Digital Material Samples for Design

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Seth Franklin Berrier

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Gary W. Meyer

August, 2012

Acknowledgements

The author wishes to acknowledge the support and rapport of his advisor, Dr. Gary Meyer, and his colleagues in the graphics department at the University of Minnesota including Dr. Baoquan Chen, Dr. Victoria Interrante, Dr. D'nardo Colucci, Dr. Daniel Keefe, Clement Shimizu, Jonathan Koniczniy, Brian Ries, Lijun Qu, Nathan Gossett, Coleman Saunders, Avery Musbach, Michael Ludwig, Jin Woo Jung, Bret Jackson, Lian Duan as well as designers Pete Wagner and Ange Tank and color scientist Dr. Patrick Chong for their input and motivation (not to mention their willingness to bounce and absorb ideas) during the development of this work and the process of my higher education. He also wishes to acknowledge the members of his PhD dissertation committee for their time and input.

We would also like to acknowledge those industry partners that supported this work including Benjamin Moore and Co., Hirschfield's, Inc., DuPont, Ford Motor Company, and the Digital Technology Center at the University of Minnesota.

For Nica, Babesy and Bee

Abstract

Designers work from example. They surround themselves with inspirational artifacts and material samples; blocks of marble, wood and metal, swatches of paint and fabrics, photographs and magazine clippings, anything that could inspire creativity or be a building block in a new product. Digital representations of these artifacts would enhance the design experience and facilitate the development of computerized collections of materials and novel appearance design tools.

In this thesis we explore techniques for capturing and rendering these material samples. We employ both the traditional rendering pipeline and the emerging body of work in image based rendering for digitizing and developing these materials. We build interactive programs for selecting paint samples and interpolating metallic paint colors and we develop a new approach to morphing surface light fields to meld these materials together into novel shapes and appearances.

The first tool allows interaction with a large set of household paint colors. It affords easy viewing of the colors on 3D objects and a novel interface for traversing and searching the large collection. Our searching tool employs the science of color perception as well as a unique 3D user interface to enable easy navigation of the large collection. We evaluate this interface with a user study and report the results.

The second tool allows for the creation of new metallic paint colors. Given two input colors it can generate an interpolated sequence of metallic colors that smoothy fit between them. This tool was used by an architect to design and exhibit a wall prototype and we detail his experience as a case study. We were able to enlist a commercial paint manufacturer to help us formulate the designed paints and construct this prototype.

The last tool provides a novel morphing algorithm for surface light fields represented as unstructured lumigraphs. The use of surface light fields allows for nearly arbitrary complexity of material appearance properties and affords scanning of real material samples into a digital representation. The morphing algorithm provides a new way of computing a 3D shape morph using shape-from-silhouette techniques. It also simultaneously morphs the appearance of the object giving a full surface light field at every point in-between.

The surface light field morphing algorithm represents a novel contribution to the field. It approaches 3D shape morphing in a new way and calculates a morph for a type of surface light field that has not been previously undertaken. Unlike other surface light field morphing algorithms, it explicitly computes light field samples at every in-between point providing detailed and physically plausible reflection data. The interface for searching color collections also represents a unique contribution that melds the science of perceptual color spaces with a unique interface that is useful for designers in their day-to-day color selection tasks.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

We are, at our core, aesthetic creatures inspired by creativity, compelled by beauty. Our culture and society champions the practice of creating art in any form, from the stacks of a library to the walls of an art gallery to the spacious hollows of a music hall; we proudly display the fruits of our aesthetic labors and seek the seeds of inspiration within them, the essential visions, sounds and constructs of creative pursuits. We are taught at a young age to appreciate the finer representative works: from the pastoral scenes of van Gogh to Picasso's twisted and grotesque image of war, from Shakespeare's comparison to a summer's day to Joyce's stately plump Buck Mulligan, from Beethoven's powerful joyous call to romanticism to Reich's minimalist repetition of a bleak "come out to show them", from the freedom of Tchaikovsky's classic, festive ballet to Kurt Jooss's constrained reality of politics in modern dance. We are made to examine these works inside and out not simply that we may discuss them with others or reference them in our dissertation, not simply that we may experience their ability to evoke emotion or even challenge conventional wisdom, not simply that we may examine their structure and achievement. Each of these works is a sample that serves to inspire further creation. By exposure to and tactile examination of representative work we find a means to inspire aesthetic creation which otherwise can be an elusive muse.

However, it is not simply the works themselves but their underlying principle components that we strive to understand. To view them only on the surface, restricted to the final product, would inspire repetition. We instead must break them down to their constituent parts both physical and metaphysical. We examine literary themes and poetic techniques, we extract melodies and motifs and identify chords, we examine the flow of paint strokes or the light and shadow at

play, we classify the individual movements or the contrast of body positions. We seek out the essence of consonance and dissonance and once identified, restructure and repurpose it for a new original work to inspire a new generation and identify new principle components of harmony.

This process of aesthetic inspiration by examination of representative component materials finds its ultimate genesis in the field of design. Design tames the creative process by coupling it with the rigorousness of engineering. By working with raw properties (sometimes even as basic as color, texture and pattern) representative materials come together to define an overall feel for a project. We only need to look at what a designer surrounds themselves with to see this. An interior decorator will collect paint chips, flooring and molding along with photographs of inspiring interiors or exteriors or even more basic materials that connect to the customer. A graphic designer may collect colors and images or even something as simple as single words or phrases that represent the client and their product. Fashion designers may collect fabrics and other raw materials along with photographs and sketches of previous designs all to develop the direction of a new design. This is particularly the case in automotive design where designers work with a variety of raw materials from upholstery and vinyl to steel and fiberglass along with photographs and images of textures and colors both static and in motion. Often these representative materials are brought together and mounted on a board to give cohesion to the collection and express an underlying harmony. This collage is called a *mood board* or *trend board*. From an engineering standpoint, the mood board establishes requirements and gives a clearer direction to the ensuing creative process. From an aesthetic standpoint it fuels and develops the inspiration necessary for creativity to occur. It is the ultimate embodiment of what the design process strives to achieve; the introduction of commercial viability into the creative process.

When a mood board is used in the design process, it can become digital by sampling the items placed on the board, or more importantly, the key characteristics of these items. These characteristics brought together represent the underlying harmony of the collage (the themes and melodies and forms that we worked so hard to extract from a finished work) that will come together to form a new product. For the simplest of representative materials (pure colors, images, basic uniform surfaces) equipment exists in the form of spectrophotometers, digital cameras, gloss meters and various industry standard appearance measurement devices all of which can

sample the key characteristics. There are also well known techniques for storing and rendering these characteristics. However, most commercial materials posses qualities that these devices cannot capture. In the presence of non-homogeneous materials and objects with complex geometry or with complex reflective properties (such as sub-surface scattering or anisotropy) commercial devices can give us nothing more than static images or primitive unrealistic renderings. Adding insult to injury, these materials are often the most sought after for inspiration as they break from the mundane, the commonplace that we experience day-to-day. This forces a designer to work with physical samples and avoid digital tools. They are left to work with ink and paper, with clay and knife, with tools that require craft beyond creativity when working in the early stages of aesthetic design.

Much of the history of computer graphics has worked to allow mimicking of real world materials and objects but little has been done to capture and digitize the objects themselves. However, recent advances in digital photography and an expanded understanding of the field nature of light has led to a new area of research that turns the the traditional rendering pipeline on its head. Where traditional rendering starts with a mathematical representation of the scene and generates an image, this new approach starts with images of that scene and produces alternative views of the scene. Consequently, it is called image based rendering or IBR.

IBR offers a promising avenue for expanding the range of digital tools available to a designer. First and foremost, it starts with real world objects capturing their appearance using digital photography and preserving even the most complex of reflectance properties. This means the physical objects and material samples that are familiar to a designer are the starting point of the digital representation. Some IBR techniques can systematically segment a set of registered images identifying all the constituent materials and fitting them to mathematical models. Other techniques rely on statistical reduction of the data into principal components for faster rendering while still preserving the key characteristics of the object. Irrespective of the approach, IBR techniques have a much wider applicability and present the potential to create a general scanning device for digitizing any material or object that could be used on a mood board or for any creative inspiration in general.

1.1 An Impetus for Digital Materials

But why pursue digital design tools at all? What becomes possible when materials are made digital? It is no small undertaking to enter the world of image based rendering where the software and hardware is entirely proprietary (even a type of haute couture for research), nor is it a small disruption to a designer to have new tools thrust into their complex and personal process.

As a basic initial observation, we note that physical samples of materials are flawed in ways that digital samples are not. They deteriorate over time, they take up space, they can be lost or stolen, they may even be dangerous to interact with. By digitizing an object or material you preserve it at a specific instant when it possessed the qualities that were desired. The digital representation is as permanent as the media that stores it. It takes up no physical space and has no sensitivity to light or the other elements beyond the space and sensitivities of the storage medium. Furthermore, the material can be altered and re-used as many times as desired without being destroyed and it is easy to share across long distances enabling broader collaboration.

A designer must store all of their sources of inspiration (such as mood boards) should they ever desire to consult or reuse them for a later project. Being limited to physical samples means they must devote a significant portion of their personal or work space to keep these sources available. Looking at mood boards in particular, they may want to include these as part of their portfolio or they may want to reuse a sample from a previous board in a new one. Transporting a real mood board may damage it and can be cumbersome. To reuse a material the designer must first locate the previous board where it was used, remove it from the board (changing or even destroying the original board in the process) and then place it on the new mood board. If the mood boards were created and stored digitally then sharing or moving them becomes effortless and reusing a representative item would be as simple as searching for it in a database. They can then add it to the new digital mood board leaving the previous one intact.

Once digitized, material samples could be maintained in a database offering state-of-the-art search and filtering options as well as all the benefits of data mining. There is additional potential to analyze this data as it becomes more populous and to make use of statistical or numerical techniques to identify patterns and further inform the design process. Augmenting the data with key characteristics, kansei adjectives or even more esoteric terms describing the object (such as ‘luxurious’ or ‘sporty’ or ‘pedestrian’) would enhance the usefulness of the

database. The extra data could then train an artificial intelligence system or pattern recognition algorithm so that it could automatically apply these terms to new materials. It could also drive a kansei recommender or some other artificial aesthetics algorithm enabling further research into the artificial subjective.

The database of digital materials could also extend beyond the mood board to the manufacturing process. Materials from the database used to make a prototype could be sent to vendors in a digital form. The vendor could then attempt to mock up a mass producible equivalent and make a bid to supply it to the manufacturer. This is a common process when mass producing a product. The digital form of our material supplied by our material scanner could enable both the manufacturer and the vendor to communicate the target properties and the raw materials back and forth until a final product (or something close to it) was agreed upon. The digital representation generated by the material scanner could also become a system for quality control. Using quantitative difference metrics the final design could be automatically compared to the original intent expressed in the mood board. In addition, products rolling off an assembly line could be compared to this digital representation automatically connecting the final product all the way back to the original design intent without human intervention.

Digitization of aesthetic design could have benefits to specific types of design as well. Take, for example, fashion design. If a process employing digital materials was used to design clothing then the digital representation of different garments (containing a detailed description of the reflective properties of the clothing) could form the basis of an enhanced database. More realistic representations of products could be offered over computer based media such as the Internet using this database. The flat panel displays that are increasingly present throughout the shopping experience could also display these enhanced representations. Sophisticated recommender systems could be employed to target a customer's tastes or to match a supplied garment that had also been scanned. All of these and a flurry of other commercial applications can be imagined and would require little or no additional work beyond utilization of the data generated by a digital aesthetic design tool.

Another specific type of aesthetic design that stands to benefit from the use of digital materials is automotive design. We can envision a similarly seamless integration of design and manufacture. It is common practice in the automotive field to build a model of the vehicle that is visually accurate down to the finest detail but that is made entirely of fake or impractical

materials (sometimes called a *real fake*). Once a real fake has been finalized and the design approved, the task of finding non-fake materials to use in the manufacturing process begins. If the materials in the real fake were scanned using our material scanner they could be communicated to potential vendors in the manor described above and help speed the process of converting the prototype into a marketable vehicle. The digital representation of the real fake material and the material offered by the vendor could also be compared for accuracy of match and used as a means of quality control once the vendor has begun providing the agreed upon material.

In this thesis we start the journey that these changes suggest with some initial steps towards using digital material samples in the design process. We hope to lay the foundation that is necessary to pave this road connecting design and image based rendering. To that end, we will demonstrate some simple tools that either make manual design tasks easier or enable manipulations that would be impossible in the real world. We begin simply, focusing on restricted classes of materials and planning our roadway. By building a tool that allows users to search through a collection of paint colors we learn what ways these digital tools can help and hinder the design process. It is a first step towards something both usable and acceptable by designers. We level the path for our roadway with a second tool that engages a specific task brought to us by an architect. Our class of materials is only slightly broader but the tool is more useful and leads to a real product designed precisely and quickly. Our final tool paves that roadway by leaping from a restricted class of materials to the most general representation currently possible, the surface light field. This thesis will be the signage on that road that will help others to find their way to all these solutions we have outlined in this section.

1.2 Overview

This thesis begins with an examination of the design process. Design has many different disciplines all of which have their own needs and practices but an essential underlying process does exist. We examine this process and the general field of design in Chapter 2. We explore the technical, objective side of the process as well as the more subjective side. Works by psychologists and engineers to codify the subjective side of human aesthetics are examined in detail and ultimately we consider the possibility of 'artificial aesthetics' as a type of artificial intelligence.

In Chapter 3 we delve into the extensive body of related work in computer science. Here we seek to carefully document the field of image based rendering focusing on work with light

fields and particularly surface light fields. Some time is also spent discussing work in morphing which will prove useful as we set out to design our own morphing technique. We discuss briefly the work in deriving 3D shape from silhouettes (a.k.a. visual hull algorithms) and we finish by looking at existing computer tools for designing appearance.

The next three chapters present the new work undertaken as part of this thesis. Three projects are discussed that start with traditional rendering techniques and limit the scope of what materials can be represented. As the work progresses we expand the capabilities and generality of the materials that can be included eventually demonstrating the power of digital tools with image based rendering techniques. At each stage we demonstrate what is possible when design becomes digital.

First, in Chapter 4 we show a set of tools that enable navigation and selection within a large set of paint colors. The colors themselves cannot be changed; they are fixed formulas specified by the manufacturer. But selecting the right one among so many choices is a difficult task. Our suite of tools allows for the design of a harmonious color palette used in a decorating scheme. We demonstrate these digital tools that aid in this process in ways that are not possible or are extremely cumbersome in the physical world and evaluate their usefulness with a user study.

Next, in Chapter 5 we expand our material representation to include colors that shift as you change your viewing angle. These are often referred to as metallic or pearlescent colors and are becoming more readily available to designers particularly in household paints and automotive finishes. With a digital representation of a metallic color we develop a tool that allows one to smoothly generate a series of colors between two endpoints. This is a type of interpolation and allows one to quickly design and realize many metallic colors which would be difficult to develop individually. This work was carried out in conjunction with an architect who used the tools to design a physical prototype with shifting metallic colors. We were able to evaluate its effectiveness and demonstrate how new digital tools for appearance design make difficult real world tasks easy and more accurate.

Finally, in Chapter 6 we leap from the world of paint finishes to a completely generalized material representation by employing surface light field rendering. This IBR technique allows for the capture of almost any type of material regardless of its reflectance properties. Using this representation we develop a new algorithm for morphing between two different surface light fields. Our new algorithm allows the blending of appearance properties of several target objects into a new, unique material. A creative idea that previously could only be suggested

by placing two materials next to each other becomes a digital reality with this technique. The designer can not only interact with this new blended material digitally but can generate accurate reflectance data that can inform the engineering side of the design process all the way through to a marketable product. This algorithm approaches morphing of 3D shape in a new way using a shape-from-silhouette technique reducing a complex 3D problem into a set of simpler 2D problems. It also works with a particular representation of a surface light field for which no previous algorithm for morphing exists.

Chapter 7 summarizes the efforts of this thesis. Here, we look to the future and movements in industry and education that could enable the average consumer to leverage the power of IBR. We see smaller and easier devices that emulate the generality of the mammoth hardware that currently exists. We see an industry trying to get started that takes photographs up to the power of a full light field. We see researchers starting to commercialize their ideas and to bring their tools to the mass market. All of this makes the world of image based rendering a dynamic and forward looking field that can revolutionize the work of any discipline that has the foresight to pursue it now. Design is uniquely positioned to benefit from this revolution.

Chapter 2

The Design Process

In this chapter we will establish what is meant by 'design' and more specifically what is meant by the 'design process'. Design is "the act of conceiving and planning the structure and parameter values of a system, device, process, or work of art." [Parker, 2007] There is a tendency to think of design as being a highly creative process as in graphic design or fashion design but it is more than just an applied art. It is engineering and architecture; it is applied science. To begin, these two aspects of design (the applied science and applied art) will be clearly delineated and their practice will be established. Then a process that appeals to both aspects will be developed and described. This process (particularly as related to applied art) will be the 'design process' referred to throughout the remainder of this work. This chapter concludes with a discussion of the study of the cognitive process of the designer and its relation to automation of the design process.

2.1 Technical vs. Industrial Design

To begin we focus on a statement by Ashby and Johnson:

> "Technical and industrial design are the essential, complimentary, parts of the design of *anything*." [Ashby and Johnson, 2002]

This is the first main division of the design process; into technical, engineering tasks and industrial, aesthetic tasks (applied science vs. applied art). *Technical design* deals with the definition of technical limits (such as mechanical, thermal and structural constraints) and appeals

heavily to simulation and scientific modeling. It concerns itself with the physical properties of materials such as strength, density, weight and malleability as well as more practical concerns such as cost and availability. Technical design appeals to other practical concerns such as compatibility with other designs, resistance to the elements and the ability to cheaply form raw materials into the components needed in the design. *Industrial design* deals with aesthetics and usability and appeals to human emotions and the sense of artistic content. It concerns itself with the perceptual aspects of the material such as visual, tactile and acoustic qualities but also with practical concerns such as adherence to market trends, color forecasts and demographic considerations. Industrial design strongly influences the form of the design (from the overall shape to individual lines), the appearance of the design (from color to texture, pattern and gloss) as well as the 'sound' of the design (the pitch it produces when operated or struck and its persistence). These two separate parts of the design process must be considered concurrently as decisions made appealing to one aspect will affect the other. For example, if a particular material is chosen for its ease of manufacture and availability (a technical aspect) it may place constraints on what lines and shapes are achievable for the form of the design (an industrial aspect). Conversely, if a particular pitch becomes a requirement of the design so that it harmonizes with other objects (an industrial aspect) this will limit the materials that can be used for manufacture as not all materials are capable of resonating at all pitches (a technical aspect).

The Technical Design Process

The vast majority of technical design decisions are engineering decisions, making it the more studied and well defined of the two aspects of design. Pahl et al. [2007] have compiled a text that systematizes almost all of the considerations of technical design boiling them down to precise task flows and equations that will ensure the success of the end design specifications. In particular they identify the *flow of work* during the process of designing. The key phases of this flow of work are:

1. Planning and task clarification
2. Conceptual design
3. Embodiment design
4. Detail design

The first phase, *planning and task clarification*, involves an analysis of the market and company situation, selection of product ideas and the formulation of the product proposal. It results in the design specifications listing key requirements for the product being developed. The second phase, *conceptual design*, will result in the principle solution by clearly identifying the essential problems that must be overcome and establishing the product's functional structures. The principle solution *is* the overall concept for the end product being designed. Some common examples of principle solutions include schematics, circuit diagrams, flow charts, line sketches and many other possibilities. The *embodiment design* phase will produce the design layout. It involves an iterative refinement of the structure of the product and key components such as material and form. The final layout will include parts lists and production/assembly documents. The final phase, *detail design*, finalizes the product documentation with all of the technical details needed for the product's production, assembly, transport and operation. It irons out the key details needed to ensure the final design will be realizable in the existing industrial infrastructure that will ultimately produce it. Each of these phases may overlap and the whole process may iterate between phases so that the design can remain dynamic and improvements can be introduced at any point in the process.

The Industrial Design Process

Industrial design is a more subjective discipline than technical design. It can involve a highly detailed, rigorous process just like technical design, or a very abstract, artistic processes that is deeply qualitative and creative in nature. Ashby and Johnson pay particular attention to industrial design and how it impacts material selection [Ashby and Johnson, 2002]. They describe the industrial design process using the same phases as Pahl et al. [2007]. Starting with the second phase, Ashby and Johnson extend the process and describe what industrial tasks should be taking place concurrent with these technical phases. In *conceptual design*, the industrial designer is seeking the desired aesthetics by considering the widest range of ideas. The concept's perceptual requirements are key and augment the design specifications produced. The third phase, *embodiment design*, is principally concerned with exploring and expanding the ideas developed in the previous phase. Alternative forms, colors and textures are all considered and the aesthetic requirements from the concept are worked into the principle solution. Note that Ashby and Johnson refer to this phase as 'Development' but acknowledge that it comes from the work flow developed by Pahl and Beitz. Finally, in the *detail design* phase the form is visualized using

1. **Planning and task clarification**

Technical Design	Industrial Design
- Market and company analysis	- Develop aesthetic requirements
- Idea selection	
- Formulation of product proposal	

Results in: Design Specifications

2. **Conceptual design**

Technical Design	Industrial Design
- Identify essential problems	- Seek desired aesthetics
- Establish functional structures	- Consider widest range of ideas

Results in: Principle Solution

3. **Embodiment design**

Technical Design	Industrial Design
- Refinement of structure	- Consideration of alternative forms,
- Refinement of key components	colors and textures
	- Expand key ideas from last stage
	- Explore sources of inspiration

Results in: Design Layout

4. **Detail design**

Technical Design	Industrial Design
- finalization of:	- Visualization of form
· Technical details	- Use of Computer Aided Industrial
· Assembly instructions	Design Tools (CAID)
· Transportation and Operation	- Use of physical prototypes

Results in: Product Specifications

Table 2.1: The flow of work with technical design and industrial design details. Based in part on information from [Pahl et al., 2007] and [Ashby and Johnson, 2002]

Computer Aided Industrial Design tools (CAID) as well as physical prototypes that will influence the final product documentation and help finalize the production, assembly and use of the product. It is worth noting that industrial design is also concerned with the product's disposal and aims to minimize the environmental impact at all phases of this process. Consequently disposal documentation is also part of the final phase of industrial design.

The full design process is summarized in Table 2.1 with details about technical and industrial aspects of each phase. This is the design process we will consider throughout this work.

2.2 Aesthetic Design and Cognition

The act of designing and its associated process is a 'cognitive' process. While engineering has brought much of what was in the mind of the designer out into concrete algorithms and processes there is still much in the designer's mind to be explored and studied. In particular, much of industrial design is a creative, subjective process where decisions are made based on the designer's taste and feelings; decisions of aesthetics and emotions, not of functionality and usability. We further subdivide industrial design by separating out these subjective decisions and will refer to them as *aesthetic design*. While cognitive science and its computer science counterpart artificial intelligence are prepared to model and automate the objective parts of the cognitive design process these subjective aesthetic decisions have been avoided. This leads us to the question, how do we model and formalize these creative, subjective thought processes. How do we apply artificial intelligence to aesthetics? These questions are pertinant to the development of an automated design process as they attempt to make quantitative the otherwise subjective processes of aesthetic design. In this section we will review the state of AI applied to aesthetic design (referred to here as *artificial aesthetics*) paying close attention to its relation to cognitive science and engineering. Our purpose is to catalog attempts at automating the underlying theory of aesthetics in design and to describe the aesthetic design process in a more objective and automateable way.

2.2.1 Artificial Aesthetics

As is implied by the phrase 'artificial aesthetics', there is a natural connection between automating the cognitive process of the designer and the field of artificial intelligence. Generally speaking, AI treats the mind as a black-box. On one end we know what input is available (or

at least part of the input) and we know what the resulting output will be but we do not know the function inside the mind that maps the input to the output. AI seeks these mappings (or something equivalent). It proposes various techniques based on simulation, numerical methods, statistics, probability and logic that for certain tasks behave remarkably similar to the mind. If we adhere to a physicalist view of the mind (namely that the totality of what we call the mind is reducible to physical attributes such as neurons firing) It seems reasonable to attempt the same with subjective thought processes. If we then assume the techniques used in AI are in any way similar to the actual mappings that are undertaken by the mind then perhaps they can also predict and/or model subjective, aesthetic decisions. On the other hand, it is more difficult to identify what influences an aesthetic decision (e.g. the input to the aesthetic mapping in the mind) and this input is unquestionably different for every mind. Perhaps due to this, or perhaps due to the difficulty of understanding the subjective compared with the objective, cognitive science and AI have avoided modeling subjective thought processes. Regardless, designers and engineers have shown an interest in this direction of research and over the last 20 years have started to develop their own systematic approaches to artificial aesthetics some of which borrow from or are remarkably similar to artificial intelligence.

Both Suh [1990, 2001] and Lin and Zhang [2006] offer a more mathematical definition of design and the design process based on cognition. They define design as "a mapping from the requirement space to the function space" subject to constraints. They define the design process as "a set of activities ... to proceed with the design mapping" [Lin and Zhang, 2006]. In this form, design itself becomes a black box much like the mind in AI. Within this black box is a mapping from the requirements specifications defined early in the design process to functional components such as those defined in blueprints or product specifications. The mind of the designer must perform this functional transformation using an internal mapping. Several researchers have attempted to make this mapping external and explicit.

2.2.2 Kansei Engineering

The word 'Kansei' comes from the Japanese language and refers to the perceptual and emotional aspects of the mind (including aesthetic aspects). *Kansei engineering* (abbreviated KE) is the incorporation of perceptions and feelings into the design process of a product. It's importance was first suggested in the 1950's by Jujiro Matsuda, president of the company that would become the auto manufacturer Mazda [Shiizuka and Watada, 2006b]. These concepts were later

championed and developed into a concrete science by Mitsuo Nagamachi in the 80's and 90's [Nagamachi, 1995]. Nagamachi used lists of words (usually adjectives) to identify aesthetic requirements and then used questionnaires to develop a statistical correlation between these adjectives and functional aesthetic components. Other researchers have taken KE beyond this questionnaire approach employing techniques from artificial intelligence such as fuzzy logic and neural networks [Shiizuka and Watada, 2006a]. More recently, Nagamachi has proposed the employment of rough set theory to incorporate uncertainty and vagueness into the extraction of rules for the design mapping in KE [Nagamachi, 2006].

2.2.3 Other Approaches to Aesthetic Design

While recent literature abounds with references to KE it is not the only approach to aesthetic engineering, and for good reason. While seeking a unified design process that allows for aesthetic requirements (in addition to functionality and usability requirements) Lin and Zhang critiqued KE and identified several shortcomings [Lin and Zhang, 2006]. They found that KE overlooks the problem of aesthetic requirements conflicting with other product requirements and provides no model for reconciling these conflicts. They also found that KE lacks a thorough evaluation of the final product specifications to ensure that they meet the original requirement specifications. As an alternative they cite the work of Reich [1993] and van Breemen et al. [1998] that separates aesthetics into an antiquated 'rationalist' approach (commonly attributed to the Baroque philosopher Baumgarten [1954, 1961]) and a more modern 'romanticist' approach [Hegel, 1975; Kant, 1952]. These two views propose very different models of evaluating the resulting design. Ultimately, Lin and Zhang propose their own system of aesthetic design that is a synthesis of the ideas of van Breemen et al. and their own ideas. Their system uses both rationalist and romanticist aspects of aesthetics and, in their own judgement, reconciles the shortcomings of KE.

2.3 Making Aesthetic Design Digital

As technology advances, even the simplest of products and processes become more and more complicated and the need for digital design tools and automation of the design process increases. Already there are a significant number of tools that help with certain phases of this process. The last phase, detail design, often uses classical 3D modeling tools (CAD and CAID applications)

both for the technical prototype development and for aesthetic drafting. In contrast, the first and second phases remain largely manual with basic organizational and communication tools helping but not truly making these stages digital. These two phases benefit from free thinking and exploration and may suffer if bogged down by restrictive tools. We choose, instead, to focus on the third phase, embodiment design, where real potential exists for the creation of digital tools (particularly related to industrial design) and where the design process itself has reached a point beyond pure brainstorming and free-thinking. The industrial aspects of embodiment design are chiefly concerned with developing the perceptual requirements into the product solution. It is here that the designer prunes down the collection of 50 – 100 potential materials to only 2 or 3 [Ashby and Johnson, 2002]. Ashby and Johnson make reference to two specific sources of inspiration that our work has the potential to make digital: material collections and mood boards.

A *material collection* is a set of physical samples of materials (the larger the better) that a designer can examine as potential candidates for their design. These collections should be more than just raw manufacturing materials but should be customized with a variety of specialized finishes and treatments to affect their appearance. It is not the physical attributes that a designer wants to examine. Those can be looked up in a design manual. They should be kept for their aesthetic qualities and maintained in a way that the designer can pick them up and interact with them. Ashby and Johnson stress the importance of material collections noting that one of their major shortcomings is the difficulty in keeping them up-to-date. New materials are constantly being made available and any material collection will have trouble keeping stocked with the latest creations. Consequently there is a need for a database of new materials that is connected with a store of aesthetic attributes and qualities that are vital to the design process [Ashby and Johnson, 2002]. A digital material database would be an ideal candidate to reconcile these problems while still providing a designer with the feedback they need to make an aesthetic decision. While the tactile feedback of holding an object is lost in a digital database the other advantages of physical samples (the ability to examine the material from many angles under different lighting conditions and in different environments) would be preserved. Furthermore, it would be very easy to update with the latest materials and would be easy to make available to customers around the world. This need for an up-to-date database of materials is clearly identified by Ashby and Johnson and is one that our work has the potential to fulfill.

A *mood board* (sometimes called a *trend board* [Bouchard et al., 2008]) is “a personalized,

project focused image collection enhanced by a collection of material samples" [Ashby and Johnson, 2002] (see Figure 2.1). While work is already underway to build a digital mood board [Bouchard et al., 2008; Shimizu and Meyer, 2010] they are currently based around static images and Ashby and Johnson are careful to include material samples in their description of the mood board. Aesthetically, a material's visual features (reflective properties) are key to the design process and these features are not static across different viewing and lighting angles. A single still photograph cannot account for this and as such a more sophisticated representation of material is necessary to enhance this part of Shimizu and Meyer's digital mood board and Bouchard et al.'s TRENDS project. An initial goal of this work will be to add functionality to Shimizu and Meyer's software that re-creates the complex reflective properties present on any material that a designer may wish to place on these boards. We also foresee the digital representation of a material enabling the creation of a tool that helps a designer enhance their mood board with new materials found in a database that fit with the underlying mood expressed in the board. Here we see a clearly identified tool used in the design process that our work has the potential to enhance and provide in a digital form.

Figure 2.1: Top: a mood board by designer Eva Artinger as part of her 'Relaxing Plant' project (retrieved from [Artinger, 2012]). Bottom: a screenshot of Shimizu's Digital Mood Board software [Shimizu and Meyer, 2010] showing a collection of inspiring photographs.

Chapter 3

Background and Related Work

In this chapter we examine the body of related work that informs the research in this thesis. The majority of this chapter will be devoted to previous work in computer graphics concluding with a discussion of related work in computer aided aesthetic design.

3.1 Related Work in Computer Graphics

In this section we examine previous publications related to the capture of reflective properties in computer graphics. Presently there is no standard approach for making complex material appearance available in a digital setting, so we begin by examining research in the literature that could offer such a standard representation as well as a standard photographic device or method for building this representation. They must be based around images acquired with a digital camera and must produce renderings of the target material from unique angles under lighting conditions different from those in use during acquisition. These techniques are often referred to in the industry as *image-based rendering* techniques. We will briefly discuss the use of a digital camera to acquire reflectance properties in general. Then we will identify three main approaches to this problem. First, we examine techniques that treat the object being acquired as a space occupied by a field of light. The general mathematics of fields are employed to render an object from unique angles possibly under unique lighting conditions without knowing the object's geometry. Second, we will look at techniques based on a spatially varying form of the BRDF known as a BTF that maps reflection properties to the surface of an object similar to the way a texture maps an image onto an object. Lastly, we examine a more recent technique known

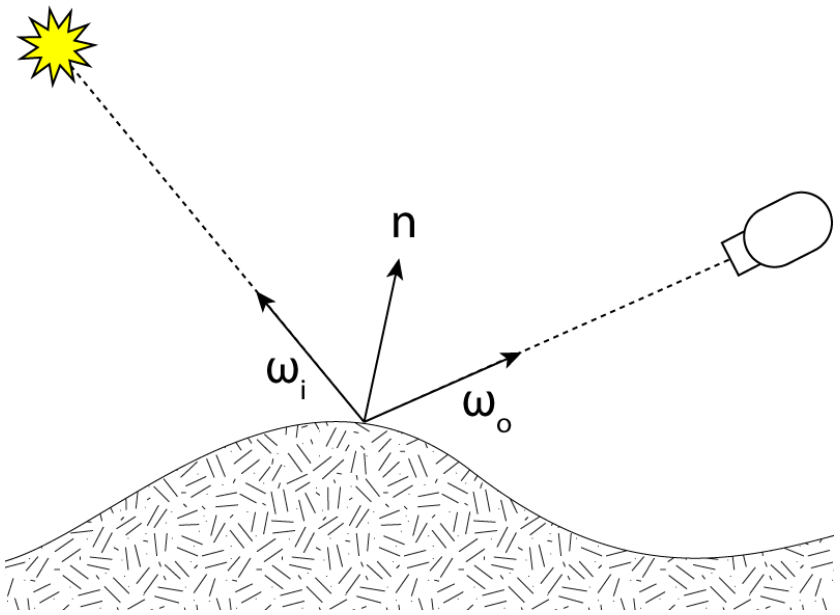


Figure 3.1: A diagram of the bidirectional reflection distribution function (BRDF) usually notated as $f_r(\omega_i, \omega_o)$. Note that both ω_i and ω_o have a minimum of two dimensions when described in three-space (usually elevation and azimuth angles).

as inverse rendering. This technique aims to take images of a scene with some unknown properties that would normally be present when rendering and to derive those properties from these images. The derived information can then be used to generate new images through the usual rendering process. We conclude by discussing some alternative reflection models proposed in the literature that are relevant to image-based rendering.

3.1.1 Photographic Gonioreflectometer

The most general description of how light reflects off a particular surface is the bidirectional reflection distribution function (or BRDF) (see Figure 3.1). A BRDF takes as input an incoming direction and an outgoing direction belonging to the hemisphere of directions above a single point on the surface of a sample. It returns the amount of light that would be reflected for those two directions. Much work in computer graphics has attempted to create analytical versions of the BRDF that can be computed and adjusted at the time of rendering. Some of these analytical

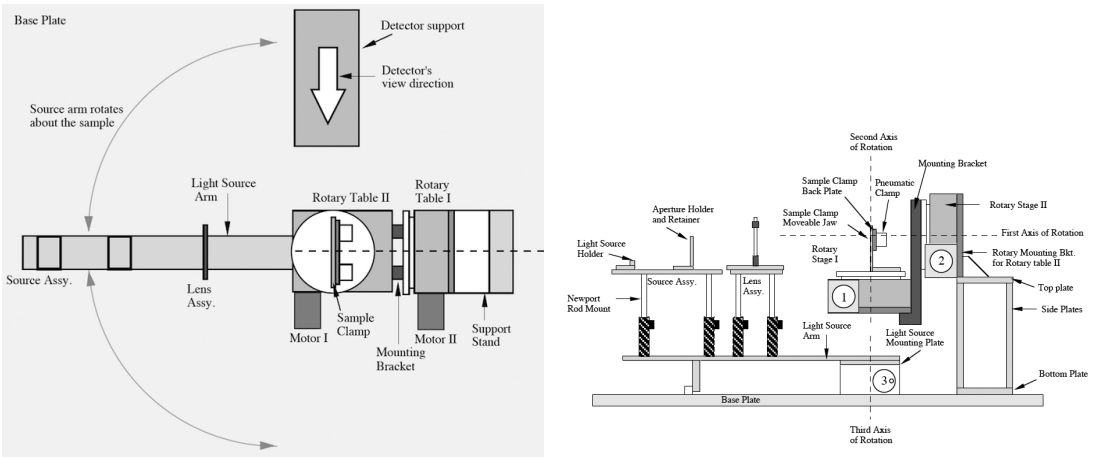


Figure 3.2: A basic gonioreflectometer from Kodak. From [Foo, 1997].

models exploit our understanding of light and optics to create physically plausible reflections while others focus on mathematical simplicity to achieve faster rendering times. It is also possible to directly measure a BRDF. Many devices have been proposed that possess a controllable light source and light sensor such as can be positioned in any direction in the hemisphere around the surface being sampled. The light source position coincides with the incoming parameter for the BRDF and the light sensor coincides with the outoing parameter. These devices are collectively referred to as gonioreflectometers. An excellent summary of a basic gonioreflectometer can be found in [Foo, 1997] (see Figure 3.2). When using a basic gonioreflectometer, it is necessary to systematically sample the entire hemisphere of outgoing directions for every possible incoming direction to fully characterize the BRDF. While a certain understanding of the specular and diffuse natures of reflections can help guide the sampling process this is still a daunting amount of data to sample. Furthermore, carefully calibrated light sources, sensors and the equipment to precisely position them around a hemisphere can be expensive and difficult to use. There are few off-the-shelf gonioreflectometers and they are very expensive. It was the effort of researchers to simplify this device and the cumbersome sampling process that led to the use of a digital camera in place of the light sensor and gave birth to the photographic gonioreflectometer.

The goal of a photographic gonioreflectometer is the same as the goal of basic gonioreflectometer, namely to provide measurements of the underlying BRDF describing how light reflects off the material's surface. After some processing, this data can be applied to other surfaces and

used as a basis for rendering. Unlike the sensor in a basic gonioreflectometer, a camera can capture data for many points in parallel. This is due to the fact that the internal sensor in a camera (usually a charge-coupled device, or CCD) is actually a 2D array of smaller sensors that coincide with the pixels of a photograph. Each pixel in the image produced by the camera's CCD represents a sampling of a different point on the surface of the object. This parallel capture of many points makes sampling the spatial variance present in real-world objects much more feasible. Regardless, there are fundamental differences in the information that is produced by a digital camera's sensor and the data measured by a single light sensor. In particular, the digital camera measures radiant energy (the integration of flux across an aperture) while a light sensor measures radiant flux. A digital camera also processes the data using techniques such as interpolation, white balancing and gamma correction that modify the radiant energy values to perceptual values ideal for human viewing but not for radiometric measurements. Dorsey et al. [2008] discuss these issues in more detail.

Two advances in the field of digital photography have helped alleviate these differences. First is the development of high dynamic range photographic techniques by Madden [1993], Mann and Picard [1995] and Debevec and Malik [1997]. The refined method developed by Debevec and Malik uses a series of digital images of a single scene taken from the same position and orientation but at different exposure levels (see Figure 3.3). These exposures are used to compute a camera response curve capable of describing the non-linearity present in CCD's and introduced by the image development process. More importantly, this response curve can be used to reverse this internal processing and return radiometrically calibrated values from the perceptual values generated by the camera. The result is a high dynamic range (HDR) photograph that stores the actual radiance values present at each pixel rather than human viewable RGB values. Radiance values require higher precision and range than human viewable RGB values and as such an HDR image is much larger and requires a different format than a typical digital photograph. The second advancement in digital photography is the availability of the raw values reported by the CCD elements rather than the result of the camera's internal processing and balancing of the image. These raw values still only cover a small portion of the full dynamic range of a scene and may exhibit some non-linearity due to some processing still being performed or imprecision and noise in the electronics. As such, a response curve and technique for combining exposures is still necessary but the results produced are more accurate.

Figure 3.3: A sequence of images at different exposure levels used to construct a high dynamic range image. From [Debevec and Malik, 1997].

3.1.2 Field Theory of Light

Most of computer graphics exploits the particle nature of light allowing a geometric approach to rendering an image. By tracing paths of light particles through a scene composed of objects, lights and observers, an approximation of the observable image can be generated. However, the widely acknowledged dual nature of light as a particle and a wave leads to another rendering method. By considering light as a wave the scene can be treated as a contiguous field of electromagnetic radiation where the behavior of light at each point in the three-dimensional space is described for every possible traversal direction. The mathematics of this field theory of light yield a very different approach to generation of an image. This technique is often referred to as *light field rendering*.

Theory

The treatment of light as a field of electromagnetic radiation is as old as the wave theory of light. One of the most important early sources, and the source that is often credited with coining the phrase *light field* is a book by Gershun [1939]. It presents the theoretical foundation of the light field defining it as a function of five dimensions (three for the spatial location and two for a direction through that location). Gershun describes the field as a 3D space of vectors where the length of each vector encodes the radiant flux for that direction at every point in the space (see Figure 3.4). Computing these vectors is as simple as the addition of the vectors

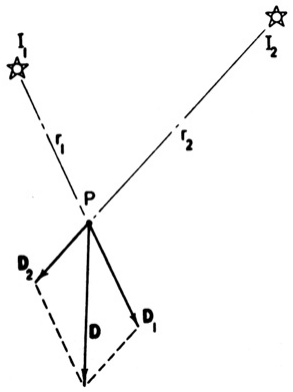


Figure 3.4: Gershun uses a sum of vectors to encode the radiance at each point in space. From [Gershun, 1939].

describing the radiant emittance from the light sources in the scene. This theory was developed again by Adelson and Bergen [1991] in the context of human vision. Adelson and Bergen refer to the light field as the plenoptic function but define it in much the same way as a traditional five-dimensional light field.

Gershun's book was translated to English by the lighting engineer Parry Moon who refined Gershun's theory throughout his career eventually presenting a modern perspective on what he termed the photic field in his own book [Moon and Spencer, 1981]. Part of Moon and Spencer's contribution to the field theory of light was the realization that when the space occupied by the field is empty then there is no change in the radiance along a single direction from point to point. This allows the elimination of a dimension and a reparameterization of the five-dimensional light field into a four-dimensional function. The five-dimensional function can be projected onto different surfaces (such as two planes or a unit sphere) where pairs of points on the surfaces can be described with only four dimensions (see Figure 3.5).

For the remainder of this section we will distinguish between the original five-dimensional light field and the reparameterized four-dimensional version. We will refer to the 5D form using Adelson and Bergen's term *plenoptic function* and will only use the term *light field* to refer to the 4D form regardless of how it is parameterized.

4D Light Field Rendering

One of the earliest examples of image based rendering in practice comes from the Apple Quicktime VR system (QTVR). Today, virtual tours are pervasive on the internet and object movies are often used to allow a user to rotate and examine a product for sale on the Internet. Both of these techniques originated in the QTVR system. The object movie in particular has a basis in the field theory of light although very primitive compared to full light field rendering systems. Work by Chen in developing the QTVR system [Chen, 1995; Chen and Williams, 1993] along with similar work by McMillan and Bishop [1995] laid the groundwork for more advanced techniques that would follow. Panoramic virtual tours offer a view from a fixed point inside the light field looking out into the world. QTVR object movies offer an external view of an object (fixed or moving over time) however the viewer's position (or the position of the object), the incident light and the reflective properties of the object are all fixed. Despite the primitive nature of these techniques they are perhaps the most pervasively used examples of image based rendering today due in no small part to commercially available equipment for automating their creation. QTVR object movies often take images at 10 degree intervals in both horizontal and vertical dimensions. As such, only $36 \times 18 = 648$ images are required to create a full object movie and in practice much fewer can be used for partial object movies. Using modern automated gantrys and turntables all of these images can be acquired in less than an hour (sometimes only a few minutes) without user intervention.

Levoy and Hanrahan were the first to acknowledge the work of Gershun and Moon [Levoy and Hanrahan, 1996] and develop a light field approach to rendering from the ground up. From

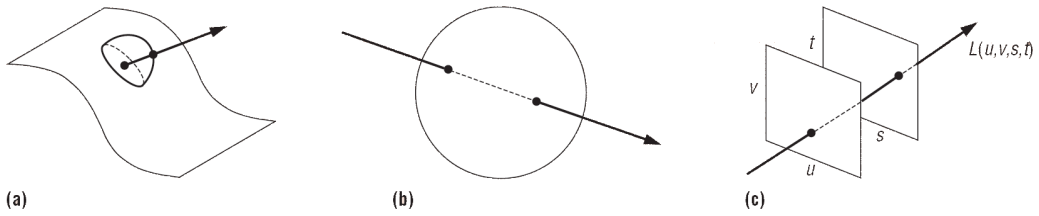


Figure 3.5: Different parameterizations of the plenoptic function (as suggested by Marc Levoy) to reduce the dimensionality to four dimensions. a) Points on a surface and directions leaving each point. b) Pairs of points on the surface of a sphere. c) Pairs of points on two planes in any position. From [Levoy, 2006].

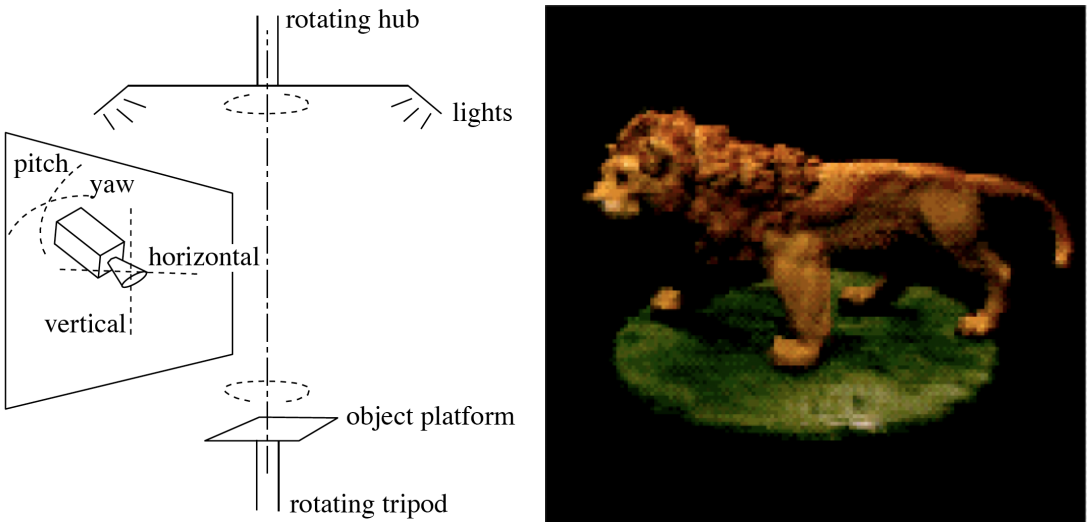


Figure 3.6: Setup and example rendering of work by Levoy and Hanrahan. From [Levoy and Hanrahan, 1996].

the beginning, their technique was centered around capturing and preserving real-world objects without having to know the object's geometry. Their technique improves upon the purely image based techniques of Chen and McMillan and Bishop by treating the set of images as 2D planar samples of the 4D light field. Sampled densely enough, the underlying light field (which was represented using the two plane parameterization, see Figure 3.5c) could be approximated using interpolation and directly rendered. Sampling the light field adequately proved tedious and time-consuming. In one example, 2048 images were taken over 4 hours time generating 402MB of data for a single figurine. The size of the recreated light field also presented a problem as it was too large to process and render in real-time. Perhaps the most significant contribution of their work was a technique for efficiently storing (and subsequently rendering) the massive amount of information present in these fields. With their technique the previously mentioned example was reduced from 402MB to only 3.4MB (a compression ratio of 118:1). Ultimately, Levoy and Hanrahan's technique involves a very complex and difficult capture process and yielded only mediocre results at the time. Repeated today with modern photographic equipment and faster computers the quality would no doubt improve but the capture process would remain prohibitive.

At the same time as Levoy and Hanrahan, Gortler et al. developed an approach to light field rendering similar to Levoy and Hanrahan calling their version of the light field a Lumigraph

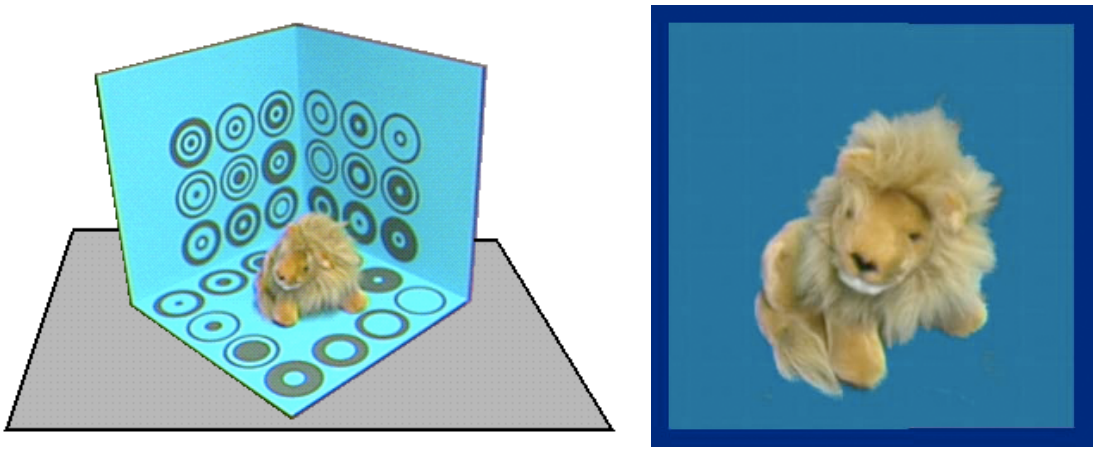


Figure 3.7: Setup and example rendering of work by Gortler et al. From [Gortler et al., 1996].

[Gortler et al., 1996]. Like Levoy and Hanrahan the light field was parameterized using two parallel planes however, a full Lumigraph has six such sets of planes forming a complete cube around the area of interest. Gortler et al. took a slightly different approach to capturing the light field. They still employed a camera but instead of using a computer controlled gantry system a special stage was created with registration markings. The object to be captured was placed in the stage and a user would take pictures of the stage holding the camera in hand. A computer program would monitor the capture process telling the user where more samples were needed. The equipment required is much cheaper and the process can be quicker but more processing of the data after capturing is necessary to properly register all images and reconstruct the light field and a human subject must operate the camera throughout the capture process. Gortler et al. also augmented the light field data with rough geometry called the visual hull that is computed using image based volumetric techniques. The visual hull is a useful approximation to the full geometry of an object and will be discussed in more detail later. This rough geometry helps guide the interpolating of the images into the light field improving the accuracy by adding approximate depth information to each pixel. The Lumigraph techniques simplify the capture process over Levoy and Hanrahan at the expense of more image processing and human interaction. However, it still requires a large number of samples making for a very lengthy capture time and difficulty storing and rendering the data.

3.1.3 Surface/Texture Based Reflectivity

Texturing is a technique in computer graphics where a two-dimensional array of colors is mapped onto the surface of an object. It allows the placement of a different color at each point on the surface without having to compute a complex reflection model. The two-dimensional array of colors is stored in an image which is often derived from photographs or synthesized using another rendering engine. The main benefit of this approach is that it allows an approximation of full spatial variance on a surface avoiding the limitation of modeling only homogeneous materials. However, this spatial variance only describes what color is produced from a fixed viewing angle under fixed lighting conditions. These values can be combined with analytical BRDF models to simulate the reflection under varying lighting conditions and viewing angles but the parameters of the BRDF will not vary across the surface, only the base color provided by the texture will vary (usually the diffuse color parameter). To improve texturing and extend its applicability to physically based BRDFs several researchers have modified the array of values in a texture to include either parameters for a reflection model or sampled BRDF data at each element instead of just a single color. Every point on the surface can compute a completely different reflectance value that accounts for the specific lighting and viewing direction in effect. Even beyond this, some techniques have stored the information usually present in a light field at points on the surface of the object. As with a light field, these values can be directly measured from the real world making this an ideal technique for scanning artifacts or materials that would otherwise be difficult to preserve.

Bidirectional Texture Function Rendering

The BRDF describes the reflection properties for a single, homogeneous material. It does not posses the flexibility to describe surfaces that are composed of different sub-materials (referred to as spatially varying surfaces). For these types of surfaces the BRDF is not homogenous. It varies along with the different materials that make up the object or with the small-scale local geometry that would be difficult or impossible to model. To accommodate this a generalization of the BRDF that accounts for this spatial variance was developed by Dana et al. [1999]. They call this generalization the bidirectional texture function (or BTF). Just as a standard texture defines a color for each coordinate in two dimensions, the BTF adds these two dimensions to the four dimensions of the BRDF resulting in a six-dimensional function that describes a

Figure 3.8: Setup and example rendering of work by Dana et al. From [Dana et al., 1999].

different BRDF for every point on the surface of the object.

Dana et al. presented their technique as an extension to texturing that does for a texture what the BRDF does for a single reflectance value; namely, it allows you to vary lighting and viewing conditions while accurately computing the behavior of light reflection off the surface. As was emphasized by Dana et al., the difficulty with their representation is in the creation of data to fit this model. Consequently the work discusses their device created to sample a real world BTF and store the information for rendering in great detail (see Figure 3.8). They created a system composed of a video camera and gantry for viewing and lighting a sample from 11 different directions. Low resolution frames were captured using the video camera and stored in a large database that is freely available on the Internet. On the other hand, very little attention is paid to efficient storage and rendering of a BTF. Later work by Dana and Nayar [1998, 1999] led to empirical models of 3D textures useful in computer vision and work by other researchers [Tong et al., 2002; Vasilescu and Terzopoulos, 2003] has led to advanced techniques for compressing BTF data and rendering it in an efficient manner. Ultimately, BTF rendering takes the already difficult to sample BRDF and adds two dimensions. However, these extra dimensions do not inherently complicate the acquisition process because a camera is the central light sensor in their gonioreflectometer which samples many different points on the surface in parallel. The number of images and the length of the sample process is therefore similar to pure light field rendering techniques. It is the processing and storage of BTF data that makes it potentially more difficult to work with than a light field as well as the fact that it describes a two-dimensional surface rather than a 3D spatial area. As with Levoy and Hanrahan's and Gortler et al.'s work there is no inherent geometry in the data. Instead the resulting BTF is intended to be wrapped

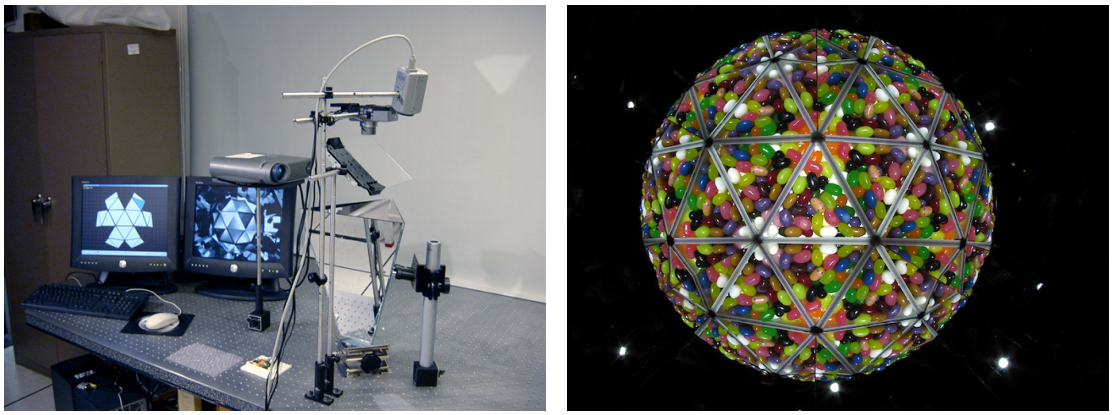


Figure 3.9: Setup and sample kaleidoscope image from work by Han and Perlin. From [Han and Perlin, 2003].

around a geometric model created or acquired in a separate process. Its original intended use is similar to the intended use of textures. BTFs would model a diverse set of generic materials that can be applied to complex objects by mapping the materials to different portions of a geometric model.

Recently a clever BTF capture technique was proposed by Han and Perlin [2003] that captures multiple viewing angles of a generic surface material in a single image using a kaleidoscopic setup (see Figure 3.9). They developed a system where several mirrors were aligned much like a kaleidoscope and illuminated by use of a projector redirected with a beam splitter allowing the light-source and camera to be co-located (optically speaking). The kaleidoscope enabled the camera to observe the object from multiple viewpoints (sometimes in the hundreds) in a single image. The ability to capture many viewpoints in parallel at the expense of each image being lower resolution could greatly speed up the process of sampling a real-world object. Furthermore, the density of sensor arrays in digital cameras seems to be increasing at a very rapid rate which this technique is well designed to leverage. It fit well with the original purpose of BTFs to acquire a diverse set of generic materials and use these in place of basic textures. However, the inherent loss of resolution in each viewing angle would necessitate capturing of only a very small spatial area and make this technique ill-suited for capturing an entire object. It is also necessary to implement a complex image processing step that separates and un-warps each viewing angle from each image.

Instead of attempting to store the physically sampled BRDF data at each point in a texture another approach is to store parameters for an analytical or numerical model of a BRDF. This approach was used by Malzbender et al. [2001] to create a reflectance representation called polynomial texture maps (or PTM). In this approach, again a photographic capture device is used to sample a surface from many different lighting angles in parallel but only one viewing angle. The data collected is fitted to a polynomial representation and the polynomial coefficients are stored in a texture (instead of R, G and B components). Since the viewing angle is fixed for a PTM the camera in the sampling device can also be fixed saving much complexity in the device and sampling time. Malzbender et al.'s device uses a fixed overhead camera integrated into a hemispherical dome with many different light sources attached to it (see Figure 3.10). Lights are turned on and off and multiple images are taken to sample the BRDF at each point on the surface inside the dome. The resulting data is fitted to a polynomial basis affording a dramatic reduction in the amount of data. The PTM is viewable from only a single angle but can be viewed under different lighting environments. The polynomial representation of the data is best suited for diffuse surfaces and will not accurately encode a specular highlight. This limitation could be overcome by using a different underlying BRDF model and storing different parameters in the texture but still employing the same capture device. The simplicity and automatic nature of Malzbender et al.'s technique gives it an advantage over both BTFs and light field techniques but its limitation to a single viewing angle constrains its applicability. Relatively flat objects can be represented quite well with a PTM but objects with more depth and geometric detail will not fare well in this representation.

A state-of-the-art review of BTF related research and a thorough discussion of capturing, storing, compressing and rendering this form of reflectance data can be found in [Müller et al., 2005].

Surface Light Fields

As mentioned earlier, one key factor in light field rendering is the ability to reduce the plenoptic function from five dimensions to four. One natural way of doing this is to use the hull of the object present in the light field (or even the object's exact geometry if available). Just like a texture can be wrapped around any object by creating a mapping from the texture coordinates to the points on the object's surface, the plenoptic function can be mapped to the surface as well. The mapping to the surface requires two dimensions of the 4D light field. The other two encode

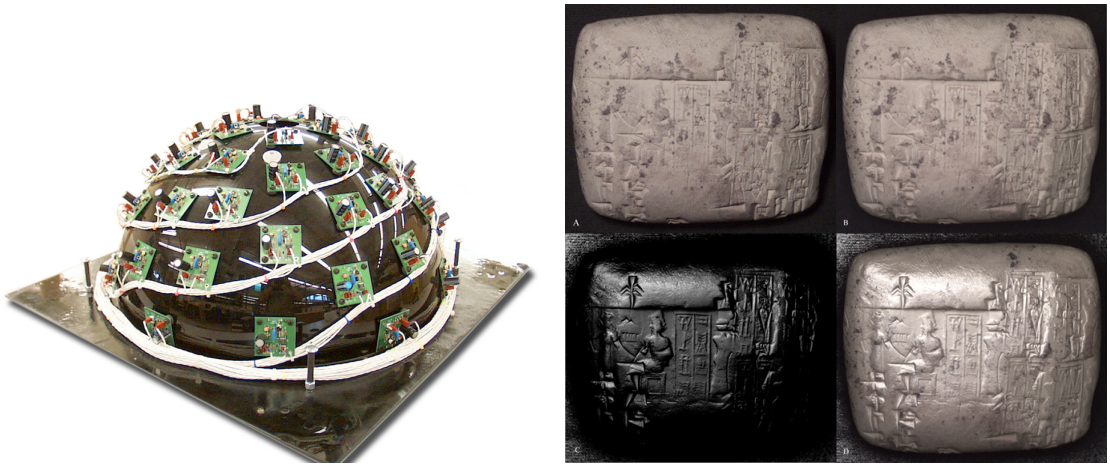


Figure 3.10: Setup and example rendering of work by Malzbender. From [Malzbender et al., 2001].

the direction light is traveling through this point. This particular approach leads to the definition of a *surface light field*. Even more sophisticated techniques refine this representation to describe how light is transformed by a surface irrespective of the incident lighting. This essentially closes the gap between the geometry free light field representation and a more traditional geometry and BRDF representation of an object. While one of the primary benefits of light field rendering is that it may be done in the absence of geometry it is worth noting that most traditional rendering techniques are centered around geometric definitions and as such rendering of light fields could benefit from some geometric information. Furthermore, as was observed by Gortler et al., interpolation of unique viewing angles not present in the source images is more accurate when there is approximate depth information for each pixel. This 'depth corrected' interpolation is automatic for a surface light field.

While Gortler et al. do use a rough approximation of object geometry to aid in view interpolation, Wood et al. [2000] took this a step further and developed a new approach that combines the light field information with precise laser scanned object geometry to create what they call a surface light field. In their representation, the light exiting in all directions is encoded for some dense sampling of points on the surface of the object. Each sample point contains what they call a lumisphere. The lumisphere describes the exitant radiance in all directions from that point. Rendering a surface light field is achieved by interpolating between angles in a lumisphere and

Figure 3.11: Comparison of results of work by Wood et al. Left are real photographs and right are renderings from the same angle. From [Wood et al., 2000].

interpolating between adjacent spheres to fill in the holes. In addition to this new representation they extended the compression and storage techniques of Levoy and Hanrahan developing a higher-fidility approach with similar compression ratios. Specialized forms of vector quantization and principle component analysis are both proposed as compression techniques. While Wood et al. consider their primary contribution to be the representation of the data and not its capture they do spend some time discussing the capture technique used. Other than the use of a laser scanning device to capture surface geometry the system described is very similar to a

standard system used for light field capturing (i.e. Levoy and Hanrahan). It involves a camera mounted on a spherical gantry system centered on the object of interest with a relocatable light outside the gantry. Wood et al. achieved much better results than Levoy and Hanrahan and Gortler et al. but do so at the expense of needing the object's precise surface geometry. This adds a geometry acquisition step to the capture process resulting in an even more complex capture device and the need for more sample images. Still, given the better photographic equipment available and the addition of geometric information to guide the sampling process only 638 images were required for the most complex example in their publication. An even simpler object required only 388 photographs. Some user interaction is necessary to register the photographs with the scanned geometry.

Buehler et al. [2001] surveyed the field of image-based representations and rendering algorithms including those devoted to the area of plenoptic photography (not discussed in this thesis). They examined the various trade-offs between representations including the need to re-parameterize the input prior to rendering. They proposed a new way of storing and rendering light field data that bridges the rather wide gap between the two-plane approach of Levoy and Hanrahan and the surface geometry approach of Wood et al.. They call their technique unstructured lumigraph rendering after the work of Gortler et al. [1996] (Gortler was also an author on the paper). The key observation that unstructured lumigraph rendering exploits is that every pixel of the output image is the result of a weighted sum of one or more pixels from the input images. By directly searching at every pixel for the best images that see that pixel and properly weighting them the same result could be achieved. This weighted function of the input images is called the *camera blending field* and is unique for every viewing angle of the light field. Every algorithm in light field rendering was achieving this same result but using very different techniques with different strengths and weaknesses. The technique of Buehler et al. looks at light field rendering backwards, starting with the output image and computing the camera blending field at key points. The result is interpolated across the output image to produce a new view that is an accurate blend of the input images. It is still necessary to determine which pixels in the output images come from which set of pixels in the input images. This is a re-projection problem that Buehler et al. resolve using the view-dependent texture mapping work of Debevec et al. [1998, 1996]. For view-dependent texture mapping to work, the original imagery must be projected onto some proxy geometry. For light fields that are sampled in a 2-plane arrangement, a simple plane can suffice for the proxy geometry. However, surface light fields

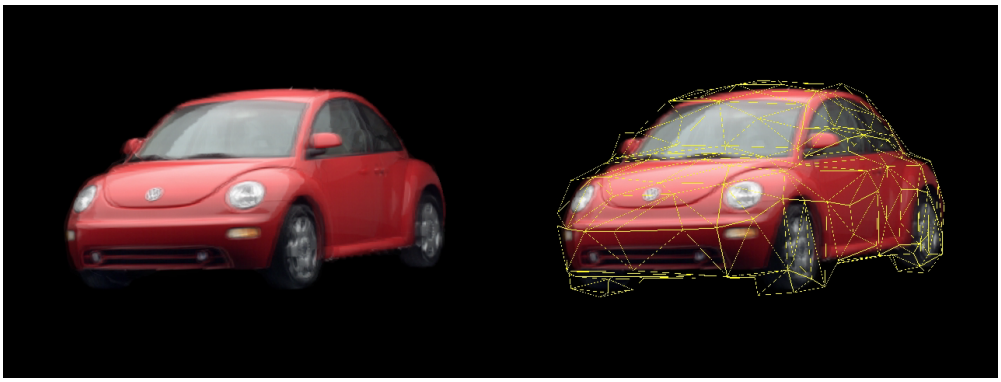


Figure 3.12: An unstructured lumigraph of a Volkswagen Beetle (left), and the proxy geometry used to render it (right). From [Buehler et al., 2001].

can use the underlying surface as their proxy geometry particularly in the form of the visual hull as used by Gortler et al. [1996]. Buehler et al. demonstrate many different results with their approach but focus on two-plane type light fields with a limited viewing range. However, they also demonstrate how a full-surround surface light field can be rendered using their technique and the visual hull of the object. Matusik et al. [2002] takes this technique even further (see page 39 for discussion of this work).

A final surface light field representation is proposed by Chen et al. [2002], called light field mapping, that has some similarities to the technique of Buehler et al. Like Buehler et al., Chen et al. use the surface of the object as proxy geometry and locate the light field samples directly on this surface. They also accelerate the rendering of the light field with texturing techniques and graphics hardware. However, they do not use the input images directly. Instead, they refactor the input images as samples of the full light field into two sets of two-dimensional functions called surface maps and view maps. These two maps are computed for small partitions across the surface that roughly correspond to the neighborhood around a vertex. The factorization is achieved numerically with either principle component analysis or non-negative matrix factorization. Rendering the resulting representation is very efficient on modern graphics hardware since the two functions amount to a pair of texture lookups. This representation is efficient to render and gives high-quality results but unlike Buehler et al. it requires lengthy processing and refactoring of the input images in the light field instead of rendering from them directly.

Figure 3.13: Three surface light field objects rendered into a larger synthetic scene. From [Chen et al., 2002].

Reflectance Fields

Debevec et al. [2000] took the surface light field one step further by computing what they term the *reflectance field*. The reflectance field is constructed by controlling the lighting environment and sampling the sphere of incident lighting directions. This introduces an additional degree of complexity to the capturing process as the incident lighting direction must be sampled for *every* viewing angle. Most early work in reflectance field capture [Debevec et al., 2000; Hawkins et al., 2001; Koudelka et al., 2001] resolved this additional capturing complexity by drastically reducing the number of viewing angles captured (only two for Debevec et al. [2000]). The resulting images of incident lighting directions allow the application of any incident light field to relight the sampled object including those sampled using traditional environment mapping techniques. Debevec et al. note that this representation is essentially equivalent to a BSSRDF (or bidirectional sub-surface scattering reflection distribution function) (see Figure 3.14). A

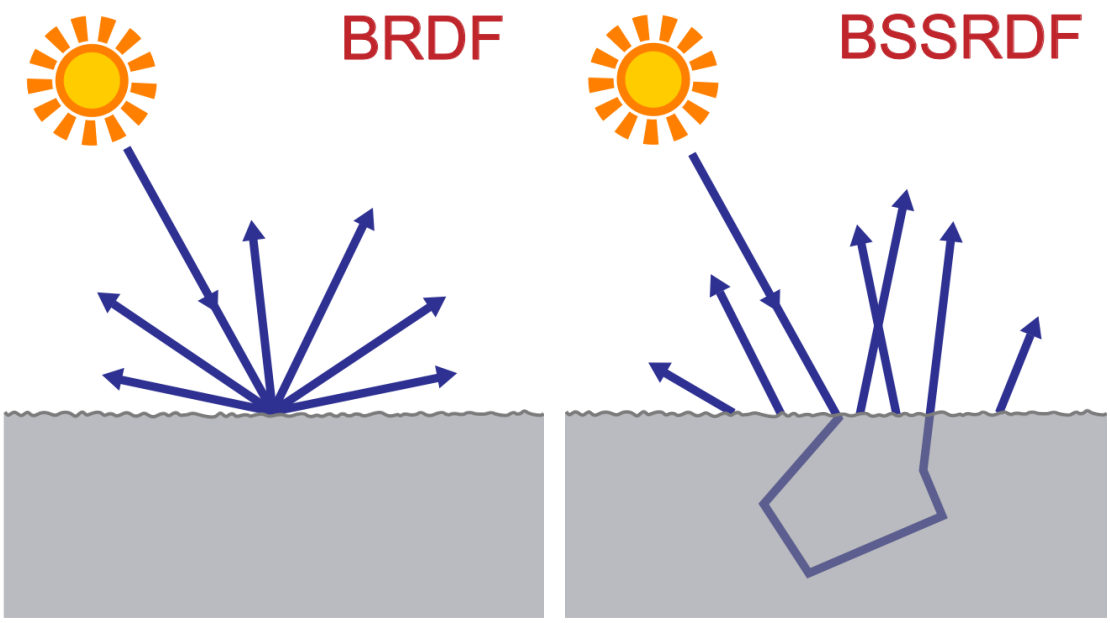


Figure 3.14: The BRDF assumes the incident light and reflected light share the same surface point. The BSSRDF does not make this assumption allowing the modeling of material where the light may penetrate the surface and bounce randomly before exiting from an entirely different surface point.

BSSRDF is an extension of the BRDF that allows light striking a surface at a particular point A from a particular direction ω_i to enter the material and leave at an entirely different point B in an entirely different direction ω_o. The points A and B introduce a minimum of four additional dimensions making the BSSRDF an eight-dimensional function. It is useful for materials that posses translucent qualities (particularly human skin) where the light does in fact enter the material and may exit it at an entirely different point. This correlation between the BSSRDF and reflectance fields provides the connective glue between light field based techniques and inverse rendering techniques discussed later in this chapter. In essence, surface light fields and reflectance fields are spatially varying representations of the BRDF and BSSRDF respectively.

The capture device used in the work of Debevec et al. was the first iteration of a series of devices developed by a research group from the University of Southern California called a *light stage* [Debevec et al., 2000, 2002; Einarsson et al., 2006; Hawkins et al., 2001; Wenger et al., 2005] (see Figure 3.17). It consists of a light mounted on a spherical gantry system with two

Figure 3.15: Setup and example rendering of work by Debevec et al. On left, the device known as Light Stage 1. On right, the top left image is one of the original photographs. The other three images are rendered from a unique angle using environment map lighting to relight the face. From [Debevec et al., 2000].

cameras positioned outside of the gantry to capture views of the object from the right and left. While two viewing angles are not sufficient to fully sample a typical object it was sufficient for Debevec et al. who were capturing human heads (specifically, faces). While there is no mention of acquisition of geometry of the face the paper considers the enclosing geometry of their light field to be co-incident with the surface of the face which would require at least some approximate geometry of the face. The video cameras working at 30 frames per second each capture 2048 images of the subject with different lighting yielding a total of 4096 images. The total capture time is only one minute due to the high rate of capture employed by the video cameras. A separate example discussed in this publication used a scene composed of diffuse,

Figure 3.16: Setup and example rendering of work by Matusik et al. On left, the capture device used. On right, the top image is one of the original photographs while the lower image is rendered from the resulting light field. From [Matusik et al., 2002].

specular and transparent glass objects. This scene required many more images (40,960) to accurately represent the hi-frequency changes in a specular reflection which is less dominant in human skin. While the resulting reflectance field could be rendered in any environment no attempt was made to render from a different viewing angle (other than the two angles used during capture). The use of video cameras for acquisition and the ability to relight the scene using environment maps make this a very versatile capture technique but the complex and large gantry system limit its use outside of a research lab environment.

Matusik et al. [2002] bring the different branches of light field research together by combining the flexible lighting of a reflectance field with the fully sampled viewing angles of a

surface light field to create a surface reflectance field. By fine tuning the number of incident light angles and viewing angles and leveraging the simplicity of a Quicktime VR style turn-table Matusik et al. propose a capture device that approaches practicality for the average researcher yet maintains the power of both a surface light field and a reflectance field (see Figure 3.16). Matusik et al. started with the representation and rendering scheme developed by Buehler et al. but focused on surface light fields and augmented the use of the visual hull with a new geometry representation called the opacity hull. Like the visual hull used in Gortler et al. [1996] the opacity hull can be computed from silhouette images of the object but also handles objects with imprecise or fuzzy boundaries (such as feathers, fur or leaves). The resulting surface light field interpolates the upper hemisphere of viewing angles and fully samples the incident lighting directions for rendering from any angle above the horizon under any lighting conditions. It is one of the most flexible light field representations to date.

This same approach to light field capture and rendering comes to an impressive climax with the sixth edition of the light stage [Einarsson et al., 2006] (see Figure 3.17). This expansive device replaces the rotating light source from Debevec et al. [2000] with a large hemispherical grid of LEDs and a set of floor mounted LEDs and mirrors for lighting angles below the horizon. The two static cameras of the older light stage are replaced with hundreds of high-framerate video cameras. The end result is a surface reflectance field *video*. An actor in this light stage can be filmed performing a task and each frame of video will capture over 230 lighting directions from hundreds of viewing angles along with hull geometry images effectively allowing view interpolation across the full sphere of viewing directions and re-lighting of the subject into any lighting environment. The scale and cost of this immense capture device make it impractical for even well-funded research labs to implement. As such light stage six is a powerful yet one-of-a-kind entry into the history of light field research.

3.1.4 Inverse Rendering

A typical rendering task starts by defining the geometry of a scene (including the position and orientation of an observer), the lighting of that scene and the reflective properties of the objects in the scene and their surfaces. Then, using various techniques, the image visible to the observer is generated. So, geometry, lighting and reflective properties yield images. Recent efforts in computer graphics have attempted to partially reverse this process. Starting with images of a scene (either photographed or rendered) from many different angles and positions they

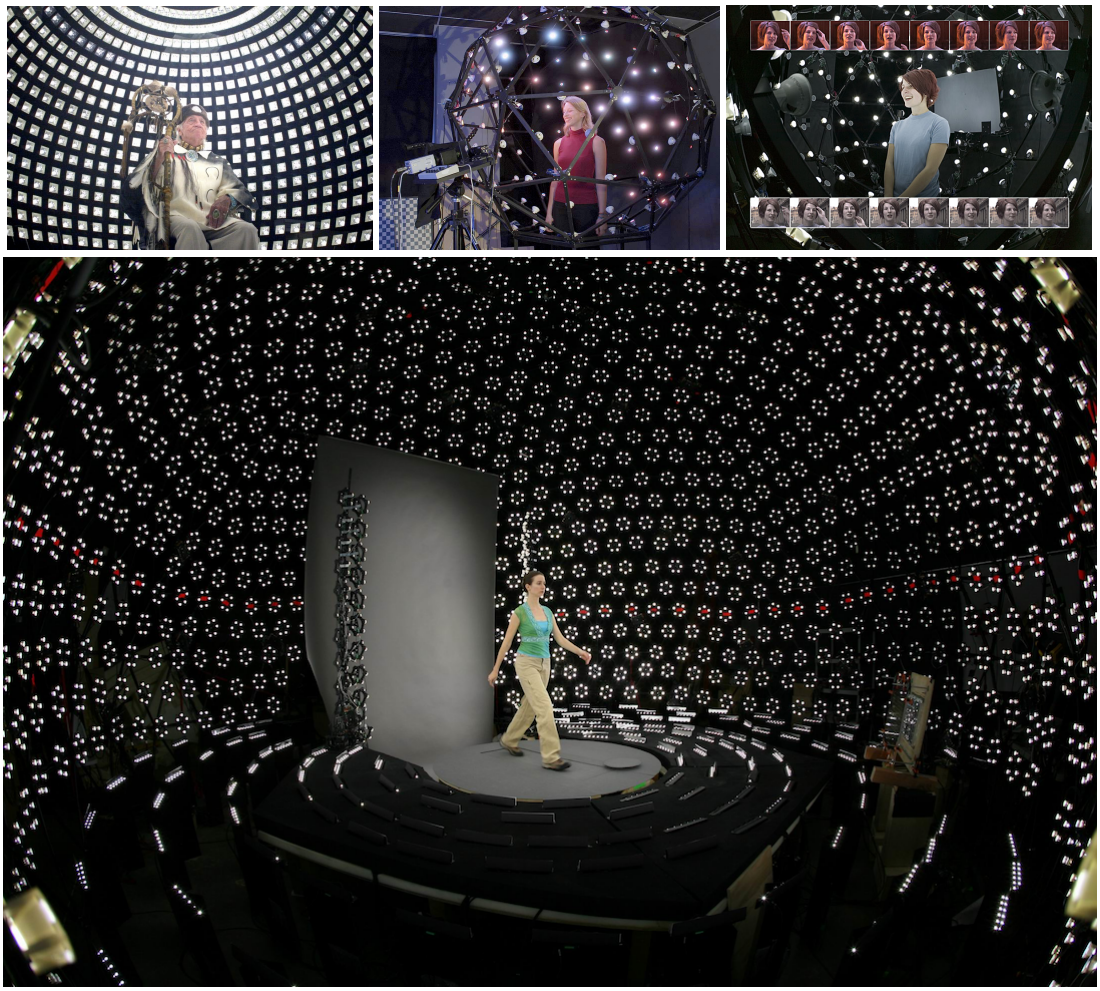


Figure 3.17: Work at the University of Southern California has produced these four light stages in addition to the original light stage mentioned above. Across the top: light stages 2, 3 and 5. Lower: light stage 6. From [Debevec et al., 2002; Einarsson et al., 2006; Hawkins et al., 2001; Wenger et al., 2005].

attempt to back out some of the scene's original properties (geometry, lighting and/or reflective properties). These techniques are collectively known as *inverse rendering* techniques. While the ultimate application of inverse rendering would be to start with a small set of photographs of an object or scene in the real world and then back out all of the traditional rendering properties (geometry, lighting and reflectivity) this is a nearly impossible task as the desired data has been

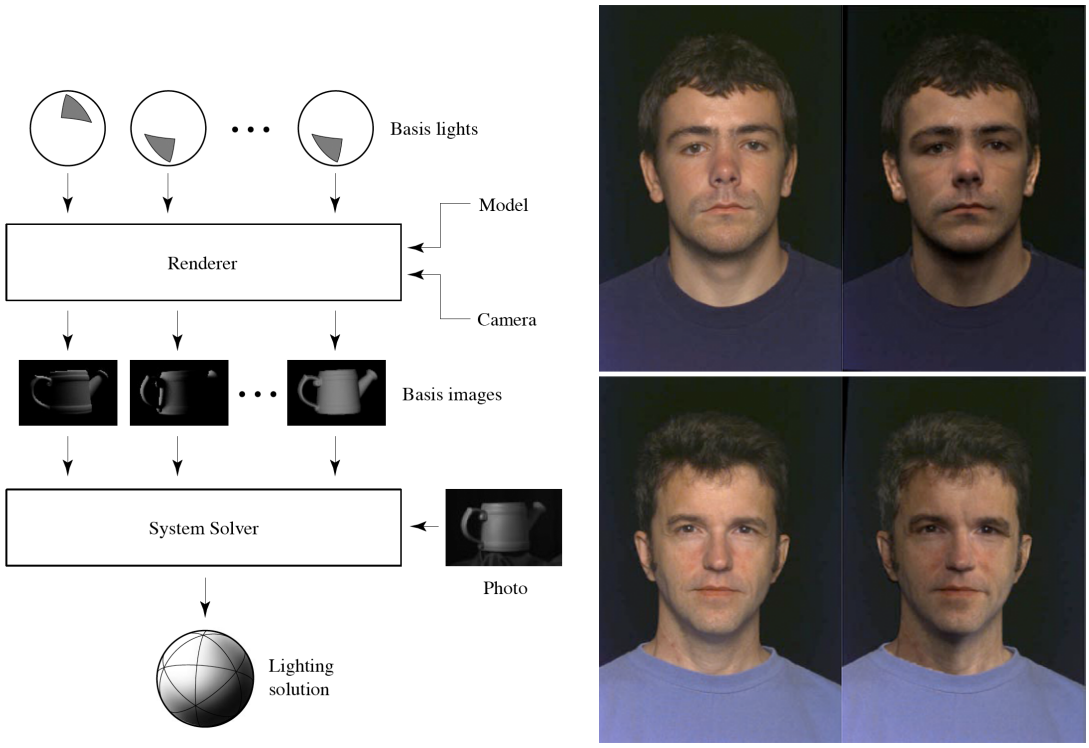


Figure 3.18: Algorithm (left) and example relightings of two photographs (right) using Marschner's inverse lighting technique. From [Marschner, 1998].

convolved into a single signal and that signal cannot be sampled sufficiently. Most inverse rendering techniques mitigate this by starting not just with images but also with a full or partial definition of the scene geometry and an approximation of the lighting information so that all that needs to be derived is the reflective properties of the surfaces. Since inverse rendering starts with photographs and attempts to compute the information necessary to generate new views under new lighting it is another form of image based rendering.

While the conceptual aspects of inverse rendering existed prior to the work of Marschner [1998] his dissertation is a logical and very complete starting point for discussion of this field. Marschner clearly identifies the task of inverse rendering and describes three different approaches to the problem all centering around recovery of the information necessary for rendering real world scenes and objects. Marschner's first technique is called *inverse lighting* where the lighting of a scene is recovered by examining a single photograph with known geometry

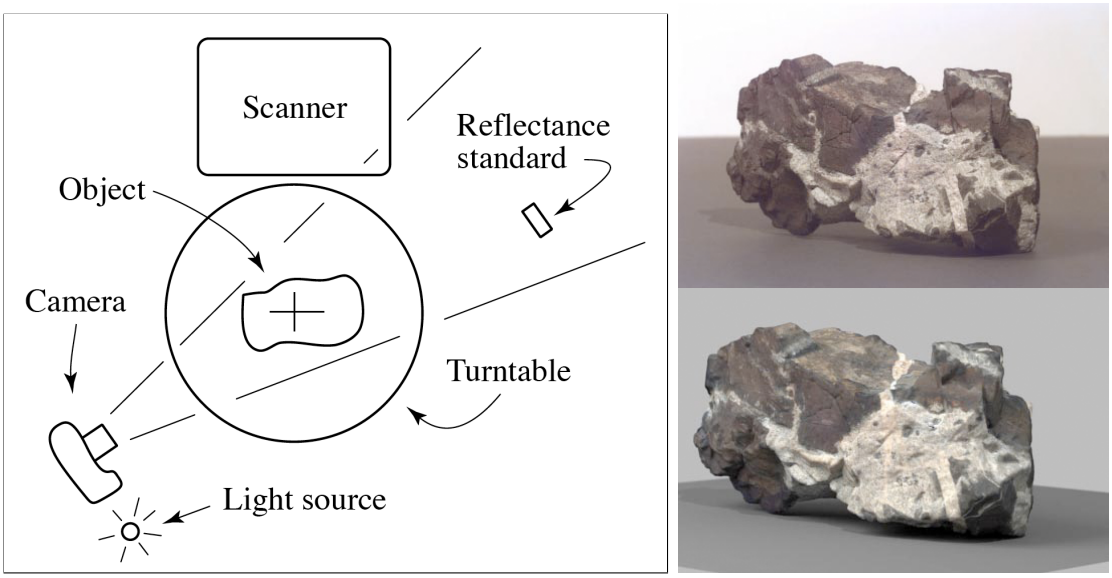


Figure 3.19: Setup (left) and results (right) of Marschner's photographic texture measurement technique. The top right image is a photograph of a rock and the bottom right image is a rendering of that rock using a textured object. From [Marschner, 1998].

and reflectance. With the lighting determined the photograph can be changed so as to appear to be located in a different environment (see Figure 3.18). This technique assumes the availability of geometry registered with the image pixels which is a difficult assumption to accommodate in practice but it requires only a single image of the scene to work. The second inverse rendering technique is called *photographic texture measurement*. This approach uses texturing to represent the reflective properties of the surfaces in the scene. It requires several photographs (the more images given, the more precise the end result) and requires known geometry and lighting of the scene. It efficiently computes a mapping from the photographs onto the geometry of the scene using overlapping texture patches (see Figure 3.19). It is only well suited to diffuse objects although extensions are proposed and demonstrated that handle higher-frequency specular surfaces. Finally, Marschner describes a technique called *Image-based BRDF Measurement*. This technique again works from multiple registered photographs of a single homogenous object with known geometry and lighting and recovers the BRDF of the object. Unlike the photographic texture measurement technique there are very few constraints placed on the underlying BRDF. In order to keep the acquisition device as simple as possible a homogenous BRDF is

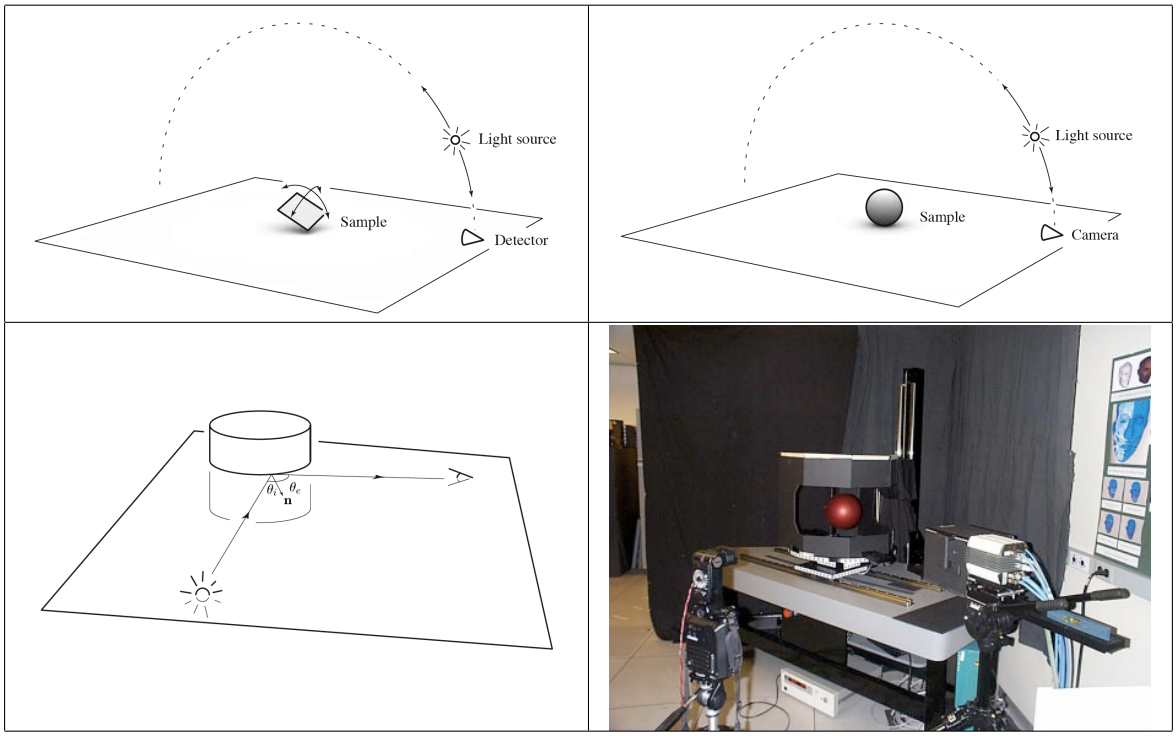


Figure 3.20: Different setups proposed by Marschner for measuring a BRDF. Top left: sample being measured tilts while light source moves in arc. Top right: sample is spherical and fixed while light source moves in arc. Bottom left: sample is cylindrical, light and detector move within a common plane perpendicular to the cylinder. Bottom right: the actual setup used in Marschner's BRDF measurement work. From [Marschner, 1998].

assumed and a spherical sample is used so that all viewing angles are visible from a single photograph (see Figure 3.20). All three of these techniques rely on the geometry of the scene being known which is an undesirable quality, however, there are many techniques in the literature that enable the recovery of accurate or approximate geometry using only images of a scene. This complements nicely with inverse rendering and is an integral part of most modern inverse rendering work. While none of Marschner's techniques constitute a complete system for acquiring the reflectance properties of real-world objects they all lay important foundations that are built upon in subsequent research.

More recent work by Lensch et al. at the Max Plank Institute for Computer Science has resulted in a complete inverse rendering system capable of acquiring reflectance properties of

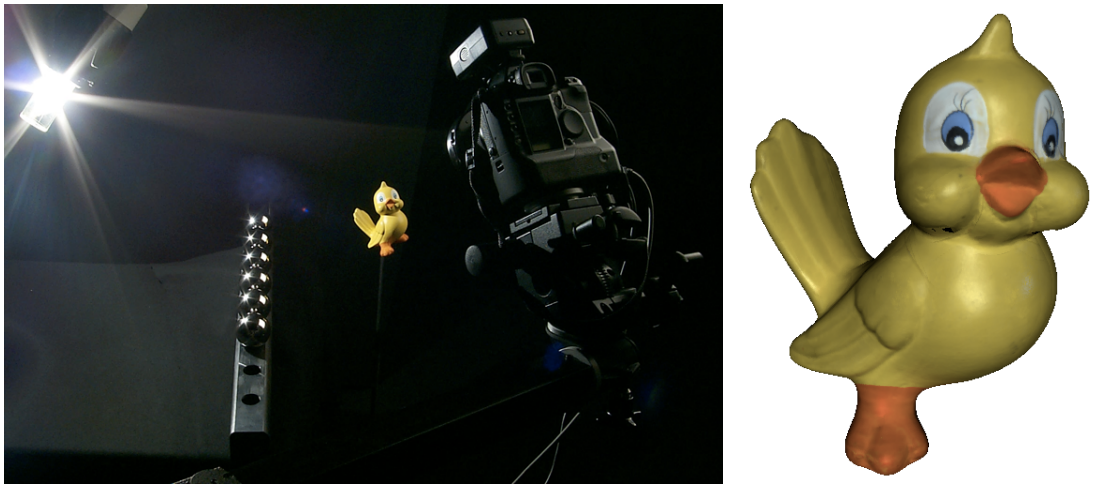


Figure 3.21: Setup and example rendering of work by Lensch et al. From [Lensch et al., 2003].

spatially varying objects [Lensch et al., 2001, 2003]. The lighting is not assumed to be known but instead is recovered from specific images of the object. A special lamp that approximates a point light source is necessary for this technique. In addition, a laser scan of the object's geometry is necessary. Unlike Wood et al. registering the images with this laser scanned geometry is fully automated. While this approach generates some very impressive results it requires a considerable amount of time and effort to execute and occupies a very large space. Each viewing and lighting angle was manually positioned in a room covered in black felt to reduce the amount of diffusely reflected light. The size and complexity of the environment required for capture make this a technique confined to a research environment. The processing required to generate the final representation of the reflectance properties is also very complex but yields a set of fitted parameters for an analytical BRDF representation (one for each identified submaterial in the object). This makes rendering the object an efficient and simple task. Further techniques from the Lensch research group offer ways to photographically acquire translucent objects [Goesele et al., 2004] and focus specifically on human faces [Fuchs et al., 2005].

A different inverse rendering technique built around a special photographic gonioreflectometer was developed by Gardner et al. [2003]. This technique resolves the tradeoff between accuracy and capture process complexity in a different way than the work of Lensch et al. They propose a *Linear Light Source Apparatus* that acquires simplified reflectance properties for relatively flat objects and materials. Working much like a flat-bed scanner the device passes a

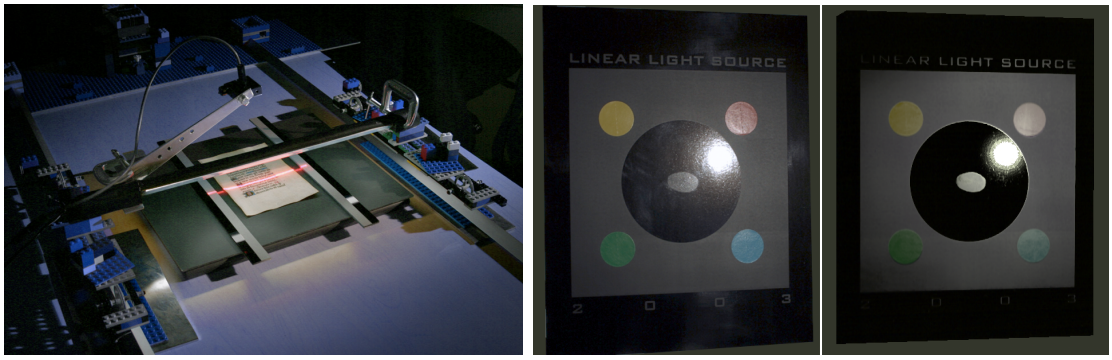


Figure 3.22: Setup and example rendering of work by Gardner et al. (center image is real poster, right image is rendered). From [Gardner et al., 2003].

tubular light source over the object (they use a white neon tube normally used for accent lighting on an automobile) along with a laser light which together allow for approximating diffuse and specular colors as well as specular roughness and normal displacement at every point on the object (see Figure 3.22). The diffuse color, specular color and specular roughness provide the parameters for a very basic reflection model (such as the Phong model) while the normal displacement gives an approximate normal map for the object that captures small geometric details in the surface. The entire capture process is automated as is the extraction of reflection properties. However, the system will only handle flat objects with relatively little surface detail and the reflection model the data is fit to is far too simplistic to handle the full spectrum of real-world materials. A surprisingly large number of materials used in the automotive, fashion and interior decorating industries are flat materials. Fabric swatches, metal panels, wood veneers, countertops, wall treatments, carpeting, just to name a few all would be acceptable materials to scan with Gardner's Linear Light Source Apparatus. A more sophisticated reflection model would be necessary to handle all of these materials or to facilitate acquisition of full BTF data with this device.

3.1.5 Other Image-Based Rendering Reflection Models

Inverse rendering techniques are concerned with more than just generating new views of an object or scene as was the purpose of light field rendering. Inverse rendering attempts to reproduce the data that would normally be used in a forward rendering task. For our purposes, we

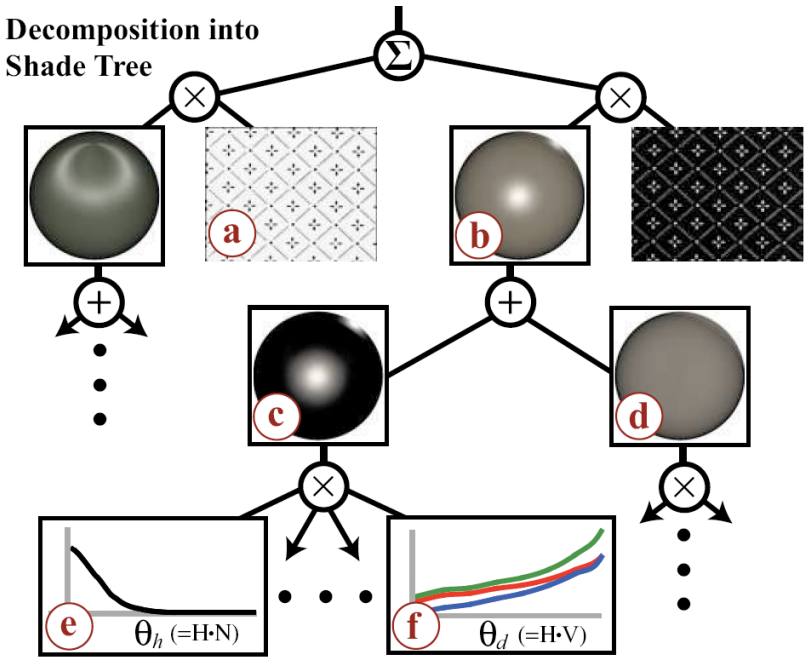


Figure 3.23: Structure of an inverse shade tree. From [Lawrence et al., 2006].

are most concerned with data related to surface reflectance properties (the BRDF of the object or scene in question). In most forward rendering techniques an analytical model of the BRDF is used instead of real BRDF data which is difficult to acquire and cumbersome to work with. As such, most inverse rendering tasks will propose a fitting of the reflectance data backed out of the images to some analytical reflectance model. In this section we will discuss some models of interest that have been proposed for general inverse rendering tasks separate from the capture technique.

The work of Lawrence et al. [2006] presents a technique for modeling spatially varying reflectance data as a 'shade tree'. Their work is particularly well suited to inverse rendering tasks where geometry will either be known or acquired along with the reflective properties. Their technique employes matrix factorization in constrained spaces to produce what they call an inverse shade tree. This inverse shade tree represents the reflective properties as combinations of one or two dimensional functions that capture directional behavior of reflection as well as spatial mixing patterns. The most useful characteristic that comes from this technique is the ability to edit the resulting reflective properties in a very intuitive and natural manner. It builds

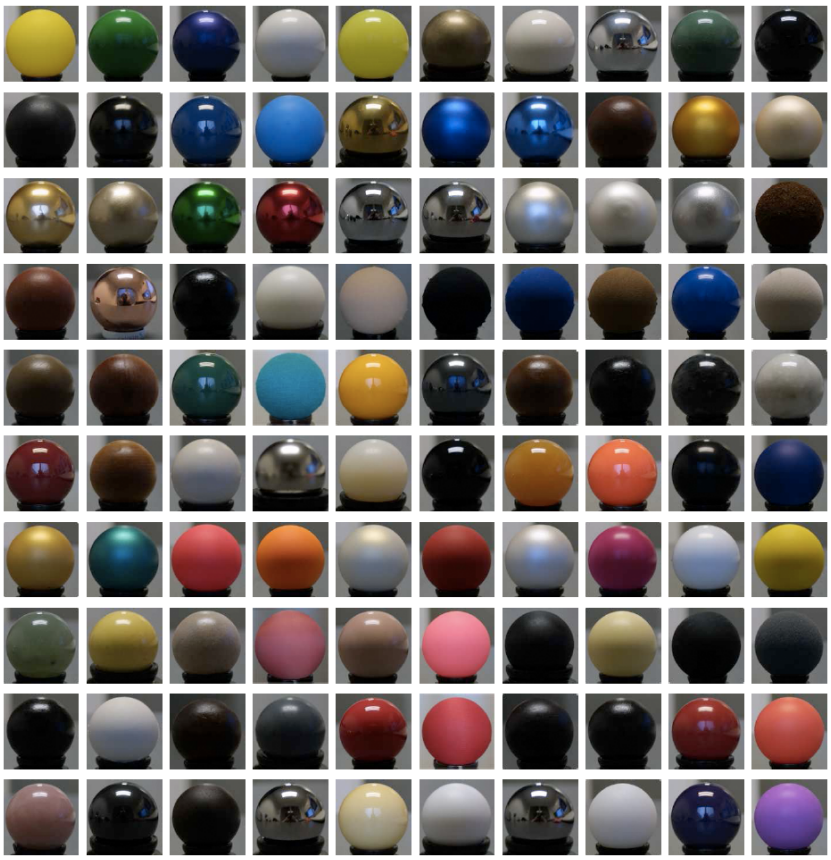


Figure 3.24: Several of the sampled materials represented using the data-driven model proposed by Matusik et al. From [Matusik et al., 2003].

upon previously developed ideas such as well known empirical models of reflective properties and texture mapping which have already received much attention for real-time user editing. This characteristic is particularly well suited to a design task where the captured material is to serve as inspiration and need not be an exact duplicate of the real world material. While this technique does not specifically describe a photographic acquisition process its applicability to data from other photographic techniques and its editable nature makes it important to consider as part of a tool for the design process.

Matusik et al. [2003] worked to develop a way of modeling all homogeneous isotropic BRDFs with a single, common manifold space. They fit photographically sampled reflectance data to this manifold to produce a general BRDF for rendering. They used a capture device

identical to the device in Marschner's image-based BRDF measurement system [Marschner, 1998]. As such, they are limited to sampling small homogeneous spheres (see Figure 3.24). Their alternate representation is beneficial in that it relates all BRDFs to one another using a common manifold space. This provides connections between different aspects of every possible BRDF along the dimensions of this space. With this comes the possibility of computing the difference between BRDFs and even interpolation between them. Matusik et al. also demonstrate some interesting blending techniques to simulate weathering of real-world materials such as rust, oxidation and dust. They also attempt to define several perceptually-based dimensions within their manifold space making editing of a BRDF more natural than physically based parameters of common analytical models. The BRDF may be adjusted along these perceptual dimensions in an intuitive manner to fine-tune the properties of a physically measured BRDF. The only significant drawback of this method is its limitation to isotropic BRDFs. The capture technique used could be replaced with a more sophisticated system to allow it to apply to non-homogeneous materials. However, the density of sampling required would be greater than with other reflectance models creating a tradeoff between flexibility in the representation and difficulty in the acquisition process. Furthermore, considerable work and analysis would be necessary to extend their manifold analysis beyond isotropic BRDFs if this is even possible.

3.1.6 Summary

In Table 3.1 we summarize the various photographic techniques. Particular attention is paid to ease of execution compared with accuracy and versatility of the final representation.

3.2 Related Work in Morphing

In this section we look at the technique of morphing (short for metamorphosis). Morphing is the practice of taking two objects and distorting them while simultaneously blending between them to give the illusion that one is smoothly melting into the other. This can be done to 2D images and shapes or 3D volumes and surfaces. Morphing is related to, but distinct from, warping. In warping a single object is distorted in some controllable way to produce a consistent and predictable result. Morphing consists of two or more warps (one defined for each target in the morph) and a simultaneously executed blend.

- Chen; McMillan & Bishop 1995

Images: 36 – 650	Capture Time: minutes
Summary: Both Chen and McMillan & Bishop developed the beginning of the Apple QuickTime VR system which uses static images in a linear sequence, played back according to user interaction to allow rotation of a single virtual object under fixed lighting.	
Pros:	Cons:
- Robust and supported by existing software	- Rotational control only
- Can be created with off-the-shelf hardware	- Lighting is static
- Relatively few images and time required to create	

- Levoy & Hanrahan 1996

Images: 100s to 1000s	Capture Time: hours
Summary: One of the first attempts at capturing, storing and rendering light field. Results suffered from inferior imaging technology and from limiting light-slab representation. Storage and compression techniques presented work quite well.	
Pros:	Cons:
- Full viewing control of object	- Lengthy and cumbersome capture
	- Lighting is static
	- Custom scanning device and software

- Gortler et al. 1996

Images: 100s	Capture Time: hours
Summary: Another first attempt at capturing, storing and rendering a light field. The capture process is much less restrictive than Levoy and Hanrahan and storage and rendering is aided by rough geometry determined from the same set of images as the original light field. Capture process is still lengthy.	
Pros:	Cons:
- Full viewing control of object	- Capture is lengthy
- Simpler, less-restrictive capture	- Custom processing and storage
	- Relies on approximate geometry

- Dana et al. 1999

Images: 200+	Capture Time: hours
Summary: An extension of the concept of texturing that lays a BRDF across the surface of an object instead of a color (a BTF). Describes a technique for capturing BTFs but does not attempt to make storage and rendering efficient.	
Pros:	Cons:
- Lighting is dynamic	- Capture is lengthy and complex
- Full viewing control	- BTF must be placed on geometry
	- Custom scanning device and software

- Han & Perlin 2003

Images: 20 – 100	Capture Time: hours
Summary: A unique method for capturing multiple viewing angles of the same object in parallel using a kaleidoscope. Dramatically reduces number of images and capture time.	
Pros:	Cons:
- Quicker process, fewer images needed	- Custom scanning device and software

Table 3.1: Summary of each technique discussed in Chapter 3.

- Malzbender et al. 2001

Images: 50	Capture Time: minutes
Summary: A technique for capturing surface details of a fixed object. Geometric details are encoded as a polynomial BRDF representation and the coefficients are stored in a texture called a polynomial texture map.	
Pros:	Cons:
- Simpler capture device	- Viewing angle is fixed
- Shorter capture process	- Designed for flat objects with surface detail
- Very efficient storage and rendering	- Best suited for diffuse materials

- Wood et al. 2000

Images: 400 – 600	Capture Time: hours
Summary: A more recent approach to light field capture and rendering, this work encodes the light field using the surface of the object as the parameterization. Light field is stored as a sphere of exitent light at each point on the surface of the object.	
Pros:	Cons:
- Full control of object	- Capture process is lengthy and cumbersome
- Efficient storage and rendering	- Uses precise scanned geometry
- Very impressive end results	- Numerical refactoring required

- Beuhler et al. 2001

Images: 10's – 100's	Capture Time: N/A
Summary: New rendering algorithm that handles light field images directly and renders surface light fields at interactive rates. Supports two-plane representations and surface based object representations.	
Pros:	Cons:
- Fast rendering implementation	- Not as precise as algorithms that re-factor
- Intuitive rendering algorithm	- Cannot extrapolate to fill holes in sampling
- No refactoring of input images needed	

- Chen et al. 2002

Images: 200 – 400	Capture Time: hours
Summary: A new technique is presented that combines the accuracy of Wood et al.'s lumisphere representation and Beuhler et al.'s unstructured lumigraph rendering algorithm. It is well suited for execution on modern graphics hardware. They also describe a capture process similar to that used by Gortler et al.	
Pros:	Cons:
- Re-posing of final object	- Numerical refactoring required
- Efficient storage and rendering	- Uses precise scanned geometry
- Very impressive end results	

Table 3.1 (continued)

- Debevec et al. 2000

Images: 1000s	Capture Time: minutes
Summary: Focusing on rendering and re-lighting human faces, this work extends the concept of a light field to a more general representation called a reflectance field. The reflectance field is independent of the light falling on the object and can therefore be relit using environment based lighting.	
Pros:	Cons:
- Lighting is dynamic	- New viewing angles limited
- Supports environment based lighting	- Capture designed for human faces
	- Custom scanning device and software

- Matusik et al. 2002

Images: 100s	Capture Time: minutes – hours
Summary: The work of Buehler et al. is extended to include sampling of the reflectance field and support for fuzzy objects like fur and feathers.	
Pros:	Cons:
- Lighting can be changed	- New viewing angles limited
- Fuzzy objects are well supported	- Capture designed for human faces
	- Custom scanning device and software

- Einarsson et al. 2006

Images: 1000s	Capture Time: minutes
Summary: Able to capture surface reflectance light field videos allowing smooth playback of a performance by an actor from any angle and in any lighting. Data in the temporal dimension enhances the quality of the surface light field representation.	
Pros:	Cons:
- Lighting is dynamic	- New viewing angles limited
- Re-light and re-pose a performance	- Designed for walking or running subjects only
	- One-of-a kind, full room scanning device

- Lensch et al. 2001

Images: 20 – 100	Capture Time: hours
Summary: Using the object's geometry this technique factors out the constituent materials in the object and fits BRDFs to each of these.	
Pros:	Cons:
- Very impressive end results	- Difficult and lengthy capture process
- Full control of object	- Custom scanning device and software
- Lighting is dynamic	

- Lawrence et al. 2006

Images: N/A	Capture Time: N/A
Summary: A special representation is proposed that is well suited to inverse rendering techniques. The representation factors the spatially varying BRDF information into a hierarchical combination of non-spatially varying BRDFs and maps indicating where they apply. This representation is well suited for editing inverse rendering BRDF information after capture.	
Pros:	Cons:
- Resulting representation is editable	- Processing software is custom
- Can be used with any inverse rendering technique	

Table 3.1 (continued)

A good, but somewhat out-of-date, overview of morphing and warping in general is available in the book by Gomes et al. [1998].

3.2.1 2D Image Morphing

Image morphing began in the movies and has been very successful over the years. Many of the different techniques described here were used in music videos and Hollywood films to great effect. A good overview of 2D image morphing is available in Wolberg [1998]. In general, all image morphing techniques have these two key components:

- **Placing Features**: The user identifies elements on both images that should stay matched as the images warp.
- **Computing the Warp Field**: The sparse correspondences represented by the features identified by the user must be expanded into a dense correspondence known as the *warp field*.

These two components are usually interrelated as the nature of the identification scheme afforded by a particular morphing algorithm often dictates a certain way of computing the warp field. Here, we will discuss only three of the simpler image morphing techniques; mesh warping, field morphing and scattered data interpolation.

In mesh warping [Wolberg, 1990], the user must design a pair of matching meshes of points and edges on both images (see Figure 3.25). To aid in this, the user is usually presented with a regular grid of quads on both images and they must move the intersecting points to elements that should correspond between the targets. This approach allows for easy generation of the warp field and execution of the final morph simply by interpolating this mesh. However, the grid can be non-intuitive to edit requiring the user to choose corresponding elements where there is nothing logical to choose from. This can result in unintended input from the user. Alternatively, the grid may be established in a more natural fashion by allowing the user to place individual points and then triangulating them afterwards. In this situation it is difficult to maintain consistent topology between the target meshes. For example, a single point that moves too far between the two targets might cause a triangle in the mesh to fold over on itself and become degenerate. Furthermore, it is unclear which positions (e.g. on one of the targets or somewhere in-between) should be used to compute the triangulation of the freely positioned points.

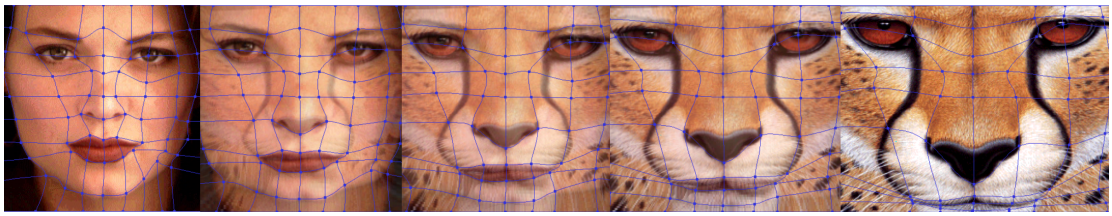


Figure 3.25: A mesh morph example from the book [Gomes et al., 1998].

In field morphing [Beier and Neely, 1992], the user can place lines around the image to indicate which landmarks should stay matched. Each feature influences every pixel in the image according to how close the pixel is to that feature. By considering proximity to all the features and not just the closest feature, a field of influence is created that drags the image pixels into new positions as the warp executes. For the user, it is easier to understand and simpler to input these features, but the algorithm to execute the warp is more complicated. Additionally, if the movements from one target to another are too great, field morphing can result in undesirable folding or ghosting artifacts. These artifacts can often be overcome by adding new features to counteract unexpected behavior.

The warp field portion of the morphing process can be thought of as a form of scattered data interpolation. The user is allowed to lay down any features either as points or as other primitives (lines, curves, etc.) that can be sampled into points. On the source image, let features points be (u,v) and on the target image let features points be (x,y). Consider two separate sets of scattered 3D points defined as (x,y,u) and (x,y,v), one point in each set for each feature pair in the input. These two sets give us a way to start with a feature on the source and go to a feature on the target by associating the coordinates with one component on the other image. Determining how to warp points in the image that are not on these features can be thought of as fitting two smooth surfaces to these two sets. Points that are not on the features will be on those surfaces and a dense warp field can be generated. There is a rich body of work in solving this problem that can be drawn upon. One successful technique from this area uses thin plate splines to interpolate the scattered points [Lee et al., 1994; Litwinowicz and Williams, 1994] (see Figure 3.26) and another uses radial basis functions [Arad et al., 1994].

Other techniques in image morphing involve energy and work minimization schemes that ensure the resulting warp field between the targets is one-to-one [Lee et al., 1996, 1995].

Figure 3.26: A feature morph the employs sparse data interpolation with thin plate splines. The morph is between two images with different facial expressions. From Lee et al. [1994].

3.2.2 Shape Morphing

The morphing of shapes in both two and three dimensions is an area rich in prior work and a full examination of that work is beyond the scope of this thesis. Instead, we focus on two key techniques which have proven successful: distance field morphing and parameterization/embedding.

The distance field of a boundary is defined as the shortest distance from the current position to the boundary. For a silhouette image where the boundary is black and everything else is white (both interior and exterior) the distance field is the length in pixels to the nearest black pixel. Sometimes it is useful to consider the signed distance field where everything inside the object is positive and everything outside is negative. The distance field is particularly useful because it represents a well behaved 'implicit' function for the shape. This means that the shape is located at the zero points of the function. Furthermore, the gradient of the unsigned distance field will naturally guide you to both the shape (the global minima) and the central skeleton of the shape (the local maxima within the shape's boundary).

From the stand point of computing a morph between two shapes, the distance field offers an intriguing alternative to a warp field defined on the shape itself. Instead of warping the shape,

we consider the shape as embedded in the distance field and warp the space around it. So, the shape is embedded in the field where the field equals zero. By computing this field at each target we have a warp field and interpolate between the values to produce the morph. This technique is equally successful when applied to 3D objects. Here instead of a shape we have a surface and we must define the distance field over a three space volume that contains the surface. It works the same way; at every point in the volume the value of the distance field is the shortest distance to the surface.

The distance field is discussed in many publications [Cohen-Or et al., 1998; Payne and Toga, 1992; Raya and Udupa, 1990] and has many desirable properties. First, because it morphs the space around the shape or surface instead of the object itself it is free of topological constraints. Objects of any arbitrary topology can be morphed into objects of a different topology and the process is the same (see Figure 3.27). Furthermore, no user interaction is required (though some work to roughly align the objects through rotations and or scalings prior to computing the morph can help control the results). However, this leads us to one of the undesirable properties of distance field morphing; it does not afford precise control over the resulting morph. Cohen-Or et al. [1998] describe some means to control the overall motion and quality of the morph through affine transformations but forcing specific landmarks to match up between the morph targets is difficult if not impossible. Another disadvantage to this approach is that all sense of the original object is lost. The space around the object is morphing not the object itself and as such any parameters that were defined on that object are lost. This is particularly problematic for surface appearance. Textures or light field samples that were associated with points on the surface cannot be morphed along with the object's geometry. The connection between those parameters and the surface cannot be encoded by the distance field.

To overcome these problems the only clear alternative is to attempt to compute a warp field directly on the object. In 2D this means we must map every point on one contour to the other (in 3D, every point on one surface to the other). This can be achieved by mapping both objects to a common and simpler shape. For 2D contours, the shape of choice is usually a circle. The contour can be mapped to the circle through a process called normal evolution. In normal evolution, the contour is moved everywhere along its normal (a vector perpendicular to the shape tangent and pointing outward). The direction of motion is determined by the local curvature. Initially only points that have negative curvature are moved outward along the normal so as to eliminate concavities. To speed up the process points with positive curvature can be

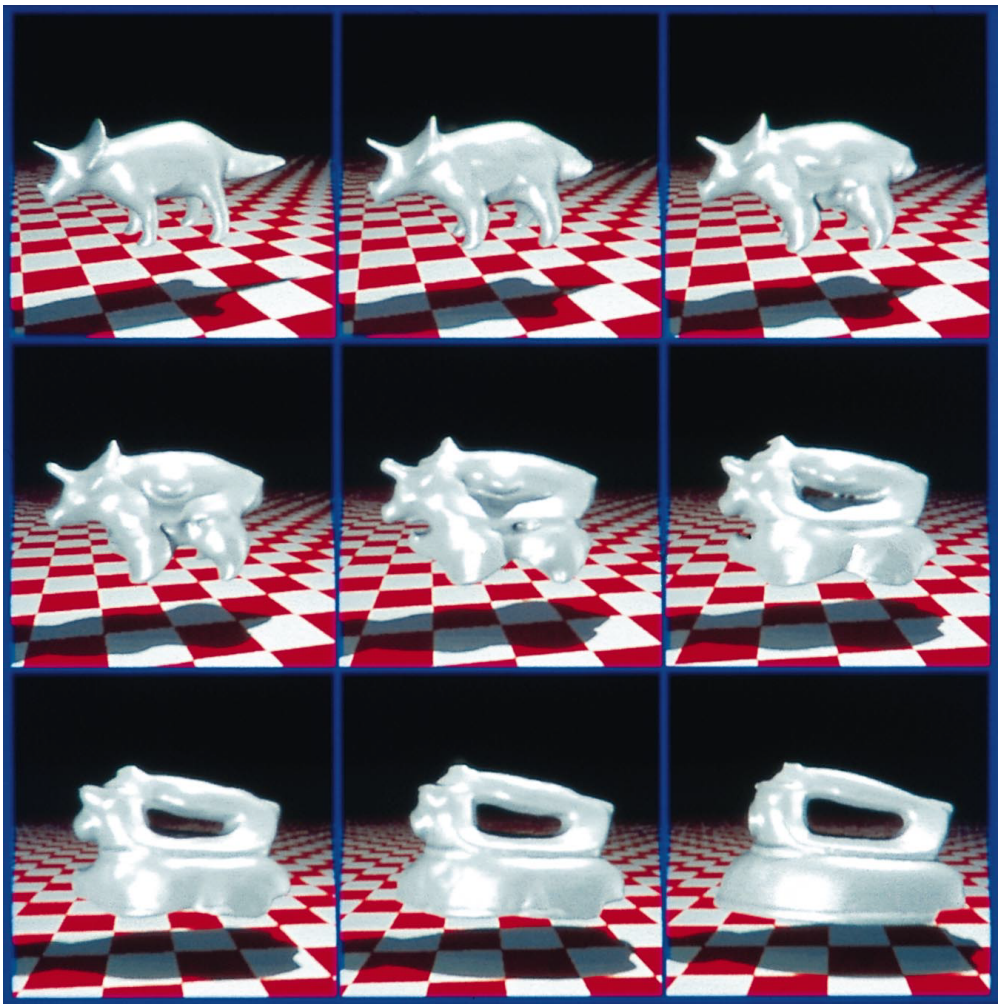


Figure 3.27: A 3D shape morph computed between objects of different topology using the distance field technique. From Cohen-Or et al. [1998].

moved inward against the normal at the same time. As these points move they will naturally collide or fold-over each other. This must be detected and conflicting points must be merged. Merges must also be remembered or the process cannot be reversed later. Once the curvature is everywhere non-negative the object can be taken out to the sphere through projection through the center of mass.

Once both targets have been mapped to a circle, all of the points on each target can be paired up by comparing their positions on this circle. The warp field naturally falls out of this common shape. While there is a natural extension of this idea to 3D where the circle becomes a sphere, the algorithm becomes considerably more complex and collisions and foldings are harder to detect and handle properly. Furthermore, it is only well defined for objects that are homeomorphic to a sphere. Objects with a genus greater than zero cannot be handled with this approach. The same is true in 2D but here there are good alternatives and work arounds and objects that are not homeomorphic to a circle are less important.

As an alternative to the normal evolution process in 3D the general problem of mapping an object to a sphere can be considered (which also has a large body of related work). Botsch et al. [2010] and Hormann et al. [2007] both offer a good starting point for surveying this problem. In computer graphics, we often look at this as 'parameterizing' the object so we can use various texturing techniques. The parameterization we want should have a spherical layout but mapping to a planar geometry (which is also common) can work as the resulting plane can be folded into a cube and then deformed into a sphere.

Regardless of how the geometry is mapped to the circle or sphere, this morphing algorithm solves both of the problems of the distance field technique. Once you have computed the mapping to the common geometry you can then move and pull individual points from the original shape all you want to control what gets mapped to where on the other object. A natural way to afford this is to define a sparse correspondence between the shapes to begin with and then use the embedded common geometry only when you need to make that sparse correspondence dense (through interpolation across the common geometry). The other benefit is that the shape itself is never lost during the construction of the warp field. Consequently, parameters like appearance can be brought along and morphed with the shape of the object. Naturally, since the warp field is defined on the shape itself, changes in topology will require special attention and are generally hard to deal with particularly in 3D.

The tradeoffs between topology dependency and control over the morph as well as the

ability to bring along parameters on the object make these two techniques quite complimentary to one another. Both will be considered as we design our light field morphing technique.

3.2.3 Light Field Morphing

Morphing between light fields in general is not a new idea. The pervasiveness of image morphing and the fact that light fields are based on sets of images makes the connection to morphing of a light field an easy one to make. However, simply computing a set of 2D morphs from all the images in the separate light fields with no regard for the underlying surface implied by the object will not yield reasonable results. Alternatives have been presented that work within the pure definition of light as a field to overcome these problems and essentially ignore the underlying structure of the light field's representation.

The first light field morphing technique was described by Zhang et al. [2002]. They approach the problem by considering the light field directly as a collection of rays (this is natural from our earlier definition of light as a field). The user provides 3D polygons that bundle the rays of the light field. Bundles are paired up between the targets and this constitutes a warp field. During the morph it is possible that a portion of an object that was previously not visible from that particular angle may become visible. In this case, there is no information in the light field samples to fill in this hole. Zhang et al. solve this problem by filling in the hole using nearby rays within the bundle. In the end, they must compute the morph itself on the raw light field rays which is inefficient both in space and time but they achieve high quality results and do not need any explicit or implicit geometry to achieve their morph.

Jeong et al. [2003] propose a surface light field morphing technique (see Figure 3.28). In their approach they use the surface light field mapping technique proposed by Chen et al. [2002]. They separate the computation of the morph into two key problems. First, morphing of the surface geometry (achieved by a slightly modified mesh morphing technique) and second, morphing the light fields. Recall that in light field mapping the input images are not directly used but are refactored into a pair of 2D functions for surface position and viewing angle. Directly morphing the input imagery is therefore not compatible with using this representation. Jeong et al. choose to avoid computing the in-between light field and instead map the point on the morphed surface back to both targets and then blend their values. Because of this, the morphed reflectance values are never fully computed but are a consequence of the rendering of the morph.

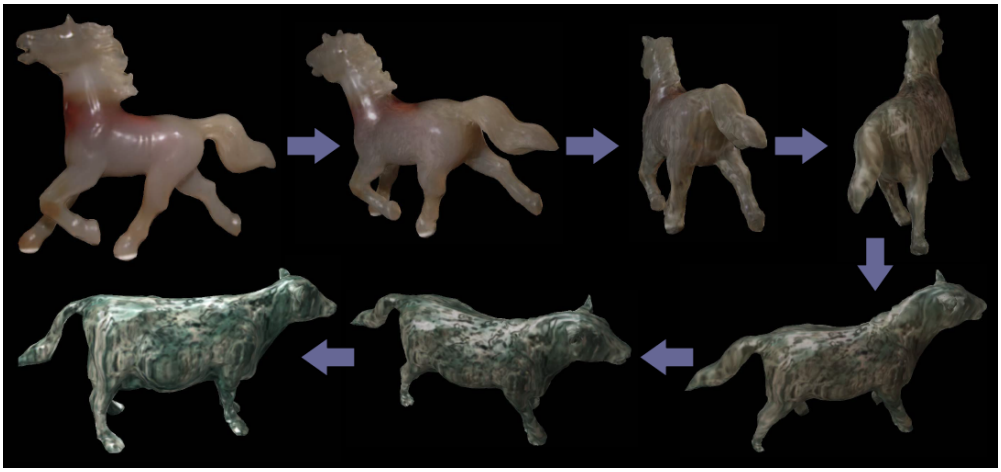


Figure 3.28: A morph between two surface light fields. From Jeong et al. [2003].

Wang et al. [2005] offer an extension to the work by Zhang et al. [2002] that allows the user to specify 2D features instead of the 3D features of the previous work. The user must locate the same feature in several of the key views that makeup the input to the light field. The algorithm will then extrapolate the location of the feature in the other key views. From here, the morph proceeds in much the same manner where the full light field rays are bundled according to the specified features and warped from one target to the other.

3.3 Shape-From-Silhouette

In computer vision there is an important problem whereby one needs to compute an approximation of an object's 3D shape from several views of its silhouette. This is often referred to as *shape-from-silhouette*. There is a very extensive history of research into this problem and many proposed solutions that are beyond the scope of this thesis. A good survey of solutions and some unique applications of this problem can be found in Cheung et al. [2005a,b].

This problem is related to surface light fields because it can offer a way to avoid explicitly defining the geometry needed to parameterize the light field. By extracting the silhouette present in each view that makes up the input to the light field a shape-from-silhouette algorithm can approximate the object's geometry. When computed in this manner, only features that are visible somewhere in the silhouette of the object will be captured. True concavities (different

from saddle points) will not be properly captured. The resulting geometry is often referred to as the *visual hull* of the object; visual because it comes from the features visible in the silhouette and hull because it completely contains the actual geometry of the object.

There are two solutions to the shape-from-silhouette problem that are of particular interest for our work. First, one can use the silhouette images to define an implicit function of the visual hull. This implicit function exists in 3D (much like the distance function described previously) and must be zero at the silhouette edge, positive inside the silhouette and negative outside it. From here we have a volumetric definition of the surface and can use any polygonizing technique from volumetric rendering (such as marching cubes [Lorensen and Cline, 1987]) to obtain our geometry. The signed distance field of the object would make an ideal implicit function and can be well approximated from the silhouette images by computing the 2D signed distance field for each silhouette and then returning the smallest value mapped from any of the views.

The other solution to the shape-from-silhouette problem that is of interest is the geometric approach. Here the silhouettes themselves are projected back into the 3D space they came from when they were imaged. This creates a cone with the cross-sectional shape of the original silhouette. All of these cones will intersect and naturally carve out the visual hull of the original object. Implementing this process is more complicated than the volumetric approaches but it yields precise results whereas the volumetric approaches are discrete and imprecise. The geometric approach can also lead to a real-time visual hull algorithm described by Matusik et al. [2000]. In this approach, called image-based visual hulls, the desired view is computed directly from the input silhouettes using epipolar geometry. The depth at each pixel is computed sparsely across the output image and then interpolated to produce pleasing results in real-time from a small set of silhouettes.

For our work, we use a volumetric shape-from-silhouette algorithm that uses the 2D distance fields of the input silhouettes and a simple marching cubes based polygonizer.

Chapter 4

The Wall of Inspiration

Choosing a surface coating is a form of color selection. Ashby and Johnson devote their entire text to the task of material selection in design and surface appearance is a part of that selection process. Applying a coating to a material can dramatically affect or even completely change the appearance of that material. Thus color selection and surface coating selection are part of the industrial design process. The *Wall of Inspiration* project was a joint research effort between the University of Minnesota's Digital Design Consortium and Benjamin Moore Paints and was undertaken with the intention of creating a digital color selection tool. This endeavor centered around the collection of surface coatings available from Benjamin Moore as a simple material collection. This collection is often presented as a deck of color cards and so the project worked to digitize this deck. Care was taken to insure that more than just color, but also appearance properties such as gloss and metallic effects, were presented. As such, this tool represents a very simple digital material collection. Emphasis was placed on the ability to navigate and select materials from the collection which was implemented in one of the tools in the Wall of Inspiration called *the Color Navigator*. The Color Navigator illustrates how a digital material collection could enable superior interfaces for searching and exploring.

In this chapter we will examine the Color Navigator in detail both as a means of interacting with a digital color collection and as a simple digital design tool. We attempt to carefully examine the problem that the navigator is trying to solve and undertake a user study that evaluates

Parts of this chapter have been previously published by the author in [Berrier et al., 2011] and [Berrier et al., 2008b].

whether or not it successfully addresses that problem. We will also look briefly at the larger project, the Wall of Inspiration, which represents a significant attempt by a major paint manufacturer to provide useful digital tools to its customers who are executing a common design task (perhaps without even being aware). Finally, we look back at this work in the context of the larger goal of this thesis and evaluate its generality and success as a digital tool for appearance design.

4.1 Introduction

In this chapter we present a new, computer-based interface for examining and searching through collections of paint colors, one that acknowledges the existing tools (color cards) and connects back to them physically. This new system, called the Color Navigator, is rooted in perceptual color organization and affords simple, direct 3D interaction. We developed this tool as an alternative to the standard fan decks of color cards used by almost all major paint manufacturers and we evaluated its effectiveness in comparison to these decks. We found that the Color Navigator can be faster than the fan deck and received a favorable evaluation by users over this traditional tool.

4.2 Background

4.2.1 Current Paint Selection

For quality control purposes, most paint manufacturers will pre-formulate all the colors that are achievable with their set of bases and tints. Tints are combined in certain proportions to achieve different pure hues and are added to the bases in increasing amounts to create shades of that color. These related shades are the basis for multi-color cards which present a single 'hue' (roughly speaking) in several levels of lightness, attributable to increasing amounts of tint. These multi-color cards are often presented on large racks (see Figure 4.1), or bound together in a single deck of cards that can fan out and be easily transported (see Figure 4.3).

It is informative to plot the colors for a single fan deck card in CIE L*a*b* color space (and its polar equivalent, CIE LCH_{ab}). For more on these colors spaces, see Judd and Wyszecki [1963]; Wyszecki and Stiles [1967]. The locus for these colors is not a straight line and the

Figure 4.1: Searching through the color cards in a display rack.

colors do not have a constant L* value, a constant chroma, or a constant hue. This means that CIE Lightness, Chroma, and Hue are all changing simultaneously on a single card (see Figure 4.2). This curved trajectory is the result of the tinting process and, while each color card follows a different path, the example shown is typical of all fan deck cards.

4.2.2 The Wall of Inspiration

The color navigator was only one module in a larger suite of tools all designed to help with the color selection process. This suite was paired with a large display system that was custom designed from off the shelf components for setup and use in a paint store. The full system of hardware and software was called *The Wall of Inspiration* (see Figure 4.4).

The software for the Wall of Inspiration accurately simulates different gloss levels and metallic reflection properties using existing techniques in computer graphics. These special

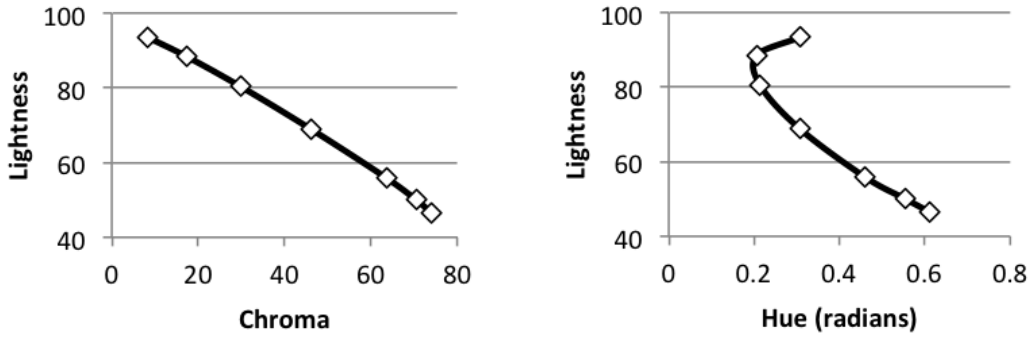


Figure 4.2: A single color card plotting CIE Lightness against CIE Chroma (left) and CIE Lightness against CIE Hue (right) (based on CIE L*a*b* space). Note how the card's locus changes in all three dimensions at once.

Figure 4.3: A pair of color card fan decks and some example multi-color cards showing multiple shades of a single hue. Note the similar hues separated by physical distance in the fan decks.

Figure 4.4: The Room of Inspiration which houses the Wall of Inspiration

appearance properties are rendered on three-dimensional objects illuminated by photographically captured environments. These environments can range from a room with professionally designed studio lighting to large day-lit outdoor spaces to the exact room in which the subject is using the software. To evaluate the appearance of the paint we have provided a representative set of three-dimensional objects including a piece of crown molding, a door, a wooden chair, a ceiling medallion and even a radiator. We harness modern graphics hardware to generate the appearance effects and manipulate the objects in real-time at fully interactive speeds. Our system has different environments available for reflection calculations as well as different sets of fine-tuned RGB values that have been selected for a particular monitor and illuminant.

The Wall of Inspiration software suite offered three different modules: the 3D Color Viewer, the Color Navigator and the Studio Finishes Viewer.

The 3D Color Viewer

To perceive the detailed reflective properties of a paint it is helpful to view it on an object with complex geometry. This is particularly the case when evaluating metallic or glossy reflections. A module called *The 3D Color Viewer* was created in our system for this task (see Figure 4.5). This module provides a library of common household objects. Any paint color from our database of pre-sampled colors can be placed on the surface of these objects. In addition, all of these paints are available in several gloss levels ranging from matte to high-gloss. The 3D Color Viewer is able to simulate five distinct gloss levels for each paint color (see Figure 4.6). This module is an attempt to overcome some of the problems inherent in the small chips of color that are available from the manufacturer which are inherently flat and lacking detailed geometry and which are only available in a single gloss finish.

Our system also has the ability to visualize certain goniochromatic colors; particularly

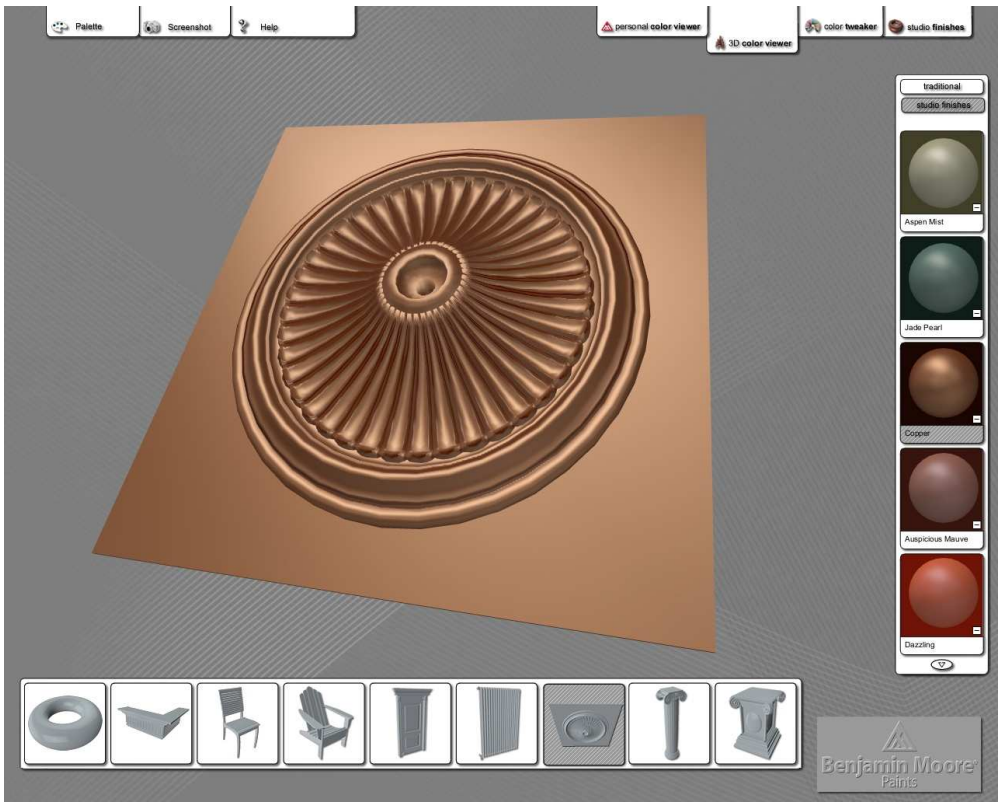


Figure 4.5: The 3D Color Viewer Module

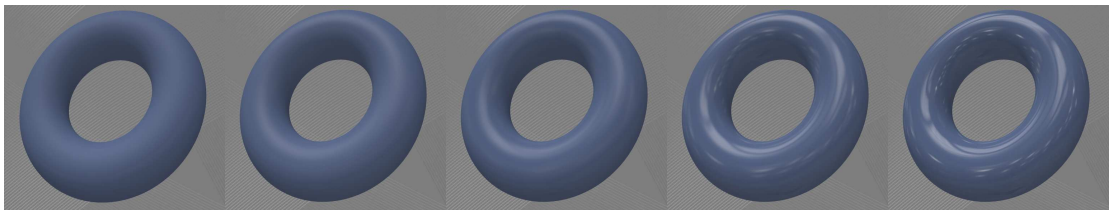


Figure 4.6: Gloss levels - matte, eggshell, perl, semi-gloss & high gloss

metallic and pearlescent tints which are becoming increasingly more popular in consumer interior design applications. These paints possess special reflective properties that cause their hue and chroma to shift slightly as the viewing angle changes. It is very difficult to see this shift on a flat chip lacking detailed geometry. By placing these colors on complex objects with curvature and fine detail, the unique reflective properties of these colors are much easier to perceive then when viewed on a flat paint chip. The ceiling medallion in Figure 4.5 is rendered with a copper version of this paint.

The 3D Color Viewer also allows the user to grab the object and rotate it in real time. This enables them to explore the full range of reflectance angles and lighting across the surface. Moving the object in front of a light source is vital for evaluating glossy finishes and metallic paints.

The Studio Finishes

A popular trend among household paint is to offer special latex based finishes that possess metallic and pearlescent reflective properties. One such product is the set of metallic colors in the Studio Finishes® collection from Benjamin Moore©. This is a relatively small collection of tints as are products offered by most household paint manufacturers that possess metallic or pearlescent qualities. Consequently, the set composed entirely of these colors will be much smaller than the full color database. Furthermore, it is impossible with standard color models and visual difference metrics to quantify simple perceptual relationships between these colors because they do not posses a single dominating tristimulus value. Because of this, *The Studio Finishes®* module uses a simple two dimensional grid layout (see Figure 4.7). This module allows all of the metallic and pearlescent colors to be viewed at once (presently using the aforementioned metallic and pearlescent tints from the Benjamin Moore© Studio Finishes®

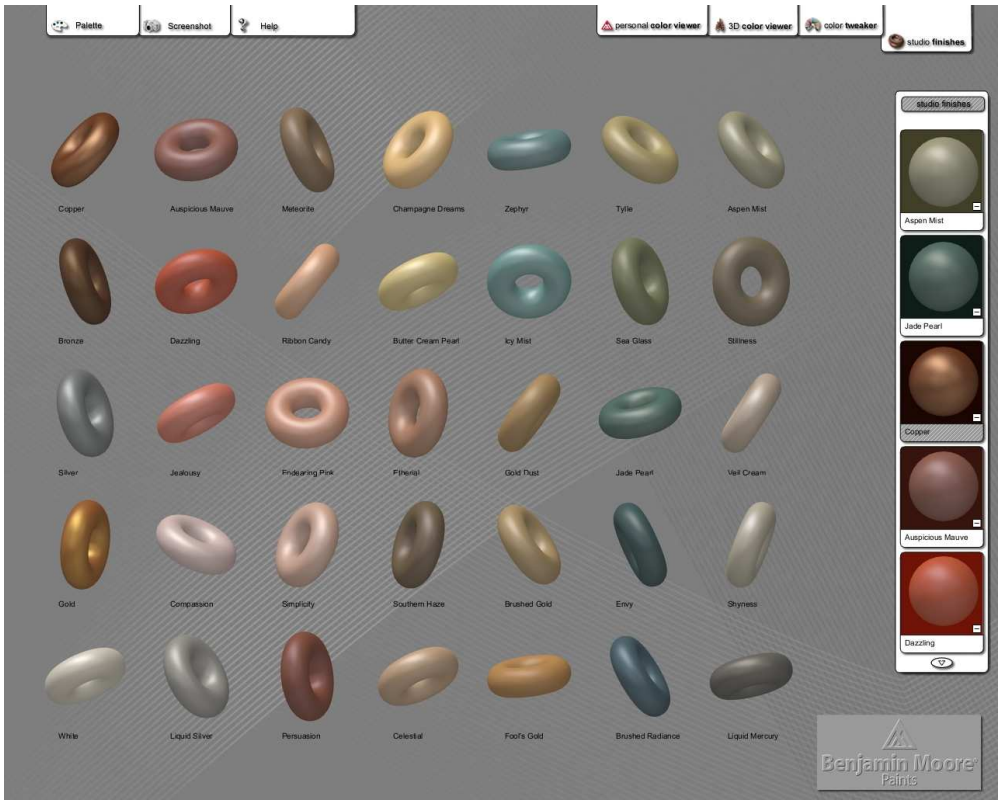


Figure 4.7: The Studio Finishes Navigator Module

collection). It is also very important for the user to be able to evaluate the complex reflective properties of these paints during the selection process. When these paints are placed on at objects (such as the square paints chips available in any paint store) much of their appeal is lost. To accommodate this, the system has the ability to place each color on a three dimensional object. At present each color is shown on a torus (a donut shaped ring) though the system has the ability to use any 3D object. The torus has the advantage that any given view of the object creates a multitude of different angles between the surface and the viewer.

4.2.3 Problem Statement and Requirements

After talking with experts in the paint industry and interior designers helping customers in a paint store, the following problem was identified:

> Given a collection of thousands of color choices, what is the best way to locate

individual colors that are good candidates to match a specific or abstract target?

Here, the 'target' may be either a real artifact with some or all of the visual properties desired or just an abstract concept of a color that a customer has in mind.

This problem led to the following requirements for a successful interface:

Similar Proximity *Colors that are similar to each other should be physically close to one another.*

This is a natural requirement for any search problem. Because of the non-perceptual nature of the color cards, this requirement is not met in the fan deck or color card rack where multiple, separate groups of related colors are noticeable (see Figure 4.3). This could lead the user to miss some good candidates for their target.

Color Perception *The color collection should map to the commonly accepted perceptual color dimensions of hue, chroma, and lightness*

This requirement should further ease searching as it leverages an instinctive perceptual structure. The color cards give the appearance of changing mostly in lightness but this is not the case. This can mislead a naïve user as these multi-dimensional perceptual changes could affect their perception of color similarity.

Direct Comparison *To facilitate direct comparisons with a target based on an artifact, the interface should make it possible to place the artifact next to the colors in the color collection.*

While it is easy to compare an artifact with a single color or a color card, it is difficult, if not impossible, to determine the artifact's relation to the collection as a whole. This is an important concept as customers often envision a target color that is related to but slightly different from the artifact chosen. Searching for proper alternatives requires a broader context than a single color card.

Current Tools *In support of users already familiar with the current tools, the interface must not completely abandon the traditional color cards.*

Given the financial investment that paint companies have already made in their fan decks, simply rearranging the colors on the existing cards was not an option. In addition, those working in the paint industry emphasized their experience and comfort with the cards.

4.2.4 Color Selection and 3D Interaction

The problem statement and requirements led to the development of a 3D computer graphic interface. Due to the requirement to incorporate current tools the cards themselves had to be part of the final solution. However, supporting color perception suggested the use of a color space with familiar perceptual dimensions such as CIE LCH_{ab} space. While many color selection interfaces have been created using 2D projections of three dimensional hue, saturation, and brightness color spaces (see Douglas and Kirkpatrick [1999]; Joblove and Greenberg [1978]; Smith [1978]), this did not meet our needs because each color card varied simultaneously in all three perceptual color dimensions. The cards could not easily be unwrapped, separated, and unambiguously projected onto a 2D surface.

In developing our interface we followed the guidelines of Bier [1987]; Chen et al. [1988]; Strauss et al. [2002] for developing 3D interaction techniques that are similar to existing techniques in 3D CAD and modeling tools.

4.3 The Color Navigator

In this section, we describe our alternative color search tool called the Color Navigator; an interactive visualization of the color card fan deck.

4.3.1 Visualizing the Fan Deck

The Color Navigator is a visualization of the fan deck in CIE LCH_{ab} space (see Figure 4.8). Each card is plotted as a strand with separate squares of color along it so that it looks like a multi-color card. The strands are created by fitting a curve to the LCH values of the colors on each card. We chose a 2nd degree polynomial fit as it does a good job of preserving the location of the colors in CIE LCH_{ab} space while eliminating the oscillations and high order curvature

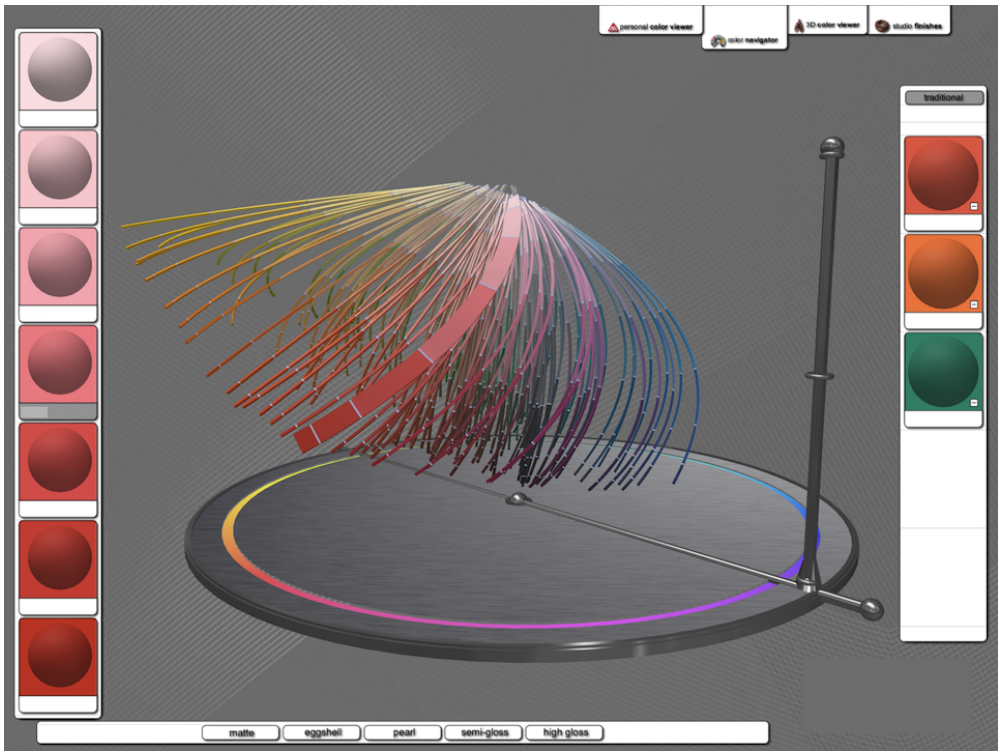


Figure 4.8: The Color Navigator interface. In the center is the color platform with the currently selected card (larger) shown to the left and the favorites palette to the right.

found in other fittings. The strands are placed above a circular platform with an embedded hue rainbow. This grounds the visualization and guides the user to a particular point in the spectrum.

To support the direct comparison requirement, our system can plot additional colors provided in the LCH representation (available from in-store scanners). This scanned color is drawn as a small sphere or, for better comparison with the cards, a flat chip (see Figures 4.10 and 4.11). The sample color is placed at the coordinates indicated by it's LCH values putting the sample directly in the context of the full color collection where physical distance will be a good approximation of color similarity. The scanned color is drawn sitting on top of a small pedestal that extends down to the circular platform. This helps the user perceive the location of the object along the projected depth dimension, something that can be difficult without the help of expensive stereoscopic rendering or other depth cues like shadows that would adversely interfere with perception of the colors.

As with any color seen on a display device or observed under a viewing illuminant, there are

colorimetric and environmental influences that must be addressed. For the computer display, a neutral background with an abstract design was chosen so that it would contrast with the visualization structurally but not in color, reducing the effects of simultaneous contrast. A commercially available calibration device was used to measure monitor chromaticity and map the collection to the specific monitor in use. For this initial work, the calibration of the monitors and the illumination of the fan deck were held constant to minimize their influence and reduce the complexity of the system and the user study. While different illuminants will change the CIE L*a*b* coordinates of the colors and affect our visualization, we chose to use a single illuminant (D65 daylight) for this work and leave the importance of this factor for future study.

4.3.2 Interaction

The color cards initially appear thin with a square cross-section to minimize obstruction of interior cards. When the mouse is above a color card it expands to a rectangular cross-section similar to the dimensions of a physical card from the fan deck. The card under the mouse is also drawn as part of the interface overlay to the left side of the color platform. On this larger card at the side, the current color under the mouse is highlighted and tracks the user's movement. To the right of the color platform is an area where the user can add and remove colors for consideration separate from the collection. We call this the favorites palette.

To control the view of the color platform we provide simple, constrained controls for rotation, zoom and translation. Rotation behaves like a 'turntable' spinning the color platform around its central L axis by dragging with the left mouse button. The camera can be zoomed in and out with the mouse's scroll wheel or by dragging with the center mouse button. The user can also shift the color platform around the screen by dragging with the right mouse button.

To support the task in our user study, we allow the user to move the color target pedestal. It moves in the C and H dimensions by grabbing the pedestal and dragging it around the platform. It can be moved along the L dimension by grabbing the color at the top of the pedestal and dragging up or down.

4.3.3 Data Obstruction and Clutter

To deal with obstruction and clutter in the visualization, the system provides a novel set of controls collectively called *the basket* (see Figure 4.9).

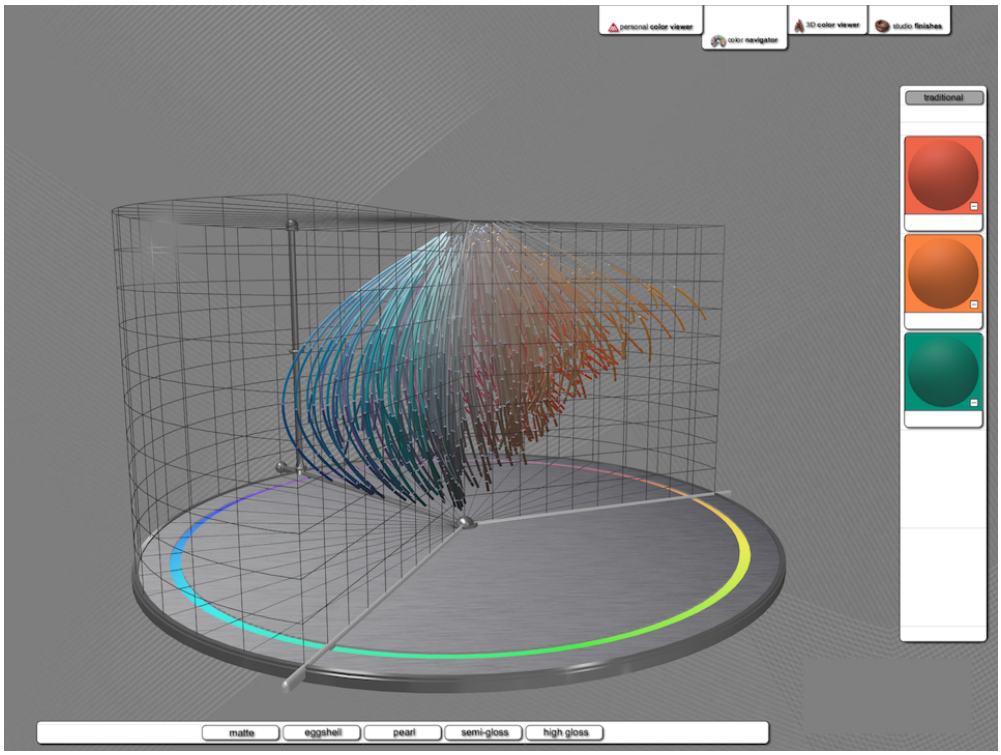


Figure 4.9: Using the basket with a very crowded collection that obscures interior colors and looks cluttered.

The basket defines limits in each of the perceptual dimensions of Lightness, Chroma and Hue. Any cards that are inside those limits are drawn and any that are outside are hidden. If a card is partially inside the basket, only the colors that fall inside are drawn while the rest of the card is indicated by a thin grey line. This provides some context to the shape and location of the entire card to the user.

Manipulators are provided for setting each of the basket limits (see Figure 4.11). These controls sit on the platform itself (or for the lightness dimension, on a pillar rising out of the platform) and are adjusted by directly grabbing and dragging them around the space. Whenever the user is adjusting the limits a wire grid is drawn to visualize the basket volume providing instant and continuous feedback.

Adjusting the basket affords hiding of the exterior strands and access to interior colors mitigating the problem of obstruction. The user can also eliminate colors that are of no interest and decrease the volume of information on the screen to afford zooming into a small area where

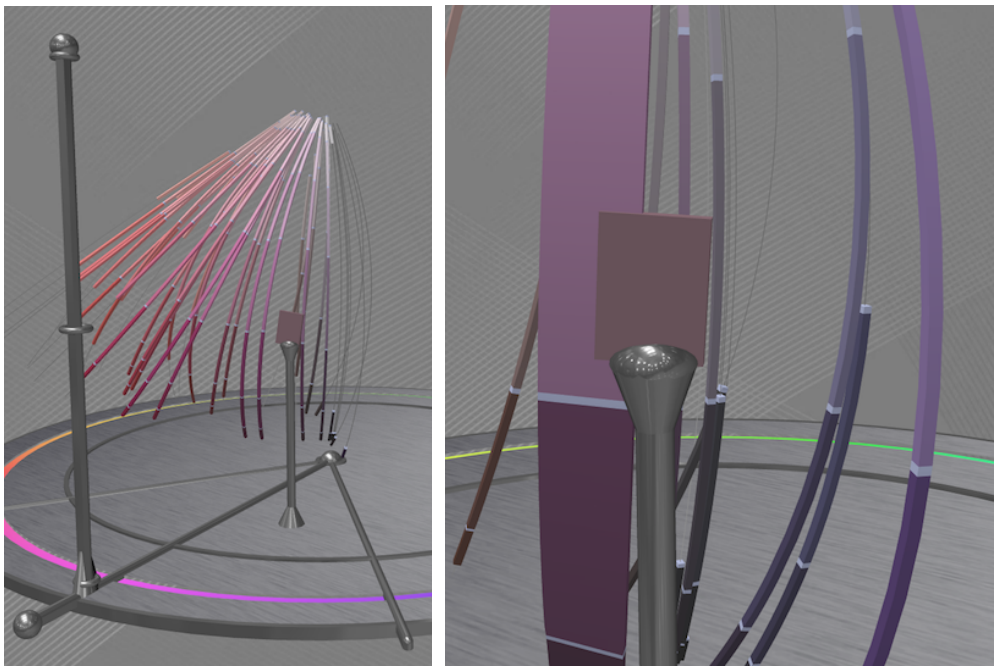


Figure 4.10: The Color Navigator displaying the target search color for the multi-color search task.

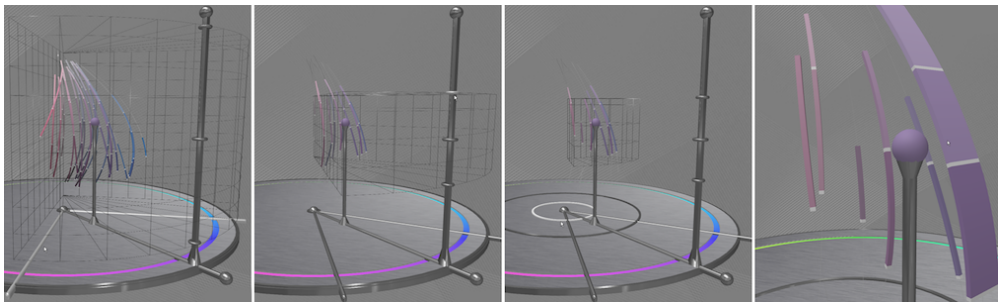


Figure 4.11: Using the basket controls to isolate a small precise section of color for comparison. The sphere represents a target color (possibly scanned). From left to right: Limiting hue, lightness and chroma then zoomed in close.

colors can be critically examined and compared.

All interaction with the color navigator and the basket is consistent with the guidelines laid out by John Schrag in Strauss et al. [2002].

4.4 User Studies

To evaluate the effectiveness of our interface versus the traditional fan deck we conducted two user studies; one without any training in color perception or 3D interaction and the second with training in both. For both studies we were testing the following hypotheses:

- The Color Navigator can be used faster than the fan deck for search tasks.
- Users will choose colors using the Color Navigator that are at least as similar to the target as those chosen with the fan deck.

For the second hypothesis we compute 'similarity' using euclidian distance in CIE L*a*b* space; a simple form of the Delta E metric (CIE76, ΔE^*_{ab}) where a value of 2.3 corresponds to a just noticeable difference Sharma [2002].

4.4.1 Search Task

With a real-world color search process in mind, we defined the following task (we call it the multi-color search task):

> The subject is given a target color (not present in their collection) and asked to search for and identify the five colors that are as similar as possible to the target color.

Participants using the fan deck were given a color card containing a single, unidentified color as the target and provided pen and paper for note taking and to indicate their answer. Participants using the Color Navigator were shown the target as a chip on top of the color target pedestal (see Figure 4.10). They were instructed to utilize the favorites palette to 'take notes' and indicate their final answer. Each subject performed three to four iterations of this task. We recorded the time it took to complete the task and the five colors identified in the search.

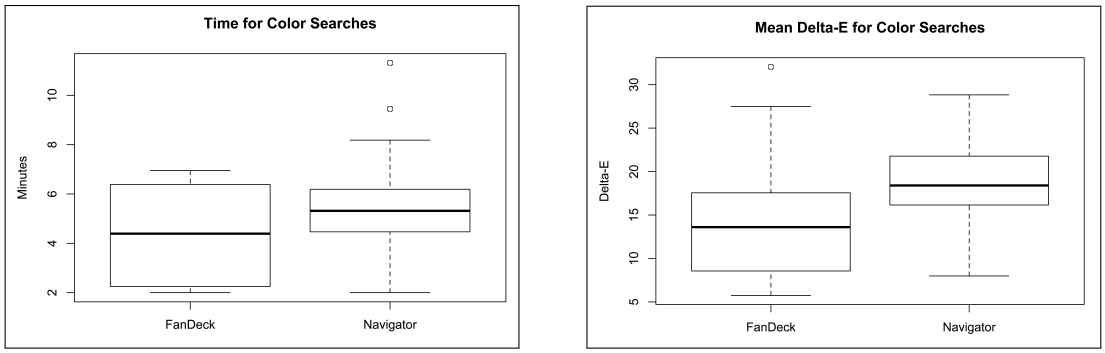


Figure 4.12: Box and whisker plots of the 5 number summary statistics (minimum, first quartile, median, third quartile and maximum) for completion times (left) and accuracies (right) in the first study. The circles are outliers and are beyond 1.5 times the inner quartile range.

Subjects using the fan deck were given a brief introduction to its organization and allowed to experiment with the deck prior to starting the task. Subjects using the Color Navigator were shown all the interaction available to set their view and control the basket and were given some simple tasks to perform to encourage using all these controls prior to starting trials. They were also shown the color target pedestal and its controls and allowed several preliminary task trials to ensure they were comfortable with the interface.

All subjects were screened for color blindness using the color plates from the Dvorine Testing Charts Dvorine [1944].

4.4.2 Tests Performed Without Extra Training

Over the course of two Saturdays, 32 subjects were recruited (30 female) from a local paint store either via their newsletter or on the store floor. Only 2 subjects identified themselves as being designers or in the design field. Only 1 participant reported prior experience with 3D computer interfaces. Each subject used only one of the two tools (either the fan deck or the Color Navigator). The participants were divided evenly among these tools. Subjects were not compensated.

Results

Independent samples t-tests were performed to compare both time and accuracy using the fan deck versus the Color Navigator. We looked at each of three different color targets separately

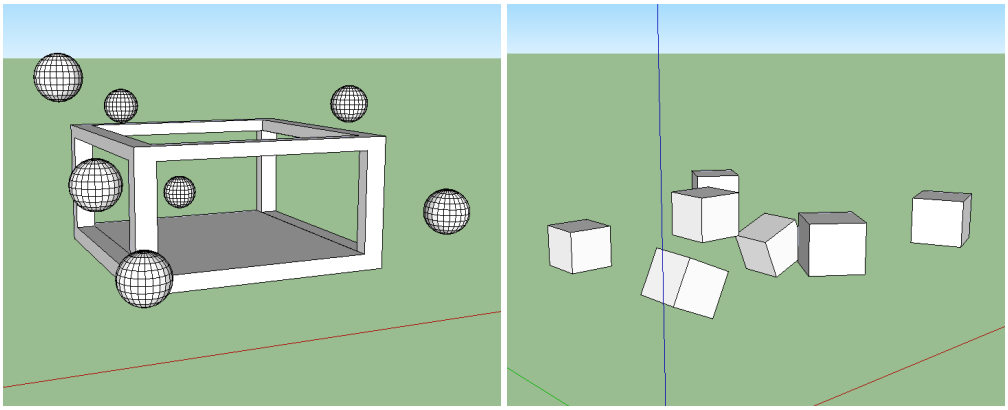


Figure 4.13: The object movement (top) and rotation (bottom) training tasks using Google SketchUp.

and then with all trials pooled in one.

With respect to search time (in minutes), with the trials pooled there was significant difference in the scores for the fan deck ($M = 4.37$ mins, $SD = 1.925$) and the Color Navigator ($M = 5.51$ mins, $SD = 2.140$); $t(44.022) = -1.912$, $p = 0.0312$. Only one of the individual color targets had a significant difference in scores and showed similar results, the other two were not significant by themselves. This is a negative result that contradicts our first hypothesis and indicates that the fan deck was faster than the Color Navigator.

With respect to accuracy (in Delta E), with the trials pooled there was significant difference in the scores for the fan deck ($M = 14.39\ \Delta E$, $SD = 6.95$) and the Color Navigator ($M = 18.68\ \Delta E$, $SD = 5.37$); $t(43.079) = -2.377$, $p = 0.022$. Again, only one of the individual color targets had a significant difference in scores with similar results. This is also a negative result that contradicts our second hypothesis and indicates that the Color Navigator was not as accurate as the fan deck (the data summary shows that it was in fact less accurate). The data is summarized in Figure 4.12.

4.4.3 Tests Performed With Extra Training

Over the course of one week, 10 subjects were recruited (2 female) from our university using signs and interviews during passing periods. 7 subjects reported some significant prior 3D interaction experience. 2 subjects reported working in a design discipline (interior design and architecture). Unlike the previous study, all users were asked to perform the task using both

the fan deck and the Color Navigator. To eliminate ordering effects, half the participants used the deck first while the other half started with the Color Navigator. Subjects were compensated with a $15 gift card.

Extra Training

In this study, all subjects were trained in the following areas:

- Munsell's color system and perceptual color.
- general 3D interaction using Google SketchUp.
- the Color Navigator interface controls.
- the relationship between Munsell's system and the Color Navigator.

The first phase of training was done with the Munsell Student Color Set Long and Luke [2001]. Subjects were asked to reconstruct the initial hue, value and chroma page using the provided paint chips as well as the value and chroma page for the 5BG hue. This initial color training was done immediately after screening for color blindness and prior to any trials. All other training was only done prior to searching with the Color Navigator.

The second phase of training consisted of two simple tasks using Google SketchUp (see Figure 4.13). It allowed us both to train the subjects that were not as familiar in 3D interaction concepts and to assess each subject's familiarity with a 3D interface in general. They were trained in 3D object movement and rotation.

Results

The same t-tests as the first study were performed to compare time and accuracy. However, this time, since both samples were the same, we used a paired samples t-test.

With respect to search time (in minutes), with the trials pooled there was significant difference in the scores for the fan deck ($M = 4.41$ mins, $SD = 1.724$) and the Color Navigator ($M = 3.20$ mins, $SD = 1.46$); $t(29) = 3.8404$, $p = 0.000308$. All but one of the individual color targets also had a significant difference in scores and showed similar results. This is a positive result that agrees with our first hypothesis and indicates that the fan deck was slower than the Color Navigator.

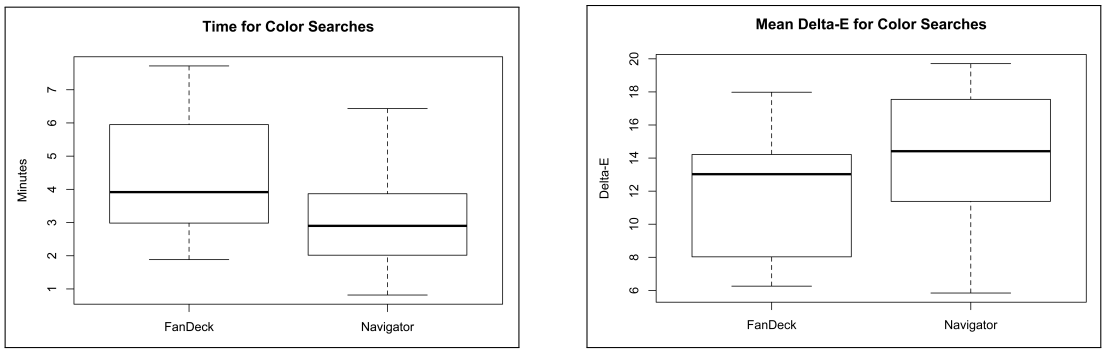


Figure 4.14: Box and whisker plots of the 5 number summary statistics for completion times (left) and accuracies (right) in the second user study.

With respect to accuracy (in Delta E), with the trials pooled there was not a significant difference in the scores for the fan deck ($M = 12.140\ \Delta E$, $SD = 3.880$) and the Color Navigator ($M = 13.653\ \Delta E$, $SD = 4.538$); $t(29) = -1.313$, $p = 0.1995$. Also, two of the individual color targets did not have a significant difference in scores (although one of the more difficult trials, 'H,' did). This is also a positive result that supports our second hypothesis and indicates that the Color Navigator was at least as accurate as the fan deck. The data is summarized in Figure 4.14. Complete t-test results are in Table 4.1.

4.4.4 Discussion

The different results between the two studies show that successful use of the Color Navigator requires facility with 3D interfaces and knowledge of traditional LCH color organization schemes. When these two conditions were met our hypotheses were confirmed. Observing the subjects as they performed the extra training tasks (and analysis of the times that were recorded for these tasks) showed considerable variance. Some subjects found one or both of these tasks easy while others struggled. This suggests that the new training regimen was beneficial in overcoming these confounding factors.

The time savings provided by the Color Navigator would be beneficial to individuals who make a significant number of color selections throughout the week (such as design professionals and paint store color stylists). These professionals may be willing to invest the effort necessary to learn how to use a 3D interface (if they don't already know how to do so), and are likely to already understand LCH color organization schemes.

	t-score	*p-value*	*Mean of Diffs.*
All	$t(29) = 3.8404$	0.0003	1.24
'B'	$t(9) = 1.6106$	0.0709	0.60
'H'	$t(9) = 4.4237$	0.0008	1.81
'F'	$t(9) = 1.9208$	0.04347	1.21
All	$t(29) = -1.313$	0.1995	-1.51
'B'	$t(9) = -1.3361$	0.2143	-1.44
'H'	$t(9) = -3.8517$	0.0039	-3.05
'F'	$t(9) = -0.0621$	0.9519	-0.05

Table 4.1: Data from the paired t-tests comparing time (top) and accuracy (bottom) in the second study.

It is worth noting that in the study with extra training, both of the two design students who participated had prior experience with SketchUp and had previously taken a color theory course that utilized the Munsell Student Color Set Long and Luke [2001] in some capacity. This suggests that today's design professionals may already have the experience necessary to effectively interact with the Color Navigator.

4.5 Conclusions and Future Work

The Color Navigator is an interface that appears to be useful to a specialized audience. It could save time for interior designers or architects who must make hundreds of color selections throughout a typical month. This group is also likely to use other computer graphic tools to visualize their interior spaces and buildings. The Color Navigator allows them to put the color fan deck aside just as computer based rendering systems permit them to avoid the use of drafting tools. It is another step along the way to virtualizing the design process in traditional fields of design.

While it was not measured in any of our experiments, it should also be pointed out that the Color Navigator made a favorable impression on almost every user. It was described as more compelling and fun than the fan deck and 90% of participants in the second study said they would use the Color Navigator again if given the option, either by itself or in conjunction with the fan deck.

4.6 The Wall of Inspiration as a Tool for Design

As mentioned earlier, the collection of Benjamin Moore colors represents a very simple, existing material collection and the Wall of Inspiration software digitizes that collection. Therefore, we can consider it to be a simple design tool to aid in the industrial design process. We had a unique opportunity to observe designers using the Wall of Inspiration software and in doing so gained an appreciation for what a digital design tool must do to ensure the design community will be willing to accept it.

First and foremost, it must build on existing tools that designers have trained to use and with which they have extensive experience. An earlier iteration of the Color Navigator (named the Color Tweaker) visualized the collection as spheres floating in L*a*b* color space. It was largely rejected due to its pure color science approach to visualizing the collection and its striking similarity to the balls and sticks used to visualize molecules in chemistry rather than color cards or the fan deck. The subsequent Color Navigator attempted to address this by basing the visualization on the existing color cards rather than purely on the L*a*b* values of the colors.

Second, the software must treat the user interface as carefully as any engine used for rendering or any reflection calculations. While we as computer graphics specialists appreciate a well thought out reflection model or the delicate subtleties of such phenomena as the Fresnel effect or caustics, the power of these simulations is lost next to a poorly conceived user interface. It is more important that the designer be able to control the system and work in a manner to which they are accustomed than to perfect the visual accuracy of the rendering. We had our assumptions about the intended audience for this software challenged when we tried to evaluate it in a user study. The results of the first study showed that despite being grounded in perception, color spaces like L*a*b* do not afford any useful structure to an untrained user. Similarly, a bare bones 3D interface that is extremely natural for a computer scientist to use will be quite challenging for the average user. These preconceptions (which are natural from the point-of-view of a researcher) can hurt the utility of any user interface that must be accessible outside our world.

Lastly, the hardware used to display or capture colors must be understood as imperfect. While colorimetry and display science has made tremendous advancements in recent decades it has not reached the point that a display device can stand next to a physical sample or even another display and attain a match in perceived color. This is particularly the case when dealing

with tasks that place importance in accurate color reproduction, as design tasks do. While we may be confident that the latest display technologies are approaching this problem much better than in the past, a designer that is familiar with physical samples and trained to recognize subtle color differences will not be impressed by the accuracy of current technology.

While the hardware used in the Wall of Inspiration project was not successful the software did achieve a level of acceptance. It was in use in a commercial paint store in Manhattan for more than a year and is now being tested in other environments with the hope that it can become an integral part of the Benjamin Moore in-store experience. The process of working with a corporate partnership proved to be a difficult but worth-while experience. The reception of our software by both color scientists and interior decorators and designers working in the industry helped codify a different emphasis on the software development task than is typical in research. While the hardware display devices never performed at the necessary level, the software was versatile enough to be adapted to many other display configurations and this flexibility did not come by accident. All of these lessons from both the successes and failures of the Wall of Inspiration project are relevant as we move forward with digitizing the aesthetic design process.

Chapter 5

The Cloak Wall

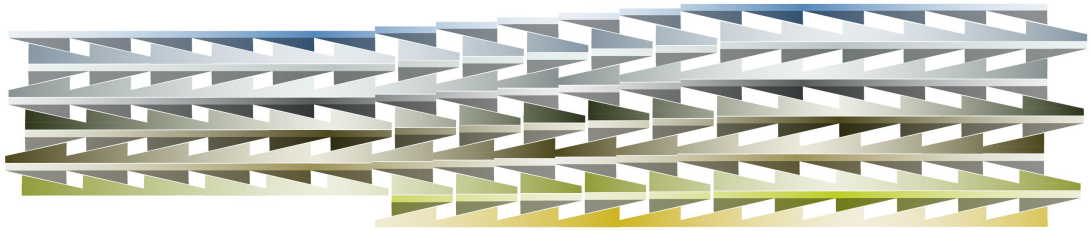
The final appearance of most designs does not consist of a single solid color. Instead we see pleasing patterns and sequences of colors; palettes chosen to give a sense of harmony to the design. There are many standard practices and tools for developing these palettes and sequences ranging from the color wheel of Johannes Itten [Itten and Birren, 1970] to books filled with hundreds of pre-selected palettes of two to four colors. We set out to extend this idea beyond a solid color to materials with more interesting properties by creating a simple design tool for producing sequences of metallic colors. Based on an industry standard practice of modeling metallic colors with five aspecular angle measurements we used interpolation to assist in developing a pleasing arrangement of metallic colors across a unique modular wall system. In this chapter we describe this project and the algorithm we developed for interpolating metallic colors.

5.1 Introduction

Creating a sequence of related solid colors is a natural, straight-forward task. We have an intuition about it and when that intuition fails we can appeal to well defined mathematical techniques such as interpolation to help. However, creating a sequence of related metallic colors is a more difficult task. Our intuition on the matter may be non-existent or misleading. Even expressing a metallic color is difficult as their qualities go beyond a single triple. They cannot be

Portions of this chapter were previously published by the author in [Berrier et al., 2008a].

(a) Intended site of Cloak House



(b) Illustration of Cloak Wall

Figure 5.1: The Cloak Wall - The key colors in the color scheme come directly from this photograph of the intended site.

constructed from three primaries alone. Consequently, they are typically used as accent colors. They form borders or trims in facades or appear only when the raw material itself is a metal.

Appealing to a standard expressing metallic colors as five angular measurements we will discuss in detail the task of creating a sequence of metallic colors by using the technique of interpolation and by extending it to a multi-dimensional domain for these complex surface coatings. We finish by discussing how an interpolation tool was used to design a complex color scheme for a prototype architectural wall.

5.2 The Cloak Wall

Swackhamer and Satterfield [2007] developed *The Cloak Wall* as a modular, energy-efficient and affordable housing project. While many other types of building materials and elements have

been considered throughout the lifespan of the project the current form of the project is shown in Figure 5.1(b). The exterior is composed of modular bricks with many desirable properties each constructed of sheet metal. The colors used on the wall should come directly from the site where the Cloak House would eventually be built (see Figure 5.1(a)) helping it to blend in with its surroundings and even disappear (or *cloak*). The colors from this site were analyzed with an online tool [Watson, 2012] to produce a palette of solid colors. This became the basis of the color scheme and was used to design key metallic colors. It was decided that these key colors would be sequenced across the rows of the wall interpolating through a common neutral color similar to Figure 5.1(b).

5.3 Sequences of Metallic Colors

The mathematical process of interpolation is useful for creating sequences of solid colors. With two colors, A and B, described in the same color space, we can produce a sequence of colors of size n by creating a line between A and B and sampling that line at n evenly spaced points. This is called linear interpolation. If we work in a perceptually uniform space (such as CIE $L^*a^*b^*$ then those even spacings along the line will be perceptually even as well. If, for example, A is a very dark forest green and B is a very light lime green the line between them with change mainly in the lightness dimension. Sampling this line at even intervals will give us a color sequence that slowly changes in lightness from the dark green to the lighter green.

This same process can be used for metallic colors but it must first be extended to deal with the extra dimensionality of the viewing angle. Each metallic color is described by five $L^*a^*b^*$ values corresponding to the five aspecular angles. If we treat each of these five values as a single color then we can interpolate it as we would a solid color. By connecting each of these five values with a line and sampling those five lines at n regular intervals we get a sequence of $n+2$ sets of triples each of which describes its own metallic color. These sequences can range from lightness gradients (similar to the example given for green colors) to trips around the entire rainbow of hues and everything in between.

Interpolation of both solid and metallic colors is done in $L^*a^*b^*$ space so that the evenly spaced samples will produce perceptually uniform steps. Alternatively the values can be transformed from $L^*a^*b^*$ space to LCH so that interpolation occurs along the hue and chroma dimensions. Linear interpolation can still be performed in this space but care must be employed with

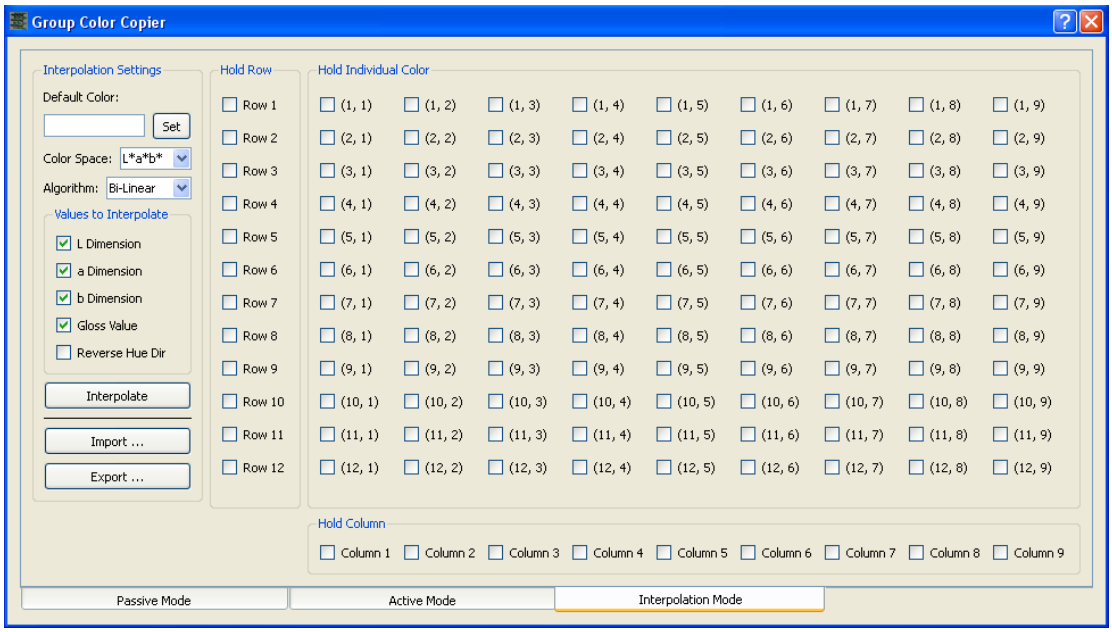


Figure 5.2: User interface for the interpolation tool. Individual bricks or entire rows and columns can be set as key colors for interpolation.

the hue dimension which is circular. Interpolation can move either clockwise or counterclockwise and the choice of direction will make a dramatic difference in the sequence produced. No single choice will be appropriate all the time so it should be left to the user to decide in which direction to proceed when interpolating in *LCH* space.

Metallic Sequence Tool

A tool was developed to help with interpolation of metallic colors across the Cloak Wall (see Figure 5.2). The tool presents a grid of checkboxes, one for each brick in the wall. To create a sequence, the user assigns colors to the key bricks that form the ends of the sequence. These colors can be selected from a catalog of existing colors or they can be custom designed using an interface described in [Meyer and Shimizu, 2005]. The user then checks off these bricks in the interface and clicks 'interpolate' to fill in the color sequence. The user has the option to select the abstract a and b dimensions for interpolation or the more natural hue and chroma dimensions. They can also specify the type of interpolation to be used by choosing from linear

and cosine interpolation. Cosine interpolation spaces the sequenced colors unevenly using a cosine function. If they are interpolating in *LCH* color space they can also reverse the direction around the hue circuit to produce a different sequence of colors. The Cloak Wall is roughly arranged in a row-column organization. Interpolation can occur in one dimension along a row; interpolation can also occur in between rows along columns effectively making this a bilinear interpolation scheme. Only rows and columns with two or more key colors checked off are interpolated; the rest are left alone.

Rendering the Cloak Wall

Throughout the color selection process, the architect worked with a real-time rendering of the painted wall along side the interpolation interface. This system performs all calculations in real-time by employing modern hardware shader technology permitting per-pixel calculation of the metallic reflection model. The angular dependence of this model necessitates per-pixel calculations as the color of the primary underlying pigment must be re-calculated at each point on the surface. Recent developments in computer graphics hardware have opened up this portion of the hardware rendering pipeline to allow a small program (called a shader) to be executed for each fragment/pixel on the surface of the object being rendered. The shader used for this program calculates the lighting and viewing angles along with the surface normal at each pixel and uses them to compute the color according to the curves fitted to the L, a and b dimensions. This color modulates the lighting calculation to give the surface a metallic appearance.

The latest version of this rendering engine uses environment maps instead of point light sources to compute the metallic model. This provides a superior effect when the surface has a high-gloss finish, however it was not as useful for our task as no clear-coat was applied to the bricks giving the coatings a matte finish. Regardless, it was known that a small portion of the wall would be exhibited in the Goldstein Museum of Design in St. Paul, Minnesota. To fully exploit the flexibility of the environment map rendering engine and visualize the wall in this gallery, we created environment maps of the Goldstein (see Figure 5.3). These environment maps were used for rendering and visualization throughout the design task along with several other ‘stock’ lighting situations with their own desirable characteristics.

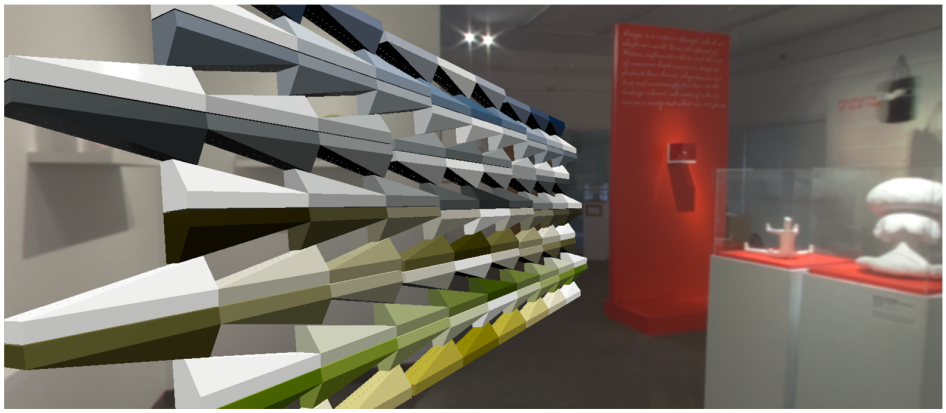


Figure 5.3: Rendering of the final design of the Cloak Wall. There are more columns and rows in this design than were used in the final exhibit due to space constraints.

Exhibiting the Cloak Wall

A unique opportunity became available to exhibit the cloak wall at a local design museum. For this a sub-section of the full prototyped wall pictured in Figure 5.4 was carefully finalized and the measurements were sent to Dupont. Dupont has developed a computerized system to match paints based on these measurements. Normally it is used in auto body shops for automotive paint repairs. In our case it was used to formulate new colors. Dupont calculated the formulations as well as the predicted measurements of the coatings which were as close to the requested measurements as possible but never equal. The predicted measurements were loaded back into our rendering system and problem areas were identified. In these cases work was done to change the colors and bring them closer to existing Dupont paints. A database of Dupont's SpectraMaster color library as well as a visualization system comparing designed colors to known colors was utilized in this step. Once the predicted measurements of the formulated coatings were found to be acceptable the paints were mixed and applied using standard automotive paint techniques. The final wall installed in its exhibit can be seen in Figure 5.5.

5.4 Conclusions

Computer aided tools will play an increasingly important role in the design and selection of surface finishes for architectural applications. Architects have been able to use computer graphics

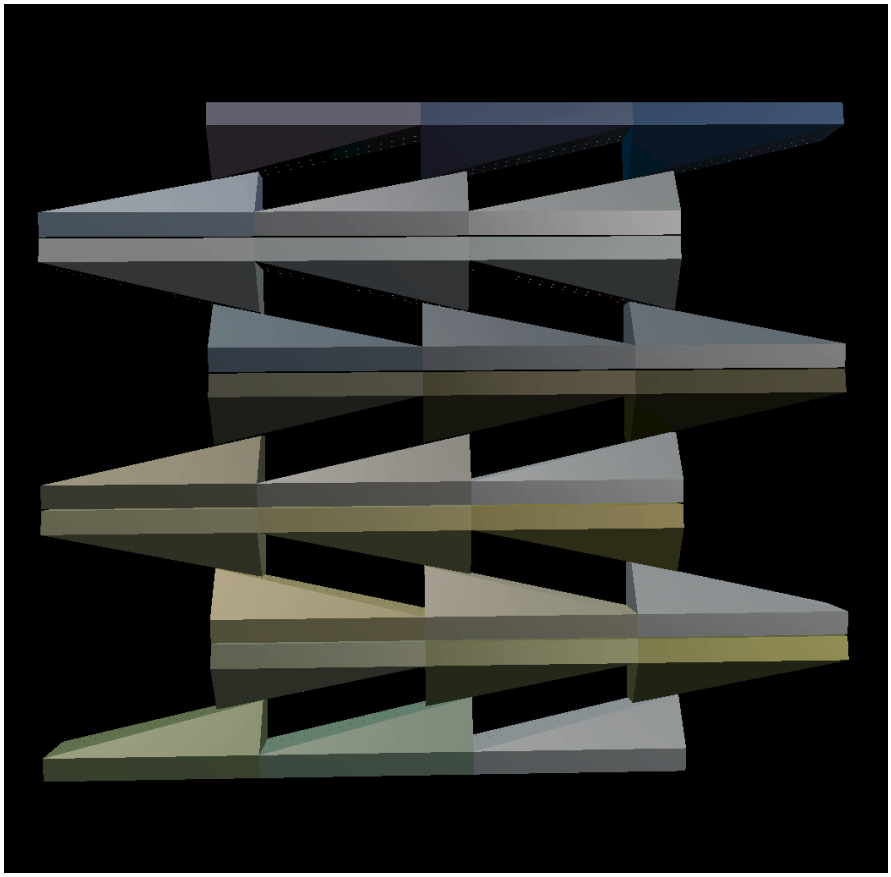


Figure 5.4: Subsection of the Cloak Wall for the Goldstein Museum showing the predicted appearance of the formulated coatings.

to manipulate the shape and form of their buildings for over forty years. Within the last twenty years advanced computer graphic illumination algorithms have made it possible to consider the effect of natural and artificial lighting on the appearance and functionality of a building. The work presented in this chapter demonstrates how recent developments in computer graphics hardware permit the reflectance properties of complex architectural coatings to be displayed *and* manipulated in real time. This capability allowed one and two dimensional sequences of metallic colors to be designed for use on a prototype architectural wall. The realistic display, iterative adjustment, and automatic formulation of these metallic paint colors demonstrates the basic steps necessary to achieve computer aided color appearance design in an architectural

application.

5.5 Metallic Interpolator as a Tool for Design

It is important that the range of techniques already employed by artists and designers remain viable in the context of general scanned materials. In this work we showed that something as simple as creating a sequence of colors becomes a considerably more complex task even when limited to just one type of material. Yet this is an important task for a designer in the early stages of the aesthetic design process. While the lead designer on the Cloak Wall project started to do this manually using the BRDF designer tools it was a painstakingly slow and imprecise task.

Figure 5.5: Prototype of the Cloak Wall exhibited in the Goldstein Museum of Design

The introduction of our interpolator insured even spacing of the colors and freed the designer to think about the key components of the design's overall palette rather than all the in between shades. This is the ultimate goal of any tool for industrial design; to free the designer from the tedium of careful craft and concentrate on their creative process. As we continue our work in digitizing industrial design we should focus on tools, like the metallic interpolator, that speed the design process and help make subjective tasks more precise.

Chapter 6

Morphing of Surface Light Fields

Figure 6.1: A morph between two surface light fields.

6.1 Introduction

Modern design studios are usually littered with material samples; stacked floor to ceiling, overflowing cabinets and bookshelves, everything from upholstery to paint swatches to marble, wood and textile samples. Sometimes these samples serve as the building blocks of a design being physically attached to mood or finish boards that express the early design concept. Other times they are a jumping off point in the creative process, the inspiration for ideation. Working from example is the modus operandi of the design process cannon. This part of the design process remains grounded in the physical world. In the absence of adequate digital representations of the myriad of design samples and examples there is little alternative to rooms filled with physical artifacts. Inevitably, these design libraries are largely incomplete particularly when it

comes to new, cutting edge products resulting from technological advances in manufacturing and materials engineering or the latest trends and patterns in the ever changing world of textiles and fashion. Furthermore, when these physical samples are the source of the final design they are often a key part of that design and must be integrated into the portfolio of the project consuming small amounts of the samples in the process. If the parties involved do not have the identical samples in their own library then the original swatches and cards must be physically traded and every person must subjectively evaluate the intent of that sample.

For all these reasons and more there is a need for digitization in the world of design; faithful digital representations of all these design samples and example artifacts. It can free up physical space in the studio, fill in the holes in a design library and keep it up-to-date on the latest options, and provide a viable, quantitative way to exchange the design intent based on these samples.

The emerging work in image based rendering offers an answer to this need. Using modern light field representations (such as those described by Buehler et al. [2001]; Chen et al. [2002]; Wood et al. [2000]) we can 'digitize' an object and reproduce it on the screen as if had been meticulously modeled by an expert computer artist. All of the subtleties of geometry and appearance and complex reflectance properties are faithfully reproduced (including inherent flaws in the object). They all behave correctly as you rotate the model and examine it virtually, and with proper sampling of the incident light field, the object can even be placed in other lighting environments with the click of a mouse. The physical material artifacts that designers know and trust become the starting point for this type of digital representation and it is not limited in the types of reflectance properties it can encode as are most fitted reflectance models.

This immediately offers solutions to the many problems posed above. But, being in the digital domain, it also makes completely new modifying and editing techniques possible. We can envision a similar kind of revolution that the photographer and illustrator felt with the introduction of Photoshop. Palettes of tools that would normally destroy a physical object can be provided that enable previously unheard of design of material appearance. Tools that have no physical counterpart could be realized that give a high level of control over reflectivity without having to worry about underlying BRDF properties and rigid physical constraints. In this chapter we will present one such technique that has no physical counterpart: morphing of physical appearance. By specifying two or more digital artifacts with desirable appearance characteristics, the designer can proceed to 'morph' between these artifacts and generate a new digital representation with properties from all of the targets. The resulting morphed object would be

rendered as a 3D object viewable from any angle with reflectance properties that behave properly as you rotate it. This is achieved by a new surface light field morphing technique designed around the unstructured lumigraph representation developed by Buehler et al. [2001].

For the remainder of this chapter we will briefly discuss some related work and then describe our morphing algorithm in detail. We will then describe how we implemented this algorithm with existing open technologies and what tradeoffs and simplifications were chosen. We will then demonstrate its effectiveness by computing a morph of two unstructured lumigraphs and draw some conclusions and discuss future directions of this work.

6.2 Related Work

We have already established the background of image based rendering and light fields in Chapter 3. In the context of this work, we want to highlight the three main representations of surface light fields that have shown particular effectiveness in the literature. First, there is the lumisphere representation developed by Wood et al. [2000]. This representation has explicit geometry on which encodings of reflectance output (one half of the BRDF) called lumispheres are located. This representation requires input imagery to be resampled and processed to generate the individual lumispheres and employs custom numerical methods to compute the final representation. The unstructured lumigraph representation was developed by Buehler et al. [2001] and seeks to provide a more natural way of rendering both surface light fields and geometry-free, two-plane light fields. By re-projecting the collection of appearance images onto proxy geometry, the unstructured lumigraph technique blends between all of the available images to choose the best representation of the view requested. The proxy geometry can be as simple as a flat plane or, for inward looking object light fields, can be the visual hull of the object. This latter approach makes it a surface light field representation. Unstructured lumigraphs are fast to render and easy to implement but less accurate than the lumisphere representation especially when the object is poorly sampled. Finally, the light field mapping technique proposed by Chen et al. [2002] offers something in-between lumispheres and unstructured lumigraphs. It parameterizes an explicit representation of the surface to lookup into a set of maps that are calculated from the appearance data instead of using the imagery directly. It avoids the complexity of the lumisphere but still requires a refactoring of the imagery by means of numerical analysis. We have chosen to use the unstructured lumigraph representation for our work due to its ease of

implementation and its direct use of the imagery that samples the object.

Surface light field morphing has already been demonstrated by Wang et al. [2005] for the light field mapping technique of Chen et al. [2002]. In this approach, an explicit light field is not computed in the in-between state of the morph. It is simply implied by a combination of the end targets. For design, it is important that the reflectance properties of any in-between representation be explicitly accessible so that they can be examined and communicated to other parts of the design process. Other light field morphing techniques [Jeong et al., 2003; Zhang et al., 2002] use the pure light field data and employ ray-warping techniques to achieve a reasonable warping. This direct manipulation of the light field data is undesirable and newer representations, as described above, could not easily be employed for morphing in ray space.

6.3 Morphing Algorithm

Our algorithm computes a morph between two or more surface light fields represented as unstructured lumigraphs. A morph is constructed for both the geometry and appearance data associated with the objects. The user must first establish sparse correspondence points on the surface of each object in 3D. These points are connected into a coarse mesh that is used to automatically construct a set of dense 2D correspondences, one for each view in the light field. Silhouette alpha-matte images for each view are used along with the dense 2D correspondence to construct a shape morph. Photographic imagery is used for a separate image morph. The resulting 2D morphs are executed concurrently to create a 'tweened' unstructured lumigraph with appearance and shape properties from all of the original targets. We will now discuss each of these steps in more detail.

6.3.1 Establishing the Correspondence

Creating a morph involves two key problems [Gomes et al., 1998]:

1. Pairing up points on the different targets that should remain matched as the objects distort (the correspondence problem). A *sparse* correspondence consists of a set of discrete points on all targets whereas a *dense* correspondence is continuously defined over the entire object.

2. Distorting the objects to the positions of the matching points on the other targets in a fluid and continuous manner (the warping problem).

A well-defined morph consists of a dense correspondence between the targets and a warp designed around that correspondence and defined for each individual target [Gomes et al., 1998]. Our work focuses on the correspondence problem as it pertains to an unstructured lumigraph and is outlined in Algorithm 6.1. We strike a balance between a purely manual correspondence and an automated matching algorithm by requiring the user to seed the key matched features on the objects, creating a sparse 3D correspondence. We then carefully propagate these features to the internal data structure of the unstructured lumigraphs to create a set of sparse 2D correspondences that are ultimately turned into dense correspondences.

Initially, the user must place six matched points that are roughly located at the apex, nadir and four points spaced around the equator of each target object. These six points are immediately connected by edges to form two pyramids stacked on their ends (see Figure 6.2). As the user matches more points on the targets the edges and/or triangles of this initial mesh are subdivided maintaining a rough triangle mesh over the surface of the object (see Figure 6.2). This mesh of matched points (P^3) and edges (E^3) is the *correspondence mesh* (M^3). This mesh can also be thought of as a graph (undirected and unweighted) with nodes at the matched points and the same edges connecting them (the *correspondence graph*, or G). There are several correspondence meshes in the morph (one for each target) but only one correspondence graph as it is only concerned with the adjacency of nodes and not their spatial location.

$l = (I_\alpha, I_{\text{photo}}, V)$	Light field with alpha-mattes, photos and views
$L = \{l_1, l_2 \dots l_n\}$	Set of target light fields
$P^3(l) = \{p_1^3, p_2^3 \dots p_x^3\}$	Matched points for target l
$E^3 = \{e_1^3, e_2^3 \dots e_y^3\}$	Edges connecting the matched points
N	Enumeration of E^3 for use with G
$M^3(l) = (P^3(l), E)$	3D Correspondence mesh for target l
$G = (N, E)$	Correspondence graph
$P^2(v)$	Projection of $M^3(l)$ to v for all targets l
$\chi(v)$	The silhouette-edge crossings from E^3 that are in v for all targets l
$C_1(v), C_2(v), C_3(v)$	Primary, secondary and tertiary 2D correspondences

Table 6.1: Symbols used in algorithm 6.1

Algorithm 6.1 Algorithm for establishing the sparse 2D correspondences

Require: All l share a common V

user creates initial sparse 3D correspondence, M^3 & G
repeat
  for all v in V **do**
    for all l in L **do**
      update $P^2(v)$ with projection of $M^3(l)$ to v
      update $\chi(v)$ with identified silhouette-edge crossings of l in v
    end for
    // Update Primary Correspondences
    for all p in $P^2(v)$ **do**
      if p is visible in all l **then**
        $C_1(v) \leftarrow C_1(v) \cup p$
      end if
    end for
    // Update Secondary and Tertiary Correspondences
    for all x in $\chi(v)$ **do**
      if x is visible in all l **then**
        $C_2(v) \leftarrow C_2(v) \cup x$
      else
        for all l where x is not visible **do**
          search G for a p or e that is visible in l and update x
        end for
        if x is now visible in all l **then**
          $C_3(v) \leftarrow C_3(v) \cup x$
        end if
      end if
    end for
    // Update original graph and meshes
    for all c_2 in $C_2(v)$ and c_3 in $C_3(v)$ **do**
      for all l in L **do**
        update G & $M^3(l)$ to include c_2 and c_3
      end for
    end for
  end for
until $|C_1(v)| > \varepsilon \forall v$

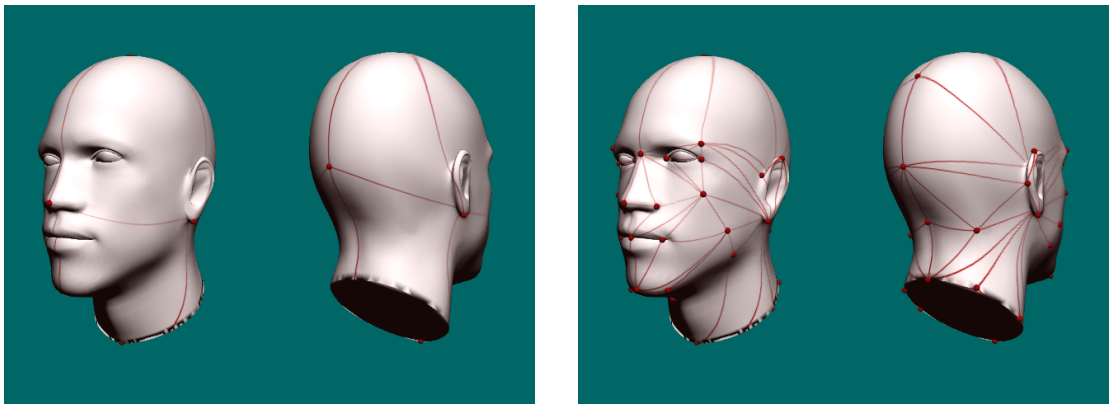
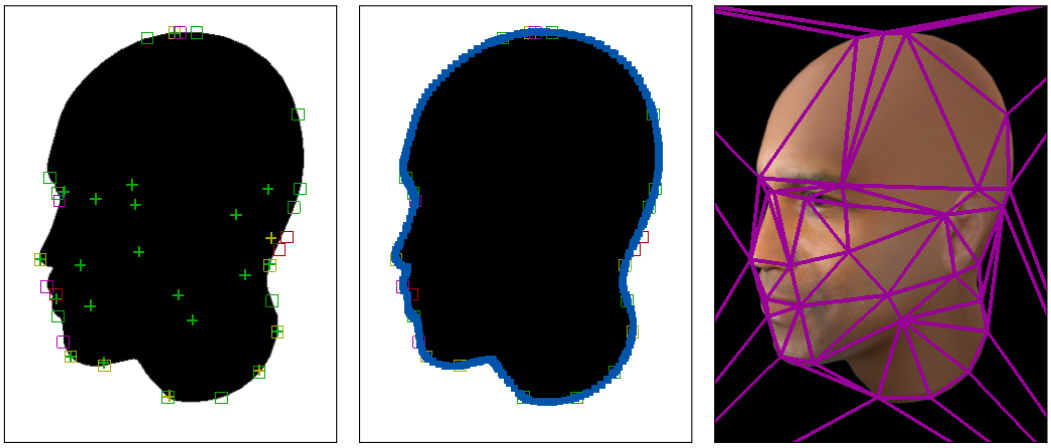


Figure 6.2: The initial six feature points (left) and a completed correspondence mesh (right).

When considered as a mesh, the matched points exist in three-space directly on the surface of each target. The edges of the mesh also need to follow the surface of the target. To achieve this, they are projected from a sphere onto the surface of the object so that they follow its contours and crevices tightly. They start as straight lines but become curved in projection. The mapping used to project these edges from the sphere to the surface is very important. If the objects being morphed are convex then a simple spherical mapping can be used where the edges are projected along a line from the center of mass of the object out to the surface of a sphere. This approach presents problems for objects with strong concavities and negative curvature. In this case, an explicit, one-to-one mapping must be computed at all the vertices of the object. There are many parameterization algorithms that aim to compute texture coordinates on mesh objects with a spherical layout. Any one of these algorithms could be employed to compute an explicit mapping between the target and a sphere. The algorithm used is an implementation choice which we will discuss in Section 6.4.

By projecting the correspondence meshes M^3 in this manner we can map them to each of the views in the surface light field and generate a set of sparse 2D correspondences between the targets (line 5). We assume that each surface light field used in the morph shares a common set of viewing directions (this may require the original image sets to be re-sampled). After projection, only a portion of the original correspondence graph nodes and edges will be visible while the rest are occluded and removed as hidden surfaces. The graph nodes that do remain visible in each target are paired up and become the *primary correspondences* (C_1 at line 10).



(a) 2D Correspondences (b) Contour Samples (c) Image Morph Mesh

Figure 6.3: Different components for creating the 2D shape and image morphs.

This small collection of correspondences would be insufficient to establish a good morph particularly around the outline of the silhouette where all the shape information is contained (see Figure 6.3(a)).

To compensate, additional correspondence points are automatically generated using the edges from the correspondence graph. Everywhere one of these edges crosses the silhouette a new correspondence point can be established. After locating and naming all of these silhouette-edge crossings in each target (line 6) we pair them up as follows:

a) Crossings that are caused by the same edge on all targets are paired (line 15).

b) Remaining crossings that have no match in one or more other targets must search for a match by traveling along the correspondence graph looking for points or other edges that do cross the silhouette in the other target (line 21).

These are referred to as the *secondary* and *tertiary* correspondences respectively (C_2 and C_3). In practice the search described in step 2 must be limited to not travel too far from the original edge or the path of the warping will not match from view-to-view causing ghosting. Consequently, some crossings might remain unpaired and these are discarded.

We distinguish between secondary and tertiary correspondences as, in general, the tertiary correspondences will be made between points that are further away from each other on the surfaces of the original targets than secondary correspondences. Furthermore, only primary correspondences are guaranteed to represent matching features on the surface of the targets. We find in practice, that this is an important distinction when computing the 2D image morph but less important for the 2D shape morph. In fact, a user may wish to ignore tertiary correspondences completely and only allow secondary correspondences to be used.

The secondary and tertiary correspondences were established in image space where we can know the location of the silhouette shape. For the single view where they were established they are guaranteed to stay matched as the object warps but the same is not true of other views that have no knowledge of these points. Because of this they cannot be directly used in the image morph or the appearance of the object will not stay registered and ghosting may occur. To rectify this, we map these points back to the object surface and create a new 3D feature point. Each secondary and tertiary point has a 2D location for one view of each target (n total 2D locations for n targets). These locations do not necessarily represent a matched point on the 3D objects. They are useful for creating a 2D shape correspondence but cannot be combined as an overall 3D feature point. Instead, each of the n locations determines a new feature point that will have a unique position on all of the targets. Some work must be done here to unproject the 2D position back to one of the 3D objects and then re-map that point to the other 3D objects. We use both the existing correspondence mesh and the spherical parameterization to achieve this re-mapping between the various target objects. Afterwards, the process of converting the 3D correspondence mesh into sparse 2D correspondences is repeated (line 27). We can iterate like this until some maximum number of primary correspondences is reached or until the distance between 3D feature points is below a minimum threshold. In practice, a single iteration of this process introduces enough new feature points to be sufficient.

The new 3D feature points added in this process are necessarily located on an edge of the original correspondence mesh. This is because they were originally identified as a point where an edge meets the silhouette. We could choose to subdivide that edge and the mesh that contains it as we add a new 3D feature. This would create new edges that could in turn cause new secondary or tertiary correspondences. However, since the new edges introduced would, in most cases, run nearly parallel to the silhouette edge, they would create problems for identifying the edge-silhouette intersections automatically. Given that a single iteration of

this process introduces so many new feature points (usually several hundred for our examples) and further iteration is not necessary, we choose to avoid this problem by *not* letting the mesh be subdivided by the new 3D feature points. The points simply sit on an edge providing well established correspondences that are useful for the reflectance imagery and not just the object shape.

At the completion of this process we have a set of sparse 2D correspondences; one for each view in the original surface light fields. To complete the morph process we need to convert this sparse correspondence into a dense correspondence for both the shape and appearance imagery in the unstructured lumigraph.

6.3.2 Building Dense 2D Correspondences

With a reasonable set of sparse 2D correspondences available we can build the 2D morphs, starting with the shape morph. A contour shape is extracted from each silhouette alpha-matte. This contour is a combination of smooth curve patches and lines accounting for sharp corners. To construct the shape morph we must use our sparse 2D correspondence to make a dense correspondence on these contours. There may be more than one continuous contour in any given view caused by holes in the original object or by overlapping protrusions that visually fuse in projection creating a hole in the 2D image (regardless of the topology of the original objects). When there are multiple contours in a single view we must pair the contours between the different targets. These multiple contours will never intersect (due to the spatial continuity of the original object geometry). This allows the contours to be arranged in a tree structure where each curve is a child of the curve that completely contains it. Contours on one target will map to contours on the other target only if they are in the same level of this tree but this mapping is not necessarily one-to-one. For example, two contours may need to merge into a single contour or a single contour may need to close and disappear as the views warp. These situations can arise when the original targets were of a different genus or if a hole visible in one target is occluded in the other. While it is necessary to establish the full mapping between the two trees, the tree structure itself is only used to maintain the constraint that contours cannot be paired with contours in a different level of the tree. As long as this constraint is maintained the mapping (including non one-to-one situations) will naturally occur as we search for matching points between the various contours.

For each contour pairing, we need to create a dense correspondence using the sparse 2D

correspondence previously generated. As noted before, in the shape morph, all of the correspondences can be used including secondary and tertiary ones. All of the primary, secondary and tertiary correspondences are examined to see which ones lie on the silhouette edge in all targets. In practice, we allow some wiggle room here of a few pixels to be considered 'on the silhouette'. In each target, the nearest point on a contour to these silhouette matches becomes a matched point for that contour and maps it to the contours of the other targets as long as they are in the same level of the tree. This sparse correspondence on the contours is made dense by linearly sampling the contour between the matched points with the same number of samples for each target (see Figure 6.3(b)). The new samples are easily paired and what results is a polygon that represents the shape as precisely as the sample density (up to the limit of the resolution of the original image).

Once all matched points are located any remaining contours that are not paired up are assumed to have nothing to map to in one or more of the other targets. These contours must close as they warp towards the target with no match. This can be achieved by morphing the contour into a single point. The location of this single point might be the 2D center-of-gravity of the contour. However, this would not be consistent from view to view. Instead, we should first attempt to pair any identified correspondence point on that contour with a primary correspondence point in the view of the un-matched target that is not on the silhouette (not previously considered when searching for a match). Such a point should always exist and give us a consistent location in each view. Only when such a point can't be found should we resort to the center-of-gravity (although, this may indicate that the number of 3D feature points is insufficient and should be expanded).

In addition to the shape morph we also build a separate image morph for the photographic data. In contrast to the silhouette alpha-mattes, the interior of the shape (and not just the edge) contains important information and needs to move consistently from view to view. Using the polygon created for the 2D shape morph (which was seeded with the secondary and tertiary correspondence points) causes unnatural swirling to occur in the interior. Consequently we ignore the contour shape as well as the secondary and tertiary correspondences and rely entirely on the primary correspondences, which are guaranteed to represent consistent features on the surface of the original targets. To make the sparse set of correspondences dense, we triangulate the points in 2D and use texturing to achieve a natural linear interpolation across the entire image (see Figure 6.3(c)).

6.3.3 Generating the Morphed Unstructured Lumigraphs

With our set of dense 2D shape and image correspondences established all that remains is the warping portion of our morph. It is important that the path taken by the shape morph matches that taken by the image morph as closely as possible. With some 2D warping/morphing algorithms (such as as-rigid-as-possible warping [Alexa et al., 2000]) the warped path chosen is tightly coupled with the structure of the underlying correspondence. Since we do not establish an identical correspondence for the shape and image parts of the structure this may cause radically different warping paths. For this reason, we make the simple choice of linear interpolation to compute the warping path which will have minimal differences for the two components.

All of the 2D morphs for each view are computed in parallel, so at any point we have a new set of silhouette alpha-mattes resulting from the shape morphs and appearance data from the image morphs. This set of images serves as the basis for a new unstructured lumigraph. The alpha-mattes can be re-projected and intersected to compute the visual hull as proxy geometry and the appearance images can be reprojected onto that geometry to make a tweened version combining the appearance properties of all the input targets. In practice, the shape and image morphs can be interpolated different amounts so that geometry and appearance are explored independently. This allows for interesting cases where the appearance of one object can be transferred in its entirety to the geometry of a different target. However, one is cautioned that, as in all image morphs, the more the photographic images are distorted the less natural the end result will look.

6.4 Implementation

We implemented our surface light field morphing algorithm using existing technologies from image based rendering. We accepted the following simplifications to limit the scope of the problem:

- We only consider objects that are simply connected (genus zero, consisting of one connected piece).
- We assume the input unstructured lumigraphs already share a common set of views and a common lighting arrangement.

- We limited the number of targets in the morph to be at most three.

While a common lighting arrangement for the input unstructured lumigraphs is not required, the appearance in the resulting morph would likely not behave as expected if this is not the case. Our input surface light field objects are represented and rendered as unstructured lumigraphs originally developed by Buehler et al. [2001]. The lumigraphs contain photographic imagery encoding the reflectance of the object as well as silhouette alpha-mattes encoding its shape. The alpha-mattes are used to compute a polygonal visual hull for the object which becomes the proxy geometry needed in Buehler's algorithm. Ultimately, our representation has calibrated images with intrinsic and extrinsic camera parameters only. For the rest of this section we will first describe some initial steps including some necessary pre-processing, the interface for specifying the correspondence mesh and the process of projecting that mesh onto the object and remapping points between the targets. We will then describe the rest of the algorithm in five discrete steps:

1. Creation of the 2D imagery and extraction of the shape contour
2. Identifying primary, secondary and tertiary correspondences
3. Automatically refining the correspondence mesh and iterating
4. Building the 2D shape and image morphs
5. Executing the resulting unstructured lumigraph morph

6.4.1 Preprocessing

There is one key preprocessing step necessary to enable the re-mapping of points between different targets in the morph. As mentioned previously, this is achieved through the use of a mapping of each target to a sphere. We choose to limit our input geometry to simply connected objects because this enables the use of many more spherical mapping algorithms.[1] We employ the technique developed by Angenent et al. [1999] which was designed to allow texturing of undulating objects (in their case, the geometry of the human brain) providing a spherical parameterization (see Figure 6.4). This technique was chosen particularly because it provides the

[1] A good overview of existing techniques for mesh parameterization (and specifically spherical parameterizations) is available in the SIGGRAPH 2007 course "Mesh Parameterization: theory and practice" presented by Hormann et al. [2007] or chapter 5 of the book "Polygon Mesh Processing" by Botsch et al. [2010].

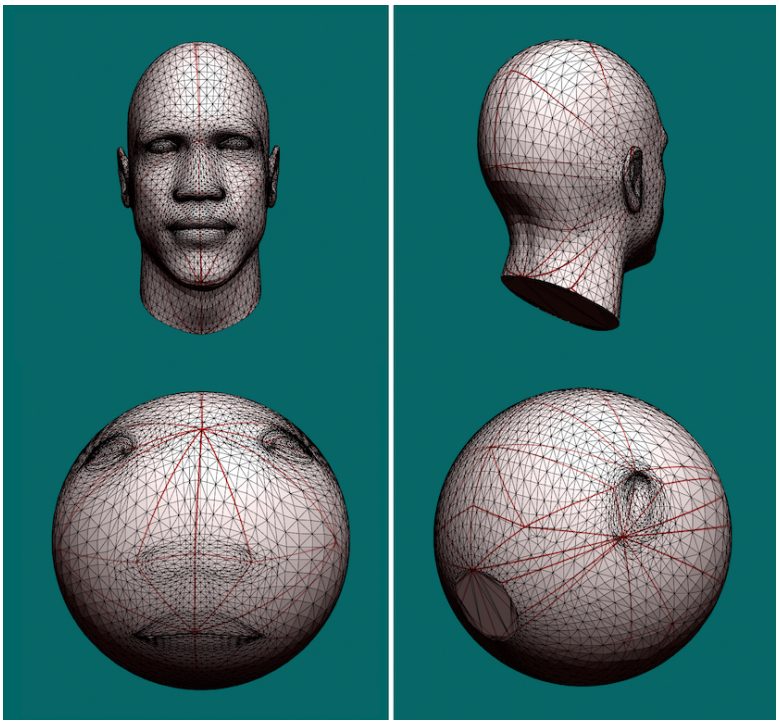


Figure 6.4: A target object with its spherical parameterization as computed by the Angenent et al. [1999] method

spherical parameterization instead of just a planer mapping, and because there is a robust implementation available as part of the Insight Toolkit [Gao et al., 2006], an open source library for computer vision in medicine. The input geometry to this algorithm must be simply connected to guarantee a one-to-one mapping as output. Since we compute the geometry as the visual hull of our object this is a guaranteed property. If finer geometry is used (as the result of a scan or by manual modeling) the geometry must be made simply connected manually. This involves making the geometry consist of one connected piece that is 2-manifold (i.e. there are no handles or holes in the surface). There are many geometry filters in existing modeling tools that can simplify this process. Objects of higher order genus (with holes or handles) cannot be processed with the technique of Angenent et al. [1999]. Other algorithms that depend on a homeomorphism with a plane instead of a sphere can deal with higher genus objects by cutting them. While we have theoretically described how to deal with objects of higher genus in the later parts of our algorithm, we leave exploration of these objects and the parameterization

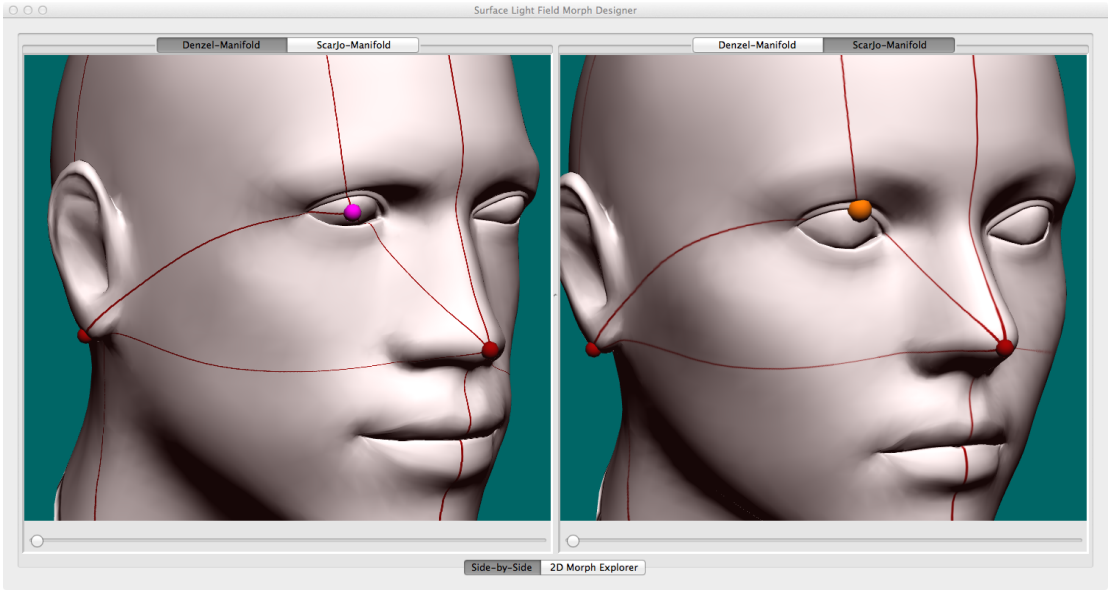


Figure 6.5: The user adding a new feature to a target (left purple point) and the proposed location automatically computed on the second target (right orange point).

techniques that support them to future work.

6.4.2 Initial Creation of the Correspondence Mesh

Prior to processing the lumigraphs with our algorithm, the user must define the mapping of corresponding features on the different targets. We provide a simple interface where the user clicks directly on the object to lay down these features. This interface is implemented in OpenGL and employs the well defined projections of the OpenGL pipeline to determine the location of each feature precisely on the surface of the target geometry. The user can freely rotate the object as features are placed around the surface (see Figure 6.5).

Initially, the user must place six specific points on the object roughly located at the apex, nadir and around the equator (positions corresponding to the different faces of a cube map). We guide the user to these positions by using reflection mapping and a specially colored cube map that distinguishes the various parts of the object. Note that instead of looking up into the cube map using the reflection vector (as in normal reflection mapping) we use the spherical position calculated by our spherical mapping computed in the preprocessing step. This gives us a well

defined mapping out to the cube map where every position is unique. These first six positions are then connected by edges in a fixed manner that produces the initial correspondence mesh. The geometry will always be two rectangular pyramids stacked end-to-end (see Figure 6.2).

By starting with this regular geometry we can then proceed to refine this correspondence mesh with a simple subdivision process as the user lays down more feature points. If a new point is added near the center of an existing triangle in the correspondence mesh then that triangle is regularly subdivided into three new triangles. If a new point is added close to the edge of an existing triangle then the edge is split into two edges and the triangles that are adjacent to the original edge are each split into two new triangles. Proceeding in this manner the user can add as many points as necessary to all of the targets and 'design' the quality of the morph to their liking.

6.4.3 Projection Onto the Targets and Remapping

As mentioned previously, the feature points laid down by the user exist on the surface of the object but the lines connecting them do not. Instead, we must project them onto the surface so that they follow its contours precisely. We active this by using a modified reflectance mapping technique, similar to the way we guide the user to the location of the initial six points. The feature points are mapped out to the sphere using the same parameterization that was computed for the geometry. The points are then connected by straight lines and this structure is rendered into a cube map centered at the origin of the sphere. When rendering the target geometry we lookup into the cube map and select pixels using the coordinates of the spherical parameterization instead of a reflection vector (as in traditional reflectance mapping). This provides a consistent projection of the correspondence mesh onto the surface that is one-to-one and reversible. In addition to projecting the edges onto the geometry to visualize its connections, we also render the edges and the triangles of the mesh to an off-screen buffer colored to uniquely identify each component. As the user places down feature points we can use this information to know which triangle or edge in the mesh to subdivide.

Another important task made possible by the spherical parameterization is the ability to map a feature point from one target to the others. As a user places a point we want to automatically place that point on the other targets in a reasonable location. Each target has its own spherical parameterization and there is no guarantee that this parameterization is well aligned between the various objects. If we were to use it directly the feature points would show up in undesirable

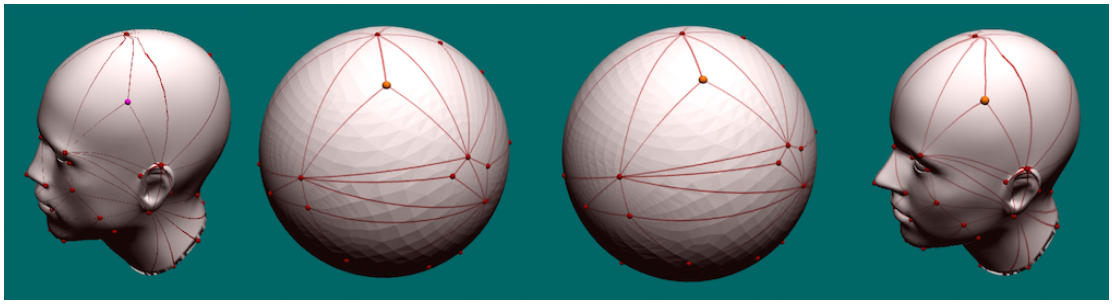


Figure 6.6: Re-mapping a new feature (left to right): the original location in purple; mapped to the sphere of original target; on the sphere of a new target (using barycentric coords); mapped to the surface of the new target.

locations on the other targets. Instead, we embed the existing correspondence mesh into the spherical geometry and use it to achieve the desired alignment between the targets. The process of re-mapping a feature from one target to another is as follows (also see Figure 6.6):

1. Map the new feature point to the spherical parameterization for the target on which it was placed.

2. Associate the feature with the triangle in the correspondence mesh that contains it and compute its coordinates relative to that triangle (barycentric).

3. Transfer the point to the other correspondence mesh and convert it back to a spherical position by reversing the barycentric calculation.

4. Map the point to the surface of the object using that object's spherical parameterization.

By following this process with each new feature point, correspondences are maintained and new locations are placed very reasonably on the other targets. The user will still need to adjust the re-mapped locations to design the overall morph between the targets. The net effect is a system for aligning and warping the spherical parameterizations of the different targets so that the quality of the parameterization is not as important and need not be as constrained when it is calculated.

6.4.4 Stage1: Correspondence Projection and Shape Extraction

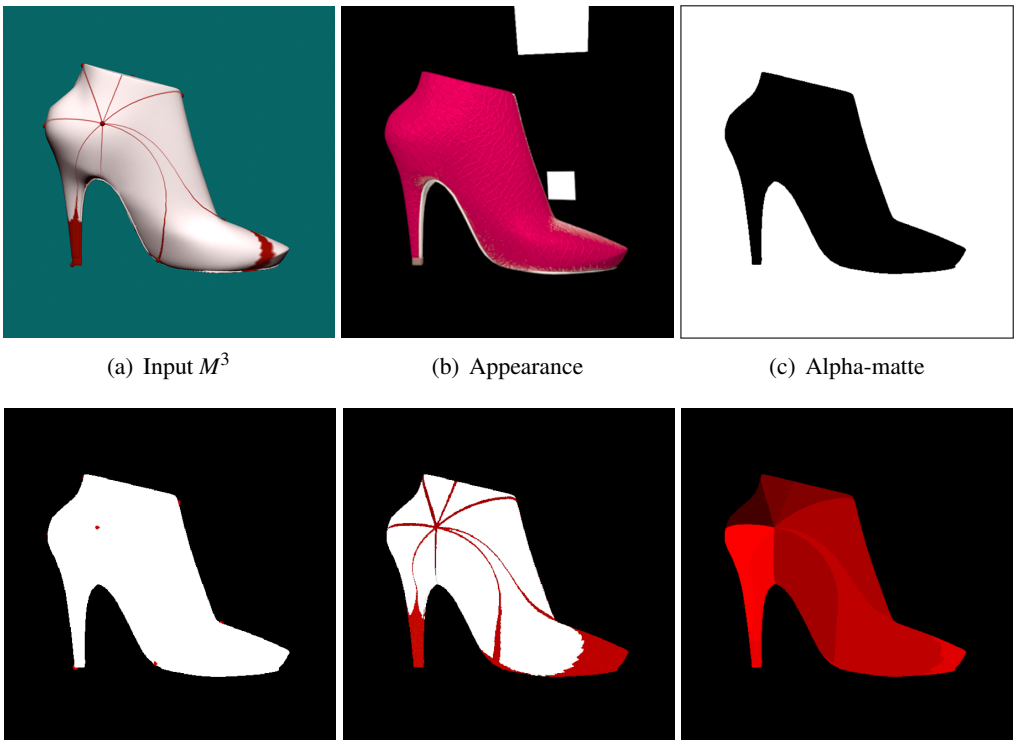
With the correspondence mesh well defined by the user on each target, we can proceed with the first stage of processing. From each view described in the lumigraph we render the object in several different forms (see Figure 6.7):

1. With natural appearance and reflectance (the result of normal unstructured lumigraph rendering)
2. As a silhouette alpha-matte (a solid black object with no shading on a white background)
3. With the feature points of the correspondence mesh drawn directly as small spheres (colored for identification)
4. With the edges of the correspondence mesh projected on the surface (colored for identification)
5. With the triangles of the correspondence mesh projected on the surface (colored for identification)

Note that for the images colored for identification, the numerous anti-aliasing features of OpenGL (including full screen supersampling, texture mipmapping and edge antialiasing) are turned off so as not to obscure the colors used to identify the elements on the object. The alpha-matte image (Image 2) is processed with the potrace algorithm from Peter Selinger [Selinger, 2003] to extract the contour tree (see Figure 6.8). At present, we use only the root of the tree and ignore any other silhouette curves. Therefore, objects that have holes in projection to any of the views are not supported and these views are discarded. The remaining set of images along with this contour are passed on to the next stage of processing.

6.4.5 Stage 2: Correspondence and Silhouette-Edge Crossing Extraction

The image containing only the features of the correspondence mesh (Image 3) is inspected to identify what features remain visible from that view. OpenGL will have removed any hidden features automatically as it removes hidden surfaces. Because the precise location of the correspondence points is known they are drawn as small spheres centered at the given point. They have some volume and will be slightly visible around corners and protrusions on the object



(a) Input M^3 (b) Appearance (c) Alpha-matte

(d) Features (e) Edges (f) Triangles

Figure 6.7: The various types of images generated as input to the 2D morphs (color ID images are enhanced for visibility).

which aids in building an accurate 2D correspondence. As each visible feature is identified we re-project its original location to the 2D image space for that view and record it precisely. The list of visible features on each target is compared and paired up according to the feature ID. These paired features become the primary correspondences for that view. Any un-paired visible features (ones that are visible on some targets but not all) are remembered for use later when we are searching for tertiary correspondences.

While inspecting the features image we also inspect the edges image (Image 4) and build a list of visible edges. To identify which of these edges crosses the silhouette we employ a computer vision technique using OpenCV. For each visible edge we filter the triangle image (Image 5) to isolate the two triangles that share that edge (see Figure 6.9). This image is made grayscale and then processed to locate strong corners using the method described by Shi and

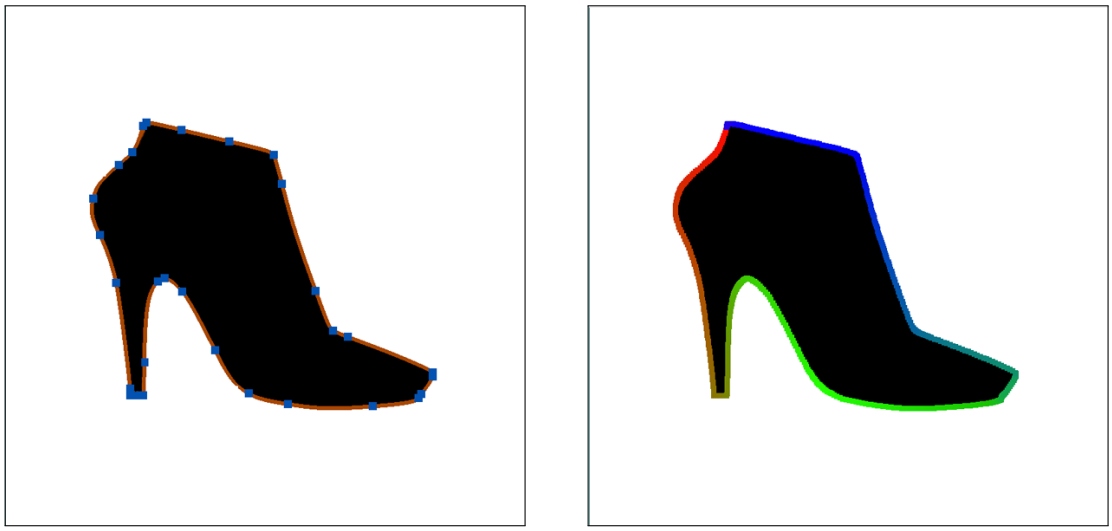
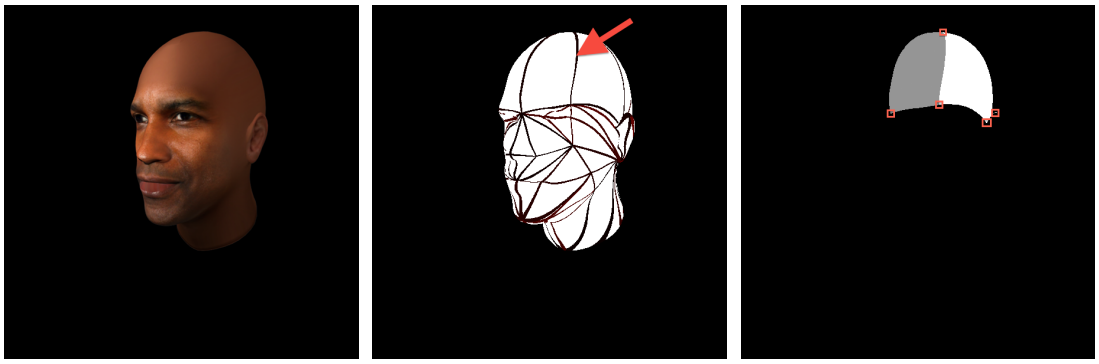


Figure 6.8: The poly-bezier contour computed by potrace (left) and a discretized version shown as colored boundary pixels (right).

Tomasi [1994] (as implemented in OpenCV). Each strong corner is compared with the location of the silhouette contour and the correspondence mesh edge to identify corners that are both on the desired edge and on the silhouette. Points that remain are saved as silhouette-edge crossing points.

At present we find this process to be imperfect failing to find several important crossings and mis-identifying some other image features as crossings that are not. Further refinement of this portion of the algorithm is necessary in order to fine-tune the parameters used in the Shi and Tomasi [1994] detector. In addition, software verification is still necessary to ensure that it was implemented correctly. For the examples demonstrated in this thesis, we have manually modified the output from this portion of the algorithm to correct these problems. This manual interaction, while hopefully unnecessary in a future iteration of this work, is not necessarily breaking with the overall design of the algorithm as the user is already expected to manually provide the initial 3D morph features.

With the silhouette-edge crossings identified, we compare them across all targets and attempt to pair them up based on the edge ID associated with the crossing. ID's that are present in all targets are paired and become the secondary correspondences. For the crossings that remain un-paired we search the correspondence graph for either a feature or edge to pair with that



(a) Appearance Image (b) Edges Image (c) Isolated Triangles with Corners

Figure 6.9: The edge-silhouette crossings detection process: 6.9(a)) The input pose; 6.9(b)) The edges of the control mesh (detected edge indicated with arrow); 6.9(c)) The triangles that share that edge isolated and the corners detected by the Shi and Tomasi [1994] detector in red.

is visible in the targets that have no direct match. They start at the edge associated with the crossing and do a breadth-first search until a node or edge is found that is visible in the target in question. We halt the search after a depth of three is reached not wanting to pair the crossing with anything that is too distant from itself. Any silhouette-edge crossings that fail to pair up after this search are discarded.

6.4.6 Stage 3: Refinement of Correspondence Meshes and Iteration

At this point we have the primary, secondary and tertiary correspondences computed for each view. If the number of primary correspondences is adequate or the distance between nodes in the correspondence graph is below a small threshold then we can proceed. For the work in this thesis we designed the original correspondence meshes with sufficient features to not require any iteration. Exploration of the iteration and refinement portion of this algorithm (described earlier in Section 6.3.1) is left as future work.

6.4.7 Stage 4: Building the 2D Shape and Image Morphs

With a sufficient number of primary correspondences we can proceed to build our 2D morphs. For the shape morph, we use the contour that was extracted by potrace. This contour is sampled into a series of points with a density that is controlled by the user. Each of the points is compared

to the list of primary, secondary (and optionally tertiary) correspondences to establish matches with similar points on the sampled contours for the other targets. With these correspondences in place, the remaining sample points are paired up linearly to achieve a complete, dense correspondence between the contours of each target. Occasionally we find that the pairings with secondary or tertiary points introduce discontinuities in the contour. Specifically, if you traverse the contours in a specific winding direction you may encounter these pairing points in a different order on different targets. These problems could cause the contour to turn itself inside-out as it morphs. When this happens we attempt to repair these contour pairings by eliminating one of the out-of-order secondary or tertiary correspondence points. If the repair is not successful (i.e. there are still out-of-order samples) then that particular view is eliminated (it is also marked so the user can intervene and repair the view if desired). The densely correlated sample points are connected into a potentially concave polygon that is tessellated into a set of convex polygons (usually just triangles) for rendering. We use simple linear interpolation to warp the vertices of this polygon between their locations in each target.

To construct the image morph, we ignore the contours as well as the secondary and tertiary correspondences (for the reasons discussed previously) and use only the primary correspondences. These are laid out across the photographic imagery associated with that view. We process these points (along with boundary lines that mark the edges of the image) with the Triangle library by Jonathan Richard Shewchuk. This library generates a Delaunay triangulation in 2D of these points and lines. The result is a triangulated quad with vertices located at the points of primary correspondence. We draw this quad with the photographic imagery textured on it. All that remains is to linearly move the vertices of the quad between their positions in each target for the warping component of the morph. In practice, we find that superior results are achieved by using the average positions of the correspondences when computing the triangulation to avoid any degenerate triangles occurring as the vertices are moved. Another alternative is to recompute the triangulation at every moment as the vertices are moved. However, this also produces undesirable effects as the triangulation suddenly jumps from one to another creating a visible pop in the morph.

6.4.8 Stage 5: Executing the 3D Surface Light Field Morph

With a 2D shape and image morph constructed for every view, we can now execute them in parallel and at any moment build a new unstructured lumigraph. The shape morphs give us

the silhouette view of the morphed geometry at any point in-between the original targets. We process these silhouettes with a visual hull algorithm to generate a polygonal visual hull for use as the proxy geometry in unstructured lumigraph rendering. The image morphs become the reflectance imagery that is projectively mapped onto the proxy geometry according to the camera blending field. There are two considerations that are unique to the morphing problem that must be addressed:

- In some views a portion of the surface of one target may be obstructed that is visible in one of the other targets. The information provided by that view's image morph would be incorrect. This can be accounted for in the camera blending field calculation of the unstructured lumigraph rendering algorithm. We can recognize these cases as we build the 2D image morphs and mark the problematic regions in the Delaunay triangulation accordingly. When computing the camera field this is an additional metric that could be considered. If the pixel in that view is part of a bad region it is given an infinite weight. Otherwise, it is given a zero weight. In this manner, the information will come from a view where that portion of the surface is visible for every target. For the work presented in this thesis we did not implement this technique but leave it for future exploration.

- There is no reason that the shape and image morphs need to be locked to the same blending value. Different values can be given for each with the effect that geometry and appearance will morph independently. However, the further the geometry is distorted towards a different target the more the source imagery will end up being stretched. This will yield visually inferior results in the extremes. Nevertheless, this provides a means for retargeting the appearance of one object to the geometry of another without the use of texture synthesis.

In our implementation, we use a relatively slow, volumetric visual hull algorithm based on code provided by Forbes. It builds a polygonal mesh by walking a uniform grid in space and examining an approximation of the distance field of the object calculated form the distance transforms of the individual silhouette images (see Figure 6.10). The running time of this algorithm makes real-time calculation of the 3D surface light field impossible. Alternative visual hull calculation methods could mitigate this and we leave their exploration to future work. Additionally, we need to process the resulting visual hull with a simplification algorithm to decimate the mesh to a reasonable number of vertices for use in the rendering of the unstructured

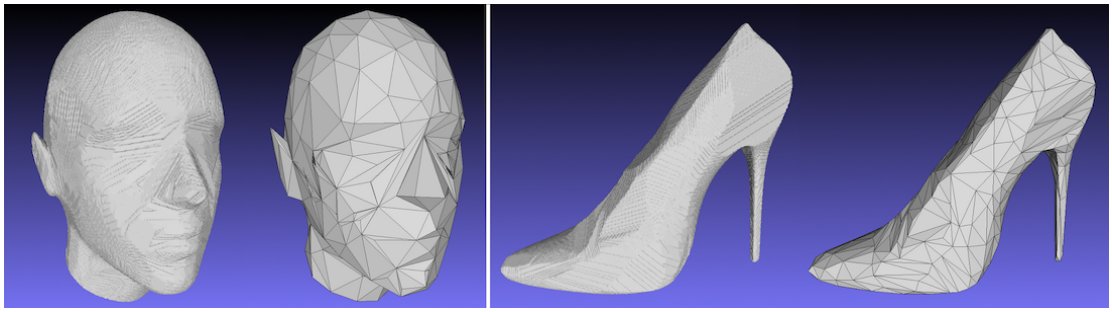


Figure 6.10: Two intermediate visual hulls in full detail and after decimation (left to right: $\sim$ 1.4m triangles, 666 triangles, $\sim$ 840k triangles, 666 triangles)

lumigraph (for the examples in this paper we reduce the polygonal visual hulls to about 600 vertices using a standard edge-collapse approach as implemented in CGAL). We find this to also be a slow process. Because of this, our current implementation samples the morph at discrete intervals and computes the visual hulls offline. We can then step through these samples of the continuous morph to experience it in real-time. At every point we have a complete unstructured lumigraph that can be rotated and viewed in 3D with accurate appearance properties that are a combination of the various targets that contributed to the morph.

6.5 Results

Using models purchased from a commercial 3D model web site (and motivated by the appeal of 2D face morphing) we created surface light fields of two celebrity heads of different races and different genders. The images needed for the unstructured lumigraphs were rendered offline using a clone of the PBRT renderer called LuxRender. We calculated 101 views evenly spaced around the sphere to sample the light field. The input to the unstructured lumigraph is still just appearance and silhouette imagery. We processed these lumigraphs with our algorithm to generate a morph that shows one celebrity changing into another (see Figure 6.11). The full morph was sampled at 15 discrete steps each of which accurately captures the appearance of both targets while staying well registered throughout the continuum of the morph.

We designed and computed a second morph between two very different shoes. These models we also purchased from a commercial web site and rendered using LuxRender with the same set of 101 views. We chose shoes as objects that would be of significance to a particular design

Figure 6.11: A full unstructured lumigraph morph between the heads of two celebrities (starting at the upper-left and proceeding clockwise). The intermediate representations can be rotated and will maintain accurate reflectance properties such as specularity and subsurface scattering.

field (fashion design) yet stay within the constraints of our implementation. The first shoe is an ankle boot with a high heel modeled with a red cracked leather material. The second shoe is a pump with a very long opening, a similarly high heel and a slightly specular snakeskin material. Of particular interest in this morph is the need to handle extreme geometry with long thin protrusions (the heel and toe) and strong concavities (the underside of the sole). There is also a very large change in the size of the opening at the top of the shoe. We show the results of the morph in Figure 6.12.

6.6 Conclusions and Future Work

We see in the head morph that our technique can be successful. We are able to compute a morph that accounts for not only geometry and shape but also appearance and maintains a reasonable reflectance quality even in the in-between states of a morph. At the sample points we have explicitly computed a new unstructured lumigraph that provides the surface light field of this

Figure 6.12: A full unstructured lumigraph morph between two types of shoes (starting at the upper-left and proceeding clockwise). Notice the difficulty in keeping the opening registered as it moves a large distance.

theoretical in-between object. Its reflective properties can be known and used in the design process. We see good registration maintained between the different views minimizing artifacts due to the approximate nature of the visual hull geometry and the unstructured lumigraph technique as a whole.

The shoe morph represents a significant challenge for our algorithm that is well met even in its partially complete state. Keeping the opening registered as it moves such a great distance is very difficult and requires many feature points to be placed around the openings. The thin heel also has a tendency to disappear in the visual hull due to slight errors in the shape morph. This could only be avoided by manually eliminating the views where the heel disappeared (requiring removing 7 views out of 101). The results are reasonable and show promise for our technique as it is further improved and future enhancements are explored. The retargeted appearance is not as pleasing for the shoes due to the amount of distortion necessary to warp the smaller opening into the larger one. This distortion is particularly noticeable when the relatively small area of the side of the pump is re-mapped to the large area on the side of the boot.

The use of shape-from-silhouette to reduce the 3D surface morph to a set of 2D shape morphs is a novel contribution. This approach is considerably simpler than mesh based approaches that require precise parameterizations. We are able to achieve a high level of quality with this approach and can potentially handle more general topologies with future improvements. The world of 2D morphing has seen more success than its 3D counterpart. Our technique brings that success to bear on the 3D problem.

Our technique explicitly computes an unstructured lumigraph at in-between locations in the morph which is an advantage over other light field morphing techniques where the in-between representation is only implicit in the rendering algorithm. However, a more thorough examination of the other morphing techniques and comparison with our results is necessary to fully appreciate the trade-offs between them and to clearly enumerate when our technique is most applicable. The in-between light fields are physically plausible and represent a blending of the reflected intensity values. The effect is visually compelling for real materials with spatially-varying texture and diffuse color.

While the spherical parameterization technique used was effective for the head models it begins to cause problems when we use objects that are not as close to spherical in shape. When large protrusions exist on the object they are naturally compressed to occupy a very small area in the resulting sphere. This is due to the method of Angenent et al. [1999] attempting to preserve angles of triangles instead of area (a conformal mapping). A parameterization technique that preserves area would be better suited to a more generic class of objects. However, our primary goal is not to support a large class of shapes but rather a large set of material appearances. This is particularly the case given that many material samples used in the design process are flat samples (or nearly so).

While we have shown that this technique can produce good results we have only just begun to explore the morphing possibilities implied by its use. We have employed only the simplest of 2D image morphing techniques allowing for single point features and linear interpolation. More advanced techniques can now be explored that allow specification of lines (from the connectivity of the correspondence graph) as well as points and that use more advanced warping techniques such as radial basis functions [Cohen-Or et al., 1998; Ruprecht and Müller, 1995].

There are other ways that we could attempt to find good correspondence points between the various targets without involving the user. These could include processing the appearance imagery with a vision algorithm, like SIFT [Lowe, 1999] or a form of optical flow, to identify

positions from different views that strongly correlate. If the starting objects are similar enough (like those of the celebrity heads) then the features identified in this process could be enough to seed the entire process and produce pleasing results with almost no user interaction.

Chapter 7

Conclusion

7.1 Review

In this thesis we started simple, building tools around a conventional rendering engine on consumer graphics hardware. We limited ourselves not just to paint but to specific color formulations from a manufacturer. With a well understood set of materials in a vernacular paradigm like household paint we set some simple goals. We showed our users realistic appearance properties and gave them the power to visualize simple choices, like gloss level, with the Wall of Inspiration software suite. We guided the user and augmented their natural understanding of color with a digital tool that fit their perceptual model of color; the Color Navigator. Our study of this tool showed that it can be useful as long as the user understands perceptual color organization and is comfortable with the interface. But more than the small gains in time, we provided the user with a more fulfilling experience that can spark new understanding of the collection as a whole and the ranges of colors that it provides. These simple digital design tools sow the seeds of digital material appearance and teach us much about what the design industry needs and will really find useful. But color selection was only a beginning, an exercise that lead us naturally to designing the paint color itself.

From our beginnings with individually fixed, solid colors we moved to color shifting paints. Here we had an opportunity to build on the industry standards for representing these colors and the formulation process that already existed to allow the creation of entirely new colors. An architect had a very specific source of inspiration and wanted to draw from that, not just one new color but an entire continuum of colors in smoothly progressing shades. We turned to

interpolation as a natural technique and with tight feedback from the designer we very quickly moved from an idea to a real prototype with real paint materials and proper appearance. We were still very limited in the scope of materials that we are able to work with (metallic paint coatings from the automotive industry). But this limitation gave us an obvious and versatile digital representation for our materials that we could immediately exploit to realize a new and useful digital design tool. Our interpolator had a specific problem to solve and may not be of considerable use beyond this one case but it solves that problem faster and more precisely than any hand made solution could ever approach. This quickness is by virtue of the digital nature of the tool and it affords the kind of iteration and rapid prototyping that the early design process encourages. With this success, the natural next step was to generalize our model and support more materials (ideally, as many materials as possible).

Finally, we dove into the most general representation currently available in computer graphics, image based rendering. This was a huge step both in generality of our digital material representation but also in effort needed to get results. We could no longer lean so lavishly on standard programming interfaces like OpenGL and the hardware that it drives. But just as with the color shifting paints, we had the potential here to design an entirely new appearance that goes far beyond our metallic color interpolator. Properties of any existing digital materials can inform the design and lead us in new directions. The natural structure underlying the unstructured lumigraph suggested a powerful way to achieve this through image morphing. By reducing the inherently 3D nature of a light field morph to a set of 2D morphs we open up the rich toolbox of 2D image and shape morphing. The relative difficulties of 3D morphing and its failure to generalize to all topologies and parameters is immediately overcome. What we show affords all of the generality of a light field morphing technique to accurately blend material reflection properties as well as shape and does so through direct use of the images that constitute the light field data itself. No re-factoring or numerical analysis is necessary and at every point in-between we have all the data we need to represent the light field of the blended material.

We offer in this thesis two significant and novel contributions to the field. First, our surface light field morphing algorithm is a significant contribution in two ways. It approaches 3D shape morphing from a unique perspective utilizing the visual hull calculation to reduce the complexity of the problem. This approach is simpler and more intuitive than other 3D morphing techniques that require a parameterization of the mesh. Additionally, while other light field morphing techniques exist, none of them work with the unstructured lumigraph representation

directly. The morphing of this specific representation (which has different strengths and weaknesses than the light field mapping and lumigraph representations) is novel. The interface of the Color Navigator is a second unique contribution. The combination of color science and a perceptual color space with an existing collection of colors represents a new approach for the problem of picking paint for your walls. It is a tool that could benefit designers who perform this task with some frequency and need a good understanding of the full gamut of options available.

7.2 Going Beyond

There would be little to get excited about with image based rendering if it stayed relegated to the domains of the research lab. However, we have already seen it leap into use in Hollywood and in the commercial market. Blockbuster movies like The Matrix have already employed image based rendering techniques to create special effects and the entire Apple QuickTime VR system is a realization of some of the earliest ideas in the field. But even more recent indicators show a desire to push this technology into the marketplace and drive the research towards more practical uses much like we have proposed in this thesis.

One such example of IBR becoming a real product is the Lytro light field camera [Lytro, 2012]. This is a realization of another branch of image based rendering that we did not discuss called plenoptic photography. A plenoptic camera uses special apertures or arrays of lenses to sample many layers of the light field in front of it with a single click of the shutter instead of just one slice (as with a normal photograph). This yields an image that can be re-focused and slightly tilted around after it has been taken. The Lytro offers all the complicated optics and software necessary to capture and process this data in a unique and inexpensive package. This is the work of Ng [2006] who proposed the hardware and software used in the Lytro in his Ph. D. thesis. Similar to the Lytro, the work of Gardner et al. [2003] is also moving towards commercialization with a company having been created to turn the research scanner built from legos into a practical device for material appearance capture.

Other research projects are working towards similarly practical ends. We have already seen the complex and expensive 3D laser scanner equipment reduced to the size of a dictionary and the cost of a laptop [NextEngine, 2012]. Similar efforts are being made towards practical and pervasive equipment for capturing material appearance. Dong et al. [2010] show the use of a simple handheld device to capture detailed local samples so fewer wide angle pictures are

necessary when sampling a light field. Ren et al. [2011] describe capturing BRDF samples of a material using the camera in your smartphone and a battery powered hand-held light tube. These works push us towards a scanning device that sits on our desk and is capable of capturing digital material appearance.

It is an exciting time to be a part of image based rendering. The field has long promised us the ability to take photographs far beyond their current static nature. But for designers it offers a promise far beyond this. In digital material appearance is a whole new language with which to communicate design intent. When a material lives and breaths on the computer screen responding to light and movement the same as a sample in your hand there is a power to express ideas at a deeper level. Art has always sought to engage the human race at a deeper level than the standards of prose, math and reason allow us. With digital design tools taking us beyond the physical limits of our material samples the technical side of design can serve this purpose as well.

References

Adelson, E. H. and Bergen, J. R. (1991). The plenoptic function and the elements of early vision. *Computational Models of Visual Processing*, 3 – 20.

Alexa, M., Cohen-Or, D., and Levin, D. (2000). As-rigid-as-possible shape interpolation. In *Proceedings of SIGGRAPH '00*, 157 – 164, New York, NY, USA. ACM.

Angenent, S., Haker, S., Tannenbaum, A., and Kikinis, R. (1999). On the laplace-beltrami operator and brain surface flattening. *IEEE Trans. Medical Imaging*, 18(8):700 – 711.

Arad, N., Dyn, N., Reisfeld, D., and Yeshurun, Y. (1994). Image warping by radial basis functions: Application to facial expressions. *CVGIP: Graphical Models and Image Processing*, 56(2):161 – 172.

Artinger, E. (2012). *Relaxing Plant : Concept : Moodboard*. `http://www.eva-artinger.de/relax/concept_moodboard.htm`.

Ashby, M. and Johnson, K. (2002). *Materials and Design: the art and science of material selection in product design*. Butterworth-Heinemann, Oxford.

Baumgarten, A. G. (1954). *Philosophical Reflections on some Matters Pertaining to Poetry*. University of California Press, Berkeley.

Baumgarten, A. G. (1961). *Aesthetica*. G. Olms, Hildesheim.

Beier, T. and Neely, S. (1992). Feature-based image metamorphosis. *Computer Graphics*, 26(2):35 – 42.

Berrier, S., Meyer, G., and Rado, D. (2011). A 3d interface for selecting household paint colors. In *Proceedings of the 19th Color Imaging Conference*, 160 – 165. IS&T.

Berrier, S., Meyer, G., and Shimizu, C. (2008a). Creating metallic color sequences for an architectural wall. In *ACADIA 08: Proceedings of the 28th annual conference of the Association for Computer Aided Design in Architecture*, 308 – 313. ACADIA.

Berrier, S., Shimizu, C., Chong, P., Colucci, D., and Meyer, G. (2008b). The wall of inspiration: A computer aided color selection system. In *CGIV '08: Fourth european conference on colour in graphics, imaging and vision*, 132 – 137. IS&T.

Bier, E. A. (1987). Skitters and jacks: interactive 3d positioning tools. In *Proceedings of the 1986 workshop on Interactive 3D graphics*, 183 – 196, New York, NY, USA. ACM.

Botsch, M., Kobbelt, L., Pauly, M., Alliez, P., and Levy, B. (2010). *Polygon Mesh Processing*. A K Peters, Natick, Mass.

Bouchard, C., Omhover, J.-F., Mougenot, C., Aoussat, A., and Westerman, S. J. (2008). Trends: A content-based information retrieval system for designers. In *Design Computing and Cognition '08*, 593 – 611. Springer.

Buehler, C., Bosse, M., McMillan, L., Gortler, S., and Cohen, M. (2001). Unstructured lumigraph rendering. In *Proceedings of SIGGRAPH '01*, 425 – 432, New York, NY, USA. ACM.

Chen, M., Mountford, S. J., and Sellen, A. (1988). A study in interactive 3-d rotation using 2-d control devices. *Computer Graphics*, 22(4):121 – 129.

Chen, S. E. (1995). Quicktime vr: an image-based approach to virtual environment navigation. In *Proceedings of SIGGRAPH '95*, 29 – 38, New York, NY, USA. ACM.

Chen, S. E. and Williams, L. (1993). View interpolation for image synthesis. In *Proceedings of SIGGRAPH '93*, 279 – 288, New York, NY, USA. ACM.

Chen, W.-C., Bouguet, J.-Y., Chu, M. H., and Grzeszczuk, R. (2002). Light field mapping: Efficient representation and hardware rendering of surface light fields. *ACM Trans. Graph.*, 21(3):447 – 456.

Cheung, K. M., Baker, S., and Kanade, T. (2005a). Shape-from-silhouette across time part i: Theory and algorithms. *Int. J. Comput. Vision*, 62(3):221 – 247.

Cheung, K. M., Baker, S., and Kanade, T. (2005b). Shape-from-silhouette across time part ii: Applications to human modeling and markerless motion tracking. *Int. J. Comput. Vision*, 63(3):225 – 245.

Cohen-Or, D., Solomovic, A., and Levin, D. (1998). Three-dimensional distance field metamorphosis. *ACM Trans. Graph.*, 17(2):116 – 141.

Dana, K. J. and Nayar, S. K. (1998). Histogram model for 3d textures. In *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 618 – 624, Washington, DC, USA. IEEE Computer Society.

Dana, K. J. and Nayar, S. K. (1999). Correlation model for 3d texture. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, 1061 – 1067, Washington, DC, USA. IEEE Computer Society.

Dana, K. J., van Ginneken, B., Nayar, S. K., and Koenderink, J. J. (1999). Reflectance and texture of real-world surfaces. *ACM Trans. Graph.*, 18(1):1 – 34.

Debevec, P., Hawkins, T., Tchou, C., Duiker, H.-P., Sarokin, W., and Sagar, M. (2000). Acquiring the reflectance field of a human face. In *Proceedings of SIGGRAPH '00*, 145 – 156, New York, NY, USA. ACM.

Debevec, P., Wenger, A., Tchou, C., Gardner, A., Waese, J., and Hawkins, T. (2002). A lighting reproduction approach to live-action compositing. *ACM Trans. Graph.*, 21(3):547 – 556.

Debevec, P., Yu, Y., and Borshukov, G. (1998). Efficient view-dependent image-based rendering with projective texture-mapping. In *9th Eurographics Rendering Workshop*, 105 – 116, Vienna, Austria.

Debevec, P. E. and Malik, J. (1997). Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH '97*, 369 – 378, New York, NY, USA. ACM.

Debevec, P. E., Taylor, C. J., and Malik, J. (1996). Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proceedings of SIGGRAPH '96*, 11 – 20, New York, NY, USA. ACM.

Dong, Y., Wang, J., Tong, X., Snyder, J., Lan, Y., Ben-Ezra, M., and Guo, B. (2010). Manifold bootstrapping for svbrdf capture. *ACM Trans. Graph.*, 29(4):98:1 – 98:10.

Dorsey, J., Rushmeier, H. E., and Sillion, F. X. (2008). *Digital Modeling of Material Appearance*. Morgan Kaufmann/Elsevier, Burlington, MA.

Douglas, S. A. and Kirkpatrick, A. E. (1999). Model and representation: the effect of visual feedback on human performance in a color picker interface. *ACM Trans. Graph.*, 18(2):96 – 127.

Dvorine, I. (1944). *Dvorine Color Perception Testing Charts*. The Waverly press, inc, Baltimore, Md.

Einarsson, P., Chabert, C.-F., Jones, A., Ma, W.-C., Lamond, B., Hawkins, T., Bolas, M., Sylwan, S., and Debevec, P. (2006). Relighting human locomotion with flowed reflectance fields. In *Proceedings of Eurographics Symposium on Rendering*, 183 – 194, Nicosia, Cyprus. Eurographics Association.

Foo, S. C. (1997). A gonioreflectometer for measuring the bidirectional reflectance of materials for use in illumination computations. Master's thesis, Cornell University, Ithaca, New York, USA.

Fuchs, M., Blanz, V., Lensch, H. P. A., and Seidel, H.-P. (2005). Reflectance from images: A model-based approach for human faces. *IEEE Trans. Visualization and Computer Graphics*, 11(3):296 – 305.

Gao, Y., Melonakos, J., and Tannenbaum, A. (2006). Conformal flattening itk filter. *The Insight Journal - 2006 MICCAI Open Science Workshop*.

Gardner, A., Tchou, C., Hawkins, T., and Debevec, P. (2003). Linear light source reflectometry. *ACM Trans. Graph.*, 22(3):749 – 758.

Gershun, A. (1939). The light field. *Journal of Mathematics and Physics*, XVIII:51 – 151.

Goesele, M., Lensch, H. P. A., Lang, J., Fuchs, C., and Seidel, H.-P. (2004). Disco - acquisition of translucent objects. *ACM Trans. Graph.*, 23(3):835 – 844.

Gomes, J., Darsa, L., Costa, B., and Velho, L. (1998). *Warping and Morphing of Graphical Objects*. Morgan Kaufmann.

Gortler, S. J., Grzeszczuk, R., Szeliski, R., and Cohen, M. F. (1996). The lumigraph. In *Proceedings of SIGGRAPH '96*, 43 – 54, New York, NY, USA. ACM.

Han, J. Y. and Perlin, K. (2003). Measuring bidirectional texture reflectance with a kaleidoscope. *ACM Trans. Graph.*, 22(3):741 – 748.

Hawkins, T., Cohen, J., and Debevec, P. (2001). A photometric approach to digitizing cultural artifacts. In *VAST '01: Proceedings of the 2001 conference on virtual reality, archeology, and cultural heritage*, 333 – 342, New York, NY, USA. ACM.

Hegel, G. W. F. (1975). *Aesthetics: Lectures on fine art*. Clarendon Press, Oxford.

Hormann, K., Lévy, B., and Sheffer, A. (2007). Mesh parameterization: theory and practice. In *ACM SIGGRAPH 2007 Courses*, New York, NY, USA. ACM.

Itten, J. and Birren, F. (1970). *The Elements of Color: A treatise on the color system of Johannes Itten, based on his book The Art of Color*. Van Nostrand Reinhold Co, New York.

Jeong, E., Yoon, M., Lee, Y., Ahn, M., Lee, S., and Guo, B. (2003). Feature-based surface light field morphing. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, 215 – 223, Washington, DC, USA. IEEE Computer Society.

Joblove, G. H. and Greenberg, D. (1978). Color spaces for computer graphics. *Computer Graphics*, 12(3):20 – 25.

Judd, D. B. and Wyszecki, G. (1963). *Color in Business, Science, and Industry*. Wiley, New York.

Kant, I. (1952). *The Critique of Judgement*. Clarendon Press, Oxford.

Koudelka, M. L., Belhumeur, P. N., Magda, S., and Kriegman, D. J. (2001). Image-based modeling and rendering of surfaces with arbitrary brdfs. In *Proceedings of Computer Vision and Pattern Recognition*, 568 – 575.

Lawrence, J., Ben-Artzi, A., DeCoro, C., Matusik, W., Pfister, H., Ramamoorthi, R., and Rusinkiewicz, S. (2006). Inverse shade trees for non-parametric material representation and editing. In *Proceedings of SIGGRAPH '96*, 735 – 745, New York, NY, USA. ACM.

Lee, S. Y., Chwa, K. Y., and Hahn, J. (1994). Image morphing using deformable surfaces. In *Proceedings of Computer Animation '94*, 31 – 39. IEEE Computer Society Press.

Lee, S. Y., Chwa, K. Y., Hahn, J., and Shin, S. Y. (1996). Image morphing using deformation techniques. *The Journal of Visualization and Computer Animation*, 7(1):3 – 23.

Lee, S. Y., Chwa, K. Y., and Shin, S. Y. (1995). Image metamorphosis using snakes and free-form deformations. In *Proceedings of SIGGRAPH '95*, 439 – 448, New York, NY, USA. ACM.

Lensch, H. P. A., Kautz, J., Goesele, M., Heidrich, W., and Seidel, H.-P. (2001). Image-based reconstruction of spatially varying materials. In *Rendering Techniques 2001: Proceedings of the 12th Eurographics Workshop on Rendering*, 104 – 115, London, Great Britain. Springer.

Lensch, H. P. A., Kautz, J., Goesele, M., Heidrich, W., and Seidel, H.-P. (2003). Image-based reconstruction of spatial appearance and geometric detail. *ACM Trans. Graph.*, 22(2):234 – 257.

Levoy, M. (2006). Light fields and computational imaging. *IEEE Computer*, 39(8):46 – 55.

Levoy, M. and Hanrahan, P. (1996). Light field rendering. In *Proceedings of SIGGRAPH '96*, 31 – 42, New York, NY, USA. ACM.

Lin, Y. and Zhang, W. J. (2006). Integrated design of function, usability, and aesthetics for automobile interiors: state of the art, challenges, and solutions. *Proceedings of the Institution of Mechanical Engineers: Part I, Journal of Systems & Control Engineering*, 220(8):697 – 708.

Litwinowicz, P. and Williams, L. (1994). Animating images with drawings. In *Proceedings of SIGGRAPH '94*, 409 – 412, New York, NY, USA. ACM.

Long, J. and Luke, J. T. (2001). *The New Munsell Student Color Set*. Fairchild Publishing, New York.

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163 – 169.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, volume 2, 1150 – 1157.

Lytro (2012). *Lytro Light Field Camera*. `http://www.lytro.com`.

Madden, B. (1993). Extended intensity range imaging. Technical report ms-cs-93-96, Department of Computer and Information Science, University of Pennsylvania.

Malzbender, T., Gelb, D., and Wolters, H. (2001). Polynomial texture maps. In *Proceedings of SIGGRAPH '01*, 519 – 528, New York, NY, USA. ACM.

Mann, S. and Picard, R. W. (1995). Being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures. In *Proceedings of the IS&T 46th annual conference*, 422 – 428.

Marschner, S. R. (1998). *Inverse Rendering for Computer Graphics*. PhD thesis, Cornell University.

Matusik, W., Buehler, C., Raskar, R., Gortler, S. J., and McMillan, L. (2000). Image-based visual hulls. In *Proceedings of SIGGRAPH '00*, 369 – 374, New York, NY, USA. ACM.

Matusik, W., Pfister, H., Brand, M., and McMillan, L. (2003). A data-driven reflectance model. *ACM Trans. Graph.*, 22(3):759 – 769.

Matusik, W., Pfister, H., Ngan, A., Beardsley, P., Ziegler, R., and McMillan, L. (2002). Image-based 3d photography using opacity hulls. *ACM Trans. Graph.*, 21(3):427 – 437.

McMillan, L. and Bishop, G. (1995). Plenoptic modeling: An image-based rendering system. In *Proceedings of SIGGRAPH '95*, 39 – 46, New York, NY, USA. ACM.

Meyer, G. W. and Shimizu, C. (2005). Computational automotive color appearance. In *Computational Aesthetics in Graphics, Visualization and Imaging*, 217 – 222. Eurographics Association.

Moon, P. H. and Spencer, D. E. (1981). *The Photic Field*. MIT Press, Cambridge, MA.

Müller, G., Meseth, J., Sattler, M., Sarlette, R., and Klein, R. (2005). Acquisition, synthesis, and rendering of bidirectional texture functions. *Computer Graphics Forum*, 24(1):83 – 109.

Nagamachi, M. (1995). Kansei engineering: A new ergonomic consumer-oriented technology for product development. *International Journal of Industrial Ergonomics*, 15(1):3 – 11.

Nagamachi, M. (2006). Kansei engineering and rough sets model. In *Rough Sets and Current Trends in Computing*, volume 4259 of *Lecture Notes in Computer Science*, 27 – 37. Springer Berlin / Heidelberg.

NextEngine (2012). *NextEngine 3D Scanner*. `http://www.nextengine.com`.

Ng, R. (2006). *Digital light field photography*. PhD thesis, Stanford University.

Pahl, G., Beitz, W., Feldhusen, J., and Grote, K.-H. (2007). *Engineering Design: A systematic approach*. Springer, London, 3rd edition.

Parker, S. P. (2007). *McGraw-Hill Encyclopedia of Science & Technology*. McGraw-Hill, New York.

Payne, B. A. and Toga, A. W. (1992). Distance field manipulation of surface models. *IEEE Comput. Graph. Appl.*, 12(1):65 – 71.

Raya, S. and Udupa, J. (1990). Shape-based interpolation of multidimensional objects. *IEEE Trans. Medical Imaging*, 9(1):32 – 42.

Reich, Y. (1993). A model of aesthetic judgment in design. *Artificial Intelligence in Engineering*, 8(2):141 – 153.

Ren, P., Wang, J., Snyder, J., Tong, X., and Guo, B. (2011). Pocket reflectometry. *ACM Trans. Graph.*, 30(4):45:1 – 45:10.

Ruprecht, D. and Müller, H. (1995). Image warping with scattered data interpolation. *IEEE Comput. Graph. Appl.*, 15(2):37 – 43.

Selinger, P. (2003). *Potrace: a polygon-based tracing algorithm*. `http://potrace.sourceforge.net/potrace.pdf`.

Sharma, G., editor (2002). *Digital Color Imaging Handbook*. CRC Press, Hoboken.

Shi, J. and Tomasi, C. (1994). Good features to track. In *CVPR '94: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 593 – 600.

Shiizuka, H. and Watada, J. (2006a). Overview of the kansei system. *Proceedings of the Institution of Mechanical Engineers: Part I, Journal of Systems & Control Engineering*, 220(8):659 – 665.

Shiizuka, H. and Watada, J. (2006b). *Special Issue On Kansei Engineering.*, i – ii. Professional Engineering Publishing.

Shimizu, C. and Meyer, G. W. (2010). Color styling tools. In *18th Color and Imaging Conference*, 272 – 279. IS&T.

Smith, A. R. (1978). Color gamut transform pairs. *Computer Graphics*, 12(3):12 – 19.

Strauss, P. S., Isaacs, P., and Schrag, J. (2002). Course 20: Design and implementation of direct manipulation in 3d. In *SIGGRAPH 2002 Course Notes*, New York, NY, USA. ACM.

Suh, N. P. (1990). *The Principles of Design*. Oxford University Press, New York.

Suh, N. P. (2001). *Axiomatic Design: Advances and applications*. Oxford University Press, New York.

Swackhamer, M. and Satterfield, B. (2007). drapeWALL + drapeHOUSE. *Community Design*, (23):60 – 67.

Tong, X., Zhang, J., Liu, L., Wang, X., Guo, B., and Shum, H.-Y. (2002). Synthesis of bidirectional texture functions on arbitrary surfaces. *ACM Trans. Graph.*, 21(3):665 – 672.

van Breemen, E. J. J., Horváth, I., Knoop, W. G., and Vergeest, J. S. M. (1998). Developing a methodology for design for aesthetics based on analogy of communication. In *Proceedings of the 1998 ASME Design Engineering Technical Conferences*.

Vasilescu, M. A. O. and Terzopoulos, D. (2003). Tensortextures: Multilinear image-based rendering. *ACM Trans. Graph.*, 22(3):7 – 9.

Wang, L., Lin, S., Lee, S., Guo, B., and Shum, H.-Y. (2005). Light field morphing using 2d features. *IEEE Trans. Visualization and Computer Graphics*, 11(1):25 – 34.

Watson, J. (2012). *Palette Generator*. `http://bighugelabs.com/flickr/colors.php`.

Wenger, A., Gardner, A., Tchou, C., Unger, J., Hawkins, T., and Debevec, P. (2005). Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Trans. Graph.*, 24(3):756 – 764.

Wolberg, G. (1990). *Digital Image Warping*. Wiley-IEEE Computer Society Press, Los Alamitos, Calif.

Wolberg, G. (1998). Image morphing: a survey. *The Visual Computer*, 14:360 – 372.

Wood, D. N., Azuma, D. I., Aldinger, K., Curless, B., Duchamp, T., Salesin, D. H., and Stuetzle, W. (2000). Surface light fields for 3d photography. In *Proceedings of SIGGRAPH '00*, 287 – 296, New York, NY, USA. ACM.

Wyszecki, G. and Stiles, W. S. (1967). *Color Science: Concepts and methods, quantitative data and formulas*. John Wiley & Sons, New York, NY.

Zhang, Z., Wang, L., Guo, B., and Shum, H.-Y. (2002). Feature-based light field morphing. *ACM Trans. Graph.*, 21(3):457 – 464.