I N F O R M A T I K

Tutorial Notes for ACM Solid Modeling 2002

A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models

Michael Goesele, Jan Kautz, Jochen Lang,
Hendrik P. A. Lensch, Hans-Peter Seidel

MPI–I–2002–4-001 June 2002
FORSCHUNGSBERICHT RESEARCH REPORT

M A X - P L A N C K - I N S T I T U T
F Ü R
I N F O R M A T I K

Stuhlsatzenhausweg 85 66123 Saarbrücken Germany

Author's Address

Michael Goesele, Jan Kautz, Jochen Lang, Hendrik P. A. Lensch, Hans-Peter Seidel
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany
{goesele, kautz, lang, lensch, hpseidel}@mpi-sb.mpg.de

Keywords

3D data acquisition, Deformable model acquisition, Surface details, High quality 3D models, Rendering

Abstract

This tutorial highlights some recent results on the acquisition and interactive display of high quality 3D models. For further use in photorealistic rendering or interactive display, a high quality representation must capture two different things: the shape of the model represented as a geometric description of its surface and on the other hand the physical properties of the object. The physics of the material which an object is made of determine its appearance, e.g. the object's color, texture, deformation or reflection properties.

The tutorial shows how computer vision and computer graphics techniques can be seamlessly integrated into a single framework for the acquisition, processing, and interactive display of high quality 3D models.

1 Introduction

The rapid advances of consumer level graphics hardware make it possible to render increasingly complex and accurate models in real time. Computer-generated movies are getting more and more realistic and users will soon demand a similar level of realism in a wide range of every day applications such as computer games, digital libraries and encyclopedias, or e-commerce applications. Being able to efficiently generate, process and display the necessary models will become a more and more important part of computer vision and computer graphics.

To fulfill these requirements a high quality representation must capture two different things: the shape of the model represented as a geometric description of its surface and the physics of the material or materials it is made of, e.g. the object's color, texture, deformation or reflection properties. Subsequently, geometry, deformation and surface appearance data must be integrated into a single digital model which must then be stored, processed, and displayed, trying to meet several conflicting requirements (such as realism versus interactive speed).

As more and more visual complexity is demanded, it is often infeasible to generate these models manually. Automatic and semi-automatic methods for model acquisition are therefore becoming increasingly important.

Systems built to acquire and to process the necessary data rely increasingly on computer vision techniques as well as on computer graphics techniques. 3D scanners are becoming the method of choice in acquiring the geometry of an object. The output from these scanners has to be transformed into a mesh representation and further processed to reduce noise and complexity. In the photo studio at the Max-Planck-Institut presented in this tutorial, the optical surface properties of the object are acquired by taking a number of images with constrained lighting. These images have to be registered to the 3D geometry by use of camera calibration techniques. By inspecting the images, the object's texture, the spatially varying reflection properties and microstructure (normal maps) can be extracted. The University of British Columbia active measurement facility ACME is an example of a measurement system capable of acquiring additional physical object properties like an object's sound response, physical surface texture and deformation properties. A brief ACME overview is included in this tutorial and acquisition of deformation behavior in ACME is detailed.

Combining all the data, a compact representation of the object can be obtained that allows for accurately shaded, photorealistic rendering from new viewpoints under arbitrary lighting conditions. In addition, the high quality 3D model may be used for object recognition and material investigation.

This tutorial highlights some recent results on the acquisition and interactive display of high quality 3D models. It shows how computer vision and computer graphics techniques can be seamlessly integrated into a single framework for the acquisition, processing, and interactive display of high quality 3D models. Some examples will illustrate the approach. Finally, we point out some remaining questions and important areas for future research concerning both computer graphics and computer vision.

2 3D Object Acquisition Pipeline

In this tutorial we focus on the generation of high quality 3D models containing the object's geometry and the surface appearance Such a model contains information needed for many computer graphics or computer vision applications. However, there are also other types of high quality models such as volumetric or image-based models (e.g., computer tomography data sets, light fields [40]) that are suitable for different applications.

In our case, the generation of a high quality 3D model for a real world object includes several, partially independent steps. Figure 2.1 shows an overview of these steps.

First, the geometry and the texture of the object are acquired. Typically, different techniques and acquisition devices for the geometry and the texture are applied which makes it necessary to align both data sets in a separate registration step. However, it is also possible to derive geometry information from texture data and vice versa. Various subsequent processing steps are necessary to extract information such as reflection properties or normal maps from the input data.

Once a complete model is created it can be resampled, converted to a different data representation, or compressed to make it suitable for a particular application scenario. Finally, the target application should be able to display the model interactively without omitting any important information.

In the following sections we give a detailed description of all the steps of the 3D object pipeline. Next, we discuss image-based acquisition techniques and introduce two complete measurement set-ups: the photo studio at the Max-Planck-Institut and the ACME facility at the University of British Columbia. In Section 4 we consider the acquisition of an object's deformation behavior, followed by acquisition techniques for appearance properties in Section 5. We give an overview over the acquisition of 3D geometry in Section 6 and describe a technique to register texture and image data in Section 7. Section 8 introduces several methods to display the acquired models interactively. We present some examples of acquired models in Section 9 before we conclude with Section 10.

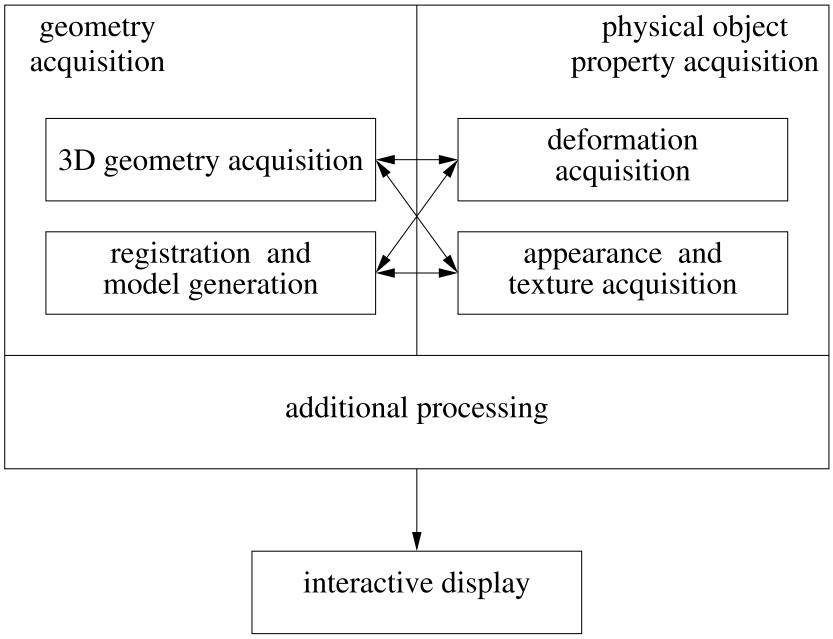


Figure 2.1: *The 3D object pipeline. Depending on the applied techniques geometry acquisition, physical object property acquisition, and registration depend on each other in different configurations.*

3 Image-Based Measurement Equipment

Today, image-based techniques become more and more popular to acquire models of real world objects (see Section 5). A key element of these methods is a camera to capture images of the object from which various properties of the object can be derived. The large number of measurements that can be made in parallel (i.e. one per pixel for a digital camera) lead to efficient methods to sample complex functions such as four-dimensional BRDFs. However, these measurements can only be meaningful if the equipment used is appropriate for the measurements, if the properties of the devices are known, and if the relevant parts are calibrated. Another important consideration in designing a measurement facility is the level of automation the facility is to provide. Beside the appropiate choices in sensor and lighting equipment, robotic devices can dramatically reduce human effort during model acquisition. Additionally, these devices are able to provide calibrated position and orientation of objects, sensors and actuators. Robots have the ability to perform tasks with high repeatability and consistent accuracy. This is extremely important if a large number of registered measurements of an object are to be collected.

3.1 Photographic Equipment

Both analog and digital cameras can be used for measurement purposes. The advantages of analog photography include the high resolution of analog film (especially in combination with commercial high quality digitization services as the Kodak Photo CD), its comparably large dynamic range, and the huge selection of available cameras, lenses and film types. However, the development and scanning of film can take quite long and the resulting images are not naturally registered against the camera lens system.

In contrast to that, the position of the imaging sensor in a digital camera re-

Figure 3.1: *A view of our photo studio with black, diffuse reflecting material on the floor, walls, and ceiling. This image was generated from a High Dynamic Range image to which a tone-mapper has been applied.*

mains fixed with respect to the lens system which makes it easy to capture several aligned images from the same position under different lighting conditions. If the digital camera is capable of returning the raw image sensor data it is possible to calibrate the individual sensor elements to account for variations on the sensor [1, 17].

Most consumer quality digital cameras use the lossy JPEG compression format to store their images although more recent cameras are often also capable of producing images in a lossless compressed format. The lossy JPEG compression introduces compression artifacts which makes them rather unsuitable for measurement purposes. Additional artifacts can occur due to various steps in the image processing chain of digital cameras such as sharpening operations or the color reconstruction in single chip cameras. The imaging community developed a large number of methods to characterize various aspects of a digital camera such as the modulation transfer function (MTF) [68]. These methods are not only helpful to choose an appropriate camera but can also be used to debug a measurement setup when an error occurs.

3.2 Lighting Equipment

For most algorithms that reconstruct the appearance properties of an object from images, it is important to control the lighting conditions exactly. Although this is also true for images taken by a regular photographer, the requirements differ strongly. A point light source, i.e. a light source where all light is emitted from a single point is ideal for many of the techniques mentioned above but is rarely used in photography as it casts very hard shadows. A perfectly constant and diffuse lighting is ideal to capture the color of an object but leads from a photographers point of view to very flat looking images due to the absence of shadows.

The surrounding of an object has also a huge influence on the lighting situation, especially if the object has a specular reflecting surface. In order to minimize this influence the measurement region should be surrounded with dark material that absorbs as much light as possible. Furthermore, the light that is not absorbed should be reflected in a very diffuse way. Figure 3.1 shows a view of our photo studio whose floor, walls, and ceiling are covered with black, diffuse reflecting material to reduce the influence of the environment on the measurements as much as possible.

A more technical and in-depth discussion of camera and lighting issues can be found in [15].

3.3 Robotic Equipment

This tutorial dicusses the automation of the measurement process with an example: the University of British Columbia Active Measurement Facility (ACME). ACME is an integrated robotic facility designed to acquire measurements of interactions with objects. At the core of ACME is a contact manipulation system (CMS) which executes contact interactions with an object to be measured. These interactions can be recorded with various sensors including a force-torque sensor, a trinocular stereo system, a microphone and a high quality 3-CCD color video camera. Sensors which are employed to measure the response of an object from a distance are combined in a field measurement system (FMS). The test station is a positioning robot to move the object to be measured with precision. The CMS is built around a robot arm positioned by a motion stage. A gantry robot places the FMS relative to an object. All these robots are commercial devices but their assembly, control and programming are unique to ACME. This tutorial discusses how ACME is employed in the acquisition of measurements during active object deformation. The geometric modeling capability of ACME are not discussed here.

3.3.1 Acme Overview

The design goal of ACME is to make it efficient to build *reality-based* models [49], i.e., interactive computational models of real, physical objects based on actual measurements. The physical layout of the facility with its variety of sensors and actuators is shown in Figure 3.2. The control of these devices is distributed over a number of computers and layers. This allows one to employ commercial con-

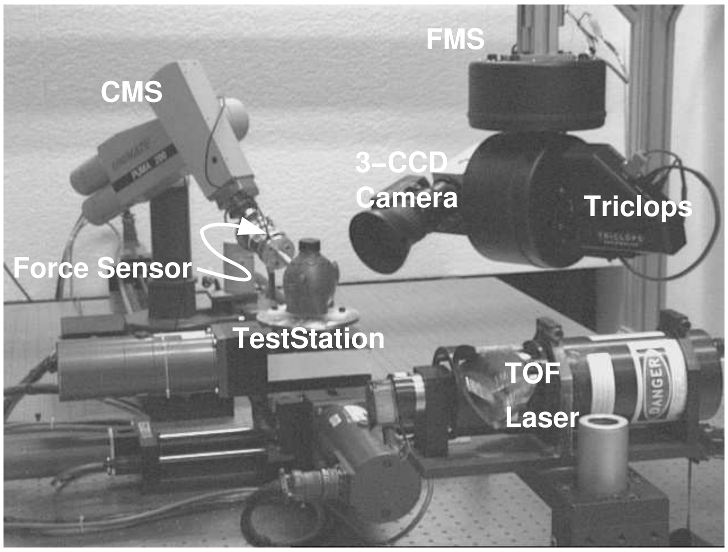


Figure 3.2: *ACME Facility Overview*

trollers when possible but adding advanced features as required. In particular, the control architecture provides the foundation for the high-level control of the system which makes programming it simple.

The architecture is based on tele-robotic control and integrates a kinematic simulator with the tele-robotic paradigm in a number of ways. On the one hand the simulator serves as a debugging tool for ACME experiments. An ACME experiment is the top-level user program written in Java[1]. On the other hand the simulator is available during execution to verify motion requests. This allows for a simple generate-and-test approach to motion planning. The control architecture of ACME is client-server based, i.e., an ACME client with which a user interacts and an ACME server which controls the robotic subsystems. The client program can execute independently from the facility or act as a terminal for ACME. The ACME server system is distributed between several computers; separating low-level closed-loop control, trajectory generation, data acquisition, and networking. The four layers of the ACME server software from top to bottom are the user's *Experiment*, the Java ACME device classes, the native bindings to the devices and finally the low-level device controller. The native bindings to most of the actuators and to some sensors are through the Robot Control C Library (RCCL) [42] which generates real-time trajectories for each actuator device from a high-level *MotionPlan* composed of device indepedent *Motion* objects.

The interested reader is directed to [51, 50] for a more complete description of ACME. Section 4.2 also briefly summarizes the ACME deformation experiment.

[1]Java is a trademark of Sun Microsystems Inc., MountainView, Ca., USA

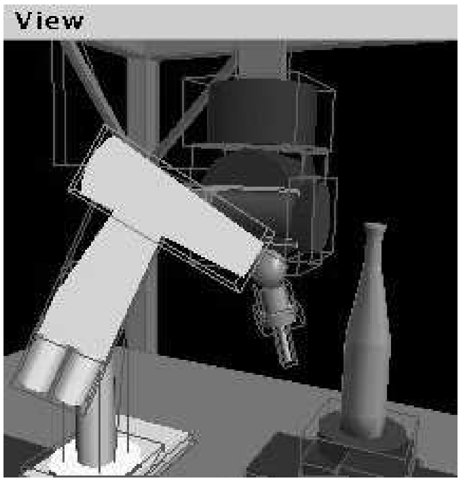


Figure 3.3: *ACME 3-D Display. A collision event produced by the simulation is rendered with the colliding boxes highlighted.*

3.4 Camera Calibration

When using a camera as a measurement device various aspects should be calibrated in order to guarantee high-quality results and the repeatability of the measurements.

3.4.1 Geometric Calibration

The properties of the camera transformation which describes how an object is projected onto the camera's image plane should be recovered e.g. using [65, 71, 20]. These methods generally use an image or a set of images of a calibration target (e.g. a checkerboard pattern) to determine camera parameters such as the focal length of the lens, the location of the optical axis relative to the imaging sensor (principal point), and various distortion coefficients. Once this information is known, a ray in space can be assigned to each pixel in an image.

3.4.2 High Dynamic Range Imaging

The dynamic range of a camera, i.e. the ratio between the brightest and the darkest luminance sample that can be captured in a single image, is for most cameras quite small (on the order of $10^2 - 10^3$). As the dynamic range of a scene can be much higher (e.g., about 10^6 between highlight and shadow regions), some techniques have to be used to capture the full dynamic range of a scene.

Several manufacturers have developed CMOS cameras that are capable of capturing a sufficiently large dynamic range by either combining multiple exposures or by the use of special imaging sensors. These cameras are typically video cameras and provide only a limited resolution. Furthermore, the measured values are quantized to 8–12 bits per pixel and color channel leading to a rather low precision.

In the computer graphics community, several authors proposed methods to extend the dynamic range of digital images by combining multiple images of the same scene that differ only in exposure time. Madden [44] assumes linear response of the imaging sensor and selects for each pixel an intensity value from the brightest non-saturated image. Debevec and Malik [11] and Robertson et al. [55] recover the response curve of the imaging system and linearize the input data before combining them into a single high dynamic range image. In [16], Goesele et al. proposed a technique to combine high dynamic range imaging with color management techniques (see Section 3.4.3).

3.4.3 Color Issues

Accurately recording the continuous spectrum of the visible light is difficult – especially if the spectrum is not smooth but contains sharp peeks such as the spectrum of a discharge lamp or even a laser. Likewise, the spectral response curve that describes the way light is reflected by an object is not always smooth. Measurement devices such as a spectrophotometer perform therefore a very dense sampling of the spectrum and output large data sets.

In contrast to that, most analog and digital cameras record only three color values per pixel (tristimulus values). Each sensor in a digital camera integrates the amount of incoming light weighted by its response curve over the whole visible spectrum. This is inspired by the human visual system that also contains three types of sensors behaving in a similar way [21]. A camera can record the colors of objects as perceived by a human observer most accurately if the corresponding response curves are identical [43], but the true spectrum of the light hitting the sensor can never be reconstructed and different spectra can result in the same tristimulus values (metamerism). Color measurements done with a tristimulus device are therefore always an incomplete representation of the actual spectrum.

White Balance

The human visual system can adapt to a wide range of illumination conditions. Within this range, colored objects look roughly the same even if the spectrum of the light source changes and therefore the spectrum of the reflected light hitting the retina is different. A digital camera can mimic this behavior with a white

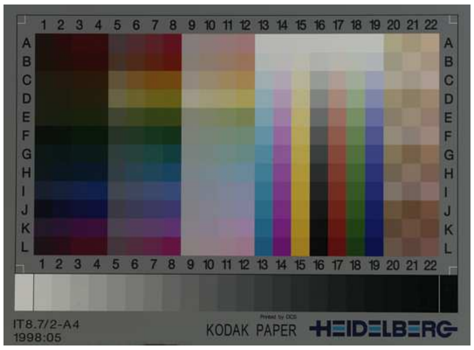


Figure 3.4: *IT8.7/2 target used to capture the color properties of an imaging system in order to generate an ICC profile.*

balancing step: the tristimulus values are multiplied with constant factors so that the color of the light source is recorded as white. The influence of the light source on the recorded color of an object is hereby minimized.

Color Management Systems

For a digital camera, the recorded color of an object depends not only on the light source but also on several other factors including the properties of the optical system, the sensor, and the image processing steps applied by the camera itself or other software.

In order to relate the recorded color to well defined standards, color management systems have become a standard tool. An image of a well known test target such as the IT8.7/2 target (see Figure 3.4) is taken and processed in the same way all later images are processed. The relation between the color values of the test target patches and the color values reported by the camera is analyzed and used as calibration data. The International Color Consortium (ICC) introduced the so called ICC profiles [23, 66] as a standard way to store this information.

The basic mechanism behind ICC based color management systems is to use a well defined color space as profile connection space (PCS). All input data is converted into the PCS using an ICC input profile associated with the input device. Other profiles are used to convert data from the PCS into the color space of display or output devices such as monitors and printers.

One of the color spaces used as PCS is the linear CIEXYZ space [8]. In [16], Goesele et al. have shown that this color space can be used to generate color calibrated high dynamic range images which are a tool to improve the color fidelity of appearance acquisition methods.

4 Deformation Acquisition

This tutorial shows how to efficiently scan deformation behavior in the ACME facility. The scanning of deformation behavior in ACME is a direct measurement method based on the observation of a deforming object (see Figure 4.1). Elastic object deformation is characterized by surface displacements and contact forces. ACME records the force and displacement at the contact area by a robotic probe while it actively deforms the object. The global displacement of the free surface is observed visually with a trinocular stereo vision system. The three-dimensional displacement vectors over the surface are obtained with simultaneous (geometric) stereo and optical flow. The scanning of the initial (undeformed) shape of an object in ACME is also based on stereo range data.

Deformable models of physical objects for interactive simulation are desirable for numerous applications in the areas of virtual and augmented reality. Many techniques have evolved in interactive simulation over the years to model deformations [14]. These techniques vary from full continuum models for elastic materials [7, 25, 72, 53, 62] through mass-spring particle systems [10], ad-hoc methods [14]. But even the most elaborate deformation model will be unrealistic if the simulated material does not match the real physical object. The measurement of these material properties is conventionally done in a "material oriented" way, e.g., in physical and mechanical testing laboratories. However, if deformable models are not based on individual physical objects, simulation results can not be validated by observation of individual behavior. As a result, most approaches to deformable simulation lack a validation step [14, 13].

Closest to the acquisition method discussed in this tutorial is the model acquisition in the work on interactive medical simulators by Laugier and his group [10]. They model the force interaction between an echo-graphic sensorhead and the human thigh with a particle system for which parameters are derived from local measurements with a robotic probe. However, the approach discussed in this tutorial is unique in that it captures the complete static deformation response of an object. In a general approach, the discrete Green's functions of an object's deformation behavior are robustly estimated from measurements of global deformation.

Figure 4.1: *Active Deformation Measurement in ACME: The trinocular stereohead records the global surface deformation while the robotic probe applies a force and measures local displacement.*

4.1 Deformable Model

In this tutorial the global deformation of a solid in static equilibrium is modeled with a discrete Green's functions matrix. The discrete Green's functions matrix can be derived for a homogeneous isotropic linear-elastic solid based on continuum mechanics. Given a suitable discrete representation of the undeformed object shape and known material constants, either the finite element solution method with condensation [7] or the boundary element solution method [25] may be employed to this end. The discrete Green's functions of the boundary value problem associated with an elastic solid can generate the solution for all possible boundary conditions similarly to the impulse response of an electric circuit. Next, a brief introduction to the discrete Green's functions matrix is given.

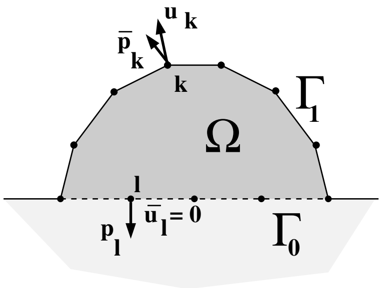


Figure 4.2: *Elastic Solid: Discrete Boundary Value Problem*

The domain of the problem is a solid Ω with its boundary Γ. On the boundary Γ_0 displacements are prescribed and on the boundary Γ_1 tractions[1] are prescribed,

[1]A traction $\mathbf{p}$ is a force $\mathbf{f}$ normalized over the surface area A it is affecting, i.e., $\mathbf{p} = \mathbf{f}/A$.

(see Figure 4.2), i.e., Γ_0 is fixed while Γ_1 is free. The Green's functions relate a field of displacement vectors $\mathbf{u}$ to a field of traction vectors $\mathbf{p}$ on the boundary of the elastic solid. In our work the boundary of the model is discretized into a triangular mesh with $k = 0 \ldots n$ vertices (see Figure 4.2). For a given boundary configuration, the block vector $\bar{\mathbf{v}}$ holds all prescribed boundary values. Entries $\bar{\mathbf{v}}_k = \bar{\mathbf{u}}_k$ if the vertex k is on the surface Γ_0, while $\bar{\mathbf{v}}_k = \bar{\mathbf{p}}_k$ if the vertex k is on the surface Γ_1. The complementary displacement and traction vectors are entered into a block vector $\mathbf{v}$. The matrix relating the prescribed values $\bar{\mathbf{v}}$ and $\mathbf{v}$ is the discrete Green's functions matrix Ξ (see Equation 4.1). Equation 4.1 is the deformable model employed in this tutorial.

$$\mathbf{v} = \Xi \bar{\mathbf{v}} \tag{4.1}$$

In estimation, the Green's functions matrix Ξ is to be determined from over-specified displacement and traction boundary conditions. The block vectors $\mathbf{v}$ and $\bar{\mathbf{v}}$ are measured or known in the estimation of the Green's functions matrix Ξ.

4.2 Deformation Measurement in ACME

A deformation *Experiment* in ACME applies a concentrated load with a robotic arm on the free surface of the object while the object is fixed on a positioning *TestStation* as shown in Figure 4.1. During measurement the probe applies a force until the maximum force or displacement is reached. Force and displacement are recorded at 100Hz. The measurement of the global deformation is based on range-flow calculated from imagery acquired with a trinocular stereo-head [35]. The vertex displacements are measured visually for all vertices which are in view from the stereo-head at a given location. During the scanning of an object, data is gathered for each vertex location on the free surface of of the undeformed object mesh. Multiple load applications in different orientations are combined with multiple viewpoints for the trinocular stereo-head.

4.3 Model Estimation

The estimation of the discrete Green's functions is treated here as a linear estimation problem based on Equation 4.1. The measurement set-up defines the boundary configuration for the estimation procedure. In the measurement configuration, vector $\bar{\mathbf{v}}$ of Equation 4.1 consists of zero displacements for the fixed surface and zero tractions for the free surface except for the contact area. As a result the block vector $\bar{\mathbf{v}}$ is all zero, except for a single traction vector at the contact point at vertex k. This leads to the structure illustrated in Equation 4.2. The behavior of the model is characterized by the $m \times m$ block submatrix of Green's functions matrix Ξ which can be measured in the approach discussed in this tutorial. Columns in Ξ which correspond to vertices of the fixed support surface are not excited during measurement. The rows of Ξ which correspond to the same vertices of the fixed support surface are not observed since the corresponding tractions in $\bar{\mathbf{v}}$ are not sensed.

$$\left[\begin{array}{ccc|c} & \Xi_{1k} & & \\ & \vdots & & never \\ \ddots & \Xi_{kk} & \ddots & excited \\ & \vdots & & \\ & \Xi_{mk} & & \\ \hline \multicolumn{4}{c}{unobservable} \end{array}\right] \left[\begin{array}{c} 0 \\ \vdots \\ \bar{\mathbf{p}}_k \\ \vdots \\ 0 \\ \hline 0 \end{array}\right] = \left[\begin{array}{c} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_k \\ \vdots \\ \mathbf{u}_m \\ \hline not \\ sensed \end{array}\right] \tag{4.2}$$

The estimation problem is: given a set of block vectors $\bar{\mathbf{v}}$ and a set of corresponding block vectors $\mathbf{v}$ find the $m \times m$ observable submatrix of Ξ. We divide the problem into m separate estimation problems for each of the m columns. There are m estimation problems per column which consist of finding a 3×3 block element Ξ_{ik} of matrix Ξ given a set of observed displacement vectors $\mathbf{u_i}$ and the corresponding traction vectors $\bar{\mathbf{p}}_\mathbf{k}$. The problem when $i = k$ is the estimation of the compliance of the object at location k.

Noise in the measurement data requires regularization in the estimation process [36]. The estimation process must also deal with missing observations in the global deformation response ($i = k$) due to occlusion, failure of the range-flow technique and incomplete sensor coverage of the object's surface. In the approach described in Pai et al. [52], hole filling is achieved by a diffusion process on the triangular surface mesh. It is also shown how to estimate the response for different level of detail of the surface mesh.

4.4 Modeling Examples

Below we summarize results obtained in ACME with the above outlined method. The first example is the result for the local stiffness estimation of a plush toy (the global response can be found in [34]). In the second example, the global deformation results for modeling an anatomic soft-tissue human wrist model[2] with

[2]The object is made by Sawbones, Pacific Research Laboratories, Inc. (http://www.sawbones.com).

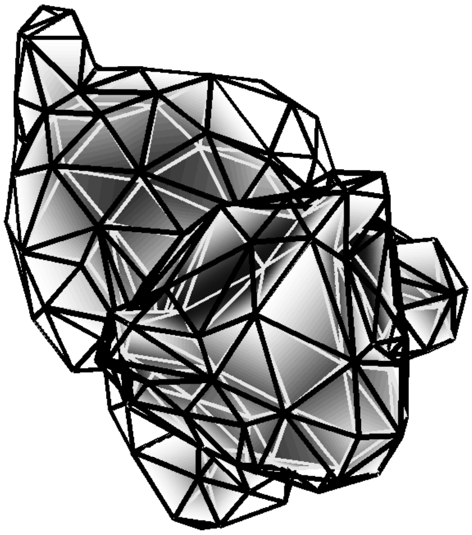


Figure 4.3: *Normal Local Compliance. Compliance normal to the surface is shown ($\Xi_{kk}n_k$). The figure is vertex colored, dark indicates a large value. Vertices not probed due to reachability constraints result in Ξ_{kk}^T not being estimated and are shown in white. A maximum of eight probes observed from two different viewpoints are employed.*

this method are given.

The plush toy has a quite uniform compliance with exception of the rear of the head which moves (hinges) more easily out of the way of the probe (see Figure 4.3). The anatomic soft-tissue human wrist model has a bone structure made of hard plastic with a cancellous inner core. The bone structure is surrounded by soft foam with a plastic skin layer. Therefore, the deformation response of the wrist model is quite non-uniform. We arbitrarily pick two Green's functions for illustration. The load application for these two cases are at vertices $k = 4$ and $k = 98$, respectively (see Figure 4.6). The lower arm deforms locally when probed at Vertex 98 (see Figure 4.6(c) and Figure 4.6(d)) with no preferred direction. This response is caused by the *soft tissue* of the object being compressed between the probe and the object's bone structure. When the model is probed at vertex $k = 4$ above the wrist joint, the complete object bends (see Figure 4.6(a) and Figure 4.6(b)) since the bone structure is not rigid enough to restrict the deformation.

Lang et al. [36] also compare the results of the least squares estimation of the discrete Green's function matrix with a linear-elastostatic continuum mechanics model obtained by assuming isotropy and homogeneity. The limitations of the homogeneous isotropic linear elastostatic model have been pointed out in the lit-

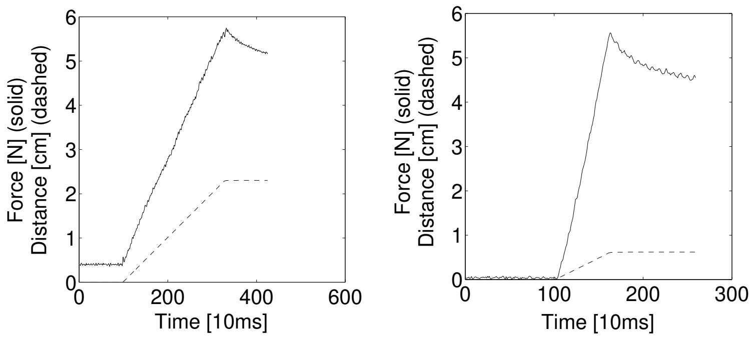
(a) Vertex 4 (b) Vertex 98

Figure 4.4: *Probe Tip of ACME's CMS at Vertex Location*

erature before [72, 53]. The solution suggested is to model material with the non-linear large deformation strain tensor [72, 53]. In contrast, the method discussed here will find a linear approximation to the physical deformation of an object. This linear approximation achieves realistic global deformation by making full use of all the degrees of freedom of the discrete Green's functions matrix. It also maintains the linearity of the model which then can be rendered easily with several thousands of boundary nodes in real-time while being stable by design [24].

4.5 Summary of Deformation Acquisition

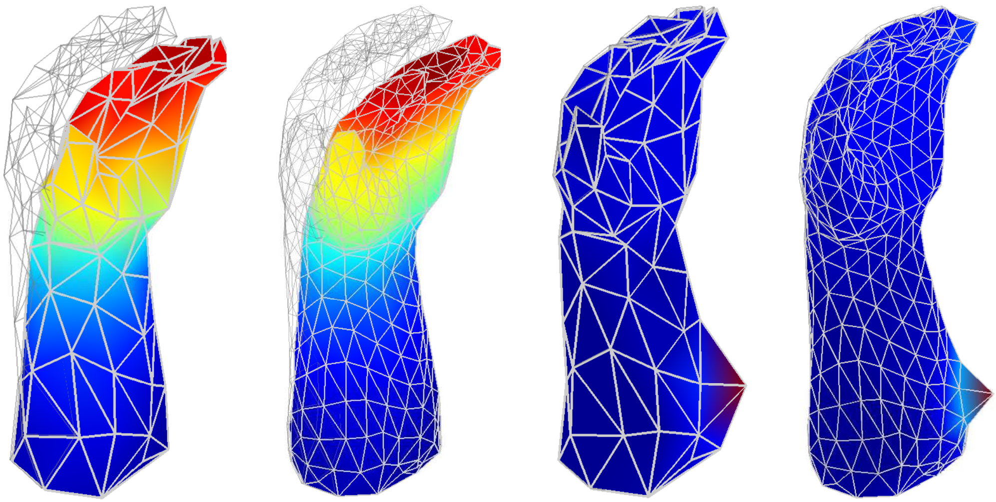
The robotic acquisition of deformable models leads to an increase in realism of global deformation models of physical objects. The method discussed here estimates the discrete Green's functions matrix for elastic objects based on direct observations of the deformation behavior. The robotic measurement facility ACME enables the convenient acquisition of the necessary measurements. ACME measures the local contact behavior at a robotic probe, as well as the global displacement of the object's surface. The least squares estimation of the block elements of the discrete Green's function matrix uses regularization in the solution method. The local compliances for a plush toy are successfully estimated with this method. The discrete Green's functions estimation also provides a reasonable linear approximation to the deformation behavior even in the case of the anatomic soft tissue human wrist model. This is despite the fact that the anatomic soft tissue human wrist model is an inhomogeneous, anisotropic and articulated soft body.



(a) Probe at Vertex 4 of Figure 4.4(a)

(b) Probe at Vertex 98 of Figure 4.4(b)

Figure 4.5: *Magnitude of Force and Displacement. The profiles of displacement and force magnitude as recorded by ACME's CMS.*



(a) Probe at Vertex 4 of Figure 4.4(a)

(b) Probe at Vertex 4 of Figure 4.4(a)

(c) Probe at Vertex 98 of Figure 4.4(b)

(d) Probe at Vertex 98 of Figure 4.4(b)

Figure 4.6: *Discrete Green's Functions for Different Mesh Resolution. Base mesh (level* 0*) of the subdivision hierarchy (4.6(a) and 4.6(c)) and level* 1 *(4.6(b) and 4.6(d)) are shown. The estimation for both meshes is based on the same observations.*

5 Appearance Acquisition

The appearance of an object consists of several surface properties including color, texture, reflection properties, and normal directions or the local tangent frame in the case of anisotropic materials. Due to their large number they are difficult to acquire but nevertheless necessary to generate a convincing looking representation of an object. It is therefore justifiable to put a lot of effort into this acquisition step.

Traditionally the appearance of an object is captured using a variety of special devices [22]. But many surface properties can be acquired by the use of a photographic camera – preferably a digital camera – in a controlled lighting setup. Captured images can for example be used to color the 3D geometry model during rendering. The digital pictures are simply projected onto the model as image textures using texture mapping [18]. To ensure that each part of the object is colored, a sufficient number of images must be taken from different view points [47, 61]. During the projection a perspective correction must be performed to gain a seamless transition between textures of different images (see also Section 7). To obtain more precise surface properties than just a single color value, further processing is needed.

5.1 Reflection Properties

Constant, diffuse lighting during the acquisition phase would reproduce only the object's color. More realistic models can be obtained by considering further aspects of a material's appearance, for example the reflection properties. The intensity and color of any material typically varies if viewed from different directions or under different illumination (see Figure 5.1).

When light interacts with a perfectly reflective surface, i.e. a mirror, the reflected light leaves the surface at the same angle it hits the surface. However, perfect mirrors do not exist in reality. In contrast, most surface have a very complex micro-structure. This micro-structure makes different materials appear differently.

When light hits such a surface, it is not reflected toward a single direction, but

Figure 5.1: *A teapot with complex reflection properties illuminated from two different directions.*

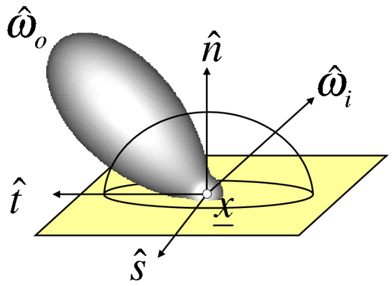


Figure 5.2: *Here you can see the values of a BRDF (depicted as a lobe) for one incident light direction $\hat{\omega}_i$ and every possible outgoing direction $\hat{\omega}_o$.*

rather to a cone of directions. If the surface is perfectly diffuse (e.g. for a piece of chalk), light even scatters equally in all directions.

In computer graphics the *bidirectional reflectance distribution function* (BRDF or also reflectance model) is used to describe the way a surface reflects light. The BRDF yields the fraction of light arriving at a point from one direction to the light that is reflected off the surface at the same point into an exitant direction.

Hence a BRDF is a four-dimensional function $f_r(\hat{\omega}_o, \hat{\omega}_i)$ that depends on the incident light direction $\hat{\omega}_i$ and the viewing direction $\hat{\omega}_o$ (see Figure 5.2). It should be noted, that it also depends on the wavelength, which is usually represented by three samples (RGB) only. In the following, the wavelength dependency is not stated explicitly.

A number of analytical BRDF models have been developed to approximate the reflection properties of real materials (e.g. [64, 67, 33, 2]).

5.2 Measuring Reflection Properties

In addition to these analytical models, it is possible to measure real-world BRDFs directly. There are special devices available to accomplish this task: The most general approach is to use a gonioreflectometer which measures the light that is emitted in every direction when the object is illuminated from a given direction. However, this measurement procedure can be very time consuming and captures only the properties of a single point on the surface of an object. If the surface is not uniform, this is not very helpful.

One way to overcome the ”single point” constraint for appearance measurements is the use of a digital camera. When an image is taken with such a camera it corresponds to millions of parallel measurements of radiance samples hitting the sensor. The main challenge is to recover the appearance information from images taken from different positions under controlled lighting conditions.

Marschner [45] used this approach to determine a single BRDF for an object by combining all the pixel data. Compared to a gonioreflectometer this technique is considerably faster, but it still assumes that the entire object consists of a single material, represented by a large number of tabulated BRDF samples. A specific BRDF model can be fitted to these BRDF samples by optimizing for the parameters of the BRDF model as it is for example done in [59]. The set of BRDF samples is then replaced by a few parameters resulting in a more compact representation.

To allow for variations of the reflectance properties over the object’s surface Marschner et al. [46] extracted the purely diffuse part (albedo map) of the object’s texture for each visible point using a similar technique. The resulting texture includes only view-independent color information and no specular reflection. Albedo maps plus one reflection model per surface patch have been acquired for indoor scenes by Yu et al. [70] which assumed that material properties only change from patch to patch.

An approach to acquire distinct reflection properties for every surface point has been published by Debevec et al. [12]. A set of images of an object, e.g. a person’s face, is taken from one viewpoint while the position of a point light source is changed. Hereby, the set of incident light directions is densely sampled. The collected data allows for realistic relighting of the object illuminated by arbitrary virtual environments. Unfortunately, a very large amount of data is needed both during the acquisition and for display.

5.3 Measuring Spatially Varying BRDFs

Based on Marschner's approach, Lensch et al. [37] developed a technique that is able to reconstruct spatially varying reflection properties by just a very few images (around 25). The key idea here is that most objects typically consist of a small number of materials only, i.e. many points on the object's surface have approximately the same reflection properties. By clustering points with different normals but consisting of the same materials, a large number of BRDF samples of that material can be collected by just a few images. After measuring the BRDF for clusters of points, separate reflection properties for each single point are determined to account for subtle details and small changes. The BRDF for each point is determined as a weighted sum of the clusters' BRDFs.

Thus, a high quality and very compact representation of the original object can be obtained with moderate acquisition effort.

5.3.1 Data Acquisition

The entire procedure is as follows: The geometry of the object is obtained by use of a 3D scanner, e.g. a structured light or computer tomography scanner, yielding a triangle mesh. In order to capture the reflection properties a small number of high dynamic range (HDR) images of the object are taken showing the object lit by a single point light source. In a next step the camera position (see Section 7) as well as the light source position relative to the geometric model are recovered for all images.

For every point on the object's surface all available data (geometric and photometric) is collected from the different views in a data structure called *lumitexel*. It contains the position of the surface point and its normal derived from the triangular mesh. Additionally, a lumitexel stores a list of radiance samples together with the corresponding viewing and lighting directions, one radiance sample for every HDR image where the point is visible and lit. The radiance sample is obtained by resampling the color value at the position of the surface point projected into the image.

5.3.2 Clustering of Materials

Because only a limited number of different views and lighting directions is acquired a single lumitexel does not carry enough information to reliably fit a BRDF model to the radiance samples. To provide more data from which the parameters can be derived, the lumitexels are grouped into clusters of similar materials. Starting with a single cluster containing all lumitexels, the parameters of an average

BRDF are fitted using the Levenberg-Marquardt algorithm to perform a non-linear least square optimization.

In order to separate the distinct materials the initial cluster has to be split. Given the average BRDF, two new sets of parameters are generated by varying the fitted parameters along the direction of maximum variance, yielding two slightly distinct BRDFs.

The lumitexels of the original cluster are then assigned to the nearest of these BRDFs, forming two new clusters. A stable separation of the materials in the clusters is obtained by repeatedly fitting BRDFs to the two clusters and redistributing the original lumitexels. Further splitting isolates the different materials until the number of clusters matches the number of materials of the object as illustrated in Figure 5.3.

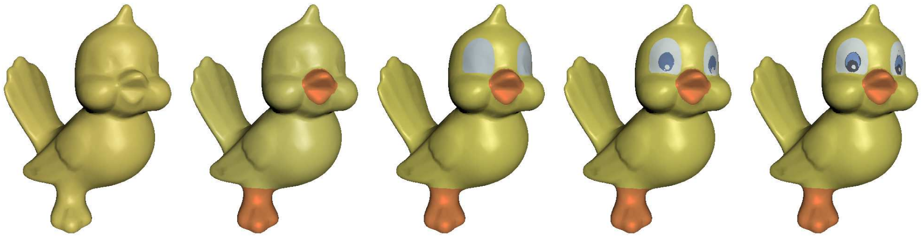


Figure 5.3: *The clustering process at work. In every image a new cluster was created.The object was reshaded using only the single BRDFs fitted to each cluster before the projection into a basis of multiple BRDFs.*

5.3.3 Spatially Varying Behavior

After the clustering the same reflection behavior is assigned to all lumitexels/points in one cluster. However, small features on the surface and smooth transition between adjacent materials can only be represented if every lumitexel is assigned its own BRDF.

In the algorithm, this BRDF is a weighted sum of the BRDFs recovered by the clustering procedure. The spatially varying reflection properties can be represented by a set of basis BRDFs for the entire model plus a set of weighting coefficients for each lumitexel.

The weighting coefficients are found by projecting the lumitexel's data into the basis of per cluster BRDFs. An optimal set of weighting coefficients minimizes the error between the measured radiance and the weighted sum of radiance values obtained by evaluating the basis BRDFs for the viewing and lighting direction of the measured sample. To recover the coefficients the least square solution of the

Figure 5.4: *Left: Last result of the clustering step. Right: Bird with the spatially varying BRDF determined by projecting each lumitexel into a basis of BRDFs. Note the subtle changes of the materials making the object look realistic.*

corresponding system of equations is computed using singular value decomposition (see [37] for more details).

In Figure 5.4 the result of projecting the collected data for every point into a basis of BRDF is shown. The method allows for accurately shaded, photorealistic rendering of complex solid objects from new viewpoints under arbitrary lighting conditions with relatively small acquisition effort. The reconstructed BRDFs can further be used to classify the objects based on their materials.

5.4 Normal Maps

The resolution of the acquired geometry of an object is typically limited by the used 3D scanning device (see Section 6). Additional processing of the 3D data like combining multiple scans, smoothing the surface to remove noise, and mesh simplification to reduce the complexity of the model further erases fine scale geometric detail.

When reconstructing the object using a coarse geometric model, smaller features in the surface's structure like bumps, cracks or wrinkles can be simulated by the use of normal maps or bump maps [5] (see Figure 8.1). These textures store a perturbation of the surface normal for each surface point. After applying the perturbation, the modified normals are used for the lighting calculations. This results in a change of the angle between the viewing direction and the surface at that point as well as between the light direction and the surface. This step approximates the correct lighting of a fine scale geometry model.

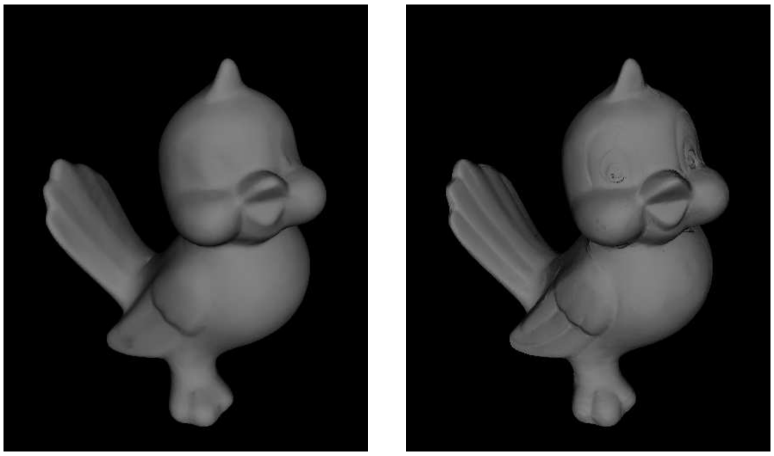


Figure 5.5: *Left: Normals of the original mesh. Right: Normals optimized using spatially varying BRDFs*

Normal maps recording small imperfections of the surface can be acquired for real world objects: Rushmeier et al. calculated normal directions from a set of images showing the same view of the object illuminated by a point light source placed at different but known positions for each image [58]. The surface is assumed to be perfectly diffuse (Lambertian), reflecting incident light equally in all directions, and thus its color can again be represented by an albedo map [57].

The restriction of a purely diffuse surfaces can be removed if techniques like [37] (see Section 5.3) are used to first measure the approximate reflection properties at each surface point and then use this data to measure the normal directions.

Since the BRDF at one point is defined for viewing and lighting directions with respect to the local tangent frame at that point, all directions have to be transformed based on the point's surface normal. To measure the exact normal at a point, an initial normal is obtained from the triangular mesh. Given the viewing and lighting directions for the radiance samples in world coordinates, the current estimate of the normal is used to transform them into the local coordinate frame. Then, the error between the measured radiance values and the reconstructed radiance values is computed where the reconstructed radiance values are obtained by evaluating the measured BRDF using the transformed directions. If enough radiance samples are provided for each point the actual normal direction at the point can be found by minimizing this error using a non-linear least square optimization technique. Figure 5.5 shows the quality of the reconstructed normals compared to the normals of the original mesh.

6 Acquisition of 3D Geometry

In most cases there exists no high quality 3D geometry model of real world objects like pieces of art. But even if it would exist (e.g. because the object was manufactured using computer based manufacturing methods) it is often only available to a very limited number of persons. Therefore, it is most often necessary to acquire the geometry of objects using a 3D scanner.

Several research groups including [41, 3] have built their own 3D scanner – some of them tailored to specific requirements. Furthermore, there is a broad range of commercial products made by companies like Cyberware, Minolta, or Steinbichler.

There are several different approaches to acquire the 3D geometry of an object (for an overview see [9]) but most of the systems for small or medium sized objects are based on an active stereo structured light approach. One or several patterns are projected onto the object with a computer controlled projection system (e.g. a video projector, a color coded flash stripe projector, or a laser beam). The projected light patterns on the object are observed by a digital camera which is rigidly connected to the projection system. The 3D location of a point on the surface of an object is then defined by the intersection of a ray from the projected pattern with the viewing ray that corresponds to the pixel in the digital image that observed this ray (see Figure 6.1).

The position of these rays in space is determined in a separate calibration step: The patterns are projected onto a calibration target – typically a flat board or a three-dimensional structure with a regular pattern whose geometric properties are exactly known. The acquired images are analyzed to recover the intrinsic parameters (e.g. focal length, lens distortion) and extrinsic parameters (the relative position and orientation) of the projection system and the camera using standard camera calibration techniques (e.g. [65, 71, 20]).

Using the active stereo approach most objects cannot be acquired with a single scan either because front and back part of the object cannot be scanned with a single scan or because for a given configuration not all parts of the object are visible from both the position of the projection system and the digital camera. Therefore

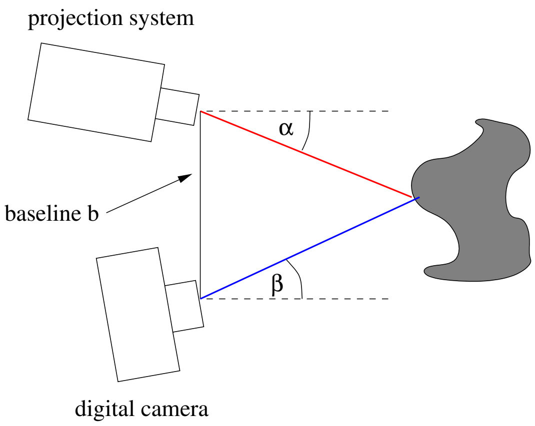


Figure 6.1: *Schematic drawing of an active stereo 3D scanner. Given the intrinsic parameters of the projection system and the camera, the baseline* b *and the angles* α *and* β*, the position of a surface point can be recovered using triangulation.*

several scans have to be registered against each other in order to combine them into a single set of surface points. This is commonly done using a variant of the iterative closest point method (ICP) [4, 54]. The resulting point cloud is triangulated leading to a single triangular mesh using one of a large variety of methods (for an overview see [9]). Further processing steps include smoothing to reduce noise (e.g. using [63, 30]) and editing of the resulting mesh for which a huge selection of tools is available including [31].

Kobbelt et al. [32] give a detailed description of the techniques used for the acquisition and processing of 3D geometry data.

7 Registration of Geometry and Texture Data

Since texture and geometry are typically acquired by two different processes the collected data has to be merged afterwards. This requires the alignment of the geometry data and the captured images. Only for scanning devices that capture geometry and texture data with the same sensor, the alignment or registration is already given. But in such a case the user is limited to the texture data provided by the scanner and the lighting setup cannot be changed to perform appearance measurements. Because of this, we further consider the case of two different sensors, a 3D scanner and a digital camera.

7.1 Manual Registration

In order to align or register the 3D model to the texture data one has to recover the parameters of the camera transformation that maps points in 3-space (the 3D geometry) onto the 2D image. These parameters describe the camera position, its orientation and the focal length (see Section 3.4.1). Further parameters are the aspect ratio, the principle point and the lens distortion, which are in the following assumed to be already known.

A simple approach to recover the camera position and orientation is to manually select corresponding points on the geometric model and in the picture [56]. If enough correspondences are established the transformation can be directly determined using one of various kinds of camera calibration methods (e.g [65, 71, 20]). But selecting corresponding points for a set of images is a time-consuming and tedious task. Additionally, the precision is limited by the user, although accuracy could be improved by selecting more points.

7.2 Automatic Registration

In order simplify the registration process some semi-automatic approaches have been published [47, 48]. The user is asked to roughly align the 3D model to the image. The algorithm then tries to optimize for the camera parameters by minimizing the distance between the outline of the 3D model rendered with the current set of camera parameters and the outline of the object found in the image. For each tested set of camera parameters the distance between the outlines has to be computed. This is a time-consuming step since the 3D model has to be rendered, its outline must be traced and for some points on it the minimum distance to the other outline must be computed.

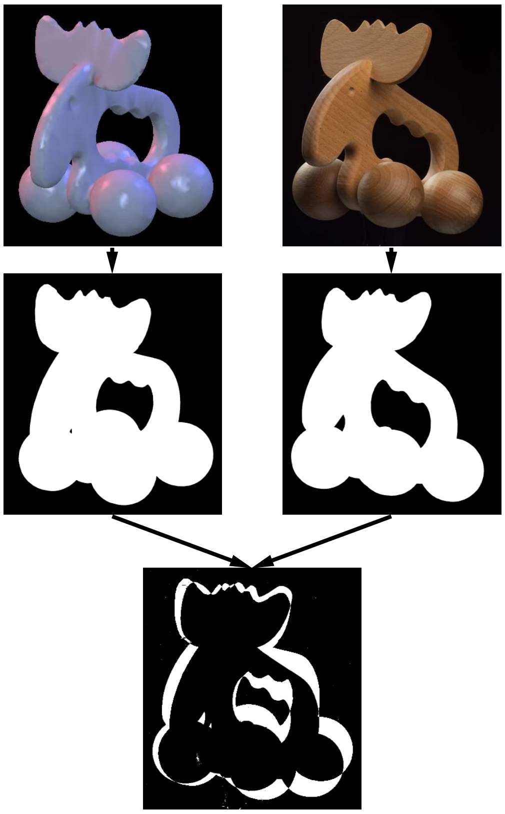


Figure 7.1: *Measuring the difference between photo (right) and one view of the model (left) by the area occupied by the XOR-ed foreground pixels.*

In [38, 39], Lensch et al. proposed a method to compute the distance between a view of the 3D model and the 2D image in a different way. Here, silhouettes are compared directly instead of using their outlines. At first the silhouette of the object in the images is extracted by classification of the image in foreground and background pixels, which can be done by any segmentation algorithm. Then, the geometry is rendered in front of a black background using a monochrome color. It is combined with the segmented image using the XOR-operation as is visualized in Figure 7.1. The resulting image will be black except for those pixels which are covered by just one silhouette but not by the other, that is to say exactly those pixels where the silhouettes differ. The number of remaining pixels is a measure for the distance between the silhouettes. These pixels can be counted by evaluating the histogram. The optimal set of camera parameters can be found by minimizing the number of remaining pixels.

Note that all three steps, rendering, combining, and histogram evaluation can be performed using graphics hardware and thus can be computed very fast, speeding up the optimization.

Additionally, it is also possible to automatically find a rough initial guess for the camera parameters. The effective focal length is first approximated by the focal length of the applied lens system. Depending on the focal length and the size of the object, the distance to the object can be approximated. It is assumed that the object is centered in the image. What remains to be estimated is the orientation of the camera. The optimization is simply started for a number of equally distributed sample orientation allowing just a few optimization steps per sample. The best result is then taken as a starting point for further optimization.

7.3 Texture Preparation

Knowing all camera parameters or the entire camera transformation for one image, it can be stitched onto the surface of the 3D model. The image is projected onto the the 3D model using projective texture mapping. Given a triangular mesh the stitching is done by computing texture coordinates for each vertex of the model that is visible in the image. Texture coordinates are calculated by projecting the 3D coordinates of the vertices into the image plane using the recovered camera transformation. All visible triangles can then be textured by the image as shown in Figure 7.2.

Further, the exact transformation for projecting surface points into the images is known. This information is required when collecting all radiance samples for one point on the objects surface into a lumitexel (compare Section 5.3.1).

A task that is still left is to determine the set of surface points for which a lumitexel should be generated. In order to obtain the highest quality with respect to the input images, the sampling density of the surface points must match that of the images. To achieve this, every triangle of the 3D model is projected into each image using the previously determined camera parameters. The area of the projected triangle is measured in pixels and the triangle is assigned to the image in

Figure 7.2: *The 3D model is aligned to a captured picture which then can be mapped as a texture onto the geometry.*

which its projected area is largest. For every pixel within the projected triangle a lumitexel is generated. The position of the surface point for the lumitexel is given by the intersection of the ray from the camera through the pixel with the mesh (see Figure 7.3).

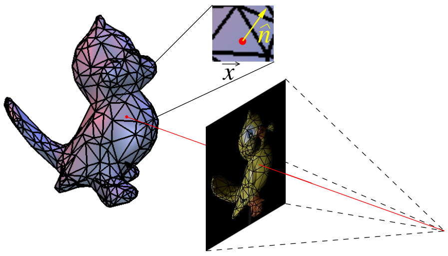


Figure 7.3: *The correspondence between pixel position and point position on the object is computed by tracing a ray through the image onto the object.*

Since every lumitexel is assigned to a triangular region within one of the HDR images it is possible to construct a 2D texture of lumitexels. This texture will unfortunately consist of a large number of separate triangles. Larger patches can be obtained by grouping adjacent triangles of the same input image. However, a significant number of isolated regions will remain. Instead of treating these regions as independent textures, it is more convenient to pack the regions into a single image, e.g. using the technique proposed by Rocchini et al. [56]. A result of this packing is shown in Figure 7.4 where the original color values of the input images are used to show the regions for which lumitexels are constructed.

During texture generation all parts of the original images where only the background is visible are discarded. Combined with dense packing of the remaining parts into one image, this reduces the size of the texture compared to the overall volume of the original images. A single image has the further advantage that it can be compressed and transformed into a streamable representation with less effort.

Figure 7.4: *Packing of the constructed texture regions for the elk model. Only three pictures were considered in this case to better visualize the layout .*

8 Interactive Display

After measuring the reflection properties of the object and transforming the images into a single texture, we explain in this section how the combined data can be displayed interactively.

8.1 Lighting Models

In Section 5 you have already seen what a reflectance model is. A reflectance model can be seen as a material description that modulates the intensity of the light that arrives at the surface. For every light incident direction it tells you how much light is being scattering to which exitant direction. For example the Blinn-Phong model that is used by OpenGL can be described as:

$$\begin{aligned} f_r(\hat{\omega}_o, \hat{\omega}_i) &= k_d + k_s \frac{(\hat{n} \cdot \hat{h})^N}{\hat{n} \cdot \hat{\omega}_i}, \\ \hat{h} &= \frac{\hat{\omega}_i + \hat{\omega}_o}{|\hat{\omega}_i + \hat{\omega}_o|}, \end{aligned}$$

where $\hat{\omega}_i$ is the incident light direction, $\hat{\omega}_o$ is the exitant light direction (i.e. viewing direction), $\hat{h}$ is the half-way vector between $\hat{\omega}_i$ and $\hat{\omega}_o$, all of which are in coordinates relative to the surface, i.e. relative to the local tangent frame consisting of the normal $\hat{n}$, the tangent $\hat{t}$, and the bi-normal $\hat{b}$. The parameters k_d, k_s, and N describe the diffuse coefficient, the specular coefficient, and the Phong exponent.

This not a complete lighting model, since the BRDF only tells you how light is scattered. A lighting model includes much more: how the light intensity decreases with the distance from the light source (e.g. quadratically), what kind of light sources are supported (e.g. point light or parallel light), if shadows are included, and so on. We will use the following simple lighting model for the rest of the chapter (which is similar to the OpenGL lighting model):

$$
\begin{aligned}
L(\hat{\omega}_o) &= f_r(\hat{\omega}_o, \hat{\omega}_i) L_i(\hat{\omega}_i)(\hat{n} \cdot \hat{\omega}_i), \qquad (8.1)\\
L_i(\hat{\omega}_i) &= \begin{cases} \frac{I}{r^2} & \text{for point lights} \\ I & \text{for parallel light} \end{cases}
\end{aligned}
$$

This lighting model uses only a single point or parallel light source that has the brightness I, and is r units away from the illuminated surface. $L(\hat{\omega}_o)$ describes the radiance leaving at the surface point in direction $\hat{\omega}_o$ towards the eye; this is then perceived by the eye.

The standard OpenGL lighting model does not allow to change the function f_r, it always uses the Blinn-Phong model introduced above. In the next section, we will explain how this can be changed. Furthermore standard OpenGL only evaluates $L(\hat{\omega}_o)$ at every vertex and uses Gouraud shading to interpolate values within the triangle.

8.2 Rendering with Arbitrary BRDFs

At first we will investigate the case of one homogeneous material, i.e. one BRDF per object. Standard OpenGL only supports the empirical and physically implausible Phong model, which makes surfaces always look "plastic"-like.

The main idea is to approximate a given reflectance model (e.g. for velvet or brushed metal) so that it can be used with OpenGL. A new algorithm called "separable decomposition" is employed [27].

Each of the two directions that a BRDF uses can be modeled as a 2D parameter, hence a reflectance model usually depends on 4 parameters. For an accurate representation this 4D function could just be sampled, but graphics hardware does not support 4D texture and a lot of memory would be needed for this representation.

Instead a separable decomposition is used, which approximates the 4D function with a product of two 2D functions.

$$
\begin{aligned}
f_r(\hat{\omega}_o, \hat{\omega}_i) &= g(\hat{\omega}_o) \cdot h(\hat{\omega}_i), \\
L(\hat{\omega}_o) &= g(\hat{\omega}_o) h(\hat{\omega}_i) L_i(\hat{\omega}_i)(\hat{n} \cdot \hat{\omega}_i),
\end{aligned}
$$

Using texture mapping the equation for $L(\hat{\omega}_o)$ can easily be evaluated on the graphics hardware. Each of these 2D functions $g(\hat{\omega}_o)$ and $h(\hat{\omega}_i)$ can be sampled and stored in a texture map. For every vertex of every polygon you have to compute $\hat{\omega}_o$ and $\hat{\omega}_i$ and use it as texture coordinates. Then the polygon has to be texture mapped with the textures containing $g(\hat{\omega}_o)$ and $h(\hat{\omega}_i)$ and the computed texture coordinates. Blending has to be set to modulate, so that $g(\hat{\omega}_o)$ and $h(\hat{\omega}_i)$ are multiplied together. The term $L_i(\hat{\omega}_i)(\hat{n} \cdot \hat{\omega}_i)$ can be multiplied to the result of $g(\hat{\omega}_o) \cdot h(\hat{\omega}_i)$ by enabling OpenGL lighting with only a diffuse component.

Rendering of arbitrary materials using this approximation is very fast because it boils down to computing texture coordinates and blending two texture maps together.

The main trick is to reparameterize the original 4D reflectance model in a smart way, such that the approximation works well. We refer the reader to [27], [28], or [69] for more detailed descriptions. See Figure 5.1 for an example rendered with this technique at real-time rates.

8.3 Rendering with Normal Maps

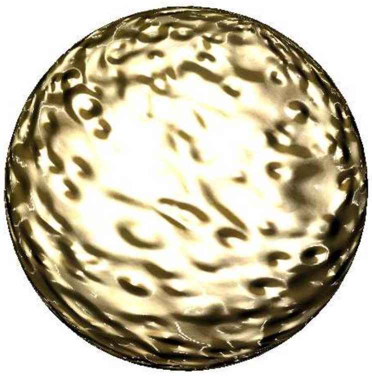


Figure 8.1: *A normal map applied to a sphere*

Blinn [5] has shown how wrinkled surfaces can be simulated by only perturbing the normal vector, without changing the underlying surface itself. The perturbed normal is then used for the lighting calculations instead of the original surface normal. This technique is generally called bump mapping.

If we have another look at the lighting model equation (see Equation 8.1), we can see a dependence on the normal $\hat{n}$. As mentioned in Section 8.1, the lighting is usually only evaluated at every vertex and not within a triangle, so the normals from the vertices are used to evaluate the Equation 8.1.

In order to simulate wrinkles, bump mapping requires a per-pixel normal, which is used for the evaluation of this equation. Graphics cards now support complex per-pixel operation which allow to perform this bump mapping technique at interactive rates [29].

Bump mapping is fairly simple to implement with these new features. These features include per-pixel dot-products, multiplication, addition, subtraction, so lighting models using only these operations can be used to do bump mapping. For every pixel we simply have to evaluate the lighting model.

Usually the Blinn-Phong model that was introduced in Section 8.1 is used to do bump mapping, because this model only mainly uses dot-products. Let us have a look at the lighting model using the Blinn-Phong reflectance model:

$$L(\hat{\omega}_o) \quad = \quad k_d I(\hat{n} \cdot \hat{\omega}_i) + k_s I(\hat{n} \cdot \hat{h})^N$$

If this is used in conjunction with bump mapping, the first term of the sum is usually entitled diffuse bump mapping and the second term is entitled specular bump mapping. Using the new per-pixel operations, this formula can be easily computed at every pixel. First, the normals are encoded in a texture map. Then $\hat{\omega}_i$ and $\hat{h}$ are computed on a per-vertex basis (will be interpolated across the triangle). Now, the graphics card has to be configured, such that it computes the equation above. An example rendering can be seen in Figure 8.1. For more details, please see [29].

8.4 Spatially Varying BRDFs

As just mentioned, bump mapping usually uses simple lighting model such as the Blinn-Phong model [6] for the lighting calculations. While this is an appropriate and fast method to do bump mapping, it is not very flexible. The Blinn-Phong model does not have many parameters that can be tweaked to change the appearance of the bumpy surface and the chosen parameters (i.e. at least the exponent) have to remain constant over a polygon.

We will introduce a new bump mapping technique [26] which uses a modified version of the Blinn-Phong model, which offers more flexibility concerning the parameters. And what's more, those parameters can even change on a per-pixel level. See Figure 8.2 for an example of what can be done.

The lighting model using the modified Blinn-Phong model can be written as follows:

$$\begin{aligned} L(\hat{\omega}_o) = k_d(\hat{n} \cdot \hat{\omega}_i) + \\ k_s \left(1 - \left(\hat{h} \cdot \frac{\hat{t}}{\alpha_x} \right)^2 - \left(\hat{h} \cdot \frac{\hat{b}}{\alpha_y} \right)^2 \right) . \end{aligned}$$

This lighting models uses new parameters. The specular part does not depend on the normal $\hat{n}$ anymore, but on the tangent $\hat{t}$ and the bi-normal $\hat{b}$. These two vectors are divided by α_x resp. α_y, which have to be in the range $[0, 1]$. The smaller

Figure 8.2: *A spatially varying BRDF applied to a sphere.*

these values are, the smaller the highlight will be. The model is anisotropic, which means, that the shape and the orientation of the highlight depends on the orientation of the surface. If different α_x and α_y are chosen, the model is anistropic, otherwise it is isotropic. In the anisotropic case, the tangent and bi-normal define the main orientation of the highlight.

The implementation of this new lighting model works very much like standard Blinn-Phong bump mapping, only that the $\hat{t}/\alpha_x$ has to be stored in one texture map, and $\hat{b}/\alpha_y$ has to be stored in a second texture map. Of course, the graphics card has to be set up, that it performs the necessary dot-products etc., but this can be done using OpenGL extensions or DirectX 8.

Other BRDFs can be used as well, if they can be implemented with the supported per-pixel operations. If not than an upcoming feature called dependent texture lookup can be used to implement arbitrary functions on the graphics hardware. Dependent texture lookup uses the values from a texture map to lookup into another texture map, which nothing else than evaluating an arbitrary function. We refer the reader to [26] for more details.

8.5 Bump Mapping with Shadows

Bump maps usually do not cast shadows onto themselves, which of course is very unrealistic. There are two novel techniques [19, 60] that add shadows to bump maps. See Figure 8.3 for an example of a bump map casting a shadow.

Both techniques precompute information on when a pixel is in shadow and

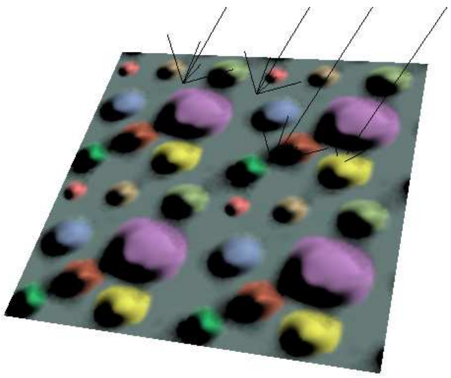


Figure 8.3: *Simple bump map, where all the bumps are casting shadows according to the light direction.*

store this per-pixel information in texture maps. To decide whether a pixel is in shadow, you only have to know whether the light source position (point or parallel light only) is above or below the horizon visible from that point.

The technique proposed by Sloan and Cohen [60] samples the height of the horizon at a number of position and stores these heights in texture maps. When rendering the bump map they transform the light position into a height value, and use per-pixel operations to perform the comparison between the stored per-pixel height values and the light source height value.

The other technique [19] fits an ellipse to the horizon, stores the parameters of the ellipse in texture maps and performs per-pixel operations to check whether a pixel is inside or outside the ellipse, i.e. whether it is lit or in shadow.

9 Examples

In this section we describe some examples for high quality 3D object acquisition. Geometry and reflection data have been acquired for a bronze bust of Max Planck, a clay bird, and a painted models of two angles. Some statistics about the meshes and the number of acquired views are listed in Table 9.1.

The model of the angels was generated by extracting an isosurface of a computer tomography scan. The 3D geometry model of the bust and the bird were acquired using a Steinbichler Tricolite structured light 3D scanner. More than 20 scans per object were necessary to cover most of the surface. After a manual approximate alignment the scans were pairwise registered against each other. Finally, an optimization procedure reduced the global error. The resulting point clouds were triangulated to form triangle meshes.

Because a structured light scanner can only acquire surface points that are visible from the camera and projector position at the same time the bust mesh contained several holes – mainly around the ears. They were filled manually. Afterwards, a filtering step was applied to improve the smoothness of the meshes. In order to accelerate further processing the triangle count of the initial models was reduced by simplifying the meshes.

The images for the textures and reflection properties were taken with a Kodak DCS 560 professional digital camera, which outputs images consisting of 6 mil-

model	triangles	views	lumitexels	rad. samples	clusters	basis BRDFs
angels	47000	27	1606223	7.6	9	6
bird	14000	25	1917043	6.3	5	4
bust	50000	16	3627404	4.2	3	4

Table 9.1: *This table lists the number of triangles of each model, the number of views we used to reconstruct the spatially varying BRDFs, the number of acquired lumitexels and the average number of radiance samples per lumitexel, the number of partitioned material clusters, and the number of basis BRDFs per cluster.*

Figure 9.1: *A bronze bust rendered with a spatially varying BRDF, which was acquired with the presented reconstruction method.*

lion pixels. To acquire data for the entire surface several views with varying light source positions were captured per model (see Table 9.1). For each view around 15 photographs were necessary: two for recovering the light source position, one to extract the silhouette of the object for the 2D–3D registration, and the rest to provide the necessary high dynamic range.

The acquisition takes about 2.5h. The high dynamic range conversion, registration with the 3D model, and the resampling into lumitexels takes about 5h but is a completely automated task. The clustering and the final projection to recover the BRDFs takes about 1.5h[1].

Figure 5.3 shows how five successive split operations partition the lumitexels (the surface points) for the bird into its five basic materials. Only the per-cluster BRDFs determined by the clustering process are used for shading. Because of this the object looks rather flat. After performing the projection step every lumitexel is represented as a linear combination in a basis of four BRDFs, now resulting in a much more detailed and realistic appearance, see Figure 5.4.

The bust in Figure 9.1 shows another reconstructed object with very different reflection properties. The bronze look is very well captured.

A comparison between an object rendered with an acquired BRDF (using the presented method) and a photograph of the object is shown in Figure 9.2. They are very similar, but differences can be seen in highlights and in places where not enough radiance samples were captured. Capturing more samples will increase

[1]All timings were measured on a single processor SGI Octane 300 MHz.

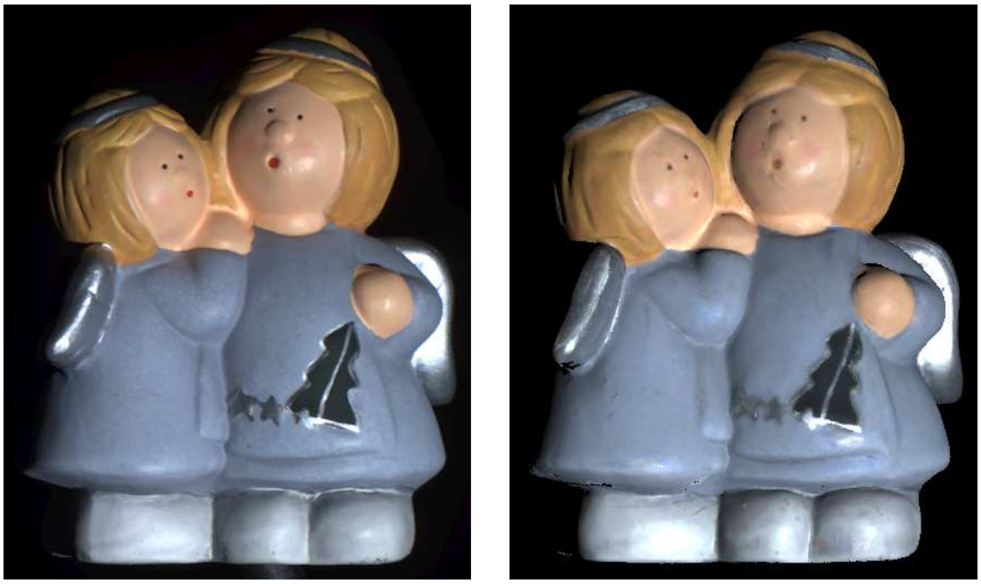


Figure 9.2: *Left side: Photograph of model. Right side: Model with acquired BRDF rendered from the same view with similar lighting direction. The difference in the hair region is due to missing detail in the triangle mesh.*

the quality. The difference in the hair region is due to missing detail in the triangle mesh. Those would be resolved by recovering the normal map for the object as described in Section 5.4.

Generally it can be said that for all the models only a few clusters were needed to accurately represent all the materials since the projection takes care of material changes. In our experiments even Lafortune BRDFs [33] consisting of a single lobe were sufficient to form good basis for the clustering and projection.

Further examples and movies of the acquired objects can be found at *http://www.mpi-sb.mpg.de/∼lensch/proj/BRDFMeasurement/BRDFMeasurement.html*.

10 Conclusion

We presented a framework for acquiring high quality 3D models of real world objects. The resulting models include both geometry and physical properties such as appearance properties including textures, normal maps or spatially varying BRDFs and interactive deformation behavior. Each of these is captured with a different setup. Afterwards all data of desired modalities are merged into a single model. In the discussed example this is a fairly complete representation of the geometry and surface properties of a large class of real world objects. In order to achieve the highest possible quality, state-of-the-art computer vision and computer graphics techniques need to be combined in the acquisition and model generation stage of the framework.

Given such a detailed model, many computer vision algorithms such as the reconstruction of surface normals [58] or the detection of different materials can be improved or extended to other types of objects. Common assumptions about the characteristics of the object (e.g., pure diffuse reflection) are no longer necessary.

The demand for high quality 3D models will further increase in applications such as computer games, interactive applications in medicine and automation, digital libraries and encyclopedias, or e-commerce applications. In order to satisfy these demands the presented methods need to be further improved with respect to acquisition speed, automation and quality. Currently, the class of materials for which appearance properties can be acquired and displayed are limited to isotropic materials. Future algorithms should also take effects like anisotropy and subsurface scattering into account. The deformation behavior captured is the static linear object response, however, often a full dynamic model supporting large deformation is desirable, e.g., in medicine. There is a lot of opportunity for future work.

11 Acknowledgments

Parts of this work have been funded by the DFG SPP V3D2 "Verteile Verarbeitung und Vermittling Digitaler Dokumente" and by the European Community within the scope of the ViHAP3D Project IST-2001-32641 "Virtual Heritage: High-Quality 3D Acquisition and Presentation."

We would like to thank Thomas Neumann, Kolja Kähler, Christian Rössl, Mario Botsch and the IMP Erlangen for their help in acquiring some of the presented data sets. Thanks also to Philippe Bekaert for proofreading parts of this paper.

The ACME facility is the result of a group effort at the University of British Columbia under the project leadership of Dinesh K. Pai. We would like to thank D.K. Pai, R.J. Woodham, D.L. James, J.E. Lloyd, J.L. Richmond and S.H. Yau for their part in the project and would like to acknowledge their respective contribution. The financial support of ACME (and the work on deformation modeling) through the IRIS Network of Centres of Excellence NSERC, and the University of British Columbia is gratefully acknowledged.

Bibliography

[1] T. M. C. Abbott. In situ CCD testing. Available at http://www.cfht.hawaii.edu/~tmca/cookbook/top.html, 1995.

[2] D. Banks. Illumination in Diverse Codimensions. In *Proceedings of SIGGRAPH 1994*, pages 327–334, July 1994.

[3] F. Bernardini, J. Mittleman, and H. Rushmeier. Case study: Scanning michelangelo's florentine pietà. In *Course Notes for SIGGRAPH 1999*, August 1999.

[4] P.J. Besl and N.D. McKay. A method for the registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–258, 1992.

[5] J. Blinn. Simulation of Wrinkled Surfaces. In *Proceedings of SIGGRAPH 1978*, pages 286–292, August 1978.

[6] J. Blinn. Models of Light Reflection For Computer Synthesized Pictures. In *Proceedings SIGGRAPH*, pages 192–198, July 1977.

[7] M. Bro-Nielsen and S. Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. In J. Rossignac and F.X. Sillion, editors, *Eurographics*, Computer Graphics Forum, 15(3), pages 57–66, Poitiers, France, 1996.

[8] Colorimetry. Publication CIE No. 15.2, 1986.

[9] Brian Curless and Steven Seitz. 3D Photography. In *Course Notes for SIGGRAPH 2000*, July 2000.

[10] D. d'Aulignac, M.C. Cavusoglu, and C. Laugier. Modeling the dynamics of the human thigh for a realistic echographic simulator with force feedback. In *Int. Conf. on Medical Image Computer-Assisted Intervention*, pages 1191–1198, Cambridge, Uk, Sept 1999.

[11] P. Debevec and J. Malik. Recovering High Dynamic Range Radiance Maps from Photographs. In *Proceedings of SIGGRAPH 97*, pages 369–378, August 1997.

[12] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. *Proceedings of SIGGRAPH 2000*, pages 145–156, July 2000.

[13] H. Delingette. Toward realistic soft-tissue modeling in medical simulation. *Proceedings of the IEEE*, 86(3):512–523, 1998.

[14] S.F.F. Gibson and B. Mirtich. A survey of deformable modeling in computer graphics. Technical Report TR-97-19, Mitsubishi Electric Research Laboratory, Cambridge, MA, USA, Nov. 1997.

[15] Michael Goesele, Wolfgang Heidrich, Hendrik P.A. Lensch, and Hans-Peter Seidel. Building a Photo Studio for Measurement Purposes. In *Proceedings of the 5th Conference on Vision, Modeling, and Visualization (VMV-00)*, November 2000.

[16] Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Color calibrated high dynamic range imaging with ICC profiles. In *Proceedings of the 9th Color Imaging Conference*, Scottsdale, USA, 2001.

[17] Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Entropy-based dark frame subtraction. In *Proceedings of PICS 2001: Image Processing, Image Quality, Image Capture, Systems Conference*, Montreal, Canada, April 2001. The Society for Imaging Science and Technology (IS&T).

[18] P. Haeberli and M. Segal. Texture Mapping As A Fundamental Drawing Primitive. In *Fourth Eurographics Workshop on Rendering*, pages 259–266, June 1993.

[19] W. Heidrich, K. Daubert, J. Kautz, and H.-P. Seidel. Illuminating micro geometry based on precomputed visibility. In *Proc. of SIGGRAPH 2000*, pages 455–464, July 2000.

[20] J. Heikkila and O. Silven. A Four-Step Camera Calibration Procedure With Implicit Image Correction. In *CVPR97*, 1997.

[21] Robert W. G. Hunt. *The reproduction of colour.* Fountain Press, 5. ed. edition, 1995.

[22] Richard S. Hunter and Richard W. Harold. *The measurement of appearance*. Wiley, 2. ed., 5. print. edition, 1987.

[23] Specification ICC.1:1998-09, File Format for Color Profiles. available from http://www.color.org, 1998.

[24] D. James and D.K. Pai. A unified treatment of elastostatic contact simulation for real time haptics. *Haptics-e*, 2(1), 2001.

[25] D.L. James and D.K. Pai. ArtDefo accurate real time deformable objects. In *Computer Graphics, Annual Conference Series*, pages 65–72, Los Angles, USA, Aug 1999. ACM SIGGRAPH.

[26] J.Kautz and H.-P. Seidel. Towards interactive bump mapping with anisotropic shift-variant brdfs. In *Proceedings of the Eurographics/SIGGRAPH Workshop on Graphics Hardware 2000*, pages 51–58, August 2000.

[27] J. Kautz and M. McCool. Interactive Rendering with Arbitrary BRDFs using Separable Approximations. In *10th Eurographics Rendering Workshop 1999*, pages 281–292, June 1999.

[28] J. Kautz, C. Wynn, J. Blow, C. Blasband, A. Ahmad, , and M. McCool. Achieving Real-Time Realistic Reflectance. *Game Developer Magazine*, January 2001.

[29] M. Kilgard. *A Practical and Robust Bump-mapping Technique for Today's GPUs*. NVIDIA Corporation, April 2000. Available from http://www.nvidia.com.

[30] Leif Kobbelt. Discrete fairing. In *Proceedings of the Seventh IMA Conference on the Mathematics of Surfaces*, pages 101–131, 1996.

[31] Leif Kobbelt, Swen Campagna, Jens Vorsatz, and Hans-Peter Seidel. Interactive multi-resolution modeling on arbitrary meshes. *Proceedings of SIGGRAPH 98*, pages 105–114, July 1998.

[32] Leif P. Kobbelt, Stephan Bischoff, Mario Botsch, Kolja Kähler, Christian Rössl, Robert Schneider, and Jens Vorsatz. Geometric modeling based on polygonal meshes. Technical Report MPI-I-2000-4-002, Max-Planck-Institut für Informatik, July 2000.

[33] E. Lafortune, S.-C. Foo, K. Torrance, and D. Greenberg. Non-Linear Approximation of Reflectance Functions. In *Proceedings of SIGGRAPH 1997*, pages 117–126, August 1997.

[34] J. Lang. *Deformable Model Acquisition and Validation*. PhD thesis, University of British Columbia, October 2001.

[35] J. Lang and D.K. Pai. Estimation of elastic constants from 3D range-flow. In *3rd International Conference on 3-D Digital Imaging and Modeling*, pages 331–338, Québec City, Canada, 2001. IEEE.

[36] J. Lang, D.K. Pai, and R.J. Woodham. Robotic acquistion of deformable models. In *International Conference on Robotics and Automation*, pages 933–938, Washington, D.C., USA, May 2002.

[37] Hendrik Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatially varying materials. In Steven Gortler and Karol Myszkowski, editors, *Proceedings of the 12th Eurographics Workshop on Rendering*, pages 104–115, London, Great Britain, 2001. Springer.

[38] Hendrik P. A. Lensch, Wolfgang Heidrich, and Hans-Peter Seidel. Automated texture registration and stitching for real world models. In *Pacific Graphics '00*, pages 317–326, October 2000.

[39] Hendrik P. A. Lensch, Wolfgang Heidrich, and Hans-Peter Seidel. A silhouette-based algorithm for texture registration and stitching. *Graphical Models*, 63:245–262, 2001.

[40] Marc Levoy and Pat Hanrahan. Light field rendering. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 31–42, August 1996.

[41] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy, Ginsberg, Jonathan Shade, and Duane Fulk. The digital michelangelo project: 3D scanning of large statues. In *Proceedings of SIGGRAPH 2000*, pages 131–144, July 2000.

[42] J. Lloyd and V. Hayward. Trajectory generation in multi-RCCL. *Journal of Robotic Systems*, 10(3):369–390, 1993.

[43] R. Luther. Aus dem Gebiet der Farbreizmetrik. *Zeitschrift fr technische Physik*, 8(12):540–558, 1927.

[44] Brian C. Madden. Extended Intensity Range Imaging. Technical report, University of Pennsylvania, GRASP Laboratory, 1993.

[45] S. R. Marschner. *Inverse rendering for computer graphics*. PhD thesis, Cornell University, 1998.

[46] S. R. Marschner, B. Guenter, and S. Raghupathy. Modeling and rendering for realistic facial animation. In *Eurographics Rendering Workshop 2000*, pages 231–242, June 2000.

[47] Kenji Matsushita and Toyohisa Kaneko. Efficient and handy texture mapping on 3d surfaces. *Computer Graphics Forum*, 18(3):349–358, September 1999.

[48] Peter J. Neugebauer and Konrad Klein. Texturing 3d models of real world objects from multiple unregistered photographic views. *Computer Graphics Forum*, 18(3):245–256, September 1999.

[49] D.K. Pai, editor. *Reality-based modeling and applications in reverse engineering, computer graphics and VR*, Workshop at the International Conference on Robotics and Automation. IEEE, 2000.

[50] D.K. Pai, J. Lang, J.E. Lloyd, and J.L. Richmond. Reality-based modeling with ACME: A progress report. In D. Rus and S. Singh, editors, *7th International Symposium on Experimental Robotics, LNCIS 271*, pages 121–130, Honolulu, USA, December 2000. Springer-Verlag.

[51] D.K. Pai, J. Lang, J.E. Lloyd, and R.J. Woodham. ACME, a telerobotic active measurement facility. In P. Corke and J. Trevelyan, editors, *6th International Symposium on Experimental Robotics, LNCIS 250*, pages 391–400, Sydney, Australia, March 1999. Springer-Verlag.

[52] D.K. Pai, K. van den Doel, D.L. James, J. Lang, J.E. Lloyd, J.L. Richmond, and S.H. Yau. Scanning physical interaction behavior of 3D objects. In *Computer Graphics, Annual Conference Series*, pages 87–96, Los Angles, Aug 2001. ACM SIGGRAPH.

[53] G. Picinbono, H. Delingette, and N. Ayache. Non-linear and anisotropic elastic soft tissue models for medical simulation. In *International Conference on Robotics and Automation*, pages 1371–1376, Seoul, South Korea, May 2001.

[54] K. Pulli, T. Duchamp, H. Hoppe, J. McDonald, L. Shapiro, and W. Stuetzle. Robust meshes from multiple range maps. In *Proceedings of IEEE International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, 1997.

[55] Mark A. Robertson, Sean Borman, and Robert L. Stevenson. Dynamic Range Improvement Through Multiple Exposures. In *Proceedings of the 1999 International Conference on Image Processing (ICIP-99)*, pages 159–163. IEEE, oct. 1999.

[56] C. Rocchini, P. Cignoni, and C. Montani. Multiple textures stitching and blending on 3D objects. In *Eurographics Rendering Workshop 1999*. Eurographics, June 1999.

[57] Holly Rushmeier, Fausto Bernardini, Joshua Mittleman, and Gabriel Taubin. Acquiring input for rendering at appropriate levels of detail: Digitizing a pietà. *Eurographics Rendering Workshop 1998*, pages 81–92, June 1998.

[58] Holly Rushmeier, Gabriel Taubin, and André Guéziec. Applying shape from lighting variation to bump map capture. *Eurographics Rendering Workshop 1997*, pages 35–44, June 1997.

[59] Hartmut Schirmacher, Wolfgang Heidrich, Martin Rubick, Detlef Schiron, and Hans-Peter Seidel. Image-based BRDF reconstruction. In Bernd Girod, Heinrich Niemann, and Hans-Peter Seidel, editors, *Proceedings of the 4th Conference on Vision, Modeling, and Visualization (VMV-99)*, pages 285–292, nov 1999.

[60] P.-P. Sloan and M. Cohen. Hardware accelerated horizon mapping. *11th Eurographics Workshop on Rendering*, pages 281–286, June 2000.

[61] Wolfgang Stuerzlinger. Imaging all visible surfaces. In *Graphics Interface '99*, pages 115–122, June 1999.

[62] G. Székely, C. Brechbühler, J. Dual, R. Enzler, J. Hug, R. Hutter, N. Ironmonger, M. Kauer, V. Meier, P. Niederer, A. Rhomberg, P. Schmid, G. Schweitzer, M. Thaler, V. Vuskovic, G. Tröster, U. Haller, and M. Bajka. Virtual reality-based simulation of endoscopic surgery. *Presence*, 9(3):310–333, 2000.

[63] G. Taubin. A signal processing approach to fair surface design. *Proceedings of SIGGRAPH 1995*, pages 351–358, 1995.

[64] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of the Optical Society of America*, 57(9):1105–1114, September 1967.

[65] R. Tsai. A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4), August 1987.

[66] Dawn Wallner. Building ICC profiles – the mechanics and engineering. available at http://www.color.org/iccprofiles.html.

[67] G. Ward. Measuring and modeling anisotropic reflection. In *Proceedings of SIGGRAPH 1992*, pages 265–272, July 1992.

[68] Don Williams and Peter D. Burns. Diagnostics for digital capture using MTF. In *Proceedings of PICS 2001: Image Processing, Image Quality, Image Capture, Systems Conference*, pages 227–232, Montreal, Canada, April 2001. The Society for Imaging Science and Technology (IS&T).

[69] C. Wynn. BRDF-Based Lighting. Technical report, NVIDIA Corporation, 2000.

[70] Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. *Proceedings of SIGGRAPH 99*, pages 215–224, August 1999.

[71] Zhengyou Zhang. A Flexible New Technique for Camera Calibration. Technical Report MSR-TR-98-71, Microsoft Research, 1999. Updated version of March 25, 1999.

[72] Y. Zhuang and J. Canny. Haptic interactions with global deformations. In *International Conference on Robotics and Automation*, pages 2428–2433, San Francisco, USA, April 2000.

I N F O R M A T I K

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-I-2002-4-002	F. Drago, W. Martens, K. Myszkowski, H. Seidel	Perceptual Evaluation of Tone Mapping Operators with Regard to Similarity and Preference
MPI-I-2002-4-001	M. Goesele, J. Kautz, J. Lang, H.P.A. Lensch, H. Seidel	Tutorial Notes ACM SM 02 A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2002-2-008	W. Charatonik, J. Talbot	Atomic Set Constraints with Projection
MPI-I-2002-2-007	W. Charatonik, H. Ganzinger	Symposium on the Effectiveness of Logic in Computer Science in Honour of Moshe Vardi
MPI-I-2002-1-008	P. Sanders, J.L. Träff	The Factor Algorithm for All-to-all Communication on Clusters of SMP Nodes
MPI-I-2002-1-003	I. Katriel, P. Sanders, J.L. Träff	A Practical Minimum Scanning Tree Algorithm Using the Cycle Property
MPI-I-2002-1-002	F. Grandoni	Incrementally maintaining the number of l-cliques
MPI-I-2002-1-001	T. Polzin, S. Vahdati	Using (sub)graphs of small width for solving the Steiner problem
MPI-I-2001-4-005	H.P.A. Lensch, M. Goesele, H. Seidel	A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2001-4-004	S.W. Choi, H. Seidel	Linear One-sided Stability of MAT for Weakly Injective Domain
MPI-I-2001-4-003	K. Daubert, W. Heidrich, J. Kautz, J. Dischler, H. Seidel	Efficient Light Transport Using Precomputed Visibility
MPI-I-2001-4-002	H.P.A. Lensch, J. Kautz, M. Goesele, H. Seidel	A Framework for the Acquisition, Processing, Transmission, and Interactive Display of High Quality 3D Models on the Web
MPI-I-2001-4-001	H.P.A. Lensch, J. Kautz, M. Goesele, W. Heidrich, H. Seidel	Image-Based Reconstruction of Spatially Varying Materials
MPI-I-2001-2-006	H. Nivelle, S. Schulz	Proceeding of the Second International Workshop of the Implementation of Logics
MPI-I-2001-2-005	V. Sofronie-Stokkermans	Resolution-based decision procedures for the universal theory of some classes of distributive lattices with operators
MPI-I-2001-2-004	H. de Nivelle	Translation of Resolution Proofs into Higher Order Natural Deduction using Type Theory
MPI-I-2001-2-003	S. Vorobyov	Experiments with Iterative Improvement Algorithms on Completely Unimodel Hypercubes
MPI-I-2001-2-002	P. Maier	A Set-Theoretic Framework for Assume-Guarantee Reasoning
MPI-I-2001-2-001	U. Waldmann	Superposition and Chaining for Totally Ordered Divisible Abelian Groups
MPI-I-2001-1-007	T. Polzin, S. Vahdati	Extending Reduction Techniques for the Steiner Tree Problem: A Combination of Alternative-and Bound-Based Approaches
MPI-I-2001-1-006	T. Polzin, S. Vahdati	Partitioning Techniques for the Steiner Problem

MPI-I-2001-1-005	T. Polzin, S. Vahdati	On Steiner Trees and Minimum Spanning Trees in Hypergraphs
MPI-I-2001-1-004	S. Hert, M. Hoffmann, L. Kettner, S. Pion, M. Seel	An Adaptable and Extensible Geometry Kernel
MPI-I-2001-1-003	M. Seel	Implementation of Planar Nef Polyhedra
MPI-I-2001-1-002	U. Meyer	Directed Single-Source Shortest-Paths in Linear Average-Case Time
MPI-I-2001-1-001	P. Krysta	Approximating Minimum Size 1,2-Connected Networks
MPI-I-2000-4-003	S.W. Choi, H. Seidel	Hyperbolic Hausdorff Distance for Medial Axis Transform
MPI-I-2000-4-002	L.P. Kobbelt, S. Bischoff, K. Kähler, R. Schneider, M. Botsch, C. Rössl, J. Vorsatz	Geometric Modeling Based on Polygonal Meshes
MPI-I-2000-4-001	J. Kautz, W. Heidrich, K. Daubert	Bump Map Shadows for OpenGL Rendering
MPI-I-2000-2-001	F. Eisenbrand	Short Vectors of Planar Lattices Via Continued Fractions
MPI-I-2000-1-005	M. Seel, K. Mehlhorn	Infimaximal Frames: A Technique for Making Lines Look Like Segments
MPI-I-2000-1-004	K. Mehlhorn, S. Schirra	Generalized and improved constructive separation bound for real algebraic expressions
MPI-I-2000-1-003	P. Fatourou	Low-Contention Depth-First Scheduling of Parallel Computations with Synchronization Variables
MPI-I-2000-1-002	R. Beier, J. Sibeyn	A Powerful Heuristic for Telephone Gossiping
MPI-I-2000-1-001	E. Althaus, O. Kohlbacher, H. Lenhof, P. Müller	A branch and cut algorithm for the optimal solution of the side-chain placement problem
MPI-I-1999-4-001	J. Haber, H. Seidel	A Framework for Evaluating the Quality of Lossy Image Compression
MPI-I-1999-3-005	T.A. Henzinger, J. Raskin, P. Schobbens	Axioms for Real-Time Logics
MPI-I-1999-3-004	J. Raskin, P. Schobbens	Proving a conjecture of Andreka on temporal logic
MPI-I-1999-3-003	T.A. Henzinger, J. Raskin, P. Schobbens	Fully Decidable Logics, Automata and Classical Theories for Defining Regular Real-Time Languages
MPI-I-1999-3-002	J. Raskin, P. Schobbens	The Logic of Event Clocks
MPI-I-1999-3-001	S. Vorobyov	New Lower Bounds for the Expressiveness and the Higher-Order Matching Problem in the Simply Typed Lambda Calculus
MPI-I-1999-2-008	A. Bockmayr, F. Eisenbrand	Cutting Planes and the Elementary Closure in Fixed Dimension
MPI-I-1999-2-007	G. Delzanno, J. Raskin	Symbolic Representation of Upward-closed Sets
MPI-I-1999-2-006	A. Nonnengart	A Deductive Model Checking Approach for Hybrid Systems
MPI-I-1999-2-005	J. Wu	Symmetries in Logic Programs
MPI-I-1999-2-004	V. Cortier, H. Ganzinger, F. Jacquemard, M. Veanes	Decidable fragments of simultaneous rigid reachability
MPI-I-1999-2-003	U. Waldmann	Cancellative Superposition Decides the Theory of Divisible Torsion-Free Abelian Groups
MPI-I-1999-2-001	W. Charatonik	Automata on DAG Representations of Finite Trees
MPI-I-1999-1-007	C. Burnikel, K. Mehlhorn, M. Seel	A simple way to recognize a correct Voronoi diagram of line segments
MPI-I-1999-1-006	M. Nissen	Integration of Graph Iterators into LEDA
MPI-I-1999-1-005	J.F. Sibeyn	Ultimate Parallel List Ranking ?
MPI-I-1999-1-004	M. Nissen, K. Weihe	How generic language extensions enable “open-world” desing in Java
MPI-I-1999-1-003	P. Sanders, S. Egner, J. Korst	Fast Concurrent Access to Parallel Disks
MPI-I-1999-1-002	N.P. Boghossian, O. Kohlbacher, H.-. Lenhof	BALL: Biochemical Algorithms Library
MPI-I-1999-1-001	A. Crauser, P. Ferragina	A Theoretical and Experimental Study on the Construction of Suffix Arrays in External Memory