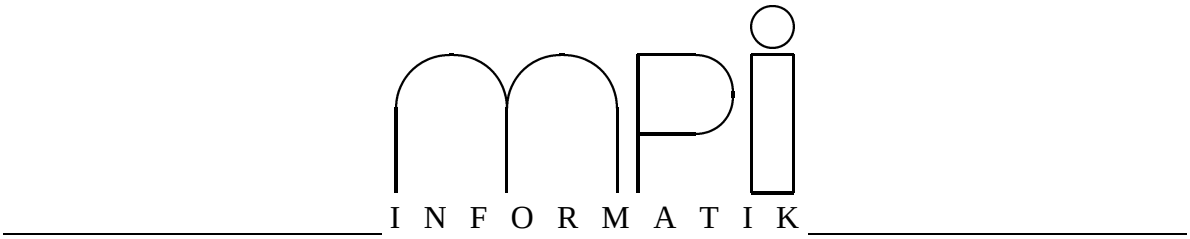

A Framework for the Acquisition, Processing, Transmission, and Interactive Display of High Quality 3D Models on the Web

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

MPI–I–2001–4-002 May 2001

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

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

Keywords

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

1 Introduction

"A picture is worth a thousand words" – the core of this proverb is that it is usually much easier to convey information with an image than by the use of language. During the last years this fact led for example to a change in how information is represented on the Internet from a purely text-based approach (e.g. represented by the web browser Lynx) to web pages filled with images and other graphical elements.

The next trend was (and still is) to include animated GIFs or small movies in web pages to give the user more information which cannot be expressed in a single image. Similar developments can be seen for other types of digital documents – modern word processors are able to include all kinds of multimedia contents into a text document. However the ultimate goal for many applications should be to give the user the ability to virtually inspect an object in the same way he can inspect it in reality: Any visitor of a museum, any shopper in a supermarket sees an interactive, real-time, high resolution, high quality, three-dimensional representation of an object – the object itself.

For some applications it is not necessary to use a high quality representation of an object: online book stores usually include only an image of the book cover. But for all e-commerce applications where the "look" of an object is important (e.g. clothes, mobile phones, jewellery) it is essential. This is also true for online encyclopaedias or digital libraries where multimedia descriptions of real world objects are given. Likewise for a piece of art in a virtual museum it is not sufficient to represent it by a single photo since its effect on the viewer may change from one viewpoint to the next. The user should be able to move freely around the object visualized using a high quality representation. The object may even be placed into different virtual environments with changing illumination while the impression of the real object should be preserved.

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 on the other hand the appearance of the material or materials it is made of, e.g. the object's color, texture, or reflection properties. Subse-

quently, geometry and surface appearance data must be integrated into a single digital model (requiring registration) which must then be stored, transmitted and displayed, trying to meet several conflicting requirements (such as realism vs. interactive speed).

This tutorial highlights some recent results on the acquisition and interactive display of high quality 3D models and shows how these results can be seamlessly integrated with previous work into a single framework for the acquisition, processing, transmission, and interactive display of high quality 3D models on the Web. Two examples illustrate the approach. Finally, we point out some remaining questions and important areas for future research.

2 Related Work

Recently, there are a lot of developments and approaches which try to improve the quality of 3D models on the Web, to unify the transmission process and to make 3D models available directly within documents without requiring a special viewer. This shows the importance of 3D models on the Web.

A standard for 3D hierarchical mesh coding is for example included in the MPEG-4 specification. Textured 3D models can be transmitted and displayed by the IBM HotMedia architecture [31]. They are integrated into a web page without the need to download any plug-in. In order to handle the growing complexity of 3D scenes Metastream [1] describes a framework for the representation of multi-resolution 3-D models including associated geometry and texture data. Progressively structured, compressed geometry and texture information is used to allow streamable computer graphics over the Internet. A first, coarse representation of the scene can be obtained already before the entire data set is transmitted.

Universal media follows another approach to strongly reduce the amount of transmitted data that is needed to represent multimedia objects including 3D models. The idea is to define a small, cross-platform library of locally resident media elements (sounds, textures, and 3D objects) that can be addressed uniformly and assembled within an online document. For a wooden sculpture for example only the shape and the name of the predefined wood texture that should be applied have to be transmitted. But besides the shape the real texture may also be important to communicate the intentions of the artist to the user. If the real texture is replaced by a standardized one relevant details of the object may be lost.

Instead of building an explicit 3D geometric model for real world objects, they can also be represented using image-based techniques. Light fields [34] store the outgoing radiance of an object in a 4D data structure. The object can be viewed from arbitrary positions by reconstructing a new image from the acquired radiance samples. A very simple but impressing image-based product for the Web has been developed by RealSpace. The object is placed on a turntable and a number of images are taken at different rotation angles. The user can then switch between adjacent images perceiving a smooth rotation. Although even small details are

represented, the possible interactions of the user are limited.

A problem common to all these approaches is to achieve a correct color representation on a user's monitor. The sRGB default color space [3] provides a common color space for the exchange of images. Several commercial products such as E-Color or WebSync use a calibration step to capture the monitor settings and deliver custom-corrected images for each user over the Internet. However, Hunt points out in [22] that it is still not guaranteed that a color displayed on a monitor matches the color of the original object.

3 3D Object Pipeline

The generation of a high quality 3D model for a real world object includes several, partially independent steps. Figure 3.1 shows an overview over these steps. First the geometry and the texture of the object are acquired – often using different techniques and acquisition devices 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.

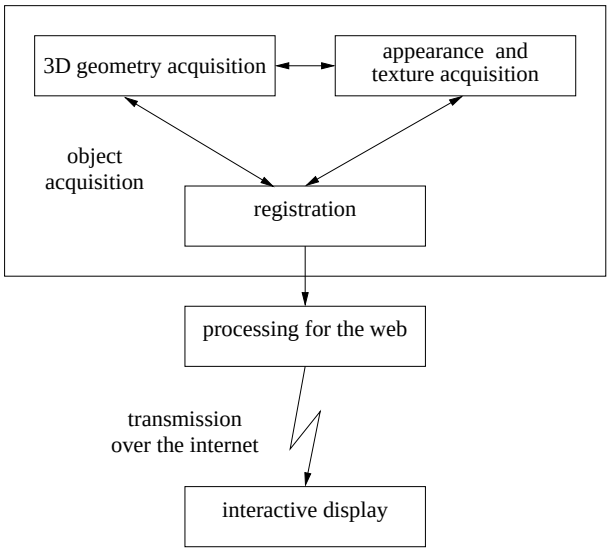

Figure 3.1: *The 3D object pipeline. Depending on the applied techniques geometry acquisition, texture and appearance acquisition, and registration depend on each other in different configurations.*

Once a complete model exists it can be resampled, converted to a different data format, or compressed to make it suitable for transmission over the Internet. Finally, an application running on a client computer should display the model in an interactive and intuitive way without omitting any information.

In the following sections we give a detailed description of all the steps of the 3D object pipeline.

4 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 exists (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 necessary to acquire the geometry of objects using a 3D scanner.

Several research groups including [35, 5] 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 [11]) 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 4.1).

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. [50, 54, 19]).

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

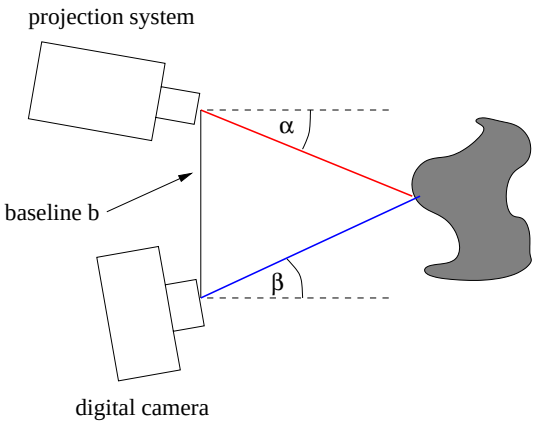

Figure 4.1: *Schematic drawing of an active stereo 3D scanner. Given the intrinsic parameters of the projection system and the camera, the baseline b and the angles α and β, the position of a surface point can be recovered using triangulation.*

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) [6, 41]. The resulting point cloud is triangulated leading to a single triangular mesh using one of a large variety of methods (for an overview see [11]). Further processing steps include smoothing to reduce noise (e.g. using [48, 27]) and editing of the resulting mesh for which a huge selection of tools is available including [29].

Kobbelt et al. [30] give a detailed description of the techniques used for the acquisition and processing of 3D geometry data.

5 Appearance Acquisition

The appearance of an object consists of several surface properties including color, texture, reflection properties, and normal directions. Due to their large number they are difficult to acquire but nevertheless needed 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 [23]. 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 be used to color the 3D geometry model during rendering. The digital pictures are projected onto the model as image textures using texture mapping [16]. To ensure that each part of the object is colored, a sufficient number of images must be taken from different view points [38, 47]. During the projection a perspective correction must be performed to gain a seamless transition between textures of different images (see also Section 6).

5.1 Reflection Properties

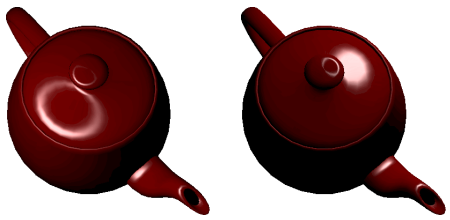

Figure 5.1: *A teapot with complex reflection properties illuminated from two different directions.*

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 towards a single direction, but 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. 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.

Hence a BRDF is a four-dimensional function $f_r(\hat{\omega}_o, \hat{\omega}_i)$ which 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, but since usually only three wavelengths are used, this parameter is not stated explicitly.

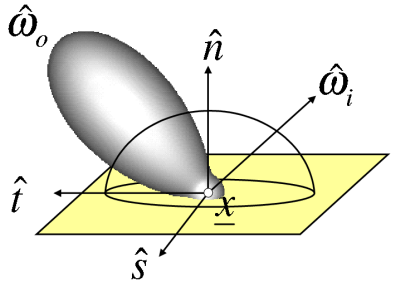

Figure 5.2: *Here you can see the values of a BRDF (depicted as a lobe) for one incident light direction $\hat{\omega}_i$ and every possible outgoing direction $\hat{\omega}_o$.*

A number of analytical BRDF models are known (e.g. [49, 52, 32, 4]). 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 again 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 [36] 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, described by a number of 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 [46]. The set of BRDF samples is then replaced by a few parameters resulting in a more compact representation.

In [37], Marschner et al. 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 reflections. Albedo maps plus one reflection model per surface patch have been acquired for indoor scenes by Yu et al. [53].

An approach to acquire distinct reflection properties for every surface point has been published by Debevec et al. [13]. 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 prohibiting fast transmission via the Internet.

5.2 Normal Maps

The resolution of the acquired geometry of an object is limited by the used 3D scanning device. 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 [7] (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 changing 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.

Normal maps recording small imperfection 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 [45]. 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 [44].

5.3 Camera and Lighting 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 servies as the Kodak Photo CD) and the huge selection of available cameras, lenses and film types. However, the developement 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 remains 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 [2, 15].

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 artefacts which makes them rather unsuitable for measurement purposes. Additional artefacts can occur due to various steps in the image processing chain of digital cameras.

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. 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 5.3 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

Figure 5.3: *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.*

as possible.

In order to use a setup as described above for measurements, various aspects should be calibrated. 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 [50, 54, 19]. To record meaningful and accurate color information, a photometrical calibration is necessary: As the acquired pixel values are usually not proportional to the amount of light hitting the sensor or the film, the response curve of the imaging system should be computed [12, 42]. The spectral characteristics of the light sources can be captured in a white balanceing step. Alternatively an ICC profile [51] can be generated for each each setup that captures the camera and light source characteristics and creates a link to color management systems.

A more technical and in-depth discussion of camera and lighting issues can be found in [14].

6 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 for appearance measurements. Because of this, we further consider the case of two different sensors, a 3D scanner and a digital camera.

6.1 Manual Registration

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. 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 [43]. If enough correspondences are established the transformation can be directly determined using one of various kinds of camera calibration methods (e.g [50, 54, 19]). 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.

6.2 Automatic Registration

To simplify the registration process some semi-automatic approaches have been published [38, 39]. 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.

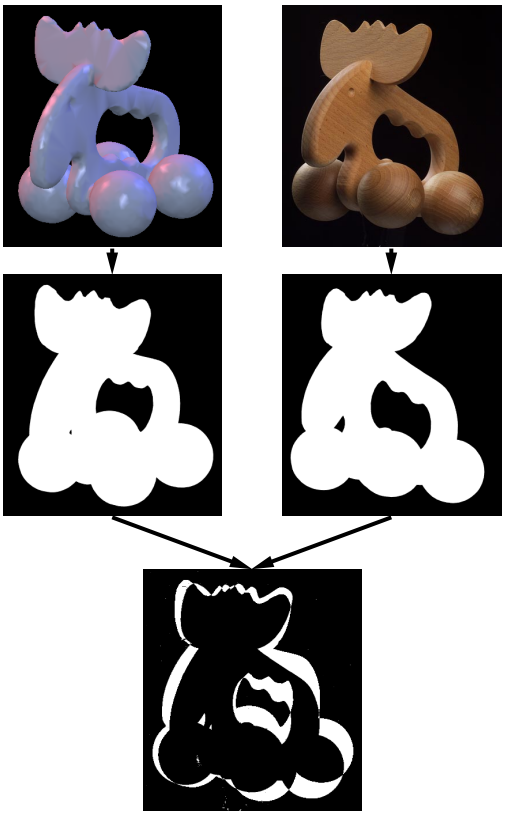

Figure 6.1: *Measuring the difference between photo (right) and one view of the model (left) by the area occupied by the XOR-ed foreground pixels.*

In [33], 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 6.1. 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. 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 Processing for the Web

7.1 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 7.1.

Figure 7.1: *The 3D model is aligned to a captured picture which then can be mapped as a texture onto the geometry.*

As already stated, multiple images are necessary to cover the entire surface since only some parts of the object's surface are visible in each image. But one triangle can of course be visible in more than one image. In this case we select the image for texturing the triangle that corresponds to the best viewing angle.

Figure 7.2: *Combining visible parts of the original images as well as blended triangles into one large image. In this case it is done for three images only.*

Unfortunately, this will result in neighboring regions which are textured by different images. The transition at the boundary between neighboring regions may become clearly visible as a sharp discontinuity. A smooth transition can be achieved if we blend between the images across the boundary triangles using alpha blending (see [33] for further explanations). Now, there are two different kinds of triangles: triangles textured by exactly one image and triangles where two or three images are blended together. In order to apply the same rendering steps to all triangles it is useful to precompute the actual blending, storing the result into a new image. This leads to a considerable number of small images, one image per blended triangle. Instead of using a set of images, meaning a set of textures, we copy the relevant parts of the original images and the blended triangles into one large texture as is shown in Figure 7.2. A similar technique has also been applied by Rocchini et al. [43]. By removing those parts of the images where only the background is visible and through intelligent packing the size of the resulting texture can be decreased compared to the overall volume of the original images and transmission speed can be increased.

A single image has the further advantage that it can be compressed and transformed into a streamable representation with less effort.

7.2 Mesh Preparation

As already stated, streamable representations exist for 3D models as well. The progressive meshing approach which was introduced by Hoppe [20, 21] is ideal for the transmission of complex 3D models via the Web. To generate a progressive representation of a triangle mesh a simplification algorithm is used to remove most of the vertices from the mesh until only a coarse mesh is left consisting of a few triangles. This small mesh is transmitted first allowing the client computer to display a coarse version of the model. The remaining vertices are reinserted into the mesh in reverse order as soon as the data is available. If the coarse appearance of the partially transmitted mesh is unacceptable a mesh subdivision algorithm (e.g. [28]) can be applied at the client side to produce a smooth surface.

In cases where it is not possible to display a highly detailed mesh (e.g. when the client is a small PDA or a mobile phone) it is possible to transmit only that part of the progressive mesh that can be displayed on the device. Figure 7.3 shows an example where a mesh consisting of 2000 triangles is rendered interactively on a PalmPilot PDA using the Palm 3D-Viewer software [9].

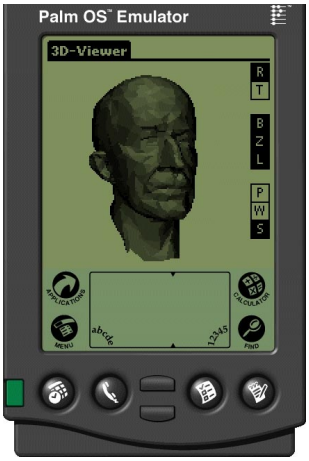

Figure 7.3: *Interactive rendering of a 2000 triangle mesh on a PalmPilot PDA*

8 Interactive Display

Representing the entire texture data as one large image simplifies the rendering process of the model because it has to deal with one texture only. On the other hand, since neighboring triangles of the model may still be textured using non-neighboring parts of the image more than one texture coordinate has to be assigned to each vertex. Actually, each triangle possesses its own set of texture coordinates for its vertices. Rendering single textured triangles is unfortunately slightly slower than rendering many adjacent triangles as a triangle strip. Nevertheless, interactive rates can still be achieved.

8.1 Rendering with Arbitrary BRDFs

If not only a color texture was obtained, but also an average BRDF as described in Section 5 the question remains how it can be rendered interactively. Standard OpenGL only supports the empirical and physically implausible Phong model, which makes surfaces always look "plastic"-like.

In order to render surfaces with other BRDFs two similar approaches [18, 25] can be used. Both approaches decompose the four-dimensional BRDF $f_r(\hat{\omega}_o, \hat{\omega}_i)$ into a product of two two-dimensional functions $g(\hat{\omega}_o)$ and $h(\hat{\omega}_i)$. These two functions are stored in two texture maps and re-multiplied together using blending. The approach by Heidrich and Seidel [18] decomposes the analytical Cook-Torrance model [10]. The approach by Kautz and McCool [25] numerically decomposes (almost) any BRDF by choosing a better parameterization for the BRDF.

Rendering is very simple. 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.

For an example of this technique please see Figure 5.1.

8.2 Rendering with Normal/Bump Maps

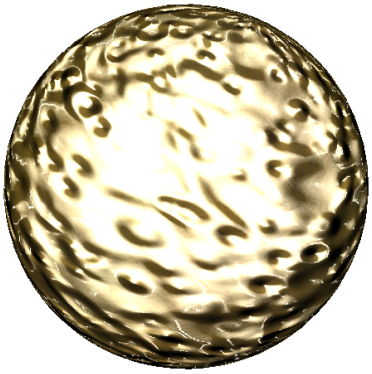

Figure 8.1: *A bump map applied to a sphere*

Blinn [7] 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.

A new algorithm has been proposed to render bump maps [26] (as shown in Figure 8.1) at interactive rates using texture maps containing per-pixel normals, which are used to perform the lighting calculations instead of per-vertex normals.

This algorithm relies on features now supported by many graphics cards. These features include per-pixel dot-products, multiplication, addition, subtraction, so lighting models/BRDFs using only these operations can be used to do bump mapping.

Usually the Blinn-Phong model [8] is used to perform bump mapping, because this model mainly uses dot-products. For more details, please see [26].

Heidrich et al. [17] also computed consistent illumination on bump maps in fractions of a second exploiting regular graphics hardware.

8.3 Shift-Variant BRDFs

One reflection model per surface can be evaluated very fast using the approach presented in [18, 25]. If the reflection properties vary across the surface shift-variant BRDFs must be considered which have been interactively rendered by Kautz et al. [24], see Figure 8.2.

Some of these algorithms take advantage of new features of current graphics hardware, e.g. multi texturing and texture combiners [40]. Although they are

Figure 8.2: *A shift-variant BRDF applied to a sphere*

currently not available on all client machines they will become more and more widespread. In the future there should be a standardized way of transmitting and rendering more complex appearance models including color, BRDFs and bump maps.

9 An Example

In this section we describe two practical examples for high quality 3D object acquisition. Geometry and texture data has been acquired for a wooden elk and clay bird. Some example images are given in Figure 9.1.

Both 3D geometry models have been 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.

Due to the fact that a structured light scanner can only acquire surface points that are visible from the camera and projector position at the same time the elk mesh contained several holes – mainly around the axes – which were filled manually. Afterwards a filtering step was applied to improve the smoothness of the meshes. To accelerate further processing the initial models consisting of nearly 150000 triangles each were simplified to 11000 triangles for the elk and 7000 triangles for the bird model.

The images for the textures were taken with a Kodak DCS 560 professional digital camera, which outputs images consisting of 6 million pixels. Since they were too large we downsampled them immediately. The elk texture consists of 15 different images. Only a small part of the surface has not been covered. Because of its simpler geometry 10 images were sufficient to acquire the texture of the bird model.

Instead of moving the objects, the camera was moved around them in order to keep the illumination constant. Large diffusor boxes were used to illuminate all parts of the surface and to obtain smooth shadow regions.

The registration of the images with the 3D models took about one hour for the elk and half an hour for the bird because the bird model consists of less triangles and has been registered using lower resolution images. Most of the time was spent finding a starting point for the optimization. This time could be saved by a rough manual alignment performed by the user.

photo texture

Figure 9.1: *Novel viewpoint. The images in the left column have not been used to generate the textures. The reconstructed models in the right column were rendered using the acquired textures.*

Except for some artefacts due to non-diffuse surface reflections and missing data the reproduced images (Figures 9.1) approximate the original photos very well. The textures are precisely aligned with the geometry and although a rather coarse 3D model has been used the applied textures visually reproduce surface features that are not present in the geometry.

10 Conclusion

We presented a framework to acquire high quality 3D models of real world objects including both geometry and appearance information represented by textures, bump maps or BRDFs. The collected data can be converted into a progressive representation to meet the constraints of incremental transmission and of clients with low computing power. Sophisticated rendering algorithms are required to visualize the acquired model.

Delivering high quality 3D models of objects to a user's computer via the Internet is a complex and challenging problem. But unless it is solved many ambitious e-commerce, virtual museum, or digital document projects will not succeed.

During the last couple of years a lot of effort has been put into developing techniques that are necessary to achieve this goal: The computer graphics community has been working hard to efficiently acquire models that contain all essential details of an object. Simultaneously, the graphics hardware industry – driven by the computer game industry – developed comparably cheap but still powerful graphics hardware for ordinary PCs that are required for the interactive display of these large models. Fast Internet connections, streaming file formats, and standard software plug-ins make it already possible to transmit smaller 3D models to the users and to display them.

Nevertheless, there is still a considerable effort required to build a complete and easy to use system that can acquire, transmit, and display realistic 3D models on every users computer.

11 Acknowledgments

We would like to thank Thomas Neumann, Kolja Kähler, and Christian Rössl for their help in acquiring the data sets used in this paper. Thanks also to Mario Botsch, and Anshuman Aggarwal for rendering some of the images.

Bibliography

[1] Vadim Abadjev, Miguel del Rosario, Alexei Lebedev, Alexander Migdal, and Victor Paskhaver. Metastream. *VRML 99: Fourth Symposium on the Virtual Reality Modeling Language*, pages 53–62, February 1999.

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

[3] Matthew Anderson, Ricardo Motta, Srinivasan Chandrasekar, and Michael Stokes. Proposal for a standard default color space for the internet – sRGB. In *Proceedings of the 4th Color Imaging Conference: Color Science, Systems and Applications*, pages 238–245. IS&T, 1995.

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

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

[6] 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.

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

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

[9] M. Botsch. Palm 3D-Viewer. Available at http://www.mpi-sb.mpg.de/~botsch/palmpilot.html.

[10] Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. In *Computer Graphics (Proceedings of SIGGRAPH 81)*, pages 307–316, August 1981.

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

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

[13] 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.

[14] 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.

[15] 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).

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

[17] 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.

[18] Wolfgang Heidrich and Hans-Peter Seidel. Realistic, hardware-accelerated shading and lighting. In *Proceedings of SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, pages 171–178, August 1999.

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

[20] H. Hoppe. Progressive meshes. In *Proceedings of SIGGRAPH 1996*, pages 99–108, 1996.

[21] H. Hoppe. Efficient implementation of progressive meshes. *Computer and Graphics*, 22(1):27–36, 1998.

[22] Robert W. G. Hunt. How to shop on the web without seeing red. In Jennifer Gille and James King, editors, *Proceedings of the 8th Color Imaging Conference on Color Science and Engeneering Systems, Technologies and Applications (CIC-00)*, pages 2–7. IS&T, November 2000.

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

[24] 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.

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

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

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

[28] Leif Kobbelt. sqrt(3) subdivision. In *Proceedings of SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, pages 103–112, July 2000.

[29] 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.

[30] 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.

[31] Keeranoor G. Kumar, James S. Lipscomb, Arun Ramchandra, Subrina Chang, William L. Gaddy, Ross Leung, Steve Wood, Liang-Jie Zhang, Jeane Chen, and Jai Menon. The hotmedia architecture: Progressive & interactive rich media for the internet. IBM White Paper. Available at http://www.developer.ibm.com/library/articles/hotmedia.html.

[32] 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.

[33] 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.

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

[35] 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.

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

[37] 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.

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

[39] 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.

[40] NVIDIA Corporation. *NVIDIA OpenGL Extension Specifications*, October 1999. Available from http://www.nvidia.com.

[41] 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.

[42] Mark A. Robertson, Sean Borman, and Robert L. Stevenson. Dynamic range improvement through multiple exposures. In *Proceedings of ICIP 1999*, 1999.

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

[44] 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.

[45] 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.

[46] 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.

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

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

[49] 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.

[50] 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.

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

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

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

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