Time-Varying BTFs

Tobias Langenbucher, Sebastian Merzbach, David Möller,
Sebastian Ochmann, Richard Vock, Welf Warnecke, Michael Zschippig
Supervised by: Martin Rump

Institute of Computer Science
Rheinische Friedrich-Wilhelms-Universität Bonn
Bonn/Germany

Abstract

There have been several approaches to model and capture time-varying materials. Modeling approaches provide good results but are sometimes hard to apply because underlying processes are not yet understood or very complex. In this paper, we present a data-driven approach to record aging effects of metal and car paint with the help of Bidirectional Texture Functions. BTFs precisely capture spatially varying reflectance properties of a given material. However, once captured, one cannot change the appearance of a material that ages. Instead, we measure at several distinct aging steps and combine this information into a time-varying BTF which allows the user to interpolate between different stages of the aging process.

1 Introduction

One goal of computer graphics is to create photo-realistic images. In order to accomplish this goal, it is necessary to take reflectance properties of materials into account. These properties depend on color, shininess, translucency and inner structure of a given material.

Additionally, almost all materials age, e.g. through corrosion, scratches, or bleaching, leading to a change of their visual properties over time. It is important to capture these effects as well since they strongly contribute to the realism of rendered materials and being able to smoothly interpolate the age of materials offers vast possibilities for industrial, artistic and scientific applications. This can be used for example to examine what a product will look like after months or years of use.

This paper is a practical report about our work measuring and rendering TV-BTFs (Time-Varying Bidirectional Texture Function). We chose time-varying BTFs because they capture these effects very well and have not been analyzed before.

Solving the rendering equation by Kajiya [4] for any given scene is the primary challenge in realistic rendering. One important component of this equation is a function that returns the amount of reflected light given the direction of incoming and outgoing light. A popular approach to represent this function is called BRDF (Bidirectional Reflectance Distribution Function). The BRDF is a function that describes how much light from an incoming angle is reflected to an outgoing angle. It does not take subsurface-scattering into account and is restricted to homogeneous surfaces only. Some other groups have already worked on Time-Varying BRDFs [8]; those however, unlike BTFs, are not capable of capturing visual properties of whole patches of a material.

There are two popular ways to generate a BTF. The first one uses phenomenological and analytical models, the second one is a data-driven approach. One big disadvantage of the model-based concept is the fact that you have to create a very detailed and physically accurate model of the material, which is often excessively time-consuming or even impossible. We therefore use a data-driven approach, measuring the BTF of a material using a dome consisting of 151 digital consumer cameras. An in-depth description of this setup is given in chapter 4.1.

We examined our approach by letting a simple metal plate rust and scratching a sample of green car paint. To get time-varying results, we measured several times with our increasingly aged materials and combined the data into a single Time-Varying BTF. Using this TV-BTF, it is possible to interpolate over the lifetime of the material.

In the following chapters we will describe the basics of rendering techniques, global illumination, capturing, compression and decompression of Time-Varying BTFs. In chapter 6 we will present some images of basic geometric figures rendered with our Time-Varying BTF.

2 Basics

To achieve photo-realistic rendering, one has to measure reflectance behavior of materials. This behavior is characterized by how a ray of light with wavelength λ_i, hitting the surface in a point $\mathbf{x_i}$ in direction $\omega_\mathbf{i}$ at time t_i, travels through the material and leaves at another point $\mathbf{x_r}$ in direction $\omega_\mathbf{r}$ at time t_r and with possibly altered wavelength λ_r. As shown in Figure 1, this leads to a twelve-dimensional function (the points $\mathbf{x_i}$, $\mathbf{x_r}$ have two spatial coordinates and the directions $\omega_\mathbf{i}$, $\omega_\mathbf{r}$ are represented by two angles).

It is, however, computationally way too complex and thus currently impractical to fully describe arbitrary ma-

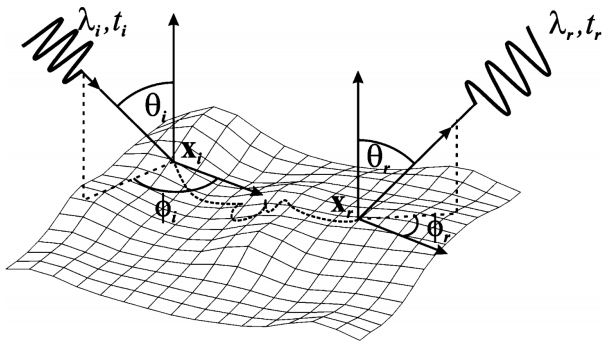

Figure 1: The twelve-dimensional function describing light transport in materials. [5]

terials because of the high dimensionality of a physically correct surface reflectance function. Measuring or storing this information is beyond today's hardware capabilities. As a result, several simplifications are used to reduce the parameter count. Phenomena such as fluorescence (change of wavelength during reflection) or phosphorescence (re-emission of absorbed light after a certain amount of time) are usually neglected, i.e. $\lambda_i = \lambda_r$ and $t_i = t_r$. Furthermore, the electro-magnetic spectrum is usually discretized into three wavelengths, i.e. red, green and blue light.

These simplifications lead to the time-independent eight-dimensional BSSRDF (Bidirectional Surface Scattering Reflectance Distribution Function). The BSSRDF is still quite complex and hence, further reductions have to be applied. A simplification of the BSSRDF is the BTF (Bidirectional Texture Function) which does not fully account for subsurface scattering.

$$\begin{aligned} &\mathrm{BTF}(\mathbf{x_r}, \theta_i, \phi_i, \theta_r, \phi_r) \\ &= \int_{\mathrm{Surface}} \mathrm{BSSRDF}(\mathbf{x_i}, \mathbf{x_r}, \theta_i, \phi_i, \theta_r, \phi_r) d\mathbf{x_i} \end{aligned} \tag{1}$$

One can easily see in the above equation that the BTF only depends on *one* position. It accumulates light that is scattered in the material from neighboring regions. Most measurement setups that are used to capture BTFs of planar materials (c.f. chapter 4.1) usually gather the subsurface scattering part; it is, however, inseparably contained in the measurements.

Even further simplifications include the four-dimensional BRDF (Bidirectional Reflectance Distribution Function) that discards any positional information and only depends on incident and outgoing directions.

$$\mathrm{BRDF}(\theta_i, \phi_i, \theta_r, \phi_r) = \frac{\int_{\mathrm{Surface}} \mathrm{BTF}(\mathbf{x_r}, \theta_i, \phi_i, \theta_r, \phi_r) d\mathbf{x_r}}{||\mathrm{Surface}||_{BRDF}} \tag{2}$$

Rendered with a BRDF, a material shows the same reflectance behavior over the whole surface.

3 Previous Work

In the field of time-varying materials, BRDFs, variable textures and aging-models have already been analyzed. [2] provides a good overview of this.

Time-varying BRDFs: At the CAVE laboratory at Columbia University, Sun et al. have created a database for time-varying BRDFs. They measured BRDFs of aging materials every 36 seconds. With this setup, they obtained a wide range of data about different time-varying materials. In the next step, they fit BRDF functions to this data and extracted the parameters. After that, they used these parameters to develop time-varying BRDFs. In [8], Sun et al. focused on drying materials and accumulation of dust.

Time-varying textures: In [3], Enrique et al. captured several hundred pictures of a given material over time. They extended the texture function with a time parameter. Afterwards, they were able to texture a material at any point of time.

Aging-models: To model aging effects on metallic patinas, Dorsey et al. use a collection of operators applied to a layer model. Each of these operators represent a different weathering effect. The result is a surface of different thickness and therefore varying reflectance properties [1].

4 BTF-Pipeline

4.1 Acquisition

A bidirectional texture function (BTF) is a six-dimensional function which provides information about the appearance of a material depending on the position on the object's surface as well as the viewing and lighting directions. Practical measurement of this function on a material sample, or rather an approximation thereof, is performed by taking several thousand pictures from different, discretized viewing and lighting angles.

We obtain the required images by using a hemispherical camera dome by Sarlette et al., consisting of 151 consumer cameras as described in further detail in [7]. All cameras simultaneously shoot photos of a material sample placed in a fixed position inside the dome while the integrated flash light of one of the cameras illuminates the sample. This process is repeated for each lighting direction which yields photos for all 22,801 combinations of camera and lighting positions.

The setup allows for rapid, automated measurement of the materials thanks to the highly parallelized process as well as a relatively simple post-processing phase due to the rigid camera setup with known extrinsic (position and orientation of the camera) and intrinsic (focal length, etc.) camera parameters.

4.2 Geometric Calibration

To determine these parameters, we use the same calibration technique as presented in [7], which uses a planar cal-

ibration object with 121 well-known LED features. The use of active markers compared to passive ones leads to a more accurate and robust detection.

We first measure without optical zoom. As we can make the assumption that the intrinsic parameters are the same, the extrinsic parameters can be estimated using Zhang's camera calibration algorithm [9]. Afterwards, the intrinsic parameters are calculated zoom step by zoom step until the measurement zoom step is reached.

4.3 Radiometric Calibration

To compensate slight differences of the camera sensors and the spectral distribution of the flash lights, four reflectance targets with different albedo values and known reflectance behavior are positioned near the target.

Using these markers, white-balance factors are calculated to achieve consistent color reproduction and to calculate the effects originating from spectral differences in flash light. In a post-processing step, these effects are eliminated by simply multiplying measured values with white-balance factors.

4.4 Image post-processing

After the raw images have been acquired they need to be transformed into a common coordinate system where all images have the same orientation and size.

We can compute the necessary transformation to reproject the region of interest within each image to the desired coordinate system by using the known extrinsic camera parameters as well as corner markers in the images. Small inaccuracies during measurement are compensated by searching a small region around the assumed corner positions and using the adjusted positions for the transformation.

For decent rendering results, we need high dynamic range images. However, one measurement pass from the consumer cameras with some given ISO speed (i.e. CCD-sensitivity) and flash light intensity only provides low dynamic range images. Thus, we perform multiple passes with different settings for the ISO speed and flash intensity which we later merge into one HDR image for each combination of the camera and light directions. The relative position of the single LDR steps in energy space is extracted from the white markers.

4.5 Compression and Decompression

Storing the resulting images uncompressed - assuming an image size of 768×768 pixels each - takes up approximately 150 GB of storage capacity, which is not feasible for later usage in a shader.

In order to use the captured BTF data for fast rendering, a decent compression algorithm is needed which has three key requirements:

1. It should allow for fast random access to a single pixel.
2. It should exploit recurring patterns and similarities in the data in order to achieve high compression rates, feasible for GPU calculations.
3. It should compress the data in reasonable time.

While the latter (as an offline computation) is a less critical requirement than the other two, the time spent compressing should meet at least decent practical considerations.

Traditionally, algorithms that fit these requirements project the data into a lower-dimensional space by building a set of orthogonal base vectors and dropping the least significant ones.

Since the uncompressed TV-BTF data is a seven-dimensional tensor, common ways to compress it involve either tensor or matrix decomposition, i.e. unfolding the tensor into a matrix beforehand. Tensor decomposition techniques allow for exploitation of patterns and similarities in each dimension separately and hence tend to achieve higher compression ratios than a matrix decomposition which is restrained to one or two dimensions.

However, known tensor decomposition algorithms take too much time to precompute and are thus not applicable. We therefore chose a matrix decomposition approach by using principal component analysis (PCA) to project the BTF data into a theoretically optimal orthogonal base. A PCA currently fits our requirements stated above best:

1. Access to a single value of the BTF data involves only a scalar product of 2 c-dimensional vectors - c being the dimension of our orthogonal subspace (i.e. the number of principal components kept).
2. Considering an intelligent unfolding into a matrix, a PCA builds an "optimal" subspace by using orthogonalized directions of the highest covariance, thus exploiting similarities and patterns in our data.
3. By using an EM-PCA algorithm introduced in [6] we are able to compute only the c most important components in a fast way without losing computation time on components we would drop later anyways.

In order to calculate the PCA, a matrix representation of the measured data is needed. For the PCA to work properly, i.e. generate quickly descending eigenvalues and hence allow for good compression, similarities in the data should be placed close to each other in the matrix. The expected variance between adjacent pixels is less than the expected variance between one pixel under different lighting conditions.

To meet these requirements, we decided to store each color channel for each time step in separate matrices, compressing each of them using PCA. Therefore, one time step consists of three matrices. The scheme works as follows:

Each column of each matrix holds the values of one color channel (red, green or blue) of one entire HDR-image. All images are written into the matrix sorted by light and view directions (see Figure 2). The dimension of the resulting matrix is $(w \cdot h) \times (v \cdot l)$, where w and h are the width and height of the HDR-images, v is the number of view directions and l is the number of light directions.

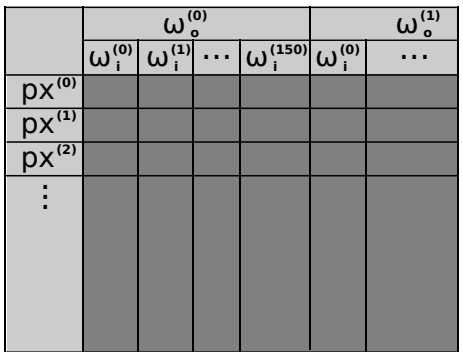

Figure 2: BTF data matrix layout for one color channel.

Regarding the large amount of data involved, composing the BTF data matrix in this way involves the problem of an out-of-core matrix transposition, since only a limited amount of images can be kept in memory simultaneously. However, this computation is a very time-consuming task if the matrix is big. As IO (reading images from hard disk, writing the data to the matrix file) is the bottleneck of this task, we used an algorithm, that writes the matrix in chunks that fit into memory (see Figure 3). This minimizes the number of times each image has to be read.

Creating a single matrix including all measured time steps and thus using a global compression for one large matrix would significantly increase compression time and memory usage (beyond what we are able to compute using our GPU-implementation in decent quality). For this reason, we chose to use separate matrices for each time step and interpolated individual pixel values from these matrices linearly in the shader. In the case of rusting metal, color and geometric properties of the material change dramatically over time so that a global compression would not provide any reasonable benefits concerning compression ratio and quality. Using separate matrices, we achieved a sufficient compression ratio to use for realtime-shading (in case of the metal we cut down matrix size from 17.5 GB to 78.7 MB).

5 Measurement

The following table gives a quick overview of the measurement setup. In detail explanation follows in the next two sections.

```
imgList = getSortedHDRImageFilenames();
largeFileR = reserveMemory();
largeFileG = reserveMemory();
largeFileB = reserveMemory();
written = 0;
// chosen according to resulting memory usage:
numScanlines = 20;

blocks = emptyListOfImages();
while (written < imgHeight) {
  // load images into RAM
  for (i = 0; i < sortedList.size(); ++i) {
    // get i'th block of scanlines
    blocks[i] = readImageBlock(
      imgList[i],
      // first scanline
      written,
      // last scanline
      min(imgHeight, written + numScanlines)
    );
  }

  // write data into matrix file pixel-wise
  for (px = 0; px < imgHeight*imgWidth; ++i) {
    for (img = 0; img < blocks.size(); ++i) {
      // write R, G and B values
      largeFileR.write(blocks[img][px][R]);
      largeFileG.write(blocks[img][px][G]);
      largeFileB.write(blocks[img][px][B]);
    }
  }

  written += numScanlines;
}
```

Figure 3: Pseudo code for writing the BTF data matrix.

light directions	151
view directions per flash	151
iso levels	4
aging levels	4
total pictures taken	364,816

5.1 Metal

To measure visual properties of metal rusting over time, we took measurements of a metal plate in a progressively rusted state. Between the measurements in the dome we sped up the aging process using a rusting-chamber as schematized in Figure 4. The bottom was filled with warm (approx. 28°C), salted water which was tempered and circulated with an aquarium heater and pumps. The probe stood nearly upright (with an angle of about 75°) on a support frame. Because the rusting process is greatly accelerated by the presence of acid we placed a small basin filled with vinegar in front of it. To counteract water drops on the probe, we heated the roof above with a lamp, so that the vapor could not condense in this area.

Whenever we considered a new stage of aging to have been reached, a new measurement was taken. The (entirely experimental) time intervals used for the aging of the material are listed in Table 1. We measured four

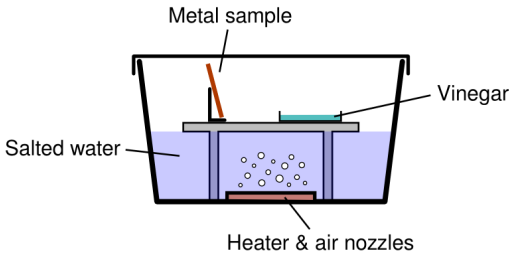

Figure 4: Our setup to accelerate the rusting process between measurements.

states of the rusted material, each measurement consisting of four different exposure time and flash settings for later HDR combination. Thus we acquired a total of $151^2 \cdot 4^2 = 364,816$ individual images, resulting in about 1,075 GB of storage space for the unprocessed shots.

Figure 5 shows photo-montages of the different rusting states assembled from raw measurement footage of the metal experiment. The pictures from the four different rusting states have been combined and faded from left to right, showing the different rusting states.

Measurement #	Time rusted
1	0 min
2	+10 min
3	+30 min
4	+30 min

Table 1: Time intervals for the rusting process.

5.2 Car Paint

Before taking initial measurements of the car paint, we had to clear it of pre-existent scratches by applying common car polish. We then recorded the pictures for a BTF. Afterwards, we added new scratches with a coarse and raspy sponge that is normally used to wipe insect remnants off your windshield. We tried to move the sponge in circles to scratch the surface evenly along all directions. Utmost care had to be taken not to add too many new scratches because the procedure is irreversible. After each step of scratching, we carefully determined the amount of scratches by taking photos with a flash light, because the scratches are hardly visible in a diffuse lighting as it is common indoors.

Just like the patina material, the car paint shows highly specular reflections, so we had to use four different ISO speeds and flash intensities to get qualitatively acceptable HDR pictures. We used five different stages of altered material, leading to a total of 364,816 pictures of the material and about 1 TB storage space.

A problem exists in our measurement setup with highly reflective materials, as there are reflections of cameras from the opposite side of the dome from low angle shots. This problem and its solution are discussed in [7]. However, we did not apply this solution because these artifacts are negligible with our materials.

6 Results

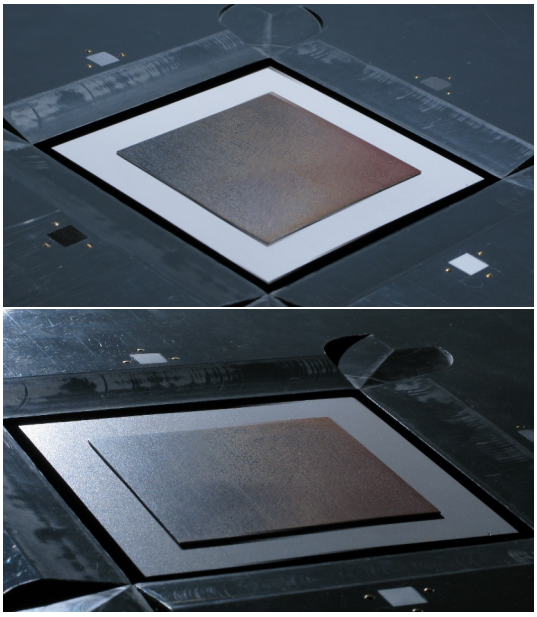

Figure 5: Montages of raw measurement footage.

We were able to compress more than one terabyte of raw image data to a few megabytes. Using only the 100 most significant components in the PCA, we achieved compression ratios of $42,000:1$ for the car paint.

Figure 6 depicts path tracer renderings of a sphere with the different rusting states and linear interpolations between them applied to the surface.

In Figure 7, three actual photographs of the first, second and fifth measurement are shown. The third and fourth picture were created by linear interpolation between the real data. In Figure 8 you can see actual renderings of the first and last aging stage.

Interpolation works well on the data and it can be used for TV-BTFs. One can argue that linear interpolation is unsuitable for our approach. Scratches appear suddenly and do not fade in smoothly, as it occurs when interpolating linearly. However, these artifacts are hardly visible and handling this problem properly is beyond the scope of this paper.

7 Conclusion

The results show that our approach can be used to render images of time-dependent materials. The data can be com-

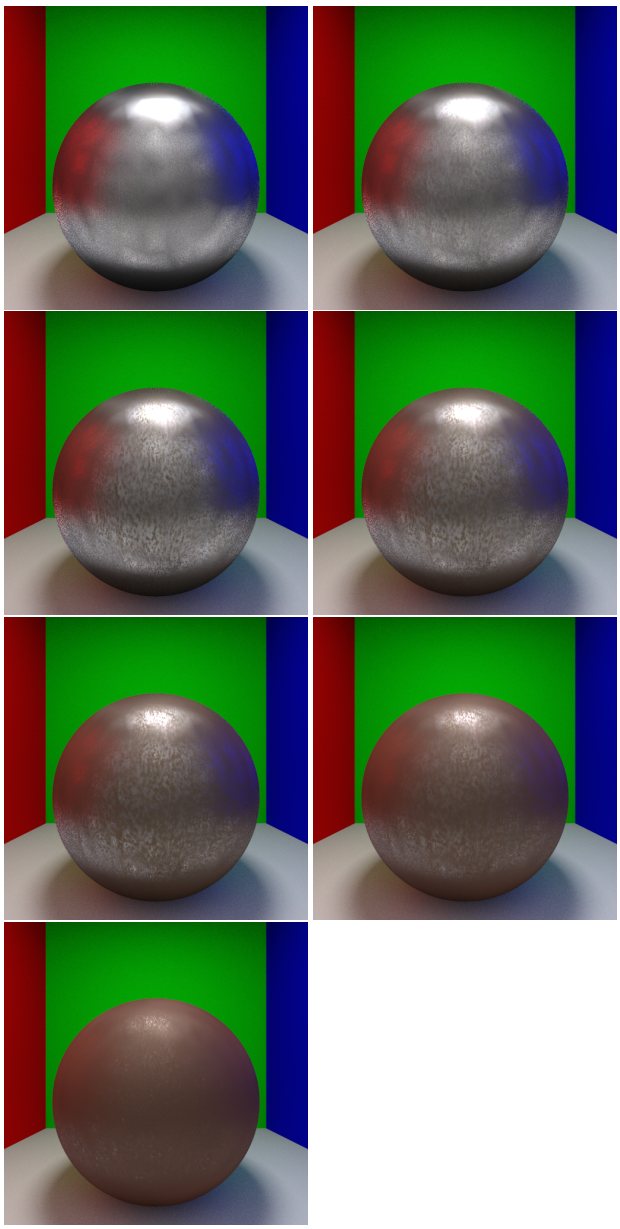

Figure 6: Rusted metal renderings. Left column: Rendered images of the four measured rusting states; right column: linear interpolations between the neighboring states within the shader.

pressed very well and the interpolated aging levels appear realistic.

There are however some problems that have to be dealt with in future projects. Small errors in measurement can lead to big errors in the resulting BTFs, e.g. scratches on the car paint do not match in different aging steps. This problem is caused by the sample holder not being fixed in the dome. It is possible for the sample to lie in a slightly different angle or position relative to the flashlights in subsequent shots, resulting in different highlights on the scratches. This effect can not be compensated by

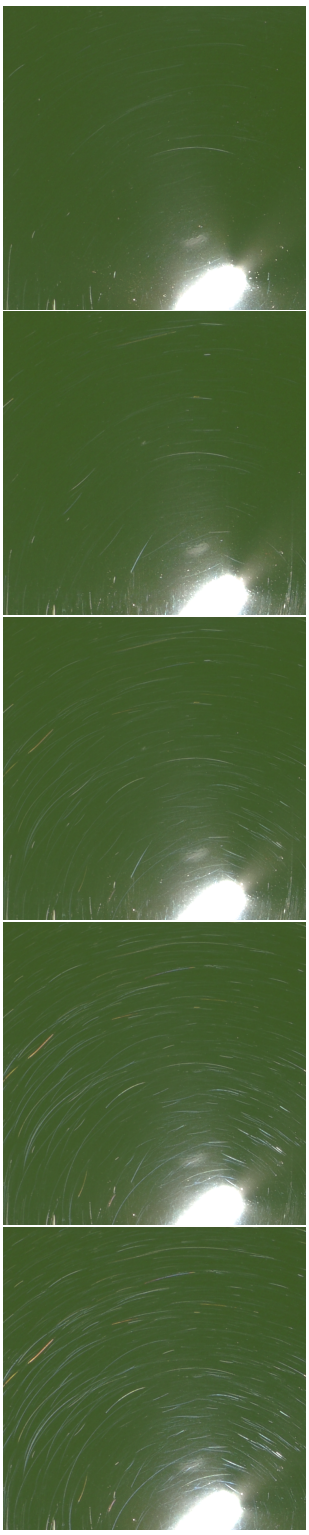

Figure 7: Montages of raw measurement footage, from top to bottom: First measurement, second measurement, followed by two interpolations and the last measurement

the rectifications used so far. It is, however, hardly visible because of the small scale of these highlights.

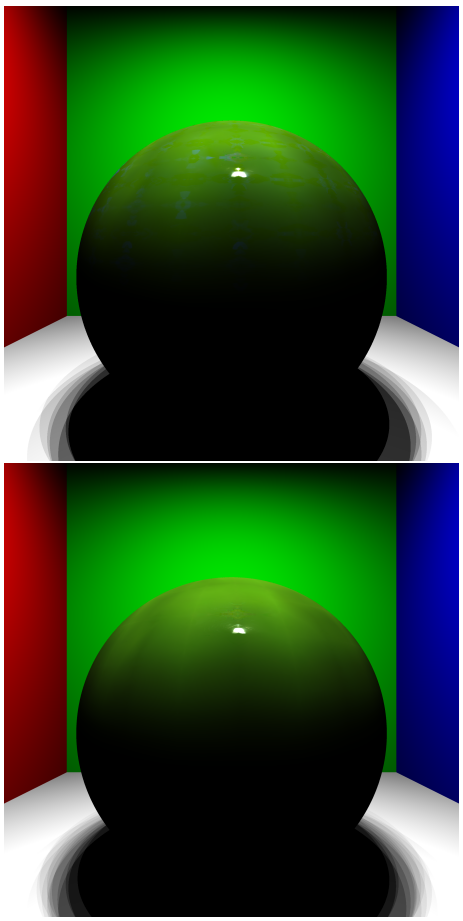

Figure 8: Actual renderings of the first and last aging stage of the car paint material.

8 Acknowledgment

We would like to thank our project supervisor Martin Rump, as well as Ralf Sarlette and Professor Dr. Reinhard Klein who allowed us to gain better knowledge of BTFs and insight into the research activities and work in our university's computer graphics department.

References

[1] Julie Dorsey and Pat Hanrahan. Modeling and rendering of metallic patinas. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 387–396, New York, NY, USA, 1996. ACM.

[2] Julie Dorsey, Holly Rushmeier, and François Sillion. Advanced material appearance modeling. In *SIGGRAPH '08: ACM SIGGRAPH 2008 classes*, pages 1–145, New York, NY, USA, 2008. ACM.

[3] Sebastian Enrique, Melissa Koudelka, Peter Belhumeur, Julie Dorsey, Shree Nayar, and Ravi Ramamoorthi. Time-varying textures: definition, acquisition, and synthesis. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches*, page 130, New York, NY, USA, 2005. ACM.

[4] James T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, 1986.

[5] Gero Müller, Jan Meseth, Mirko Sattler, Ralf Sarlette, and Reinhard Klein. Acquisition, synthesis and rendering of bidirectional texture functions. *Computer Graphics Forum*, 24(1):83–109, March 2005.

[6] Roland Ruiters, Martin Rump, and Reinhard Klein. Parallelized matrix factorization for fast btf compression. In *Eurographics Symposium on Parallel Graphics and Visualization*, pages 25–32, March 2009.

[7] Martin Rump, Gero Müller, Ralf Sarlette, Dirk Koch, and Reinhard Klein. Photo-realistic rendering of metallic car paint from image-based measurements. *Computer Graphics Forum*, 27(2), April 2008.

[8] Bo Sun, Kalyan Sunkavalli, Ravi Ramamoorthi, Peter Belhumeur, and Shree Nayar. Time-Varying BRDFs. In *Eurographics 2006 Workshop on Natural Phenomena*, Sep 2006.

[9] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000.