

A reflectometer setup for spectral BTF measurement

Daniel Lyssi*

Institute of Computer Science II, University of Bonn, Germany

Abstract

In computer graphics, information about a material's reflectance properties is needed to visualize it correctly in a virtual scene. A common representation of these properties is the BTF (Bidirectional Texture Function). Ideally, the BTFs should be available in a *spectral* form (more than 10 non-overlapping wavelength bands) to allow for correct color calculations. Currently, however, all BTF measurement methods rely on RGB cameras or similar tristimulus-based devices. Reasons for this are the large amounts of data generated by spectral measurements, which would have been too much to handle just a few years ago, the long measurement times and the enormous cost of specialized hardware needed for a spectral acquisition. Since RGB values are not sufficient for the correct reproduction of a material's visual appearance under arbitrary lighting conditions, we now present a measurement setup for the acquisition of spectral BTFs.

Keywords: BTF, material measurement, spectral

1 Introduction

The correct visualization of materials in virtual environments is needed in many parts of the industry. It is used, amongst other things, to make decisions during product design or to generate images for advertisements. The visualization requires exact information about the material's reflectance properties. These properties depend on a multitude of parameters: Incoming and outgoing direction of the light waves, the points on the surface where they enter and leave the material (subsurface scattering), the points in time when they enter and leave (phosphorescence), the wavelength of the incoming and outgoing light (fluorescence and other wavelength dependent effects) and its polarization. These parameters are visualized in figure 3. Together, they lead to a reflectance function dependent on fourteen variables:

$$\rho(x_i, x_r, \theta_i, \phi_i, \theta_r, \phi_r, t_i, t_r, \lambda_i, \lambda_r, p_i, p_r) \quad (1)$$

In this function, x_i and x_r represent the surface points of incoming and outgoing light, (θ_i, ϕ_i) and (θ_r, ϕ_r) zenith and azimuth angle of the incoming and outgoing light, t_i and t_r the points in time when the light enters and leaves the material, λ_i and λ_r the wavelengths of incoming and

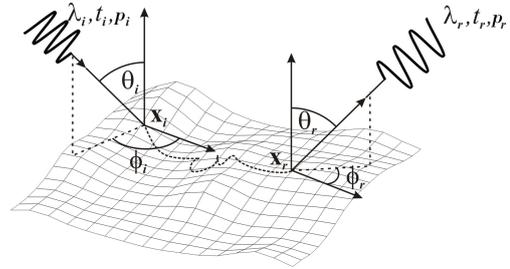


Figure 3: Visualization of the parameters describing the reflectance properties of a material

outgoing light and, finally, p_i and p_r the polarization of the incoming and outgoing waves.

While it is possible to measure a selection of these parameters, it is nearly impossible to densely sample the whole fourteen dimensional function. Instead, one uses material representations that disregard some of these dependencies, usually those that are only needed for a few special effects, like the time dependency for phosphorescence, the wavelength dependency for fluorescence and the polarization. If one also discards the information where the radiation leaves the material, and thus information about subsurface scattering, one gets a material representation called BTF (Bidirectional Texture Function). Introduced by Dana et al. [1], this is a function over the surface points x , the incoming (θ_i, ϕ_i) and the outgoing (θ_r, ϕ_r) light direction:

$$\rho_{BTF}(x, \theta_i, \phi_i, \theta_r, \phi_r) \quad (2)$$

One should note, however, that while the dependencies mentioned above are not parameterized, their visual effects, such as subsurface scattering or fluorescence, are still present in the BTF data, as well as effects resulting from the material's spatial properties such as interreflection and self-shadowing. Another simplification made is the discretization of all wavelength information to tristimulus values, usually representing the colors red, green and blue. This decision was not made without reason, since it corresponds to the way the human eye reacts to visible light, which is electromagnetic radiation with wavelengths between about 380 nm and 800 nm. In the eye, the spectrum of the incoming light is discretized to three values by the cones, one of two kinds of receptor cells on the retina. Thus, three values are enough to represent all color impressions a human observer can perceive. This behavior is recreated by convoluting the spectral information, given

*lyssi@cs.uni-bonn.de

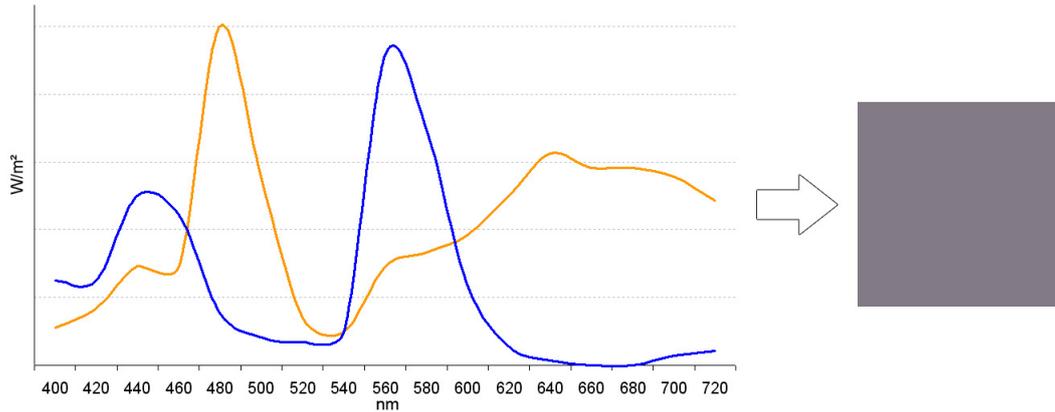


Figure 1: Two different spectral power distributions discretized to the same RGB values by convolution with the CIE 1931 RGB color matching functions

as a spectral power distribution $I(\lambda)$, with color matching functions, e.g. the CIE 1931 RGB color matching functions \bar{r} , \bar{g} and \bar{b} :

$$\begin{aligned} R &= \int_{\lambda} I(\lambda) \bar{r}(\lambda) d\lambda \\ G &= \int_{\lambda} I(\lambda) \bar{g}(\lambda) d\lambda \\ B &= \int_{\lambda} I(\lambda) \bar{b}(\lambda) d\lambda \end{aligned} \quad (3)$$

Tristimulus values are, however, not enough to simulate the interaction of light with a material. This is due to the fact that the reflection happens on a spectral level, i.e. the reflectance properties are wavelength-dependent. Unfortunately, there is no 1-to-1 relation between spectra and RGB values. Multiple (theoretically infinite) spectra can be represented by the same RGB-triple. This effect is called *metamerism*. Figure 1 shows an example of two different spectra with the same corresponding RGB values. The reflectance properties of a material can be given as a spectrum, where the values at each wavelength represent the fraction of reflected light at that wavelength. Such a spectrum is needed for all possible incoming and outgoing light directions and surface points.

Now suppose one has a spectrum representing a light source and two spectra representing the color of two different materials. A problem arises if the spectra of the materials are discretized to the same RGB values. No matter what illumination one might choose, the resulting color impression will always be the same for both materials. This is not an accurate representation of reality, where the differing spectra might result in different color impressions depending on the illumination. For an example, see figure 2. The first row shows the spectra and corresponding identical color impression of two different materials, the second row the spectrum of a light source. The last row shows the result of illuminating the materials with the lightsource, in the first two columns by convolution of the

spectra and subsequent convolution with the color matching functions, and in the third column by multiplication of the RGB values. The results are three different color impressions.

This is the reason why a tristimulus based material representation and thus the measurement of reflectance properties with devices like RGB cameras is insufficient in areas where a correct reproduction of a material's appearance, especially under varying illuminations, is necessary. Instead, a dense sampling of the spectrum is needed. This should be acquired with spectral band filters that have minimal overlap, so that the multiplication of the spectral bands of two materials can be understood as the multiplication of the two spectra.

The measurement of spectral BRDFs (Bidirectional Reflectance Functions) using a gonioreflectometer setup with spectrometers is already commonplace in some parts of the industry. The BRDF, however, is an even more simplified material representation, with the only parameters left being the incoming and outgoing light direction. The spatial parameter present in the BTF is being neglected. This means that, in contrast to the BTF, the BRDF cannot faithfully reproduce the reflectance behavior of spatially varying materials.

In this paper we therefore present a measurement setup for the acquisition of spectral BTFs, which is a modification of the setup described by Sattler et al. [7].

The rest of the paper is structured as follows: in chapter 2 we will examine previous work on the subject of BTF measurement and spectral rendering. Following that, we present our measurement setup in chapter 3, the calibration procedure we had to employ in chapter 4 and further information about our implementation and the measurement process in chapter 5. Some results we have achieved so far are show in chapter 6. Finally, we give a conclusion of our work in chapter 7.

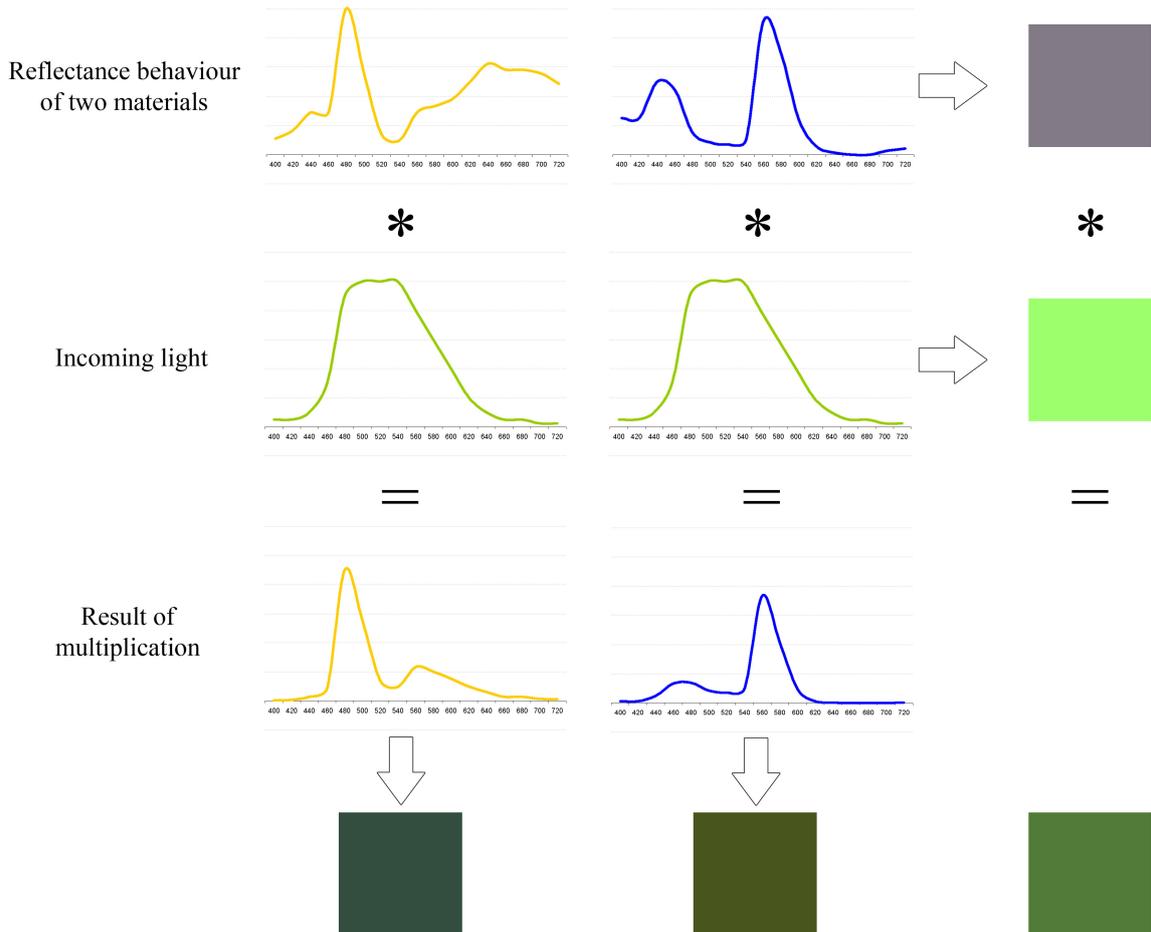


Figure 2: An example for the error produced by the usage of RGB values. Two materials with different spectra but the same corresponding RGB values are illuminated by a light source. The last row shows the resulting, differing color impressions.

2 Related Work

The previous work can be divided into the two areas of BTF measurement and spectral rendering.

2.1 BTF measurement

Bidirectional Texture Functions were introduced by Dana et al. [1]. These are six dimensional functions depending on a surface point as well as the incoming and outgoing light directions:

$$\rho_{BTF}(x, \theta_i, \phi_i, \theta_r, \phi_r) \quad (4)$$

This can be interpreted as a collection of 2D images, one for each possible view/light direction. They also introduced a first measurement setup, consisting of a robot holding a planar material sample, a lightsource and a videocamera. To take pictures from a multitude of view and lighting directions, the orientation of the material towards the lightsource was changed from picture to picture by movements of the robot and the positioning of the cam-

era on a limited number of different locations, leading to a whole of 205 different view/light directions. The position of the lightsource remained fixed. Using this technique, they measured 61 material samples and released the results in the form of the CURET Database.

A modified version of this setup was also used by Sattler et al. at the University of Bonn [7]. One adjustment was made by mounting the camera on a rail, which made it easier to automatically collect images from a high number of directions compared to manual positioning of the camera needed in Dana's setup. Measurements taken with this setup can be found in the BTF Database Bonn¹

Koudelka et al. [8] described another variation on this setup. In their case, the light source, a white LED, is mounted on a robot arm and moved while the camera remains fixed. Moving the robot arm provides for the different lighting directions, while moving the sample (which is mounted on a pan/tilt head) provides for the different viewing directions. Using this setup, $90 \times 120 = 10800$ images were taken. While the use of a video camera leads

¹<http://btf.cs.uni-bonn.de/>

to relatively fast measurement time of under 10 hours, the resolution of the resulting images (480×360 pixels) is rather low. Additionally, some of the image cannot be used due to the robot arm obstructing the view.

Apart from these there are a few measurement setups which vary greatly from the original setup. Han and Perlin [6] were inspired by a kaleidoscope for their setup. They positioned a number of mirrors around the material. When looking down onto the material, this generates a hall-of-mirrors effect that can be used to measure a number of different viewing directions with only one photograph. While this allows for a fast measurement and also has the advantage that the hardware setup is relatively small and easily portable, it has the disadvantage that the spatial resolution of the images is rather low and gets worse the higher the angular resolution (depending on the slope of the mirrors) gets. Also, interreflections between the mirrors pose a problem.

Another measurement setup was described by Müller et al. in their STAR on the acquisition, synthesis and rendering of BTFs [10]. There, an array of 151 cameras was mounted on a hemispherical gantry surrounding the material sample. This allowed for a parallel acquisition of the images for 151 view directions. The flashes of the cameras were used as the light source, making for 151 light directions and resulting in $151^2 = 22801$ images taken all together. Due to the parallel acquisition the whole measurement process could be completed in under one hour. The disadvantage of this method is the high cost because of the large amount of cameras. This is especially true if specialized cameras are needed (as in our case, see section 3).

2.2 Spectral rendering

Despite all the work on RGB-BTF acquisition, there has been no previous work on the acquisition of spectral Bidirectional Texture Functions. There has, however, been significant work on the subject of spectral rendering, some of which is mentioned in the following.

Early research was done by Hall and Greenberg [5]. They developed an image synthesis software based on a raytracer which can perform all color calculations on an arbitrary spectral resolution. Transformation to displayable tristimulus values is done as the last step, after the spectral information for each image point has been calculated.

Another spectrally based rendering framework was developed by Sun et al. [14]. A compact representation of spectra is introduced, which is based on the decomposition of the spectrum into a smooth component and a number of peaks. They also give an overview over natural phenomena like interference, diffraction and fluorescence which can only be correctly simulated in a spectral rendering environment, and describe how they can be realized in their framework. Additionally, an error metric is introduced which can be used to compare images generated with a varying density of spectral sampling.

Many reflectance models have been proposed that focus on the simulation of a particular set of the aforementioned phenomena. Gondek et al. [4] synthesized wavelength based BRDFs based on an optics model which incorporates the light's phase. This allowed for the simulation of interference effects, which, for example, can be observed on thin film surfaces like soap bubbles or the feathers of peafowls. Wilkie et al. [15] presented an analytical BRDF model for diffuse fluorescent surfaces based on a layered multifacet model. Stam [12] and Sun et al. [13] introduced reflectance models that simulate diffraction effects, which can be observed on compact discs. Finally, an overview of the reasons for spectral rendering and the work done in this area can be found in the STAR on tone reproduction and physically based spectral rendering by Devlin et al. [3].

While the necessity of spectral rendering and spectral material representations is reiterated in all of these works, they do not elaborate on how the material representations, especially in the form of BTFs, could be measured.

3 Measurement setup

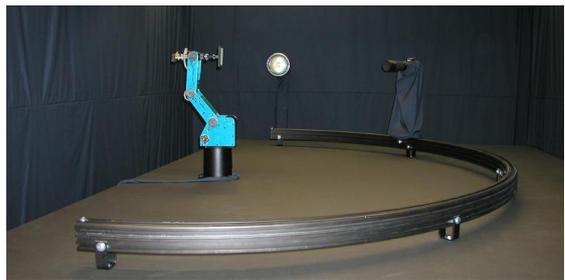


Figure 4: Measurement setup consisting of an HMI lamp, a CCD camera and a robot holding the material sample

Our goal is to measure the reflectance of a material for each surface point, incoming and outgoing light direction and wavelength of the outgoing light:

$$\rho_{specBTF}(x, \omega_i, \omega_r, \lambda_r) \quad (5)$$

The spatial dimension is sampled by taking two-dimensional photographs with a CCD camera. Incoming and outgoing light directions are sampled by changing the materials orientation in comparison to the lightsource and the camera. To sample the spectrum of the outgoing light, we install a spectral filter in front of the camera. The spectrum of the incoming light was not sampled, since adding this parameter would have had a large impact on the measurement time. This means that while the effects of fluorescence will be present in the measured data, exact information of the the material's fluorescence behaviour cannot be extracted. Also, filtering the light at the source would have been technically complicated. The filtered light is given off as thermal energy, and as the energy level directly at the light source is very high, this would have produced

high temperatures which quickly would have damaged the filter. Using a tunable laser as another way to parameterize the incoming spectrum is not applicable, since the coherent light waves produced by the laser would result in unwanted interference effects. Additionally, the small waveband of a laser would force a very dense and thus resource-demanding sampling.

The hardware setup used for the measurement is a modification of the setup used at University of Bonn introduced by Sattler et al.[7] and depicted in fig. 4, which itself was based on the original setup by Dana et al. [1]. It consists of an Intelitek SCORBOT-ER 4u robot arm holding a material sample of 10 x 10 cm, a combination of a camera and a spectral filter mounted on a rail and an 575 Watt HMI (Hydrargyrum Medium Arc Length Iodide) lamp. A lamp with this power is needed because due to the filter, only a very small fraction of the light will reach the camera when measuring each wavelength band. The camera is a Photometrics CoolSnap 4k, which has a 4-megapixels resolution, a 12 bit color depth and is sensitive to electromagnetic radiation from 350 nm to 1000 nm. With it, we reach a target resolution of about 800×800 pixels. Since, as mentioned above, the incoming energy level is low due to the filter, it is important that we have chosen a cooled camera. This leads to a reduction of thermal noise, and thus to an improvement of the signal-to-noise ratio. The spectral filter is a CRi VariSpec, which can be tuned to a wavelength between 400 nm and 720 nm and filters all electromagnetic radiation outside of a 10 nm band surrounding that wavelength. Since the filter has to be tuned multiple times during the measurement of one light/view direction, depending on the density of the spectral sampling, this has to happen very fast in order keep the measurement time as short as possible. Tuning to a new wavelength takes about 50 ms with our filter.

All hardware elements are controlled by a self written central computer program, which employs the SDKs provided by the hardware manufacturer. Thus, the measurement process is fully automatized, with the program responsible for the movement of the robot and the camera, the tuning of the filter and the activation of the camera. The computer we used had an Intel 2.67GHz Core2 Quad CPU, 2 GB of RAM and was running Windows XP. Using this setup, images from a combination of 81 view and 81 light directions and are taken.

4 Calibration and Post-Processing

Unfortunately, the raw images taken by the camera cannot be used to generate an accurate representation of the reflectance properties of the material. The first problem lies in the nonlinearity of the camera's response to incoming radiance, which is what we want to measure. Second, the pictures include the influence of the spectral distribution of the HMI lamp as well as the influence of wavelength-dependent variations in the transmissibility of lens and fil-

ter and the sensitivity of the CCD.

During the measurement, the camera is subjected to the exposure X , defined as the product between the irradiance E and the exposure time Δt :

$$X = E\Delta t \quad (6)$$

From this exposure, the pixel values of the digital image are obtained. Unfortunately the mapping between exposure and pixel value by the response function $f(X)$ is non-linear. To solve this problem, the camera's response function has to be determined. We used the algorithm from Debevec et al.[2] to do this. Several pictures of the same static scene with different exposure times are used as the input. The dependency between pixel value, response function, irradiance and exposure time is given by the film reciprocity equation $Z_{ij} = f(E_i\Delta t_j)$, with Z_{ij} being the value of pixel i in image j , f the response function, E_i the irradiance at pixel i and Δt_j the exposure time of image j . While the values Z_{ij} and the exposure times Δt_j are known, the response function f and the irradiance values E_i are unknown. They are found by minimizing a quadratic objective function derived from the film reciprocity equation. Debevec et al. include a Matlab implementation that does this using the singular value decomposition method.

To eliminate the influence of the light and the properties of the camera system, the data had to be modified as if taken with a lamp with a constant spectrum, a lens and filter with a constant transmissibility across the spectrum and a camera with a constant sensitivity. Thus, factors had to be calculated by which to multiply the pixel values at each wavelength. To get these factors, we first used a spectrometer (X-Rite i1) to measure the reflectance of the whitefield on a GretagMacbeth color chart. Then a frontal picture was taken of the whitefield using our measurement setup. The difference between the measurement with the spectrometer and our measurement setup was due to the influences mentioned above, and the correction factors could now be calculated.

This, however, revealed a different problem. For wavelengths shorter than 430 nm, we mainly measured thermal noise. This was due to the extremely low transmissibility of the filter at these wavelengths, which can also be observed when looking at the correction factors (see figure 5). Thus, the results for this spectral range were useless. We decided on the measurement of 30 wavelength bands between 430 nm and 720 nm in 10 nm steps and disregarded all wavelengths out of this range. The problem may be circumvented by using a different filter or a lamp with a higher radiation at short wavelengths. Unfortunately this was not possible during this work.

Another challenge were the long measurement times (see chapter 5) compared to the lifetime of our HMI lamp. This means we had to anticipate changes in the illumination like small drifts in the spectrum or dimming over the course of a measurement. To cope with this, a diffuse white border was placed on the sample holder around the

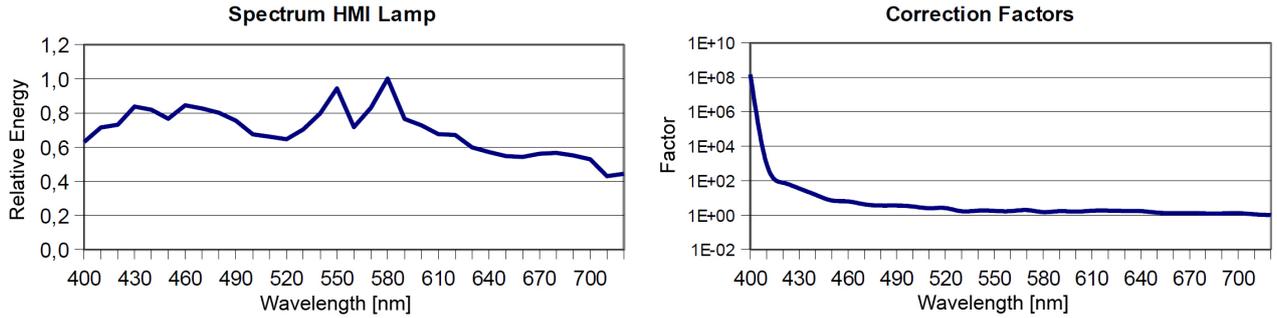


Figure 5: The spectral power distribution of our HMI lamp and the correction factors for the elimination of influences by our illumination and by the camera system’s properties on our measurement. Notice the logarithmic scale of the y-axis in the second graph and the high factors for wavelengths ≤ 430 nm.

sample. From the changes in its spectrum, correction factors could be calculated and applied to each image.

After the acquisition and calibration of the images, they had to be registered and rectified. To do this each image has to be projected on the frontal plane. This is also done with the help of the white border. Its corners can automatically be detected and mapped onto their corresponding positions in the frontal view. The mapping is then used for the projection of the whole image. Finally, the region of interest, i.e. the region of the material sample, has to be cut out and saved. These images have a spatial resolution of about 800×800 pixels. They are the final output of the measurement process and are ready for further use, e.g. for rendering.

5 Implementation and Measurement

Our measurement consists of the acquisition of 6561×30 12 bit greyscale images, since we measure 6561 view/light directions and 30 wavelength bands. Each image has a resolution of 2048×2048 pixels. During our tests we used an exposure time of 20 miliseconds per image.

After taking the photographs, the data has to be transferred to the host computer, calibrated, and finally be saved. For the last point, we decided to use ILM’s OpenEXR format [9], since it allows for an arbitrary amount of channels. This means we could collect and save all 30 images for one view/light directions in a single image file. Each channel can be given a name, which we used to also save the wavelength band each channels represents. Since a sequential approach to these subtasks would lead to a measurement time of about 100 seconds for one view/light direction, and about seven and a half days for all 6561 directions, this was a problem we had to address. To partially solve it, the process was parallelized using a quadcore CPU. Acquisition/readout, calibration and saving the data were each given their own thread, so that three threads were running in parallel. A visualization of this scheme can be seen in figure 6. This resulted in a reduced measurement time of about 45 seconds per view/light di-

rection and three and a half days for all 6561 directions, which is more than twice the speed of the original implementation. The calibration and saving threads were the most time consuming: acquisition and readout of the data took about 19 seconds per 30 channels, while calibration and saving took about 40 to 42 seconds. Additionally, after each view/light direction there is a small interruption of a few seconds in the measurement process in order for the camera and robot arm to move to their new position. In the future, CPUs with more than four cores could be used to start additional calibration and saving threads. This would further reduce the measurement time until the acquisition and readout become the bottleneck. Our program was designed to easily enable this extension.

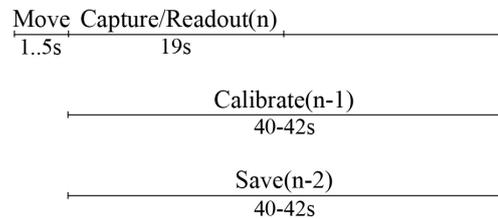


Figure 6: Visualization of the parallelization. After robot arm and camera have moved to their new position n , the acquisition of the images for this position ist started as well as the calibration of the images gathered at position $n-1$ and the saving of the images from position $n-2$.

A very large amount of data is created during the measurement. The raw data for 6561 images of the given size and number of channels, using the 16 bit half datatype of the OpenEXR format, has a size of about 1.65 terabyte. Since a lossless compression is supplied by the OpenEXR library, the saved images take up a size of about 985 gigabytes.

Since this is still too large an amount of data to handle, specialized compression methods have to be employed. Various compression methods for BTfs already exist. So far we have modified the implementation of the Per-View Factorization introduced by Sattler et al. [11], since it can

easily be generalized to multi-channel data. This compression method performs a PCA (principal component analysis) for each of the j viewing directions on vectors containing the image data for that particular direction. Each vector contains the image data for one particular light direction. In their original form, these vectors are of the form

$$X_{ij} = (r_{1,1}, g_{1,1}, b_{1,1}, \dots, r_{h,w}, g_{h,w}, b_{h,w}) \quad (7)$$

where i is the number of the light direction, j is the number of the view direction and h and w are height and width of the image. The only change is that we now place the pixel values of thirty channels instead of the three RGB channels into the vectors:

$$X_{ij} = (c_{1,1}^1, c_{1,1}^2, \dots, c_{1,1}^{30}, \dots, c_{h,w}^1, c_{h,w}^2, \dots, c_{h,w}^{30}) \quad (8)$$

The rest of the compression algorithm remains unchanged. The decompression code can be altered accordingly.

6 Results

To display the acquired spectral images they have to be converted to RGB images first. This is done by convoluting with the CIE 1931 color matching functions (see eq. 3) to retrieve the R, G, and B values for each pixel. All spectral images presented in the following have been converted this way. Samples for the greyscale images acquired and the resulting spectral image can be seen in figure 10. The difference between color calculations on a spectral scale and with RGB values is shown in figure 8. For these images, we simulated an illumination using the spectra of the CIE Standard Illuminants A, D65, FL4 and FL12 as well as a self measured spectrum of a flashlight's LED (see figure 7). Calculations were done once on a purely spectral scale and once only with RGB values. The sample used was a self made color chart measured with our setup. We've highlighted some of the fields where the difference can most easily be seen. To show that not only these highlighted fields are different, we computed an RGB difference image for the FL12 illuminant which can be seen in figure 9.

7 Conclusions

We have presented a goniospectrometer-like setup for the measurement of spectral BTFs and described the necessary steps to acquire correctly calibrated data. These spectral BTFs can be used in a spectral rendering environment to create images with a more accurate color reproduction compared to color calculations done with RGB values. There are, however, several areas we plan to work on in the future.

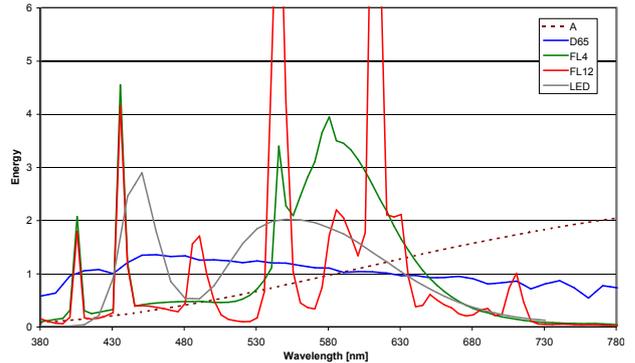


Figure 7: Illumination spectra used to create the results in figure 8.

The measurement time of more than three days still poses a problem, especially compared to the fast measurement times of just a few hours that can be achieved using camera arrays [10]. One way to do this would be to further exploit parallelization techniques as mention in chapter 5. Also, we are planing to put additional work into the adaption and further development of compression algorithms to cope with the large amount of data in our spectral images.

References

- [1] Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, 1999.
- [2] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 369–378, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [3] K. Devlin, A. Chalmers, Alexander Wilkie, and Werner Purgathofer. Star report on tone reproduction and physically based spectral rendering. In *Eurographics 2002*. Eurographics Association, 2002.
- [4] Jay S. Gondek, Gary W. Meyer, and Jonathan G. Newman. Wavelength dependent reflectance functions. In *In Computer Graphics Proceedings, Annual Conference Series*, pages 213–220. Press, 1994.
- [5] Roy Hall and Donald Greenberg. A testbed for realistic image synthesis. *IEEE Comput. Graph. Appl.*, 3(8):10–20, 1983.
- [6] Jefferson Y. Han and Ken Perlin. Measuring bidirectional texture reflectance with a kaleidoscope. *ACM Trans. Graph.*, 22(3):741–748, 2003.

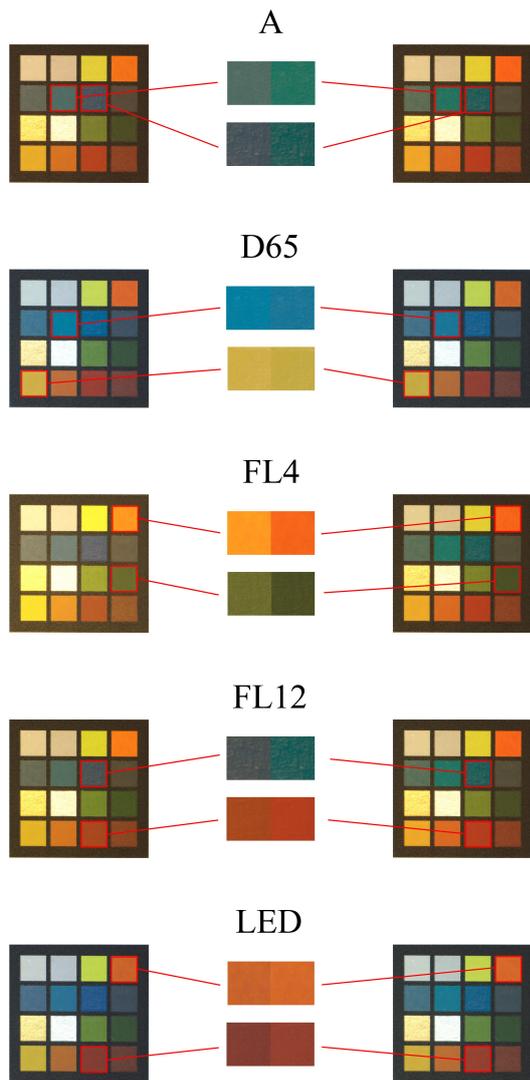


Figure 8: An example for the different results when doing color calculations on a spectral scale and with RGB values. Illumination with CIE Illuminants A, D65, FL4 and FL12 and with the LED of a flashlight were simulated. The left column shows the results of the spectral calculations. For the results in the right column, the images and the light spectra were first converted to RGB values as described in section 6. Areas where the differences can most easily be seen are highlighted.

- [7] M. Hauth, O. Etmuss, B. Eberhardt, R. Klein, R. Sarlette, M. Sattler, K. Daubert, and J. Kautz. Cloth animation and rendering, September 2002. Tutorial.
- [8] Melissa L. Koudelka and Sebastian Magda. Acquisition, compression, and synthesis of bidirectional texture functions. In *In Proc. 3rd Int. Workshop on Texture Analysis and Synthesis (Oct)*, pages 59–64, 2003.
- [9] Industrial Light & Magic. Openexr library.

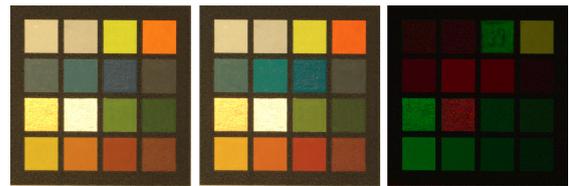


Figure 9: Our color chart, illuminated with CIE Illuminant FL12 as shown in figure 8. In the left picture shows the result of spectral calculations, the middle picture is the result of RGB calculations. The right picture is an RGB difference image between the first two images. It can be seen that deviations can be found across the whole image.

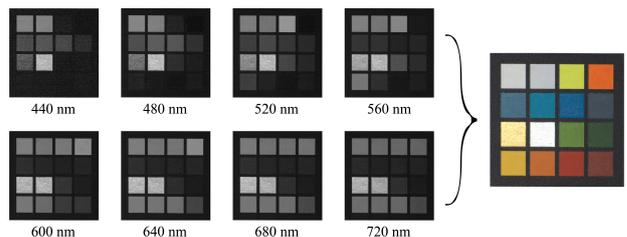


Figure 10: An example of the greyscale images taken for each wavelength band. Eight of the 30 images making up the spectral image on the right side are shown.

<http://www.openexr.com>.

- [10] G. Müller, J. Meseth, M. Sattler, R. Sarlette, and R. Klein. Acquisition, synthesis and rendering of bidirectional texture functions. *Computer Graphics Forum*, 24(1):83–109, March 2005.
- [11] M. Sattler, R. Sarlette, and R. Klein. Efficient and realistic visualization of cloth. In *Eurographics Symposium on Rendering 2003*, June 2003.
- [12] Jos Stam. Diffraction shaders. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 101–110, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [13] Yinlong Sun, F. David Fracchia, Mark S. Drew, and Thomas W. Calvert. Rendering iridescent colors of optical disks. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*, pages 341–352, London, UK, 2000. Springer-Verlag.
- [14] Yinlong Sun, F. David Fracchia, Mark S. Drew, and Thomas W. Calvert. A spectrally based framework for realistic image synthesis. *The Visual Computer*, 17(7):429–444, 2001.
- [15] Alexander Wilkie, Andrea Weidlich, Caroline Larboulette, and Werner Purgathofer. A reflectance model for diffuse fluorescent surfaces. In *Proceedings of Graphite 2006*, pages 321–328, November 2006.