DOI: 10.1111/j.1467-8659.2010.01730.x
Eurographics Symposium on Rendering 2010
Jason Lawrence and Marc Stamminger
(Guest Editors)
Volume 29 (2010), Number 4

Spectralization: Reconstructing spectra from sparse data

Martin Rump and Reinhard Klein

Institute for Computer Science II, University of Bonn †

Abstract

Traditional RGB reflectance and light data suffers from the problem of metamerism and is not suitable for rendering purposes where exact color reproduction under many different lighting conditions is needed. Nowadays many setups for cheap and fast acquisition of RGB or similar trichromatic datasets are available. In contrast to this, multi- or even hyper-spectral measurements require costly hardware and have severe limitations in many cases. In this paper, we present an approach to combine efficiently captured RGB data with spectral data that can be captured with small additional effort for example by scanning a single line of an image using a spectral line-scanner. Our algorithm can infer spectral reflectances and illumination from such sparse spectral and dense RGB data. Unlike other approaches, our method reaches acceptable perceptual errors with only three channels for the dense data and thus enables further use of highly efficient RGB capture systems. This way, we are able to provide an easier and cheaper way to capture spectral textures, BRDFs and environment maps for the use in spectral rendering systems.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Digitizing and scanning

1. Introduction

Photo-realistic rendering is of great importance in many parts of the industry for the use in virtual prototyping, virtual showrooms or advertising. Here, spectral rendering systems have become more and more commonplace (e.g. [Ran], [Nex]) especially in certain fields like architecture where color correct rendering is of special interest. Moreover, the quality of rendered images has been substantially improved by measuring optical material properties and lighting conditions. But until now, these datasets are usually only available in RGB or similar trichromatic representations limiting the use in spectral rendering systems.

Unfortunately, spectral light and reflectance capture is a very tedious task due to several physical limitations. Of course, capturing the spectra of simple light sources or of homogeneous and diffuse surfaces is not difficult any more because even handheld devices exist for that purpose. However, as soon as complex lighting conditions are required, environment maps or similar representations have to be used. Likewise, the capture of multi-spectral textures, Bidirectional Reflectance Distribution Functions (BRDFs) or Bidirectional Texture Functions (BTFs) is needed for more complex materials. BRDFs can be captured using gonioreflectometer setups with a spectrometer as measurement device. Unfortunately, these setups are very slow compared to image based measurement methods like [MWL*99]. All image based methods may be directly transferred to the spectral domain if multi-spectral imaging devices are used. These are mainly based on monochrome cameras combined with a mechanically or electronically tuneable bandpass filter (e.g. [HBS00], [HHA*10], [KSKL10]). The main problem with this approach is the very low amount of light passing the bandpass filters leading to bad signal-to-noise ratios and long exposure times. This severely limits the use of this technique, because only static scenes can be captured and for example outdoor scenes or scenes of rooms with windows cannot be measured due to the movement of the sun.

Textures may be captured using either the camera-filter combination described above or using a line-scanning device which maps the spectrum to one of the spatial axes of a CCD sensor (e.g. [AFOR04]). This requires linear movement of the scanner unit or of the target which induces additional effort and costs. Lighting conditions can be acquired

† e-mail: {rump, rk}@cs.uni-bonn.de

using a rotating line-scan camera with a fish-eye lens which also excludes the application to dynamic scenes. Nevertheless, line-scan cameras provide in general a much higher efficiency than bandpass filter based systems since all spectral bands are captured at once and no time for the tuning of the filter is wasted.

In this paper, we present a method to infer reflectance and illumination spectra from dense RGB data and sparse spectral data. In contrast to previous methods that reconstruct spectra from few channels, we do not try to construct a low dimensional and/or smooth basis for the imaged spectra. Instead, we are able to deal with a very high number of different spectral shapes at once. This is of special interest, when environment maps are captured that contain multiple light sources with different spectra. Our technique is inspired by colorization, where a grayscale image is assigned colors by propagating color information under the assumption that similar connected gray values have similar colors. We assume that similar RGB values behave similar in spectral space and propagate spectra among images using an optimization algorithm. Similarity can be defined in several ways in our case, ranging from simple similarity of RGB values to similarity in RGB and spatial position or even more sophisticated methods from texture synthesis. Using our technique, it is possible to capture multi-spectral textures using an RGB image and one spectral scanned line or to capture BRDFs requiring only a very small amount of directions with multi-spectral sampling. This significantly reduces the effort to capture multi-spectral data for use in spectral rendering system since existing and efficient tools for RGB acquisition may be used further on and only a minimal amount of additional effort must be spent on multi-spectral acquisition.

2. Nomenclature

In this section we introduce necessary symbols and nomenclature to allow for a compact and understandable notation of previous work and our own method. We denote the RGB values of pixel i with R_i and those of the unknown spectral image with S_i where the number of spectral bands in S is called k. Both, R and S must be linear in energy. In this case, the RGB image is generated by applying three color filters to the respective spectra. These filters can be aligned in a $3 \times k$ matrix C so that $R = CS$. Furthermore, our method requires that for a set of pixels P the spectra are known. These will be called X having the size $|P| \times k$. Throughout the paper, C^T denotes the transposed matrix and the $k \times 3$ matrix $C^+ = C^T \left(CC^T\right)^{-1}$ the Moore-Penrose pseudo inverse of C. C^0 is the $k \times (k-3)$ matrix spanning the null space of C. C^+ and C^0 are of interest here, since $\left\{C^+R + C^0\tau \mid \tau \in \mathbb{R}^{k-3}\right\}$ is the space of all metamers leading to the RGB value R using the color filters C.

3. Previous Work

Spectral reconstruction By now, several methods have been proposed to infer spectra from RGB or similar trichromatic data. All of these methods are primarily intended for reflectances and not for lighting data. Most of them rely on the fact that reflectance spectra tend to be very smooth because of the underlying physical processes in the molecules generating the absorption.

Imai and Berns [IB99] proposed a method based on the pseudo-inverse $Q = C^+$ of the filter matrix C and reconstruct using $S = QR$. This technique does not require any multi-spectral data, but the results are in many cases unsatisfactory.

Another more heuristic approach from Smits [Smi99] reconstructs the spectrum using 7 basis spectra corresponding to white, red, green, blue, cyan, yellow and magenta. These basis spectra are generated by finding one valid and smooth metamer of the respective RGB values R of the seven basis colors in the space $C^+R + C^0\tau$ (see Section 2). The combination of the basis spectra to reproduce the pixels of an image is then done in a heuristic way. This makes the final algorithm very fast and thus allows for direct integration into the shading process and it does furthermore not require prior knowledge of multi-spectral data X. Unfortunately, it does not guarantee the spectra to have the right RGB equivalent and it will not perform well on non-smooth spectra.

The method from Hardeberg et al. [HSB*99] extends the pseudo-inverse method from Imai by incorporating prior knowledge of the imaged spectra into the determination of the inverse filter matrix: $Q = XX^TC^T\left(CXX^TC^T\right)^{-1}$. This way the reconstruction of non-smooth spectra is possible as long as all the imaged spectra are similar enough to each other that the three basis spectra in Q are sufficient. A further approach was proposed by Imai et al. [IBC98]. They combined a high-resolution RGB image with a low-resolution spectral image and upsampled the multi-spectral image having the RGB data as a constraint. However, this method still requires a very large number of spatially distributed spectral samples X whereas our method is in many cases able to reconstruct spectra from just a single line.

In recent work, Hullin et al. [HHA*10] presented a method to measure bispectral reflectance data. They capture a certain set of directions with full spectral resolution of both incoming and outgoing light and build a PCA basis from this dense data. The whole bispectral BRDF is then reconstructed using this basis where the weighting of the basis functions per direction pair ω_i, ω_o is determined using a sparse set of samples and a least-squares optimization. In principle, this method is very similar to the one from Hardeberg et al. [HSB*99] except for the fact that Hardeberg et al. propose to measure in the constructed basis directly by choosing an appropriate set of filters whereas the approach from Hullin et al. performs the least squares reconstruction for every sample.

Colorization and Edit-Propagation In principle, infering spectra from RGB values is related to the colorization of grayscale images. In this field, much work has concentrated on color transfer from an example image to the target image (e.g. [ICOL05]) which is not applicable in our setting since no example image is present. Other methods let the user scribble regions of an image and define a color for that region which is then propagated (e.g. [LLW04], [LWCO*07]). The approach from Levin et al. [LLW04] is quite general. They formulate colorization as an optimization problem where the energy function contains a data term fixing the grayscale values as well as a smoothness term propagating known RGB values to other neighboring pixels having similar grayscale values. Additionally, our method is related to the AppWand editing technique from Pellacini et al. [PL07] where image and material edits known at a small subset of pixels are propagated among the other pixels by assuming that similar pixels should have similar edit operations applied.

4. Method

Our method reconstructs spectra from RGB data R known at all pixels A and a small set of pixels P with known spectrum X_i. In contrast to other methods described in Section 3 we neither rely on a smoothness assumption nor try to find a low dimensional basis Q for the reconstruction. This enables us to use as many different spectra as X provides as well as mixtures of these spectra. Our key assumption is that we can identify similar pixels solely using the dense RGB image. This may be done on a simple per pixel basis but also more sophisticated distance metrics based on local pixel neighborhoods can be easily integrated into the algorithm.

4.1. Energy function

We formulate spectralization as a minimization problem with the following energy functional:

$$\begin{aligned} E(S) &= \alpha \sum_{i \in P} \|S_i - X_i\|^2 \\ &+ \beta \sum_{i \in A} \|CS_i - R_i\|^2 \\ &+ \sum_{i \in A} \sum_{n \in N_i} \gamma_{i,n} \|S_i - S_n\|^2 \end{aligned} \tag{1}$$

The first term fixes the known spectra at the positions in P, the second term fixes the RGB projections CS of the spectral image and the third term leads to the propagation of the spectra among similar RGB values. N_i denotes a set of size g of neighboring pixels for the pixel i and α, β and $\gamma_{i,n}$ are weighting terms. $\gamma_{i,n}$ consists of a general weight γ for the third term as well as a specific weight for the neighbors i and n based on their distance.

4.2. Neighborhoods

One crucial subproblem of our approach is the selection of a proper way to choose the neighborhoods N_i as well as the weights $\gamma_{i,n}$. As mentioned above, the simplest method would be to choose a fixed number of closest pixels based on plain RGB color distance. This, however, may not work even in very simple cases where the pixels are only different in brightness, which is often the case e.g. for BRDFs or in non-texture images where certain parts of an object are shadowed. For this reason we propose a first simple extension by computing the pixel intensities I for all pixels as the mean of the R, G and B values. We then calculate $R'_i = \frac{R_i}{I_i}$ and $X'_i = \frac{X_i}{I_i}$, run all further steps using R' and X' computing S' and finally multiply with I again to obtain S.

For the weights $\gamma_{i,n}$ we choose a gaussian with a fixed variance σ:

$$\gamma_{i,n} = \gamma \exp\left(\frac{-\|R_i - R_n\|^2}{2\sigma^2}\right) \tag{2}$$

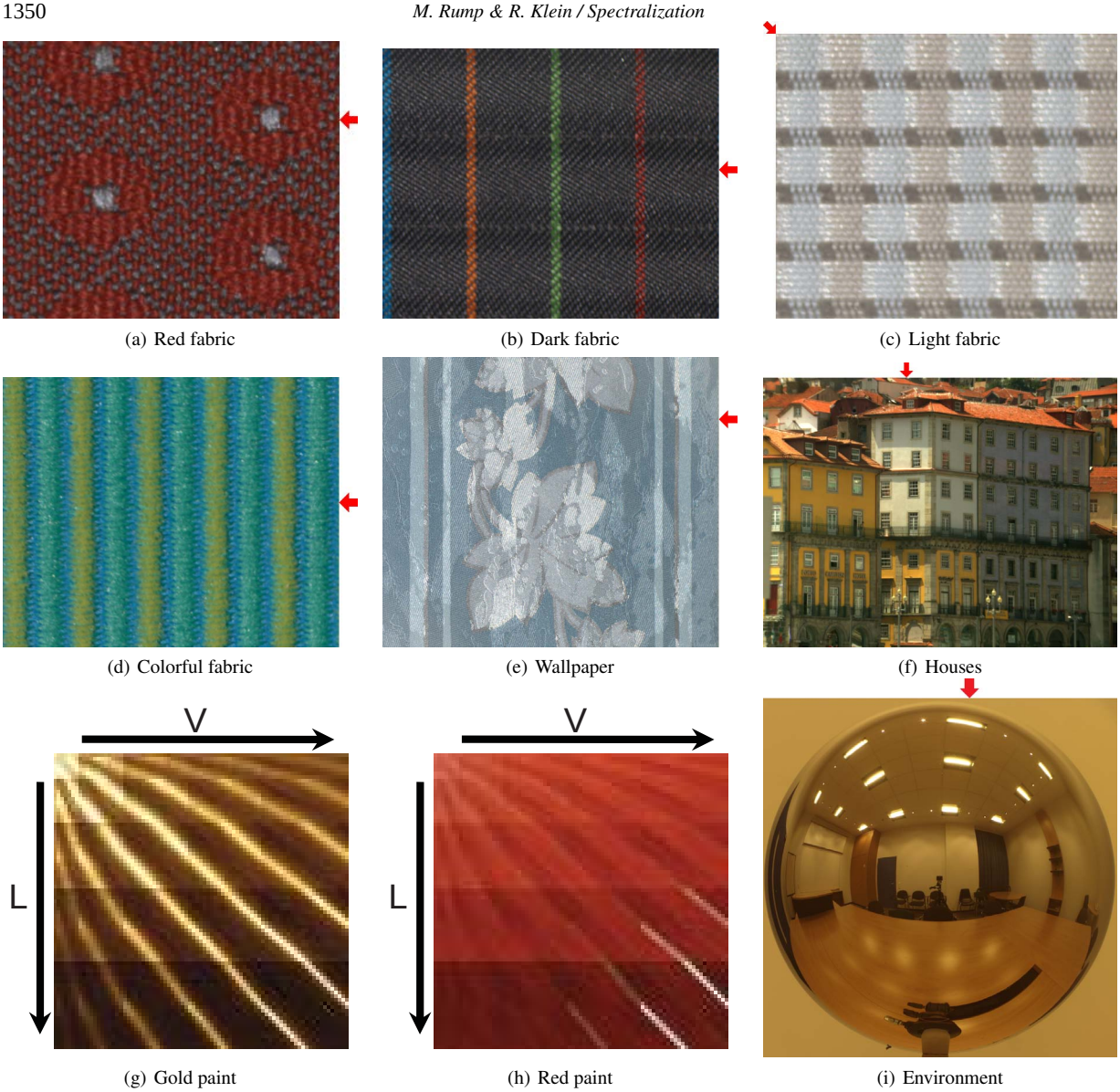
4.3. Minimization

Since the energy function (see Equation 1) is quadratic, it can be transformed into a set of linear equations $BS = b$ with a sparse matrix B. In small cases the matrix B may be built explicitly and the system can be solved for example using a QR factorization or a similar matrix decomposition. However, in most cases this is not possible since even a sparse representation of B is several gigabytes in size and thus not suitable for the use in a matrix factorization algorithm. Therefore, we propose to use an iterative algorithm that does not require to build B explicitly. For that purpose, we build the normal equations $\left(B^T B\right) S = B^T b$ with $B^T B$ being symmetric and positive definite. This makes it possible to minimize the function using a conjugate gradient solver which requires only a function that multiplies $B^T B$ with a vector y. This function can we written without explicitly constructing B and therefore the memory consumption of the method is reduced considerably. In all test cases, the error did not improve significantly after 250 iterations which does not depend on the size of the image in particular. Therefore, we use this as a hard limit without respect to the image size. Since the multiplication with $B^T B$ is linear in image size this leads to the total runtime of the algorithm being linear in the image size as well. This is in stark contrast to matrix factorization techniques which require quadratic runtime or even more with respect to matrix size.

As a starting value for the iterations we can use a black image or any of the methods described in Section 3. If a better initialization is given, the conjugate gradient solver will converge faster. Therefore, all results in Section 5 have been calculated using the Hardeberg method [HSB*99] (see Section 3) as an initialization since it performed best among all other methods.

4.4. Parameters

The energy function can be controlled by the weights α, β and γ as well as the variance σ for the weight $\gamma_{i,n}$ of pairwise



(a) Red fabric (b) Dark fabric (c) Light fabric

(d) Colorful fabric (e) Wallpaper (f) Houses

(g) Gold paint (h) Red paint (i) Environment

Figure 1: *Test textures, BRDFs and images. The BRDFs are plotted as image where each row corresponds to a light direction and every column to a view direction. The small arrows on the side of the images indicate the position and direction where the spectral line was sampled.*

neighboring pixels. Furthermore, the number g of neighbors is an important parameter. Fortunately, all of the parameters can be chosen automatically and so the amount of required user interaction is reduced to a minimum.

We propose to weight the three terms of the energy function equally with regard to the real number of summands:

$$\begin{aligned} \alpha &= \frac{k\,|P| + 3\,|A| + kg\,|A|}{k\,|P|} \\ \beta &= \frac{k\,|P| + 3\,|A| + kg\,|A|}{3\,|A|} \end{aligned} \tag{3}$$

$$\gamma = \frac{k\,|P| + 3\,|A| + kg\,|A|}{kg\,|A|}$$

Fortunately, the minimization behaves quite robust against changes in these parameters. We multiplied the β and γ with values between 0.01 and 50 and the resulting ΔE^* error did not change more than about 0.05 in all cases. For all results in Section 5 we used the values from Equation 3.

The number of neighbors g can be kept quite small. All our results were calculated using 10 neighbors per pixel and a further increase led to only slightly better errors. In all our test cases, the 10 neighbors $n \in N_i$ are that similar to i that

the choice of σ does not play a dominant role any more. We propose to use the maximum distance $d_{max} = \|R_i - R_n\|$ for all i and n and set the weight for d_{max} to 0.01:

$$\sigma = \sqrt{\frac{-d_{max}^2}{2\ln(0.01)}} \tag{4}$$

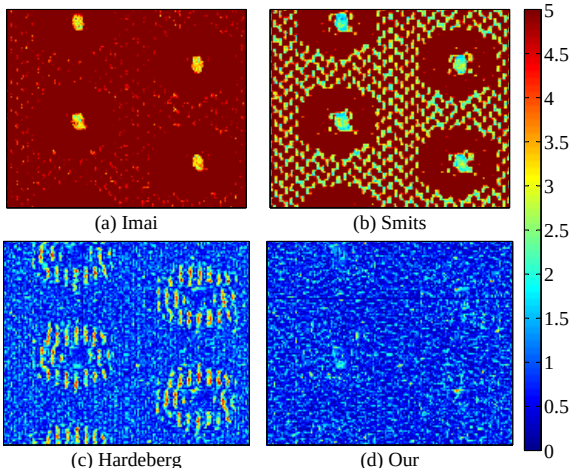


Figure 2: *Color coded ΔE^*_{94} error maps for the red fabric texture. It shows that the Hardeberg method fails to reconstruct all of the different threads whereas our method can deal with an arbitrary number of spectral shapes at once.*

4.5. Adaptive sampling

The neighborhood sets N_i define a graph with pixels as nodes and edges between pixel nodes i and j if $j \in N_i$. In cases where the graph breaks up into multiple components, whole components may not be connected to any known pixel P. But even if the graph is made of one single component it might be that single areas are only reachable over graph edges with very low weight $\gamma_{i,n}$. Usually, this means that the known pixels X do not contain enough information to reconstruct the spectra of the pixels in the area of low weight. Fortunately, we can detect regions of low weight by minimizing the energy functional (Equation 1) with $k = 1, C = 1, X_i = 1, \forall i \in P$ and $R_i = 0, \forall i \in A$. In the resulting image M lighter pixels correspond to regions that are well supported by the pixels P and darker pixels to the regions with low support. This is due to the fact that the first data term of Equation 1 will drag the well connected pixels towards one while the second data term pulls all values down towards zero. By simply finding the darkest point in M, it is possible to decide where to place the next sample point(s). This is especially useful for BRDF acquisition since it allows to specify a good set of directions to capture in beforehand based solely on the RGB data.

4.6. Practical considerations

Some practical problems arise from the registration of spectral and RGB data. First of all, both datasets have to be linearized and offsets have to be removed. In the case of BRDFs measured using a gonioreflectometer no spatial registration has to be done enabling for very easy integration of spectralization. But as soon as a spectrally sampled line should by used with an RGB image, one must consider some additional problems. We will not discuss the problem of finding the position of the spectral scan line in the RGB image here and assume that the position is already known. Additional problems arise of the different pixel size, pixel orientation and probably even focus and point spread function of the two systems. Because we assume that the RGB image R has higher resolution, we calculate the spectral image S on the resolution of R. Therefore, the second and third term of Equation 1 can stay unmodified. Problems of different pixel mapping can be described by linear combinations of pixels. That is, we calculate weights W_{ij} for a pixel i in the low resolution image and a set of pixels j in the high resolution image and write:

$$X_i = \frac{1}{\sum_l^{L_i} W_{i,j}} \sum_l^{L_i} W_{i j_{i,l}} S_{j_{i,l}} \tag{5}$$

Typically, the weights will come from some kind of kernel function, but arbitrary weighting schemes are possible. The first term of the energy functional will then change to:

$$E_1(S) = \alpha \sum_{i \in P} \left\| \frac{1}{\sum_l^{L_i} W_{ij}} \sum_l^{L_i} W_{i j_{il}} S_{j_{il}} - X_i \right\|^2 \tag{6}$$

Due to the linear combination this can still be integrated into the set of linear equations to solve the quadratic minimization problem.

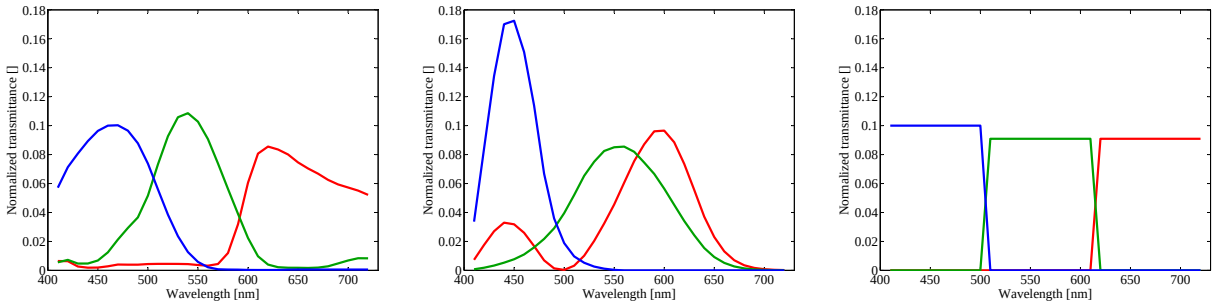
5. Results

We analyzed our method using seven different datasets namely 5 textures, 2 BRDFs including the cosine term and 2 non-texture images. Figure 1 shows an overview of the test datasets. Among the images, one is an environment map from [RSK10] of a room with complex lighting consisting of neon lamps and halogen spotlights. The other image is taken from David Foster's spectral image database [FNA04] and we will assume that it is an excerpt from an environment map since we do not have other environment data yet. The BRDFs are extracted from the Colorchecker dataset from [RSK10] and are equally sampled over the hemisphere at 81x81 directions. The other textures have been measured using a monochrome camera, a LCTF and a 1000W QTH lamp. For all datasets the complete multi-spectral ground truth was available so that the reconstruction quality can be reliably rated.

Our error analysis is twofold. On the one hand, we measure the difference on the whole spectra by computing RMSE errors and on the other hand a perceptual comparison is made in tristimulus space.

Dataset	ΔE^*_{94}					RMSE			
	RGB	Imai	Smits	Hardeberg	Our	Imai	Smits	Hardeberg	Our
Red fabric	10.56	9.83	10.45	0.92	0.67	0.104	0.088	0.010	0.007
Dark fabric	10.10	2.02	2.05	0.78	0.44	0.019	0.011	0.005	0.004
Light fabric	25.11	5.08	3.08	1.30	0.82	0.285	0.139	0.079	0.069
Colorful fabric	14.66	1.90	7.45	0.65	0.42	0.024	0.042	0.007	0.005
Wallpaper	14.66	2.72	2.25	1.40	1.08	0.070	0.031	0.027	0.026
Environment	16.94	9.75	13.13	3.80	0.77	0.322	0.344	0.073	0.043
Gold paint	13.30	3.44	6.18	0.84	0.75	0.095	0.055	0.029	0.040
Red paint	11.17	8.00	11.55	0.43	0.48	0.104	0.099	0.008	0.008
Houses	9.24	2.92	2.20	0.57	0.50	0.043	0.033	0.014	0.013

Table 1: *Comparison of reconstruction ΔE^*_{94} and RMS errors for the different datasets. Except for the BRDFs our method performed significantly better in RMS and perceptual errors than the previous ones. For simple BRDFs our method performs like the method from Hardeberg.*



(a) Real RGB camera, taken from manufacturer supplied data sheet

(b) CIE 1931 standard observer

(c) Artificial box filter

Figure 3: *The different color filter sets used in our experiments*

5.1. Perceptual comparison

Tristimulus color images are the most common target in rendering applications since most display devices are RGB based. Therefore, the correct reproduction of colors in rendered images using materials or light sources reconstructed with our technique can be rated this way. To make a perceptual evaluation of the accuracy of our method we use ΔE^*_{94} based on the perceptual linear L*a*b* color space.

For the materials we artificially illuminate the single pixels of the original and the reconstruction using a selection of 10 CIE standard illuminants (A, D65, D50, D75, FL1, FL4, FL8, FL12, HP1, HP3) which cover a wide range of commonly used light sources. We then transform the resulting spectra into the CIE XYZ color space, further on into L*a*b* using the respective light source as neutral and calculate the ΔE^*_{94}. Finally, the arithmetic mean of the single ΔE^* values of all light sources is used.

For lighting data, the scheme is reversed. We artificially illuminate 20 materials from a Gretag MacBeth color checker using the spectra of the single pixels / light directions which is quite similar to the calculation of the color rendering index (CRI). We assume that the observer is adapted to the mean of the respective environment map and thus use this as neutral during the conversion to L*a*b*.

Since the ΔE^* errors are very sensitive to acquisition noise and we do not want to overweight the noise of single pixels we downsample the two images in question by a factor of 2 with a simple box filter before calculating the ΔE^* on a per pixel basis.

5.2. Error analysis and comparison

We evaluated our method as well as the methods from Imai [IB99], Smits [Smi99] and Hardeberg [HSB*99] on all test datasets and calculated the perceptual errors as described above. The RMSE is calculated as:

$$E_{RMSE}(D,S) = \sqrt{\frac{\sum_{i\in A, j\in[1..k]} \left(D_{ij} - S_{ij}\right)^2}{k\,|A|}} \qquad (7)$$

having D as the groundtruth data.

For the image examples, we manually chose a line P to cover as much different materials as possible. As soon as whole materials are not sampled by the line, the reconstruction result will be inaccurate in those areas. This is also true for the method from Hardeberg et al. This process could be easily automated by clustering the scene in RGB space and choosing a line that intersects as much clusters as possible.

The BRDFs were sampled in the adaptive manner described in Section 4.5 starting with just one single direction

for X (light and view from the top) and ending up with 30 directions. We noticed, that for both examples the ΔE^* error drops below 1 using four directions or more. This means that using our technique the spectral acquisition of four instead of 6561 directions is sufficient to reach acceptable perceptual errors as long as dense RGB data is available. By using existing and efficient RGB BRDF measurement techniques this leads to a significant speedup for the acquisition of multi-spectral BRDFs.

The resulting ΔE^*_{94} and RMS errors for all datasets can be taken from Table 1. It shows that our technique clearly improves upon previous techniques especially for textures and images regarding both the perceptual tristimulus error as well as the spectral error. Our technique requires only the single line or four directions to reconstruct the multi-spectral images and BRDFs with acceptable perceptual errors. For simple BRDFs the similarity of spectra is that high, that the method from Hardeberg performs as well as our one. This will be different for materials showing interference effects and thus more colors in their BRDFs like pearlescent paints.

Figure 2 shows ΔE^* error maps for the red fabric dataset produced by the three previous techniques and our one. It shows that the method from Hardeberg performs similar to our one except on one of the threads of the fabric. This is because the fabric is made of very different kinds of threads and the Hardeberg method constructs a low dimensional basis Q for the reconstruction. If too many different spectral shapes are contained in the image (like in this case) this basis is no longer sufficient to cover all of them. Our method does not build such a basis and can therefore deal with very different spectral shapes in one image.

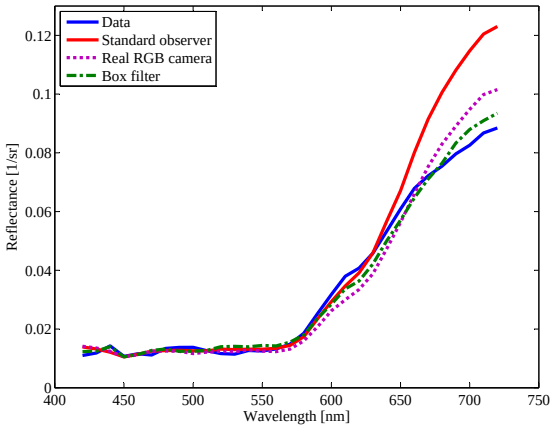


Figure 4: *Reconstruction result for one red thread of the "Red fabric" dataset using the different color filter sets from Figure 3. The low support of the human eye in the long wavelengths leads to larger differences here.*

5.3. Impact of color filter C

We wanted to analyze what the impact of the camera in use, or more specific the spectral shape of its colors filters, is. For this reason we tried the three different filter sets shown in Figure 3 regarding the reconstruction results. Figure 4 shows a reconstruction of the same pixel using these three filter sets. It shows that the smoothness of the filters does not play a dominant role for the reconstruction quality. Noticeable is only the difference in the long wavelengths bands for the imaginary camera with the sensitivity of the human eye. For light sources having much energy there (like Illuminant A) this yields much larger errors. However, the reconstruction of spectra from an imaginary camera with human eye sensitivity performs best regarding the average perceptual ΔE^* errors since many light sources (e.g. the fluorescent ones) have low energy in the very long wavelength bands and the perceptual weighting of the bands leads to a better approximation in the other bands. Unfortunately, such a camera does not exist to our knowledge and we computed all other results in this paper (especially Table 1) using the real RGB camera filters to show the use case with current cameras.

5.4. Adaptive sampling

We analyzed the quality of our adaptive sampling approach on the two BRDF datasets by comparing the adaptive sampling against a random sampling of the directions. Results can be taken from Figure 5. It shows that the adaptive technique converges faster than the random one and therefore the amount of directions for which spectral data has to be captured is minimized.

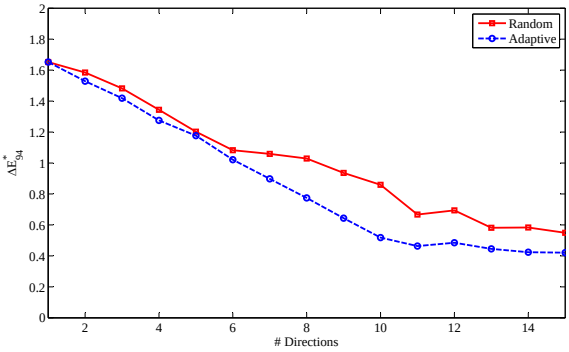
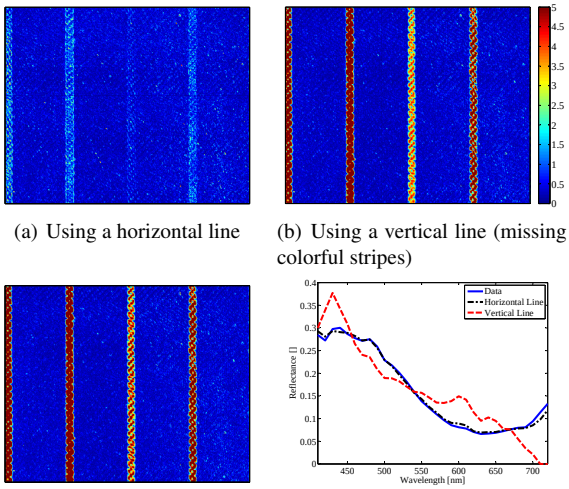


Figure 5: ΔE^*_{94} *errors for increasing number of directions used to reconstruct the gold BRDF. Directions were sampled randomly and with our adaptive method.*

5.5. Failure case

In general, there are two possibilities for failure of the proposed method. On the one hand, two or more objects in the scene might have materials that are metamers of each other. This can be handled by an additional spatial proximity term in the neighborhood search that will isolate the two objects. On the other hand, one of the objects in the scene that was not sampled by the spectral scanline might have a spectrum which is considerably different than all sampled

spectra. In this case the algorithm will create a smooth spectrum from the most similar object in RGB space. Figure 6 shows ΔE^* error maps for the dark fabric scene where the colorful stripes have (a) and have not been sampled (b, c). Our method behaves like the method from Hardeberg et al. here, since we initialize with that method and since nothing will be changed during the minimization of the energy functional since the pixels do not contribute much to the overall energy due to their low γ weights.



(a) Using a horizontal line (b) Using a vertical line (missing colorful stripes)

(c) Result with method from Hardeberg and vertical line (d) Reconstruction of the left-most blue thread

Figure 6: *Color coded ΔE^*_{94} error maps and spectra for the "dark fabric" dataset with different lines sampled.*

6. Conclusion

In this paper we presented a method to infer spectral images from dense RGB data and very sparse spectral data. Our technique simplifies multi-spectral measurements for computer graphics and enables to reuse existing, cheap and efficient RGB capture devices. Our method improves on previous work regarding both the perceptual tristimulus error as well as the RMS error on the spectra. Only for very simple examples like diffuse BRDFs the method from Hardeberg et al. [HSB*99] performs comparably.

In future work we plan to extend the method to Bidirectional Texture Functions, where effects like self-shadowing, masking and parallax make the ABRDFs much more complicated than BRDFs are. Here, neighborhoods must be searched in the higher dimensional space spanned by directions and spatial positions. Moreover, we want so examine more sophisticated neighborhood searches. In the field of texture synthesis, many techniques are known to search for similar pixels based on the local surroundings of the pixel in question. Additionally, techniques from colorization may be used, that depend on spatial proximity as well as on pixel value. This is of special interest for non-texture images with larger homogeneous objects.

7. Acknowledgment

This work was funded by the German Science Foundation (DFG) under research grant KL 1142/7-1.

References

[AFOR04] ANTONIOLI G., FERMI F., OLEARI C., REVERBERI R.: Spectrophotometric scanner for imaging of paintings and other works of art. In *Proceedings of CGIV 2004* (2004), pp. 219–224. 1

[FNA04] FOSTER D. H., NASCIMENTO S. M., AMANO K.: Information limits on neural identification of colored surfaces in natural scenes. *Visual Neuroscience 21*, 03 (2004), 331–336. 5

[HBS00] HARDEBERG J. Y., BRETTEL H., SCHMITT F.: Multispectral image capture using a tuneable filter. *Color Imaging: Device Independent Color, Color Hardcopy and Graphic Arts 3963 of SPIE Proceedings* (2000), 77–88. 1

[HHA*10] HULLIN M. B., HANIKA J., AJDIN B., SEIDEL H.-P., KAUTZ J., LENSCH H. P. A.: Acquisition and analysis of bispectral bidirectional reflectance and reradiation distribution functions. In *Proceedings of SIGGRAPH 2010* (2010), p. to appear. 1, 2

[HSB*99] HARDEBERG J. Y., SCHMITT F., BRETTEL H., CRETTEZ J.-P., MAITRE H.: Multispectral image acquisition and simulation of illuminant changes. In *Color Imaging: Vision and Technology* (1999), pp. 145–164. 2, 3, 6, 8

[IB99] IMAI F. H., BERNS R.: Spectral estimation using trichromatic digital cameras. In *Proceedings of the International Symposium on Multispectral Imaging and Color Reproduction* (1999), pp. 42–49. 2, 6

[IBC98] IMAI F. H., BERNS R. S., CARLSON C. F.: High-resolution multi-spectral image archives - a hybrid approach. In *Proceedings of IS&T and SID's 6th Color Imaging Conference: Color Science, Systems and Applications* (Scottsdale, Arizona, USA, 1998), pp. 224–227. 2

[ICOL05] IRONY R., COHEN-OR D., LISCHINSKI D.: Colorization by Example. In *Proceedings of EGSR* (2005), pp. 201–210. 3

[KSKL10] KIM D. B., SEO M. K., KIM K. Y., LEE K. H.: Acquisition and representation of pearlescent paints using an image-based goniospectrophotometer. *Optical Engineering 49*, 4 (2010), 043604. 1

[LLW04] LEVIN A., LISCHINSKI D., WEISS Y.: Colorization using optimization. *ACM Trans. Graph. 23*, 3 (2004), 689–694. 3

[LWCO*07] LUAN Q., WEN F., COHEN-OR D., LIANG L., XU Y.-Q., SHUM H.-Y.: Natural image colorization. In *Proceedings of EGSR 2007* (June 2007). 3

[MWL*99] MARSCHNER S., WESTIN S., LAFORTUNE E., TORRANCE K., GREENBERG D.: Image-based BRDF measurement including human skin. In *EGWR* (1999), pp. 131–144. 1

[Nex] NEXT LIMIT TECHNOLOGIES: Maxwell render^{TM}. http://www.maxwellrender.com/. 1

[PL07] PELLACINI F., LAWRENCE J.: Appwand: editing measured materials using appearance-driven optimization. *ACM Trans. Graph. 26*, 3 (2007), 54. 3

[Ran] RANDOMCONTROL: fryrender. http://www.randomcontrol.com/fryrender. 1

[RSK10] RUMP M., SARLETTE R., KLEIN R.: Groundtruth data for multispectral bidirectional texture functions. In *Proceedings of CGIV 2010* (2010), p. to appear. 5

[Smi99] SMITS B.: An rgb-to-spectrum conversion for reflectances. *J. Graph. Tools 4*, 4 (1999), 11–22. 2, 6