České vysoké učení technické v Praze
Fakulta elektrotechnická

Czech Technical University in Prague
Faculty of Electrical Engineering

Komprese prostorově rozšířené dvousměrové funkce odrazivosti povrchu pomocí vyhledávání a víceúrovňové vektorové kvantizace

Compression of BTF Data via Searching and Nested Vector Quantization

Ing. Vlastimil Havran, Ph.D.

Summary

The Bidirectional Texture Function (BTF) is becoming widely used for accurate representation of real-world material appearance. In this work a novel BTF compression model is proposed. The model resamples input BTF data into a parametrization, allowing decomposition of individual view and illumination dependent texels into a set of multidimensional conditional probability density functions. These functions are compressed in turn using a novel multi-level vector quantization algorithm. The result of this algorithm is a set of index and scale code-books for individual dimensions. BTF reconstruction from the model is then based on fast chained indexing into the nested stored code-books. In the proposed model, luminance and chromaticity are treated separately to achieve further compression. The proposed model achieves low distortion and compression ratios $1:233-1:2040$, depending on BTF sample variability. These results compare well with several other BTF compression methods with predefined compression ratios, usually smaller than $1:200$. We carried out a psychophysical experiment comparing our method with LPCA method. BTF synthesis from the model was implemented on a standard GPU, yielded interactive framerates. The proposed method allows the fast importance sampling required by eye-path tracing algorithms in image synthesis.

Souhrn

Prostorově rozšířená dvousměrová distribuční funkce odrazivosti povrchu (angl. *bidirectional texture function*), tedy funkce reprezentující prostorově závislou odrazivost fyzikálních povrchů, se začíná více a více používat pro reprezentaci vizuální podoby skutečných materiálů. Navržený model pro kompresi BTF dat nejprve převzorkuje data do nové parametrizace, která umožňuje rozklad funkce pro individuální úhly pozice kamery a osvětlení na funkce odpovídající marginální hustotě pravděpodobnosti. Tyto funkce jsou následně komprimovány pomocí víceúrovňové vektorové kvantizace, která používá vyhledávání nejpodobnější funkce z množiny již uložených kódových vektorů reprezentujících komprimovaná data. Z větší části výpočet komprimace BTF dat odpovídá vyhledávání. Datově jsou komprimovaná BTF data reprezentována jako vektor indexů a škálovacích koeficientů s tím, že jednotlivé dimenze vícerozměrné funkce jsou komprimovány jednotlivě. Rekonstrukce dat z tohoto modelu je založena na zřetězeném indexování vnořených kódových vektorů. V navrženém modelu jsou jas a chromacita dat komprimovány nezávisle pro dosažení ještě vyššího kompresního poměru. Tento ztrátový model pro kompresi BTF dat dosahuje nízkého zkreslení při dosažení velkého kompresního poměru $1:233-1:2040$ v závislosti na obsahu informace v BTF datech. Dosažené výsledky lze porovnat s dalšími metodami pro kompresi BTF dat, které typicky mají předdefinované kompresní poměry, často menší než $1:200$. Výsledky pro navržený model porovnáváme s modelem LPCA pomocí psychovizuelního experimentu. Dekomprimace dat z modelu je implementována jak na standardním procesoru tak i na grafickém hardware s tím, že je dosaženo reálného času zobrazování při osvětlení jedním bodovým světlem. Model navíc umožňuje rychlé vzorkování BTF dat pro zadaný vektor kamery, které je vyžadováno algoritmy trasování cest (angl. *path tracing*) a fotonových map.

Klíčová slova: počítačová grafika, realistická syntéza obrazu, stínování, textura, vektorová kvantizace, odrazivost povrchu

Keywords: computer graphics, three-dimensional graphics and realism, shading, texture, vector quantization, BTF, reflectance

Contents

1 Introduction 1

2 Basic Terminology and Notation 2

3 Previous Work 3

4 A Novel BTF Model 3

4.1 Model Parametrization . . . 3

4.2 Vector Quantization . . . 5

4.3 Multi-Level Vector Quantization . . . 6

4.4 Similarity Measure . . . 7

4.4.1 BRDF Data Compression . . . 7

4.4.2 BTF Data Compression . . . 8

4.5 Scalar Quantization and Compact Indices for Code-books . . . 8

5 Discussion 9

5.1 Vector Quantization Scheme . . . 9

5.1.1 Generation of Optimized Code-Books . . . 9

5.1.2 Thresholds Setting . . . 9

5.1.3 Mipmapping . . . 10

5.2 Importance Sampling . . . 10

6 Results 10

7 Comparison with Other Methods 12

8 Conclusion and Future Work 14

9 Ing. Vlastimil Havran, Ph.D. 16

1 Introduction

Realistic visualization of surface appearance has always been one of the main challenges in computer graphics. In particular, the increasing attention is paid in almost all industrial sectors to such applications as computing visual safety simulations and computer aided design (CAD) that require realistic reproduction of material behavior under complex illumination and viewing conditions.

One method to capture real material appearance is based on the measurement of reflectance with respect to varying light and viewing directions. This so called *Bidirectional Reflectance Distribution Function* (BRDF) was first described in [NJH*77]. The BRDF itself does not preserve texture information, so it is suitable only for homogeneous materials. However, a large number of real, rough surfaces have a complicated spatial structure that causes effects such as shadowing, masking, inter-reflection, and subsurface scattering, all of which vary with illumination and viewing directions.

To preserve at least some of these effects, a new representation of real-world materials, the *Bidirectional Texture Function* (BTF), was presented in [DvGNK99]. A monospectral BTF is a six-dimensional function which, unlike BRDF, accounts for the dependence of viewing and illumination measurements on planar material position. An appropriately measured BTF contains information about material properties as anisotropy, masking, or self-shadowing. Examples of rendered images using BTF are depicted in Figure 1.

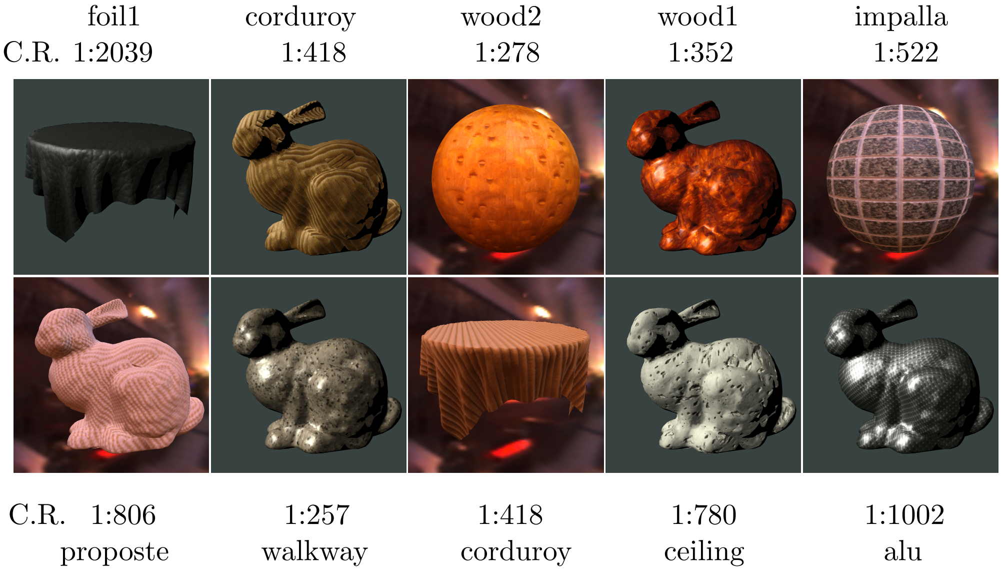


Figure 1: Example images rendered by the proposed BTF compression methods for illumination by point light and by different environment maps.

Uncompressed BTF requires a very large amount of data storage such as several giga-bytes per sample in raw format. The storage space requirements of raw BTF data prevents their direct use for fast rendering in modern graphics hardware. Hence a BTF data compression that produces a compact representation is necessary. Such a method should provide: (a) reasonably high compression ratios, (b) fast random access data synthesis (convenient for GPU implementation and rendering algorithms), (c) visual fidelity comparable with existing BTF compression algorithms.

Additionally, the method should allow fast importance sampling for high-quality rendering applications using eye path tracing algorithms, and spatial enlargement of measured BTF samples. The processing/workflow pipeline for BTF data is shown in Figure 2.

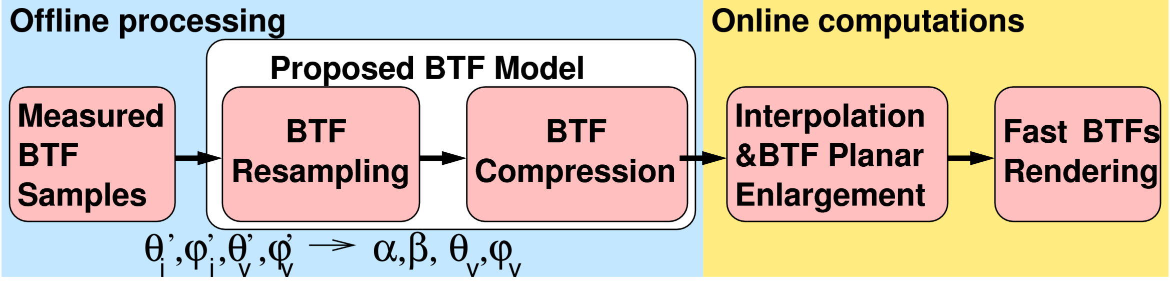


Figure 2: BTF data processing pipeline.

We present a novel BTF compression technique based on efficient multi-level vector quantization, allowing fast importance sampling for a given viewing direction as well as efficient multi-sample compression into a single shared database. To our knowledge there is no other BTF compression method that has these features.

The following section describes the basic terminology. Section 3 outlines prior work in the fields of BTF compression and importance sampling. Section 4 explains individual parts of the proposed model. Section 4.1 proposes a novel BTF data parametrization and interpolation. In Section 4.2 a vector quantization algorithm of interpolated data is explained, and this is followed by a description of a novel multi-level vector quantization method introduced in Section 4.3. Section 4.4 discusses a similarity measure applied throughout the model for BTF and BRDF data. Section 4.5 describes the use of scalar quantization to achieve further compression. Properties of the model and its application to fast importance sampling are discussed in Section 5. The results of our method are described in Section 6. A comparison of the method with other existing methods is shown in Section 7. Section 8 concludes the description of the method.

2 Basic Terminology and Notation

In this section we describe basic terminology and notation used below. The incoming light direction is denoted by $\omega_i = [\theta_i, \varphi_i]$ and viewing direction by $\omega_v = [\theta_v, \varphi_v]$. BRDF is a four-dimensional function $BRDF(\omega_i, \omega_v)$ and has two main important properties [DF97]. The first one is the *Helmholtz reciprocity rule* stating that if the illumination and viewing direction are reversed, the value of the BRDF should not change. The second property is the *energy conservation law*, which states that the ratio of total outgoing radiance from the material and incoming total radiance from the light sources must be less than or equal to one for all possible illumination directions.

Monospectral BTF is a six-dimensional function, $BTF(\mathbf{x}, \omega_i, \omega_v)$ depends on viewing and illumination measurements on planar material position $\mathbf{x} = [x, y]$. BTF can be decomposed into a set of illumination and viewing direction dependent texels specifying pixel-wise BRDFs. We will call such a texel as an *apparent BRDF* and denote it as $F_{\mathbf{x}}(\omega_i, \omega_v)$. Due to masking, shadowing, and light scattering effects the apparent BRDF fulfills neither the Helmholtz reciprocity rule nor energy balance.

3 Previous Work

We refer for BTF compression methods by the full and recent review of previous work related to our approach to the full version of our paper by Havran et al. [HFM10], and to thorough recent reviews by Filip and Haindl [FH09], and the tutorial by Lensch et al. [LGC*05].

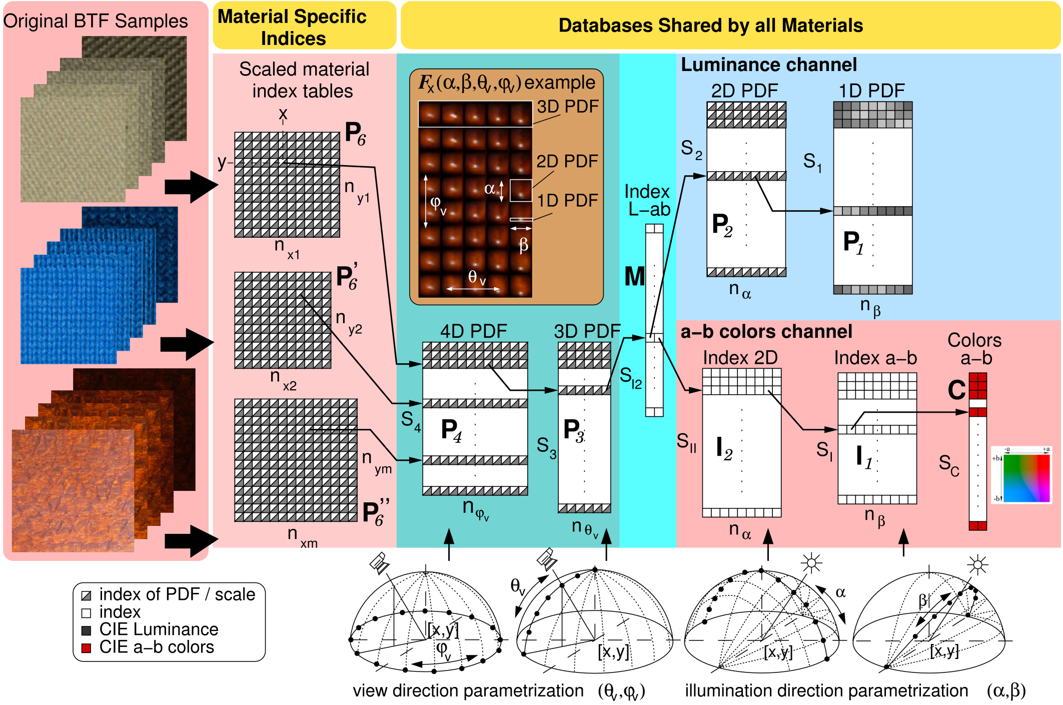


Figure 3: The proposed BTF model scheme; dependencies of individual VQ code-books.

4 A Novel BTF Model

The scheme of the proposed BTF model is shown in Fig. 3. The compression scheme is based on subsequent decomposition of 4D, 3D, 2D, and 1D dimensional slices of BTF data. These slices are obtained by resampling original BTF data to a novel parametrization of illumination and viewing directions. The model's compression is achieved by vector quantization of slices to individual dimensions to obtain a set of code-books. These code-books work as nested look-up tables of indices and scales of the individual slices while only the code-books at the lowest level contain the resampled original BTF data.

4.1 Model Parametrization

The key motivation of the model was to propose a light direction parametrization over a hemisphere that enables not only efficient data compression but also fast rendering and importance sampling. In addition, as the proposed model decomposes the function to parts in the individual dimensions separately, we want to align the data at these individual dimensions.

We considered several different parametrizations proposed for BRDFs e.g., half-angle parametrization by Rusinkiewicz [Rus98], by Stark et al. [SAS05], and Edwards et al. [EBJ*06].

However, we decided to abandon them for three reasons. First, the published parametrizations do not preserve monotonicity between the generated direction and the bivariate uniform variable in the input domain. That is, when we generate a similar pair of random numbers we want to get a similar generated direction for all random pairs. This is discussed in more detail in Section 5.2. Second, many BTF samples have distinct properties from BRDF samples. Typically, in BTF data there can be several specular highlights that are not aligned with the direction of an ideal reflected ray. The third reason is that we want to compress not only one but many apparent BRDFs. We want to align their perceptually similar features as we expect similarity among apparent BRDFs across a BTF sample. Hence the design of the parametrization proposed here specifically for BTF data compression is tightly coupled with a multi-level vector quantization method described in Section 4.3.

The proposed parametrization defines BTF slices that can be represented as conditional probability density functions (PDF). These PDFs are treated as input data into the vector quantization scheme proposed in the Section 4.2.

The proposed $[\alpha, \beta]$ parametrization is based on an "onion slices" concept of a hemisphere of illumination directions, as illustrated in Fig. 4. The hemisphere is divided into a set of meridian slices running between points **A** and **B** lying at its bottom part. Each slice is parametrized by angle $\beta \in \langle -\pi/2, \pi/2 \rangle$ with a zero value at the upper pole **E** of the hemisphere. A uniform placement of individual 1D slices over a hemisphere is controlled by angle $\alpha \in \langle -\pi/2, \pi/2 \rangle$ with zero value at the upper pole as well. A mapping $\mathcal{M}$ between standard hemispherical $[\theta, \varphi]$ parametrization and the proposed $[\alpha, \beta]$ parametrization can be stated as follows:

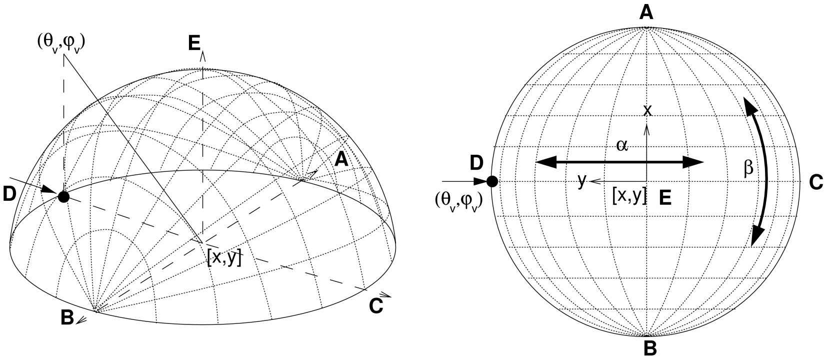


Figure 4: Illumination direction parametrization over a hemisphere.

$$\begin{array}{rcl} \mathcal{M}(\theta, \varphi) & \rightarrow & \{\alpha, \beta\} \\ \theta \in \langle 0, \pi/2 \rangle & & \alpha \in \langle -\pi/2, \pi/2 \rangle \\ \varphi \in \langle 0, 2\pi \rangle & & \beta \in \langle -\pi/2, \pi/2 \rangle \ . \end{array} \tag{1}$$

A corresponding unit 3D directional vector can be specified by means of the $[\theta, \varphi]$ and $[\alpha, \beta]$ parametrization respectively as

$$\begin{array}{rcl} [x, y, z] & = & [\cos\varphi \cdot \sin\theta, \ \sin\varphi \cdot \sin\theta, \ \cos\theta] \\ [x, y, z] & = & [\sin\beta, \ \sin\alpha \cdot \cos\beta, \ \cos\alpha \cdot \cos\beta] \ . \end{array} \tag{2}$$

While the illumination direction ω_i is specified by $[\alpha, \beta]$, the viewing direction ω_v is given by standard $[\theta, \varphi]$ parametrization only resampled to regular sampling steps of angles θ and φ. On one hand this resampling causes dense sample distribution near the pole of the hemisphere but on the other hand it allows direct factorization of samples along angles θ_v and φ_v. Such a resampling consequently allows better compression of underlying data samples.

Now we describe how $F_{\mathbf{x}}$ is resampled from original spherical parametrization $\theta'_i, \varphi'_i, \theta'_v, \varphi'_v$:

$$\begin{array}{rcll} & & \{\alpha, \beta, \theta_v, \varphi_v\} \leftarrow \mathcal{M}(\theta'_i, \varphi'_i, \theta'_v, \varphi'_v) & (3) \\ \theta_v & = & \theta'_v \qquad \beta = \arcsin(\sin\theta'_i \cdot \cos(\varphi'_i - \varphi'_v)) & \\ \varphi_v & = & \varphi'_v \qquad \alpha = \arccos\left(\dfrac{\cos\theta'_i}{\cos\beta}\right) \; . & \end{array}$$

Note that the hemisphere is oriented in such a way that an outline between points **A** and **B** is always perpendicular to the azimuth of viewing direction φ_v. Additionally, we need to achieve equitable distribution of samples on the hemisphere in $[\alpha, \beta]$ parametrization. Due to this reason the sample distribution along β slice is not chosen uniformly according to β angle shift, but uniformly in $\cos\beta$ so that the projection of the samples is equidistant.

To test the algorithm BTF data from [SSK03] were used. These data provide uniform distribution of measurement points in at 81 illumination and 81 viewing angles over the hemisphere. Such a distribution is obtained by using variable quantization of azimuth angle φ for individual elevation angles θ. These BTF data were resampled into proposed illumination $[\alpha, \beta]$ and viewing $[\theta_v, \varphi_v]$ direction parametrization by means of a two-step interpolation scheme based on radial basis functions [CBC*01]. The resulting data of interpolation (see example in orange part of Fig. 3) are then used as direct input into the proposed multi-level vector quantization based BTF model.

4.2 Vector Quantization

The proposed BTF compression model utilizes lossy block coding often referred to as vector quantization (VQ) [GG92]. VQ is based on an assumption that a set of data vectors can be represented by its representative subset – the code-book. This subset is obtained by representing similar vectors m by a suitable code-vector $\hat{m}$ according to a predefined maximal allowed distance. The similarity between them is defined by a distance measure $d(m, \hat{m}) > 0$.

Let us mention the important theorem for lossy compression methods related to our work [GG92, p.313]: *When the code-book is set optimally then no other coding system exists that can do better than VQ.* So a careful design of the code-vectors is the main issue. This theoretical result both motivates and justifies the use of vector quantization in lossy compression schemes.

In our method a vector code-book is based on selective elimination of input data-vectors until a final set of input data-vectors remains as the code-book, a procedure also known as pruning [TG74]. This idea of code-vectors generation can be explained in three steps: **Step 1:** Put the first input data-vector V_1 in the empty code-book. **Step 2:** With each new input data-vector V_x, find the nearest code-vector V_{NN} in the code-book. **Step 3:** If the minimum found distance between the vector V_x and V_{NN} is not within some threshold ε, add the input data-vector V_x to the code-book and return its index. Go to step 2. Otherwise, return index of the nearest code-vector V_{NN} and continue to step 2.

The selection of a distance measure appropriate for input data and the setting of the corresponding threshold ε have a crucial influence on the performance of the vector quantizer.

4.3 Multi-Level Vector Quantization

Now we can connect all the building blocks described above and explain our BTF compression model. As input data a $F_{\mathbf{x}}$ is converted from original θ'_i, φ'_i and θ'_v, φ'_v data parametrization into a novel parametrization $[\alpha, \beta]$ and $[\theta_v, \varphi_v]$ as described in Section 4.1. An example of $F_{\mathbf{x}}$ for lacquered wood material is depicted in the orange part of Fig. 3.

The general scheme of the proposed BTF model is shown in Fig. 3. The resulting four-dimensional function $F_{\mathbf{x}}(\alpha, \beta, \theta_v, \varphi_v)$ is decomposed along a viewing azimuth angle φ_v into a set of three-dimensional functions. Similarly, each 3D function is decomposed along a viewing elevation angle θ_v into a set of two-dimensional functions. Each 2D function describes the behavior of material reflectance along all slices in $[\alpha, \beta]$ parametrization, where data of a single slice can be considered as a one-dimensional function. To enable perceptually correct matching of individual data patterns and sharing of some common material features, the input BTF data were converted from standard RGB space into more perceptually uniform color space. YCrCb color space was used for LDR BTF samples and LogLUV [Lar98] for HDR BTF samples. In the rest of the text, regardless of the color-space that is used the luminance channel is denoted by L and chromaticity channels by a and b.

The original 1D, 2D, 3D, and 4D luminance functions are normalized to obtain corresponding conditional probability functions (PDF), which are used as training vectors for our VQ scheme. The proposed BTF model is based on BTF data decomposition into several code-books of indices and scale coefficients, while only the code-books on the lowest level contain the resampled original BTF data as one-dimensional vectors.

The vector quantization of luminance BTF data is carried out separately for individual dimensions as shown in Fig. 3. As a result of 1D PDFs quantization we obtain code-book $\mathbf{P}_1$(size $S_1 \times n_\beta$) of normalized 1D data slices along illumination angle β. This code-book is indexed by the $\mathbf{P}_2$(size $S_2 \times n_\alpha$) code-book of 2D PDFs representing luminance values where each item contains indices and scales c_{P_2} of individual 1D slices along illumination angle α in $\mathbf{P}_1$. Items in $\mathbf{P}_2$ are indexed from the auxiliary code-book $\mathbf{M}$(size $S_{I2} \times 2$). $\mathbf{M}$ in fact only merges indices pointing into luminance and color code-books ($\mathbf{P}_2$ and $\mathbf{I}_2$) and is indexed from the code-book of 3D PDFs $\mathbf{P}_3$(size $S_3 \times n_{\theta_v}$), where for each item indices and scales c_{P_3} corresponding to viewing angle θ_v are stored. The $F_{\mathbf{x}}$ encoding is finished by the last shared code-book $\mathbf{P}_4$(size $S_4 \times n_{\varphi_v}$), which provides items corresponding to viewing angle φ_v with indices and scales c_{P_4} to $\mathbf{P}_3$.

Chromaticity channels a and b (Cr/Cb or U/V) of BTF data are quantized in a slightly different way. The $\mathbf{C}$(size $S_C \times 2$) code-book stores basic a and b color values. Possible color variations along 1D slices are described in items of the $\mathbf{I}_1$(size $S_I \times n_\beta$) code-book and the corresponding color can be looked-up by indexing into $\mathbf{C}$. Color variations for all illumination directions are obtained by means of items of the $\mathbf{I}_2$(size $S_{II} \times n_\alpha$) code-book. Each such item of length n_α determines which color variations from $\mathbf{I}_1$ are used for individual positions of angle α.

The code-books $\mathbf{P}_2$ and $\mathbf{I}_2$ are stored separately to allow the use of different color variations for the same luminance distribution over a hemisphere of different illumination directions. This arrangement also makes it possible to save considerably fewer $\mathbf{P}_2$ slices when, e.g., BTFs of similar material structure but different color are encoded. The luminance and color information is merged by means of the auxiliary code-book $\mathbf{M}$ indexed from $\mathbf{P}_3$. $\mathbf{M}$ contains only index to $\mathbf{P}_2$ and index to $\mathbf{I}_2$. The remaining $\mathbf{P}_4$ and $\mathbf{P}_3$ code-books have the same function as in the luminance channel.

During BTF compression, individual $F_{\mathbf{x}}$ are compared with reconstructed $\hat{F}_{\mathbf{x}}$ in $\mathbf{P}_4$ by means of nested indexing through all code-books. If a similar code-vector is not found, $F_{\mathbf{x}}$ is decomposed into a set of less dimensional slices and the same process continues on all levels of the model either until the similar slice is found, or the $\mathbf{P_1}$ or $\mathbf{C}$ code-books are reached. Then the new unique data are inserted into the code-book $\mathbf{P_1}$ in the form of a luminance vector of length n_β along a slice parametrized by angle β or a chromaticity in the code-book $\mathbf{C}$. The insertion to $\mathbf{P_1}$ and $\mathbf{C}$ corresponds to standard vector quantization. During insertion the data-vector is compared so that the luminance is normalized in both the inserted data-vector and the data-vector in the code-book. When the match is found, this then provides a corresponding scale for upper-level code-book.

All the code-books described so far enable efficient coding of color $F_{\mathbf{x}}$ and can be shared by more BTF samples (i.e., materials). However, individual apparent BRDFs $F_{\mathbf{x}}$ do not provide any information about sample structure, so for coding of an entire BTF a material-specific planar index is needed. Such an index is obtained by VQ of individual $F_{\mathbf{x}}$ and stored in a form of $\mathbf{P}_6(n_{x_m} \times n_{y_m})$ code-book where $n_{x_m} \times n_{y_m}$ is the spatial resolution of the m-th BTF sample. $\mathbf{P}_6$ contains an index to $\mathbf{P}_4$ together with its scale value c_{P_6}.

The scale values are used for scaling of the stored PDFs to obtain correct reconstruction of 4D PDF function, i.e., $F_{\mathbf{x}}$, in the form of a compound function as follows

$$\begin{aligned}
c_{scale} &= c_{P_6} \cdot c_{P_4} \cdot c_{P_3} \cdot c_{P_2} \\
k &= \mathbf{P}_3(\mathbf{P}_4(\mathbf{P}_6(x,y),\varphi_v),\theta_v) \\
F_{\mathbf{x}L} &= c_{scale} \cdot \mathbf{P}_1(\mathbf{P}_2(\mathbf{M}(k,1),\alpha),\beta) \\
F_{\mathbf{x}\{a,b\}} &= \mathbf{C}(\mathbf{I}_1(\mathbf{I}_2(\mathbf{M}(k,2),\alpha),\beta),\{1,2\}) \\
F_{\mathbf{x}\{L,a,b\}} &\to F_{\mathbf{x}\{R,G,B\}} \ .
\end{aligned} \tag{4}$$

4.4 Similarity Measure

For VQ in the proposed BTF compression model we need a similarity measure between the input data-vector and the stored code-vector; this is of crucial importance for the compression algorithm. The data-vector corresponds to either a 1D, 2D, 3D, or 4D slice of $F_{\mathbf{x}}$ of BTF at a given planar position. We studied and tested several possibilities used for probability density functions (PDF) and traditional distance measures such as Euclidean function. Below we describe our final choice for our compression method, but the selection of the optimal similarity measure in BTF compression remains an open problem and can be improved in future.

4.4.1 BRDF Data Compression

As BRDF data lacks the spatial neighborhood information, we decided to use the mean square error (MSE) as a distance function between the original and the compressed data. The computation of MSE, which corresponds to computing Euclidean distance, has one big advantage. We can specify for each code-book $\mathbf{P}_1, \mathbf{P}_2, \mathbf{M}, \mathbf{P}_3$, and $\mathbf{P}_4$ the maximum MSE that is acceptable for compression. While the maximum MSE for $\mathbf{P}_4$ is user specified, the

MSE thresholds for other code-books are smaller by multiplicative constants such as 0.4 and squares of multiplicative constants among code-vectors.

The MSE for $\mathbf{P}_4, \mathbf{P}_3$, and $\mathbf{M}$ is computed directly in sRGB color space according to the definition of MSE, but the MSE for $\mathbf{P}_2$ and for $\mathbf{P}_1$ is computed for luminance only. The thresholds for code-books $\mathbf{I}_1, \mathbf{I}_2, \mathbf{C}$ of color components of YCrCb/LogLUV space are set to small constants; $\mathbf{I}_2$ threshold=0.5, $\mathbf{I}_1$ threshold=0.2, $\mathbf{C}$ threshold=0.1.

4.4.2 BTF Data Compression

After experiments with several similarity measures we finally decided to analyze BTF samples using a structural similarity index measure (SSIM) [WBSS04]. Another advantage of SSIM over other standard image quality measures as MSE, PSNR, etc. is that SSIM also takes into account both the surroundings of the compared pixels and local visual masking effects. SSIM measures the local structure similarity in their local neighborhood of an $R \times R$ window of pixels in an image (usually 11×11, [WBSS04]). The basic idea of SSIM is to separate the task of similarity measurement into comparisons of luminance, contrast, and structure combined into one similarity function:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \ , \tag{5}$$

where μ_x, μ_y, σ_x, σ_y and σ_{xy} are mean values, standard deviations, and mutual variance of values in the local neighbourhood of compared images X and Y. C_1, C_2 are specific non-zero constants. The valid range of SSIM for a single pixel is $[-1, 1]$, with higher values indicating higher similarity. The mean value accross the set of pixels in two images is understood as MSSIM in the rest of the text.

The MSSIM is computed for $\mathbf{P}_2$ and for $\mathbf{P}_1$ over luminance only, for $\mathbf{P}_4, \mathbf{P}_3$, and $\mathbf{M}$ in YCrCb/LogLUV color space, for $\mathbf{I}_1$ and $\mathbf{I}_2$ and $\mathbf{C}$ in two chromaticity channels of YCrCb/LogLUV color space. To combine luminance and chromacity in SSIM we propose a novel method. For example for $\mathbf{P}_4$ we compute MSSIM for all three channels (e.g Y, Cr, and Cb) for all combinations of viewing and illumination direction for the selected discretization over the hemisphere, which yields $3 \times \mathbf{n}_{\varphi_\mathbf{v}} \times \mathbf{n}_{\vartheta_\mathbf{v}} \times \mathbf{n}_\alpha \times \mathbf{n}_\beta$ values. As a similarity measure, we then compute the 98th-percentile from MSSIM values for all three channels.

The proposed approach is computationally efficient, as it allows us to prune the vector comparisons during the search as soon as we achieve the percentile value already found as the current best in the code-book found. The Nth-percentile of MSSIM values is consistently computed over all code-books either from luminance $(\mathbf{P_2}, \mathbf{P_1})$, two chromaticity channels $(\mathbf{C}, \mathbf{I_2}, \mathbf{I_1})$, or all three channels $(\mathbf{P_4}, \mathbf{P_3}, \mathbf{M})$. There is no need for a multiplicative factor for thresholds as the percentile method propagates the results from code-book of higher dimensionality to those of lower dimensionality.

4.5 Scalar Quantization and Compact Indices for Code-books

Scalar Quantization. During compression, we store the indices and scale values in code-books simply by 32-bits for an integer index and for floating point in 32-bits in IEEE-754 format. As the scale values are limited to a small range of values we apply a simple scalar quantization [GG92] for floating point values. For simplicity and ease of decompression we use scalar quantization to 8 bits for LDR BTF samples for all levels. For HDR BTF samples

it is necessary to increase the precision for $\mathbf{P}_2$ to 16 bits. The maximum relative error is far below 1% in all cases, the relative error yields values in range from 10^{-4} to 10^{-3}.
Compact Indices. Similarly, the size of code-books is reduced. Therefore the index in a code-book $\mathbf{P}_i$ pointing to another code-book $\mathbf{P}_{i-1}$ of size $\mathbf{S}_{i-1}$ can be represented only by $N = \lceil \log_2 (S_{i-1}) \rceil$ bits instead of 32 bits.

5 Discussion

This section discusses features of the proposed model and its application for importance sampling of BTF data.

5.1 Vector Quantization Scheme

There are several advantages of the proposed model. The individual code-books of luminance and color slices and their indices can be shared by an arbitrary number of BTF samples and can therefore enable even higher compression. Data compression is carried out on all levels of the proposed model. We set the thresholds for code-books so that the low-level code-books contain most of the code-vectors. In contrast methods based on PCA [MMK03, VT04] or spherical harmonics [WL03], which have predefined compression ratios, our approach can adapt to variance in input data-vectors. Additionally, whenever a new BTF sample arrives it can be easily processed by our model. This is much more difficult or infeasible with the other methods mentioned above.

5.1.1 Generation of Optimized Code-Books

The order of processing apparent BTFs across a BTF sample has a large impact on the final results. To guarantee that code-books describe a perceptual variety of pixels across a whole BTF sample, and to ensure a sufficient compression ratio, the individual code-books are generated in a progressive sampling algorithm. First, a small set (e.g. 1%) of apparent BTFs $F_{\mathbf{x}}$ across the whole BTF sample is randomly progressively sampled from BTF data with a predefined threshold and is used in the VQ scheme. The samples are taken from a sampling sequence by two-dimensional Halton random number generator. Second, the threshold is increased (e.g. by 2.5 times) and the same process is repeated for a larger set of $F_{\mathbf{x}}$ (e.g. 4%), again for the whole BTF sample. In the third step we do not modify the code-books and compress the rest of the pixels (e.g. 95%) in any order.

The number of generated items in individual data sets can become so high that finding the closest code-vector can be very slow. For BRDF data comparison we take advantage of the fact that we are using MSE, which is a true metric, and so we address this problem by implementing a dynamic version of the LAESA method [MOV94] (see book [Sam06] for other nearest neighbor search algorithms in high dimensional spaces). For SSIM a 98th-percentile the efficient search pruning is implemented as described in Section 4.4.

5.1.2 Thresholds Setting

A common problem of all VQ algorithms is finding the optimal quantization thresholds that provide either a required compression ratio or satisfy a defined quality measure. In our case, such a measure is the computational model of perceived difference between rendered images using the VQ BTF compression scheme and images rendered using the original BTF data. When MSE is used as similarity measure, the maximum MSE difference allowed or required for each generation of a code-book is specified by an user.

When SSIM is a similarity measure the situation is also simple, because the measure directly estimates the perceptual difference between the original and the modeled data. The setting of thresholds effectively controls the trade-off between the compression ratio of the proposed VQ compression scheme and the visual fidelity of the resulting rendered images. There is an obvious limitation of our current approach - the SSIM is only a mathematical model of visual fidelity given two images. So the visual fidelity achieved is limited by accuracy of SSIM.

5.1.3 Mipmapping

Since rendering at different scales is important for direct visualization of BTF data on the visible object, it is necessary to address an anti-aliasing. In our model, mipmapping [Wil83] can be used in the same way as for ordinary textures. The reflectance data are averaged, and the data are compressed from fine scale to coarse scale, each level separately. This requires extension of the spatial index $\mathbf{P}_6$. Obviously, the compression ratio is decreased up to one third as for standard texture mipmapping.

5.2 Importance Sampling

Importance sampling is not supported by current BTF models, but our algorithm design allows it efficiently. It is implemented via a standard inverse transform method for discrete PDFs [Fis96] directly from the $\mathbf{P_2}$ code-book, without the necessity to compute the 2D PDF, as in for several other BTF compression methods. The proposed parametrization over 2D slices guarantees that for strictly positive values $F_{\mathbf{x}}$, and given the viewing direction ω_v and a couple of random numbers $\xi_{1,2} \in [0-1]^2$, we can generate the illumination direction ω_i using proper interpolation.

In contrast to [Mat03, LRR04, EBJ*06] our hemispherical parametrization of apparent BRDF allows us to preserve monotonicity between the generated direction and the bivariate uniform variable in the input domain, and avoids discontinuities at the same time over the surface of the hemisphere. If for random numbers ξ_1, ξ_2 the function generating direction is $\omega_i = DirIS(\xi_1, \xi_2)$, then it holds $\lim_{\delta_1 \to 0, \delta_2 \to 0} |DirIS(\xi_1 + \delta_1, \xi_2 + \delta_2) - DirIS(\xi_1, \xi_2)| = 0$ for $\forall \xi_1, \xi_2 \in [0,1]^2$.

6 Results

For our experiments we have used BTF data from the University of Bonn [SSK03] having spatial resolution 256×256 and angular resolution $|\omega_i| \times |\omega_v| = 81 \times 81$. Single BTF material in RGB for LDR data (8 bits per color channel) takes up to 1.2 GBytes. HDR data are considered to have resolution 12 bits per color channel (1.8 GBytes per material). All results presented below were computed for resampling of the original data using the discretization $n_\alpha = 11$, $n_\beta = 11$, $n_{\theta_v} = 7$, and $n_{\varphi_v} = 16$.

Implemented on a CPU, our model achieves 310,000 - 1,360,000 BTF evaluations per second. According to our comparison, it is about 1.5 times faster than the standard single-lobe Lafortune model [FH05]. All the tests were performed on a single core of PC with the processor 2.83 GHz, Intel(R) Xeon(R) CPU, 6 MBytes L2 cache, 16 GBytes RAM DDR2 400 MHz. The performance of the proposed model GPU implementation was tested on two graphics cards and their results for various 3D objects. The speed achieves from 8 to 170 frames per second for ATI Mobility graphics cards and NVidia GeForce 8800 GT for 3D models with 2k and 115k triangles.

No	BTF sample	our C.R.^{1}	our **C.R.**^{2}	our C.R.^{3}	our size^2 (Bytes)	time [h]	our SSIM	LPCA‡ SSIM
1	alu*	1:253	**1:1002**	–	80,460	0.39	0.850	0.941
2	corduroy	1:128	**1:418**	1:484	3,084,692	19.2	0.748	0.916
3	fabric1	1:117	**1:362**	1:419	3,558,788	29.3	0.737	0.915
4	fabric2	1:217	**1:710**	1:822	1,815,996	18.1	0.779	0.993
5	foil1	1:574	**1:2040**	1:2364	632,352	19.5	0.859	0.923
6	foil2	1:334	**1:1138**	1:1319	1,133,896	17.2	0.814	0.994
7	impalla	1:162	**1:522**	1:604	2,466,808	21.8	0.730	0.970
8	leather	1:366	**1:1244**	1:1441	1,036,352	17.2	0.802	0.994
9	proposte	1:236	**1:806**	1:934	1,599,412	18.0	0.710	0.988
10	pulli	1:87	**1:264**	1:306	4,873,008	27.1	0.699	0.955
11	wallpaper	1:222	**1:728**	1:843	1,771,420	28.8	0.776	0.963
12	wood1	1:101	**1:352**	1:408	3,664,060	22.3	0.866	0.811
13	wood2	1:75	**1:278**	1:322	4,625,772	17.2	0.886	0.957
14	wool	1:77	**1:233**	1:270	5,514,260	50.2	0.684	0.964
15	ceiling$^\diamond$	1:235	**1:780**	1:1102	2,653,188	20.1	0.711	–
16	floortile$^\diamond$	1:136	**1:360**	1:509	5,383,352	28.7	0.772	–
17	pinktile$^\diamond$	1:711	**1:2267**	1:3205	853,496	15.6	0.961	–
18	walkway$^\diamond$	1:102	**1:257**	1:363	7,514,860	37.4	0.884	–
		1:230	**1:764**	1:924	average C.R.		–	1:275

* sample size 64 × 64 only, $^\diamond$ HDR sample

‡ 128 × 128 pixels only due to extreme computational demands

Table 1: Comparison of our method with three other methods in terms of compression ratio and MSSIM$_W$ [WBSS04] values in YCrCb space for all tested materials. The range of MSSIM is $\langle 0.0, 1.0 \rangle$, where value 1.0 corresponds to equal images. C.R.^{1} is the compression ratio for representing code-book indices by 32-bits and floating point values by 32 bits. C.R.^{2} and size^2 is the compression ratio and the compressed size of BTF sample for representing indices by minimum numbers of bits and floating point values by 8 bits for LDR samples and 16 bits for HDR samples. Compression ratio C.R.^{3} uses the same representation as C.R^2, but several BTF samples are compressed together for sharing luminance characteristics.

Compression ratios achieved for individual BTF samples with corresponding compression times are shown in columns 2–6 of Table 1. From the results we can conclude, that lower compression ratios correspond to textile materials having higher structural variability and complex occlusion/translucency effects, such as corduroy, impalla, proposte, and pulli.

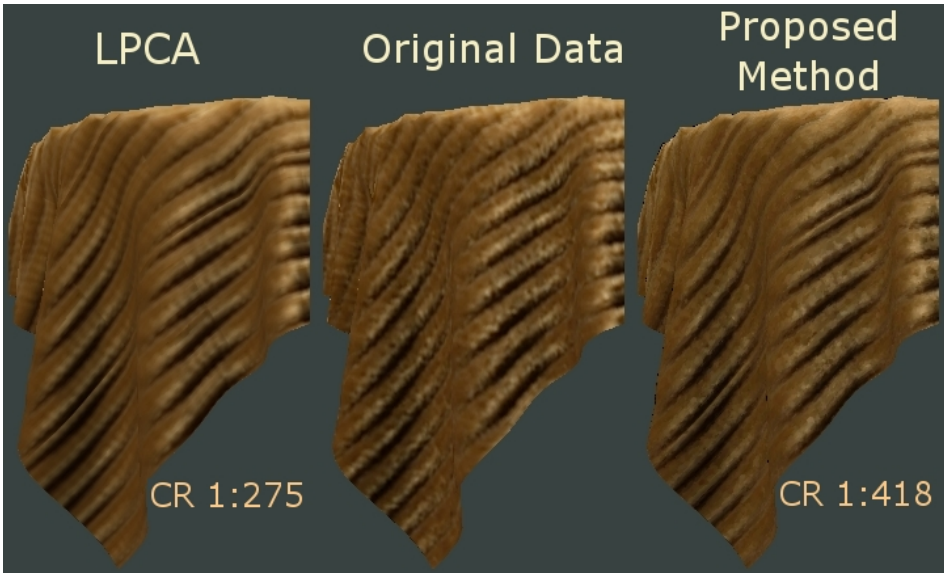
The average compression time of a BTF sample (size 256^2) using unoptimized implementation of the proposed VQ algorithm on a single CPU core, was about 23.4 hours, including BTF data resampling to the proposed parametrization. When we compress 13 BTF LDR samples (except alu) to a shared representation, the compression ratio is increased further by a factor from 15%. When compressing 4 HDR samples to a shared representation, the compression ratio is increased by 40% (not reported in Table 1). Images rendered using our BTF model for point light and environment lighting (Grace Cathedral, St. Peter's Basilica courtesy of Paul Debevec (http://www.debevec.org), and grassplain) are depicted in Fig. 1.

The proposed BTF compression method can also be used for BRDF when the BRDF samples are understood as apparent BRDFs. We compressed 100 isotropic BRDF measured samples (courtesy of Wojciech Matusik and MERL BRDF database [Mat03]) with an original data size of each sample of 90×90×180×3 numbers (=16.69 MBytes of data) for various discretizations. For example, for the discretization $n_\alpha = 91$, $n_\beta = 91$, $n_{\theta_v} = 45$, and $n_{\varphi_v} = 1$ we compressed 100 BRDF samples with a negligible average MSE error to 41 MBytes (compression ratio C.R.≈1:42). The shared data in the code-books are luminance characteristics in the code-books $\mathbf{P}_1$ and $\mathbf{P}_2$.

We have measured the speed of the importance sampling algorithm using the proposed model. Given a viewing direction the computed illumination direction achieves 450,000 - 1,600,000 samples per second on a single core CPU.

7 Comparison with Other Methods

We have compared the proposed method in terms of data compression with local PCA compression (LPCA) method representing BTF clusters [MMK03] using 7 clusters, 5 components/cluster.



(a) (b) (c)

Figure 5: Example comparison of the methods for corduroy sample. (a) LPCA based compression (b) reference uncompressed data (c) the proposed method.

Fig. 5 shows compression of an example data for BTF sample corduroy, other results can be found at http://www.cgg.cvut.cz/members/havran/btfbase/. In average the proposed method provided subjectively comparable overall visual quality across all tested samples, however, in average provides compression ratio more than twice higher than the LPCA method (settings 7 clusters with 5 components per cluster - C.R. 1:275, our method on average C.R. 1:764).

In order to objectively compare visual fidelity of these two methods we performed a simple psychological experiment with 19 participants. The subjects with normal or corrected vision of average age 27 years were shown 14 animated sequences of the rotating tablecloth objects with mapped BTF as shown in Fig. 5, i.e., the video rendered from original data always in the middle and from the compressed data by the proposed method and the LPCA, side-by-side in random order. The video for each BTF sample has been shown for 25 seconds, the whole test took between 7 to 9 minutes for each subject. The subjects' task was to evaluate which of the method provides more realistic visual experience given the reference data in the middle. For each person 14 LDR BTF samples were shown, which gave 266 individual answers.

As can be seen in Figure 6 summarizing our perceptual experiment the LPCA works better for materials with relatively small spatial appearance variations across images (please refer to a list of materials in Table 1). This is typical for such materials as alu, fabric2, foil1, foil2, and leather. Our compression allows better adaptation to complex materials having large variety of non-typical features such as corduroy, impalla, proposte, pulli, wood1, and wool. This is to be expected because our approach assumes a similarity on the level of apparent BRDFs allowing the efficient representation of irregularities thanks to the multi-level decomposition of data, while in LPCA the features are easier to represent by limited set of basis functions. The very small p-value ($p = 2.2 10^{-16}$) of ANOVA test indicates that differences between samples' means are highly significant. The mean evaluation is 5.15, where 5 means undecided and 6 very low preference of the LPCA. This means that the proposed method is comparable with the LPCA in terms of visual performance. However, it has much lower memory requirements during sample analysis, it compresses each sample according to its variability, it allows more materials to be compressed efficiently into one data set, and it performs fast importance sampling (about 300 times faster than approach using data reconstruction from LPCA).

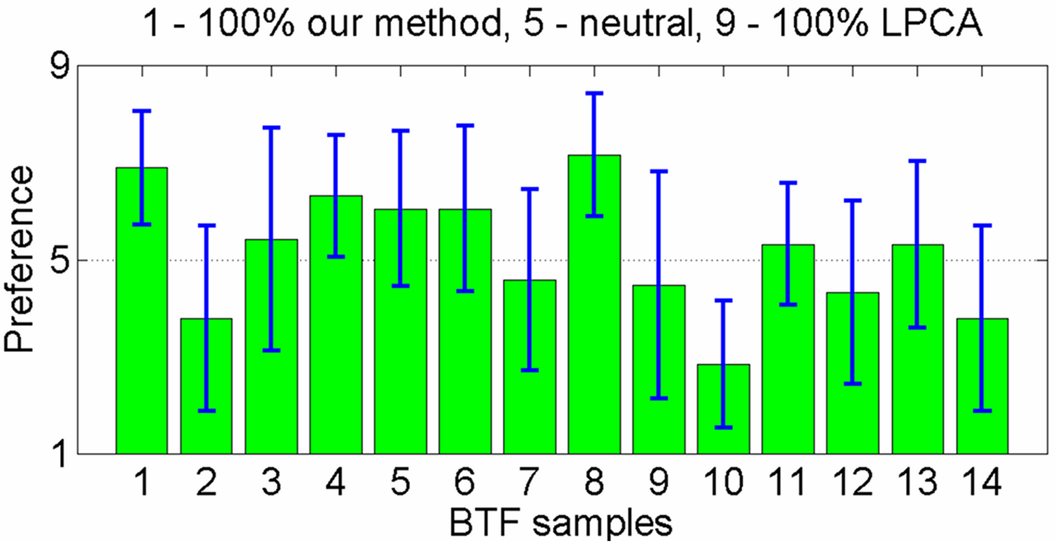


Figure 6: Evaluation of the psychological experiment for 19 participants and 14 LDR BTF samples. The error bars represent twice the standard error across subjects.

We can also compare the proposed method based on multi-level vector quantization with standard (i.e., one-level) vector quantization. We omit the discussion of the implementation here due to the lack of space. The standard vector quantization for the same parametrization reaches compression ratios up to 1:80, which is about 10 times less than achieved by the multi-level approach.

8 Conclusion and Future Work

The main contribution of this work is a novel BTF compression method based on vector quantization enabling high compression ratios between $1:233-1:2267$ (on average $1:764$) depending on material sample variability suitable also for BRDF data. This is further increased by 15% to 40% when several BTF materials are compressed to a common representation. For compression of BTF samples we directly use the SSIM metric to control the estimated visual similarity between the original and the compressed data. The proposed algorithm can efficiently control quality versus a compression ratio. The quality metric can be changed in future to a more efficient one. Additionally, the proposed method allows fast importance sampling of BTF/BRDF data.

We tested the functionality of the algorithm for 18 distinct BTF materials in HDR and LDR format, and have thoroughly compared the achieved results with results of LPCA method. High fidelity of the results was also verified against true measured data in a z-buffer based renderer for both point and environment lighting. Additionally, we have implemented the BTF decoding algorithm on the standard GPU with framerates up to 170 FPS depending on the scene complexity.

The proposed BTF framework can be further elaborated in several directions. First, other similarity measures among apparent BRDFs that better exploit known perceptual properties of human vision can be researched. Second, when multi-spectral BTF measurements are available, we believe that our model can be simply extended by a more accurate color models.

Acknowledgments

Most of the material can be found in the extended form in [HFM10]. I would like to thank the co-authors of this work, Jiří Filip (UTIA, Academy of Sciences of the Czech Republic) and Karol Myszkowski (MPI für Informatik, Saarbrücken, Germany).

References

[CBC*01] Carr J., Beatson R., Cherrie J., Mitchell T., Fright W., McCallum B., Evans T.: Reconstruction and representation of 3D objects with radial basis functions. *ACM SIGGRAPH 2001, ACM Press* (2001), 67–76.

[DF97] DeYoung J., Fournier A.: Properties of tabulated bidirectional reflectance distribution functions. In *Proceedings of the Graphics Interface 1997 Conference* (1997), pp. 47–55.

[DvGNK99] Dana K., van Ginneken B., Nayar S., Koenderink J.: Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics 18*, 1 (1999), 1–34.

[EBJ*06] Edwards D., Boulos S., Johnson J., Shirley P., Ashikhmin M., Stark M., Wyman C.: The halfway vector disk for brdf modeling. *ACM Trans. Graph. 25*, 1 (2006), 1–18.

[FH05] Filip J., Haindl M.: Efficient image based bidirectional texture function model. In *Texture 2005* (October 2005), Heriot-Watt University, pp. 7–12.

[FH09] Filip J., Haindl M.: Bidirectional texture function modeling: A state of the art survey. *IEEE Transactions on PAMI 31* (2009), 1921–1940.

[Fis96] Fishman G. S.: *Monte Carlo: Concepts, Algorithms, and Applications.* Springer-Verlag, New York, NY, 1996.

[GG92] Gersho A., Gray R. M.: *Vector Quantization and Signal Compression.* Communications and Information Theory. Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[HFM10] Havran V., Filip J., Myszkowski K.: Bidirectional texture function compression based on the multilevel vector quantization. *Computer Graphics Forum 29*, 1 (Jan 2010), 175–190.

[Lar98] Larson G. W.: LogLuv encoding for full-gamut, high-dynamic range images. *Journal of Graphics Tools: JGT 3*, 1 (1998), 15–31.

[LGC*05] Lensch H., Gösele M., Chuang Y.-Y., Hawkins T., Marschner S., Matusik W., Müller G.: Realistic materials in computer graphics. In SIGGRAPH 2005 Tutorials, 2005.

[LRR04] Lawrence J., Rusinkiewicz S., Ramamoorthi R.: Efficient BRDF importance sampling using a factored representation. *ACM Trans. Graph 23*, 3 (2004), 496–505.

[Mat03] Matusik W.: *A Data-Driven Reflectance Model.* Ph.D. thesis, Massachusetts Institute of Technology, September 2003.

[MMK03] Müller G., Meseth J., Klein R.: Compression and real-time rendering of measured BTFs using local PCA. In *Vision, Modeling and Visualisation 2003* (November 2003), pp. 271–280.

[MOV94] Micó L., Oncina J., Vidal E.: A new version of the nearest-neighbour approximating and eliminating search algorithm (aesa) with linear preprocessing time and memory requirements. *Pattern Recognition Letters 15*, 1 (1994), 9–17.

[NJH*77] Nicodemus F., J.C. R., Hsia J., Ginsburg I., Limperis T.: Geometrical considerations and nomenclature for reflectance. *NBS Monograph 160, National Bureau of Standards,* (October 1977), 1–52.

[Rus98] Rusinkiewicz S.: A new change of variables for efficient brdf representation. In *Proceedings of Eurographics Workshop on Rendering* (1998), pp. 11–22.

[Sam06] Samet H.: *Foundations of Multidimensional and Metric Data Structures.* Morgan Kaufmann, 2006.

[SAS05] Stark M. M., Arvo J., Smits B.: Barycentric parameterizations for isotropic brdfs. *IEEE Transactions on Visualization and Computer Graphics 11*, 2 (2005), 126–138.

[SSK03] Sattler M., Sarlette R., Klein R.: Efficient and realistic visualization of cloth. In *Eurographics Symposium on Rendering 2003* (2003), pp. 167–177.

[TG74] Tou J., Gonzalez R.: *Pattern Recognition Principles.* Addison-Wesley, 1974.

[VT04] Vasilescu M., Terzopoulos D.: TensorTextures: Multilinear image-based rendering. *ACM SIGGRAPH 2004, ACM Press 23*, 3 (Aug 2004), 336–342.

[WBSS04] Wang Z., Bovik A., Sheikh H., Simoncelli E.: Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing 13*, 4 (2004), 600–612.

[Wil83] Williams L.: Pyramidal parametrics. *SIGGRAPH Comput. Graph. 17*, 3 (1983), 1–11.

[WL03] Wong T., Leung C.: Compression of illumination-adjustable images. *IEEE Transactions on Circuits and Systems for Video Technology 13*, 11 (November 2003), 1107–1118.

9 Ing. Vlastimil Havran, Ph.D.

Address:
Department of Computer Graphics and Interaction, Faculty of Electrical Engineering, Czech Technical University in Prague, Karlovo nám. 13, 12135 Prague 2, Czech Republic

E-mail:
havran@fel.cvut.cz

Education:

- 2001-2003: PostDoc at the Max-Planck-Institute for Computer Science, Saarbruecken, Germany
- 1996-2001: Ph.D. in Informatics and Computer Scienece, Doctoral Programme at the Czech Technical University in Prague
- 1990-1996: M.Sc. (Ing.) at Czech Technical University in Prague, Faculty of Electrical Engineering, Computer Engineering, graduated with distinction

Employment:

- 2009–now, adjunct professor, Czech Technical University in Prague
- 2006–2008, assistant professor/researcher, Czech Technical University in Prague
- 2003–2006, senior researcher, Max-Planck-Institute for Computer Science, Saarbrücken, Germany.

Journal and conference publications (May 2005 to May 2010):

1. J. Novák, V. Havran, C. Dachsbacher: *Path Regeneration for Interactive Path Tracing*, Eurographics 2010, short papers, Norrkoeping, Sweden, pp. 61–64, May 3–7, 2010.
2. V.Havran, J.Filip, K.Myszkowski: *Bidirectional Texture Function Compression based on the Multilevel Vector Quantization*, in journal Computer Graphics Forum, Volume 29, Issue 1, pp. 175–190, March 2010.
3. M. Zlatuška, V. Havran: *Ray Tracing on a GPU with CUDA – Comparative Study of Three Algorithms*, in WSCG 2010 conference, pp. 69–76, proceedings of Communication Papers, Pilsen, Czech Republic, February 2010.
4. J. Bittner, O. Mattausch, P. Wonka, V. Havran, M. Wimmer: *Adaptive Global Visibility Sampling*, journal ACM Transactions on Graphics (proceedings of SIGGRAPH 2009), Volume 28, Issue 3, pp. 94:1–94:10, August 2009.
5. R. Herzog, V. Havran, S. Kinuwaki, K. Myszkowski, H.-P. Seidel: *Global Illumination using Photon Ray Splatting*, journal Computer Graphics Forum (Proceedings of Eurographics 2007), Volume 26, Issue 3, pp. 503–513, September 2007.
6. V. Havran: *About the Relation between Spatial Subdivisions and Object Hierarchies Used in Ray Tracing*, Spring Conference on Computer Graphics, pp. 55–60, Budmerice, Slovakia, April 2007.

7. V. Havran and J. Bittner: *Ray Tracing with Sparse Boxes*, Spring Conference on Computer Graphics, Budmerice, pp. 49–54, Budmerice, Slovakia, April 2007.

8. V. Havran, J. Bittner: *Stackless Ray Traversal for kD-Trees with Sparse Boxes*, electronic journal Computer Graphics & Geometry (online journal), 2007, vol. 9, no. 3, pp. 16–30, ISSN 1811-8992.

9. V. Havran, R. Herzog, H.-P. Seidel: *On the Fast Construction of Spatial Hierarchies for Ray Tracing*, RT06 conference, pp. 71–80, Salt-Lake City, USA, September 2006.

10. I. Wald, V. Havran: *On building fast kd-trees for ray tracing, and on doing that in O(N log N)*, RT06 conference, pp. 61–69, Salt-Lake City, USA, September 2006.

11. V. Havran, J. Bittner: *Efficient Sorting and Searching in Rendering Algorithms*, a half-day tutorial T4 at Eurographics 2006, 74 pages, Vienna, Austria, September 2006.

12. V. Havran, R. Herzog, H.-P. Seidel: *Fast Final Gathering via Reverse Photon Mapping*, journal Computer Graphics Forum (Proceedings of Eurographics 2005), Volume 24, Issue 3, pp. 323–333, September 2005.

13. V. Havran, J. Bittner, R. Herzog, H.-P. Seidel: *Ray Maps for Global Illumination*, in 16th Eurographics Symposium on Rendering, pp. 43–54, Konstanz, Germany, June/July 2005.

14. V. Havran, M. Smyk, G. Krawczyk, K. Myszkowski, H.-P. Seidel: *Interactive System for Dynamic Scene Lighting using Captured Video Environment Maps*, in 16th Eurographics Symposium on Rendering, pp. 31–42, Konstanz, Germany, June/July 2005.

15. A. Efremov, V. Havran, H.-P. Seidel: *Robust and Numerically Stable Bezier Clipping Method for Ray Tracing NURBS Surfaces*, at Spring Conference on Computer Graphics, pp. 123–131, Budmerice, Slovakia, May 2005.

16. V. Havran, A. Neumann, G. Zotti, W. Purgathofer, H.-P. Seidel: *On Cross-Validation and Resampling of BRDF Data Measurements*, Spring Conference on Computer Graphics, pp. 154–161, Budmerice, Slovakia, May 2005.

Professional research services:

- March 2010 - March 2013, member of editorial board of journal Computer Graphics Forum, ISSN 0167-7055
- Tutorial co-chair for Eurographics 2007
- IPC Member for the Eurographics Symposium on Rendering (2008–2010), Eurographics short papers 2010, WSCG(2009, 2010), HPG(2009, 2010), GRAPP(2007–2010), SCCG(2006–2010), Symposium on Interactive Ray Tracing (2006–2008)
- Regularly reviewing for Eurographics, Siggraph, Graphics Interfaces, Pacific Graphics, SCCG, TVCG, WSCG, IEEE Symposium on Interactive Ray Tracing, High Performance Grapphics, GRAPP, Eurographics Symposium on Rendering conferences and symposia, Computer Graphics Forum, IEEE Computers and Graphics, Computer Graphics and Applications, IEEE TVCG, Visual Computer, Journal of Graphics Tools