



Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics

Supervising workplace:



Academy of Sciences of the Czech Republic
Institute of Information Theory and Automation
Pattern Recognition Department

Colour Rough Textures Modelling

Jiří Filip

Prague, May 2005

Thesis supervisor: Michal Haindl

PhD programme: Electrical Engineering and Informatics

Branch of study: Artificial Intelligence and Biocybernetics

Acknowledgements

I would like to thank to all the people who supported me during my research and writing of this thesis and here I try to mention those who have made the main contributions.

My main acknowledgements belong to my supervisor Michal Haindl who encouraged me during recent years and provided me with many inspiring ideas, advices and comments on this thesis. Without his help I would not be able to achieve any goal of this thesis.

I would like say thanks also to all my colleagues at Pattern Recognition Department at UTIA who enabled my pleasant work at the institute and offered me professional help. Namely to Petr Somol for BTF tiling and help during solving of various technical and implementation problems as well as to Jiří Grim for his valuable advices on data modelling. They helped me to put this thesis in much better shape by careful reading and valuable comments.

Further I wish to thank to Institute of Information Theory and Automation AS CR for enabling me to work at the institute during of my PhD studies and writing of this thesis.

Research and industrial partners had an important influence on development of proposed BTF models and BTF visualisation. Namely I would like to mention Reinhard Klein from University of Bonn for providing us with BTF measurements and Jan Meseth from the same institute for useful hints on texture mapping and BTF visualisation. Michael Arnold from DaimlerChrysler must be thanked for providing us with car 3D model and interior textures.

I am also very grateful to the following grants and projects which funded my research: EC projects IST-2001-34744 Real Reflect and FP6-507752 MUSCLE, grant of the Grant Agency of the Academy of Sciences CR No. A2075302, CTU internal grant No. CTU0407313 and grant MŠMT No. ME567 MIXMODE.

Of course I thank to my family and to Andrea for wide support and patience during those years.

List of Notations and Acronyms

r_1, r_2, r_3	row, column and spectral index
r	multi-index $r = \{r_1, r_2, r_3\}$
s_1, s_2, s_3	row, column and spectral shift
•	means all possible values of the corresponding index
θ, ϕ	elevation and azimuthal angle
d	the number of spectral bands
\mathcal{D}	BTF dimensionality
I	underlying rectangular image lattice
I_r	causal contextual neighbourhood defined on I
η	cardinality of causal contextual neighbourhood I_r
γ	parameter vector
Θ	parameter matrix
Y_r	image values on location defined by multi-index r
X_r	data vector on CN location defined by multi-index r
$\beta(r)$	the number of CN shifts on image plane
n_c	the number of preserved principal components
n_i	the number of view positions in BTF
n_v	the number of illumination positions in BTF
n_l	the number of reflectance lobes
n_p	the rank of mapping polynomial in PLM
ρ	the material albedo
σ	white noise variance
Σ	white noise variance matrix
ψ	exponential forgetting factor
2D, 3D, 4D, 6D, 8D	two, three, four, six, eight D imensional
BRDF	B idirectional R eflectance D istribution F unction
ABRDF	A pparent BRDF
BSSRDF	B idirectional S ubsurface S cattering R eflectance D istribution F unction
BTF	B idirectional T exture F unction
CAR	C ausal A uto- R egressive
CPU	C entral P rocessing U nit
CN	C ontextual N eighbourhood
GMRF	G aussian MRF
GPU	G raphics P rocessing U nit
LM	L afortune (reflectance) M odel
LM-C	C lustered L afortune M odel
PCA	P rincipal C omponent A nalysis
PLM	P olynomial extension of L afortune M odel
PLM-C	C lustered P olynomial extension of L afortune M odel
PS	P hotometric S tereo
MAE	M ean A verage E rror
MRF	M arkov R andom F ield
SFS	S hape F rom S hading
SVD	S ingular V alue D ecompositon
VR	V irtual R eality

Contents

1	Introduction and Motivation	1
1.1	Objectives of the Thesis	3
1.2	Chapter Outlines	4
2	Rough Surface Reflectance Representation	6
2.1	Bidirectional Reflectance Distribution Function	7
2.1.1	BRDF Measurement	8
2.1.2	BRDF Modelling	9
2.1.3	BRDF Factorisation	9
2.1.4	BRDF Approximation Using Reflectance Models	10
2.2	Bidirectional Subsurface Scattering Reflectance Distribution Function	13
2.3	Bidirectional Texture Function	13
2.3.1	BTF Measurement	14
2.3.2	BTF Data Representation	15
2.4	BTF Compression Methods Based on Analytical ABRDF Models	17
2.4.1	Pixel-Wise Analytical ABRDF Models	17
2.4.2	Pixel-Wise Analytical Reflectance Field Models	18
2.4.3	Surface Light Field Modelling	20
2.5	BTF Compression Methods based on PCA	20
2.5.1	Entire BTF Space Factorisation	21
2.5.2	Multimodal BTF Space Factorisation	21
2.5.3	Reflectance Field Factorisation	22
2.5.4	BTF Segmentation and Local PCA	23
2.6	BTF Synthesis and Modelling Approaches	23
2.6.1	BTF Synthesis from Parametric Tiles	23
2.6.2	Copy and Paste Based BTF Synthesis	24
2.7	Alternative BTF Modelling and Classification Methods	25
2.7.1	BTF Classification Based on Image Descriptive Histograms	25
2.7.2	BTF Compression Based on Image Histograms	25
2.7.3	Prediction of Lambertian Rough Texture Illumination Characteristics	26
2.7.4	Image Based Relighting	26
3	BTF Databases	28
3.1	Columbia-Utrecht Reflectance and Texture Database	28
3.2	Yale University BTF Database	29
3.3	Bonn University BTF Database	29
3.4	Comparison of BTF Databases' Parameters	32

4	BTF Rendering	33
4.1	BTF Mapping on a 3D Object	33
4.1.1	Additional Texture Mapping Tasks	35
4.2	BTF Interpolation	36
4.3	Surface Height Simulation	37
4.3.1	Bump Mapping	38
4.3.2	Displacement Mapping	41
5	Range Data Acquisition	42
5.1	Range-Map Measurement	42
5.1.1	Structured Light Range Sensors	43
5.1.2	Time of Flight Range Sensors	43
5.2	Range-Map Estimation	44
5.2.1	Shape from Stereo	44
5.2.2	Shape from Shading	45
5.2.3	Photometric Stereo	47
5.3	Range-Map Techniques Overview	51
5.4	Range-Map Synthesis	51
6	Segmentation of BTF Data	52
6.1	BTF segmentation	52
6.1.1	Data Representation	52
6.1.2	Histogram similarity	53
6.1.3	Colour-Space Transformation	53
6.1.4	K-Means Clustering	53
6.2	Optimal Number of Clusters	54
6.2.1	PCA on Raw Data	55
6.2.2	PCA on Raw Data Correlations	55
6.3	Clustered BTF Representation	57
7	Probabilistic BTF Modelling	59
7.1	BTF Segmentation into Subspaces	60
7.2	Surface Height Data Estimation and Enlargement	60
7.3	Multiscale Multispectral Subspace Models	61
7.3.1	3D Models	61
7.3.2	Approximation using 2D Model	62
7.3.3	Spectral Factorisation	64
7.3.4	Spatial Factorisation	64
7.3.5	3D Causal Auto-Regressive Subspace Model	66
7.3.6	2D Causal Auto-Regressive Subspace Model	69
7.3.7	2D Gaussian-Markov Random Field Subspace Model	70
7.4	Results	72
7.4.1	Synthesis of Individual BTF Images	73
7.4.2	Rough Texture Model from Spare Set of Textures	77
7.4.3	BTF on 3D Objects	79
7.5	MRF BTF Model Fast Implementation Issues	84
7.5.1	Synthesis of BTF Subspace Images	84

7.5.2	BTF Interpolation	85
7.5.3	Surface Height Simulation	85
8	BTF Modelling Using Reflectance Models	87
8.1	Lafortune Model	87
8.2	Sample-Size Lafortune BTF Model	89
8.2.1	Non-Linear Estimation of Model Parameters	90
8.2.2	Proposed Polynomial Extension of the Lafortune Model	90
8.2.3	Results of Sample-Size Reflectance Models	93
8.2.4	Compression of Lafortune Parametric Images	95
8.2.5	BTF Data Interpolation	99
8.3	Unrestricted Resolution BTF Reflectance Model	99
8.3.1	Reflectance Model Parametric-Plane Tiling	100
8.3.2	Modelling of the Reflectance Model Parameters	106
8.4	Summary of the Reflectance Models for BTF Modelling	106
9	Results Verification and Testing	112
9.1	Probabilistic BTF Models Verification	112
9.2	Reflectance BTF Models Verification	113
10	Conclusions and Future Work	116
10.1	Contributions of the Thesis	116
10.2	Future Research	118
	References	128
A	Appendix - Proposed MRF Model Results	129
B	Appendix - Proposed Reflectance Model Results	131
C	Appendix - Contents of the Enclosed CD	133

Chapter 1

Introduction and Motivation

In contemporary virtual reality (VR) systems demands grow for realistic and accurate modelling of visual properties of real materials in particular for purposes of visual safety simulation in automotive industry and car/architectural interior design among others. Standard smooth textures combined with reflectance models and bump mapping show not to be able to accurately capture real material behaviour, which depend on mesostructure roughness and anisotropic reflectance properties. The solution is employing rough textures that imitate reflective properties as close as possible to an original material with respect to illumination and view positions.

Although there is not any definition of texture generally accepted we can assume texture as random field realisation with spatially homogeneous properties whose principal characteristic is a repetition of a basic visual pattern. We can principally divide textures into two major categories. The *smooth textures* are textures which fulfil Lambertian law, i.e., their appearance is independent on illumination and view direction and represent ideally diffuse material. However, many of real-world materials violate this Lambertian assumption which results into the second major category - *rough textures*. Rough texture reflectance depends on illumination and viewing direction and this is caused by surface structure of the original material (shadowing, masking) or its underlying physical properties (anisotropic specularity, inter-reflections, subsurface scattering).

Although the rough textures allows accurate simulation of surface appearance, due to their high dimensionality their measurement, representation and modelling was significantly limited in the past. Nowadays with constant increase of available computational power the number of available VR systems increases as well as their overall complexity. Moreover, contemporary graphics hardware became a very powerful tool for computations performed in each individual fragment of virtual scenes. These advanced and continually developing graphics techniques enable to process relatively complex models that represent appearance of real-world materials used to cover objects in VR systems.

Visual perception of such objects significantly depends not only on their shape but particularly on the representation of the surface materials. The most common, computationally simple approach in the past was based on using a single smooth texture lit according to empirical reflectance model, optionally combined with a bump-mapping technique. Later the real material reflectance was measured to capture the original reflectance with respect to varying light and camera positions. This so called *Bidirectional Reflectance Distribution Function* (BRDF) was compressed and approximated by variety of analytical models in the past. BRDF, in itself, does not preserve texture information, so this

approach was suitable mainly for homogeneous materials as, e.g., metals, plastics, ivory and another polished homogeneous surfaces. However, a large number of real rough surfaces such as plaster, leathers, fabrics etc. have complicated spatial structure that causes effects such as shadowing, masking, inter-reflection and sub-surface scattering dependent on different illumination and view directions. During last years a new precise rough texture representation has appeared in form of *Bidirectional Texture Function* (BTF). BTF is a six-dimensional function which introduces additional dependency of measurements on material planar position. This means that BTF preserves, in contrary to BRDF, also the desired spatial texture information.

The importance of BTF is shown in Fig. 1.1. The figure depicts objects covered by two different BTF materials lighted from three different illumination directions. The change of illumination and view directions considerably affects the visual appearance of real materials and each of them in different way - this can be relatively easily recorded by means of BTF.



Figure 1.1: Example of BTF behaviour for two different materials.

The BTF measurements are usually represented by means of thousands of images taken for different illumination and viewing directions. Consequently, the storage size of such measurements is several GB depending on planar and angular BTF resolution. Due to this huge storage size the raw BTF data can not be directly used for any fast application even when high-end contemporary graphics hardware is employed.

For fast rendering of BTF on objects in VR an efficient BTF model is required. The main motivations and goals of the BTF modelling are:

- **Data compression** - Due to limited GPU memory the data size of original BTF measurements prevents any fast application. The size of BTF have to be considerably reduced to enable rendering of complex scenes comprehending several different materials ¹.
- **BTF enlargement** - Measured BTF is always too small to seamlessly cover virtual object surface of required size.
- **BTF reconstruction** - Ideal BTF modelling method should allow BTF reconstruction for unmeasured combinations of view and illumination angles within range of possibly spare set of BTF measurements.
- **Visual quality preservation** - Measured BTF describes real material reflectance properties and its main visual features and characteristics that have to be preserved.
- **Fast rendering of data from a BTF model** - Fast applications implemented directly in graphics hardware require efficient synthesis algorithms working preferably in pixel-wise manner.

¹All compression ratios in this thesis are related to a size of the original University Bonn BTF measure-

As BTF modelling is quite a new research area at the frontiers of computer graphics, computer vision and image processing, not too many publications exist so far in this field when compared to other research areas. Most of them address sample based BTF compression methods to store original or pixel-wisely parametrised BTF samples, thus the problem of texture enlargement remains open. Most recent methods are based either on clustered pixel-wise BRDF models (ABRDF) or on linear basis decomposition. The BTF measurements are enlarged by means of simple image repetition or seamless image tiling approaches. The rest of the methods synthesises novel BTF images by interpolation of sparse set BTF measurements according to Lambertian reflectance function or by matching estimated range-map of material with these sparse BTF measurements.

However, till now no generic BTF modelling approach is available which does not need to store neither BTF samples nor any of their pixel-wise parametric representation. None of contemporary BTF models can be considered perfect for all application fields, types of approximated material, required speed and visual quality.

1.1 Objectives of the Thesis

The main objectives of this thesis are as follows:

- Provide a survey of rough texture measurement, representation, compression and modelling methods published up to now.
- Provide an overview and comparison of publicly available BTF databases.
- Introduce two novel BTF modelling approaches and the corresponding models.

The first of them is probabilistic BTF modelling approach based on BTF segmentation and subsequent modelling of obtained subspace images by means of several distinct Markov Random Field (MRF) models. Due to the fact that simple MRFs have difficulties to reproduce regular low frequency structure of measured material, we have used displacement mapping filter which combines the synthesised subspace images with estimated range-map of modelled material.

The second proposed BTF modelling approach is a polynomial extension of pixel-wise Lafortune reflectance model. Synthesised BTF data are enlarged by means of image tiling of the model parameter space. To achieve higher compression ratios we employ an additional parameter clustering technique.

Both of these approaches enable fast hardware implementation in contemporary VR systems and were developed in the scope of EC project IST-2001-34744 *RealReflect* (Real Time Visualization of Complex Reflectance Behaviour in Virtual Prototyping) [1]. The main objective of this project was the development of advanced VR system aimed to high-end interior design in architecture and automotive industry. For design purposes the appearance of virtual objects has to be realistic, whereas the speed of rendering is assumed to be at interactive frame-rates. Although raw BTF measurements can with an additional enlargement method reproduce the reflectance properties of observed material very realistically, due to huge size of this BTF data

ments [98]. Each such a BTF dataset has angular resolution $n_i \times n_v = 81 \times 81$ and corresponding planar resolution of BTF images is 800×800 (see Section 3.3).

the rendering speed is unfortunately very low. However, the proposed BTF models offer reasonable visual quality while the amount of data to be stored is considerably reduced and fast rendering is guaranteed.

1.2 Chapter Outlines

The thesis is divided in chapters as follows:

Chapter 2: Rough Surface Reflectance Representation Contains a short review of material surface appearance measurement and modelling methods published up to now. Includes brief description of methods for representation and modelling of BRDFs, BSS-RDFs and BTFs and discusses their basic properties.

Chapter 3: BTF Databases This chapter describes three publicly available BTF databases and discusses their properties and limitations.

Chapter 4: BTF Rendering The basic problems occurring in BTF rendering and corresponding suggested solutions are proposed in this chapter. Includes BTF mapping, interpolation and additional surface macrostructure simulation by means of bump or displacement mapping.

Chapter 5: Range-Map Acquisition Range-maps and normal-maps enable one way of surface height simulation and takes part in proposed probabilistic BTF model introduced in Chapter 7. Different methods of range data acquisition and estimation are discussed and compared, and their results on real materials are provided.

Chapter 6: Segmentation of BTF Data This chapter describes the proposed approach of BTF segmentation of space of illumination and view directions into several BTF subspaces. This enables efficient subspace modelling by means of probabilistic BTF model introduced in Chapter 7.

Chapter 7: Probabilistic BTF modelling A novel BTF model based on combination of BTF segmentation, MRF probabilistic model and rough structure modelling based on displacement filter is introduced together with numerous results for distinct materials. Additionally, properties of the proposed model are discussed as well as possibilities of its fast implementation in hardware.

Chapter 8: BTF Modelling Using Reflectance Models This chapter introduces the novel reflectance BTF model based on polynomial extension of one-lobe Lafortune reflectance model and its clustered variant. Several ways of BTF enlargement are discussed and the chapter provides obtained results in comparison with original BTF measurements.

Chapter 9: Results Verification and Testing This chapter summarises the methods for quality verification and testing of individual proposed BTF models.

Chapter 10: Conclusions and Future Work This chapter summarises achieved results of BTF modelling and also mention open problems beyond the scope of our research and suggests further research directions of BTF modelling.

Chapter 2

Rough Surface Reflectance Representation

The main purpose of this chapter is to provide general information about research performed so far in the field of realistic modelling of real-world materials appearance. Each such a rough material can be characterised with respect to its visual appearance using following three major appearance levels:

- **Macrostructure level** - representing texture pattern repetition (usually low frequencies in corresponding texture image). Texture information on this level can be approximated by means of surface height measurements using displacement mapping with additional polygonal tessellation.
- **Mesostructure level** - includes relatively small yet still visible geometric details (usually higher frequencies in corresponding texture image), e.g., small bumps, woollen knits, etc.
- **Microstructure level** - involves surface microfacets which are visually indistinguishable and which control the overall appearance of material depending on view angles according to physical rules, e.g., smooth fibers of textile, which reflect light more efficiently if the light shines parallel to their orientation.

Depending on actual application task the real materials are modelled in scope of appropriate level. This approach enables significant reduction of the complexity of input high-dimensional data and consequently allows using simpler data representation. E.g., material modelling in scope of microstructure level reduces to single colour reflectance modelling while the macrostructure level requires a model that preserves the spatial structure of the material.

Several types of material representation have been used in computer graphics up to now whose complexity considerably depends on concrete application field. The most common material representations as well as their mutual relations are shown in Fig. 2.1. First we describe the most common BRDF representation that provides material reflectance information depending on illumination and viewing position. Several of BRDF modelling methods based on factorisation or reflectance models published up to now are discussed. However, the BRDF have proved to be insufficient for satisfactory representation of rough or textured materials, what led to further research and introduction of more complex

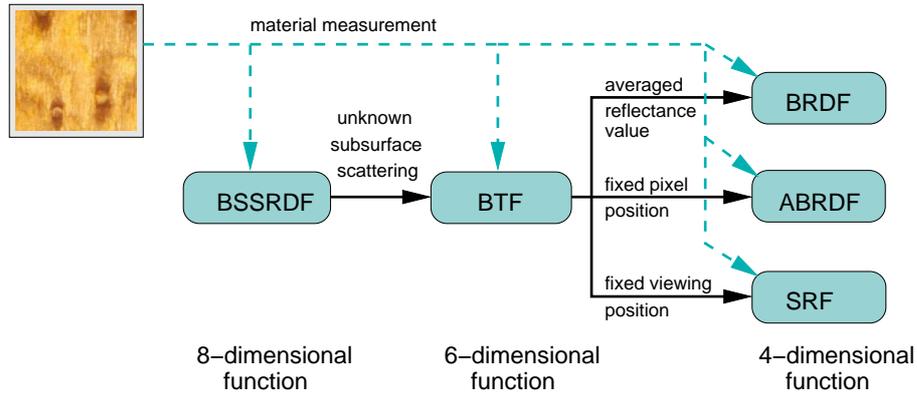


Figure 2.1: The overview of real-world material representations in computer graphics.

representation models. The most general of them is Bidirectional Subsurface Scattering Reflectance Distribution Function (BSSRDF). This eight-dimensional function comprises reflectance values for any combination of incoming / outgoing planar positions on material sample and illumination / viewing directions. Since it is quite difficult to measure and even model such a complex function, its simplification consisting in unification of incoming and outgoing planar position was necessary. This simplification results in six dimensional BTF which is used for photo-realistic rendering of real-world materials nowadays. This function is usually represented by a set of illumination / view dependent images and thus its size is still considerably large. Thus the development of efficient BTF compression and synthesis algorithms to enable fast BTF rendering using standard graphics hardware is one of main challenges in computer graphics community as well as main topic of this thesis. As BTF modelling is quite a new research area, there is not as many related results as in other research fields available. However, many contemporary image data processing algorithms can be favourably employed.

2.1 Bidirectional Reflectance Distribution Function

Material surface at micro-structure level can be represented by *Bidirectional Reflectance Distribution Function* (BRDF). BRDF is a 4D function introduced in work of Nicodemus [88] which describes the relation between incident irradiance E from direction $\omega_{\mathbf{i}} = [\theta_{\mathbf{i}}, \phi_{\mathbf{i}}]$ and radiance L reflected off observed material to direction $\omega_{\mathbf{v}} = [\theta_{\mathbf{v}}, \phi_{\mathbf{v}}]$ (see Fig. 2.2 and (8.1)) according to the following equation

$$BRDF(\theta_{\mathbf{i}}, \phi_{\mathbf{i}}, \theta_{\mathbf{v}}, \phi_{\mathbf{v}}) = \frac{dL_{\mathbf{v}}(\theta_{\mathbf{i}}, \phi_{\mathbf{i}}, \theta_{\mathbf{v}}, \phi_{\mathbf{v}})}{dE_{\mathbf{i}}(\theta_{\mathbf{i}}, \phi_{\mathbf{i}})} \quad (2.1)$$

where θ and ϕ depict elevation and azimuthal angles respectively (see Fig. 2.2). BRDF captures physical reflectance behaviour of uniform surface elements. It is called *bidirectional* because the incident and reflected directions can be reversed for the function to return the same value. This fact follows from the physics of light [5].

BRDF has two main important properties. The first one is the *Helmholtz reciprocity*

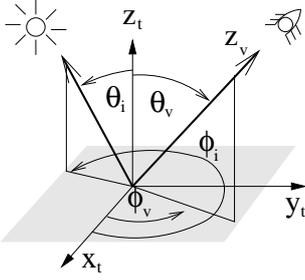


Figure 2.2: Relationship between illumination and viewing angles within texture coordinate system.

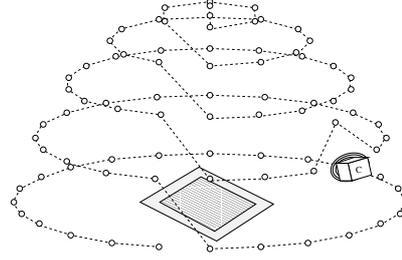


Figure 2.3: Light vector trajectory above the sample. The movement of illumination starts at the top of.

rule [10] fulfilling

$$\forall \omega_v \in \mathcal{H} \quad \forall \omega_i \in \mathcal{H} \quad (2.2)$$

$$BRDF_{r_3}(\theta_{i,1}, \phi_{i,1}, \theta_{v,1}, \phi_{v,1}) = BRDF_{r_3}(\theta_{i,2}, \phi_{i,2}, \theta_{v,2}, \phi_{v,2})$$

where $\theta_{i,2} = \theta_{v,1}$, $\phi_{i,2} = \phi_{v,1}$, $\theta_{v,2} = \theta_{i,1}$, $\phi_{v,2} = \phi_{i,1}$ and \mathcal{H} means the set of all possible positions on a hemisphere above the material.

The second property is the *energy conservation law* where the BRDF has to fulfil following normalisation condition

$$\forall \omega_v \in \mathcal{H} \quad \int_{\omega_i \in \mathcal{H}} BRDF_{r_3}(\theta_i, \phi_i, \theta_v, \phi_v)(N \cdot \omega_v) d\omega_v \leq 1 . \quad (2.3)$$

where N is surface normal at a given point and ω_v is viewing vector.

According to BRDF shape two kinds of surfaces are distinguished:

- **Diffuse surfaces** – the light is reflected in every direction. The limit case is Lambertian surface, i.e., the BRDF becomes a constant function when the light is reflected in every direction equally.
- **Specular surfaces** – the light is reflected only in a small area close to the mirror reflection. The limit case, i.e., the perfectly specular (Fresnel) surface, is obtained when the BRDF becomes a Dirac function when the light is reflected in one single direction.

2.1.1 BRDF Measurement

BRDF can be measured using gonioreflectometers. This device mechanically moves the position of light source and spectral sensor owing to measured material and collects large number of point samples. Because the BRDF is in general a function of four angles, two incident and two reflected, such a device must have four degrees of freedom to measure the complete function [116].

Another advanced BRDF measurement system together with data-driven reflectance model was presented by Matusik et al. in [74]. This measurement process gives 20-80 millions of BRDF samples per material. These samples are acquired at special coordinate

system based on view and illumination angles with respect to the half angle vector instead of the normal vector as it is common. This coordinate system allows to vary the sampling density near the specular highlight. This results in assigning higher angular resolution near specular reflection and vice versa.

Direct measurement is not the only way of obtaining BRDF. Ashikhmin introduces in [2] an efficient method that takes as input 2D micro-facet orientation distribution and produces a 4D BRDF. This method uses a simple shadowing term allowing to handle general micro-facet distributions while maintaining reciprocity and energy conservation.

Finally, the simplest way to obtain BRDF data is BTF data averaging in contextual neighbourhood containing all kinds of structure elements presented in original material.

2.1.2 BRDF Modelling

The main goal of BRDF modelling has been to develop a compact BRDF representation to enable fast rendering in graphics hardware. The BRDF measurements were approximated by several methods in the past. We present the main ones in the following.

2.1.3 BRDF Factorisation

BRDF can be represented by means of spherical harmonics [120] which are analogous to Fourier series, but in the spherical domain. Spherical harmonics are especially suitable for representing smooth spherical functions. This is often the case with reflectance functions. Using this method the BRDF (containing hundreds of measurements) can be represented using only up to 25 coefficients. 4D BRDF representation for real-time rendering applications utilising a 2D table of spherical harmonics coefficients was introduced in [55, 102].

Another BRDF factorisation method [52] exploits singular value decomposition (SVD) for separable BRDF decomposition. BRDF is replaced by the sum of products of two 2D functions stored in texture maps. Final BRDF reconstruction can be performed by means of a fast hardware-implemented multiplication of these texture maps. Moreover, the authors recommend normalised decomposition instead of SVD. This decomposition is faster, simpler and uses no more components than is required for final representation.

Even more efficient BRDF factorisation method based on homomorphic factorisation (HF) was introduced in [76]. Homomorphic factorisation, similarly to SVD, decomposes BRDF into several factors of lower dimensionality, each factor dependent on a different interpolated geometric parameter. Compared to SVD this technique generates a factorisation with only positive factors, enables control over the result smoothness and works well with scattered, sparse data without a separate resampling and interpolation algorithm. This approach was extended in [65] for isotropic BRDF lighting computation using environment maps.

A 4D surface light field function factorisation by means of non-negative matrix factorisation was shown in [9] to be significantly easier to implement than the homomorphic factorisation mentioned above. Compared to PCA the HF produces non-negative basis images that form a parts-based representation and all of them are needed to reproduce a consistent approximation of the input data.

2.1.4 BRDF Approximation Using Reflectance Models

Another BRDF modelling approach employs reflectance models for BRDF approximation. Several reflectance models have been commonly used for surface rendering in computer graphics. They can be divided into two major categories. The first includes simple but physically incorrect empirical models while the second comprises theoretical, physically valid and more complex models. Both kinds of models attempt to approximate the real reflectance function, represented by the BRDF, but the respective approaches are quite different.

Empirically Derived Reflectance Models

The empirical model is usually based on a very simple formula with several adjustable parameters designed to fit certain class of reflectance functions. Empirical model design does not pay attention to physical derivation or significance of individual parameters. Although these models are not physically plausible, they can offer computational simplicity following from low number of model parameters. The simplicity of empirical models enables their fast hardware implementation, what is the reason of their wide use to this day.

Probably the best known empirical model was introduced by Phong in 1975 [91]. This model has only three parameters in separated diffuse and specular terms [64]. The model is not physically plausible, therefore it is very hard to find the relation between the parameters of the model and the physical characteristics of the represented material. Thus the Phong model can not capture important reflectance effects, e.g., it enables to emit more light than is received.

A modification of the original Phong model targeted to achieve more realistic reflections is the Blinn-Phong model introduced by Blinn in [6]. This model is usually used for hardware accelerated bump-mapping.

Next empirical anisotropic reflectance model was introduced by Banks et al. in [4]. This model assumes small fibers along the given tangent, resulting in anisotropic reflections. It can be computed using a combination of dot-product computations and blending operations.

Another BRDF model combining the advantages of the various empirical models was introduced in Ashikhmin et al. [3]. The authors use a Phong-based specular lobe but make this model anisotropic and incorporate Fresnel behaviour while attempting to preserve the simplicity of the initial model as well as physical plausibility. For improving energy conservation of the Phong model for metallic materials the result of [87, 86] is used. Although the model is mostly empirical it interprets certain parts of specular term using physical-based micro-facet models [12, 116]. The approximation presented in [99] is used as a Fresnel factor. The diffuse term is realised in form of a modification of non-Lambertian BRDF presented in [100] which allows the diffuse-specular trade-off to conserve energy. This model is easy to use in Monte Carlo frameworks.

A different empirical BRDF model was introduced by Matusik et al. [74]. A BRDF measurement device is introduced and used to acquire BRDFs for more than hundred different materials was measured by means of this device, including metals, plastics, painted surfaces, etc.. Each measured BRDF is treated as high-dimensional vector and its dimensionality is subsequently reduced by means of linear analysis (PCA) and non-linear reduction (charting - subspace projections of measured samples). These reductions lead to

10D manifold which is approximately consistent with many theoretical isotropic reflectance models. Finally the model is tuned for actual material by visual inspection evaluating 16 user defined directions to navigate in the reduced-dimensions of BRDF space. On the low-dimensional manifold, movement along these directions produces novel but valid BRDFs. This empirical model fulfil reciprocity, non-negativity and energy conservation.

Another empirical model is described in [84]. Authors use the same V-facets as Torrance and Sparrow [112], but assume Lambertian reflectance of these facets. First a reflectance model is developed for anisotropic surfaces with one type of V-facets with all facets aligned in the same direction of surface plane. This result is then used to derive a model for the more general case of isotropic surfaces that have normal facet distributions with zero mean and arbitrary standard deviation which parametrises the macroscopic roughness of the surface. The authors have pointed out that several real-world objects have diffuse components that are significantly non-Lambertian. They intended their model for use in algorithms that analyse images of diffuse surfaces and recover accurate shape information.

Physically Derived Reflectance Models

One of the first theoretical reflectance models has been introduced in computer graphics by Torrance and Sparrow [112]. This analytical BRDF model assumes a surface consisting of vertical V grooves – perfectly specular micro-facets. The micro-facets normals deviation from the average surface normal is assumed to be a zero-mean Gaussian - the higher is the variance of deviation the rougher the surface. The model can be divided to two parts. The first one is associated with bulk material effects leading to a Lambertian lobe coloured by the diffuse albedo at a particular position on the surface. The second is entirely related to surface scatter.

The Torrance and Sparrow model was later enhanced by Cook and Torrance [12], who take use of the work previously done in physics by Torrance and Sparrow [112] about reflection of electromagnetic waves on rough surfaces. In that model, a surface is again supposed to be composed of micro-facets, i.e., small smooth planar elements. Only a part of micro-facets contribute to the reflection. The approximated BRDF depends on five different angles and is expressed as a linear combination of a diffuse and specular reflector.

The more complete BRDF model has been proposed by He et al. [42]. This sophisticated model based on physical optics incorporates the specular, directional diffuse, and uniform diffuse reflections of a surface. The model accounts for every physical phenomena involved in light reflection on rough surfaces such as polarisation, diffraction, interference, conductivity. In comparison to Cook and Torrance model [112] an additional term appears in the linear combination to represent coherent reflection on the mean plane of the surface (i.e., not the micro-facets).

Ward [116] presented even more complex, physically plausible anisotropic reflectance model based on Gaussian distribution of micro-facets. In contrast to previous work of He et al. [42] where not enough attention had been paid to normalisation of the reflectance function the presented model has built-in normalisation as well as other desirable features, such as permitting quick evaluation for data reduction and Monte Carlo sampling [117]. The model has the necessary bidirectional characteristics and all four of its parameters have physical meaning and can be fit independently to measured BRDF data to produce physically valid reflectance function. This reflectance model was fitted to BRDF measure-

ments of real materials obtained by means of measurement setup presented in the same paper [116]. Ward's measurement method exploits hemispherical mirrors reflecting the whole hemisphere of the flat probe at once onto a CCD camera equipped with fish-eye lens. This method captures the entire hemisphere of reflected directions simultaneously what considerably accelerates the whole measurement procedure.

Schlick's BRDF model published in [99] stands halfway between empirical and theoretical models. In this paper a rational fraction distribution is utilised for reflectance function representation. The idea is based on kernel conditions, which can be any intrinsic characteristic of the function (value at a given point of the function and one of its derivatives, integral or differential equations, etc.). Introduced rational fraction approximation scheme enables to speed-up the computation of reflectance model. Moreover, the author introduces an approximation of Fresnel factor, geometrical attenuation coefficient and slope distribution while the BRDF gets separated into spectral and directional factors. The model requires only a few intuitively describe reflectance parameters to define a material and a formulation of varying complexity is provided which is well suited to Monte Carlo rendering methods.

One of the first BRDF models which take into account the wave-like properties of light was published by Stam in [108]. This physical model is based on Kirchhoff integrals and is able to approximate diffraction of light on arbitrary surface structure and can be taken as a common generalisation of earlier approximative physical models mentioned in this section above. This model does not comprise such effects as multiple scattering and subsurface scattering and enables relatively easy implementation.

Phong's, Blinn-Phong's and Ward's models were decomposed into several factors to enable their computation directly in graphics hardware [53]. The individual models were combined with material textures (range-map, etc.) which control strength of the model anisotropy.

The main goal in optimal BRDF model development is finding a compact representation which can accurately describe the dominant behaviour of the BRDF. The representation should be computable using an iterative algorithm that would monotonically converge to a correct solution. Moreover, such a model should be physically plausible, reciprocal and energy conserving as well as easy to implement in graphics hardware. These conditions are fulfilled by the model presented in Lafortune et al. [63]. This model is based on cosine lobes and represents material reflectance by means of a new class of primitive functions with nonlinear parameters. These functions are reciprocal, energy-conserving, capture important BRDF features as off-specular reflection, increasing reflectance with angle of incidence, retro-reflection while the representation is compact and uniform.

Generally, the reflectance function representation by means of BRDF works well for smooth non-textured materials as, e.g., metal or glass, but is absolutely insufficient for rough textures like textiles or leathers due their local non-homogeneity, which causes:

- **Masking** - parts of the material surface are invisible from view position due to their occlusion by other surface parts (masking by intersection) or due to their orientation away from view position (self-masking).
- **Shadowing** - parts of the surface are not irradiated due to their orientation (self-shadowing) or because the incident beam is intersected (shadowing by intersection).

- **Interreflections** - parts of the surface irradiate other parts, thus producing multiple scattering.
- **Subsurface scattering** occurs in slightly translucent materials. Light enters their surface, is scattered around inside the material, and then exits the surface, potentially at a different point from where it entered.

To represent at least some of these effects the following material description functions have been introduced recently.

2.2 Bidirectional Subsurface Scattering Reflectance Distribution Function

A general material surface introduces subtle lighting effects, such as masking, shadowing, interreflections and subsurface scattering. To capture all these effects a more general approach than simple BRDF is inevitable.

The most general description of material reflectance properties offers a 8D *Bidirectional Subsurface Scattering Reflectance Distribution Function* (BSSRDF) described as follows

$$BSSRDF_{r_3}(r_{i1}, r_{i2}, r_1, r_2, \theta_i, \phi_i, \theta_v, \phi_v) \quad (2.4)$$

where r_{i1}, r_{i2} represent planar coordinates where the light enters into the material, r_1, r_2 represent planar coordinates where the light emits from the material while θ and ϕ are elevation and azimuthal angles of illumination i and view v positions (see Fig. 2.2).

BSSRDF describes the light transport between every point on the surface for any illumination and view position. Obviously, the complexity of BSSRDF based methods [45] limits their application mainly to homogeneous diffuse materials such as fluids, marble, etc..

Recent measurement techniques [32] enable to capture BSSRDF for translucent inhomogeneous materials with strong subsurface scattering effects. However, the authors assumed diffuse surface reflections and did not pay attention to angular dependency of laser beam light source or light sensor. The rendering method exploiting subsurface scattering effect is presented in [67]. Nowadays there is no method enabling us to measure or even model the BSSRDF in its whole complexity available.

2.3 Bidirectional Texture Function

Since there is not any fast method available neither for measurement nor for modelling of BSSRDF up to now, it is inevitable to use a simplified variant of the BSSRDF enabling universal measurement of reflectance data for rough materials. One solution is the *Bidirectional Texture Function* (BTF). BTF is a six dimensional function, similarly to BSSRDF, dependent on illumination and view angles as well as on planar position on observed material surface (2.5).

$$BTF_{r_3}(r_1, r_2, \theta_i, \phi_i, \theta_v, \phi_v) \quad (2.5)$$

Actually, the BTF represents one planar texture for each combination of illumination and view position. Thus the BTF integrates subsurface scattering light intensity from

neighbouring material surface locations in individual BTF images by means of sample measurements. Thus the subsurface scattering effect can not be figured out, however, it is preserved in BTF measurements and can be modelled together with other reflectance properties of an observed material sample.

BTF was first presented in work of Dana et al. [16]. Only three BTF measurement databases have been made public so far. Real BTF measurements consist of several thousands material sample images taken for different combinations of illumination and view positions. The standard BTF measurement comprehends whole hemisphere of possible light and camera positions of observed material sample according to preselected angular quantisation steps (see Fig. 2.3 for University of Bonn BTF measurement quantisation).

An appropriately measured BTF offers enough information about material properties as is anisotropy, masking or self-shadowing. In contrast to a regular 2D texture or to BRDF, BTF is high-dimensional and involves large amounts of data. To render BTF on graphics hardware, a compact representation of the BTF is needed. The best currently available BTF [98] takes up about 2GB of storage space per sample. BTF database for moderately complex VR scenes can take up to several TB of data space. Hence some compression and modelling method of this huge BTF datasets is inevitable. Such a method should provide compact parametric representation and preserve main visual features of the original BTF as much as possible, while enabling its fast rendering in contemporary graphics hardware.

The modelling of BTF as a 6D function is not trivial, thus task so many researchers factorised BTF into individual subsets of textures. The two most frequently used BTF subsets are *Surface Light Field* (SLF) and *Surface Reflectance Field* (SRF).

The surface light field (2.6) represents the BTF slice containing all the BTF images corresponding to a fixed illumination position i .

$$SLF_{r_3}(r_1, r_2, \theta_i, \phi_i) \sim \mathcal{L}_i \quad (2.6)$$

Similarly the surface reflectance field (2.7) represents the BTF slice containing all the BTF images corresponding to a fixed view position v .

$$SRF_{r_3}(r_1, r_2, \theta_v, \phi_v) \sim \mathcal{R}_v \quad (2.7)$$

2.3.1 BTF Measurement

Only few BTF measurement systems exist up to now. These systems are (similarly to BRDF measurement systems) based on light source, video or still camera and material sample moving using a robot arm. The main difference between individual BTF measurement systems is in type of measurement setup allowing four degree of freedom and type of measurement sensor (CCD, video, etc.). In some systems the camera is fixed and the light is moving while in others it is contrariwise. The main requirement on BTF measurements is accurate image rectification, i.e., aligning of texture normal with view vector, mutual registration of single BTF measurements and visual constancy during measurement. The rectification accuracy strongly depends on used light/camera and robot positioning errors while the visual constancy depends on stability of material properties during long measurement time when exposed to strong light source. Most of these problems are solved by the recently proposed measurement setup [83] based on array of 151 digital still cameras mounted on hemispherical chassis above the material sample. Camera built-in flashes are

used as light sources. The homomorphic transformation necessary for rectification of individual BTF images can be precisely calculated in advance for each camera, since there are not any moving parts. This approach can significantly decrease rectification errors to sub-pixel values and speeds up the whole measurement process considerably. The main known BTF measurements systems are described in more detail in Chapter 3.

2.3.2 BTF Data Representation

It is possible to define at least two possible ways of BTF data representation which are commonly used in variety of BTF modelling and compression methods as depicted in Fig. 2.4.

Texture representation

$$BTF_{TEX}(i, v) \quad (i, v) \in \mathcal{B} \quad (2.8)$$

where (i, v) is illumination and view position and \mathcal{B} represents a set of all BTF images. The $BTF_{TEX}(i, v)$ describes BTF as a set of discrete textures depending on view and illumination direction. The number of images is product of the numbers of all view and illumination positions (in the case of University of Bonn BTF datasets it is 81×81).

Apparent BRDF (ABRDF) representation

$$BTF_{ABRDF}(r_1, r_2) \quad (r_1, r_2) \in I \subset (M \times N) \quad (2.9)$$

where (r_1, r_2) is planar index in BTF image and $M \times N$ is resolution of original BTF images. In this representation the BTF can be seen as a set of apparent BRDFs (ABRDF), one for each planar position (r_1, r_2) . The term apparent BRDF was first introduced in [120]. The ABRDF encodes reflection of a single point given a illumination and view position. It is called *apparent* because the underlying geometry is of a much larger scale compared to normal BRDFs. The ABRDF, in contrary to BRDF, does not fulfil physical reciprocity since it includes strong shadowing and masking effects as well as scattering effects from neighbouring parts of the surface as depicted in Fig. 2.5. Technically this representation can be considered as a set of subsequent images where each image corresponds to one pixel in BTF. Such an image provides information about pixel reflectance value (ABRDF) when the view, illumination position represent horizontal, vertical index in ABRDF image respectively. The number of images in the ABRDF representation is given as the number of pixels in original BTF images $M \times N$. From our observations as well as from [82] it follows that the image to image variance in BTF_{ABRDF} representation depends mainly on the structure of observed material surface producing spatial variation of reflectance properties as well as on scattering effects on observed pixel neighbourhood. These effects are nicely illustrated in Fig. 2.5. Two ABRDFs of four different materials for different planar positions representing distinct reflective properties are shown.

In BTF_{ABRDF} representation the reflectance changes occurring between individual ABRDF images are caused by shadowing and masking effects. However, the specular reflectance peaks (light parts in Fig. 2.5) are placed at the fixed positions for all ABRDF images.

Whereas, in BTF_{TEX} representation additional visual variances between individual BTF images are presented. These variances can be caused by measurement and registration errors of individual BTF images or by different material specularity and shadowing for

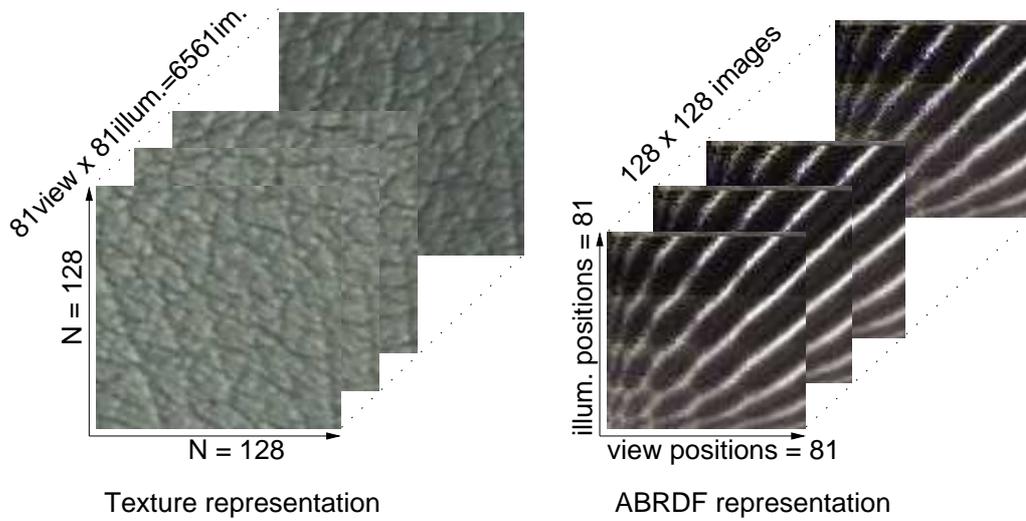


Figure 2.4: Two common BTF representations - *Texture representation* (left) and *ABRDF representation* (right).

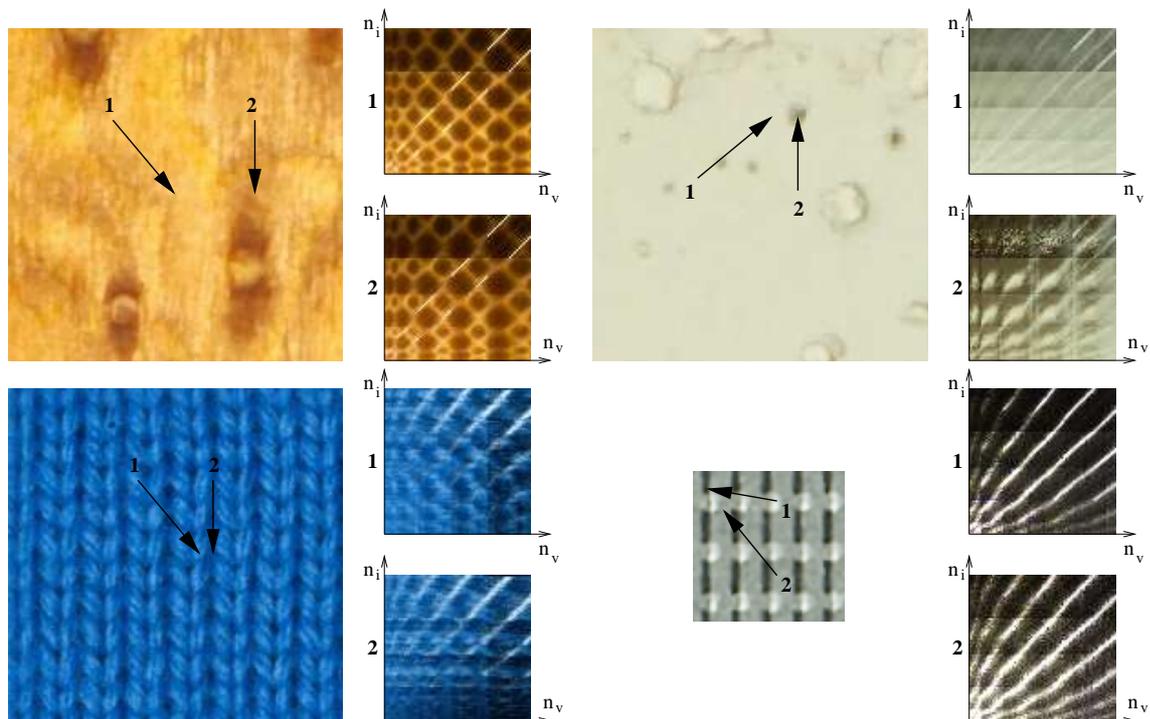


Figure 2.5: ABRDFs corresponding to four different BTFs (*wood02*, *plaster*, *knitted wool* and *aluminium*) measured for each material in two different planar positions (highlighted by arrows) on BTF image.

different view and illumination directions. Moreover, due to an occlusion effect the same pixel in all BTF images does not necessarily corresponds to the same point on material surface (see Fig. 2.6).

2.4 BTF Compression Methods Based on Analytical ABRDF Models

2.4.1 Pixel-Wise Analytical ABRDF Models

Similarly to other high-dimensional data the BTF exhibit local linearity which can be exploited for data factorisation and subsequent use of simpler models or methods. One example can be pixel-wise factorisation of original BTF. In this factorisation individual pixel data represents apparent BRDF and can be approximately modelled by means of various BRDF models.

Pixel-wise ABRDF Lafortune Model

One of the first attempts to hardware accelerate BTF approximation in graphics hardware using analytical BRDF model was done by McAllister [75]. This model introduces spatial BRDF (SBRDF) which represents ABRDF of each pixel in BTF by means of sum of several reflectance lobes according to Lafortune model parametrisation

$$BTF(r_1, r_2, i, v) \approx \rho_{d,r_1,r_2} + \sum_k^{n_l} \rho_{s,r_1,r_2,k} [\omega_i^T \mathbf{D}_{r_1,r_2,k} \omega_v]^{n_{r_1,r_2,k}}, \quad (2.10)$$

where ω_i, ω_v are illumination and view vectors in local coordinate system, \mathbf{D} is a diagonal matrix with model parameters. The remaining parameters ρ_d, ρ_s, n_k represent diffuse, specular albedo and specular exponent, respectively.

The number of reflectance lobes n_l is less or equal to three. The model parameters are estimated according to Levenberg-Marquardt non-linear fitting process [94]. Although the McAllister's model provides very compact BTF representation together with one of the first real-time BTF rendering application, its use is limited to materials with more or less smooth structure. For coarse materials with surface height variations this method can not capture ABRDF accurately and produces blurred results.

Scaled Pixel-wise ABRDF Lafortune Model

An extension of McAllister's BTF model was presented by Daubert et al. [19]. The approach was originally intended for realistic cloth modelling. It consists of two Lafortune reflectance lobes scaled by additional coefficients stored in a look-up table T as follows

$$BTF(r_1, r_2, i, v) \approx T_{\omega_v, r_1, r_2} \left(\rho_{d,r_1,r_2} + \sum_k^{n_l} [\omega_i^T \mathbf{D}_{r_1,r_2,k} \omega_v]^{n_{r_1,r_2,k}} \right), \quad (2.11)$$

where ω_i, ω_v are illumination and view vectors in local coordinate system, \mathbf{D} is diagonal matrix with model parameters, ρ_d is diffuse albedo and n_k is specular exponent.

Each spectral channel uses dedicated lobe parameters obtained by means of non-linear fitting process and multiplicative look-up table T_{ω_v, r_1, r_2} whose values are result of an additional iterative process. The look-up table stores pixel-wise colour and alpha values for

each of the original viewing directions that model shadowing and masking effects of individual rough material structure elements as well as strong specular highlights and other effects caused by complex material geometry. However, the look-up table requires to store significantly more parameters than in McAllister’s approach. This is not a big problem for BTFs of regular materials, as shown by the authors in the case of fabrics, but for general materials this approach involves quite large data storage space.

2.4.2 Pixel-Wise Analytical Reflectance Field Models

Another straightforward BTF decomposition avoiding high-dimensionality is per-view factorisation. This factorisation enables separate modelling of BTF images corresponding to one given view direction only, i.e., so called *Surface Reflectance Field* \mathcal{R}_v . This approach avoids problem caused by non-corresponding pixels in BTF images for different view directions as it is shown in Fig. 2.6.



Figure 2.6: Pixel-wise inconsistency for different view directions in BTF. The first image shows original BTF image compared with pixel-wisely non-corresponding rectified images for two different view positions, respectively.

Polynomial Texture Maps

In the approach of Malzbender et al. [72] the surface reflectance field is approximated by means of per-pixel polynomials. Therefore the authors call this method *Polynomial Texture Maps* (PTM). PTM models illumination dependence of individual pixels using following pixel-wise biquadratic

$$\begin{aligned} \mathcal{R}_v(r_1, r_2, i) \approx & a_0(r_1, r_2)u_{r_1}^2 + a_1(r_1, r_2)u_{r_2}^2 + a_2(r_1, r_2)u_{r_1}u_{r_2} + \\ & + a_3(r_1, r_2)u_{r_1} + a_4(r_1, r_2)u_{r_2} + a_5(r_1, r_2) , \end{aligned} \quad (2.12)$$

where u_{r_1}, u_{r_2} are projections of the normalised light vector into the local coordinate system r_1, r_2 . The six polynomial coefficients $a_0 - a_5$ are fitted in each pixel by means of singular value decomposition (SVD) [30].

The authors found this image-based method useful for colour pixel reconstruction in fixed-point hardware as well as for production of number of other effects such as anisotropic and Fresnel shading models or variable depth of focus. The method can be also used for contrast enhancement or for temporal scene compression. Moreover, a device for surface reflectance fields measurement is presented as well.

This method enables very fast rendering since its per-pixel costs for PTM evaluation depend only on 11 multiplications and 5 additions. However, as mentioned by the authors, the method assumes that the modelled surfaces are either diffuse or their specular contribution had been separated in the previous modelling step. This separation can be quite problematic for reflectance fields obtained as a BTF slice. For such a reflectance field the PTM exhibits considerable errors mainly for high grazing angles as shown in [78]. For BTF rendering this method requires six parametric images to be stored for each reflectance field \mathcal{R}_v and colour channel. Since this model is computed in pixel-wise manner for each reflectance field separately the final BTF rendering requires additional interpolation between individual view directions.

Reflectance Fields Using Lafortune Model

Another BTF compression approach based on factorisation into individual reflectance fields was introduced by Meseth et al. in [78]. The approach exploits, similarly to McAllister's [75] and Daubert's [19] work, a pixel-wise model based on Lafortune reflectance lobes of the following form

$$\mathcal{R}_v(r_1, r_2, i, v) \approx \rho_d, r_1, r_2 + \rho_s, r_1, r_2, k \sum_k^{n_l} [\omega_i^T \mathbf{D}_{r_1, r_2, k}]^{n_{r_1, r_2, k}} , \quad (2.13)$$

where ω_i represents illumination vector in local coordinate system, \mathbf{D} is diagonal matrix with model parameters and ρ_d, ρ_s, n_k represent diffuse, specular albedo and specular exponent, respectively.

Due to the expensive non-linear parameters fitting, the number of Lafortune lobes n_l is practically limited to three lobes. Unlike previously mentioned BTF models based on Lafortune lobes where each colour channel used individual set of fitting lobes this model uses lobes to compute luminance values only. These luminance values are further used to scale the albedo of individual colour channels. This arrangement reduces the number of parameters needed to be stored, but on the other hand decreases approximation accuracy.

Similarly to other reflectance field based BTF modelling approaches this method requires additional interpolation between individual view directions during BTF rendering to suppress disturbing edges. The model can be implemented in graphics hardware and ensures reasonable BTF approximation for less specular materials. However, for complex anisotropic and highly specular materials its performance is not satisfactory [79].

Spatial BRDF Factorisation Models

One of the first BRDF models based on factorisation techniques was presented by Kautz in [52]. This model exploits SVD for BRDF factorisation and produces two 2D factors instead of 4D BRDF. It can be exploited for pixel-wise BTF compression according to the formula

$$BTF(r_1, r_2, i, v) \approx \sum_{k=1}^{K_j} P_{k, r_1, r_2}(\pi_1(\omega_i, \omega_v)) Q_{k, r_1, r_2}(\pi_2(\omega_i, \omega_v)) , \quad (2.14)$$

where the functions π_1, π_2 are projection functions which map the 4D space determined by illumination and view directions to a 2D space. The factors $P_{j,k}$ and $Q_{j,k}$ are eventually stored in texture maps from which the ABRDF is easily reconstructed in graphics hardware.

The main limitation of this method is the decomposition into two factors only. This restriction has been overcome by McCool [76] using single term ABRDF approximation according to equation

$$BTF(r_1, r_2, i, v) \approx \prod_{j=1}^J P_{j,r_1,r_2}(\pi_j(\omega_i, \omega_v)) . \quad (2.15)$$

This approach exploits homomorphic factorisation which unlike the previous SVD based method generates a factorisation with only positive factors. This makes it more suitable for hardware implementation.

Even more efficient multiple term ABRDF approximation was suggested by Suykens et al. in [110]. This model decomposes ABRDF of each pixel into a product of three or more two-dimensional positive factors using a technique called *chained matrix factorisation* (CMF). This technique uses a sequence of matrix decompositions, each in a different parametrisation, allowing to obtain the multiple factor approximation as follows

$$BTF(r_1, r_2, i, v) \approx \prod_{j=1}^J \sum_{k=1}^{K_j} P_{j,k,r_1,r_2}(\pi_{j,1}(\omega_i, \omega_v)) Q_{j,k,r_1,r_2}(\pi_{j,2}(\omega_i, \omega_v)) . \quad (2.16)$$

This decomposition enables easier factor computation in comparison to previously discussed methods. As the authors claim, the CMF factors have lower dynamic range so their quantisation into 8-bits is much safer.

When using any of these factorisation approaches, an efficient BTF representation is usually obtained by factor clustering in form of a compact set of 2D textures. BRDF factorisation approaches enable BTF rendering at interactive frame-rates with a compression ratio of more than $\frac{1}{100}$.

2.4.3 Surface Light Field Modelling

Alike per-view factorisation also per-illumination factorisation is possible. This approach is often referred to as *Surface Light Fields* (SLF). It describes material reflectance for different view positions while the illumination position is fixed. These SLFs were first introduced and parametrised by Miller et al. in [81]. Large SLFs are very common in high-quality rendering systems, however, their size is a limiting factor which prevents fast applications in graphical hardware. Several SLF compression methods have been therefore introduced as, e.g., in [9, 89]. Since SLF modelling is not subject of this thesis we will do not discuss it in more detail.

2.5 BTF Compression Methods based on PCA

Although pixel-wise BRDF models discussed in the previous section can provide satisfactory quality of BTF approximation, they are often limited to certain kinds of real materials or can not handle all complex effects that appear due to using of ABRDF instead of assumed BRDF. These effects include sub-surface scattering from neighbouring pixels, using not perfectly directional light, inaccuracies in rectification or image filtering processing of original BTF data that violate Helmholtz reciprocity rule and result in

asymmetric BRDF. The reflectance models often fail to fit this complex data as a result of violating basic model assumptions.

These problems hinder the use of another category of BTF models represented by image statistics linear model *Principal Component Analysis* (PCA) typically computed by means of *Singular Value Decomposition* (SVD). Basic information can be found in [94] while a rigorous mathematical background is given in [30]. The SVD is linear algebra technique for solving a set of linear equations which provides the closest possible solution in a least-square sense. This method decomposes input matrix \mathbf{A} (even singular one), into matrices \mathbf{U} , \mathbf{V}^T containing orthonormal columns and rows, respectively, so called eigen-vectors (see 2.17). The non-negative diagonal matrix \mathbf{D} contains so called eigen-numbers or eigen-values. The size of the eigen-values determines importance of the corresponding eigen-vector for original data reconstruction.

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T \quad . \quad (2.17)$$

This technique enables high data compression since only a relatively small number of eigen-vectors have to be stored to achieve reasonable approximation error. This error is given by ratio of squared sum of preserved eigen-values to squared sum of all eigen-values. This technique can lead to significant compression of redundant input data and as such stands behind many compression algorithms applicable to BTF data as described in following sections.

2.5.1 Entire BTF Space Factorisation

A PCA based BTF factorisation approach was published by Koudelka et al. [59]. Individual BTF sub-images are arranged into vectors forming matrix \mathbf{A} of size $3MN3 \times n_v n_i$. The principal components are the eigen-vectors E_k of the symmetric matrix $\mathbf{A}\mathbf{A}^T$. However, the computation requirements for larger BTF image of resolution $M \times N$ are enormous. Computing eigen-images (i.e., the eigen-vector with image attributes) for non-homogeneous materials takes often several days. BTF reconstruction is stated by the following equation

$$BTF(r_1, r_2, i, v) \approx \sum_{k=1}^{n_c} \alpha_k(i, v) E_k(r_1, r_2) \quad . \quad (2.18)$$

To obtain good BTF approximation results the authors suggest the number of preserved principal components to be at least to $n_c = 150$. However, any fast computation of linear combination of such a number of eigen-images is impossible nowadays even on high-end hardware. Direct use so using of this method for any real-time BTF rendering application is currently considered impossible.

2.5.2 Multimodal BTF Space Factorisation

The image-based BTF compression method by Vasilescu and Terzopoulos [114] is based on N-mode SVD, being an extension of the conventional unimodal matrix SVD. The method assumes the individual BTF colour images to be ordered into vector features forming input BTF data tensor $\mathcal{B} \in \mathbf{R}^{(dNN) \times n_i \times n_v}$. N-mode SVD performs the following tensor decomposition

$$\mathcal{B} = \mathcal{Z} \times_1 \mathbf{U}_{tex} \times_2 \mathbf{U}_i \times_3 \mathbf{U}_v \quad , \quad (2.19)$$

where $\mathcal{A} \times_n M$ denotes mode- n product of tensor \mathcal{A} , \mathcal{Z} is core tensor that steers the interaction between the different modes, mode matrix \mathbf{U}_v spans the view space, its rows encode illumination and texel invariant representation for each of the different views, mode matrix \mathbf{U}_i spans the illumination space, its rows encode view and texel invariant representation for each of the different illumination and mode matrix \mathbf{U}_{tex} spans the texel space and are, the PCA eigen-vectors (i.e., eigen-images). The authors recommend following tensor-texture representation varying with view and illumination positions

$$\mathcal{T} = \mathcal{B} \times_2 \mathbf{U}_i \times_3 \mathbf{U}_v . \quad (2.20)$$

This representation requires to store more than ten times less parameters in comparison with unimodal SVD [59] and enables efficient BTF data compression that can be controlled independently with respect to viewing and illumination positions. The method enables BTF rendering for both planar and non-planar surfaces. Planar version follows

$$BTF(i, v) = \mathcal{T} \times_2 \mathbf{i}^T \times_3 \mathbf{v}^T , \quad (2.21)$$

where \mathbf{i} and \mathbf{v} are, respectively, view and illumination representation vectors corresponding to desired view and illumination directions.

Although, as reported by authors, RMS error of this method is higher in comparison with unimodal SVD with the same number of components, the visual performance is significantly better. However, this method involves high computational times for multi-modal SVD decomposition of BTF data with higher view and illumination angular resolution, e.g., [98]. This concerns offline decomposition as well as online rendering. Unfortunately, the authors have tested the method mostly on artificial BTF data; their tests on real University Bonn BTF were not compared with methods [98, 59] neither in terms of BTF compression rate ($\frac{1}{10} - \frac{1}{100}$?) nor offline and online time complexity. Due to the mentioned limitations this method is not suitable for fast hardware implementations of BTF rendering. However, its compact representation sets it as a good candidate for offline high quality rendering tasks.

2.5.3 Reflectance Field Factorisation

The huge factorisation demands of whole BTF space as well as the necessity of storing relatively high number of components which can not be evaluated in graphics hardware led to development of another approach by Sattler et al. [98]. The basic idea consists in computation of maximally $n_c = 16$ principal components for individual reflectance fields instead of the whole BTF space. Individual images corresponding to reflectance field \mathcal{R}_v are used as matrix input vectors to produce 16 eigen-images $E_{v,k}$ for each \mathcal{R}_v together with corresponding weights $\alpha_{v,k}$ by means of SVD. The reflectance field reconstruction formula is

$$\mathcal{R}_v(r_1, r_2, i) \approx \sum_{k=1}^{n_c} \alpha_{v,k}(i) E_{v,k}(r_1, r_2) . \quad (2.22)$$

The final BTF image is obtained by interpolation between the closest reflectance fields \mathcal{R}_v . Although this approach enables fast BTF rendering, the size of stored weight parameters and eigen-images is still quite high. To ensure reasonable results best compression ratio cannot exceed $\frac{1}{10}$.

2.5.4 BTF Segmentation and Local PCA

A BTF compression method well suited to contemporary graphical hardware was presented by Müller et al. in [82]. Unlike BTF factorisation approaches employing PCA as mentioned before, this method exploits the fact that high dimensional datasets, in this case BTF, show a locally linear behaviour. The authors propose a BTF compression algorithm based on combination of local PCA [49] computed in both BTF_{ABRDF} and BTF_{TEX} representations and vector quantisation. The BTF space is iteratively divided up to 32 clusters, each to be represented by means of local PCA. The squared eigen-texture/eigen-ABRDF reconstruction error is used as distance measure in the clustering process.

The described BTF factorisation can be stated as

$$BTF(r_1, r_2, i, v) \approx \sum_{k=1}^{n_c} \alpha_{m(r_1, r_2), k}(r_1, r_2) E_{m(r_1, r_2), k}(i, v) , \quad (2.23)$$

where $m(r_1, r_2)$ is a cluster index look-up table given by planar coordinates (r_1, r_2) , n_c is number of preserved principal components, α_k are PCA weights and E_k are either eigen-images or eigen-ABRDFs. The entire BTF reconstruction together with illumination and view interpolation is performed in graphics hardware enabling fast BTF rendering.

The authors claim the BTF_{ABRDF} arrangement more appropriate in terms of compression ratio which is approximately ten times higher than in the case of BTF_{TEX} arrangement. It follows from the observation that resembling material areas lead to natural clustering of similar ABRDF images and lower dimension of principal components in BTF_{ABRDF} arrangement than in BTF_{TEX} arrangement. This method provides BTF compression ratio of about $\frac{1}{100}$ while ensuring high reconstruction quality and rendering speed [79].

2.6 BTF Synthesis and Modelling Approaches

BTF models described in the previous section are intended mainly for efficient BTF compression enabling fast hardware supported BTF rendering. Most of these approaches preserve excellent visual quality of restored BTFs, but several significant disadvantages remain inherited in this kind of models. They can not produce larger BTF images than those present in original BTF and they offer only mild compression ratio. To enable BTF synthesis of large objects in VR by means of this kind of models it is necessary to employ additional BTF enlargement methods. These methods are usually based either on simple texture repetition with edge blending or on more or less sophisticated image tiling methods [24, 11, 62, 103]. Up to now there are only few image-based methods available that enable BTF synthesis of arbitrary resolution. Moreover, these methods are often too slow to enable fast BTF rendering. Till now there is no generally applicable generative BTF modelling approach available. All described BTF modelling methods require to store samples of BTF either in form of image tiles, textons or in form of some kind of their pixel-wise parametric representation.

2.6.1 BTF Synthesis from Parametric Tiles

The most common way of arbitrary resolution BTF synthesis is employing image tiling methods. The BTF synthesis approach based on image tiling was introduced in [124]. This

approach involves BTF compression based on polynomial texture maps [72]. Estimated resulted parametric images containing polynomial coefficients are subsequently enlarged by means of Efros’s image quilting algorithm [24]. Dong and Chantler [22] present a survey of several BTF synthesis approaches. The authors have tested an image based relighting method [21] based on BTF image reconstruction from several known BTF images according to Lambertian reflectance function, overdetermined photometric stereo based on SVD of 36 images, polynomial texture maps [72] and finally PCA analysis of all BTF images. BTF synthesis in all of these methods is accomplished again by means of tiling algorithm [24].

2.6.2 Copy and Paste Based BTF Synthesis

Copy and paste BTF synthesis is generally based on reproducing the BTF data from original BTF measurements to generate corresponding large BTF synthesis. These synthesis methods usually do not introduce any compression and are often unusable for fast real-time BTF synthesis applications.

One of the first BTF synthesis algorithms was developed by Liu et al.[70]. The method starts with range-map estimation using modified shape-from-shading algorithm based on [66]. The range-map is enlarged to the required size by means of accelerated non-parametric sampling [25]. This enlarged range-map is used to generate synthetic template image for the given view/illumination direction to be subsequently covered using albedo map.

For each illumination and view direction the nearest original BTF image is taken as a reference image. The final stage of BTF synthesis consists of copying image blocks from reference image that are similar to those in synthesised template image. Better results can be obtained by employing additional reference images obtained by means of weighted BTF image averaging. The authors tested the method performance on CURET BTF data [16, 96]. Two main drawbacks have shown up; the synthesis is too slow and the method does not guarantee any reasonable data compression.

Later Liu et al. [71] developed another BTF model similar to [82]. This method works with BTF_{ABRDF} data arrangement. The BTF synthesis on arbitrary surfaces is based on the smallest texture elements in BTF, so called *3D textons* introduced in [68]. To capture surface appearance at different illumination and viewing conditions the 3D textons are constructed using K-means clustering of appearance vectors, i.e., the vectors containing responses to a set of orientation and spatial-frequency selective linear filters applied at certain planar position in BTF. Thus only selected appearance vectors, those corresponding to textons in cluster centers, are stored. This leads to considerable data reduction.

In the following step a matrix is created so that its rows correspond to ABRDFs of previously selected textons. The matrix is then decomposed by means of SVD to obtain 2D geometry map $\alpha_i(\cdot)$ together with a set ($n_c = 5 - 40$) of eigen-ABRDFs E_i . BTF reconstruction from these eigen-ABRDFs is then described by the following equation

$$BTF(r_1, r_2, i, v) \approx \sum_{k=1}^{n_c} \alpha_k(r_1, r_2) E_k(i, v) . \quad (2.24)$$

During BTF rendering a surface is covered by ABRDFs restored from PCA components according to previously computed texton vocabulary assigning a texton label to each

pixel-ABRDF. The authors implemented this method in graphics hardware for fast BTF rendering on arbitrary surfaces with maximal BTF compression ratio about $\frac{1}{100}$.

Very similar BTF synthesis approach based on the 3D texton analysis was published in [111]. The paper describes 3D texton search algorithm in more detail and suggests BTF rendering based on surface synthesis from individual textons.

Another copy and paste BTF synthesis method was published by Neubeck et al. in [85]. The authors extend standard smart copy and paste smooth texture synthesis (e.g., [25, 24]) to BTF synthesis. The authors introduce their own BTF measurement setup and discuss the impact of projection plane position during rectification of BTF images on BTF smoothness and suggest as optimal the plain alignment that corresponds to maximal height of material. This considerably increases BTF smoothness during varying illumination and view conditions. The described BTF synthesis method does not copy all BTF pixel values, but only the values corresponding to individual viewing position which produces novel synthetic BTF pixel reflectance values. For this purpose the authors match the original input frontal view with synthesised support view using multiscale dynamic weighting scheme to obtain correct BTF pixel values. The method starts with single view synthesis and continues with sequential synthesis of other views. Although this method produces high quality BTF synthesis, it does not solve the issue of BTF compression and it is relatively slow with problematic hardware implementation.

2.7 Alternative BTF Modelling and Classification Methods

2.7.1 BTF Classification Based on Image Descriptive Histograms

Cula and Dana [14, 15] present a method for BTF representation and classification (CURET [96]). The authors modify the method presented in [68], so that instead of computing filter bank responses for BTF pixel during varying view and illumination directions K-means clustering of filter bank responses for individual BTF images is performed separately to obtain so-called image texton library. This representation is favourable since it does not require spatial registration of individual BTF images. The distributions of texton library labels over individual images are subsequently approximated by the texton histograms which forms BTF dataset representation. PCA is employed to reduce the dimensionality of these texton histograms. The resulting representation in the eigen-space is the reference manifold indexed by illumination and view positions. When novel BTF measurement (material) is to be classified, its image texton histogram is projected onto the universal eigen-space and the closest manifold is found corresponding to BTF which is the class of the query.

2.7.2 BTF Compression Based on Image Histograms

Exploiting of individual BTF image (CURET [96]) histograms for purpose of estimation of material parameters, verification of texture models, BTF classification, geometry estimation and texture rendering is introduced by van Ginneken in [113]. According to this method BTF images corresponding to the same view position are generated by means of a histogram matching technique [31] using only one original BTF image, its histogram and histograms corresponding to other BTF images taken from the same view position. This technique enables relatively high BTF compression since only one image and sev-

eral histograms have to be stored for satisfactory approximation of remaining BTFs for many kinds of real-world materials. The authors found this method significantly better in comparison with simple image brightness adjustment according to BRDF data.

2.7.3 Prediction of Lambertian Rough Texture Illumination Characteristics

Directional characteristics of rough textures was also studied by Chantler in [8] with conclusion that illumination during image capture can act as a directional texture filter and that directional characteristics of such a texture are not just a function of surface relief but are also affected by the illumination angles. The author presents a simple texture model based on a linearised version of Lambertian law, applied to a frequency domain representation of the surface texture, which successfully predicts the directional filtering effect for closely Lambertian materials.

2.7.4 Image Based Relighting

Another alternative approach of rough texture modelling based on Lambertian reflectance function was presented by Dong and Chantler in [21]. The proposed method - *Image-Based Relighting* (IBR) uses a specific linear combination of BTF images to generate new one with different illumination conditions. If Lambertian reflectance law is assumed, the following reflectance function is valid

$$I_{(\phi_i, \theta_i)}(x, y) = \lambda \rho \frac{-p \cos \phi_i \sin \theta_i - q \sin \phi_i \cos \theta_i + \cos \theta_i}{\sqrt{p^2 + q^2 + 1}} \quad (2.25)$$

where I is the intensity of image pixel at position (x, y) , λ is the incident intensity to the surface, ρ is the albedo value of the Lambertian reflection, (θ_i, ϕ_i) are elevation and azimuthal illumination angles and (p, q) are the partial derivatives of the surface in x and y direction.

According to these assumptions the interpolating formula is a linear sum of three images captured in three different illumination directions ($\phi_i = 0^\circ, 90^\circ, 180^\circ$ when $\theta_i = 60^\circ$):

$$\begin{aligned} I_{(\phi_i, \theta_i)}(x, y) = & \left(\frac{\cos \phi_i \sin \theta_i}{2 \sin 60^\circ} - \frac{\sin \phi_i \sin \theta_i}{2 \sin 60^\circ} + \frac{\cos \theta_i}{2 \cos 60^\circ} \right) \cdot I_{(0^\circ, 60^\circ)}(x, y) \\ & + \frac{\sin \phi_i \sin \theta_i}{\sin 60^\circ} \cdot I_{(90^\circ, 60^\circ)}(x, y) \\ & + \left(\frac{\cos \theta_i}{2 \cos 60^\circ} - \frac{\cos \phi_i \sin \theta_i}{2 \sin 60^\circ} - \frac{\sin \phi_i \sin \theta_i}{2 \sin 60^\circ} \right) \cdot I_{(180^\circ, 60^\circ)}(x, y) \quad (2.26) \end{aligned}$$

Fig. 2.7 shows the three original captured images as well as two interpolated images obtained by applying the formula to each pixel. The described method is useful for generation of new views not included in a sparse sampled BTF dataset, while assuming that the material more or less fulfils the Lambertian law. Nevertheless, this interpolation method does not solve BTF modelling or compression problems. To compress one reflectance field this method requires to store relatively high number of BTF images. Moreover, it does not solve the BTF enlargement problem so some additional image tiling method is necessary.

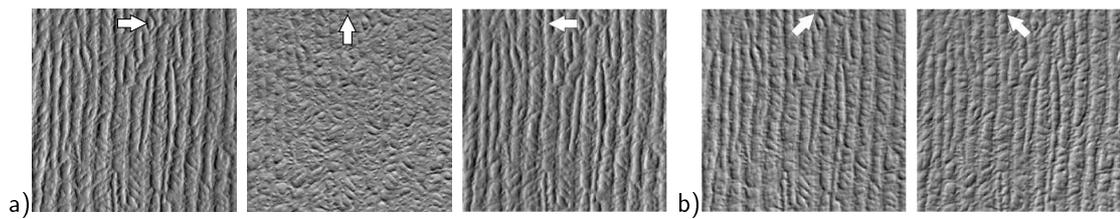


Figure 2.7: Original BTF images with illumination elevation angle $\theta_i = 60^\circ$ and azimuthal angles $\phi_i = 0^\circ$, $\phi_i = 90^\circ$, $\phi_i = 180^\circ$ during constant viewing position (a), generated images based on IBR formula with illumination elevation angle $\theta_i = 60^\circ$ and azimuthal angles $\phi_i = 45^\circ$, $\phi_i = 135^\circ$ during constant viewing position.

Chapter 3

BTF Databases

Representation of real-world materials by means of BTF is relatively new approach, however, its importance constantly increases. Due to a complex measurement procedure involving accurate data acquisition setup and long acquisition and post-processing time there are only several BTF databases available comprising tens of different materials. Various attributes of three main publicly available BTF databases are subject of this chapter.

3.1 Columbia-Utrecht Reflectance and Texture Database

Dana et al. [16] acquired one of the first real BTF measurements for 61 different natural and man-made materials - *Columbia-Utrecht Reflectance and Texture Database - CURET*. This database contains materials as fabric, leather, plaster, paper, pebbles, velvet, feather, leaf, human skin and several others. On image Fig. 3.1-a is illustrated measurement setup of CURET BTF database. During measurement the material sample position is adjusted by robot arm and the 3-CCD colour camera moves around the material sample in seven different positions (see Fig. 3.1-b) while the light position is fixed. Vertices on hemisphere in Fig. 3.1-b correspond to possible orientations of material sample surface normals and images are acquired for the subset of sample orientations which are visible and illuminated. As a result each datasets comprises of 205 BTF images for isotropic material samples and 410 BTF images for anisotropic material samples where extra measurements are obtained by sample rotation by 45° or 90° . For fixed camera position there is from 1 to 13 different BTF images, i.e., from 2 to 26 for anisotropic samples. Size of raw BTF images is 640×480 pixels with 24bits per pixel. However, exploitable size of such BTF images is not more than 500×400 pixels and used video camera had low dynamic range and, moreover, produced visible discontinuity, interlace and moire artifacts. The major drawback of CURET BTF data is absence of any points of correspondence so performing of correct pixel-wise rectification according to image contents is impossible. Nevertheless CURET BTF samples can be useful for testing purposes where dense angles quantisation and images registration is not necessary and this BTF data are available via project web site [96].

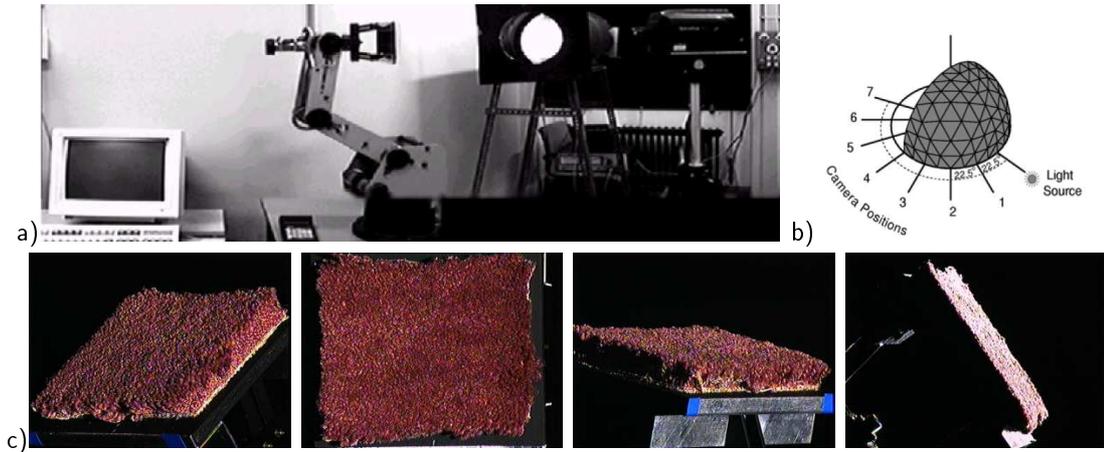


Figure 3.1: (a) BTF CURET measurement setup, (b) Camera positions according to material sample and illumination position, (c) Examples of CURET BTF images of fabric.

3.2 Yale University BTF Database

The Yale university BTF database have been created by Koudelka et al. [59]. This database contains BTFs of more than ten different rough materials: carpet, fur, gravel, lichen, moss, sponge, velvet and even lego brick. BTF images were acquired by a static digital video camera (CANON XL-1) while the illumination (LED array) was moving in a robot arm with two degrees of freedom over hemisphere above the surface of texture sample. The texture sample was mounted on pan/tilt head providing two degrees of freedom in viewpoint as well. Overall measurement setup is depicted in Fig. 3.2 Each such a BTF contains 90 possible view positions and 120 possible illumination positions which results in 10 000 BTF images per sample. Individual BTF images (Fig. 3.3) can be rectified using marks on sample support. Obtainable resolution of rectified BTF measurement is 192×192 pixels. The resolution of the BTF measurements is unsatisfactory for statistical models training, mainly for materials with slightly non-homogeneous appearance containing lower spatial frequencies. On the other hand, an advantage of this dataset is high angular resolution in both view and illumination angles. Due to use of video camera with relatively low resolution the measurement time is about 10 hours. The Yale University BTF database is available for research purposes upon request to author [18].

3.3 Bonn University BTF Database

An extension of the CURET BTF measurement setup have been constructed on Bonn university [98] and is illustrated in Fig. 3.4. This measurement device consists of a robot for material sample setting, a light source, a digital camera mounted on a rail leading around the robot and a computer for robot arm, rail and camera control. The light source with a HMI (Hydrargyrum Media Arc Length Iodide) bulb (Broncolor HMI F575) is equipped with the Fresnel lens to generate parallel light rays and should guarantee homogeneous illumination of the sample and constant emission over whole measurement period. The six-megapixels digital camera Kodak DCS Pro 14N is fully remote controlled. This measurement setup enable to measure every combination of view and illumination

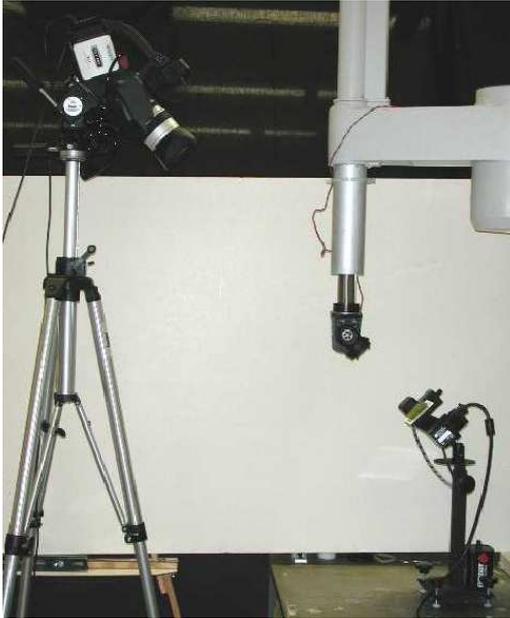


Figure 3.2: BTF measurement setup of Yale University.

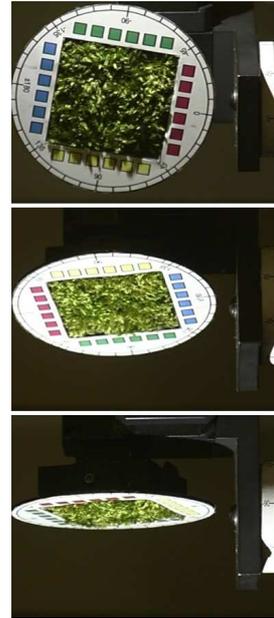


Figure 3.3: Acquired BTF images of Yale University (material: *moss*).

direction over the material sample. This is useful mainly for samples with anisotropic reflectance properties. Acquired raw images are of size 4300×3000 pixels in 12-bit RGB format. The measurement time for one material using described setup is about 14 hours and most of it takes data transfer from the camera to the computer.

Sample holder contains features for easy raw image rectification (Fig. 3.6-a). The rectification procedure consists of two steps. The first is detection of sample holder corners as intersections of four main outlines (Fig. 3.6-b) in Hough space illustrated in Fig. 3.6-c. In second step is computed a homomorphic projection matrix from four points corresponding to outlines intersection of four points corresponding to corners positions of rectified image. By means of this matrix the raw BTF image is transformed into a rectified head-on position ($\theta_v = 0^\circ, \phi_v = 0^\circ$) as it is shown in Fig. 3.6-d. Rectification error is less than five pixels. The size of material sample is $10 \times 10 \text{ cm}^2$ and corresponding rectified BTF images have resolution of 256×256 pixels. The Bonn university BTF datasets contains 6561 images per texture sample which corresponds to all combinations of 81 view and 81 illumination directions. Each BTF dataset takes up about 1.2GB. A subset of these BTF datasets are publicly available via project web page [17].

In the framework of EU project IST-2001-34744 RealReflect we were provided with even more accurate Bonn University BTF measurements with a maximal resolution of 800×800 pixels. Thanks to improved images post-processing and rectification algorithms the rectification error of these measurements is less than three pixels, while all other parameters and the measurement setup remains unchanged.

At this point we have to mention that for all experiments with BTF data in this thesis the extended Bonn university BTF datasets were used of size 800×800 pixels introducing 81 different view and illumination directions.

To improve correspondence of the individual BTF images as well as a speed up whole



Figure 3.4: BTF measurement setup of Bonn University.

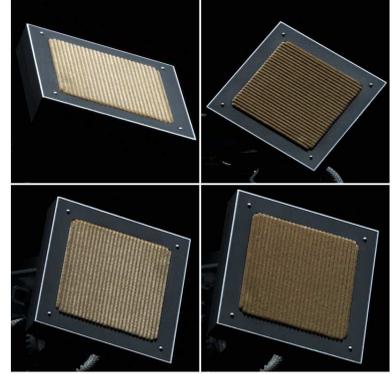


Figure 3.5: Acquired BTF images of Bonn University (material: *corduroy*).

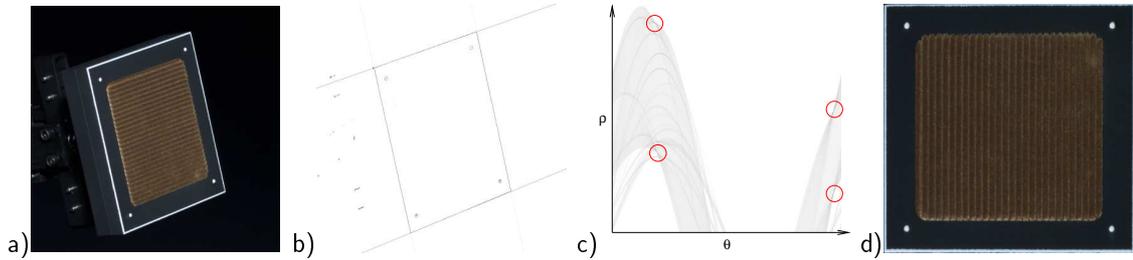


Figure 3.6: BTF image rectification: (a) measured image, (b) binary image for Hough transform, (c) Hough transform accumulator with four main lines corresponding to sample holder edges, (d) final rectified image.

the BTF measurement procedure Bonn University proposes [83] a new BTF measurement device based on array of 151 digital still cameras mounted on the hemispherical chassis above the material sample. As light sources will be used the built-in flashes in these cameras. Angular resolution of this planned measurement setup depends only on a number of cameras. The biggest advantage of this system is that it does not involve any moving parts, so the acquired images can be finally rectified at sub-pixel precision. System will enable fast measurement as individual cameras can take images subsequently and do not need to wait on a previous image data transfer. The time consuming post-processing should be much faster and more accurate as the individual images are rectified and corrected according to exactly known parameters and positions of the individual cameras. The spatial resolution of this setup would be up to 280DPI resulting to rectified BTF textures of size 1024×1024 pixels. The measurement time is expected to be less than one hour.

3.4 Comparison of BTF Databases' Parameters

The following table summarises properties of individual BTF databases.

parameter	BTF database			
	CURet	Yale	Bonn	Bonn ext.
Number of materials	61	~17	5	~12
Raw BTF images resolution [pixels]	640×480	480×360	3032×2008	3032×2008
Rectified BTF images resolution [pixels]	500×400	192×192	256×256	800×800
Number of view positions	7	90	81	81
Number of illumination positions	max. 50	120	81	81
Number of BTF images / material	205	10 000	6561	6561
Material sample size [cm]	10×12	< 10 ²	10×10	10×10
Average size of rectified BTF dataset in PNG format	~100MB	~700MB	~700MB	~5GB
Rectification accuracy [pixels]	–	?	~5	~2
Camera type [V ideo / S till]	V	V	S	S
Movement of [S ample / C amera / L ight] during measurement	S,C	S,L	S,C	S,C
Raw data publicly available	yes	yes	no	no
Rectified data publicly available	no	yes	yes	no

Chapter 4

BTF Rendering

The BTF is relatively new approach for material reflectance properties description which is only currently possible due to recent progress in computer technology. However, even the most sophisticated graphics hardware have limited capability of fast raw BTF data rendering therefore employing of BTF modelling methods is inevitable. Additionally there are neither any standards for BTF rendering and modelling nor BTF rendering support available implemented in widely used 3D rendering software packages. This is caused by both huge dimensionality of BTF data and the principal dissimilarity of individual BTF modelling methods so the BTF rendering is usually inevitably tailored to a needs of these individual methods. Therefore in the scope of this thesis we have implemented a BTF rendering scheme, for proposed BTF modelling methods, based on the OpenGL library for one point-light source.

During implementation several technical problems occurred and this section describes the most important of them as well as their proposed solution: the BTF mapping on a 3D object surface, the BTF interpolation for given view and illumination directions and surface height simulation techniques.

4.1 BTF Mapping on a 3D Object

A texture mapping lays the texture onto an object in a VR scene. During this process an image is applied to a polygon (or some other surface facet) of an object shape surface by assigning texture coordinates to the polygon's vertices. These coordinates index a texture image, and are interpolated across the polygon to determine, at each of the polygon's pixels, a texture image value. The result is that some portion of the texture image is mapped onto the polygon when the polygon is viewed on the screen.

Additionally to geometrical mapping in the case of BTF (assuming homogeneous illumination for each polygon) we have to compute the two pairs of polar angles (θ_i, ϕ_i) and (θ_v, ϕ_v) for each polygon in texture coordinate system according to actual illumination and camera direction as is illustrated in Fig. 4.1. These angles are necessary for identification which BTF image from the whole BTF dataset should be synthesised and subsequently mapped on the polygon. For this the 3D world coordinates of polygon vertices $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ are known as well as their corresponding 2D texture coordinates $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$.

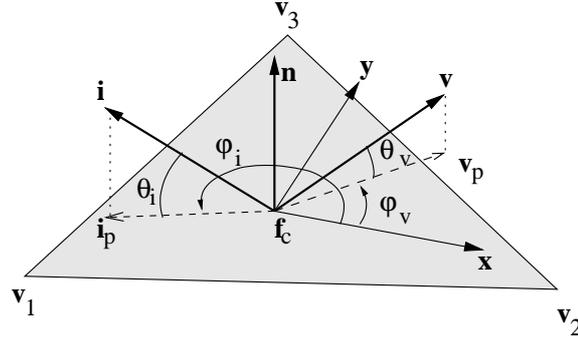


Figure 4.1: The local coordinate system of object surface polygon specified by angles $\theta_i, \phi_i, \theta_v, \phi_v$.

As a first step the polygon normal vector is computed using equation (4.1).

$$\mathbf{n} = \frac{(\mathbf{v}_1 - \mathbf{v}_2) \times (\mathbf{v}_3 - \mathbf{v}_2)}{\|\mathbf{n}\|} . \quad (4.1)$$

The center of the polygon (triangle) is

$$\mathbf{f}_c = \frac{\mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3}{3} . \quad (4.2)$$

Vectors pointing from the polygon center \mathbf{f}_c to the illumination and camera positions in world coordinates $\mathbf{i}_w, \mathbf{v}_w$ are

$$\mathbf{i} = \frac{\mathbf{i}_w - \mathbf{f}_c}{\|\mathbf{i}\|} \quad \mathbf{v} = \frac{\mathbf{v}_w - \mathbf{f}_c}{\|\mathbf{v}\|} . \quad (4.3)$$

The elevation angles θ_v and θ_i are then

$$\theta_i = \arccos \frac{\mathbf{n} \cdot \mathbf{i}}{\|\mathbf{n}\| \|\mathbf{i}\|} \quad \theta_v = \arccos \frac{\mathbf{n} \cdot \mathbf{v}}{\|\mathbf{n}\| \|\mathbf{v}\|} . \quad (4.4)$$

To determine azimuthal angles ϕ_i and ϕ_v we need to compute the projection of illumination and view vectors \mathbf{i}, \mathbf{v} to a polygon plane (4.5):

$$\mathbf{i}_p = \mathbf{n} \times (\mathbf{n} \times \mathbf{i}) \quad \mathbf{v}_p = \mathbf{n} \times (\mathbf{n} \times \mathbf{v}) \quad (4.5)$$

and to obtain vector \mathbf{x} giving x-axis of a local texture coordinate system. Vector \mathbf{x} is projection of a 2D vector $(1, 0)$ to 3D space. Our task is to compute the 3×3 projection matrix \mathbf{M} having set of three 2D texture coordinates $(t_{u1}, t_{v1}), (t_{u2}, t_{v2}), (t_{u3}, t_{v3})$ and corresponding 3D vertices coordinates $(v_{x1}, v_{y1}, v_{z1}), (v_{x2}, v_{y2}, v_{z2}), (v_{x3}, v_{y3}, v_{z3})$ for a given polygon according to

$$\mathbf{M} \begin{bmatrix} t_u \\ t_v \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ a_5 & a_6 & b_3 \end{bmatrix} \begin{bmatrix} t_u \\ t_v \\ 1 \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} . \quad (4.6)$$

Parameters of the matrix M can be determined as the Gauss elimination [94] of matrix equation:

$$\mathbf{AM}^T = \mathbf{B} \quad , \text{ i.e., } \begin{bmatrix} t_{u1} & t_{v1} & 1 \\ t_{u2} & t_{v2} & 1 \\ t_{u3} & t_{v3} & 1 \end{bmatrix} \begin{bmatrix} a_1 & a_3 & a_5 \\ a_2 & a_4 & a_6 \\ b_1 & b_2 & b_3 \end{bmatrix} = \begin{bmatrix} v_{x1} & v_{y1} & v_{z1} \\ v_{x2} & v_{y2} & v_{z2} \\ v_{x3} & v_{y3} & v_{z3} \end{bmatrix} \quad (4.7)$$

where \mathbf{M} is the wanted solution. In the case that we need only projection of vector $\mathbf{x}_{2D} = (1, 0)$ – it means only difference of 3D coordinates of points A(0,0), B(1,0) using the equation (4.6) with the result

$$\mathbf{x} = \begin{bmatrix} x_x \\ x_y \\ x_z \end{bmatrix} = \begin{bmatrix} x_B - x_A \\ y_B - y_A \\ z_B - z_A \end{bmatrix} = \begin{bmatrix} 1 \cdot a_1 + b_1 - b_1 \\ 1 \cdot a_3 + b_2 - b_2 \\ 1 \cdot a_5 + b_3 - b_3 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_3 \\ a_5 \end{bmatrix}. \quad (4.8)$$

Thus we see that only three parameters have to be computed, which can be done analytically with the following result

$$\begin{aligned} x_x = a_1 &= \frac{v_{x3}(t_{v2}-t_{v1})-v_{x2}(t_{v3}-t_{v1})-v_{x1}(t_{v2}-t_{v3})}{t_{u3}(t_{v2}-t_{v1})-t_{u1}(t_{v3}-t_{v1})-t_{u1}(t_{v2}-t_{v3})} \\ x_y = a_3 &= \frac{v_{y3}(t_{v2}-t_{v1})-v_{y2}(t_{v3}-t_{v1})-v_{y1}(t_{v2}-t_{v3})}{t_{u3}(t_{v2}-t_{v1})-t_{u1}(t_{v3}-t_{v1})-t_{u1}(t_{v2}-t_{v3})} \\ x_z = a_5 &= \frac{v_{z3}(t_{v2}-t_{v1})-v_{z2}(t_{v3}-t_{v1})-v_{z1}(t_{v2}-t_{v3})}{t_{u3}(t_{v2}-t_{v1})-t_{u1}(t_{v3}-t_{v1})-t_{u1}(t_{v2}-t_{v3})}. \end{aligned} \quad (4.9)$$

The azimuthal angles are then computed as angles between the vector x and projections $\mathbf{i}_p, \mathbf{v}_p$ of illumination and view vectors onto the polygon. To obtain information of computed angles greater than 180° we need to introduce auxiliary vector $\mathbf{y} = |\mathbf{n} \times \mathbf{x}|$. The azimuthal angles ϕ_i and ϕ_v are computed as follows

$$\begin{aligned} \phi_i &= \begin{cases} 2\pi - \arccos \frac{\mathbf{i}_p \cdot \mathbf{x}}{\|\mathbf{i}_p\| \|\mathbf{x}\|} & \text{if } \arccos \frac{\mathbf{i}_p \cdot \mathbf{y}}{\|\mathbf{i}_p\| \|\mathbf{y}\|} > \frac{\pi}{2} \\ \arccos \frac{\mathbf{i}_p \cdot \mathbf{x}}{\|\mathbf{i}_p\| \|\mathbf{x}\|} & \text{else} \end{cases} \\ \phi_v &= \begin{cases} 2\pi - \arccos \frac{\mathbf{v}_p \cdot \mathbf{x}}{\|\mathbf{v}_p\| \|\mathbf{x}\|} & \text{if } \arccos \frac{\mathbf{v}_p \cdot \mathbf{y}}{\|\mathbf{v}_p\| \|\mathbf{y}\|} > \frac{\pi}{2} \\ \arccos \frac{\mathbf{v}_p \cdot \mathbf{x}}{\|\mathbf{v}_p\| \|\mathbf{x}\|} & \text{else} \end{cases}. \end{aligned} \quad (4.10)$$

All this computation can be done in advance and stored in a cube-map texture where a vector to illumination/camera is used as an index in the 4-dimensional images covering cube sides storing the already precomputed values. This enables to considerably speed up the whole process of BTF mapping. For more details see [101].

4.1.1 Additional Texture Mapping Tasks

Constrained Texture Size

The texture is normally stored as a sampled array of constrained size, so a large continuous image must first be reconstructed from the samples/tiles. The repeatable tiles published in [103] were computed from the original (see Section 8.3.1) BTF image using the fast and adjustable sub-optimal path search algorithm for finding the minimum error boundaries between the overlapping images. Another image tiling approaches are mentioned in [24, 11, 62]. Alternative solution of BTF enlargement using probabilistic texture models approximating real measurements is proposed in Chapter 7. The presented probabilistic models enable to produce the BTF images of an arbitrary size. The tiling approach was used to obtain textured objects for comparison with methods developed in the thesis.

Aliasing

The texture image must be warped to match any distortion (caused, perhaps, by perspective) in the projected object being displayed. Then this warped image is filtered to remove

high-frequency components that would lead to aliasing in the final step. This problem is usually solved using mip mapping approach [119, 101] which stores pyramid of images of the same texture with subsequently decreasing resolution. During the texture mapping a certain texture image from the pyramid is taken according to distance of observer from the object polygon being mapped. This suppress aliasing artifacts and save computation time too because there is not necessary to have the same level of detail mapped on both closed and distant polygons of the object.

Due to time and memory complexity of BTF synthesis and mapping we do not deal with mip mapping in the scope of this thesis at all. Thus all the polygons of rendered objects are covered with BTF with the same level of details.

However, all of the mathematical texture models presented in Chapter 7 uses the Gaussian-Laplacian pyramid whose individual image levels can be stored and used as mip map images.

4.2 BTF Interpolation

For purpose of the BTF correct visualisation on objects in VR scene, there is necessary to perform interpolation between individual view and illumination directions with respect to original quantisation step of the measured BTF. In the case that only the closest BTF image is chosen for texture mapping on actual polygon there became visible seams of different colour / brightness on object surface. These artifacts can be suppressed using interpolation between three or more view / illumination position from BTF database which are closest to actual polygon's view and illumination directions. Next problem is computation of optimal weights of these closest view/illumination directions. In computer graphics are for this aim commonly used barycentric coordinates which were derived by Möbius in 1827 [13, 26]. Using this technique the weights of three closest triangle vertices ($\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$) (view/illumination positions) corresponding to point \mathbf{p} are computed as ratio of masses of particular sub-triangle and whole triangle. The relative magnitude of the coordinates (weights) corresponds to area ratios in triangle depicted in Fig. 4.2.

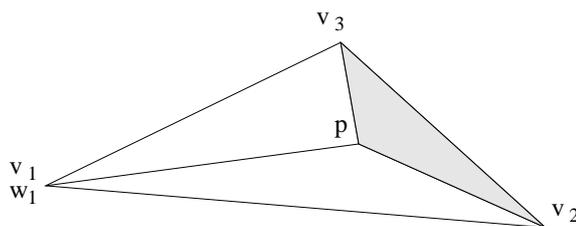


Figure 4.2: The barycentric coordinates in triangle.

Then barycentric weights are then computed as

$$w_1 = \frac{\Delta(\mathbf{p}, \mathbf{v}_2, \mathbf{v}_3)}{\Delta(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)} \quad w_2 = \frac{\Delta(\mathbf{p}, \mathbf{v}_1, \mathbf{v}_3)}{\Delta(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)} \quad w_3 = \frac{\Delta(\mathbf{p}, \mathbf{v}_1, \mathbf{v}_2)}{\Delta(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)} \quad (4.11)$$

where $\Delta(a, b, c)$ means area of triangle a, b, c . The point \mathbf{p} is interpolated from closest three points using equation

$$\mathbf{p} = w_1 \mathbf{v}_1 + w_2 \mathbf{v}_2 + w_3 \mathbf{v}_3 \quad (4.12)$$

while following condition is fulfilled

$$w_1 + w_2 + w_3 = 1 . \quad (4.13)$$

From this is obvious that following scheme can be simply utilised for BTF interpolation if the three closest measurements are known. These measurements can be obtained as three closest vectors to actual camera/light positions. Now having these three closest view/illumination positions the barycentric weights (w_i, w_v) are computed for the closest view and illumination directions. To determine the BTF image corresponding to these directions there is necessary to specify both view and illumination directions at a time. By permutation of three closest view and illumination positions we obtain nine pairs of view and illumination position when each of them specify one of BTF measurement. The texture finally mapped on each polygon is then linear combination of these nine BTF images while the final weight factors for each such a image are obtained using

$$w_{i,v} = w_i w_v . \quad (4.14)$$

Implementation of barycentric coordinates is quite straightforward when assuming that each triangle in 3D can be represented using two vectors and the area of this triangle is proportional to vector norm of a cross-product of these two vectors. So the barycentric weights from equation (4.11) are finally computed as

$$\begin{aligned} w_1 &= \frac{\|(\mathbf{v}_2 - \mathbf{p}) \times (\mathbf{v}_3 - \mathbf{v}_2)\|}{\|(\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_2)\|} \\ w_2 &= \frac{\|(\mathbf{v}_3 - \mathbf{p}) \times (\mathbf{v}_1 - \mathbf{v}_3)\|}{\|(\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_2)\|} \\ w_3 &= \frac{\|(\mathbf{v}_1 - \mathbf{p}) \times (\mathbf{v}_2 - \mathbf{v}_1)\|}{\|(\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_2)\|} . \end{aligned} \quad (4.15)$$

The barycentric coordinates have the following interesting properties:

- If w_1, w_2 and w_3 are all greater than zero, \mathbf{p} is strictly inside the triangle.
- If $w_i = 0$ and the other two coordinates are positive, \mathbf{p} lies on the edge opposite \mathbf{v}_i .
- If $w_i = 0$ and $w_j = 0$, \mathbf{p} lies on \mathbf{v}_k .
- If $w_i < 0$, \mathbf{p} lies outside the edge opposite \mathbf{v}_i .

There is a lot of research work, mainly in computer graphics, concerning barycentric coordinates. The article [80] describes generalisation of barycentric coordinates to any irregular polygon (not only triangle).

The same interpolation scheme can be exploited also for cluster interpolation within BTF model introduced in Chapter 7. In this case there are not used original BTF images for interpolation but synthesised subspace images instead of them specified by means of a cluster index file.

4.3 Surface Height Simulation

Rough material surfaces can be in certain conditions efficiently approximated by means of methods exploiting surface height information. Two main approaches of rough surface

representation based on height information were developed in computer graphics. The first one is relative computationally cheap method proposed by Blinn in [7] called *bump mapping*. The second one, *displacement mapping* is more computationally demanding method modelling texture surface by means of direct move of surface vertices according to the range-map. Both of these methods are supported by contemporary graphics hardware and thus their fast implementations and different modifications are available [122, 118, 56, 90, 92].

4.3.1 Bump Mapping

Bump mapping [7] is a normal-perturbation rendering technique for simulating lighting effects caused by patterned irregularities on otherwise locally smooth surfaces. This technique adds more realism to synthetic images without adding a lot of geometry. Bump mapping adds per-pixel surface relief shading and increasing the apparent complexity of the surface, however its real geometry remains unchanged as is illustrated in Fig. 4.3 [77].

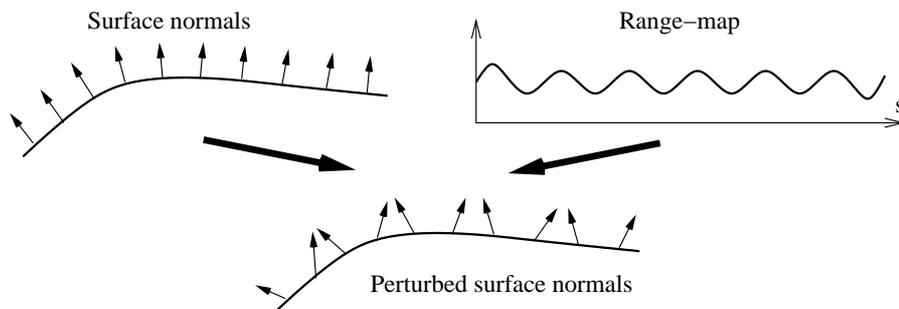


Figure 4.3: Principle of bump mapping.

The colour of a surface is determined by the angle between the normal vector \mathbf{N} of that surface and the light vector \mathbf{L} . On a flat surface the normal vector \mathbf{N} is the same everywhere on that surface, so the colour of that surface will be the same everywhere, however, if the normal vector is perturbed at various points on that surface, it would yield areas that are darker or lighter, thereby creating the perception that parts of the surface are raised or lowered.

The information about normals perturbation can be stored in a texture map so called range-map or bump-map which is an array of values that represent an object's height variations. The material range-map can be measured and estimated by means of variety methods and most common of them are described in Chapter 5.

The normals perturbation can be performed by means of the first derivative of the range-map values which can be efficiently found by the following process (see 1D example in Fig. 4.4):

1. Render the range-map image as a texture (A).
2. Shift the texture coordinates at the vertices towards the light (B).
3. Re-render the range-map as a texture, subtracting from the first image (A-B).

In order to perform accurate texture shift, the light source direction \mathbf{L} must be rotated into a tangent space. The tangent space has three perpendicular axes, \mathbf{T} , \mathbf{B} and \mathbf{N} . \mathbf{T} ,

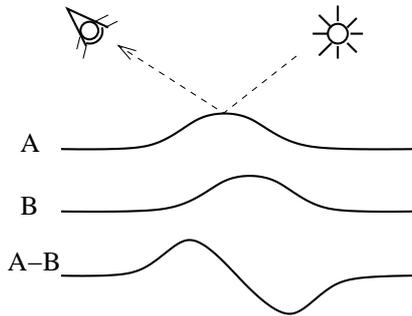


Figure 4.4: Subtraction of bump shifted towards light source from original bump produces visual perception of lit bump.

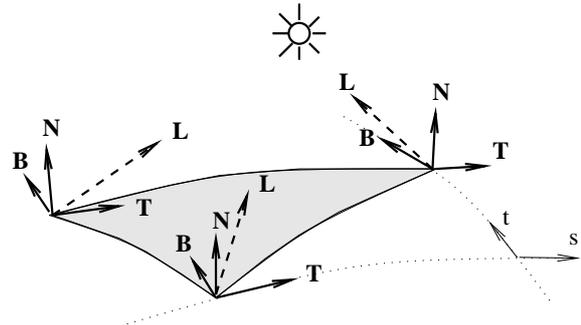


Figure 4.5: The tangent space defined on polygonal 3D surface.

the tangent vector, is parallel to the direction of increasing texture coordinates s or t on a parametric surface (see Fig. 4.5). \mathbf{N} , the normal vector, is perpendicular to the local surface. \mathbf{B} , the bi-normal, is perpendicular to both \mathbf{N} and \mathbf{T} , and like \mathbf{T} , also lies on the surface. They can be thought of as forming a coordinate system that is attached to the surface, keeping the \mathbf{T} and \mathbf{B} vectors pointing along the tangent of the surface, and pointing away. If the surface is curved, the tangent space orientation changes at every point on the surface.

In order to create a tangent space for a surface, it must be mapped parametrically. But since this technique requires applying a 2D texture map to the surface, the object must already be parametrically mapped in texture coordinates s and t . The only requirement for well working bump mapping is consistent parametric mapping on the polygon.

A vector pointing to the light source must be rotated into tangent space at each vertex of the polygon. To find the tangent space vectors at a vertex, use the vertex normal for \mathbf{N} , find the tangent axis by finding the vector direction of increasing s in the object's coordinate system. Find \mathbf{B} by computing the cross-product of \mathbf{N} and \mathbf{T} . The normalised values of these vectors can be used to create a rotation matrix:

$$\mathbf{R} = \begin{bmatrix} T_x & T_y & T_z & 0 \\ B_x & B_y & B_z & 0 \\ N_x & N_y & N_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.16)$$

The matrix (4.16) rotates the \mathbf{T} vector, defined in object space, into the x axis of tangent space, the \mathbf{B} vector into the y axis, and the normal vector \mathbf{N} into the z axis. It rotates a vector from object space into tangent space. For all non-planar surfaces, this matrix will differ at each vertex of the polygon.

Now we can apply this matrix to the light direction vector \mathbf{L} , transforming it into tangent space at each vertex. Then the transformed x and y components of the light vector are used to shift the texture coordinates at the vertex. The overall description of three pass bump mapping algorithm is given in Alg. 1.

Example of bump mapping for *leather02* smooth texture on sphere with additional diffuse lighting is depicted in Fig. 4.6-left.

Although this technique approximates the surface efficiently, there are following limitations to its accuracy [77]:

Algorithm 1: Bump mapping algorithm

1. Render the polygon with the range-map textured on it (*pass 1*).
 2. Find \mathbf{N} , \mathbf{T} and \mathbf{B} at each vertex according to Fig. 4.5.
 3. Use these vectors to create a rotation matrix \mathbf{R} (4.16).
 4. Use the matrix \mathbf{R} to rotate the light vector into tangent space ($\mathbf{L}' = \mathbf{R}\mathbf{L}$).
 5. Use the rotated L'_x and L'_y components of \mathbf{L}' to shift the s and t texture coordinates at each polygon vertex in direction of the light.
 6. Re-render the textured polygon with the range-map using the shifted texture coordinates (*pass 2*). See 1D example in Fig. 4.4.
 7. Subtract the second image (*pass 2*) from the first (*pass 1*).
 8. Render the polygon with original colour texture (i.e., smooth texture or BTF synthesised and interpolated according to illumination and view position) and add it to previously “bumped” polygon (*pass 3*).
-

- **Bump Map Sampling** - the range-map is not continuous, but is sampled into the texture. The resolution of the texture affects how faithfully the bump map is represented. Increasing the size of the bump map texture can improve the sampling of the high frequency height components.
- **Texture Resolution** - the shifting and subtraction steps produce the directional derivative. Since this is a forward differencing technique, the highest frequency component of the bump map increases as the shift is made smaller. As the shift is made smaller, more demands are made on the texture coordinate precision. The shift can become smaller than the texture filtering implementation can handle, leading to noise and aliasing effects.
- **Surface Curvature** - the tangent coordinate axes are different at each point on a curved surface. This technique approximates this by finding the tangent space transformation at each vertex. Texture mapping interpolates the different shift values from each vertex across the polygon. For polygons with very different vertex normals, this approximation can break down. A solution would be to subdivide the polygons until their vertex normals are parallel to within some error limit.
- **Maximum Bump Map Slope** - the range-map normals used in this technique are good approximations if the bump map slope is small. If there are steep tangents in the range-map, the assumption that the perturbed normal is length one becomes inaccurate, and the highlights appear too bright. This can be corrected by creating a fourth pass, using a modulating texture derived from the original bump-map.

Many of these problems are avoided when using displacement mapping approach instead.

Due to its simplicity we have used variant of bump mapping called parallax mapping [118] to introduce regular macro-structure to proposed probabilistic BTF model in Chapter 7. This approach approximates correct appearance of uneven surfaces by modifying the texture coordinate for each pixel and thus does not require to draw any additional polygons and can be implemented directly in graphics hardware.

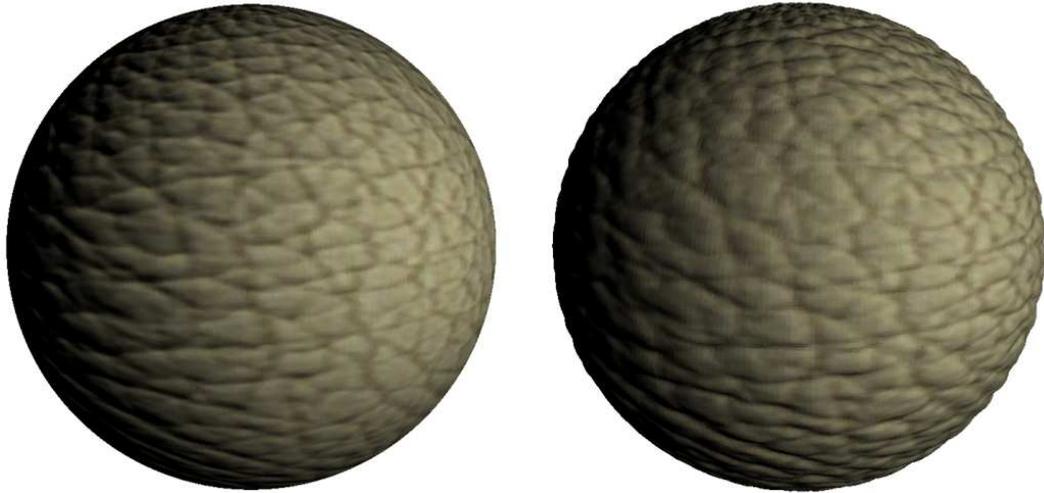


Figure 4.6: Bump Mapping (left) vs. Displacement Mapping (right) for *leather02* material (smooth texture with diffuse lighting).

4.3.2 Displacement Mapping

Displacement mapping [115] is a powerful technique that allows the range-map to manipulate the position of rendered object faces. Unlike bump mapping, where the normals are perturbed to give the perception of a bump, this creates real surface relief. They cast shadows, occlude other objects, and do everything real geometry can do. The displacement mapping is in comparison with bump mapping significantly more computationally demanding since it requires to tessellate original polygon mesh into even finer one. Displacement mapping works for all 3D object types. However, because of need for fine render-faces, not all objects are equally well suited for displacement mapping. Fig. 4.6-right shows example of displacement mapping for *leather02* material on sphere with additional diffuse lighting compared with corresponding bump mapping result. Note the difference mainly near an object's silhouette, where the bump mapping fails to perform realistic occlusions. The main advantage of this method is no limitation on bump height so it can be used for approximation of variety of real-world materials, e.g., fur as shown in [54].

To sum up, the displacement mapping technique provides better results than the bump mapping alternative especially at contours of textured objects, where the bump mapped textures remains flat while displacement mapping preserve material relief correctly (see Fig. 4.6). On the other hand the displacement bump mapping is not so computationally expensive since this method doesn't require further tessellation of each polygon of the textured object.

Chapter 5

Range Data Acquisition

For the sake of rough textures modelling as is often the case of BTF, the original material structure should be preserved as reliably as possible. One way of material surface representation is surface height field so called range-map. In this thesis we have found the range-map useful for enhancing the quality of probabilistic BTF model introduced in Chapter 7. Stochastic nature of the proposed models has difficulties with modelling of textures consisting of regular patterns. So the regularity is introduced into these models by means of surface height information stored in the range-map. Therefore this chapter summarises and discusses different approaches for material surface height acquisition and their suitability for BTF height acquisition.

The range-map is usually stored in a form of monospectral image whose intensity in individual pixels corresponds to relative height of observed surface. In our case the lighter the pixel is, the closer the corresponding surface is to the camera. The range-map enables us to produce 3D representation of rough texture from single texture image using rendering techniques which are widely supported by today's graphics hardware.

There are several options how to obtain the range-map. The easiest but costly is direct measurement using a *range-scanner*. Much cheaper but computationally more demanding alternative is using of analytical methods for range-map estimation. There are numerous methods for range-map estimation and their modifications. The range-map can be estimated by means of shape from stereo [58], shape from shading (SFS) methods [44, 66, 126, 29], specific case of SFS - photometric stereo [121, 104] or other alternative methods [69]. For purposes of this thesis we have implemented and tested the two of the most know and widely used approaches:

- Shape from shading
- Photometric stereo
 - using local integration of surface normals
 - using global integration of surface normals

5.1 Range-Map Measurement

Range-scanners are widely used for obtaining depth information of environment for robot navigation, shape reconstruction or other image processing tasks and can be basically

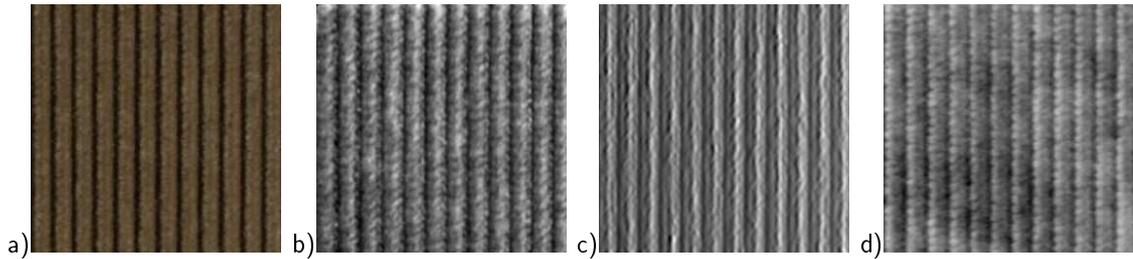


Figure 5.1: Range-map estimation results of corduroy material (a). The following figures illustrate: (b) direct measurement using FM beat sensor, (c) shape from shading estimate, (d) photometric stereo estimate.

divided into following two main categories.

5.1.1 Structured Light Range Sensors

The first category is based on a structured light scanner which uses two optical paths, one for a CCD sensor and one for the structured light projected on measured object and computes depth via triangulation. Light source can be laser beam or light pattern. Using multiple cameras can improve measurement accuracy and exploit two dimensional light pattern to speed up measurement. The main drawback of this triangulation technique is low acquisition speed and missing data at parts of the scene visible to the camera but not visible to light projector. Moreover, for relatively flat and smoothly structured objects such as rough textures for instance, the method is not accurate enough. An additional difficulties appear also due to specularity of measured material.

5.1.2 Time of Flight Range Sensors

The second group of range sensors involves a signal transmitter, a receiver and electronics for measuring the time of flight of the signal during its round trip from the range sensor to the surface of interest. There are three main classes of time of flight sensors:

- *Pulse time delay sensor* emits intense pulses of light or ultrasound and the distance measurement is obtained as the amount of time the pulse takes to reach the target and return to the sensor.
- *AM phase shift sensor* measures the phase difference between the beam emitted by an amplitude-modulated laser and the reflected beam, a quantity proportional to the time of the flight. This sensor suffers from inherent ambiguities since depth differences corresponding to phase shifts that are multiples of 2π cannot be resolved.
- *FM beat sensor* measures the frequency shift (or beat frequency) between a frequency-modulated laser beam and its reflection which is proportional to the round trip flight time. The range-map of corduroy rough material (Fig. 5.1-a) obtained using FM beat sensor is shown in Fig. 5.1-b.

All these sensors have problems when imaging specular surfaces and can be relatively slow due to long integration time at the receiver. Compared to structured light based systems, time of flight sensors offer greater operating range during variable light conditions.

5.2 Range-Map Estimation

Direct range-map measurement device could not be always available or such a measurement could not be in principle possible. In such a situation the analytical range-map estimation methods described in this section can be convenient.

5.2.1 Shape from Stereo

Surface height can be restored from at least two images taken from different positions. The problem reduces mainly to dense correspondence matching. Approaches to the correspondence problem can be broadly classified into two categories: the intensity-based matching and the feature-based matching techniques. In the first category, the matching process is applied directly to the intensity profiles of the two images, while in the second, features are first extracted from the images and the matching process is applied to the features.

Intensity-Based Stereo Matching

The intensity-based stereo matching employs epipolar geometry which reduces the search for correspondences from two-dimensions (the entire image) to one-dimension. This is possible if we assume that an epipolar lines coincide with the horizontal scanlines if the cameras are parallel so the corresponding points in both images must therefore lie on the same horizontal scanline. From the corresponding row of the image pair reveals that the two intensity profiles differ only by a horizontal shift and a local foreshortening. The advantage of this intensity profile matching is that as an output is a dense disparity map and, consequently a dense range-map. Unfortunately, like all constrained optimisation problems, whether the system would converge to the global minimum is still an open problem. An alternative approach in intensity-based stereo matching, commonly known as the window-based method, only matches those regions in the images that are “interesting” [58].

Feature-Based Stereo Matching

In the feature-based approach, the image pair is first preprocessed by an operator so as to extract the features that are stable under the change of viewpoint, the matching process is then applied to the attributes associated with the detected features. Edge elements, corners, line segments, and curve segments are features that are robust against the change of perspective, and they have been widely used in stereo vision. Edge elements and corners are easy to detect, but may suffer from occlusion; line and curve segments require extra computation time, but are more robust against occlusion.

Stereo matching process is a very difficult search procedure. In order to minimum false matches, some matching constraints must be imposed. Several matching constrains were used in the past as for instance similarity, uniqueness, continuity, ordering and epipolar constrain [73].

When the matching is finished the essential matrix of both cameras can be computed based on correspondence between two images. From this essential matrix it can be derived the translation and rotation between the both camera positions. Finally according these information about mutual camera position and orientation the range-map reconstruction

can be performed up to a scale factor. Note that the camera is assumed to be calibrated so that its intrinsic parameters are known. For more see [104].

5.2.2 Shape from Shading

Shape-from-shading (SFS) is problem of determining the shape of a smooth surface given a single image of that surface illuminated from know direction. The pioneer work in this field has been done by Horn [44]. The SFS task can be regarded as calculating the set of partial derivatives $(\tilde{z}_x, \tilde{z}_y)$ corresponding to surface $z(x, y)$ assuming as a input single intensity image illuminated under light direction β . The problem reduces to solving the image irradiance equation

$$E(x, y) = \rho R(z_x, z_y, \omega_{\mathbf{i}}) \quad (5.1)$$

where E is the intensity value of the pixel at position x, y , ρ is albedo of the object surface and R is Lambertian reflectance map (5.2) that maps surface gradients $z_x = \frac{\partial z(x, y)}{\partial x}$ and $z_y = \frac{\partial z(x, y)}{\partial y}$ to an intensity value as follows

$$R(z_x, z_y, \beta) = \frac{u_x z_x + u_y z_y + u_z}{\sqrt{u_x^2 + u_y^2 + u_z^2} \sqrt{1 + z_x^2 + z_y^2}}, \quad (5.2)$$

where $\omega_{\mathbf{i}} = (u_x, u_y, u_z)$ is vector to illumination source in sample coordinate system. The values $(z_x, z_y, -1)$ represents normal vector of surface in observed object location. Unfortunately the equation (5.1) is underconstrained. To overcome this underconstrained nature of Lambertian SFS several constraints were proposed (see SFS survey in [126]). One of them has been introduced by Frankot and Chellappa in [29]. This integrability constrain transforms estimated slopes (\hat{z}_x, \hat{z}_y) to nearest integrable slopes $(\tilde{z}_x, \tilde{z}_y)$ where following integrability equation is valid

$$\frac{\partial}{\partial y} \tilde{z}_x = \frac{\partial}{\partial x} \tilde{z}_y. \quad (5.3)$$

A possibly non-integrable estimate of surface slopes is represented by a finite set of basis functions and integrability is enforced by computing the orthogonal projection onto a vector subspace spanning the set of integrable slopes. This projection maps closed convex sets into convex sets in each iteration step of SFS algorithm (Alg. 2) while following distance measure is minimised

$$d\{(\tilde{z}_x, \hat{z}_x), (\tilde{z}_y, \hat{z}_y)\} = \int \int |\tilde{z}_x - \hat{z}_x|^2 + |\tilde{z}_y - \hat{z}_y|^2 dx dy. \quad (5.4)$$

At the beginning of each pass of the algorithm the previous slopes are smoothed by approximation of Laplacian with the center pixel left out according to

$$\begin{aligned} \hat{\hat{z}}_x(x, y) &= \frac{1}{5} [\tilde{z}_x(x, y+1) + \tilde{z}_x(x, y-1) + \tilde{z}_x(x+1, y) + \tilde{z}_x(x-1, y)] + \\ &+ \frac{1}{20} [\tilde{z}_x(x-1, y-1) + \tilde{z}_x(x-1, y+1) + \tilde{z}_x(x+1, y+1) + \tilde{z}_x(x+1, y-1)] \end{aligned} \quad (5.5)$$

and similarly for $\hat{\hat{z}}_y$.

The following step represents finite difference approximation which generates a new set of raw slope estimates during each iteration

$$\begin{bmatrix} \hat{z}_x(x, y) \\ \hat{z}_y(x, y) \end{bmatrix} = \begin{bmatrix} \hat{z}_x(x, y) \\ \hat{z}_y(x, y) \end{bmatrix} + \frac{1}{\lambda} (I(x, y) - R(\hat{z}_x, \hat{z}_y)) \begin{bmatrix} R_x(\hat{z}_x, \hat{z}_y) \\ R_y(\hat{z}_x, \hat{z}_y) \end{bmatrix}, \quad (5.6)$$

where $R_x = \frac{\partial R}{\partial \hat{z}_x}$ and $R_y = \frac{\partial R}{\partial \hat{z}_y}$ are partial derivations of reflectance function in (5.2) and $I(x, y)$ is intensity in image at location (x, y) and λ is regularisation parameter which influences convergence speed of the whole iterative algorithm.

Surface slopes are represented using integrable basis functions in Fourier domain

$$\begin{aligned} \hat{C}_x(u, v) &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \hat{z}_x(x, y) \cdot e^{j(ux+vy)} dx dy, \\ \hat{C}_y(u, v) &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \hat{z}_y(x, y) \cdot e^{j(ux+vy)} dx dy. \end{aligned} \quad (5.7)$$

And finally in the last step the raw slopes estimates are projected onto the nearest integrable solution using surface slopes representation in Fourier domain as follows:

$$\begin{aligned} \tilde{C}(u, v) &= \frac{-j \frac{u}{N} \hat{C}_x(u, v) + \frac{v}{N} \hat{C}_y(u, v)}{2\pi \frac{u^2}{N} + \frac{v^2}{N}}, \\ \tilde{C}_x(u, v) &= \frac{u}{N} \tilde{C}(u, v), \\ \tilde{C}_y(u, v) &= \frac{v}{N} \tilde{C}(u, v), \end{aligned} \quad (5.8)$$

where \hat{C} is Fourier spectrum corresponding to surface height \hat{z} ,
 u is frequency corresponding to width of the image $1 \dots N$,
 v is frequency corresponding to height of the image $1 \dots N$.

The raw slope estimates which fulfil integrability condition are obtained as the inverse Fourier transformation of slopes represented in Fourier domain:

$$\begin{aligned} \tilde{z}_x(x, y) &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \tilde{C}_x(u, v) \cdot e^{-j(ux+vy)} du dv, \\ \tilde{z}_y(x, y) &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \tilde{C}_y(u, v) \cdot e^{-j(ux+vy)} du dv. \end{aligned} \quad (5.9)$$

The surface height is obtained as a byproduct of integrating surface slope estimates

$$z(x, y) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} C(u, v) \cdot e^{-j(ux+vy)} du dv. \quad (5.10)$$

Result of this algorithm applied on texture with complex surface illuminated by light source with tilt $\phi = 0^\circ$ and slant $\theta = 60^\circ$ is illustrated in Fig. 5.2-b and Fig. 5.1-c. Another example of shape from shading algorithm performance is given in Fig. 5.3. The range-maps of three leather materials (right resp.) were estimated from single texture images (left resp.) for roughly estimated position of illumination source.

Algorithm 2: Shape from Shading Algorithm (Frankot,Chellappa) [29]

1. FOR all $(x, y) \in I_{N \times N}$ set slopes \tilde{z}_x, \tilde{z}_y to zero
 2. WHILE $(d\{(\tilde{z}_x, \hat{z}_x), (\tilde{z}_y, \hat{z}_y)\} < threshold)$
 - (a) FOR all $(x, y) \in I_{N \times N}$ compute smoothing by (5.5)
 - (b) FOR all $(x, y) \in I_{N \times N}$ compute new slopes (5.6)
 - (c) FOR all $(x, y) \in I_{N \times N}$ represent slopes using integrable basis functions in Fourier domain, i.e. $z_x(x, y) \rightarrow C_x(u, v)$ and $z_y(x, y) \rightarrow C_y(u, v)$ (5.7)
 - (d) FOR all $(x, y) \in I_{N \times N}$ project the raw slope estimates onto the nearest integrable solution using (5.8) in Fourier coefficients space
 - (e) FOR all $(x, y) \in I_{N \times N}$ convert slopes from Fourier domain into raw form, i.e. $C_x(u, v) \rightarrow z_x(x, y)$ and $C_y(u, v) \rightarrow z_y(x, y)$ (5.9)
 3. FOR all $(x, y) \in I_{N \times N}$ compute height data $z(x, y)$ from last Fourier coefficients $C(u, v)$ (5.10) by inverse DFT .
 4. FOR all $(x, y) \in I_{N \times N}$ normalise height data
-

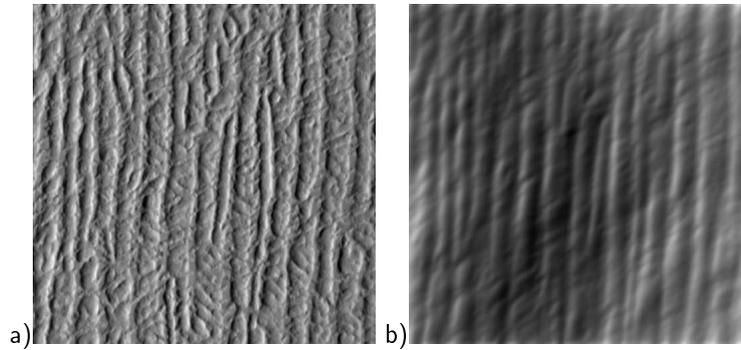


Figure 5.2: Result of SFS algorithm [29]: (a) original image (b) its range-map ($\lambda = 500$, 28 iterations)

5.2.3 Photometric Stereo

The idea of photometric stereo is based on a change illumination position between successive views, while holding the viewing direction constant. Woodham in [121] demonstrated that these intensity measurements from multiple images provide sufficient constraint to determine surface orientation locally. Thus assumptions of photometric stereo are known reflectance function of observed material and existence of three or more images obtained from the same viewing position during known illumination position change. This method enables to compute normal vector in each image pixel [104]. The desired height data



Figure 5.3: Range maps (left resp.) estimated from one image (right resp.) using shape from shading technique for *white leather*, *cushion leather* and *snake leather* from DaimlerChrysler and UTIA texture database.

(range-map) are obtained from the estimated normal-map using integration techniques.

All surfaces are assumed Lambertian and light sources are assumed as point light sources in infinity.

$$I(x, y) = \rho R(z_x, z_y, \omega_i) \quad (5.11)$$

where $I(x, y)$ is image intensity at location (x, y) , ρ is albedo - the surface reflectivity coefficient, $R(z_x, z_y)$ is reflectance map which determine reflected intensity dependently on surface slopes (z_x, z_y) (for example see (5.2)) and ω_i is illumination position.

All directions to light sources are ordered into a matrix \mathbf{L} and corresponding intensity of the same pixel for different illumination directions are ordered to in matrix form $\mathbf{I}(x, y)$ as follows

$$\mathbf{L} = \begin{bmatrix} L_1 \\ \vdots \\ L_n \end{bmatrix}, \quad \mathbf{I}(x, y) = \begin{bmatrix} I_1(x, y) \\ \vdots \\ I_n(x, y) \end{bmatrix}. \quad (5.12)$$

Then surface slopes $\mathbf{n} = [z_x, z_y, 1]$ for each pixel are computed by

$$\mathbf{n}(x, y) = \frac{\mathbf{L}^{-1}\mathbf{I}(x, y)}{\|\mathbf{L}^{-1}\mathbf{I}(x, y)\|} \quad (5.13)$$

Equation (5.13) holds if only three images are used ($n = 3$), otherwise it is necessary to use pseudo-inversion \mathbf{L}^+ of matrix \mathbf{L} where $\mathbf{L}^+ = (\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T$ instead of \mathbf{L}^{-1} . The optimal placement of the illumination for three-image photometric stereo when used for capturing 3D surface texture is derived and verified experimentally in [107]. The gradient images can be obtained even from images where the illumination positions are unknown. This problem is called uncalibrated photometric stereo and is discussed in papers [23, 106].

As was already mentioned photometric stereo computes normal \mathbf{n} for every pixel in image lattice. Based on these normals, the height data are obtained using either local integration techniques introduced, e.g., in [57] or global integration techniques as for example one-pass of shape from shading algorithm, e.g., [126, 29].

Local integration techniques

Local techniques perform local calculation of height increments by curve integral. These methods differ in specifying an integration path, i.e., scan lines and a local neighbourhood for local approximation of height increments. Basic idea is multi-pass surface integration in defined neighbourhood starting from different parts of gradient image containing pixel-wise normals orientation. The resulted range-map is normalised average image of individual passes. These techniques are easy to implement and very fast, however, the locality of

the calculations causes high data dependency and the propagation of height increments propagate the errors as well. These techniques doesn't have impose any restrictions on surface integrability.

One of such methods is described in [57, 97]. The method require four scans through the gradient image according to Fig. 5.4 During the scan the height information is computed

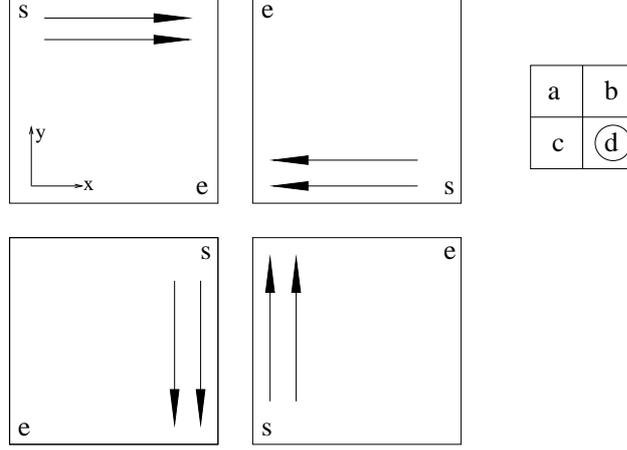


Figure 5.4: Directions of scanning in four different scans through array of normals and used contextual neighbourhood.

for each pixel by following process. Let $\mathbf{n} = [z_x, z_y, -1]$ be the normal vector at position (x, y) (see contextual neighbourhood in Fig. 5.4). This vector is normalised and reordered according to

$$\mathbf{n}' = \frac{[z_x, z_y, -1]}{\sqrt{z_x^2 + z_y^2 + 1}} \quad , \quad \bar{\mathbf{n}}' = [\bar{n}'_x, \bar{n}'_y, -\sqrt{1 - \bar{n}'_x{}^2 - \bar{n}'_y{}^2}] \quad (5.14)$$

then the normals are averaged within contextual neighbourhood

$$\bar{n}'_x = \frac{n'_{a,x} + n'_{b,x} + n'_{c,x} + n'_{d,x}}{4} \quad , \quad \bar{n}'_y = \frac{n'_{a,y} + n'_{b,y} + n'_{c,y} + n'_{d,y}}{4} \quad . \quad (5.15)$$

Applying of following formula for each slope position give as range-map from one scan.

$$z(d) = \frac{z(b) + z(c)}{2} + \frac{\frac{-\bar{n}'_x}{\bar{n}'_z} + \frac{\bar{n}'_y}{\bar{n}'_z}}{2} \quad (5.16)$$

To obtain final range-map it is necessary to perform all four scans as it is depicted in Fig. 5.4 and average their values for each slope position.

Result of photometric stereo with local integration of slopes applied on texture with a rough surface illuminated by light source with tilt $\phi = 0^\circ$ and slant $\theta = 60^\circ$ is illustrated in Fig. 5.5-b. In the image we can see the slanted artifacts caused by the incremental error during contextual neighbourhood movement along image grid from all four scans. The solution avoiding these possible artifacts is using global surface integration techniques.

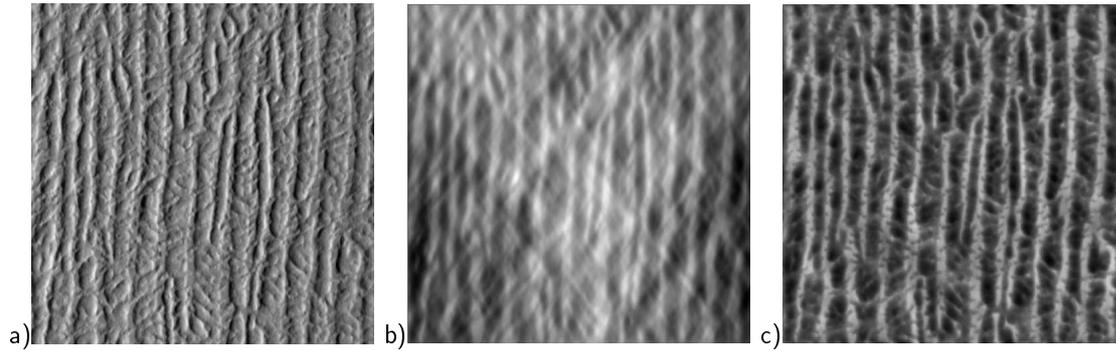


Figure 5.5: Result of photometric stereo normals integration : (a) the original image, the normals integration (b) using local techniques, (c) using shape from shading integration.

Global integration techniques

Global integration techniques treat surface integration as optimisation problem where the goal is to minimise certain function. The global techniques are more robust against noise in comparison to local integration techniques because the surface gradient data have global impact on solution process. One example of these techniques is one iteration of shape from shading algorithm Alg. 2 presented in [29]. Results of this integration approach is depicted in Fig. 5.5-b and Fig. 5.1-d. Fig. 5.6 shows examples of estimated normal and range maps for six different materials from University of Bonn BTF database and one material from UTIA rough textures database. Normal maps (second row) were obtained by means of photometric stereo using three different BTF images with fixed viewing position and corresponding range-maps (third row) were reconstructed using mentioned shape from shading global integration technique.

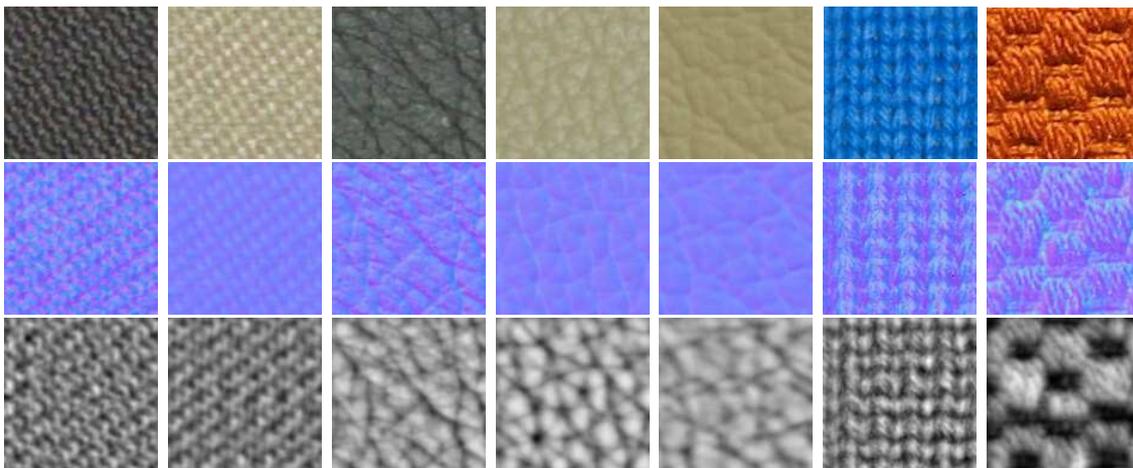


Figure 5.6: Normal and range maps (second and second row) estimated using photometric stereo and global integration technique for six materials from University Bonn BTF database (*fabric01*, *fabric02*, *foil01*, *foil02*, *leather02* and *knitted wool*) together with *cushion fabric* from UTIA rough texture database.

5.3 Range-Map Techniques Overview

The following table describes main advantages and disadvantages of individual range-map measurement and estimation methods.

method	\oplus	\ominus
Structured Light Sensors	easy reconstruction of complex shapes	slow, missing data in some areas, problem with specularities, accurate calibration required
Time of Flight Sensors	accurate, wide operation range	relatively slow, costly device, problem with specularities
Shape from Stereo	wide operation range	slow, for deep scenes only, only approximate results
Shape from Shading	only one image with known illumination position necessary	relatively slow iterative method, only approximate results
Photometric Stereo	fast, accurate	three registered images required (with known lights position), additional normals integration required

5.4 Range-Map Synthesis

The range-map is usually obtained as some kind of material image processing when both: size of the observed material and an acquisition device resolution are limited. However, this constrained size of a range-map is inconvenient for surface approximation of large objects in VR. Thus the estimated range-map can be enlarged to required size by means of image tiling as shown, e.g., in Section 8.3.1. Another possibility is employing of a probabilistic smooth texture synthesis model introduced in Chapter 7. Apart from image tiling methods this option is most suitable for range-map modelling of irregular materials, e.g., plaster, leather, etc. Another range-map synthesis method with real-time performance is described in [51]. Although the method enable fast range-maps synthesis on arbitrary level of details according to given normal density function the possible range-map representations obtained using this model are limited to bumpy surfaces without any low frequency details.

Chapter 6

Segmentation of BTF Data

6.1 BTF segmentation

BTF datasets contain thousands of images taken for varying illumination and view directions. This overall number of images corresponds to angular resolution of BTF measurement setup. In the case of used Bonn university database with 81 different view and 81 different illumination position we obtain 6561 BTF images.

Due to a limited computational capability of contemporary hardware, the both analysis and real-time synthesis of all BTF images by means of probabilistic MRF-based BTF models presented in Chapter 7 would be too costly and also superfluous so a data reduction method is inevitable.

This reduction can be performed due to the fact that an individual BTF measurements have similar colour and brightness properties with nearby illumination and viewing angles. Image similarity depends mainly on individual material reflective properties and it is quite probable mainly for close spatial positions of light and camera. These facts lead us to idea of BTF segmentation into a finite set of BTF clusters.

During segmentation the BTF data space, spanning all view and illumination directions, is divided into set of clusters. Each such a cluster is represented by the BTF image closest to cluster center according to similarity function. Such an ideal similarity function should favour the overall brightness and colour hue similarity regardless of texture or shadow information presented in compared BTF data.

6.1.1 Data Representation

One of suitable data arrangements for BTF segmentation is BTF_{TEX} containing individual images taken for different illumination and view positions (see Section 2.3.2). Thus the resulting data space contains 81×81 data features. An appropriate form of these features considerably influence performance and speed of segmentation algorithm. Natural solution would be using directly pixel values of BTF image ordered into a feature vector. However, this solution has several shortcomings. The first of them is length of these feature vectors. Even for relatively small data window containing the largest structure elements involved in simple materials cannot be smaller than 20×20 . This window in three spectral channels, results into feature vector of size $3 \times 20 \times 20 = 1200$. Such long image features cause very time demanding segmentations. Moreover, the pixel-wise registration of individual BTF images is far from being perfect (accuracy $\max \pm 3$ pixels) so the Euclidean distance of such

features can lead to relatively high distance even in cases when both images are the same but shifted for a few pixels. The second problem is caused by a nature of BTF including images with masked and shadowed areas. The pixels in two BTF images when one was taken from head-on position and the other for high grazing angle does not necessarily correspond to the same location on the measured material sample due to masking effect caused by rough surface structure (see Fig. 2.6). Additionally, when the view angles are fixed the changing illumination produces shadows occurring in different portions of the BTF image so again the simple Euclidean distance between two such data features fails.

6.1.2 Histogram similarity

The simple image statistics present in image histogram is handful for the purpose of accurate BTF segmentation. As a similarity measure can be used difference between corresponding histogram bins. However, the difference of two histograms do not pay attention on mutual positions of histogram bins in both images so using the cumulative histograms \bar{m} instead is convenient.

For a colour BTF images a corresponding histograms in individual colour channels are used. So each BTF image is represented by data features containing subsequently three cumulative histograms corresponding to individual colour channels. The size of such feature vectors is $3 \times 256 = 768$. The histogram similarity for one colour channel r_3 is

$$\bar{d}_H(k, i, v, r_3) = \sum_{i=1}^{n_l} (\bar{m}_{c_k} - \bar{m}_{i,v})^2, \quad (6.1)$$

where the n_l is a number of image quantisation levels (i.e., in our case $n_l = 256$), c_k represents BTF image corresponding to k -th cluster center and $r_3 = 1 \dots d$ is spectral channel. The final distance for all three colour channels is

$$\Delta d_H(k, i, v) = \Delta d_H(k, i, v, 1) + \Delta d_H(k, i, v, 2) + \Delta d_H(k, i, v, 3). \quad (6.2)$$

6.1.3 Colour-Space Transformation

The treatment of colour data according to human visual system is an important aspect of used cumulative histogram similarity measure. To avoid weighting of individual RGB values according to subjective human eye sensitivity we transformed all BTF images into perceptually uniform CIE Lab (D65) colour-space [123, 28]. In CIE Lab the same colour shift in terms of Euclidean distance results into the same difference of visual perception of resulted colours and due to this reason the Euclidean distance become appropriate metric for BTF histograms clustering.

6.1.4 K-Means Clustering

We have used common iterative K-means clustering algorithm [105, 20] for the sake of BTF histograms clustering. This technique exploits first order statistics of the data and finds a predetermined number of clusters in data space, by minimising the sum of squared errors criterion. The exploitation of clustering algorithm for aim of BTF data segmentation into \mathcal{D} clusters forming set K is introduced in Alg. 3.

Algorithm 3: K-means Algorithm for BTF segmentation

1. Give random position to $\forall k \in K$ centres of all clusters, i.e., $[i = 0 \dots 81, j = 0 \dots 81]$
 2. WHILE ((inner class variance $J >$ threshold) OR (number of iterations $<$ max. iterations) OR (no further data shifts between clusters))
 - (a) FOR each feature vector (BTF image with view position v and illumination position i compute distance $d(k, i, v)$ to every cluster centre k and assign this vector to cluster with the closest distance (Step 1):

$$k^* = \arg \min_{k \in K} d(k, i, v) .$$
 - (b) FOR each cluster $k \in K$ recompute its centre according to all assigned vectors. (Step 2)
 3. Corresponding representative of cluster $k \in K$ is a BTF image having the closest distance $d()$ to the center of the cluster k .
-

The segmentation quality criterion of correct data distribution into individual clusters is the inner class variance defined by

$$J = \sum_{k=1}^{\mathcal{D}} \sum_{j \in S_k} |x_j - \mu_k| , \quad (6.3)$$

where S_k is a set of data features belonging to class k , x_j vector representing j -th data point and μ_k is a mean vector of data in class k , i.e., $\mu_k = \frac{1}{|S_k|} \sum_{j \in S_k} x_j$.

The whole iterative process is stopped when J drops under predefined threshold or there is no any further shift of data features between individual classes which indicates optimal distribution of data features into the clusters.

Finally for each cluster is chosen one representative which is the closest BTF image to cluster centre in sense of mentioned Euclidean metric between multispectral CIE Lab histograms.

6.2 Optimal Number of Clusters

The estimate of optimal number of BTF clusters is very important step, when using K-means clustering method. We will denote this number a *BTF dimensionality* \mathcal{D} . There are several methods based usually on BTF linear analysis using the PCA performed on all 6561 BTF images or on their spatial correlations. Here have to be noted that finding of principal components for whole BTF space is tremendous computational task which even for relatively small part of original BTF images takes many hours of computations.

An alternative solution avoiding these problems consist in starting with predefined minimal number of clusters. The number of clusters is then iteratively increased by splitting the cluster of the biggest inner-variance J (6.3). The cluster adding iterations are repeated till the number of data features (corresponding BTF images) in each cluster or the inner variance in all individual clusters is smaller than predefined threshold.

6.2.1 PCA on Raw Data

One way to obtain optimal BTF dimensionality \mathcal{D} is employing the image statistical methods. One of them is PCA of whole BTF data space. All pixels from window of size $M \times N$ cut from individual BTF images are ordered into vectors \mathbf{X} and centered using mean BTF image vector $\hat{\mathbf{X}}$. All these vectors form matrix \mathbf{A} of size $3MN \times n_i n_v$ as follows

$$\mathbf{A} = [\mathbf{X}_1 - \hat{\mathbf{X}}, \mathbf{X}_2 - \hat{\mathbf{X}}, \dots, \mathbf{X}_{n_i \cdot n_v} - \hat{\mathbf{X}}] \quad (6.4)$$

In the following step the symmetric matrix $\mathbf{M} = \mathbf{A}^T \mathbf{A}$ is computed. Note that even this symmetrisation step is quite long since the size of matrix A , even if taking into account only BTF image area 20×20 , is 1200×6561 . So for material with large structure elements the matrix can be considerably larger which causes very long computational times.

When having the symmetric matrix \mathbf{M} the SVD is performed resulting into the following decomposition

$$\mathbf{M} = \mathbf{U} \mathbf{D} \mathbf{V}^T \quad (6.5)$$

where \mathbf{U} , \mathbf{V}^T contain orthonormal columns, rows respectively and \mathbf{D} is non-negative diagonal matrix containing sorted eigen-values.

The individual eigen-values from diagonal matrix \mathbf{D} weight the importance of eigenvectors contained in matrices \mathbf{U} and \mathbf{V}^T for matrix \mathbf{M} reconstruction. The number of preserved eigen-numbers approximately corresponds to the BTF dimensionality \mathcal{D} according to the equation

$$\frac{\sum_{i=1}^{\mathcal{D}} \sigma_i^2}{\sum_{i=1}^{n_i n_v} \sigma_i^2} \geq F \quad (6.6)$$

where σ_i^2 are squared eigen-values sorted downwardly on diagonal of matrix \mathbf{D} and F is a fraction close to 1 which enables preservation of the most important BTF features. In our experiments on several BTFs we suppose $F = 0.9$ as satisfactory approximation.

From this follows that only \mathcal{D} eigen-values and corresponding eigen-vectors (i.e., eigen-BTF images) have to be stored for BTF dataset reconstruction holding most of the original BTF information.

Plotting of the first thirty eigen-values and corresponding logarithmic eigen-values, illustrated in Fig. 6.1, shows that only about 10 to 30 BTF images covers most of the information in BTF of tested materials. Fig. 6.1 shows that most of the materials can be satisfactory approximated by means of linear combination of relatively low number \mathcal{D} of eigen-BTF images.

6.2.2 PCA on Raw Data Correlations

The BTF dimensionality \mathcal{D} should correspond to number of texture images, which are necessary for correct reconstruction of whole BTF dataset. Another possible way to obtain \mathcal{D} was presented in paper of Suen and Healey [109].

The method determines texture dimensionality using correlation functions computed in small area of each BTF image. If we assume that all BTF images are already rectified (rotated to head-on position, i.e., the texture normal is identical to camera axis), the correlation between spectra i and j is computed in following way

$$R_{ij}(\mathbf{m}) = \frac{1}{|\Lambda(\mathbf{m})|} \sum_{\mathbf{x} \in \Lambda, (\mathbf{x} + \mathbf{m}) \in \Lambda} [I_i(\mathbf{x}) - \bar{I}_i][I_j(\mathbf{x} + \mathbf{m}) - \bar{I}_j] \quad (6.7)$$

where $\mathbf{x} = [r_1, r_2]^T$ are planar coordinates in BTF image, $\mathbf{m} = [m, n]$ are correlation shifts,

$$\bar{I}_i = \frac{1}{|\Lambda|} \sum_{\mathbf{x} \in \Lambda} I_i(\mathbf{x})$$

and similarly for \bar{I}_j , Λ is the region in the image which the sample occupies, $|\Lambda(\mathbf{m})|$ is number of locations where $\mathbf{x} \in \Lambda$ and $(\mathbf{x} + \mathbf{m}) \in \Lambda$.

If three spectral bands in BTF images are assumed (RGB), we have namely following nine correlations per each image pixel $R_{RR}, R_{RG}, R_{RB}, R_{GR}, R_{GG}, R_{GB}, R_{BR}, R_{BG}, R_{BB}$. The values of m, n can vary in intervals $0 \leq m \leq m_{max}$ and $-n_{max} \leq n \leq n_{max}$. Thus finally $9(m_{max}+1)(2n_{max}+1)$ correlations for every image are obtained and their values are ordered into a column vector of this length. Such a vector is obtained for every BTF image of all $n_i n_v$ BTF images. Finally the matrix \mathbf{R} (6.7) with size $9(m_{max}+1)(2n_{max}+1) \times n_i n_v$ is built.

On matrix R is performed SVD which finds diagonal matrix \mathbf{D} with eigen-values and

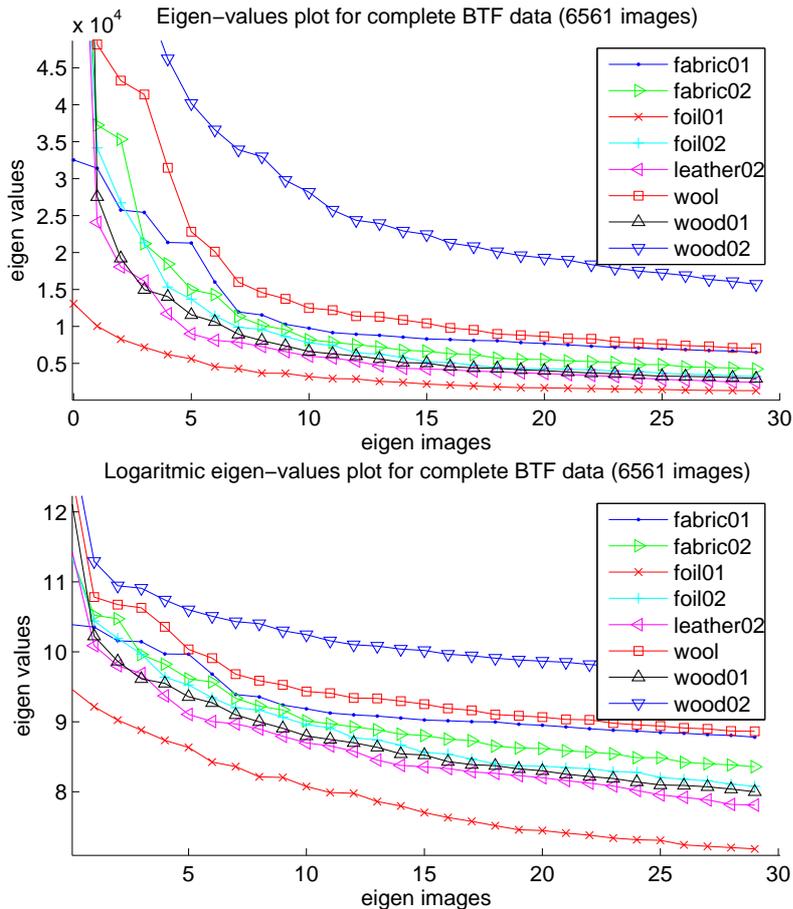


Figure 6.1: The thirty highest eigen-values plotting for eight distinct BTFs. The first image depicts standard eigen-values plotting while the second image shows corresponding logarithmic values.

matrices \mathbf{U} , \mathbf{V} with corresponding eigen-vectors.

$$\mathbf{R} = \mathbf{U}\mathbf{D}\mathbf{V}^T \quad (6.8)$$

The eigen-values are sorted according to their size downwardly so $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{n_i n_v}$. The BTF dimensionality \mathcal{D} is obtained from the formula (6.6)

6.3 Clustered BTF Representation

The final BTF reconstruction is based on cluster index file which stores indices of the individual cluster representative images being used for certain light and camera positions \mathbf{i} and \mathbf{v} . The example of cluster index for $\mathcal{D} = 15$ together with corresponding cluster representative images for material *leather01* is shown in Fig. 6.2.

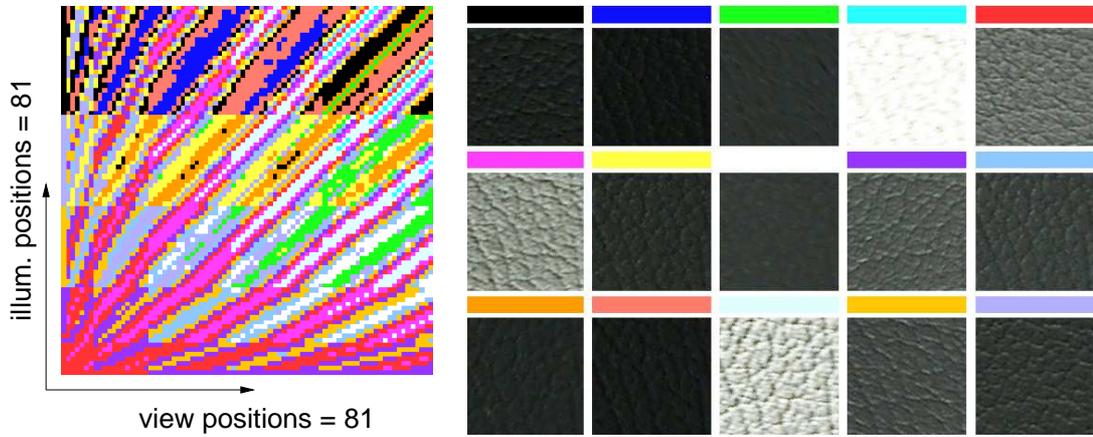


Figure 6.2: Cluster subspace index for the *leather01* BTF containing 15 clusters (left). The BTF images withdrawn according to clustering results (right). Each image represents the closest BTF image to center of each cluster with respect to colours in cluster index (left).

Only these images so-called BTF subspaces (Fig. 6.2-right) are synthesised using MRF models presented in Chapter 7 and their parameters are stored along with cluster index table (Fig. 6.2-left). According to these data the BTF texture image with mesostructure and microstructure corresponding to required illumination and view direction are synthesised.

From cluster index in Fig. 6.2-left and Fig. 6.3 it is apparent that the change of illumination direction, in comparison with the change of view direction, results in higher cluster difference as was mentioned already in [59]. See horizontal stripes of different “background” clusters which corresponds to change of illumination elevation angle.

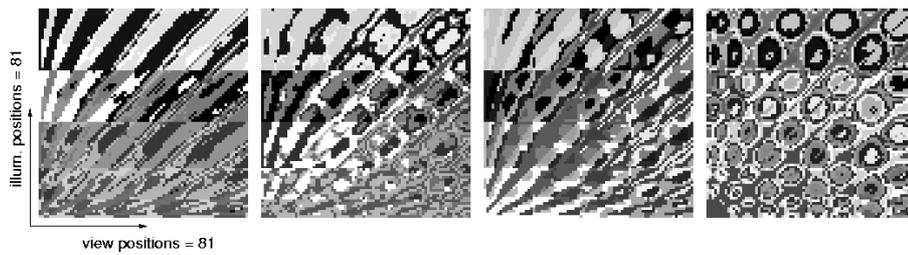


Figure 6.3: Subspace index images for four different materials: leather02 (12 clusters), fabric02 (13 clusters), wool (15 clusters) and wood02 (15 clusters).

Chapter 7

Probabilistic BTF Modelling

The main requirements on ideal BTF model are preservation of visual quality, compact size of parametric set and low computational demands. All the BTF compression and modelling methods mentioned in Chapter 2 are based on repetition of stored BTF images or on some kind of their pixel-wise parametric representation. These approaches enable relatively low BTF data compression and require additional, usually sample-based, methods for enlargement of synthesised BTF images.

This chapter introduces several novel BTF models which remedy mentioned disadvantages of contemporary BTF models at the price of compromise visual quality for some materials. Proposed models are generative so they do not need to store any form of original BTF measurements but only very restricted model statistics. This approach allows to reach huge compression ratio of the original BTF measurements, while the fast BTF data synthesis and rendering is guaranteed.

The proposed probabilistic BTF model is based on two main parts: BTF segmentation into subspaces and subspace modelling as depicted in a block scheme on Fig. 7.1.

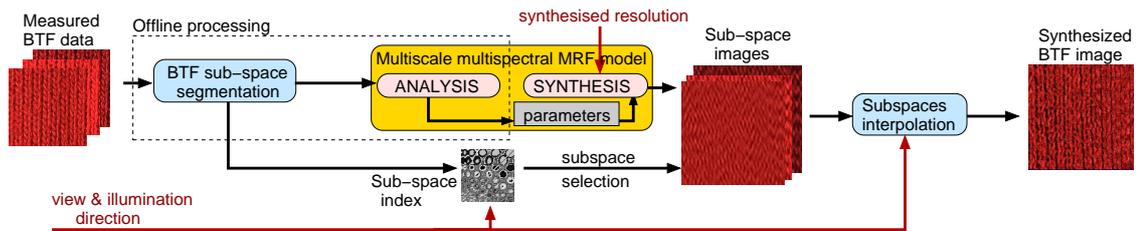


Figure 7.1: The overall scheme of proposed probabilistic BTF model.

The BTF segmentation method is in detail described in the previous chapter while the subspace modelling is subject of this chapter. The proposed BTF modelling approach performs well on smooth BTFs with stochastic structure, e.g., wood, plaster, etc. However, due to its inherent principle it has difficulties to reproduce regular low frequency structures often present for instance in BTFs of textiles. To overcome this drawback we introduce modified BTF model with additional rough surface structure processing pipeline based on height data estimation and modelling as it is illustrated in Fig. 7.2. The additional processing consist in range and normal maps estimation and their subsequent enlargement. The final BTF image is obtained as combination of original interpolated subspace syn-

thesis and height data in displacement or bump mapping filter. Such an approximation is generally possible only for materials fulfilling Lambertian reflectance law, however, it gives satisfactory results also for materials where this assumption does not hold. This

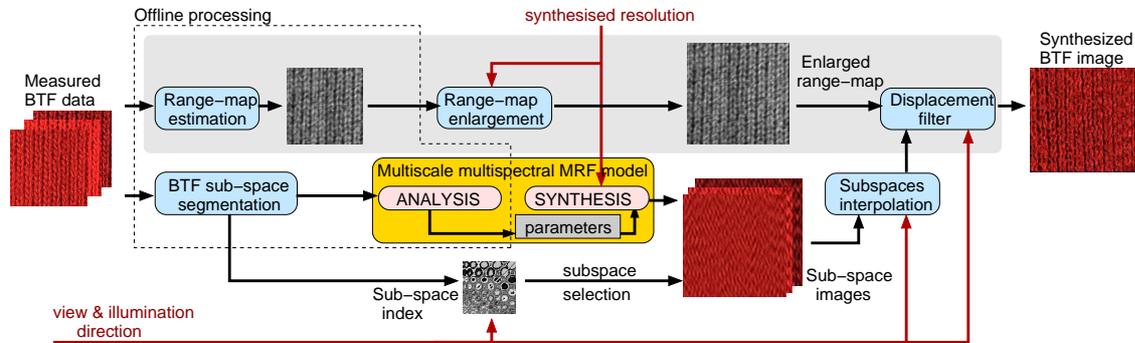


Figure 7.2: The overall scheme of proposed probabilistic BTF model with additional rough information processing pipeline.

approach simplifies the underlying probabilistic MRF models structure as well as their corresponding robust parameters estimation problem and allow shading and scattering approximation by means of GPU.

7.1 BTF Segmentation into Subspaces

All three proposed BTF models (Fig. 7.1 and Fig. 7.2) starts with the BTF image space segmentation into several subspaces. This is done using the K-means algorithm in the feature space that consists of all view and illumination combinations using colour cumulative histogram data as data features. Currently the best available BTFs are represented by $81 \text{ view} \times 81 \text{ illumination}$ directions (for University of Bonn data [98]). To obtain more accurate segmentation results the cumulative histograms are computed in perceptually uniform CIE Lab colour-space. Described segmentation divides BTF space into a set of subspaces and each of them is represented by the nearest BTF image that corresponds to cluster center in sense of Euclidean distance between two cumulative histograms.

An important issue is the optimal number of subspaces. The eigen-value analysis (PCA) of whole BTF data space leads us to the conclusion that the intrinsic BTF space dimensionality for most BTF texture samples is between ten and thirty. Hence the first largest 10 to 30 eigen-values contain often 90% of the whole information. Several examples of BTF space segmentations for different materials are depicted in Fig. 6.3. We restricted the maximal number of subspace images in our implementations to twenty with respect to limited GPU memory, computational demands as well as satisfactory visual BTF reconstruction. More details on BTF segmentation are given in Chapter 6.

7.2 Surface Height Data Estimation and Enlargement

The BTF models exploit range and normal map estimated from the original BTF measurements for representation of rough material macrostructure. This can be performed by

means of several method as it is discussed in Chapter 5. We have used a simple photometric stereo which is preferable for BTF data having tens of ideally mutually registered images with known light positions. The photometric stereo enables to acquire the normal and albedo fields from intensity images when the Lambertian opaque surface is assumed (for details see [121]).

This method provide relatively accurate approximation of material macrostructure. We obtained satisfactory results for many real-world materials such as for instance plaster, leather, etc. Range data enlargement to required resolution was accomplished by means of image tiling method presented in [103] (see Section 8.3.1) or using the MRF modelling approach.

7.3 Multiscale Multispectral Subspace Models

In this section we propose the three different *Markov Random Field* (MRF) subspace models which are responsible for material modelling on mesostructure and microstructure level. Individual cluster representative subspace images obtained using BTF segmentation are subsequently analysed by means of one of these probabilistic MRF models. The simultaneous modelling of such multispectral subspace images generally requires three dimensional models. If a 3D data space can be factorised then these data can be modelled using a set of less-dimensional 2D random field models, otherwise it is necessary to use some 3D random field model. Although, full 3D models allow unrestricted spatial-spectral correlation modelling its main drawback is the large number of parameters that have to be estimated and consequently more time demanding analysis as well as synthesis.

7.3.1 3D Models

The modelling of general multi-spectral textures requires three dimensional model that allow unrestricted spatial-spectral correlation representation. Three dimensional models can be divided into two major categories with respect to the type of model's contextual neighbourhood (CN): *causal* and *non-causal*. The non-causal models do not have any particular restriction on the shape of CN, however, their computation is very slow since in most MRF models it requires an iterative Monte Carlo methods. On the other hand, the causal models restrict the CN shape to be either causal or unilateral, i.e., during computation the CN take into account during computation only the known or already computed image pixels (see example in Fig. 7.6). This enables use some exceptional models for fast analytical parameter estimation methods as well as to perform fast model synthesis.

The basic block scheme of 3D multiscale multispectral model is depicted in Fig. 7.3 The whole subspace images modelling process can be split into two major parts. The first one is a simultaneous *analysis* of all subspace images by means of 3D MRF model. The second part is a fast subspace images *synthesis* of arbitrary resolution based on MRF model parameters computed in the previous analytical step.

In comparison with the offline analysis the synthesis has to be fast enough to enable real-time rendering of synthesised BTFs in VR systems.

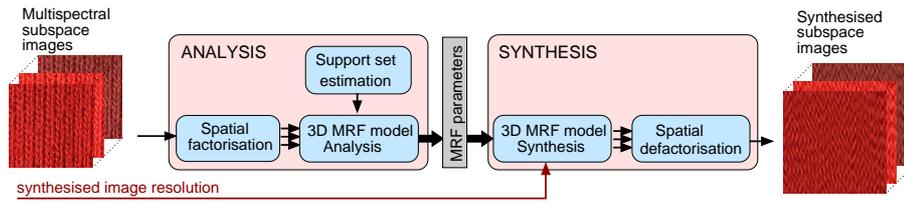


Figure 7.3: The scheme of 3D MRF multiscale multispectral subspace model.

Analysis

Individual subspace images are decomposed into a multi-resolution grid and each resolution data are independently modelled by their dedicated MRF model. This enable efficient modelling of all visual features of subspace images. The multi-resolution grid is created by means of a Gaussian-Laplacian pyramid as is described in full details in Section 7.3.4.

The results of the analytical part are several MRF models with different parameters corresponding to different synthetic results. Since there is not any suitable similarity measure to compare visual quality of colour textures available so far, we choose the optimal model structure according to subjective visual observation as is discussed in Chapter 9.

The analysis of 3D MRF models generally leads to the computationally demanding Monte Carlo iterative techniques. However, after several restrictions, as mentioned above, we can avoid this iterative parameter estimation and use the fast analytical estimation techniques.

Generally the MRF model is generative so it does not require to store any samples of original subspace textures. The number of the 3D model parameters depends only on the number of input data spectral channels and on size of model's contextual neighbourhood. Thus the 3D MRF model enable unbeatable compression of subspace images. The BTF model parameters for one material in form of the floating point numbers take up typically about 100KB in average for contextual neighbourhood including six support pixels and three pyramid layers.

Synthesis

The synthesis of 3D model is process complementary to the analysis, however, in comparison with analytical part of proposed analytical 3D MRF model, its synthesis is very fast and consist of individual multi-resolution pyramid planes synthesis and their subsequent interpolation during spatial defactorisation step (inversion process to (7.4),(7.6)). Based on the parameters of 3D MRF model the subspace images of arbitrary resolution can be synthesised.

7.3.2 Approximation using 2D Model

The factorisation alternative of 3D data models is attractive because it allows using simpler 2D data models with less parameters and consequently also easier implementation. In our case the spectral factorisation was employed to decompose individual multispectral subspace images into monospectral factors which are modelled independently by their dedicated 2D models. The block scheme of general 2D multiscale multispectral model is

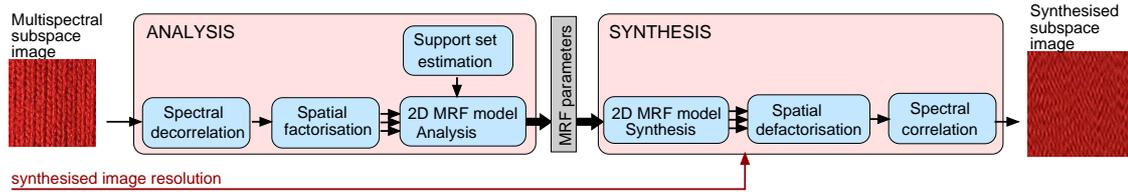


Figure 7.4: The scheme of 2D MRF multiscale multispectral subspace model.

illustrated in Fig. 7.4.

The whole subspace images modelling process can be again split into two major parts. The first one is subsequent *analysis* of subspace images by means of a 2D MRF model. The second part is fast subspace images *synthesis* of arbitrary resolution based on MRF model parameters computed in the previous analytical step.

Analysis

Apart from 3D model the 2D model starts with a spectral factorisation step which results into a set of monospectral subspace factors. The spectral factorisation is accomplished by the Karhunen-Loeve transformation of input multispectral data (see Section 7.3.3 for details). Each monospectral decorrelated plane is subsequently spatially factorised using a Gaussian-Laplacian multi-resolution pyramid and each monospectral subband image is analysed by means of a 2D MRF model resulting into a texture representation by the model parameters. So each monospectral subspace factor have dedicated multi-resolution pyramid and each pyramid resolution layer is analysed by dedicated 2D MRF model.

As a result of multispectral data analysis a matrix of inverse Karhunen-Loeve transformation is stored together with 2D MRF model parameters for individual subspace images. The number of the 2D MRF model parameters depends again on the number of input data spectral channels of individual subspace images and on the size of model's contextual neighbourhood. Thus the 2D MRF model enable even higher compression of subspace images than 3D model. The BTF model parameters for one material take up typically about 60KB in average for contextual neighbourhood with six support pixels and three pyramid layers.

Synthesis

The synthesis of monospectral subspace factors requires reconstruction of individual planes of the multi-resolution pyramid. Each monospectral fine-resolution component is obtained from the pyramid collapse procedure (inversion process to (7.4),(7.6)). Finally the resulting synthesised multispectral subspace image is obtained from the set of synthesised monospectral images using the inverse Karhunen-Loeve transformation. Based on the parameters of 2D MRF model and the inverse Karhunen-Loeve transformation matrix the proposed 2D model enable synthesis of subspace images in arbitrary resolution. Despite the essential computation of the inverse Karhunen-Loeve transformation the synthesis of the proposed 2D model is very fast. Moreover, the 2D multiscale multispectral model can be appropriate candidate for fast hardware accelerated implementation since the synthesis

consists of relatively simple operations. The idea of such an implementation is described in Section 7.5.1.

7.3.3 Spectral Factorisation

A real data space can be decorrelated only approximately, hence the independent spectral component modelling approach suffers from some loss of image information, however, this loss of spectral information is only visible in textures with many substantially different colours. Spectral factorisation using the Karhunen-Loeve expansion transforms the original centered data space \tilde{Y} defined on the rectangular $M \times N$ finite lattice I into a new data space with K-L coordinate axes \bar{Y} . This new basis consists of the eigenvectors of the second-order statistical moments matrix (7.1)

$$\Phi = E\{\tilde{Y}_r \tilde{Y}_r^T\} \quad (7.1)$$

where the multiindex r has two components $r = \{r_1, r_2\}$, the first component is row and the second one column index, respectively. The projection of random vector \tilde{Y}_r onto the K-L coordinate system uses the transformation matrix

$$T = [u_1^T, u_2^T, \dots, u_d^T]^T \quad (7.2)$$

which has rows u_j that are eigenvectors of the matrix Φ . The number d of eigenvectors depends on the number of spectral bands in the original multispectral data \tilde{Y}_r (for applications in RGB colourspace the $d = 3$). Components of the transformed vector

$$\bar{Y}_r = T \tilde{Y}_r \quad (7.3)$$

are mutually uncorrelated and if we assume that they are also Gaussian then they are independent thus each transformed monospectral factor can be modelled independently of the remaining spectral factors.

7.3.4 Spatial Factorisation

The spatial factorisation is technique that enables separate modelling of individual frequency components of input image data. So these multi-spectral image data are decomposed into a multi-resolution grid and each resolution data are independently modelled by dedicated MRF model. Each grid resolution represents a single spatial frequency band of the texture which corresponds to one layer of Gaussian-Laplacian pyramid.

The input multi-spectral image is decomposed into a multi-resolution grid and all resolution data factors represents the Gaussian pyramid $\check{Y}_r^{(k)}$ of level k . The Gaussian pyramid $\check{Y}_r^{(k)}$ is a sequence of k images in which each one is a low-pass downsampled version of its predecessor. Gaussian filter is approximated by the weighting function (FIR generating kernel) w which is chosen to comply:

$$\begin{array}{ll} \text{separability} & w_s = \hat{w}_{s_1} \hat{w}_{s_2} \\ \text{normalisation} & \sum_i \hat{w}_i = 1 \\ \text{symmetry} & \hat{w}_i = \hat{w}_{-i} \\ \text{equal contribution} & \hat{w}_0 = 2\hat{w}_1 \quad (l = 1) \end{array}$$

where l represents size of kernel function and $s = \{s_1, s_2\}$ in row and column index in the kernel.

The equal contribution constraint requires that all nodes at the given level contribute the same total weight to the nodes at the next higher level. The solution of above constrains for the reduction factor 3 ($2l + 1$) is $\hat{w}_0 = 0.5$, $\hat{w}_1 = 0.25$, for reduction factor 5 it is $\hat{w}_0 = \alpha$, $\hat{w}_1 = 0.25 - 0.5\alpha$ where usual choice is $\alpha = 0.4$.

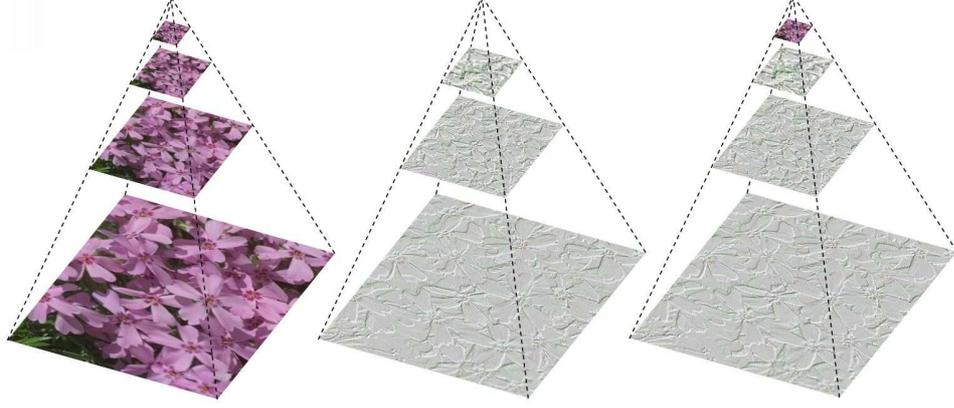


Figure 7.5: Multiscale texture decomposition into the Gaussian-Laplacian pyramid. The Gaussian, Laplacian and Gaussian-Laplacian pyramids respectively.

The Gaussian pyramid for a reduction factor n (for $n = 2$ the $N \times N$ image is down-sampled to $\frac{N}{2} \times \frac{N}{2}$) is

$$\ddot{Y}_{r,i}^{(k)} = \downarrow^n (\ddot{Y}_{\bullet,i}^{(k-1)} \otimes w) \quad k = 1, 2, \dots, \quad (7.4)$$

where $\ddot{Y}_{\bullet,i}^{(0)} = \bar{Y}_{\bullet,i}$, \downarrow^n denotes down-sampling with reduction factor n and \otimes is the convolution operation. Convolution can be substituted using

$$\ddot{Y}_r^{(k)} = \sum_{i,j=-l}^l \hat{w}_i \hat{w}_j \ddot{Y}_{2r+(i,j)}^{(k-1)}. \quad (7.5)$$

An analysed texture is decomposed into multiple resolutions factors using the Laplacian pyramid and the intermediary Gaussian pyramid $\ddot{Y}_{\bullet,i}^{(k)}$ which is a sequence of images in which each one is a low-pass down-sampled version of its predecessor. Each level of Laplacian pyramid generates a single spatial frequency band of the texture and is independently modelled by its dedicated 2D subspace model (see sections 7.3.7, 7.3.6 and 7.3.5). Such a Laplacian pyramid $\dot{Y}_{r,i}^{(k)}$ contains band-pass components and provides a good approximation to the Laplacian of the Gaussian kernel. It can be constructed by differencing single Gaussian pyramid layers:

$$\dot{Y}_{r,i}^{(k)} = \ddot{Y}_{r,i}^{(k)} - \uparrow^n (\ddot{Y}_{\bullet,i}^{(k+1)}) \quad k = 0, 1, \dots, \quad (7.6)$$

where \uparrow^n is the up-sampling with an expanding factor n . The example of resulted Gaussian-Laplacian pyramid is illustrated in Fig. 7.5.

There are several alternative spatial factorisation approaches to the Gaussian-Laplacian pyramid available such as a steerable pyramid introduced in [43, 93]. However, this technique is much slower than the approach proposed above so we did not use it in this thesis.

7.3.5 3D Causal Auto-Regressive Subspace Model

Multi-spectral subspace images are decomposed into a multi-resolution grid and each resolution data is modelled independently by independent Gaussian noise driven 3D CAR MRF model that enable simultaneous modelling of all subspace images.

Let the digitised colour texture Y is indexed on a finite rectangular three-dimensional $N \times M \times d$ underlying lattice I , where $N \times M$ is the image size and d is the number of spectral bands (i.e., $d = 3$ for usual colour textures). Let us denote a simplified multiindex r to having two components $r = \{r_1, r_2, r_3\}$. The first component is a row index, the second one is a column index and the third is a spectral index, respectively.

I_r specifies shape of the contextual neighbourhood (CN) around the actual index $r = \{r_1, r_2, r_3\}$. Causality is fulfilled when all data obtained from CN are known (not missing pixels). The example of 2D causal and non-causal CN is depicted in Fig. 7.6.

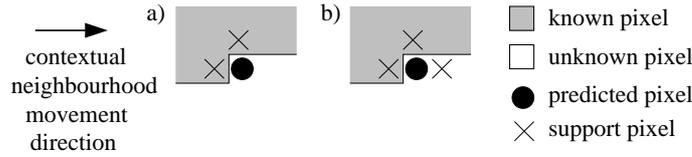


Figure 7.6: The example of 2D causal (a) and non-causal (b) contextual neighbourhood.

From this causal contextual neighbourhood the known data are arranged into a vector:

$$X_r = [Y_{r-s}^T : \forall \{s\} \in I_r^c]^T . \quad (7.7)$$

The (CAR) random field is a family of random variables with a joint probability density on the set of all possible realisations Y of the $M \times N \times d$ lattice I , subject to the following condition:

$$p(Y | \Theta, \Sigma^{-1}) = (2\pi)^{-\frac{d(MN-1)}{2}} |\Sigma^{-1}|^{\frac{(MN-1)}{2}} \exp \left\{ -\frac{1}{2} \text{tr} \left\{ \Sigma^{-1} \begin{pmatrix} -I \\ \Theta^T \end{pmatrix}^T \tilde{V}_{MN-1} \begin{pmatrix} -I \\ \Theta^T \end{pmatrix} \right\} \right\} , \quad (7.8)$$

where I is identity matrix, Θ is parameter matrix, Σ is covariance matrix of Gaussian white noise and

$$\tilde{V}_{r-1} = \begin{pmatrix} \tilde{V}_{YY(r-1)} & \tilde{V}_{XY(r-1)}^T \\ \tilde{V}_{XY(r-1)} & \tilde{V}_{XX(r-1)} \end{pmatrix} . \quad (7.9)$$

The used notion is:

$$\tilde{V}_{XX(r-1)} = \sum_{k=1}^{r-1} X_k X_k^T ,$$

$$\begin{aligned}\tilde{V}_{XY(r-1)} &= \sum_{k=1}^{r-1} X_k Y_k^T , \\ \tilde{V}_{YY(r-1)} &= \sum_{k=1}^{r-1} Y_k Y_k^T .\end{aligned}\quad (7.10)$$

Simplified notation $r, r-1, \dots$ denotes the multispectral process position in I , i.e., $r = \{r_1, r_2, r_3\}$, $r-1$ is the location immediately preceding $\{r_1, r_2, r_3\}$, etc. A direction of movement on the underlying image sub-lattice is common rows scanning. $r-1 = (r_1 - \Delta_1, r_2 - \Delta_2, r_3)$, $r-2 = (r_1 - 2\Delta_1, r_2 - 2\Delta_2, r_3), \dots$. The data from model history obtained during adaptation are denoted as $Y^{(r-1)}$.

For the sake of proper model adaptation the standard exponential forgetting factor technique in parameter learning part of the algorithm [35] can be used. This approach enable to suppress an influence of distant data in model history during parameter estimation step. The exponential forgetting factor is stated by parameter ψ and afterwards the equations (7.10) become

$$\begin{aligned}\tilde{V}_{XX(r-1)} &= \psi \sum_{k=1}^{r-2} X_k X_k^T + X_{r-1} X_{r-1}^T , \\ \tilde{V}_{XY(r-1)} &= \psi \sum_{k=1}^{r-2} X_k Y_k^T + X_{r-1} Y_{r-1}^T , \\ \tilde{V}_{YY(r-1)} &= \psi \sum_{k=1}^{r-2} Y_k Y_k^T + Y_{r-1} Y_{r-1}^T .\end{aligned}\quad (7.11)$$

The 3D CAR model can be expressed as a stationary causal uncorrelated noise driven 3D autoregressive process:

$$Y_r = \Theta X_r + e_r , \quad (7.12)$$

where Θ is the $d \times d\eta$ parameter matrix

$$\Theta = [A_1, \dots, A_\eta] , \quad (7.13)$$

and

$$A_i = \begin{pmatrix} a_{1,1}^i & \dots & a_{1,d}^i \\ \vdots & \ddots & \vdots \\ a_{d,1}^i & \dots & a_{d,d}^i \end{pmatrix} \quad \forall i \in \{1 \dots \eta\} , \quad (7.14)$$

$\eta = \text{card}(I_r^c)$, I_r^c is a causal CN, e_r is a Gaussian white noise vector with zero mean and a constant but unknown covariance matrix Σ .

Optimal Support Set Estimation

The selection of an appropriate CAR model support is important to obtain good modelling results. Too small contextual neighbourhood can not capture all details while inclusion of surplus neighbours add the computational burden and can potentially degrade the performance of the model as an additional source of noise. The optimal neighbourhood can be found using the Bayesian decision rule for minimising the average probability of

decision error. Let us assume a set of CAR models (7.12) M_1, M_2, \dots which can differ either in the contextual neighbourhood I_{r_1, r_2, r_4} or / and in their exponential forgetting factor ψ (7.11). The optimal decision rule for minimising the average probability of decision error chooses the maximum a posterior probability model, i.e., a model whose conditional probability given the past data is the highest one. The presented algorithm can be therefore completed [41] as:

$$\tilde{Y}_r^i = \tilde{\Theta}_{r-1}^{i,T} X_{i,r} + e_{i,r} \quad \text{if } p(M_i|Y^{(r-1)}) > p(M_j|Y^{(r-1)}) \quad \forall j \neq i \quad (7.15)$$

where $X_{i,r}$ are data vectors corresponding to I_{r_1, r_2, r_4}^i . The Following Bayesian framework used in our paper, choose uniform a priori model in the absence of contrary information, $p(M_i|Y^{(t-1)}) \sim p(Y^{(t-1)}|M_i)$, and assume conditional pixel independence.

Thus the most probable CAR model given past data, the normal-Wishart parameter prior and the uniform model prior is the model which maximise the statistics [41] is then

$$p(M_j|Y^{(r-1)}) = k \exp\{D_j\} , \quad (7.16)$$

where

$$D_j = -\frac{d}{2} \ln |V_{XX(r-1)}| - \frac{\beta(r) - d\eta + d + 1}{2} \ln |\lambda_{(r-1)}| + \frac{d^2 \eta}{2} \ln \pi \\ \sum_{i=1}^d \left[\ln \Gamma \left(\frac{\beta(r) - d\eta + d + 2 - i}{2} \right) - \ln \Gamma \left(\frac{\beta(0) - d\eta + d + 2 - i}{2} \right) \right]$$

where k is a common constant and $\Gamma(n)$ is the Euler function. All statistics related to a model M_j $\tilde{V}_{XY(r-1)}$, $\tilde{V}_{XX(r-1)}$, are computed from data in $X_{j,r}$. The determinant $|V_{XX(r)}|$ as well as λ_r can be evaluated recursively see [40].

Parameter Estimation

There are two parameters $\hat{\Theta}_r, \hat{\Sigma}_r$ to estimate / update in each step, i.e., CN shift on image lattice. The first one is parameter matrix $\hat{\Theta}_r$ and the second one is noise covariance matrix $\hat{\Sigma}_r$. Because of the model causality the parameter estimations (7.17),(7.18) of the CAR model using the Bayesian method and the normal-Wishart parameter prior can be found analytically [40]. The estimate of parameter matrix is

$$\hat{\Theta}_{r-1}^T = V_{XX(r-1)}^{-1} V_{XY(r-1)} , \quad (7.17)$$

while the estimate of process-history-data covariance matrix is

$$\hat{\Sigma}_{r-1} = \frac{\lambda_{(r-1)}}{\beta(r)} , \quad (7.18)$$

where

$$\lambda_{(r)} = V_{YY(r)} - V_{XY(r)}^T V_{XX(r)}^{-1} V_{XY(r)} , \quad (7.19)$$

$$\begin{aligned} V_{XX(r-1)} &= \tilde{V}_{XX(r-1)} + V_{XX(0)} , \\ V_{XY(r-1)} &= \tilde{V}_{XY(r-1)} + V_{XY(0)} , \\ V_{YY(r-1)} &= \tilde{V}_{YY(r-1)} + V_{YY(0)} \end{aligned} \quad (7.20)$$

$$(7.21)$$

and matrices $V_{XX(0)}, V_{XY(0)}, V_{YY(0)}$ are the corresponding matrices from the normal-Wishart parameter prior. The estimates (7.17),(7.18) can be also evaluated recursively if necessary. Where the $\beta(r)$ represents number of model movements on image plane:

$$\beta(r) = \beta(0) + r - 1 , \quad (7.22)$$

$$\beta(0) > 1 . \quad (7.23)$$

Subspace Synthesis

The CAR model synthesis is very simple and the Markov random field can be directly generated from the model equation (7.12) with respect to CN data vector X_r and parameter matrix $\hat{\Theta}_r$ using a multivariate Gaussian white-noise generator. The fine-resolution synthetic texture is obtained from the pyramid collapse procedure, which is inverse process to the spatial factorisation (7.4),(7.6) described in Section 7.3.4.

7.3.6 2D Causal Auto-Regressive Subspace Model

Spectral factorisation (7.3) of multispectral subspace images into individual monospectral factors allows to use simpler 2D CAR model [37]. These single orthogonal monospectral factors of subspace image are further decomposed into a multi-resolution grid and each resolution data are independently modelled by their dedicated independent Gaussian noise driven autoregressive random field model (CAR) as follows.

The causal autoregressive random field (CAR) is a family of random variables with a joint probability density on the set of all possible realisations Y of the $M \times N$ lattice I , subject to following condition:

$$p(Y | \gamma, \sigma^{-2}) = (2\pi\sigma^2)^{-\frac{(MN-1)}{2}} \exp \left\{ \frac{-1}{2} tr \left\{ \sigma^{-2} \begin{pmatrix} -\alpha \\ \gamma^T \end{pmatrix}^T \tilde{V}_{MN-1} \begin{pmatrix} -\alpha \\ \gamma^T \end{pmatrix} \right\} \right\} , \quad (7.24)$$

where α is a unit vector, γ is parameter vector, σ is variance of Gaussian white noise and the following notation is used

$$\tilde{V}_{r-1} = \begin{pmatrix} \tilde{V}_{YY(r-1)} & \tilde{V}_{XY(r-1)}^T \\ \tilde{V}_{XY(r-1)} & \tilde{V}_{XX(r-1)} \end{pmatrix} ,$$

where $\tilde{V}_{XX(r-1)}, \tilde{V}_{XY(r-1)}, \tilde{V}_{YY(r-1)}$ are matrices defined in (7.10). Similarly to 3D CAR model we can employ model adaptation by means of exponential forgetting factor technique according to equation (7.11).

The 2D CAR model can be expressed as a stationary causal uncorrelated noise driven 2D autoregressive process:

$$Y_r = \gamma X_r + e_r , \quad (7.25)$$

where

$$\gamma = [a_1, \dots, a_\eta] \quad (7.26)$$

is the parameter vector, I_r^c is a causal neighbourhood with $\eta = \text{card}(I_r^c)$ and e_r is a white Gaussian noise with zero mean and a constant but unknown variance σ^2 and X_r is a corresponding vector of Y_{r-s} (see (7.7)).

Parameters Estimation

Parameter estimation of a CAR model using the maximum likelihood, the least square or Bayesian methods can be found analytically. The Bayesian parameter estimations of the causal AR model with the normal-gamma parameter prior which maximise the posterior density are:

$$\hat{\gamma}_{r-1}^T = V_{XX(r-1)}^{-1} V_{XY(r-1)} \quad (7.27)$$

and

$$\hat{\sigma}_{r-1}^2 = \frac{\lambda_{(r-1)}}{\beta(r)} \quad , \quad (7.28)$$

where

$$\lambda_{(r-1)} = V_{YY(r-1)} - V_{XY(r-1)}^T V_{XX(r-1)}^{-1} V_{XY(r-1)} \quad , \quad (7.29)$$

$$\begin{aligned} V_{XX(r-1)} &= \tilde{V}_{XX(r-1)} + V_{XX(0)} \\ V_{XY(r-1)} &= \tilde{V}_{XY(r-1)} + V_{XY(0)} \\ V_{YY(r-1)} &= \tilde{V}_{YY(r-1)} + V_{YY(0)} \quad , \end{aligned} \quad (7.30)$$

$$\beta(r) = \beta(0) + r - 1 \quad (7.31)$$

and submatrices in $V_{XX(0)}, V_{XY(0)}, V_{YY(0)}$ are from normal-gamma parameter prior. The estimates (7.27) can be also evaluated recursively if necessary.

Subspace Synthesis

The CAR model synthesis is very simple and a causal CAR random field can be directly generated from the model equation (7.25). Single CAR models synthesise spatial frequency bands of the texture. Each monospectral fine-resolution component is obtained from the pyramid collapse procedure (inversion process to (7.4),(7.6)). Finally the resulting synthesised colour texture is obtained from the set of synthesised monospectral images using the inverse K-L transformation (7.32).

$$\tilde{Y}_{r,\bullet} = T^{-1} \bar{Y}_{r,\bullet} \quad (7.32)$$

If a single visualised scene simultaneously contains BTF texture view and angle combinations which are modelled by different probabilistic models (i.e., models supported by different BTF subspaces) for the same material all such required subspace images are easily synthesised simultaneously. Simultaneous synthesis allows to avoid difficult subspace images registration problems.

7.3.7 2D Gaussian-Markov Random Field Subspace Model

Similarly to 2D CAR model the single orthogonal monospectral factors of subspace image are further decomposed into a multi-resolution grid and each resolution data are independently modelled by their dedicated independent Gaussian Markov random field model (GMRF).

The Markov random field is a family of random variables with a joint probability density on the set of all possible realisations Y of the lattice I , subject to following conditions:

$$p(Y_{\bullet,i}) > 0, \quad \forall Y, \quad (7.33)$$

and the Markov property:

$$p(Y_{r,i} | Y_{s,i} : \forall s \in I \setminus \{r\}) = p(Y_{r,i} | Y_{s,i} : \forall s \in I_{r,i}), \quad (7.34)$$

where $I_{r,i}$ is a contextual support set (CN) of the i -th monospectral field.

If the local conditional density of the MRF model (7.35) is Gaussian, we obtain the continuous Gaussian Markov random field model (GMRF):

$$p(Y_{r,i} | Y_{s,i} \forall s \in I_{r,i}) = (2\pi\sigma_i^2)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\sigma_i^{-2}(Y_{r,i} - \tilde{\mu}_{r,i})^2\right\}, \quad (7.35)$$

where the conditional mean value is

$$\tilde{\mu}_{r,i} = E\{Y_{r,i} | Y_{s,i} \forall s \in I_{r,i}\} = \mu_{r,i} + \sum_{s \in I_{r,i}} a_{s,i}(Y_{r-s,i} - \mu_{r-s,i}), \quad (7.36)$$

where μ_r is a local mean value and $\sigma_i, a_{s,i} \forall s \in I_{r,i}$ are unknown parameters. The 2D GMRF model can be expressed as a stationary non-causal correlated noise driven 2D autoregressive process:

$$\tilde{Y}_{r,i} = \sum_{s \in I_{r,i}} a_{s,i} \tilde{Y}_{r-s,i} + e_{r,i} \quad (7.37)$$

where the noise $e_{r,i}$ is random variable with zero mean. The $e_{r,i}$ noise variables are mutually correlated

$$R_{e_i} = E\{e_{r,i}e_{r-s,i}\} = \begin{cases} \sigma_i^2 & \text{if } s = (0, 0), \\ -\sigma_i^2 a_{s,i} & \text{if } s \in I_{r,i}, \\ 0 & \text{otherwise.} \end{cases} \quad (7.38)$$

Correlation functions have the symmetry property $E\{e_{r,i}e_{r+s,i}\} = E\{e_{r,i}e_{r-s,i}\}$ hence the neighbourhood support set and their associated coefficients have to be symmetric, i.e. $s \in I_{r,i} \Rightarrow -s \in I_{r,i}$ and $a_{s,i} = a_{-s,i}$.

Optimal Support Set Selection

The selection of an appropriate GMRF model support is important to obtain good results in modelling of a given random field. If the contextual neighbourhood is too small it can not capture all details of the random field. Inclusion of the unnecessary neighbours on the other hand add the computational burden and can potentially degrade the performance of the model as an additional source of noise. We use hierarchical neighbourhood system $I_{r,i}$, e.g., the first-order neighbourhood is $I_{r,i} = \{r - (0, 1), r + (0, 1), r - (1, 0), r + (1, 0)\}$, etc. thus the contextual neighbourhood is symmetric. An optimal neighbourhood is detected using the correlation method [39] favouring neighbours locations corresponding to large correlations over those with small correlations.

Parameter Estimation

The GMRF model does not fulfil causality assumption and mutual dependencies of pixels in non-causal symmetric contextual neighbourhood (7.38) leads to numerical non-linear Monte Carlo parameters estimation methods. To avoid this slow iterative estimation process the individual pixel values in CN are assumed to be conditionally independent and thus parameters estimation can be performed analytically by means of pseudo-likelihood or alternatively least-squares estimators. The corresponding pseudo-likelihood estimate for a_s parameters has the form

$$\gamma_i = [a_{s,i} : \forall s \in I_{r,i}] = \left[\sum_{\forall r \in I} X_{r,i}^T X_{r,i} \right]^{-1} \sum_{\forall r \in I} X_{r,i}^T Y_{r,i} , \quad (7.39)$$

where

$$X_{r,i} = [Y_{r-s,i} : \forall s \in I_{r,i}] \quad (7.40)$$

and

$$\sigma_i^2 = \frac{1}{MN} \sum_{r=1}^{MN} (Y_{r,i} - \gamma_i X_{r,i}^T)^2 . \quad (7.41)$$

Subspace Synthesis

The non-causal GMRF model generally requires a time consuming iterative Monte Carlo methods for data synthesis. However, when the input image, i.e., underlying regular rectangular random field index set, is defined on toroidal image lattice a simpler non-iterative methods can be employed [33] for a finite lattice GMRF synthesis. The most effective synthesis method uses the discrete fast Fourier transformation, which somewhat limits using of this model for fast GPU applications. According to [50] the GMRF can be generated from

$$Y_{\bullet,i} = \mathcal{F}^{-1} \{ \hat{Y}_{\bullet,i} \} + U_i , \quad (7.42)$$

where U_i the mean vector of the whole filed and $\hat{Y}_{\bullet,i}$ is generated from the Gaussian generator $\mathcal{N}(0, NMS_Y(r, i))$. $S_Y(r, i)$ is the associated power spectrum [34] and $N \times M$ is the underlying generated lattice size. Single GMRF models synthesise spatial frequency bands of the texture. Each monospectral fine-resolution component is obtained from the pyramid collapse procedure (inversion process to (7.4),(7.6)). Finally the resulting synthesised multispectral texture is obtained from the set of synthesised monospectral images using the inverse K-L transformation (7.32) in the same way as in previous 2D CAR model.

7.4 Results

The results of the proposed probabilistic BTF models are presented in two different ways. The first one is an approximation of spare set of the original BTF measurements which are visually compared with their original counterparts. The second one is BTF rendering on 3D object. The object surface exhibits many different combination of view and illumination directions and show overall behaviour of BTF model while it is compared with original tiled BTF data.

7.4.1 Synthesis of Individual BTF Images

The most straightforward way of verification of the proposed BTF model results is a comparison of synthesised images with original BTF measurements. In this initial experimental part we picked up three BTF measurements corresponding to fixed viewing position (θ_v, ϕ_v) while the illumination elevation angle θ_i and the illumination azimuthal angle ϕ_i significantly differs: $\theta_i = 45^\circ, \phi_i = 0^\circ$, $\theta_i = 60^\circ, \phi_i = 90^\circ$ and $\theta_i = 75^\circ, \phi_i = 180^\circ$ (see Fig. 2.2).

We used three distinct BTF materials from the University of Bonn BTF database [98] containing regular surface structure: *fabric02*, *foil02* and *knitted wool*. Additional test was performed on *cushion fabric* rough texture measurements from the UTIA texture database. Figs. 7.7, 7.8, 7.9 and 7.10 shows results of our tests for all proposed probabilistic BTF models: GMRF, 2D CAR and 3D CAR. The upper row in each of these images illustrates three distinct cutouts from original BTF measurements accompanied by material range-map estimated by photometric stereo (see Section 5.2.3). The second row shows corresponding underlying synthesised images for individual MRF models. The third row shows combination of these synthesised images with the range-map according to light position by means of displacement mapping technique (see Section 4.3.2). The depicted synthesised images are encouraging, however, our approach can not handle all effects occurring when an arbitrary real-world material is lit from different directions. E.g., fabric materials woven from distinct material threads being oriented to different directions. This combination leads to distinct reflectance properties of the material dependently on a given surface location. These material attributes can not be reproduced by means of displacement filter and their correct modelling require substantially more complex physical model of individual mesostructure elements, e.g., woven knits, metal grooves etc. This effect can be observed for *fabric02* material in Fig. 7.7 and *cushion fabric* in Fig. 7.10. In the first original image (upper left) for the both materials are visible light areas which were not correctly reproduced by proposed probabilistic BTF model.

A similar problem occurs for translucent materials. One example of such a material where the Lambertian assumption does not hold is *knitted wool* in Fig. 7.9. Mainly the synthesis of the second image does not correspond to the original and the synthesised images looks like a rigid surface. In spite of these shortcomings the proposed BTF model is capable of reliable approximation of many real-world materials. The best model performance was obtained for leather. For instance the synthesised results for *foil02* material in Fig. 7.8 are almost visually indistinguishable from their original patterns. However, since the overall mesostructure appearance is driven by underlying smooth MRF model the selection of suitable synthesis is very important. Generally we can say, that the more structured texture is in the synthesised image the less noticeable is the influence of range information on final BTF image synthesis. This is apparent in 2D CAR model synthesis in Fig. 7.8 in comparison with GMRF model synthesis. Choosing the optimal model, i.e., the corresponding set of parameters, have significant influence on MRF model stability during synthesis of subspace images. Additionally, in some cases the MRF model produces synthesised images containing artifacts caused by simplifying causality assumption (see last 3D CAR model synthesis in Fig. 7.8). This can be avoided by visual inspection of all synthesised subspace images and using the set of model parameters producing stable synthesis for all subspace images.



Figure 7.7: *Fabric02* synthesised BTF images obtained using proposed probabilistic BTF models: GMRF, 2D CAR and 3D CAR respectively compared with corresponding raw BTF measurements for three distinct illumination directions. Range data were introduced into the models by means of displacement mapping.



Figure 7.8: *Foil02* synthesised BTF images obtained using proposed probabilistic BTF models: GMRF, 2D CAR and 3D CAR respectively compared with corresponding raw BTF measurements for three distinct illumination directions. Range data were introduced into the models by means of displacement mapping.

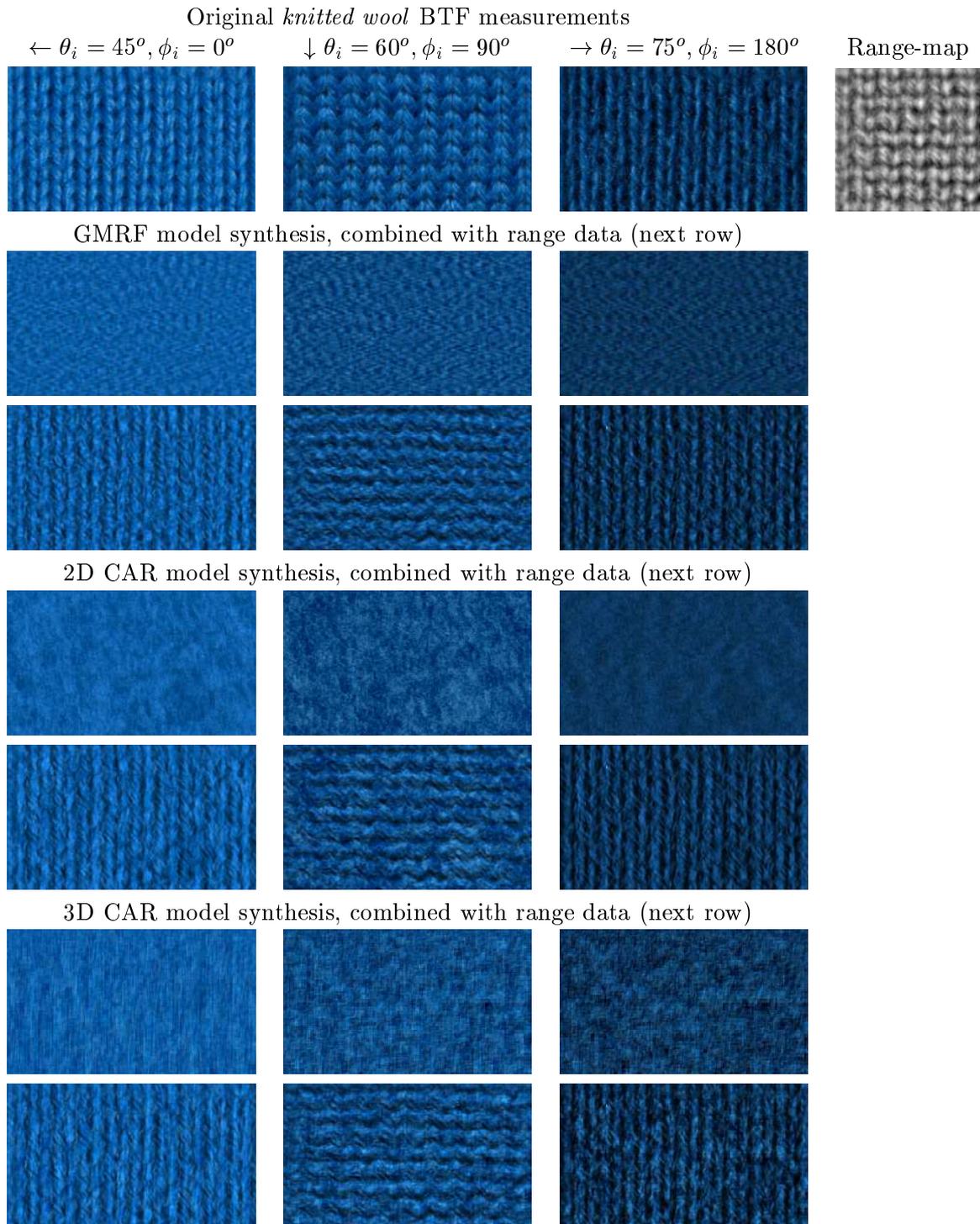


Figure 7.9: *Knitted wool* synthesised BTF images obtained using proposed probabilistic BTF models: GMRF, 2D CAR and 3D CAR respectively compared with corresponding raw BTF measurements for three distinct illumination directions. Range data were introduced into the models by means of displacement mapping.

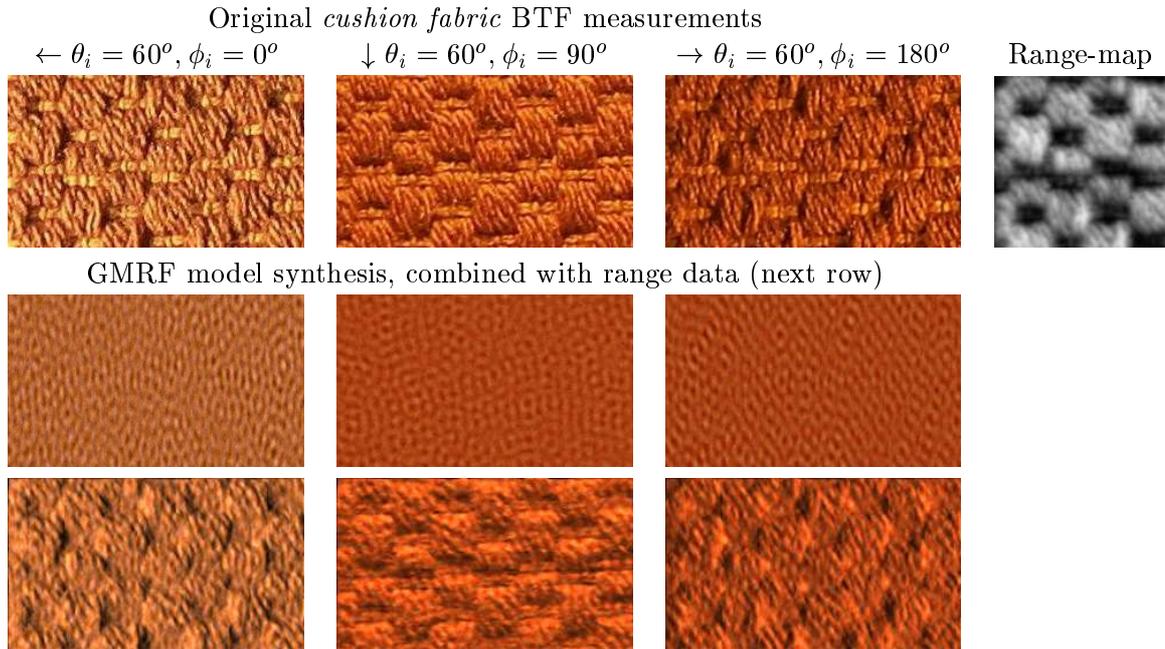


Figure 7.10: *Cushion fabric* synthesised BTF images obtained using proposed probabilistic GMRF BTF model compared with corresponding raw BTF measurements for three distinct illumination directions. Range data were introduced into the model by means of displacement mapping.

7.4.2 Rough Texture Model from Spare Set of Textures

The proposed rough texture model can be obtained from several or in extreme case from one texture image/images as illustrated in this section. However, such a simplification strictly requires Lambertian surface so only a limited group of materials can be represented in this way otherwise the results will not correspond to real material reflectance properties. However, satisfactory rough textures synthesis can be obtained also for slightly non-Lambertian surfaces as shown in case of three different kinds of leather from UTIA texture database: *white leather*, *snake leather* and *cushion leather*. Fig. 7.11 shows three different synthesised rough texture images for different light azimuth compared with the original texture for each material, its range-map estimate and the 3D CAR model synthesis.

Additionally, Fig. 7.12 depicts sphere covered with two mentioned material (*cushion*, *snake leather*) to provide better demonstration of rough texturing. The rough material regular structure was introduced into the model by means of displacement mapping technique. The ultimate advantage of this model is option of rough texture synthesis of arbitrary resolution from a spare set of BTF images or in the extreme case from a single texture image only. The range-map of all three example materials illustrated in Fig. 7.11 was estimated from one image only by means of the shape from shading method described in Section 5.2.2. The range map was further tiled to produce range image of arbitrary size so the final storage demands of such a model are approximately 150KB including tiled range-map and MRF model parameters.

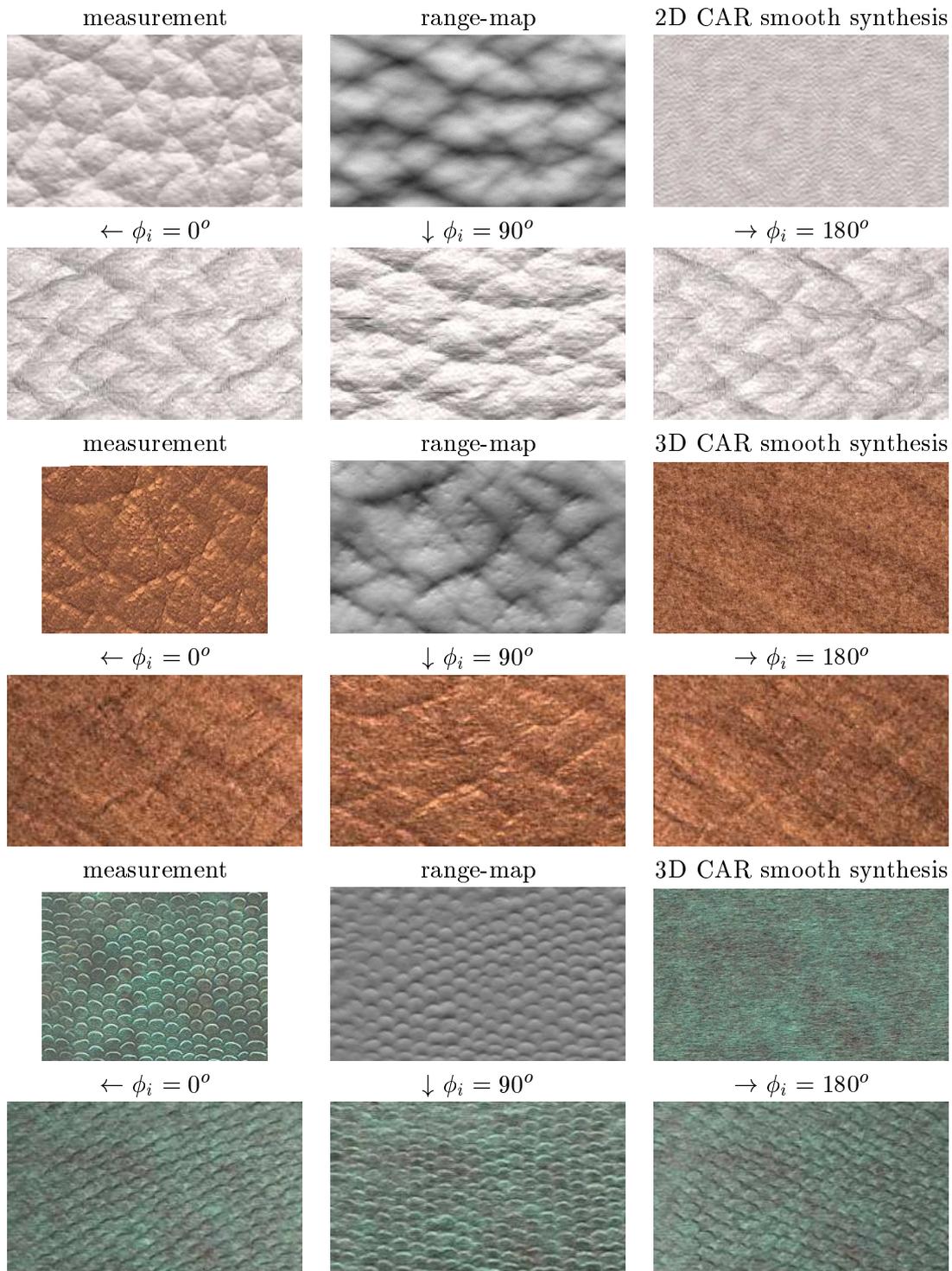


Figure 7.11: *White, cushion* and *snake leather* examples, corresponding estimated range-map, smooth 3D CAR synthesis and their BTF synthetic results rendered for illumination elevation angle $\theta_i = 60^\circ$ and azimuth angles $\phi_i = 0^\circ, 90^\circ$ and 180° .

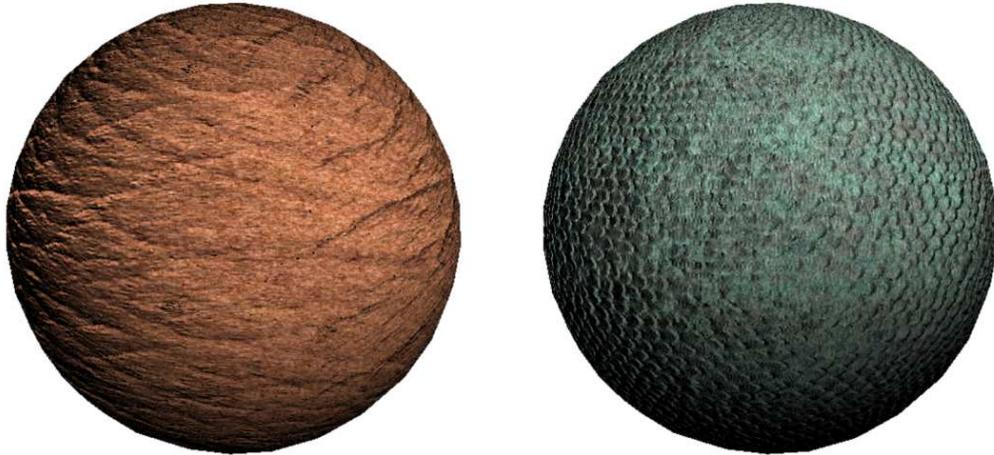


Figure 7.12: Rough texture rendering example by means of bump mapping technique. Sphere covered by *cushion leather* (left) and *snake leather* (right).

7.4.3 BTF on 3D Objects

We have tested the proposed probabilistic BTF models again on materials from BTF texture database of the University of Bonn [98] such as corduroy, upholstery, carpet, knitwear or leather textures and several UTIA BTF measurements. The resolution of University of Bonn BTF images (800×800) is satisfactory for parameters estimation of proposed MRF models. As a 3D object we used Mercedes Class-C armrest (courtesy of DaimlerChrysler). The resolution of synthesised texture mapped on this object was set to 512×512 . To reproduce all visible structure details, the range and normal maps were enlarged by means of image tiling method presented in [103]. This method cuts range and normal tiles according to sub-optimal path search algorithm from raw range and normal maps estimates. These surface height data are estimated using a photometric stereo technique in resolution of original BTF measurements. Therefore only several small image tiles together with the tile index file are stored taking about 100KB in average dependently on structure of the material. The 3D object is lit by single point-light source and textures on each polygon for given illumination and view direction are result of BTF interpolation between the closest BTF measurements available as is described in Section 4.2. Detail description of illumination and view angles computation for arbitrary scene polygon is given in Section 4.1. Figures 7.13, 7.14, 7.15 and 7.16 illustrate the results of the proposed probabilistic MRF models for the individual BTF materials from the University of Bonn compared with the original tiled data mapped on a car armrest 3D model. One can observe a slightly compromised visual quality of the proposed modelling approach for translucent and strongly non-Lambertian materials (e.g., fabrics in Figures 7.14 and 7.16). The regular rough structure in Fig. 7.13 was introduced into the model by means of the parallax bump-mapping (see Section 4.3.1). The results of proposed probabilistic BTF modelling approach with underlying 2D CAR model mapped on car gearbox are compared with original tiled BTF data and with results of proposed one-lobe reflectance model (PLM-C) in Figs. 9.2 and 9.3.

Note that the shown examples of synthesised BTFs are brighter than their original



Figure 7.13: Original BTF on car armrest for *wood01* and *wood02* materials (first rows) compared with results of GMRF, 2D CAR and 3D CAR BTF models respectively. Regular surface structure introduced by means of bump mapping.



Figure 7.14: Original BTF on car armrest for *fabric01* and *fabric02* materials (first rows) compared with results of GMRF, 2D CAR and 3D CAR BTF models respectively. Regular surface structure introduced by means of bump mapping.



Figure 7.15: Original BTF on car armrest for *foil01* and *foil02* materials (first rows) compared with results of GMRF, 2D CAR and 3D CAR BTF models respectively. Regular surface structure introduced by means of bump mapping.



Figure 7.16: Original BTf on car armrest for *leather02* and *wool* materials (first rows) compared with results of GMRF, 2D CAR and 3D CAR BTf models respectively. Regular surface structure introduced by means of bump mapping.

counterpart (first row). This is caused by additional ambient lighting which was necessary for bump mapping to work. The original tiled BTF data in the first row are interpolated on individual polygons without any additional lighting which is, however, important to produce shadows of objects in scene etc.

The comparison of smooth texture, only enhanced by bump mapping, with results of the proposed 3D MRF BTF model for all eight tested materials is shown in Figures A.1 and A.2. The first column shows combination of smooth albedo texture combined with bump mapping only while the second and third columns represent the proposed BTF synthesis mapped on a cylinder and lit from two distinct positions.

7.5 MRF BTF Model Fast Implementation Issues

The fast implementation of the synthesis part of the proposed BTF model based on underlying MRF model can be performed with hardware support of contemporary low-end graphics card. This equipment enable to run user defined fragment and vertex programs directly in GPU avoiding bandwidth problems caused by huge data transfer between GPU and CPU and enabling significant increase of computational performance.

The most appropriate candidate for such an implementation from MRF BTF models presented in this chapter is the BTF model with underlying 2D CAR texture model. This approach enables factorisation of subspace synthesis into synthesis of individual monospectral planes. The synthesis of remaining models requires either relatively slow FFT (GMRF model) or involves simultaneous evaluation of high number of models parameters (3D CAR model).

A fast hardware implementation of chosen 2D CAR BTF model requires three main steps. The first one is synthesis of individual subspace images, the second one is interpolation of synthesised images with respect to actual view and illumination direction while the third step handles the bump-mapping of a rough material structure.

7.5.1 Synthesis of BTF Subspace Images

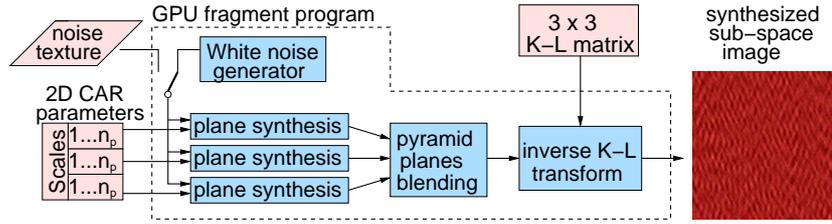
The synthesis of subspace images requires several operations to be performed for each individual image plane. The first one is subsequent image generation according to the 2D CAR model parameters with respect to a white-noise generator. In this way all layers of multiresolution pyramid (at most 3) are generated and finally blended together. The noise generator can be represented by means of the float-point texture for fixed image-size applications or it can be implemented in GPU as well. Finally, every vector of RGB pixels of each subspace image is multiplied by the inverse Karhunen-Loeve 3×3 matrix to preserve original colour correlations. According to the schema in Fig. 7.17, the input of the fragment CG program for one subspace image reconstruction is a set of the CAR model parametric vectors of the length 3 and the inverse K-L matrix of size 3×3 . The usual number of model parameter vectors n_p varies between 6 and 18 depending on the size of model's contextual neighbourhood. The output of the program is the synthetic colour BTF image.

All these computations can be efficiently performed using CG fragment programs and rendering-to-texture technique taking advance of contemporary graphics hardware. The CPU synthesis of all subspace images takes at most several seconds in average on Athlon 1.9GHz as shown in Tab. 7.1. Obviously any hardware implementation can considerably

Table 7.1: The synthesis time of 15 BTF subspace images (512×512) using proposed MRF BTF models on Athlon 1.9GHz.

<i>No. of G-L pyramid planes / CN size</i>	<i>3D CAR</i>	<i>2D CAR</i>	<i>GMRP</i>
1 / 2	5.9 s	3.3 s	16.2 s
3 / 2	8.0 s	3.9 s	19.4 s

speed up this process to enable synthesis at interactive frame rates. Subspace synthesis

**Figure 7.17:** Fast subspace synthesis scheme.

is performed only once for the first time when the corresponding material appears on the scene so this part does not later require any additional computation during an online rendering. So the numerical efficiency of whole rendering depends mainly on the efficiency of following two implementation steps and on GPU attributes.

7.5.2 BTF Interpolation

Due to relatively sparse measurement of the original BTF data space, the replacement of unmeasured BTF data with the nearest available illumination and view directions often produces significant seams visible on the surface of textured object. This problem occurs when the BTF data are used either for raw data mapping or learning the BTF texture synthesis model according to the cluster index. These artifacts were considerably suppressed when an interpolation scheme based on barycentric coordinates was applied (see Section 4.2). The three closest BTF measurements, in sense of vector Euclidean distance on hemisphere, containing individual measurements points are found for the given view and illumination directions. The resulting barycentric weights are computed for both three closest view and illumination directions. By multiplication of view and illumination weights we obtain the nine weights corresponding to nine synthesised images which are picked up with respect to cluster index file and combined by means of multitexturing or fragment programs.

7.5.3 Surface Height Simulation

There are variety on bump-mapping GPU implementations presented so far. In our case we have used the parallax bump mapping presented in [118]. This method enables fast bump-mapping effects by means of simple vertex and fragment programs. It approximates correct appearance of rough surfaces by modifying the texture coordinate for each pixel with no extra polygon requirements using only surface range a normal maps. These maps

are obtained from BTF measurements using the photometric stereo (see Section 5.2.3) and arbitrarily enlarged by image-tiling technique computed in accordance with [103] (see Section 8.3.1). Thus only a few range/normal-map tiles have to be stored together with the corresponding tile-index.

Chapter 8

BTF Modelling Using Reflectance Models

Reflectance models are mostly introduced as parametric functions representing amount of energy reflected by a material dependently on illumination and view direction with respect to a surface normal (see Fig. 2.2). If we are able to fit a parameters of such a model to all characteristic features of the original material reflectance and if they are fitted correctly, then the model can produce reflectance values which are indiscernible from those of the original material.

The simplest reflectance model is the *Lambertian model*, representing material with constant reflectance function for arbitrary view direction. Unfortunately such assumption for most of the materials does not hold due to their variable reflectance values for different view directions. Several reflectance models have been applied in the computer graphics in the past. These models are discussed in Chapter 2.

Most of BRDF models presented in Chapter 2 can be extended to BTF modelling using dedicated pixel-wise BRDF (i.e., ABRDF) models for each planar BTF positions. Thus in this case modelling of the BTF image for a given illumination and view direction consists of pixel-wise computation of the corresponding reflectance values from the model parameters estimated in a preceding offline phase.

For the purpose of fast BRDF rendering in graphics hardware the Lafortune model [63] became popular, because it enables relative easy and fast evaluation, compact BRDF representation and physical plausibility.

8.1 Lafortune Model

Monospectral BRDF is a four-dimensional function depending on a local viewing (ω_v) and illumination (ω_i) direction where

$$\omega_i(\theta_i, \phi_i) = \begin{bmatrix} \cos \theta_i \cos \phi_i \\ \cos \theta_i \sin \phi_i \\ \sin \theta_i \end{bmatrix}^T = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}^T, \quad \omega_v(\theta_v, \phi_v) = \begin{bmatrix} \cos \theta_v \cos \phi_v \\ \cos \theta_v \sin \phi_v \\ \sin \theta_v \end{bmatrix}^T = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}^T. \quad (8.1)$$

For multispectral modelling three different BRDFs are used for the individual spectral channels. BRDF usually represented as a 4D table involves storing large amount of data so some way of BRDF space parametrisation is inevitable.

For parametrisation of BRDF in the scope of this thesis we have developed the generalisation of Lafortune reflectance model because of its efficient and compact reflectance representation. Moreover, the simplicity of this model enables its application in real-time rendering algorithms implemented directly in contemporary graphics hardware.

The Lafortune model [63] provides a physically correct BRDF approximation using set of reflectance lobes. The model is a generalisation of the original cosine model:

$$Y_{i,v} = \rho K \cos^n \alpha \quad (8.2)$$

where α is the angle between the view direction ω_v and the mirror direction of the illumination direction ω_i , denoted by ω_m and K is the normalisation factor enforcing maximum lobe albedo ρ into a value between 0 and 1. The cosine part of the model can be written as a dot product

$$Y_{i,v} = \rho K [\omega_m \omega_v]^n \quad (8.3)$$

and the mirroring around the normal \mathbf{n} can be written using a Householder matrix

$$Y_{i,v} = \rho K [\omega_i^T (2\mathbf{nn}^T - \mathbf{I}) \omega_v]^n . \quad (8.4)$$

The model can be generalised by replacing the Householder matrix and normalisation factor K by a general 3×3 matrix \mathbf{M} according to

$$Y_{i,v} = \rho [\omega_i^T \mathbf{M} \omega_v]^n . \quad (8.5)$$

In order to obtain the reciprocal reflectance function (the same reflectance value if positions of light and camera are swapped) the matrix has to be symmetrical $\mathbf{M} = \mathbf{M}^T$. When a singular value decomposition of matrix \mathbf{M} is applied we obtain $\mathbf{Q}^T \mathbf{D} \mathbf{Q}$ where \mathbf{Q} is transformation matrix into the new orthogonal coordinate system. In this new system the matrix \mathbf{M} simplifies to the diagonal matrix \mathbf{D} . In this situation the axes are aligned to the normal and to the principal directions of anisotropy. The diagonal matrix can be assumed to be composed of the weights of individual terms of the dot-product $\omega_i \cdot \omega_v$ as it is shown in the following statement of the Lafortune reflectance model:

$$Y_{i,v} = \rho [\omega_i^T \mathbf{D} \omega_v]^n = \rho (D_x u_x v_x + D_y u_y v_y + D_z u_z v_z)^n . \quad (8.6)$$

The model can be extended to n_l reflectance lobes to be able to fit the complex reflectance functions as follows

$$Y_{i,v} = \sum_{k=1}^{n_l} \rho_k [\omega_i^T \mathbf{D}_k \omega_v]^n = \sum_{k=1}^{n_l} \rho_k (D_{x,k} u_x v_x + D_{y,k} u_y v_y + D_{z,k} u_z v_z)^{n_k} . \quad (8.7)$$

The representation using this model (8.7) is compact and relatively memory efficient since each reflectance lobe is determined by means of only five parameters ρ, D_x, D_y, D_z, n . The model is able to handle noisy data and even in case when data are sparse the model provides their correct interpolation. The obtained reflectance functions are physically plausible, inherently reciprocal and they satisfy the rule of energy-conservation. Moreover, the individual spectral channels of reflectance data can be modelled separately.

These properties and the simplicity make this model suitable for the fast BTF rendering algorithms implementable directly in graphics hardware. The following section explains the way of employing of the described Lafortune model for this task.

8.2 Sample-Size Lafortune BTF Model

Monospectral BTF is a six-dimensional function whose correct modelling involves either very complex and computationally demanding model or some kind of data factorisation into lower-dimensional data spaces where simpler models can be used. Actually, the six-dimensional BTF can be considered a spatially varying four-dimensional “apparent” BRDF (ABRDF). This fact enables the modelling of these per-pixel ABRDFs by dedicated Lafortune models as proved in [75], [19] and [78]. If all measured reflectance values are used as input data for a BRDF model (see [75] and [59]) then the results are unsatisfactory especially in the case of rough textures. The reason for this is a self-occlusion effect in BTF images of a rough materials where some parts of the material are occluded especially for high grazing angles and as a result individual pixels in the rectified BTF images do not correspond to the unique planar position on the material surface. Therefore the only way to obtain a real pixel-wise registration of BRDFs is to model only the individual reflectance field \mathcal{R} , i.e., n_i images taken for a fixed view and varying illumination direction. So for each of n_v reflectance fields \mathcal{R}_v the parameters of the Lafortune model are computed for every BTF planar position. This procedure is done separately for all RGB colour channels. As the parameter computation is independent of the view direction ω_v (which is fixed for a given reflectance field) the model equation becomes

$$Y_{i,v}(r) = \sum_{k=1}^{n_l} \rho_{v,k}(r) [\omega_i^T \mathbf{D}_{v,k}(r)]^{n_{v,k}(r)} = \quad (8.8)$$

$$= \sum_{k=1}^{n_l} \rho_{v,k}(r) (D_{v,x,k}(r)u_x + D_{v,y,k}(r)u_y + D_{v,z,k}(r)u_z)^{n_{v,k}(r)} , \quad (8.9)$$

$$(8.10)$$

where $\omega_i(\theta_i, \phi_i) = [u_x, u_y, u_z]^T$ is a unit vector pointing to light and parametrised by the illumination elevation and azimuthal angles $[\theta_i, \phi_i]$ respectively (see Fig. 2.2). As a reflectance data the set of pixels $\mathcal{R}_v(r_1, r_2, r_3, \omega_i)$ is considered, where $i = 1, \dots, n_i$ is the illumination position index and v is the actual view position index ranging from 1 to n_v . A multiindex $r = \{r_1, r_2, r_3\}$ represents the planar horizontal, vertical and spectral index in the BTF image respectively.

The individual colour reflectance fields \mathcal{R}_v are represented by means of n_l sets of five floating point parametric images corresponding to the model parameters ρ, D_x, D_y, D_z, n respectively. This is still relative large amount of data in comparison with the 81 original images particularly when more than one reflectance lobe is used. In addition the parameter fitting for several lobes is underconstrained, time consuming problem and as a result the estimation is often numerically unstable. For these reasons we decided to use only simplified one-lobe variant of the Lafortune reflectance model (LM):

$$Y_{i,v}(r) = \rho_v(r) [\omega_i^T \mathbf{D}_v(r)]^{n(r)} = \rho_v(r) (D_{v,x}(r)u_x + D_{v,y}(r)u_y + D_{v,z}(r)u_z)^{n(r)} \quad (8.11)$$

The representation of BTF by means of one-lobe LM with five parameters involves storing of $5 \times n_v = 405$ floating point colour parametric images instead of $81 \times 81 = 6561$ original BTF images.

8.2.1 Non-Linear Estimation of Model Parameters

For every planar position (r_1, r_2) in the estimated BTF image one model is used, so for each view direction from the BTF database we have to estimate $N \times N$ models for each spectral channel. The model parameters are estimated using Levenberg-Marquardt non-linear estimation technique as described in [94]. The Levenberg-Marquardt method is a “single-shot” method which attempts to find the local fit-statistic minimum nearest to the initialising point. Its principal advantage is that it uses information about the first and second derivatives of the fit-statistic as a function of the estimated parameter values to guess the location of the fit-statistic minimum. It will not work reasonably well with complex statistic surfaces. Apart from that there is no guarantee to find the global fit-statistic minimum.

In practical experiments including estimation of model parameters for every view direction contained in the BTF database, it turned out that the fitting quality of the optimised model parameters strongly depends on their initial value. When the initial values were manually tuned and fixed for all computed models then in the estimated BTF images appeared isolated dots with completely different spectral values than those expected from the original. This situation is illustrated in Fig. 8.1-b in comparison with the original BTF image Fig. 8.1-a. To overcome this problem the parameter estimation process was split in two separate steps. In the first step the parameters of the model are fitted to manually tuned initial values. The correct initial values for the following second estimation step are finally obtained as median of parameter values over all models in \mathcal{R}_v . Results of such a correct parameter initialisation is depicted in Fig. 8.1-c.

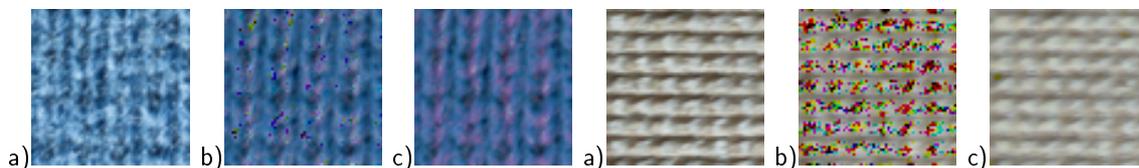


Figure 8.1: Example how the improper initial values of model parameters influence the restored BTF image quality for two different materials. The first image is an original (a), the second one is a result based on the estimated model parameters with wrong initial values (b) and the third image illustrates the result with the correctly set initial values (c).

8.2.2 Proposed Polynomial Extension of the Lafortune Model

The one-lobe Lafortune model needs to store considerably lower number of parameters than the n_l -lobes model. However, one can assume that the performance of such a simplified model on reflectance data would not be satisfactory. In our experiments testing it became obvious that the reflectance function approximation by one-lobe LM is erroneous especially in cases of complex BRDF when the reflectance values strongly depend on the illumination direction. This situation is most apparent at higher grazing angles when the light shines from direction opposite to camera which causes significant specular reflections. The problem of one-lobe LM fitting to original data is illustrated in Fig. 8.2. We computed BRDF for BTF images of *knitted wool* material from $\mathcal{R}_v - \theta_v = 60^\circ$, $\phi_v = 54^\circ$ as an average RGB value in a window of size 20×20 (solid line) and compared it with

BRDF obtained in the same way from estimated BTF images corresponding to the same \mathcal{R}_v (dashed line). From the results we conclude that the used one-lobe model is not able to follow such steep changes of reflectance function, which are present in almost all natural materials. The significance of this one-lobe model error considerably depends on properties of individual material sample and on the actual viewing direction. Generalisation

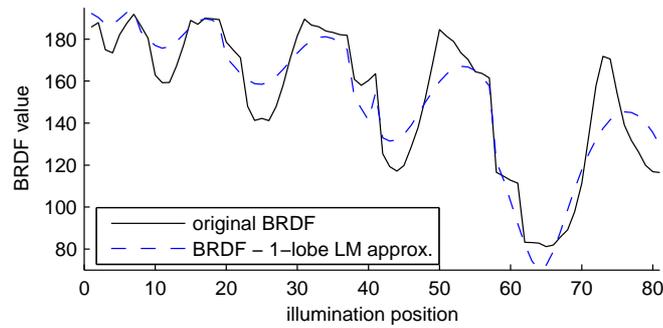


Figure 8.2: Comparison of the original BRDF (solid line) and BRDF approximated by one-lobe Lafortune model (LM) (blue, dashed line) for *knitted wool* material for $\mathcal{R}_v - \theta_v = 60^\circ$, $\phi_v = 54^\circ$.

of the Lafortune model with additional reflectance lobes improves its performance, however, such a model still fails at grazing angles where the considerable difference between original and estimated reflectance values persists. In addition the n_l lobes model implies undesirable increase of n_l -times more parameters to be stored. Moreover, the lobes-fitting procedure is then more complex and time consuming. Even the general Lafortune model with all the parameters in the parameter matrix \mathbf{M} (not only the diagonal or symmetric ones) did not significantly suppress the undesirable artifacts.

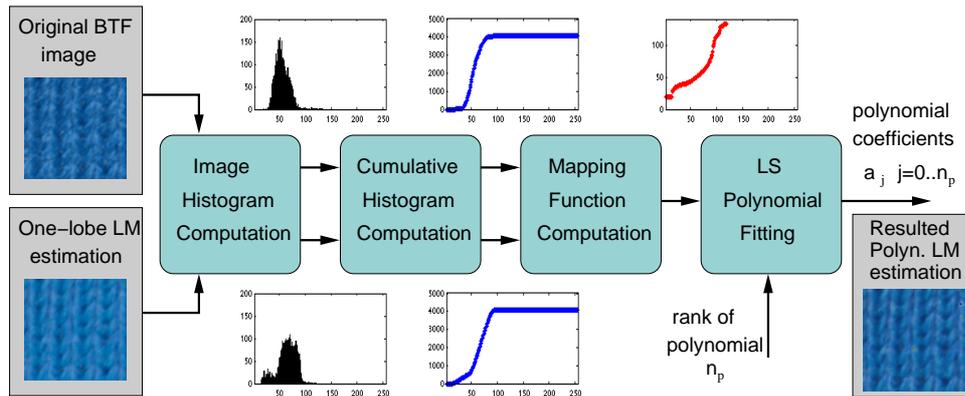


Figure 8.3: Procedure of polynomial coefficients computation.

In order to improve the low fitting accuracy of the described one-lobe model we introduced its polynomial extension [27]. This method is based on a histogram matching technique [31] adopted and extended for BTF data and polynomial fitting as illustrated on the scheme in Fig. 8.3. At the beginning the image histograms in all spectral channels are computed for both the original and the one-lobe LM approximation of every BTF

image. From these histograms we compute the cumulative histograms. These histograms represent the inputs of the algorithm Alg. 4 which computes the coefficients for the polynomial mapping of grayscale levels of the LM estimated image to the original one. The algorithm is based on histogram matching of both images so that the histogram of the LM estimated image is fitted with respect to the original BTF image. The resulting mapping between both cumulative histograms is approximated by polynomial using least squares fitting scheme.

The resulting polynomial coefficients $a_{r_3,v,i,j}$ are stored for individual colour channels of every BTF image. The proposed polynomial extension of the one-lobe Lafortune model (PLM) using coefficients $a_{r_3,v,i,j}$ can be described by following equation

$$\tilde{Y}_{i,v}(r) = \mathcal{M}_{r_3,v,i}(Y_{i,v}(r)) = \sum_{j=0}^{n_p} a_{r_3,v,i,j} Y_{i,v}(r)^j \quad (8.12)$$

which results in a novel model expressed by the following formula:

$$\tilde{Y}_{i,v}(r) = \sum_{j=0}^{n_p} a_{r_3,v,i,j} [\rho_v(r) (\omega_i^T \mathbf{D}_v(r))^{n_v(r)}]^j . \quad (8.13)$$

Here $a_{r_3,v,i,j}$ are polynomial parameters specifying the mapping function $\mathcal{M}_{r_3,v,i}$ between the histogram values of the image $Y_{i,v}(r)$ (synthesised from one-lobe model parameters) and the original BTF image and $(n_p - 1)$ is a rank of this polynomial. The parameters $a_{r_3,v,i,j}$ are estimated by least squares fitting on the original 8 bits quantised mapping function. We obtained satisfactory results already with $n_p = 5$.

Algorithm 4: Mapping function coefficients computation

1. **Input:** Cumulative histograms of the original BTF image and its one-lobe LM estimate
 2. $i = 0, j = 0$
 3. WHILE ($j \leq 255$)
 - (a) WHILE ($histCumLM[i] \leq histCumOrig[j]$)
 - i. $\mathcal{M}[i] = j$
 - ii. $i = i + 1$
 - (b) $j = j + 1$
 4. Fit 5-order polynomial to \mathcal{M} using Least Squares method.
 5. **Output:** Polynomial coefficients $a_j, j = 0 \dots n_p$ representing mapping function \mathcal{M} .
-

Finally for each BTF image we have to store fifteen additional float polynomial parameters (five float numbers for each colour channel). The number of these parameters stored within each BTF image is negligible in comparison to the number of one-lobe Lafortune

model parameters stored for each image pixel. The extended model involves only several additional linear operations which can be possibly implemented in contemporary graphical hardware. Moreover, the results of the proposed polynomial extension of one-lobe model are encouraging as it is shown, e.g., in the shape of reflectance lobe which is compared with original measured BRDF lobe and lobe estimated by one-lobe model only in pixel ($r_1 = 13, r_2 = 3$) of *knitted wool* as illustrated in Fig. 8.4. More thorough comparison of model performance is provided in the following sections.

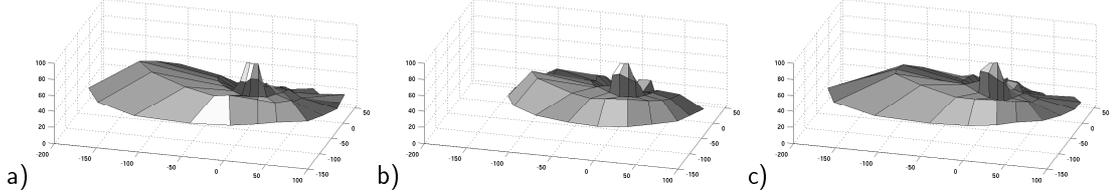


Figure 8.4: Reflectance lobes in pixel ($r_1 = 13, r_2 = 3$) of *knitted wool* and illumination angles $\theta_i = 75^\circ, \phi_i = 54^\circ$. Original reflectance lobe (a), approximated by the one-lobe LM (b) and by the proposed PLM (c).

8.2.3 Results of Sample-Size Reflectance Models

In order to verify the considered reflectance models we performed extensive experiments using eight BTFs provided by University of Bonn [98]. The LM parametric images were estimated for all 81 surface reflectance fields \mathcal{R}_v , $v = 1 \dots n_v$ as well as the polynomial coefficients corresponding to the histogram matching mapping functions $\mathcal{M}_{r_3, v, i}$. All BTF images were synthesised by means of the estimated parameters. For all materials the synthesised BTF images based on the one-lobe LM were compared with the BTF synthesis using the proposed PLM.

For the sake of the BTF results comparison the standard mean average error (MAE) between original data (Y) and estimated data (\hat{Y}) was used (8.14).

The mean average error for one BTF image is given by

$$\mathcal{E}_v(i) = \frac{1}{255\nu} \sum_{\forall r_1, r_2, r_3 \in Y} |Y(r_1, r_2, r_3, v, i) - \hat{Y}(r_1, r_2, r_3, v, i)|, \quad (8.14)$$

where r_1, r_2, r_3 represent the planar horizontal, vertical and spectral index respectively in BTF the image with resolution $\nu = N_x \times N_y$ and i is the illumination position index in \mathcal{R}_v . For computation of the average error for all estimated images from \mathcal{R}_v we used the formula

$$\mathcal{E}_v = \frac{1}{n_i} \sum_{i=1}^{n_i} \mathcal{E}_v(i), \quad (8.15)$$

where n_i is the number of images included in \mathcal{R}_v .

Graphs in Fig. 8.5 demonstrate the performance in terms of MAE for both methods (LM depicted as blue curve and PLM depicted as red curve) on the whole BTF, i.e., for all 81 reflectance fields \mathcal{R}_v along x axis, for eight different materials from the Bonn University BTF database. Individual reflectance fields are ordered according to camera position circular movement from top to bottom of a hemisphere above the observed material as

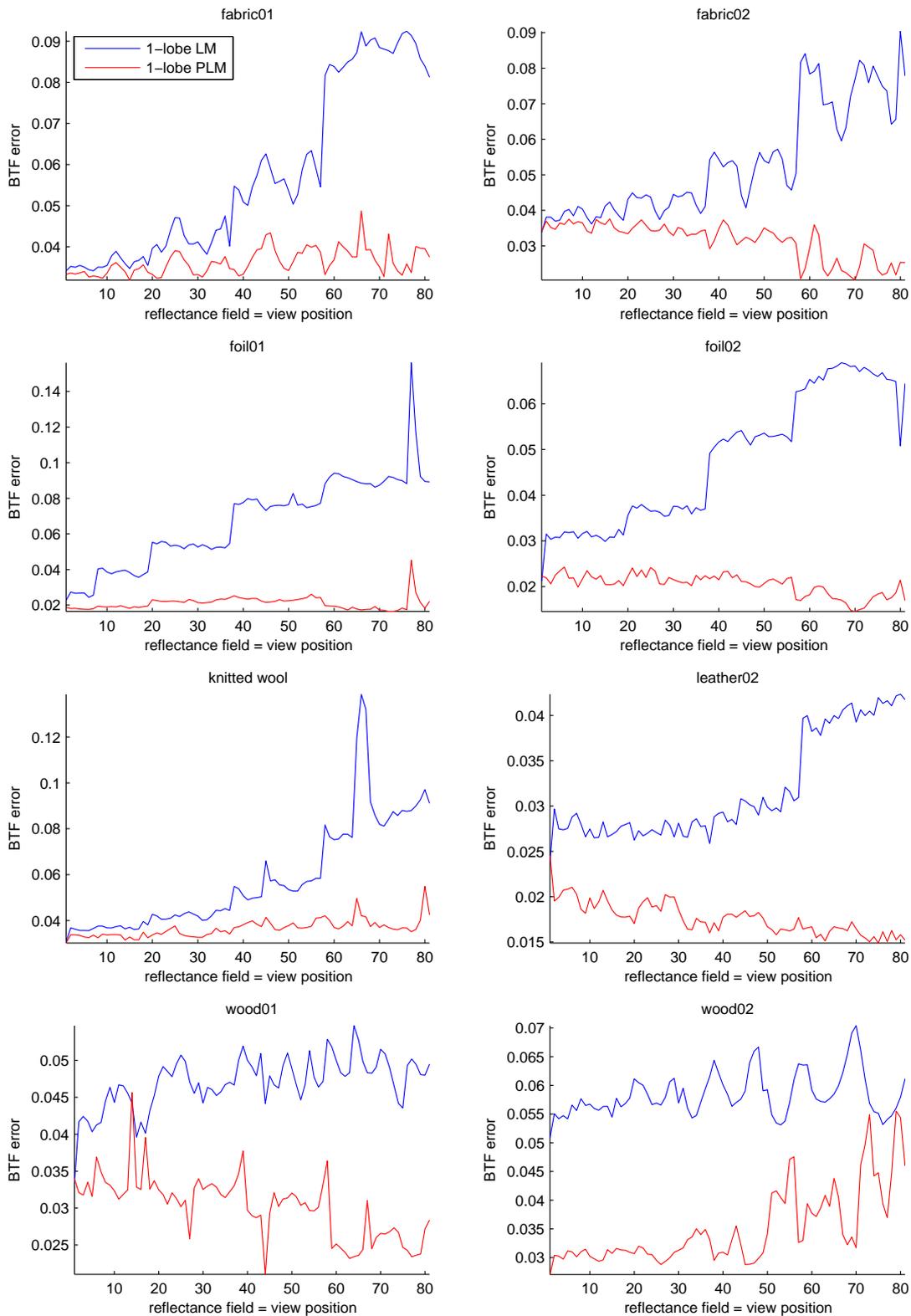


Figure 8.5: The mean average error (MAE) (according the equation (8.15)) of one-lobe lafortune model (LM – blue line) and its polynomial extension (PLM – red line) for all 81 reflectance fields of eight different BTFs: *fabric01*, *fabric02*, *foil01*, *foil02*, *wool*, *leather02*, *wood01* and *wood02* respectively.

illustrated in Fig. 2.3. Similarly, the overall MAE errors computed as average value of \mathcal{E}_v for all reflectance fields and spectral channels of individual BTFs are figured in Tab. 8.1. From the first two columns of the table it is apparent that the proposed polynomial extension of one-lobe Lafortune model (PLM) yields considerably lower MAE values in comparison with one-lobe model. At the same time, the storage space required by the proposed model is maximally 5% higher in comparison with one-lobe Lafortune model. This small increase is caused by storing of fifteen additional polynomial coefficients for every BTF image.

Table 8.1: The MAE (according to the equation 8.15) of the synthesised BTFs for one-lobe Lafortune model (LM), its polynomial extension (PLM) and clustered polynomial extension (PLM-C) for different materials.

<i>material</i>		Mean Average Error		
		<i>LM</i>	<i>PLM</i>	<i>PLM-C</i>
	wool	0.058	0.037	0.038
	proposte	0.054	0.052	-
	fabric01	0.058	0.036	0.038
	fabric02	0.053	0.032	0.033
	foil01	0.067	0.021	0.023
	foil02	0.048	0.020	0.023
	leather02	0.032	0.018	0.021
	wood01	0.047	0.030	0.031
	wood02	0.058	0.035	0.038

Fig. 8.6 shows the results for several BTF images from two different reflectance fields \mathcal{R}_v . The upper row represents the original BTF images, the middle row shows the corresponding synthesised images by means of one-lobe LM while the bottom row represents the results of the proposed PLM.

From these images it is apparent that the proposed method offers both better approximation of colour hues than one-lobe LM and increased contrast of the estimated BTF images. This is due to stretching the histograms which results in the increased contrast, i.e., the increased distance of individual colour levels. This makes possible to recognise, e.g., two neighbouring colour levels perceived as one colour hue in one-lobe LM. The next application example of the proposed model is given in Fig. 8.7. The figure shows how the averaged BRDF computed by the proposed polynomial extension (dash-dot line) follows the original average BRDF (solid line) for reflectance field $\mathcal{R}_v - \theta_v = 60^\circ, \phi_v = 54^\circ$ of the *knitted wool* material. From the figure it is apparent that the original BRDF data fitting based on the proposed method is much more accurate in comparison with the unsatisfactory results of the one-lobe Lafortune model.

8.2.4 Compression of Lafortune Parametric Images

Using the polynomial extension of Lafortune model described above we were able to achieve the maximal compression ratio of a real BTF data about $\frac{1}{20}$ depending on the resolution of parametric images. Unfortunately even this compression implies the necessity to store

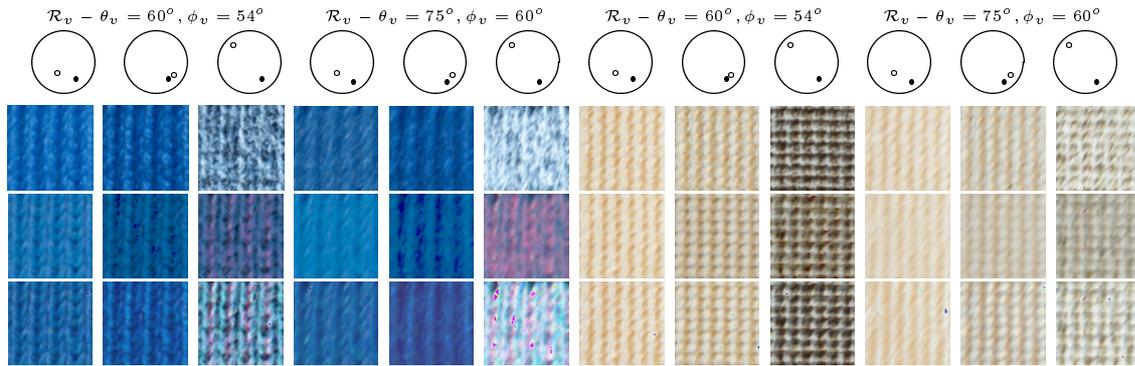


Figure 8.6: Synthesised BTF examples for *knitted wool* and *proposte* materials respectively. The first row describes mutual position of light (empty circle) and camera (filled circle) above the sample, the second row shows original raw BTF data. The third row shows results of one-lobe LM on registered BTF data and finally the fourth row illustrates results of proposed PLM.

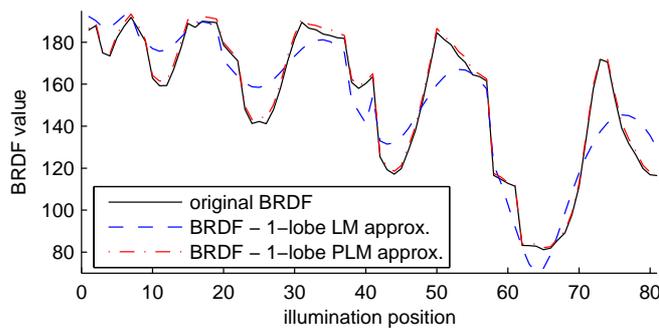


Figure 8.7: Comparison of the original BRDF (solid line), the BRDF approximated by one-lobe Lafortune model (LM) (blue, dashed line) and finally the BRDF approximated by proposed polynomial extension (PLM) (red, dash-dot line) for *knitted wool* material for $\mathcal{R}_v - \theta_v = 60^\circ$, $\phi_v = 54^\circ$.

several hundreds megabytes of data per one material. The data resolution to be stored depends mainly on the homogeneity of material, i.e., for materials with regular texture pattern, as for example fabrics, the required size of parameter images is rather small in comparison with the materials containing large, irregular patterns as for example leather, wood, etc.

Using further extension of the proposed BTF reflectance model we can even reduce the space required for the storage of model parameters while the computational costs increase only slightly and the visual quality is almost the same. The extension consists in the segmentation of the individual reflectance field image data to obtain the corresponding clusters in parametric images. Finally only a cluster index and model parameters corresponding to the individual clusters are saved for each RGB spectrum. The number of clusters for each spectrum is set to 256 to enable reproduction of 256 different grayscale levels. Thus the theoretical number of colour hues within this setup is $256^3 = 16777216$. Actually the number of clusters can be increased arbitrarily, while for BTF tiles segmen-

tation purposes the computationally reasonable number is less than 1000.

The whole LM parameter segmentation procedure works as follows. At the beginning the K-means segmentation algorithm (Alg. 3 in Section 6.1.4) is employed using pixels from all 81 images corresponding to actual reflectance field \mathcal{R}_v as data features. Note that the segmentation cannot be performed on the model parametric feature space containing directly model parameters as these individual parameters have strong non-linear impact on the restored pixel value and any general weights cannot be attached to them.

The K-means segmentation process is computationally very demanding and the segmentation of relatively small parametric images of resolution 256×256 for all 81 reflectance fields takes several hours. The segmentation can be significantly speed up for example by means of modified K-means algorithm described in [61]. This method exploits random sampling and enable time complexity linear in the size of the input.

During model synthesis in a particular pixel the corresponding cluster with the model parameters is obtained according to number at pixel coordinates in cluster index look-up file. These model parameters are found for each colour channel and the pixel value is figured out. Using this approach the storage size of model parameters reduces considerably since only one colour parametric look-up image and several cluster parameters have to be stored.

Choosing the Reflectance Field Subset

Unfortunately the size of pixel reflectance data features (3×81) is too big for segmentation of large parametric images which results to many hours of computation for each reflectance field. To avoid this exhaustive computational demands only a constrained set of images from those 81 in reflectance field are included into a data feature vector. To choose an appropriate subset of images bearing the most information of whole set we used an algorithm based on the Kullback-Leiber distance [60] of two different histogram distributions P, Q

$$d_{KL}(P, Q) = \sum_{i=1}^{256} P(i) \log \frac{P(i)}{Q(i)}. \quad (8.16)$$

However, the Kullback-Leiber divergence is non-symmetric and it is sensitive to histogram binning. For this reason we have used its modification called Jeffrey divergence [95] which is represented by the following equation

$$d_J(P, Q) = \sum_{i=1}^{256} \left(P(i) \log \frac{P(i)}{M(i)} + Q(i) \log \frac{Q(i)}{M(i)} \right) \quad (8.17)$$

where $M(i) = \frac{P(i)+Q(i)}{2}$. This divergence is numerically stable, symmetric and robust with respect to noise and the size of histograms bins in comparison with previous one.

The algorithm choosing a subset of the most different images (n_{max}) from reflectance field \mathcal{R}_v is described in Alg. 5. Basically a new image for the subset \mathcal{S} is taken as an image with the minimal distance d_J to all images already included in \mathcal{S} .

BTF Image Synthesis

When the segmentation is finished we obtain cluster indices $I_v(r_1, r_2, r_3)$ for the individual colour spectra r_3 of each reflectance field \mathcal{R}_v . Cluster indices are stored in form of colour

Algorithm 5: Choosing a subset \mathcal{S} of n_{max} images from the reflectance field \mathcal{R}_v

1. Compute normalised histograms $H_i, i = 1, \dots, 81$ for all 81 images.
2. Add first image corresponding to $(\theta_i = 0^\circ, \phi_i = 0^\circ)$ into \mathcal{S} .
3. FOR $n = 1 \rightarrow n_{max}$
 - (a) Compute minimal Jeffrey divergence for all images in \mathcal{S} to all 81 images in reflectance field \mathcal{R} and choose such j^* -th image histogram from \mathcal{R} for which the computed divergence is maximal:

$$j^* = \arg_j \left\{ \max_{i=1, \dots, n} \left[\min_{j=1, \dots, 81} d_J(H_i, H_j) \right] \right\}$$

- (b) Add j^* -th image into \mathcal{S} .
 4. From n_{max} images contained in \mathcal{S} are built up the data features for segmentation.
-

images of original parameter images resolution, i.e., in each colour channel we store the corresponding cluster index. An important product of segmentation is the table containing individual cluster centers $K_v(c)$ where c is the cluster index. For each cluster five LM parameters are stored for individual colour channel. The number of these clusters in our implementation was fixed to 256 for all colour channels.

The final synthesis is straightforward. The parameters ρ, D_X, D_Y, D_Z and n of the original model (8.19) are computed as

$$\begin{aligned} \rho(r)_v &= K_{v,1}(I_v(r)) \\ D(r)_{v,X} &= K_{v,2}(I_v(r)) \\ D(r)_{v,Y} &= K_{v,3}(I_v(r)) \\ D(r)_{v,Z} &= K_{v,4}(I_v(r)) \\ n(r)_v &= K_{v,5}(I_v(r)) . \end{aligned} \tag{8.18}$$

When the parameters are known the computation of the polynomial expansion of Lafortune model is the same according to the equation

$$\hat{Y}_{i,v}(r) = \sum_{j=0}^{n_p} a_{r3,v,i,j} [\rho_v(r) (\omega_i^T \mathbf{D}_v(\mathbf{r}))^{n_v(r)}]^j . \tag{8.19}$$

We refer to this clustered polynomial extension of the Lafortune Reflectance model as PLM-C in the following text. The synthesis based on the described approach is quite fast, requiring the look-up index tables only which can be implemented using standard OpenGL features. Besides, the storage size of LM parameters is reduced considerably as it

is figured out in the fifth column of Tab. 8.2. Thus by means of this method we are able to achieve the average BTF data compression ratio more than $\frac{1}{100}$, while the computational expenses are almost the same. Moreover, the graphical hardware can take advantage of this compact model size to reduce the amount of data loaded to GPU memory. This advantage may become more apparent primarily for VR scenes containing objects covered with many different materials which appear and disappear during scene observation. The average MAE for all reflectance fields of PLM-C remains almost on the same level as for non-clustered variant of PLM as it is apparent from the last column in Tab. 8.1.

8.2.5 BTF Data Interpolation

Chapter 4 describes the basic approaches concerning BTF rendering and interpolation. As it was mentioned the correct raw BTF interpolation involves at least nine BTF images which correspond to permutation of three different view and three different illumination positions. The proposed polynomial extension of Lafortune reflectance model have the same requirements since the polynomial fitting is performed according to angular resolution of the BTF data set. However, the nature of Lafortune model enables to compute the reflectance smoothly for arbitrary illumination position. To take advantage of this smooth property of the proposed model the mapping functions computed for given illumination and view angles have to be properly interpolated with respect to actual illumination position. For the sake of this interpolation we utilised barycentric coordinates as interpolation weights between the three closest illumination positions. The barycentric weights are obtained proportionally to triangle areas using the method described in Chapter 4. Additionally, interpolation of the corresponding mapping functions enables us to use only three different BTF images (corresponding to the three closest view positions) for final interpolation so the whole BTF rendering speeds up three times. In Fig. 8.8 is shown an example of rendering of *wood02* material using all nine images (left) in contrast with the three images interpolation only (right). There is an obvious difference between these two images especially for high elevation illumination angles (i.e., $t_i > 75^\circ$) where the texture becomes darker due to incorrect interpolation. The reason for these problems is interpolation of those angles which do not correspond to the actual illumination position. Thus the BTF extrapolation for higher elevation illumination angles would be more appropriate approach. An overview of results of view angles interpolation for all tested materials compared to original BTF data is given in Fig. B.1.

8.3 Unrestricted Resolution BTF Reflectance Model

The size of the raw BTF measurement is always constrained by the size of material sample and by the resolution of the measurement device. However, to cover large objects in VR the method for BTF enlargement is necessary. There are several methods on this topic available mainly in computer graphics. One of them is based on intelligent sampling technique [70], which produces synthesised BTF images according to specific samples from sparse set of original BTF measurements in combination with synthetic image obtained by means of a range-map. The second approach combines texture-synthesis using mathematical models based either on Gaussian-Markov random fields [36] or on multi-dimensional causal auto-regressive model [38] of the original BTF image with estimated range-map. The most often approaches to BTF enlargement are based on different extensions of intelligent



Figure 8.8: BTF data interpolation using the proposed PML model of *wood02* material. Full 9-images interpolation of both view and illumination position (left) and 3-images interpolation of view position only (right).

sampling techniques [24, 25, 11, 62, 85]. These methods produce set of image/parametric tiles and the resulted image is obtained as a combination of these tiles. In the context of this thesis two approaches are suggested:

- **Image tiling** based on generation of set of LM parametric tiles. The large image is a simple combination of these tiles fulfilling certain constraints. This method cannot produce colour values not present in the original image and cannot reproduce texture elements with lower spatial frequencies than those corresponding to the tile resolution.
- **Image modelling** based on Markov random field (MRF) models. These models may generate texture image of arbitrary size while only few parameters have to be stored. On the other hand a quality of these models can be compromised in comparison with image tiling mainly for materials with regular pattern which cannot be easily reproduced by means of these stochastic-based models.

Both these methods can be extended to enable the enlargement of reflectance model parameters instead of ordinary colour image as it is explained in Sections 8.3.1 and 8.3.2.

8.3.1 Reflectance Model Parametric-Plane Tiling

Some form of tiling the model parameter planes is inevitable when a large object is to be covered by BTF. A simple seamless one-tile repetition does not provide satisfactory results mainly for non-regular textures where certain pattern occurs on the same position subsequently. Thus to obtain more realistic results we need more than one tile per texture to support the visual variability of generated images. There is a variety of image-based texture synthesis methods published recently [24, 11]. In this thesis we use the image tiling method based on the image stitching introduced in [103]. The idea of stitching is based on the minimum error boundary cut, as used in the image quilting algorithm [24]. The stitching procedure is demonstrated in Fig. 8.9.

Let us assume that each oriented stitch is created between two equally sized overlapping image regions: a source and a target. Creating such stitch can be imagined as attaching a cropped part of source to target (Fig. 8.9). To make the transition between two images as invisible as possible the source (i.e., the tile surrounding) is cropped from along the

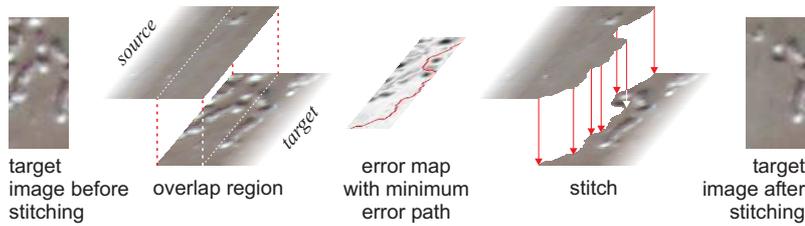


Figure 8.9: Image stitching. The source image is cropped from the right along the minimum error path and placed over the target background image.

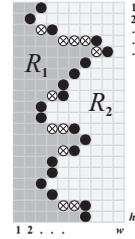


Figure 8.10: Simplified path representation model. Each row contains one control point (black dot) and complementary points (crossed dots).

minimum error path before attaching to target (i.e., the tile content). The minimum error path is constructed to lead through the error map (8.20) which represents the visual difference between source and target for each pixel of the overlapping region $w \times h$.

$$E[i, j] = d(I_S[i, j], I_T[i, j]), \quad i = 1, \dots, w \quad j = 1, \dots, h \quad (8.20)$$

where $d(., .)$ is the Euclidian distance of two pixel colour values. Each error path is represented unambiguously by a sequence of control points \mathbf{c} , one for each row:

$$\mathbf{c} = \langle c_1, c_2, \dots, c_h \rangle, \quad c_j \in 1, \dots, w \quad (8.21)$$

To obtain a continuous sequence the additional points have to be added to actual path as is illustrated in Fig. 8.10. From several possible complete path definitions the one that suits the oriented-stitch approach is adopted:

$$Path^{\mathbf{c}} = \{(i, j) : j = 1, \dots, h; i = \mu_j, \dots, c_j\} \quad (8.22)$$

where $\mu_j = \min(c_{j-1} + 1, c_{j+1} + 1, c_j)$. For each path a criterion can be evaluated to assess the expected visible transition inconsistency:

$$\epsilon(Path^{\mathbf{c}}) = \sum_{(i,j) \in Path^{\mathbf{c}}} E[i, j] \quad (8.23)$$

In [103] is defined a sub-optimal minimum path algorithm on error map E of size $w \times h$ - the *oscillating search*. The algorithm first evaluates single control point shifts in each step as long as the criterion value can be decreased. This algorithm is used as a fast alternative to the slow optimal path search procedures like the dynamical programming. The main advantage is the computational speed. The *oscillating search* has polynomial complexity while the optimal search is always exponential. This method is a step-wise procedure that sequentially improves some actual solution and thus it can be stopped at any moment to yield a usable result. Differences between the optimal and suboptimal search occur in the areas of evenly distributed error and thus they remain visually indistinguishable. However, it is very important to assure that the overlap image region itself is positioned and sized not to rule out the existence of low error path.

The minimum path based stitching often produces good natural appearance of image transition areas. However, if no good path exists in the error map, visible artifacts can not

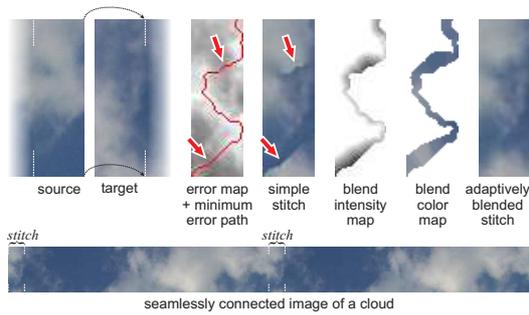


Figure 8.11: Adaptive blending to improve visual consistency of stitched image areas.

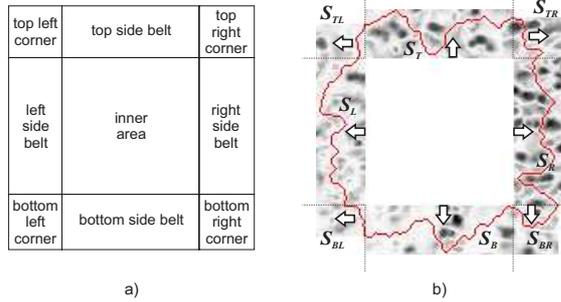


Figure 8.12: Patch creation. the stitching technique is used to create sides and corners of the patch.

be avoided (as demonstrated in simple stitch in Fig. 8.11). Therefore the authors defined the *adaptive boundary blending* (Fig. 8.11) to reduce the visibility of such unwanted high-error artifacts. The idea is to interpolate between the overlapped source region to the target with a locally adjusted intensity while utilising the minimum error path. The adaptive blending process can be visualised using the blend intensity and blend colour-maps separately. The boundary can be made almost unnoticeable in this way, except of cases when the transition is made between principally incompatible texture image areas.

The previously described image stitching method can be extended to transfer general continuous image regions while keeping the transition between the old and new unnoticeable. During tile generation the algorithm searches for such a rectangular region in the source texture image, where the opposite border areas are the most visually consistent in both the horizontal and vertical directions. As a criterion of visual consistency it is used RGB Euclidean distance. New tiles can be obtained using the described patching technique with respect to Fig. 8.12. New tiles can be created by making a copy of the template tile and subsequently covering its inner area by patches taken from different positions in the source texture image. Example image synthesis of the described method is shown in Fig. 8.13.

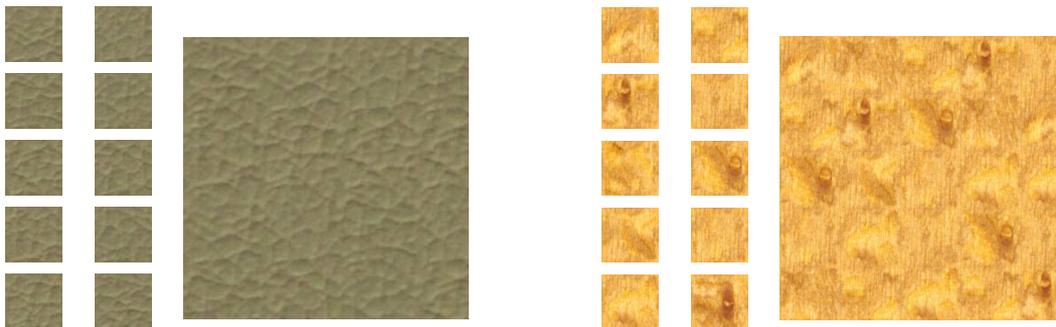


Figure 8.13: Example of tiling using the described method. Ten original tiles were used to produce synthesised image of *leather02* and *wood02* materials.

However, BTF tiling is much more complex task as the stitch should appear consistent in all BTF planes. To decrease the computational complexity of such an extensive data

processing we adopted a two-stage tiling process. In the first stage we only determine the stitching and other parameters to be used later for actual tile creation. For this purpose only a few sample parametric images are taken (prepared in full size, i.e., 800×800 pixels) to represent different azimuthal and elevation view positions. The optimal stitching paths are found in this subset of parametric images by means of the method described in [103]. In the second stage the complete raw BTF data are processed using the pre-computed stitching parameters. Once all tiles become available, the final parameters of the proposed Lafortune model are computed based on the BTF tiles. This procedure saves considerable computational demands of Lafortune parameters estimation algorithm. For the sake of the synthesised BTF rendering quality the number of computed tiles used is usually more than one. In this thesis the maximal number of computed tiles is ten what we found to be a reasonable compromise between the computational complexity and quality of the output.

The time complexity can be described as follows. Preparation (estimation) of sample Lafortune parametric planes in original BTF resolution to be used in the first tiling stage takes about 1 hour. The first tile creation stage (stitching parameters learning) takes usually less than 1 hour. The second stage, i.e., cutting the complete raw BTF data to obtain raw data tiles based on the first-stage-parameters takes 3-5 hours, mainly due to lots of slow data access operations involving thousands of files. The final non-linear estimation of one-lobe Lafortune model parameters for ten different BTF image tiles of size 64×64 for all 81 reflectance fields comprised in the BTF database takes about 10 hours. Note that if more reflectance lobes were used the time required for parameter fitting would considerably increase and thus the storage space for these parameters would be much more larger.

The time complexity mentioned above strongly depends on the BTF image size and computational power involved. We used the BTF dataset from Bonn University of size 800×800 pixels and computer PC Athlon 1.9GHz.

Results

Fig. 8.14 shows the error curves (MAE) for individual test materials. For each material the MAE is computed for all 81 view positions \mathcal{R}_v (depicted on x axis) of clustered one-lobe Lafortune model (LM-C, blue solid line) and its clustered polynomial extension (PLM-C, red solid line) are compared with the corresponding non-clustered variants of LM and PLM (both depicted as dash-dot line). The overall MAE values of all tested materials were computed as averaged MAE of all reflectance fields and are shown in Tab. 8.1 in contrast to the corresponding values of non-clustered PLM. The MAE for PLM-C is slightly higher in comparison with PLM but this higher error is well counterbalanced by the model size. The number of parameters to be stored have been reduced using proposed parameter clustering at least ten times in contrast to the non-clustered PLM as it is evident from the last column of Tab. 8.2. The tile resolutions for individual materials are listed in this table as well. Generally, the less homogeneous is the material appearance the larger image tiles are required for proper reproduction of the material characteristics.

Finally for subjective visual comparison of results we used the same 3D object as previously. This car armrest is covered by different BTFs generated using the proposed clustered PLM-C and it is compared with the non-clustered PLM as well as with the raw tiled BTF as it is illustrated in Fig. 8.15. The first column shows the object covered by

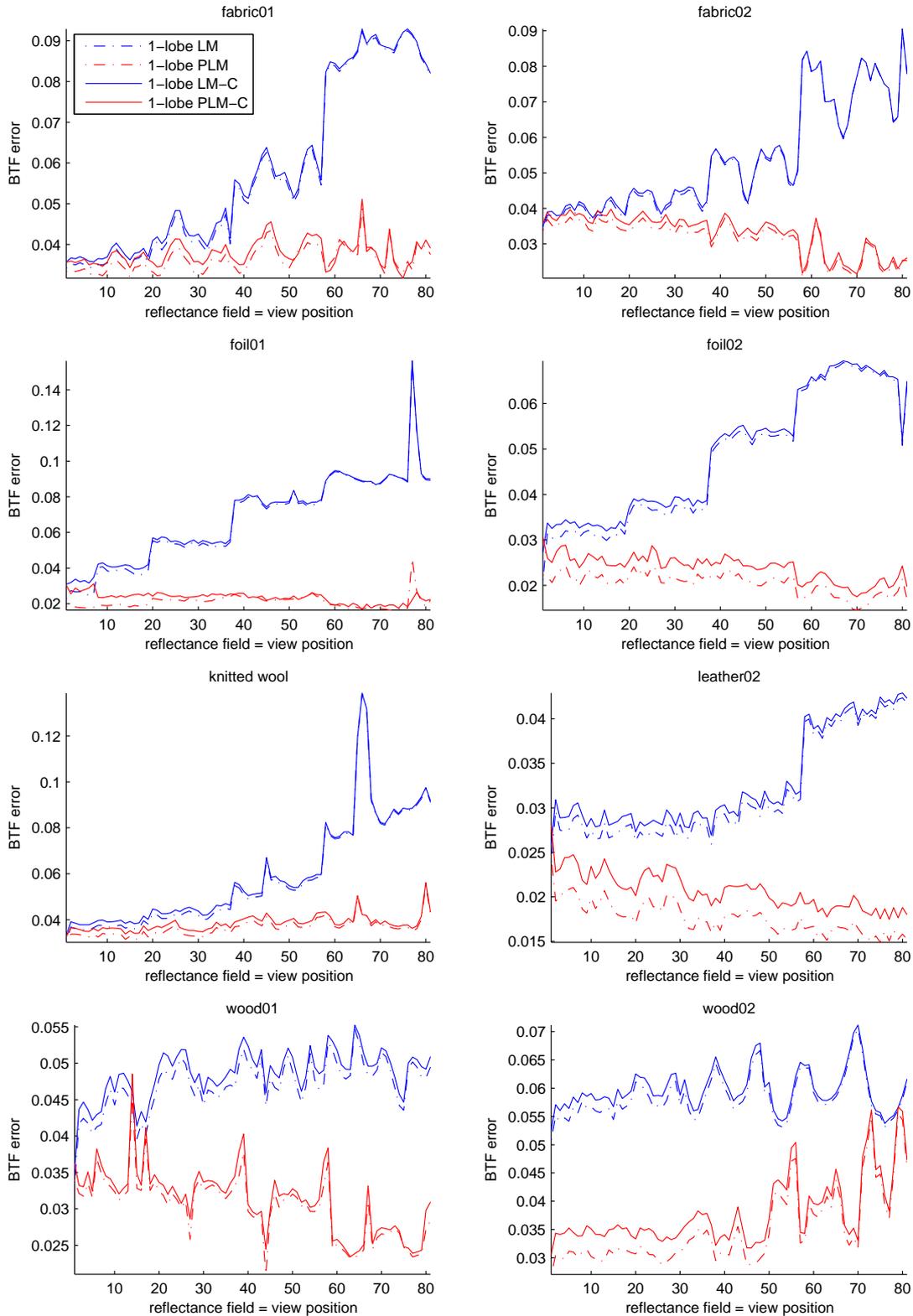


Figure 8.14: The mean average error (MAE) (according the equation (8.15)) of clustered one-lobe lafortune model (LM-C – blue line) and its clustered polynomial extension (PLM-C – red line) compared with non-clustered variants LM and PLM (dash-dot lines) for all 81 reflectance fields of eight different BTFs: *fabric01*, *fabric02*, *foil01*, *foil02*, *wool*, *leather02*, *wood01* and *wood02* respectively.

Table 8.2: Storage size of the proposed PLM and PLM-C in comparison with size of the raw BTF data and their tiled representation.

		storage size in MB				
<i>material</i>	<i>raw BTF</i>	<i>10 BTF tiles</i>	<i>PLM</i>	<i>PLM-C</i>	<i>tile size [pixels]</i>	
 wool	733.3	103.4	33.5	4.3	25 × 25	
 fabric01	6766	87.1	24.9	2.9	21 × 23	
 fabric02	5863	77.5	24.1	4.0	19 × 23	
 foil01	5190	728.1	406.8	19.2	86 × 96	
 foil02	5065	527.5	296.7	13.8	74 × 79	
 leather02	5074	659.7	381.0	18.6	86 × 87	
 wood01	5330	1333.2	771.8	31.8	122 × 125	
 wood02	5083	2405.0	973.4	29.1	137 × 142	

the raw BTF tiles, the second column shows results of PLM while the last column shows the results of PLM-C. From these images it is apparent that the visual difference between PLM-C and PLM results is almost indiscernible.

To sum up this section, we recall that the compression ratio of PLM-C ($\frac{1}{100}$) is approximately ten times higher and computational expenses are much the same.

Parametric BTF Tiles Rendering on 3D Object

Till this point the performance of the proposed method was expressed mainly in terms of mean average error. In this section the synthesised BTF data are mapped on 3D triangulated object to enable subjective visual comparison of the obtained results. In the scope of this thesis we have used 3D model of Mercedes Class-C interior by courtesy of Daimler-Chrysler and Bonn University. The car armrest in Fig. 8.15 is covered by eight tested BTFs approximated by means of the proposed model. The first column represents armrest covered by the original tiled BTF measurements, while the second column represents BTF data approximated using the one-lobe LM and finally the third column depicts the armrest covered by the BTF data obtained using the proposed polynomial extension of one-lobe Lafortune model (PLM). According to graphs in Fig. 8.5 the images in the second column were dim and less contrast losing information in dark parts as it is apparent for example for *foil01* material in comparison with the original BTF data in the first column. On the other hand, observation of the third column show clear improvement of the previous drawbacks and the images are more or less visually indiscernible from the original in the first column.

Fig. 8.17 shows part of car gearbox covered with four BTFs approximated by means of all proposed reflectance models. The first row shows original BTF tiling, the second row shows result of one-lobe LM, the third row shows results of proposed PLM, the fourth row shows clustered variant PLM-C and the last row shows PLM-C result for view interpolation only. The result of PLM-C variant on car gearbox for different illumination angles is shown in Fig. 1.1. Fig. 8.4 depicts two different examples of car interior covered with seven BTFs approximated again using PLM-C. The results of proposed PLM-C model are compared

with the results of probabilistic 2D CAR BTF model in Figs. 9.2 and 9.3.

Note that in the presented examples it is not used any shading method and the visual appearance is produced by a combination of the BTF images, corresponding to the light and camera position, on each object polygon.

8.3.2 Modelling of the Reflectance Model Parameters

An alternative solution to parameter tiling is their synthesis by means of probabilistic image synthesis algorithm. Chapter 7 describes the following three different Markov random field based probabilistic texture models: 3D Causal Auto-Regressive model, 2D Causal Auto-Regressive model and Gaussian-Markov Random Field model. These stochastic models enable synthesis preserving important statistical properties of the original texture image.

The estimated Lafortune parametric images can be considered as texture images and thus they can be synthesised by means of these stochastic models. The main motivation of this parametric images modelling is a huge compression ratio of the probabilistic models and parametric image synthesis of unrestricted size.

There are five Lafortune parametric images corresponding to parameters ρ, D_x, D_y, D_z, n for each colour channel resulting in fifteen parametric planes in total. For the correct synthesis all these planes have to be synthesised simultaneously to guarantee the correspondence of the individual structural features. We synthesised the parametric images of one-lobe Lafortune model for two different materials *foil02* and *wood02* by means of all three probabilistic models introduced in Chapter 7. The obtained examples of BTF images computed from the synthesised parametric images (corresponding to the individual probabilistic texture models GMRF, 3DCAR, 2DCAR respectively) are shown in Fig. 8.16.

Although this approach seems to be a promising way of compression and modelling of the Lafortune parametric images there are several practical problems which are not solved yet. One of them is the synthesis time so the only way to use this modelling approach is using hardware implementation. The next problem is preserving of the structure elements correspondence in parametric images synthesised in the individual synthesis passes for different view positions. Unlike the reflectance BTF model the probabilistic models are driven by a Gaussian white noise generator so that the individual synthesised images can differ considerably. The most straightforward solution can be using the same noise values for each \mathcal{R}_v 's parameter synthesis. Moreover, the probabilistic models cannot preserve regular structure of the original pattern sufficiently so they are suitable mainly for materials with irregular structure as , e.g., wood, leather, etc. as it is shown in Fig. 8.16.

8.4 Summary of the Reflectance Models for BTF Modelling

Application of the reflectance models to BTF modelling is one of the most common approaches which can yield very realistic results. One can compare results of the proposed PLM model with the original BTF tiles as applied on a part of car gearbox in Fig. 8.17. The figure compares original BTF measurement mapped on car gearbox with results of all reflectance models discussed in this chapter. These kind of BTF models can take advantage of steadily increasing power of graphics hardware enabling to perform relatively complex processing of individual vertices and fragments of the textured object surface. So the pixel-wise reflectance models can be easily implemented in such a way, however their



Figure 8.15: Part of a car armrest covered using BTF when only view angle interpolation is used. The figure includes the tiled original BTF data (first column), the results of one-lobe LM (second column), the result of the proposed one-lobe PLM (third column) and results of the proposed one-lobe PLM-C (fourth column) for eight different materials: *fabric01*, *fabric02*, *foil01*, *foil02*, *knitted wool*, *leather02*, *wood01*, *wood02*.

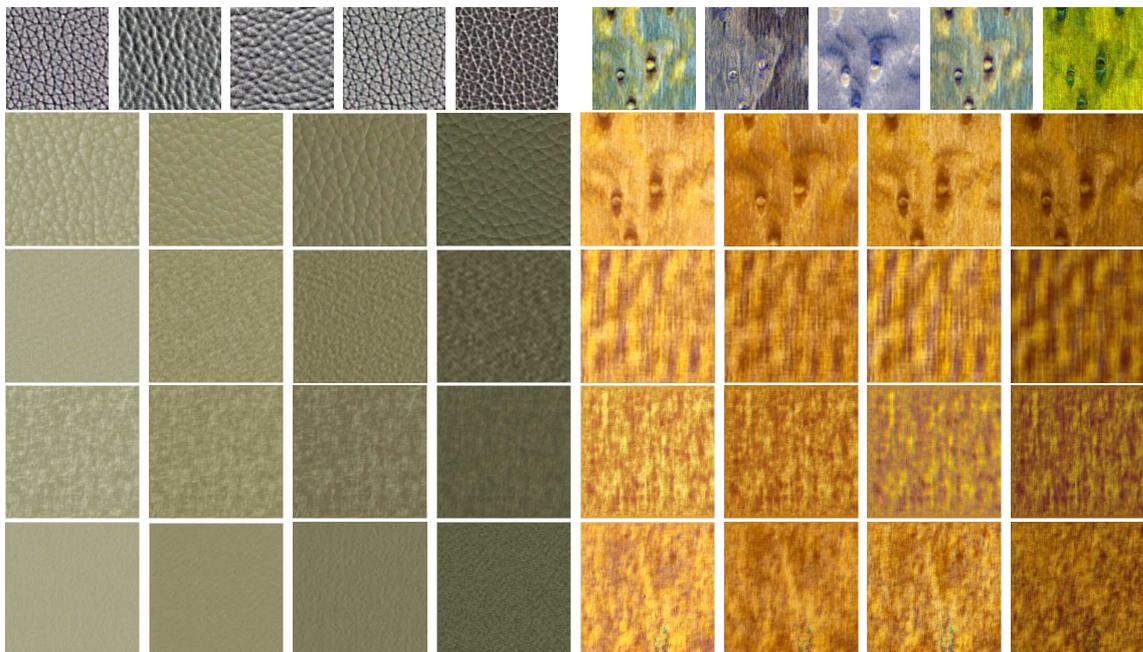


Figure 8.16: BTF images from the Lafortune parametric images obtained by probabilistic synthesis. The figure shows the Lafortune parametric images estimated from the original BTF images (first row), the original BTF images (second row), the BTF images obtained from the Lafortune parametric images synthesised using GMRF model (third row), 2DCAR model (fourth row) and 3DCAR model (fifth row), respectively, for materials: *foil02* and *wood02*.

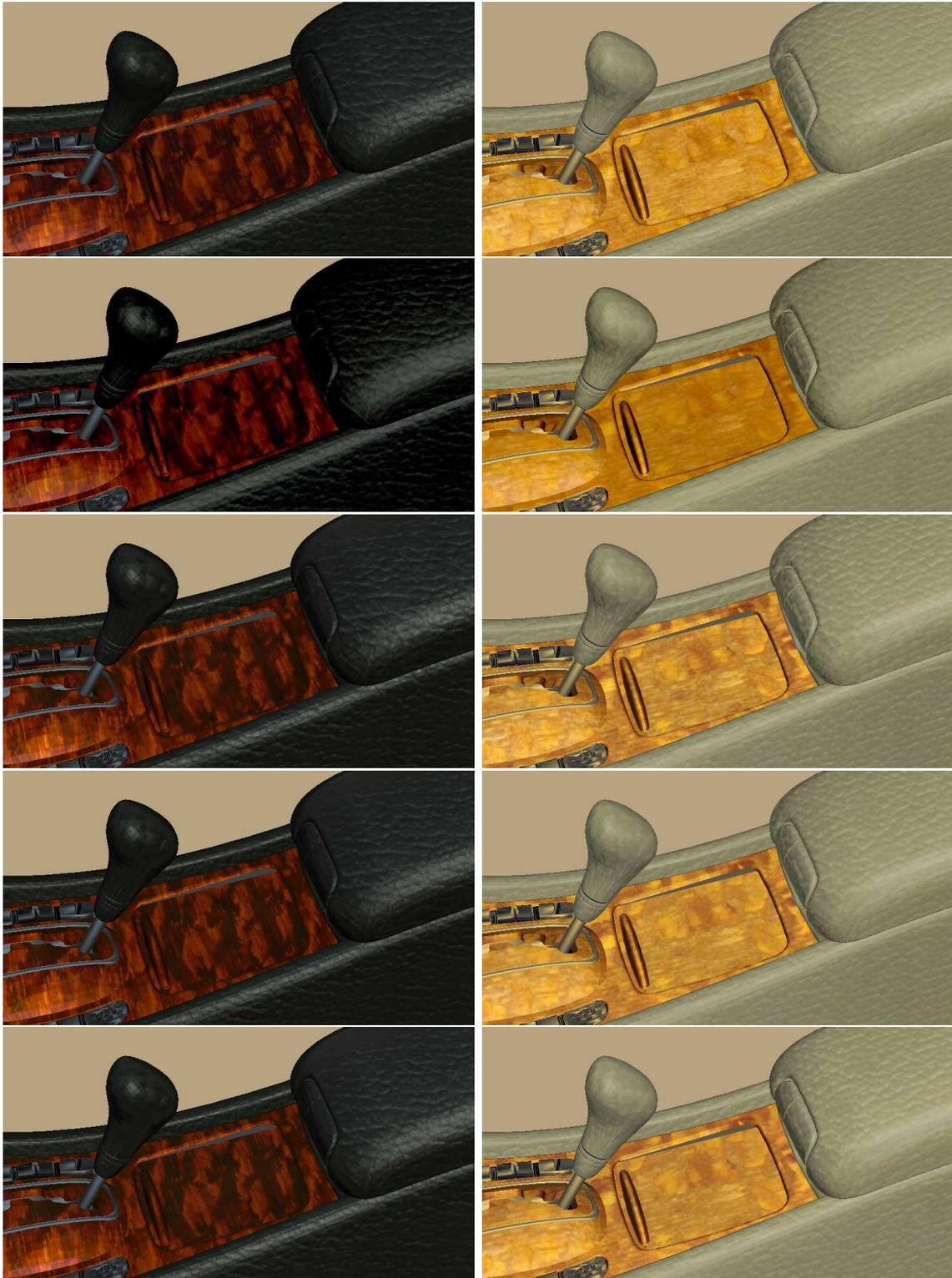


Figure 8.17: A part of Mercedes C-class gearbox covered using four BTFs: *foil01*, *wood01* and *foil02*, *wood02*. The first row illustrates using of original tiled BTF data, the second row depicts approximation using one-lobe LM, the third row is result of proposed one-lobe PLM model, the fourth row shows result of proposed clustered PLM-C model and the last row shows result of fast PLM-C with view angles pixel-wise interpolation only.

performance is tightly related to their complexity and kind of mathematical functions involved. Nowadays, the pixel-wise BTF reflectance models can be rendered in real-time frame-rates but an additional problems occur when the rendering of complex VR scenes is required containing large number of distinct materials. This is quite common requirement for example in computer aided design or safety simulation systems where preservation of a real material appearance is essential. Fig. 8.4 depicts two distinct examples of car interior covered by seven different BTFs. The total memory requirements for the storage of all these materials are 119MB so they can be efficiently stored in GPU for fast BTF rendering of simpler VR scenes. However, the rendering of more complex VR scenes containing tens of different materials becomes time consuming due to two main reasons. The first is non-zero time of the pixel-wise reflectance model computation with necessary BTF interpolation and the second lies in loading-time of the reflectance model parameters into graphics hardware memory. Thus the number of model parameters and their representation is a crucial problem of all pixel-wise BTF reflectance models. However, a reduction of model parameter number leads to compromised visual quality of obtained results. For instance the BTF data compression ratio of the most advanced BTF models is about $\frac{1}{100}$ with the resulting size of 10MB in average per material. The following main drawback of the reflectance models is their inability to synthesise BTF image of arbitrary size without additional image synthesis methods as for instance image tiling presented in this thesis. So the resolution of the synthesised BTF images is limited by the resolution of the original BTF measurements.

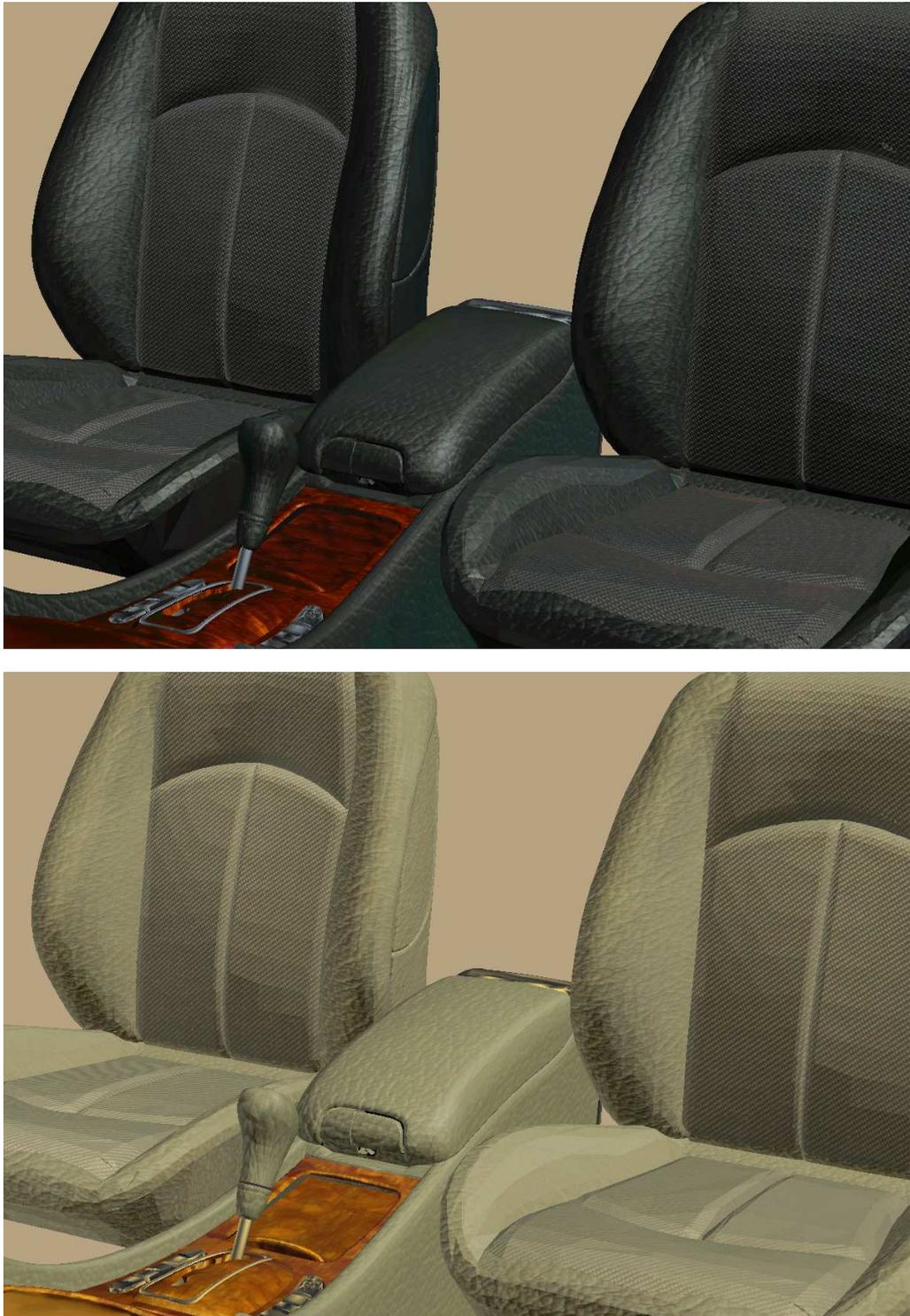


Figure 8.18: Two partial examples of car interior modelling (Mercedes-Class C). Images illustrate seven different materials: *wood01*, *wood02*, *foil01*, *foil02*, *fabric01*, *fabric02*, *leather02* approximated by means of proposed PLM-C BTF model.

Chapter 9

Results Verification and Testing

9.1 Probabilistic BTF Models Verification

Probabilistic BTF model results verification is a difficult problem. All probabilistic BTF models presented in this thesis are based on some kind of MRF model - GMRF, 2DCAR, 3DCAR. However, these MRF models are stochastic models which do not produce exact copy of an original texture but its accurate approximation. The original texture is substituted by the analytically synthesised texture preserving the major statistical properties of the original. The quality of this representation depends on chosen model type and its initial parameters as is the support set shape and size, direction of movement on image grid, etc.. For this reason any differential metric based on pixel-wise image comparison between original and estimated texture image does not make sense. Unfortunately, no robust method is available yet for computation of this similarity. Thus the only known way is to compare the overall visual similarity of two textures by an independent viewer. There were several attempts of texture similarity metrics definition in the past as, e.g., the work of Julesz [46] which suggested similarity measure based on the second order statistical moments. However, this promising method was questioned later by the same author in [48, 47] since many counter-examples of proposed similarity measure failure have been shown. Another method based on the same assumption but using third order statistics was introduced in [125]. Although, this method seems to be robust, it can only decide whether two texture images are identical or not, thus the method does not provide any similarity measure. So it is clear that till today there is no algorithm available providing acceptable texture similarity measure. The only possible way of comparing two texture images is based on subjective visual observation. To provide reliable similarity results by means of visual observation we build up a relatively large group of voting observers. However, this is beyond the scope of this thesis. Moreover, these tests should be performed for all processed BTF materials comprehending hundreds of distinct synthesised results. Thus we compared all results in this thesis according to subjective visual observation of a small group of the department colleagues. Despite the dependence on personal observer's preferences this method provides satisfactory results as can be seen on examples in Fig. 9.2 and Fig. 9.3.

In the case of BTF synthesis all synthesised subspace images had to be compared with the corresponding set of original cluster representatives, i.e. 15-25 couples of colour images. The ranking priority was set to emphasise preservation of colour hues as well as the mesostructure of the material.

9.2 Reflectance BTF Models Verification

Unlike MRF BTF models the verification of the proposed reflectance BTF models is much easier, since the overall structure of the material together with its original localisation is preserved. Therefore we can employ differential measure between individual pixels of original and approximated texture image. We used a mean average error (MAE - see (8.14)) computed for individual BTF images. Fig. 9.1 illustrates the MAE course for *knitted wool* reflectance field $\mathcal{R}_v - \theta_v = 60^\circ, \phi_v = 54^\circ$. Each point on the curves represents one illumination position for fixed view position. The MAE is computed in each point with respect to (8.14) between original BTF image and its approximation using the LM and PLM.

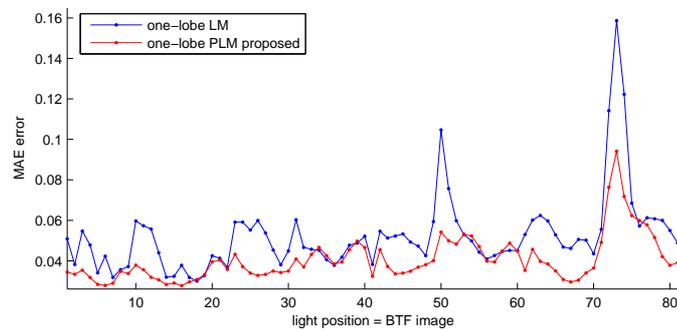


Figure 9.1: Mean Average Error for *knitted wool* reflectance field $\mathcal{R}_v - \theta_v = 60^\circ, \phi_v = 54^\circ$.

The final BTF data verification is performed when original BTF data and their approximation by proposed BTF model are mapped onto the same test 3D object and visually compared for the same view and light positions as is depicted in Fig. 9.2, Fig. 9.3 and Fig. 1.1.

During our experiments it became clear that no ideal BTF model exists. MRF probabilistic BTF models yield utmost data compression while producing compromised visual quality results for some materials. Whereas the models based on reflectance models or BTF factorisation have excellent visual quality almost indiscernible from originals, their memory requirements are still too high for complex scene rendering. Moreover, in contrast to proposed MRF models the reflectance models can only reproduce original BTF, thus one has to deal with BTF enlargement by means of some additional modeling method.

For these reasons the type of optimal BTF model should be chosen carefully according to application purpose depending on bandwidth of used graphics hardware, speed of BTF rendering, quality requirements, data compression, etc.. For example, in game industry's VR systems there is no need to have so accurate material reflectance approximation as in VR systems aimed to car or architectural interior design or for safety simulation in automotive or aerospace industry. Moreover, the BTF rendering speed is a crucial factor for real-time simulations, where different BTF models than are suitable for instance for high quality rendering applications and so on.



Figure 9.2: Comparison of measured BTF with two proposed BTF models. First image shows tiled original BTF measurements of *foil01* and *wood01* materials mapped on part of car gearbox. Following images show results of proposed polynomial extension of Lafortune reflectance model (PLM) (second) and probabilistic BTF model based on 2D CAR subspace MRF modelling (third).

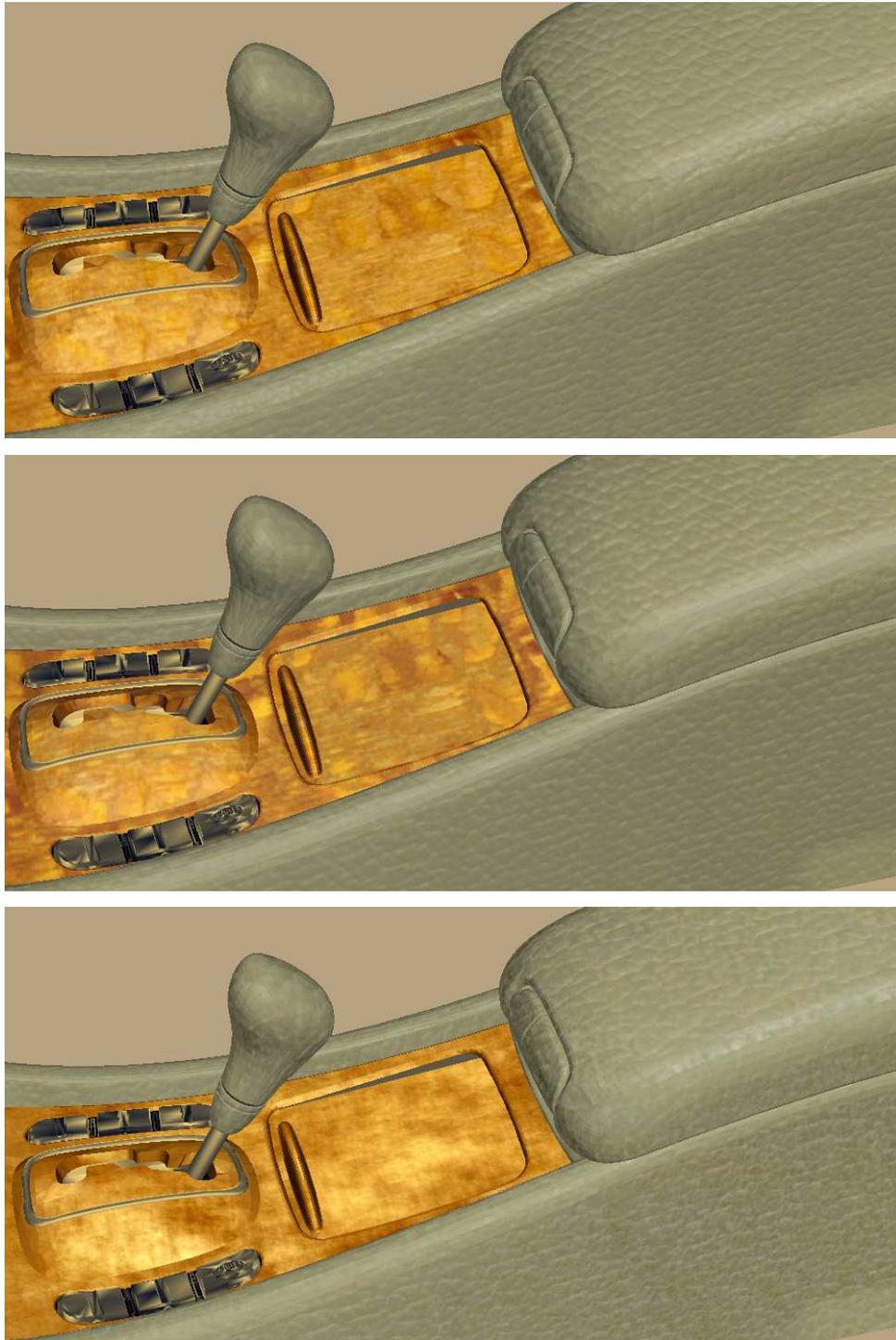


Figure 9.3: Comparison of measured BTF with two proposed BTF models. First image shows tiled original BTF measurements of *foil02* and *wood02* materials mapped on part of car gearbox. Following images show results of proposed polynomial extension of Lafortune reflectance model (PLM) (second) and probabilistic BTF model based on 2D CAR subspace MRF modelling (third).

Chapter 10

Conclusions and Future Work

The main objective of this thesis was development of novel *Bidirectional Texture Function* (BTF) models enabling high BTF data compression and their fast rendering being suitable for direct hardware implementation, while the major visual characteristics of approximated BTF are preserved.

10.1 Contributions of the Thesis

In this thesis we present an overview of BTF compression and modelling methods published up to now and propose two novel BTF modelling approaches with several corresponding BTF models:

- Probabilistic BTF models based on a set of the following underlying texture models:
 - 2D causal autoregressive model
 - 3D causal autoregressive model
 - Gaussian-Markov random field model
- Polynomial extension of pixel-wise reflectance BTF model

During development of these BTF models a variety of image processing, statistical and computer graphics methods were employed to develop two different BTF data processing pipelines consisting of BTF analysis, compression and modelling, synthesis and visualisation of synthesised BTF results on 3D objects.

We should emphasise that no ideal BTF model can be claimed the best. Each BTF model has its advantages and disadvantages and is tailored for different application and / or kind of approximated material. In the following text we shortly describe and compare the both BTF modelling approaches proposed in this thesis, discuss their pros and cons and proper application areas.

In the first proposed BTF modelling approach we actually published the first generative BTF models. This approach is based on statistical analysis of BTF subspace images by means of several Markov random field multi-scale models. BTF subspace images are obtained using BTF segmentation. Based on to these MRF parameters the novel subspace images are synthesised with corresponding spectral and spatial information. These subspace images are finally interpolated with respect to actual view and illumination direction and for rough materials also combined with surface height information by means

of bump/displacement mapping filter. Surface height information is enlarged using image quilting method and the underlying MRF model enables synthesis of subspace images in arbitrary resolution so the resolution of synthetic BTF is actually limited only by hardware limits. Due to the model's inherited stochastic character these methods can not exactly reproduce BTF spatial patterns locations as it is common for reflectance models. For regular materials the regularity has to be introduced into the model by means of surface height information. Our test results prove extreme BTF data compression ratio of the proposed BTF model simultaneously with very realistic visual quality. Some synthetic BTF textures reproduce given measured images so that both the natural and synthetic textures are visually almost indiscernible and even the worst results can be used for preliminary BTF texturing applications at the least. These models enable huge BTF compression ratio unattainable by any other BTF compression method ($\sim \frac{1}{10^5}$). Similarly to the second proposed reflectance BTF model, this kind of models enables also very fast BTF synthesis and rendering implemented in graphics hardware.

Advantages:

- extreme compression ratio $\sim \frac{1}{10^5}$,
- synthesis of BTF of arbitrary size,
- fast BTF synthesis implementable in GPU (2D CAR model),
- mip mapping for free due to a multi-scale nature of the model,
- possibility of BTF approximation from spare set of BTF images.

Disadvantages:

- compromised visual quality for highly non-Lambertian or translucent materials,

Application field:

- computer games industry or other VR applications running on low-end hardware offering satisfactory BTF approximation capturing its most apparent visual features.

The second proposed BTF modelling approach is based on polynomial extension of pixel-wise Lafortune reflectance model computed for individual spectral channel of every pixel. The advantage of this model consists in using only one reflectance lobe while the remaining fitting is done by means of polynomial extension of one-lobe Lafortune model. Using of one-lobe model considerably reduces the number of model parameters which have to be stored. Moreover, the memory requirements of introduced polynomial coefficients are negligible in comparison to Lafortune parameters. Proposed reflectance model has similar computational requirements as pixel-wise Lafortune model while using only few additional linear operations so it can be easily implemented in graphics hardware. To increase the model's BTF compression ratio even more we introduce a clustering variant which enables ratios $\sim \frac{1}{2 \cdot 10^2}$ whereas the computational requirements remain similar. Due to the fact that the original Lafortune model itself can not enlarge BTF to arbitrary size our generalisation applies a simple image quilting of Lafortune parametric images. The results of this model show its excellent performance for all eight tested BTFs even for materials with complicated underlying structure producing strong subsurface scattering effects, e.g., in the case of two kinds of lacquered woods.

Advantages:

- excellent visual quality for all tested materials,
- fast BTF synthesis implementable in GPU,
- moderate compression ratio $\sim \frac{1}{2 \cdot 10^2}$.

Disadvantages:

- includes additional method for enlargement of synthesised BTF images,
- time consuming offline parameters estimation and clustering.

Application field:

- professional VR systems (CAD) with high requirements on visual quality and accuracy.

An overall visual comparison of both proposed modelling methods to original BTF is given in form of a VR scene showing a part of car gearbox (see Fig. 9.2 and Fig. 9.3). The images demonstrate high visual quality of both proposed BTF approaches as well as principal differences between probabilistic and pixel-wise reflectance BTF models, which is apparent mainly in part of the object covered by BTF of smooth lacquered wood. Attributes of the individual BTF modelling approaches implemented in the scope of this thesis are compared in Tab. 10.1.

Table 10.1: A comparison of the proposed BTF modeling methods attributes.

	Tiling	PLM-C	GMRF	3D CAR	2D CAR
compression ratio	$\frac{1}{3}$	$\frac{1}{10^2}$	$\frac{1}{10^5}$	$\frac{1}{10^5}$	$\frac{1}{10^5}$
seamless enlargement	Y	Y	Y	Y	Y
anal./synt. separated	Y	Y	Y	Y	Y
block-wise processing	Y	Y	N	Y	Y
GPU implementation	Y	Y	N	Y-	Y
parallel synthesis	N	Y	Y	Y	Y
unseen data	N	Y-	Y	Y	Y

10.2 Future Research

Even though both proposed BTF modelling methods enable fast and visually correct BTF modelling, several problems remain to be solved in this research field to enable wider use of BTFs.

- **BTF interpolation for arbitrary view/illumination direction** is necessary for producing of BTF renderings without visible seams. However, this is costly operation which can take the same time as enumeration of BTF model and consequently considerably prolongs the rendering pipeline.
- **Accuracy of BTF data** is limited in contemporary BTF databases. This is caused by limited accuracy of robotic sample holder and the rectification procedure itself. However, new methods of BTF measurement should solve this problem in near future.
- **BTF standards** are not suggested or developed yet, because there is no standardised BTF compression or modelling method agreed upon. This may be due to the relative novelty of BTF and its complexity, which allows so far only development

of methods precisely tailored to required application and hardware/software platform. Owing to this reason there is not common commercial or public domain BTF renderer available yet.

- Obviously there is vast space of other possible statistical models (not only MRFs) which can be investigated for BTF modelling purposes.

Although the Bidirectional Texture Function is a novel research area and BTF is not used yet as a standard material description in computer graphics its potential is very high as it enables relatively fast photo-realistic modelling of simpler virtual reality scenes. With continually developing BTF measurement systems and graphics techniques as well as increasing computational power it is obvious that the number of possible BTF applications will increase considerably in near future and BTF modelling will add a new level of perfection to contemporary virtual reality systems.

Bibliography

- [1] RealReflect (real time visualization of complex reflectance behaviour in virtual prototyping). URL: <http://www.realreflect.org>, 2001.
- [2] M. Ashikhmin, S. Premoze, and P. Shirley. A microfacet-based BRDF generator. In K. Akeley, editor, *ACM SIGGRAPH 2000*, ACM Press, pages 65–74, 2000.
- [3] M. Ashikhmin and P. Shirley. An anisotropic phong light reflection model. *Journal of Graphics Tools*, 5(2):25–32, 2000.
- [4] D. Banks. Illumination in diverse codimensions. In *ACM SIGGRAPH 1994*, ACM Press, pages 327–334. ACM, July 1994.
- [5] P. Beckmann and A. Spizzichino. *The Scattering of Electromagnetic Waves from Rough Surfaces*. Artech House Publishers, Norwood, MA, 1987.
- [6] J.F. Blinn. Models of light reflection for computer synthesized pictures. In *Computer Graphics Proceedings, Annual Conference Series*, pages 192–198. ACM SIGGRAPH 1977, ACM Press, 1977.
- [7] J.F. Blinn. Simulation of wrinkled surfaces. *ACM SIGGRAPH 1978*, ACM Press, 12(3):286–292, August 1978.
- [8] M.J. Chantler. Why illuminant direction is fundamental to texture analysis. *IEE Proceedings: Vision, Image and Signal Processing*, 142(4):199–206, August 1995.
- [9] W.Ch. Chen, J.Y. Bouguet, M.H. Chu, and R. Grzeszczuk. Light field mapping: Efficient representation and hardware rendering of surface light fields. In *ACM SIGGRAPH 2002*, ACM Press, pages 447 – 456, July 2002.
- [10] F.J.J. Clarke and D.J. Parry. Helmholtz reciprocity: Its validity and application to reflectometry. *Lighting Research and Technology*, 17(1):1–11, 1985.
- [11] M.F. Cohen, J. Shade, S. Hiller, and O. Deussen. Wang tiles for image and texture generation. In *ACM SIGGRAPH 2003*, ACM Press, volume 22, pages 287–294, New York, NY, USA, July 2003.
- [12] R.L. Cook and K.E. Torrance. A reflectance model for computer graphics. *ACM SIGGRAPH 1981*, ACM Press, 15(3):307–316, August 1981.
- [13] H. S. M. Coxeter. *Introduction to Geometry*. New York: Wiley, 1969.

- [14] O.G. Cula and K.J. Dana. Compact representation of bidirectional texture functions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1041–1047, December 2001.
- [15] O.G. Cula and K.J. Dana. 3d texture recognition using bidirectional feature histograms. *International Journal of Computer Vision*, 59(1):33–60, August 2004.
- [16] K.J. Dana, B. van Ginneken, S.K. Nayar, and J.J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, 1999.
- [17] Bonn University BTF Database. *URL: <http://btf.cs.uni-bonn.de/>*, 2003.
- [18] Yale University BTF Database. *URL: <http://pantheon.yale.edu/~mlk34/BTFResearch.html>*, 2003.
- [19] K. Daubert, H. P. A. Lensch, W. Heidrich, and H.-P. Seidel. Efficient cloth modeling and rendering. In *Proceedings of Rendering Techniques '01 - the 12th Eurographics Workshop on Rendering 2001*, Springer Verlag, Berlin, 2001.
- [20] P. A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice-Hall, 1982.
- [21] J. Dong and M. Chantler. Capture and synthesis of 3D surface texture. In *Proceedings of 2nd international workshop on texture analysis and synthesis, Copenhagen (co-located with ECCV)*, volume 1, pages 41–45, June 2002.
- [22] J. Dong and M. Chantler. Comparison of five 3D surface texture synthesis methods. In *Proceedings of the 3rd International Workshop on texture analysis and synthesis (Texture 2003)*, pages 47–52, October 2003.
- [23] O. Drbohlav and R. Šára. Unambiguous determination of shape from photometric stereo with unknown light sources. In *Proceedings 8th ICCV*, pages 581–586, July 2001.
- [24] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In E. Fiume, editor, *ACM SIGGRAPH 2001*, ACM Press, pages 341–346, 2001.
- [25] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *ICCV (2)*, pages 1033–1038, 1999.
- [26] J. Fauvel, R. Flood, and R. J. Eds. Wilson. *Möbius and his Band: Mathematics and Astronomy in Nineteenth-Century Germany*. Oxford University Press, Oxford, England, 1993.
- [27] J. Filip and M. Haindl. Non-linear reflectance model for bidirectional texture function synthesis. In *Proceedings of 17th International Conference on Pattern Recognition*, volume 1, pages 80–84. IEEE Computer Society Press, August 2004.
- [28] EasyRGB Color Conversion Formulas. *URL: <http://www.easyrgb.com/math.php>*, 2000.

- [29] R.T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on Image Processing*, 10(4):439–451, July 1988.
- [30] G. H. Golub and C. F. Van Loan. *Matrix Computations, (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, 1996.
- [31] R.C. Gonzales and R.E. Woods. *Digital Image Processing*. Addison-Wesley, 75 Arlington Street, Suite 300 Boston, MA 02116, USA, 1993.
- [32] M. Gosele, H.P. Lensch, J. Lang, C. Fuchs, and H.-P Seidel. Disco-acquisition of translucent objects. In *ACM SIGGRAPH 2004, ACM Press*, Los Angeles, USA, August 2004. ACM.
- [33] M. Haindl. Texture synthesis. *CWI Quarterly*, 4(4):305–331, December 1991.
- [34] M. Haindl. Texture synthesis. Technical Report CS-R9139, Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands, 1991.
- [35] M. Haindl. An Adaptive Image Reconstruction Method. Technical Report CS-R9537, CWI, Amsterdam, 1994.
- [36] M. Haindl and J. Filip. Fast BTF texture modelling. In *Proceedings of the 3rd International Workshop on texture analysis and synthesis (Texture 2003)*, pages 47–52, October 2003.
- [37] M. Haindl and J. Filip. A fast probabilistic bidirectional texture function model. In *Proceedings of International Conference on Image Analysis and Recognition. (Lecture Notes in Computer Science. 3212)*, volume 2, pages 298–305, Berlin Heidenberg, September 2004. Springer-Verlag.
- [38] M. Haindl, J. Filip, and M. Arnold. BTF image space utmost compression and modelling method. In *Proceedings of 17th International Conference on Pattern Recognition*, volume 3, pages 194–198. IEEE Computer Society Press, August 2004.
- [39] M. Haindl and V. Havlíček. Prototype implementation of the texture analysis objects. Technical Report 1939, ÚTIA AV ČR, Praha, 1997.
- [40] M. Haindl and S. Šimberová. *Theory & Applications of Image Analysis*, chapter A Multispectral Image Line Reconstruction Method, pages 306–315. World Scientific Publishing Co., Singapore, 1992.
- [41] M. Haindl and S. Šimberová. A high-resolution radiospectrograph image reconstruction method. *Astronomy and Astrophysics, Suppl.Ser.*, 115(1):189–193, January 1996.
- [42] X. He, K.E. Torrance, F.X. Sillion, and D.P. Greenberg. A comprehensive physical model for light reflection. *Computer Graphics*, 25(4), July 1991.
- [43] D.J. Heeger and J.R. Bergen. Pyramid based texture analysis/synthesis. *ACM SIGGRAPH 2005, ACM Press*, pages 229–238, 1995.

- [44] B.K.P. Horn and M.J. Brooks. The variational approach to shape from shading. *Computer Vision Graphics Image Processing*, 33:174–208, February 1986.
- [45] H.W. Jensen, S.R. Marschner, M. Levoy, and P. Hanrahan. A practical model for subsurface light transport. *ACM SIGGRAPH 2001, ACM Press*, pages 511–518, August 2001.
- [46] B. Julesz. Visual pattern discrimination. *IRE Transaction on Information Theory*, 8(1):84–92, February 1962.
- [47] B. Julesz. Textons, the elements of texture perception and their interactions. *Nature*, 290:91–97, 1981.
- [48] B. Julesz, E.N. Gilbert, and J.D. Victor. Visual discrimination of textures with identical third-order statistics. *Biol. Cybernetics*, 31:137–140, 1978.
- [49] N. Kambhatla and T.K. Leen. Dimension reduction by local PCA. *Neural Computation*, 9:1493–1516, 1997.
- [50] R.L. Kashyap. Analysis and synthesis of image patterns by spatial interaction models. In L.N. Kanal and A. Rosenfeld, editors, *Progress in Pattern Recognition 1*, North-Holland, 1981. Elsevier.
- [51] J. Kautz, W. Heidrich, and H.-P. Seidel. Real-time bump map synthesis. In *Proceedings of the Eurographics/SIGGRAPH Workshop on Graphics Hardware 2001*, pages 109–114, August 2001.
- [52] J. Kautz and M.D. McCool. Interactive rendering with arbitrary BRDFs using separable approximations. In *Proceedings of the 10th Eurographics Workshop on Rendering*, pages 281–292, 1999.
- [53] J. Kautz and H.-P. Seidel. Towards interactive bump mapping with anisotropic shift-variant BRDFs. In *Proceedings of the Eurographics/SIGGRAPH Workshop on Graphics Hardware 2000*, pages 51–58. Eurographics Association, Switzerland, August 2000.
- [54] J. Kautz and H.-P. Seidel. Hardware accelerated displacement mapping for image based rendering. In *Graphics Interface 2001*, pages 61–70, June 2001.
- [55] J. Kautz, P.-P. Sloan, and J. Snyder. Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics. In *Proceedings of the 12th Eurographics Workshop on Rendering*, pages 301–308, June 2002.
- [56] M.J. Kilgard. A practical and robust bump-mapping technique for today’s GPUs. In *Game Developers Conference - GDC 2000*, March 2000.
- [57] R. Klette and K. Schlüns. Height data from gradient fields. In *Proceedings of Machine Vision Applications, Architectures, and System Integration V SPIE, Boston, Massachusetts*, volume 2908, November 1996.
- [58] J. Kostková and R. Šára. Stratified dense matching for stereopsis in complex scenes. In *Proceedings BMVC*, pages 339–348, 2003.

- [59] M.L. Koudelka, S. Magda, P.N. Belhumeur, and D.J. Kriegman. Acquisition, compression, and synthesis of bidirectional texture functions. In *Proceedings of the 3rd International Workshop on texture analysis and synthesis (Texture 2003)*, pages 47–52, October 2003.
- [60] S. Kullback. *Information Theory and Statistics*. Dover Books, New York, USA, 1997.
- [61] A. Kumar, Y. Sabharwal, and S. Sen. A simple linear time $(1+\epsilon)$ -approximation algorithm for k -means clustering in any dimensions. In *45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04)*, pages 454–462. IEEE, October 2004.
- [62] V. Kwatra, A. Schodl, I. Essa, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM SIGGRAPH 2003, ACM Press*, 22(2):277–286, July 2003.
- [63] E. P. Lafortune, S. Ch. Foo, K. E. Torrance, and D. P. Greenberg. Non-linear approximation of reflectance functions. *Computer Graphics*, 31(Annual Conference Series):117–126, 1997.
- [64] E. P. Lafortune and Y. D. Willems. Using the Modified Phong BRDF for Physically Based Rendering. Technical Report CW197, Department of Computer Science, KU.Leuven, Leuven, Belgium, 1994.
- [65] L. Latta and A. Kolb. Homomorphic factorization of BRDF-based lighting computation. *ACM SIGGRAPH 2002, ACM Press*, 21(3):509–516, July 2002.
- [66] Y.G. Leclerc and A.F. Bobick. The direct computation of height from shading. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 552–558, 1991.
- [67] H.P.A. Lensch, M. Goesele, P. Bekaert, J. Kautz, M.A. Magnor, J. Lang, and H.-P. Seidel. Interactive rendering of translucent objects. In *Proceedings of Pacific Graphics '02*, pages 214–224, Los Alamitos, July 2002. IEEE Computer Society.
- [68] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.
- [69] D. Levesque and F. Deschênes. Sparse scene structure recovery from atmospheric degradation. In *Proceedings of 17th International Conference on Pattern Recognition*, pages 85–88. IEEE Computer Society Press, August 2004.
- [70] X. Liu, Y. Yu, and H. Y. Shum. Synthesizing bidirectional texture functions for real-world surfaces. In E. Fiume, editor, *ACM SIGGRAPH 2001, ACM Press*, pages 97–106, 2001.
- [71] X. Liu, J. Zhang, X. Tong, B. Guo, and H.-Y. Shum. Synthesis and rendering of bidirectional texture functions on arbitrary surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 10(3):278–289, June 2004.

- [72] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. In *ACM SIGGRAPH 2001*, *ACM Press*, pages 519–528. Eurographics Association, Switzerland, 2001.
- [73] J. Matas and O. Chum. Randomized ransac with t , d test. *Image Vision Comput.*, 22(10):837–842, 2004.
- [74] W. Matusik, M. Pfister, H.P. Brand, and L. McMillan. A data-driven reflectance model. In *ACM SIGGRAPH 2003*, *ACM Press*, Los Angeles, USA, 2003. ACM.
- [75] D. K. McAllister, A. Lastra, and W. Heidrich. Efficient rendering of spatial bidirectional reflectance distribution functions. *Graphics Hardware*, pages 77–88, 2002.
- [76] M.D. McCool, J. Ang, and A. Ahmad. Homomorphic factorization of BRDFs for high-performance rendering. In E. Fiume, editor, *ACM SIGGRAPH 2001*, *ACM Press*, pages 185–194, 2001.
- [77] T. McReynolds and D. Blythe. *Advanced Graphics Programming Techniques Using OpenGL*. Morgan Kaufmann, 2005.
- [78] J. Meseth, G. Müller, and R. Klein. Preserving realism in real-time rendering of bidirectional texture functions. In *OpenSG Symposium 2003*, pages 89–96. Eurographics Association, Switzerland, April 2003.
- [79] J. Meseth, G. Müller, M. Sattler, and R. Klein. BTF rendering for virtual environments. In *Virtual Concepts 2003*, pages 356–363, November 2003.
- [80] M. Meyer, H. Lee, and M. Desburn. Generalized barycentric coordinates on irregular polygons. *Journal of graphics tools*, 7(1):13–22, November 2002.
- [81] G. Miller, S. Rubin, and D. Ponceleon. Lazy decompression of surface light fields for precomputed global illumination. In *Proceedings of 9th Eurographics Workshop on Rendering*, pages 281–292, June 1998.
- [82] G. Müller, J. Meseth, and R. Klein. Compression and real-time rendering of measured BTFs using local PCA. To appear in proceedings of Vision, Modeling and Visualisation 2003, November 2003.
- [83] G. Müller, J. Meseth, M. Sattler, R. Sarlette, and R. Klein. Acquisition, synthesis and rendering of bidirectional texture functions. To appear in Proceedings of Eurographics 2004 State of the Art Reports, September 2004.
- [84] S.K. Nayar and M. Oren. Generalization of the lambertian model and implications for machine vision. *International Journal of Computer Vision*, 14:227–251, 1995.
- [85] A. Neubeck, A. Zalesny, and L. Van Gool. Viewpoint consistent texture synthesis. In *3D Data Processing, Visualization and Transmission (3DPVT)*, pages 388–395. CS, IEEE, September 2004.
- [86] A. Neumann. *Constructions of Bidirectional Reflection Distribution Functions*. PhD thesis, Vienna University of Technology, Institute of Computer Graphics and Algorithms, Favoritenstr. 9-11 / E186, A-1040 Vienna, Austria, 2001.

- [87] L. Neumann, A. Neumann, and L. Szirmay-Kalos. Compact metallic reflectance models. *Computer Graphics Forum*, 18(13), 1999.
- [88] F.E. Nicodemus, Richmond J.C., J.J. Hsia, I.W. Ginsburg, and T. Limperis. Geometrical considerations and nomenclature for reflectance. *NBS Monograph 160, National Bureau of Standards, U.S. Department of Commerce, Washington, D. C.*, pages 1–52, October 1977.
- [89] K. Nishino, Y. Sato, and K. Ikeuchi. Eigen-texture method: Appearance compression and synthesis based on a 3D model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1257–1265, 2001.
- [90] M.M. Oliveira, G. Bishop, and D. McAllister. Relief texture mapping. In *27th annual conference on Computer graphics and interactive techniques*, pages 359–368, New York, NY, USA, March 2000. ACM Press/Addison-Wesley Publishing Co.
- [91] B.-T. Phong. Illumination for computer generated images. *Communications of the ACM*, 18(6):311–317, June 1975.
- [92] F. Policarpo, M.M. Oliveira, and J. Comba. Real-time relief mapping on arbitrary polygonal surfaces. In *Proceedings of ACM Siggraph Interactive 3D Graphics and Games 2005*, New York, NY, USA, 2005. ACM Press/Addison-Wesley Publishing Co.
- [93] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–71, 2000.
- [94] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992.
- [95] J. Puzicha, T. Hofmann, and J. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 267–272, 1997.
- [96] Columbia-Utrecht Reflectance and Texture Database. *URL: <http://www1.cs.columbia.edu/CAVE/curet/>*, 1999.
- [97] V. Rodehorst. Vertiefende Analyse eines Gestalts-Constrains von Aloimonos und Shulman. Technical Report CV-Bericht 8, Institut für Technische Informatik, TU Berlin, 1993.
- [98] M. Sattler, R. Sarlette, and R. Klein. Efficient and realistic visualization of cloth. In *Eurographics Symposium on Rendering 2003*, June 2003.
- [99] Ch. Schlick. An inexpensive BRDF model for physically-based rendering. *Computer Graphics Forum (EUROGRAPHICS'94)*, 13(3):149–162, August 1994.
- [100] P. Shirley, H. Hu, B. Smith, and E. Lafortune. A practitioners' assessment of light reflection model. In *Proceedings of Pacific Graphics '97*, pages 40–49, Los Alamitos, October 1997. IEEE Computer Society.

- [101] D. Shreiner, M. Woo, J. Neider, and T. Davis. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.4, 4th Edition*. Addison Wesley Professional, 2003.
- [102] P.-P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *ACM SIGGRAPH 2002*, *ACM Press*, pages 527–536, July 2002.
- [103] P. Somol and M. Haindl. Novel path search algorithm for image stitching and advanced texture tiling. In *Proceedings of 13-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG05*, February 2005.
- [104] M. Šonka, V. Hlaváč, and R.D. Boyle. *Image Processing, Analysis and Machine Vision*. PSW, Boston, USA, 1998.
- [105] H. Spath. *Cluster Analysis Algorithms*. Ellis Horwood Publishers, 1980.
- [106] A.D. Spence and M.J. Chantler. On capturing 3D isotropic surface texture using uncalibrated photometric stereo. In *Proceedings of the 3rd International Workshop on texture analysis and synthesis (Texture 2003)*, pages 83–88, October 2003.
- [107] A.D. Spence and M.J. Chantler. Optimal illumination for three-image photometric stereo acquisition of surface texture. In *Proceedings of the 3rd International Workshop on texture analysis and synthesis (Texture 2003)*, pages 89–94, October 2003.
- [108] J. Stam. Diffraction shaders. *ACM SIGGRAPH 1999*, *ACM Press*, pages 101–110, 1999.
- [109] P. Suen and G. Healey. The analysis and recognition of real-world textures in three dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):491–503, May 2000.
- [110] F. Suykens, K. Berge, A. Lagae, and P. Dutr. Interactive rendering with bidirectional texture functions. In *Computer Graphics Forum*, pages 463–472. Blackwell Publishing on behalf of the European Association for Computer Graphics, September 2003.
- [111] X. Tong, J. Zhang, L. Liu, B. Wang, X. Guo, and H.-Y. Shum. Synthesis of bidirectional texture functions on arbitrary surfaces. In *ACM SIGGRAPH 2002 ACM Press*, pages 665–672, Los Angeles, USA, 2002. ACM.
- [112] K. Torrance and E. Sparrow. Theory for off-specular reflection from rough surfaces. *Journal of the Optical Society of America*, 57(9):1105–1114, February 1967.
- [113] B. van Ginneken, J.J. Koenderink, and K.J. Dana. Texture histograms as a function of irradiation and viewing direction. *International Journal of Computer Vision*, 31(2-3):169–184, April 1999.
- [114] M.A.O. Vasilescu and D. Terzopoulos. TensorTextures: Multilinear image-based rendering. *ACM SIGGRAPH 2004*, *ACM Press*, 23(3):336–342, August 2004.

- [115] X. Wang, X. Tong, S. Lin, S. Hu, B. Guo, and H.-Y. Shum. View-dependent displacement mapping. *ACM SIGGRAPH 2002, ACM Press*, 22(3):334–339, 2003.
- [116] G.J. Ward. Measuring and modeling anisotropic reflection. *Computer Graphics*, 26(2), July 1992.
- [117] G.J. Ward, F.M. Rubinstein, and R.D. Clear. A ray tracing solution for diffuse interreflection. *Computer Graphics*, 22(4):85–92, July 1988.
- [118] T. Welsch. Parallax mapping with offset limiting: A per-pixel approximation of uneven surfaces. Technical Report Revision 0.3, Infiscape Corporation, 2004.
- [119] L. Williams. Pyramidal parametrics. In *ACM SIGGRAPH 1983, ACM Press*, pages 1–11, 1983.
- [120] T.-T. Wong, P.-A. Heng, S.-H. Or, and W.-Y. Ng. Image-based rendering with controllable illumination. In J. Dorsey and P. Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 13–22, New York, NY, 1997. Springer Wien.
- [121] R.J. Woodham. Analysing images of curved surface. *Artificial Intelligence*, 17(5):117–140, 1981.
- [122] C. Wynn. Implementing bump-mapping using registers combiners. Technical report, NVIDIA Corporation, 2003.
- [123] G. Wyszecki and W.S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae: 2nd Edition*. John Wiley & Sons, 2001.
- [124] H.O. Yacov, T. Malzbender, and D. Gelb. Synthesis of reflectance function textures from examples. In *Proceedings of the 3rd International Workshop on texture analysis and synthesis (Texture 2003)*, pages 47–52, October 2003.
- [125] J.I. Yellott. Implications of triple correlation uniqueness for texture statistics and the Julesz conjecture. *Journal of Optical Society of America A*, 10(5):777–793, May 1993.
- [126] R. Zhang, P.S. Tsai, J.E Cryer, and M.. Shah. Shape from shading: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999.

Appendix A

Proposed Probabilistic Models' Results

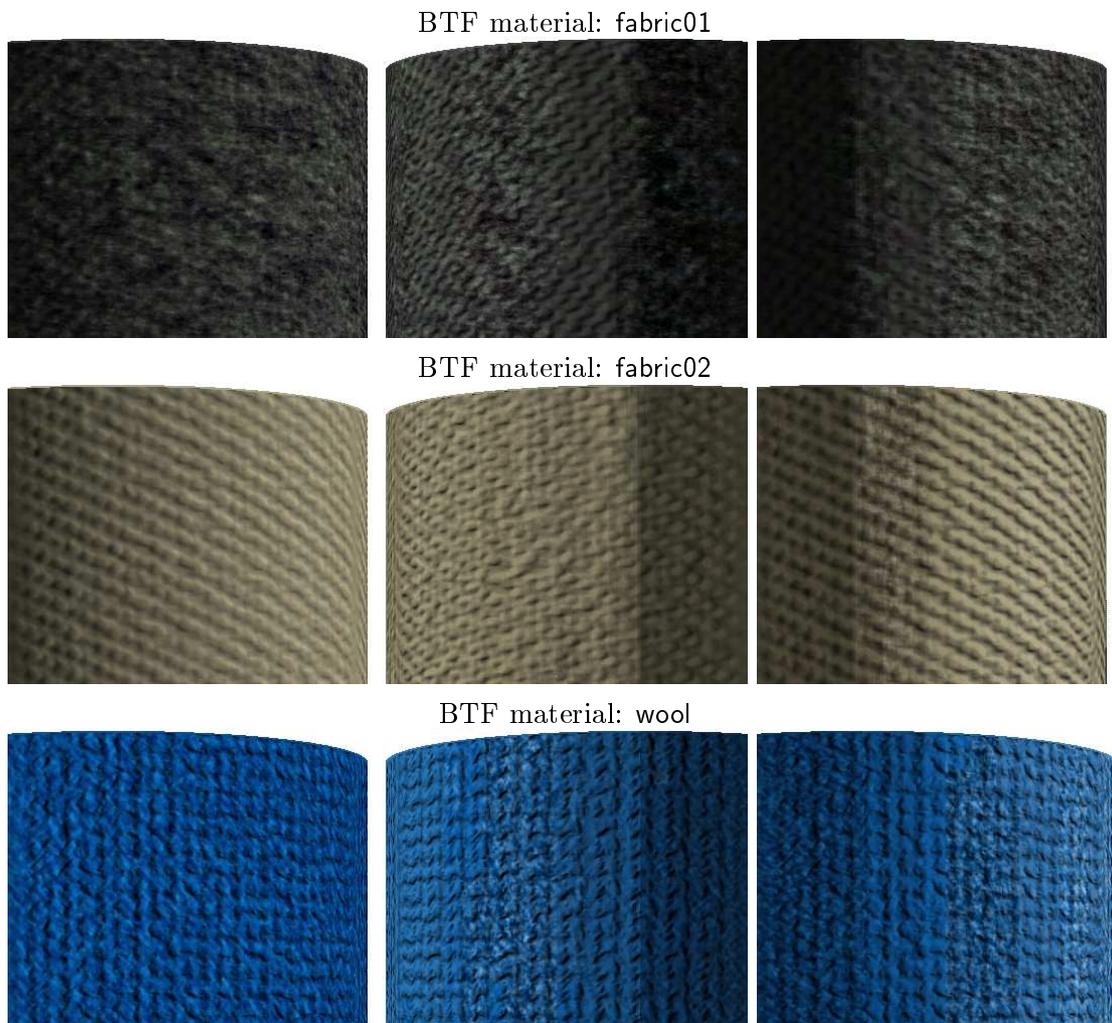


Figure A.1: Bump-mapping (left) in comparison with proposed 2D CAR probabilistic BTF model on part of cylinder lighted from left and right respectively.

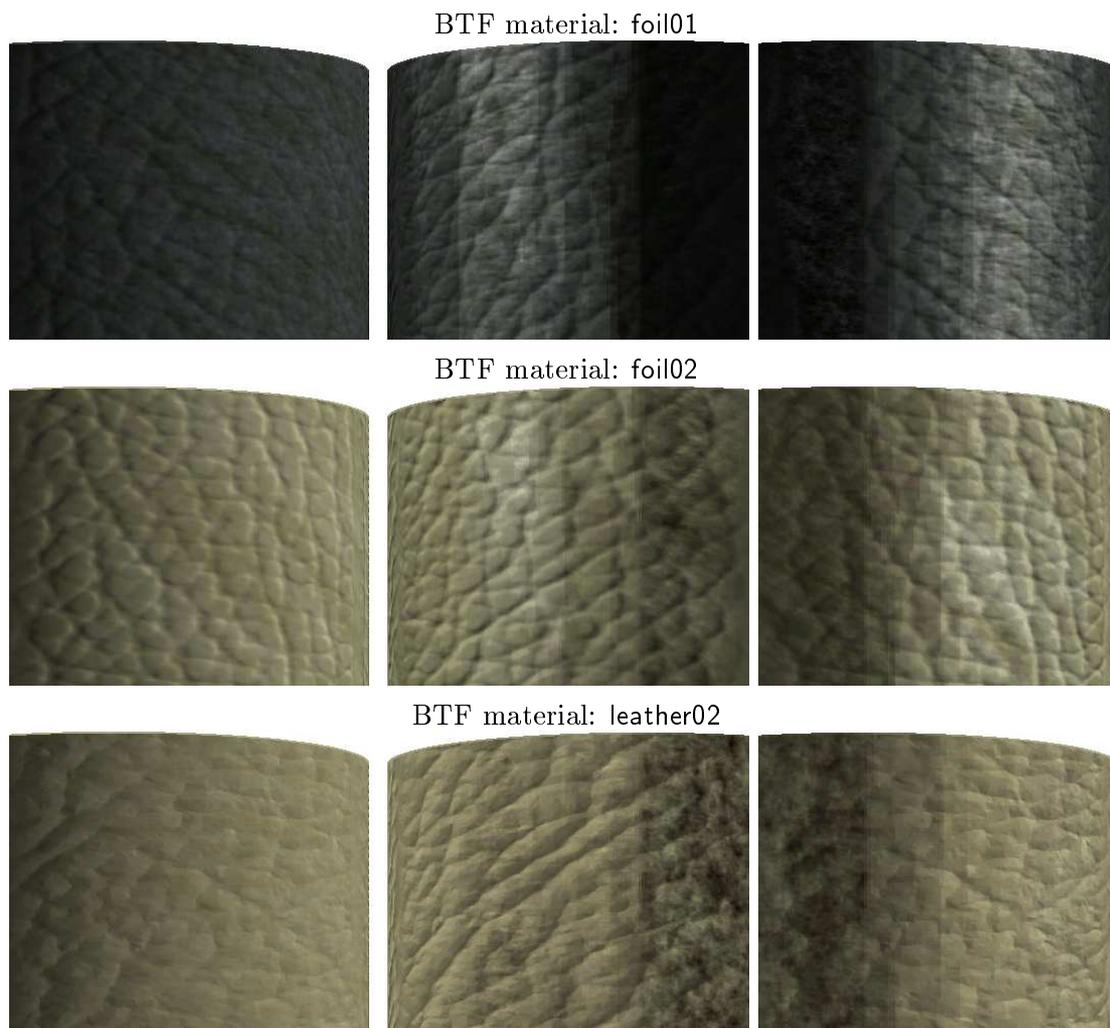


Figure A.2: Bump-mapping (left) in comparison with proposed 2D CAR probabilistic BTF model on part of cylinder lighted from left and right respectively.

Appendix B

Proposed Reflectance Models' Results

See other page.



Figure B.1: Part of a car armrest covered with BTf when only view angle interpolation is used. Tiled original BTf data (first column), results of one-lobe PLM (second column) and one-lobe PLM-C (third column) for eight different materials: *fabric01*, *fabric02*, *foil01*, *foil02*, *knitted wool*, *leather02* *wood01*, *wood02*.

Appendix C

Contents of the Enclosed CD

With this thesis is enclosed a CD comprising:

- HTML presentation of proposed BTF model's results. This document is also available on html://www.utia.cas.cz/R0/demos/dt_jf/dt_jf.html.
- Animations of car interior parts covered with results of the proposed BTF models for varying view and illumination directions.
- Electronic version of this document in PDF format.