Rendering of Complex Materials for Driving Simulators

Thierry Lefebvre [1,2]
+33 (0)1.76.85.06.64
thierry.t.lefebvre@renault.com

Andras Kemeny [1]
+33 (0)1.76.85.19.85
andras.kemeny@renault.com

Didier Arquès [2]
+33 (0)1.60.95.77.17
didier.arques@univ-mlv.fr

Loïc Joly [1]
+33 (0)1.76.85.19.65
loic.joly@renault.com

Sylvain Michelin [2]
+33 (0)1.60.95.77.31
michelin@univ-mlv.fr

[1] Technical Centre for Simulation
Technocentre Renault, TCR AVA O13
1 avenue du golf, 78288 Guyancourt Cedex.
Renault, France.

[2] Laboratory of Image synthesis
Batîment Charles Cros
6 Cours du Danube -77700 Serris.
University of Marne-la-Vallée, France.

Abstract

The visual realism of a virtual scene is very important for driving simulators. In real life, a scene is composed of complex materials (asphalt, vegetation, road mark...) interacting with light sources (sun, headlamps...). Our goal is to simulate such environments in real time under various weather conditions (dry or wet road). The lighting interaction cannot be simulated directly using standard lighting model implemented in graphic board (Phong shading). In this paper, we present a method for rendering complex materials with BTF (Bidirectional Texture Function). This method, which allows to view various and complex lighting phenomenon (such as interreflections, self-shadowing and masking effects), requires a long computation chain and uses a large data set (one or more gigabytes). The main steps of this method are exposed : BTF acquisition (both virtual and through measurement), data compression (a dedicated algorithm can achieve compressions rates above 2000:1), and real time implementation techniques using vertex and pixel shaders. We present our first results, applications and discuss about limitations and future works of this rendering technique.

Introduction

Objectives

The realistic simulation of virtual scenes is one of the main challenges in today's computer graphics. It depends mostly on the interactions between light sources (sun, headlamp…) and materials (asphalt, road marking, vegetation…). These interactions are much more complex than the lighting model implemented in graphic board (Phong lighting model). This paper presents a rendering technique of complex materials based on BTF (Bidirectional Texture Function). In the first part, we present main concepts of material representation. In the second part, we describe an algorithm dedicated to real time rendering of materials using BTF. Finally, we discuss about advantages and limitations of the method before presenting results and future works. Our final goal is to simulate realistic environments under various weather conditions (dry or wet road) for both headlight simulator and daytime driving simulator.

Today's lighting simulation

The Technical Centre for Simulation of Renault (CTS) has developed a driving simulation software (SCANeR©II). It relies on several independant modules using a network protocol. The software enables programmers to build complete driving simulator with immersive stereoscopic visualisation, dynamic models, trafic and scenario generation, sound simulation, and comprises many tools for data analysis and road network modeling. It can be used for many different studies such as human factors, vehicle systems ergonomics, vehicle behaviour, perception (navigation), basic and advanced driver training, headlight assessment or even road infrastructure.

Figure 1 : headlight simulator (left), lighting simulation within fog (center), glare effects (right)

The real-time headlight simulation module is used for the assessment of new projectors since 1998 [LK99]. Its headlight main features comprises lighting simulation within fog and simulation of glare effects coming from trafic vehicles (figure 1).

To compute lighting in real time, the tool calls upon classical rendering techniques : projected textures (for headlamps) and hardware lighting model (Phong [Pho75]) for surfaces reflection.
Headlamps are described by their photometric data (luminance and color information). Surfaces are purely diffuse, the same amount of light is reflected in all directions (figure 2).

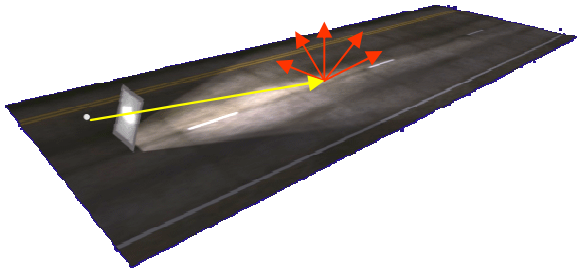


Figure 2 : diffuse reflection on road surface.

Material representations

Overview

Traditionally the geometry of a surface is modeled explicitly (e.g. with mesh) only up to a certain scale (figure 3).

Fine and planar details on surfaces are represented by textures. Geometric details are generally modeled using tricks, such as bump mapping or displacement mapping.

The micro-structure responsible for the reflectance behaviour of a material is simulated using lighting models (Phong or reflectance function described in the following section).

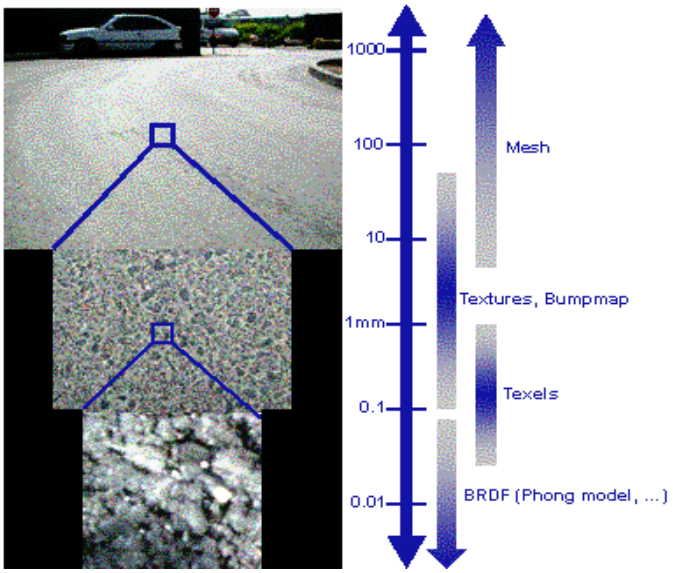
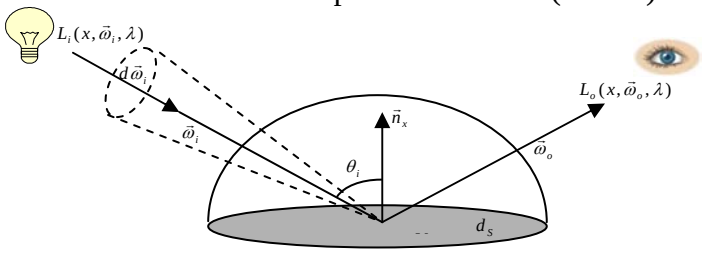


Figure 3 : modeling techniques.

BRDF (Bidirectional Reflectance Distribution Function)

Reflectance properties of a material are commonly represented by a function called BRDF (figure 4). This function describes, for an incoming illuminance, the amount of reflected luminance in a specific direction (albedo) and can be described as follow:



$$f_r(\vec{\omega}_i,\vec{\omega}_o,x,\lambda)=\frac{L_o(x,\vec{\omega}_o,\lambda)}{L_i(x,\vec{\omega}_i,\lambda)\cos\theta_i d\omega_i}$$

$\vec{\omega}_i$ = Light vector
$\vec{\omega}_o$ = View Vector
$d\omega_i$ = Solid angle around light vector
θ_i = Angle between light vector and surface normal
λ = Wavelength
L_i = Received Luminance
L_o = Reflected luminance

Figure 5 : BRDF representation

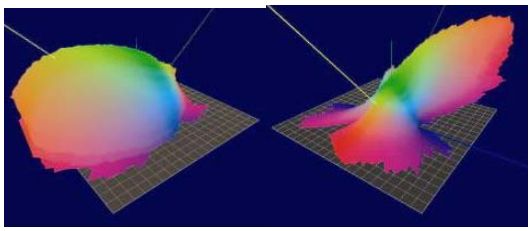


Figure 4 : measures of dry (left) and wet (right) asphalt.

BRDF measurements of wet and dry asphalt are represented in false colors on figure 5. You can see a large retro diffusion around the incoming ray (right) for dry asphalt whereas a fine specular peak around the reflected ray for wet asphalt. Some noise is also visible (especially on wet asphalt) at grazing angles due to solid angle (more the signal is weak, more the noise is relevant). Thus, it could be interesting to apply a noise filter or find a numerical model that match the measured data.

Using a BRDF functions, materials with coarse variations can be well represented, only for far viewers (lighting variations are smaller than a pixel size). The difficulty arises from the complex mesostructure and reflectance behaviour defining the unique look-and-feel of a material.

Figure 6 : rendering based on measured BRDF of wet asphalt combined with normal mapping.

Since the BRDF already contains light interactions within the asphalt, it is not physically correct to apply bump-mapping techniques to render asphalt roughness. Thus, we need a spatially variant function to describe complex and heterogeneous materials (asphalt, vegetation…). We describe in the next part a suitable function recently introduced by Dana and al. in 1999 [DvGNK99].

BTF (Bidirectional Texture Function)

Overview

The BTF can be seen as a set of 2D-textures (figure 8), where each one corresponds to a given lighting and viewing direction.
A BTF is very similar to BRDF but provides a texture instead of an albedo. Thus, it can model the fine-scale self-shadows, self-occlusions, and specularities caused by surface mesostructure.
A wide class of materials (asphalt, marble, wool…) can be described by a such function.

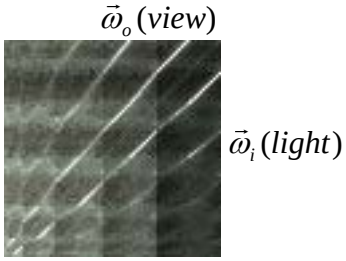


Figure 7 : example of ABRDF

A BTF can also be represented as a set of Apparent BRDF (figure 7). An ABRDF is not exactly defined as a BRDF since it does not respect fundamental physic principles (Helmotz reciprocity and energy conservation). Effectively, there are strong non-linear effects dues to self-shadowing (and masking) of the material.
This function has at least six dimensions : 4D as non-spectral BRDF and 2D for spatiality information.

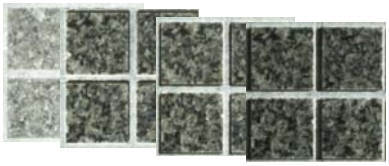


Figure 8 : samples of measured BTF from Bonn database [BonnDB] for few angles of lighting and viewing direction.

Texture resolution	x	Eye positions	x	Light positions	x	Component RGB	=	**1,3 Gb**
256x256		81		81		3		

Due to its huge size (1,3 Gb in our example), a BTF cannot be implemented directly in real time with today's graphic boards. The second difficulty arises from the dimension of this function: 6D textures are not available.

In order to achieve real-time frame rates and acceptable memory consumption, some sort of data-compression has to be achieved. Method should of course preserve as much as possible, the relevant features of the BTF. It should also exploit the data redundancy in an efficient way and provide a real-time decompression stage for our application.

BTF: from generation to real-time implementation

Synthetic BTF Generation

The acquisition of BTF can be achieved by several ways. It can be measured using an installation based on CCD camera [HP03], [KMBK03]. Few measured BTF databases are available online : CUReT database [CURetDB] and Bonn University database [BonnDB].The samples from CUReT database are not spatially registered, thus its use in our experiments is tiresome.

Measurement of BTF requires an expansive system and a controlled sophisticated process. In order to experiment BTF, we thought of data processing sequences for generating synthetic BTF samples. We generate good quality images (figure 9), using a ray-tracer (*Povray)* because it is easily customizable (animation scripts are used for both camera and light displacements). We have developed a tool that computes homographic transformation in order to replace the BTF samples in the camera space (spatially registered samples)

Figure 9 : synthetic BTF sample

Compression algorithm

We have chosen to implement an algorithm introduced by Suykens and al. [SvBLD03]. They provide a BTF rendering method with high compression rates at interactive frame rates.

This algorithm is based on the Singular Value Decomposition (SVD) and k-mean clustering.

The SVD is used in order to represent a 4D function (ABRDF) by a product of 2D function (singular vectors). This idea has been introduced in [KM99] and [MAA01]. The compression is achieved by reducing the size *n* of the matrix *D* (for n = 1, size decreases from $n.m$ to $2n+1$).

$$M = U.D.V^T$$

M = matrix (n x *m)*
U = orthogonal matrix (m x *n)*
D = diagonal matrix (n x *n)*
V = orthogonal matrix (n x *m)*

Results of SVD compression are much better using the GSHD (Gramm-Schmidt Halfway-Difference) parameterization, since it correspond to the optimal space for representing ABRDF.

Although the singular value decomposition of all ABRDFs, the BTF remains too big.
A sort of k-mean clustering is applied to reduce datasize by gathering similar singular vectors (We keep only 256 different ABRDFs).

An overview of the complete process can be described as follow:

- Apply GSHD parameterization on input ARBDFo.
- Compute SVD, keep only one singular vector for halfway component (H).
- Reconstruct the compressed ABRDFh and divide the original one (ARBDFo) by ABRDFh, you obtain the resultant ABRDFr
- Compute SVD on ABRDFr, keep two singular vectors (U and V)
- Apply k-mean clustering on singular vectors, keep 256 vectors represented as parabolic maps (figure 10) for each group U, V and H. Save an index map to retrieve the corresponding clustered map to each texel.

Interactive rendering

The Final representation of material requires only 640 Kb, corresponding to a compression rate of 2030:1, independently on the material complexity.

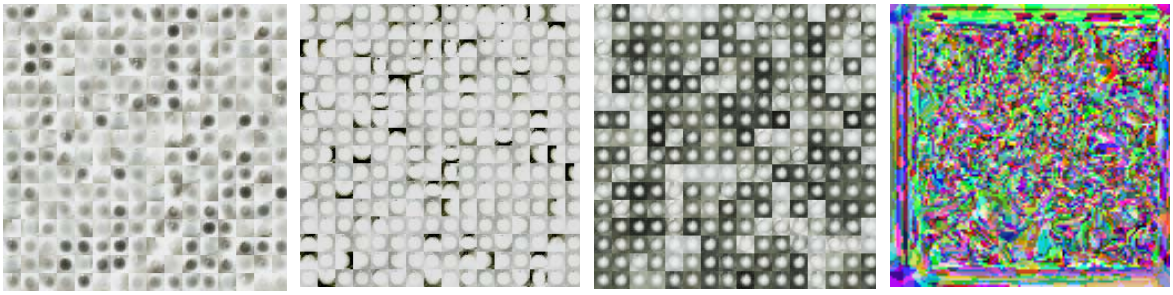


Figure 10 : BTF sample (impalla from Bonn database) after compression process (U, V, H and an index map).

The indexmap is a classical 2D texture mapped on geometry as a standard texture. Each group U, V, H is stored in a 3D texture. We can rebuild the BTF very easily by multiplying components of singular vectors (U*V*H). Therefore, this algorithm is very suitable for real time rendering using a shading language (GLSL or other).

Results

We have tested two natural BTF samples provided by Bonn University (figure11 and 12). We have also tested the algorithm on a synthetic BTF (figure 13). As expected, results are good for very repetitive materials (Corduroy sample). Shadows are also well visible on squares borders on impalla sample but the overall image quality is not very satisfying.

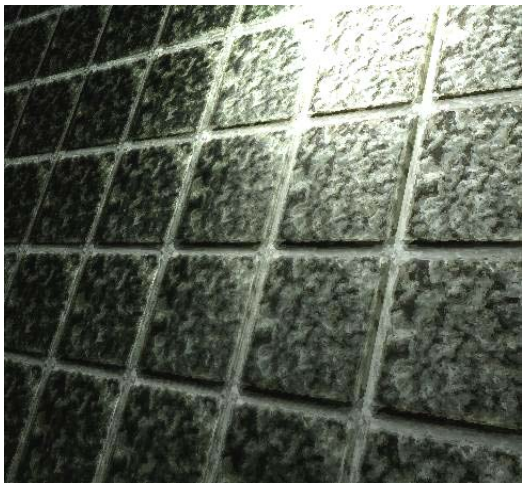


Figure 11 : Impalla sample (from Bonn database)

Figure 12 : Corduroy sample (from Bonn database)

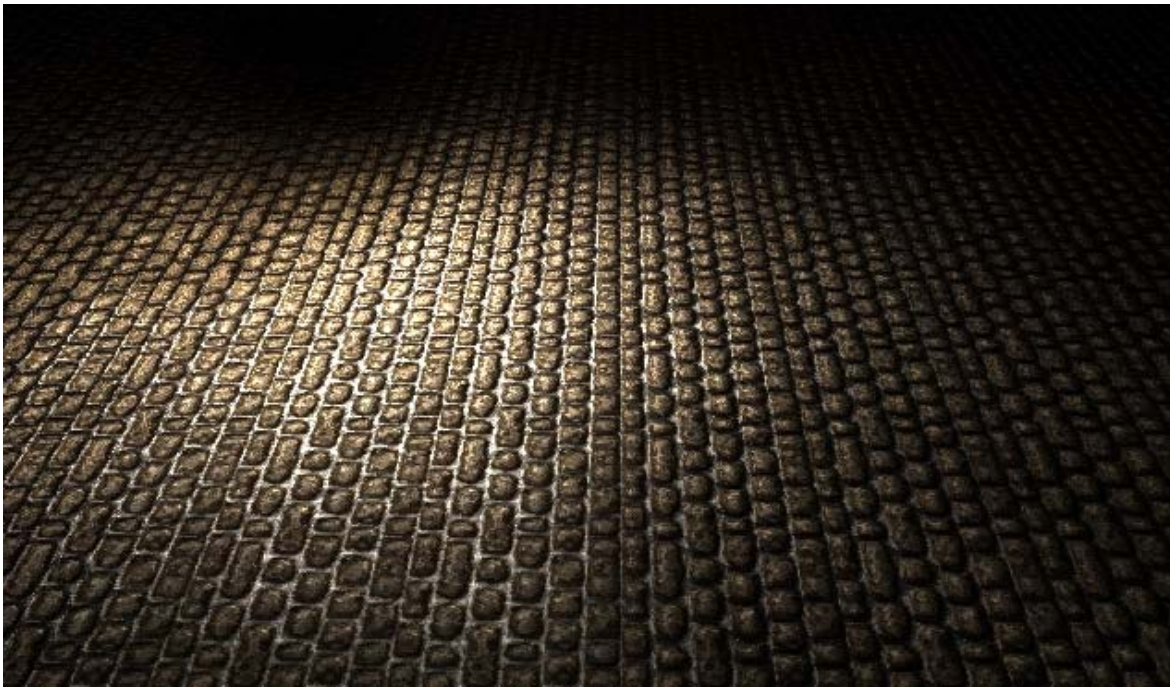


Figure 13 : synthetic pavement BTF

The synthetic BTF of pavement (Figure 13) has been generated using a ray-tracer. Specular reflections can be seen between paving stones that coming from shallow water. Image quality is quite damaged compared to the original sample (see Figure 9) which is very detailed.

The rendering performances for the given examples is in the range of about 20 to 60 frames per second, at a resolution of 1280×1024. All performance measurements were obtained on a BiXeon 2.6GHz using an Nvidia Quadro FX2000.

Conclusion and future work

In this paper, we have presented a real time implementation of measured BRDF combined with bump mapping techniques (geometry information). We have also presented an image based rendering method (BTF) for realistic material simulation, without assumption on the underlying geometry. The presented method provides high compression rate, easy and fast reconstruction (only three texture products) which allows real time rendering of BTF. We have developed a method to generate synthetic BTF samples (image computation and homographic transformation) and we have tested this algorithm on both synthetic and natural BTF samples.

However, this method has a few disadvantages. Image quality can be poor depending on the material complexity. A vertex local space (tangent, binormal, normal) must be correctly defined to compute angles for accessing the BTF, therefore, high quality mesh are required.

This method does not support mip-map filtering which is very important for scene observed at a driver position (grazing angles with long egocentric distances).

In the future, we will make more precise compression error assessments and comparisons with further algorithms based on principal component analysis (PCA) [Jol86], or Local PCA [MMK03], [Sch04]. We will also study new algorithms more suitable for driving simulator (mipmaping support and random mapping to avoid visible repetitive patches [TZL02]).

We are also interested by considering human perception of speed. Since our display system has limitations (spatial resolution, temporal aliasing, luminance…), it could be very important to consider specific algorithms (motion blur, tone mapping…) to generate a virtual scene in accordance with the most relevant human perception properties.

References

[LK99] Lecocq P., Kelada J-M., Kemeny A., *Interactive Headlight Simulation*, in Proceedings of DSC'00 Driving Simulation Conference, 1999, pp. 173-180.

[DvGNK99] DANA K. J., VAN GINNEKEN B., NAYAR S. K., KOENDERINK J. J.: Reflectance and texture of real world surfaces. *ACM Transactions on Graphics 18*, 1 (1999), 1–34.

[HP03] HAN J. Y., PERLIN K.: Measuring bidirectional texture reflectance with a kaleidoscope. *ACM Trans.*
Graph. 22, 3 (2003), 741–748.

[Jol86] JOLLIFFE I.: *Principal Component Analysis*. Springer-Verlag, 1986.

[KM99] KAUTZ J., MCCOOL M.: Interactive Rendering with Arbitrary BRDFs using Separable Approximations. In *Tenth Eurographics Workshop on Rendering* (1999), pp. 281–292.

[KMBK03] KOUDELKA M. L., MAGDA S., BELHUMEUR P. N., KRIEGMAN D. J.: Acquisition, compression and synthesis of bidirectional texture functions. In *3rd International Workshop on Texture Analysis and Synthesis* (2003).

[LHZ04] LIU X., HU Y., ZHANG J., TONG X., GUO B.,SHUM H.-Y.: Synthesis and Rendering of Bidirectional Texture Functions on Arbitrary Surfaces. *IEEE Transactions on Visualization and Computer Graphics* 3 (2004), 278–289.
[MAA01] MCCOOL M. D., ANG J., AHMAD A.: Homomorphic Factorization of BRDFs for High-Performance Rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM Press, pp. 171–178.

[MMK03] MÜLLER G., MESETH J., KLEIN R.: Compression and real-time Rendering of Measured BTFs using local PCA. In *Vision, Modeling and Visualisation 2003* (November 2003), pp. 271–280.

[MMSK04] J., MÜLLER G., SATTLER M., KLEIN R : Acquisition, Synthesis and Rendering of Bidirectional Texture Functions. State of The Art Report (STAR) in Eurographics 2004, pp.70-93.

[MMSK03] MESETH J., MÜLLER G., SATTLER M., KLEIN R.: Btf rendering for virtual environments. In *Virtual Concepts 2003* (November 2003), pp. 356–363.

[Pho75] PHONG B. T.: Illumination for computer generated pictures. *Communications of the ACM 18*, 6 (1975), 311–317.

[Sch04] SCHNEIDER M.: Real-Time BTF Rendering. In *Proceeding of CESCG 2004* (2004).

[SvBLD03] SUYKENS F., VOM BERGE K., LAGAE A., DUTRÉ P.: Interactive Rendering of Bidirectional Texture Functions. In *Eurographics 2003* (September 2003), pp. 463–472.

[TZL02] TONG X., ZHANG J., LIU L., WANG X., GUO B., SHUM H.-Y.: Synthesis of bidirectional texture functions on arbitrary surfaces. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (2002), ACM Press, pp. 665–672.

[BonnDB] BTF DATABASE BONN: http://btf.cs.uni-bonn.de.

[ColDB] COLUMBIA-UTRECHT REFLECTANCE AND TEXTURE DATABASE: http://www1.cs.columbia.edu/cave/curet/.