WHITE PAPER

Photorealistic Imaging

Luc Claustres - CTO

Capturing the appearance of real-world materials and applying them to virtual prototypes in real-time

This white paper introduces the Bidirectional Texture Function (BTF) describing the appearance of real-world materials and the compression/decompression algorithms used to capture them for real-time prototyping.

Table of Contents

Executive Summary

Real-world materials are often very complex and significant color changes can appear across a surface due to small-scale features such as bumps, holes, etc. Models have been developped but are too much specific, hard to tune or require a prohibitive amount of processing power. As a consequence acquired data usually provide the best knowledge of a material appearance under the guise of a Bidirectional Texture Function (BTF).

This paper introduces an accurate but fast representation to overcome the main problem preventing BTF usage for real-time applications, which is storage. By fitting the reflectance data upon a physical reflection model and using an additional residual term we can accurately capture the fine details of a material. Then, we evaluate this representation through a dedicated pixel shader in order to reproduce real-world material appearance for real-time digital prototyping on consumer graphics hardware.

The challenges of real-time photorealism

The ultimate goal of computer-generated graphics is the creation of images that are indistinguishable from photographs. Today, the more realistic images are produced by offline renderers simulating the actual physics of light. However, this rendering technique known as *raytracing* often takes hours or days to compute a simple animation using powerful workstations or even render farms (many computers).

At the same time, last few years have seen the development of a competitive rendering technique known as *rasterization* thanks to dedicated accelerators available on mainstream computers. Indeed, a growing number of graphics applications need high quality (photorealistic) imaging in real-time, especially in the areas of design or ergonomic research, but also in the field of training. The main reason is to reduce the time-to-market through the use of interactive digital models, which is known as *virtual prototyping*. Another concern is to get huge savings as most physical models are no longer required.

Rendering is a fascinating yet complex process. On the one hand, computing interactions between light and objects (shadows, reflections, etc.) is the hardest part of the problem addressed by raytraced or raster graphics. This requires an amazing amount of processing power for real-time performances but the most recent advances in multi-core processors or graphics processing units (GPUs) provide one part of the answer. On the other hand, how we accurately capture the essence of a product (its color and texture), is probably the other part of the answer. In this paper we take a closer look at a main deficiency of today's real time graphics and present a solution for the identified problem of capturing real-world material appearance.

Handling the appearance of real-world materials

Real-world materials are never simple, and in certain cases, very complex. Let's examine for instance corduroy as a material. It has a surface appearance that dramatically changes with regard to the angle of lighting or the position of the observer. Small-scale features are playing a key role, producing a color variation across the surface, which usually characterizes the so called *texture* of the material.

The main obstacle to capture real-world materials is that underlying lighting phenomena, such as micro-scale shadowing and reflection effects, cannot be tackled easily from scratch. Models have been created according to a theoretical or empirical approach, but both lack generality. On the one hand physical models are very specific, complex, and require a prohibitive amount of processing power. On the other hand, empirical models are only suitable for a particular class of surfaces (metal, wood, etc.) and are really hard to tune according to the target material. This is precisely the reason why most rendered images look "plastic", no matter the lighting simulation accuracy, the lack of a complex surface behavior.

Instead of creating appearances from scratch, Orealia proceeds the other way around by extracting appearances from the real-world and bring it to the digital world. We believe this approach is more powerful and flexible as our customers do not need to know much about the material but simply needs to have a small sample of it.

Capturing the reflectance properties

The most basic notion of a surface's appearance is its color. The surface's color comes from the fraction of light of each wavelength that it reflects. So when discussing appearance, we will actually speak of reflectance, rather than color. The reflectance is physically described through the bidirectional texture function (BTF) as the ratio of outgoing to incoming light at each surface point. Some materials do not show visible patterns across the surface, and thus exhibit a homogeneous behavior (i.e. a plain color). In this particular case, we will speak of a bidirectional reflectance distribution function (BRDF) rather than a BTF, but this is a specific case of a more general issue.

The reflectance usually varies by the direction to the light and the direction from which it is viewed. For instance, most materials (from a sheet of paper to the ocean) are more reflective at grazing angles, which is known as the Fresnel effect (see Figure 1). These variations are produced by the interaction between the light and small-scale features of the material such as bumps, holes, scratches, etc. This leads to complex occlusion and inter-reflection phenomena that are prohibitive to compute on the fly or exceed the expressive power of current models. As a consequence, acquired data often provides the best knowledge of the reflectance of a real material.

Figure 1 : illustration of the Fresnel effect, reflected light increases at grazing angles

In order to correctly reproduce the look and feel of real-world materials, one needs to acquire their reflectance properties. We acquire them by measuring the BTF of the material by taking series of pictures for varying view and light directions, which are distributed approximately equally on the hemisphere. From this knowledge of the real-world material, we can compute a digital counterpart that encompasses the complex behavior of the surface. This contrasts with the traditional approach known as *texture mapping*, which apply a single shot image and fails in reproducing the changes of color with regard to the lighting/viewing configuration.

Equipment and technique

The traditional device for acquiring reflectance of real materials is the gonioreflectometer, a specialized device that positions a light source and a sensor relative to the material. These devices obtain a single sample (i.e. a color from which a BRDF can be computed) for each light–sensor position and are therefore relatively slow. Imaging devices such as CCD cameras sacrifice spectral resolution but obtain a large number of samples simultaneously (i.e. an image from which the BTF can be computed).

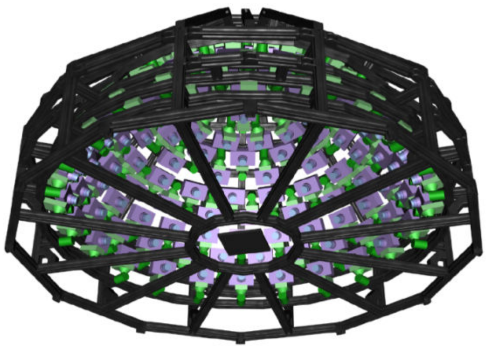

Figure 2 : a measurement dome used for BTF acquisition, the sample is at the center and varying viewing and lighting conditions are achieved through the activation of a different camera/light couple

From this set of images, we know how a material looks like for each possible lighting and viewing configuration. Since the number of pictures to be taken is very high in order to achieve an accurate sampling of the real reflectance behavior, an automatic setup acquires and post-processes these images. Post-processing includes rectification, color corrections and conversion to high dynamic range (HDR) format. A dense sampling of the entire BTF – between 300 and 8000 poses – can produce a set of reflectance images – typically on the order of 512 × 512 resolution – that measures up to 8 GB in size.

Bidirectional Texture Function (BTF)

The biggest problem introduced by BTFs is storage: high quality samplings of BTFs easily require several GBs of storage. In order to allow for efficient rendering of this data, a data compression scheme has been developed and employed. This compression technique especially permits real-time decompression while simultaneously achieving low approximation error.

Following the idea introduced by Mac Allister et al. [1] the BTF is viewed as a spatial-varying BRDF, i.e. the directional part of the BTF is represented by a reflection model while the spatial variation comes from different setup for each point of the surface. The reflectance values from the corresponding pixel of all images are used as input to a numerical solver to compute the best fit parameters at that pixel, of the specified physical reflection model. To enhance the result, we also approximate a residual function [2], i.e. the error between the reflection model and the real BTF, at each pixel. This allows to take self-shadowing and self-occlusion into account because it is not naturally handled by the BRDF model (Figure 3).

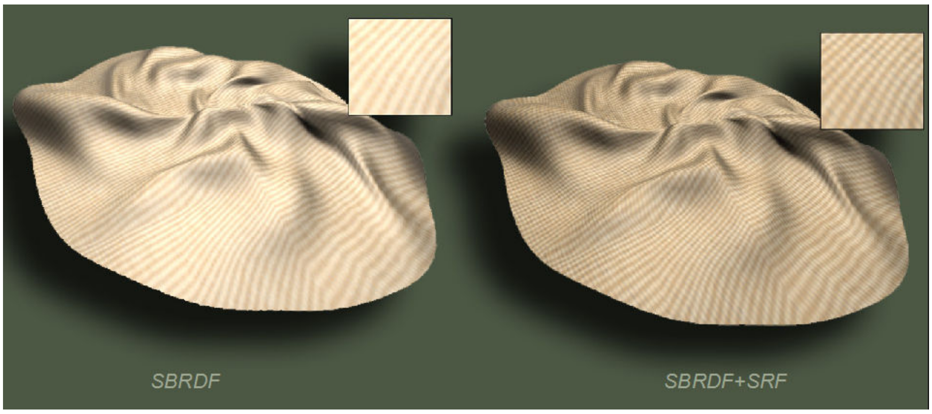

Figure 3 : this picture illustrates how the residual contributes to the realism of the material appearance in addition to the spatial-varying bidirectional texture function (SBRDF)

We use the efficient Lafortune's model [3] and Levenberg-Marquardt data fitting method [4] to achieve accurate results. The processing can take place in a dozen of minutes thanks to multi-threading for medium datasets (about 600 images), while the resulting data are less than 5 MB in size. These data are encoded as 4 texture files on account of the available texture-mapping and computational capabilities of GPUs. This task needs only to be performed once per acquired material, data is then used on demand by the visualization algorithm.

The real-time visualization pipeline

The past few years have seen impressive improvements in the capabilities of GPUs. Among many features, the most fascinating achievement is the realization of GPU programmability for real-time high quality imaging by directly evaluating physically-based reflection models as each pixel is shaded.

At rendering time, we compute the lighting and viewing direction for each pixel of the rendered image. Depending on the position of the pixel within the surface, we also select the appropriate parameters from our encoded data. Using mip-mapping when accessing the parameters stored in textures, we get filtering for free[1]. Then, we evaluate back the reflection model and the residual through a dedicated pixel shader to compute the final color of the pixel in real-time. Our acquisition, encoding and rendering pipeline is summarized Figure 4.

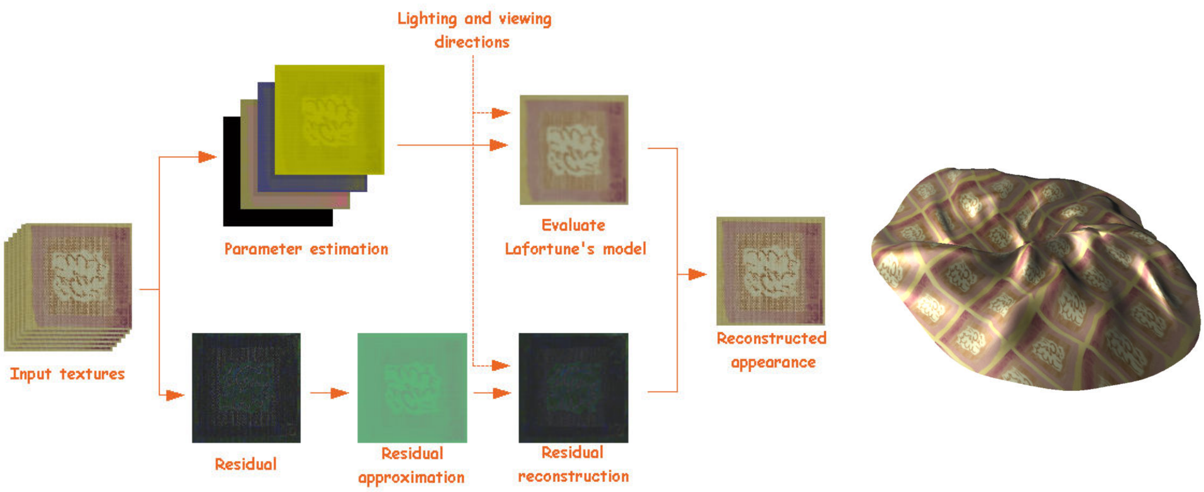

Figure 4 : the Orealia's BTF pipeline, data fitting over acquired pictures produces a compressed set of textures used by a pixel shader to visualize the virtual prototype in real-time

Results and performances

Figure 5 shows the results achieved on a set of different materials and how an accurate representation remains possible with regard to our dramastic compression rates (256:1). Indeed, the total resulting textures for all materials are less than 8 MB while input images measure 2 GB in size. Figure 6 illustrates the rendering speed for a simple 3D model (50,000 triangles) using a GeForce 7800 GTX. Such efficiency allows to use many materials at the same time without compromising real-time performances.

[1] This is not theoretically right as the Lafortune's model is non-linear, but it performs suprisingly well in our experiments.

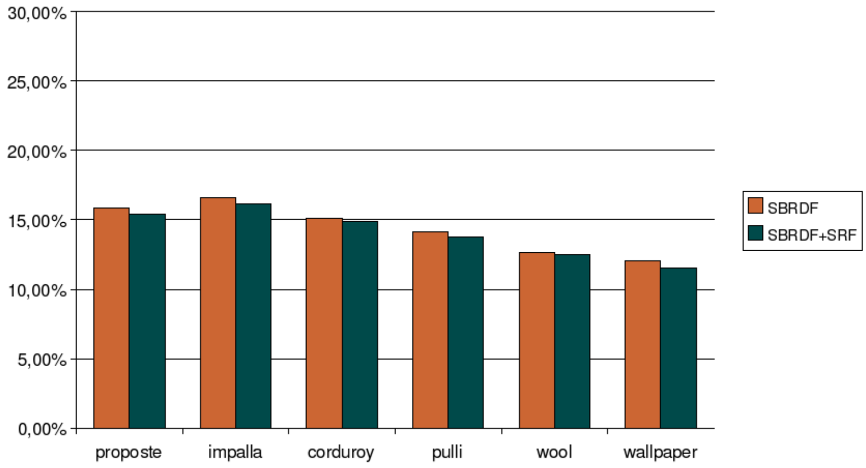

Figure 5 : total relative error achived on different materials with our BTF representation

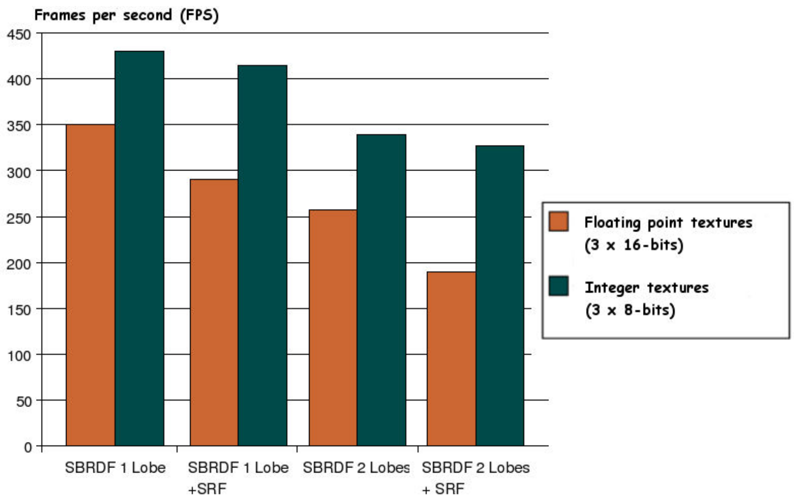

Figure 6 : performances achieved on a simple 3D object textured with one of our material depending on the reflection model complexity and the use of unsigned or floating point textures (GeForce 7800 GTX)

Convincing visual results are also achieved by our technique compared to a traditional texture mapping algorithm (see Figure 8 and Figure 7). Complex lighting effects such as anisotropy and self-shadowing are handled naturally on the velvet, while simple texture looks flat and is unable to produce plausible images. A very particular effect, which is paint flakes appearing at near distances and for specific viewing angles, can also be tackled with our BTF pipeline (see Figure 9). This effect is correctly filtered at larger distance and the paint looks smoother according to the behaviour of the real sample. Standard texture mapping simply fails to reproduce such a view and location dependent specular highlights and make its use impratical in this case.

Figure 7 : real-time rendering achieved with acquired velvet on the left seat, comparison with a traditional texture mapping algorithm on the right seat

Figure 8 : real-time rendering achieved with acquired velvet on the left bag, comparison with a traditional texture mapping algorithm on the right bag

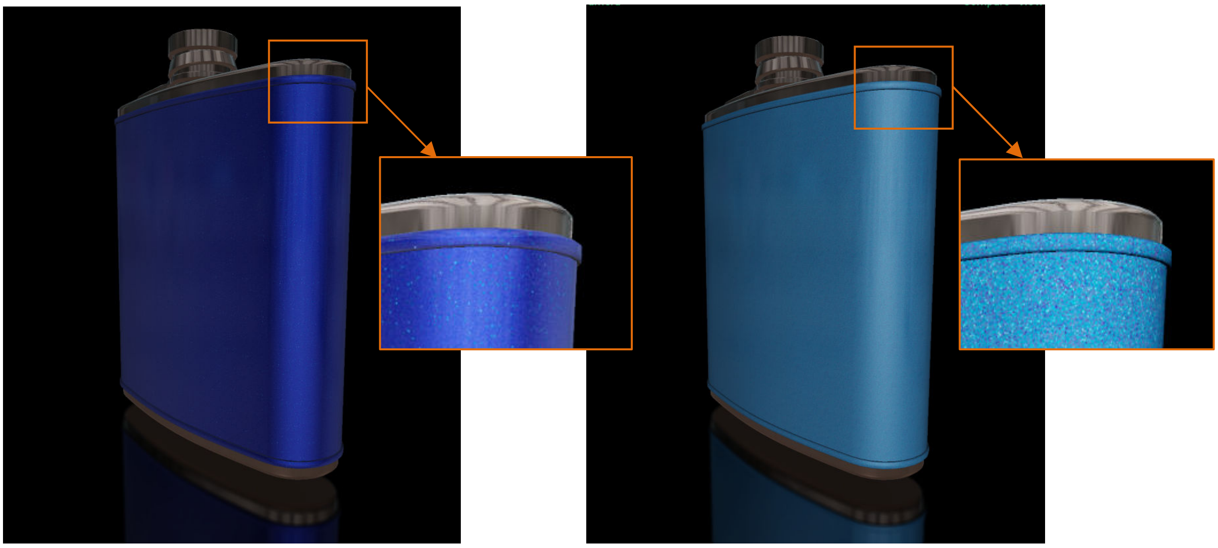

Figure 9 : real-time rendering achieved with acquired paint on the left bottle, comparison with a traditional texture mapping algorithm on the right bottle. When zooming in you can see the sparkle effect of the paint, which is correctly handled with our BTF representation.

Conclusion

We presented a complete digital pipeline to extract real-world material appearance and apply it to virtual prototypes in real-time. A wide range of complex lighting effects such as anisotropy and self shadowing can be tackled compared to traditional texture mapping. Filtering is also naturally handled in order to reduce aliasing and sparkling effect at large distance. At last, an efficient compression scheme, as well as a GPU-based reconstruction, allows usage for industrial design even in a large scene composed of many different surfaces.

Rendering objects with realistic materials puts some restrictions on the rendered objects. One of them is that they contain a valid, consistent parameterization that minimizes texture stretch and might fulfill additional requirements like orientation and placement constraints. By the way, within the next version of Orealia, algorithms for computing parameterization are as automatic as possible, making application of real-materials to virtual prototypes as simple as possible.

About Onesia

Founded in 2004, Onesia provides interactive real-time 3D rendering software solutions and services for the professional designer, at an affordable price, without compromising performance or quality.

Onesia provides architects, engineers, designers, computer graphics artists and hobbyists with the tools they need to easily review and present photo-realistic 3D models in their real-life environment.

Onesia enables production and marketing departments to speed up product conception, design, and ultimately the decision-making process by evaluating several scenarios in real-time. By communicating ideas and concepts through this powerful real-time 3D rendering software, companies can now cut prototyping costs and time in half and bring their products to market faster.

Onesia Inc. is headquartered in Los Gatos, Silicon Valley. To learn more about Onesia, visit www.orealia.com or call 408.313.8492.

Acknowledgements

This work is based on a technological tranfer with the Institut de Recherche en Informatique de Toulouse (IRIT) – Université Paul Sabatier (UPS) - Centre National de la Recherche Scientifique (CNRS), Toulouse, FRANCE.

This work used public BTF data provided by the Computer Graphics Lab at Bonn University.

References

[1] David K. McAllister, Anselmo Lastra, and Wolfgang Heidrich. Efficient rendering of spatial bi-directional reflectance distribution functions. In *HWWS '02 : Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 79–88. Eurographics Association, 2002.

[2] Wan-Chun Ma, Sung-Hsiang Chao, Bing-Yu Chen, Chun-Fa Chang, Ming Ouhyoung, and Tomoyuki Nishita. An efficient representation of complex materials for real-time rendering. In *VRST '04 : Proceedings of the ACM symposium on Virtual reality software and technology*, pages 150–153, New York, NY, USA, 2004. ACM Press.

[3] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-linear approximation of reflectance functions. In *SIGGRAPH '97 : Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 117–126. ACM Press/Addison-Wesley Publishing Co., 1997.

[4] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, 1992.