Real-Time Rendering of Rough Refraction

Charles de Rousiers, Adrien Bousseau, Kartic Subr, Nicolas Holzschuch, Ravi Ramamoorthi

Real-Time Rendering of Rough Refraction

Charles de Rousiers, Adrien Bousseau, Kartic Subr, Nicolas Holzschuch and Ravi Ramamoorthi

Abstract—We present an algorithm to render objects made of *transparent materials with rough surfaces* in real-time, under all-frequency distant illumination. Rough surfaces cause wide scattering as light enters and exits objects, which significantly complicates the rendering of such materials. We present two contributions to approximate the successive scattering events at interfaces, due to rough refraction: First, an approximation of the Bidirectional Transmittance Distribution Function (BTDF), using spherical Gaussians, suitable for real-time estimation of environment lighting using pre-convolution; second, a combination of cone tracing and macro-geometry filtering to efficiently integrate the scattered rays at the exiting interface of the object. We demonstrate the quality of our approximation by comparison against stochastic ray-tracing. Furthermore we propose two extensions to our method for supporting spatially varying roughness on object surfaces and local lighting for thin objects.

Index Terms—Real-time Rendering, Translucent Material, Bidirectional Transmittance Distribution Function, Normal Distribution Function.

1 Introduction

We introduce a real-time algorithm to render translucency due to rough surfaces such as frosted and misted glass. Translucent appearance can be attributed to a combination of two factors: Surface scattering, for example in frosted glass due to its surface roughness, or volume scattering in participating media. In this paper we focus on the former and assume that the media are non-scattering. In the remainder of this paper we refer to such materials as being *rough-refractive*.

In contrast to transparent materials with specular surfaces handled by existing methods [1], [2], transparent materials with rough surfaces present the unique challenge that light is scattered in multiple directions as it enters and exits the object. The first consequence of these scattering events is that a light ray entering at a single point of the surface will exit over an area that depends on the scattering properties of the material and the geometry of the object. The second consequence of scattering in rough translucent objects is that the lighting simulation involves numerical integration of light over multiple directions.

Much of the computation performed by brute-force methods might be misdirected, since human vision has not evolved to detect refractive effects accurately [3], [4]. However, we are sensitive to variation in blur [5] introduced by surface roughness. Our approximations exploit these two observations to enable plausible rendering of rough-refractive materials in real-time.

First, we observe that the scattering function used in previous work on rendering of rough transparent materials [6] can be well approximated by spherical Gaussians. Consequently, we replace stochastic integration of lighting with a pre-integration, in the spirit of existing work on pre-convolution of environment maps for real-time glossy reflections [7], [8].

Our second contribution is a combination of cone tracing [9] and geometry filtering [10] to approximate the exiting rays due to scattering of a single incident ray through a homogeneous material. At points where scattered rays exit the object, we construct a distribution of normals to approximate the local geometry. We then convolve this resulting normal distribution function (NDF) with the material BTDF (Bidirectional Transmittance Distribution Function) to obtain a new scattering function. We then query the pre-convolved environment map using this scattering function.

Finally, we present two extensions to support spatially varying BTDF (SV-BTDF) and local lighting of thin geometry due to spherical light sources. To summarize, our paper makes the following contributions:
- we describe a method to render rough-refractive objects under all-frequency distant illumination (see Fig. 19), in real time.
- we introduce a new spherical Gaussian approximation of a microfacet BTDF to extend pre-convolution based methods to refractive materials.
- we propose an innovative formulation for approximating sequential refractions as convolutions involving the BTDF and the NDF at the exiting surface.
- we extend our method to support spatially varying roughness.
- we propose a local lighting model for thin objects.

The first three contributions were also published in [11], while the last two are new extensions from our previous work.

- *INRIA Rhone-Alpes, France.*
- *INRIA Sophia-Antipolis, France.*
- *Laboratoire Jean-Kuntzmann, Grenoble, France.*
- *University College London, UK.*
- *University of California, Berkeley, CA, USA.*

$\mathbf{x}$	Point
ω_x	Normalized direction
$\vec{\omega_x}$	Un-normalized direction
ω_o	Outgoing direction
ω_i	Incoming direction
$\mathbf{n}$	Normal
ω_{h_r}	Reflected half vector
ωh_t	Transmitted half vector
η_i	Index of refraction of the incident media
η_o	Index of refraction of the refracting media
f_r	BRDF (*Bidirectional Reflectance Distribution Function*)
f_t	BTDF (*Bidirectional Transmittance Distribution Function*)
$\mathbf{p}$	Intersection point on the front interface
$\mathbf{s}$	Intersection point on the back interface
S	Surface of integration on the back interface
D	NDF(*Normal Distribution Function*)
F	Fresnel function
G	Geometric function for microfacet
Ω^+	Upper hemisphere
Ω^-	Lower hemisphere
$L(\omega_o)$	Outgoing radiance
$L(\omega_i)$	Incoming radiance
$\cdot$	Dot product
$\chi^+(a)$	Sign function (1 if $a \geq 0$ and 0 if $a < 0$)

TABLE 1: *Notations used in this paper.*

2 Related work

We review work on real time rendering of both reflection and refraction as well as related work in off-line rendering of refraction at rough surfaces.

Real-time reflection of distant illumination: in the case of glossy reflections, Kautz et al. [7] and Ramamoorthi and Hanrahan [8] have shown that for radially symmetric BRDFs, the contribution of distant incident illumination can be pre-integrated by means of a convolution of the environment map with the BRDFs. We extend these pre-convolution based methods to transparent materials with non-specular BTDFs.

Real-time refractions: most existing work on real-time refractions focus on the special case of specular BTDFs, for which a single light ray travels through the object at each pixel [1], [2]. Recent methods on specular refractions handle complex light transport scenarios in real time, including non-homogeneous media and caustics [12]–[14]. While Heidrich et al. [15] and Eisemann and Décoret [16] address the rendering of rough refraction in real-time using pre-convolved environment maps, they approximate light transport inside the object with a single light ray. In contrast, our method relies on cone-tracing [9] to efficiently account for multiple refracted rays inside the object.

BTDF models : just as microfacet BRDFs model rough reflections, a microfacet BTDF for rough refractions has been derived by Walter et al. [6]. This model is expressed as (see notation in Table 1):

$$f_t(\omega_{h_t}, \mathbf{n}) = \frac{|\omega_i \cdot \omega_{h_t}||\omega_o \cdot \omega_{h_t}|}{|\omega_i \cdot \mathbf{n}||\omega_o \cdot \mathbf{n}|} \frac{\eta_o^2(1 - F(\omega_o, \mathbf{n}))D(\omega_{h_t}, \mathbf{n})G(\omega_o, \omega_i, \omega_{h_t})}{(\eta_i(\omega_i \cdot \omega_{h_t}) + \eta_o(\omega_o \cdot \omega_{h_t}))^2} \quad (1)$$

where η_i and η_o are the indices of refraction (IOR) of the incident and refracting media respectively, D is the normal distribution function (NDF) and G is the geometric term that models effects such as masking and shadowing. Several choices are possible for the NDF D, such as the Beckmann and GGX [6] distribution. We use the Blinn-Phong model for its simplicity. The Blinn-Phong NDF is parametrized with κ_γ^D, the inverse roughness (high values of κ_γ^D yield sharp refractions) :

$$D(\omega_{h_t}, \mathbf{n}) = \chi^+(\mathbf{n} \cdot \omega_{h_t}) \frac{\kappa_\gamma^D + 2}{2\pi} (\mathbf{n} \cdot \omega_{h_t})^{\kappa_\gamma^D} \quad (2)$$

This microfacet BTDF model is expressed according to the refractive half vector ω_{h_t}, equivalent of the half vector for reflective models ω_{h_r} [17]. The refractive half vector is defined as :

$$\omega_{h_t} = \frac{\vec{\omega_{h_t}}}{||\vec{\omega_{h_t}}||} = -\frac{\eta_o\omega_o + \eta_i\omega_i}{||\eta_o\omega_o + \eta_i\omega_i||} \quad (3)$$

When ω_{h_t} and the surface's normal $\mathbf{n}$ are collinear, ω_i and ω_o follow Snell's law:

$$\eta_i \sin\theta_i = \eta_o \sin\theta_o \quad (4)$$

Dai et al. [18] propose a similar model for the particular case of thin surfaces, for which light transport in the object can be omitted. While these microfacet models can easily be integrated in a raytracer, the large number of samples required to handle rough materials is prohibitive for real-time applications. In this paper we identify that Walter's BTDF model is well approximated by a spherical Gaussian function, which allows us to pre-convolve environment maps for real-time rendering.

3 Theory

The exiting radiance $L(\mathbf{p}, \omega_o)$ at a point $\mathbf{p}$ on the surface of a transparent material, along the direction ω_o, is the integral of the incoming radiance weighted by the reflectance function. For clarity, we separate the full scattering function into a BRDF (f_r), expressing scattering effects over the upper hemisphere (Ω^+), and a BTDF (f_t), expressing scattering effects over the lower hemisphere (Ω^-). Both hemispheres, $\Omega^+$and $\Omega^-$are defined with respect to the tangent plane at $\mathbf{p}$.

$$L(\mathbf{p}, \omega_o) = \int_{\Omega^+} L(\mathbf{p}, \omega_i)\, f_r(\omega_o, \omega_i)\, \mathrm{d}\omega_i + \int_{\Omega^-} L(\mathbf{p}, \omega_i)\, f_t(\omega_i, \omega_o)\, \mathrm{d}\omega_i \quad (5)$$

For specular refraction, the BTDF f_t is simply a Dirac along the direction given by Snell's law of refraction (see Figure 1a). For rough surfaces, the micro-geometry around point $\mathbf{p}$ defines the distribution f_t(see Figure 1b).

Under our two-interface assumption, the exiting radiance at $\mathbf{p}$ is the result of successive scattering processes at two

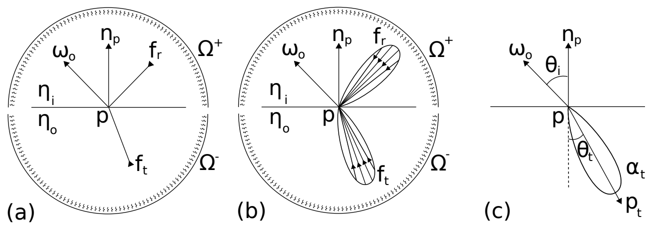


Fig. 1: *BSDF and notation. (a) A specular BRDF f_r and a specular BTDF f_t, (b) A glossy BRDF f_r and a rough BTDF f_t, (c) Spherical Gaussian parametrization used for approximating the lobe of the refractive microfacet model.*

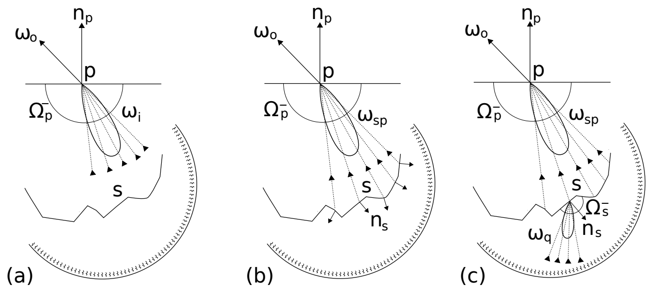


Fig. 2: *Refraction at two interfaces. (a) Integration of the BTDF at the front interface. (b) Stochastic integration of incident directions over the BTDF lobe at* **p**. *(c) For each sampled direction in (b), integration over the BTDF lobe at* **s** *along which radiance is queried from the environment map.*

boundaries (see Figure 2). That is, ignoring the BRDF in Equation 5 and accounting for refraction at only two interfaces — entry and exit — on a homogeneous rough object, the outgoing radiance at an entry point **p** can be expressed as an integral over the exiting surface S

$$L(\mathbf{p},\omega_o) = \int_S L(\mathbf{p},\omega_{\mathbf{sp}})\ f_t(\omega_{\mathbf{sp}},\omega_o)\ J(\mathbf{n_p},\mathbf{n_s},\mathbf{p},\mathbf{s})\ \mathrm{ds} \quad (6)$$

where $\omega_{\mathbf{sp}} = \vec{\mathbf{sp}}/||\vec{\mathbf{sp}}||$, $\mathbf{n_p}$ and $\mathbf{n_s}$ are normals at **p** and **s** respectively, $J(\mathbf{n_p},\mathbf{n_s},\mathbf{p},\mathbf{s})$ is the Jacobian associated with the change of variables from solid angle to the surface S and

$$L(\mathbf{p},\omega_{\mathbf{sp}}) = \int_{\Omega_{\mathbf{s}}^+} L(\mathbf{s},\omega_q)\ f_t(\omega_q,\omega_{\mathbf{ps}})\ \mathrm{d}\omega_q. \quad (7)$$

3.1 Refraction using convolution

The primary challenge in real-time rendering of reflections and refractions for rough surfaces is rapid evaluation of the integral over $\Omega^+$and Ω^-, which corresponds to a convolution of the incident illumination with a BSDF [19]. Assuming distant lighting, and ignoring visibility, existing methods for real-time rendering of glossy reflections pre-compute the integral over $\Omega^+$by convolving the environment map with the BRDF f_r. This pre-convolution is independant of incident direction for radially symmetric

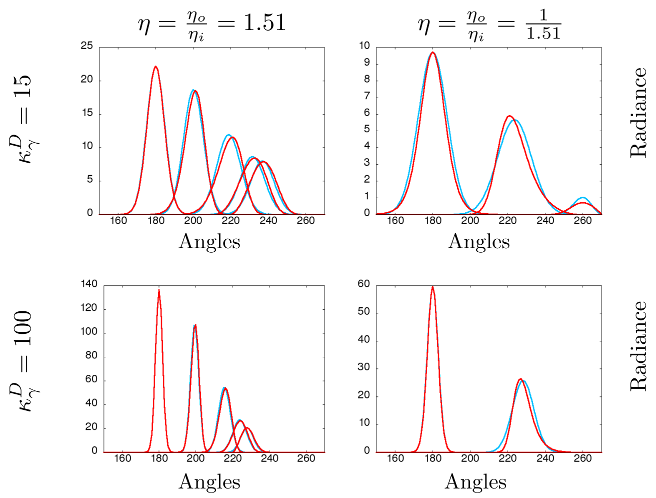


Fig. 3: *Comparison between the actual BTDF model (red) and our Gaussian approximation (blue) for different incident angles (0,30,60,80 and 88 degrees), for two different values of roughness κ_γ^D. We show both values for light entering the material ($\eta = 1.51$) and for light leaving the material ($\eta = \frac{1}{1.51}$). At the exiting interface, total internal reflection can happen, resulting in a smaller number of curves.*

BRDFs and the integral is evaluated at run time with a single look-up in the environment map in the direction of specular reflection.

We extend this pre-convolution approach to the case of refraction and pre-compute the integral over Ω^-as a convolution of an environment map with the BTDF f_t. However, the pre-convolution approach requires a radially symmetric representation of the BTDF, which is not the case for the BTDF model of Walter et al. [6]. We observe however that the BTDF response for a given incident angle and index of refraction is radially symmetric and *can be well approximated by a spherical Gaussian*, as illustrated in Figure 3. We express this spherical Gaussian representation of the BTDF as:

$$f_t(\omega_o,\omega_i) = \alpha_t e^{\kappa_t(\mathbf{p}_t\cdot\omega_i - 1)} \quad (8)$$

where α_t is a scaling factor controlling the amplitude of the Gaussian lobe, κ_t is an exponent controlling the width of the lobe, and $\mathbf{p}_t$ is the principal direction of the lobe (see Figure 1). These parameters are fitted for each outgoing angle θ_o, each index of refraction η and each BTDF exponent κ_γ^D and stored into a table T :

$$T(\theta_o,\eta,\kappa_\gamma^D) \rightarrow (\alpha_t,\kappa_t,\mathbf{p}_t) \quad (9)$$

For a given index of refraction η, we precompute the functions corresponding to the two interfaces: $T(\theta_o,\eta,\kappa_\gamma^D)$ and $T(\theta_o,1/\eta,\kappa_\gamma^D)$ and store them in 2D textures indexed by θ_i and κ_γ^D.

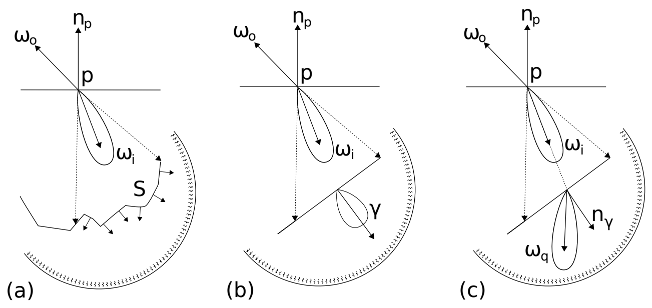


Fig. 4: *Approximated transport. (a) Transport from the front to the back interface with the actual geometry. (b) The geometry on the back interface is replaced by a single lobe NDF (mean + variance) (c) A new BTDF is derived from the NDF to include the local geometry.*

3.2 Light transport as geometry filtering

We now focus on light transport inside the object. As described by Equation 6 and Figure 2, light enters the transparent object at the front interface, is scattered by the BTDF and transported through the object, before finally exting at the back interface, where it is again scattered by the BTDF. Our goal is to find a good approximation of equation 6 for achieving real-time performance.

Our key idea is to replace the transport problem with a filtering problem (Figure 4): we extend the work of [10] and represent the geometry intersected by the scattered rays at the front interface as a flat proxy enriched with a Normal Distribution Function (NDF). We then convolve this NDF with the BTDF to obtain a new BTDF that models the effect of both the transmittance function and the aggregated exiting geometry.

This approach is valid under *three assumptions*: (1) that all rays are parallel as they reach the exiting interface; (2) that the shape of the BTDF lobe is independent of the incident angle; and (3) that a single lobe NDF will be sufficient to represent the underlying geometry. While refraction is an important phenomenon, humans have difficulty perceiving or assessing the accuracy of results. Consequently, our model leads to plausible results with real-time frame rates despite the above assumptions. Our results, in Section 6, demonstrate the efficacy of our model.

To compute the parameters of the resulting BTDF, we convolve the NDF of the geometry γ (with an exponent κ_γ^G) with the original NDF of the BTDF, D (with an exponent κ_γ^D). The result is a new NDF, D', whose exponent $\kappa_\gamma^{D'}$ is defined as:

$$\kappa_\gamma^{D'} = \frac{\kappa_\gamma^G \, \kappa_\gamma^D}{\kappa_\gamma^G + \kappa_\gamma^D} \tag{10}$$

As shown in Figure 4c, we orient the proxy surface perpendicular to the average normal n_γ, and use it to refract the direction of the central lobe $\mathbf{p}_t$ of the front interface using the new BTDF.

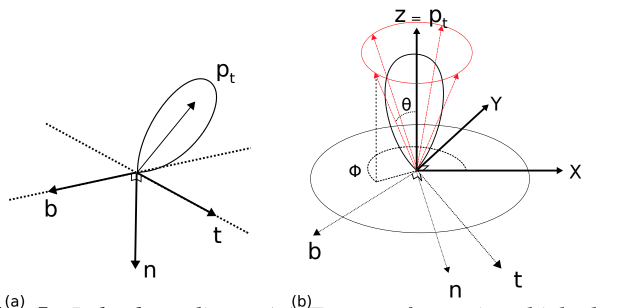


Fig. 5: *Lobe bounding : (a) Tangent frame in which the BTDF is expressed. (b) A tangent frame is built around the main axis of the BTDF, and 4 rays generated in spherical coordinates by using the bounding angle θ_b computed by equation 12.*

4 Algorithm

Figure 6 describes the main steps of our algorithm, that we implement using three rendering passes. The first two passes compute geometric information about the scene, such as the NDF at the back interface at multiple scales and the range of depth covered by the back faces of the mesh. The third pass uses this information to approximate the transport of refracted rays inside the object with cone tracing. We finally convolve the NDF of the intersected geometry at the back interface with the BTDF and evaluate the illumination by a lookup in the pre-convolved environment map by the resulting distribution.

We describe here the pre-computation and the three rendering passes of our algorithm in more detail.

4.1 Pre-computation

Gaussian BTDF : We pre-compute the parameters θ_t, α_t,κ_t of the Gaussian BTDF model for a regular sampling of incident angle θ_i, index of refraction η and exponent of the BTDF κ_γ. We fit the BTDF model by sampling, using the importance sampling scheme given in [6] and then compute the scaling, mean and variance of the distribution.

While the resulting table could be stored in a 3D texture, a few 2D textures parameterized by θ_i and κ_γ are often sufficient to store the parameters for the most common indices of refraction.

Environment Map : Similar to previous pre-convolution based methods [7], we store the pre-convolved environment map as a MIP-map of 2D textures, where MIP-map levels contain the environment map convolved with spherical Gaussian functions of decreasing exponent κ.

4.2 First and Second Pass : Geometry filtering

In the first pass of our algorithm we render the normals and depth of back-facing geometry and store the result in off-screen buffers. We render the farthest back faces by setting up the depth test function to `GL_GREATER` in

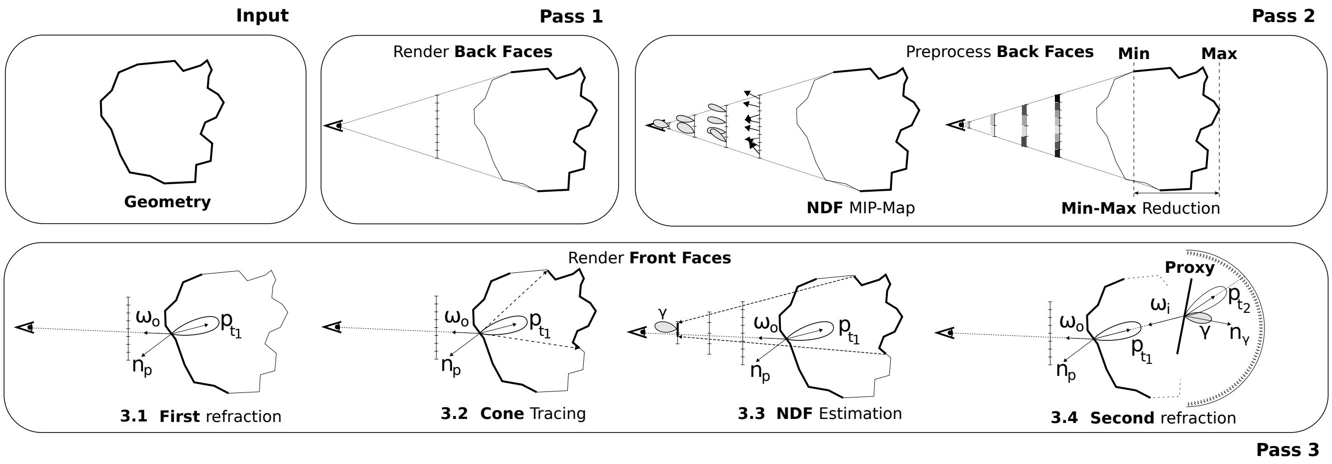


Fig. 6: *Overview of our algorithm. (**Pass 1**) During the first pass, the depths and the normals of back faces of the mesh are rendered into two separate buffers. (**Pass 2**) Based on the normal buffer, a NDF buffer is built, by estimating for each texel the mean direction and the variance. The min and max value of the depth buffer are also extracted. (**Pass 3**) During the last pass, the front faces of the mesh are rasterized. For each fragment, the parameters of the lobe at the front interface are extracted, and four rays are cast by ray marching in the depth texture for estimating its footprint. The NDF corresponding to this footprint is retrieved from the NDF buffer and is used as a proxy surface for the second refraction. The final radiance value is looked up into the pre-convolved environment map.*

OpenGL, as in [2]. In a second pass we recursively downsample the back-face normal buffer to compute NDFs of the back interface at multiple scales. We use Toksvig's method [20] to compute the downsampled normal buffer. In this method, the main direction of each NDF in the downsampled buffer is computed as the mean direction of the underlying normals. The variance σ^2 of the NDF is expressed as:

$$\kappa_\gamma^G = \frac{2}{\sigma^2} = \frac{2\,(1 - ||\vec{\mathbf{n}}||)}{||\vec{\mathbf{n}}||} \tag{11}$$

with $\vec{\mathbf{n}}$ the main direction of the NDF.

The normalized main direction $\vec{\mathbf{n}}/||\vec{\mathbf{n}}||$ is stored in the `xyz` components and the norm $||\vec{\mathbf{n}}||$ in the `w` component in order to interpolate them independently. This way, the variance of the NDF is smoothly interpolated unlike Toksvig's original method where variance can change rapidly between two texels if the NDF directions are very different.

We also compute the **min** and **max** value of the back-face depth buffer during the second pass using MIP-map reduction (For more details see [2]). The depth buffer and its **min** and **max** value will then be used to trace rays inside the object using the ray marching algorithm of Oliveira and Brauwers [2].

4.3 Third Pass : Estimating macro-geometry NDF

In the third pass, we render the front facing geometry and evaluate the final radiance at each pixel. To do so, we first compute the refraction at the front interface, and then trace a cone bounding the refracted rays inside the object. We finally use the NDF buffer to retrieve the NDF covered by the intersection of the cone with the back interface (pass 3 in Figure 6).

Refraction at the front interface: For each pixel, we use the incident angle θ_i relative to the local normal $\mathbf{n}$ and the BTDF exponent κ_γ^D to read in the pre-computed texture values of α_{t1} κ_{t1} and $\mathbf{p}_{t1}$ for the spherical Gaussian BTDF model.

Approximate cone tracing: We build a tangent frame where the Z axis is oriented along the spherical Gaussian axis $\mathbf{p}_{t1}$, as shown in figure 5b. We use this local frame to generate four rays bounding the spherical Gaussian lobe (Figure 5). We follow [21] and set the orientation θ of the rays in respect to the Z axis such that the integral of the bounded lobe represents 75% of its total integral.

$$\theta_b = \arccos((1 - 0.75)^{\frac{1}{\kappa_t + 1}}) \tag{12}$$

We use the ray marching algorithm of Oliveira and Brauwers [2] to trace the four bounding rays inside the object. The four intersection points suggest the extent of the footprint of the refracted rays that intersect the back interface.

NDF estimation: The screen positions of the four intersection points define an axis-aligned bounding box of the pixels in the geometry buffers that are intersected by the refracted rays (pass 3.2 and 3.3 in Figure 6). We estimate the local texture coordinate derivatives ($\frac{\partial s}{\partial x}$ and $\frac{\partial t}{\partial y}$) from this bounding box and use the OpenGL `textureGrad` function to look up the NDF value with an anisotropic sampler in the NDF buffer. The NDF texture has to be set up with anisotropic sampling for non-square bounding boxes.

Radiance computation: We use the NDF as a proxy surface for the second refraction (pass 3.4 in Figure 6). The angle θ_o between the central direction of the lobe on the front interface $\mathbf{p}_{t1}$ and the NDF normal and the new exponent $\kappa_\gamma^{D'}$ are used to look up into the parameter texture

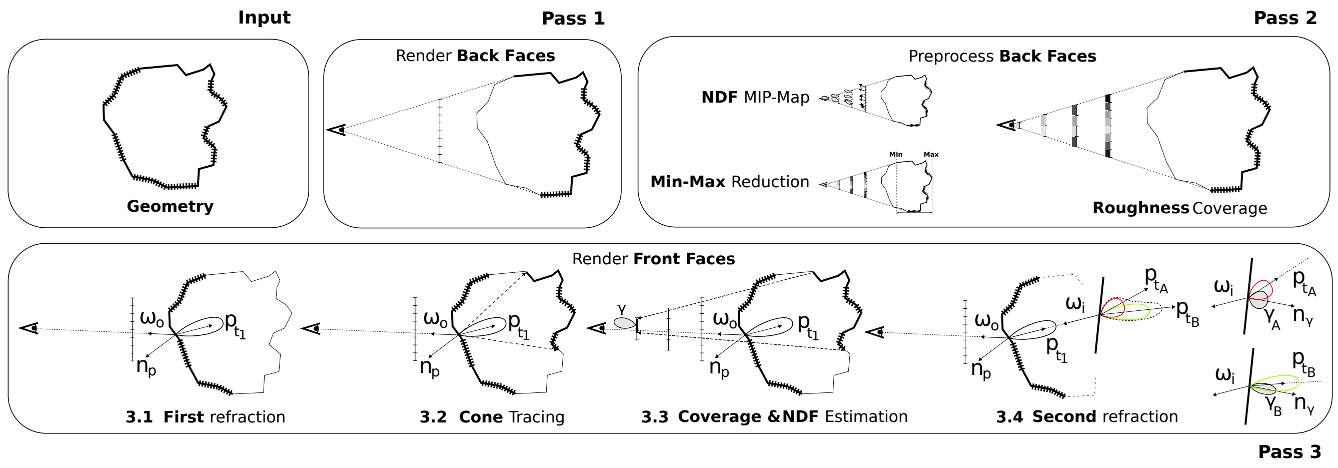


Fig. 7: *Our algorithm for spatially varying roughness. (**Pass 1**) During the first pass, the depths, the normals and the coverage ratio of back faces of the mesh are rendered into three separate buffers. (**Pass 2**) Based on the normal buffer, an NDF buffer is built, by estimating for each texel the mean direction and the variance. The min and max value of the depth buffer are also extracted and the coverage ratio are computed for each roughness at different scales. (**Pass 3**) During the last pass, the front faces of the mesh are rasterized. For each fragment, the parameters of the lobe at the front interface are extracted, and four rays are cast by ray marching in the depth texture for estimating its footprint. The NDF and the coverage ratios corresponding to this footprint are retrieved from the NDF and coverage buffers. They are used to compute the second refraction for each roughness. The final radiance is computed as the sum of the radiance for each roughness weighted by their coverage ratios.*

for retrieving the BTDF parameters α_{t2}, κ_{t2} and $\mathbf{p}_{t2}$. Finally, those 3 parameters are used to look up the value of the incoming radiance in the pre-convolved environment map.

Total internal reflection: Correctly handling total internal reflection for objects with rough boundaries is expensive since it requires explicit tracing of individual rays. To avoid this, we adopt the popular solution introduced by Wyman [1] and clamp the incident ray to the critical angle, at the upper bound, when the relative index of refraction is less than 1.

5 Extensions

We derived our real-time rendering algorithm assuming uniform roughness and distant lighting. We now describe how to extend our method to handle spatially varying roughness and local lighting.

5.1 Spatially-varying roughness

So far, the roughness was assumed to be constant over the surface of the object and the BTDF was only expressed as a function of the incoming and outgoing vectors $f_t(\omega_i, \omega_o)$. To represent materials for which roughness might vary over the surface, we introduce the roughness function $R(\mathbf{x})$ that depicts the roughness at point $\mathbf{x}$. We represent this roughness function with a texture map in our implementation. The resulting BTDF function becomes $f_t(\omega_i, \omega_o, R(\mathbf{x}))$ (Fig. 8).

We propose two solutions to render such materials: A simple approach that proceeds exactly as for the homogeneous case, except that the coefficient for local roughness at the front interface is obtained by looking up the roughness texture; a more detailed approach that also accounts for varying roughness at the back interface (Fig. 8).

We can rewrite equation 6 using the roughness function R, (here, J is a simplified notation for the Jacobian) :

$$L(\mathbf{p}, \omega_o) \quad = \quad \int_S L(\mathbf{p}, \omega_{\mathbf{sp}})\ f_t(\omega_{\mathbf{sp}}, \omega_o, R(\mathbf{s}))\ J\ \mathrm{ds}$$

Assuming that the surface is made of a small set of different roughnesses, we approximate the rendering equation as a discrete weighted sum :

$$\begin{aligned} L(\mathbf{p}, \omega_o) \quad = \quad & c_0 \int_S L(\mathbf{p}, \omega_{\mathbf{sp}})\ f_t(\omega_{\mathbf{sp}}, \omega_o, R_0)\ J\ \mathrm{ds} \\ + \quad & c_1 \int_S L(\mathbf{p}, \omega_{\mathbf{sp}})\ f_t(\omega_{\mathbf{sp}}, \omega_o, R_1)\ J\ \mathrm{ds} \\ + \quad & c_2 \int_S L(\mathbf{p}, \omega_{\mathbf{sp}})\ f_t(\omega_{\mathbf{sp}}, \omega_o, R_2)\ J\ \mathrm{ds} \\ + \quad & \ldots \end{aligned}$$

where c_i represents the ratio of the exiting area covered by the roughness R_i.

This decomposition allows the use of previous transport approximations for each roughness independantly and to accumulate their contributions.We assume that the roughness function R is *piecewise constant* and is *not correlated with the lighting*.

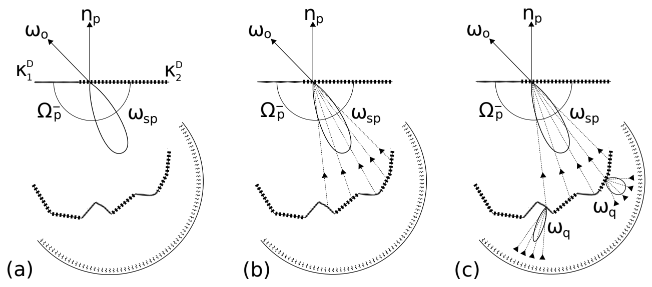


Fig. 8: *Refraction at two interfaces with a spatially varying BTDF (κ_1^D and κ_2^D represent different roughness). (a) A first intersection the underlying roughness determines the BTDF lobe shape. (b) Stochastic integration of incident directions over the BTDF lobe. (c) For each sampled direction in (b) integration over the BTDF lobe is done according to the underlying roughness.*

Figure 7 gives an overview of our pipeline to handle spacially varying roughness. In the first pass, we rasterize the back faces of the geometry and store the roughness $R(\mathbf{x})$ in addition to normals and depths. During the second pass, we compute the ratio c_i of each roughness at different scales and store them in a MIP-map pyramid (pass 2 in Figure 7). The third pass generates the final rendering by rasterizing the front faces. We evaluate the roughness $R(\mathbf{x})$ at the front interface by a lookup in the roughness texture map. We evaluate the radiance due to each roughness R_i at the back interface and compute the final radiance as the sum of these radiance values weighted by their coverage ratio c_i (pass 3.4 in Figure 7).

5.2 Local light sources for thin objects

Our geometry filtering approach assumes that the lighting is distant and as such is purely directional. However, when dealing with local lighting, the incident light direction varies according to the position of the surface point with respect to the light. Aggregating the surface normals in a single NDF would ignore the relative position of the surface points and lead to incorrect lighting. In this section we assume instead that the object is thin so that the exit footprint of the BTDF lobe is well approximated by a single point. With this thin geometry assumption we can express the radiance computation as a single integral :

$$L(\mathbf{p},\omega_o) = \int_{\Omega^+} f_t(\omega_i,\omega_o) L(\mathbf{s},\omega_i)\,\mathrm{d}\omega_i \tag{13}$$

where s is the center of the lobe footprint at the back interface.

In the case of distant lighting, this integral is precomputed with environment map convolution. For local lighting, we rely on the analytical computation of the inner product of two spherical Gaussian functions. We follow Wang et al. [21] and represent a spherical light source as a spherical Gaussian in the reference frame where we need to integrate its contribution (Fig. 16)

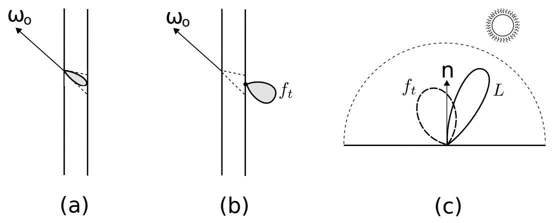


Fig. 9: *(a) Footprint of the first lobe on the opposite surface for thin geometry. (b) Point approximation of the foot print for thin geometry. (c) Lighting integration for local light source. The local light source is approximated by a spherical Gaussian function.*

$$(\alpha_l,\kappa_l,p_l) = \left(\frac{I}{||\mathbf{p}_x-\mathbf{p}_l||^2}, \frac{2\,\pi\,r^2}{||\mathbf{p}_x-\mathbf{p}_l||^2}, \frac{\mathbf{p}_x-\mathbf{p}_l}{||\mathbf{p}_x-\mathbf{p}_l||}\right) \tag{14}$$

The local lighting integration is then expressed as

$$\int_{\Omega^+} f_t(\omega_i,\omega_o) L(\omega_i)\,\mathrm{d}\omega_i = \frac{4\pi\alpha_t\alpha_l}{e^{\kappa_t+\kappa_l}}\frac{\sinh d_m}{d_m} \tag{15}$$

with $d_m = ||\kappa_t\mathbf{p}_t + \kappa_l\mathbf{p}_l||$.

This representation allows us to handle point and spherical light sources. While it is still possible to approximate other types of light sources with a set of spherical light sources, it might be interesting to study more complex light source shapes in the future.

6 Results

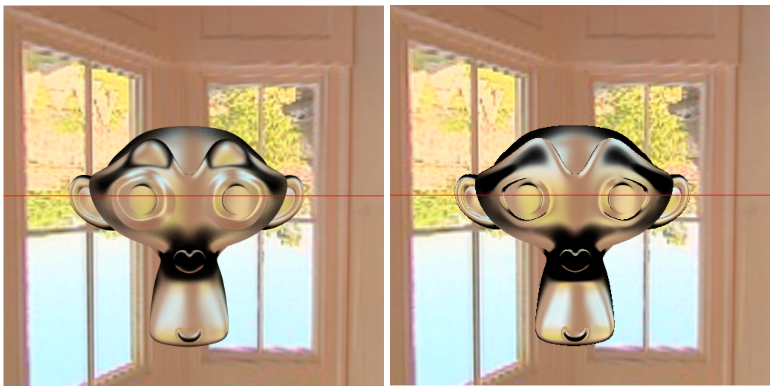
We have implemented the technique described in the paper in C++ with OpenGL/GLSL. The images and videos have been rendered on an Intel Xeon 2.67GHz with a nVidia 260 GTX graphics card. All images and videos are captured with a 512×512 resolution. The ground truth results are generated using a path tracer with importance sampling implemented using nVidia Optix [22]. Ground truth images are generated with 512 samples per pixel.

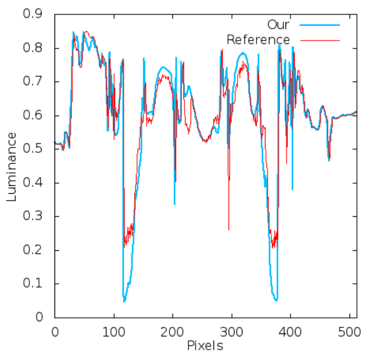
Since our technique does not require any precomputation on the mesh, we naturally support fully dynamic scenes.

6.1 Constant roughness

Figure 10 and Figure 11 show rough refractions through complex models in a comparison with ground truth. For fair comparison, images are rendered without managing total internal reflections. The scan line plot in Figure 10 illustrates the accuracy of our method for nearly convex objects.

Figure 12 shows comparisons of our method with ground truth with approximate total internal reflections. While


(a) Reference (b) Our model



(c) Line plot

Fig. 10: *Comparison with ground truth image without total internal reflection. Scan line plot of the values along the red line are shown. The graph shows a good fit between the ground truth and our technique. The arches of the monkey's eyebrow appear different using our model, since the model violates the convexity hypothesis.*

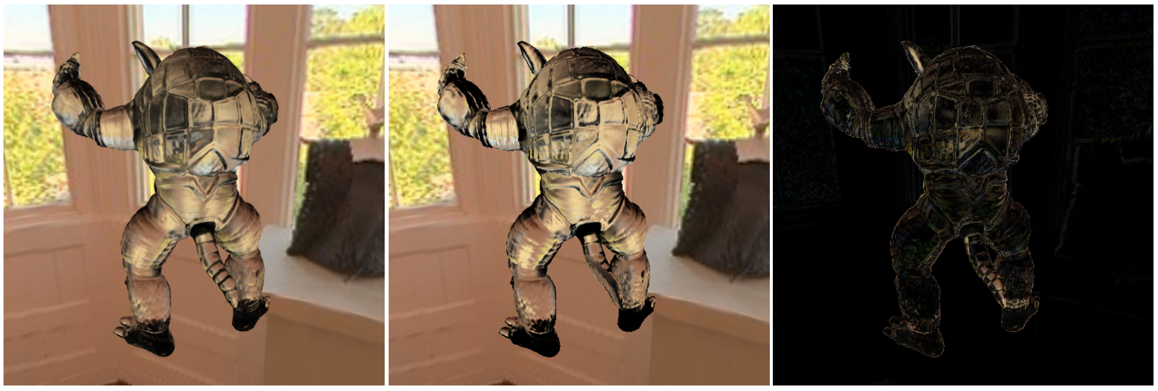

(a) Ground truth (b) Our method (c) Difference image

Fig. 11: *Comparison with ground truth without TIR. (Left) Ground truth. (Middle) Our method. (Right) The difference between the two images. Although our result is plausible, some artifacts are visible due to our assumptions (see Sec. 3.2).*

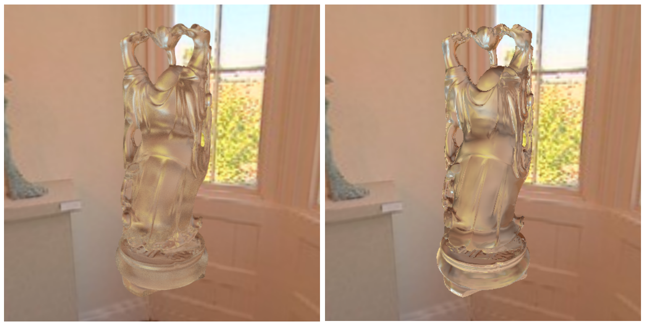

(a) Ground truth (b) Our method

Fig. 12: *Comparison with ground truth with total internal reflection: While images have some differences, the result remains plausible.*

some subtle differences can be noticed, our method produces plausible rough refractions despite approximate rendering of total internal reflections. See section 7 for a deeper analysis of approximations.

Figure 19 shows the Bunny and Armadillo models with several BTDF exponents κ_γ^D to illustrate the range of materials that can be achieved with our method. Our model supports modification of the BTDF's exponent κ_γ^D in real-time, allowing artists to adjust the glossiness of the refractions to obtain a desired appearance.

Figure 14 summarizes the performance of our method. While our method for rendering rough materials is on average two times slower than rendering of specular materials [2], we achieve a speed up of several orders of magnitude compared to ground truth raytracing for comparable image quality. Since our technique is mainly excuted in a fragment shader, we indicate the number of pixels covered by the geometry in addition to the number of triangles. The graph in Figure 14 illustrates the rendering cost of each pass of our algorithm, as well as the rendering cost for purely specular materials [2].

6.2 Spatially-varying roughness and local lighting

Figure 13 shows a globe with two different roughnesses — one for the continents and one for the oceans. We compare our result (13c) to ground truth (13b) and to light transport assuming uniform roughness on the back interface (13a). All three results are rendered without TIR. The shape of the continents on the back interface are visible through the sphere in both the ground truth image and our result. In contrast the simple transport with uniform roughness results

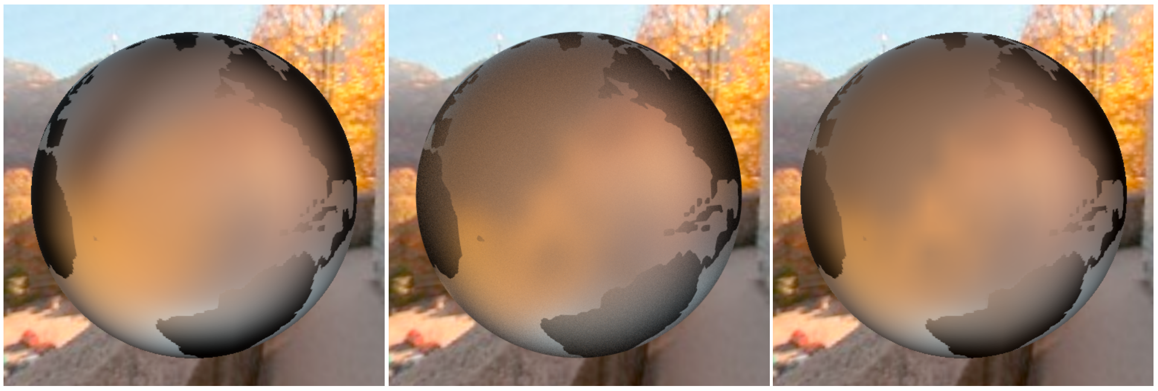


Fig. 13: *Comparison of approaches for spatially varying roughness on a sphere. The roughness is set to* $\kappa_\gamma^D = 4.2$ *for the continents and* $\kappa_\gamma^D = 100$ *for the oceans. Images have been rendered without TIR.*

Models	#Tri	#Pix	Specular	Our	GTruth
Sphere	20k	80k	1.4 ms	3.5 ms	47 s
Bunny	65k	58k	2.2 ms	7.2 ms	47 s
Monkey	70k	76k	6.3 ms	11.9 ms	46 s
Buddha	100k	65k	10.8 ms	19.6 ms	51 s
Dragon	100k	87k	10.0 ms	19.8 ms	65 s

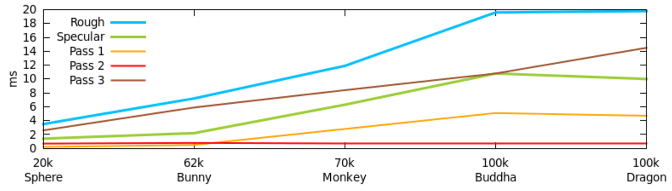


Fig. 14: *This table shows the performance of our methods according to the number of triangles #Tri and the number of pixels #Pix covered by the geometry. We compare the timings of our method to a GPU specular refraction algorithm and to the ground truth rendering generated with Optix. Ground truth timings are given for 512 samples per pixel.*

in an excessive blur of these shapes. See the accompanying video for an animated version of this figure.

Figure 15 shows additional results with spatially varying roughness on the bunny and the dragon models rendered under different lighting conditions with total internal reflection.

Figure 16 shows a plane with a SV-BTDF lit by distant lighting and a spherical local light. The areas with different roughnesses react differently to the lighting.

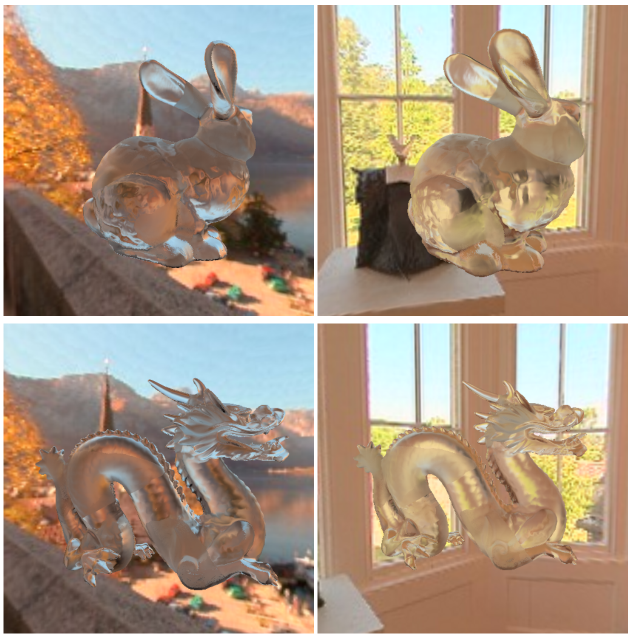


Fig. 15: *Additional results with Bunny and Dragon models under various environment maps with a spatially varying roughness. The roughness texture contains 2 different roughnesses :* $\kappa_\gamma^D = 3.4$ *and* $\kappa_\gamma^D = 100$

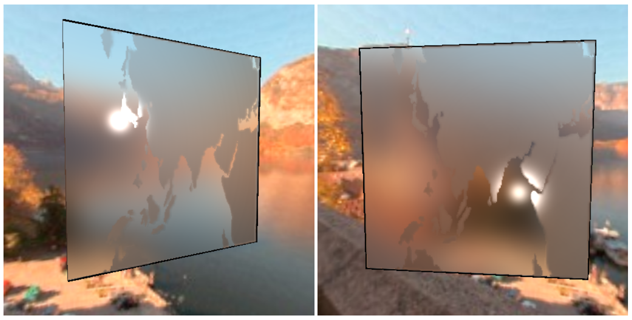


Fig. 16: *Rendering of a thin slab with a spatially varying roughness lit by a spherical local light behind the surface.*

7 Discussion and Limitations

7.1 Validity of our hypothesis

We made the following assumptions in deriving the transport operator inside the object (see section 3):

- *Rays are parallel as they arrive at the back interface*: this assumption is approximately true as long as the distance between the front and back interfaces is at least twice the size of the footprint. For geometry with locally high-curvature, this assumption does not hold.

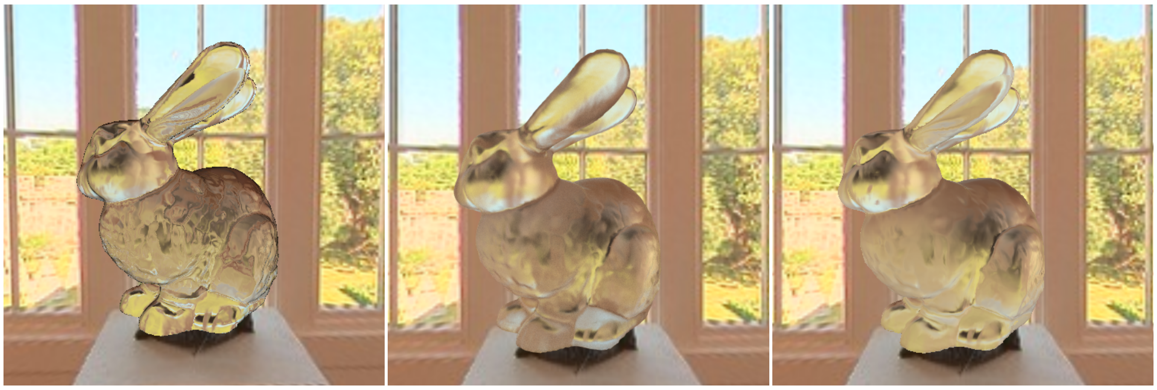


Fig. 17: *Comparison of transport methods including total internal reflections: (Left) Specular transport with fixed mipmap level look up into the environment map based on depth traveled by the ray inside the object as done by Eismann et al. [16]. (Middle) Ground truth. (Right) Our approach, also accounting for the lobe tranport. Our technique accounts well for the low-pass filtering effect of the BTDF on the geometry at the back interface.*

This partially explains the incorrect appearance of the tail of the Armadillo model (Figure 11).

- *The shape of the BTDF lobe is independent of the incident angle*: this hypothesis holds if the NDF variance is not too large. If this hypothesis is violated, transmitted energy tends to be over-estimated.
- *A single lobe NDF will be sufficient to represent the underlying geometry*: we suppose the geometry of the back interface can be approximated by a single gaussian NDF. Extreme cases, such as v-grooves, violate this assumptions. In this case, our method would over-smooth object appearance.

We find that, while the violation of these hypotheses would produce images which diverge from ground truth, they would not introduce distracting artifacts. At worst, our method tends to over-smooth the solution.

Since we rely on a screen-space algorithm to estimate the NDF at the back interface, our method is restricted to geometry that lies inside the view frustum. Besides, normal estimation and ray-marching are not precise near silhouettes. This produces deviations from ground truth.

7.2 Approximation and perception

Khan et al. [3] and Yeung et al. [4] observed that humans are poor at assessing the geometric accuracy of specular refractions. Similar to existing work on real time refractions [1], [2], our assumption of convex shape and our approximation of total internal reflections benefit from this tolerance of human observers to geometric errors in portrayed refractions. Figure 18 illustrates different strategies to approximate total internal reflections and depicts that the results remain plausible even in comparison to reference solutions.

Studies on material perception such as the work of Fleming et al. [5] shows, however, that people are sensitive to variations in roughness. Figure 17 compares our light transport based on cone-tracing and geometry filtering with existing methods that only trace one ray inside the object. Our method better captures the blur produced by scattering at the back interface compared to the results of existing methods that appear overly sharp. Figure 13 shows how our extension for spatially varying roughness matches the ground truth compared to uniform roughness that tends to over-blur the image. Additional studies on the perception

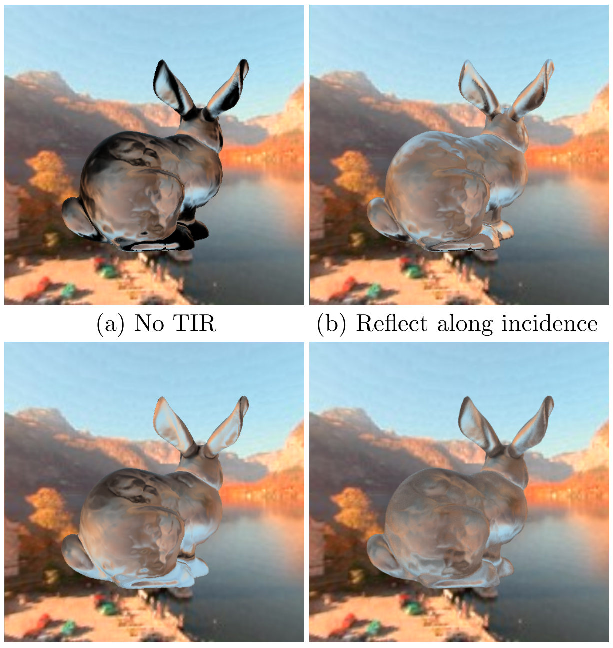
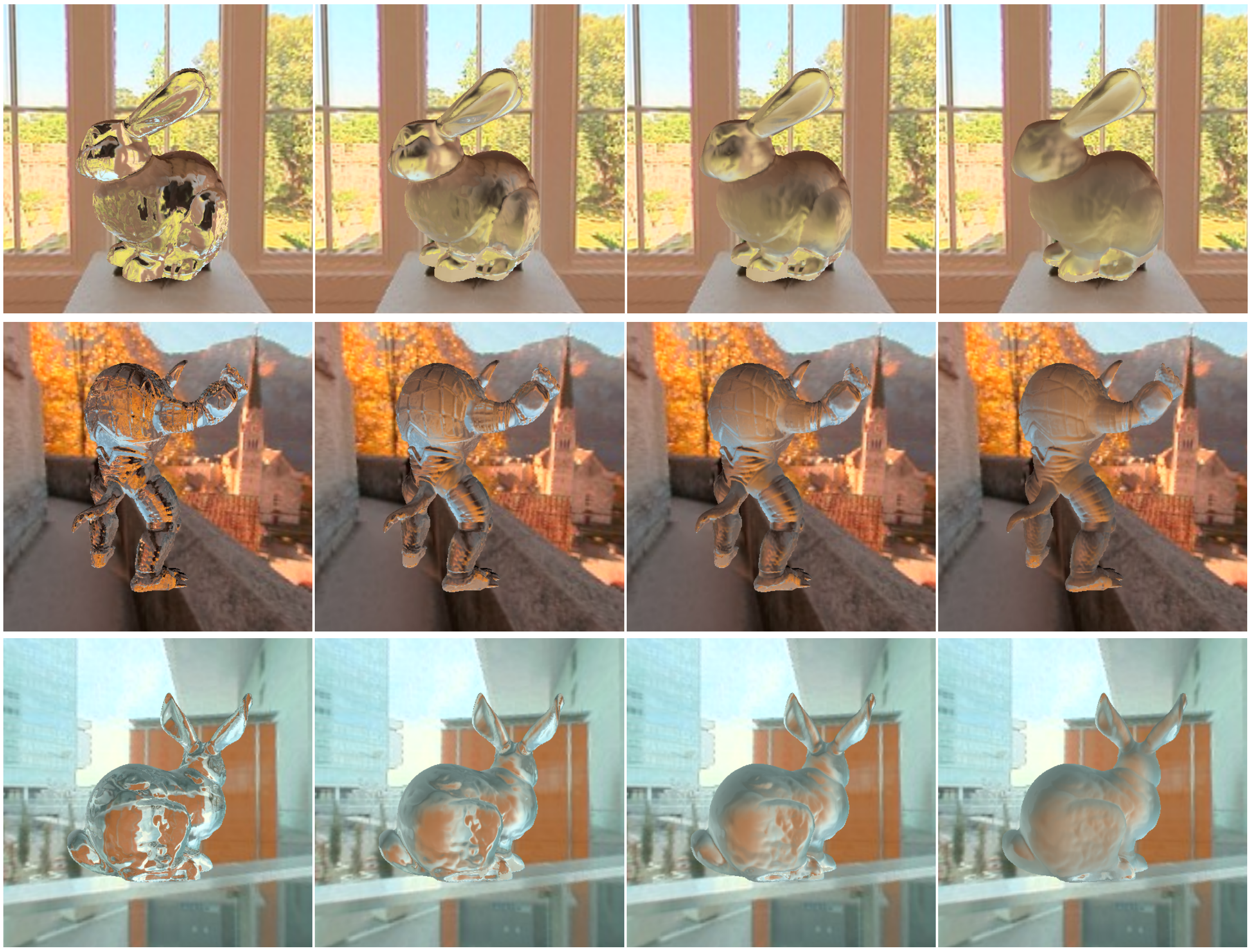


Fig. 18: *Different approaches to approximate total internal reflection on a bunny with a spatially varying roughness. (a) No TIR. (b) The incident direction is used as reflected direction. (c) The refracted direction is clamped to the tangent plane. (d) Ground truth.*



(a) $\kappa_\gamma^D = 20000$ (b) $\kappa_\gamma^D = 90$ (c) $\kappa_\gamma^D = 24$ (d) $\kappa_\gamma^D = 6.4$

Fig. 19: *Additional results with Bunny and Armadillo models under various environment maps for different roughness.*

of rough refractions would be needed to rigorously assess the validity of our assumptions for geometry filtering and spatially varying roughness as listed in Section 3.2 and 5.1 respectively.

8 Conclusion

We have presented a new algorithm for rendering transparent objects with rough surfaces. Our algorithm renders plausible images in real time. We presented a new formulation for the scattering function at the interface, and a combination of cone tracing and geometry filtering to approximate the rays leaving at the exiting interface.

We exploited results from studies which observed that humans are poor at assessing the accuracy of portrayed refractions, by providing simpler alternatives for various stages in rendering rough-refractive objects. Finally, we extended our model to include spatially-varying roughness and simple local lights such as points and spheres.

In future work we plan to investigate lighting models for arbitrary thick objects. Furthermore, an analytical description of spherical gaussian parameters would remove the fitting step and avoid storing those parameters into textures. This is likely to accelerate the shader evaluation.

Acknowledgment

This work was supported in part by NSF grant 0924968, ANR grant ANR-07-MDCO-001, the ExploraDoc program and the Intel STC for Visual Computing. We also acknowledge additional support from NSF grant 1011832, NVIDIA through a professor partnership award and graphics cards, and equipment and funding from Adobe and Pixar. The environment map used in this paper were provided by Bernhard Vogl (http://dativ.at/lightprobes/). Kartic Subr acknowledges support from the Royal Society and Royal Academy of Engineering by means of the Newton International Fellowship. The models used in this paper come from the Stanford 3D scanning respository, except the monkey model which comes from the Blender project.

References

[1] C. Wyman, "An approximate image-space approach for interactive refraction," *ACM Transactions on Graphics (SIGGRAPH 2005)*, vol. 24, no. 3, pp. 1050–1053, 2005.

[2] M. M. Oliveira and M. Brauwers, "Real-time refraction through deformable objects," in *Symposium on Interactive 3D Graphics and Games (I3D)*, 2007, pp. 89–96.

[3] E. A. Khan, E. Reinhard, R. W. Fleming, and H. H. Bülthoff, "Image-based material editing," *ACM Trans. Graph.*, vol. 25, pp. 654–663, July 2006. [Online]. Available: http://doi.acm.org/10.1145/1141911.1141937

[4] S. K. Yeung, C.-K. Tang, M. S. Brown, and S. B. Kang, "Matting and compositing of transparent and refractive objects," *ACM Transactions on Graphics*, vol. 30, no. 1, p. 2, 2011.

[5] R. W. Fleming, R. O. Dror, and E. H. Adelson, "Real-world illumination and the perception of surface reflectance properties," *Journal of Vision*, vol. 3, no. 5, pp. 347–368, 2003.

[6] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance, "Microfacet models for refraction through rough surfaces," in *Eurographics Symposium on Rendering*, 2007.

[7] J. Kautz, P.-P. Vázquez, W. Heidrich, and H.-P. Seidel, "Unified approach to prefiltered environment maps," in *Eurographics Workshop on Rendering*, 2000, pp. 185–196.

[8] R. Ramamoorthi and P. Hanrahan, "Frequency space environment map rendering," *ACM Transactions on Graphics (SIGGRAPH '02)*, vol. 21, no. 3, pp. 517–526, 2002.

[9] J. Amanatides, "Ray tracing with cones," *Computer Graphics (SIGGRAPH '84)*, vol. 18, no. 3, pp. 129–135, 1984.

[10] C. Han, B. Sun, R. Ramamoorthi, and E. Grinspun, "Frequency domain normal map filtering," *ACM Transactions on Graphics (SIGGRAPH 2007)*, vol. 26, no. 3, p. 28, 2007.

[11] C. de Rousiers, A. Bousseau, K. Subr, N. Holzschuch, and R. Ramamoorthi, "Real-time rough refraction," in *Symposium on Interactive 3D Graphics and Games*, ser. I3D '11. New York, NY, USA: ACM, 2011, pp. 111–118 PAGE@7. [Online]. Available: http://doi.acm.org/10.1145/1944745.1944764

[12] X. Sun, K. Zhou, E. Stollnitz, J. Shi, and B. Guo, "Interactive relighting of dynamic refractive objects," *ACM Transactions on Graphics (SIGGRAPH '08)*, vol. 27, no. 3, pp. 1–9, 2008.

[13] I. Ihrke, G. Ziegler, A. Tevs, C. Theobalt, M. Magnor, and H.-P. Seidel, "Eikonal rendering: Efficient light transport in refractive objects," *ACM Transactions on Graphics (Siggraph'07)*, vol. 26, no. 3, p. 59, Aug. 2007.

[14] C. Cao, Z. Ren, B. Guo, and K. Zhou, "Interactive Rendering of Non-Constant, Refractive Media Using the Ray Equations of Gradient-Index Optics," *Computer Graphics Forum (EGSR proceedings)*, vol. 29, no. 4, pp. 1375–1382, 2010. [Online]. Available: http://www.eg.org/EG/CGF/volume29/issue4/v29i4pp1375-1382.pdf

[15] W. Heidrich, H. Lensch, M. F. Cohen, and H.-P. Seidel, "Light field techniques for reflections and refractions," in *Eurographics Symposium on Rendering*, 1999.

[16] E. Eisemann and X. Décoret, "Fast scene voxelization and applications," in *Symposium on Interactive 3D Graphics and Games (I3D)*, 2006, pp. 71–78.

[17] J. F. Blinn, "Models of light reflection for computer synthesized pictures," *Computer Graphics (SIGGRAPH '77)*, vol. 11, no. 3, pp. 192–198, 1977.

[18] Q. Dai, J. Wang, Y. Liu, J. Snyder, E. Wu, and B. Guo, "The dual-microfacet model for capturing thin transparent slabs," *Computer Graphics Forum*, vol. 28, no. 7, pp. 1917–1925, 2009.

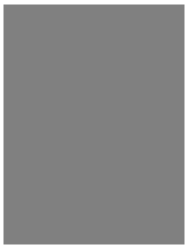
[19] R. Ramamoorthi and P. Hanrahan, "A signal-processing framework for inverse rendering," in *SIGGRAPH '01*, 2001, pp. 117–128.

[20] M. Toksvig, "Mipmapping normal maps," *journal of graphics, GPU, and game tools*, vol. 10, no. 3, pp. 65–71, 2005.

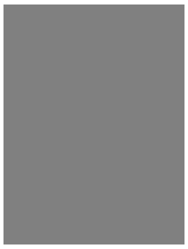
[21] J. Wang, P. Ren, M. Gong, J. Snyder, and B. Guo, "All-frequency rendering of dynamic, spatially-varying reflectance," *ACM Transactions on Graphics*, vol. 28, no. 5, 2009.

[22] S. G. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, D. McAllister, M. McGuire, K. Morley, A. Robison, and M. Stich, "Optix: a general purpose ray tracing engine," *ACM Transactions on Graphics*, vol. 29, no. 4, pp. 1–13, 2010.

Charles de Rousiers Charles de Rousiers is currently a PhD candidate at INRIA Grenoble Rhône-Alpes in the ARTIS research team. He studies complex materials representation for realistic rendering under the supervision of Nicolas Holzschuch. He was a visiting scholar at UC Berkeley under the supervision of Ravi Ramamoorthi for 6 months, during which time this research was initiated.



Adrien Bousseau Adrien Bousseau is a junior researcher at INRIA Sophia-Antipolis in the REVES research group. He did his PhD from 2006-2009 in the Artis group at INRIA Rhône-Alpes under the supervision of Joëlle Thollot and Francois X. Sillion. He was a postdoc at UC Berkeley, working with Maneesh Agrawala and Ravi Ramamoorthi. During his PhD he also spent 6 months in Seattle as an intern at Adobe's Advanced Technology Labs under the supervision of David Salesin, and 3 months in Cambridge at MIT CSAIL under the supervision of Fredo Durand and Sylvain Paris. Adrien Bousseau works on non photorealistic rendering (NPR), and more generally image synthesis and image processing. He received the 2011 Eurographics PhD Award for his work on expressive image manipulations.



Kartic Subr Kartic is currently a Newton Fellow and Honorary Research Associate at University College London. He was a post-doctoral researcher in the computer graphics group at INRIA-Grenoble from 2008-2010. He received his PhD (2008), under supervision by James Arvo, from the University of California, Irvine. During his PhD, he enjoyed a variety of short stints interning at venues including Rhythm and Hues Studios, NVIDIA Inc. and Columbia University. Kartic's research interests span most aspects of realistic picture generation and manipulation.

Nicolas Holzschuch Nicolas Holzschuch is a Senior Researcher at INRIA Grenoble Rhône-Alpes, and the scientific leader of the ARTIS research team. He got his PhD from Grenoble University in 1996, and joined the INRIA in 1997, first in Nancy, then in Grenoble. His research interests include photorealistic rendering and real-time rendering, with an emphasis on scalability, and multi-scale phenomena.

Ravi Ramamoorthi Ravi Ramamoorthi received the BS degree in engineering and applied science and the MS degree in computer science and physics from the California Institute of Technology in 1998. He received the PhD degree in computer science from Stanford University's Computer Graphics Laboratory in 2002, upon which he joined the computer science department at Columbia University, as an assistant and later associate professor. He assumed his current faculty position at the University of California, Berkeley in January 2009. His research interests cover many aspects of computer vision and graphics, including mathematical foundations, real-time photorealistic rendering, image-based and inverse rendering, and lighting and appearance in computer vision. He has published papers on all of these topics in leading graphics and vision conferences and journals. His work has been recognized my many awards including the NSF Career award, the Sloan Fellowship, an Okawa Foundation Research grant, an ONR Young Investigator Award, the ACM SIGGRAPH Significant New Researcher Award, and the white house's Presidential Early Career Award for Scientist and Engineers (PECASE).