Visually-Uniform Reparametrization of Material Appearance through Density Redistribution

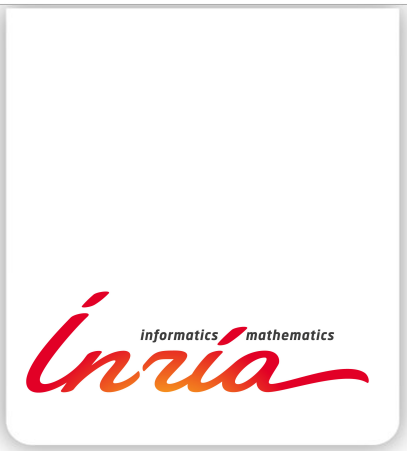
Pascal Barla, Laurent Belcour, Romain Pacanowski

Visually-Uniform Reparametrization of Material Appearance through Density Redistribution

Pascal Barla, Laurent Belcour, Romain Pacanowski

RESEARCH REPORT
N° 9424
October 6, 2021
Project-Team Manao

ISSN 0249-6399 ISRN INRIA/RR--9424--FR+ENG



Visually-Uniform Reparametrization of Material Appearance through Density Redistribution

Pascal Barla*, Laurent Belcour†, Romain Pacanowski*

Project-Team Manao

Research Report n° 9424 — October 6, 2021 — 13 pages

Abstract: Variations of material parameters such as roughness, sheen, etc. do not usually produce visually-uniform changes in rendered images. This might make it detrimental to artistic edition or prefiltering. In this work, we provide a methodology to reparametrize non visually-uniform parameters, by uniformly distributing the visual changes across the range of parameter values. For 1D and 2D parameters, our solution boils down to inverting a Cumulative Distribution Function. We provide three concrete applications of this method to reparametrize the complex refractive index, roughness and sheen.

Key-words: Computer Graphics, Material appearance, Artistic control

* Inria Bordeaux Sud-Ouest
† Unity Labs

RESEARCH CENTRE
BORDEAUX – SUD-OUEST

200 avenue de la Vieille Tour
33405 Talence Cedex

Reparametrization Uniforme de l'Apparence des Matériaux par Redistribution de Densité

Résumé : Des variations de paramètres de modèles matériaux tels que la ruguosité ne produisent généralement pas de changements qui apparaissent visuellement uniformes en synthèse d'images. Cela a pour conséquence de compliquer le contrôle artistique ainsi que les méthodes de pré-filtrage. Dans ce rapport, nous proposons une méthodologie qui permet de reparamétriser des modèles de matériau en redistribuant de manière uniforme les changements d'apparence en jouant sur les valeurs des paramètres. Pour des paramètres à une ou deux dimensions, notre solution revient à inverser une fonction de distribution cumulative. Nous présentons trois applications concrètes de cette méthode portant sur trois types de paramètres: l'indice de réfraction complexe, la rugosité et les effets de lustre diffus.

Mots-clés : Synthèse d'images, Apparence des matériaux, Contrôle artistique

1 Introduction and Related Work

Most material models in Computer Graphics expose a set of parameters that characterize statistical optical properties of a surface or volume. From a user perspective, a desired feature of such models is to yield *visually-uniform* changes in appearance when their parameters are modified in uniform increments. It is usually difficult to come up with visually-uniform parametrizations for multiple reasons: parameters may have physical origins that produce non-linearities in the image formation process; multiple parameters may be related to each other, so that their impact on appearance depends on other parameter values; complex appearance may also crucially depend on object shape and the surrounding lighting environment.

In this paper, we introduce a method to *reparametrize* material models to produce visually-uniform changes in appearance with a simple approach (see Figure 1). The key idea is to cast the reparametrization problem as a density redistribution problem: intuitively, we want pockets of parameter space values that yield abrupt changes in appearance to be spread around. As explained in Section 2, this is very similar to the inversion of Cumulative Distribution Functions (CDFs) used in rendering with importance sampling.

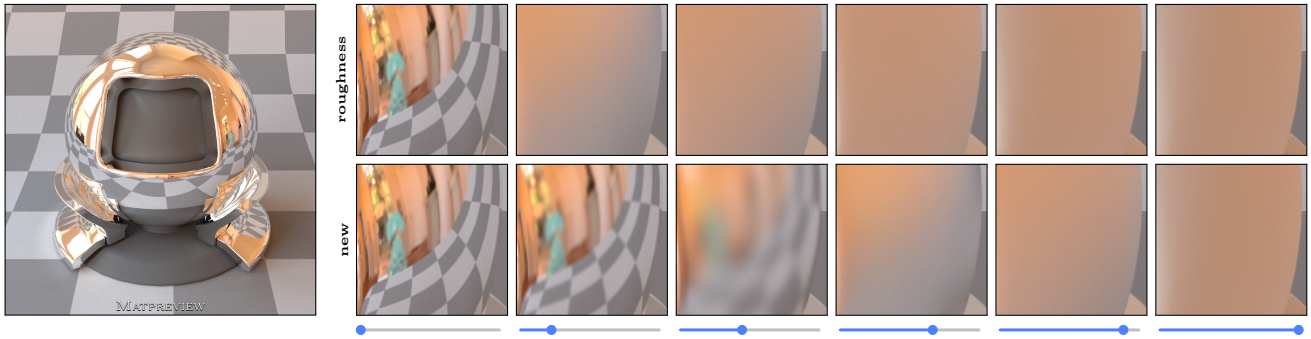


Figure 1: Varying material parameters does not usually produce visually-uniform variations of material appearance. For example, in the case of microfacet roughness (top row), the distinctness of the reflected image changes abruptly. We propose a methodology to correct such non-uniform parametrizations (bottom row) by redistributing differences in material appearance to yield visually-uniform variations.

One important difference is that we must characterize infinitesimal changes in material appearance. This raises two practical questions. First, in which space should the materials be compared? As detailed in Section 3, we choose to focus on material space, hence ignoring the effects of shape or lighting. The rationale behind this choice is that it provides a more generic solution to the reparametrization problem, one that can be reused for different types of asset. As shown in Figure 9, performing comparisons in image space instead leads to significant variations with mere changes of lighting environments. The second question is whether we need a single material difference function, or one for each new effect. We have found the latter to be necessary, since different material properties have different effects (e.g., multiplicative, convolutive) in relation to the environment lighting.

In practice, our CDF-based approach is used to reparametrize three common aspects of surface material appearance: the edge tint of conductors [Gul14], the roughness of GGX distributions [TR75, WMLT07], and specular sheen effects [CK17]. In each case, we provide analytical fits to reparametrization functions, which makes our approach readily testable in rendering engines implementing the corresponding material models, since it amounts to modifying material parameters prior to their use. We provide detailed results and comparisons with alternative image-based solutions in Section 4.

1.1 Related work

Previous approaches have proposed navigation tools to explore appearance spaces [MPBM03, NDM06, SGM+16] that can mix materials of different nature (metallic, plastic, etc). In contrast, ours is restricted to the reparametrization of an existing parametric material model within its own appearance space: we thus focus on providing enhanced low-level control over an existing material appearance, instead of exploring new appearance combinations.

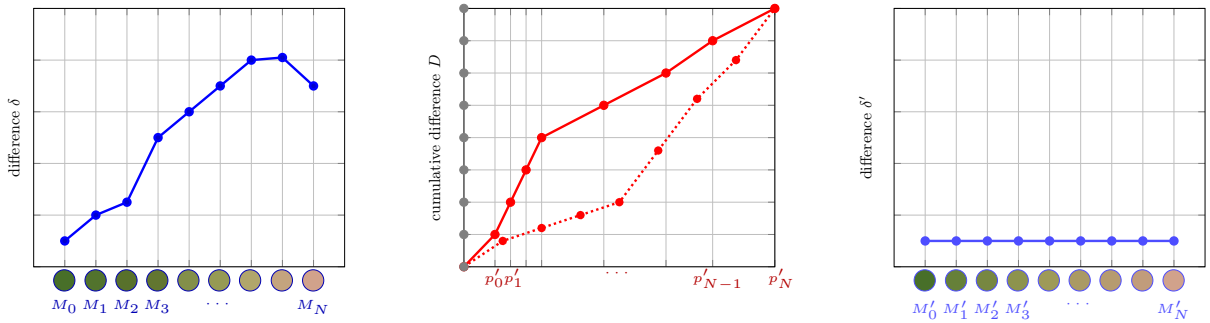
Methods based on multi-dimensional scaling (MDS) [PFG00, WAKB09] also reparametrize an existing material model, but focus on finding new parametric axes: pairs of material configurations are compared using perceptual experiments, yielding pairwise differences that are fed to MDS to obtain the best rotation and linear scaling of

the parametric space. Our method assumes the axes are already properly identified (which may be obtained through MDS), and rather addresses the issue of finding non-linear scaling functions that produce fine-grained, uniform visual differences.

A similar approach is taken for the specific case of hair in the work of Chiang et al. [CBTB16]: they let artists pick a visually-uniform sequence of renderings with their model, which produces a *sparse* set of constraints that are fitted using different analytic functions for each parameter. The main difference with our work is that we consider cases where the visual differences may be determined for a *dense* set of parameter values and with no human intervention. As discussed in Section 5, the two approaches may be seen as instances of a same framework.

2 General approach

We first outline our general approach in the 1D case. Let us thus assume we have a material model $M \in \mathcal{M}$ that depends on a single parameter $p \in \mathcal{P}$, where $\mathcal{P} \subset \mathbb{R}$ and $\mathcal{M}$ are left unspecified for the time being. A uniform sampling of $\mathcal{P}$ produces a set of parametric values $\{p_i\}, i \in [0, N]$, which in turn yields a set of corresponding materials $\{M_i\}$. Now let us further assume we can measure differences between pairs of materials, and call $\delta : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ the corresponding difference function. Computing the pairwise differences between successive materials in $\{M_i\}$ might look like Figure 2a (it is only shown for illustration purpose and does not correspond to any concrete case). This means that when varying the p parameter in uniform increments, the material M will not show a uniform change in appearance according to δ.



(a) Original differences δ_i (b) CDF-based reparametrization (c) Final differences δ'_j

Figure 2: **The three steps of our reparametrization method.** For (a) a set of materials $\{M_i\}$ obtained by uniform sampling $\{p_i\}$ of a single control parameter, we compute the differences $\{\delta_i\}$ between each material and its neighbors. Then (b) a Cumulative Distribution Function (CDF) D (dashed curve) is built from these differences. The inverse CDF D^{-1} (solid curve) is computed numerically, and used to obtain a new sequence of samples $\{p'_j\}$ (red ticks) from a set of samples $\{\xi_j\}$ distributed in the $[0, 1]$ range (light gray points). The resulting non-uniform sequence is used in (c) to generate a sequence of materials $\{M'_j\}$ that exhibit constant differences δ'_j.

Our approach reparametrizes $\mathcal{P}$ by computing the inverse of the Cumulative Distribution Function (CDF) of material differences. We first compute *unnormalized* CDF samples $\tilde{D}_k$, $k \in [\![0, N]\!]$ and then renormalize them to the $[0, 1]$ range to yield the CDF samples D_k:

$$\tilde{D}_k = \sum_{i=0}^{k} \delta_i, \qquad D_k = \frac{\tilde{D}_k - \tilde{D}_0}{\tilde{D}_N - \tilde{D}_0}, \tag{1}$$

where we use $\delta_i := \frac{\delta(M_{i-1}, M_i) + \delta(M_i, M_{i+1})}{2}$ for the difference between the i-th material and its neighbors, and assume mirror boundary conditions: $\delta_0 = \delta(M_0, M_1)$ and $\delta_N = \delta(M_{N-1}, M_N)$.

As shown in Figure 2b (dashed curve), the collection of CDF samples $\{D_k\}$ forms a discretized version of the CDF D. The inverse CDF, D^{-1} (solid curve), may now be computed numerically with arbitrary resolution, and stored in a lookup table. Alternatively, D^{-1} may be fitted with an analytical function as discussed in details in the following section.

Finally, as with importance sampling, a sequence of uniformly distributed samples $\{\xi_j\}$ in the $[0,1]$ range is input to D^{-1} to obtain a non-uniform sequence $\{p'_j\}$ of parameter values in the $\mathcal{P}$ domain, with the result that differences $\delta'_j := \frac{\delta(M'_{j-1},M'_j)+\delta(M'_j,M'_{j+1})}{2}$ between successive the new materials $\{M'_j\}$ and their neighbors are now constant as shown in Figure 2c. Note that $p_0 = p'_0$ and $p_N = p'_N$ by construction, hence $M_0 = M'_0$ and $M_N = M'_N$.

For material models that are controlled by more than one parameter, several strategies could be followed. One is to choose a difference function δ that cancels out the effect of the other parameters, which is well adapted to the case of roughness or sheen. Another strategy is to define an order on parameters: if one parameter is independent of the others, then for each of its values we may compute an inverse CDF for the second parameter; then follow the same strategy for the second and third parameters; etc. This is the natural strategy to follow for complex refractive indices.

3 Reparametrizations

The three reparametrizations presented in this paper all rely on microfacet theory [TS67, CT81]. In this context, a glossy BRDF M is defined as:

$$M(\omega_i,\omega_o) = \frac{D(\omega_h)G(\omega_i,\omega_o)F(\omega_h\cdot\omega_i)}{4(\omega_i\cdot\mathbf{n})(\omega_o\cdot\mathbf{n})}, \tag{2}$$

where ω_i and ω_o are the incoming and outgoing directions in the upper hemisphere Ω, $\omega_h = \frac{\omega_i+\omega_o}{\|\omega_i+\omega_o\|}$ is the halfway vector, $\mathbf{n}$ is the surface normal, D is the microfacet distribution function, G is the geometric attenuation factor, and F is the Fresnel reflectance term.

In Section 3.1, we present a reparametrization of the complex refractive index used in the Fresnel term for the case of conductors. We then consider two types of reparametrizations affecting the distribution of microfacets in Sections 3.2 and 3.3, for roughness and sheen effects respectively.

3.1 Refractive Index

In this section, we consider Equation 2 in the smooth case (i.e., with a flat micro-surface), which amounts to directly compute differences on Fresnel terms F. We choose the maximum absolute difference among all incidence angles $\theta_i = \arccos(\omega_i\cdot\mathbf{n})$ for the δ function:

$$\delta(M_1,M_2) = \max_{\theta_i} |F_1(\theta_i) - F_2(\theta_i)|. \tag{3}$$

However, we cannot use Equation 3 directly in the approach of Section 2 since for conductors, the Fresnel term depends on two scalars, the real and imaginary parts of the refractive index. We first convert from this complex refractive index to Gulbrandsen's parametrization [Gul14], which exposes two parameters: $r \in [0,1]$, the reflectance at normal incidence, and $g \in [0,1)$, the edge tint toward grazing angles. In this case, r is clearly independent of g, but the converse is not true; it is thus natural to reparametrize g as a function of r.

Figure 3a shows the difference functions of Equation 3 obtained by varying g, for 10 fixed values of r uniformly sampled in the $[0,1]$ range (different curves colored cyan to blue). This suggests that most of the variation is concentrated toward the largest g values. Using these differences in Equation 1 and inverting yields a family of inverse CDF functions D^{-1}, one for each r value, as shown in Figure 3b. This 2D reparametrization function produces an edge tint value g given two inputs: the reflectivity r and a new parameter g' that is used to control grazing angle effects in a visually-uniform manner.

We have found that directly fitting this 2D reparametrization function required a very large number of coefficients to achieve reasonable accuracy. Instead, we fit 10 1D inverse CDFs for uniformly sampled values of r (including $r = 0$ and $r = 1$), and interpolate linearly interpolating. In terms of implementation, for each of the inverse CDF, we fit a rational polynomial of the form:

$$f(x) = \frac{\sum_{i=0}^{N_p} p_i x^i}{\sum_{j=0}^{N_q} q_j x^j}, \tag{4}$$

where $p_0 = 0$ since we have $f(0) = 0$, and $p_{N_p} = \sum_{j=0}^{N_q} q_j - \sum_{i=1}^{N_p-1} p_i$ due to the $f(1) = 1$ constraint. Fitting is performed using $N_p = 4$ and $N_q = 3$. The fitted functions are shown as dashed curves in Figure 3b. The

reparametrization code is given in Listing 1, along with computed coefficients: it gives an edge tint parameter g to be used in Gulbrandsen's parametrization. The distance functions δ' after reparametrization are near constant as desired, as can be seen in Figure 3c. A slight increase of δ' remains for very high r and g', which could be taken care of with higher-order rational polynomials; but we have not found the need to do so in practice.

Listing 1: Python code for **complex refractive index reparametrization**

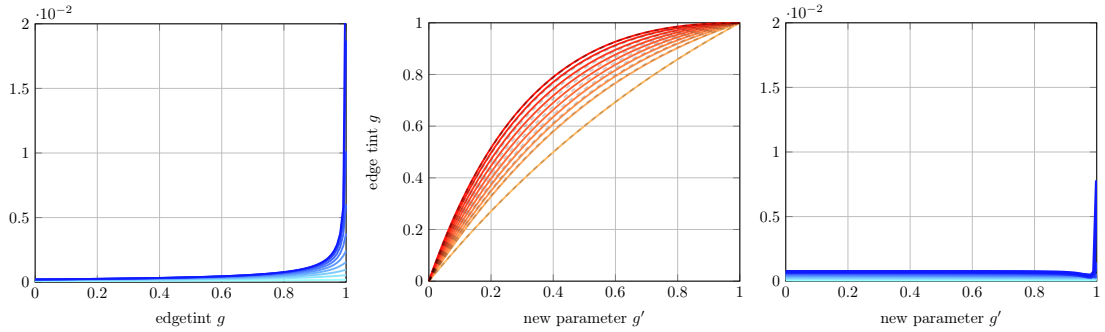
```
ps =  numpy.array(\
[[0.761624521141218,-0.791800432837066,0.0340827353164465],\
[0.641219841829447,-0.955579526199114,0.367892231209608],\
[0.504293924525993,-0.975402107109498,0.574532710606476],\
[0.481015815391423,-0.986276750233156,0.63819163114932],\
[0.605570645152696,3.22912208169963,-4.6982426622247],\
[2.1980451003527,2.95340236634721,-6.76690219773513],\
[22.9272845767138,-15.1479975487854,25.3494965059587],\
[33.8138100735803,-29.7606985550111,15.8360520401615],\
[46.5152821507814,20.2259875291415,11.3779731175398],\
[87.330261285565,-99.8250322198085,15.2137457152897]])

qs =  numpy.array(\
[[0.518069987834517,-0.316430132798066,-0.18604770546933,-0.0154366907005347],\
[0.346610178136577,-0.309405427918929,-0.0296966338284561,-0.0260019481493535],\
[0.251910475737378,-0.314299098363677,0.0117300366318092,0.0473607327210316],\
[0.220056146352031,-0.270329104447671,-0.0225640124759208,0.0883194317854664],\
[0.253912722931515,1.58917910291272,-0.559420460217356,-1.04253579451892],\
[0.846439111816821,2.02877957568105,-0.886138131681798,-1.96141584349659],\
[8.12957863756479,4.44444739611676,6.56739901192009,7.08131699922802],\
[11.1090971151016,4.9034952067542,-0.796562371723013,1.0094782510825],\
[14.2221714149087,26.5474528542614,22.6132169939666,11.2793328269011],\
[25.3464944622573,9.15386779106598,-17.5005019538263,-16.2402235904496]])

def ratPoly(p, q, x):
    p4 = q[0] + q[1] + q[2] + q[3] - p[0] - p[1] - p[2]
    return (p4*x**4 + p[2]*x**3 + p[1]*x**2 + p[0]*x ) /  (q[3]*x**3 + q[2]*x**2 + q[1]*x + q[0])

def getEdgeTint(r, beta):
    idx = int(r*10)
    g_p = ratPoly(ps[idx], qs[idx], beta)
    if idx==9:
        return g_p

    g_n = ratPoly(ps[idx+1], qs[idx+1], beta)
    alpha = r*10 - idx
    return (1-alpha)*g_p + alpha*g_n
```



(a) Difference functions δ (b) CDF-based remapping D' (c) Difference functions δ'

Figure 3: **Reparametrization of the complex refractive index.** (a) When using Gulbrandsen's parametrization, material differences δ along the edge tint dimension g for various reflectivities r exhibit a strong increase toward $g = 1$. (b) Our CDF-based approach provides a mapping from a new edge tint parameter g' to Gulbrandsen's edge tint g. The dashed curves show the rational polynomial fits we obtain. (c) After remapping, the difference functions δ' are close to constant throughout the g' dimension. The only exception occurs for very high r and g' where the fit slightly looses its accuracy, which we have found sufficient in practice though.

3.2 Roughness

We now switch our attention to rough materials. The most common choice for the D term in Equation 2 is the GGX distribution [TR75, WMLT07]. For the sake of simplicity, we assume that the BRDF is isotropic, in which case D and G should be controlled by a same scalar parameter $\alpha \in [0, 1]$ [Hei14]. Taking into account the Fresnel term F, the difference function δ should then be a function of either two or three parameters for dielectrics or conductors respectively.

Besides dimensionality, another difficulty with finding a reparametrization for roughness is that the microfacet distribution has a *convolutive* rather than *multiplicative* effect on material appearance. Intuitively, increasing roughness does not merely affect image intensity but it also blurs out reflections. The amount of blur in images is correlated to the magnitude of the image gradient; this suggests looking at the BRDF gradient, since the derivative of a convolution is equivalent to convolving with the derivative of the blur kernel.

In practice, we make a number of choices for defining the material difference δ in the case of roughness: we restrict the difference to normal incidence (i.e., $\omega_i = \mathbf{n}$); we take the logarithm of the BRDF, which not only removes dependency on the refractive index but is also relevant to avoid dominance of very small roughness configurations; and we take the average absolute difference between BRDF *derivatives* along the $\theta_h = \arccos(\omega_h \cdot \mathbf{n})$ dimension. This yields:

$$\delta(M_1, M_2) = \frac{1}{N_o} \sum_{\omega_o \in \Omega} \left| \frac{\partial}{\partial \theta_h} \log\left(M_1(\mathbf{n}, \omega_o)\right) - \frac{\partial}{\partial \theta_h} \log\left(M_2(\mathbf{n}, \omega_o)\right) \right| \sin \theta_o, \tag{5}$$

where N_o is the number of ω_o samples, and the sine term is used to account for the solid angle associated to each ω_o direction (with $\theta_o = \arccos(\omega_o \cdot \mathbf{n})$).

Figure 4a visualizes the material difference δ as a function of roughness α as given in Equation 5. It exhibits a sharp peak for small values of α, implying that most of the variations in gradient magnitude are concentrated around small roughness values.

Using this difference in Equation 1 and inverting yields an inverse CDF function D^{-1}, plotted in Figure 4b. It is accurately fitted with a rational polynomial of the form of Equation 4 of degrees $N_p = 6$ and $N_q = 2$ respectively, as shown by the dotted curve. For reference, we also show the α^2 reparametrization function classically employed in production [Bur12]. Our reparametrization function along with fitted coefficients are provided in Listing 2.

The distance function δ' after reparametrization is near constant as desired, as shown in Figure 4c. In comparison, the α^2 reparametrization still exhibits a sharp peak.

Listing 2: Python code for **roughness reparametrization**

```
p = numpy.array([0.062511315492697,0.039120758967145,0.880443180680444, 1.118615811570759,-4.280213095575128])
q = numpy.array([4.894549426661904,-6.875101250309577,2.240268420585049])

def getRoughness(alpha_prime):
    x = alpha_prime
    p6 = q[0]+q[1]+q[2]-p[0]-p[1]-p[2]-p[3]-p[4]
    return (p6*x**6+p[4]*x**5+p[3]*x**4+p[2]*x**3+p[1]*x**2+p[0]*x) / (q[2]*x**2+q[1]*x+q[0])
```

3.3 Sheen

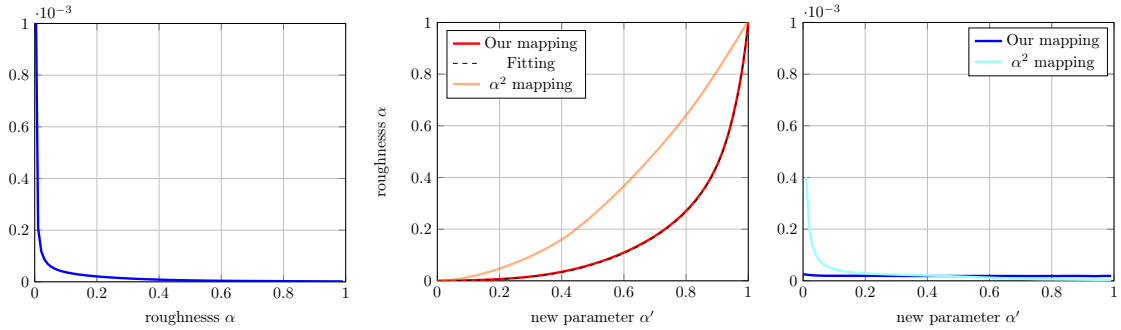
Materials may also exhibit roughness due to tiny structures sticking out of the surface, called asperities. Asperity scattering has been modeled empirically in the microfacet framework by Conty and Kulla [CK17], to typically reproduce back-scattering effects in cloth materials, which they call 'sheen'.

Since the main effect occurs in retro-reflection, we are going to concentrate on the subset of configurations where $\omega_i = \omega_o = \omega_h$. These are naturally parametrized by θ_i, and we are thus left with the task of finding a material difference function for this subset of light/view directions. A visually important criterion for sheen is the point of transition to the back-scattering effect, also called 'terminator'. We choose to identify the terminator as the location T along the θ_h dimension where the sheen effect has dropped to 5% of its maximal intensity:

$$T = \{\theta_i / M(\omega_i, \omega_i) = 0.05 M(\bar{\omega}_i, \bar{\omega}_i), \}$$

with $\bar{\omega}_i$ a grazing incoming direction (i.e., $\bar{\theta}_i = \frac{\pi}{2}$), since maximal back-scattering occurs at grazing angles. We may now simply define the difference function δ by:

$$\delta(M_1, M_2) = |T_1 - T_2|. \tag{6}$$



(a) Difference function δ (b) CDF-based remapping D' (c) Difference function δ'

Figure 4: **Reparametrization of GGX roughness.** (a) Material differences computed using Equation 5 show that the original roughness α induces strong visual differences δ for small parametric values, and negligible differences for large values. (b) The numerical inverse CDF D^{-1} is shown in red, along with the fit using a rational polynomial (dotted curve), that maps from the new parameter α' to the original one α. For reference, we show the α^2 reparametrization used in production (orange curve). (c) Using α' yields near constant material differences δ'. In comparison, using the α^2 reparametrization still exhibits a non-constant difference function.

Figure 5a visualizes the material difference δ as a function of roughness α as given in Equation 6. As in the previous section, it exhibits a sharp peak for small values of α, implying that most of the variations in back-scattering are concentrated around small roughness values.

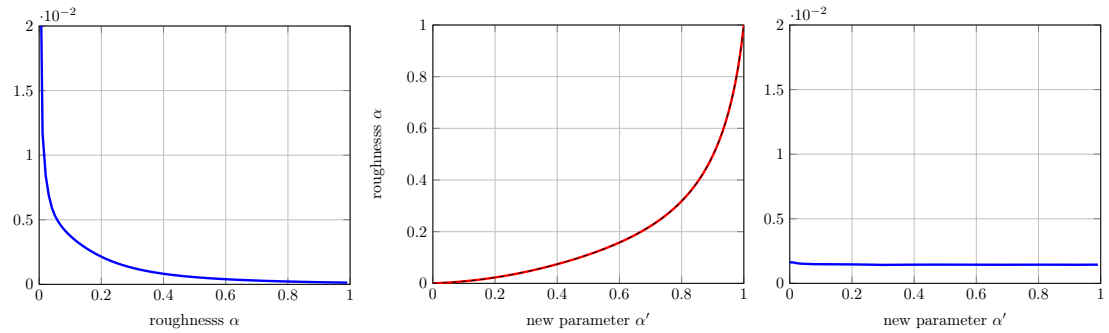
Using this difference in Equation 1 and inverting yields the inverse CDF function D^{-1} plotted in Figure 5b. As before, it is accurately fitted with a rational polynomial of the form of Equation 4 of degrees $N_p = 6$ and $N_q = 2$ respectively, as shown by the dotted curve. The reparametrization is thus identical to the previous section; only the fitted coefficients differ, as shown in Listing 3.

Once again, the distance function δ' after reparametrization is near constant as desired, as shown in Figure 5c.

Listing 3: Python code for **sheen reparametrization**

```
p = numpy.array([0.523399756604875,3.155082420442268,-5.010590077248909, -2.211394588430514,7.072030032363751])
q = numpy.array([11.853773649248811,-19.377593363553615,7.731762872228672])

def getSheen(alpha_prime):
    x = alpha_prime
    p6 = q[0]+q[1]+q[2]-p[0]-p[1]-p[2]-p[3]-p[4]
    return (p6*x**6+p[4]*x**5+p[3]*x**4+p[2]*x**3+p[1]*x**2+p[0]*x) / (q[2]*x**2+q[1]*x+q[0])
```



(a) Difference function δ (b) CDF-based remapping D' (c) Difference function δ'

Figure 5: **Reparametrization of sheen effects.** (a) Material differences computed using Equation 6 show that the original roughness α induces stronger visual differences δ for small parametric values than for larger ones. (b) The numerical inverse CDF D^{-1} is shown in red, along with the fit using a rational polynomial (dotted curve), that maps from the new parameter α' to the original one α. (c) Using α' yields near constant material differences δ'.

4 Results and Comparisons

In the following sections, we show rendering results obtained using the reparameterizations produced by our method. All renderings have been done using MITSUBA [Jak10].

4.1 Refractive Index

Results for remapped complex refractive indices (Section 3.1) are provided in Figure 6. Since the mapping is two-dimensional, we presents results for two values of reflectivity r and a linear range for either the edge tint parameter g, or our alternative edge tint g' obtained by reparametrization. We show the effect of both edge tint parameters only on the red channel to make the visual differences more salient.

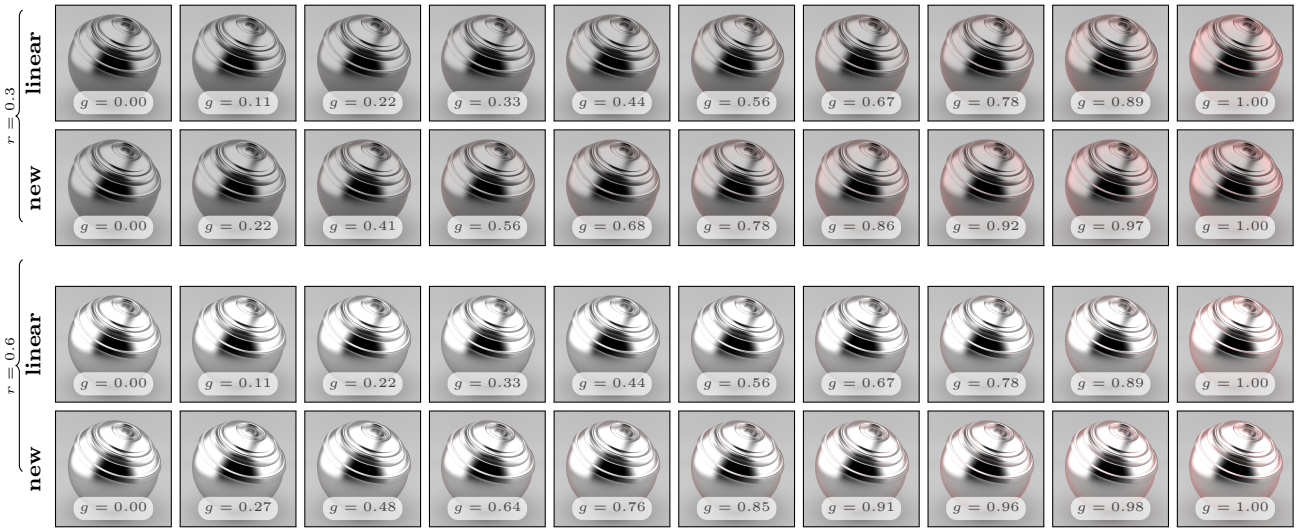


Figure 6: **Results for complex refractive index remapping.** We display results of a linear range of edge-tint values for a low ($r = 0.3$) and high ($r = 0.6$) reflectivity setting. Compared to renderings using Gulbransen's parameterization (r, g) (top rows), our new mapping (r, g') (bottom row) better distributes the redish edge tint effect. The associated g value is displayed for each configuration.

For both reflectivity values, our proposed mapping better distributes edge tint effects across parameter values. Note that the range of possible edge tint effects depends on the reflectivity r: the higher r, the smaller that range; in the extreme case where $r = 1$, a change in g or g' does not affect the output appearance.

4.2 Roughness

Results for roughness remapping (Section 3.2) are provided in Figure 7. We zoomed on a reflection off the MATPREVIEW scene available for MITSUBA to better observe the change made by our reparameterization. In this configuration, we display the linear parametrization of the roughness α, the quadratic parametrization α^2 used in production [Bur12], and our new parametrization α'.

The linear roughness mapping is known to tend quickly toward an appearance where reflections are completely blurred. The quadratic mapping is a clear improvement in this regard, but it still allocates more than half of the roughness range to appearances where surrounding objects cannot be distinguished in reflections. In contrast, our new roughness parametrization provides fine control over low roughness values, enabling artists to control accurately the distinctness of the image reflected off shiny objects. This is of course done at the expense of the roughest configurations, which are compressed to a small portion of the roughness range. Note that other types of reparametrizations could be readily explored with our approach, provided an alternative material difference function δ is given.

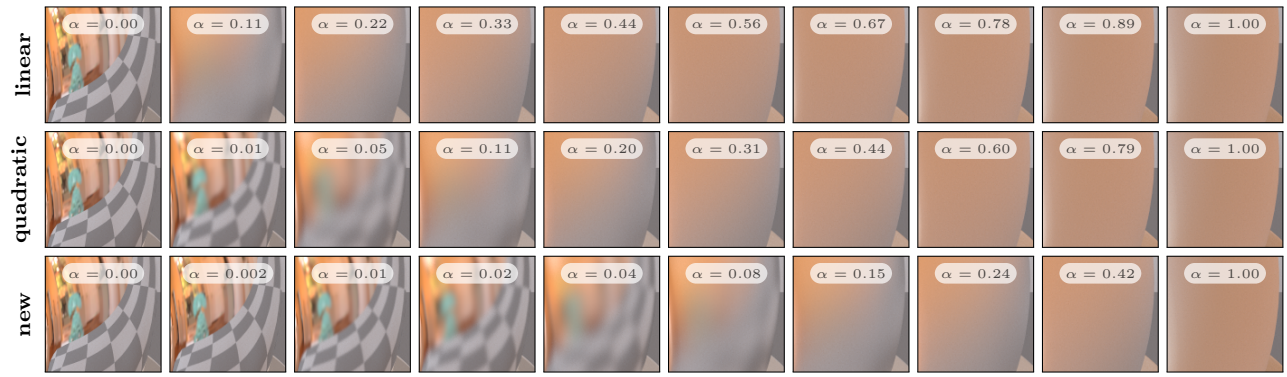


Figure 7: **Results for roughness remapping.** We display a series of crops for the rendering of the MATPREVIEW scene using a linear range of roughnesses for the GGX microfacet model (top row). We also showcase the traditionnal remapping of Burley [Bur12] (middle row), as well as our new remapping (bottom row), which better distributes the blurring of reflections off the object surface. For each configuration, the corresponding roughness parameter is provided.

4.3 Sheen

Results for sheen remapping (Section 3.3) are provided in Figure 8. Here we used a cloth model draped over a sphere that exhibits folds where sheen effects are commonly encountered. Besides the linear parametrization and our new parametrization, we have also tried, out of curiosity, a quadratic mapping as in the case of roughness.

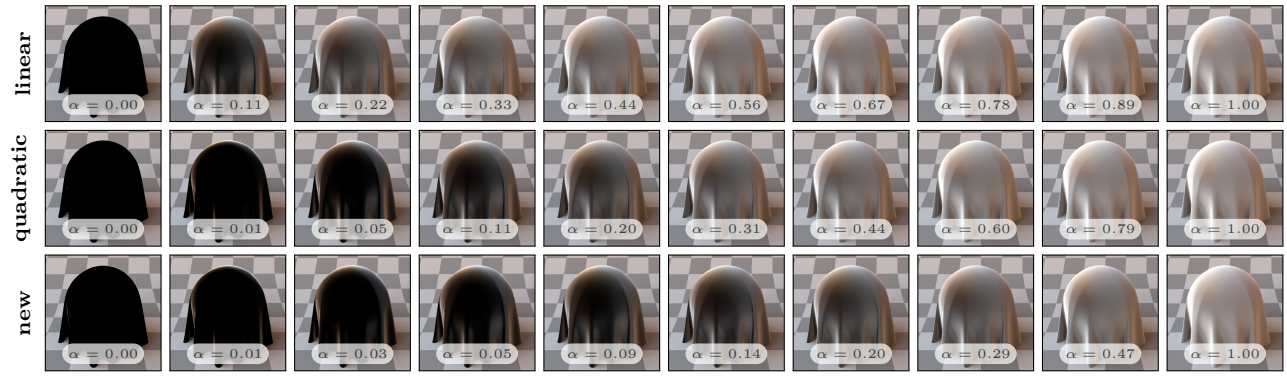


Figure 8: **Results for sheen remapping.** Rendering of a CLOTH scene using a linear range of roughnesses for the Sheen model of Conty and Kulla [CK17] (top row). We also showcase the remapping of Burley [Bur12] in this case (middle row) as well as our new remapping (bottom row). Our remapping better distributes the range of appearance, letting the sheen effect progressively spread toward the front-facing parts of the object surface. For each configuration, the corresponding roughness parameter is provided.

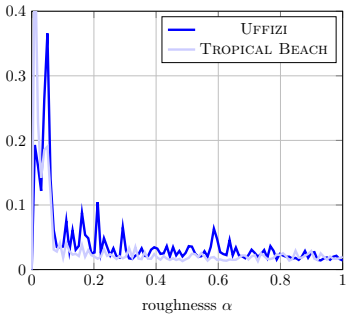
The linear mapping very quickly produces sheen effects that cover most of the cloth surface, similarly to roughness with reflection blur. As before, the quadratic mapping corrects that issue, but still devotes about half of the sheen range to effects that are hardly distinguishable as they nearly entirely cover the object surface. In contrast, our new mapping provides fine control over the extent of the sheen effect away from the object occluding contours, which is due to our choice of basing our material difference on sheen terminators.

4.4 Comparison with image-based metrics

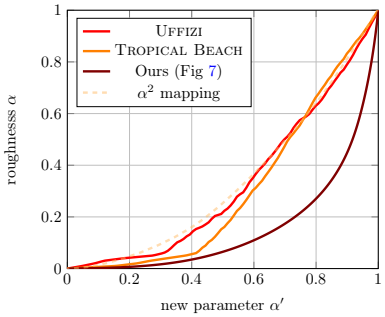
We also experimented with an image-based perceptual metric to act as the material difference function in our reparameterization approach. The neural network of Laguna et al. [LMS+19] was thus used to compute the pairwise material differences δ. More precisely, we rely on the L2 difference between normalized feature vectors

output by their neural network. Since the difference in lighting can affect material differences in this case, we have computed two reparametrizations, using two different HDR environments: Uffizi, where the dominant light is the overcast sky, and Tropical Beach, where the dominant light is the sun.

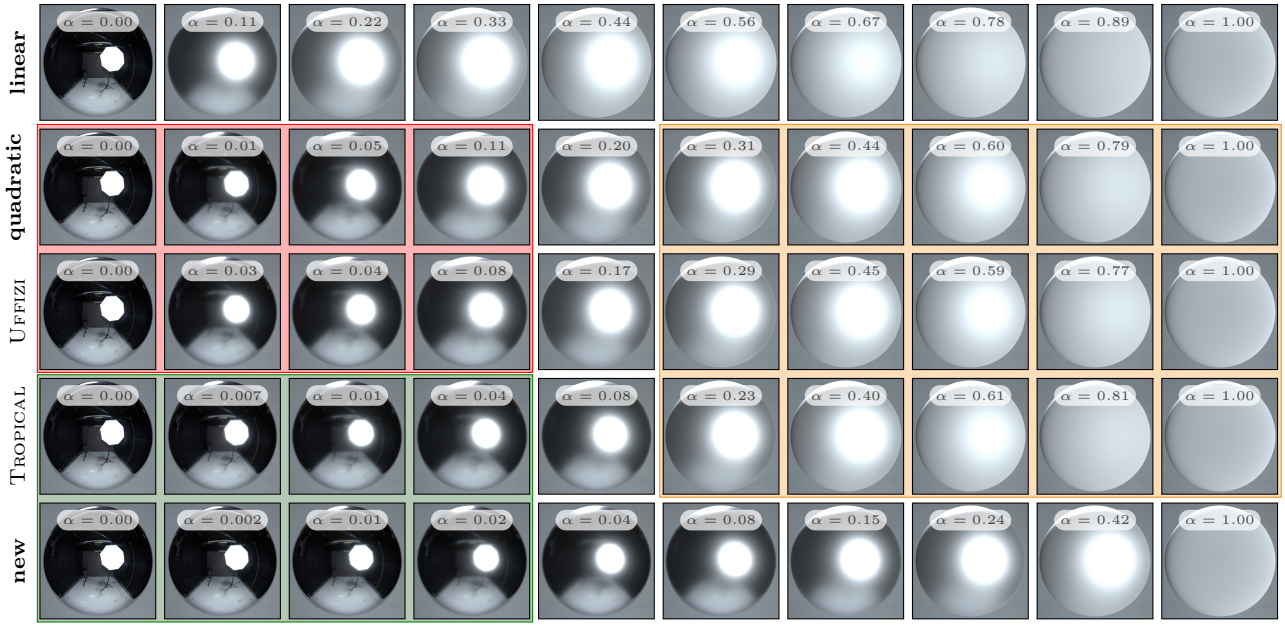
(a) Inputs to [LMS+19]



(b) Difference function δ



(c) CDF-based remapping D'



(d) Visualizing the different reparameterizations with the Studio HDR environments.

Figure 9: **Image-based perceptual metrics depend on lighting conditions.** We run the experiment of using [LMS+19] similarity measure as our δ function with (a) renderings of spheres lit by two different HDR environments. Depending on the environment, (b) the differences between images varies, and (c) the inverse CDF remappings differ. Visually, (d) both reparameterization match the quadratic remapping for large input values (orange box) but differ for small input values. In this range, an environment with a large area light (Uffizi) matches the *quadratic* remapping (red box) while an environment with a more punctual light (Tropical) better matches our new mapping (green box).

Results of this experiment are provided in Figure 9. The two image-based reparametrizations are rather similar in the high roughness range, but differ for low roughness values. Using the environment with a large angular light results in a parameterization close the quadratic case, whereas using the environment with a more punctual light source results in a parameterization closer to our gradient-based reparameterization. We believe that this dependence on lighting is generally not well adapted to the context of appearance reparametrization, where a single type of control that works reasonably well for different lighting environments will likely be preferred. Nevertheless, an image-based solution might still be chosen for the fine control of material appearance in specific shots. An

additional advantage of using a difference function in material space is that the resulting reparametrization is smooth, as can be seen in Figure 9(c); this might prove useful with smooth spatial variations of material parameters.

5 Discussion and future work

We have presented a simple approach for the reparametrization of material appearance, and demonstrated it on three types of material effects commonly encountered in Computer Graphics. In each case, we have shown that our approach provides a finer control over material appearance.

In the case of roughness, we have limited our analysis to an isotropic model, and did not consider multiple reflections among micro-facets. For anisotropic distributions, we simply suggest to reuse the roughness reparametrizations of Section 3.2 along each of the two surface tangent directions. We have considered providing a different remapping function in cases where multiple scattering is considered. However, we have found that with the approach of Turquin [Tur18], this does not change the material difference function δ, and hence the reparametrization. Indeed, the corrective term in his Equation 16 is only dependent on the incident direction, which we assume to be constant in our approach. Hence this term becomes a multiplicative constant, which vanishes in Equation 5.

An alternative approach to compute automatically material differences (either in material or image space) is to rely on artists themselves. This is the approach taken by Chiang et al. [CBTB16], who let artists directly provide the resampled parameter values $\{p'_i\}$ (red ticks in Figure 2b) by letting them order and space a few renderings made with different values of a single material parameter. Compared to our approach, this amounts to provide directly a discretized version of the inverse CDF D^{-1}. The fitted reparametrization function does not have to contain many coefficients since the artist-provided sampling is sparse to start with.

In future work, we would like to investigate appearance reparametrizations on more complex materials. Complexity may appear with material models that produce more than one BRDF lobe. This is the case of layered materials (e.g., [Bel18]), but also of translucent materials (e.g., [GXZ+13]). In that latter case, an image-based approach might become necessary due to the very different appearance one observes between front- and back-lighting conditions. Another dimension of complexity includes spatial variations, which pose an even more difficult challenge, since the material difference function should also consider texture properties.

References

[Bel18] Laurent Belcour. Efficient rendering of layered materials using an atomic decomposition with statistical operators. *ACM Trans. Graph.*, 37(4), July 2018.

[Bur12] Brent Burley. Physically-based shading at disney. In *ACM SIGGRAPH*, volume 2012, pages 1–7. vol. 2012, 2012.

[CBTB16] Matt Jen-Yuan Chiang, Benedikt Bitterli, Chuck Tappan, and Brent Burley. A practical and controllable hair and fur model for production path tracing. In *Computer Graphics Forum*, volume 35, pages 275–283. Wiley Online Library, 2016.

[CK17] Alejandro Conty and Christopher Kulla. Production friendly microfacet sheen brdf, 2017.

[CT81] Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *SIGGRAPH Comput. Graph.*, 15(3):307–316, August 1981.

[Gul14] Ole Gulbrandsen. Artist friendly metallic fresnel. *Journal of Computer Graphics Techniques (JCGT)*, 3(4):64–72, December 2014.

[GXZ+13] Ioannis Gkioulekas, Bei Xiao, Shuang Zhao, Edward H. Adelson, Todd Zickler, and Kavita Bala. Understanding the role of phase function in translucent appearance. *ACM Trans. Graph.*, 32(5):147:1–147:19, October 2013.

[Hei14] Eric Heitz. Understanding the masking-shadowing function in microfacet-based brdfs. *Journal of Computer Graphics Techniques (JCGT)*, 3(2):48–107, June 2014.

[Jak10] Wenzel Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org.

[LMS+19] Manuel Lagunas, Sandra Malpica, Ana Serrano, Elena Garces, Diego Gutierrez, and Belen Masia. A similarity measure for material appearance. *ACM Transactions on Graphics (SIGGRAPH 2019)*, 38(4), 2019.

[MPBM03] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Transactions on Graphics*, 22(3):759–769, July 2003.

[NDM06] Addy Ngan, FrÃ©do Durand, and Wojciech Matusik. Image-driven Navigation of Analytical BRDF Models. In Tomas Akenine-Moeller and Wolfgang Heidrich, editors, *Symposium on Rendering*. The Eurographics Association, 2006.

[PFG00] Fabio Pellacini, James A. Ferwerda, and Donald P. Greenberg. Toward a psychophysically-based light reflection model for image synthesis. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 55–64, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[SGM+16] Ana Serrano, Diego Gutierrez, Karol Myszkowski, Hans-Peter Seidel, and Belen Masia. An intuitive control space for material appearance. *ACM Transactions on Graphics (SIGGRAPH ASIA 2016)*, 35(6), 2016.

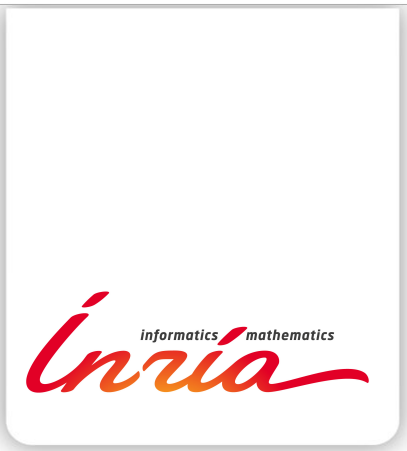
[TR75] T. S. Trowbridge and K. P. Reitz. Average irregularity representation of a rough surface for ray reflection. *J. Opt. Soc. Am.*, 65(5):531–536, May 1975.

[TS67] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces∗. *J. Opt. Soc. Am.*, 57(9):1105–1114, Sep 1967.

[Tur18] Emmanuel Turquin. Practical multiple scattering compensation for microfacet models, 2018.

[WAKB09] Josh Wills, Sameer Agarwal, David Kriegman, and Serge Belongie. Toward a perceptual space for gloss. *ACM Trans. Graph.*, 28(4), September 2009.

[WMLT07] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, pages 195–206, Goslar, DEU, 2007. Eurographics Association.



RESEARCH CENTRE
BORDEAUX – SUD-OUEST

200 avenue de la Vieille Tour
33405 Talence Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399