BRDF Fitting Using Inverse Global Illumination and Stochastic Optimization*

Gustavo Pfeiffer, Ricardo Marroquim
Laboratório de Computação Gráfica (LCG) – COPPE – UFRJ

***Abstract*—We present a comprehensive method for recovering bidirectional reflectance distribution functions (BRDFs) of objects with known geometry from photographs. Most existing image-based methods consider only direct illumination, a simplification which might be inappropriate for highly specular objects or objects with significant non-convex geometric features. Meanwhile, algorithms that consider indirect illumination often use iterative methods that alternate between solving an approximation of the problem and a series of "black-box" optimizations. We present a different approach that considers indirect illumination and uses derivative-based optimization. Our method uses stochastic ray tracing to compare photographs and renderings, thus leading to a stochastic objective function, i.e., an objective function that cannot be measured precisely. We present an unbiased estimator for the derivative of the ray-traced images with respect to the BRDF parameters and stochastic optimization techniques for properly solving the proposed optimization problem.**

***Keywords*-BRDF fitting; inverse global illumination; stochastic optimization; stochastic ray tracing**

I. Introduction and related work

At present, bidirectional reflectance distribution functions (BRDFs) provide us the most complete model of light reflection at a macroscopic observation level. They describe the visual appearance of a surface point by modeling how light interacts with it (or roughly speaking, they describe the material from which the surface is made), and can be conveniently measured from an actual object. However, since BRDFs are four-dimensional functions, thorough measurement can be extremely laborious. It is often preferred to fit the BRDF to a parametric equation, which may derive from a physically-based model (such as the Oren-Nayar model [1] or the He-Torrance model [2]), from a mathematical model designed for fitting (such as Lafortune's model [3]), or even from a linear combination of several measured BRDFs [4].

Image-based BRDF fitting is an efficient way of measuring a BRDF, since a few photographs often provide sufficient data for an accurate fit. Several techniques have been presented, including the ones by Marschner et al. [5], Lensch et al. [6], and Sato et al. [7] [8]. Standardly, methods suppose we already know the geometry (often acquired from 3D scanner data), the camera position and rotation (which may be recovered aligning photograph silhouettes [9]) and the light source position (by for instance analyzing specular highlight [6]).

However, a deficiency of the methods above is that they only consider direct illumination. This significantly simplifies the optimization method, but may fail in cases where this simplification is not appropriate — for instance, when the geometry of the object causes interreflection spots, specially when the surface is mostly specular, or when a physical setup for isolating surrounding surfaces was not properly done or is not even possible. We present therefore a BRDF fitting method that takes into account global illumination (i.e., considers multiple reflections). Stochastic ray tracing is used to evaluate the objective function, which consequently turns the objective function stochastic, characterizing one of the major challenges of the proposed method.

Our method solves the Inverse Global Illumination problem, which has been solved by [10] [11] [12], among others. Approaches for this problem however are limited to specific BRDFs and often resort to iterative algorithms that alternate between solving an approximated model and a series of "black-box" optimizations. On the other hand, as we developed a dedicated ray tracer that computes the rendered image and its Jacobian matrix with respect to the unknown parameters, we can use a more comprehensive, derivative-based optimization approach.

Our main contributions are:

- A comprehensive method for fitting BRDFs, taking into account mutual illumination. In addition, it can be used to fit emission distribution functions of light sources, and it also solves the inverse global illumination problem.
- A few optimization strategies that, although designed for this problem, are generic enough to be applied to other nonlinear least-squares stochastic problems.

II. Initial mathematical formulation and notation

Our problem consists in finding the BRDF parameters that characterize the surface of the object, given a set of photographs of an object, the position and rotation of the camera in each photograph (i.e., its extrinsic and intrinsic matrices; distortions are disregarded), the position of all light sources, and the geometry of the object.

We model it as a nonlinear least squares problem that compares each photograph with a rendering of the scene. We would like to find p that minimizes the following objective function:

$$f(p) = \frac{1}{2}||R_*(p) - \bar{R}_*||^2 = \frac{1}{2}\left\| \begin{bmatrix} R_1(p) \\ ... \\ R_{N_\phi}(p) \end{bmatrix} - \begin{bmatrix} \bar{R}_1 \\ ... \\ \bar{R}_{N_\phi} \end{bmatrix} \right\|^2$$

where:

** This document is a revised version of the manuscript formerly sumbitted to SIBGRAPI-WUW 2012.*

- $p \in \mathbb{R}^{N_p}$ is a vector of the N_p parameters we want to obtain, usually the BRDF parameters. It may also include, for instance, light source intensity and albedo textures.
- N_ϕ is the number of photographs multiplied by 3 (supposing 3 color channels, i.e., RGB). We render each channel separately, as if they were different images, so, for convenience, we will often use the terms "photograph" and "image" to refer to only one channel.
- $R_i(p) \in \mathbb{R}^{N_{i,x} \times N_{i,y}}$ is the image correspondent to a perfect rendering (without noise) of the scene for parameters p, where $N_{i,x}$ and $N_{i,y}$ are the dimensions of the photograph i.
- $\bar{R}_i$ is the i-th photograph (target image).
- The notation R_* indicates a concatenation of the R_i.

In the following sections this objective function will suffer modifications due to optimization techniques, but as of now we keep this simplified formulation.

The renderings $R_i(p)$ cannot be computed exactly, algorithms such as stochastic ray tracing compute otherwise unbiased estimates $\widehat{R}_i$ of $R_i(p)$, i.e., that satisfy $\mathrm{E}[\widehat{R}_i|p] = R_i(p)$.

For simplicity, we use the $\widehat{\ }$ notation to indicate that a random variable is generated given another random variable. We write $\widehat{F}(x)$ when $E[\widehat{F}|x] = F(x)$ for a given function $F(x)$. Note that $\widehat{F}(x)$ is not a function, it merely represents this relationship between random variables $\widehat{F}$ and x. Also, we use the symbol $'$, as in $\widehat{F}(x)$ and $\widehat{F}'(x)$ to indicate when two measurements are independent, i.e., $\mathrm{pdf}[\widehat{F}, \widehat{F}'|x] = \mathrm{pdf}[\widehat{F}|x]\mathrm{pdf}[\widehat{F}'|x]$.

In order to choose a search direction in the parameter space, we use a derivative-based method, thus we need to be able to compute an unbiased estimator $\widehat{\partial_p R_i}(p)$ of the Jacobian $\partial_p R_i(p)$. Note that $\widehat{\partial_p R_i}(p)$ is not the derivative of $\widehat{R}_i(p)$, since $\widehat{R}_i(p)$ is not a function of p, but $E\left[\widehat{\partial_p R_i}\middle| p\right] = \partial_p R_i(p)$. In the next section we show how this derivative can be estimated.

III. The ray tracer derivative

A. Mathematical model of the path tracer

For the ray tracer, we use a path tracer with next event estimation (following the terminology from [13]), i.e., rays are traced from the eye to the light, and direct illumination is computed separately from indirect illumination. A pixel $\widehat{R}_{i,x,y}$ of image i in coordinates (x, y) can be written as

$$\widehat{R}_{i,x,y} = \frac{1}{N_{\text{samples}}} \sum_{j=1}^{N_{\text{samples}}} \widehat{R}_{i,x,y,j}$$

where $\widehat{R}_{i,x,y,j}$ is a sample.

From Kajiya's rendering equation [14], a point x of a surface satisfies:

$$L_{\text{out}}(x, \omega_{\text{out}}, p) = L_e(x, \omega_{\text{out}}, p) + \int_\Omega f_r(x, \omega_{\text{in}}, \omega_{\text{out}}, p)\langle \omega_{\text{in}}, \hat{n}\rangle L_{\text{in}}(x, \omega_{\text{in}}, p) d\omega_{\text{in}}$$

where f_r is the BRDF, p are the BRDF parameters, $\hat{n}$ is the normal vector at x, L_{in} and L_{out} are the radiance functions entering and exiting the surface point from directions ω_{in} and ω_{out}, L_e is the emitted radiance, and Ω is the hemisphere of entering directions.

We separate the direct and indirect illumination components:

$$\begin{aligned} L_{\text{out}}(x, \omega_{\text{out}}, p) = L_e(x, \omega_{\text{out}}, p) + \\ \int_\Omega f_r(x, \omega_{\text{in}}, \omega_{\text{out}}, p)\langle \omega_{\text{in}}, \hat{n}\rangle L_{\text{direct}}(x, \omega_{\text{in}}, p) d\omega_{\text{in}} \\ + \int_\Omega f_r(x, \omega_{\text{in}}, \omega_{\text{out}}, p)\langle \omega_{\text{in}}, \hat{n}\rangle L_{\text{indirect}}(x, \omega_{\text{in}}, p) d\omega_{\text{in}} \end{aligned}$$

where L_{direct} anticipates the value of L_e coming directly from a light source, while L_{indirect} refers to light that has been reflected by at least one surface before reaching x.

Using the Monte Carlo integration technique, an unbiased estimator $\widehat{y}$ for a function $y(p) = \int z(x,p)dx$ can be generated as follows:

$$\begin{aligned} E[\widehat{y}|p] = y(p) &= \int z(x,p)dx = \int \frac{z(x,p)}{\mathrm{pdf}[x|p]}\mathrm{pdf}[x|p]dx \\ &= E\left[\left.\frac{z(x,p)}{\mathrm{pdf}[x|p]}\right| p\right] \end{aligned} \tag{1}$$

$$\widehat{y}(p) := \frac{z(x,p)}{\mathrm{pdf}[x|p]} \tag{2}$$

where $\mathrm{pdf}[x|p]$ can be conveniently chosen and must be non-zero where $z \neq 0$.

Thus, we compute $\widehat{R}_{i,x,y,j}$ recursively, using:

$$\begin{aligned} \widehat{R}_{i,x,y,j} &= L_e(x, \omega_{\text{out}}, p) + \widehat{R}_{i,x,y,j,0} \\ \widehat{R}_{i,x,y,j,k} &= s_{i,x,y,j,k}\widehat{E}_{i,x,y,j,k} + \widehat{r}_{i,x,y,j,k}\widehat{R}_{i,x,y,j,k+1} \end{aligned}$$

where

$$\begin{aligned} s &= f_r(x, \omega_{\text{direct}}, \omega_{\text{out}}, p)\langle \omega_{\text{direct}}, \hat{n}\rangle \\ \widehat{E} &= \frac{L_{\text{direct}}(x, \omega_{\text{direct}}, p)}{\mathrm{pdf}[\omega_{\text{direct}}|p]} \\ \widehat{r} &= \frac{f_r(x, \omega_{\text{indirect}}, \omega_{\text{out}}, p)\langle \omega_{\text{indirect}}, \hat{n}\rangle}{\mathrm{pdf}[\omega_{\text{indirect}}|p]} \end{aligned}$$

We remark that $\widehat{E}$ is sampled in the light source position domain ($\mathrm{pdf}[x_{\text{light}}]$), and not in the hemisphere of reflection ($\mathrm{pdf}[\omega_{\text{direct}}]$), yielding (in case of area light sources):

$$\widehat{E} = \frac{\max\{\langle -\omega_{\text{direct}}, \hat{n}_{\text{light}}\rangle, 0\} \cdot L_{\text{light}}(x_{\text{light}}, -\omega_{\text{direct}}, p)}{\mathrm{pdf}[x_{\text{light}}|p] \cdot ||x_{\text{light}} - x||^2}$$

$$\text{where } \omega_{\text{direct}} = \frac{x_{\text{light}} - x}{||x_{\text{light}} - x||}$$

Also, in case the ray does not hit any surface, we set $\widehat{R}_{i,x,y,j,k} = 0$, which ends the recursion. Similarly, $\widehat{E}$ is set to zero if there is another surface blocking the light in direction ω_{direct}.

B. Derivative of the path tracing recursion

The derivative of the path tracing recursion may be computed recursively using:

$$\widehat{\partial_p R}_{i,x,y} = \sum_{j=1}^{N_{\text{samples}}} \frac{\widehat{\partial_p R}_{i,x,y,j}}{N_{\text{samples}}}$$
$$\widehat{\partial_p R}_{i,x,y,j} = \widehat{\partial_p L_e}(x, \omega_{\text{out}}, p) + \widehat{\partial_p R}_{i,x,y,j,0}$$

$$\widehat{\partial_p R}_{i,x,y,j,k} = \partial_p s_{i,x,y,j,k} \widehat{E}_{i,x,y,j,k} + s_{i,x,y,j,k} \widehat{\partial_p E}_{i,x,y,j,k} + \widehat{\partial_p r}_{i,x,y,j,k} \widehat{R}_{i,x,y,j,k+1} + \widehat{r}_{i,x,y,j,k} \widehat{\partial_p R}_{i,x,y,j,k+1}$$
$$\text{or } \widehat{\partial_p R}_{i,x,y,j,k} = 0 \text{ if the ray does not hit any surface}$$

In order to avoid summing up and stacking gradient vectors such as $\widehat{\partial_p R}_{i,x,y,j,k}$ and $\widehat{\partial_p r}_{i,x,y,j,k}$, it is preferrable to compute $\widehat{\partial_p R}_{i,x,y}$ incrementally, by starting with $\widehat{\partial_p R}_{i,x,y} = 0$ and summing the terms separately:

$$\begin{aligned}
q_{i,x,y,j,0} &= \frac{1}{N_{\text{samples}}} \\
q_{i,x,y,j,k+1} &= \widehat{r}_{i,x,y,j,k} q_{i,x,y,j,k} \\
\widehat{\partial_p R}_{i,x,y} &\mathrel{+}= q_{i,x,y,j,0} \widehat{\partial_p L_e}(x, \omega_{\text{out}}, p) \\
\widehat{\partial_p R}_{i,x,y} &\mathrel{+}= q_{i,x,y,j,k} \widehat{E}_{i,x,y,j,k} \widehat{\partial_p s}_{i,x,y,j,k} \\
\widehat{\partial_p R}_{i,x,y} &\mathrel{+}= q_{i,x,y,j,k} s_{i,x,y,j,k} \widehat{\partial_p E}_{i,x,y,j,k} \\
\widehat{\partial_p R}_{i,x,y} &\mathrel{+}= q_{i,x,y,j,k} R_{i,x,y,j,k+1} \widehat{\partial_p r}_{i,x,y,j,k} \qquad (3)
\end{aligned}$$

where the notation $A \mathrel{+}= B$ should be read as $A \leftarrow A + B$.

In addition, since eq. 3 requires knowing the value of $\widehat{R}_{i,x,y,j,k+1}$, this parcel is only summed when returning from the recursion.

C. Derivative of s, r, and E

The derivative of s is straightforward:

$$\partial_p s = \langle \omega_{\text{direct}}, \hat{n} \rangle \partial_p f_r(x, \omega_{\text{direct}}, \omega_{\text{out}}, p)$$

On the other hand, r has the particularity that the pdf of ω_{in} may change with p, so we have to review the Monte Carlo integration.

Using the same technique shown in Equations 1 and 2 to generate an estimator $\widehat{y}$ of a function $y(p)$, we can compute an unbiased estimator for its partial derivative $\partial_p y$ as follows:

$$E[\widehat{\partial_p y}|p] = \partial_p y(p) = \int \partial_p z(x,p) dx = \int \frac{\partial_p z(x,p)}{\text{pdf}[x|p]} \text{pdf}[x|p] dx = E\left[\frac{\partial_p z(x,p)}{\text{pdf}[x|p]} \middle| p \right]$$
$$\widehat{\partial_p y} := \frac{\partial_p z(x,p)}{\text{pdf}[x|p]}$$

We use the same pdf for solving the direct problem $\widehat{R}_*(p)$ and the derivative $\widehat{\partial_p R}_*(p)$, though this is arguable, since probably the most adequate pdf for sampling a function is not the same as for sampling its derivative.

Now we can calculate the r term derivative:

$$\widehat{\partial_p r} = \frac{\langle \omega_{\text{indirect}}, \hat{n} \rangle \partial_p f_r(x, \omega_{\text{indirect}}, \omega_{\text{out}}, p)}{\text{pdf}[\omega_{\text{indirect}}|p]}$$

The derivative of $\widehat{E}$ follows the same principle, though the pdf does not usually change with p, since uniform distribution is often used:

$$\widehat{\partial_p E} = \frac{\max\{\langle -\omega_{\text{direct}}, \hat{n}_{\text{light}} \rangle, 0\} \cdot \partial_p L_{\text{light}}(x_{\text{light}}, -\omega_{\text{direct}}, p)}{\text{pdf}[x_{\text{light}}|p] \cdot ||x_{\text{light}} - x||^2}$$

IV. Optimization

A. Choice of direction

In the deterministic approach, one could use the Gauss-Newton method, which chooses a search direction that minimizes:

$$\min_{\delta p} ||R_*(p) + \partial_p R_*(p) \delta p - \bar{R}_*||^2$$

resulting in

$$\delta p = (\partial_p R_*(p)^T \partial_p R_*(p))^{-1} \partial_p R_*(p)^T (\bar{R}_* - R_*(p))$$

In the stochastic setting, a similar method can be applied:

$$\widehat{\delta p} = (\widehat{\partial_p R_*}''(p)^T \widehat{\partial_p R_*}''(p))^{-1} \widehat{\partial_p R_*}'(p)^T (\bar{R}_* - \widehat{R_*}(p)) \quad (4)$$

where $\widehat{R_*}(p)$, $\widehat{\partial_p R_*}'(p)$ and $\widehat{\partial_p R_*}''(p)$ are independent measurements. This choice is a biased estimate of δp from the Gauss-Newton method since:

$$\begin{aligned}
&E\left[\widehat{\delta p} \middle| p\right] \\
&= E\left[\left(\widehat{\partial_p R_*}^T \widehat{\partial_p R_*} \right)^{-1} \middle| p \right] E\left[\widehat{\partial_p R_*} \middle| p \right]^T E\left[\bar{R}_* - \widehat{R_*} \middle| p \right] \\
&= E\left[\left(\widehat{\partial_p R_*}^T \widehat{\partial_p R_*} \right)^{-1} \middle| p \right] \partial_p R_*(p)^T \left(\bar{R}_* - R_*(p) \right)
\end{aligned}$$

However, since

$$E\left[\widehat{\partial_p R_*}^T \widehat{\partial_p R_*} \middle| p \right] = \partial_p R_*(p)^T \partial_p R_*(p) + \sum_{i,x,y} \text{Var}\left[\widehat{\partial_p R}_{i,x,y} \middle| p \right]$$

we can expect that the variance term should have a regularizing effect, yielding a more cautious step $E[\widehat{\delta p}|p]$ than the one provided by the Gauss-Newton method (δp). Naturally, this is only valid under a number of hypothesis, including the fact that the number of pixels must be much greater than the number of parameters, and that $\widehat{\partial_p R_*}(p)^T \widehat{\partial_p R_*}(p)$ is invertible.

Nevertheless, we follow the often called Gauss-Newton-Krylov approach, which solves the system of eq. 4 only approximately, using a Krylov-space method such as linear conjugate gradients up to k iterations. Theoretically, this yields a solution that minimizes:

$$\min_{x \in \{b, Ab, A^2 b, \ldots, A^{k-1} b\}} (x - x^*)^T A (x - x^*)$$

where

$$
\begin{aligned}
&Ax^* = b \\
&b = \widehat{\partial_p R_*}'(p)^T(\bar{R}_* - \widehat{R_*}(p)) \\
&A = \widehat{B}''(p) = \widehat{\partial_p R_*}''(p)^T \widehat{\partial_p R_*}''(p) \\
&x = \widehat{\delta p}
\end{aligned}
$$

The value of k is controlled by a simple heuristic in which it starts at 1, increases gradually and decreases in case of step rejection (see Section IV-C).

We would also like to emphasize the importance of solving $\widehat{B}''(p)^{-1}\widehat{\partial_p R_*}'(p)^T(\bar{R}_* - \widehat{R_*}(p))$ instead of other alternatives.

For instance, if we use the same measurement for $\widehat{R}_*$, $\widehat{\partial_p R_*}$ and $\widehat{B}(p)$, i.e., solving instead $\widehat{B}(p)^{-1}\widehat{\partial_p R_*}(p)^T(\bar{R}_* - \widehat{R_*}(p))$, the direction generated will be severely biased toward decreasing variance, which is unwanted, since our ideal objective function uses only the expected values of the renderings. Effects of this alternative include decreasing the albedo of the objects in the scene in order to diminish the contribution of indirect illumination in the scene, as it is a significant source of variance in the rendered images.

On the other hand, if we use different measurements of $\widehat{B}$ in different iterations of the linear conjugate gradient method, we verified experimentally that convergence slows down drastically (in terms of the number of iterations).

B. Constraints

We use the barrier method from optimization in order to constrain variables (many BRDF parameters are constrained with $p_k > 0$ or $0 < p_k < 1$). Thus, we use $p := p(h)$, and apply the Gauss-Newton-Krylov process to h, which means solving:

$$
\widehat{\delta h} = \left(\widehat{\partial_h R_*}''(h)^T \widehat{\partial_h R_*}''(h)\right)^{-1} \widehat{\partial_h R_*}'(h)^T(\bar{R}_* - \widehat{R_*}(h))
$$

where we use the abuse of notation $R_*(h) = R_*(p(h))$ and $\partial_h R_*(h) = \partial_p R_*(p(h))\partial_h p$. Generally, we use $h_k = \log(p_k)$ for strictly positive variables and $h_k = \log(-\log(p_k))$ for variables between 0 and 1.

However, as we are using the Gauss-Newton-Krylov method, this technique is also useful to induce the optimizer to favor a certain direction. For instance, we use $h_k = \log(p_k)/3$ for the light intensity, where the factor $1/3$ was chosen experimentally. As a result, the gradient ∇f is intensified in the light direction, which, by its turn, is modified more frequently than, for instance, the albedo of the objects. This is desirable because changing the albedo before adjusting light intensity may drag the optimization process to a local minima or to a point which is hard to recover from.

C. Step acceptance

Some sort of line search or trust region strategy is essential to nonlinear optimization problems to avoid that the optimizer takes an excessive step and diverges. In the stochastic setting, this is more significant since a bad measurement may lead to a mistaken search direction. However, standard line search methods are prohibitive in stochastic optimization, since they become severely unstable. An otherwise more stable approach of line search would be to fit an approximation of the objective function in the line and minimize the fitted function; however, its stability comes from the supposition that a sufficiently large number of samples is given, which is costly.

We use instead a very simple heuristic of accepting the step or retroceding. We unbiasedly estimate $f(p(h))$ using two independent measurements:

$$
\widehat{f}(h) = \frac{1}{2}\langle \widehat{R}_*(h) - \bar{R}_*, \widehat{R}'_*(h) - \bar{R}_* \rangle
$$

Let h_t be the value of h in the t-th iteration, and $h_{t+1} = h_t + \widehat{\delta h_t}$. If $\widehat{f}(h_{t+1}) \in [\frac{1}{10}\widehat{f}(h_t), \frac{3}{2}\widehat{f}(h_t)]$, the step is accepted and $h \leftarrow h_{t+1}$. Otherwise, we retrocede $h \leftarrow h_{t-1}$. If h retrocedes twice or a previous value of h is not available, we refine resolution (see Section IV-D). If resolution is already maximum, optimization stops.

The explanation for this heuristic is as follows. A step can be considered "bad" in mainly two situations: when the step $\widehat{\delta h_t}$ is an outlier or when $\widehat{f}(h_{t+1})$ is an outlier much below $f(h_{t+1})$. The first case yields $\widehat{f}(h_{t+1}) >> \widehat{f}(h_t)$ and may be solved by retrying measuring $\widehat{\delta h_t}$. The second case, however, would make h_{t+1} be accepted and h_{t+2} be rejected with very high probability, which would be more efficiently solved by retroceding to h_t. We could combine retrying measuring $\widehat{\delta h}$ a few times and then retrocede h, however, as the second case is much more frequent than the first, we verified experimentally that not retrying measuring $\widehat{\delta h}$ is more efficient.

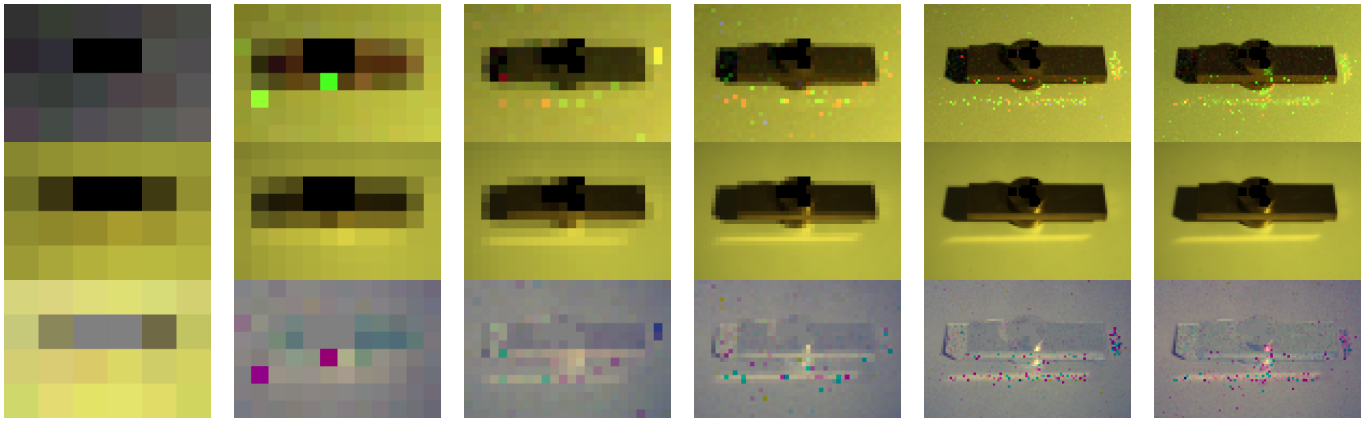
D. Refining resolution

Refining resolution means doubling the width and height of rendered images, and keeping the samples-per-pixel ratio. For performance issues, we do not start optimization at full resolution, but in fact at a very low resolution (See Figure 1). The target images are shrunk to the same size of the rendered resolution, and the objective function is changed to work with the number of pixels defined. This strategy reduces the variance of $\widehat{f}(h)/N_{\text{pixels}}$ (where $N_{\text{pixels}} = \sum_{i=1}^{N_\phi} N_{i,x}N_{i,y}$), which helps the step acceptance algorithm get better results. It does not change the bias of $\widehat{B}(p)/N_{\text{pixels}}$, although it diminishes its variance, which has a regularizing effect, as explained in Section IV-A. It also diminishes the variance of $\widehat{\partial_p R}'_*(p)^T\left(\widehat{R}_*(p) - \bar{R}_*\right)/N_{\text{pixels}}$.

V. Photograph setup

In order to avoid interference from external light sources, we take two pictures from the same viewpoint with our controlled light turned on and off and subtract them pixelwise. This is valid since the rendering problem is linear with respect to light intensity, though we have to turn off the Gamma filtering from photographs.

Additionally, in the photograph with the light on, we mark saturated pixels (with intensity above 250 out of 255), generating what we call the specular mask (see Figure 2). We do this

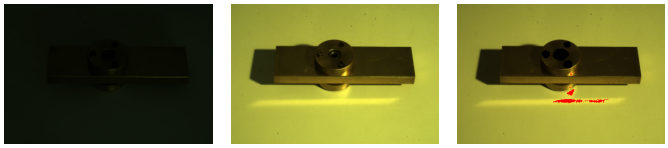


(a) At start of optimization (0s) (b) Refined once (45s). (c) Refined twice (2min35s). (d) Refined three times (18min50s). (e) Refined four times (1h2min). (f) Refined four times (5h30min).

Fig. 1. Optimization progress and resolution refining. Each subimage shows the rendering on top, the target image in the center and the difference in the bottom. Most noise is due to inefficiency in computing caustics.

to avoid, in specular highlight regions, that the rendered pixel is compared to a much lower value than it should be. We adapt our objective function to $f(h) = ||g(R_*(h)) - \bar{R}_*||^2$, where $g(\widehat{R}_{i,x,y}) = \max\{\widehat{R}_{i,x,y}, \bar{R}_{i,x,y}\}$ if (i, x, y) is marked in the specular mask or $\widehat{R}_{i,x,y}$ otherwise. This however introduces bias to all estimators (rendering, objective function and its gradient), which is not reduced when refining resolution.

We also use a custom mask, where pixels selected by the user can be removed from optimization (see Figure 2).



(a) Photograph with light source turned off (b) Photograph with light source turned on (c) Difference between the two and masks used. Custom mask is in black, specular mask in red.

Fig. 2. Photograph setup example.

VI. Initial guess

The initial guess p_0 for the optimization must be a somewhat balanced vector, for example, giving BRDFs the same weight for diffuse and specular components. Most variables such as albedo and Fresnel coefficients are started at 0.5, and light intensity is initially at a value that generates gray images (see Figure 1). The Gauss-Newton-Krylov approach helps the optimizer change first the directions that have a great contribution to the objective function. Thus, it first focuses on parameters such as light intensity and albedo, then on the weight of each BRDF in the linear combination of BRDFs used, and finally on more subtle parameters such as roughness and lobe position.

VII. Results

We applied our algorithm first to a synthetic case (Figure 3) then to a real object (Figure 4). In the synthetic case, geometry was exact and the object was mostly diffuse, yielding good results. Our real case, however, had its geometry measured manually and was highly specular. Misalignment of the geometry and light position caused the ghost caustic effect seen in the figure. In spite of the difficulties, results were better than the ones obtained when considering only direct illumination and deterministic rendering: From Figure 4, we note that the deterministic approach cannot distinguish the strong specular component of the material. We remark that the quality of the results was determined only by visual inspection.

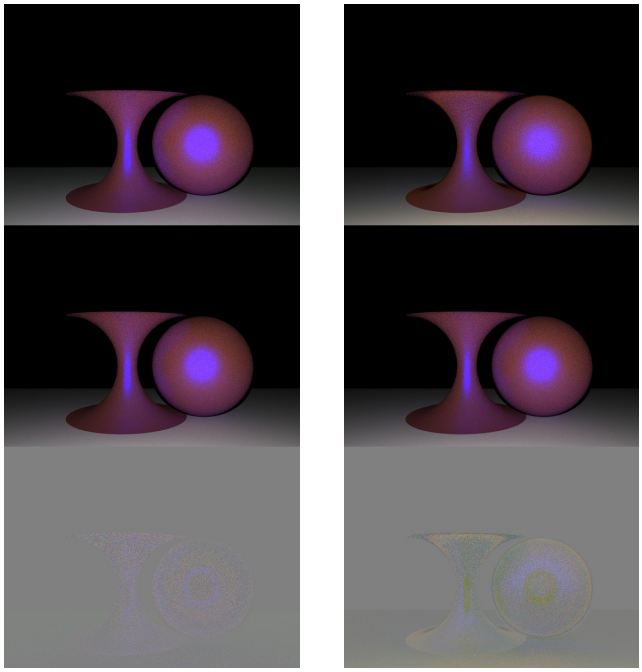
VIII. Limitations, conclusions and future work

The most notable limitation of the method is optimization time, since resolution or sampling must be refined in order to obtain more precise results. We have made efforts in order to increase the performance of our ray tracer, including using an octree-like data structure in OpenCL code for testing ray-surface intersections, but the cost is still unsatisfactory. However, we believe that if better rendering algorithms, sampling strategies and optimization preconditioning are chosen, this approach can potentially outperform other existent solutions to the inverse global illumination problem.

The fact that one cannot use a “black-box” ray tracer (since derivatives must be computed) can also be considered a limitation.

We observed that bad sampling conditions may make the stochastic optimization techniques used misbehave. For this reason, we consider, as future work, applying this method to bidirectional path tracing [13] or metropolis light transport [15]. Another important issue that we did not study is how to make a good sampling of the Jacobian, i.e., how to choose pdfs for derivatives of BRDFs, as mentioned in Section III-C.

Line search / trust region / step acceptance is still a difficulty and we would like to experiment other algorithms. We



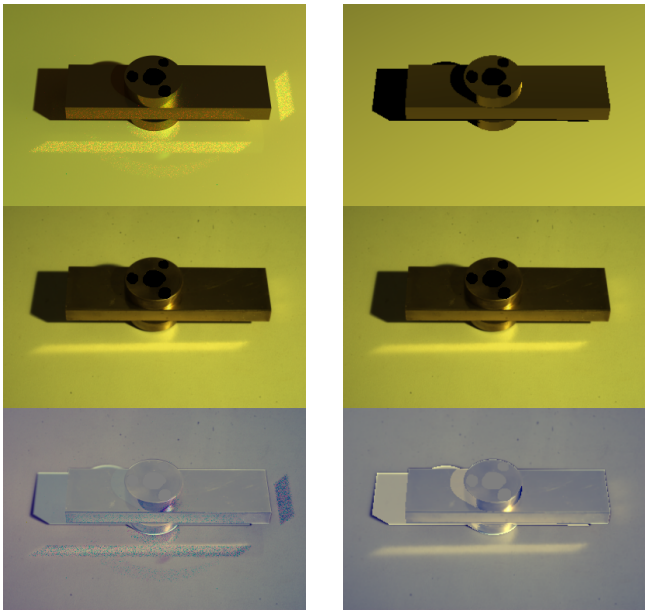
(a) Result obtained fitting the same BRDF models as the ones used to render the target image.

(b) Result obtained using fitting 3 Lafortune lobes for the diffuse component instead of Oren-Nayar.

Fig. 3. Results obtained from a synthetic scene. Target image was generated using a linear combination of Cook-Torrance and Oren-Nayar BRDF models. Each subimage shows the rendering on top, the target image in the center and the difference in the bottom.

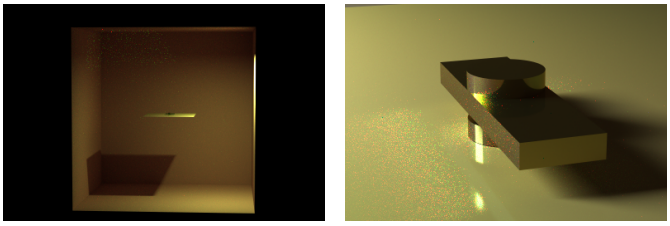
also consider changing the Gauss-Newton-Krylov approach to Levenberg-Marquardt or Quasi-Newton.

We would also like to experiment fitting albedo textures for spatially-varying materials.



(a) Result obtained using stochastic ray tracing (optimization time: about 26 hours). Most of the computation time is for computing caustics properly, since our ray tracer is not optimized for that.

(b) Result obtained using only direct illumination and deterministic rendering (optimization time: about 10 minutes)



(c) Surrounding surfaces model used.

(d) The object rendered from another angle and other lighting conditions.

Fig. 4. Results obtained from a real object. The BRDF model used was a linear combination of 3 Lafortune lobes, 2 Cook-Torrance lobes, and 1 Lambertian lobe for the metal piece and the table, and 1 Lambertian lobe for the surroundings.

References

[1] M. Oren and S. Nayar, "Generalization of Lambert's Reflectance Model," in *SIGGRAPH'94*, Jul 1994, pp. 239–246.

[2] X. D. He, K. E. Torrance, F. X. Sillion, and D. P. Greenberg, "A comprehensive physical model for light reflection," in *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH'91. New York, NY, USA: ACM, 1991, pp. 175–186. [Online]. Available: http://doi.acm.org/10.1145/122718.122738

[3] E. P. F. Lafortune, S. choong Foo, K. E. Torrance, and D. P. Greenberg, "Non-linear approximation of reflectance functions," 1997.

[4] W. Matusik, H. Pfister, M. Brand, and L. McMillan, "A data-driven reflectance model," *ACM TRANSACTIONS ON GRAPHICS*, vol. 22, pp. 759–769, 2003.

[5] S. R. Marschner, S. H. Westin, E. P. F. Lafortune, K. E. Torrance, and D. P. Greenberg, "Image-based brdf measurement including human skin."

[6] H. P. A. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H. peter Seidel, "Image-based reconstruction of spatially varying materials," in *In Proceedings of the 12th Eurographics Workshop on Rendering*, 2001, pp. 104–115.

[7] Y. Sato and K. Ikeuchi, "Reflectance analysis for 3d computer graphics model generation," in *Graphical Models and Image Processing*, 1996, pp. 437–451.

[8] Y. Sato, M. D. Wheeler, and K. Ikeuchi, "Object shape and reflectance modeling from observation," 1997.

[9] R. Marroquim, G. Pfeiffer, F. Carvalho, and A. A. F. Oliveira, "Texturing 3d models with low geometric features." in *SIBGRAPI*, 2011, pp. 1–8.

[10] Y. Yu, P. Debevec, J. Malik, and T. Hawkins, "Inverse global illumination: Recovering reflectance models of real scenes from photographs," in *SIGGRAPH'99*, 1999, pp. 215–224.

[11] S. Gibson, T. Howard, and R. Hubbold, "Flexible image-based photometric reconstruction using virtual light sources," *Computer Graphics Forum*, vol. 20, pp. 1067–7055, 2001.

[12] S. Boivin and A. Gagalowicz, "Image-based rendering of diffuse, specular and glossy surfaces from a single image," 2001.

[13] E. Lafortune, "Mathematical models and monte carlo algorithms for physically based rendering," Katholieke Universiteit Leuven, Tech. Rep., 1996.

[14] J. T. Kajiya, "The rendering equation," in *Computer Graphics*, 1986, pp. 143–150.

[15] E. Veach and L. J. Guibas, "Metropolis light transport," in *SIGGRAPH'97*. Addison Wesley, 1997, pp. 65–76.