BRDF Fitting using Inverse Global Illumination and Stochastic Optimization

XXV SIBGRAPI - Conference on Graphics, Patterns and Images
SIBGRAPI 2012
August 22-25, Ouro Preto, Brazil

Gustavo Pfeiffer, Ricardo Marroquim

Laboratório de Computação Gráfica (LCG) - COPPE - UFRJ

laboratório de computação gráfica
www.lcg.ufrj.br

Objective

• Find BRDF parameters of a surface, given photographs and geometry.

Previous Work

• Lensch et al., among others: use only direct illumination.

• Yu et al., among others: Use indirect illumination, but use "black-box" optimization methods and approximations.

Our Approach

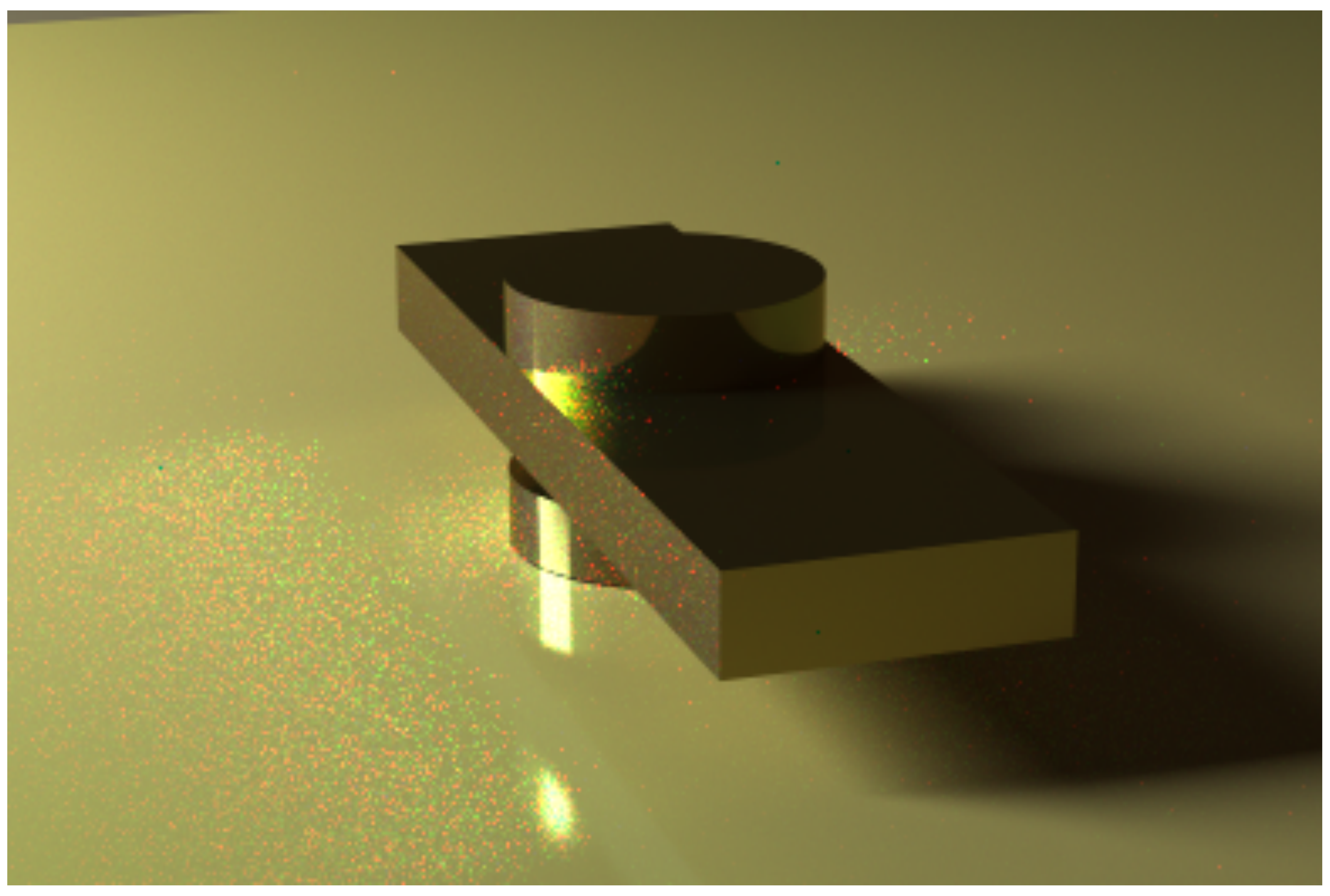
• Indirect Illumination, but derivative-based stochastic optimization.

Objective Function

$$\underbrace{f(p)}_{\text{BRDF parameters}} = \Big\|\underbrace{R_*(p)}_{\text{"perfect" renderings}} - \underbrace{\bar{R}_*}_{\text{photographs}}\Big\|^2$$

Obj. Function Estimator

$$\widehat{f}(p) = \langle \underbrace{\widehat{R_*}(p)}_{\text{stochastic ray-tracing renderings}} - \bar{R}_*, \widehat{R_*}'(p) - \bar{R}_* \rangle$$



A rendering of our recovered model, with different camera position and illumination setting from that of the photograph

Notation

Pseudorandom estimator of a function, usually unbiased (ex.: $\widehat{A}(B)$ satisfying $\mathrm{E}[\widehat{A}|B] = A(B)$).

Independence:
$\widehat{A}(p), \widehat{B}'(p) \rightarrow$ independent
$\widehat{A}''(p), \widehat{B}''(p) \rightarrow$ dependent

$$\widehat{\partial_p R_*}''(p)$$

Indicates **Jacobian**. $\widehat{\partial_B A}(B)$ estimates $\partial_B A(B)$

Concatenation, ex:

$$R_* = \begin{bmatrix} R_1 \\ \dots \\ R_n \end{bmatrix}$$

Search Direction

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

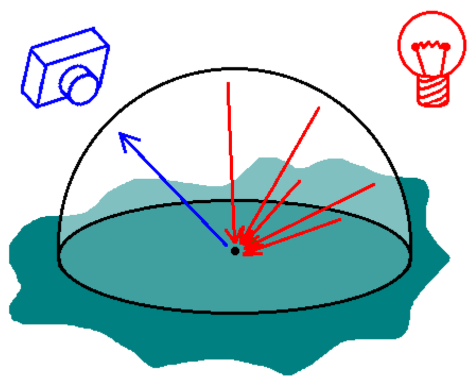
• Gauss-Newton-Krylov, linear conjugate gradients up to k iterations, variable k

Constraints and Preconditioning

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

$$p := p(h),\ \widehat{R_*}(h) = \widehat{R_*}(p(h)),\ \widehat{\partial_h R_*}(h) = \widehat{\partial_p R_*}\partial_h p$$

Monte-Carlo Derivative



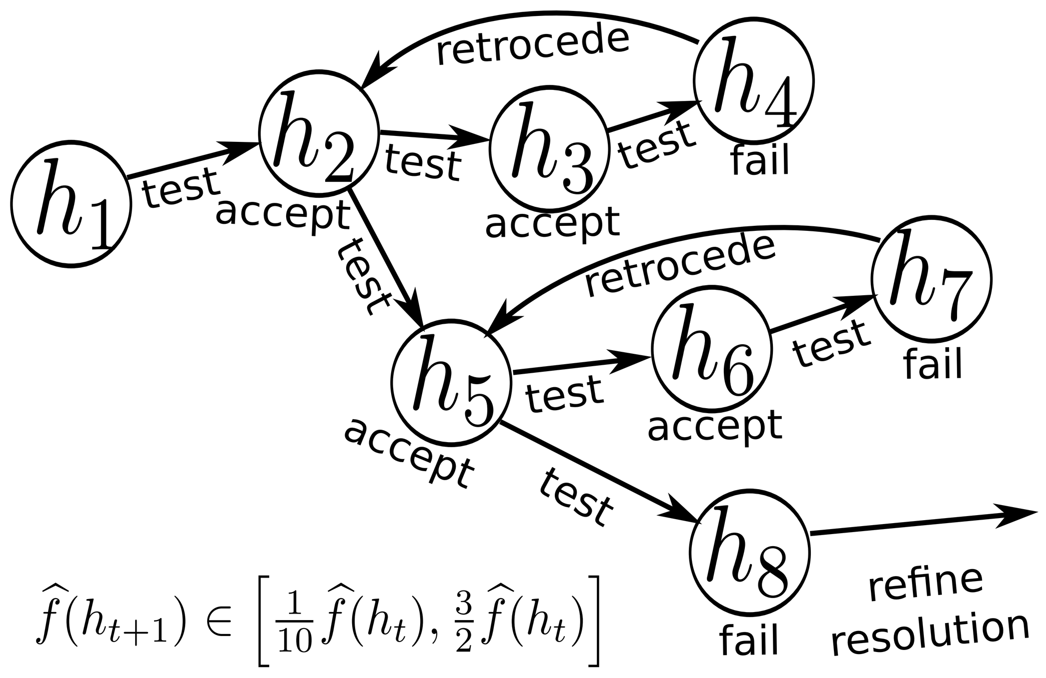
$$z(p) = \int y(x,p)dx$$

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

$$\partial_p z(p) = \int \partial_p y(x,p)dx$$

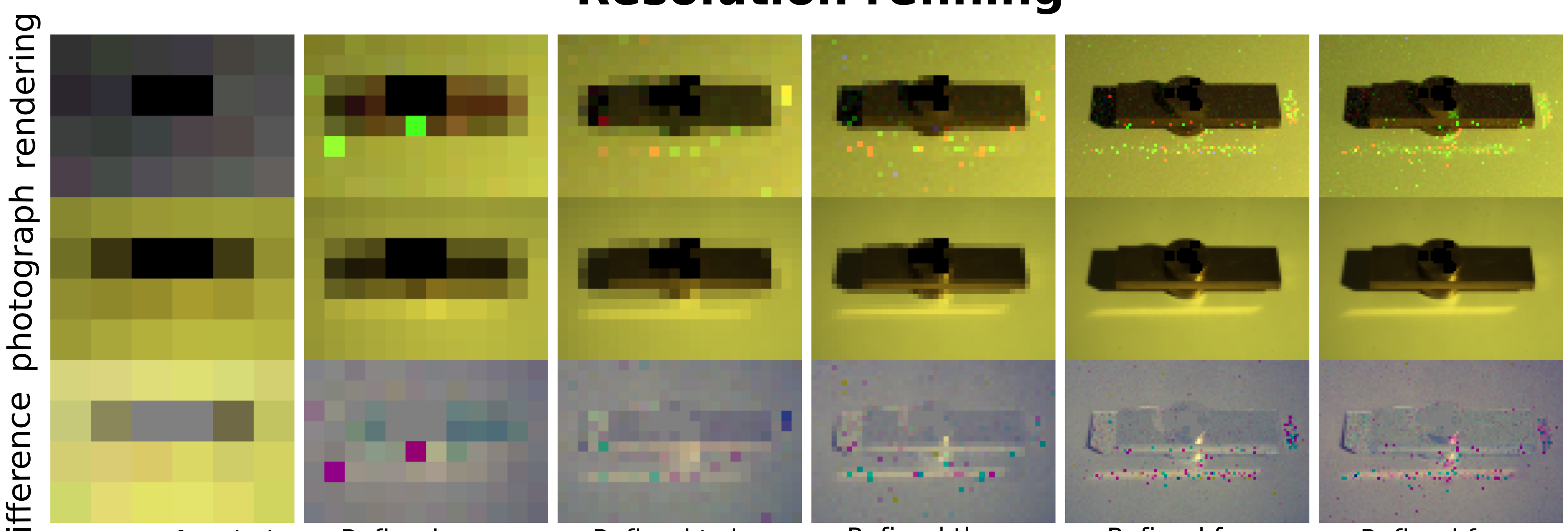
$$\widehat{\partial_p z}(p) := \frac{\partial_p y(x,p)}{\mathrm{pdf}[x|p]}$$

Step Acceptance



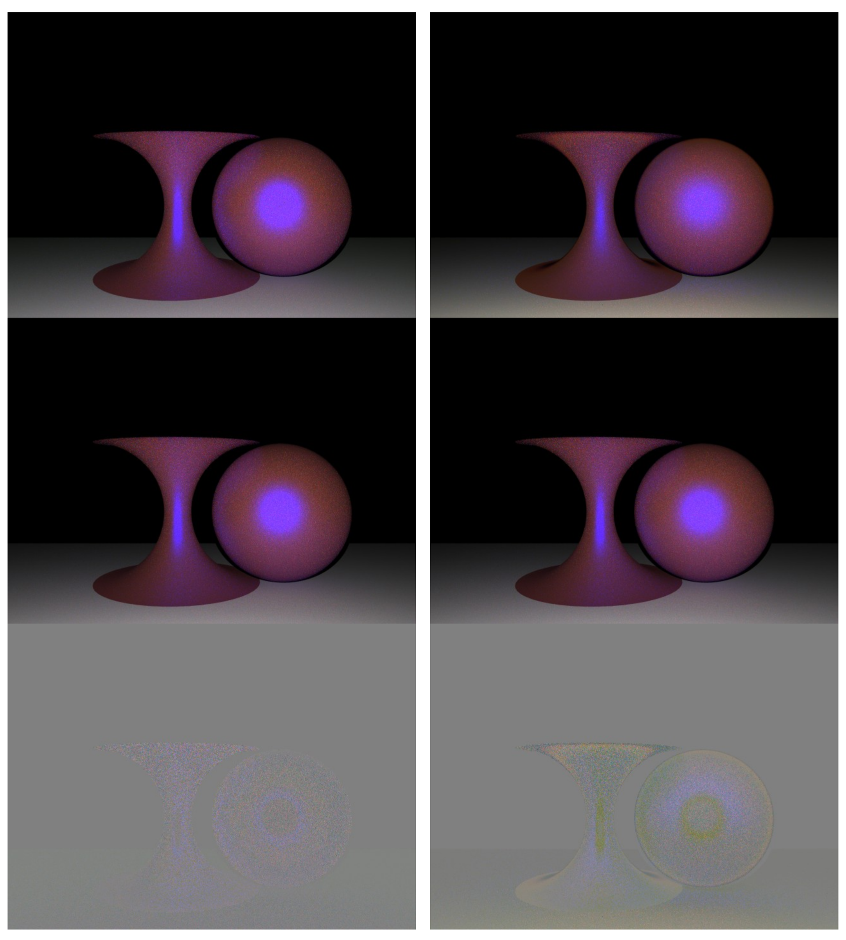
$$\widehat{f}(h_{t+1}) \in \left[\tfrac{1}{10}\widehat{f}(h_t), \tfrac{3}{2}\widehat{f}(h_t)\right]$$

Resolution refining



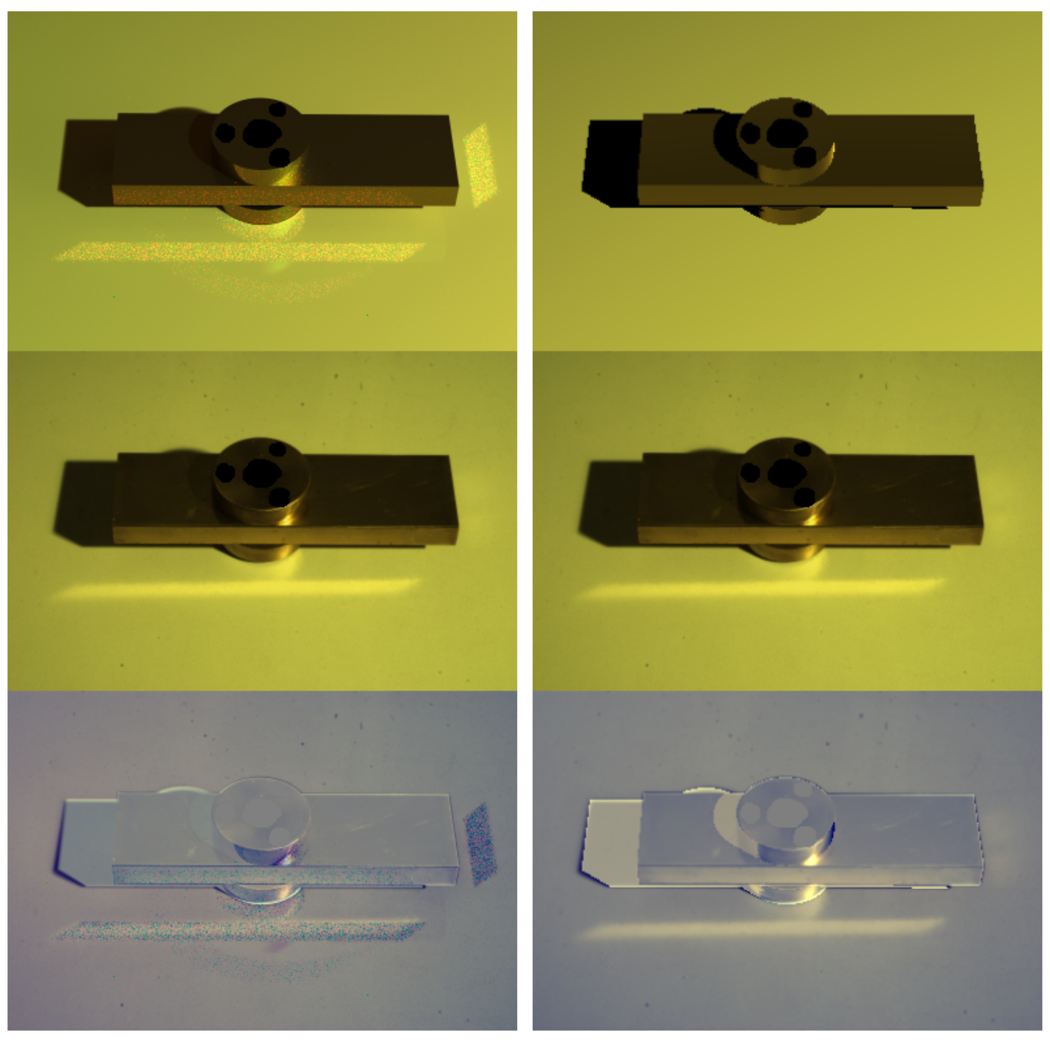
At start of optimization (0s). | Refined once (45s). | Refined twice (2min35s). | Refined three times (18min50s). | Refined four times (1h2min). | Refined four times (5h30min).

Results: Synthetic case



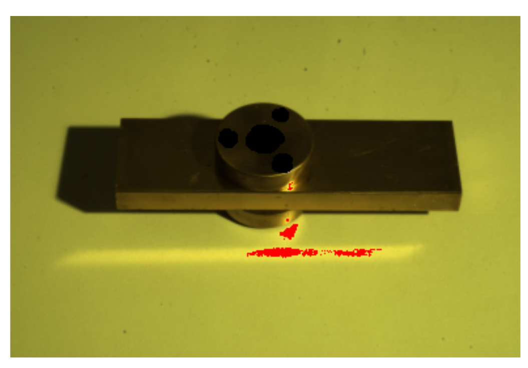
Same BRDF models for generating the target and for fitting | Different BRDF models

Results: Real case

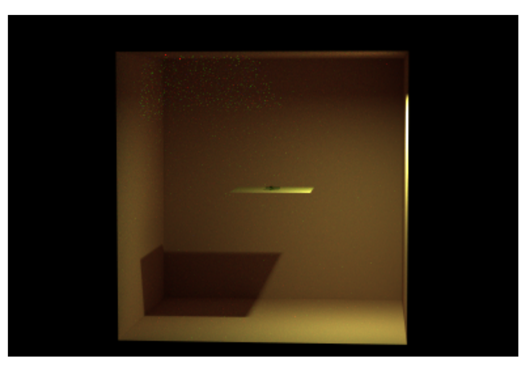


Our method (~26 hours) | Deterministic approximation (~10 minutes)

Photograph filters



Surrounding surfaces



Limitations, Conclusions, Future Work

• Optimization time

• GPU, BPT, MLT

• "Black-Box" rendering

• Choice of pdf for derivatives

• Trust region / step acceptance / line search, Quasi-Newton or Levenberg-Marquardt

• Textures