A Bayesian Inference Framework for Procedural Material Parameter Estimation

Yu Guo[1], Miloš Hašan[2], Lingqi Yan[3] and Shuang Zhao[1]

[1]University of California, Irvine [2]Adobe Research [3]University of California, Santa Barbara

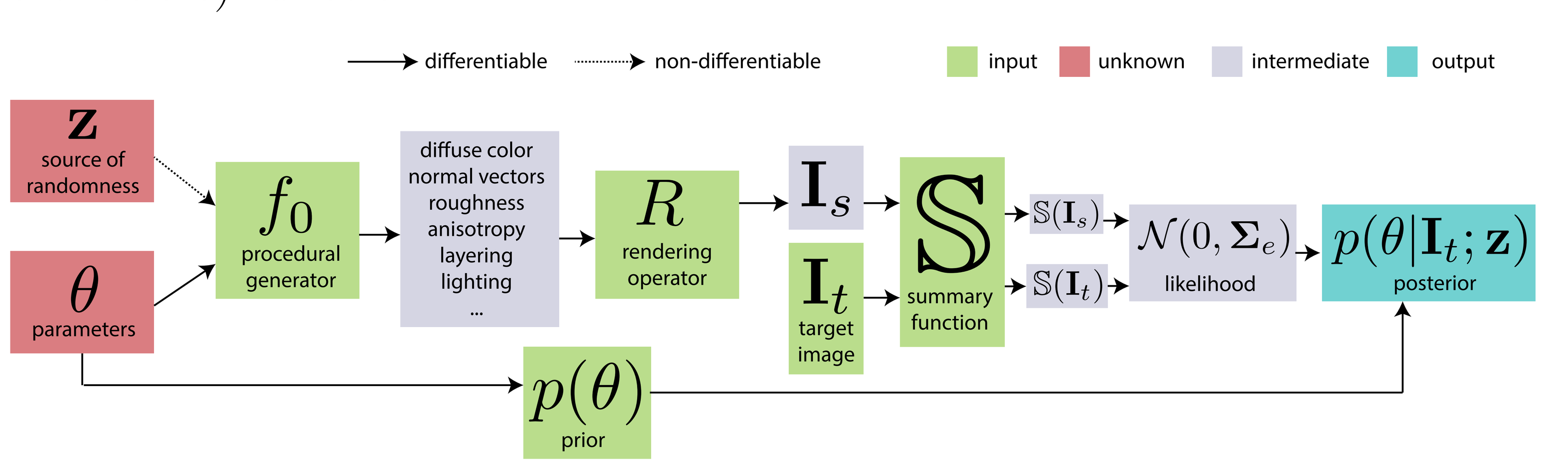
Introduction

Procedural material models have been graining traction in many applications thanks to their flexibility, compactness, and easy editability. In this paper, we explore the inverse rendering problem of procedural material parameter estimation from photographs using a Bayesian framework. We use *summary functions* for comparing unregistered images of a material under known lighting, and we explore both hand-designed and neural summary functions. In addition to estimating the parameters by optimization, we introduce a Bayesian inference approach using Hamiltonian Monte Carlo to sample the space of plausible material parameters, providing additional insight into the structure of the solution space. To demonstrate the effectiveness of our techniques, we fit procedural models of a range of materials—wall plaster, leather, wood, anisotropic brushed metals and metallic paints—to both synthetic and real target images (See Results section).

Figure 1: A scene rendered with material parameters estimated using our method: bumpy dielectrics, leather, plaster, wood, brushed metal, and metallic paint. The insets show a few examples of the initial flash photograph, and our procedural material with parameters found by posterior maximization.

Framework

We implement the procedural generation and rendering processes as (differentiable) PyTorch procedures with the priors and summary functions (classical or neural) expressed in the same framework. These four components (priors, procedural material model, rendering, summary function) together define our posterior distribution, a rather complex (but fully differentiable) function.



Methods

Forward evaluation:

$$f(\boldsymbol{\theta}; \boldsymbol{z}) := R(f_0(\boldsymbol{\theta}; \boldsymbol{z}))$$

Our forward evaluation contains procedural material generation f_0, and standard rendering process R. Extra random input z (e.g., random seeds, pre-generated noise textures, etc.) is used to create the irregularities.

Parameters estimation:

$$\text{find } \boldsymbol{\theta} \text{ s.t. } \boldsymbol{I}_t \approx f(\boldsymbol{\theta}; \boldsymbol{z}),$$

Point estimation:

$$\arg\min_{\boldsymbol{\theta}} \|\mathbb{S}(f(\boldsymbol{\theta}, \boldsymbol{z})) - \mathbb{S}(\boldsymbol{I}_t)\|^2.$$

MCMC: Hamiltonian Monte Carlo sampling[1, 2]

$$p(\boldsymbol{\theta}|\boldsymbol{I}_t, \boldsymbol{z}) \propto \mathcal{N}\left[\mathbb{S}(f(\boldsymbol{\theta}, \boldsymbol{z})) - \mathbb{S}(\boldsymbol{I}_t); 0, \boldsymbol{\Sigma}_e\right] p(\boldsymbol{\theta}),$$

Summary function:

An image summary function $\mathbb{S}$ is a continuous function that maps an image of a material ($\boldsymbol{I}_t$ or $\boldsymbol{I}_s$) into a vector in $\mathbb{R}^k$. An idealized summary function would have the property that

$$\mathbb{S}(f(\boldsymbol{\theta}_1, \boldsymbol{z}_1)) = \mathbb{S}(f(\boldsymbol{\theta}_2, \boldsymbol{z}_2)) \Leftrightarrow \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2.$$

Some techniques for constructing summary functions: Statistics of image bins; Fourier transforms; Neural summary function[3, 4]

Results (Synthetic data)

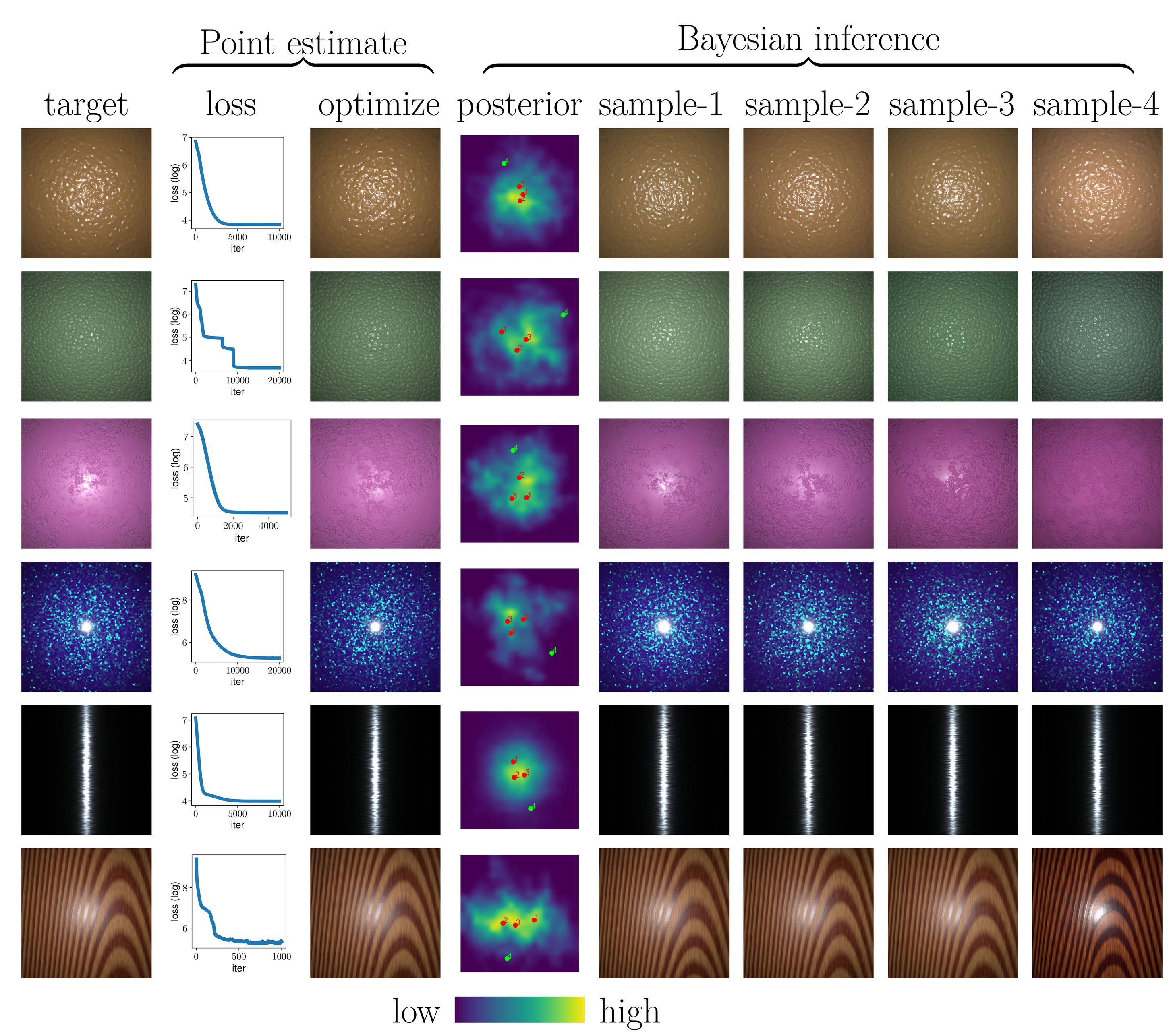


Figure 2: **Optimization and HMC sampling on synthetic images.** Each row corresponds to a different material. From top: bump, leather, plaster, metallic flake, brushed metal and wood.

Results (Real photo)

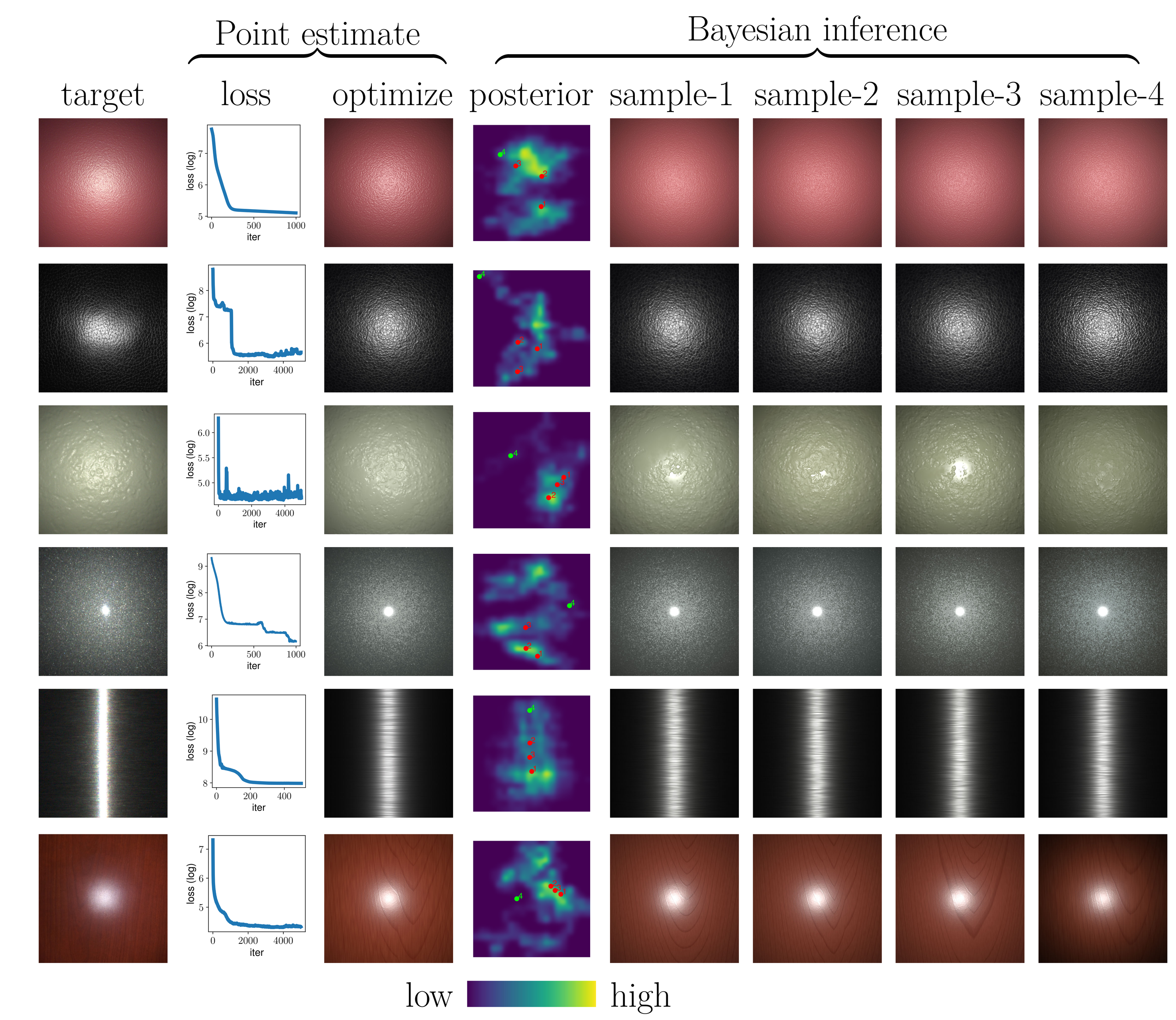


Figure 3: **Optimization and HMC sampling on real photos.** Each row corresponds to a different material. From top: bump, leather, plaster, and wood. Similar to Figure 2, except column 1 here contains real photos, columns 2 and 3 show point estimates (via non-linear optimization), and the remaining columns show HMC sampling results.

Contribution

The first major ingredient to our technique is a family of summary functions, from hand-crafted to neural-network based, that enable robust calculation of image differences (without requiring pixel-level alignments). The second ingredient is a Bayesian inference method that leverages Hamiltonian Monte Carlo (HMC) to sample posterior distributions of procedural material parameters.

References

[1] Radford M. Neal.
MCMC using Hamiltonian dynamics.
arXiv e-prints, page arXiv:1206.1901, Jun 2012.

[2] Michael Betancourt.
A conceptual introduction to hamiltonian monte carlo, 2017.
arxiv:1701.02434.

[3] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge.
A neural algorithm of artistic style, 2015.

[4] L. A. Gatys, A. S. Ecker, and M. Bethge.
Image style transfer using convolutional neural networks.
In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, June 2016.