Translucent Material Parameter Estimation

23.02.2023, Saip-Can Hasbay (01428723)

Outline

- Introduction: Essential ideas
- Problem and Goals
- Method
- Results
- Hopefully: Live example or a video

Introduction

Physics-Based Differentiable Rendering
A Comprehensive Introduction. SIGGRAPH
2020 Course, Zhao et al.[1]

Rendering

$$f(x) = y$$

“Inverse Rendering”

$$x = f^{-1}(y)$$

Scene description: geometry, materials, lights, etc.

Difficulties

- Physics-based (inverse) rendering
  - Scattering effects, complex materials, global illumination

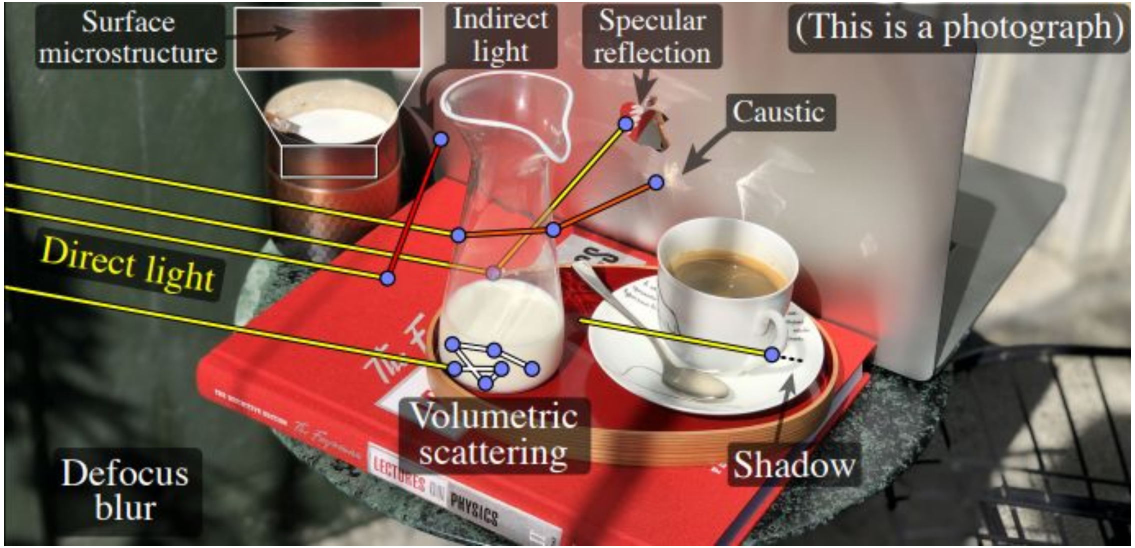
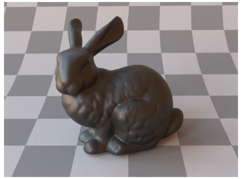
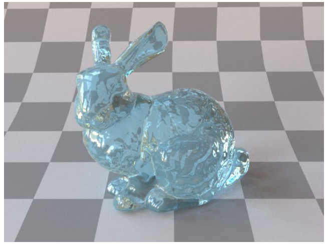


Image from Zhao et al. [1]

Objective Function

g() = ||  −  || ^{2}

Rendering

Target

minimize g(f(x))

Differentiable Rendering

x

f(x) = y

Matrix: Derivative of the rendering [2, 3, 4]

$$\delta g/\delta x = \delta y/\delta x \times \delta g/\delta y$$

$$g(y, y_{ref})$$

Problem and Goals (1)

- Main task: Translucent material parameter estimation
- But also:
    - A tool for inverse rendering
    - A gradient based optimization algorithm
    - A workflow for data acquisition
    - A naive approach for geometry and material reconstruction

Optimized material from synthetic (left) and real-world (right) alginate [5] data. Dragon model by Delatronic [14]

Problem and Goals (2)

Material parameters of interest:

(1) Disney Principled BSDF with integrated subsurface scattering [7, 8]
(2) Volumetric rendering: Rough dielectric BSDF [9] with homogeneous participating medium [6]

Roughness
Specular transmission
Index of refraction
Albedo
Extinction coefficient

$$x = \{c, \alpha, \sigma_s, \eta, \sigma_t\}$$

Problem and Goals (3)

- Estimate material parameters: $x = \{c, \alpha, \sigma_s, \eta, \sigma_t\}$
- By defining the task as an optimization problem

$$\min g(y(x)), \text{ s.t. } h(x) \leq 0,$$

where h defines additional constraints and g is either

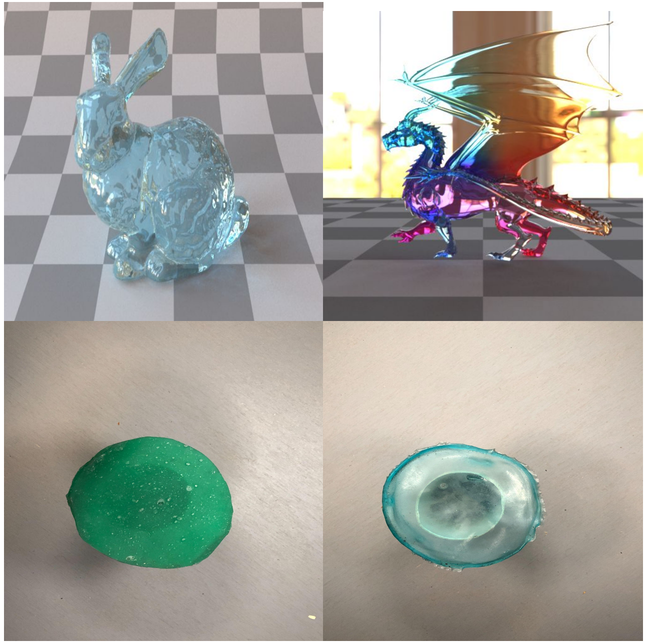
- the L2 norm $g(y) = ||y - y_{ref}||^2$, or
- the dual buffer method by Deng et al. [13]:

$$g(y_1, y_2) = (y_1 - y_{ref}) \cdot (y_2 - y_{ref})$$

Problem and Goals (4)

Requirements

(1) a (set of) reference image(s)
(2) a Mitsuba scene



Example reference images we used in our project. Top: Synthetic test cases. Bottom: Real-World alginate materials [5].

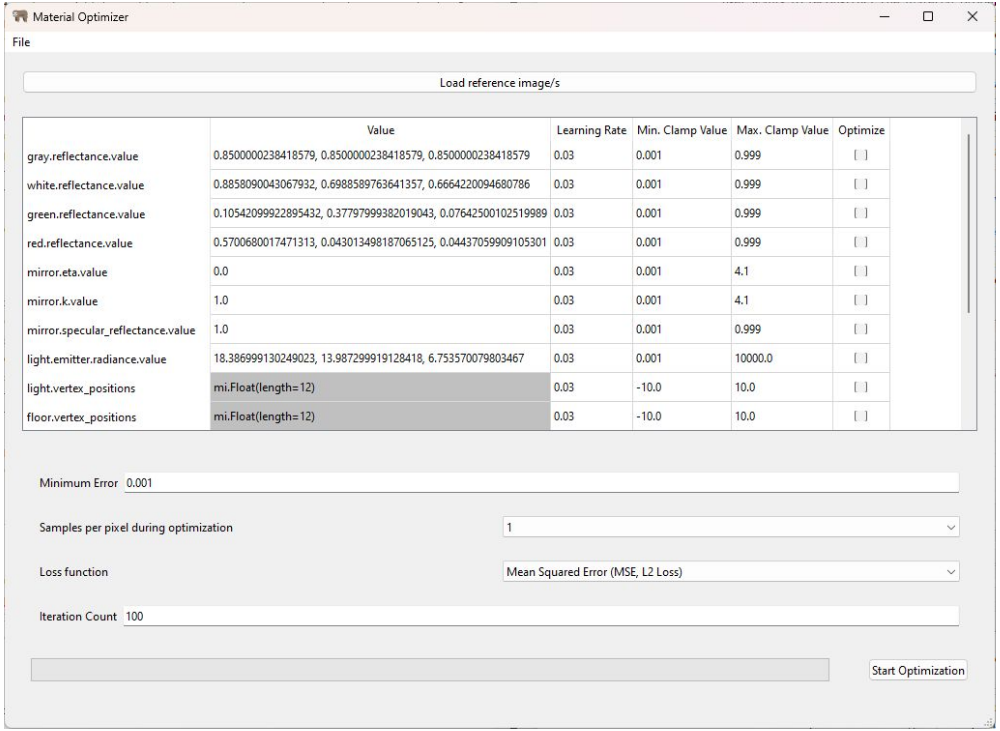
```xml
<scene version="3.0.0">
    <!-- Integrator -->
    <integrator type='$integrator'>
        <integer name="max_depth" value="$max_depth"/>
    </integrator>
    <!-- Sensor -->
    <sensor type="perspective" id="sensor">
        ...
    </sensor>
    <!-- BSDFs -->
    <bsdf type="diffuse" id="white">
        <rgb name="reflectance" value="0.885809, 0.698859, 0.666422"/>
    </bsdf>
    <bsdf type="dielectric" id="glass"/>
    <!-- Light -->
    <shape type="obj" id="light">
        <string name="filename" value="meshes/cbox_luminaire.obj"/>
        <ref id="white"/>
        <emitter type="area">
            <rgb name="radiance" value="18.387, 13.9873, 6.75357"/>
        </emitter>
    </shape>
    <!-- Shapes -->
    <shape type="obj" id="floor">
        <string name="filename" value="meshes/cbox_floor.obj"/>
        <ref id="white"/>
    </shape>
    <shape type="sphere" id="glasssphere">
        <ref id="glass"/>
    </shape>
</scene>
```

Example (simple) Mitsuba scene [5].

Method (1)

Using our tool:

1. Load a scene file which includes material parameters x_0.
2. Load a (set of) reference image(s).
3. Select x, which gets assigned to initialized ADAM optimizer.
4. (Optional) Select optimization hyperparameters (e.g. iteration count).
5. Start the optimization loop.



User interface of our tool.

Method (2)

Optimization loop

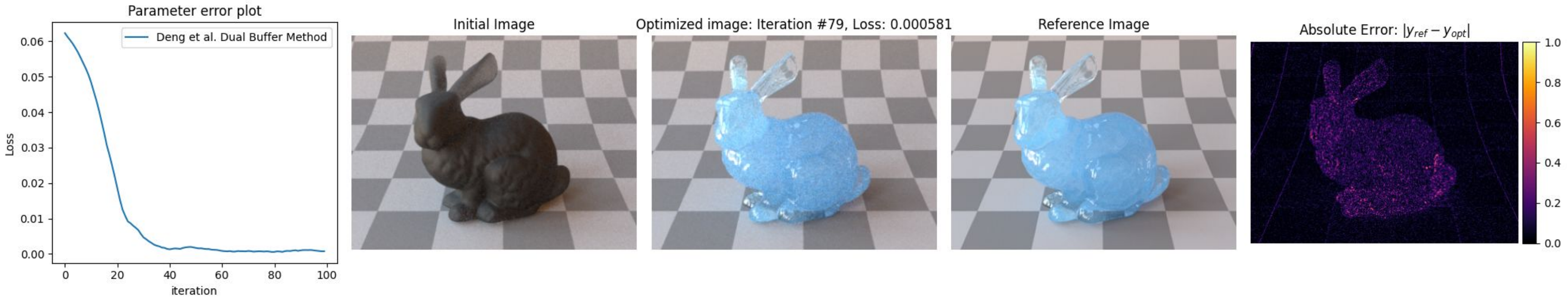
Our tool initializes $x_i = x_0$, and runs for each camera pose (i.e. reference image):

1. Perform a differentiable rendering step with respect to x_i resulting in an image y_i .
2. Evaluate the objective function $g(y_i)$.
3. Back-propagate $\delta g / \delta y$ using Mitsuba 3, to obtain $\delta g / \delta x_i$.
4. Take an ADAM optimization step to find updated parameters $\tilde{x}_{i+1}$.
5. Ensure legal values for x_{i+1} by clamping $\tilde{x}_{i+1}$ using box constraints.
6. Update the scene with x_{i+1}.
7. Repeat until either the loss tolerance or the maximal iterations is reached.

Results: Synthetic Data

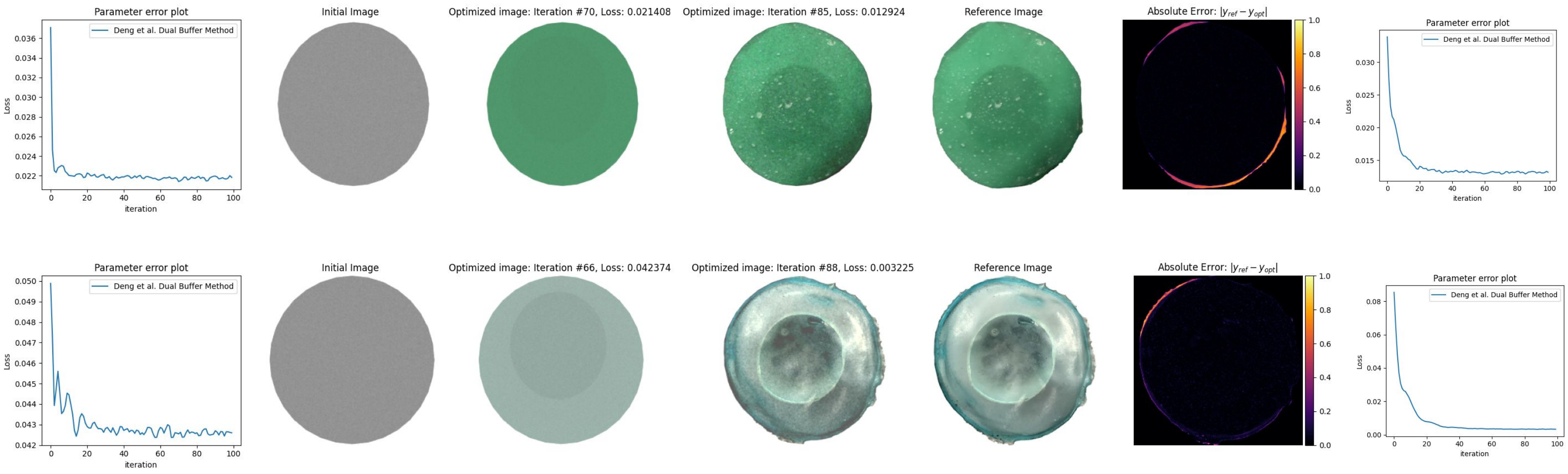
Disney Principled BSDF with integrated subsurface scattering [7, 8]

Rough dielectric BSDF [9] with homogeneous participating medium [6]



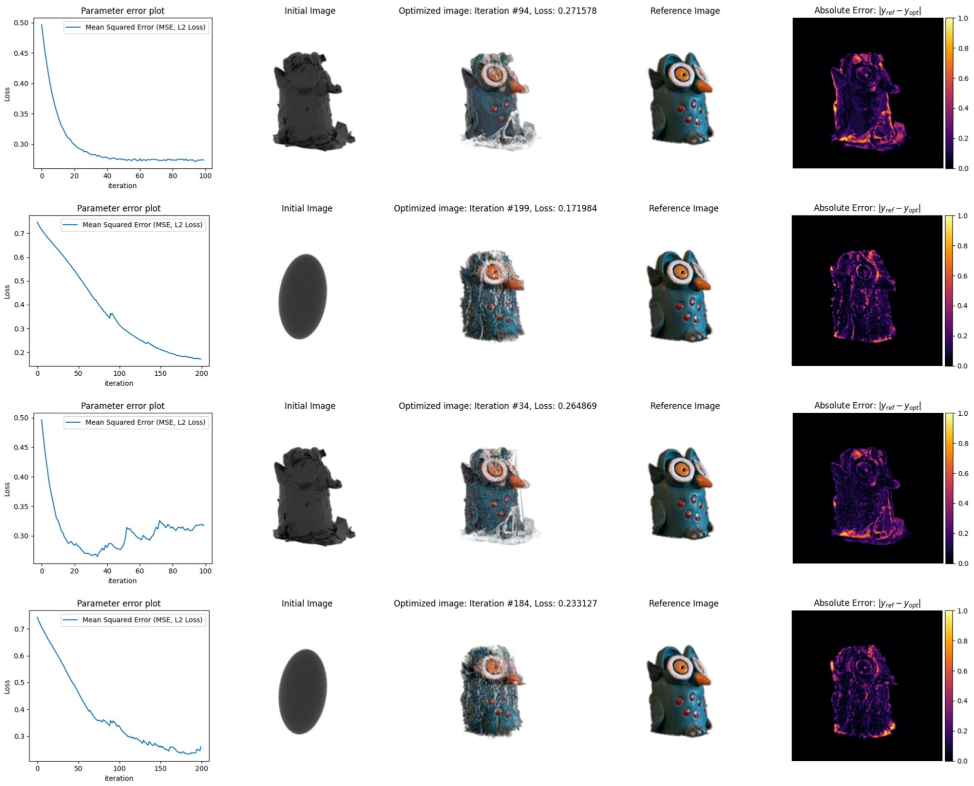
Results: Alginate [5] specimens (real-world)

Disney Principled BSDF with integrated subsurface scattering [7, 8]



Plots Left/Right: Parameter error plot from the first/second optimization.
Images–Left to right: (1) Initial image. (2/3) Optimized image from the first and second part of the optimization. (4) Reference. (5) Absolute error.

Results: Bird statue (real-world)



Geometry and material estimation. Rows: (1) Material estimation using reconstructed mesh from Metashape [16] (2) Texture and geometry estimation. (3) Material and geometry estimation using reconstructed mesh from Metashape. (4) Geometry and material estimation.

Thank you for your attention

Thesis/paper/code available:
https://github.com/sapo17/BachelorThesis

References (1)

[1] Physics-Based Differentiable Rendering: A Comprehensive Introduction. SIGGRAPH 2020 Course, Zhao et al.

[2] Nimier-David, M., Speierer, S., Ruiz, B., and Jakob, W. Radiative backpropagation: An adjoint method for lightning-fast differentiable rendering. Transactions on Graphics (Proceedings of SIGGRAPH) 39, 4 (July 2020).

[3] Zeltner, T., Speierer, S., Georgiev, I., and Jakob, W. Monte carlo estimators for differential light transport. ACM Trans. Graph. 40, 4 (jul 2021).

References (2)

[4] Vicini, D., Speierer, S., and Jakob, W. Path replay backpropagation: Differentiating light paths using constant memory and linear time. Transactions on Graphics (Proceedings of SIGGRAPH) 40, 4 (Aug. 2021), 108:1–108:14

[5] Marjanovi, I., amec, E., Vaatko, H., and Stavri, M. Alginate in architecture : An experimental approach to the new sustainable building material. In Art and Science Applied: Experience and Vision, . and . , Eds., vol. 2 of SmartArt. , , , 2022, ch. 21, pp. 394–406. 21.

References (3)

[6] Jakob, W., Speierer, S., Roussel, N., Nimier-David, M., Vicini, D., Zeltner, T., Nicolet, B., Crespo, M., Leroy, V., and Zhang, Z. Mitsuba 3 renderer, 2022. https://mitsuba-renderer.org.

[7] Burley, B. Physically-based shading at disney.

[8] Hill, S., McAuley, S., Burley, B., Chan, D., Fascione, L., Iwanicki, M., Hoffman, N., Jakob, W., Neubelt, D., Pesce, A., and Pettineo, M. Physically based shading in theory and practice. In ACM SIGGRAPH 2015 Courses (New York, NY, USA, 2015), SIGGRAPH ’15, Association for Computing Machinery.

References (4)

[9] Walter, B., Marschner, S. R., Li, H., and Torrance, K. E. Microfacet models for refraction through rough surfaces. In Proceedings of the 18th Eurographics Conference on Rendering Techniques (Goslar, DEU,2007), EGSR'07, Eurographics Association, p. 195206

[10] AgiSoft. Agisoft metashape standard, 2022. https://www.agisoft.com/.

[11] Community, B. O. Blender - a 3D modelling and rendering package. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2022.

References (5)

[12] <baptiste.nicolet@epfl.ch>, B. N. Mitsuba blender add-on. https://github.com/mitsuba-renderer/mitsuba-blender, 2022

[13] Deng, X., Luan, F., Walter, B., Bala, K., and Marschner, S. Reconstructing translucent objects using differentiable rendering. In ACM SIGGRAPH 2022 Conference Proceedings (New York, NY, USA, 2022), SIGGRAPH ’22, Association for Computing Machinery.

[14] Delatronic. Dragon. https://blendswap.com/blend/15891, August 21, 2015. [Online; accessed January 30, 2023].

References (6)

[15] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization, 2014.

[16] AgiSoft. Agisoft metashape standard, 2022. https://www.agisoft.com/.