Technical Presentation

SVBRDF Estimation using a Physically-based Differentiable Renderer

Markus Andreas Worchel

Recap

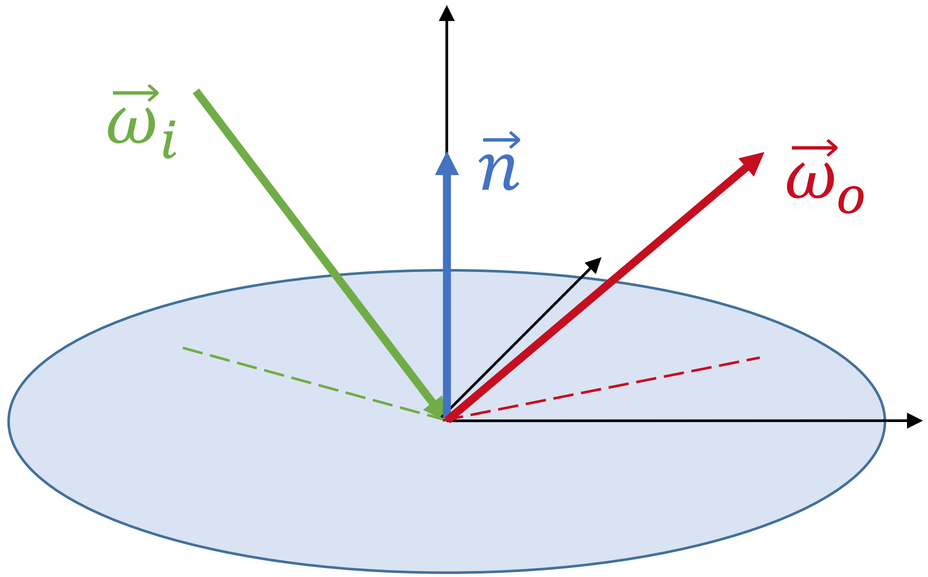
Physically-based Differentiable Renderer

Deschaintre et al.,
2018 (single view)
2019 (multi view)

Bidirectional Reflectance Distribution Function

- $f_r(\vec{\omega}_i, \vec{\omega}_o) :=$ "Fraction of the irradiance coming from $\vec{\omega}_i$ that is reflected towards $\vec{\omega}_o$"

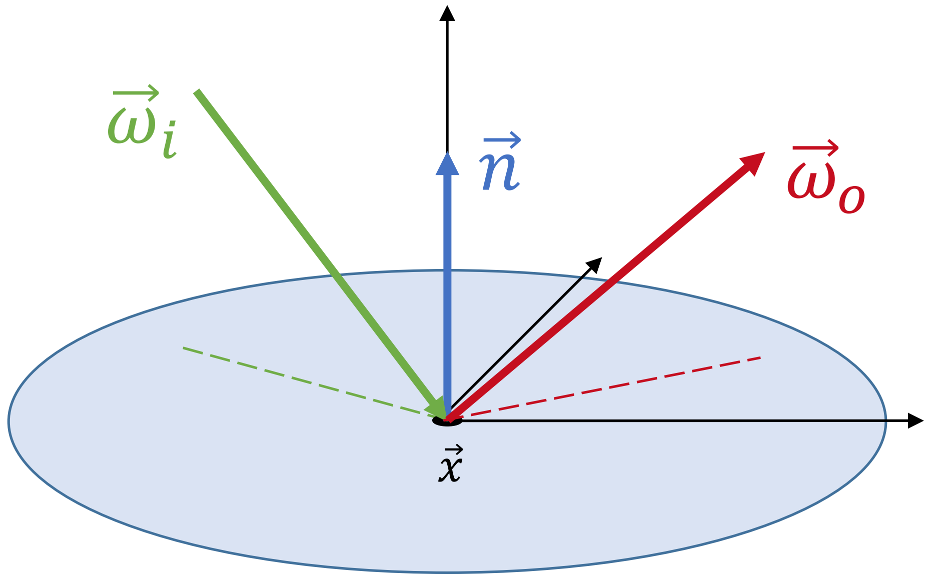
- Captures reflectance properties of the surface material



Spatially Varying BRDF

- **S**patially **v**arying reflectance properties
  $\rightarrow f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}_o)$

- Surface is assumed to be heterogeneous
  - Small variations of the same material
  - Different materials

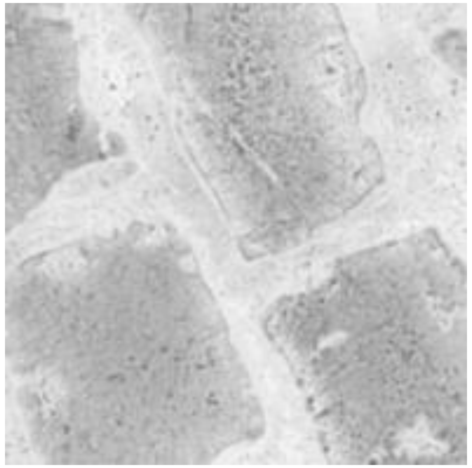


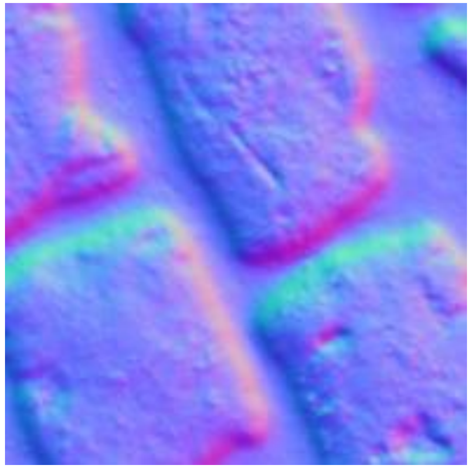
Cook-Torrance BRDF

- Anisotropic microfacet model
  - Surface consists of small differently oriented microfacets (“bumps”)
  - Only relative difference of $\vec{\omega}_i$ and $\vec{\omega}_o$ matters, not their absolute orientation
- Four parameter (maps), separation of diffuse and specular:

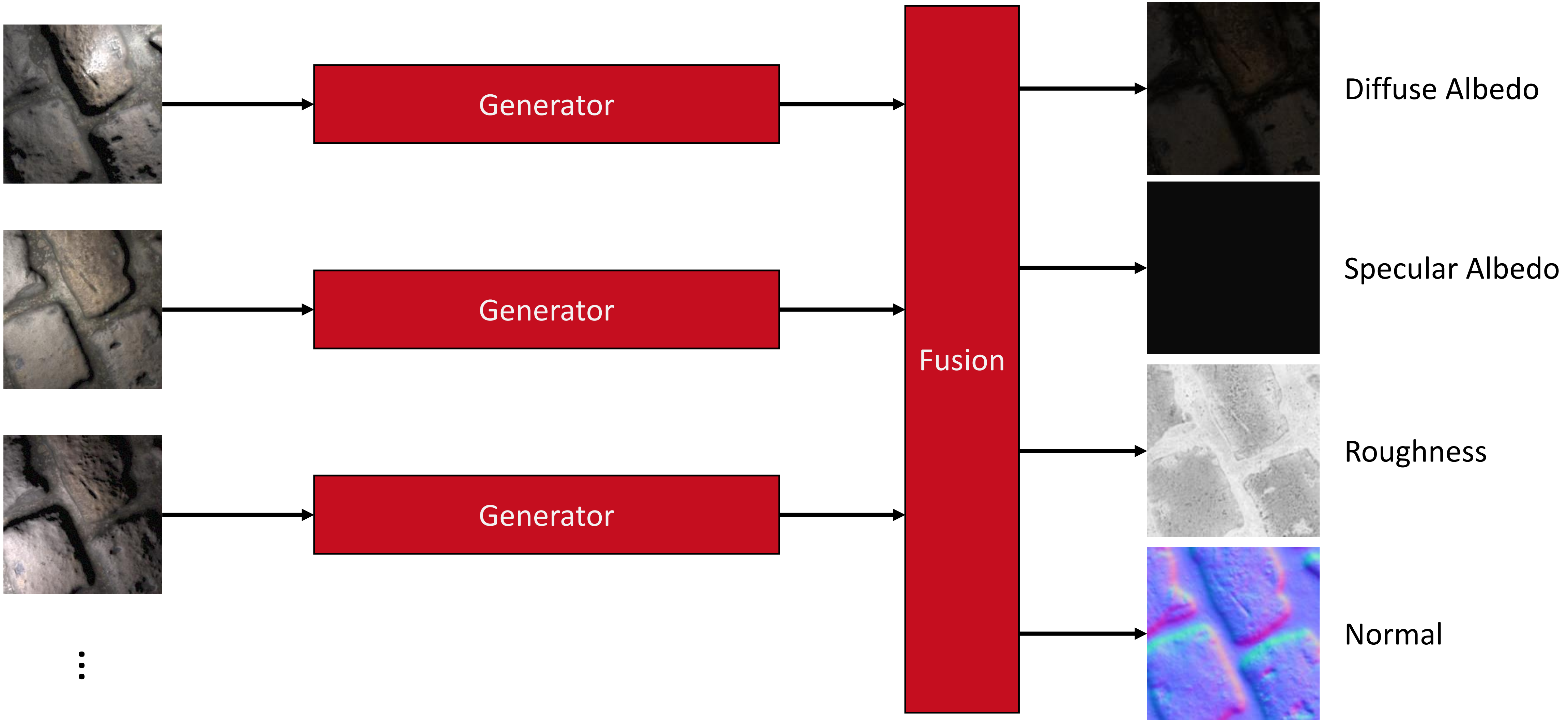

Diffuse Albedo


Specular Albedo


Specular Roughness

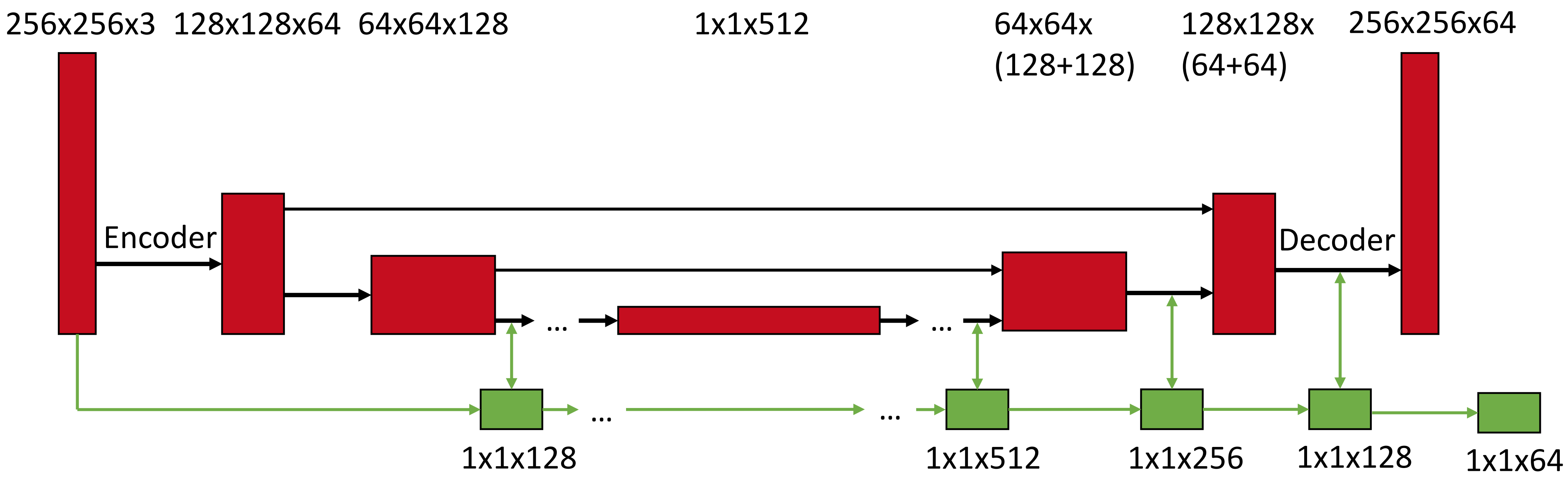

Normal

Network Architecture

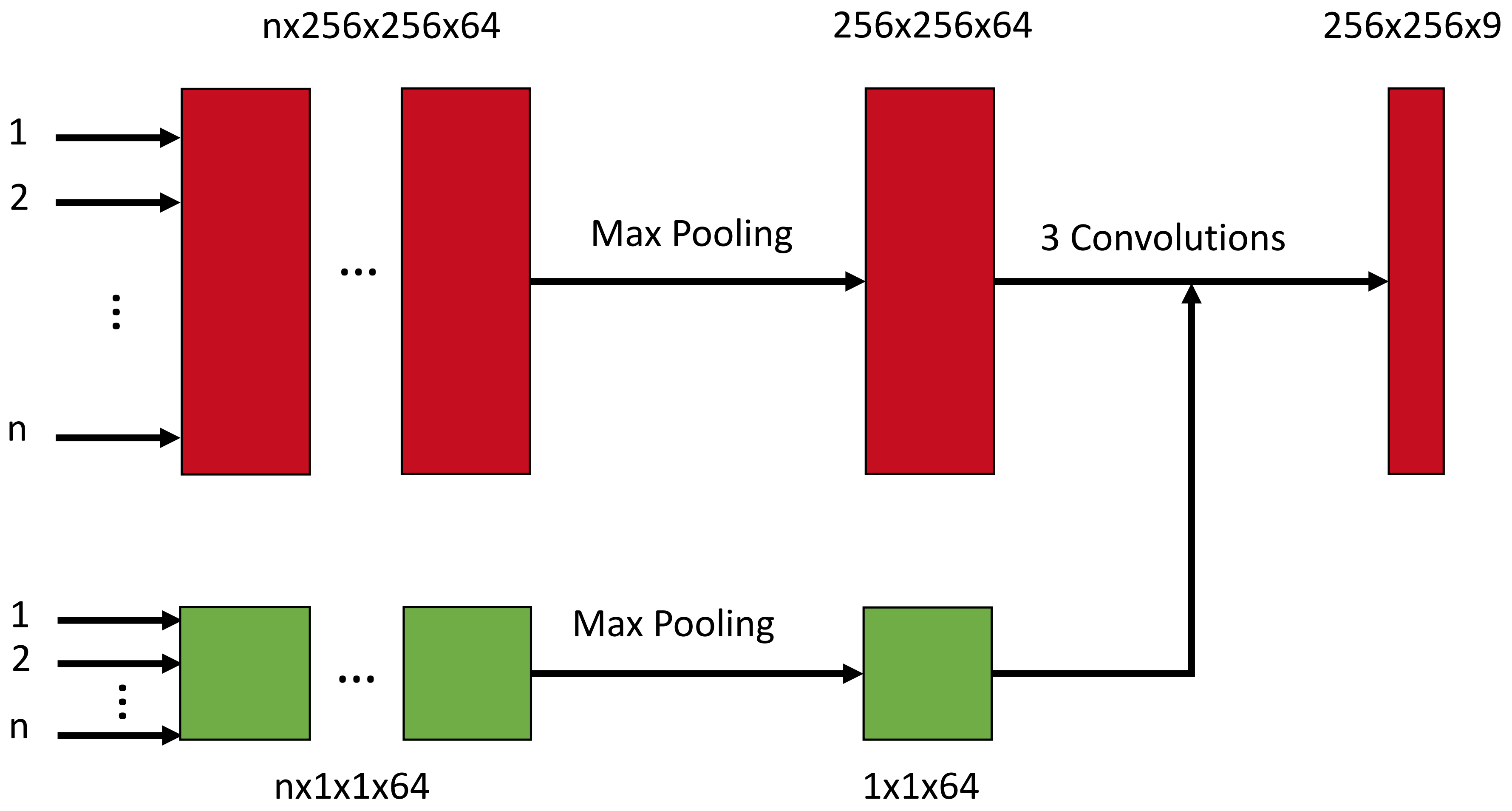


Network Architecture – Generator

- Based on U-Net architecture for image-to-image transformation
- Additional track reinjects global information lost due to instance norm



Network Architecture – Multi-View Fusion



Training – Loss

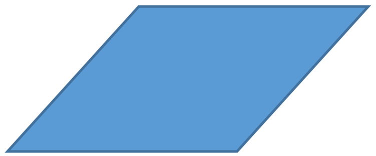
- Given for each sample:
  - Ground truth SVBRDF maps D_G, S_G, R_G, N_G
  - Predicted SVBRDF maps D_P, S_P, R_P, N_P

- Loss function is $L \coloneqq L_{Rendering} + \lambda(L_D + L_S + L_R + L_N)$

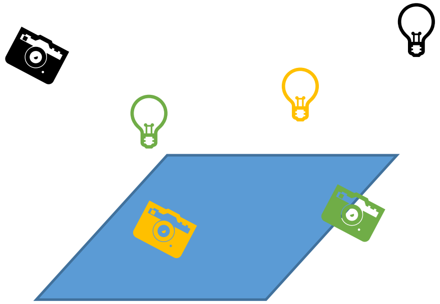
- L_D, L_S, L_R, L_N are simple L1 losses between predicted and ground truth parameter maps: $L_I \coloneqq ||I_G - I_P||_1$

- $L_{Rendering}$ is a **rendering loss**: Compares rendered appearance

Training – Rendering Loss



1. Scene definition



2. Light and camera configurations (include specular configurations)

3. Render object with ground truth SVBRDF in all different configurations

4. Render object with predicted SVBRDF in all different configurations

5. Compare the rendered images (L1 loss)

Training – Rendering Loss

- Challenge: Loss must be differentiable → **differentiable rendering**

- Solution in the papers:
  - Simple tensorflow renderer → Only direct illumination
  - Simple scene → One flat plane, one light

- My goal:
  - Differentiable path tracer → Global illumination
  - Arbitrary scene → Flat plane, other objects, multiple lights, environment maps, …

Current State – Generator

- Generator reimplemented in PyTorch
- Overfitting test (2000 epochs)
- 1 of 2 training samples (top)
- 1 test sample (bottom)
- Limitations:
  - 3 color channels per target image → 12 channels instead of 9
  - Simple L1 loss