

Final Presentation

SVBRDF Estimation using a Physically-based Differentiable Renderer

Markus Andreas Worchel

Recap – Topic



...



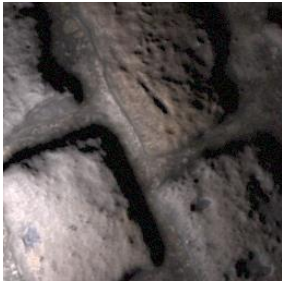
⋮

⋮

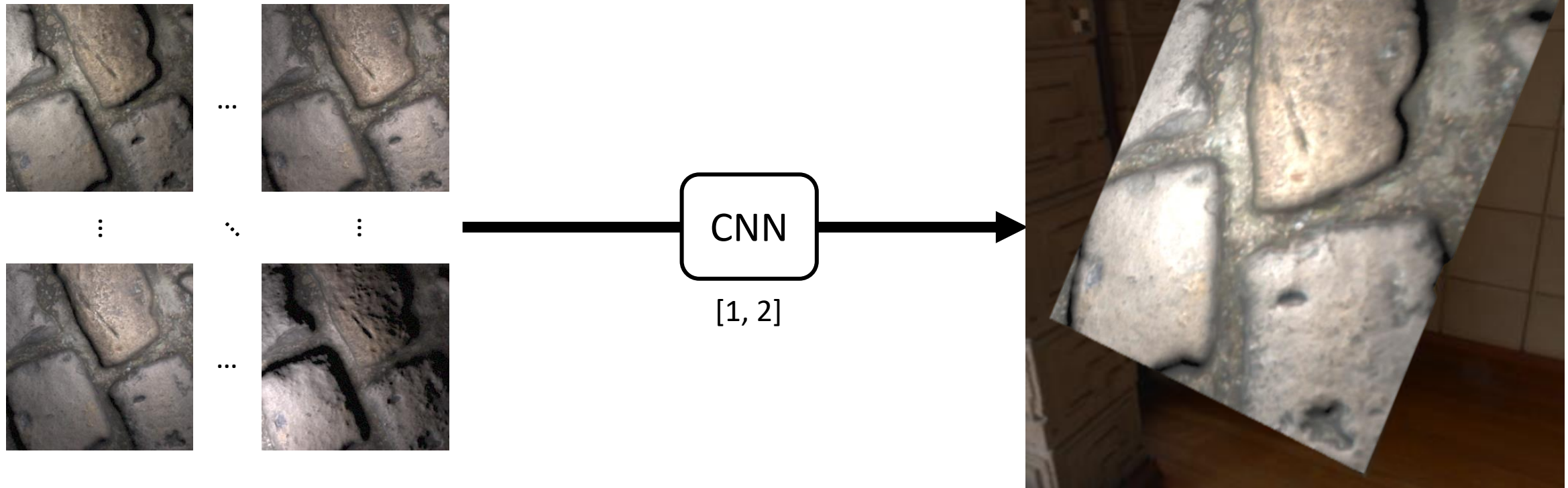
⋮



...



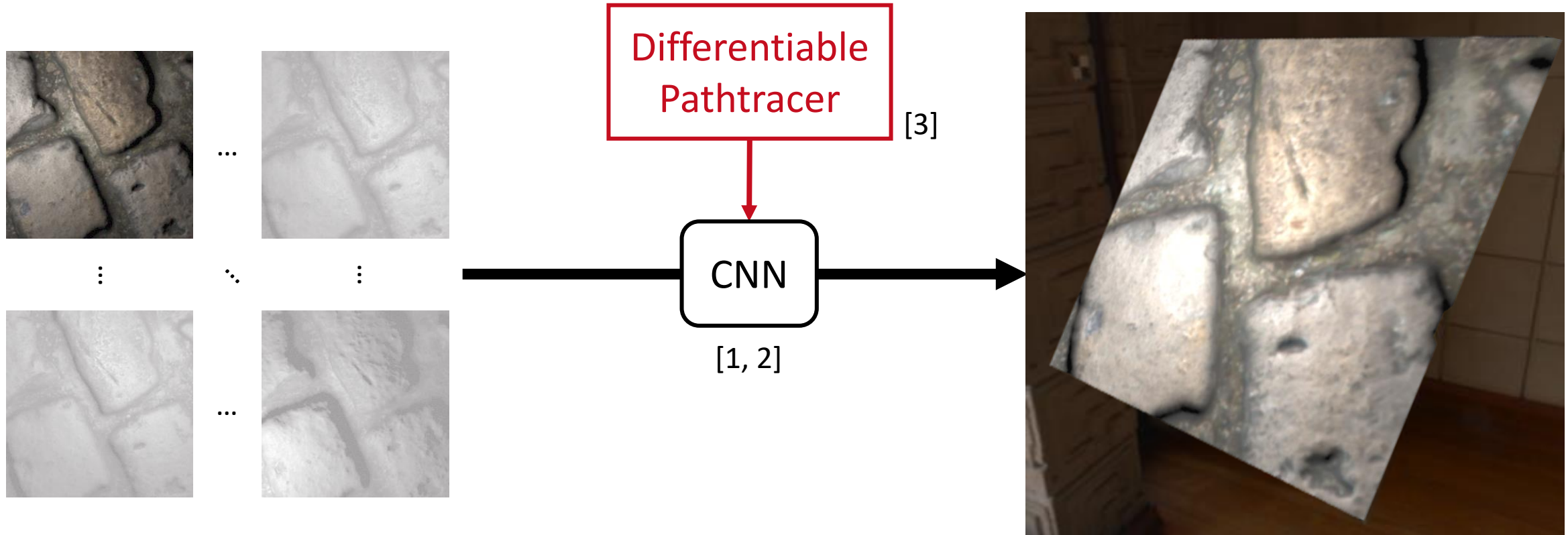
Recap – Approach



[1] Deschaintre et al., 2018, [Single-Image SVBRDF Capture with a Rendering-Aware Deep Network](#)

[2] Deschaintre et al., 2019, [Flexible SVBRDF Capture with a Multi-Image Deep Network](#)

Recap – Project Scope



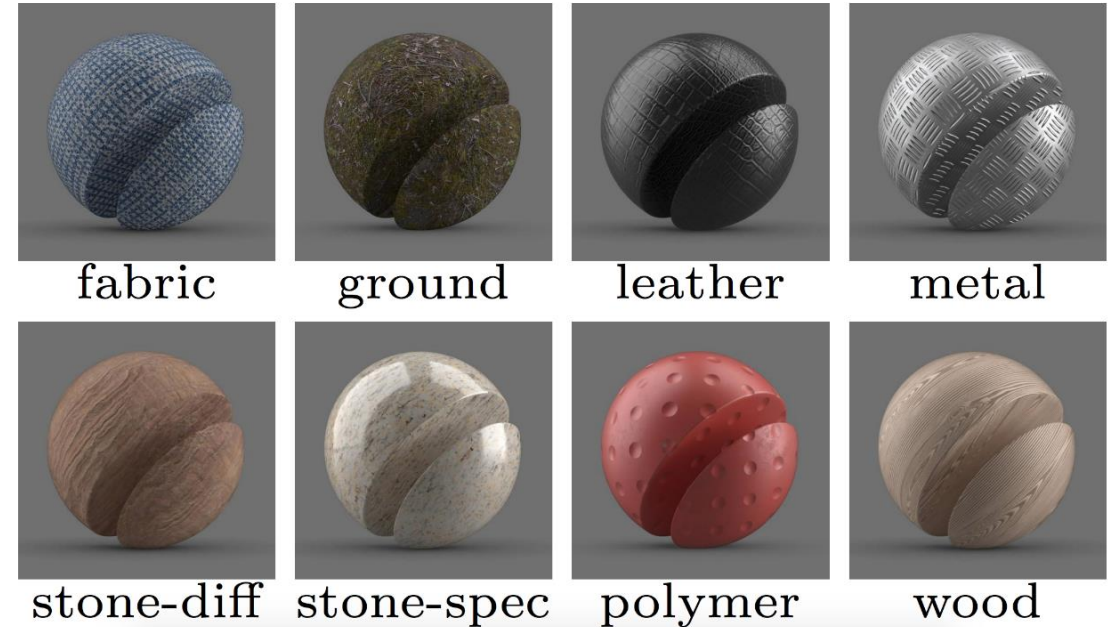
[1] Deschaintre et al., 2018, [Single-Image SVBRDF Capture with a Rendering-Aware Deep Network](#)

[2] Deschaintre et al., 2019, [Flexible SVBRDF Capture with a Multi-Image Deep Network](#)

[3] Li et al., 2018, [Differentiable Monte Carlo Ray Tracing through Edge Sampling](#)

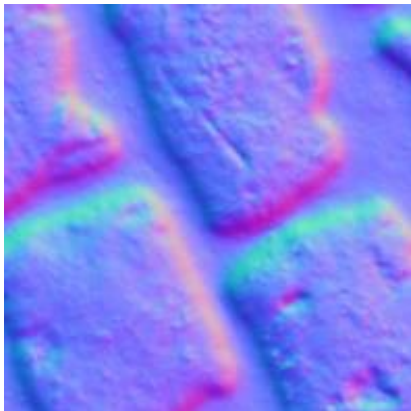
Recap – Motivation

- Why material estimation?
 - Holistic 3D reconstruction (geometry + material)
 - Photorealistic assets for
 - Games
 - Movies
 - Cultural heritage
 - ...
 - Multi material 3D printing
- Why using a differentiable pathtracer?
 - General approach (independent of use case)
 - Flexible light transport simulations





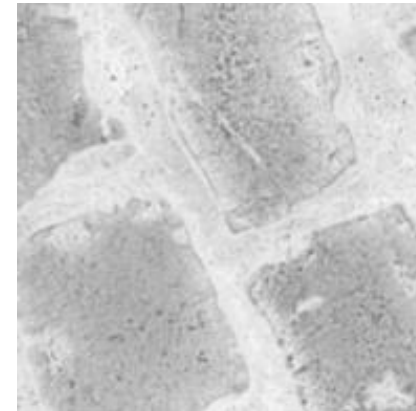
SDR Image



Normal



Diffuse Albedo

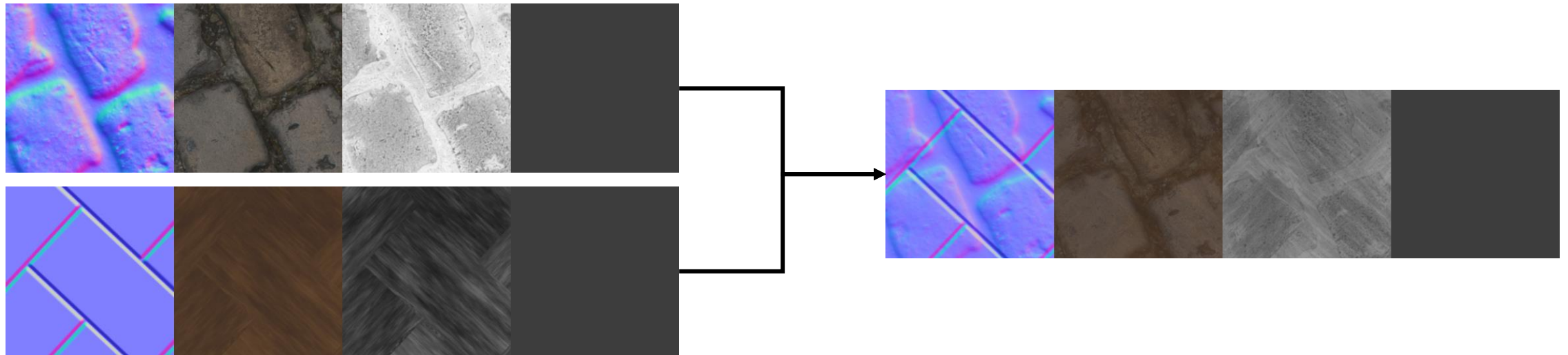


Roughness

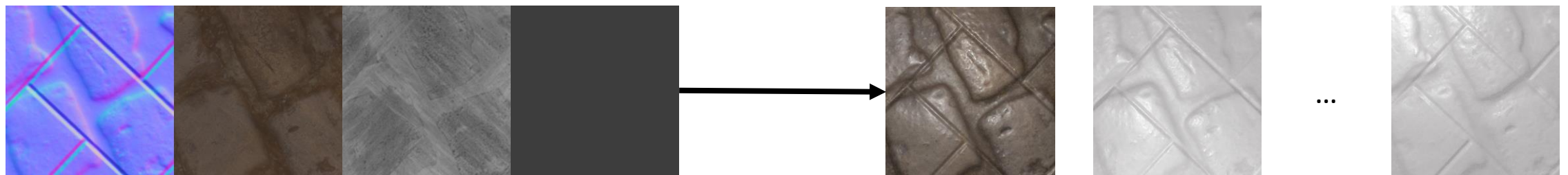


Specular Albedo

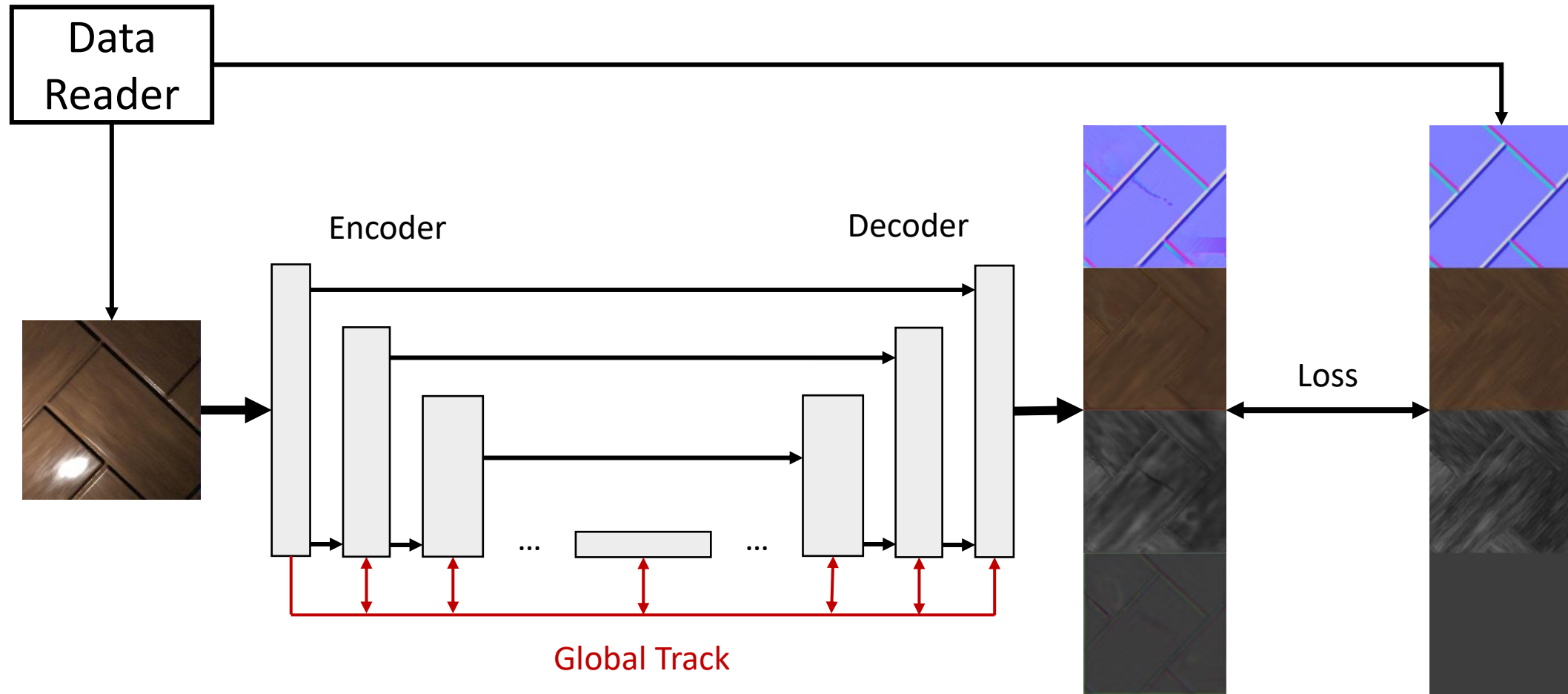
- Material mixing



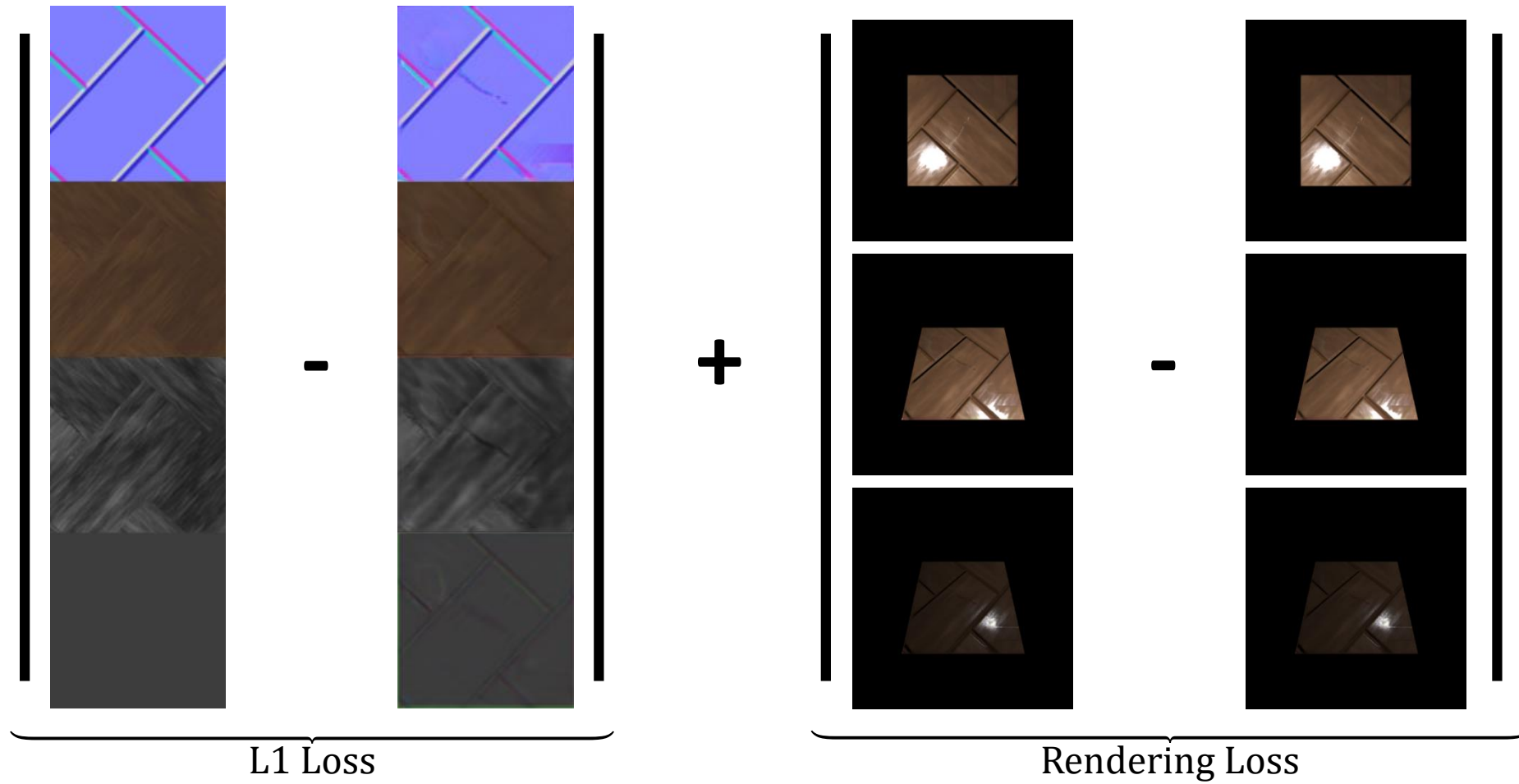
- Augmented image generation



Algorithm – Overview



Algorithm – Loss



Algorithm – Differentiable Renderers

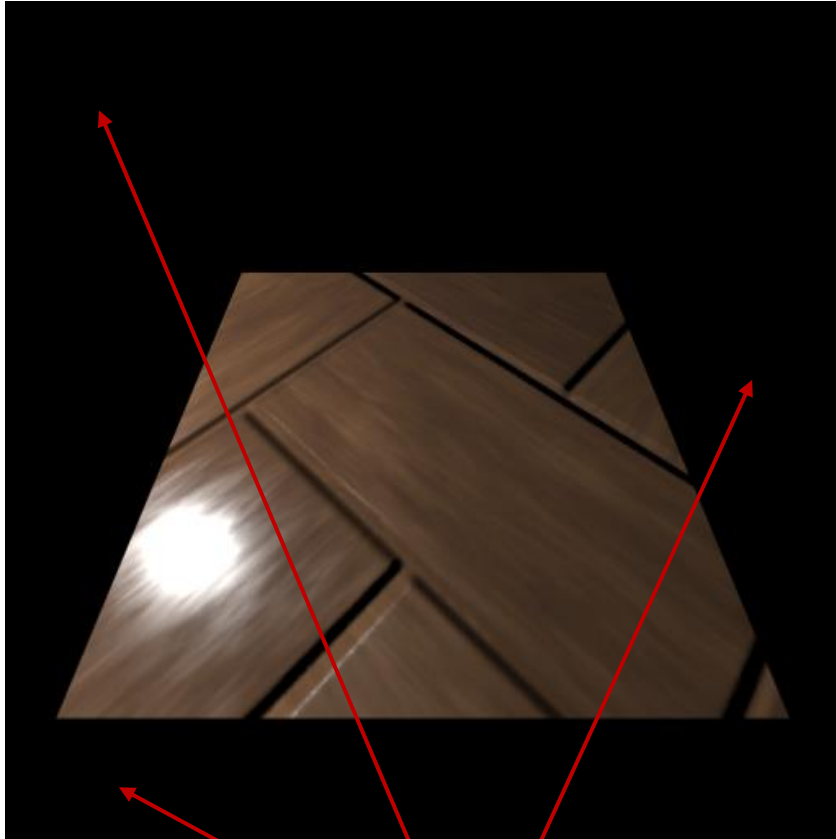


Renderer for direct illumination
implemented using PyTorch (`local`)



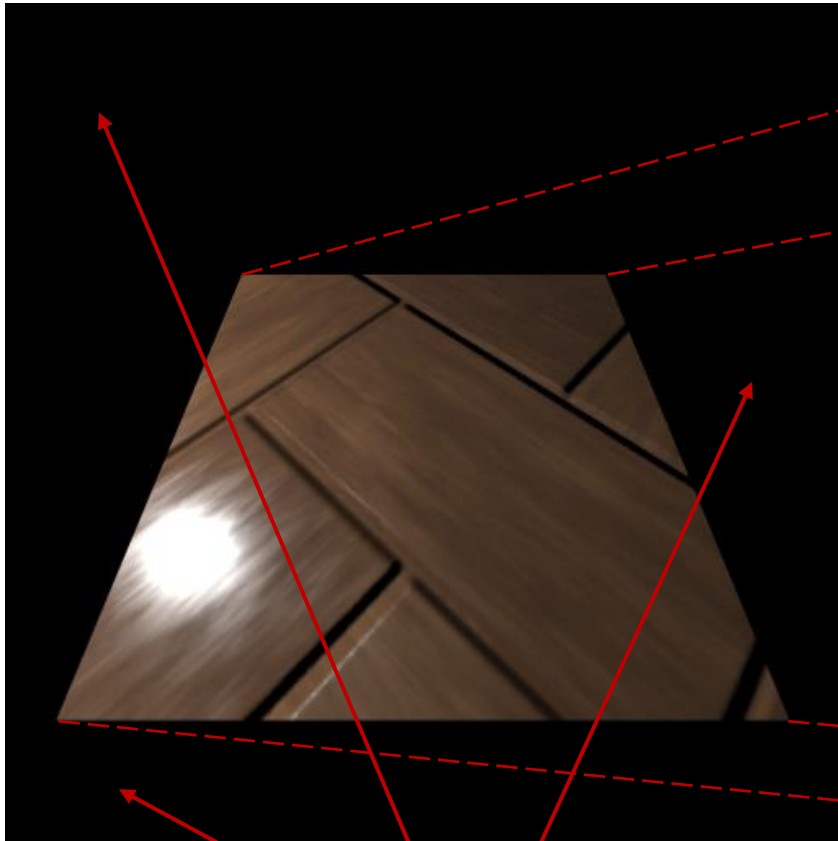
Redner pathtracer implemented in C++
with PyTorch bindings (`pathtracing`)

Algorithm – Renderer Output

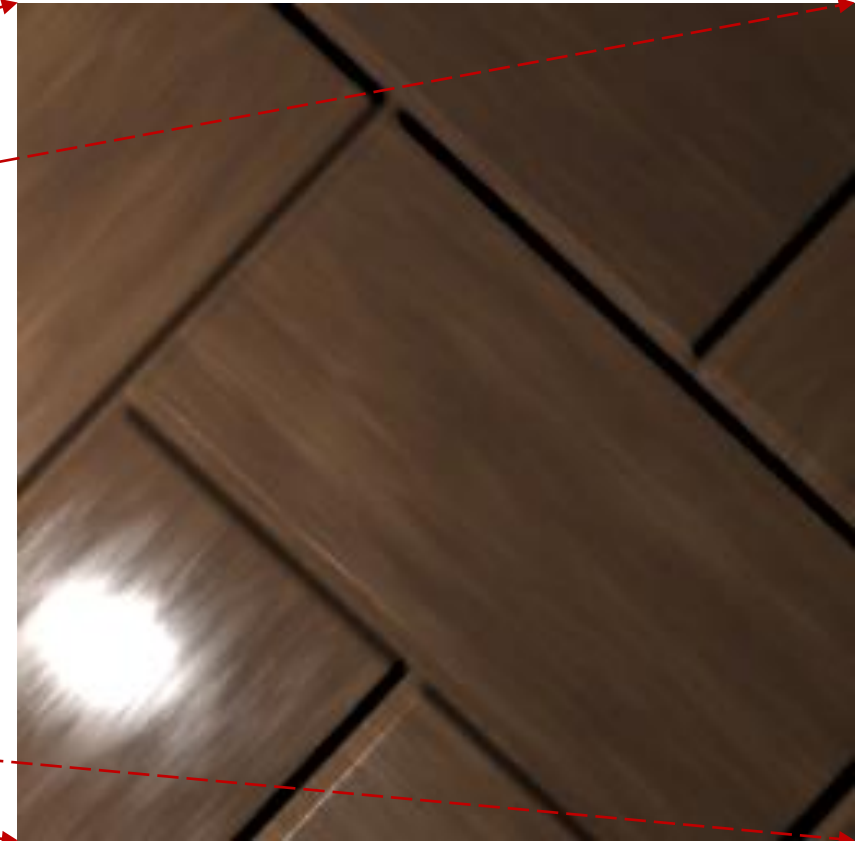


Problem: No gradient information for weight optimization

Algorithm – Renderer Output

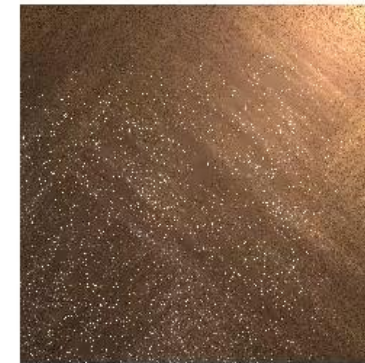
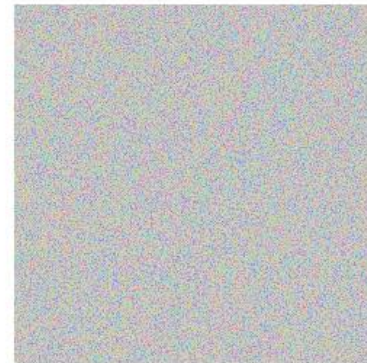
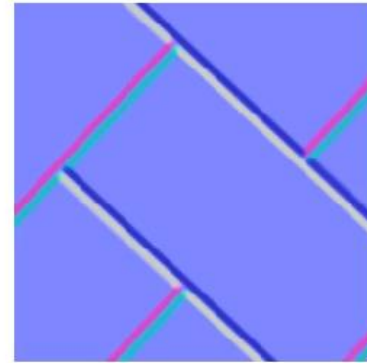
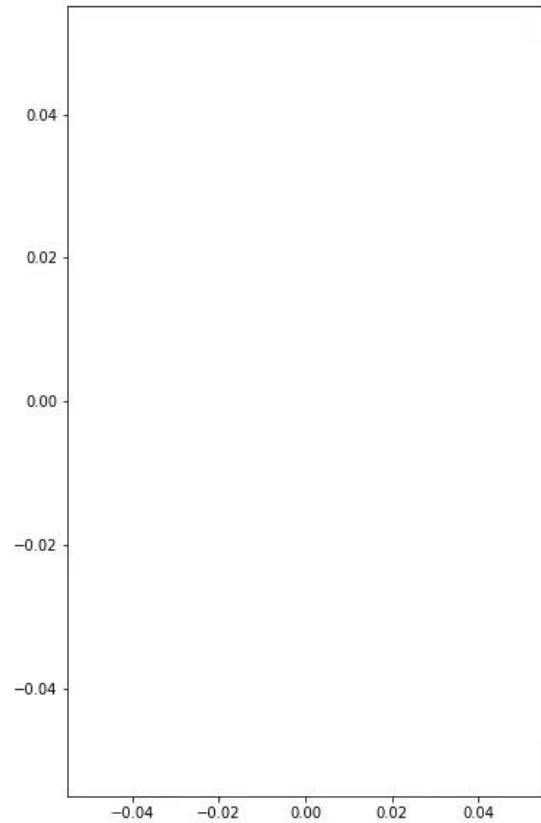


Problem: No gradient information for weight optimization

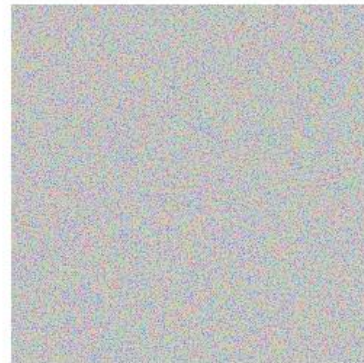
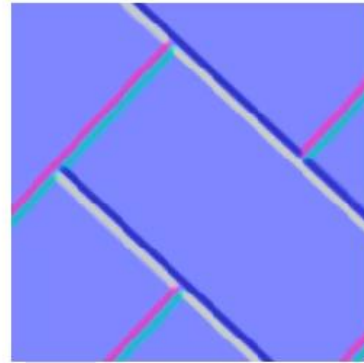
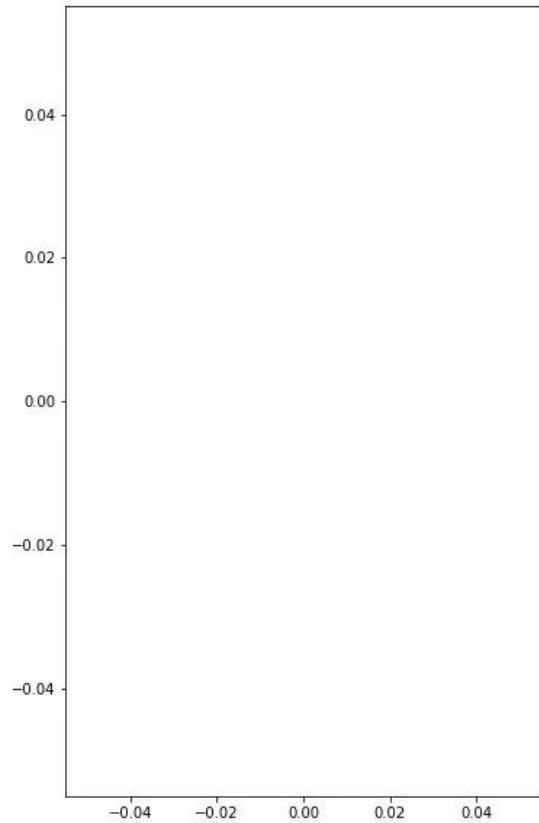


Solution: Full patch sampling

Algorithm – Patch Sampling



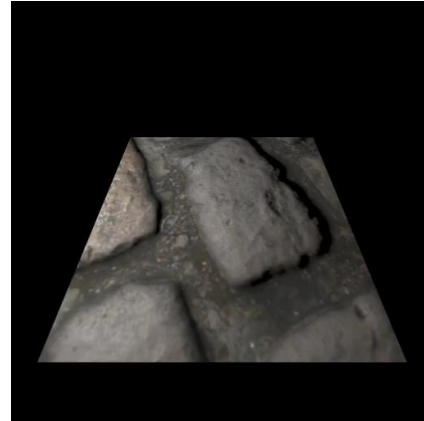
Algorithm – Patch Sampling in Redner



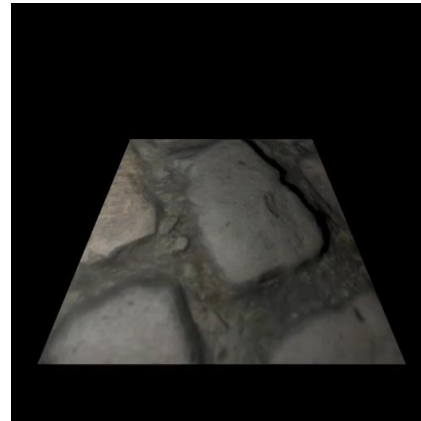
Results – Artificial Images (1)



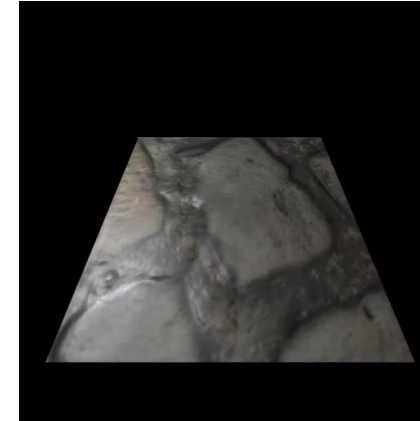
Ground Truth



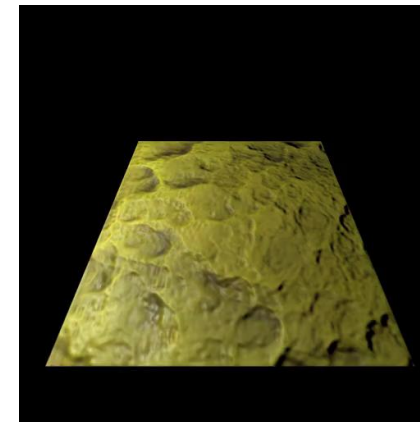
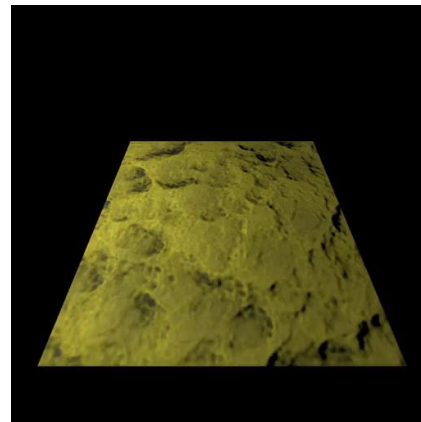
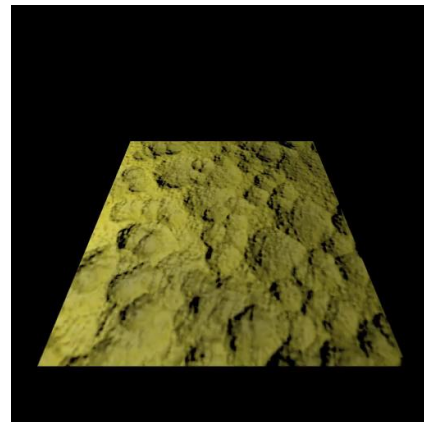
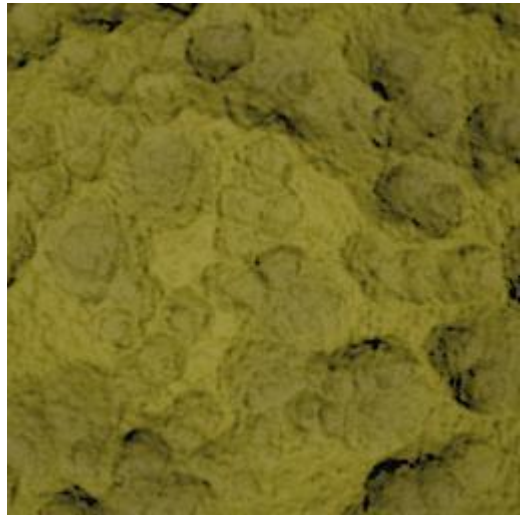
Reference (400k)



Local (285k)



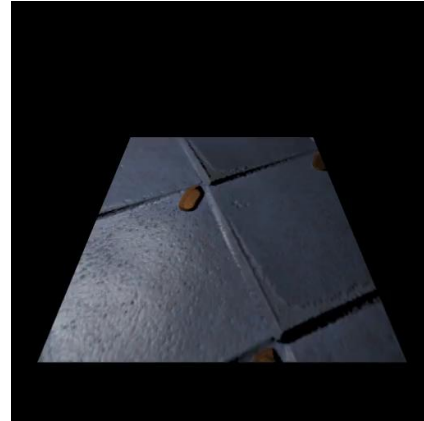
Pathtracing (17k)



Results – Artificial Images (2)



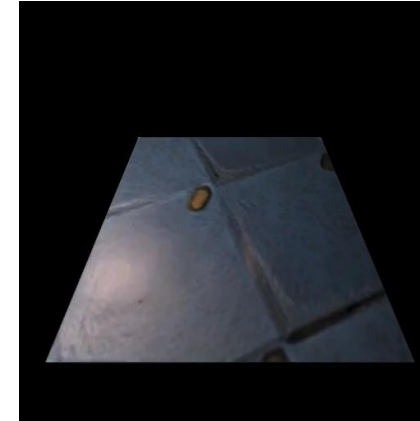
Ground Truth



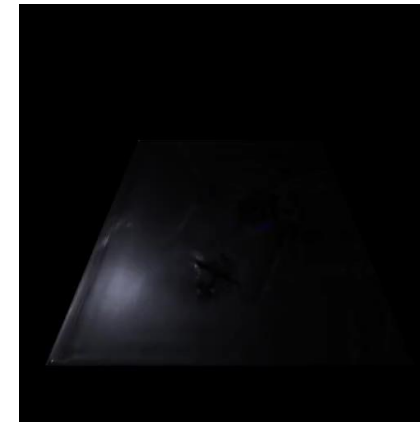
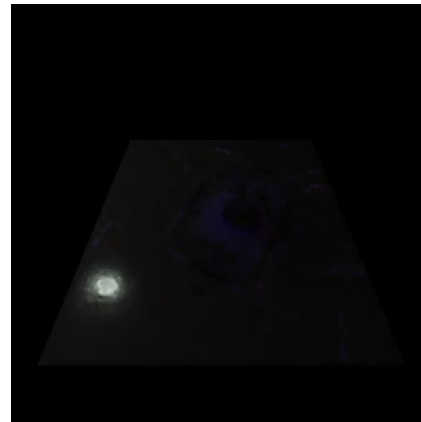
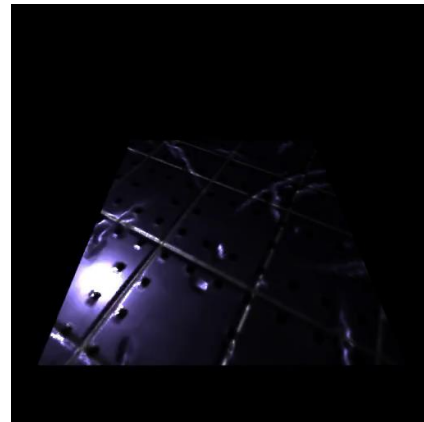
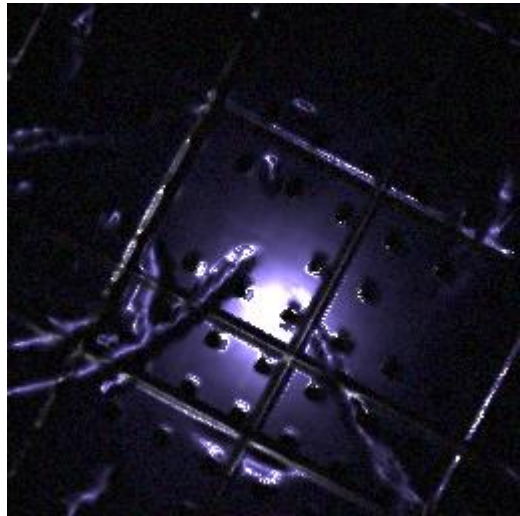
Reference (400k)



Local (285k)



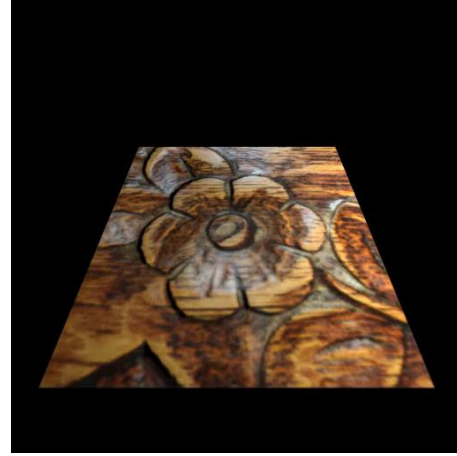
Pathtracing (17k)



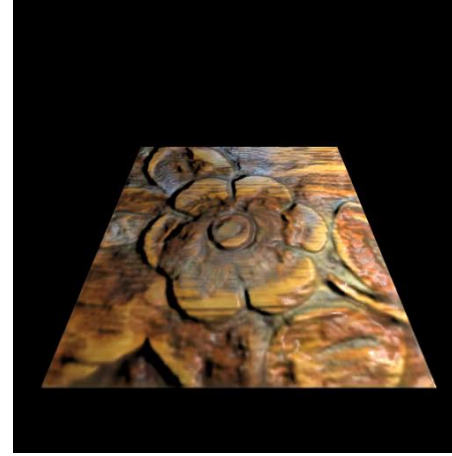
Results – Real Images (1)



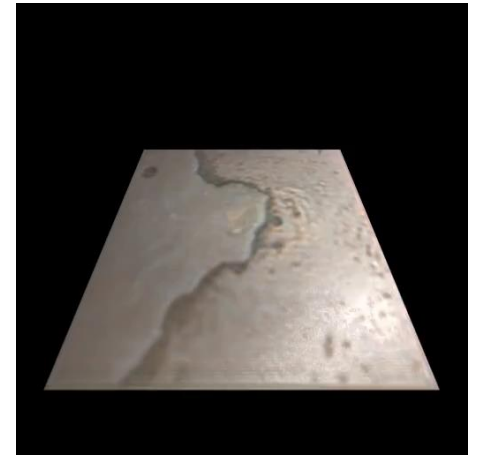
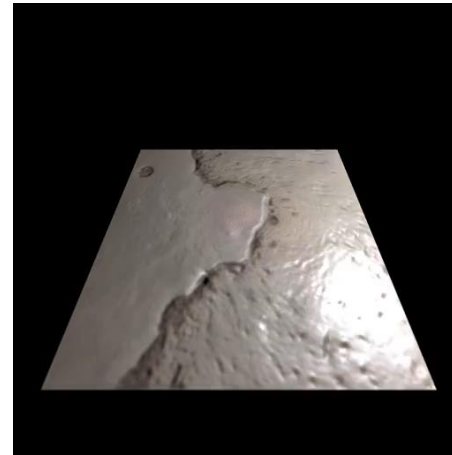
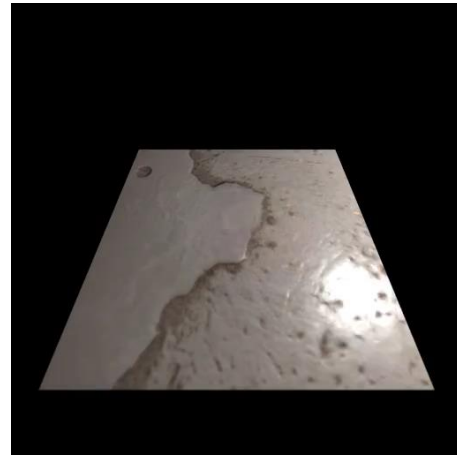
Reference (400k)



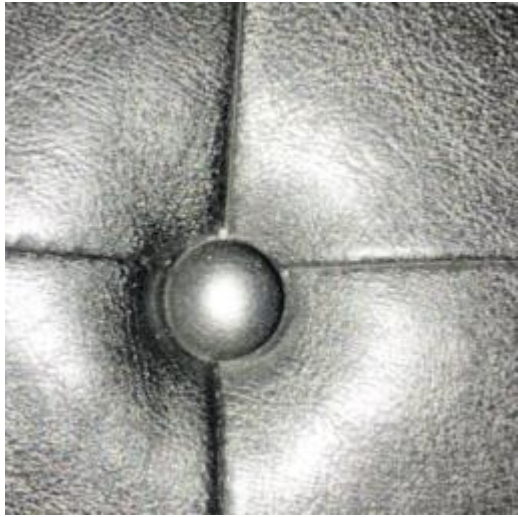
Local (285k)



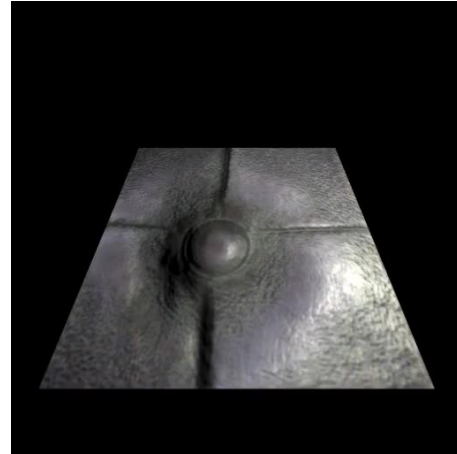
Pathtracing (17k)



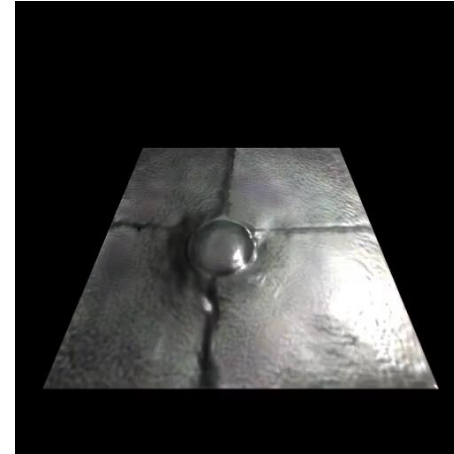
Results – Real Images (2)



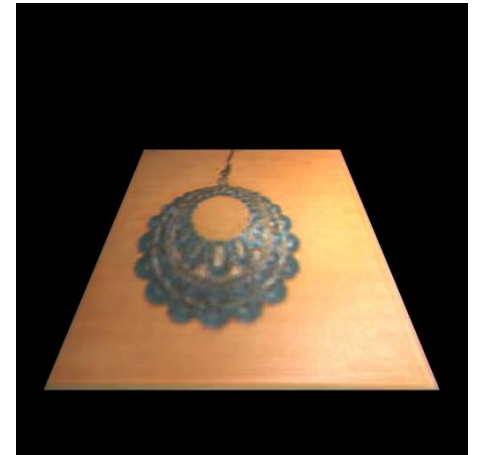
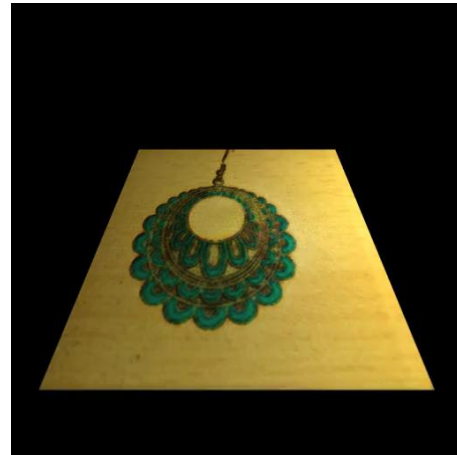
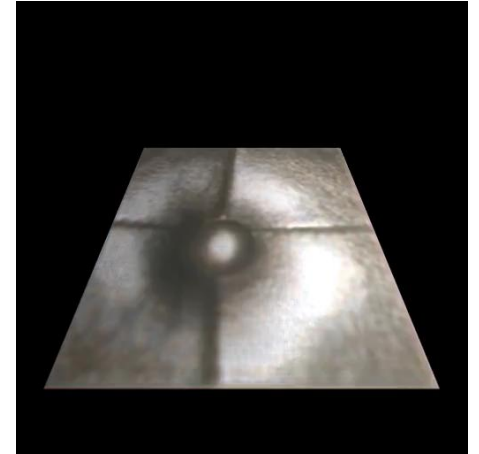
Reference (400k)



Local (285k)

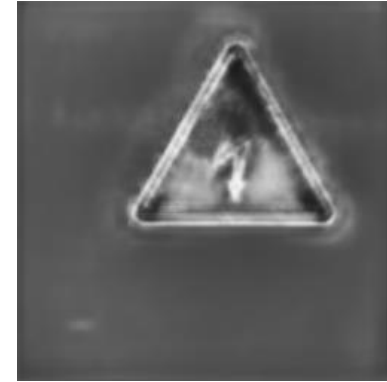
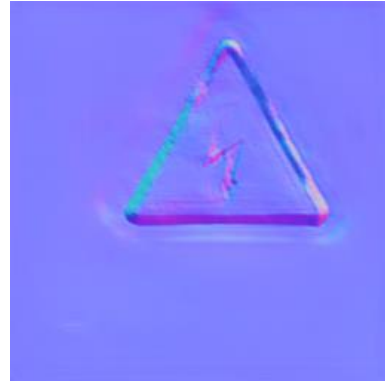


Pathtracing (17k)



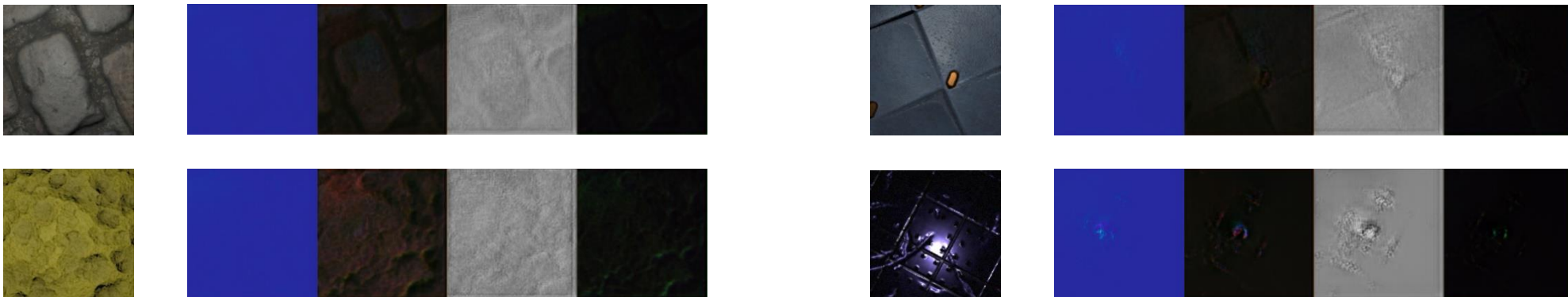
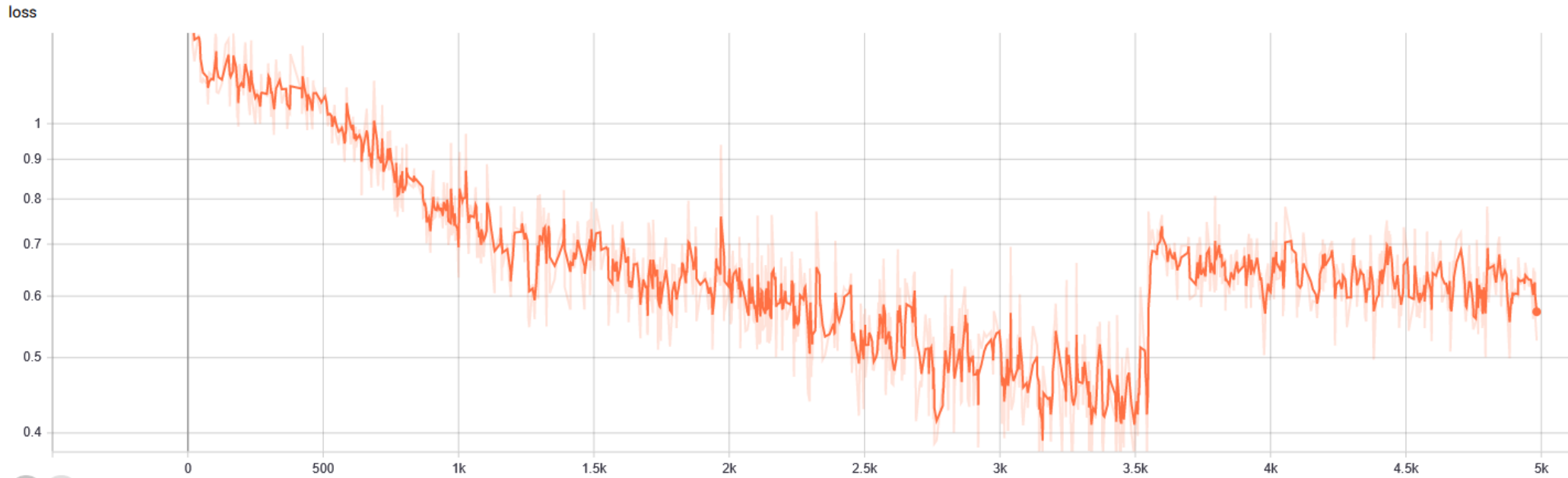
Limitations – The Obvious

- Correlated SVBRDF maps

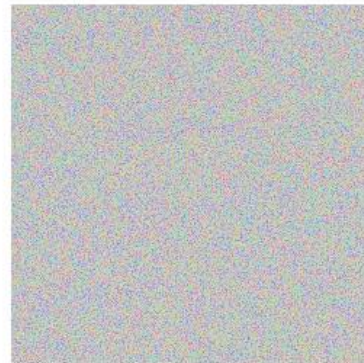
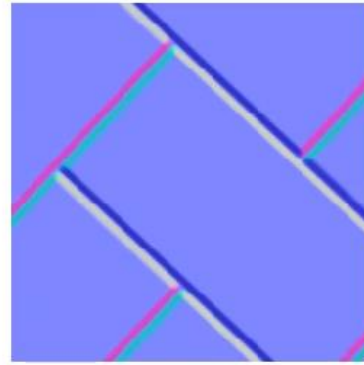
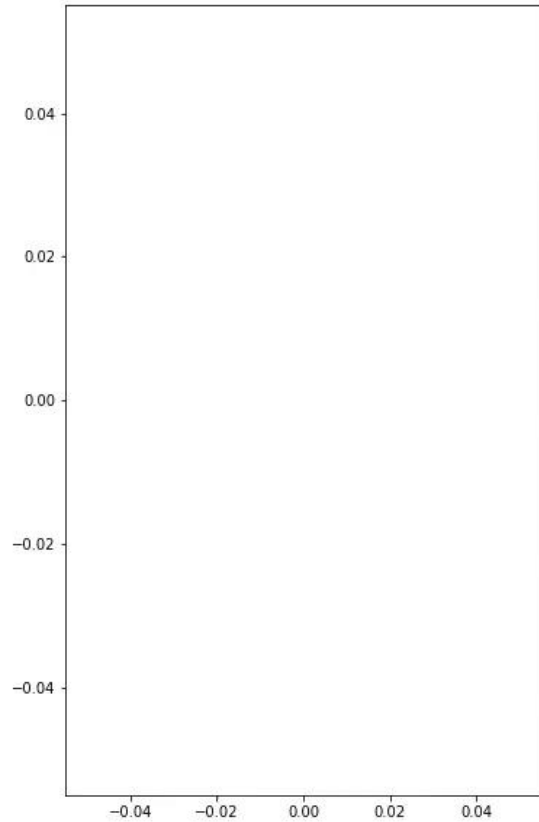


- Assumption of mostly flat macro geometry
- Training with pathtracer is ~ 25 times slower

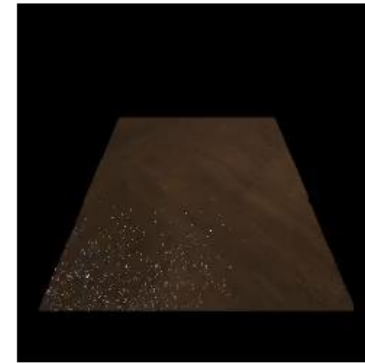
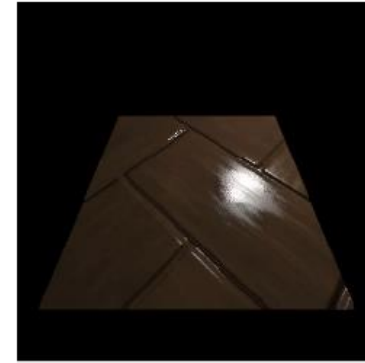
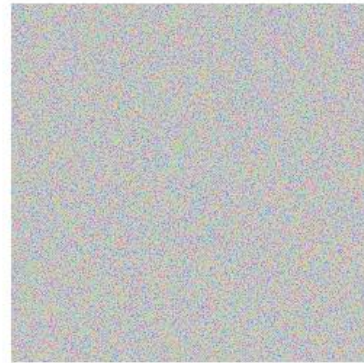
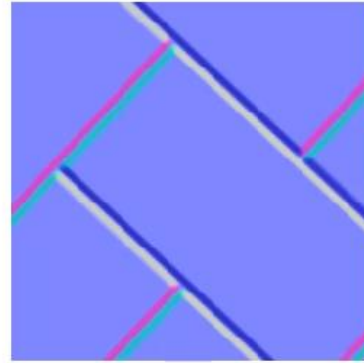
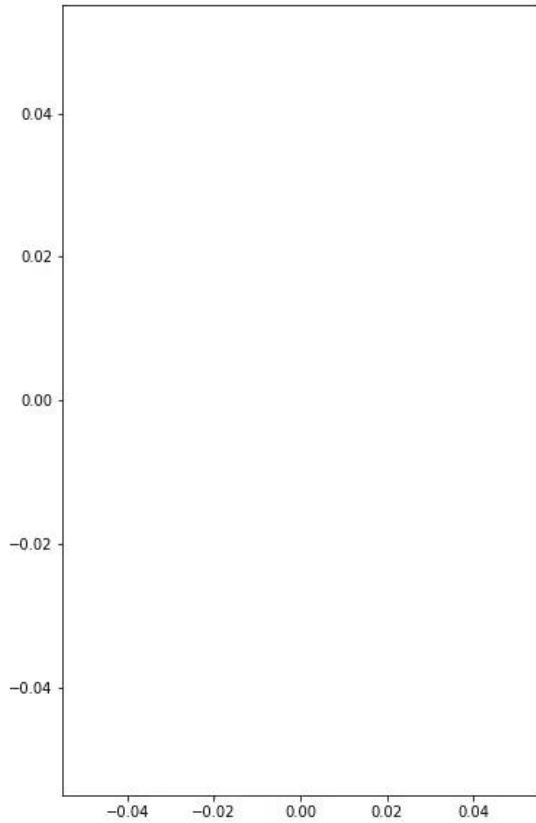
Limitations – Wrong/Noisy Gradients?



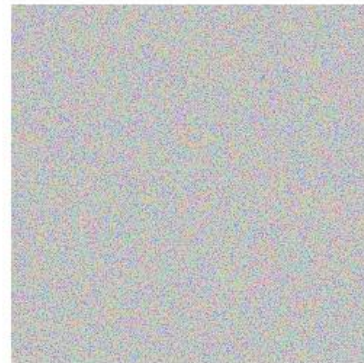
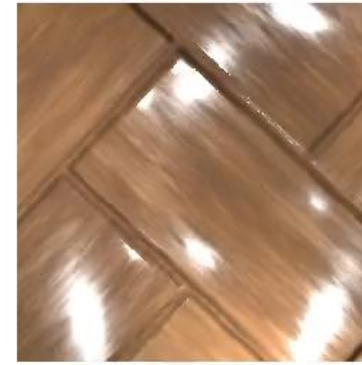
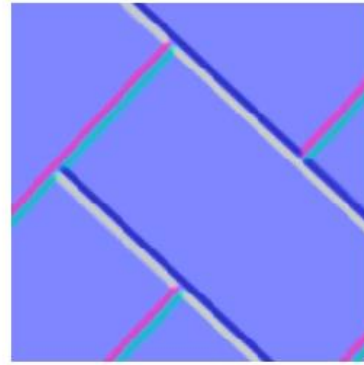
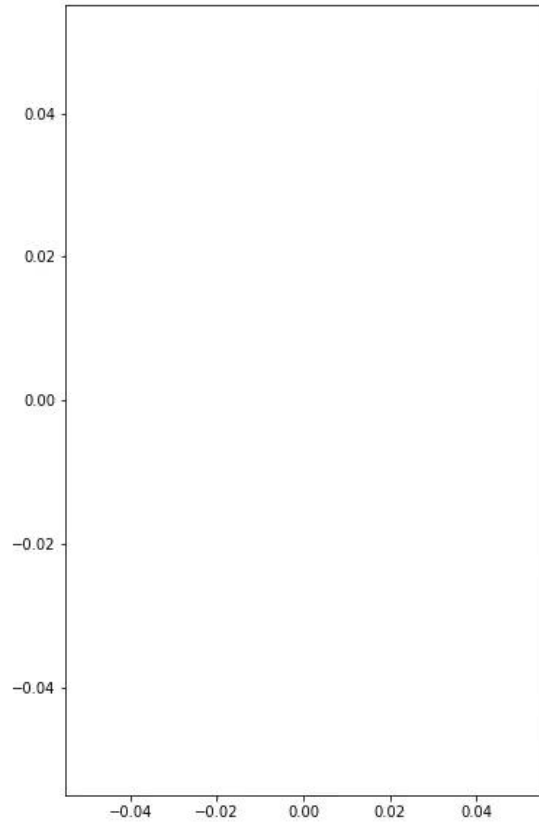
Limitations – Wrong/Noisy Gradients?



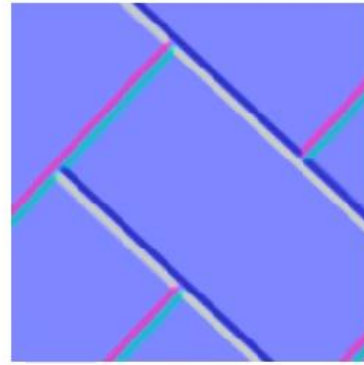
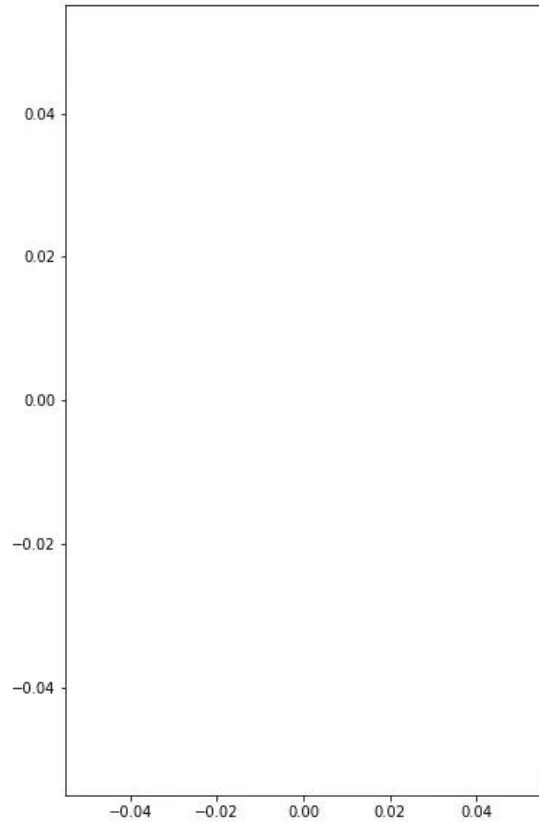
Limitations – Wrong/Noisy Gradients?



Limitations – Wrong/Noisy Gradients?



Limitations – Wrong/Noisy Gradients?



- Finding and fixing potential bugs
- More optimized implementation
- Generic geometry
- BSDF estimation