



Rendering atmospheric clouds using neural networks

Author: Mikhail Panin, ITMO

Scientific Supervisor: Anton Kovalev, Assistant Prof., ITMO

Scientific Advisor: Sergey Nikolenko, PHD, Head of AI lab, PDMI RAS

Clouds are HARD to render

- Light bounces thousands of times
- In a non-deterministic manner
- And we have to simulate it.

Old Fashioned Solution (path-tracing)

Rendering clouds by simulation of light bounces.

Pros:

- Physically correct
- Any level of accuracy

Cons:

- Takes over 24 hours per image



Modern Solution (By Disney Research)

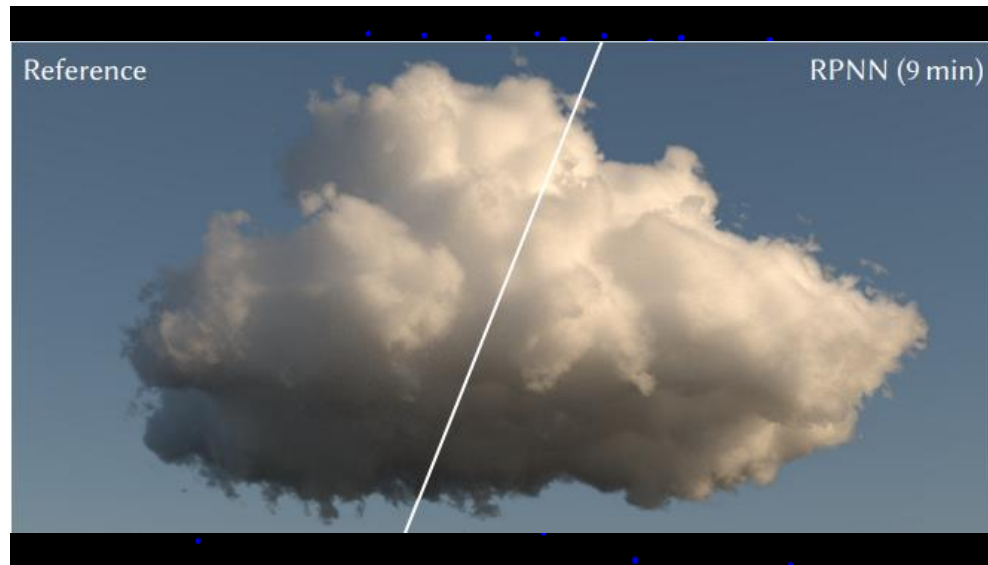
Approximating out coming light using neural network.

Pros:

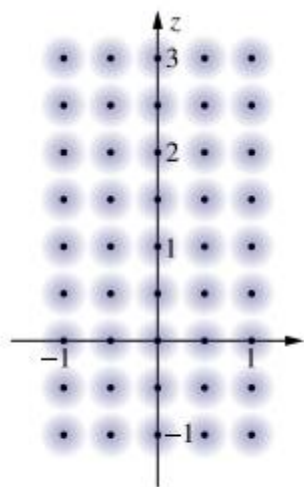
- Takes a few minutes per image
- Can be trained for various looks

Cons:

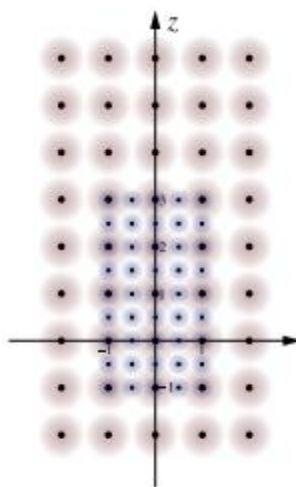
- There are corner cases
- Still not fast enough



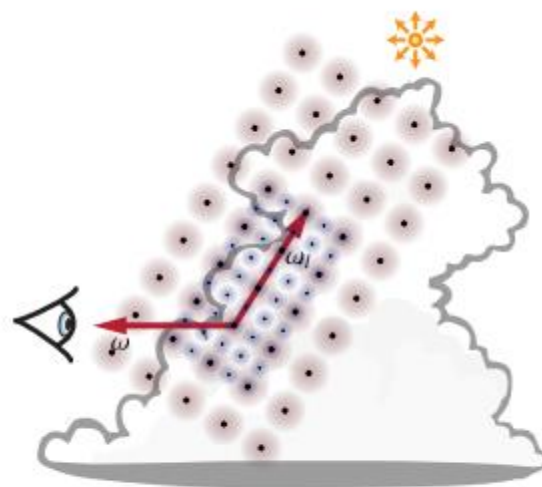
Descriptor



(a) Stencil grid



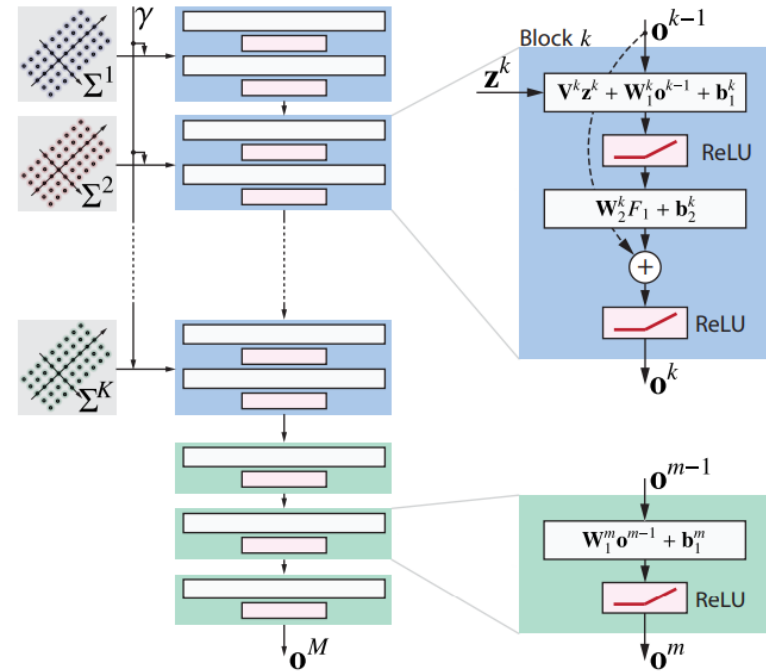
(b) Two levels



(c) Local frame

Neural Network

- Feed the descriptor layer-by-layer
- Three fully connected in the end



Goal

- Beat Disney's performance
- Come close to a real-time solution.

Objectives

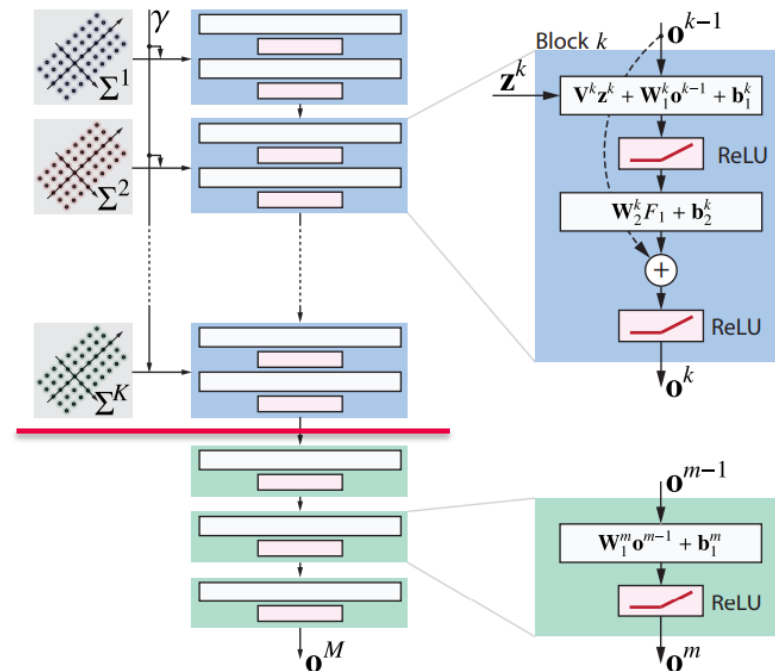
- Implement a Monte-Carlo path-tracer (Ground Truth, data source)
- Collect a dataset of clouds
- Collect a dataset of [LightDirection, SunDirection, Descriptor, Radiance]
- Train Disney style Neural Network
- Train a faster Neural Network
- Implement NN rendering on GPU
- Compare images and performance

The new idea (backing of Neural Network)

What if we could pre bake the output of the first K layers?

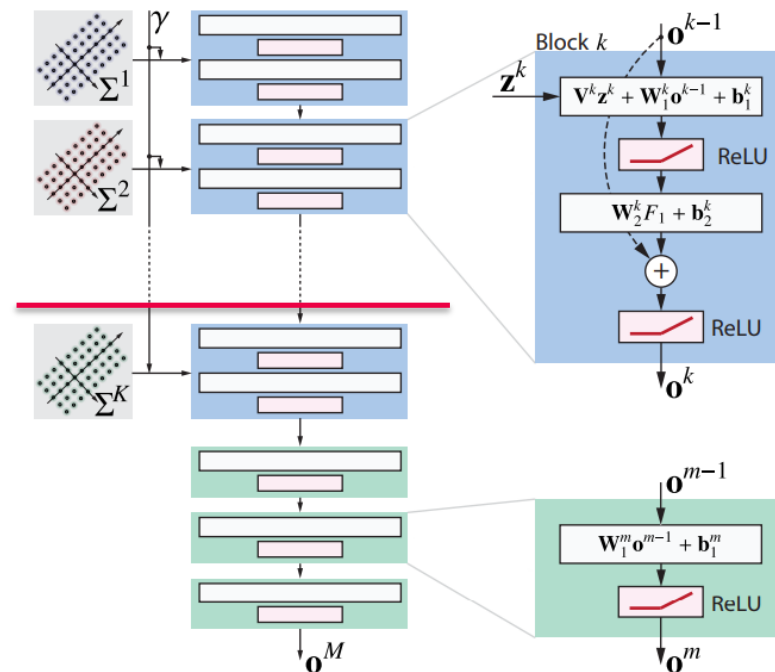
Problems:

- Variable γ
- Variable view direction
- Variable sun direction
- Memory footprint



The new idea (backing of Neural Network)

- Precompute only the first N blocks
- Freeze the sun direction
- Use the view direction only in the last layers
- Different Descriptor for the baked layers



Results

- Improved performance
- And the same image quality.



What's next?

Even faster rendering of dense clouds

- By only baking the values on the surface
- And using rasterization instead of raytracing

At 10 FPS or more?



Thank you for your attention!

Questions?