Simulation of the multi-view imaging system with differentiable ray tracing

Maxime Vandegar, Michael Kagan
Murtaza Safdari, Ariel Schwartzman

March 2021

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

Goals

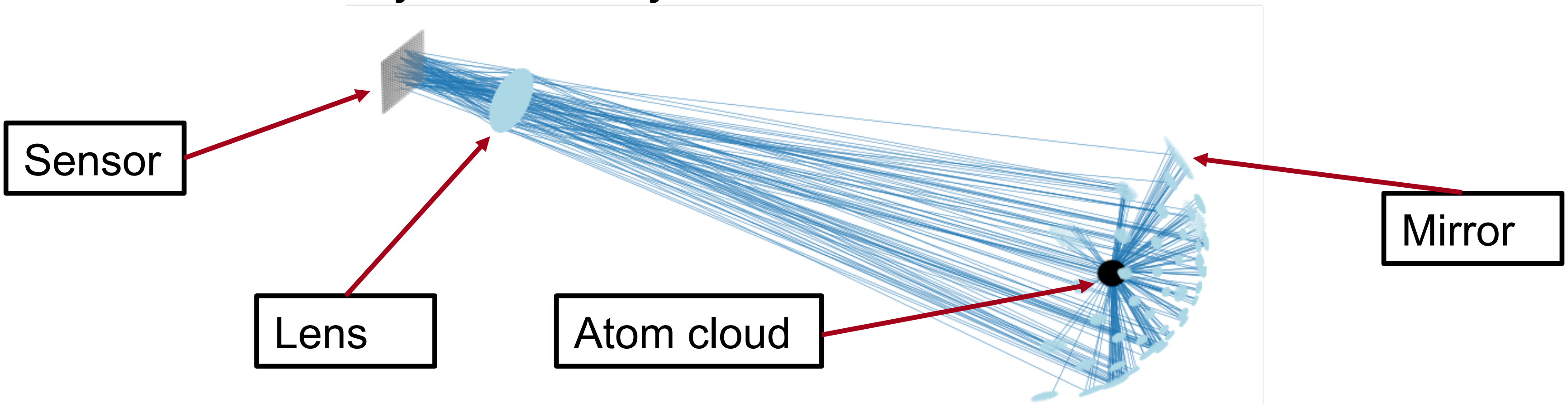
- System insights:
  - Provides image modeling for potential system design.
  - Design optimization.

- Data analysis (i.e. parameter inference):
  - Data / simulation comparison.
  - Systematic uncertainty modeling.
  - 3d reconstruction & parameter estimation.

Approach

- A simulator to map photon emission from the atom cloud to a multi-view image on sensor.

- Physics-driven approach to rendering images on sensor.
  1. Stochastic emission from the atom cloud.
  2. Trace light rays through the optical system.
     - Includes interaction with lenses, mirrors, apertures, etc.
  3. Trace rays until they hit the sensor.



Software design

- Implemented in Python within automatic differentiation framework (PyTorch).

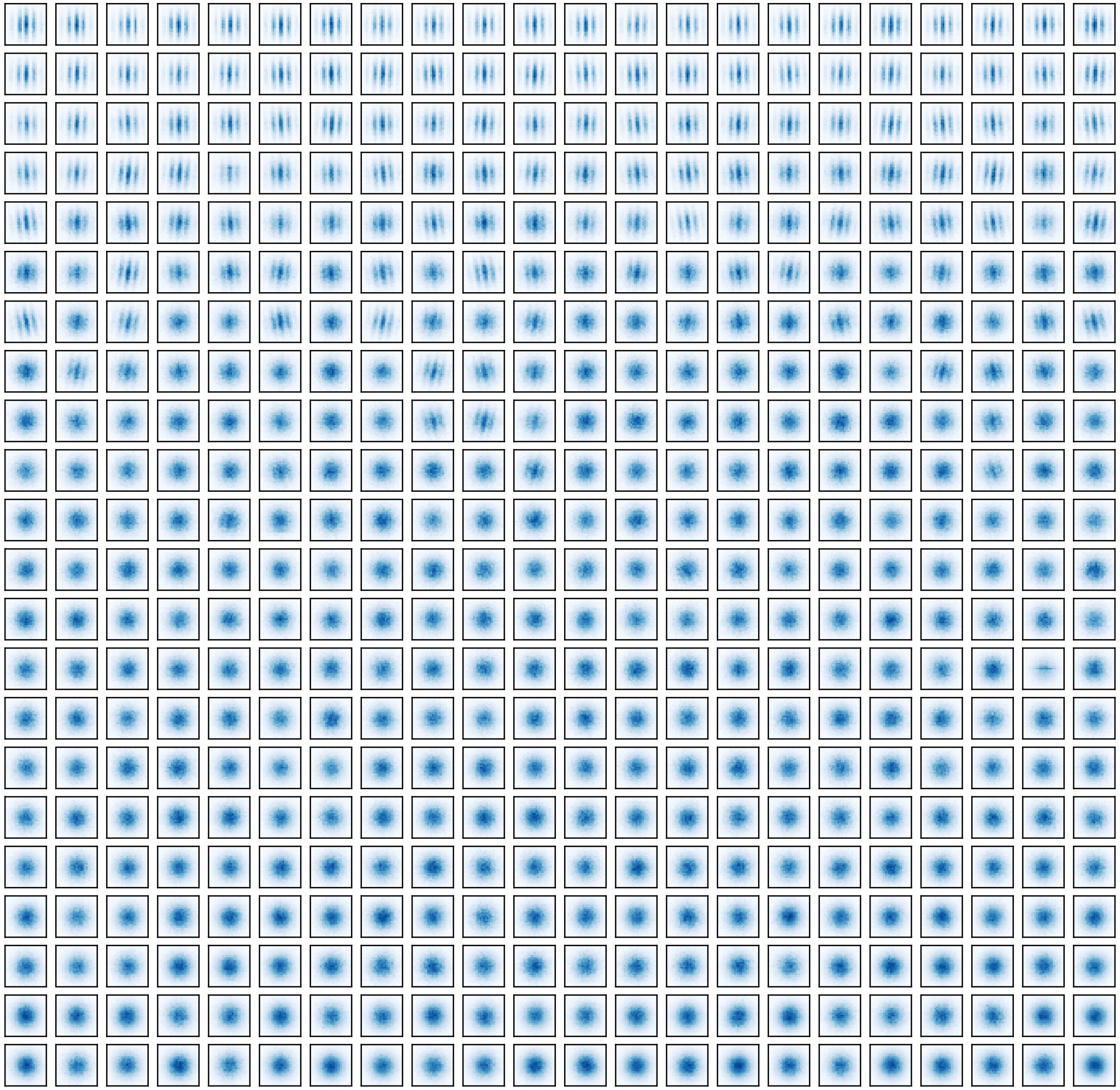
- Every traced ray is differentiable end-to-end.
  - Enables gradient descent from images back to atom cloud / system properties for optimization & parameter estimation.
  - Natively ready for interfacing with machine learning algorithms & pipeline.

- High throughput.
  - Vectorizable – trace many rays at once.
  - Parallizable – over multiple devices.
  - Support on CPU / GPU.
  - Just-in-time (jit) compilation for reducing computational time.

Simulator’s fidelity

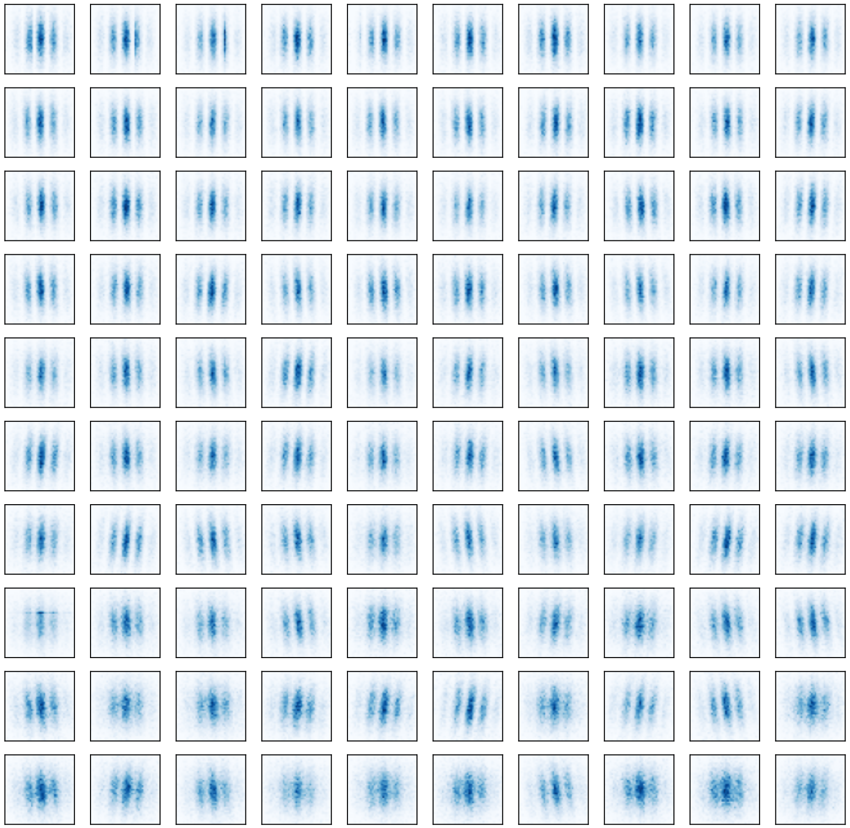
- Currently using perfect lens / mirror / sensor approximation.
  - Thin lens model, no readout noise, no quantum efficiency, …
- Easy to add complexity and noise to the system.
  - Which should we add first?

Simulated views (1)

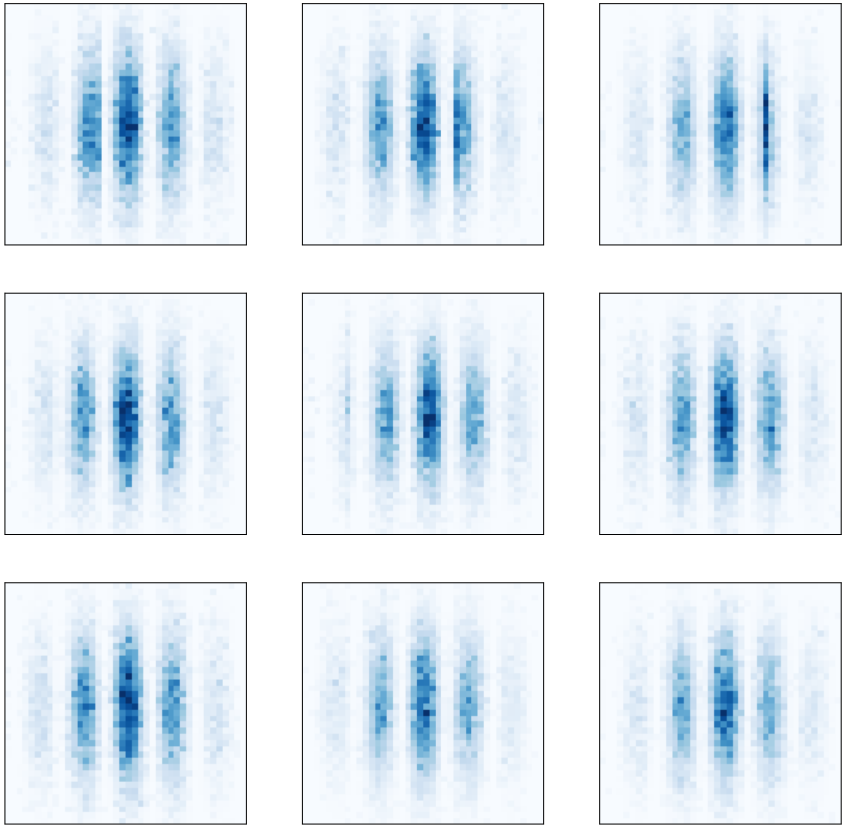


Patches of 40 x 40 pixels (500 mirrors)

Simulated views (2)

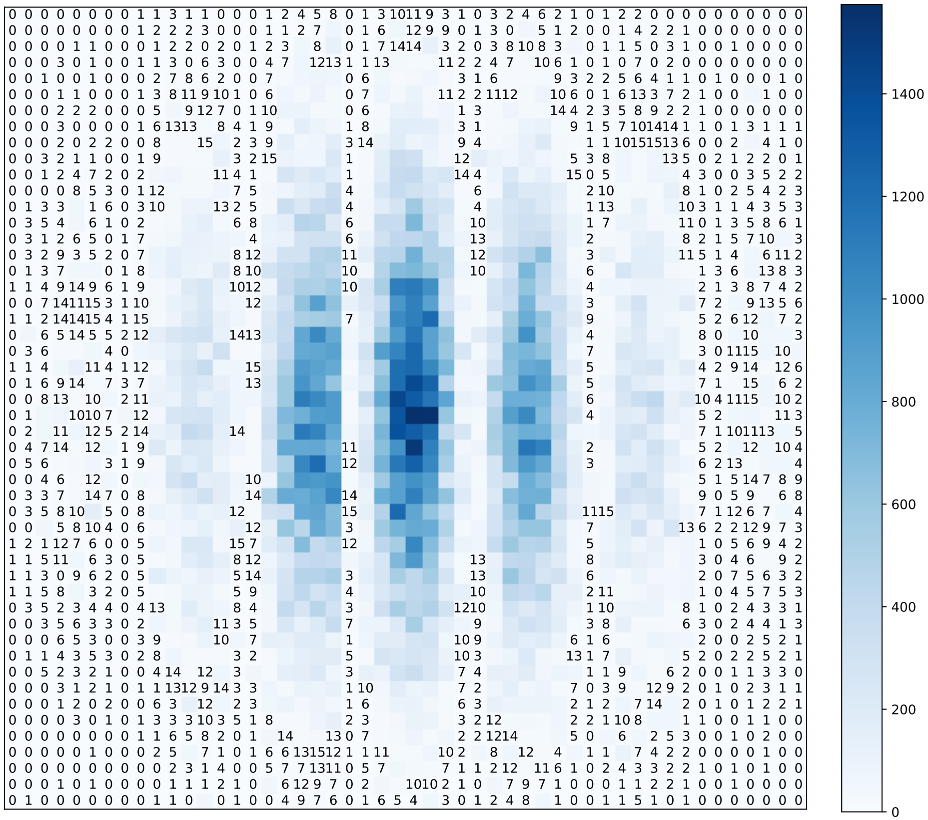


Views from the 100 mirrors (out of 500) that are the closest to the optical axis.

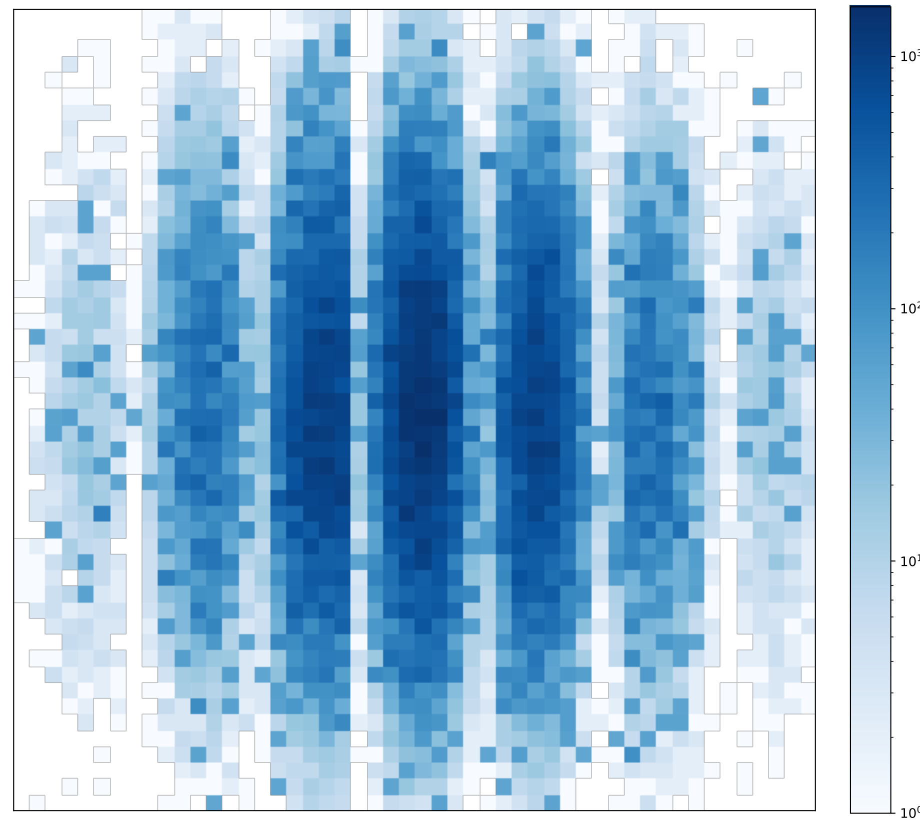


Views from the 9 mirrors that are the closest to the optical axis.

Simulated views (3)



View from the mirror that is the closest to the optical axis (50 x 50 pixels).



View from the mirror that is the closest to the optical axis (logarithmic norm).

Inference

SLAC

- The simulator will be a key component in the inference pipeline.
- Enables 3d reconstruction & simulation-based inference in order to learn confidence intervals or posterior distributions over $\boldsymbol{\phi}$ or $\Delta\boldsymbol{\phi}$.
- The simulator's differentiability allows solving $||A(x) - b||^2$ (or other reconstruction objectives) with gradient descent.
  - b is an observed image.
  - A is the differentiable simulator.
  - x is the model of the atom cloud.
    - E.g. parametrized wave function, voxel, neural network, implicit function, …

More about physically based rendering

SLAC

- Photon mapping.
  - Rays are traced from source to sensor (current approach).
  - Requires to model the interaction of 1B rays with the system (computational expensive).
    - Makes it difficult to compute $||A(x) - b||^2$ and its gradients in a tractable manner.
- Path tracing:
  - Rays are traced from sensor to source (being implemented).
    - Integrate radiant flux over the ray's path.
  - Most common approach used in computer graphics.
  - Allows to query the value of a single pixel and do per-pixel minibatch gradient descent.

Next steps

1. Finish the path tracing implementation.
2. First implementation of 3d reconstruction & maximum likelihood estimation of parameters given a wave function from.
3. Add more sources of noise & complexity to make the simulator more realistic.
4. Interface the simulator with ML approaches to simulation-based inference.

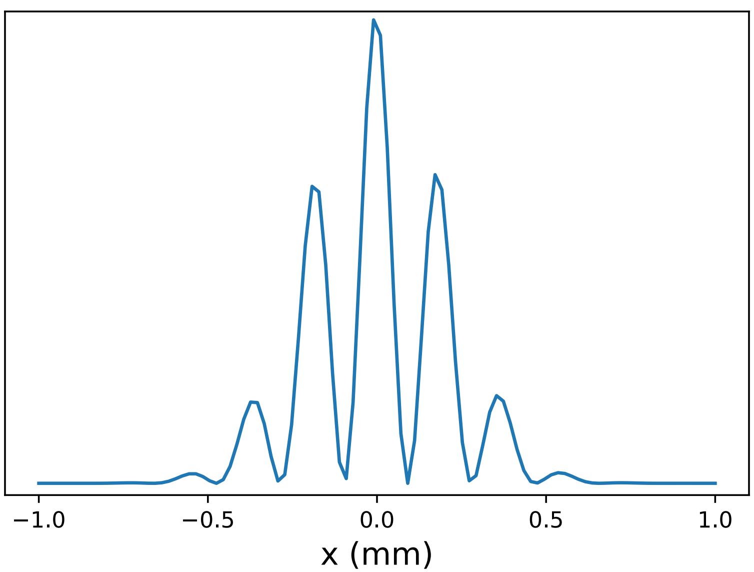
Questions

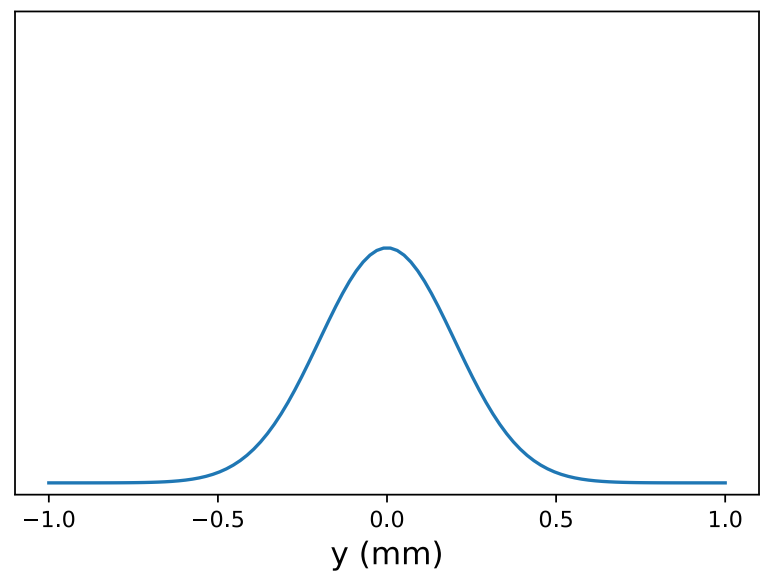
SLAC

- Does the interference pattern always face the same direction?
- Do we need to consider any coherence between emitted photons?
- Given the low number of photons, can we still model the photon paths with classical geometric optics?
- Should we target including lasers-induced noise sources in the near term?
- Are there existing simulators / resources for understanding how to simulate the wave function propagation through the pipe.

Interference pattern

$$\frac{i\left(\frac{2}{\pi}\right)^{3/4}(m w_0)^{3/2}\left(\frac{1}{\sqrt{2}}+\frac{e^{i\left(\text{aQuad kFringe}^2 x^2+\text{kFringe } x\right)+i\phi}}{\sqrt{2}}\right)e^{-\frac{m\left((x-x_A)^2+(y-y_A)^2+(z-z_A)^2\right)}{4 m w_0^2+2i\,\text{tFinalBS}\,\hbar}}}{\sqrt{-\left(2 m w_0^2+i\,\text{tFinalBS}\,\hbar\right)^3}}$$





Gradients and 3d reconstruction

- The simulator's differentiability allows forwarding per-pixel gradients to the cloud parameters (chain rule).