Detection of Atomic Fluorescence and Parallel Computing for Phase-Only Holography

Nathan Constantinides
College Park Scholars – Science & Global Change Program | Physics
nconstan@terpmail.umd.edu | CPSP359G
College Park Scholars Academic Showcase, May 3, 2024

Introduction

In the field of atomic physics, recent work on experiments of laser trapping arrays of neutrally charged atoms has advanced as a platform for investigating quantum information and computing. I study compare efficiency of experimental methods to detect fluorescence curves as well as the computationally efficient generation of holograms for creating arrays of laser traps.

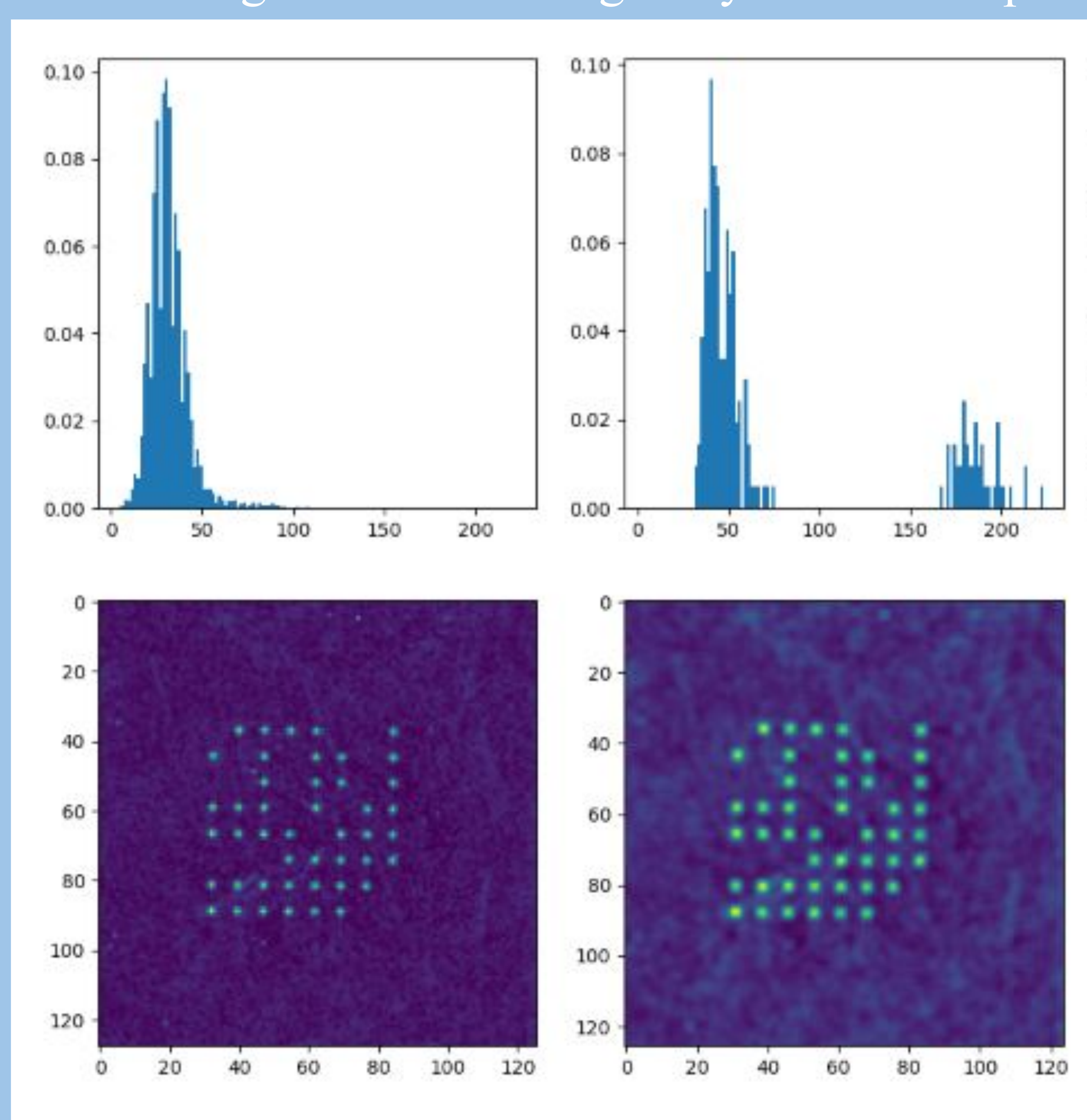

Figure: An example comparison of point resolution with and without rolling ball subtraction post processing. Bottom Left: Raw camera data. Bottom Right: Camera data after binning and post-processing Top Left: Pixel photon counts histogram. Not well resolved Top Right: Pixel photon counts histogram after post-processing. Effectively bimodal.

Site Information:

Physical Sciences Complex | 4296 Stadium Dr, College Park, MD 20742
Professors Trey Porto and Steve Rolston
Mission: To Support Research in Fundamental and Applied Physics
Site Goals: Perform novel experiments on neutral atom arrays

Methods:

To build a dataset, I set up a small 2f imaging system and took 40 images of various dot patterns. After this, I implemented the binarization and peak detection routines with Numpy [1] and Scikit-Image [2], as well as various pre and post processing methods including rolling ball filtering [2] and convolution. I compared detection fidelity using the function described in [3].

To generate phase patterns efficiently I implemented the modification of the Gerchberg-Saxton routine in [4] in CUDA, taking advantage of parallelism when performing large summations and convolutions.

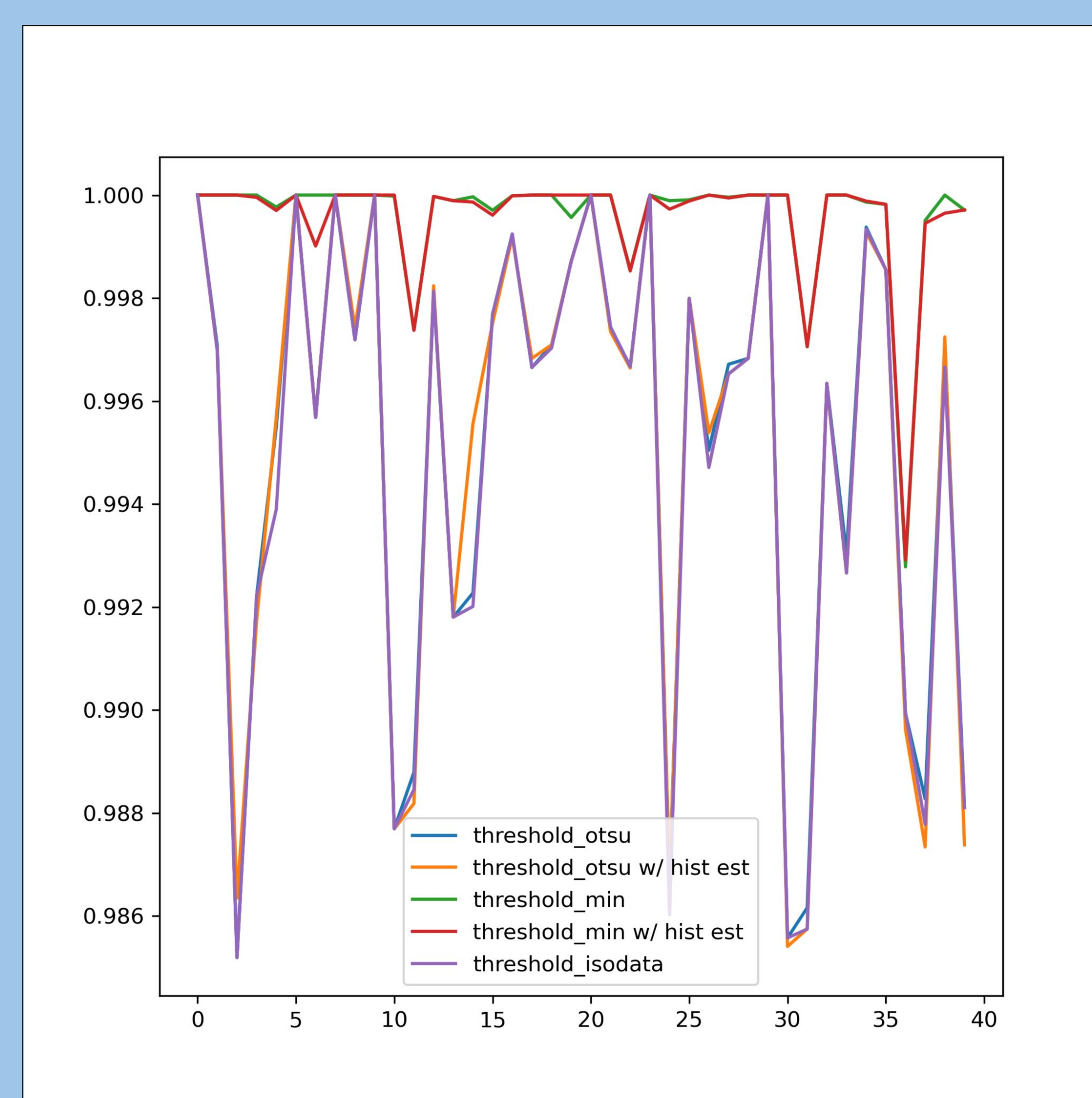

Figure: Comparison of different atom detection algorithms over an image dataset of 40 images. Y-Axis is fidelity (1 best), X-Axis is image number.

Scikit's threshold_min showed the best fidelity on all images.

Results: Optimization of the phase pattern generating routine brought runtimes down from ~2 seconds to ~4 milliseconds per frame. Comparison of detecting procedures highlighted the need for preprocessing of images with rolling ball filtering and superior performance of the threshold_min algorithm.

Discussion: Further testing of both the SLM phase pattern generation and image detection routines on experimental hardware is required to fully validate these findings. Future work in development of SLM phase retrieval algorithms is especially interesting, as current procedures are highly limited and are not guaranteed to converge.

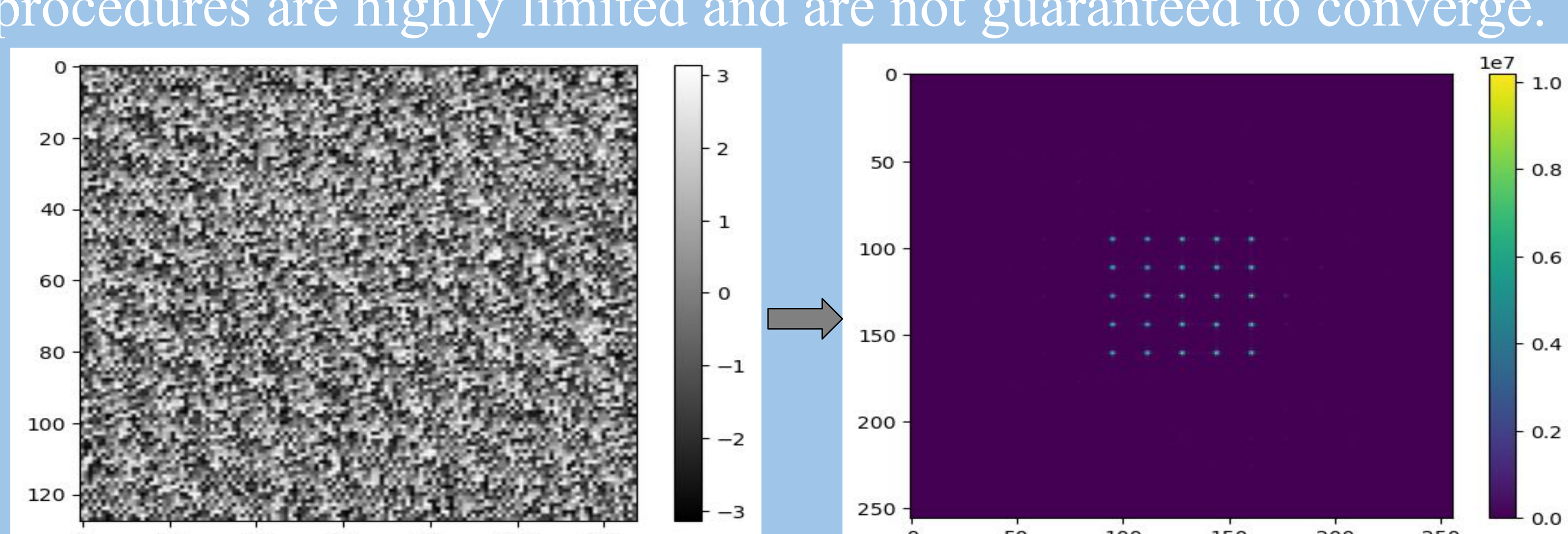

Figure: Generated phase pattern (left) and simulated associated light field intensity for an SLM (right)

Acknowledgments:

I would like to thank Professors Steve Rolston, Trey Porto, Ian Spielman, and Dr. Madison Anderson for their patience and encouragement as mentors on this project, and Professors Thomas Holtz and John Merck for their support and advice in the Science and Global Change Program.

Citations:

[1] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2. (Publisher link).
[2] Van der Walt, S., Sch"onberger, Johannes L, Nunez-Iglesias, J., Boulogne, Franccois, Warner, J. D., Yager, N., … Yu, T. (2014). scikit-image: image processing in Python. PeerJ, 2, e453.
[3] Madjarov, Ivaylo Sashkov (2021) Entangling, Controlling, and Detecting Individual Strontium Atoms in Optical Tweezer Arrays. Dissertation (Ph.D.), California Institute of Technology. doi:10.7907/d1em-dt34. https://resolver.caltech.edu/CaltechTHESIS:01292021-001639979
[4] Leseleuc de kerouara, S. de. (2018). Quantum simulation of spin models with assembled arrays of Rydberg atoms [Phdthesis]. http://www.theses.fr/2018SACLO007/document