Third Summer School on ML for Electron Microscopy

May 19 – 23 2025

Methods of Atom Position determination

- Participant Computer
- Kamyar Barakati

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Key image-analysis steps used in the notebook

1. Threshold + Centroid
2. LoG + Blob Detection
3. CNN-based atom localization → U-Net

→ Point-Spread Function fitting

→ Richardson Lucy Deconvolution

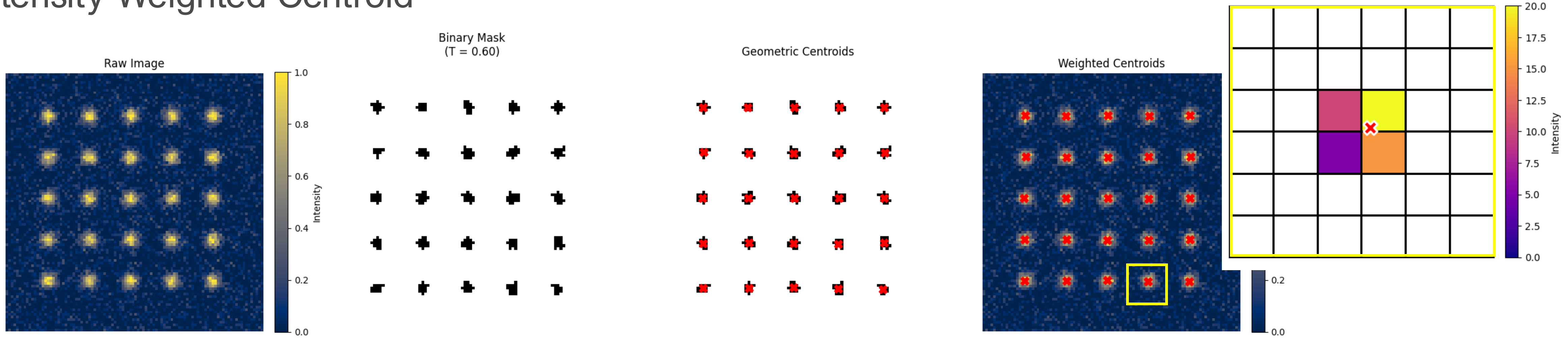
Threshold + Centroid

$$B(x,y) = \begin{cases} 1, & I(x,y) \geq T, \\ 0, & I(x,y) < T, \end{cases}$$

$$\bar{x} = \frac{1}{N} \sum_{(x,y)\in R} x, \quad \bar{y} = \frac{1}{N} \sum_{(x,y)\in R} y$$

$$\bar{x} = \frac{\sum x\, I(x,y)}{\sum I(x,y)}, \quad \bar{y} = \frac{\sum y\, I(x,y)}{\sum I(x,y)}.$$

- Cleaning up
- Centroid calculation
- Intensity-Weighted Centroid



Limitations:

Sub-pixel accuracy is limited (~0.5 px)

Sensitive to threshold choice

doesn't account for the known blur shape (PSF), so can't optimally reject outlier pixels or fit partial peaks.

LoG + Blob Detection

Raw Step Function

$$n_\sigma(t) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\!\Big(-\frac{t^2}{2\,\sigma^2}\Big).$$

Gaussian Smoothing

$$f_s(x) = (n_\sigma * f)(x) = \int_{-\infty}^{\infty} n_\sigma(t)\, f(x-t)\, dt.$$

Replaces each point f(x)) by a weighted average of its neighbors

First Derivative

$$\frac{df_s}{dx}(x) = \frac{d}{dx}(n_\sigma * f)(x) \;\approx\; \frac{f_s(x+\Delta x) - f_s(x-\Delta x)}{2\,\Delta x}.$$

rising and falling edges

Second Derivative

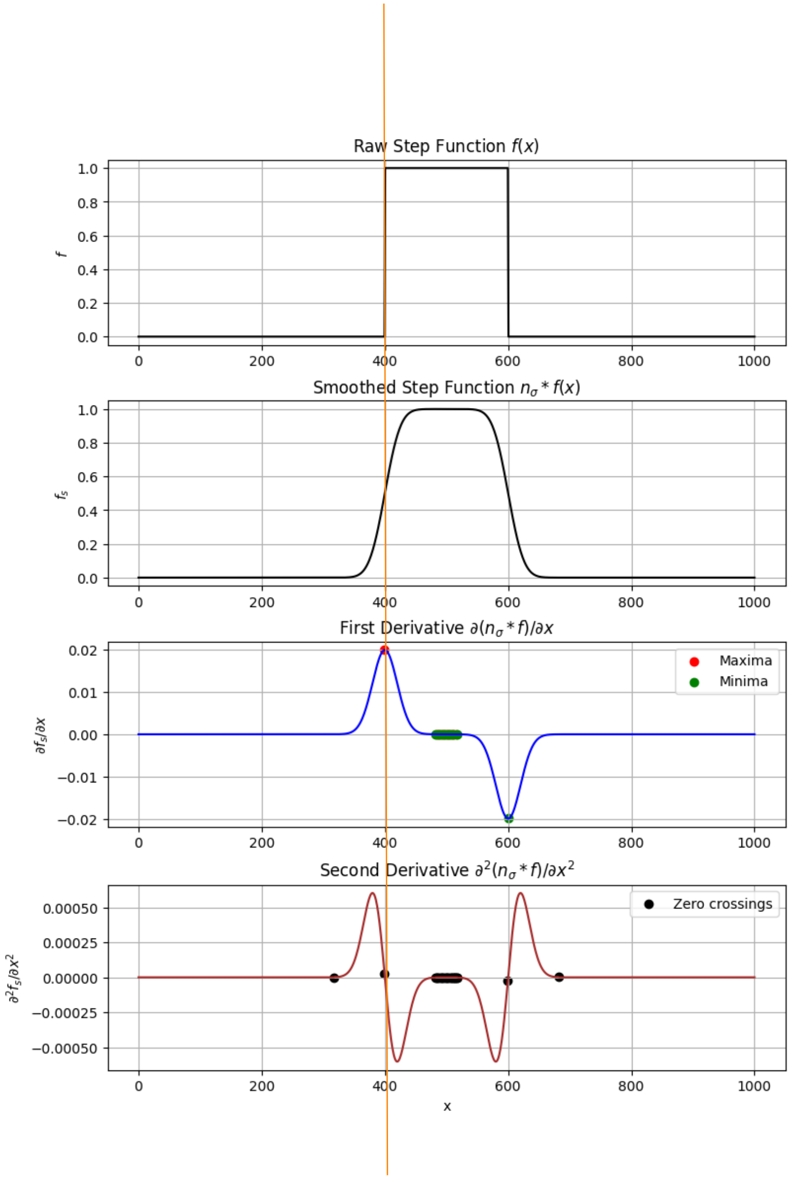
$$\frac{d^2 f_s}{dx^2}(x) = \frac{d^2}{dx^2}(n_\sigma * f)(x) \;\approx\; \frac{f_s(x+\Delta x) - 2f_s(x) + f_s(x-\Delta x)}{\Delta x^2}.$$

$$\nabla^2 I|_{i,j} \approx \frac{1}{\varepsilon^2}\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

4-Neighbor

$$\nabla^2 I|_{i,j} \approx \frac{1}{6\,\varepsilon^2}\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$

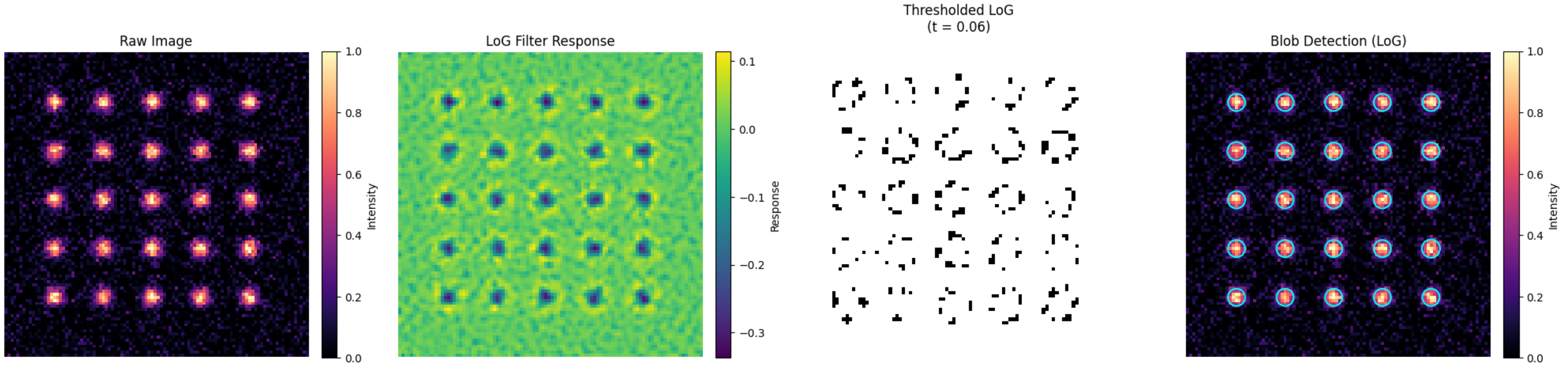
8-Neighbor



LoG + Blob Detection

Gaussian-smooth with **sigma=1**

blobs = min_sigma/max_sigma → the expected range of spot radii (in pixels)
num_sigma=10,
threshold=0.06 → the minimum normalized LoG response at which to accept a blob



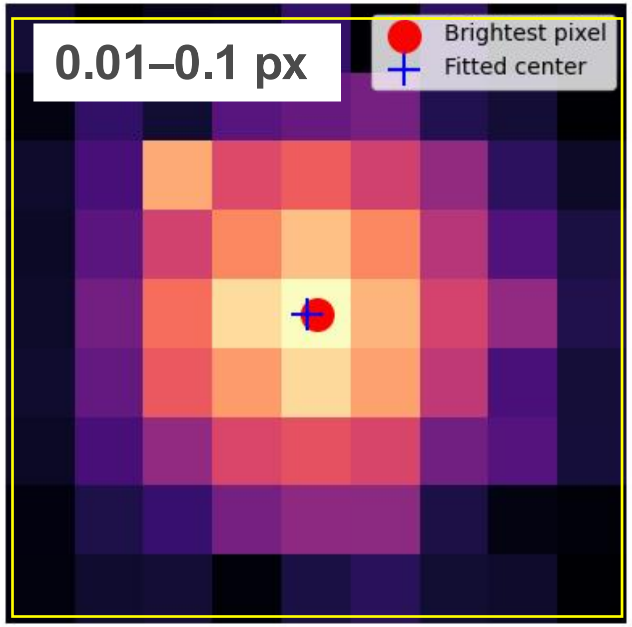
Limitations:

Hyperparameters to tune

Fixed, isotropic kernel

Real PSFs may be anisotropic or non-Gaussian. → asymmetric blur shapes

Point-Spread Function fitting

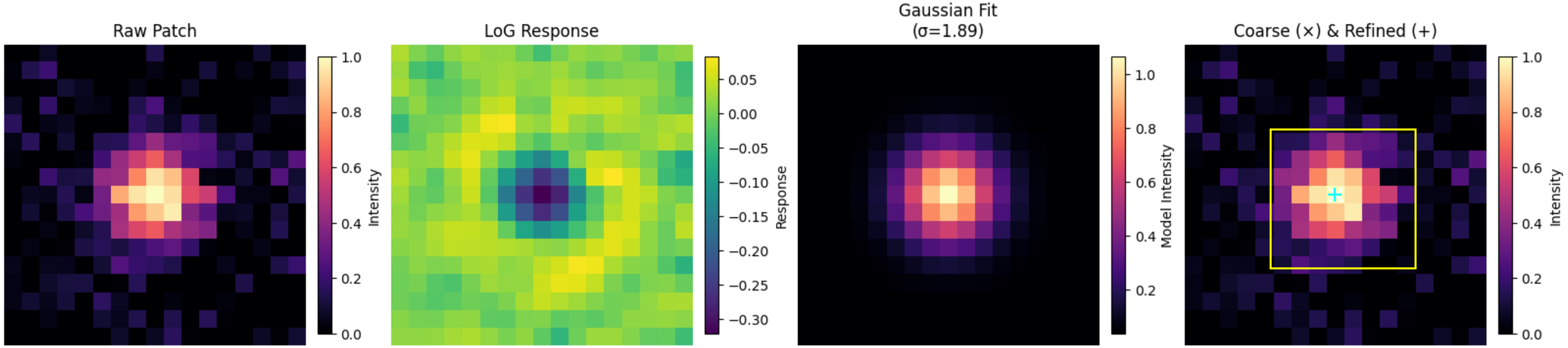


Refining atom or blob locations to sub-pixel accuracy

1. LoG → xi, yi → **0.5 px**
2. Extract ROI → cut out a small square patch
3. Fit a Gaussian Model → on each patch, assume the spot looks like a little 2D Gaussian hill plus a flat background

$$\text{model}(x, y) = A\ \exp\left(-\frac{(x-x_0)^2 + (y-y_0)^2}{2\,\sigma^2}\right) + B.$$

You adjust the hill's location, size, and brightness so its shape lines up with the actual pixel values as closely as you can.



Richardson Lucy Deconvolution

1. **observed image → $g(x, y)$**
2. **true object → $f(x, y)$**
3. **PSF → $h(x, y)$**

$$g = h * f \text{ +Poisson noise}$$

goal is to estimate **f** given **g** and **h**

$$f^{(n+1)}(x,y) = f^{(n)}(x,y) \times \left[h_{\text{flipped}} * \left(\frac{g(x,y)}{(h * f^{(n)})(x,y)}\right)\right]$$

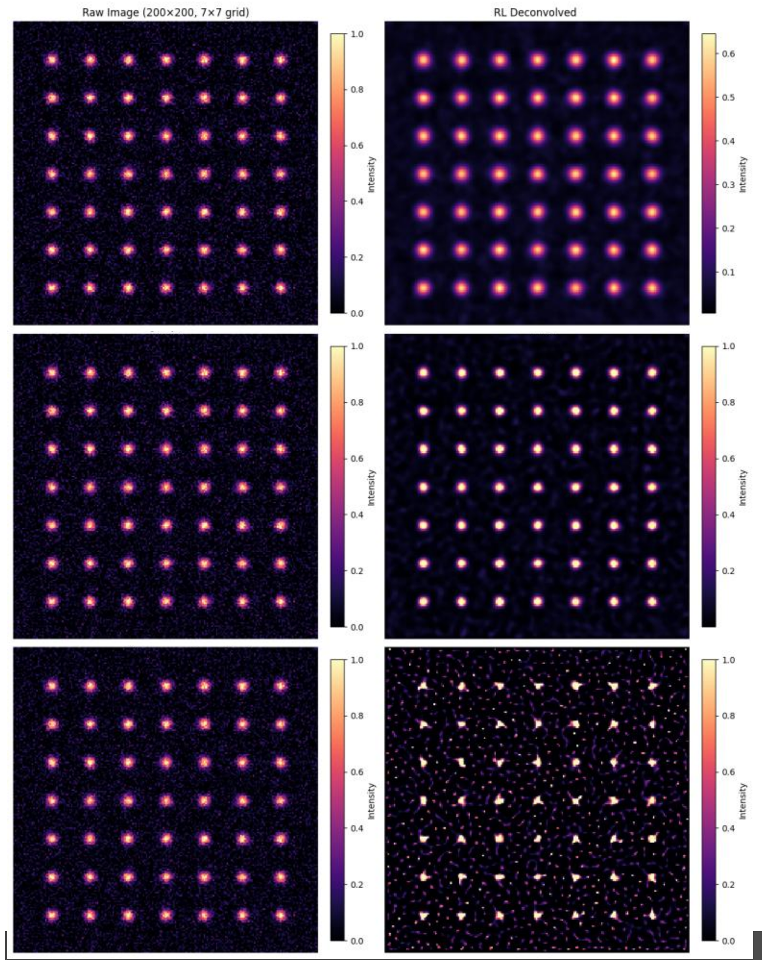
Where f_0 is our initial guess

$(h * f^{(n)})(x,y)$

→ Current blurred image

What RL Does for You? what your microscope would see if the current guess were true

1. **Sharpens** blurred spots, recovering high-frequency detail up to the optical cutoff.
2. **Suppresses** out-of-focus haze and background smear.
3. **Maintains** photometric fidelity (total intensity) under Poisson noise.
4. **Converges** to the maximum-likelihood solution for Poisson-distributed data.



1

5

100

overshoot around **bright spots**, creating "ghost" halos


Participant Computer
notebook
THE UNIVERSITY OF TENNESSEE KNOXVILLE