Automation of Optical Tweezers & Tracking Applications for EV Radii Prediction

Noah Horne

UiT The Arctic University of Norway

July 10, 2025

Overview

MSD inferred
Particle Radius & Diffusivity

Brownian Diffusion and Mean Squared Displacement (MSD) Relationship

- EV's and other nanoscale particles undergo Brownian motion.
- Nanoscale particle densities stochastically obey the diffusion relation.
- Rearranging yields a linear relationship between particle MSD and lag time.
- With temperature T and viscosity η the particle radius r is calculated.

$$\frac{\partial \rho(\vec{x}, t)}{\partial t} = D\nabla^2 \rho(\vec{x}, t), \quad \vec{x} \in \mathbb{R}^d$$

$$\Downarrow$$

$$\rho(\vec{x}, t) = \frac{1}{(4\pi Dt)^{\frac{n}{2}}} \exp\left(\frac{-|\vec{x}|^2}{4Dt}\right)$$

$$\Downarrow$$

$$\mathbb{E}[x] = 0, \; Var[x] = 2nDt$$

$$\Downarrow$$

$$\boxed{\frac{\mathbb{E}[x^2]}{2dt} = D = \frac{k_B T}{6\pi\eta r}}$$

Particle Detection & Image Preprocessing

1. Bandpass Filter
   - Low-pass truncated 1D Gaussian Convolution:
     $G(x) = \frac{1}{N}\exp(\frac{-x^2}{2\sigma^2}), \quad x \in [-\mathrm{r}\cdot\sigma, +\mathrm{r}\cdot\sigma]$
   - High-pass 1D Boxcar Convolution:
     $B(x) = \frac{1}{2r+1}, x \in [-r, r]$
   - Difference of these convolution results:
     $I_{bp}(x,y) = \underbrace{G \star I}_{\text{Low-pass}} - \underbrace{B \star I}_{\text{Long-pass}}$
2. Maximum Detection & Binary Thresholding
   - Peak detection via grayscale morphological dilation
   - Binary mask (thresholded with hyperparamter)
3. Subpixel Refinement
   - Particle c.o.m found to subpixel accuracy.

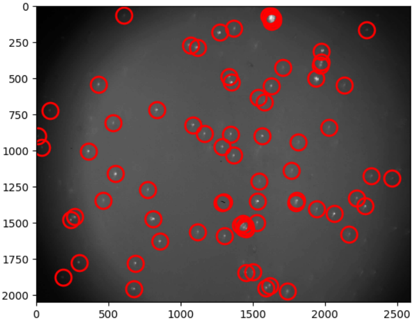


Figure: $r = 0.5\mu$m Polystyrene Detection

Trajectory Analysis

- Image preprocessing and particle location detection is performed on all frames of captured video.
- **Crocker-Grier** linking algorithm enumerates particles and assigns trajectories.
- Analysis is then free to be performed on returned dataframe.

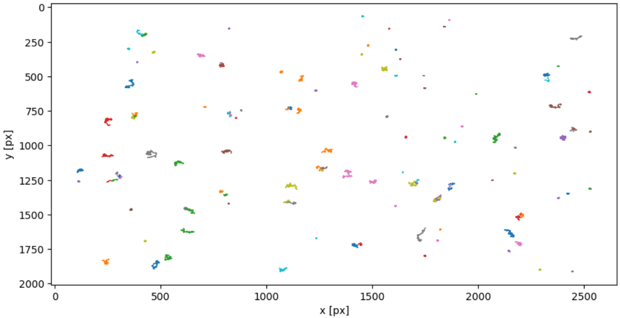


Figure: 2μm Polystyrene Trajectories over 120 frames.

Tracking Parameters / Calibration

The tracking calibration procedure allows for 5 levels of configuration:

1. Brightness Thresholding & Mask Calibration (frame annotation)
2. Custom Thresholding via Cluster Detection (frame annotation)
3. **Trajectory Linkage & Memory** (dynamics capture)
4. **Ephemeral Trajectory Thresholding** (dynamics capture)
5. Visual Pixel Density / FPS config (physical result calibration)

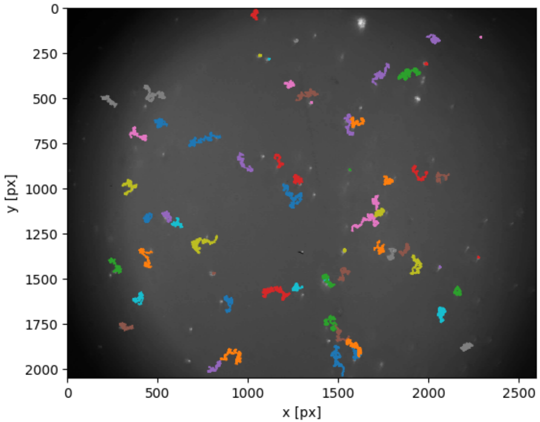


Figure: Short Memory, Long Ephemeral Threshold

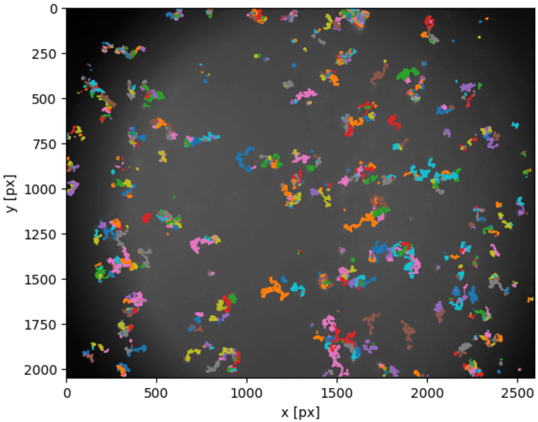


Figure: Long Memory, Short Ephemeral Threshold

Mean Squared Displacement Plots

- The gradient of the MSD plot permits Diffusivity D & Radii r calculation.

- MSD is measured as a function of lag time, returning the average displacement a particle goes through across said lag time, over the entire video sample (N frames).

$$MSD(\tau) = \frac{1}{N-\tau} \sum_{i=1}^{N-\tau} \left[(x_{i+\tau} - x_i)^2 + (y_{i+\tau} - y_i)^2 \right]$$

$$\frac{1}{4} \frac{MSD(\tau)}{\tau} = D \quad \text{(For 2D Case)}$$

- Larger τ_{max}: Richer dynamics captured, high temporal variance.
- Smaller τ_{max}: Less complete dynamics, low temporal variance. (more stable)

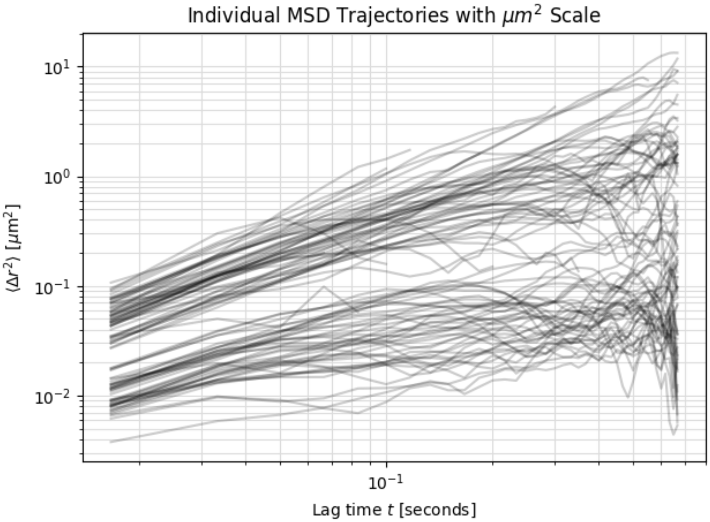


Figure: Individual 2μm Polystyrene Mean Squared Displacements fit with 15 frame memory.

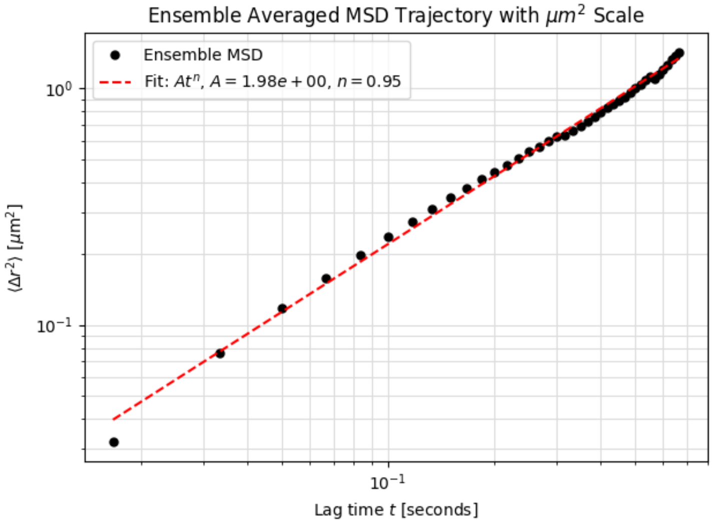


Figure: Ensemble Averaged 2μm Polystyrene Mean Squared Displacements fit with 15 frame memory.

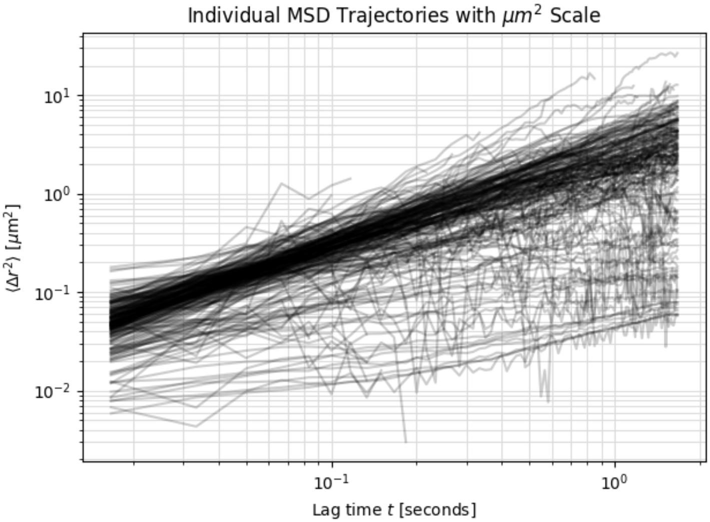


Figure: Individual 1.04μm Polystyrene Mean Squared Displacements fit with 15 frame memory.

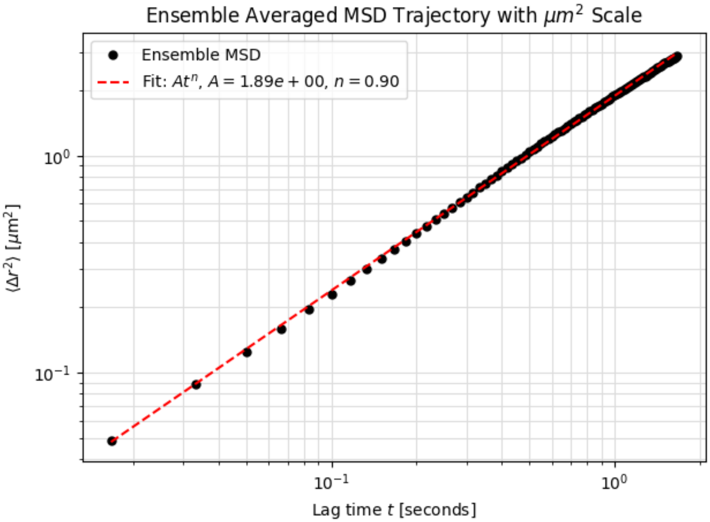


Figure: Ensemble Averaged 1.04μm Polystyrene Mean Squared Displacements fit with 15 frame memory.

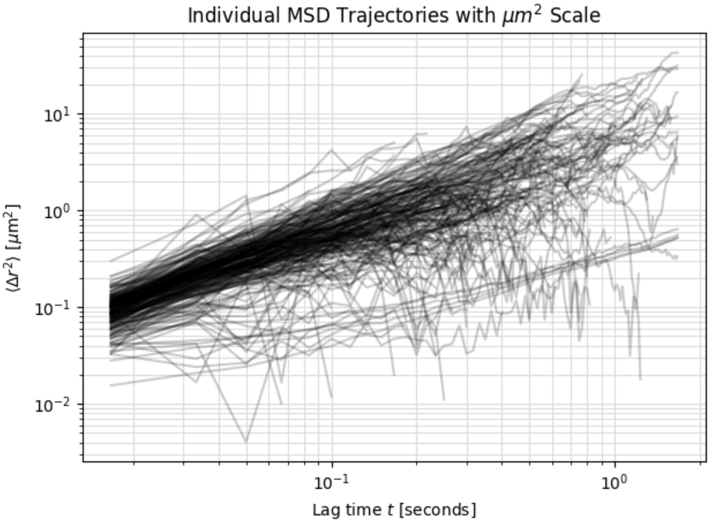


Figure: Individual 0.53μm Polystyrene Mean Squared Displacements fit with 15 frame memory.

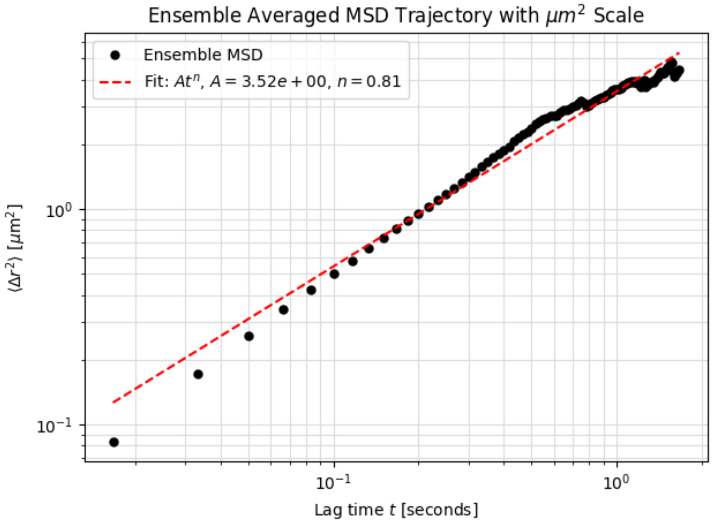


Figure: Ensemble Averaged 0.53μm Polystyrene Mean Squared Displacements fit with 15 frame memory.

Radii Prediction Accuracy

- The backend takes the gradient of the MSD fit to calculate the radii and diffusivity coefficient:

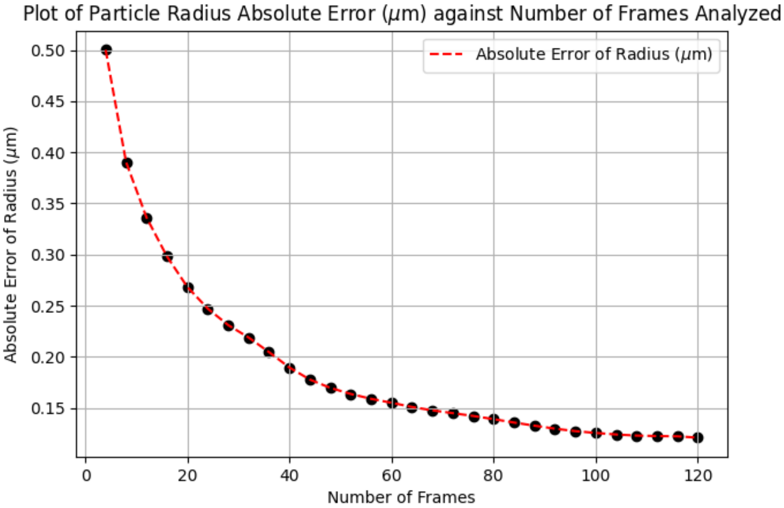
$$\frac{\mathbb{E}[x^2]}{t} = 4D = \frac{2k_B T}{3\pi\eta r}$$

Comparison between Literature Radii Values and Tracking Model Predicted Values for 100 frames			
Real Radii Values (μm)	1	0.52	0.265
Ensemble Predicted Radii Values (μm)	1.022	0.496	0.2303
Absolute Error (μm)	0.022	0.024	0.0347
Relative Error	0.022	0.046	0.1309

Note: These results use long memory, short ephemeral thresholds, and a long τ_{max}.

Estimates under this configuration break down for more diffusive particles.

Radii Prediction Accuracy: Video Length



Enhancing Stability for Single Particle Measurement

The goal is to strongly fit straight lines to single particle mean squared displacement plots for individual radii calculation.

Solution: **Short Memory, Long Ephemeral Threshold, Small** τ_{max}.

(+) Very small temporal variance, longer trajectories analyzed.

(-) Hard to ensure that the particle we want has a strong enough track. Only very stable particles are kept (careful calibration of the optical tweezer ROI necessary).

Before Changes

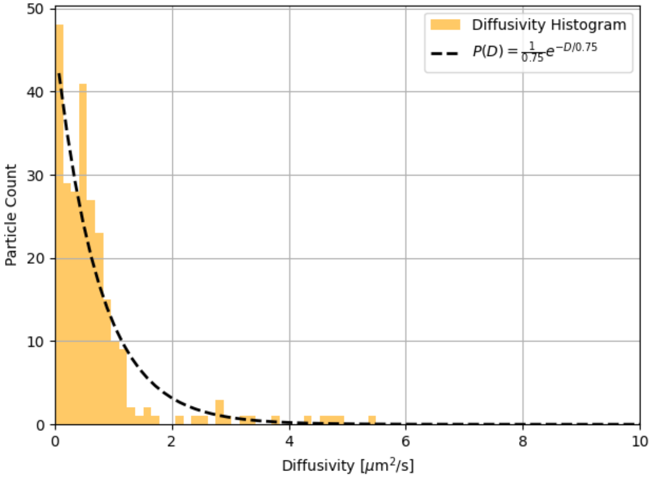


Figure: Diffusivity Histogram 1.04μm Polystyrene fit with $\tau_{max} = 60$

Before Changes

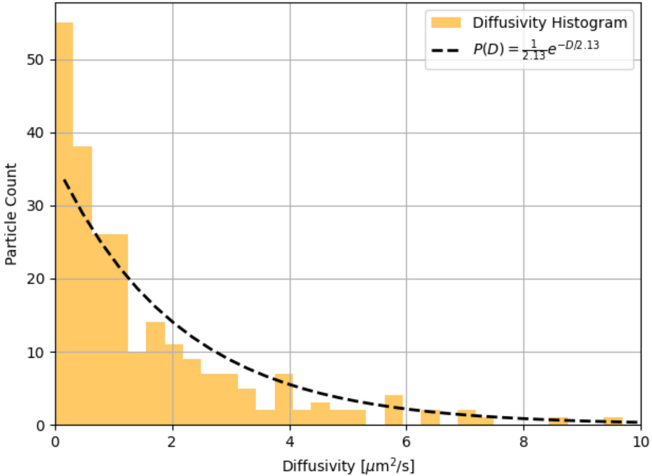


Figure: Diffusivity Histogram 0.53μm Polystyrene fit with $\tau_{max} = 60$

Reasoning for observing this exponential distribution (not desired)

- In a very noisy track with high temporal variance, the MSD ceases to be linear and anomalous diffusion begins. The diffusivity D_t consequently becomes governed by some PDF $\pi(D, t)$. (Chubynsky & Slater, 2018)

- One can further say that D is subject to some arbitrary constant noise d_0 (diffusivity), and bias force s_0 due to this temporal variance in the fit. With $J = 0$:

$$J = \frac{-\partial}{\partial D}[d_0 \pi(D, t)] - [s_0 \pi(D, t)]$$

$$\frac{\partial}{\partial D}[d_0 \pi(D, t)] = -[s_0 \pi(D, t)] \implies \boxed{\pi(D) = \frac{1}{D_0} e^{-\frac{D}{D_0}}}$$

- **Smaller particles will have diffusivity distributions of much higher variance! (scales with $\langle D \rangle^2$).**

After Changes

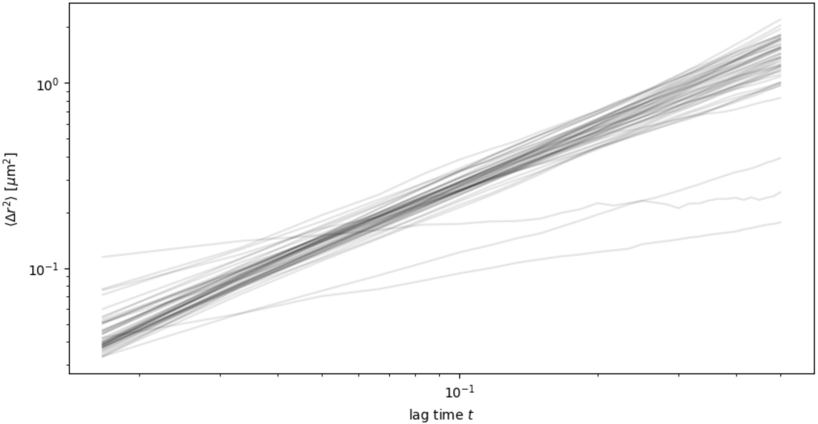


Figure: IMSD PLot for 1.04μm Polystyrene fit with $\tau_{max} = 15$

After Changes

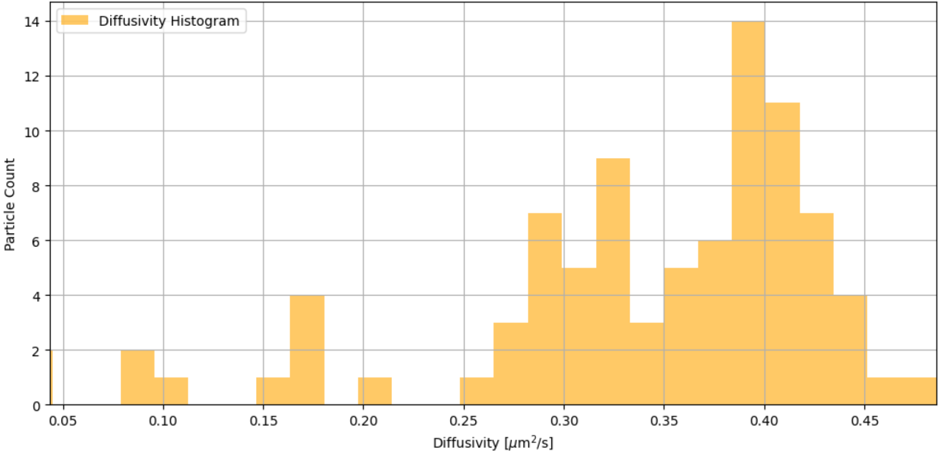


Figure: Diffusivity Histogram 1.04μm Polystyrene fit with $\tau_{max} = 5$

After Changes

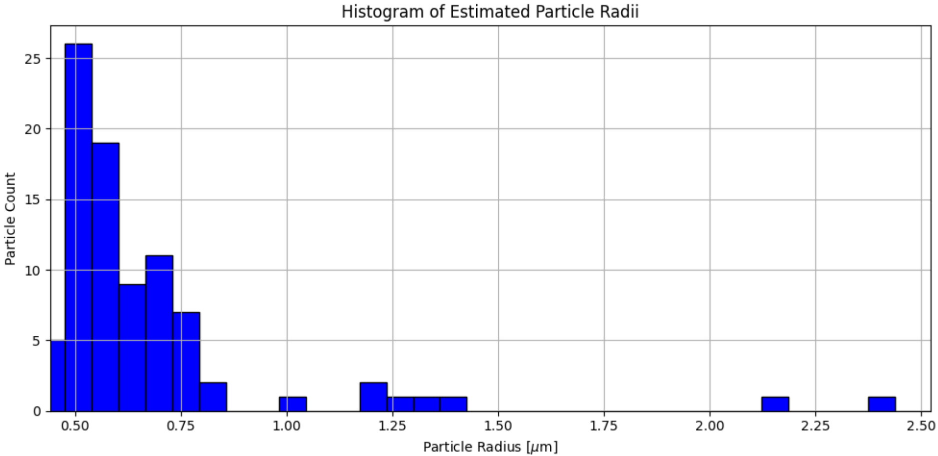


Figure: Radius Histogram 1.04μm diameter Polystyrene fit with $\tau_{max} = 5$

After Changes

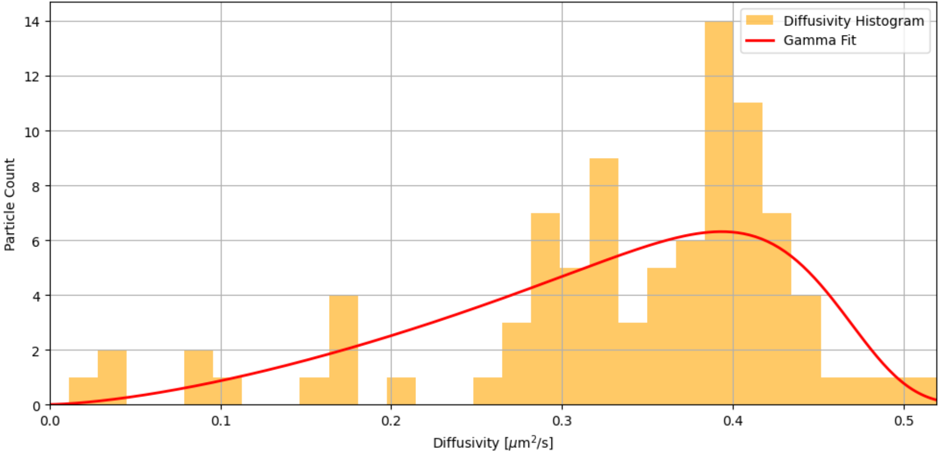


Figure: Diffusivity Histogram 1.04μm Polystyrene fit with $\tau_{max} = 5$

After Changes

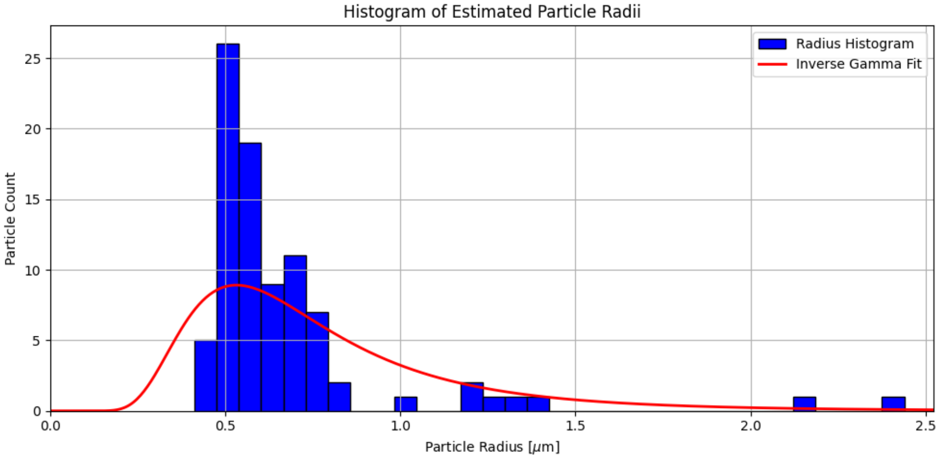


Figure: Radius Histogram 1.04μm diameter Polystyrene fit with $\tau_{max} = 5$

True Distribution of MSD-inferred Diffusivity & Radii

After some calculation, it can be shown that the MSD inferred diffusivity and radius are distributed according to:

$$\boxed{\pi_D(x; N, D, \tau, d) = \frac{x^{\frac{d(N-\tau)}{2}-1} e^{\frac{-d(N-\tau)x}{2D}}}{\left(\frac{2D}{d(N-\tau)}\right)^{\frac{d(N-\tau)}{2}} \Gamma\left(\frac{d(N-\tau)}{2}\right)}} \sim \Gamma\left(\frac{d(N-\tau)}{2}; \frac{2D}{d(N-\tau)}\right)$$

$$\boxed{\pi_r(x; N, D, d, \tau, T, \eta) = \frac{e^{\left(\frac{-k_B T d(N-\tau)}{12\pi D\eta x}\right)} \left(\frac{k_B T d(N-\tau)}{12\pi D\eta}\right)^{\frac{d(N-\tau)}{2}}}{\Gamma(\frac{d(N-\tau)}{2}) \cdot x^{\frac{d(N-\tau)}{2}+1}}} \sim \mathrm{Inv}\Gamma\left(\frac{d(N-\tau)}{2}; \frac{k_B T d(N-\tau)}{12\pi D\eta}\right)$$

Note: This holds for single points, distributions approximate Gaussians for larger τ_{max} by CLT.

Integration into Visual Interface

- Integrates directly into a visual interface.
- Tracking is performed on a parallel process during operation.
- The tracking pipeline supports individual particle analysis via tagging.
- A jupyter notebook has been made for straightforward tracking calibration to accelerate cross-setup changes.

- The following files are returned:
  1. Diffusivity distribution histogram
  2. Radii distribution histogram
  3. Ensemble & Individual mean squared displacement plots
  4. Info text file with values.
  5. Per particle diffusivity/radius csv
  6. Trajectory plots in pixel space
  7. Trajectory csv with enumerated particle trajectories.

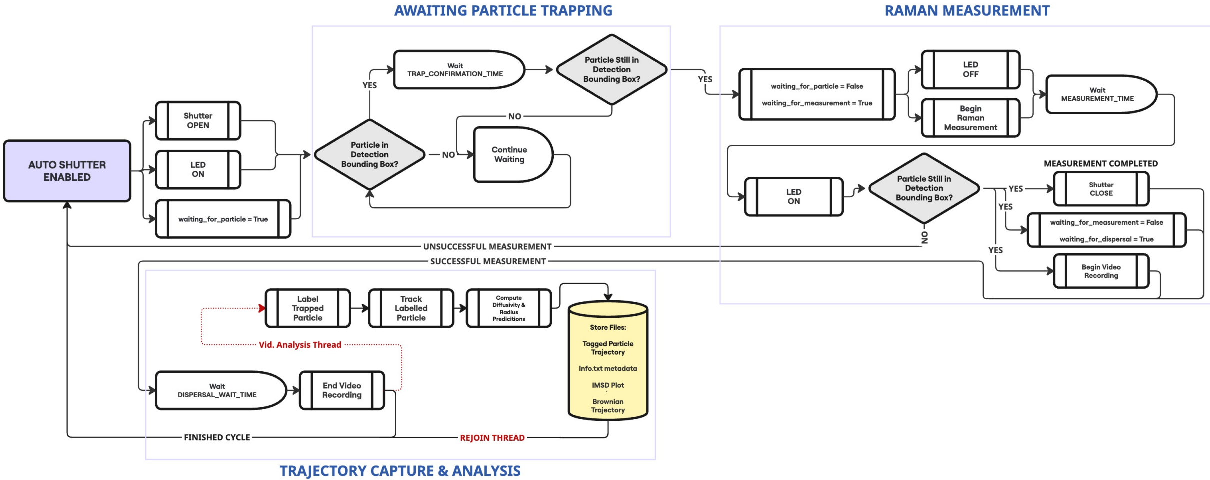
Visual Automation Interface

About the Visual Interface

- Primary Backends:
  - OpenCV
  - pypylon
  - pythonnet
  - Thorlab's Kinesis SDK
  - Arduino's PyFirmata
  - various other support packages..

- Functionality:
  - Overlayed particle detection
  - Manual detection bounding box adjustment
  - Manual shutter control
  - Screengrabbing & Video
  - Alignment crosshair
  - Radii Estimation (processed concurrently)
  - **Auto Mode** (complete autonomous control)

Auto Mode Control Flow



Final Remarks

- All features have been tested simultaneously on a webcam based simulated setup and are working in parallel.
- Programmatic control of isoplane is implemented, but not tested.
- Full testing will be done on the setup once hardware integration is complete.
- The entire visual interface backend is extremely object-based and packaged.
- An extensive documentation detailing custom visual interface creation is on Github with examples.