Jonathan Cho 04/08/2022

Accurate Imaging of Particle Motion in Dusty Plasmas

By

Jonathan Cho

Justin Burton
Advisor

Physics

Justin Burton
Advisor

Daniel Weissman
Committee Member

Tom Bing
Committee Member

2022

Accurate Imaging of Particle Motion in Dusty Plasmas

By

Jonathan Cho

Justin Burton

Advisor

An abstract of a thesis submitted to the Faculty of Emory College of Arts and Sciences of Emory University in partial fulfillment of the requirements of the degree of Bachelor of Sciences with Honors.

Physics

2022

Abstract

Accurate Imaging of Particle Motion in Dusty Plasmas

By Jonathan Cho

The use of cameras to track the trajectory of objects is important in the science of imaging. Therefore, accuracy in locating particle motion is of high priority. We explore the motions of dusty plasma in a weakly-ionized Argon-filled vacuum chamber in real experiments using a Phantom V7.11 high-speed camera. Then we simulate dusty plasma motion and track its trajectory using Trackpy, a Python tracking package. An issue called pixel locking arises when tracking few-pixel wide objects, which make particle tracking inaccurate. Pixel locking is apparent in dusty plasma experiments where the sizes of particles are $11-\mu m$. The analysis of the characteristics of dusty plasmas in a controlled environment is heavily biased due to pixel locking. There is a commonly used method called the SPIFF method which mathematically finds the true location of particles given an estimated location fed by a tracking algorithm. However, we find that the SPIFF algorithm does not correctly solve the fundamental issues caused by pixel locking as huge amounts of error are found when comparing the velocity probability density functions of a simulated particle trajectory and a SPIFF corrected particle trajectory.

Accurate Imaging of Particle Motion in Dusty Plasmas

By

Jonathan Cho

Justin Burton

Advisor

A thesis submitted to the Faculty of Emory College of Arts and Sciences of Emory University in partial fulfillment of the requirements of the degree of Bachelor of Sciences with Honors.

Physics

2022

Acknowledgements

My biggest thanks to Justin Burton, my advisor, who invested time and energy to help me investigate imaging particle motion in dusty plasmas. In addition to my other committee members, Daniel Weissman and Tom Bing, Wentao Yu were very supportive throughout this project, for which I am deeply grateful.

Table of Contents

Contents

1 Introduction

The usage of imaging has become progressively more popular and important in many fields of science. Imaging is prevalent in biology such as sizing and identifying cellular vesicles [1], in motion computation [2], in stereo matching [3], in structural analysis near the colloidal glass transition [4], and more [5, 6, 7]. Thus, reducing the amount of noise and minimizing error that occur in imaging is of high priority. However, a universal error called pixel-locking is introduced when tracking very small objects that are only a few pixels wide. Pixel-locking is the inclination of a particle image velocimetry velocity field to measure more integer values of pixel displacement [8]. This error occurs due to the lack of resolution to determine precise sub-pixel locations [8]. Pixel-locking can cause drastic errors in understanding the characteristics of flow fields when using small particles as observable objects as the tracking is heavily biased. One specific application of using pixel tracking software in experiments is tracking particles in dusty plasma experiments [9]. The presence of pixel locking inhibits investigators to accurately determine physical aspects of a dusty plasma system. We explore when pixel locking occurs in regards to particle and mask size and see if the widely accepted SPIFF method is a valid solution to fix pixel locking [10]. We simulate particle motion with Gaussian noise so that we know the exact velocities at all times. Then, using Trackpy, we track the particle motion and create a probability density function of the tracked velocities. We will create a probability density function for the simulated particle velocities and compare it to the tracked velocities and see if there are any significant differences. If tracking is done nearly perfectly, then the probability density functions of both velocities will be similar. If both probability density functions are significantly different, we can conclude that pixel locking does have significant effects on measuring fundamental data. We also want to explore if deploying machine learning strategies in place of fixing algorithms, such as SPIFF, can successfully correct measurement errors caused by pixel locking.

1.1 Dusty Plasma

Dusty plasma is observed in extraterrestrial bodies such as the rings of Saturn or in interstellar clouds [11]. Dusty plasma is also seen in Earth's ionosphere where dust particles in the atmosphere are charged mainly by solar wind plasma [12]. Dusty plasma is the phenomena in which few nanometers- to few centimeters-sized dust particles are suspended in an ionized gas environment [13, 14, 15, 16, 17]. The plasma environment is usually generated by applying a current through an electrode in a vacuum chamber. Dust particles are then suspended into the system and are charged by ions and even more electrons from the plasma, making them negatively charged [9]. This allows dusty plasma to be suspended over the electric plate due to the electric field generated. Dusty plasma obey the same plasma physics as charged particles such as repulsion, drag from ions, and stochastic effects. This allows the fluid mechanics of the plasma system to be identified through analyzing the motions of dusty plasma [9]. Core advantages of dusty plasma compared to ions and electrons are their relatively large size and slow speed. This allows us to identify some of the qualities of the plasma fluid system by observing motions of dusty plasma using a simple camera [9].

1.2 SPIFF Algorithm

A commonly used method to solve pixel-locking is the single-pixel interior filling function (SPIFF) algorithm [18]. The SPIFF algorithm looks at the histogram of the decimal distribution of a tracked sample. It then forcefully fits the histogram to a uniform distribution to find the true position of the particle [19]. The true position of the particle can be calculated by solving,

$$X_T = \pm \int_0^{X_E} P(X_E')\, dX_E$$

where $\mathrm{X_E}$ is the estimated position of the particle, $\mathrm{X_T}$ is the true position of the particle, and $\mathrm{P(X_E)}$ is the SPIFF density function [10]. Although the SPIFF method estimates the true value of the position of the particle mathematically, SPIFF still produces significant error in finding the position of the particle. This error is seen when we produce a 1-dimensional velocity distribution. We then compare it to a purely Gaussian velocity distribution, which is an indication of white stochastic noise.

1.3 Tracking Particles

Cameras capture and process images by detecting photons of the desired image and then are distributed based on the point spread-function of the camera used [20, 21]. A point spread-function explains the image produced from a point source from a specific camera. Even though the exact shape and size of the particle is limited by the resolution of the camera used, the center of the particle can be determined based on the number of photons hitting the pixels [22]. Thompson et al. used a localization algorithm based on a least-square fitting theory to determine the positions of fluorescent probes [23]. Similarly, the Crocker-Grier algorithm tracks objects through a least-square fitting method [24]. We are going to use Trackpy in our experiments, which is an open-source tracking algorithm based on the Crocker-Grier algorithm that analyzes bright spots of a video and maps their trajectory in Python [25]. Trackpy captures objects that are vastly different from the background color and pinpoints their positions by fitting their brightness to a Gaussian distribution. Trackpy is able to monitor how much of an area around the brightest spots it analyzes, control how many light-emitting objects we expect to observe at each frame, and determine each particle's trajectory based on a given memory [25]. We will explore the presence of pixel-locking in dusty plasma experiments and see if the bias can be fixed using the SPIFF method.

2 Methods

2.1 Plasma Experiment

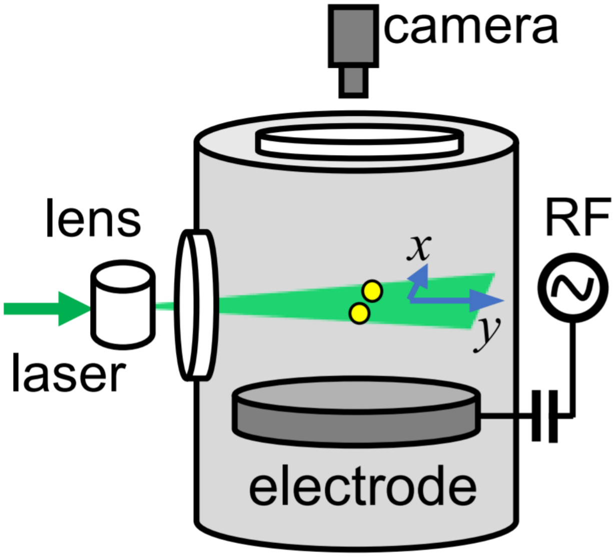

Figure 1: Experimental apparatus for dusty plasma experiment. Consists of vacuum chamber kept at 0.6-1.0 Pa and injected with argon gas [26].

We have observed the motions of dusty plasma in our custom built vacuum chamber to determine characteristics of a plasma field on charged particles. The plasma environment is generated inside of a vacuum chamber with pressure of 1.0 Pa that has an electric plate installed within the chamber to excite gas molecules. We pump argon gas into the vacuum chamber and ionize the gas by applying a 13.56 MHz radio-frequency voltage through the electrode [26]. An aluminum ring is set around the electrode to prevent ionized dust particles from escaping horizontally. A spherical melamineformaldehyde dust particle (microParticles GmbH) with diameter=11 μm is suspended into the vacuum chamber and then ionized by the electrode. The ionized dust particle "floats" in the vacuum chamber due to the electric force countering the gravitational force. The voltage of the electrical plate is controlled by a capacitor and is optimized to $2.9 \pm 0.1W$ of input power running through the system. The ionized particles are observed by shining an oscillating green laser into the vacuum chamber at the same level in which the ionized particles are suspended in the system. The laser oscillated with a 50 Hz sawtooth wave to observe vertical motion. The ionized particles are not always at a constant charge as they are susceptible to noise, differences in the plasma system, or many other external factors [14]. However, we will ignore the vertical motions of the experiments and analyze the purely horizontal planar movement of the particle. The ionized particles are visible due to the green laser and are recorded using a Phantom V7.11 high speed camera. We can introduce more ionized particles into the system by slightly disturbing the particle reservoir. We can remove ionized particles from the system by quickly pulsing the electrical plate to "drop" some particles, preventing them from being suspended again. We observe in experiments that the particle moves in an elliptical pattern before finally resting after a long time. This can occur due to the effect of the magnetic vortex on influencing the particle trajectory [27, 28, 29]. We will want to simulate the effect of the vortex on our simulated particle motion to most accurately follow our dusty plasma procedures.

2.2 Simple Harmonic Motion

To simulate experimental particle motion, we first have to understand the core physics influencing the dusty plasma. We can assume particles are suspended in mid-air due to the influence of the electric field counteracting gravity by

$$F = E * q = m * g \tag{1}$$

where:

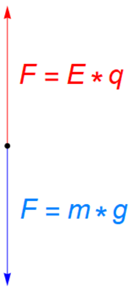

Figure 2: Free body diagram of a single particle affected by the electric field and gravity.

- E is the electric field strength, V/m
- q is charge on each particle, C.
- m is the mass of each particle, g.
- g = 9.8 m/s^2.

This simplifies our analysis by confining each particle movement to a 2-D plane with horizontal confinement.

2.3 Particle Trajectory Model

The motion of ionized particles within the system is dependent on the characteristics of the vortex. One can imagine the effects of a vortex on dusty plasma to be similar to the effects of a whirlpool on floating balls. Just as the floating balls rotate around the center of a whirlpool, a vortex affects dusty plasma by spinning the dusty plasma [30]. Let $\ddot{x}(t)$ and $\dot{x}(t)$ be the time derivatives of $x(t)$. Similarly, let $\ddot{y}(t)$ and $\dot{y}(t)$ be the time derivatives of $y(t)$. The relationship between the acceleration and the position solely due to the effects of the external electrostatic confinement can be shown as,

$$\begin{bmatrix} x''(t) \\ y''(t) \end{bmatrix} = \begin{bmatrix} w_+ & 0 \\ 0 & w_- \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \tag{2}$$

where $w_- = (1-\delta)w_0^2$ and $w_+ = (1+\delta)w_0^2$. w_0 stands for the average confinement of the plasma and δ stands for the asymmetry. Now we can introduce the damping factor, γ, dusty plasma will experience. The particle will experience higher damping effects which scales linearly with their velocities. The faster the particle goes, the more resistance it will feel from the surrounding ions. We can represent the damping factor as,

$$damping = -\gamma \begin{bmatrix} x'(t) \\ y'(t) \end{bmatrix} \tag{3}$$

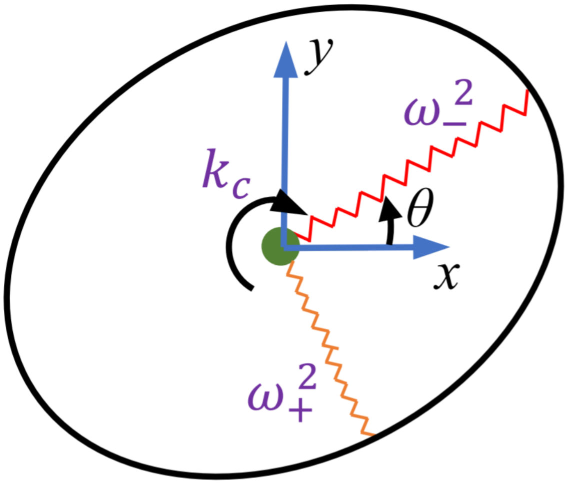

Figure 3: Visualization of the asymmetry of the confinement as visualized by the ellipse [26].

Now we have to see how the particle experiences an influence from how strong the magnetic vortex is, which is represented by $k_c > 0$. This can be visualized as how strong a whirlpool is. The stronger the magnetic vortex, the more the particle inside is affected.

$$k_c \begin{bmatrix} -x(t) \\ y(t) \end{bmatrix} \tag{4}$$

We can now show an expression of how the vortex affects the positions of dusty plasma by combining equations (2), (3), and (4),

$$\begin{bmatrix} x''(t) \\ y''(t) \end{bmatrix} = \begin{bmatrix} (1+\delta)w_0^2 & 0 \\ 0 & (1-\delta)w_0^2 \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} - \gamma \begin{bmatrix} x'(t) \\ y'(t) \end{bmatrix} + k_\mathrm{c} \begin{bmatrix} -x(t) \\ y(t) \end{bmatrix} \tag{5}$$

However, the particle motion does not fully follow the trajectory of equation (5) due to noise in the system. We assume that the random noise in the system follows a stochastic model in which randomness in particle motion is predicted based on a Gaussian probability [31]. The majority of the particle trajectory will follow closely to the path explained by equation (5). Considering stochastic noise, N(α), the resulting equation is,

$$\begin{bmatrix} x''(t) \\ y''(t) \end{bmatrix} = \begin{bmatrix} (1+\delta)w_0^2 & 0 \\ 0 & (1-\delta)w_0^2 \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} - \gamma \begin{bmatrix} x'(t) \\ y'(t) \end{bmatrix} + k_{\text{c}} \begin{bmatrix} -x(t) \\ y(t) \end{bmatrix} + N(\alpha) \tag{6}$$

$\alpha = 2$ expresses a purely Brownian randomness and $\alpha < 2$ shows a more spread out Gaussian distribution with long tails. Since we simulated particle motion with inherent stochastic noise, we expect $\alpha = 2$ for the velocity distribution. Since pixel locking affects key measurements in tracked particles,

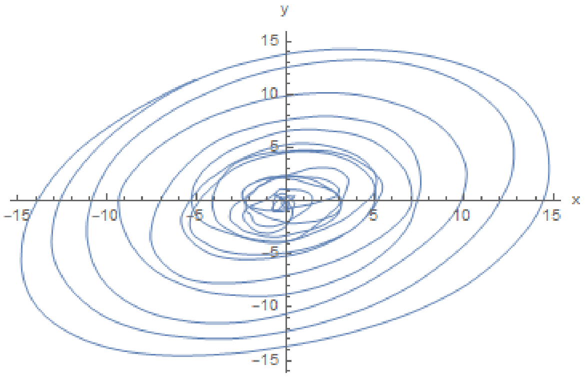

Figure 4: Used Mathematica to simulate single particle motion following equation (6) considering Gaussian noise. Simulated with $w_0 = 1, \gamma = 0.9, \delta = 0.3, k_c = 0.9604, \alpha = 2$.

we expect $\alpha \neq 2$ for the velocity distribution of tracked videos.

2.4 Particle Generation

We also want to imitate real experimental particles by simulating how real recorded particles are seen in tracked recordings. A camera captures an image based on the point spread-function of the camera. Experiments with micro-objects have shown that the light emitted from very small objects follow a least-squares Gaussian approximation [32]. Similarly, dusty plasma in experiments show a Gaussian distribution of light where the center of the particle shows a much brighter spot and deviates away from the center [33]. We can use this characteristic of how light is captured from a point source to generate our simulated particles. A spherical dusty plasma will show as a pixelated Gaussian distribution on the screen. The standard form that generates the character of the Gaussian distribution is represented as,

$$f(x) = e^{-\frac{1}{2}(\frac{x-x_0}{\sigma})^2} \tag{7}$$

where σ is the standard deviation and x_0 is the deviation from the mean. σ describes the width of the particle and can be altered to vary particle size. This equation describes the spread of a normal distribution in a 1-D scenario. Because we are working with particle motion on a 2-D plane, we

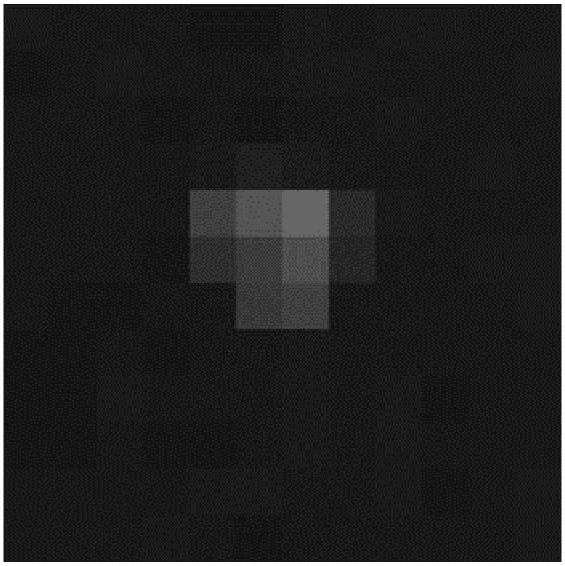

Figure 5: Real experimental picture of a dusty plasma suspended in vacuum chamber.

must introduce a secondary variable y to simulate 2-D particles:

$$f(x) = e^{-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}} \tag{8}$$

Using equation (8), we are able to simulate light-emitting symmetrical particles that are only a few pixels in width.

Since we are able to simulate both the particle and particle motion in a vortex, we run a simulation by generating a particle in regards to equation (8) and simulate its movement that follows the trajectory described by equation (6). We will then use Trackpy to locate the positions of the simulated particle over its lifetime and create a histogram of the decimal values of the coordinates to determine if pixel-locking is present or not.

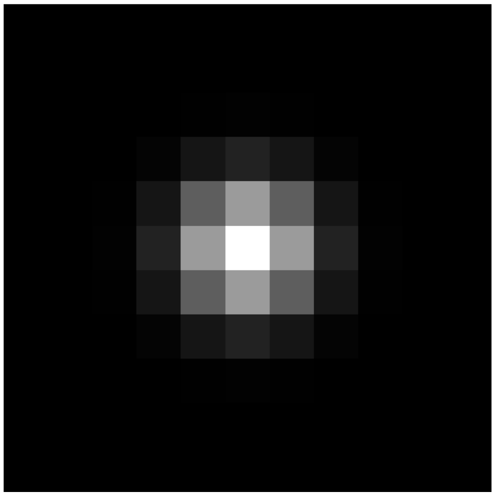

Figure 6: Simulated particle with $x_0 = y_0 = 0, \sigma = 1$

2.5 Error Analysis

To analyze the error produced by pixel lcoking, we need to determine when there is a significant bias towards the integer values. We extract the decimal values of all points of the simulated particle trajectory and then produce a histogram. To determine if bias is present, we conduct a probability test to see if the bias present is statistically significant. A noticeable trend of a biased sample is that there are much less decimal values that are near 0.4-0.6. To mark if there is a significant bias, we took the sum of the number of values between 0.4-0.6 and compared it to the number of values 0.0-0.2 and 0.8-1.0. If the number of values between 0.0-0.2 and between 0.8-1.0 is much greater than twice the number of values between 0.4-0.6, then we deem the sample as biased. Through trial and error, we found that a scale of 0.7

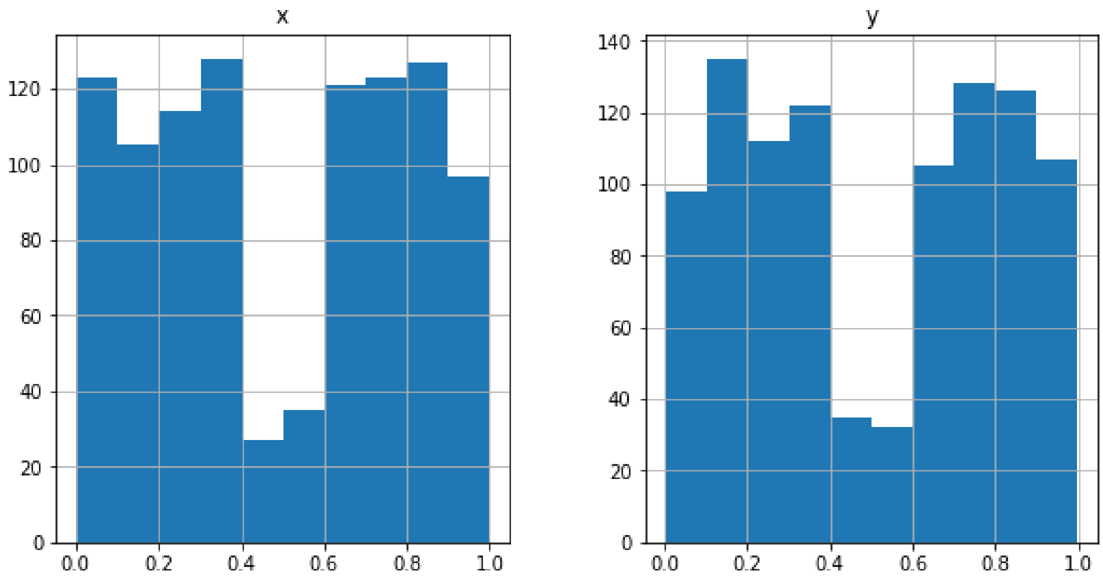

Figure 7: Pixel-locking present histogram of decimal point values.

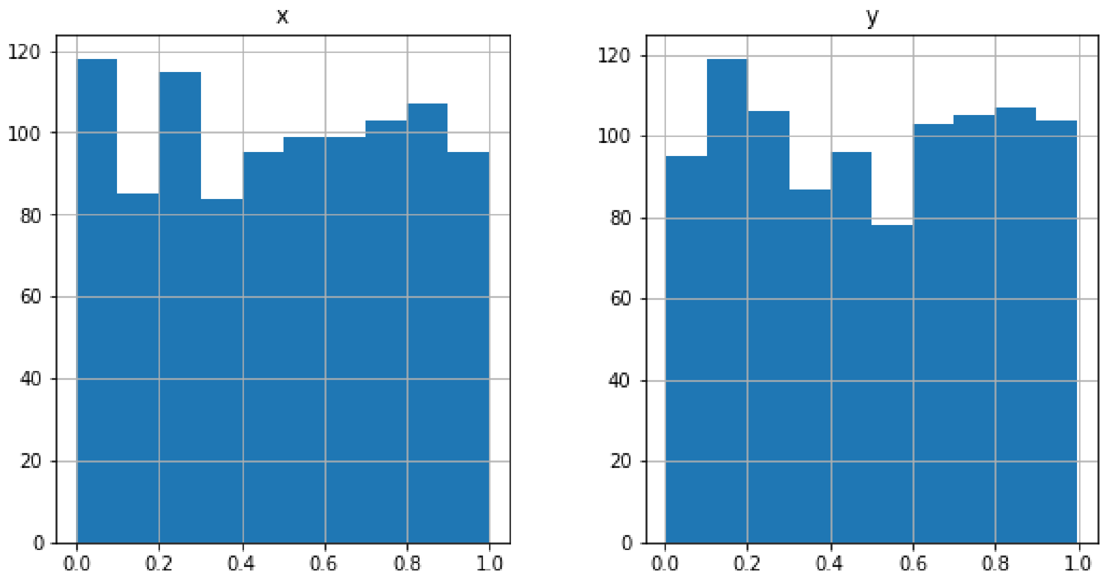

Figure 8: No pixel-locking histogram of decimal point values.

reliably detects the presence of pixel-locking.

$$2N[0.4 - 0.6] < 0.7(N[0.0 - 0.2] + N[0.8 - 1.0]) \tag{9}$$

If equation (9) turns out to be true, then the sample is biased as there is a bias towards the integer values in the histogram. The relevant variables that affects whether or not pixel-locking occurred are the mask size and the value of σ. σ affects the size of the particle and the mask size controls the size of the area around the brightest spot of an object. The value of the mask size, n, will analyze an nxn area centered around the brightest pixel on the screen and determine the position of the object with sub-pixel accuracy by fitting the observed pixels to a Gaussian distribution. The mask size needs to be an odd number greater than 1 since the even value mask sizes do not have a pixel center. We vary mask sizes and sigma values to see in which set of conditions do pixel-locking occur to understand the nature of pixel-locking in tracking programs.

2.6 Velocity Distribution

Brownian motion observed in particles are mainly due to random fluctuations within the system that interact with the particles. To analyze how effective raw tracking and SPIFF method are on determining velocities, we create a probability density function of the velocity distribution. We first normalize

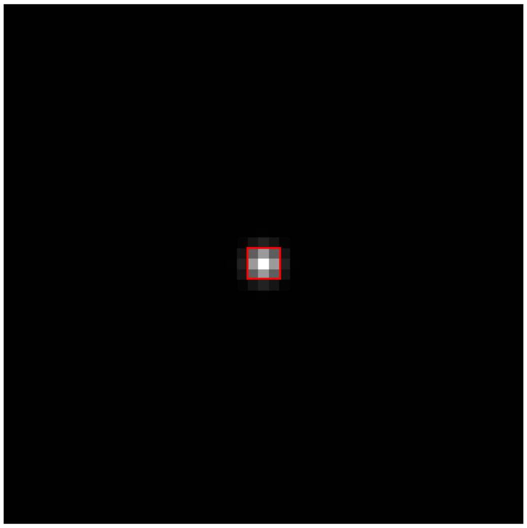

Figure 9: The area of light values analyzed with mask size=3 around the brightest pixel on screen.

all the velocities by the overall standard deviation.

$$v_{norm} = \frac{v}{\sigma_v} \tag{10}$$

Then we create a probability density function of the normalized velocities with log scaling on the y-axis. We fit the resulting probability density function to a best fit curve with an exponential term,

$$y = -A * x^{\alpha} + B \tag{11}$$

where A and B are arbitrary constants. We used $scipy.optimize.curve_fit()$ on Python to find the best fit curve for the probability density function. We expect the probability density function for a perfect stochastic motion to have $\alpha = 2$. Any other result for α means that particle trajectory did not experience purely stochastic noise.

3 Results and Discussion

Real experimental particles were recorded to be about 5 pixels in width, which correspond to the value of σ lying between 0.9-1.1. As seen in figure 10, we observe that pixel locking occurs when the mask size, m, is $m \leq 5$ and pixel locking is not present when the mask size is $m \geq 7$. We tested the validity of equation 9's ability to successfully detect pixel locking or not by looking at the decimal distribution of the tracked particles and graphed a list plot of different mask sizes vs. values of σ. We found that, no matter the

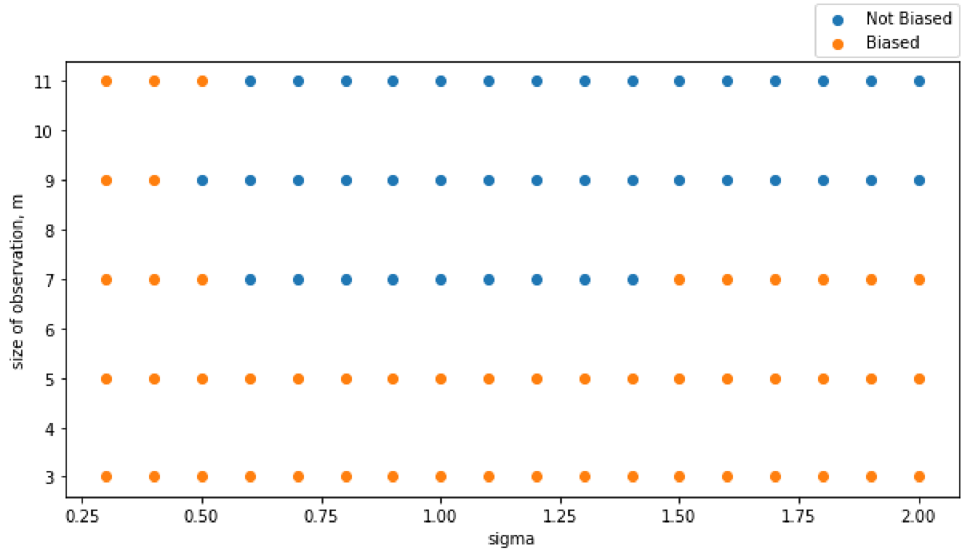

Figure 10: Mask size vs. σ plot expressing presence of pixel-locking. Decision to see if pixel locking has occurred or not is based on equation 9.

mask size, pixel locking occurs if the size of the particle is too small, which is what we expected. This occurs because Trackpy is unable to accurately fit the light distribution to a Gaussian distribution with high accuracy due to

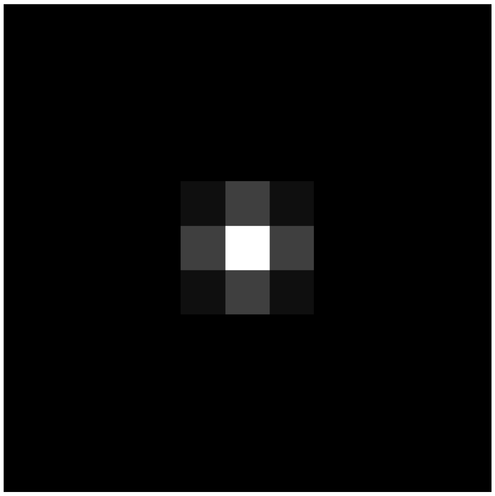

Figure 11: Visualization of particle with $x_0 = y_0 = 0, \sigma = 0.5$.

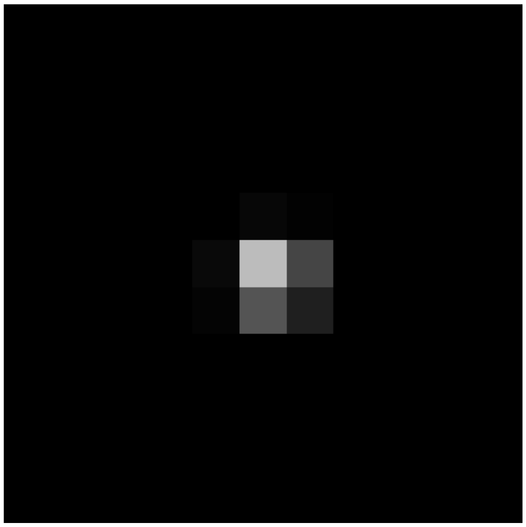

Figure 12: Visualization of particle with $x_0 = 0.3, y_0 = 0.25, \sigma = 0.5$.

the lack of information given by the dim particle. Trackpy has no problem tracking particles if they are tracking a perfectly symmetrical picture of a particle. However, pixel locking occurs once particles move away from the center of pixels due to the uneven distribution of light evident in Figure 12.

We also see pixel locking occur when the mask size is too low in comparison to the pixel size. Figure 12 shows an example in which tracking a large object still results in pixel locking due to the mask size being too small. The pixels surrounding the center of the particle are way too similar in light intensity, so Trackpy is unable to fit the particle light levels to a Gaussian distribution accurately. We see in Figure 10 that a tracked particle with $\sigma = 1.5$ and mask size $m = 7$ has pixel locking due to the mask size being too small in comparison to the pixel size. We see that pixel locking does not occur when we increase the mask size to $m = 9$ since Trackpy is

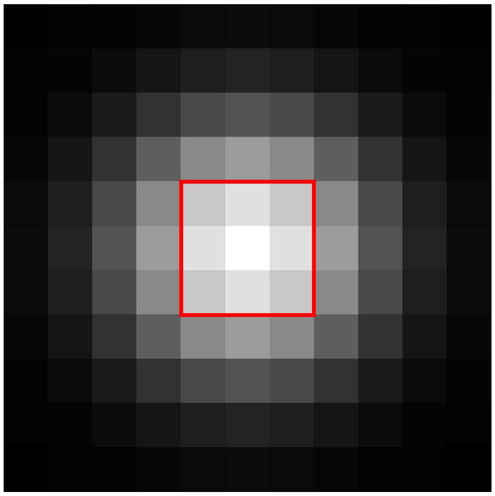

Figure 13: Visualization of particle with $\sigma = 2.0$ and mask size =3.

able to extract sufficient enough data to fit the light levels to a Gaussian distribution.

Figure 13 shows the velocity probability density function for simulated data. A velocity profile with $p = 2$ means that the only noise affecting the system is stochastic noise. Figure 13 shows that when the simulated particle with Gaussian noise is tracked, $p = 2$ which is expected. We also observed that a simulation with non-Gaussian noise has $p = 1.49$, which is also expected since we only expect a system with pure Gaussian noise to have $p = 2$. However, when we used the SPIFF method on a tracked particle with pixel locking, we got $p = 1.36$. This means that SPIFF correction does not completely fix the issue of pixel locking as the fitted exponent $p = 1.36$ does not equal the fitted exponent of the true particle trajectory velocity $p = 2.00$.

4 Conclusion

In this investigation, we observed how pixel locking drastically affects the velocity distribution from tracking dusty plasma experiments and see if implementing conventional methods to correct for pixel locking significantly improves data collected. We specifically looked into the nature of when pixel locking occurs through the use of simulated particle motion and see if the SPIFF algorithm is a viable method of fixing for the errors caused by pixel locking. We found that the general issue of sub-pixel bias in tracking minute objects still exists even after using the SPIFF method. The use of simula-

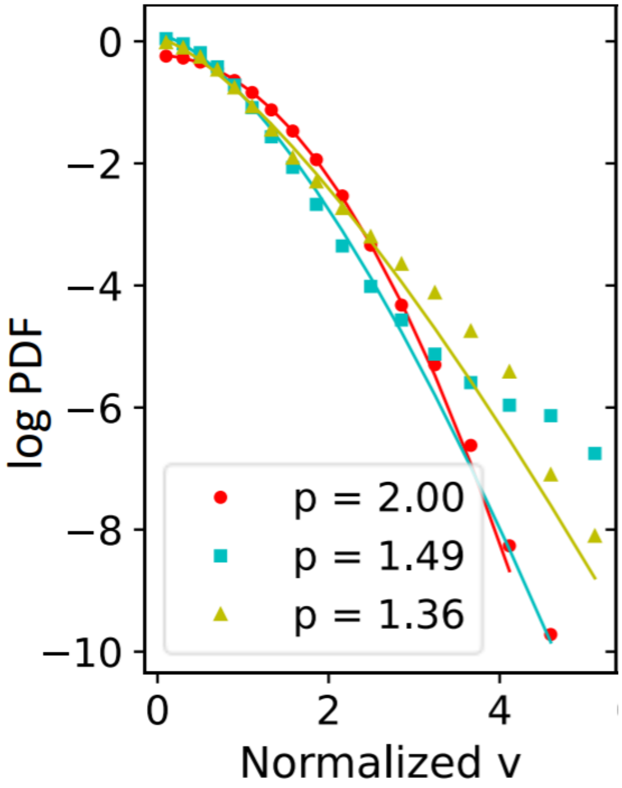

Figure 14: Log probability density function of x-velocity of particle, —$\mathrm{v_x}$—, normalized by $\sqrt{\langle v_x^2 \rangle}$. Red dots represent Gaussian simulated velocity PDF, $\alpha = 2$. Blue squares represent non-Gaussian simulated velocity PDF, $\alpha = 1.8$. Green triangles represent SPIFF correction on data with pixel locking, $\alpha = 2$. Solid lines represent their respective colors fits in regards to the form $y = Av^p$ where p is the fitted exponent profile.

tions allows us to compare how accurate tracking algorithms and the SPIFF method are at tracking the true location of a single pixel trajectory to the actual known location.

We want to explore how mask size affects the presence of pixel locking since it would be ideal to have the smallest mask size possible. A smaller mask size will significantly improve the speed of tracking. Each tracking analysis with $m = 13$ took $> 3min$ while analysis with $m = 11$ took $< 1min$. A smaller mask size also allows us to analyze multiple particles. Experiments that require multiple particles to be tracked prefers smaller mask sizes as numerous particles may be present under one mask, which makes tracking inaccurate. Trackpy will treat overlapping particles as one object as they are all under one mask. A smaller mask size will reduce the chance for multiple particles to occur in each analysis. However, the trade off for a smaller mask size as seen in Figure 10 is the higher chance of pixel locking to occur.

We observe that pixel-locking exists when tracking real dusty plasma motions in an ionized vacuum environment. Even though the SPIFF algorithm mathematically estimated the true value of the position of the particles, the velocity distribution of the tracked particle is significantly different than the velocity distribution of the actual trajectory. This error means that an alternative method to fix sub-pixel inaccuracy must be introduced to successfully resolve pixel locking. The observed particles were about 5 pixels in width, which correspond to the value of σ lying between 0.9-1.1. As seen in figure 8, we observe that pixel locking occurs when the mask size, m, is $m \leq 5$ and

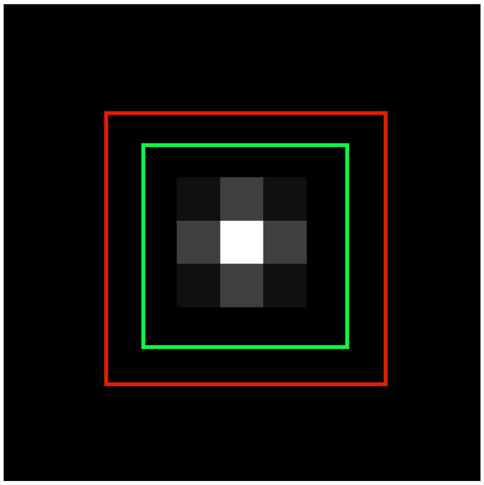

Figure 15: Image of mask size m=5 represented by green square and mask size m=7 represented by red square.

pixel locking is not present when the mask size is $m \geq 7$.

We saw that despite a mask size of $m = 5$ able to encompass the majority of the light emitted by a particle of $\sigma = 0.5$, pixel locking still occurs when the particle is tracked. However, a mask size of $m = 7$ is able to successfully detect the particle position without pixel locking. The only difference is that mask size $m = 7$ encompass more ”blank space” than $m = 5$, which we initially thought would make negligible difference. Further investigation is needed for determining the sensitivity of mask sizes and what it calculates. The present work only deals with simulated particle trajectory. Extension into using simulated data to fix real experimental data will require research in tracking projection of simulated particle movement.

References

[1] Dragovic, r. a. et al. sizing and phenotyping of cellular vesicles using nanoparticle tracking analysis. nanomedicine nanotechnology, biol. med. 7, 780–788 (2011)., 2011.

[2] Aggarwal, j.k. and nandhakumar, n. 1988. on the computation of motion from sequences of images—a review. proceedings of the ieee, 76(8):917–935.

[3] Kanade, t. and okutomi, m. 1994. a stereo matching algorithm with an adaptive window: Theory and experiment, ieee transactions on pattern analysis and machine intelligence, 16(9):920–932.

[4] Weeks, e. r. tree-dimensional direct imaging of structural relaxation near the colloidal glass transition. science 287, 627–631 (2000).

[5] Aghajan, h.k., schaper, c.d., and kailath, t. 1993. machine vision techniques for sub-pixel estimation of critical dimensions. optical engineering, 32(4):828–839.

[6] Wang, j., chen-wiegart, y. k. wang, j. in operando tracking phase transformation evolution of lithium iron phosphate with hard x-ray microscopy. nat. commun. 5, 1–10 (2014).

[7] Jiang, j., zhang, g., wei, x. li, x. rapid star tracking algorithm for star sensor. ieee aerosp. electron. syst. mag. 24, 23–33 (2009).

[8] Hearst, r. jason, and bharathram ganapathisubramani. ”quantification and adjustment of pixel-locking in particle image velocimetry.” experiments in fluids 56.10 (2015): 1-5.

[9] Feng, yan, et al. ”particle position and velocity measurement in dusty plasmas using particle tracking velocimetry.” journal of plasma physics 82.3 (2016).

[10] Burov, stanislav, et al. ”single-pixel interior filling function approach for detecting and correcting errors in particle tracking.” proceedings of the national academy of sciences 114.2 (2017): 221-226.

[11] F. verheest, waves in dusty space plasmas (kluver academic publishers, dordrecht, 2000).

[12] Mann, ingrid, et al. ”dusty plasma effects in near earth space and interplanetary medium.” space science reviews 161.1 (2011): 1-47.

[13] Shukla, p. k. mamun, a. a. 2001 introduction to dusty plasma physics, institute of physics.

[14] Melzer, a. goree, j. 2008 fundamentals of dusty plasmas. in low temperature plasmas: Fundamentals, technologies and techniques, 2nd edn (ed. hippler, r., kersten, h., schmidt, m. schoenbach, k. h.), pp. 157–206. wiley-vch.

[15] Morfill, g. e. ivlev, a. v. 2009 complex plasmas: an interdisciplinary field. rev. mod. phys. 81, 1353–1404.

[16] Bonitz, m., henning, c. block, d. 2010 complex plasmas: a laboratory for strong correlations. rep. prog. phys. 73, 066501.

[17] Piel, a. 2010 plasma physics. springer.

[18] Yifat, yuval, et al. "analysis and correction of errors in nanoscale particle tracking using the single-pixel interior filling function (spiff) algorithm." scientific reports 7.1 (2017): 1-10.

[19] Yifat, yuval, nishant sule, and norbert f. scherer. "identifying and correcting pixel locking errors with the spiff algorithm." single molecule spectroscopy and superresolution imaging xi. vol. 10500. international society for optics and photonics, 2018.

[20] Novotny l, hecht b (2006)principles of nano-optics(cambridge univ press, cam-bridge, uk).

[21] Born m, wolf e (1999)principles of optics: Electromagnetic theory of propagation,interference and diffraction of light(cambridge univ press, cambridge, uk).

[22] Bobroff, n. 1986. position measurement with a resolution and noiselimited instrument. rev. sci. instrum. 57:1152–1157.

[23] Thompson, russell e., daniel r. larson, and watt w. webb. "precise nanometer localization analysis for individual fluorescent probes." biophysical journal 82.5 (2002): 2775-2783.

[24] Crocker, john c., and david g. grier. ”methods of digital video microscopy for colloidal studies.” journal of colloid and interface science 179.1 (1996): 298-310.

[25] Allan, daniel b., caswell, thomas, keim, nathan c., van der wel, casper m., verweij, ruben w. (2021). soft-matter/trackpy: Trackpy v0.5.0 (v0.5.0). zenodo. https://doi.org/10.5281/zenodo.4682814.

[26] Yu, wentao, jonathan cho, and justin c. burton. ”extracting forces from noisy dynamics in dusty plasmas.” arxiv preprint arxiv:2203.03740 (2022).

[27] M. schwabe, u. konopka, p. bandyopadhyay, and g. e. morfill, phys. rev. lett. 106, 215004 (2011).

[28] M. choudhary, r. bergert, s. mitic, and m. h. thomas, phys. plasmas 27, 063701 (2020).

[29] E. thomas jr., u. konopka, r. l. merlino, and m. rosenberg, phys. plasmas 23, 055701 (2016).

[30] Guslienko, k. yu, et al. ”field evolution of magnetic vortex state in ferromagnetic disks.” applied physics letters 78.24 (2001): 3848-3850.

[31] Pinsky, mark, and samuel karlin. an introduction to stochastic modeling. academic press, 2010.

[32] Zhang, bo, josiane zerubia, and jean-christophe olivo-marin. "gaussian approximations of fluorescence microscope point-spread function models." applied optics 46.10 (2007): 1819-1829.

[33] Williams, jeremiah d. "application of particle image velocimetry to dusty plasma systems." journal of plasma physics 82.3 (2016).