PARTICLE TRACKING APPLICATIONS OF VIBRATING GRAINS USING THERMAL IMAGING FILTRATION

by

Philip Brown

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Mechanical Engineering

Charlotte

2024

Approved by:

Dr. Russell Keanini

Dr. Jerry Dahlberg

Dr. Artur Wolek

ABSTRACT

PHILIP BROWN. Particle Tracking Applications of Vibrating Grains using Thermal Imaging Filtration. (Under the direction of DR. RUSSELL KEANINI)

This thesis describes the development of a new macroscopic method to study the diffusion of particles using thermal imaging as a means of filtration. Using research previously conducted at The University of North Carolina at Charlotte as a basis for the experiment, ceramic grains in a vibratory polishing machine were used as a macroscopic analog for fluid molecules. A method of heating grains and using a thermal imaging camera to isolate the heated grains as they diffused was developed. Multiple methods of analyzing the video-data acquired during the experiment were tested, eventually resulting in an average grain diffusion by frame. A method was then developed to calculate the area of the diffusion event based upon the statistical probability of a particle being at a specific pixel location. This method was then modeled to determine the rate of particle diffusivity and, by using the Stokes-Einstein Relation, a diffusion coefficient, $D_{experimental}$, was calculated. It was observed that the $D_{experimental}$ was orders of magnitude larger than expected; in attempting to explain this unexpected outcome, $D_{theoretical}$ was calculated and it was discovered that the diffusion does not appear to be thermally driven. Based on the highly dynamic nature of vibrating grains, a model was created to determine a kinetically driven diffusion coefficient we call ballistic self-diffusion, $D_{ballistic}$. When comparing $D_{ballistic}$ to $D_{experimental}$, it was found that they were on the same order of magnitude and both reflected within the linear modeling of the experimental data; this indicated that the dispersion event is most likely kinetically driven.

DEDICATION

I would like to dedicate this work to my family, my veteran brothers and sisters, and the SSVE (Society of Student Veterans in Engineering) for their unwavering support and contributions.

ACKNOWLEDGEMENTS

I would like to thank and acknowledge my advisor Dr. Russell Keanini, Dr. Jerry Dahlberg, Dr. Artur Wolek, and Dr. Peter Tkacik for their time, expertise, and support in this research. I would also like to thank and acknowledge the SERVE (Shaping Experiential Research for Veteran Education) Program, The Office of Naval Research grant N000142012688, and the University of North Carolina Office of Undergraduate Research Program for providing funding for this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

FOI	Fluid of Interest
FOV	Field of View
FPS	Frames per Second
GOI	Grain(s) of Interest
MFP	Mean Free Path
MS	Molecular Simulation(s)
PIV	Particle Imaging Velocimetry

CHAPTER 1: INTRODUCTION

1.1 Background

When an outside influence disturbers a fluid, the fluid attempts to dissipate the disturbance through a relaxation process [1]. On a macroscopic scale, the fluid can be treated as a continuum and will dissipate as flow, diffusion, or thermal conduction [1]. On a microscopic scale, a disturbance results in a multitude of molecular collisions; the seemingly random dynamic interactions between the fluid particles must be considered and the continuum assumption cannot be used [1, 2]. This originated based on Smoluchowski's proposition and was continued upon by Einstein, which considers liquid as a continuous medium where inhomogeneities are created by thermal fluctuation, resulting in density and concentration fluctuations [3, 4, 5, 6]. Einstein also continued the work of Robert Brown, relating the molecular motion of particles to a random walk process, this established the theory of Brownian motion [5, 6, 7]. The 1845 work of Sir George Stokes suggested that the friction of particle interaction is proportional to viscosity and particle radius; Einstein built upon this to establish a relationship between the particle diffusion coefficient and transitional friction known as the Stokes-Einstien equation [5, 6, 7, 8]. Traditional methods for the study of molecular hydrodynamics and the determination of the diffusion coefficient include light scattering, molecular dynamic simulations, and, to an extent, neutron scattering [1, 2].

1.2 Traditional methods

1.2.1 Dynamic light scattering

As light passes through a medium, if its particles are smaller than the wavelength of the light, the light is diffracted [9]. In the visual spectrum, this can be observed as the blue color of the sky; this color is the result of sunlight scattering in a specific way as it passes through the particles in the earth's atmosphere [6]. In practice, light scattering aims a laser at a sample and detects the angle that the laser's light is diffracted. The way the light scatters is a function of the size and shape of the molecules the light encounters; this provides information about the material composition of what is encountered [10]. If the molecules in the fluid of interest (FOI) are dynamic, motion is able to be related to size based upon the fluctuations in the diffraction [10]. The fluctuations can then be processed using correlation to determine structural and dynamical information about the FOI [10]. While light scattering is a useful method to determine diffusion coefficients, the laser light may excite undesirable modes by operating at a wavelength longer than the spacing between liquid molecules [2]. This method also requires a laser, detection equipment, and specialized software.

1.2.2 Neutron scattering

Thermal neutrons have wavelengths and energies that correspond with the excitation energies of condensed matter, such as liquids and solids; additionally, they only weakly interact with matter, allowing for thicker samples to be used [11, 12]. Neutrons have an integral magnetic moment property that causes them to spin in a very specific way; this makes them susceptible to deflection from magnetic fields [11, 12]. An uninterrupted beam of neutrons will encounter a detector in a predictable way; however, when a FOI is placed between the neutron source and the detector, the neutrons are deflected and will encounter the detector in a way unique material composition [11]. Additionally, any motion/ vibration in the fluid will also affect the deflection pattern as a frequency, providing data that can be used to find the diffusion coefficient [11]. While an effective method for determining a diffusion coefficient, neutron scattering requires a neutron source and complex instrumentation; this makes it a very costly and inaccessible method for the determination of diffusion coefficients [13].

1.2.3 Molecular dynamic simulations

Molecular dynamic simulations use a computational method to define and calculate probable particle interactions [1, 14]. There are numerous existing software packages available to run these simulations or they can be created using freely available coding platforms like PYTHON [15]. These simulations are easily modified and avoid unintended experimental impurities; however, all laws of energy, any variables, and material properties must be accurately defined for the simulation to hold any level of accuracy [15, 16]. The computing cost and time required increase based on the complexity of the simulated system, often rising beyond the capabilities of commonly available equipment [17]. Additionally, there are other spatial/ temporal domain limitations, boundary conditions, and complex particle flow interaction conditions that molecular simulations (MS) are not able to address [2].

1.3 Analog methods

Analog methods for researching molecular hydrodynamics use macroscopic substitutes for molecules, allowing for the movements to be visually observed and recorded. The dynamics of confined grains subjected to a vibratory force have been observed to have fluid-like properties [14, 18]. With the obvious size difference between a molecule and a stand-in grain analog, there is a noticeable difference in mass; however, the laws of classical mechanics can be applied in both cases [18]. Unlike true molecular interaction where quantum-mechanics dictate completely elastic collisions, grain collisions are dictated by the laws of classical mechanics and collision energy is transferred as heat; however, the driving mechanism of the grain bed continues to input energy in the system, offsetting the system energy loss [18]. While molecules of the same type are identical, grains contain imperfections causing them to be similar but not identical; to simplify calculations, a spherical assumption is used, disregarding potential grain imperfections [18]. The spherical assumption is also commonly used when addressing real molecules, as they are not perfectly spherical [18]. In the real or analog case, the non-spherical properties would cause rotational components to collision interactions that are neglected by the spherical assumption in both cases [18]. Even with the notable differences between actual molecular flow and analog grain flow, no other method for the direct observation of a physical experiment that reflected molecular hydrodynamic interaction was located.

1.3.1 PIV overview

The dominate method for flow visualization through grain tracking in a vibrating bed is Particle Imaging Velocitmetry (PIV). PIV systems typically seed a fluid in a flowing channel with a reflective media that is illuminated with a laser; the laser illuminated field-of-view (FOV) is then recorded by a camera [19]. Another method for obtaining images for analysis is known as adaptive PIV. Adaptive PIV can use a standard video and does not require a laser; however, the objects being tracked must be sufficiently visible for the PIV software to differentiate them from background [20]. In PIV, the camera's FOV is calibrated to allow conversion between pixel size and a physical length measurement; the frame rate of the camera is then used to determine time [19, 20]. The software then converts pixel groups into interrogation areas and is able to differentiate the particles from background [20]. By determining the change in location and direction of particles between frames, the PIV software is able to create a flow vector for each of the interrogation areas [20]. Combining the interrogation areas, the PIV software is able to create a vector flow-field, providing a mathematical and visual representation of the flow dynamics [20].

1.3.2 Analog methods using PIV

Previously completed experiments have used a vibrated grain bed to simulate flow by facing a camera at the surface of the dynamic grain bed and recording the movement of the grain for further analysis. The University of North Carolina at Charlotte proposed a method using vibrating grain beds as a means of studying molecular hydrodynamic flow [2]. This method used an annular shaped vibratory polishing bowl filled with 10 mm thick, 10 mm equilateral triangle shaped media and a high-speed camera oriented to record the grain surface; this is the exact vibratory polishing bowl that would later be used in the experiment detailed in this paper [2]. The camera used featured a resolution of 1504 x 1128 and recorded at a rate of 1000 frames per second (FPS); this allowed the grain flow to be captured in great detail, increasing accuracy during data processing [2]. A stationary plate was placed into the grain to deflect the flow and the vibratory bowl was then turned on; the grains were then allowed to achieve a steady-flow state [2]. After the steady-flow state had been achieved, the camera was activated and recorded the grains flowing around the plate; this video data was then processed using PIV software to create vector fields at each time increment, showing the flow dynamics of the grain system [2]. Results of the PIV analysis were then compared to a Computational Fluid Dynamics (CFD) model of the same system and found to be in agreement [2].

This experiment used the same general setup; however, the deflection plate previously used was replaced with a load cell mounted cylinder to determine drag [21]. Unlike the first experiment, this experiment tested 8 different types of grain media as analogs, including the 2mm diameter spheres used in this experiment [2, 21]. This experiment was also able to determine additional fluid properties from the grain bed, such as the drag coefficient, Reynolds number, and Kinematic Viscosity [21].

CHAPTER 2: METHODOLOGY

2.1 Intent

Previous experiments conducted at The University of North Carolina at Charlotte calculated fluid properties from granular flow using a high-speed camera and PIV software; however, these calculations were made from a bulk-flow observation of all particles in the flow field passing through the camera's FOV [2, 14, 21]. To provide an alternative method of conducting grain flow analog experiments and to further support the use of vibrating grain beds in molecular dynamics research, it was desired find a method of recording the dynamics of a single grain or relatively small number of grains as they were affected by the dynamical motion of the bulk flow.

2.2 Proposed grain distinguishment methods

Multiple methods of distinguishing grains were proposed and evaluated. Since the end goal of this project was the determination of a self diffusion coefficient, maintaining the greatest amount of similarities possible between the grains of interest (GOI) and all other grains was prioritized. Painting or coating the select grains was proposed and dismissed due to the paint changing the surface properties of the select grains from the bulk grain surface properties. Acquiring a different color grain with the same physical properties was also proposed; however, removing one or a select few grains from the thousands of other grains in the bed after each experimental run was too impractical of a method.

The use of high-speed X-ray detection or fluoroscopy, a specialized form of combining MRI technique with X-ray, was also considered as a non-invasive way to track grain particles [22, 23]. While fluoroscopy shows promise as a potential method for tracking particle movement, even potentially tracking movement in 3-D, there are multiple difficulties in experimentation [22, 23]. The first challenge was creating enough contrast between the particle(s) of interest and the bulk-flow particles. The use of contrast agents is already commonly used with this technology; however, achieving the necessary definition while minimally altering the grain properties from the background grains proved difficult. Additionally, there are safety concerns due to the radiation associated with x-ray technology [22, 23]. The second difficulty is the accessibility and expense of the equipment necessary to pursue this method of research, likely requiring the cooperation of a hospital or laboratory [22, 23]. While this technique warrants future investigation as a potential method of non-invasive, direct-observation study of 3-D flow dynamics, it was set aside to pursue a technique that only required more accessible equipment.

Heating a grain or a few grains above the background temperature and using thermal imaging technology to differentiate them from the other grains was also proposed. As a basic thermal imaging camera was readily accessible, this method was tested by heating a few ceramic grains on an available hotplate and dropping them on top of the dynamic grain bed; a still image of this initial test is shown in Figure 2.1.

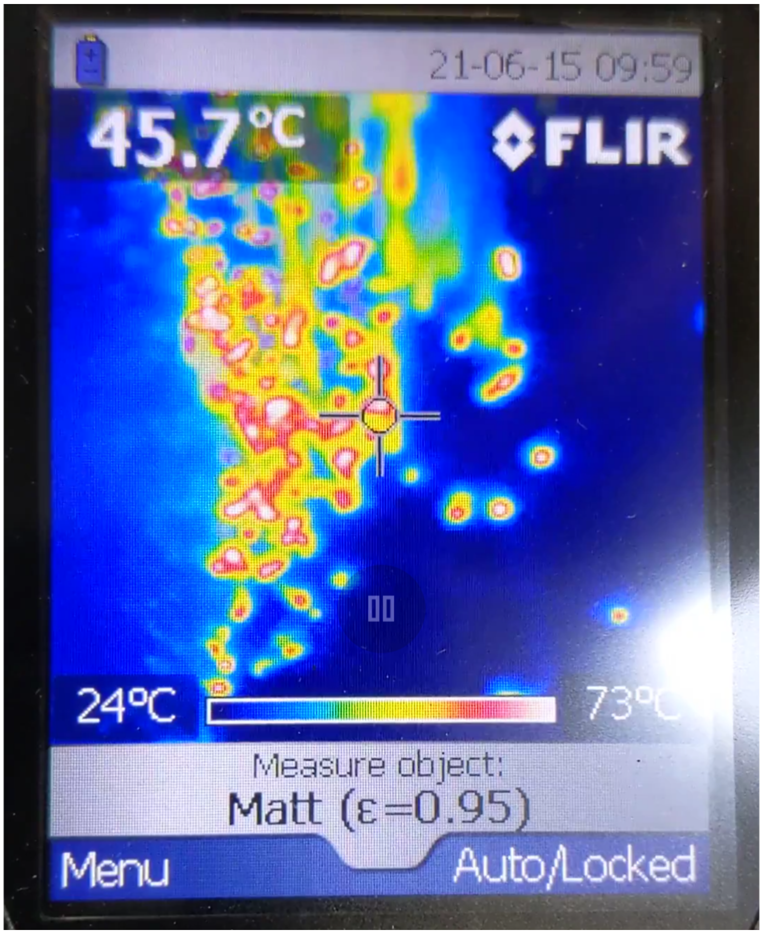


Figure 2.1: Thermal Imaging Proof of Concept test

This test showed that heated grains could be effectively distinguished from other grains that were identical in every other way. This method did not require the grain(s) to be altered in any way aside from the addition of heat and avoided the need for the grains to be retrieved after each iteration of testing. This method was also relatively simple, efficient, and cost effective since it only required a thermal imaging camera and a heat source. Based on the results of this initial test, this method was selected for the experiment. After initial testing and based on the budgetary restraints of this project, the best thermal camera available for this experiment was determined to be a Fluke TiX580 Infrared Camera; this camera is shown if Figure 2.2.

Figure 2.2: Multiple Views of the Fluke TiX580 Infrared Camera

The Fluke TiX580 recorded video in 640 x 480 at 24 FPS [24]. This pixel definition and recording FPS was a significant departure from the high-speed camera used in other experiments where vibrating grains were used to model molecular hydrodynamic processes; however, this was the best attainable camera at our disposal for this experiment [2, 21, 24]. It was decided that, if this method worked with the TiX580 camera's limitations, it would justify the acquisition of a high-speed infrared camera that would only improve upon the accuracy of the experiment.

2.3 Experimental procedure and development

As in similar experiments, a metal structure would be assembled around the vibratory grain bowl to allow the camera and other necessary items to be mounted without being in contact with the vibrating bowl [2, 21]. The intended procedural flow of the experiment was to turn on the vibratory grain bowl, allowing it to come to a steady, dynamic, non-equilibrium condition. A preheated convection oven would be used to heat the grain(s) for a short period of time; then, they would be introduced onto the vibratory bowl grain surface using the funnel and down-tube setup. A recording by the thermal imagine camera was to be initiated immediately before the grains were introduction and stopped after a period of about 20-30 seconds, long enough to ensure that all the heated grains left the FOV with a margin of extra time included in case of an abnormally long flow time. This experiment was to be repeated 100 times each for the single-grain and multi-grain methods to allow for later statistical analysis. By performing multiple tests at different heating temperatures and grain introductory points, this method was determined to work as intended, as outlined in detail in Chapter 3.

2.4 Proposed data analysis method

Following the methodology of previous experiments that used vibrating grain beds as molecular hydrodynamic analogs, the video data from the thermal imaging camera was to be processed using the available Dantec Dynamics PIV software [2, 20, 21]. Using the MATLAB script seen in Appendix A.1, videos would be converted into greyscale, single-frame images; these images would then be imported into the Dantec DynamicStudio 6.5 software [2]. Using the adaptive PIV method, vector fields were output for each experimental iteration and time interval [2, 20]. From the averaged vector field data, a mean flow path (MFP) was created; this can be seen in Figure 2.3.

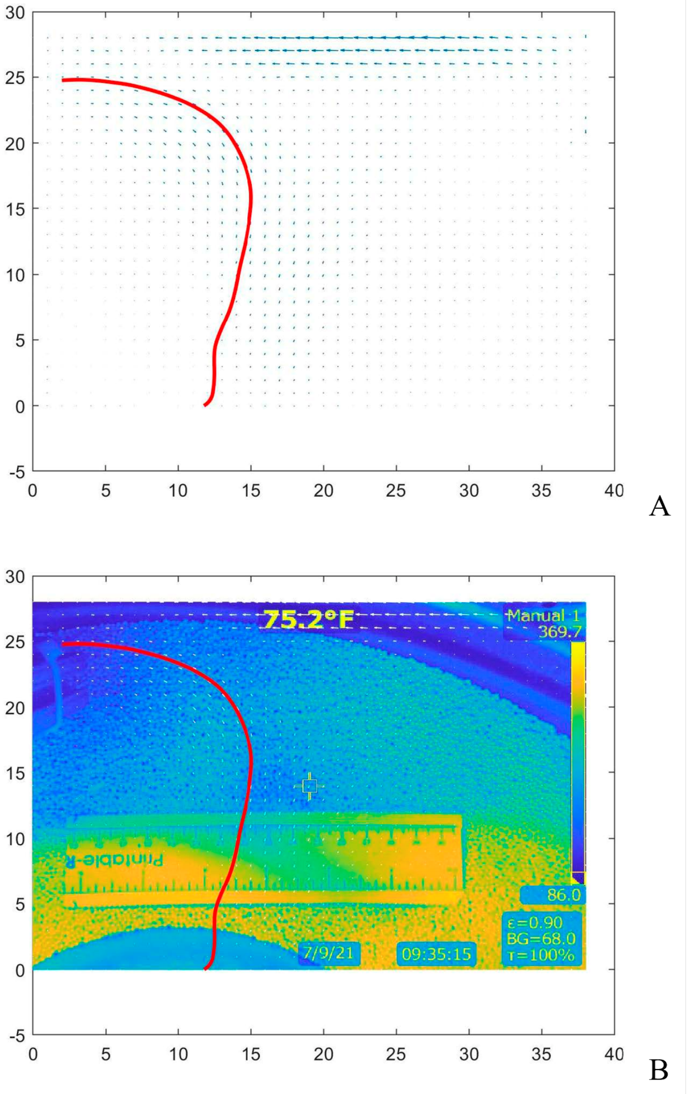


Figure 2.3: A: Estimated mean flow Path on vector field. B: Estimated mean flow path and vector field overlaying the camera's FOV Image

As further analysis of the vector field data was being completed, it was noticed that the image data appeared to lose resolution after being processed in the PIV program. To verify this, a position based average of the runs was completed using the raw data and post PIV processed data; this visually verified that the PIV software significantly degraded the precision of the data, as seen in Figure 2.4.

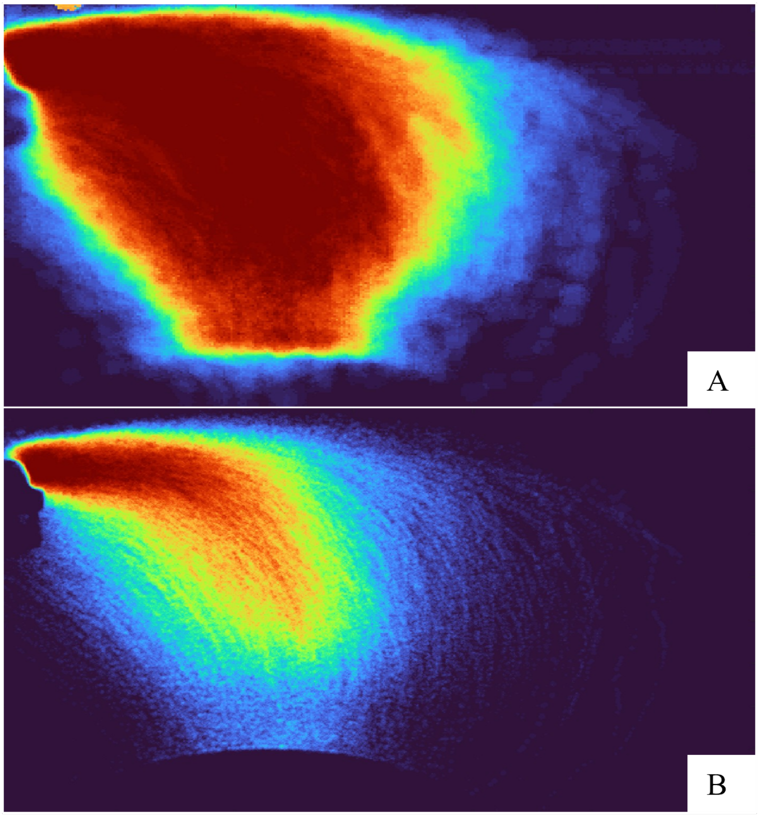


Figure 2.4: A: Position based averaged distribution processed using PIV software. B: Position based averaged distribution without PIV software

As the image precision was so negatively affected by the use of the PIV software, it was decided to avoid using PIV software and, instead, to code a method using MATLAB. Using this method for the analysis, a MFP was calculated using a polyfit function in MATLAB and a linear distribution was conducted along the line normal to locations along the curve; a single-point example of this can be seen in Figure 2.5.

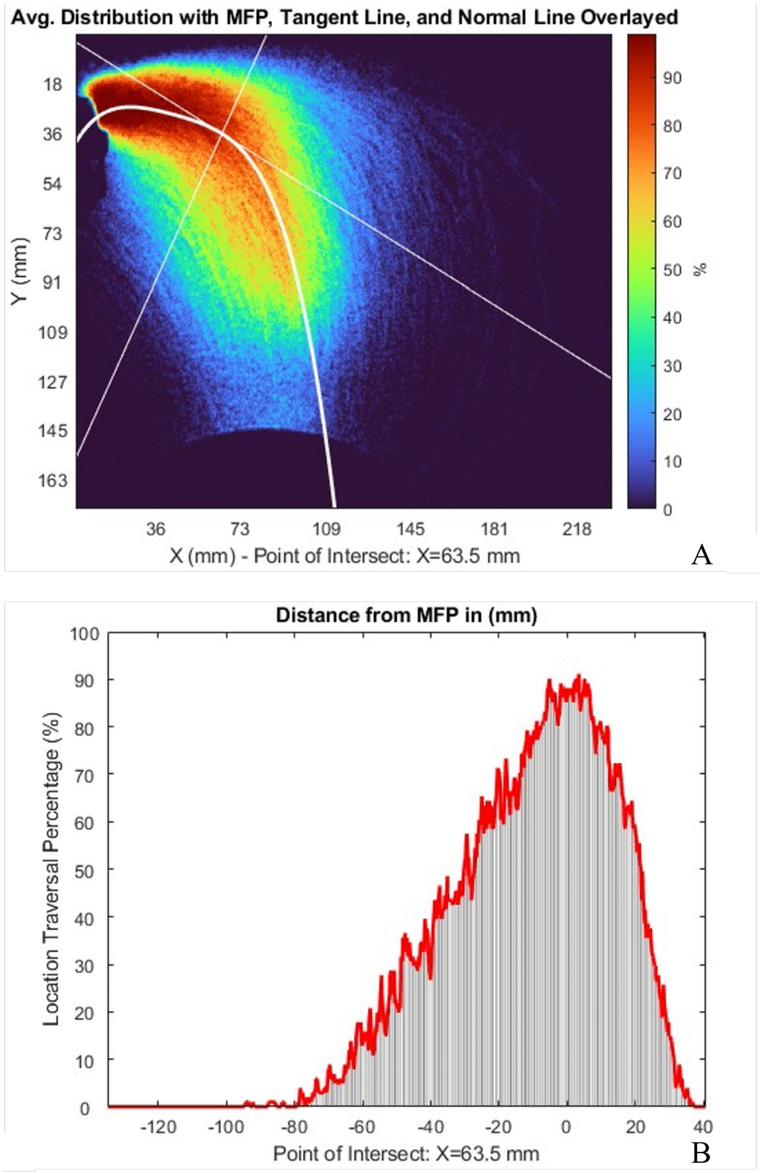


Figure 2.5: A: Overall averaged distribution with best fit MFP line, tangent line at location, and normal line at location. B: Overall averaged distribution along the normal line at location

While this method resolved the resolution issue, provided a reasonable MFP, and showed the average statistical distribution along the MFP, it did not account for the spread with respect to time. When this method was attempted at individual time intervals, the result became blob-like in nature, without a defined shape; a visualization of this can be seen in Appendix B.1. Determining the instantaneous direction of travel without the vector fields created using PIV was difficult; however, images like those seen in Appendix B.1 visually appeared to show a radial Brownian distribution, as seen in Figure 2.6.

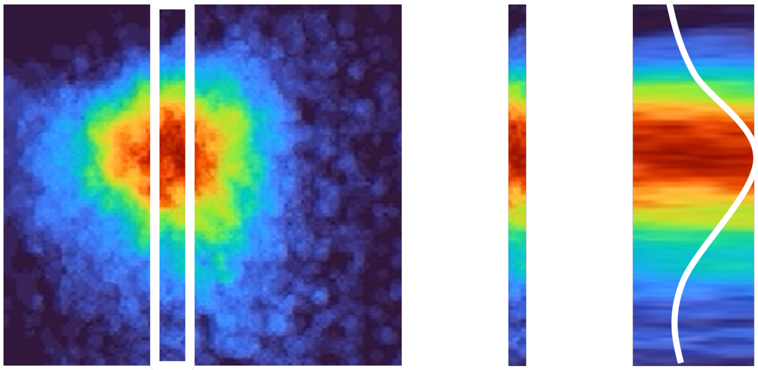


Figure 2.6: Visual observation of a select, cross-section of the blob-like distribution show a radial Brownian distribution in the statistical particle spread

Upon further observation of Figure 2.6, it was also noted that, if a cross-sectional plane was taken at any angle through the blob-like distribution, the Brownian like distribution would remain essentially unchanged. These observations led to questioning if the MFP or the direction of travel was necessary to determine the diffusion coefficient.

When considered from a Brownian motion perspective, the most critical aspect of the movement between time intervals is not the specific direction but the ability to compute the mean-squared particle displacement [25]. Using the blob-like distributions area expansion over time, the change in area over time could be calculated. An existing MATLAB function called convex hull was found to work for these area calculations; the convex hull function calculates the minimum area required to contain a set of 2D points. Then, the known frame rate in conjunction with the convex hull area calculations would be used to determine probable particle displacement at a specific time. Upon testing and verification, this method was used to analyze the experimental data, as detailed in Chapter 4.

CHAPTER 3: EXPERIMENTAL METHOD

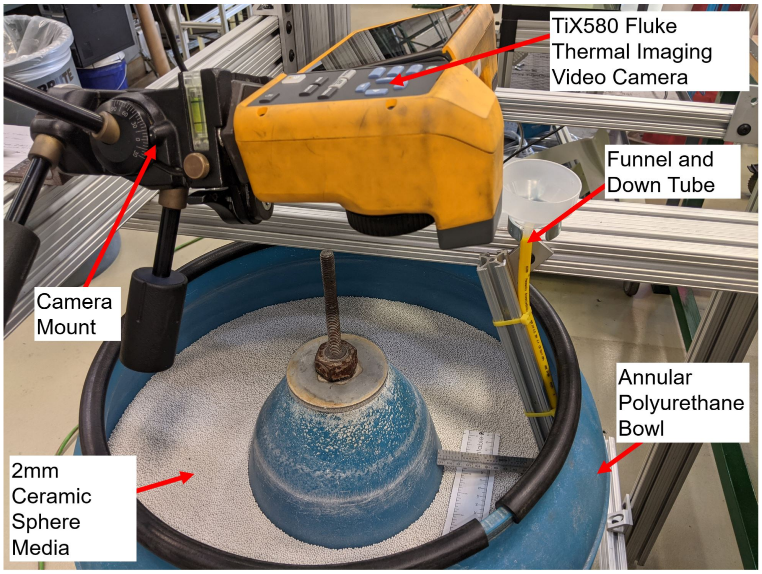


Figure 3.1: General Experimental Setup

3.1 Preparation

The initial setup and protocol of this experiment, as seen in Figure 3.1, were based upon previous experiments conducted at The University of North Carolina at Charlotte [2, 21]. As the experiment evolved, the development and progression described in Chapter 2 served as the basis for the experimental procedure detailed below. Although this procedure is detailed to the exact equipment used in this experiment, it is intended to be easily modifiable to equipment available in other laboratories.

3.1.1 Preparation of the vibratory system

1. The vibratory system consisted of an annular polyurethane bowl with an OD of 600 mm; this bowl was attached to a single-speed (1740 rpm), unbalanced motor, as seen in Figure 3.2, which generated the necessary vibrations for the ex-

periment [2]. The bowl was mounted to an assembly that consists of a weighted base and supported by a group of eight springs, as seen in Figure 3.2 [2].

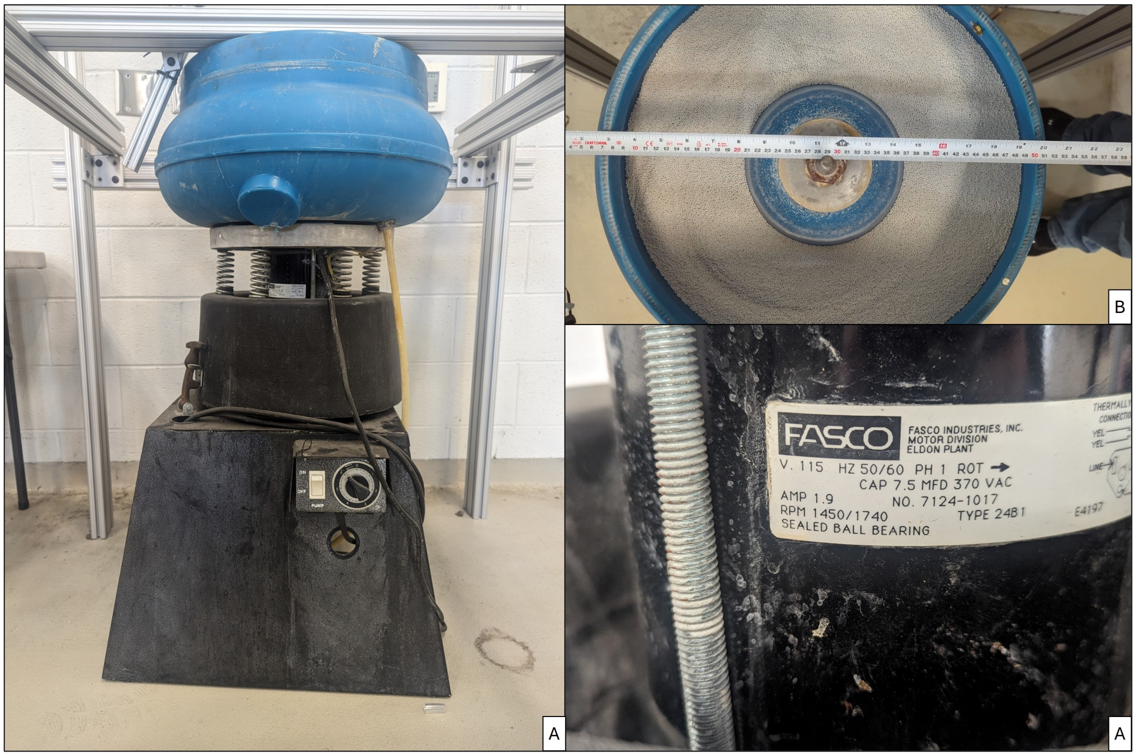


Figure 3.2: Annular shaped vibratory polishing bowl with a single-speed unbalanced motor

2. The bowl was filled approximately half way with white, 2 mm spherical ceramic polishing media.

3. A frame was constructed around the vibrating bowl assembly to mount a funnel with down-tube and a thermal imaging camera. It was important that no part of the frame assembly or anything affixed to it come in contact with the vibratory system, as this would have transmitted vibrations to the camera and funnel, affecting the accuracy of the data collected.

4. Before data collection began, the motor was plugged into a standard electrical outlet and allowed to come to full speed, vibrating the grains at a constant rate for the entirety of the experiment.

3.1.2 Preparation of the thermal imaging system

1. Using a multi-axis arm, a Fluke TiX580 Thermal Imaging camera was mounted to the frame assembly.
2. The TiX580 was connected to a standard electrical outlet, as the duration of this experiment would likely exceed the capacity of the camera's battery.
3. An SD card was inserted into the memory card port located at the bottom of the TiX580 camera.
4. Ensuring that data was being saved to the SD card, TiX580 camera settings were set to: MENU > Settings > Image Storage > SD Card.
5. To set the correct image format, TiX580 camera settings were set to: MENU > Settings > File Format > Image > JPEG.
6. To set the correct video format, TiX580 camera settings were set to: MENU > Settings > File Format > Video > AVI.
7. The TiX580's lens was approximately centered over the expected flow-field.

3.1.3 Preparation of the heat source

1. A standard convection oven was connected to power and preheated to a constant temperature; the oven that was used in this experiment is seen in Figure 3.3. The preheated temperature was later adjusted based on the thermal filtering band chosen (In this experiment, 350°F was used for multigrain experiments and 400°F was used for single grain experiments).

Figure 3.3: Convection oven heat source

2. A small, low-form beaker, capable of withstanding the required temperature, was placed in the oven. This beaker was used to hold the grains as they were heated before being used to pour them into the funnel/ down-tube setup for grain introduction. The beaker remained in the oven when it was not in use to ensure a consistent temperature and to prevent grain cooling during transfer.

3.2 Calibration of the vibratory and thermal imaging systems

1. With the motor in the off position, the outlet end of the funnel and down-tube was placed as close to the grain-bed and outer edge of the annulus as possible; this allowed the grain(s) to be introduced into the flow without disturbing the bulk-flow, as seen in Figure 3.4.

Figure 3.4: Down-tube Placement

2. The motor was turned on; then, the down-tube placement was adjusted so that it did not interact with the dynamical motion of the grains or the annulus.

3. The TiX580 camera was set to the visible spectrum by: MENU > Image > IR-Fusion > IR Level; then the arrow on the touchscreen was moved to the fully to visible spectrum.

4. The multi-axis arm was adjusted so that the cameras field of view included the entire expected flow path of the grains that were introduced to the flow.

5. The TiX580 camera was adjusted to blend the visible and thermal spectrum by: MENU > Image > IR-Fusion > IR Level. Then, the arrow on the touchscreen was moved slightly towards the IR setting, so that the visible spectrum was still clearly seen in the background.

6. The temperature threshold was set by: MENU > Measurement > Set Level/ Span > Check the Manual box > Set Level/ Span; then the max./ min. arrows on the touchscreen were moved so that the the minimum temperature setting was above background temperature but below the oven temperature.

(The grain-bed temperature rose during the long amount of time that bowl was vibrating during this experiment; to ensure a neutral image background, it was important to maintain a high enough thermal floor threshold. This prevented thermal noise in the recordings and ensured that the data was usable. In this case, Single-grain used a temperature span of 96.6°F-189.1°F and multigrain used 126.5°F-275.9°F.)

7. The TiX580 touchscreen was adjusted to display the image only by: MENU > Image > Display > check the Image Only box.

8. The amount of grains desired for the experiment (1/4 teaspoon was used for this multigrain experiment; however, the main importance here is that the same measurement was used throughout the experiment.) were placed in the beaker; then, the beaker was placed into the preheated oven.

9. The grains were heated for 2 minutes. (This amount of time was experimentally determined and can be adjusted, as long as the grains absorb enough heat to appear on camera for the duration of the experiment.)

10. Using thermal hand protection, the beaker was removed from the oven and it's contents were poured into the funnel. On the TiX580 camera screen, the thermally illuminated grains were observed flowing across the screen. The visible portion of the blended image was used to ensure that the entirety of the grain flow path in the annulus was captured by the camera. The thermal portion of the blended image was used to ensure that the grains appear clearly as they moved across the screen, as seen in Figure 3.5. If the heated grains did not clearly move across the FOV or if they flowed out of the FOV, adjustments were made as needed until a clear calibration image was obtained.

Figure 3.5: Thermal and Visible Spectrum, Blended Image of Heated Grains moving across the Flow-Field

11. The motor was turned off and the grainbed was allowed to return to a static position.

12. A scale of known length was placed on the surface of the static grain bed without disturbing the bed; it was ensured that the scale measurements could be clearly viewed by the camera, as seen in Figure 3.6.

Figure 3.6: Scale Calibration Image

13. The green button on the lens side of the TiX580 Camera was depressed to take a calibration picture; then, save on the touchscreen was pressed. It was of the upmost importance that, once this picture has been taken, the camera was not moved for the duration of the experiment; if the camera had been moved, there would have been no way to determine the scale of the data collected.

3.3 Experimental procedure

1. The TiX580 camera was set to the IR Spectrum by: MENU > Image > IR-Fusion > IR Level; then, the arrow on the touchscreen was set fully to the IR end of the scale.

2. The TiX580 camera was set to record video by: MENU > Camera > Video > check the Video ONLY box on the touchscreen; then, the Record Video button was pressed on the touchscreen.

3. Using thermal hand protection, the beaker from the pre-heated oven was removed and the desired number of particles was placed in the beaker (single particles were placed in the beaker by hand and a measuring spoon was used for multigrain experiments); then, the beaker was returned to the oven.

4. A timer was set to ensure that the grains were heated uniformly for each iteration of the experiment (2 minutes were for this experiment); when the timer expired, using thermal hand protection, the beaker was removed from the oven.

5. The green button on the TiX580 camera was pressed to begin recording video.

6. The heated grain(s) were poured into the funnel.

7. The heated grains were observed moving across the FOV of the TiX580 camera. (It was ensured that the camera did not re-calibrate during the recording, as this may have caused data corruption; if the camera re-calibrate, the data was

disregard and the experimental run was repeated.) To ensure that the entire experimental run was captured in each video and that all video recordings were at least as long as the longest experimental run, each video was recorded for a period after the experiment had exited the FOV (In this experiment, each recording was at least 20 seconds long).

8. To stop the recording, the green button was pressed again; then, done was pressed on the touchscreen, followed by save.

9. The grain heating and recording process was repeated until the required number of experimental runs had been recorded (100 runs were recorded for this experiment).

10. When the experiment was complete, power was secured to the oven, vibrating polisher, and TiX580 camera. Then, the SD card was removed from the bottom of the camera and kept for analysis.

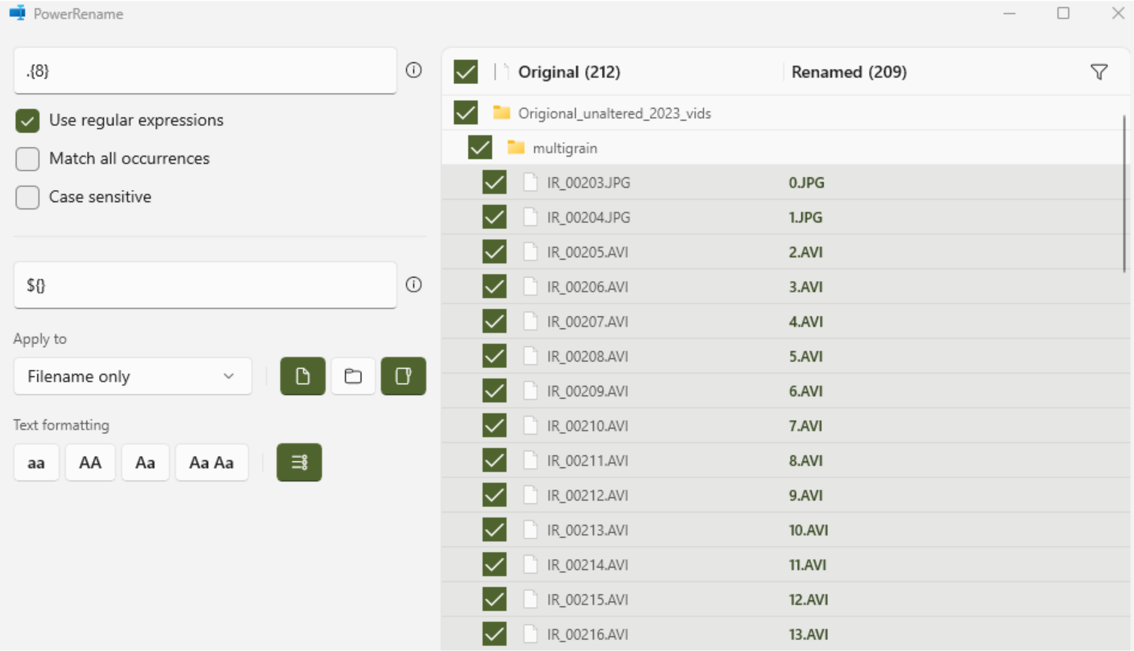
CHAPTER 4: DATA ANALYSIS

4.1 Video file management

1. The SD card containing the data files from the experiment was inserted into a computer SD card reader; the files were copied onto a secondary storage location (A portable hard drive was used for this experiment) for analysis.

2. PowerRename was opened by right clicking the folder containing the video files (If necessary, PowerRename was downloaded as a part of the Microsoft PowerToys utility); then, the drop-down menu was selected in the PowerRename drop down menu.

3. The names of all the video files were changed from their default name into a numerical sequence to assist upcoming MATLAB analysis; the PowerRename commands that accomplished this can be seen in Figure 4.1.



.{8}

☑ Use regular expressions

☐ Match all occurrences

☐ Case sensitive

${}

Apply to: Filename only

☑ Original (212)	Renamed (209)
☑ Origional_unaltered_2023_vids	
☑ multigrain	
☑ IR_00203.JPG	0.JPG
☑ IR_00204.JPG	1.JPG
☑ IR_00205.AVI	2.AVI
☑ IR_00206.AVI	3.AVI
☑ IR_00207.AVI	4.AVI
☑ IR_00208.AVI	5.AVI
☑ IR_00209.AVI	6.AVI
☑ IR_00210.AVI	7.AVI
☑ IR_00211.AVI	8.AVI
☑ IR_00212.AVI	9.AVI
☑ IR_00213.AVI	10.AVI
☑ IR_00214.AVI	11.AVI
☑ IR_00215.AVI	12.AVI
☑ IR_00216.AVI	13.AVI

Figure 4.1: PowerRename Setup for mass video file renaming

4.2 Video to binary image conversion

A MATLAB program, shown in Appendix A.1, was created to trim all of the .AVI videos to a uniform length; while, simultaneously, converting the videos into binary, .tif images. An example of this binary conversion can be seen in Figure 4.2.

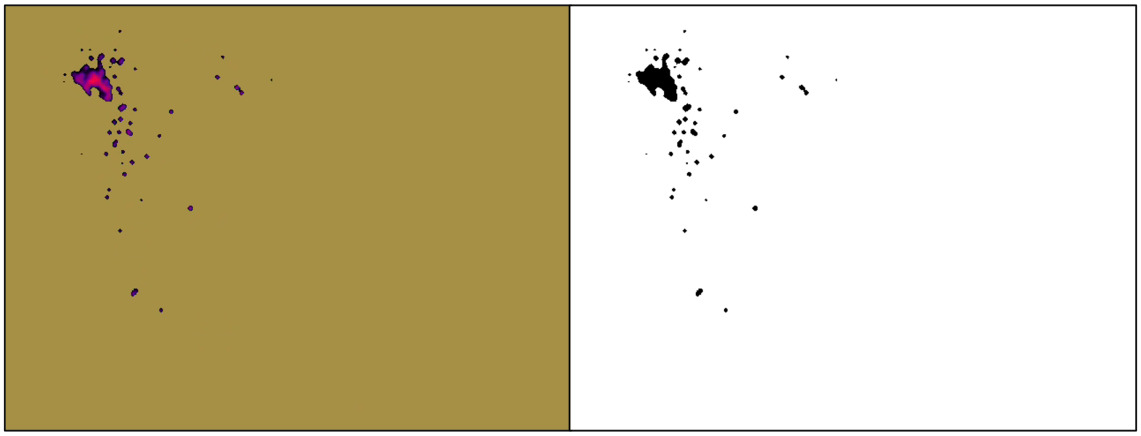


Figure 4.2: Binarized example of a single frame of an experimental run (Right: Origional, Left: Binarized

This program created numerically ordered folders for each individual video; then, the individual video frames were numbered and placed within those video folders. The purpose of the numerical ordering was to setup for subsequent MATLAB analysis of the data files.

4.3 Event identification

A MATLAB program, shown in Appendix A.2, was created to identify the first and last appearance of heated grains in the binarized video frames. Since the images had already been binarized, this was accomplished be determining when one pixel value differed from the rest of the pixels values. By identifying the first appearance of a grain, $Frame_{Initial}$, the start time of each respective grain introduction was able to be identified and set as the first frame of each respective run; this created a unified start time. Next, the last appearance of a grain, $Frame_{Final}$, in each respective run was identified; Eqn. 4.1 was then applied to each video to determine the required number of frames.

$$Frame_{required} = (Frame_{Final} - Frame_{Initial}) + 1 \tag{4.1}$$

By comparing the $Frame_{required}$ value given by Eqn. 4.1 for each video, the largest number calculated was used as the minimum length of frames necessary to complete the experiment. Then, the same MATLAB program shown in Appendix A.2 extracted the required number of frames, starting at $Frame_{Initial}$, for each run and placed them in a folder, keeping their previously assigned video number. Then, the frames were numerically renamed using $Frame_{Initial}$ as frame 1. Since the original videos files were recorded at 24 FPS, the files were able to be translated into seconds based on Eqn. 4.2; in this experiment, 212 frames were required, which translated into approximately 8.83 seconds.

$$\triangle t = (Frame_{Final} - Frame_{Initial}) * (\frac{1s}{24frame}) \tag{4.2}$$

4.4 Calculating average position at each time interval

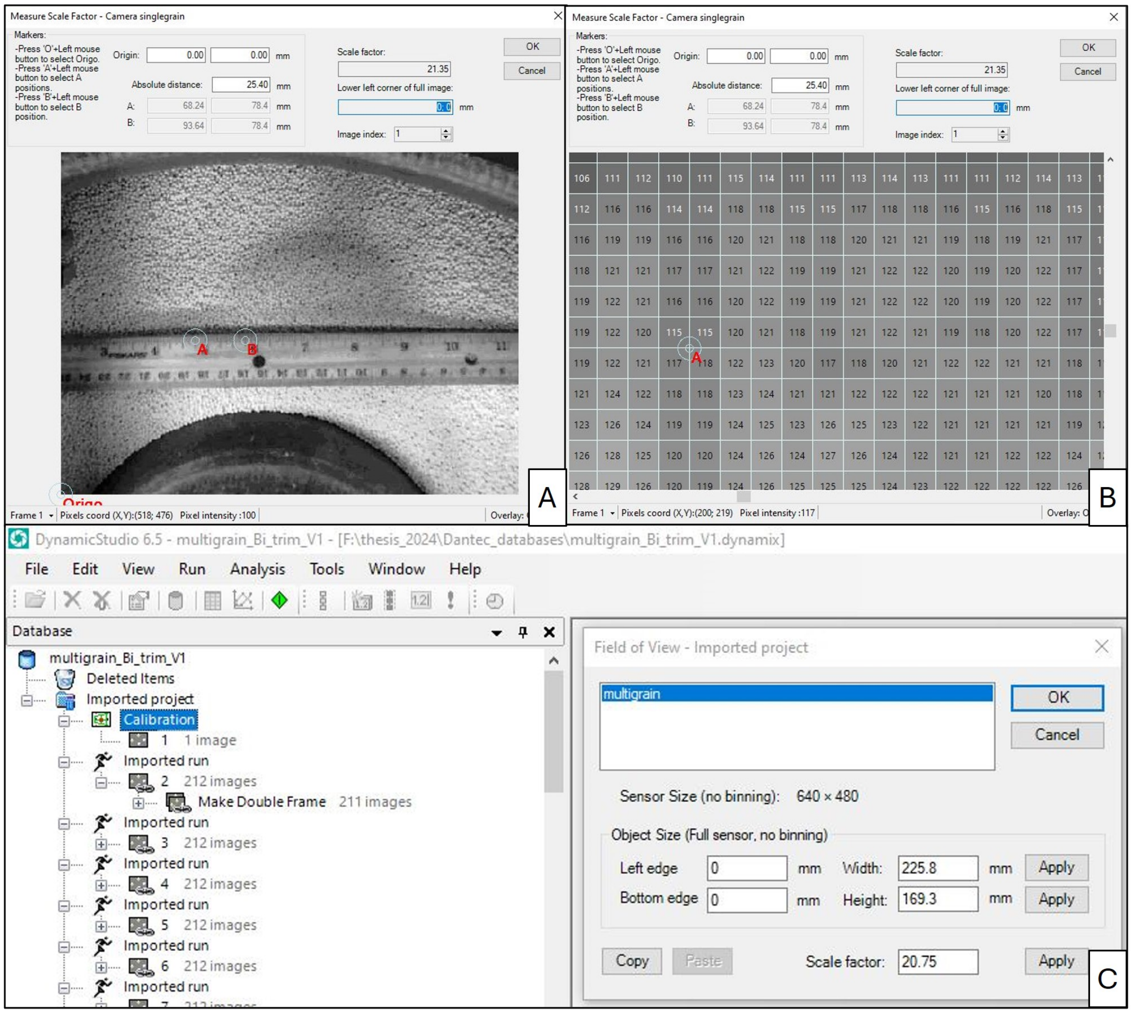
A MATLAB program was created to calculate the matrix, $\mu(t)$, which shows the probability of a grain being at a location in the 640 x 480 pixel grid, $P(x, y)$, at a specific time increment, t ; this program can be seen in Appendix A.3. By using Eqn. 4.3, the previously binarized matrices are averaged at each respective time interval; this calculates the probability that a grain will be at a specific pixel location at a specific time.

$$\mu(t) = \frac{1}{((Run_{Final} - Run_{Initial}) + 1)} * \sum_{i=Run_{Initial}}^{Run_{Final}} \begin{bmatrix} P(x_1, y_1) & \dots & P(x_{640}, y_1) \\ \vdots & \ddots & \vdots \\ P(x_1, y_{480}) & \dots & P(x_{640}, y_{480}) \end{bmatrix} \tag{4.3}$$

Where:

$$\mu(t) = \begin{bmatrix} \overline{P}(x_1, y_1) & \dots & \overline{P}(x_{640}, y_1) \\ \vdots & \ddots & \vdots \\ \overline{P}(x_1, y_{480}) & \dots & \overline{P}(x_{640}, y_{480}) \end{bmatrix} \tag{4.4}$$

Using the calibration image seen in Figure 3.6, the 640 x 480 pixel grid was converted from pixel units to mm using the calibration function in Dantec Dynamics Studio 6.5 PIV Software, as seen in Figure 4.3.



Measure Scale Factor - Camera singlegrain

Markers:
-Press 'O'+Left mouse button to select Origo.
-Press 'A'+Left mouse button to select A positions.
-Press 'B'+Left mouse button to select B position.

Origin: 0.00 0.00 mm
Absolute distance: 25.40 mm
A: 68.24 78.4 mm
B: 93.64 78.4 mm

Scale factor: 21.35
Lower left corner of full image: 0; 0 mm
Image index: 1

Frame 1 Pixels coord (X,Y):(518; 476) Pixel intensity :100

Frame 1 Pixels coord (X,Y):(200; 219) Pixel intensity :117

106	111	112	110	111	115	114	111	111	113	114	113	111	111	112	114	113
112	116	116	114	114	118	118	115	115	117	118	118	116	115	116	118	115
116	119	119	116	116	120	121	118	118	120	121	121	119	118	119	121	117
118	121	121	117	117	121	122	119	119	121	122	122	120	119	120	122	117
119	122	121	116	116	120	122	119	119	121	122	122	120	119	120	122	117
119	122	120	115	115	120	121	118	119	121	122	121	119	118	120	122	117
119	122	121	117	118	122	123	120	117	118	120	121	122	122	121	121	118
121	124	122	118	118	123	124	121	121	122	122	122	121	121	121	120	118
123	126	124	119	119	124	125	123	126	125	123	122	121	121	121	121	119
126	128	125	120	120	124	126	124	127	126	124	122	121	121	122	122	124
128	129	126	120	119	124	126	125	125	125	124	123	122	122	122	122	126

DynamicStudio 6.5 - multigrain_Bi_trim_V1 - [F:\thesis_2024\Dantec_databases\multigrain_Bi_trim_V1.dynamix]

Field of View - Imported project

multigrain

Sensor Size (no binning): 640 × 480

Object Size (Full sensor, no binning)

Left edge: 0 mm Width: 225.8 mm
Bottom edge: 0 mm Height: 169.3 mm

Scale factor: 20.75

Figure 4.3: Dantec Dynamics Studio 6.5 software used to determine the conversion factor between pixel length and mm. A: Measuring Scale Factor Screen with imported calibration image. B: Zoomed in version of "A" down to individual pixel grid view. C: FOV dimensions based on calibration image scaling.

Using the data obtained, width: $\frac{225.8}{640}\frac{mm}{pixel} = 0.3528125\frac{mm}{pixel}$ and height: $\frac{169.3}{480}\frac{mm}{pixel} = 0.352708333\frac{mm}{pixel}$, a conversion factor of $\approx 0.353\frac{mm}{pixel}$ was used to covert the 640 x 480 pixel grid to a 225.8 x 169.3 mm FOV. Using Eqn. 4.2, Eqn. 4.3, and the pixel to mm conversion seen in Figure 4.3, it was possible to calculate the statistical probability of a particle being at a location in the FOV at a specific time; this was completed for each time increment. For visualization purposes, select time increments are shown in Appendix B.1.

4.5 Determination of the grain dispersion area at each time interval

A MATLAB program was created to calculate the area of the statistical particle spread at each time interval. This program utilized the convex hull (convhull) function in MATLAB to calculate the area of the particle spread event at each time interval. However, locations of statistically low probability caused the area calculations to include large areas with no statistically determined grain appearances; additionally, if the program was run for to great a time interval, as the event dissipated to very low detection levels, the program also failed. Using trial and error, it was determined that the program was able to calculate the area of the event at frames 20-70 ($\approx 0.83 - 2.92$ seconds in 1/24 second increments) by filtering out events below 10% and above 90% probability. This program can be seen in Appendix A.4. For visualization purposes, the effect of the filtration at select percentages with the outline of the convex hull area calculation can be seen in Appendix B.2.

Filtering out low probability events allowed the dispersion area to be calculated; however, a side effect of the filtration was a reduction in the area of the dispersion event. The way that this affected the accuracy of the area calculation and the point of filtration that the calculated areas were usable needed to be determined. To accomplish this, a linear fit of the area expansion at each achievable filtration percentage was completed using the linfit and convex hull area calculations in MATLAB; the results of this calculation can be seen in Appendix C.1. The R^2 of the linear fit was then plotted at each percentage filtered, as seen in Figure 4.4.

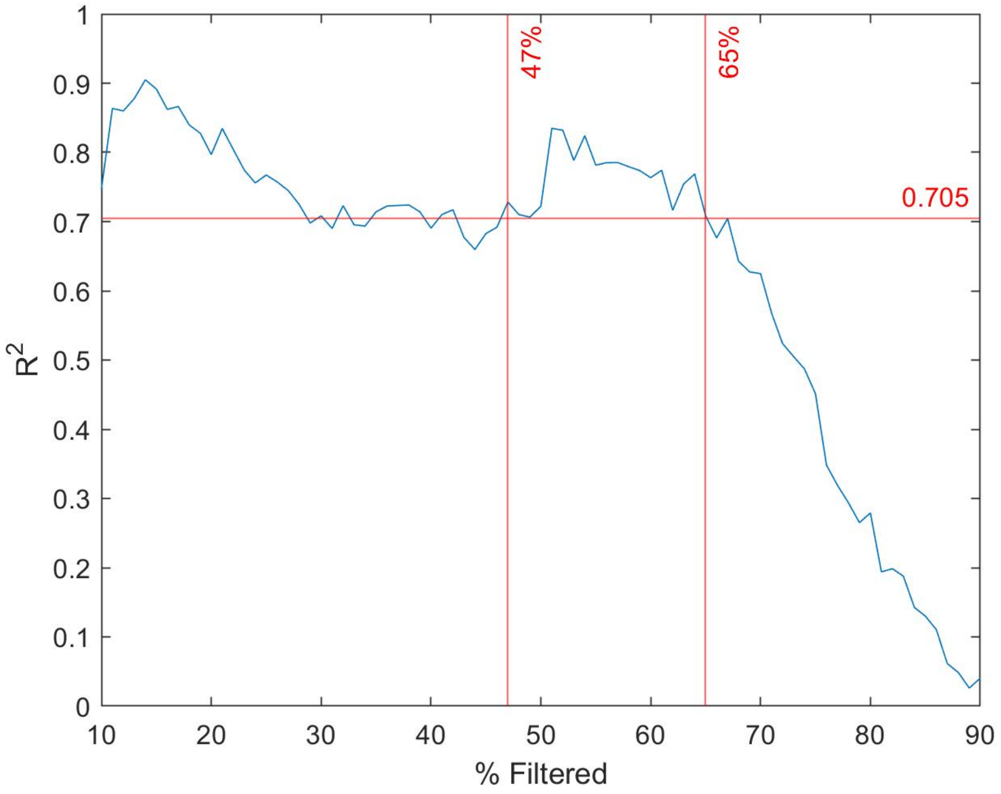


Figure 4.4: R^2 of the linear fit at varying % Filtered. Red vertical and horizontal lines represent the final filtering criteria used for determining the region of data to be used for final calculations.

Based on the results in Figure 4.4, it was observed that the accuracy of the linear fit greatly decreased at high filtration levels. To further understand the relationship of filtration to the linear fits, the slopes calculated from the linear fits were plotted at each filtration level, as seen in Figure 4.5.

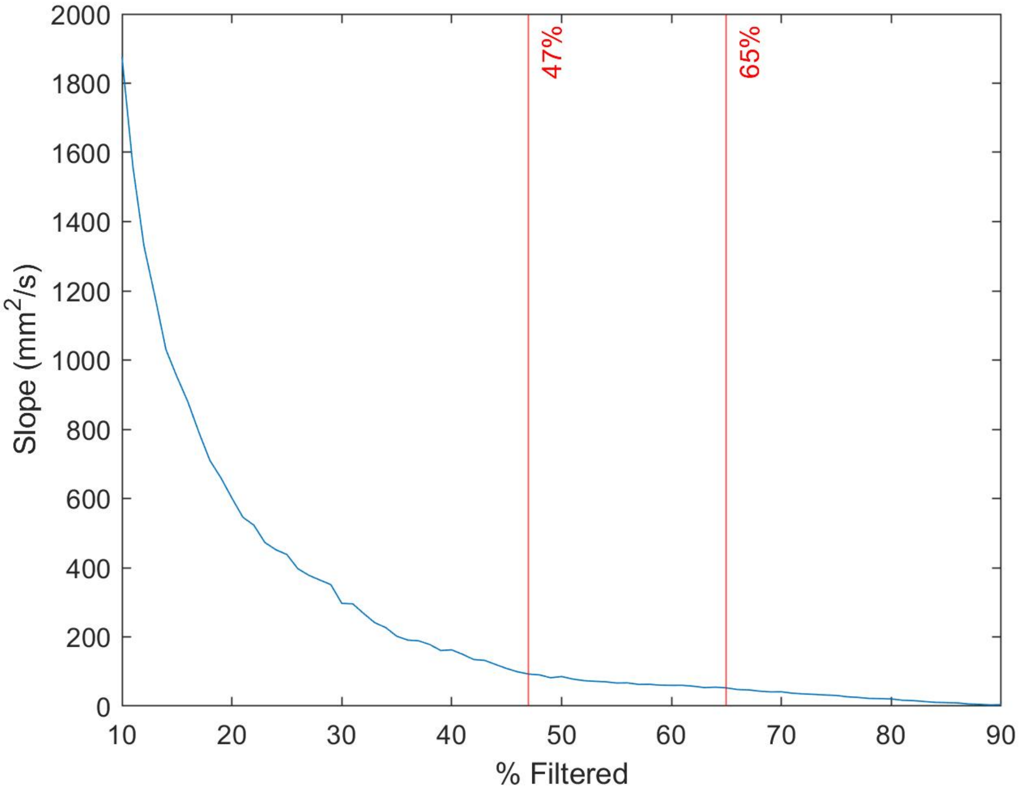


Figure 4.5: Slope of linear fit at varying % Filtered. Red vertical lines represent the final filtering criteria used for determining the region of data to be used for final calculations.

Based on Figure 4.5, it was visually observed that the change in the calculated slope became more linear as the filtration percentage increased. To determine the point of filtration that the change in the slope of Figure 4.5 became sufficiently linear, a linear fit was completed for the entire range of $10-90\%$ filtration; then, the minimum filtration point of this range was increased in 1% increments (Example: $11-90\%, 12-90\%, 13-90\%, ..., 88-90\%, 89-90\%$) for the entire filtration range. The R^2 of each of these linear fits was then plotted, as seen in Figure 4.6.

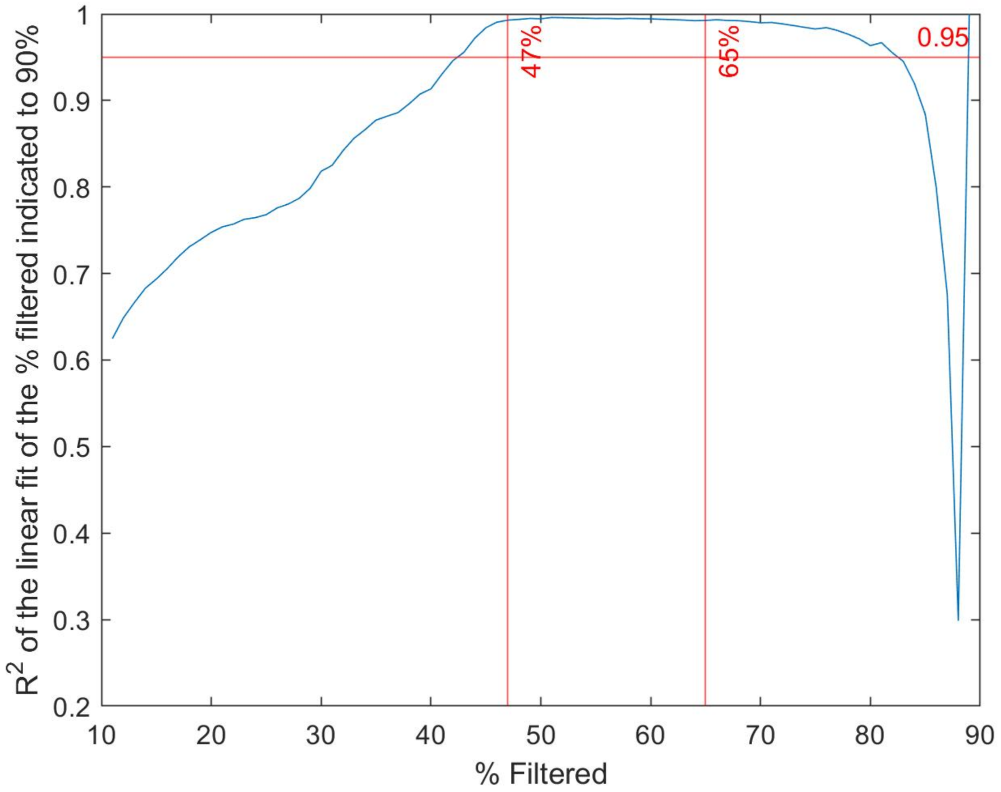


Figure 4.6: R^2 of the linear fit between the % filtered indicated to 90 %. Red vertical and horizontal lines represent the final filtering criteria used for determining the region of data to be used for final calculations.

By choosing a high R^2 value of 0.95, the data in Figure 4.6 showed that 43% was the minimum amount of filtration needed to exceed 0.95 R^2; this indicated that the change in the slope of Figure 4.5 was sufficiently linear above 43% filtration. Next, using Figure 4.4, the maximum R^2 that allowed for no breaks in the data with respect to filtration percentages and for all data to be above the 43% minimum determined in Figure 4.6 was found to be an R^2 of 0.705. This filtration method showed that a range of $47-65\%$ was an accurate range of filtered data that could be used to created a linear fit. These filtration points were indicated by red boundary lines in Figures 4.4, 4.5, and 4.6; within this filter band, the data was determined not to be over or under filtered. Using the slopes of the linear fit from the $47-65\%$ range in Table C.1, reflected in Figure 4.5, a liner fit was completed, as seen in Figure 4.7.

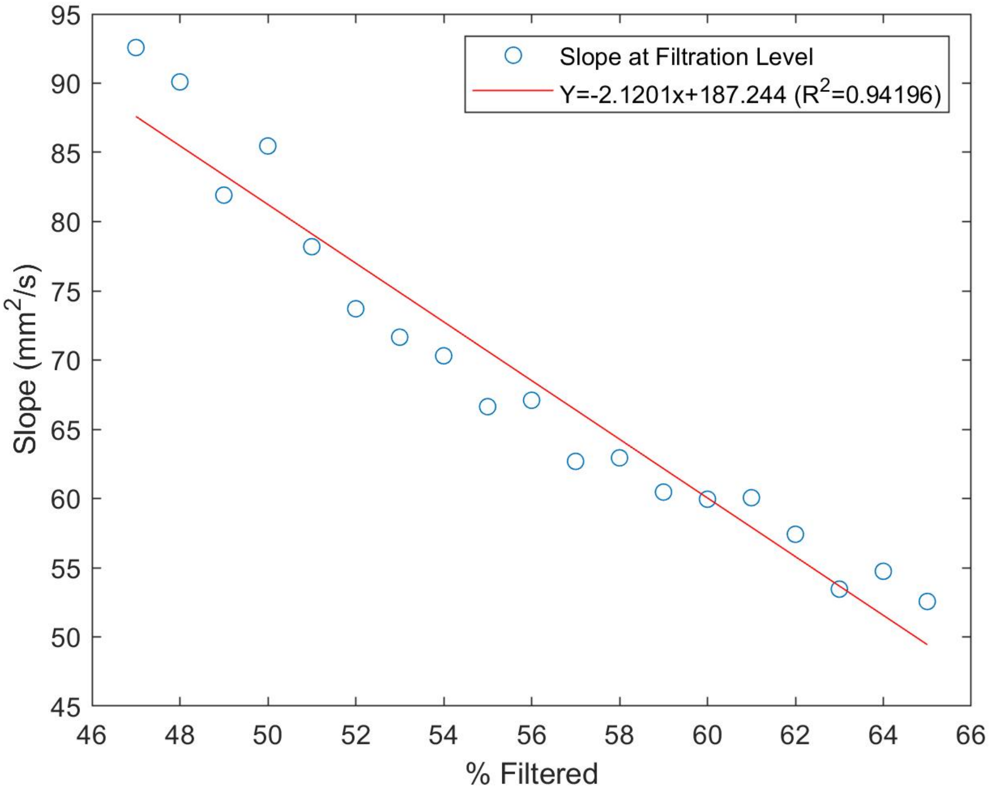


Figure 4.7: Linear fit of the calculated slopes in the $47 - 65\%$ filtered band

The linear fit calculated in Figure 4.7 provided an approximation for determining the slope (rate of area expansion) at 0% filtration at an R^2 of 0.94196. By setting $x = 0$, we calculated $187.244\frac{mm^2}{s}$ at 0% filtration; this allowed us to overcome the inability to complete convex hull area calculations without some level of filtration. This linear expansion of area was described using the point-slope formula in Equation 4.5.

$$\tilde{m} = \frac{A_1 - A}{t_1 - t} = 187.244\frac{mm^2}{s} \tag{4.5}$$

4.6 Determination of the diffusion coefficient

Mean squared displacement is defined in Equation 4.6.

$$MSD \equiv< (x(t) - x_0)^2 > \tag{4.6}$$

Dropping the time dependence notation in Equation 4.6, the particle displacement was shown as $< x^2 >$. Using this notation, a generalized form of the mean squared displacement of particles was related to the diffusion coefficient, as seen in Equation 4.7 [25].

$$< r^2 >=< x^2 > + < y^2 > +... = 2nDt \tag{4.7}$$

As our calculated area was blob-like in shape, without a clearly defined x^2 or y^2 diffusion, we idealized the expansion to a circular form by using the equation for the area of a circle; this resulted in Equation 4.8.

$$< r^2 >= \frac{< A >}{\pi} = 2nDt \tag{4.8}$$

By reorganizing this equation to solve for D, we get Equation 4.9. As this experiment was in 2-dimensions, we set $n = 2$; additionally, we rewrote the equation to reflect that the time interval was constant and equal to the experimental frame rate.

$$D = \frac{< A >}{2\pi nt} = \frac{A_1 - A}{4\pi(t_1 - t)} \tag{4.9}$$

Substituting Equation 4.5 into Equation 4.9, we get Equation 4.10.

$$D_{experiment} = \frac{1}{4\pi}\tilde{m} \approx 14.9\frac{mm^2}{s} \tag{4.10}$$

Equation 4.10 provided our calculated diffusion coefficient, $D_{experiment}$.

4.7 Physical analysis: Ballistic self-diffusion

One of the principal objectives of this project centered on testing the validity of the Stokes-Einstein relation, connecting the self-diffusion coefficient and the kinematic viscosity, ν, in a fluid, at (absolute) temperature, T [5, 6, 7, 8]:

$$D = \frac{k_B T}{3\pi\rho\nu d_g} \tag{4.11}$$

where k_B, ρ, and d_g are, respectively, Boltzmann's constant, fluid density, and the (effective) diameter of the fluid's constituent molecules. An earlier investigation by our research group measured the effective kinematic viscosities of nine different vibrated grain systems, including the 2 mm spherical grains studied here, all undergoing fluid-like flow [21].

4.7.1 Evidence suggesting equipartition of energy in vibrated grain beds

In order to obtain a theoretical self-diffusion coefficient, which we labeled $D_{theoretical}$, we first had to establish that the equipartition theorem approximately held for our vibrated grain system. This requirement arose since:

a) The grain scale version of Equation 4.11 was used, along with the previously measured grain effective kinematic viscosity to estimate $D_{theoretical}$ [21].

b) The calculation required either determination or replacement of the difficult to determine grain scale Boltzmann temperature, $k'_B T_{effective}$; here, it proved much easier to argue the validity of grain scale equipartition of energy, allowing replacement of $k'_B T_{effective}$ with a readily estimated average grain kinetic energy.

First note that our systems existed in a state of near-freezing [21]: When stationary objects are placed in one of our grain flows, a large region of (nominally) stagnating, frozen, grains appear on the downstream side of the object. Focusing on grain bed energetics on time scales shorter than the inter-grain collision time scale, we approximated that our N-grain system was at a solid in which each grain underwent a small amplitude vibration about a nominally fixed equilibrium point. Assuming that the pairwise (collision-induced elastic strain) potential between each pair of neighboring grains was quadratic in relative displacement, then, using a normal mode decomposition, the set of short-time scale vibrations were re-expressed as 3N statistically independent, collective vibration modes. In other words, the time-average, statistically stationary system potential energy, PE_{system}, was expressed as a sum over 3N, statistically independent, collective (i.e., multigrain) potential energies. Similarly, considering short-time scale grain kinetic energy, the same normal mode decomposition allowed us to express the (time-average, statistically stationary) system kinetic energy, KE_{system}, as the sum of 3N independent, collective, multigrain kinetic energies. [Note that on longer near- and multi-collision time scales, inter-grain friction dissipates all (vibrational) input energy in excess of the stationary values of KE_{system} and PE_{system}.]

Based on experimentally measured bed vibration spectra reported in [14], we assumed that the normal mode decomposition produced a set of 3N eigen-frequencies, ranging in magnitude from frequencies much smaller than the bed forcing frequency, $f_o \approx 30$ Hz, up to $n \cdot f_o$, where n was on the order of 4 or 5. Importantly, measured spectra showed that bed vibration was dominated by a large peak centered on f_o, as well as four to five weaker (smaller amplitude) harmonics of f_o. Thus, we assumed: a) that the normal mode mass matrix (the matrix multiplying normal mode velocities) and normal mode potential matrix (the matrix multiplying normal mode displacements) were both strongly diagonal, and b) that each matrix diagonal had one term - corresponding to the eigen-frequency nearest f_o - much larger than all other terms. In other words, neglecting harmonics, experimental spectra indicated that *all N grains* vibrate predominantly at f_o. Likewise, spectra indicate that system (elastic strain) potential energy, PE_{system}, was dominated by a single, whole-bed spring mode; if other spring modes were (strongly) activated, these would have, presumably, broaden the sharp spectral peak at/ near f_o.

Based on this, we assert that the equipartition theorem held for all dynamical degrees of freedom, kinetic and potential, in our system:

$$\frac{1}{2}m\dot{q}_i^2 = \frac{1}{2}k'_B T_{eff} \tag{4.12}$$

and

$$\frac{1}{2}m\omega^2 q_i^2 = \frac{1}{2}k'_B T_{eff} \tag{4.13}$$

where, neglecting grain rotational kinetic energy, q_i and $\dot{q}_i$ are average displacements (i.e., collision-induced grain elastic strains × effective grain diameters) and velocities in 3N coordinate directions, m is grain mass, and where, for simplicity, ω is a fixed, system averaged inter-grain oscillation frequency (determined presumably by the grains' Young's modulus, Poisson ratio and effective diameter, as well as the bowl vibration frequency, f_o, and amplitude, A_o). [Based on the discussion and scaling analysis in section 4.8, we don't need detailed expressions for displacements and the elastic frequency, ω. The only parameters needed are f_o and A_o.]

4.7.2 Hamiltonian (dissipationless) dynamics on sub-collision time scales and a second argument suggesting the validity of short time scale equipartition

Based on the argument above, we noted as an important corollary, that the approximate validity of the equipartition theorem implied that grain dynamics on the short, sub-collision time scale were nominally dissipationless, i.e., Hamiltonian [26]. This important and physically reasonable observation was also made in our earlier study using a different argument; there, we observed that the peculiar (random) velocity magnitude for all grains tested exhibited Maxwell-Boltzmann probability distributions [14]. Such distributions emerged in classical, interacting (as well as non-interacting) multi-molecule systems when: a) random single molecule kinetic energies are statistically independent of all other single molecule kinetic energies, as well as being independent of inter-molecular potential energies, and b) inter-molecular potential energies are likewise independent of molecular kinetic energies.

Focusing on the sub-collision time scale, we assumed that, on this time scale, each grain undergoes statistically independent, harmonic oscillations about nominally fixed equilibrium points; then, in light of the evidence suggesting Hamiltonian dynamics, we arrived at a second argument supporting the validity of the equipartition theorem in vibrated grain systems [14, 26].

4.7.3 Scaling estimate of $D_{theortical}$ using the grain scale Stokes Einstein relation

Having established the apparent validity of the equipartition theorem to our vibrated grain systems, we used Equation 4.12 to replace the grain scale Boltzmann temperature, $k'_B T_{eff}$, in the Stokes-Einstein relation, Equation 4.11, with the approximate kinetic energy of individual grains:

$$k'_B T_{eff} \approx m_g \left(A_o f_o\right)^2 \tag{4.14}$$

Using the replacement in Equation 4.11, we obtained an expression allowing an estimate of D_{theory}, under the critical, and apparently erroneous assumption that measured grain self-diffusion is thrmally driven:

$$D_{theoretical} \approx \frac{m_g \left(A_o f_o\right)^2}{3\pi \rho_g \nu_g d_g} \tag{4.15}$$

where $\rho_g = 1385$ kg m^{-3} (packing density $\approx$ 55%), $\nu_g = 0.2$ m^{2}s^{-1}, and $d_g = 2\ (10^{-3})$ m, are, respectively, the effective/bulk density, kinematic viscosity, and constituent grain diameter of the 2 mm grain flows used in our experiments, and where, again, $A_o \approx 2\ (10^{-3})$ m and $f_o \approx 30$ s^{-1}, are the approximate imposed vibration amplitude and frequency [21, 27, 28]. In order to calculate the grain mass, we used the specific weight of the grain material, $\rho_{swg} = 2.52\frac{g}{cm^3}$, and the formula for the volume of a sphere [27]:

$$m_g = \frac{\rho_{swg}\pi d_g^3}{6} \approx 0.01\text{g} \tag{4.16}$$

Using these parameter values in Equation 4.15 yielded:

$$D_{theoretical} \approx 7.3\ (10^{-9})\ \text{m}^2\ \text{s}^{-1} = 7.3\ (10^{-3})\ \text{mm}^2/\text{s}^{-1} \tag{4.17}$$

Comparing $D_{theoretical}$ with $D_{experimental}$, we observe that the Stokes-Einstein relation under-estimates D by approximately four orders of magnitude.

Crucially, this large discrepancy implies that experimentally observed diffusion was not thermally driven by the random kinetic energy of grains in the vibrated bed. Moreover, the large measured D suggests that the observed "super diffusion" may reflect *ballistic grain transport*, i.e., sub-collision time scale, collision free, elastic-deformation-driven motion of individual grains. We pursued this interpretation of experimentally observed self-diffusion in the following section.

4.8 Ballistic self-diffusion and the inapplicability of the Stokes-Einstein relation

Close examination of the origin of the Stokes-Einstein relation, Equation 4.11, revealed that it only applies to *thermally-driven* self-diffusion, specifically, the self-diffusional random hopping of single particles at some location, $\mathbf{x}$, driven by the random kinetic energy of a (small) system of particles, near $\mathbf{x}$, relative to the (system) average (bulk) velocity at $\mathbf{x}$. Unfortunately, measurement of (position-dependent) thermal energy, i.e., the random component of kinetic energy, which was measured by our group, was not possible in the present experiment [14].

Serendipitously, this limitation exposes what we believe may be a new, previously unreported mode of particle self-diffusion, a mode we termed *ballistic* self-diffusion. First, we used a simple scaling argument to show that measured self-diffusion coefficients are several orders of magnitude larger than those predicted by the Stokes-Einstein relation. We then estimated the self-diffusion coefficient using its fundamental definition, stated in terms of the velocity autocorrelation function and given below. This analysis led to a self-consistent explanation for the anomalously large self-diffusion coefficients measured here.

4.8.1 A model of ballistic self-diffusion

In order to explain the large self-diffusion coefficient measured in our experiment, we started with the Green-Kubo relationship, connecting D and the single particle velocity autocorrelation function:

$$D = \frac{1}{2}\int_0^{\infty} \langle \mathbf{v}\left(t\right) \cdot \mathbf{v}\left(0\right) \rangle dt \tag{4.18}$$

where the factor of 1/2 reflects the strongly two-dimensional grain dynamics that existed on planes parallel to the grain bed's free surface, observed and reported in [14].

The large D measured above, coupled with the fact that transport of heated spheres was driven, overwhelmingly, by the externally forced, rigid body-like vibrational motion of the entire bed, led us to the following picture of collision time scale single grain dynamics [14]:

a) Over a short (grain pair) impact time scale, τ_{impact}, much shorter than the collision time scale, $\tau_c \sim 1/f_o$, we assumed that a single grain, $\mathcal{A}$, traveling at characteristic speed $v_o \approx A_o f_o$, contacted a nominally fixed second grain, $\mathcal{B}$, elastically deformed $\mathcal{B}$, then elastically rebounded, leaving the surface of $\mathcal{B}$ with the same speed it had on impact, $v_o \approx A_o f_o$. A force balance on $\mathcal{A}$:

$$m\dot{v} = \frac{2mv_o}{\tau_{impact}} \tag{4.19}$$

applicable during the short impact time interval, $0 \leq t \leq \tau_{impact}$, yielded a differential equation describing the post-impact motion of $\mathcal{A}$, over the much longer (average/characteristic) time interval, $0 \leq t \leq \tau_c/2$, during which the grain underwent collision-free, "ballistic" motion.

During any given grain bed vibration cycle, we assumed that on average, grain $\mathcal{A}$ suffered no collisions until grain bed motion reversed itself half-way through the cycle. Thus, we integrated Equation 4.19 from the instant $\mathcal{A}$ and $\mathcal{B}$ separate ($t = 0$), to the cycle midpoint, $t = \tau_c/2$, to obtain the time-dependent velocity of $\mathcal{A}$ during its ballistic motion:

$$v\left(t\right) = v_o + \frac{2v_o}{\tau_{impact}}t \tag{4.20}$$

Next, we assumed that on time intervals ranging from $\tau_c/2$ to τ_c, the time-dependent velocity of grain $\mathcal{A}$, $v\left(t\right)$, observed over many repeat trials, remained only weakly correlated with $v_o = v\left(0\right)$. This assumption reflected the fact that, over the second half of any given vibration cycle, $\tau_c/2 \leq t \leq \tau_c$, $\mathcal{A}$ began colliding with other grains moving predominantly in the opposite direction, at a speed on the order of $A_o f_o$. Similarly, for $t > \tau_c$, due to numerous collisions between $\mathcal{A}$ and other grains, we assumed that the velocity of $\mathcal{A}$ becomes uncorrelated with v_o. Both of these features were in fact observed by our group in [14].

Based on these assumptions and using the ballistic grain velocity in Equation 4.20, we used Equation 4.18 to calculate a theoretical ballistic self diffusion coefficient:

$$D_{ballistic} = \frac{1}{2}v_o^2 \int_0^{\tau_c/2} \left(1 + \frac{2}{\tau_{impact}}t\right) dt \tag{4.21}$$

or

$$D_{ballistic} = \frac{1}{4}v_o^2 \left(\tau_c + \frac{\tau_c^2}{2\tau_{impact}}\right) \tag{4.22}$$

4.8.2 Scaling estimate of the elastic impact time, τ_{impact}, and estimated $D_{ballistic}$

In order to obtain a theoretical $D_{ballistic}$ via Equation 4.22, we used a simple physical analysis to estimate the elastic impact time, τ_{impact}, between colliding grains. Thus, imposing conservation of energy on the impact process, we first estimated the total collision-induced elastic deformation, $\delta r/2$, of the radii of spherical grains $\mathcal{A}$ and $\mathcal{B}$, achieved halfway through the impact interval, $t = \tau_{elastic}$:

$$\frac{m_g A_o f_o^2}{2} \approx E_g \frac{\delta r}{d_g} \pi \delta r^2 \tag{4.23}$$

Here:

a) m_g, E_g, and d_g were, respectively, single grain mass, Young's modulus, and diameter,

b) the relative speed of $\mathcal{A}$ and $\mathcal{B}$ prior to impact was approximated as the product of the imposed vibration amplitude and frequency, $A_o f_o$,

c) the total elastic deformation of δr was imposed on $\mathcal{B}$, while $\mathcal{A}$ remaining rigid, and

d) the circular (collision-induced) contact area on $\mathcal{B}$ was the (approximate) radius δr.

Using Equation 4.23, we solved for δr :

$$\delta r \approx \left[\frac{d_g m_g A_o f_o^2}{2\pi E_g} \right]^{1/3} \tag{4.24}$$

Next, applying conservation of linear momentum to the collision process and using the same assumptions, we obtained an expression that was used to estimate τ_{impact} :

$$\frac{2m_g A_o f_o}{\tau_{impact}} \approx E_g \frac{\delta r}{d_g} \pi \delta r^2 \tag{4.25}$$

Solving Equation 4.25 for τ_{impact} finally yielded:

$$\tau_{impact} \approx \frac{2m_g A_o f_o d_g}{\pi E_g \delta r^3} \tag{4.26}$$

Using Equation 4.24 to estimate δr, and then using δr in Equation 4.26 to obtain τ_{impact}, we finally used Equation 4.22 to obtain a rough theoretical estimate of $D_{ballistic}$. For these calculations, we used the following parameter values: $m_g = 0.01g$, $E_g(estimated) = 300GPa$, $d_g = 2\ \left(10^{-3}\right) m$, $f_o \approx 30$ Hz, and $A_o \approx 2\ \left(10^{-3}\right) m$; this resulted theoretical estimate of $D_{ballistic} = 33.75 \frac{mm^2}{s}$ [29].

4.8.3 Comparison of $D_{ballistic}$ with $D_{experimental}$

A comparison of $D_{ballistic}$ and $D_{experimantal}$ can be seen in Equation 4.27.

$$D_{ratio} = \frac{D_{ballistic}}{D_{experimental}} \approx 2.3 \tag{4.27}$$

$D_{ballistic}$ was approximately double the value of $D_{experimental}$ and on the same order of magnitude. Additionally, if $D_{ballistic}$ was substituted for $D_{experimental}$ in Equation 4.10, we approximated a $\tilde{m}_{ballistic} \approx 424 \frac{mm^2}{s}$; this value appeared within the span of the experimental linear fit data seen in Appendix C.1 at approximately the 25% filtration level, while $D_{experimental}$ more closely reflects the 37% filtration level fit. This close comparison between $D_{ballistic}$ and $D_{experiment}$ suggested that the experimental diffusion data is reflective of the externally forced, rigid body-like vibrational motion of the entire bed that we have termed "Ballistic Self-Diffusion."

CHAPTER 5: CONCLUSIONS

5.1 Summery of completed work

This project, inspired by previous research conducted at The University of North Carolina at Charlotte, endeavored to track a single or small number of grain particles in order to determine if a diffusion coefficient could be obtained. Unlike previous experiments of this type where the bulk-flow was analyzed, this experiment intended to isolate and track only the introduced particles. After potential methods of grain tracking were discussed and tested, it was determined that heating the introduced grains and tracking them using a thermal video camera would be used.

After developing the experimental setup and procedure, 100 single-grain and 100 multi-grain experiments were conducted and filmed using the thermal video camera. These videos were then exported to a computer where they were processed into sets of single-frame images in binary and grey-scale; the image files were then processed into PIV software. Analysis of the data exported from the PIV software showed a low resolution; to visually test how much this affected the resulting data, a statistical analysis of the pixel data was completed using a MATLAB script. Based upon this test, it was determined that further PIV analysis would not be usable until a thermal video camera with a greater resolution was available.

Using a statistical analysis of the data, multiple methods were attempted to determine if a Brownian distribution existed and, after several attempts, it was determined that a radial Brownian distribution was present. Using filtration and the convex hull function in MATLAB, the statistical area of the event was determined; however, the required use of filtration dictated that a data fit be used to estimate the rate that the event area grew over time with no filtration. Using the generalized form of means-squared displacement particles to relate to the diffusion coefficient and experimentally derived rate of event growth over time, an experimental diffusion coefficient, $D_{experiment}$, was calculated; however, $D_{experiment}$ was a greater value than expected.

A theoretical diffusion coefficient, $D_{theoretical}$, was calculated to test $D_{experiment}$'s validity. The results showed that $D_{theoretical}$ was approximately 4 orders-of-magnitude smaller than $D_{experiment}$. This large difference indicated that the event diffusion is not thermally driven but more reflective of super-diffusion we termed "Ballistic Self-Diffusion." In order to model $D_{ballistic}$, we used the Green-Kubo relation and modeled the grain impacts as elastic. The calculation of $D_{ballistic}$ showed that it is on the same order-of-magnitude as $D_{experiment}$; this demonstrated that this experiment is reflective of the externally forced, rigid body-like vibrational motion of the entire bed.

5.2 Future work

As this experiment has demonstrated, the use of thermal video cameras to track heated grains was successful, justifying the use of further funds and research time. The next step would be to acquire a new thermal video camera with, at minimum, twice the resolution and frame-rate of the TiX580 camera that was used in this experiment; however, the highest resolution and frame-rate achievable would be preferred. If the frame rate was, at minimum, doubled, it would allow the reintroduction of PIV analysis, as the loss in resolution was exactly 1/2 the pixel resolution. Additionally, a greater frame rate would decrease the time between event samples, improving the area calculations and linear fits. The reinclusion of PIV methodology into this experiment presents the potential to separate any thermally driven diffusion from the dominate bulk-flow, a goal of the experiment that was disregarded in this work due to the low image resolution of the current camera. Even if PIV methods were not reincluded, at minimum, the new camera would improve the accuracy of the method as it is currently outlined in this work.

REFERENCES

[1] J. Boon and S. Yip, *Molecular Hydrodynamics.* Dover books on physics, Dover Publications, 1991.

[2] J. Dahlberg, P. Tkacik, B. Mullany, E. Fleischhauer, H. Shahinian, F. Azimi, J. Navare, S. Owen, T. Bisel, T. Martin, J. Sholar, and R. Keanini, "An analog macroscopic technique for studying molecular hydrodynamic processes in dense gases and liquids," *Journal of Visualized Experiments*, vol. 130, 12 2017.

[3] M. v. Smoluchowski, "Molekular-kinetische theorie der opaleszenz von gasen im kritischen zustande, sowie einiger verwandter erscheinungen," *Annalen der Physik*, vol. 330, no. 2, pp. 205–226, 1908.

[4] A. Einstein, "Theorie der opaleszenz von homogenen flussigkeiten und flussigkeitsgemischen in der nahe des kritischen zustandes," *Annalen der Physik*, vol. 338, no. 16, pp. 1275–1298, 1910.

[5] A. Einstein, A. Beck, and P. Havas, *The collected papers of Albert Einstein.* Princeton, N.J: Princeton University Press, 1987.

[6] J. Stetefeld, S. A. McKenna, and T. R. Patel, "Dynamic light scattering: a practical guide and applications in biomedical sciences," *Biophysical reviews*, vol. 8, no. 4, pp. 409–427, 2016.

[7] A. Einstein, "Zur theorie der brownschen bewegung," *Annalen der Physik*, vol. 324, no. 2, pp. 371–381, 1906.

[8] G. Stokes, "On the theories of the internal friction of fluids in motion, and of the equilibrium and motion of elastic solids.," *Trans Cam Philos Soc*, vol. 8, pp. 287–305, 1845.

[9] R. Pecora, "Doppler shifts in light scattering from pure liquids + polymer solutions," *The Journal of chemical physics*, vol. 40, no. 6, pp. 1604–1614, 1964.

[10] B. Berne and R. Pecora, *Dynamic Light Scattering: With Applications to Chemistry, Biology, and Physics.* Dover Books on Physics Series, Dover Publications, 2000.

[11] G. Kostorz and S. Lovesey, "Neutron scattering general introduction," in *Neutron Scattering* (G. Kostorz, ed.), vol. 15 of *Treatise on Materials Science Technology*, pp. 1–67, Elsevier, 1979.

[12] W. A. Monteiro, *Neutron Scattering.* IntechOpen, 2016.

[13] W. Langel, "Introduction to neutron scattering," *ChemTexts (Cham)*, vol. 9, no. 4, 2023.

[14] R. Keanini, P. Tkacik, E. Fleischhauer, H. Shahinian, J. Sholar, F. Azimi, and B. Mullany, "Macroscopic liquid-state molecular hydrodynamics," *Sci. Rep.*, vol. 7, 01 2017.

[15] K. Vollmayr-Lee, "Introduction to molecular dynamics simulations," *American journal of physics*, vol. 88, no. 5, pp. 401–422, 2020.

[16] S. Alavi, *Molecular simulations : fundamentals and practice.* Weinheim: Wiley-VCH, 2020.

[17] S. Loukatou, L. Papageorgiou, P. Fakourelis, A. Filntisi, E. Polychronidou, I. Bassis, V. Megalooikonomou, W. MakaÅowski, D. Vlachakis, and S. Kossida, "Molecular dynamics simulations through gpu video games technologies," *J Mol Biochem*, vol. 3, pp. 64–71, 2014.

[18] P. K. Haff, "Grain flow as a fluid-mechanical phenomenon," *Journal of Fluid Mechanics*, vol. 134, p. 401â430, 1983.

[19] D. Dabiri and C. Pecora, *Particle Tracking Velocimetry.* IOP Ebooks Series, Bristol, England: IOP Publishing Ltd, first edition. ed., 2020.

[20] Dantec Dynamics, "DynamicStudio User's Guide." 2018.

[21] E. Fleischhauer, J. L. Dahlberg, J. M. Solomon, R. G. Keanini, and P. T. Tkacik, "Kinematic viscosity measurement of granular flows via low reynolds number cylinder drag experiment," *Measurement science technology*, vol. 30, no. 5, pp. 55904–, 2019.

[22] T. B. Morgan, B. R. Halls, T. R. Meyer, and T. J. Heindel, "A high-speed x-ray detector system for noninvasive fluid flow measurements," *Iowa State University Digital Repository*, 2013.

[23] W. Wang, T. Tsai, F. Tian, J. Li, Y. Zhao, R. Zhu, J. Li, Y. Liu, and S. Wang, "High-speed fluoroscopic imaging for investigation of three-dimensional knee kinematics before and after marathon running," *Gait posture*, vol. 88, pp. 231–237, 2021.

[24] Fluke Corporation, "Fluke TiX580 Infrared Camera Specifications." Accessed Apr. 29, 2024 [Online].

[25] K. Dill and S. Bromberg, *Molecular Driving Forces: Statistical Thermodynamics in Chemistry and Biology.* Garland Science, 2003.

[26] R. Pathria and P. D. Beale, *Statistical Mechanics.* Elsevier Ltd, third edition ed., 2011.

[27] Rosler Metal Finishing USA, LLC, "Consumables." 2024.

[28] Weisstein, Eric W., "Sphere Packing. From MathWorld-A Wolfram Web Resource." Accessed Apr. 29, 2024 [Online].

[29] J. A. Navare, “Experimental and computational evaluation of a vibratory finishing process,” 2017.

APPENDIX A: MATLAB PROGRAMS

A.1 MATLAB program to trim video length and convert images to binary

```
%Video length editing for vibrating grain data
%and bianary color conversion

%we will take the videos of varying length and edit them all to save only
%the first 20 sec of data and convert to binary.tif files

%By: Philip Brown
%Date: 1/10/2024


clear all; close all; clc;

%Variable inputs
File1=1;
FileEnd=100;
Frame1=1;
FrameEnd=480;
%

for i=[File1:FileEnd]

    filenamei= [ 'D:\thesis_2024\Dec_2023_vids\Renamed_2023_vids\multigrain\' num2str(i)  '.AVI' ];
    obj=VideoReader(filenamei);
    vid=read(obj);

    destdir= strcat('D:\thesis_2024\bi_pics_20sec\multigrain\' , num2str(i));
```

```
     mkdir(destdir);

    for x=[Frame1:FrameEnd]

      imwrite((imbinarize((rgb2gray(vid(:,:,:,x))),.5)),strcat(destdir , '\' ,num2str(i), '_frame_' , num2str(x), '.tif'));

    end

end
```

A.2 MATLAB program to identify the first and last grain appearance of each run; then, to set frame one to the first appearance in each video and include enough frames to capture the longest run

```matlab
%Find the first and last even occurance in a series of bianary images
%Then; trim to required length to contain the entire event in all runs
%making frame 1 the first appearance of the event in each run
%Author: Philip Brown, %1/11/2024 , Last Edited: 4/16/2024
clear all; close all; clc;
index_len=0;
count2=0;
i_start=4;
i_end=104;
for i=[i_start:i_end]
Events=0;
count=0;
for x=[1:438]
filenamei= [ 'D:\thesis_2024\bi_pics_20sec\multigrain\' , num2str(i) , '\' , num2str(i) ,  '_frame_' , num2str(x) , '.tif' ];
pic=imread(filenamei);
val=sum(pic(:)==0);
if val~0;
    count=count+1;
    Events(count,:)=x;
end
end
count2=count2+1;
index_len(count2, 1)=i; %Puts run Number in Columb 1
index_len(count2,2)=Events(1,1); %Puts first event occurance in Columb 2
```

```
index_len(count2,3)=Events(length(Events),1); %Puts last event occurance in columb 3
index_len(count2,4)=(index_len(count2,3)-index_len(count2,2))+1; % Frame length of Event this run
end
Req_len=max(index_len(:,4))+0; %If nessisary, add 1 to make an even number for PIV program use
count3=0;
for i=[i_start:i_end]
     count3=count3+1;
     count4=0;
     destdir=strcat('D:\thesis_2024\trim2event_pics\multigrain\',num2str(i));
     mkdir(destdir);
     for x=[index_len(count3,2):index_len(count3,2)+Req_len]
     count4=count4+1;
     filenamei= [ 'D:\thesis_2024\grey_pics_20sec\multigrain\' , num2str(i) , '\' , num2str(i) ,  '_frame_' , num2str(x) , '.tif' ];
     pic=imread(filenamei);
     imwrite(pic, strcat(destdir, '\' , num2str(i), '_frame_' , num2str(count4, '%03i') , '.tif'));
     end
end
```

A.3 MATLAB program to determine the probability of a particle being located at a specific pixel grid location at a specific time

```
%Determines the probibility of a grain being at a pixel grid location
%at a specific time interval
%Author: Philip Brown, 1/11/2024, Last Edited: 4/16/2024
clear all;close all;clc;

file1=2; % 1-multi
fileEnd=101; %101-multi
Num_Files=(fileEnd-file1)+1;

frame1=1; % 1-multi
frameEnd=212; % 212-multi


for i=[frame1:frameEnd]
    trimsum_avgsum=0;
    trimsum=0;
    for j=[file1:fileEnd]
        readinfile=['D:\thesis_2024\trim2event_pics\multigrain_bi\' num2str(j) '\' num2str(j) '_frame_' num2str(i,'%03i') '.tif'];
        hold=imread(readinfile);
        hold=~hold;
        for y=[1:height(hold)]
            for x=[1:width(hold)]
                if (hold(y,x)~=0)
                hold(y,x)=1;
                end
            end
        end
        trimsum=trimsum+hold;
```

```
    end
    trimsum_avg=(1/Num_Files)*trimsum;
    filenamei=['D:\thesis_2024\bi_pic_heatmapping\
   multigrain_timestep_v3\frame_' num2str(i) '.csv'];
    writematrix(trimsum_avg,filenamei);
end
```

A.4 MATLAB program to determine dispersion event area at each time interval at varying location probability percentages

```
%Determines the area of the particle dispersion at varying
    filtrations and
%times
%Author: Philip Brown, 3/23/2024, Last Edited: 5/3/2024

clear all; close all;clc;

frame1=20; % used 20 for excel slope workbook
frameEnd=70; %used 70 for excel slope workbook
step=1;
areaV=[0 0];
set(gca,'YDir','reverse');
count3=1;

for Percent=[.1:.01:.9]
    count2=0;
    count3=count3+1;
for i=[frame1:step:frameEnd]

    readinfile=[ 'E:\thesis_2024\bi_pic_heatmapping\
    multigrain_timestep_v3\frame_' num2str(i) '.csv'];
    Tdata=readmatrix(readinfile);
    TdataHold=Tdata;
    M=max(Tdata,[],"all");
    threshold=Percent*M;
    V=0;
    count1=0;
    count2=count2+1;

for wid=[1:width(Tdata)]
```

```
    for high=[1:height(Tdata)]
        if Tdata(high,wid)<threshold
            Tdata(high,wid)=0;
        else
            Tdata(high,wid)=1;
        end

        if Tdata(high,wid)==1
            count1=count1+1;
            V(count1,1)=wid;
            V(count1,2)=high;
        end
    end

end

    [k,av] = convhull(V);
    areaV(count2,1)=i;
    mm2=av*((0.353)^2); %converts pix^2 to mm^2
    areaV(count2,count3)=mm2;
    colormap('turbo');
    imagesc(TdataHold);
    hold on
    plot(V(:,1),V(:,2),'.w')
    hold on
    plot(V(k,1),V(k,2))
    set(gca,'YDir','reverse');
    xlim([0 640]);
    ylim([0 480]);
    xlabel('X (mm)');
    ylabel('Y (mm)');
    %scale plots x axis only
    xt=get(gca, 'XTick');
```

```
    set(gca,'XTickLabel',round((0.353)*xt));
    yt=get(gca,'YTick');
    set(gca,'YTickLabel',round((0.353)*yt));
    title(['Filtered: ' num2str(100*Percent) '%, Frame: ' num2str(i) ', Time: ' num2str(round(((1/24)*(i-1)),2)) ' seconds, Area: ' num2str(av) ' mm^2']);
    cb=colorbar;
    tix=cb.Ticks;
    cb.TickLabels=100*tix;
    ylabel(cb,'%');

end
end
```

APPENDIX B: FIGURES

B.1 Select time instants of the statistically averaged particle spread for visualization purposes

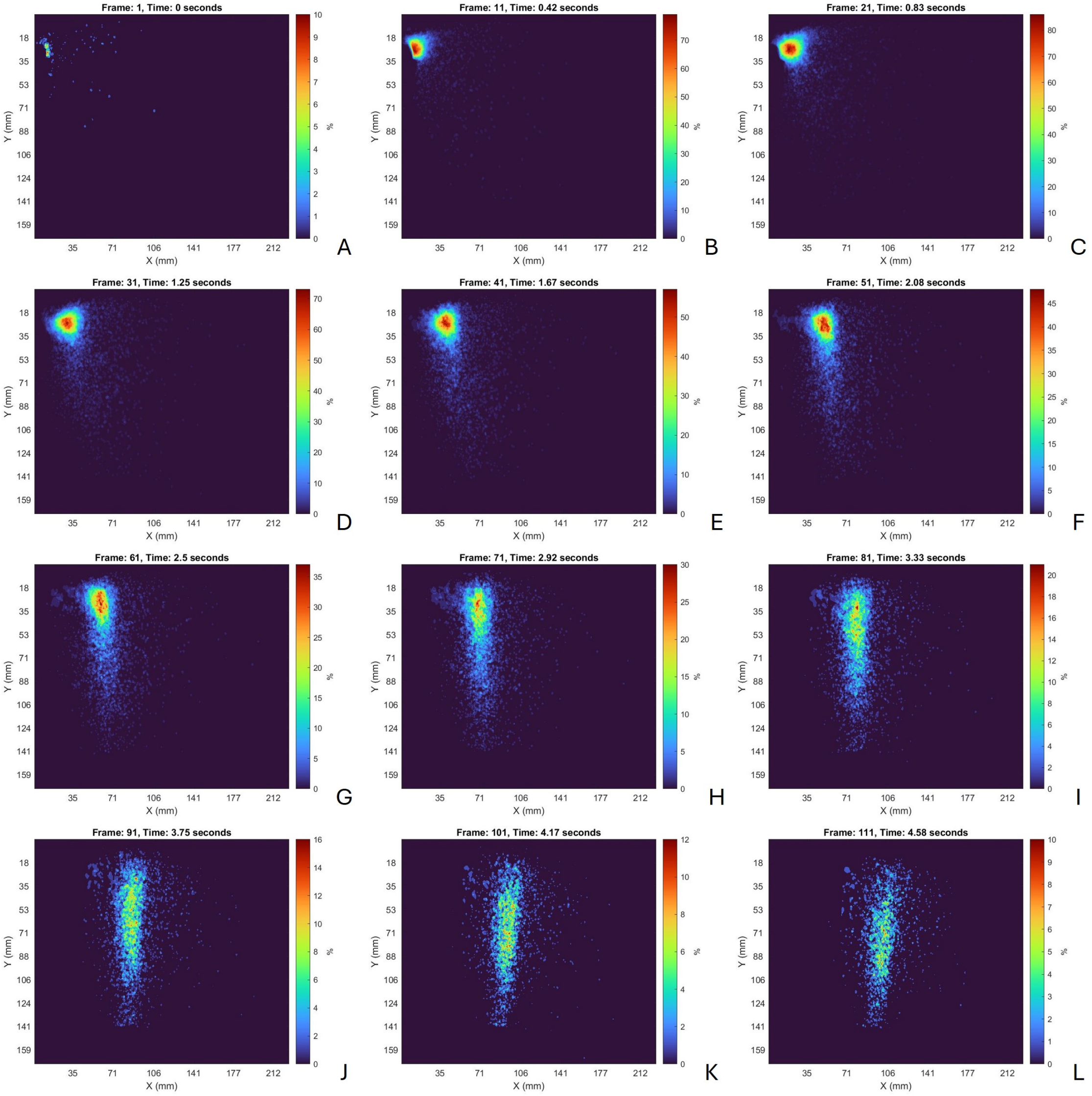


Figure B.1: Statistically averaged particle location at frame and time: A. Frame 1 - 0 seconds, B. Frame 11 - 0.42 seconds, C. Frame 21 - 0.83 seconds, D. Frame 31 - 1.25 seconds, E. Frame 41 - 1.67 seconds, F. Frame 51 - 2.08 seconds, G. Frame 61 - 2.5 seconds, H Frame 71 - 2.92 seconds, I Frame 81 - 3.33 seconds, J. Frame 91 - 3.75 seconds, K. Frame 101 - 4.17 seconds, L. Frame 111 - 4.58 seconds

B.2 Select time instants of percentage based area analysis

B.2.1 10 percent filter

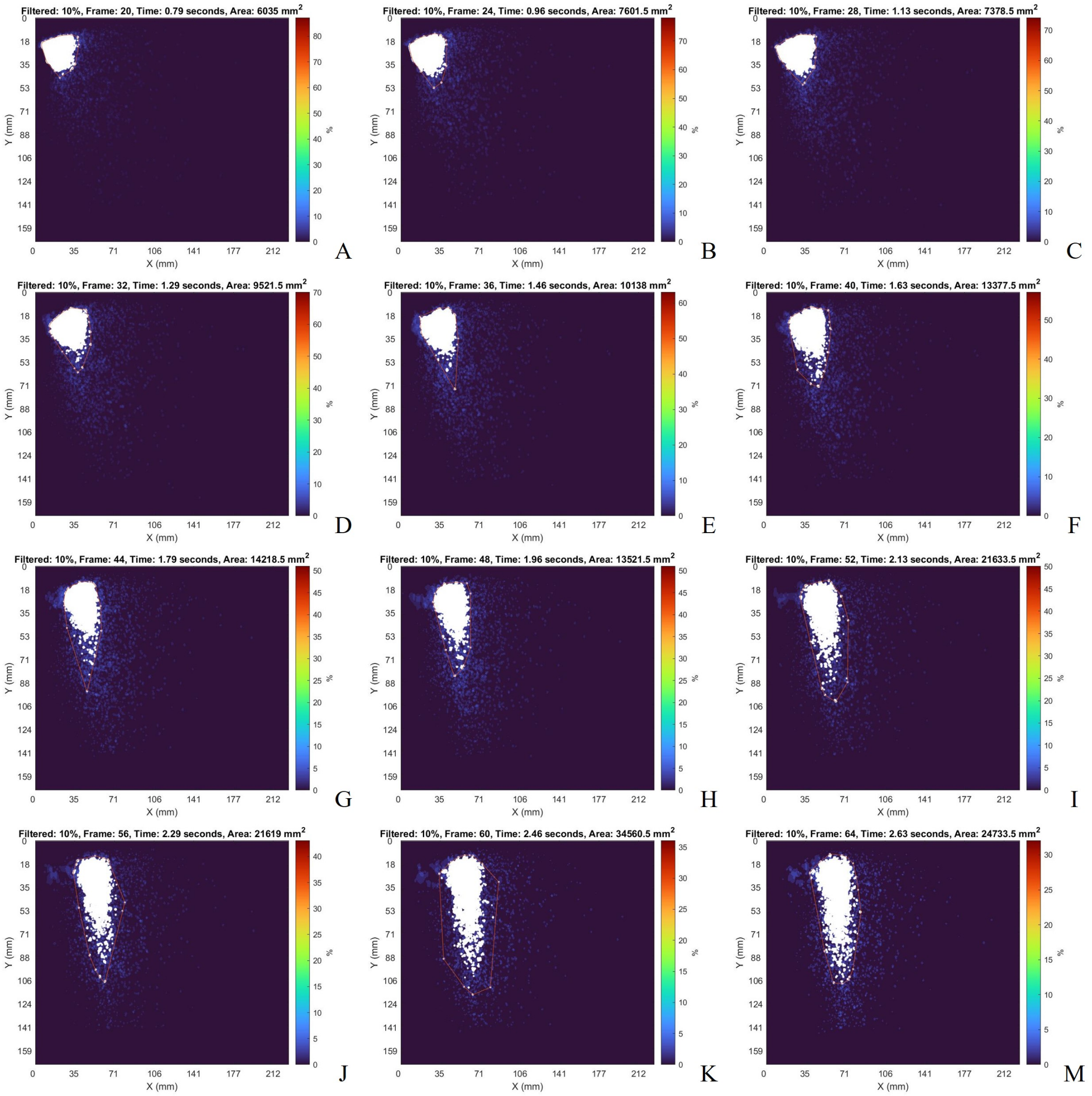
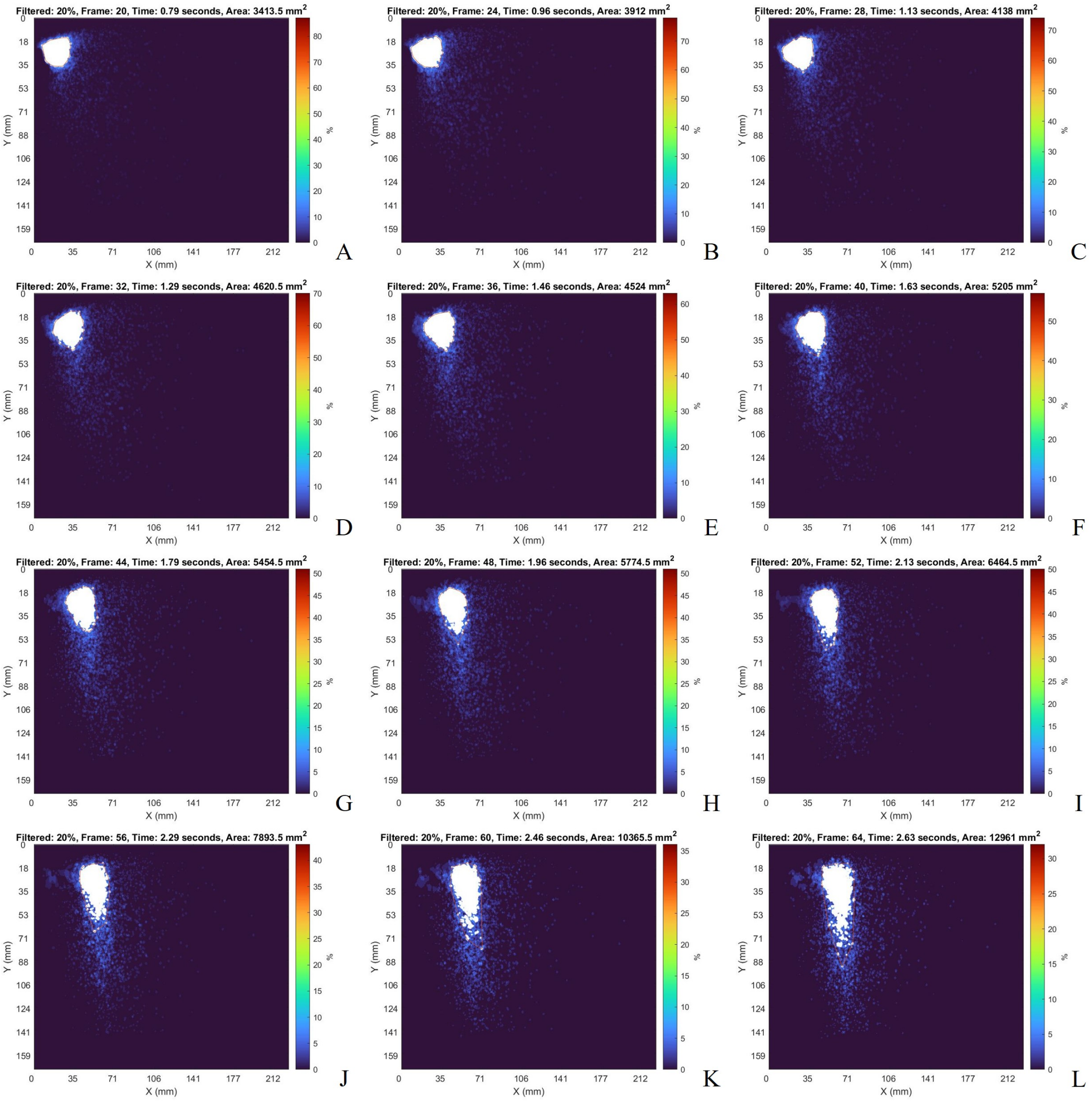


Figure B.2: Area calculated from convex hull analysis of the statistically averaged particle location at frame and time with the lowest 10 percent filtered (above 10 percent appears white): A. Frame 20 - 0.79 seconds, B. Frame 24 - 0.96 seconds, C. Frame 28 - 1.13 seconds, D. Frame 32 - 1.29 seconds, E. Frame 36 - 1.46 seconds, F. Frame 40 - 1.63 seconds, G. Frame 44 - 1.79 seconds, H Frame 48 - 1.96 seconds, I Frame 52 - 2.13 seconds, J. Frame 56 - 2.29 seconds, K. Frame 60 - 2.46 seconds, L. Frame 64 - 2.63 seconds

B.2.2 20 percent filter


Filtered: 20%, Frame: 20, Time: 0.79 seconds, Area: 3413.5 mm^2 A

Filtered: 20%, Frame: 24, Time: 0.96 seconds, Area: 3912 mm^2 B

Filtered: 20%, Frame: 28, Time: 1.13 seconds, Area: 4138 mm^2 C

Filtered: 20%, Frame: 32, Time: 1.29 seconds, Area: 4620.5 mm^2 D

Filtered: 20%, Frame: 36, Time: 1.46 seconds, Area: 4524 mm^2 E

Filtered: 20%, Frame: 40, Time: 1.63 seconds, Area: 5205 mm^2 F

Filtered: 20%, Frame: 44, Time: 1.79 seconds, Area: 5454.5 mm^2 G

Filtered: 20%, Frame: 48, Time: 1.96 seconds, Area: 5774.5 mm^2 H

Filtered: 20%, Frame: 52, Time: 2.13 seconds, Area: 6464.5 mm^2 I

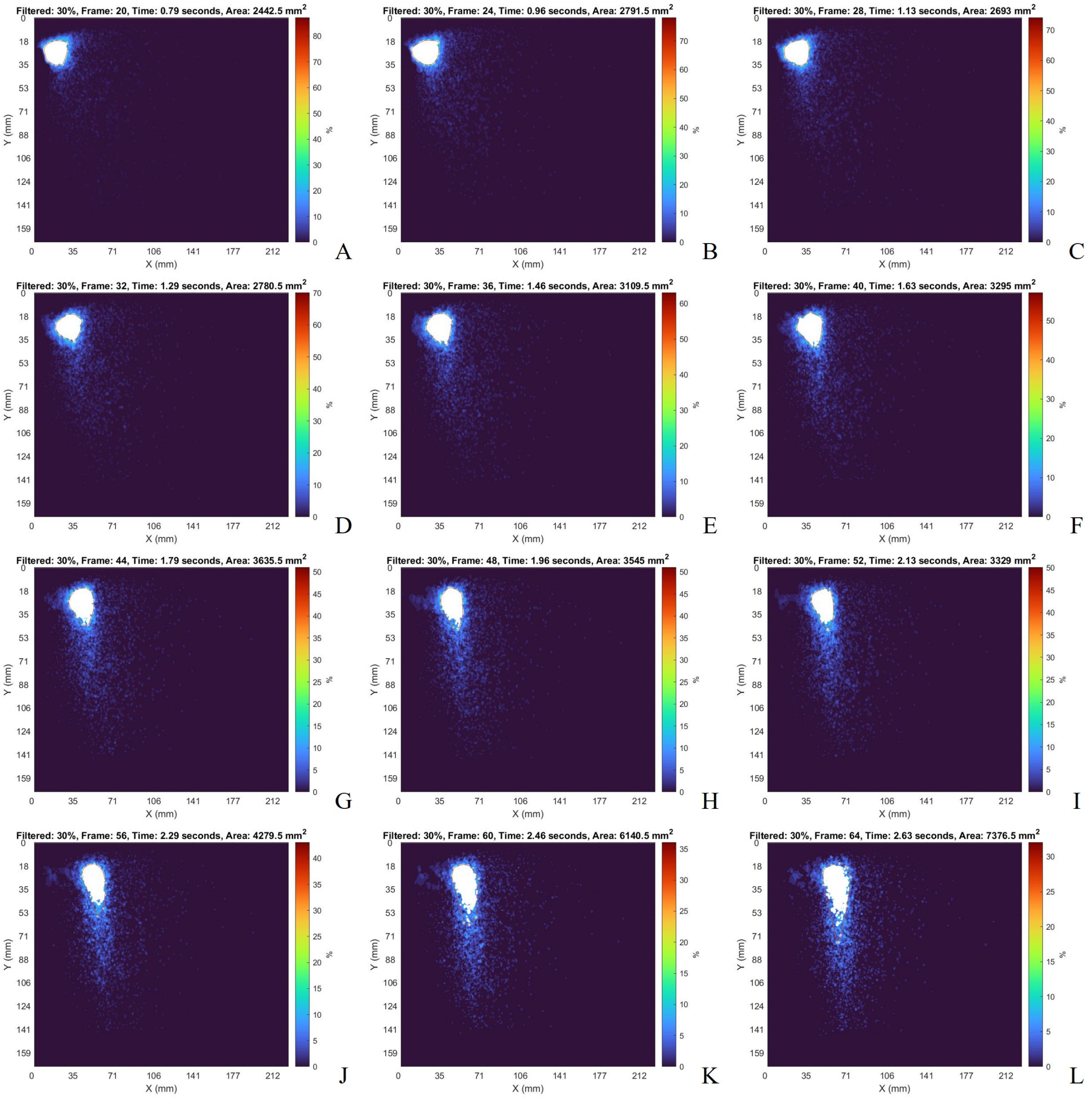
Filtered: 20%, Frame: 56, Time: 2.29 seconds, Area: 7893.5 mm^2 J

Filtered: 20%, Frame: 60, Time: 2.46 seconds, Area: 10365.5 mm^2 K

Filtered: 20%, Frame: 64, Time: 2.63 seconds, Area: 12961 mm^2 L

Figure B.3: Area calculated from convex hull analysis of the statistically averaged particle location at frame and time with the lowest 20 percent filtered (above 20 percent appears white): A. Frame 20 - 0.79 seconds, B. Frame 24 - 0.96 seconds, C. Frame 28 - 1.13 seconds, D. Frame 32 - 1.29 seconds, E. Frame 36 - 1.46 seconds, F. Frame 40 - 1.63 seconds, G. Frame 44 - 1.79 seconds, H Frame 48 - 1.96 seconds, I Frame 52 - 2.13 seconds, J. Frame 56 - 2.29 seconds, K. Frame 60 - 2.46 seconds, L. Frame 64 - 2.63 seconds

B.2.3 30 percent filter



Filtered: 30%, Frame: 20, Time: 0.79 seconds, Area: 2442.5 mm^2

Filtered: 30%, Frame: 24, Time: 0.96 seconds, Area: 2791.5 mm^2

Filtered: 30%, Frame: 28, Time: 1.13 seconds, Area: 2693 mm^2

Filtered: 30%, Frame: 32, Time: 1.29 seconds, Area: 2780.5 mm^2

Filtered: 30%, Frame: 36, Time: 1.46 seconds, Area: 3109.5 mm^2

Filtered: 30%, Frame: 40, Time: 1.63 seconds, Area: 3295 mm^2

Filtered: 30%, Frame: 44, Time: 1.79 seconds, Area: 3635.5 mm^2

Filtered: 30%, Frame: 48, Time: 1.96 seconds, Area: 3545 mm^2

Filtered: 30%, Frame: 52, Time: 2.13 seconds, Area: 3329 mm^2

Filtered: 30%, Frame: 56, Time: 2.29 seconds, Area: 4279.5 mm^2

Filtered: 30%, Frame: 60, Time: 2.46 seconds, Area: 6140.5 mm^2

Filtered: 30%, Frame: 64, Time: 2.63 seconds, Area: 7376.5 mm^2

Figure B.4: Area calculated from convex hull analysis of the statistically averaged particle location at frame and time with the lowest 30 percent filtered (above 30 percent appears white): A. Frame 20 - 0.79 seconds, B. Frame 24 - 0.96 seconds, C. Frame 28 - 1.13 seconds, D. Frame 32 - 1.29 seconds, E. Frame 36 - 1.46 seconds, F. Frame 40 - 1.63 seconds, G. Frame 44 - 1.79 seconds, H Frame 48 - 1.96 seconds, I Frame 52 - 2.13 seconds, J. Frame 56 - 2.29 seconds, K. Frame 60 - 2.46 seconds, L. Frame 64 - 2.63 seconds

B.2.4 40 percent filter

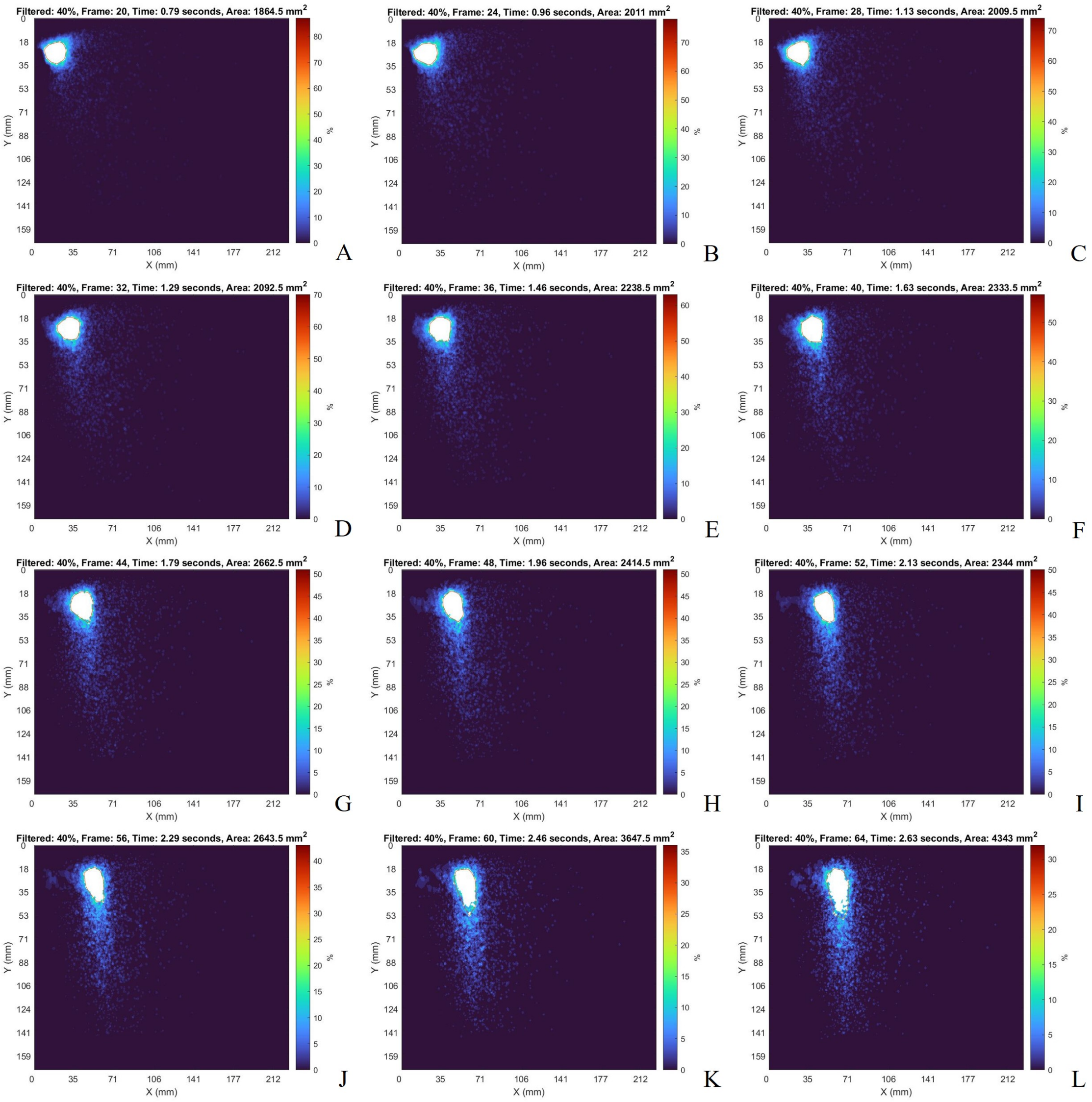


Figure B.5: Area calculated from convex hull analysis of the statistically averaged particle location at frame and time with the lowest 40 percent filtered (above 40 percent appears white): A. Frame 20 - 0.79 seconds, B. Frame 24 - 0.96 seconds, C. Frame 28 - 1.13 seconds, D. Frame 32 - 1.29 seconds, E. Frame 36 - 1.46 seconds, F. Frame 40 - 1.63 seconds, G. Frame 44 - 1.79 seconds, H Frame 48 - 1.96 seconds, I Frame 52 - 2.13 seconds, J. Frame 56 - 2.29 seconds, K. Frame 60 - 2.46 seconds, L. Frame 64 - 2.63 seconds

B.2.5 50 percent filter

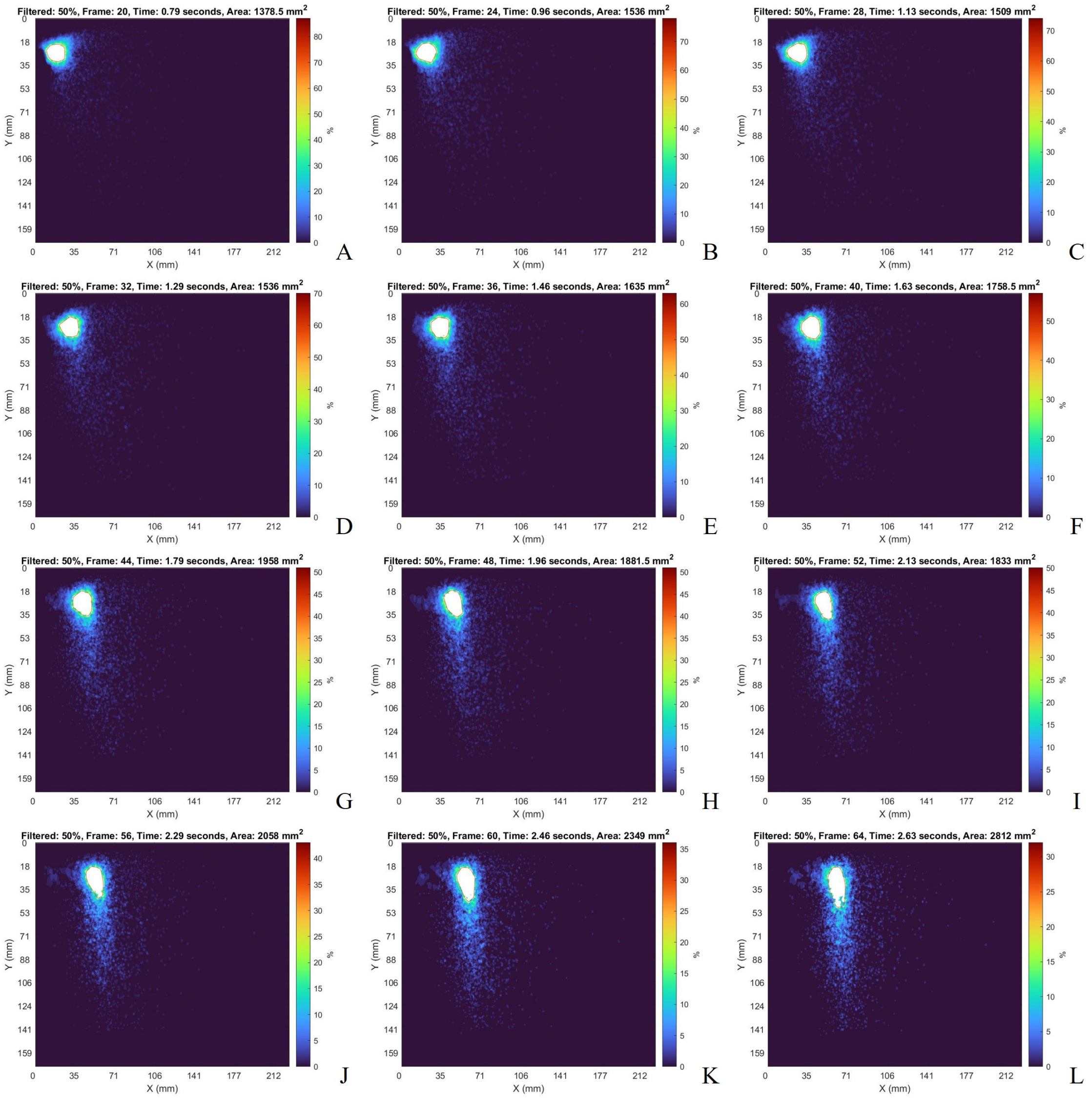
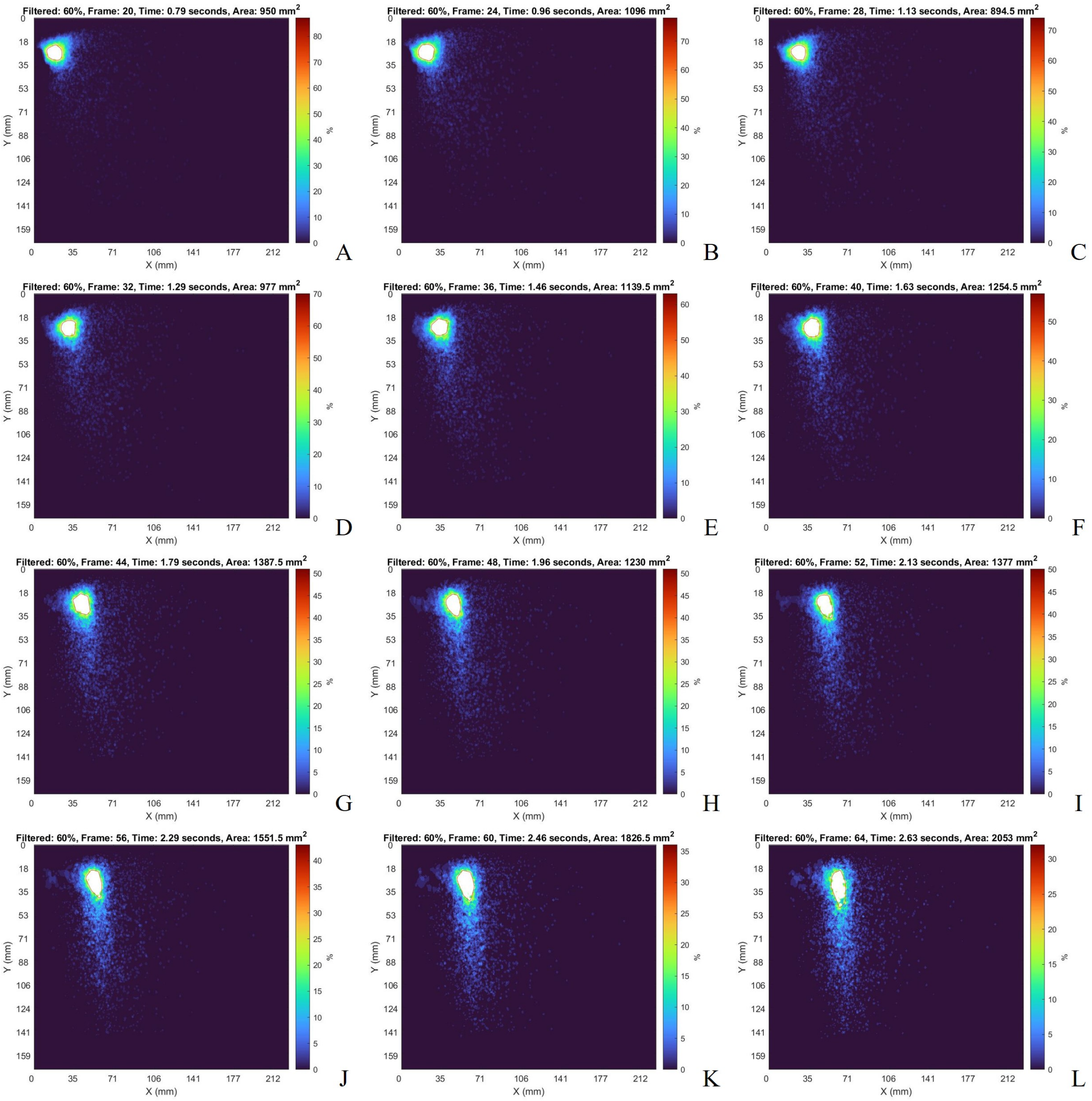


Figure B.6: Area calculated from convex hull analysis of the statistically averaged particle location at frame and time with the lowest 50 percent filtered (above 50 percent appears white): A. Frame 20 - 0.79 seconds, B. Frame 24 - 0.96 seconds, C. Frame 28 - 1.13 seconds, D. Frame 32 - 1.29 seconds, E. Frame 36 - 1.46 seconds, F. Frame 40 - 1.63 seconds, G. Frame 44 - 1.79 seconds, H Frame 48 - 1.96 seconds, I Frame 52 - 2.13 seconds, J. Frame 56 - 2.29 seconds, K. Frame 60 - 2.46 seconds, L. Frame 64 - 2.63 seconds

B.2.6 60 percent filter



Filtered: 60%, Frame: 20, Time: 0.79 seconds, Area: 950 mm^2

Filtered: 60%, Frame: 24, Time: 0.96 seconds, Area: 1096 mm^2

Filtered: 60%, Frame: 28, Time: 1.13 seconds, Area: 894.5 mm^2

Filtered: 60%, Frame: 32, Time: 1.29 seconds, Area: 977 mm^2

Filtered: 60%, Frame: 36, Time: 1.46 seconds, Area: 1139.5 mm^2

Filtered: 60%, Frame: 40, Time: 1.63 seconds, Area: 1254.5 mm^2

Filtered: 60%, Frame: 44, Time: 1.79 seconds, Area: 1387.5 mm^2

Filtered: 60%, Frame: 48, Time: 1.96 seconds, Area: 1230 mm^2

Filtered: 60%, Frame: 52, Time: 2.13 seconds, Area: 1377 mm^2

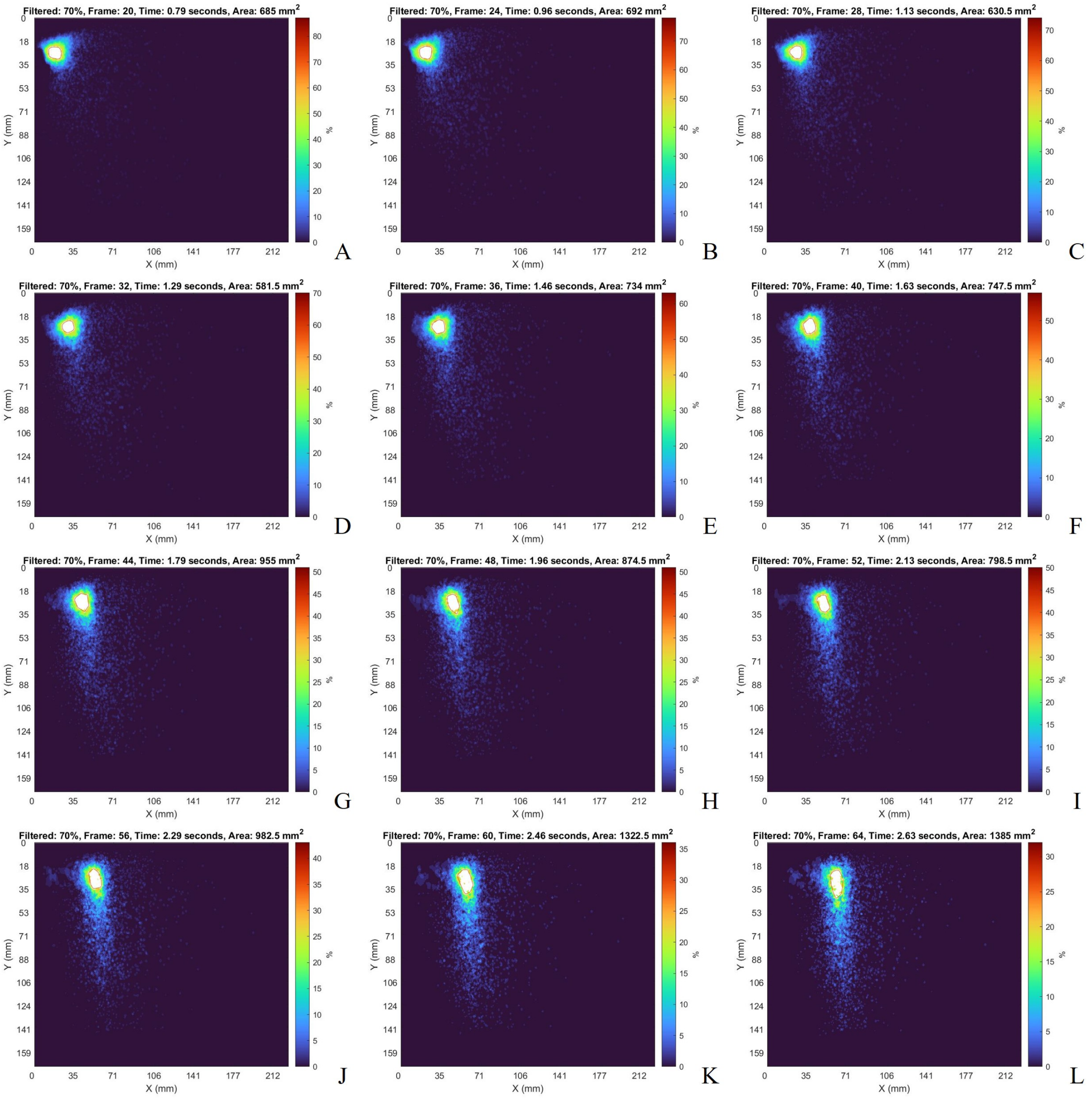
Filtered: 60%, Frame: 56, Time: 2.29 seconds, Area: 1551.5 mm^2

Filtered: 60%, Frame: 60, Time: 2.46 seconds, Area: 1826.5 mm^2

Filtered: 60%, Frame: 64, Time: 2.63 seconds, Area: 2053 mm^2

Figure B.7: Area calculated from convex hull analysis of the statistically averaged particle location at frame and time with the lowest 60 percent filtered (above 60 percent appears white): A. Frame 20 - 0.79 seconds, B. Frame 24 - 0.96 seconds, C. Frame 28 - 1.13 seconds, D. Frame 32 - 1.29 seconds, E. Frame 36 - 1.46 seconds, F. Frame 40 - 1.63 seconds, G. Frame 44 - 1.79 seconds, H Frame 48 - 1.96 seconds, I Frame 52 - 2.13 seconds, J. Frame 56 - 2.29 seconds, K. Frame 60 - 2.46 seconds, L. Frame 64 - 2.63 seconds

B.2.7 70 percent filter



Filtered: 70%, Frame: 20, Time: 0.79 seconds, Area: 685 mm^2

Filtered: 70%, Frame: 24, Time: 0.96 seconds, Area: 692 mm^2

Filtered: 70%, Frame: 28, Time: 1.13 seconds, Area: 630.5 mm^2

Filtered: 70%, Frame: 32, Time: 1.29 seconds, Area: 581.5 mm^2

Filtered: 70%, Frame: 36, Time: 1.46 seconds, Area: 734 mm^2

Filtered: 70%, Frame: 40, Time: 1.63 seconds, Area: 747.5 mm^2

Filtered: 70%, Frame: 44, Time: 1.79 seconds, Area: 955 mm^2

Filtered: 70%, Frame: 48, Time: 1.96 seconds, Area: 874.5 mm^2

Filtered: 70%, Frame: 52, Time: 2.13 seconds, Area: 798.5 mm^2

Filtered: 70%, Frame: 56, Time: 2.29 seconds, Area: 982.5 mm^2

Filtered: 70%, Frame: 60, Time: 2.46 seconds, Area: 1322.5 mm^2

Filtered: 70%, Frame: 64, Time: 2.63 seconds, Area: 1385 mm^2

Figure B.8: Area calculated from convex hull analysis of the statistically averaged particle location at frame and time with the lowest 70 percent filtered (above 70 percent appears white): A. Frame 20 - 0.79 seconds, B. Frame 24 - 0.96 seconds, C. Frame 28 - 1.13 seconds, D. Frame 32 - 1.29 seconds, E. Frame 36 - 1.46 seconds, F. Frame 40 - 1.63 seconds, G. Frame 44 - 1.79 seconds, H Frame 48 - 1.96 seconds, I Frame 52 - 2.13 seconds, J. Frame 56 - 2.29 seconds, K. Frame 60 - 2.46 seconds, L. Frame 64 - 2.63 seconds

B.2.8 80 percent filter

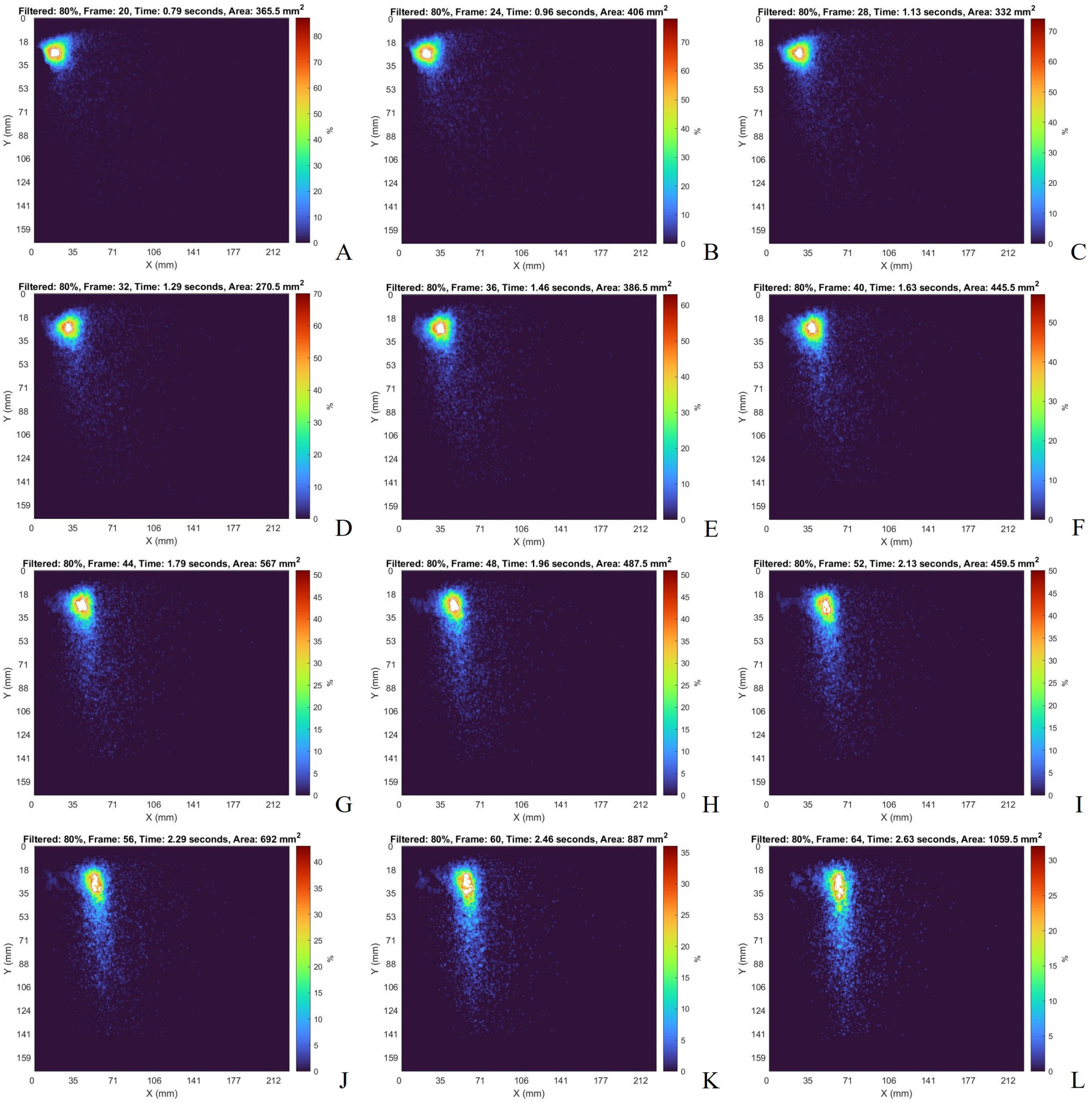


Figure B.9: Area calculated from convex hull analysis of the statistically averaged particle location at frame and time with the lowest 80 percent filtered (above 80 percent appears white): A. Frame 20 - 0.79 seconds, B. Frame 24 - 0.96 seconds, C. Frame 28 - 1.13 seconds, D. Frame 32 - 1.29 seconds, E. Frame 36 - 1.46 seconds, F. Frame 40 - 1.63 seconds, G. Frame 44 - 1.79 seconds, H Frame 48 - 1.96 seconds, I Frame 52 - 2.13 seconds, J. Frame 56 - 2.29 seconds, K. Frame 60 - 2.46 seconds, L. Frame 64 - 2.63 seconds

B.2.9 90 percent filter

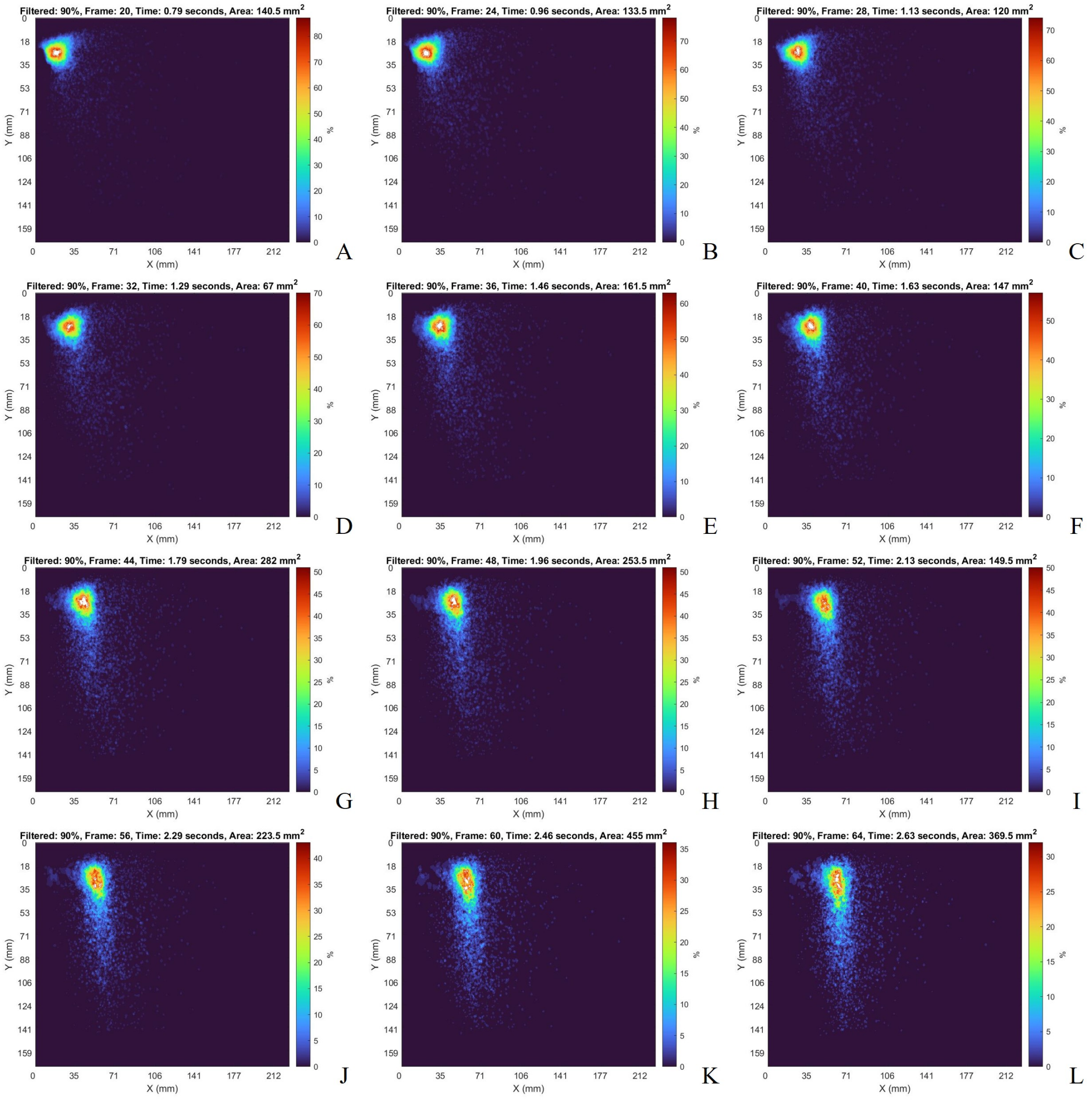


Figure B.10: Area calculated from convex hull analysis of the statistically averaged particle location at frame and time with the lowest 90 percent filtered (above 90 percent appears white): A. Frame 20 - 0.79 seconds, B. Frame 24 - 0.96 seconds, C. Frame 28 - 1.13 seconds, D. Frame 32 - 1.29 seconds, E. Frame 36 - 1.46 seconds, F. Frame 40 - 1.63 seconds, G. Frame 44 - 1.79 seconds, H Frame 48 - 1.96 seconds, I Frame 52 - 2.13 seconds, J. Frame 56 - 2.29 seconds, K. Frame 60 - 2.46 seconds, L. Frame 64 - 2.63 seconds

APPENDIX C: CALCULATED DATA TABLES

C.1 Time dependent linear fit of area flux at varying low end filtration levels

Note: The values in Table C.1 are reported as displayed in MATLAB to allow for the accurate reproduction of Figures 4.4, 4.5, 4.6, and 4.7; as the pixel measurements used for area calculation were accurate to ±0.5 pixels in the x and y directions, the values are precise to 1 decimal point.

Table C.1: Time dependent linear fit of area flux at varying low end filtration levels

Low End	Linear Fit - $mm^2(t)$		R^2
% Filtered	Slope	Intercept	
10	1876.10394721195	-1338.52660073808	0.747187200777043
11	1557.51118893611	-989.920171019909	0.863423671217023
12	1329.67442436489	-842.764196782202	0.859944803234521
13	1183.89691510588	-741.984290499999	0.877934754182034
14	1030.17397514824	-581.889701442157	0.904903851180303
15	951.711172389502	-547.824072347963	0.891546009937892
16	878.807667662986	-503.244150093589	0.862160778647703
17	790.921312098462	-416.345188351282	0.866339381011872
18	709.758970942262	-361.094497487330	0.839502286149033
19	659.680361751312	-315.315725764102	0.827738313361054
20	601.189460992941	-272.809537842157	0.796965204001926
21	545.686199150769	-215.651323760633	0.834482969468947
22	522.824225389683	-212.813931723303	0.804240988976333
23	472.695682384073	-167.420423587783	0.774565623296952
24	451.758799266244	-167.019785918326	0.755884210819413
25	438.112117766878	-159.887828332504	0.767298601659207

Table C.1: Time dependent linear fit of area flux at varying low end filtration levels

Low End % Filtered	Linear Fit - $mm^2(t)$		R^2
	Slope	Intercept	
26	397.083302042715	-124.187274653620	0.757346538297199
27	377.928880189683	-114.003270522323	0.744961875230886
28	364.268531169231	-109.547991314857	0.724790210556525
29	351.127683272036	-107.340964821342	0.697970094278112
30	296.842860636380	-43.8522000559578	0.708353317544213
31	295.495459572489	-52.0165422670437	0.690138609627599
32	267.291108930679	-25.2853932744343	0.723002744984504
33	241.048250547692	0.247921164253463	0.695380007326182
34	226.809535337919	8.88820209434391	0.693595826660961
35	201.867527253937	32.9438661733786	0.714103463262417
36	190.759080338824	36.1766803450980	0.722526041765801
37	188.593387195656	30.0658399493213	0.723353053653515
38	178.032588377919	34.1104075168929	0.723913741652814
39	160.640058847059	51.3590653382353	0.713994690197195
40	162.506284424253	38.1868133417798	0.690527997559229
41	149.587364592217	49.7363197327301	0.710337148312981
42	134.637667641991	61.9025654947964	0.717124184192034
43	132.100650955656	57.8989704189291	0.677196068085232
44	120.149824646878	67.1458603655355	0.659734397025873
45	108.621202949864	76.3182292435144	0.682716728610115
46	99.0248439616290	83.4398779248869	0.691791450418490
47	92.5741799750226	87.5073656546757	0.728332924023440
48	90.0922393390046	83.1867130334842	0.710126976289243

Table C.1: Time dependent linear fit of area flux at varying low end filtration levels

Low End	Linear Fit - $mm^2(t)$		R^2
% Filtered	Slope	Intercept	
49	81.9124309520362	90.0443334747361	0.706530917936498
50	85.4639566045249	79.2963373567119	0.722041384538694
51	78.1864752532127	80.7897940766969	0.834765691230540
52	73.7029870490498	83.1283648222474	0.831988396029756
53	71.6504230903168	79.2194011664405	0.788402550719859
54	70.3062697542082	74.8959669265460	0.824140891675796
55	66.6306538360181	73.2593602143288	0.781553859487518
56	67.0938610110407	66.0137979376320	0.785165925759689
57	62.6811462081448	66.9100523989442	0.785368955670166
58	62.9320331793665	60.6428775141780	0.779333974128394
59	60.4591591167421	57.3949429894419	0.773719416809532
60	59.9469653809955	53.2023575969080	0.763480509450878
61	60.0548170077828	46.4850397476621	0.774217196971288
62	57.4122491685068	45.8373360737557	0.716418051927422
63	53.4493446635294	45.6666330205882	0.754277731395690
64	54.7401811352036	39.4200853420815	0.768829147683924
65	52.5591966070588	36.9137059303922	0.708455291201153
66	47.8545922696833	39.2075264453243	0.676403003025785
67	46.8552393665158	36.1897990603318	0.704486750467449
68	43.2010043228959	37.6515396886878	0.642926946106799
69	41.0749605230769	35.5112626466818	0.627316219308478
70	41.2935055384615	31.9288478702866	0.624847651908137
71	37.2059694798190	33.6276275194570	0.568024875782857

Table C.1: Time dependent linear fit of area flux at varying low end filtration levels

Low End	Linear Fit - $mm^2(t)$		R^2
% Filtered	Slope	Intercept	
72	35.2654521295928	32.4026815805430	0.523811883982374
73	33.8387523790045	30.3010652795626	0.505179718330852
74	32.0317978338462	29.3314550223228	0.487112452166637
75	30.7837231113122	26.8423855486425	0.451155426675655
76	26.8439586667873	28.3202896370287	0.348225662050140
77	24.9702903800905	27.0726208804676	0.319088587226705
78	22.1143197610860	27.0840386734540	0.293710383117160
79	21.6248601198190	23.9813027655355	0.265142499415512
80	20.7053471913122	21.8486187907994	0.279012549100947
81	16.9787148825339	23.5340635854449	0.193908599185708
82	15.7192731127602	22.6886676390649	0.198490068842455
83	13.2327315290498	21.9279982947964	0.187758305315371
84	10.8173693125792	22.4705361075415	0.142628503962125
85	10.0574010114027	20.0973454859729	0.129966107957181
86	9.20094813067874	17.5458963334087	0.110491299807982
87	6.11073514533937	18.1626638966063	0.0614039410667808
88	5.19122221683258	16.5015394708899	0.0485779664328587
89	3.41349722171946	15.1252866308446	0.0260382506614388
90	4.21825710190045	11.9133721888386	0.0403473316005814