# Implementation of a Particle Tracking Method to Study Particle Spreading in Powder Bed Fusion Systems

Eric Whitenton
Justin Whiting
Alkan Donmez
Aniruddha Das
Vipin Tondare

**NIST** | NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

# Implementation of a Particle Tracking Method to Study Particle Spreading in Powder Bed Fusion Systems

Eric Whitenton
Justin Whiting
Alkan Donmez
Aniruddha Das
Vipin Tondare
*Intelligent Systems Division*
*Engineering Laboratory*

February 2024

**NIST Technical Series Policies**

**Publication History**

**How to Cite this NIST Technical Series Publication**

**NIST Author ORCID iDs**
Eric Whitenton:   0000-0002-8706-7413
Justin Whiting:   0000-0003-0264-1349
Alkan Donmez:   0000-0002-1324-1642
Aniruddha Das:   0000-0002-6451-1657
Vipin Tondare:   0000-0002-0483-1990

**Contact Information**
eric.whitenton@nist.gov

## Abstract

Additive manufacturing is a rapidly growing and increasingly important set of manufacturing techniques. One of those techniques, powder bed fusion, is often used when making metal parts. The part is built up by spreading a thin layer of metal powder, melting that layer with some form of thermal energy, and repeating. However, how that powder flows as it is spread can affect the ultimate part quality. Powder flow characteristics during spreading are poorly understood, and are affected by many things such as powder characteristics, recoater type, recoater speed, platform temperature, cover gas type, and pressure. This paper describes an image-based powder particle tracking method to better understand the motions of the particles during the powder spreading process. It presents preliminary data obtained from two prototype powder spreading testbeds. Such measurements should ultimately enable the validation of physics-based models for powder spreading spreading performance.

## Keywords

**Table of Contents**

**List of Tables**

List of Figures

## 1. Introduction

This report is a companion document to an earlier report [1] describing the development of a powder spreading testbed (PST) at the National Institute of Standards and Technology (NIST), which discusses both powder spreading behavior and powder spreading testbeds in detail. This will be briefly summarized in this section to provide context. The NIST PST, in its first version, is intended to simulate the spreading of powder with a recoater blade, which is the most common type of powder spreading in commercial Powder Bed Fusion (PBF) machines used in Additive Manufacturing (AM). The PST is constructed to mimic a PBF machine and enable the acquisition of data not easily attainable in a commercial PBF system. This data will be used to learn about the physics occurring during powder spreading and to provide data to the AM modeling community regarding the dynamic properties of the blade-type powder spreading process. Other types of spreading processes (e.g., with a roller) will be considered in future versions of the PST.

### 1.1. Particle interactions, heap characteristics, and the nature of shearing

During the blade-type spreading process, most particles are contacting other particles, the recoater blade, and the substrate (or the previously solidified layer surface). The aggregate of these interactions determines the powder's bulk flow behavior to the applied stress. A schematic of a cross-sectional view of powder spreading is provided in **Fig. 1.** The powder *heap*, indicated by the region inside the yellow line in **Fig. 1**, is defined here as any powder that is ahead of the recoater blade in the spreading direction. This is referred to as the *charge* in some commercial PBF machines, and its initial quantity is often a user-controlled variable. The region marked by the black oval in **Fig. 1** will be called the *deposited* or *spread powder*. During powder spreading, while it is likely that a large portion of the powder heap moves as a unified mass, there will be regions or interfaces of shear. The powder that is actually spread or deposited on the substrate is separated from the heap. However, the location of the exact interface of the two regions (i.e., powder heap and deposited powder) is unknown and likely variable, dependent on the conditions of spreading (e.g., coating speed, recoater blade material, environmental conditions, and powder properties). Though these conditions are of great interest, how and where the heap is sheared (shear plane characteristics) are important contributions to a full characterization of the powder spreading process. The shear plane characteristics dictate how a powder is spread and is therefore closely related to the primary characteristics of the spread powder layer such as height, density, and particle-size distribution (PSD). These shear plane characteristics have not yet been fully characterized.
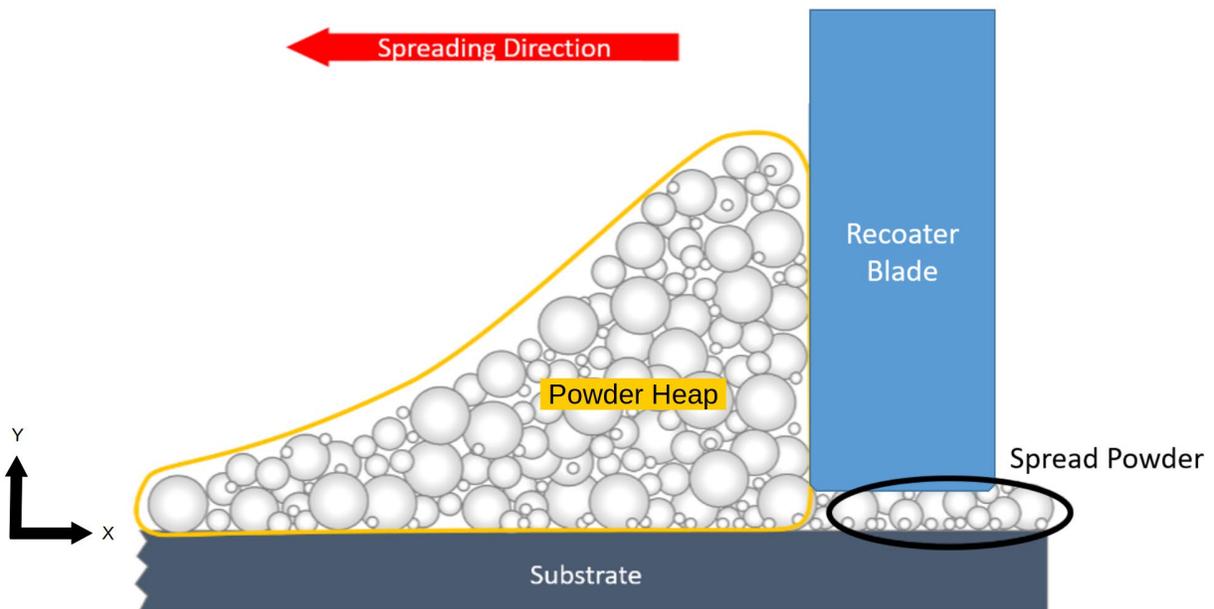
**Fig. 1. Cross-sectional view of powder spreading (note: not to scale).**

## 1.2. Preliminary particle tracking experiments using two NIST PSTs

Particle tracking is a set of techniques aimed at determining the paths, and optionally the velocities, that either individual particles or groups of particles traveled over some period of time [2-6]. Details will be discussed in Section 2.2. It may be used to investigate dynamic properties of the spreading process.

Two PSTs have been built as part of this effort. Due to the particle speeds and magnifications needed, a high-speed camera was used for both.

PST 1: Shown in **Fig. 2**, this testbed was intended as a proof of concept only. A Photron Fastcam SA5 high-speed camera was used to image the powder particles. This gave a spatial resolution of 2.5 μm per pixel.

A                                      B
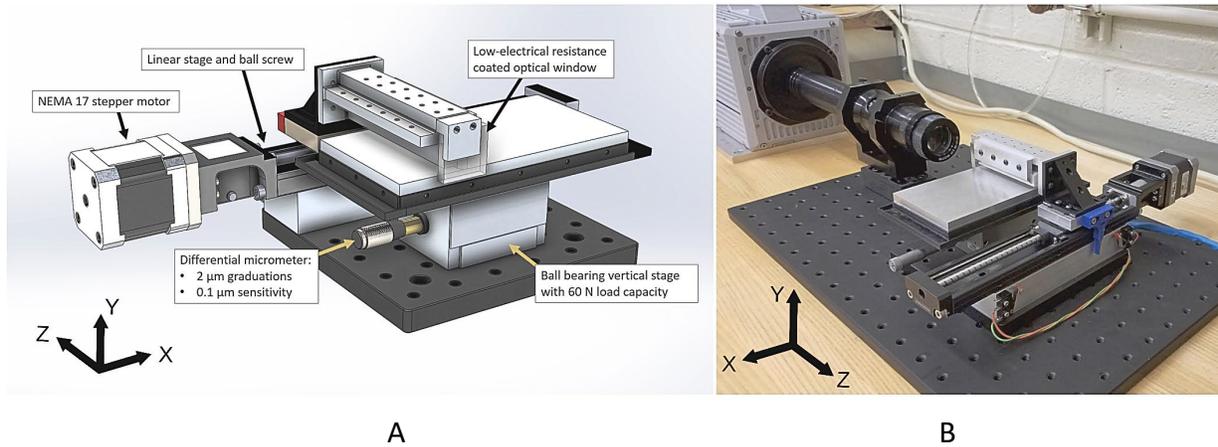
**Fig. 2. Computer aided design (A) and photo (B) of the PST 1 with the imaging camera.**

PST 2: Shown in **Fig. 3**, this testbed is an improved version of PST 1. At time of this writing, it is not fully constructed, with several features still to be added. A Photron Nova S series camera was chosen to image the powder spreading. It is discussed in detail in [1].
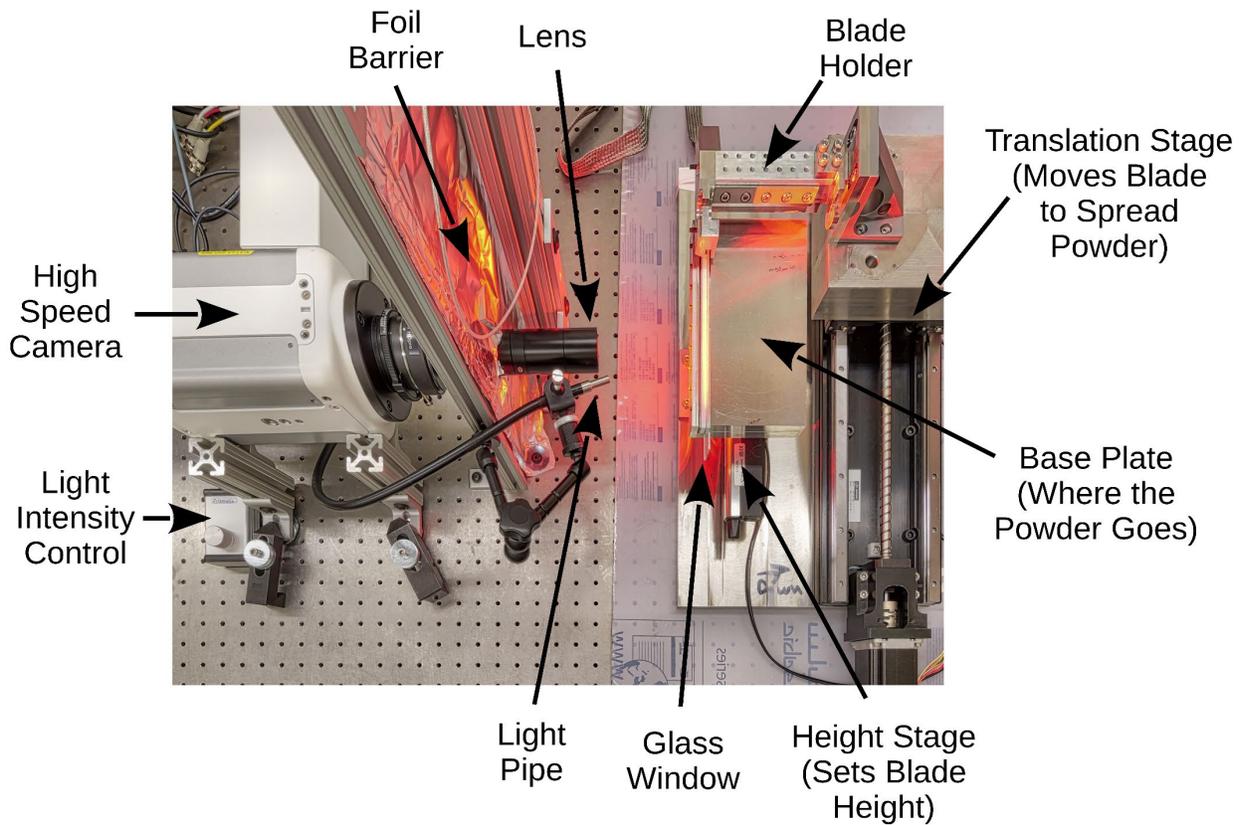


**Fig. 3. Top-down photo of the partially built PST 2 with the imaging camera.**

To test the efficacy of the particle tracking in powder spreading applications, the following three experiments (S1, A1, and S2) were performed. Their parameters are shown in **Table 1**. The powder for Experiment S1 and Experiment S2 was previously used in a commercial PBF machine, containing a few percent of dark colored heat-affected particles. Blade speed was measured by the analysis software discussed in Section 2.5. The field of view was determined by imaging calibrated targets [1]. These experiments were used as test cases to provide a proof of concept for the technique, and also to use while developing the software. Results from Experiment S2 will be emphasized in this paper, with lessons learned from Experiment S1 and Experiment A1 briefly discussed. Ultimately, these tests will be repeated as part of a larger set of experiments with PST 2.

**Table 1. The three experiments examined.**

| ID | Powder | Testbed | Blade Speed | Substrate | Field of View | Video |
|---|---|---|---|---|---|---|
| **S1** | Used. 17-4 stainless steel. Manufacturer's stated D10 of 15 μm and D90 of 45 μm. | PST1 | 27.4 mm/s (3.65 pixel/frame) | Flat plate | 0.98 mm × 0.94 mm | 390 pixel × 375 pixel × 405 frames @ 3000 frame/s |
| **A1** | Virgin. Alpha-alumina (Al2O3). Manufacturer reported D50 of 70 μm ± 5 μm. | PST1 | 5.8 mm/s (0.46 pixel/frame) | Flat plate | 0.88 mm × 0.88 mm | 350 pixel × 350 pixel × 3444 frames @ 5000 frame/s |
| **S2** | Used. 17-4 stainless steel. Manufacturer's stated D10 of 15 μm and D90 of 45 μm. | PST2 | 60.0 mm/s (1.39 pixel/frame) | Already spread powder layer | 2.71 mm × 2.33 mm | 1000 pixel × 860 pixel × 1851 frames @ 16000 frame/s |

All three experiments used telecentric lenses [7]. A telecentric lens is a lens that ensures the spacing between objects in the image remains constant even when the distance between the lens and the objects being imaged changes. This will be discussed more in Section 2.4.

When the particles are being spread, they are obviously moving. How they move over time creates a pattern that is being measured by the tracking software. If that pattern changes very little over some time period, the pattern is said to be *time invariant*. In terms of the type of video captured, there are two options for the location of the camera. One is a moving camera synchronized with the motion of the blade, and the other is a stationary camera attached to the base of the PST [1]. The moving camera always images the same area in close vicinity of the blade. This configuration is better for monitoring time-varying particle flow, but it has a narrow area to study. This contrasts with a stationary camera, where the camera images the blade and its surrounding particles as they go by, including the powder layer created behind the blade. If

one assumes that the characteristics of the particle flow (e.g., velocity field, shear band location, heap size, spread layer height, etc.) are time-invariant over short time periods, one composite plot can be created for an equivalent to a wide field of view by combining multiple individual images from a more limited field of view. Using this assumption, a stationary camera configuration was used for all three experiments.

As shown in Fig. 2 and Fig. 3, there is a piece of glass constraining the powder from falling off the camera side of the baseplate. When powder is spread in a real-world PBF, most particles contact other particles on all sides. Thus, most particles feel the influence of other particles limiting, though not eliminating, motion perpendicular to the spreading direction. The glass enables capturing high-speed videos of powder spreading from the other side of the glass, as well as approximating the constraints for the lateral motion of the particles during spreading.

There are two options regarding whether the glass should be fixed to the moving blade of the PST (moving glass) or fixed to the base of the PST (stationary glass). In the case of moving glass, its speed is close to that of the bulk powder. As a result, relative speed between the glass and the powder is minimized, which also minimizes any effects of friction between the glass and the powder. However, this makes it harder to seal the glass to the base to prevent powder leakage. This situation is reversed with case of stationary glass. There can still be powder leakage, but it is more likely to be between the glass and the side of the blade. Stationary glass was used to for all three experiments.

## 2. Practical considerations in particle tracking experiments

Particle tracking has limitations. Some are obvious. For example, a particle cannot be tracked when it is moving so fast that it is imaged by only one frame of the video before it moves completely out of the field of view. A single frame shows the location of a particle at only one specific moment in time. It does not contain information on how that particle moved. It is only by comparing that location to locations of the particle at other times that the motion of the particle may by tracked. Other limitations are more subtle. This section examines some of these.

### 2.1. Frame rate, resolution, particle size, blade speed

The process of measuring particle flow involves several sequential steps for a complete measurement chain. To determine acceptable conditions for an accurate measurement that avoids error-prone results, a constraints analysis may be performed.

Each experiment to be conducted may require different measurement settings. For example, a magnification appropriate for imaging one type of powder might not work for a different type of powder with larger or smaller particle sizes. To provide guidance for the selection of the measurement settings, a constraint analysis using an approximation with a simple 2D model of the imaging process was performed. The main variables used are listed in the following.

$G$ — A unitless generic safety factor for all constraints. Should be 1 or larger. A value of 1 is no safety factor. The larger the value, the larger the safety factor.

$W_{act}$ — Actual horizontal image width in pixels.

$H_{act}$ — Actual vertical image height in pixels.

$S$ — Image scaling in µm/pixel.

$W_{min}$ — Minimum acceptable horizontal width in µm seen by the image so you see everything you need to see.

$H_{min}$ — Minimum acceptable vertical height in µm seen by the image so you see everything you want to see. For example, the full height of the powder heap.

$F$ — Imaging frame rate in frames/second.

$D_{max}$ — Maximum particle size as a diameter in µm. For this testing, the D90 was used since the actual maximum was not known.

$D_{min}$ — Minimum particle size as a diameter in µm. For this testing, the D10 was used since the actual minimum was not known.

$B$ — Blade velocity in µm/s.

Using these variables, we defined five constraints using the following equations:

Constraint 1: Is the full desired width visible?

$$W_{act}\, S \geq G\, W_{min} \qquad\qquad (1)$$

Constraint 2: Is the full desired height visible?

$$H_{act}\, S \geq G\, H_{min} \qquad\qquad (2)$$

Constraint 3: Does the largest particle fit in the field of view?

$$\min(W_{act}\,,\, H_{act})\, S \geq G\, D_{max} \qquad (3)$$

Constraint 4: Insufficient magnification will result in the inability to resolve individual small particles. Thus, a particle should span at least two pixels. This is essentially the Nyquist criteria.

$$D_{\min}/S \geq G\, 2 \qquad\qquad (4)$$

Constraint 5: For the tracking software to work reliably, a particle should never move more than 1/2 particle diameter between two consecutive video frames. If the spacing between particles were large, then this would be less of an issue. However, in case of powder spreading, the particles are directly touching. In fact, some particles are partially obstructed by others, exacerbating the situation. Further, as the powder is spread, there exists the potential for motion along the Z-axis (closer or farther away from the camera, as shown in **Fig. 2**). Therefore, the visibility of particles may change during the spreading process (i.e., from visible to hidden or vice-versa).

$$D_{\min}/2 \; \geq \; G\,B/F \qquad\qquad (5)$$

**Table 2** shows constraints analyses for the Experiment S2 mentioned in Section 1.2. If a value exceeds a constraint (crosses the threshold from acceptable to unacceptable), values in that constraint column are highlighted red. Experiment S2 is always well within acceptable values, so all columns are colored gray. Looking at the row for G, one can see that we arbitrarily picked a target value of 1.02, or a safety factor of 2 %. However, the worst actual constraint was for Constraint 2, with a value of 1.17. Thus, the actual safety factor is 17 % or more for all constraints.

**Table 2. Constraints analysis for Experiment S2.**

| Variables | | | | Constraints Boundaries | | | | |
|---|---|---|---|---|---|---|---|---|
| Symbol | Brief Description | Units | Value | Eq. 1 | Eq. 2 | Eq. 3 | Eq. 4 | Eq. 5 |
| $G$ | Safety factor | none | 1.02 | < 1.36 | < 1.17 | < 51.79 | < 2.77 | < 2.00 |
| $W_{act}$ | Image width | pixel | 1000 | > 753 | | > 0 | | |
| $H_{act}$ | Image height | pixel | 860 | | > 753 | > 17 | | |
| $W_{min}$ | Min image width | μm | 2000 | < 2657 | | | | |
| $H_{min}$ | Min image height | μm | 2000 | | < 2285 | | | |
| $S$ | Image scaling | μm / pixel | 2.71 | > 2.04 | > 2.37 | > 0.05 | < 7.53 | |
| $F$ | Frame rate | frames / s | 16000 | | | | | > 8160 |
| $D_{max}$ | Max particle size | μm | 45 | | | < 2285 | | |
| $D_{min}$ | Min particle size | μm | 15 | | | | > 5.5 | > 7.7 |
| $B$ | Blade speed | μm / s | 60 000 | | | | | < 117 647 |

## 2.2. Effects of lens optics and lighting

Some of the impacts of the lens magnification were discussed in Section 2.1. However, there are additional impacts as well.

8

Metal powder particles are often very reflective. Often, these reflections are the brightest area seen on any individual particle. This means that the particle tracking software often is effectively tracking reflections off the particle, and not the particle as a whole. **Fig. 4** shows an image from Experiment S1. There was only one light source, so each particle should have only one bright reflection on it. However, each particle has multiple reflections, with the brightest reflection being of the light source. The generally dimmer reflections are reflections of bright reflections of neighboring particles. Green arrows illustrate some of those, with the head of each arrow pointing to the first bright reflection of the light source, and the tail of the arrow originating at the secondary reflection in a neighboring particle. In the low magnification Experiment S2, the camera cannot see each particle in as much detail as in Experiment S1, so these "reflections of reflections" are not much of an issue. For the high magnification Experiment S1, since these reflections of reflections are dimmer than the original reflections, it is tempting to try and remove them by setting a threshold value in post processing so only the brighter original reflections are processed. However, due to the reflective but also rough nature of surfaces of many particles, the brightness changes dynamically as the particles move and rotate. This does not mean that the data is unusable, but it does mean the data processing might need to be more sophisticated than simply setting a single brightness threshold. This will be discussed more in Section 2.5.
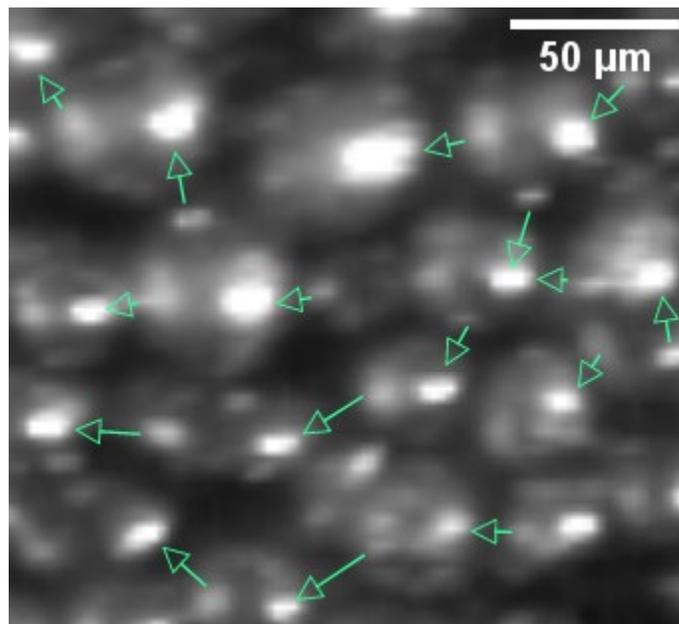


**Fig. 4. Portion of image from Experiment S1. Green arrows illustrate "reflections of reflections." This is one example of how each particle can appear as several particles to tracking software.**

**Figure 5A** shows a portion of an image from Experiment A1 described in Table 1. Two light sources were used in this test. This results in two bright reflections for each particle. As a result,

the particle tracking software produced two tracks for each particle. This can be corrected and is discussed more in Section 2.5.

**Figure 5B** shows an early attempt to use image processing to see if one could determine particle sizes as well as track the particles. At higher magnification, it should be possible to measure both, at least for some of the particles. However, both lower and higher magnification have their own trade-offs. These are summarized in **Table 3**.  For example, a low magnification system produces a relatively larger field of view and larger depth of focus, etc.



| A | B |
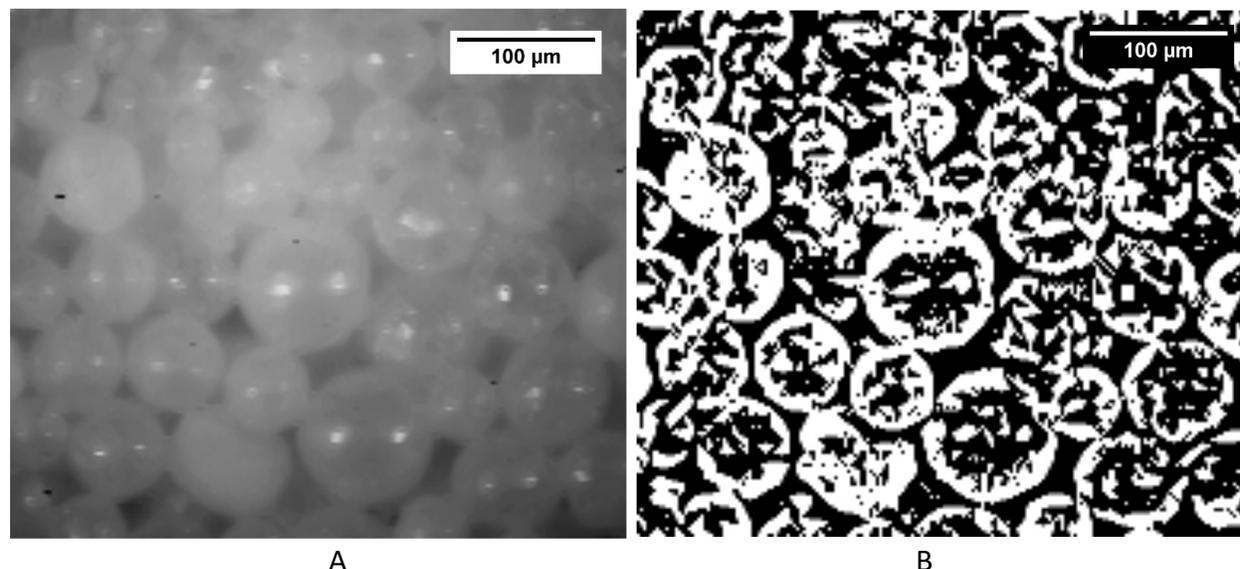
**Fig. 5. Figure A shows a portion of an image from Experiment A1. Figure B shows a processed version of the same portion to illustrate that it is possible to estimate particle sizes from the images.**

**Table 3. Summary of some of the advantages and disadvantages to consider when choosing the magnification to use.**

|  | Low Magnification | High Magnification |
|---|---|---|
| **Field of view.** | Larger. | Smaller. |
| **Depth of focus.** | Larger. | Smaller. |
| **Trackable blade speeds.** | Fast or slow. | Limited to slow. |
| **Ability to track very small particles.** | Poor. | Better. |
| **Ability to track partially obscured particles.** | Poor. | Better. |
| **Ability to characterize particle shapes and sizes along with their tracks.** | Poor. | Better. |
| **Manually editing particle and track data.** | More difficult due to having much more data. | Easy. |

Magnification is not the only lens property that can impact measurement results. For example, **Fig. 6A** shows an in-focus image of a 1951 United States Air Force (USAF) tri-bar resolution target [7], with a yellow rectangular area drawn. The gray values of the rows of pixels within that yellow rectangle are averaged to produce the plot in **Fig. 6B**. It shows the average intensity across three vertical bars. As expected, this curve looks like a square wave with some rounding of the corners.



A                                                    B
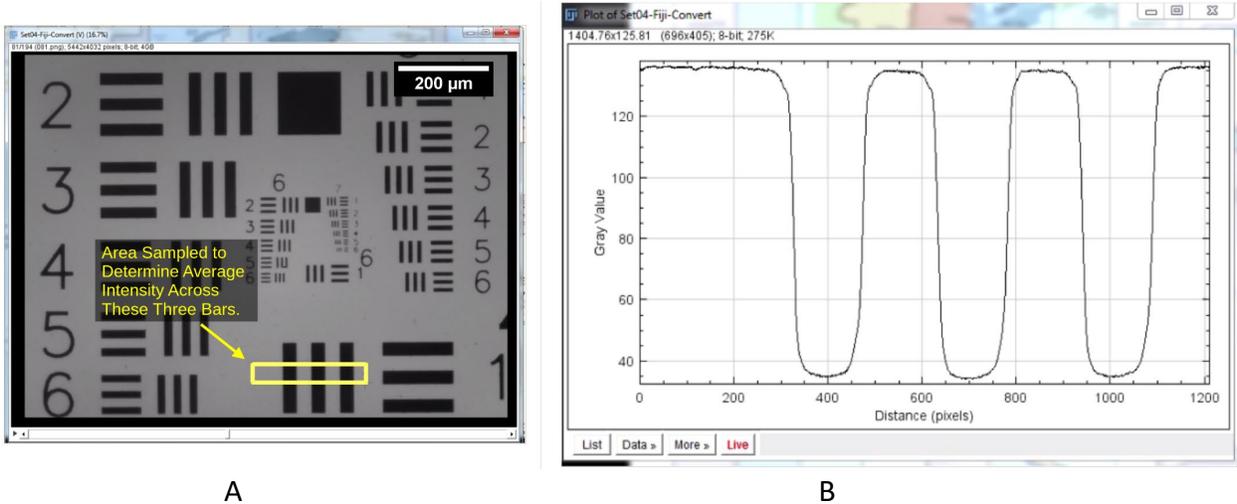
**Fig. 6. Three bar target and a trace for one set of bars when the image is sharply in focus.**

**Figure 7** shows the same analysis, but with the image out of focus. Intensity across the three bars resembles a distorted sine wave. Using a telecentric lens ensures that, no matter how out of focus the image is, the spacing between the peaks remains constant. Thus, particle velocity measurements should remain accurate even if the particle is out of focus. However, if a conventional lens is used instead, the spacing between the peaks will change.
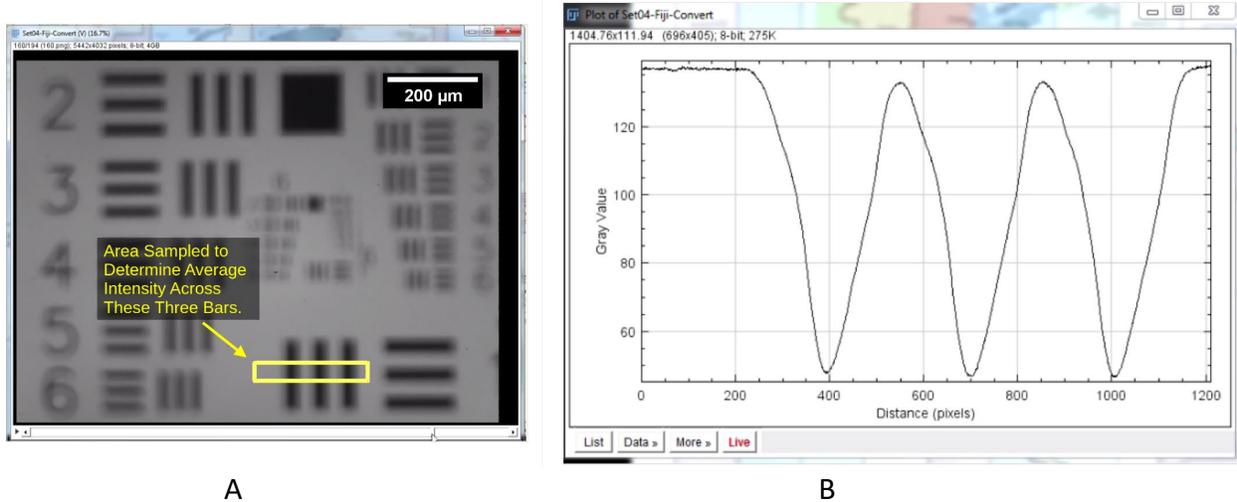
**Fig. 7. Three bar target and a trace for one set of bars when the image is out of focus by 160 μm.**

Comparing the in-focus **Fig. 6** to the out of focus **Fig. 7**, when the image is out of focus, the wave becomes more rounded and the amplitude of the wave becomes smaller. If the camera continues to move more and more out of focus, the amplitude will eventually become so small that the wave becomes lost in the pixel-to-pixel random noise in the image. The distance at which this happens depends on the size of the features and the spacing between those features. There are many ways to characterize this. A discussion of all these ways is beyond the scope of this paper. In addition, while the three-bar target is an easy to use target, the real grayscale contrast of the metal powders is quite different, so readers should note that these feature size measurements are not necessarily equivalent to powder particle size. However, it should provide a quantitative characterization of performance.

For this paper, the following procedure was used. A series of images of the three-bar target were acquired at a range of distances between the camera and the target. The camera was moved 2 μm between each image. For each set of three bars in each of those images, a procedure similar to those shown in Fig. 6 and Fig. 7 was performed, where a wave was computed and the peak height determined for each trace. The spacing between the peaks of those bars (in μm) are known from their location in the image. If the peak height is greater than a specified threshold value, the associated bars are declared sufficiently in-focus to be detectable by the tracking software at that camera distance. A reasonable threshold level was found to be 25 grayscale units, as determined by measuring image noise at various locations in the images. The range of camera distances considered in focus enough to be detectable (above the pixel-to-pixel noise) is declared the range of detectable depths for bars of that bar size.

**Figure 8** shows the results of combining the range of detectable depths data for all sizes of bars. This plot indicates that small particles are detectable over a small range of distances from the camera while large particles are detectable over a significantly larger range of distances. If one

assumes that actual particles are at approximately random distances from the camera, then there is a higher likelihood that the large particles will be in focus since they are in-focus over a wider range of camera distances. Thus, a larger fraction of tracks for large particles are likely to be detected than for small particles. This can result in sampling bias. Whether this adversely affects statistics computed from the tracks depends on what statistical metrics are chosen, and how they are used.
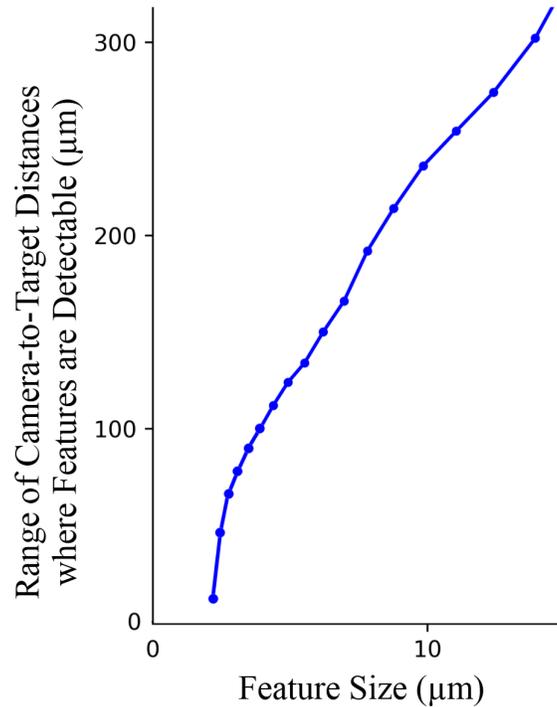


**Fig. 8. Range of camera-to-target distances where features are detectable as a function of feature size. This illustrates that smaller particles are harder for the software to detect than larger ones.**

For example, in a population of particles, assume the large particles move at velocity $V_l$ and small particles move at velocity $V_s$. Let the relative percentage of the total population for the large and small particles be $P_l$ and $P_s$, respectively. The average measured velocity will be approximately

$$[(P_l\,V_l) + (P_s\,V_s)] \,/\, (P_l + P_s). \qquad (6)$$

However, if the large particles are detected at twice the rate the small particles are detected, the average measured velocity will change to approximately

$$[(2\,P_l\,V_l) + (P_s\,V_s)] \,/\, [(2\,P_l) + P_s]. \qquad (7)$$

**Figure 9** shows an example of out-of-focus bias. Particle 5 is close enough to the plane of focus and is detected while Particle 2 might be too blurred and missed. However, other effects could bias averages toward the larger particles. For example, Particle 4 is partially obscured by Particle 3 and Particle 5 in front of it. Particle 4 is visible, but it may or may not actually be detected. A similar situation applies to Particle 1.
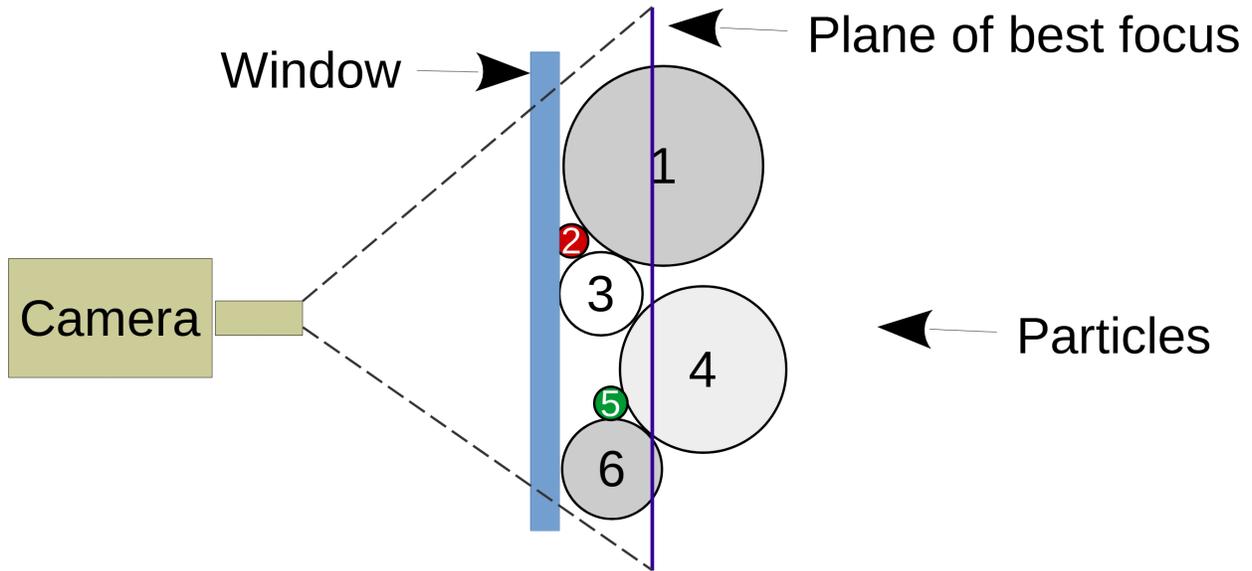


**Fig. 9. Side view of camera imaging particles of various sizes. This illustrates how particles can partially obscure other particles, making particle detection more complicated than just a 2D approximation.**

## 2.3. Particle tracking

Particle tracking techniques work primarily by processing a series of images of the objects being tracked, which are powder particles in this case. Note that some techniques also specify lighting equipment used, such as devices that emit a planar sheet of light. Also, there are similar techniques with different names, such as "optical flow." A discussion of all known tracking techniques is beyond the scope of this paper. In this paper, the term particle tracking refers to single-particle tracking. The images obtained are front lit, using a conventional high-intensity light source.

### 2.4. Comparison of a track-based representation to a vector field representation of particle tracking data

Regardless of the specific technique used to determine particle paths, the resulting data may be represented either as tracks or as vector fields. A track is a chain of points connected by line segments. An example is shown in **Fig. 10A**. Each point represents the position of the particle in the video frame coordinate system and the line segment represents how the particle moved from one frame of the video to the next frame over a fixed time interval determined by the frame rate of the camera used. Since each line segment occurred over the known inter-frame time, the velocity or acceleration can be easily computed. The entire chain represents how an individual particle moved over the period of observation. Also note that since each point represents the particle position in a different video frame, the particle position is measured at different times. Our implementation of tracks retains this time information. This will be used when we analyze the data in Section 3.

Alternatively, one can employ vector fields for flow characterization as is often done in particle image velocimetry (PIV). For vector fields, a user-specified, uniformly spaced grid is laid over each image of the video. At each node of the grid, a "typical" behavior of all the tracks in that local neighborhood is represented by a vector. There are many ways to define typical behavior, depending on the application. An example is shown in **Fig. 10B**. The direction of each resulting vector represents the typical direction of the paths in that neighborhood. The length often represents the average velocity, though other parameters could be represented such as acceleration. Tracks and vector fields each have both advantages and disadvantages.



A                                                                 B
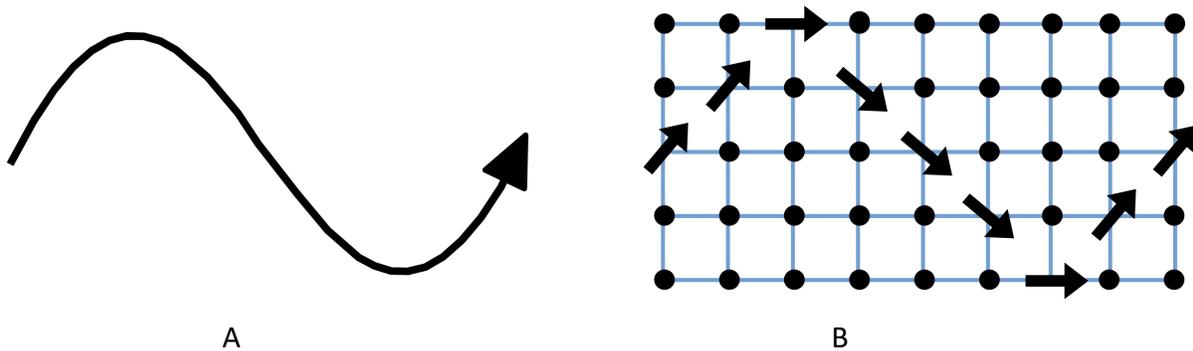
**Fig. 10. To compare track and vector field representations, Figure A shows a single track while Figure B shows one possible accompanying vector field. The grid of possible vector locations is shown by the blue grid. Vectors of zero length appear as a black dot.**

The vector field representation has the effect of averaging track information in two ways, spatial averaging and temporal averaging. For both, the amount of averaging is determined primarily by grid spacing and how the local neighborhood is defined by the user. This impacts the range of sizes of undulations in the tracks which are observable in the corresponding vector

field. Another way to think of this is that the accuracy of the tracks is limited mainly by the camera and optics used to acquire the data while vector fields average/agglomerate that data. Whether this averaging/agglomeration is helpful depends on how the vector field was derived and on the information the user is seeking. To illustrate this, **Fig. 11A** shows two tracks (green and red) of two individual particles. The tracks are identical except for being slightly shifted in the vertical direction. **Figures 11B, C** and **D** show three possible vector field representations of those same two tracks. Each representation has a different grid spacing, arbitrarily chosen solely for illustrative purposes. The neighborhood is a circular region with a radius of ½ the grid spacing, centered on the grid point (i.e., the node).
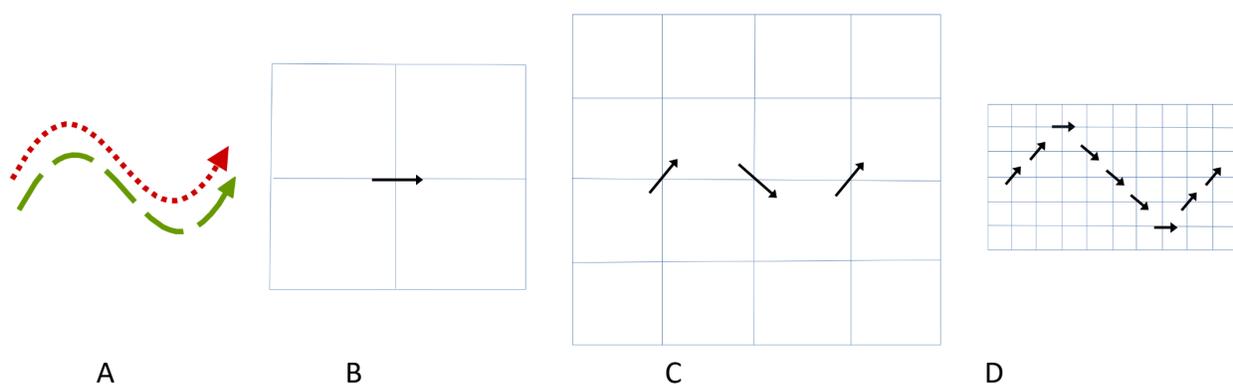


Fig. 11. Two tracks are shown in Fig. A. Figures B, C, and D show several possible vector field representations of those same two tracks, depending on the grid's coarseness. This illustrates how the user selected grid spacing can affect perceived motion of the particles.

In this report, the color coding for the tracks is used only to differentiate between the tracks of two different particles. Obviously, other coding schemes are possible. For visualizing the vector fields, the vectors are set to black, with the vector lengths set to a constant. The background color in vector field plots can be chosen for visual appearance or used to represent specific data. For **Fig. 10** and **Fig. 11**, no background color was used. Later in this paper, the background color will indicate the magnitude of the particle velocity. Also, in most plots only vectors of statistically significant non-zero length will be shown to make the plots easier to read. As will be seen in later plots, these choices were made to keep the very dense plots of actual data easier to read.

**Figure 12** shows a red and a green particle following identical but shifted paths. Only one vector of the vector field is rendered. The circle represents the neighborhood for that vector. In **Fig. 12A**, the tracks are aligned well enough that the vectors solved for in the black circle will accurately represent both tracks. **Figure 6B** shows the same paths but are effectively "out of phase" in the horizontal direction. A vector representation in the same neighborhood (black circle) averages the +45 degree of the red path and the -45 degree of the green path to yield an average vector of 0 degrees. As a result, potentially important information about how

individual particles flow can be smoothed out and lost when using vector fields, or when calculating averages over too large a grid.
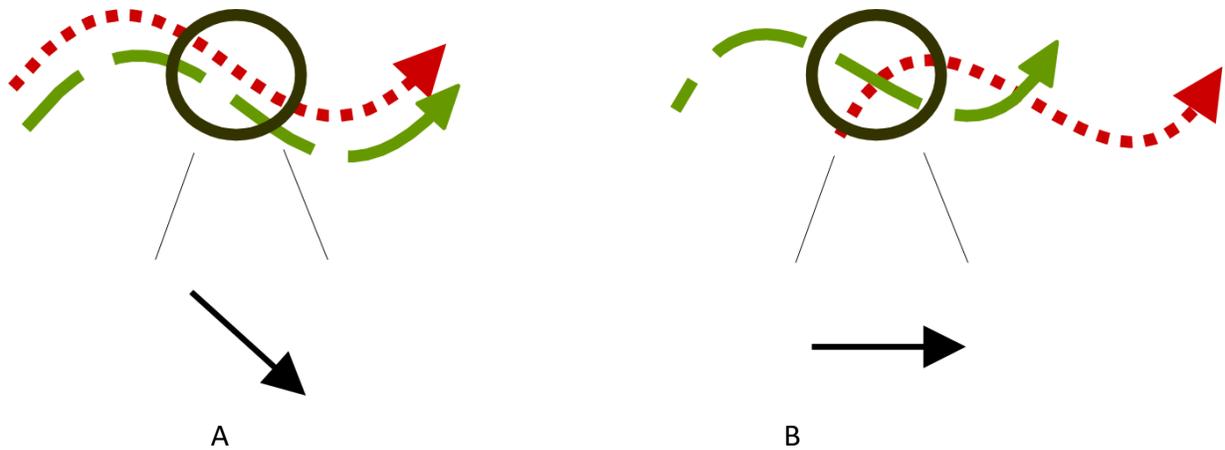


**Fig. 12. Two plots, each showing two tracks. The resulting vector in the circle shows where vector fields would be different, even though tracks are identical but shifted versions of each other. This is one illustration that the vector field representation is an averaged representation of the track information.**

As discussed in Section 2.2, by necessity researchers often simplify powder spreading analysis by using 2D models, despite the potential impact on the accuracy of the results. An example of such simplification is shown in **Fig. 13**. Both **Fig. 13A** and **Fig. 13B** have the same vector fields, even though the tracks are very different. In **Fig. 13A**, the two paths crossed, either due to one particle traveling behind the other, or one particle passing through a little before the other in time. This contrasts with **Fig. 13B**, where their paths never crossed and both particles actually changed direction. This issue is generally completely ignored by the vector field representation. By contrast, tracks inherently contain this information. However, it is possible for the tracking software to assume one is occurring when it is in fact the other. Tracking software often allows users to either manually delete such tracks or manually fix them. The software employed here currently only allows manual deletion. Some specialized tracking software will perform Multiple Hypothesis Testing (MHT) and produce two outputs, one for each possibility. MHT is practical for small numbers of particles but can become memory intensive when many particles or many outputs are required.
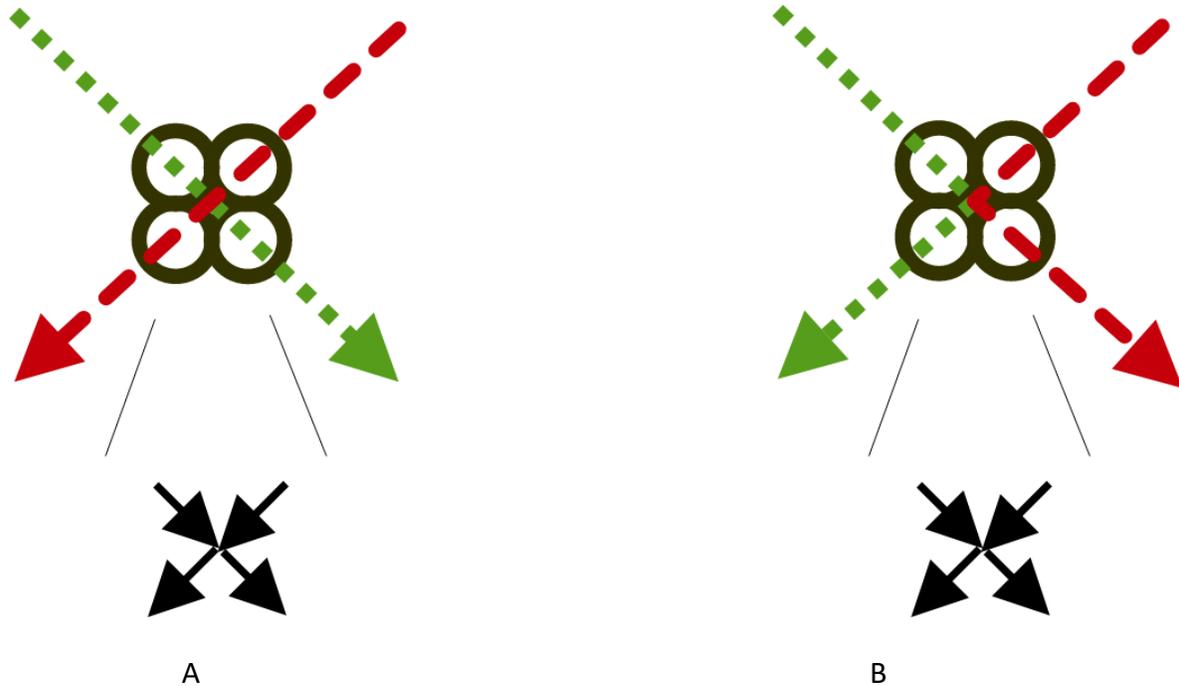
**Fig. 13. Two plots, each showing tracks. The vector field for each are the same, even though the tracks cross in one case and do not cross in the other. This illustrates how the vector field representation will always contain less information than a track representation.**

## 2.5.  Implementation of tracking software

The image analysis was performed using Knime Analytics Platform [8], which provides a framework, image processing functions, Python scripting, and Trackmate tracking software. Similar to Lab View, Knime's user interface uses a node and connecting wires style of programming. The workflow is executed automatically where possible, and pauses for user input as needed. There were subtle differences in how the data was analyzed for Experiment S2 as compared to Experiment S1 and Experiment A1. This is primarily due to Experiment S2 having a much lower magnification and higher frame rate than the other experiments. As noted in Section 2.2, high magnification and low magnification images present different challenges and advantages. The ultimate goal is to have the workflow able to process either situation with minimal intervention from the user. However, this is still a work in progress.

Next, we present two simplified versions of the workflow. First, a highly simplified version is presented to help introduce concepts.

1 - Manually trim and crop video. Any frames, or portions of frames, which do not add interesting information should be trimmed/cropped so both processing time and size of processed data are kept reasonable.

2 - Manually determine blade coordinates in video. Since the blade speed will ultimately be determined, if the edges outlining the blade are determined in any of the frames, the software can determine blade location in all frames.

3 - Generate and label spots. First, the software automatically performs simple corrections, such as converting images to grayscale if needed, where each pixel represents a brightness value without any color information. Also, the overall brightness is normalized. Next, the software detects individual particles, referred to as "spots". The Spot Detection node performs thresholding to separate the particles from the background portion of each image. The method is based on Reference [9]. Each spot is given a unique name, referred to as a label. For example, "Spot 9636."

4 - Filter out obviously "bad" spots. Perform a series of tests to filter out spots that are obviously not particles. For example, oblong spots may not really be particles since our particles are nearly spherical. Also, if a particle touches the edge of an image, only a part of the particle may be visible. To avoid ambiguity, these particles are rejected.

5 - Generate and label "raw" tracks, where all locations are relative to the camera coordinates. Use Trackmate to detect particle motion tracks. The tracks are determined using the Trackmate Tracker node, which is built on Trackmate [10-12]. Here, the term "particle tracking" refers to single-particle tracking, as opposed to tracking groups of particles. Trackmate works by particle-linking, which is a two-step process. First, track segments are created using frame-to-frame particle linking. Second, track segments are linked to achieve gap closing. The mathematical formulation used for both steps is a linear assignment problem, where a cost matrix is assembled containing all possible assignment costs. Actual assignments are retrieved by solving this matrix for the minimal total cost. Trackmate contains many options to optimize this process.

6 - Filter out obviously "bad" raw tracks. For example, if a track is only two frames long, it was ignored in this study.

7 - Determine blade speed. The user can either manually enter the blade speed, or the software will estimate the blade speed by measuring the movement of scratches on the end of the blade.

8 - Use blade speed and frame rate information to shift tracks so locations are relative to blade.

$$X_{shifted} = X_{original} + (FrameNumber \cdot BladeSpeed) \qquad (8)$$

$X_{shifted}$ and $X_{original}$ are in pixels and *BladeSpeed* is in pixels / frame.

9 - Filter out obviously "bad" shifted tracks. For example, tracks which do not represent particles, but actually are scratches on the end of the blade, should be removed.

10 – If appropriate, manually perform any needed additional filtering of shifted tracks. As discussed in Section 2.2, the higher magnification tests have the advantage that, with fewer tracks to process, many aspects of that processing are easier and faster. This includes the ability to edit tracks directly if the automatic track removal routines miss any undesirable tracks.

11 - Generate vector fields from shifted track data as discussed in Section 2.4.

12 - Output processed results such as plots, videos, and metadata files.

A more detailed version of the workflow is presented next. Only those steps which are significantly different than the highly simplified version are expanded upon.

1 - Manually trim and crop video.

2 - Manually determine blade coordinates in video.

3 - Generate and label spots.

4 - As discussed in Section 2.2, sometimes an individual particle can produce two or more roughly parallel tracks. In theory, removing these duplicate tracks could be accomplished by combining all spots for the same particle into one spot, which can be performed using the Waehlby Cell Clump Splitter [13]. This was tried while developing this workflow, but it did not work well for the three experiments mentioned in Section 1.2. Despite this, the authors decided to keep it in the workflow as an optional step in case it might be useful for future data. It proved better to delete duplicate tracks instead and will be addressed in Step 7.

5 - Filter out obviously "bad" spots.

6 - Generate and label raw tracks. This is currently performed by Trackmate. However, one of the functions Trackmate performs has been problematic in this application. It is where Trackmate attempts to fill in gaps between tracks, which was more of a challenge for the low magnification Experiment S2. Ideally, a particle would generate only one continuous track, starting from when the particle first comes into view and ends when the particle goes out of view. However, particles do not move only in X and Y, but also closer and further away from the camera. Thus, instead of one long track, that track can be broken up into several smaller pieces. Trackmate can attempt to fill in those gaps. However, Trackmate appears to do so based mainly on the distance from the end of one track to the end of another nearby track. If the user sets this distance too short, most gaps go unfilled. If the user sets this distance too long, tracks

which should not be connected become connected, which results in portions of tracks being incorrect. The results presented for Experiment S2 used a very short distance. Thus, most particles are represented by a series of short, but accurate, tracks. It may be possible to improve this situation. For example, when determining whether to close a gap, local slopes (in X, Y) of the ends of the tracks can be considered in addition to the distance. This is left as future work as of this writing.

7 - If appropriate, remove "duplicate" tracks. As discussed in Section 2.2, sometimes each particle can produce two or more roughly parallel tracks. In theory, removing these duplicate tracks could be accomplished either in Step 5, or in this step. However, in practice, the authors found that it is more robust when performed in this step. For example, **Fig. 14A** shows data from Experiment A1. The parallel, duplicate tracks were removed, and the results shown in **Fig. 14B**. Custom code written in a Knime Python Node was used for this.



A                                    B
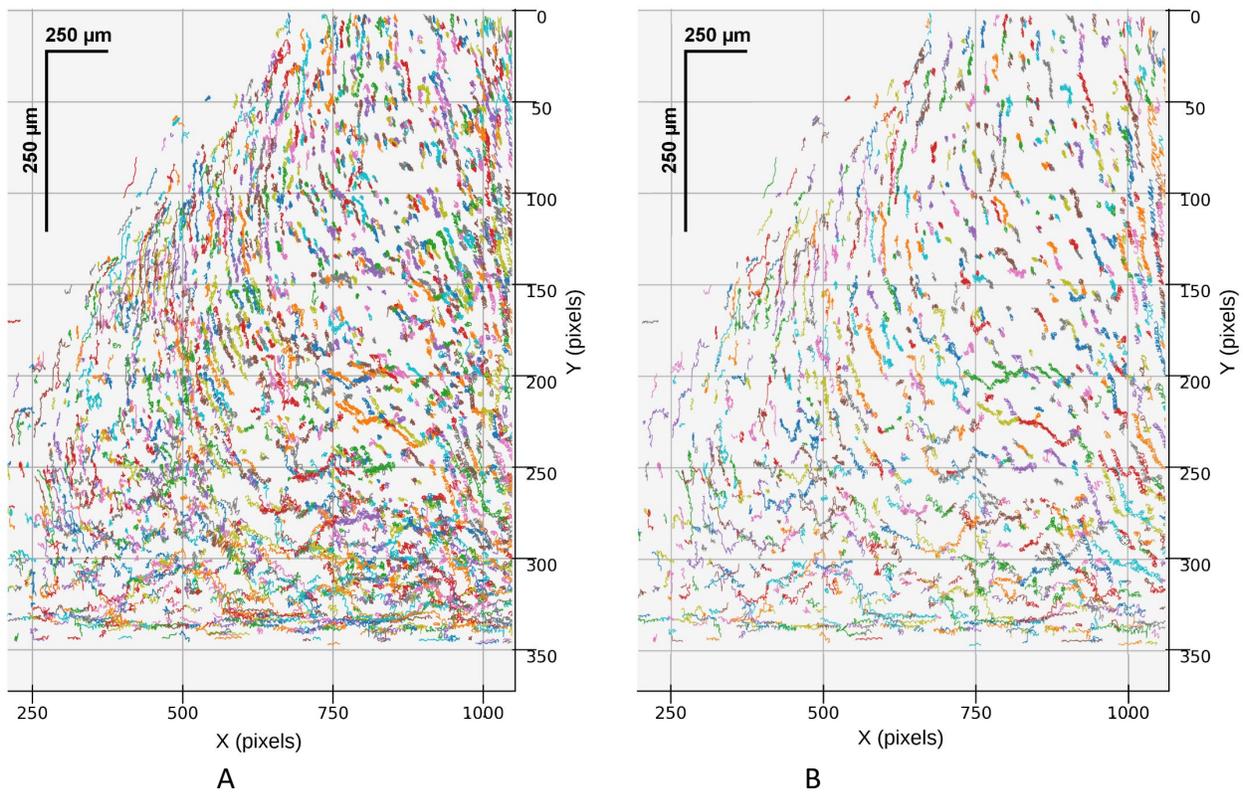
**Fig. 14. Removing redundant tracks from Experiment A1 data by analyzing track data.**

8 - Filter out obviously "bad" raw tracks.

9 - If a good estimate of blade speed is available, the user can enter it. If not, then automatically select tracks that represent blade motion to estimate blade speed and estimate it.

10 - Make a video of images shifted using Equation 8. If the estimated blade speed is correct, and the actual blade speed is constant, the blade will appear stationary in the resulting video. Use this video to manually verify blade speed is correct, and that there is no dithering. Adjust the estimated blade speed if needed. Even though other similar videos will be generated later, this video has only shifted images of the blade. The reason for making this separate, diagnostic video is that it is fast to render, and it is easiest for the user to see the blade without visual clutter. As an example, consider the "merged movie" for Experiment S2 [14], which will be discussed in Section 3. In later frames, from t = 62 ms to t = 114 ms, the apparent location of the blade appears fixed. This indicates that the chosen blade speed was correct, and that the blade speed was constant. If the blade speed is incorrect, the blade will appear to drift across the screen over time. The user can manually measure this drift (in pixels/frame) and use that information to correct the blade speed (which is also in pixels/frame). If the blade appears to oscillate left and right in the image, then either the blade speed was not constant, or there was visible vibration in the PST. If the amplitude of this oscillation is significant, the user may want to check the test bed to make sure there is nothing wrong with it.

11 - Shift tracks so locations are relative to blade.

12 - Make draft version of track-image merged movie, similar to [14]. Use this to manually determine masks. Some track filtering can be performed algorithmically, such as, for example, a track of length two. However, there are other situations where it is easier and faster to remove tracks in areas that are clearly incorrect. For example, any tracks which are on the end of the blade are either just scratches or are particles stuck in between the blade and the glass. Also, certain locations in the data are clearly just particles stuck to the glass and not representative of normal particle flow. The ability to define such closed irregular polygonal areas, and remove unwanted data from those areas, is referred to as masking. Once the user has defined said masks, the software can automatically remove data from those areas.

13 - Filter out shifted tracks with any points in masked areas.

14 - Filter out obviously "bad" shifted tracks.

15 - Generate vector fields from shifted track data, filtering out vectors whose centers are in any of the masked areas. Because vector fields measure behavior in a local neighborhood, the software can create vectors that are next to tracks, even though there were no actual tracks directly beneath. Thus, the same masks used to filter "bad" tracks are used again to filter "bad" vectors.

16 - Output processed results such as plots, videos, and metadata files.

## 3. Results

The results presented will focus on Experiment S2. **Figure 15** shows a typical frame from the raw video. The full video is available at [15]. As noted in [1], there is a small amount of leakage allowing particles to become wedged between the end of the blade and the glass. These particles will be rejected by the particle tracking software.
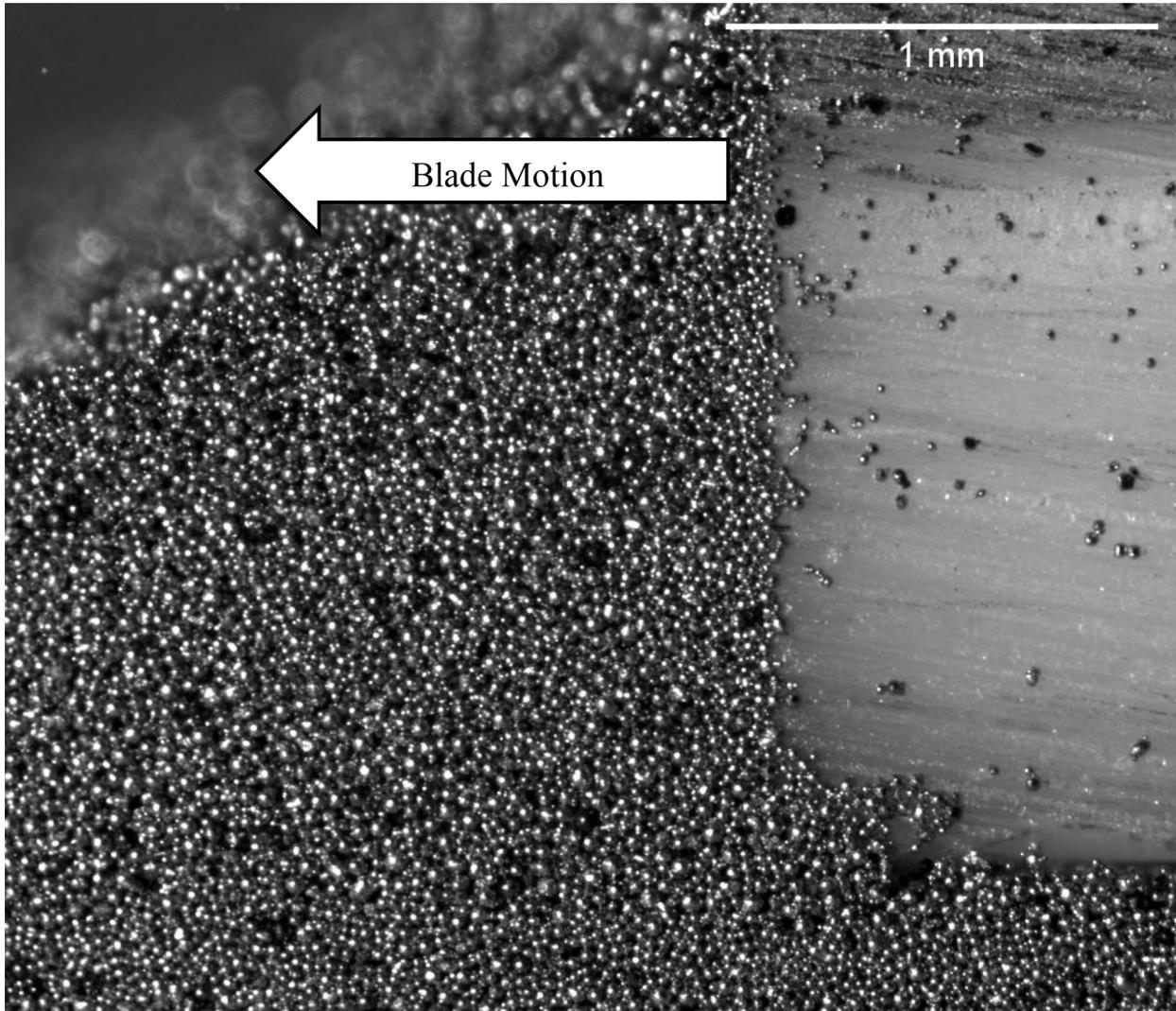


**Fig. 15. A typical frame from the raw video for Experiment S2. The field of view is 2.71 mm x 2.33 mm. The arrow shows the direction of blade motion.**

To better illustrate the range of images being processed, **Fig. 16** shows an image near the beginning, the middle, and the end of the video.
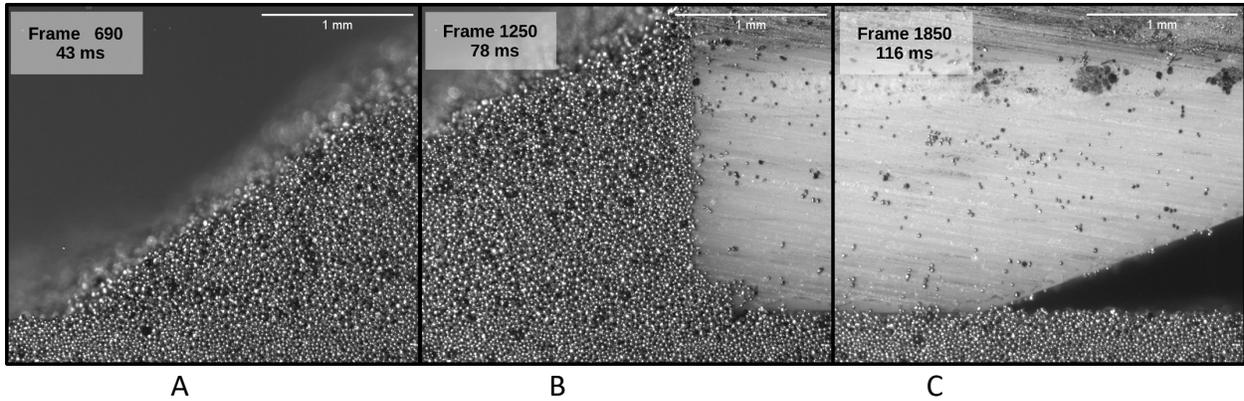
Fig. 16. Three frames from the video for Experiment S2, one near the beginning, the middle, and the end.

The processing described in Section 2.5 produces two forms of the data. For both forms, X and Y are locations relative to the blade. These locations can be made relative to the baseplate by using **Equation 8** if desired. The first form is a list of track information, where each item in the list represents one track. The data for each track is a Track ID Number paired with a list of (X, Y, Frame Number) triplets. The other form is a list of vector information, where each item in the list represents one vector. Each vector is located at an X, Y location on a grid of nodes determined by the user. The data for each vector is a set of (X, Y, dX, dY, stdX, stdY) scalars, where X and Y is the location of the center of the vector, dX and dY is the delta X and delta Y for that vector, and stdX and stdY are the standard deviations for the dX's and dY's of the tracks in that local neighborhood. Eventually, stdX and stdY may be used in uncertainty analyses for the vector data. **Figure 17** shows the track data projected onto the X, Y plane. A black line representing the outline of the blade is also shown. Note the scaling for the plots are different in X than in Y. Remember that data from many frames were combined to create these tracks, with each frame mathematically moved in X as the blade moved. Thus, the range of X becomes wider and wider as data from each frame is combined. The frames are never shifted in Y, so that image never grows in size in that direction.
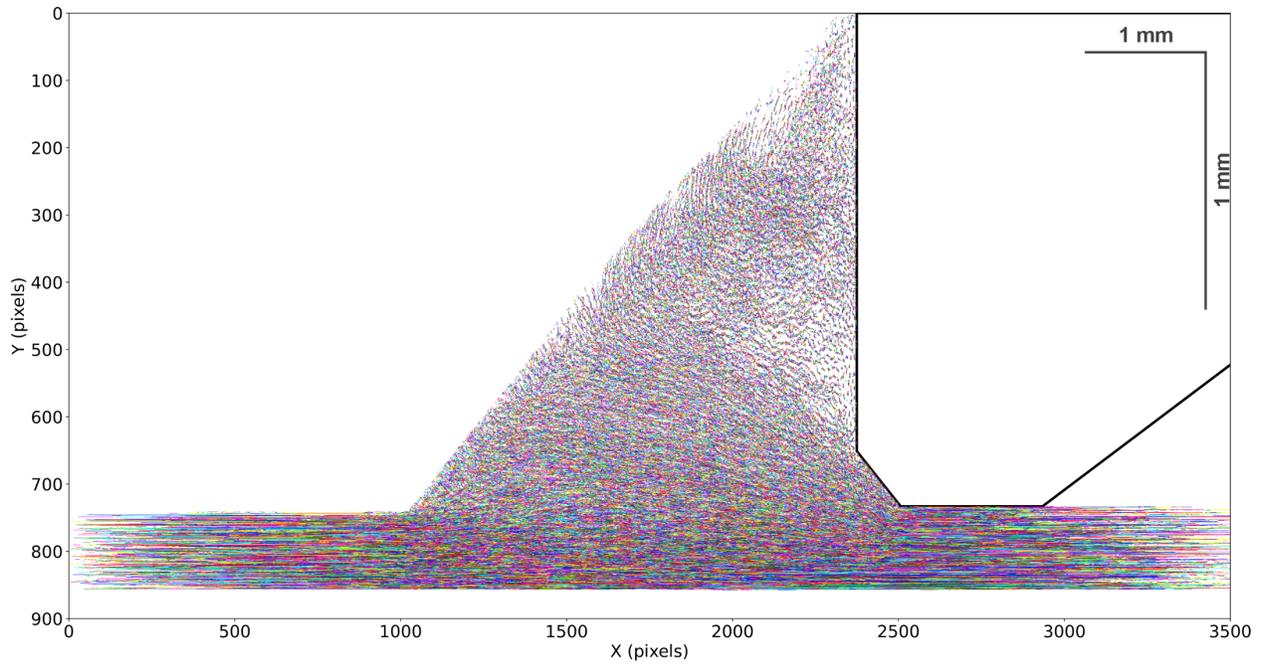
**Fig. 17. Track data (X, Y, Frame Number) triples projected onto the X, Y plane. The field of view is 9.49 mm x 2.44 mm.**

A video of this plot, merged with the original video, is available at [14]. **Figure 18** compares three frames from the original video (top of Fig. 18) to the corresponding three frames of the "merged" video (bottom of Fig. 18). Since all X, Y locations are relative to the moving blade, the stationary camera images appear to move across the screen during the video.
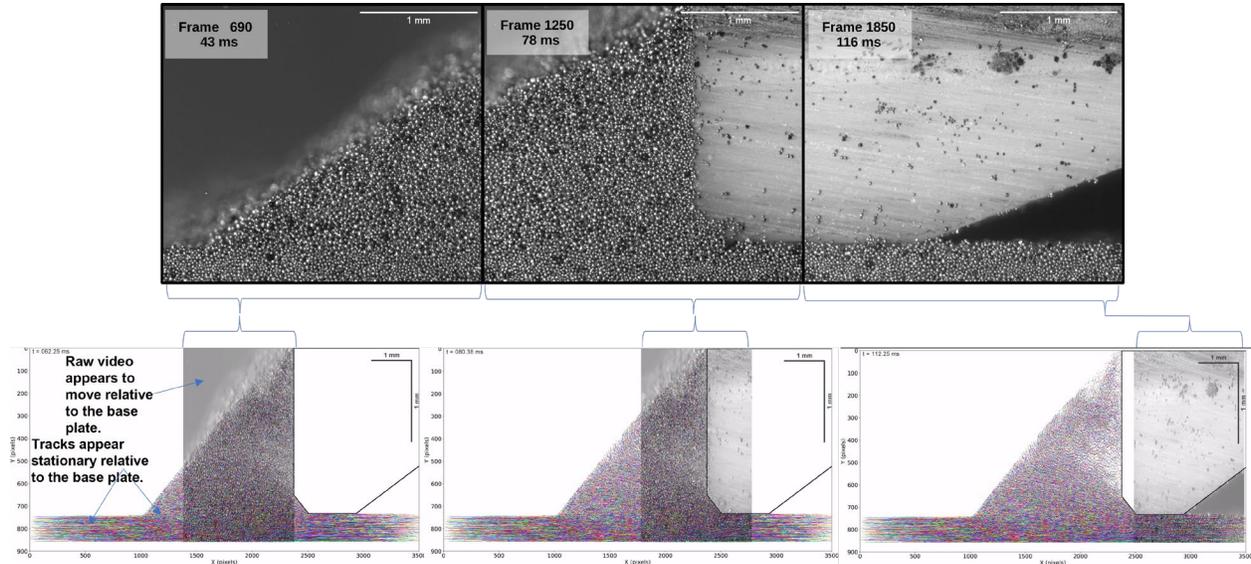
**Fig. 18. Comparing three frames from the original video (top of the figure) to the corresponding three frames of the "merged" video (bottom of the figure).**

To better understand this data, consider **Fig. 19** and **Fig. 20**. These show the same data as **Fig. 17**, but projected to the (X, Frame Number) plane. Since Frame Number scales with time, these figures illustrate the time-dependent nature of how the data was captured. This is why assumptions of time invariance in how the powder spreads must be made. Notice that the tracks form a diagonal band from the lower left quadrant to the upper right quadrant. Since X location is relative to the moving blade, any stationary particle (relative to the PST base plate) will lie precisely along the line determined by the blade speed and frame rate.

Frame 1250 of the video is shown in **Fig. 16B**. In **Fig. 19**, a line is projected from Frame number 1250, across to the diagonal band, then projected down to the X axis. Thus, what the camera sees during Frame 1250 can only effect X = 1750 pixels to X = 2750 pixels in the plot. A similar analysis is shown in **Fig. 20**. In this case, all data used to directly determine the tracks at X = 1750 pixels occurred between Frame 550 and Frame 1250, which is a 44 ms period of time.
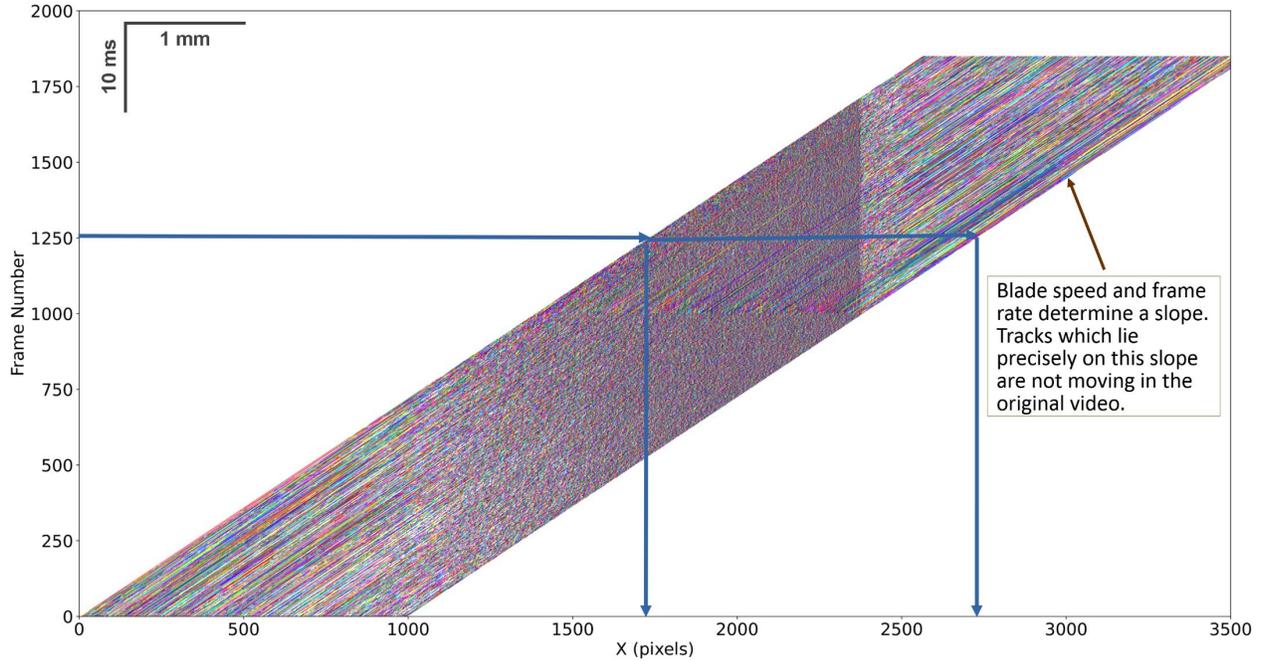
**Fig. 19. Experiment S2, with tracks projected to X, Frame Number plane. The projections along the blue lines show how Frame 1250 is only "seeing" X = 1750 pixels to X = 2750 pixels. The field of view is 9.49 mm x 125 ms.**
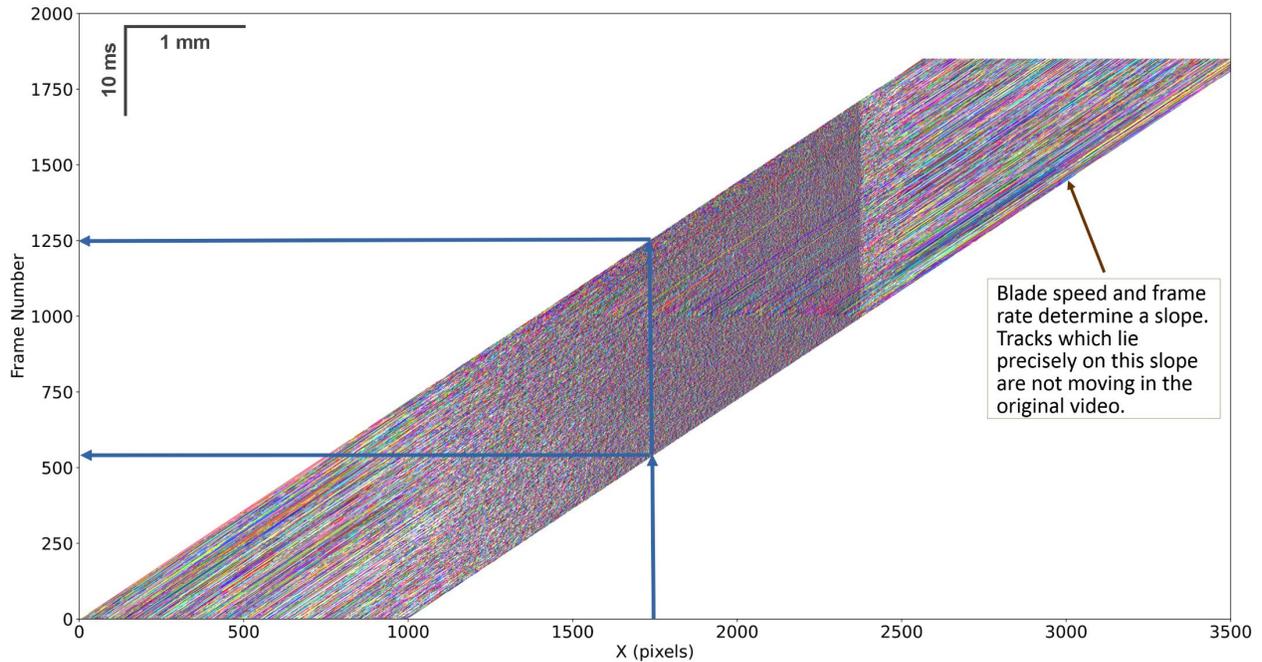


**Fig. 20. Experiment S2, with tracks projected to X, Frame Number plane. The projections along the blue lines show how location X = 1750 pixels is only "seeing" what is happening from Frame 550 to Frame 1250. The field of view is 9.49 mm x 125 ms.**

As discussed in Section 2, one can use these tracks to compute vector fields for flow characterization. This can be determined from the track data in many ways, but two similar ways are shown in **Fig. 21** and **Fig. 22**. The direction of each vector represents the typical direction of the tracks in that neighborhood. The length of each vector is a fixed constant, and the typical velocity represented by the color around that vector. Velocity relative to the stationary camera is shown in **Fig. 21**. Since the base plate of the testbed is also stationary, this plot is especially informative near the base plate, since it may indicate what frictional forces are likely between the particles and the plate. Velocity relative to the moving blade is shown in **Fig. 22**. This plot is especially informative near the blade, since it may possibly indicate what frictional forces are likely between the particles and the blade. Many other plots are possible. For example, a velocity could be relative to other velocities in a local neighborhood. This would highlight local eddies in a flow and might be useful in understanding particle – particle interactions.
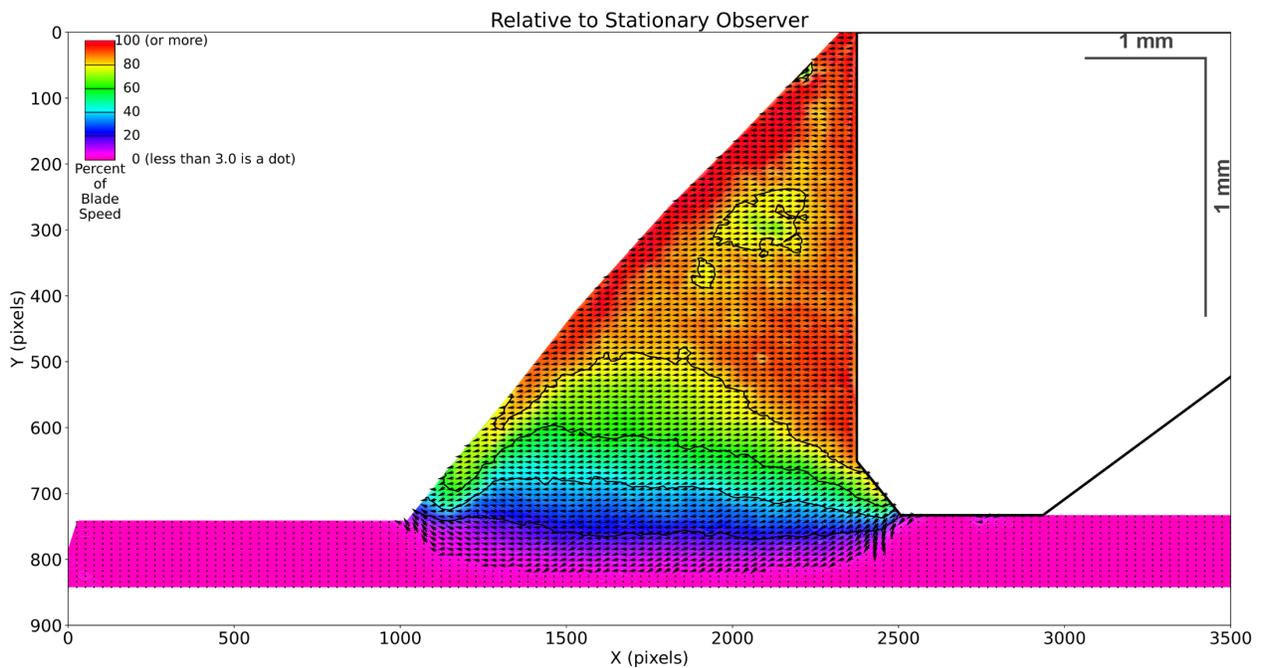


**Fig. 21. Plot of vector field with vectors indicating velocity relative to the stationary camera. The field of view is 9.49 mm x 2.44 mm.**
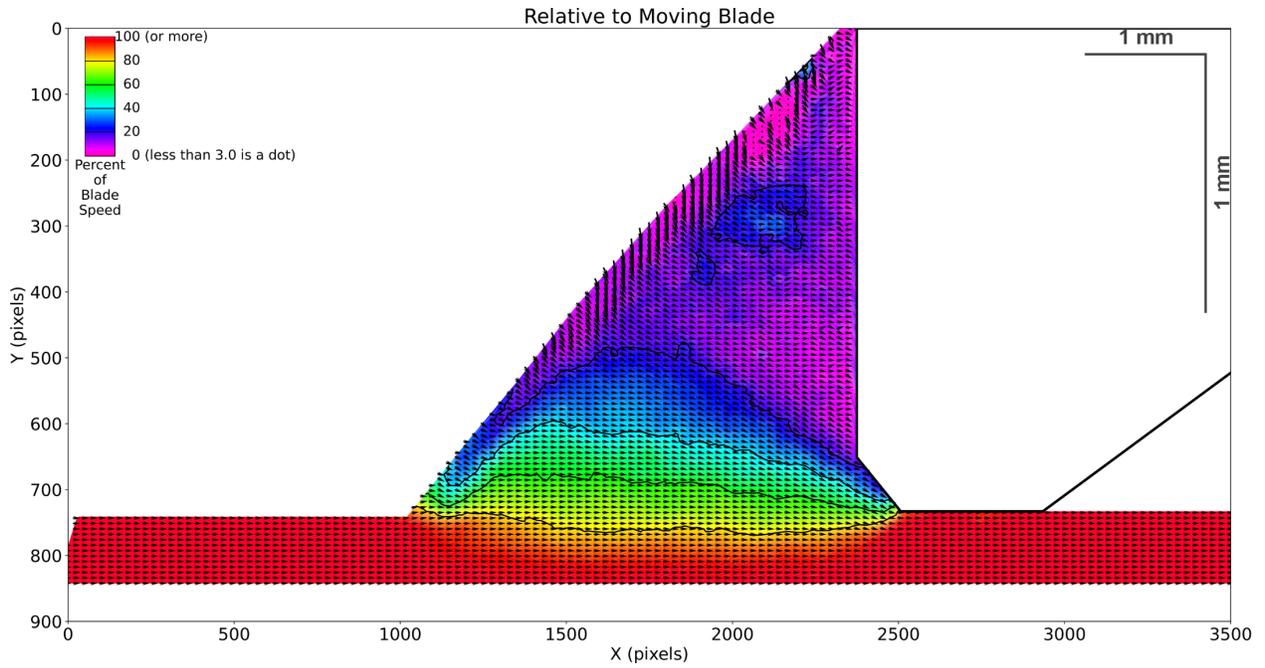
**Fig. 22. Plot of vector field with vectors indicating velocity relative to the blade. The field of view is 9.49 mm x 2.44 mm.**

## 4. Conclusions

Initial results are presented from a powder spreading testbed that provides the ability to image the cross-section of a rigid recoater blade, a substrate, and powder. Both track data and vector field data were determined from this and look very informative.

It should be noted that **Figure 15** shows that the particles are escaping from the space between the recoater blade and the glass. What matters in the PST is whether the particles near the glass surface are getting directly pushed by the recoater-blade or not. However, this paper is about explaining the method of particle tracking. The PST needs further work.

Both the hardware and software are undergoing revisions. While the hardware yielded very interesting results, there are some limitations. For example, typical powder recoaters have a much longer spread length than this testbed. Also, the camera is currently mounted on the base. Plans are being implemented to allow for a recoater-mounted camera, which would allow for an analysis that does not require the assumption of time invariance of the powder spreading process.

We would like to mention that the vector field plots shown here in **Figure 21** and **Figure 22** represent the specific PST with its setup parameters and the specific material. The ongoing work is focused on the PST setup acceptability criteria, and the setup repeatability and reproducibility. After that, the repeatability of the data generated will be verified. The sensitivity of the PST-generated results to the recoater-blade height, the recoater speed, powder material, particle size distribution, and particle density will be tested.

# References

[1] Whiting, J., Whitenton, E., Das, A., Tondare, V., Fox, J., McGlauflin, M., Donmez, A., Moylan, S., Powder Spreading Testbed for Studying the Powder Spreading Process in Powder Bed Fusion Machines, NIST Advanced Manufacturing Series 100-56, https://doi.org/10.6028/NIST.AMS.100-56 .

[2] Adrian, R.J. (1991) Particle-imaging techniques for experimental fluid mechanics. Annual Review of Fluid Mechanics, 23(1), pp.261-304.

[3] Jesuthasan, N., Baliga, B.R. and Savage, S.B. (2006) Use of Particle Tracking Velocimetry for Measurements of Granular Flows: Review and Application—Particle Tracking Velocimetry for Granular Flow Measurements—. KONA Powder and Particle Journal, 24, pp.15-26.

[4] Adrian, R.J. and Westerweel, J. (2011). Particle image velocimetry (No. 30). Cambridge University Press.

[5] Lueptow, R.M., Akonur, A. and Shinbrot, T. (2000) PIV for granular flows. Experiments in Fluids, 28(2), pp.183-186.

[6] Tischer, M., Bursik, M.I. and Pitman, E.B. (2001). Kinematics of sand avalanches using particle-image velocimetry. Journal of Sedimentary Research, 71(3), pp.355-364.

[7] Gerald C Holst, Holst's Practical Guide to Electro-Optical Systems. JDC Publishing, ISBN 0-9707749-1-5.

[8] Knime website. https://www.knime.com/knime-analytics-platform . (Accessed 10/25/2023).

[9] Olivo-Marin, J.C. (1996) Extraction of spots in biological images using multiscale products. Pattern Recognition 35.

[10] Ershov, D., Phan, M.-S., Pylvänäinen, J. W., Rigaud, S. U., Le Blanc, L., Charles-Orszag, A., Tinevez, J.-Y. (2021) Bringing TrackMate into the era of machine-learning and deep-learning. bioRxiv. doi:10.1101/2021.09.03.458852

[11] Tinevez, J.-Y., Perry, N., Schindelin, J., Hoopes, G. M., Reynolds, G. D., Laplantine, E., Eliceiri, K. W. (2017) TrackMate: An open and extensible platform for single-particle tracking. Methods, 115, 80–90. doi:10.1016/j.ymeth.2016.09.016

[12] ImageJ website. https://imagej.net/plugins/trackmate . (Accessed on 3/28/2022).

[13] Knime website. https://nodepit.com/node/org.knime.knip.base.nodes.seg.waehlby. WaehlbySplitterNodeFactory%23Waehlby%20Cell%20Clump%20Splitter . (Accessed 10/27/2023).

[14] Link to "merged" movie, https://www.nist.gov/video/merged-movie-showing-both-high-speed-video-metal-powder-spreading-using-powder-spreading

[15] Link to "raw" movie, https://www.nist.gov/video/high-speed-video-metal-powder-spreading-using-powder-spreading-testbed-pst-nist-0