UNIVERSITY OF TWENTE.

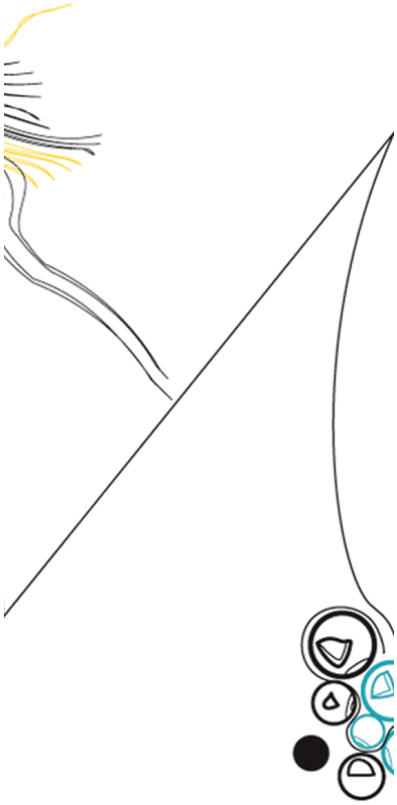
Faculty of Engineering Technology

Particle detection and tracking in avalanches

Dieuwert A. van den Broeke

Master Thesis

February 24, 2023



Supervisors:
prof. dr. ing. B. Rosic
dr. A. Jarray

Applied Mechanics and Data Analysis Group
Faculty of Engineering Technology
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

To prevent avalanches one requires their reliable prediction. For this purpose one has to study the avalanche dynamics and conditions under which such dynamics occurs. In this thesis the avalanche of granular materials is studied from an experimental and data-driven perspective. The data are collected by taking images of the rotating drum filled by a granular material with the help of the high speed camera. Initially, the individual particles in images are detected by use of circular Hough transform and the existing expert knowledge. Furthermore, the particles are tracked by use of the nearest neighbor method in combination with the unscented Kalman filter. This approach has shown to have good performance as one may easily account for missing particle detections and uncertainties about particles exact position. Thus, the initial particle position is modelled as uncertain, and its evolution is further predicted using the coordinated turn model. With the help of the proposed tracking method the particle position is further updated by assimilating the measurement data with the prior knowledge. The method is evaluated on unseen data set, and is able to predict the global avalanche dynamics such as angle of repose, and movement of particles that is not too stochastic. On the other hand, prediction of the individual movement of particles in the middle of the avalanche is shown to be more challenging.

Contents

1 Introduction

Granular flows and avalanches are currently subject of extensive studies in various research disciplines and are discussed in physics, geophysical, powder technology and other academic works due to their practical importance[1, 2, 3, 4]. The physical behavior of granular materials and the corresponding avalanches is extremely complex, which makes generation of their physics-based mathematical models difficult. Due to the non-linear particle movement, avalanches are a challenge that is difficult to predict and manage. In addition to being affected by gravity, the movement is also affected by interactions between the particles and other influencing external factors[5]. The features of granular material set it apart from other "regular" solids, fluids, and gases. These materials actually behave somewhat like solids, fluids, or gases depending on the external influencing factors[6, 7, 8, 4]. Even if the external driving mechanisms are constant, the processes themselves can be either continuous or intermittent.

Avalanche is a rapid flow of material (snow, ice, rocks, debris) that falls down the side of a slope. On a sloop, an avalanche is triggered by loose material breaking from the surrounding materials, and swiftly gathers, carrying more material down the slope[5]. A granular avalanche happens when a particle accumulation's slope exceeds its maximum angle of stability; after that, the angle starts to fall until it reaches the angle of repose. When particles are at rest, their velocity is almost zero. However, when an avalanche happens, their speed reaches a peak, and then it progressively drops as the avalanche continues. An avalanche keeps sliding down the mountain until it accumulates where the mountainous terrain flattens. The time evolution of a slope angle and particle velocity thus define the dynamics of avalanches. All avalanches have a starting zone, which is their highest point, an avalanche track, which is their natural path downhill, and a run-out zone, which is where they halt by piling up granular debris. The speed of the descending mass, made of dry particles can reach of 130 km an hour, whereas a liquid substance can reach a speed of 15 to 60 km an hour[9]. This is why avalanches cause human mortality in addition to material losses; 150 persons per year die in Europe and North America as their direct consequence[10]. Granular flows can impact geophysical formations, such as the formation of landscapes by landslides.

A granular avalanche is a gravity-driven free surface flow of continuously granulated media down a steep slope that is typically caused by the instability of a granular layer. Avalanche are characterized as "a transient, three-dimensional gravity-driven free surface motion of a mass system made up of an assemblage of granular fragments initiated by an instability of granular layer and flowing down to the run-out zone on an arbitrary steep topography with varied surface resistance"[11]. The main characteristic of an avalanche is that its surface moves faster than its base, functioning as a kinetic sieve, thus sorting the granular particles. Rapidly, an inverse grading of the particles occurs in which the larger particles overlie the smaller ones, generating a two-layered shear band.

Granular avalanches have a number of applications in technology and engineering. A better understanding of the flow of granular materials can lead to development of new technologies and improved design practices that benefit society and the environment, the improvement of industrial processes, and better methods for detecting and monitoring granular avalanches. Typical application examples are: disaster risk assessment, infrastructure design, agriculture practices, and geotechnical engineering. As can be seen avalanches can be classified as destructive or productive. The first type occurs during industrial processes and can have a detrimental impact on output because granular materials are involved in uncontrolled, swiftly moving flows that may harm infrastructure or interrupt human operations. A few examples can be seen in the agriculture and food processing industries, pharmaceutical plants, mining caves, and construction industries which require proper measures to prevent destructive granular avalanches. On the other hand, constructive avalanches are used for the production purpose where the flowing properties of granular material are utilized to achieve specific goals. Application of granular avalanches can improve productivity in industries by mixing and blending granulars to achieve homogenous products, and thus improve the process quality and efficiency while reducing costs. Productive industrial utilization of granular avalanches is used in highly lucrative pharmaceutical and metallurgic industries (coating, granulation etc)[12].

Granular systems, despite their everyday familiarity, have become paradigmatic systems of complexity. Therefore an understanding of the possible causes and dynamics of avalanches and development of technology for their prediction and prevention are needed. Tracking particles in an avalanche helps to understand and predict the behavior of the flow, including its speed, direction, and impact force. This can provide insights into the behavior of granular materials under different conditions, which can inform the design and engineering of structures and equipment in industries such as construction and transportation. By tracking granular avalanches, industries can gather data to develop models for predicting and preventing future avalanches. This can help to reduce the risks associated with avalanches and improves the overall safety and efficiency of operations.

State of the art

A common numerical method used to predict avalanche is the Discrete Element Method (DEM)[13, 14] in combination with physics based equations[15, 16]. DEM is a numerical simulation technique[17] used to model and analyze the behavior of granular materials and particles. It has become a valuable tool for researchers and engineers in the fields of particle mechanics and granular materials. The main advantage of DEM is its ability to handle complex geometries, material properties, and boundary conditions, which makes it well suited for simulating real-world systems. Additionally, DEM can be used to model the behavior of a wide range of particle sizes and can account for various types of particle interactions, such as friction, adhesion, and breakage[14, 18, 19, 13]. By considering each particle as an individual element, DEM can provide particle-level resolution, which is important in understanding the behavior of granular materials. However, approach based on DEM is limited in its use, because it is challenging to validate the dynamic behaviour of the particles, as there is often a lack of experimental data available for comparison. One way used to validate DEM model is by the angle of response. However, this validation technique cannot guarantee good mathematical model when it comes to particles interactions. Despite DEM ability to handle complex geometries and boundary conditions, the accuracy of its results is not fully known. Further, the method is computationally intensive and requires large amounts of computing resources, especially when simulating large-scale systems or systems with many particles[20, 21]. Other studies approach the matter from an experimental perspective. The revolving drum or slanted bed is a straightforward and useful tool that has been widely utilized in research to analyze the flow of granular materials[22, 23, 24]. In such a case dynamic properties are extracted from high speed camera images. Thus, this approach is purely relying on the computer vision, and does not take into account the experts knowledge. In other words, the avalanche dynamics can be described in a purely data driven way, i.e. by using machine learning[25] .

When the data-driven approach is used to track the particles it is necessary to detect the particles first. This task is usually achieved by using one of the computer vision algorithms. In this aspect the image quality has to be good as well as visibility of the particles. The results of the detection are better the more clearly the particles are visible. Particle detection can be done in a variety of ways, one of which is the superresolution[26]. The concept behind super-resolution is that a high-resolution image or image sequence can be created by combining a series of low resolution (noisy) photos of a scene. As a result, given a collection of observed photos with lesser resolution, it tries to rebuild the high-quality original scene image. Although this seems to be a viable alternative for enhancing the image quality, superresolution methods can be computationally demanding and demand significant processing resources. Further the quality of the resulting superresolved image is limited by the quality of the original image. If the original image is noisy, blurry, or has other issues, these problems amplified in the superresolved image. In some cases, increasing the resolution of an image can result in a loss of information or a degradation in image quality. Machine learning techniques, such as Convolutional Neural Networks (CNN) and its numerous variants[27, 28, 29, 30, 31], are the most widely used method for object, i.e. particle detection[25]. As these techniques rely on unsupervised learning approach, one requires large amount of data. Here under unsupervised learning it is meant that the particle positions are not manually labeled and fed to the algorithm for further learning. Instead, the algorithm has to find positions by itself and learn the appropriate model for their prediction. However, due to their transparency and dense packing, the glass particles' characteristics, such as their contour and color, are rarely visible, and thus the poor image quality makes unsupervised learning difficult. Thus, unsupervised machine learning is not a viable option to detect the particles. The methodologies that have been previously discussed center on the detection of particles using simply pictures. However, one could enhance particle detection if a depth sensor was utilized in addition to an RGBB camera[32]. In the absence of the depth sensor, one may use the so-called gradient grids. In order to detect whether the object(s) of interest are present in the test image, one first creates a template using only the gradients of the object on a background with no texture. The approach makes no assumptions on the type of the background, or the shape of the transparent object. Due to the absence of a depth sensor and the fact that the background of the particles is made up entirely of transparent particles, neither approach can be utilized to detect particles in the drum. In literature one may also find a special algorithm for detecting circular-shaped objects in low resolution images[33]. This method is used for detecting tanks in satellite images, and does not work for detecting particles in a drum. The primary cause of this, alike with satellite photographs, is once again a translucent background that does not provide enough contrast to the transparent objects. One solution to this problem is to train a machine learning model using a fictitious data set. Random packing[34] is one method of carrying this out. Examples are jammed disk packing, voronoi packing, Torquato-Jiao, and Lubachevsky-Stillinger type of packing[35, 36, 37]. If the initial conditions, i.e. a few particle placements, are known, then the drum can

be mathematically filled with particles using one of previously described methods. Thus, one has to start with identification of initial conditions. For this purpose,, one may manually identify the drum boundary and a few particles in an image, and then fill the empty area using random packing. The problem of the detecting contours of an object in an image can be solved by using the Hough transform algorithm[38]. Since the particle geometry is known a priori, the contour of a particle is a circle, so a variant of the Hough transform can be used. This variant is the Circle Hough Transform (CHT)[39, 40], and is specialized in detecting circle shaped contours. An advantage of the CHT is the capacity to accurately detect the particles even when they are partially obscured or distorted. It can also handle multiple particles and overlapping particles. Also the CHT is relatively robust to images noise and is tolerant to missing pieces in the feature boundary. So if not the full contour of the particle, a circle is visible. Further the CHT can handle particles of different sizes, orientations, and brightness, making it suitable for a wide range of granular avalanche experiments. Finally the CHT is a simple and intuitive technique which makes it easy to implement and understand. A disadvantage of the CHT is the sensitivity to the choice of parameters used in the algorithm, such as the threshold for detecting local maxima in the parameter space. This can result in missing some circles or detecting non-existing circles. The CHT can be computationally expensive for large images or images with many circles. The CHT's usefulness may be constrained for images with weak edges or poor contrast because it depends on an edge detection technique used to pre-process the image.

Once the particles are detected in an image, one has to track them in time, i.e. detect them in subsequent images in time. This can be achieved by using commercial software such as FIJI Image or particle image velocimetry[41, 42]. However, these algorithms work with denoised images. Therefore, the focus in this thesis is on the probabilistic type of trackers that are based on Bayes rule[43]. One example of simplified Bayes rule is the Kalman filter[44] that is based on the assumption that the particle dynamics is linear, and that the observation noise is of Gaussian type. The precision, efficiency, suitability for linear systems, and capacity to handle noisy observations and unknown linear dynamics are the benefits of utilizing the Kalman filter. The Kalman filter's drawbacks include its inability to handle non-linear systems, sensitivity to initialization, and potential sensitivity to outliers. To overcome these drawbacks, one may utilize particle filter[45, 46]. This is a non-parametric method for tracking particles by describing their position initially as uncertain over the probability distribution that is further sampled. This algorithm can handle non-linear systems, does not require knowledge of the system model, is robust against initialization errors, and can model the uncertainty in the measurements. The drawbacks of this filter are its high computing cost, the potential for sample degeneracy if improperly initialized, and its sensitivity to the proposal distribution selection. Similarly to particle detection, the object tracking can be also done by use of Convolutional Neural Networks (CNNs)[47, 48, 49, 50, 51] to track objects in real-time. The method can handle complex object appearances and occlusions, can learn from data, is fast, and accurate which are all advantages. However, this approach has drawbacks such as the need for a substantial amount of training data and sensitivity to changes in object scale, viewpoint, and illumination. The possible lag in training data makes this strategy ineffective. Finally, one may also utilize the optical flow approach[52, 53]. This is a motion estimation technique that tracks the movement of pixels between consecutive frames. The advantages are that it is computationally efficient, robust to illumination changes, and can handle small object motions. Disadvantages of this technique are that it may fail in cases where objects are occluded or there are significant background motions. Because the particles are closely packed and the background is not constant this method is therefore not an option.

Based on the previous discussion one may conclude that the particle detection and tracking in avalanches can be challenging. This thesis continues from a previous study[54], by making a new approach for the particle detection and tracking. Namely, in this work the data (images) are not labeled and have not clearly visible features. As a result, the most of previously described supervised and unsupervised machine learning techniques cannot be used for detection and prediction. For the detection of the particles a classic computer vision technique, the CHT is used, because with this method it is possible to search for the a priori known features of the particles; the shape and size of the particle. For the tracking of the particles the Kalman Filter (KF) is most suitable since the KF includes uncertainties in the model, so that this method is not dependent on the accuracy of the assumed particle dynamics model nor measurement noise. The way the KF represents the underlying system and deals with uncertainty sets it apart from other tracking techniques. The KF predicts the state of the particle in the following time step by using a mathematical model of the particle motion. Then, taking into consideration the uncertainty in both the model and the observations, it revises this forecast using the most recent measurements. This results in a more accurate estimate of the state of the particle compared to other tracking methods that do not take into account the uncertainty in the measurements. Since the particle movement is non-linear, the straightforward KF is no longer applicable. Therefore, the nonlinear-version of Kalman filter, also known as Unscented Kalman filter[55], is further

utilized. This method is based on the Unsented Transform[56] and offers higher performance at an equal computing cost to the extended Kalman filter[57] that linearzies the system dynamics.

Research objectives and outline

In this study, the potential for using machine learning[25] to analyze and track individual avalanche particles is explored. This study's goal is to look at how it is possible to track and detect (glass) particles in an avalanche simulation experiment. The behavior of granular material in avalanches is studied using data (pictures) from a rotating drum experiment. Then, different techniques for detecting and tracking specific particles are contrasted to see which one is the most effective. The ultimate goal of this project is to assess the methods for detecting, tracking, and forecasting individual particles in granular avalanches and to provide a framework for related future studies.

This study is divided in the following chapters: chapter 1 coveres the motivation and goal of the research, in chapter 2 are described the detection and tracking of particles with machine learning are described, chapter 3 presents the numerical results. Chapter 4 concludes the work, whereas Chapter 5 contains further recommendations.

2 Detect and track particles with machine learning

In this study the data are collected by observing the avalanches of dry particles in the half filled rotating drum. Dry partciles have a low viscosity and the highest particle velocity and thus are chosen for this experiment. Images of the rotating drum are recorded with a Canon Legria HFG40 camera, operating at 450 FPS. The experimental design used in this research is demonstrated in a schematic view in Figure 1 below. The drum consists out of a cylinder with a radius of $R = 60.5mm$ and a depth of $w = 22mm$. The drum side surfaces are made of plexiglass plates (PMMA) of a 5mm thickness. This ensures further observations of particle dynamics. To prevent sticking of particles at the plates these were coated with fluorinated ethylene propylene (FEP). The cylinder of the drum is made of poplar wood. The drum is partially filled (filling level 35 percent of its volume) with monodisperse borosillicate glass particles with a density of $\rho = 2500\frac{kg}{m3}$. The radius of the particles is $r = 2.00mm$. Tests are conducted with a constant rotation speed in various sets ranging in the interval $0.5 - 45rpm$. When utilizing dry particles of low viscosity, intermittent avalanches start to happen at a rotating speed of $0.5rpm$. Avalanches occur periodically at a low rotational speed and when a high rotational speed is used a continuous flow is created.

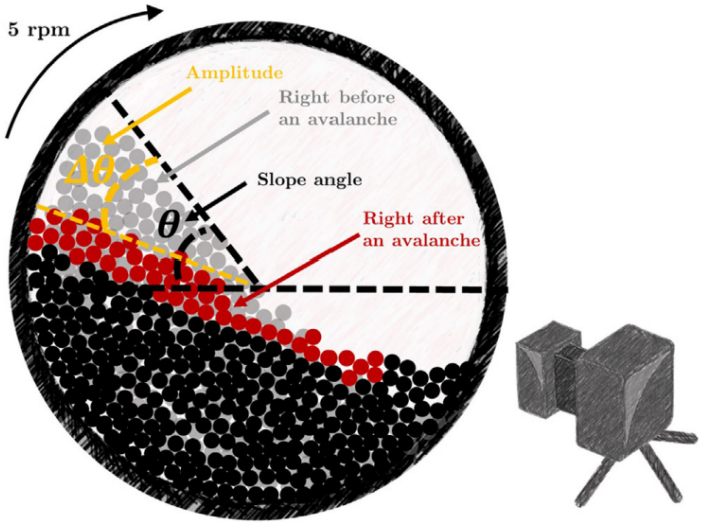


Figure 1: Schematic representation of granular avalanches in a rotating drum

Rotating drum experiments, in which a cylinder partially filled with granular material is rotated, are an ideal laboratory setup for observing the distinct facets that a granular moving system may possess. Particles in rotary drums are used to mimic granular avalanches with a continuous flow. Avalanches occur periodically at a low rotational speed and when a high rotational speed is used a continuous flow is created. Images of a rotary drum filled with dry granular materials flowing continuously, to study the flow of the individual particles are used. Using computer vision to predict an avalanche, one has to analyze images as shown in Figure2 that are collected during an experiment as described above.

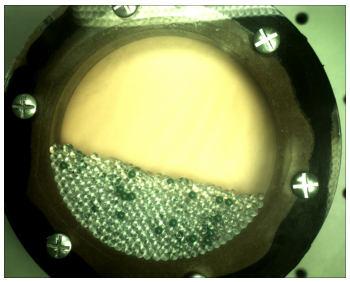


Figure 2: Image of granular avalanches in a rotating drum

The previously described experiment can be made by using only one layer of particles, and the light source in

the back of the drum as described in[58]. In such a case, the detection of the particles is quite straightforward. However, such experimental setup does not correspond to the real avalanches. Therefore, in this experiment several layers of particles are used, and the light source in front of the drum. Thus, the data that are collected by a high speed camera are similar to the one shown in Figure2. As the images are taken by a high-speed camera, one can not manually analyze images as such an analysis has to run in a real-time in industrial applications. On the other hand, if the analysis is to be done offline after the images have been collected, then manual inspection would be time consuming due to their large number. Therefore, the goal of this thesis is to build the computer-based method that would prevent any manual work. For this purpose, one has to design an algorithm that can detect and track individual particles in the granular media in the drum. Next to this, the aim is to also track the angle of response as global avalanche feature.

2.1 Particle detection

The detection of particles in images is often a cumbersome task, and its difficulty strongly depends on the shape of particles and the type of their movement. The geometrical shape, if known a priori, can help significantly in detecting particles or objects, as such a learning problem is supervised. In such a case one has to find the object with the prescribed geometrical features in the image. This can be difficult if several objects with the same geometry are present in the image especially if they have different dimensions, or if the edges of objects are not smooth. The motion of particles on the other hand can make the detection problem complex. If particle movement is deterministic and smooth then the detection of particles on the subsequent images is relatively easy and can be extrapolated from the previous images. In such a case, the particle on the next image will be in close vicinity of its position in the previous image. However, if the particle movement is stochastic and noisy, this problem becomes more difficult as the particle may suddenly jump and change its direction of motion. In such a case the position of the particle cannot be easily found.

In the problem considered in this thesis the particle shape is known a priori: all particles have identical size and are circles. The particle geometry is thus characterized by two features: the position of its center, and the radius of the particle. For the particle detection only the position of its center needs to be known, as the radius is known a priori. On the other hand, the tracking of particle movement is challenging. Most of the granular particles in the drum move according to the circular path as observed in [54]. However, the particles that are part of the avalanche are not smoothly moving, and are characterized by a stochastic movement as later described in the text.

The center position of the particles in the drum is found using the particle contours, and the known radius. As a result, the center detection largely depends on the particle contour being detected. However, in 3D packing problems as the one considered in this thesis, this can be difficult. The contours of the particles in the front 2D portion of a 3D packing problem may be challenging to distinguish from the contours of the particles in the adjacent 2D section. Next to this, the fact that the particles are composed of glass and reflect the light from the camera lamp makes it much harder to view the particles properly. Because the reflection is more prominent, it is harder to see the right particle outlines. Therefore, to partially compensate for this problem a few of green particles, also known as tracer particles, have been added to the drum. Both type of particles then allow use of color features next to the geometrical features for the particle detection. An image is represented by its height and width, which are determined by the number of pixels. A pixel represents a specific value of a point in an image. This value represents the intensity within a given range. Grayscale and RGB are the two most common image representations. Each pixel in Grayscale has one integer with a value between 0 and 255, where 0 is black and 255 is white. Each Red, Green, and Blue (RGB) pixel is made up of three integers with values ranging from 0 to 255. The intensity of the colors Red, Green, and Blue is represented by the values [59]. Thus, to find a particle contour one may use the pixel intensity. A contour is a curve that connects all pixels with the same intensity along a particle boundary. An algorithm that can achieve this is the edge detection algorithm. The Canny edge detection method has been chosen as the best performance edge detection method for creating the edge map of the particles in the drum for several reasons. First the method uses a multi-stage algorithm that is able to remove noise from the image, which results in fewer false edges being detected. Next to this, the method uses gradient information to detect edges, which further allows detection of edges that are not necessarily the brightest or darkest pixels in the image. By hysteresis thresholding one may also preserve edges that are connected to strong edges and suppress the remaining weak edges. Canny edge algorithm also gives the possibility of tunning this threshold, and is computationally efficient. All these factors contribute to the Canny edge detection method being considered one of the best edge detection methods available. An edge map, is a representation of an image that highlights the contours of the particles within

the image. It is created by applying the Canny edge algorithm[60] as further described in the text, which analyzes the image and detects areas where there are sudden changes in pixel intensity. These areas of sudden change are considered edges, and are typically represented in the edge map as white pixels on a black background. The Canny edge detection algorithm consists of several steps, including image smoothing, gradient calculation, non-maximum suppression, and hysteresis thresholding. The first step, image smoothing, is used to reduce noise in the image. This is done by convolving the image with a Gaussian filter, which helps to smooth out small variations in pixel intensity and reduce noise in the image. This improves the accuracy of edge detection by reducing the chance of detecting false edges caused by noise. The next step is the calculation of the gradient of the image in each pixel, which is used to highlight regions of intensity change, including the contours of the particles. This is done using the Sobel operator[61], which calculates the gradient at each pixel of the image in the x and y directions separately. The magnitude of the gradient is then calculated as the square root of the sum of the squares of the x and y gradients:

$$G = \sqrt{G_x^2 + G_y^2}. \tag{1}$$

And the direction of the gradient is calculated as the arctangent of the y gradient divided by the x gradient:

$$\Theta = tan^{-1}(\frac{G_y}{G_x}). \tag{2}$$

After gradient calculation, non-maximum suppression [62] is applied to thin the edges. Once the gradient's magnitude and direction have been determined, the entire image is scanned to remove out any extraneous pixels that might not be the edge. For this, it is determined whether each pixel is a local maximum in its vicinity in the gradient's direction. Pixels with a gradient magnitude less than the pixels in the direction of the gradient are removed. Figure 3 shows an example where pixel A is on the edge and where pixel C and B are in the gradient direction.

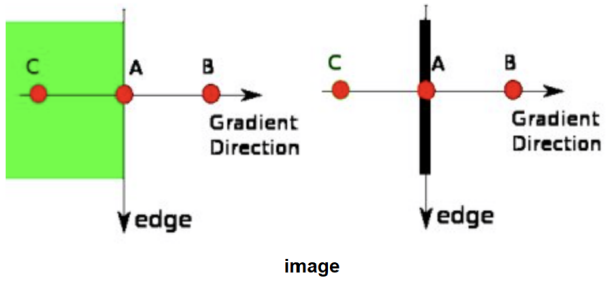


Figure 3: Illustration of Non maximum supression

The final step of the Canny edge detection algorithm is hysteresis thresholding. This step is used to remove weak edges and leave only the strongest edges in the image. Hysteresis thresholding is performed by applying two threshold values to the gradient magnitude, a high threshold and a low threshold. Pixels with a gradient magnitude greater than the high threshold are considered to be strong edges, pixels with a gradient magnitude less than the low threshold are considered to be non-edges, and pixels with a gradient magnitude between the two thresholds are considered to be weak edges. The weak edges that are connected to strong edges are considered to be edges, while the remaining weak edges are removed. For illustration an example in Figure4, point A is a strong edge and point C is a weak edge but is linked to point A and is therefore seen as an edge. Point B is a weak edge and not linked to a strong edge and is therefore removed. An example of an edge map of an image of the particles in the drum is shown in Figure5.

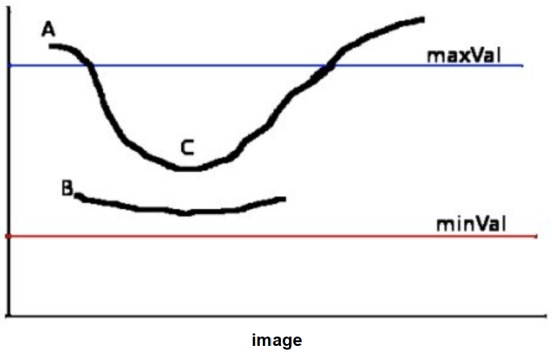


Figure 4: Illustration of hysteresis thresholding

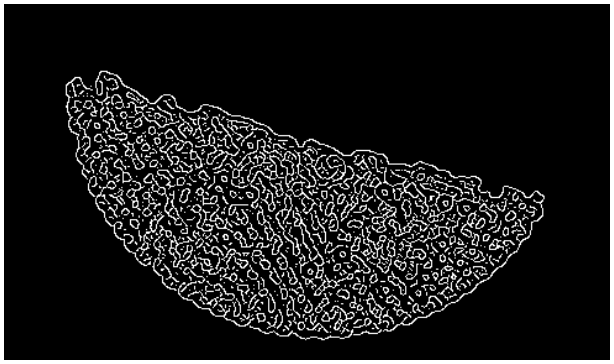


Figure 5: Canny Edge, edge map

Once the edge detection is performed, one may continue with the detection of circular objects by Circular Hough transform[39, 40] as described further. Mathematically, the 2D circle can be described by:

$$(x-a)^2 + (y-b)^2 = r^2 \tag{3}$$

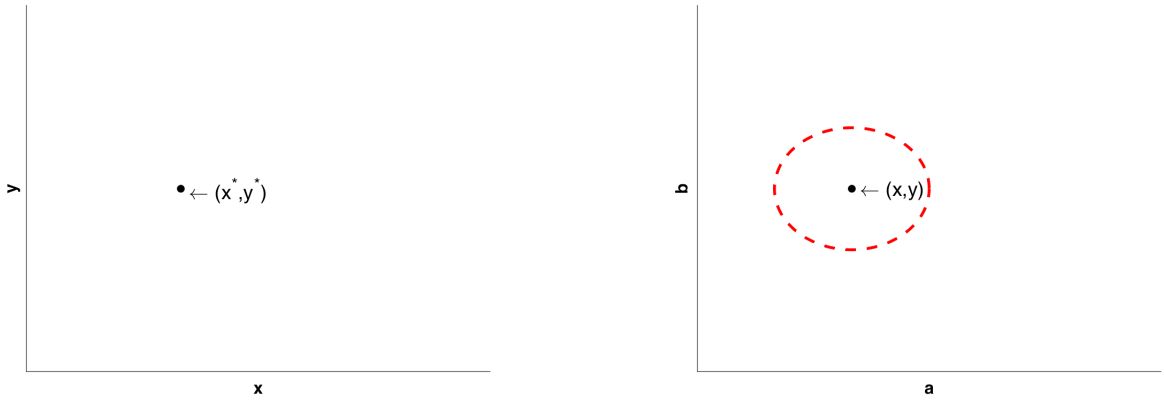
in which a and b represent the center coordinates of a circle and r is the radius. Therefore, three parameters are required to fully describe a circle. Since the radius of the particles is known a priori the parameter space is reduced to two dimensions, respectively a and b. Following this, each coordinate of the circle with the center (a,b) in the image can be expressed as:

$$\begin{aligned} x &= a + rcos\theta \\ y &= b + rsin\theta \end{aligned} \tag{4}$$

with θ being the full angle of circle. From previous equations one can express the parameters of the circle:

$$\begin{aligned} a &= x - rcos\theta \\ b &= y - rsin\theta. \end{aligned} \tag{5}$$

As sinus and cosinus are periodic functions with the switching signs one can actually say that the second pair of equations also represent the circle with the center in (x, y). Thus, the circle with the center in (a,b) in image is same as circle with the center (x, y) in the parameter space. In other words, to each coordinate (x^*, y^*) in the image one can assign the circle with a radius in the parameter space (a, b), see Figure 6.

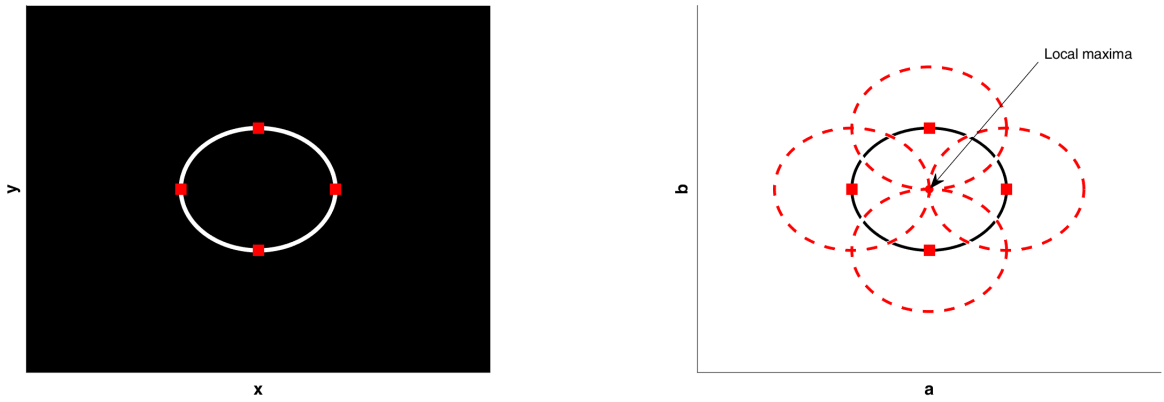


(a) Image space

(b) Parameter space

Figure 6: Edge point Circle Hough Transform

A circle with a radius of r can therefore be drawn in the (a, b) space for each edge pixel discovered by the edge detection algorithm. If the edge was a particle boundary, all sketched circles in (a, b) space would then intersect in one point, representing the center of the particle, see Figure 7.



(a) Edge map in image space

(b) Parameter space

Figure 7: Illustration Circle Houhg Transform

The CHT then transforms each circle to one point in a parameter coordinate system described by two coordinates (a, b), i.e. reduced space. CHT uses a vote process to determine the intersection's point. For this the parameter space (a,b) is first descretized into bins as shown in Figure 8, known as an accumulator matrix. It is a matrix where each cell represents a point in the the parametric space, and the value stored in that cell represents the number of times a circle passed through that point. As circles pass the most through coordinates of the original particles they will thus get most votes. By taking the point with the local maximum, one can then find the particle center.

Another possible way to detect the particles in the drum is by using the light reflection. The reflection of the light creates dominant light spots in the particles. With this method the noise (light reflection) in the image is used as a feature of the particles. Since the particles are spherical in shape, the maximum light reflection from the lamp to the camera lens is at the top of the particles. To identify and locate the light regions one may use the blob detection technique. These methods detect regions in a digital image whose intensity properties differ from those of neighbouring regions, these connected pixels are a blob. The first step for blob detection algorithm is to convert the image to a binary form (black-white) by thresholding the source image, starting at a minimal value.

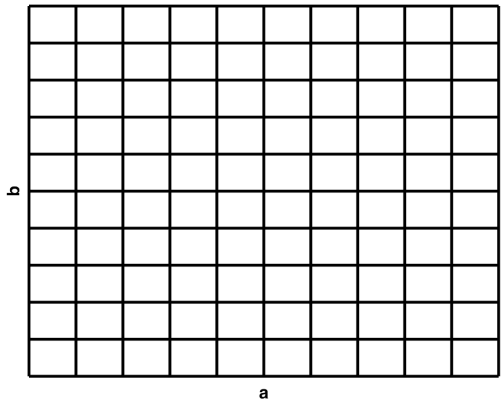


Figure 8: Accumulator matrix

Then increase it until the light reflecting becomes visible. So in this manner the light reflection points are isolated from the rest of the image. Then scanning the image for connected regions that meet a size of white pixels between one and twenty pixels, which groups these pixels into blobs. This value is chosen because there is only interest in the small light reflections that can represent the centers of the particles.

To improve the detection of the particles for both methods, the knowledge of the boundaries of the drum is used. Next to this, the particle detection is constrained such that no particle can be outside of the drum. Similarly, as particles are not compressible, they cannot overlap. However, some "overlapping" of particles can occurs due to projection of a 3D problem onto a 2D front drum plane. Thus, one may also detect particles in the layer behind the front layer. Therefore it is important to categorize the particles into particles of layer one (front layer) and layer two (layer behind front layer). To prevent detection of particles outside the boundaries of the drum, a filter is implemented that removes false detected particles outside the drum. For this purpose each particle is checked whether the distance from the centre of the drum to the particle cannot be bigger than the radius of the drum minus the radius of a particle, i.e.:

$$\text{if} \quad (x_{drum} - x_{particle_i})^2 + (y_{drum} - y_{particle_i})^2 > r_{drum}^2 - r_{particle}^2 \quad \text{then the particle is outside the drum.} \tag{6}$$

The overlapping of the particles is tested with the pair correlation, that represents the squared distance between the centres of the detected particles

$$\text{Pair correlation} = (x_i - x_j)^2 + (y_i - y_j)^2. \tag{7}$$

To determine the close neighbours, the pair correlation is normalized, and the particles in the range of 0.1 to 1.4 particles are seen as close neighbours. The value 0.1 particles is chosen to prevent pair correlation with the particle itself and the value 1.4 particles is chosen to only detect the closest neighbouring particles. Following this, each overlapping particle receives a score by using Equation7 related with the average distance to the nearest neighbour particles. In the next step all the particles are checked to see which of the neighbouring particles has the largest average overlap. A particle is overlapping if the distance between the centers between the two comparative particles is less than twice the radius of the particles. Each particle is tested with the neighbour particles for overlap. The distance of overlap is summed from all overlapping neighbouring particles, this is done for each particle. The particles with more than two times the highest score are considered as a particle of layer 2. If the particle center distance to another center of the particle is less than two times of the particle radius then this particle is a non-overlapping one. The remaining particles are tested and categorized in the same manner as described previously. Each remaining particle is individually tested with the direct neighbour particles categorized in layer 1 whereby the left over remaining particles are not included in the test. The particle in question is given a score with the average pair correlation value of the surrounding particles within a range of 0.1 and 1.4 particles. If the score is less than 1.0, the particle is classified as layer 2 because it overlaps with most of the other particles or has a strong relationship with one particular particle, which suggests that it is a layer 2 particle. The particles with a score higher or equal to 1.0 automatically belong to layer 1.

The previously described method for particle detection is iteratively applied to an image. The reason for this is that not all of particles are detected at the first instance. In order to detect the remaining particles that have not yet been identified, the search area is narrowed. This is accomplished by masking all identified particles. In this manner the search area is minimized and all unnecessary objects are removed from the image. This is realized by colouring these particles black and make them confluent.

2.2 Particle connecting

Once the particles are detected in the initial observed image of a rotating drum, the next challenge is to affiliate these with the particles detected in the next image, i.e. the image shot in the next time step. To achieve so, one may calculate the distance of all detected particles in the current image to the one detected particle in the next image, and find the particle with the shortest distance. Then it is assumed that the particles in the subsequent images that have the shortest distance are embodying the same physical particle. This process is called the Nearest neighbour[63] and is based on calculating the Euclidean distance:

$$Distance(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{8}$$

in which i represents the coordinate center of specific particle in the current image and j represents the center of one specific particle of the succeeding image. Based on Equation 8 one may form a so called distance matrix D, the elements of which are inspected for minimal distance. The issue is, however, that multiple particles of the current image may have a closest distance to a particle of the next image. Thus as shown in Figure 9, a specific particle can be a nearest neighbour more than once. In this example, particle 1 and 2 are two particles in the previous images, and particles A and B are the same physical two particles but in the current images, respectively.

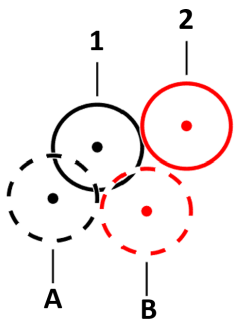


Figure 9: Multiple particles with same particle as nearest neighbour

In the example of Figure 9 the distance of particle A and B is calculated relative to 1 and 2. Particle A has the shortest distance to position 1, similarly to particle B. Thus, both A and B have a particle 1 for a nearest neighbour. The problem of multiple closest neighbour is solved by taking only one neighbour as closest, and assigning the other one to the next closest observed particle. In the assignment process, one has to make sure that the distance between particles in the current and next images does not exceed 2/3 radius of the radius of the particles[54]. This value has been chosen to prevent the wrong assignment process. Once the corresponding particle in the next image is found, the center position of the particle in the current image is recorded and rewritten by the next value. In this manner one obtains the time evolution of the particles center position recorded at the discrete time steps, also known as the particle track. Due to possible particle disappearance or reappearance, it may happen that the observed particle in the next image has no closest neighbour in the previous image or vise versa that the particle in the current image has no closest neighbour in the next image. In case of appearance of the new particle, the former case scenario holds. Then one has to introduce the new particle track in the dictionary. In case of disappearance of the current particle the later scenario happens. Then, the track of that particle is being ended irrespectively if the particle reappears later. Hence, the method is sensitive to misdetection as the particle track also stops in such a case. Therefore, the nearest neighbour detection has to be improved. For this purpose, one may introduce the so-called tracking algorithm. The main purpose of this algorithm is to predict the position of the particle in the time, and to introduce the confidence in the estimation of its position. However, as the detection as well as the nearest neighbour algorithms are not perfectly accurate, the process of tracking as well as approximation has to take into account

these errors. To achieve this, one has to model these errors as uncertain, and further approximate the path by some analytical functional, or a dynamical model including uncertainties. Therefore, this thesis considers probabilistic methods for tracking such as Kalman filter type of algorithms.

2.3 Particle tracking and predicting

The Kalman Filter (KF) and its variants are particularly suitable for tracking and predicting of the maneuvering particles. The KF is a highly efficient filter which is widely used in the field of state estimation for moving objects[64, 65]. The KF is probabilistic method that can account for the position uncertainty as well as measurement noise. It provides the optimal estimate of the system state based on the set of noisy measurements (the detections). The filter can be used for real-time, online processing of data as it operates recursively, updating the estimate of the particle position at each time step. The KF can handle systems with multiple variables, making it useful for particle tracking in two or three dimensions. The particle tracking is in this approach seen as a two-step process consisting of, state prediction and its correction given the measurement data. The prediction of the state is done by using a probabilistic approach to models the uncertainty in the particle state using probability distributions, and combines these estimates with the measurement data to produce an updated estimate. The filter is a approximation of Bayesian inference[43] in which the updated estimate is represented by the mean and covariance of the probability distribution. Due to presence of uncertainty, this method then allows prediction of the missing data. Namely, if a particle detection in the image is missed, the Kalman filter can still predict its position up to some confidence level. Thus, the tracking of the particle is still possible irrespectively of its appearance and re-appearance. The main issue in tracking particles by Kalman filter is that the prediction model has to be a priori known. In other words, one has to model the dynamics of moving particles and then use probabilistic approach to predict all possible particles states. However, particle's dynamics is not known unless obtained from DEM simulations. Similarly to the system identification, one may assume linear dynamics model and use Kalman filter to track the time evolution of the particle center position. However, the particle dynamics is not necessary linear, and thus advanced versions of Kalman Filter have to considered as further described in the text.

2.3.1 Modeling of particle dynamics:

Uncertainty in the particle motion is one of the major challenges of particle tracking. This uncertainty is due to the fact that the tracker does not have access to an accurate dynamic representation of the particle being tracked[66, 67]. When tracing the particles, only the observation of the location of the particles is known, whereas their movement is either caused by drums rotation or its direct consequence, avalanche. Thus, in general words, the motion of the particles can be described as a stochastic process. Namely, particles move but also collide with other particles in the same layer or in other layers. As a result, the full dynamic process is not known and an uncertainty must be added to the process. A probabilistic description of the movement of the particles is formulated in order to create a stochastic model. Instead of discretizing random processes, one applies a direct discrete-time approach. This means the input signals are considered to be piecewise constant random variables (depending on the model, this can be the speed, acceleration, etc) with zero mean and Gaussian distribution between sampling instants[68]. The states of the model (X) that describe the states of the particle are described in discrete form as:

$$X_{k+1} = f(X_k) + g(X_k)w_k \tag{9}$$

in which f represents the dynamics model, g describes the relation between the model and the input signal w. At which the particle movement noise covariance is computed according to

$$Q_k = g(X_k)diag(\sigma^2_{noise})g(X_k)^T. \tag{10}$$

Hereby diag σ^2 is the diagonal matrix with σ being the input standard deviation. In order to describe particle dynamics, i.e. function $f(X_k)$ here are considered three types of models:

- Constant Velocity (CV),
- Constant Acceleration (CA),
- and Coordinated Turn model (CT) .

The previous models are inspired by the modelling applied in control theory for vehicles or robots [66, 69, 68, 70, 71]. The CV model is a linear model and has been used in the previous study [54]. The CA and CT models are non-linear models and can therefore better approximate the motion of the particle since the particles maneuver non linearly. Most of the particles in the drum move in a circular path and therefore their dynamics is non linear. In contrast with the bulk, the top layer of the particles moves extremely randomly due to presence of avalanches. For this research, the priority is to trace and predict the majority of the particles, thus the ones in the bulk. As the top particles have different type of random walk, they are not the main focus of this study.

Constant Velocity model

The Constant Velocity (CV) model is used to describe the motion of a particle in a linear way. This model assumes that the speed of the particle is constant between the time steps, so there is no acceleration. According to this model the state transition of a moving object, in our case a particle, can be described as:

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + T\dot{\boldsymbol{x}}_{k-1} + w_{k-1}. \tag{11}$$

Here T is the time sampling interval, whereas the position and velocity tensors are denoted by $\boldsymbol{x}_k$and $\dot{\boldsymbol{x}}_k$, respectively and w_k represent the noise.

The state of the particle center is described as $\boldsymbol{X}_k := [x_k, y_k, \dot{x}_k, \dot{y}_k]^T$. Accordingly, the state transition model in Equation 11 rewrites to:

$$\boldsymbol{X}_k = \begin{bmatrix} x_{k-1} + \dot{x}_{k-1}T \\ y_{k-1} + \dot{y}_{k-1}T \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{bmatrix} + w_{k-1} \cdot g(X_k). \tag{12}$$

The process noise $w_k = [\ddot{x}, \ddot{y}]^T$ is described as the acceleration in the x and y direction here assumed to be unknown and thus modelled as a zero mean Gaussian distribution $w_k \sim (0, \sigma_a)$. For the CV model, $g(X_k)$ is given by:

$$g(X_k) = \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix}. \tag{13}$$

Thus the covariance matrix describing the particle state $\boldsymbol{X}$ is defined by a matrix Q_k as:

$$Q_k = \begin{bmatrix} \frac{T^4\sigma_{ax}^2}{4} & 0 & \frac{T^3\sigma_{ax}^2}{2} & 0 \\ 0 & \frac{T^4\sigma_{ay}^2}{4} & 0 & \frac{T^3\sigma_{ay}^2}{2} \\ \frac{T^3\sigma_{ax}^2}{2} & 0 & T^2\sigma_{ax}^2 & 0 \\ 0 & \frac{T^3\sigma_{ay}^2}{2} & 0 & T^2\sigma_{ay}^2 \end{bmatrix}. \tag{14}$$

As it is assumed that standard deviations describing accelerations σ_{ax} and σ_{ay} are equal in the x and y directions the noise covariance matrix can be further be simplified as follows:

$$Q_k = \begin{bmatrix} \frac{T^4}{4} & 0 & \frac{T^3}{2} & 0 \\ 0 & \frac{T^4}{4} & 0 & \frac{T^3}{2} \\ \frac{T^3}{2} & 0 & T^2 & 0 \\ 0 & \frac{T^3}{2} & 0 & T^2 \end{bmatrix} \cdot \sigma^2. \tag{15}$$

The disadvantage of this model is that it is a linear model that tries to describe the non-linear motion of the particles. This only works for very small intervals and is therefore not desirable. For this reason, further research is done on non-linear variants.

Constant Acceleration model

Similarly to the CV model, for a Constant Acceleration (CA) model the state transition of a moving object is described as:

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{x}_{k-1} + T\dot{\boldsymbol{x}}_{k-1} + \frac{1}{2}T^2\ddot{\boldsymbol{x}}_{k-1} \\ \dot{\boldsymbol{x}}_k &= \dot{\boldsymbol{x}}_{k-1} + T\ddot{\boldsymbol{x}}_{k-1}. \end{aligned} \tag{16}$$

The difference between the CV and CA model is that with the CA model the acceleration $\ddot{\boldsymbol{x}}$ is also described as a state, and is assumed constant between the time steps. The state of the particle center is described as $\boldsymbol{X}_k = [x_k, y_k, \dot{x}_k, \dot{y}_k, \ddot{x}_k, \ddot{y}_k]^T$. The CA model state transition matrix X_k is:

$$\boldsymbol{X}_k = \begin{bmatrix} x_{k-1} + \dot{x}_{k-1}T + \ddot{x}_{k-1}\frac{1}{2}T^2 \\ y_{k-1} + \dot{y}_{k-1}T + \ddot{y}_{k-1}\frac{1}{2}T^2 \\ \dot{x}_{k-1} + \ddot{x}_{k-1}T \\ \dot{y}_{k-1} + \ddot{y}_{k-1}T \\ \ddot{x}_{k-1} \\ \ddot{y}_{k-1} \end{bmatrix} + w_{k-1} \cdot g(X_k). \tag{17}$$

The process noise $w_k = [\dddot{x}, \dddot{y}]^T$, i.e. it consist of the jerk $\dddot{x}$ and $\dddot{y}$ in the x and y direction, respectively. Both are assumed to be unknown and hence described by a zero mean Gaussian distribution $\sigma \sim (0, \sigma_{\dot{a}})$. Furthermore, $g(X_k)$ is given by:

$$g(X_k) = \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{18}$$

Including all these terms, the covariance matrix Q_k for the particle at the next time moment is:

$$Q_k = \begin{bmatrix} \frac{T^4\sigma_{\dot{a}x}^2}{4} & 0 & \frac{T^3\sigma_{\dot{a}x}^2}{2} & 0 & \frac{T^2\sigma_{\dot{a}x}^2}{2} & \\ 0 & \frac{T^4\sigma_{\dot{a}y}^2}{4} & 0 & \frac{T^3\sigma_{\dot{a}y}^2}{2} & 0 & \frac{T^2\sigma_{\dot{y}x}^2}{2} \\ \frac{T^3\sigma_{\dot{a}x}^2}{2} & 0 & T^2\sigma_{\dot{a}x}^2 & 0 & T\sigma_{\dot{a}x}^2 & 0 \\ 0 & \frac{T^3\sigma_{\dot{a}y}^2}{2} & 0 & T^2\sigma_{\dot{a}y}^2 & 0 & T\sigma_{\dot{a}y}^2 \\ \frac{T^2\sigma_{\dot{a}x}^2}{2} & 0 & T\sigma_{\dot{a}x}^2 & 0 & \sigma_{\dot{a}x}^2 & 0 \\ 0 & \frac{T^2\sigma\dot{a}y^2}{2} & 0 & T\sigma_{\dot{a}y}^2 & 0 & \sigma_{\dot{a}y}^2 \end{bmatrix}. \tag{19}$$

Same as for the CV model in the CA model, one also assumes that the model error is equal in both directions, because of this the noise covariance matrix can be written as follows:

$$Q_k = \begin{bmatrix} \frac{T^4}{4} & 0 & \frac{T^3}{2} & 0 & \frac{T^2}{2} & \\ 0 & \frac{T^4}{4} & 0 & \frac{T^3}{2} & 0 & \frac{T^2}{2} \\ \frac{T^3}{2} & 0 & T^2 & 0 & T & 0 \\ 0 & \frac{T^3}{2} & 0 & T^2 & 0 & T \\ \frac{T^2}{2} & 0 & T & 0 & 1 & 0 \\ 0 & \frac{T^2}{2} & 0 & T & 0 & 1 \end{bmatrix} \cdot \sigma_{\dot{a}}^2. \tag{20}$$

In the situation of the top layer particles in the drum, the particles move extremely stochastic. So a constant acceleration between time steps does not give a good approximation.

Coordinated Turn model

For tracing the maneuvering particle, a coordinated turn (CT) target motion is applied. This makes it possible to track the particle when it makes a maneuver. The particles in the drum continuously make a maneuver because they move in a circular trajectory and change their motion due to the collisions between the other particles. The

standard CT model describes the particle with a constant known turn rate, a two dimensional motion and a nearly constant speed around the center of the drum. For particles in the drum, one assumes that the particle position is known, whereas the turn rate is not. Namely, the particle does not have the same turn rate as the drum due to possible slipping. So the particle turn rate is not constant and the particle speed is varying. This can be further extended to a non-linear variant of the CT in which one may have a varying turn rate and a varying particle speed[68].

The state of the particle for a CT model in polar coordinates is described as $\boldsymbol{X}_k = [x_k, y_k, v_k, h_k, \omega_k]^T$ where x_k and y_k both $[m]$ are the particle position, v_k $[m/s]$ is the magnitude of the particle velocity vector, h_k $[rad]$ denotes the heading angle of the particle and ω_k $[rad/s]$ is the turn rate of the particle. For clarification purpose, this is shown in Figure 10.

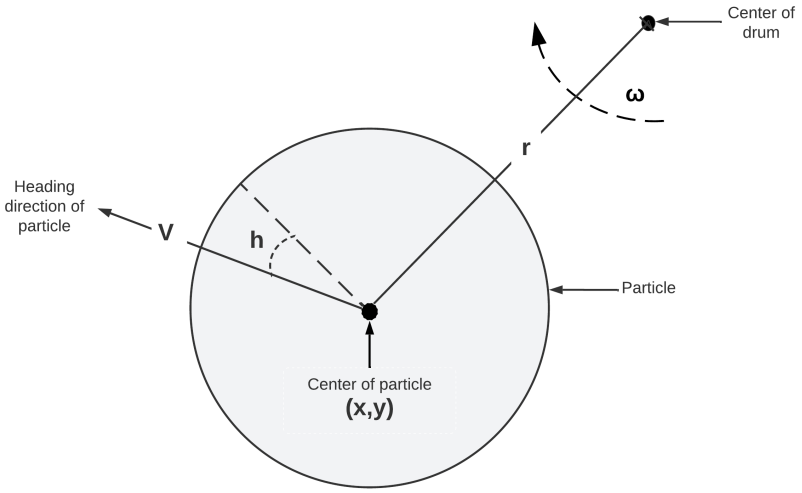


Figure 10: Explanation of the particle states in the Coordinated Turn model

The CT particle dynamics model is described by [68]

$$\boldsymbol{X}_k = \begin{bmatrix} x_{k-1} + \frac{2v_{k-1}}{\omega_{k-1}} sin(\frac{\omega_{k-1}T}{2}) cos(h_{k-1} + \frac{\omega_{k-1}T}{2}) \\ y_{k-1} + \frac{2v_{k-1}}{\omega_{k-1}} sin(\frac{\omega_{k-1}T}{2}) sin(h_{k-1} + \frac{\omega_{k-1}T}{2}) \\ v_{k-1} \\ h_{k-1} + \omega_{k-1}T \\ \omega_{k-1} \end{bmatrix} + w_{k-1} \cdot g(\boldsymbol{X}_k). \tag{21}$$

Due to the complexity of the time varying heading angle the value is set as a constant between sampling instants. In the previous equation the modeling noise is $w_k := [a, \alpha]^T$ is described as the acceleration magnitude a $[m/s^2]$ and the angular acceleration α $[rad/s^2]$. These values are considered as an impulse just before the sampling instant. Both are assumed to be unknown and hence described by a zero mean Gaussian distribution $a \sim (0, \sigma_a)$ and $\alpha \sim (0, \sigma_\alpha)$. The resulting $g(X_k)$ of the CT model is given by:

$$g(X_k) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ T & 0 \\ 0 & 0 \\ 0 & T \end{bmatrix}. \tag{22}$$

All of these terms together, give the covariance matrix Q_k of the particle state in the next time moment as:

$$Q_k = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & T^2\sigma_a^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & T^2\sigma_\alpha^2 \end{bmatrix}. \tag{23}$$

2.3.2 Kalman Filter

The Kalman filter (KF) [44] is a tracking algorithm that uses one of the models as described above and Bayesian estimation to estimate the state of the particle dynamic from the noisy particle center position measurements achieved with the CHT. The strength of the KF is that despite the exact nature of the particle movement is not known, it supports estimations of past, present, and even future states. At each time step, the KF estimates the state of the particle using the following steps:

- Predict: the state of the particle is estimated at the current time step based on the previous state. This estimate is known as the a priori estimate.
- Update: the a priori estimate is updated using the current particle measurement to produce the a posteriori estimate, which takes into account the latest information.

These steps are repeated at each measurement increment to produce an updated estimate of the state of the particle. If there is no available particle measurement, the update phase is skipped.

For the update step the measurement (the detection of the particle) is used. This is described with the measurement matrix

$$z_k = H_k X_k + v_k \tag{24}$$

in which the H matrix described the measurement operator, and v_k is the noise of the measurement. The standard deviation of the measurement is normally known. However, for observing the location of the particles, nothing is known about the measurement accuracy. The location of the particles is determined on the basis of an image in which the particles are detected by means of a number of operations on the images. Then the location of the individual particle is detected with the Circular Hough transform. The precision of the camera is unknown, but it is assumed that the accuracy of the camera is so high that it has no influence on the measurement accuracy. The measurement accuracy is determined by the accuracy of detecting the particles by means of the Circular Hough transform. For this study it is assumed that the measurement standard deviation is one pixel in both x and y directions. This value is small compared to the radius of a particle being fourteen pixels. One pixel is assumed because the large misdetections are filtered out with pair correlation between the particles as explained earlier. Thus measurement noise v_t is a zero mean, uncorrelated (white) noise $v_t \sim N(0, R_t)$ in which R_t represents the error in pixels of the accuracy of the CHT.

To track the particle position, one first has to assume its initial position in the first image. This is known as a prior or experts information. For example, one can use Gaussian distribution to describe the particle position as one is uncertain about it. For a linear dynamics model:

$$x_k = A_{k-1}x_{k-1} + Q_k \tag{25}$$

one may then predict the position of the particle in the next image as:

$$\bar{\boldsymbol{x}}_k^- = A_{k-1}\bar{\boldsymbol{x}}_{k-1}^- + Q_k. \tag{26}$$

In which $\bar{\boldsymbol{x}}_k^-$ denotes the mean of the predicted state. The minus sign in the upper index is used to denote prediction. On the other hand, the covariance of the next state can be computed as:

$$P_k^- = A_{k-1}P_{k-1}A_{k-1}^T + Q_k \tag{27}$$

in which P_k^- is the a priori estimated covariance and P_{k-1} is the a posteriori estimated covariance at the time moment k-1, and Q_k is the covariance of the model error.

After predicting the states of the particle, one observes and hence update prior knowledge with the measurement. This is called data assimilation. To do this, one may use the KF update equations:

$$\bar{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}_k^- + K(z_k - H_k\bar{\boldsymbol{x}}_k^-) \tag{28}$$

$$P_k = (I - K_k H_k)P_k^- \tag{29}$$

In Equation 28 the prediction of the mean particle position $\bar{x}_k^-$ via measurement operator H_k is compared to the real measurement z_k. By scaling the discrepancy between the real measurement and the prediction with the Kalman gain K, one can update the mean position. Here Kalman gain is defined as:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{30}$$

Similarly, in Equation 30 one may update the covariance of the particle position by using the measurement data. By using Equation 28 and 29 one gains more knowledge about the particle position, meaning that the covariance in Equation 29 is smaller than covariance in equation 27. In other words, one is becoming more confident about the position of the particle. The previously described process is repeated for each new measurement. To support the formulas, a visual representation of a piece of particle track is shown in Figure 11. The figure represents the movement of a particle from bottom right to top left. Each step represents the position of a particle in the subsequent time step. The actual particle location is shown in red and the detected particle location is shown in green, this is the initial step. In the subsequent steps, blue indicates where the KF predicts the particle to be. The location of the particle after the correction of the Kalman gain in the correcting step is depicted in black. In the final step, it can be seen that the KF predicts the location of the particle when no detection is available.

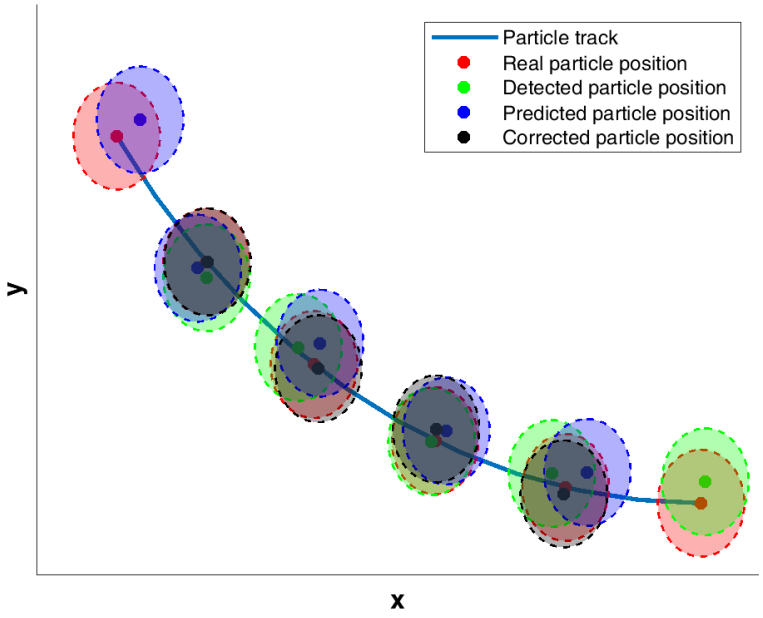


Figure 11: Visualisation of the KF for a particle track

The KF is made for linear estimation, however the movement of a particle in the drum is a non-linear movement. As a result, the KF is only suitable if the time steps are sufficiently small, so that the motion of the particle can be approximated as linear. However, the goal of this thesis is to improve the tracks by estimating where the particle is over a longer period of time. As a result, it is not possible to consider the motion of the particle as linear. To approach the motion of the particle realistically it is necessary to include non-linear terms. For this reason it is necessary to use a non-linear variant of the KF, here assumed to be the Unscented Kalman Filter.

2.3.3 Unscented Kalman Filter

In contrast to KF, the unscented version[55] is not representing the prior state over the mean and covariance as shown in Equations 9 and 24, but is defined by a small number of carefully selected sample points known as Sigma points $\boldsymbol{\chi}$. To track the particle position, one must first assume its initial position in the first image, just like the KF. This is referred to as prior or expert information. For predicting the states of the particle with the UKF all the Sigma vectors are propagated through the nonlinear function described by:

$$\boldsymbol{\chi}_{k|k-1} = f(\boldsymbol{\chi}_{k-1}) \tag{31}$$

in which $f(\boldsymbol{\chi})$ can denote one of the previously presented models, i.e. CV, CA or CT. This new state is then having the prior mean:

$$\bar{\boldsymbol{x}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \chi_{i,k|k-1} \tag{32}$$

in which W_i are the weights as further explained in the text. The covariance of the current state can be computed as:

$$\boldsymbol{P}_k^- = \sum_{i=0}^{2L} W_i^{(c)} \left[\chi_{i,k|k-1} - \bar{\boldsymbol{x}}_k^-\right] \left[\chi_{i,k|k-1} - \bar{\boldsymbol{x}}_k^-\right]^T + Q_k. \tag{33}$$

The sigma points are chosen to capture the mean and covariance of the estimate, and are transformed through the non-linear function to produce an updated estimate of the state according to:

$$x_k = f(\chi_k) + Q_k \tag{34}$$

$$z_k = h(\chi_k) + v_k. \tag{35}$$

As can be seen in Equation 34, the previous particle position is non-linearly transformed to the next particle position. Assuming that the prior knowledge on x_k is a Gaussian distribution, then one has to find a way of computing statistics of the particle position in the next image. However, due to non-linearity of the mapping this is not straightforward as in Kalman filter case. Namely, the distribution of χ_k is non-Gaussian. The technique that the UKF used for computing the statistics of a random variable that has undergone a non-linear transformation is known as the unscented transformation (UT) [56]. A random variable $\boldsymbol{x}$ (dimension L, the number of states) with a mean $\bar{\boldsymbol{x}}$ and a covariance $\boldsymbol{P_x}$ is propagated by the nonlinear function 34 . The unscented transform is graphically visualized in Figure 12 and shows that the true transformation is approached with only a few sigma points. The sigma points are passed through the non-linear function and approximate the true mean and covariance.

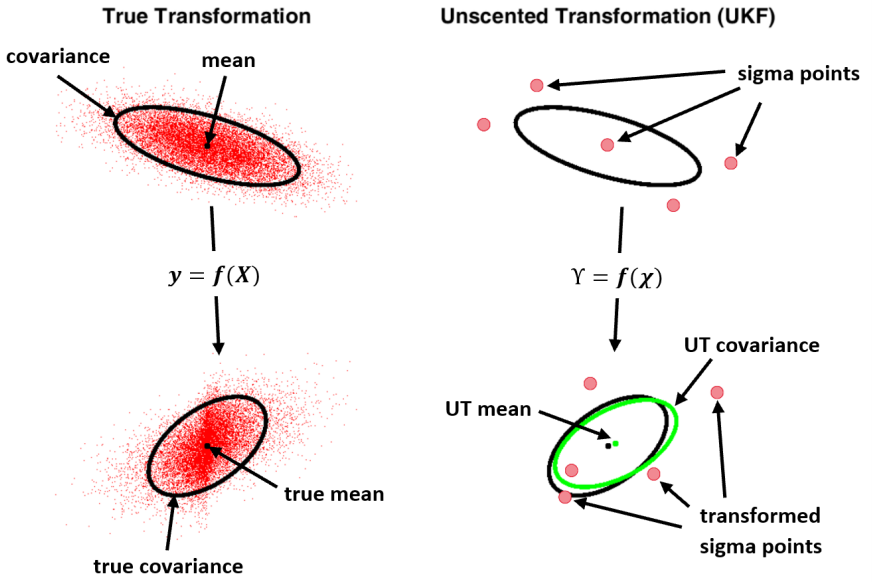


Figure 12: Visualisation of the Unscented Transform

In order to compute the statistics of $f(X_k)$, first one creates a matrix $\boldsymbol{\chi}$ consisting of $2L+1$ Sigma points χ_i that are calculated as:

$$\begin{aligned} \chi_0 &= \bar{x} \\ \chi_i &= \bar{x} + (\sqrt{(L+\lambda)\boldsymbol{P_x}})_i \qquad i = 1,..,L \\ \chi_i &= \bar{x} - (\sqrt{(L+\lambda)\boldsymbol{P_x}})_i \qquad i = L+1,..,L \end{aligned} \tag{36}$$

in which $\bar{x}$ is the mean value and P_x is the covariance. The scaling parameter $\lambda = \beta^2(L+\kappa) - L$ is used to control the spread of the sigma points used to approximate the posterior distribution. The scaling parameter called κ is a constant that is typically set to 0[72]. β controls the spread of the sigma points based on the prior belief of the distribution shape of x, with $\beta = 2$ being the ideal value for Gaussian distributions[73]. The ith column of the matrix's square root, $(\sqrt{(L+\lambda)\boldsymbol{P_x}})i$, is computed using the lower triangular Cholesky factorization.

Weighting the mean and covariance in UKF is an important aspect of the estimation process, as it helps to balance the trade-off between accuracy and robustness. The weighting of the mean and covariance determines how much importance is given to the mean and covariance of the estimate, and affects the accuracy and robustness of the estimate. The weights W_i given by:

$$\begin{aligned} W_0^{(m)} &= \lambda/(L+\lambda) \\ W_0^{(c)} &= \lambda/(L+\lambda) + (1-\alpha^2+\beta) \\ W_i^{(m)} &= W_i^{(c)} = 1/(2(L+\lambda)) \qquad i = 1,..,2L \end{aligned} \tag{37}$$

The constant α in combination with constant κ , controls how far the sigma points are spaced around the variable $\bar{\boldsymbol{x}}$[56]. The constant α is set between 10^{-4} and 1, to avoid non-local effects in non-linear systems α is set to a small value and thus is set to value $1e{-}3$ [56]. It's important that the sum of the weights assigned to the sigma points must equal 1:

$$\sum_{i=0}^{2L-1} W_i = 1 \tag{38}$$

as the sigma points represent a probability distribution. This ensures that the mean and covariance of the estimate are computed correctly and accurately reflect the state of the particle. The mean is computed based on the transformed sigma points 36 and their associated weights 37, and it represents the most likely value of the state estimate. The covariance is computed based on the transformed sigma points and their associated weights, and it represents the amount of uncertainty in the estimate.

After the states of the particle are predicted, the prior knowledge is updated if a measurement is available. The estimated output of the particle, the location is found with all the Sigma vectors propagated through the output function 35 described by:

$$\boldsymbol{\gamma}_{k|k-1} = h\left(\boldsymbol{\chi}_{k|k-1}\right). \tag{39}$$

This gives the estimated output:

$$\bar{\boldsymbol{y}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \gamma_{i,k|k-1} \tag{40}$$

The covariance $\boldsymbol{P}_{y_k y_k}$is described as:

$$\boldsymbol{P}_{y_k y_k} = \sum_{i=0}^{2L} W_i^{(c)} \left[\boldsymbol{\gamma_{i,k|k-1}} - \bar{\boldsymbol{y}}_{\boldsymbol{k}}\right] \left[\boldsymbol{\gamma_{i,k|k-1}} - \bar{\boldsymbol{y}}_{\boldsymbol{k}}\right]^T + v_k. \tag{41}$$

The Kalman gain K_k is calculated with the covariances 33 and 41 as

$$K_k = \boldsymbol{P}_k^- \boldsymbol{P}_{y_k y_k}^{-1}. \tag{42}$$

Weighting with the Kalman gain, the differences between the estimated output and the measured output, the a posteriori estimated state is determined as:

$$\bar{\boldsymbol{x}}_k = \boldsymbol{x}_k^- + K_k\left(\boldsymbol{y}_k - \bar{\boldsymbol{y}}_{\boldsymbol{k}}^-\right). \tag{43}$$

The covariance is calculated with the covariances and the Kalman gain

$$\boldsymbol{P}_k = \boldsymbol{P}_k^- - K_k \boldsymbol{P}_{y_k y_k} K_k^T. \tag{44}$$

This process is repeated for each time the new measurement is made.

3 Numerical results

In this chapter the particle detection and particle tracking results of the research are presented. The results of the various stages of the particle detections based on the images from avalanche experiment simulated by the rotating drum are shown. The chapter presents and compares results of multiple particle tracking methods. The following particle tracking methods are discussed; The nearest neighbour, the Kalman Filter in combination with the constant velocity model, and the Unscented Kalman filter in combination with the constant acceleration model and the coordinated turn model.

3.1 Particle detection

The software/program Matlab R2020b has been used to construct the computer simulations. The dataset of the research is taken from [13, 54] and is used as input for the analysis of the images. In this previous experiment transparent particles with a 2mm diameter and tracer particles with a 2mm diameter and green colour are used. The material consists of borosilicate glass and is incompressible. The data set of the consecutive image is filmed with a frame rate of 450 frames per second, with a total number of 8780 frames. The image resolution is 1024x1280 pixels. The drum has an angular velocity of 5 rotations per minute.

Before starting the experiment, a white colored circle has been drawn on the inside contours of the drum by hand as shown in Figure 15. The camera and drum setup are fixed, so the center of the drum is at the same location in each frame. For this reason, the drum's boundary only needs to be detected once, and its coordinates are used for the subsequent image processing steps. The space outside the white boundaries of the drum is colored black to narrow the scope in the image to only the area of interest. The canny edge method is used to detect the contours, and the drum boundary is detected by using CHT method. After detection of the boundary only the content of the drum is made visible as shown in Figure 14.

(a) Original image

(b) Drum boundary drawn

Figure 13: Drum boundary preparation

Since there is only interest in the particles inside the drum and the drum is partially filled with particles, the part without particles is also blacked out by removing the unnecessary objects and colors in this part of the image. To achieve this, the image with only the drum visible is used for this (as shown in the right bottom in Figure14. As first step the colour channels are reduced from three color channel to one color channel by displaying the image in gray scale as shown in Figure 15a . Afterwards the image is filtered through an entropy filter as shown in Figure 15b. The local entropy of an image is proportional to the neighbourhood complexity. The entropy filter may detect minor fluctuations in the local distribution of gray levels. This separation is required to distinguish between particles and background in the image. The image is converted to black and white,this is necessary for segmentation, and a threshold value of 0,79 for the color gray intensity is chosen; this value is found by trail and error and gave the best results. The bright parts (such as the contours of the drum) shown in Figure 16 that do not belong to the particles are filtered out by removing narrow pixel groups in a binary image. Because the tracer particles are green in color, they become clearly visible when the image is made black and white as shown in Figure 16. The detection of the tracer particles is done in a subsequent step. To segment the particles in the drum, the segment of the particles is made as one, after which all pixels outside this segment were blackened in the original image as shown in Figure 16. The image with only the particles visible is shown in Figure17.

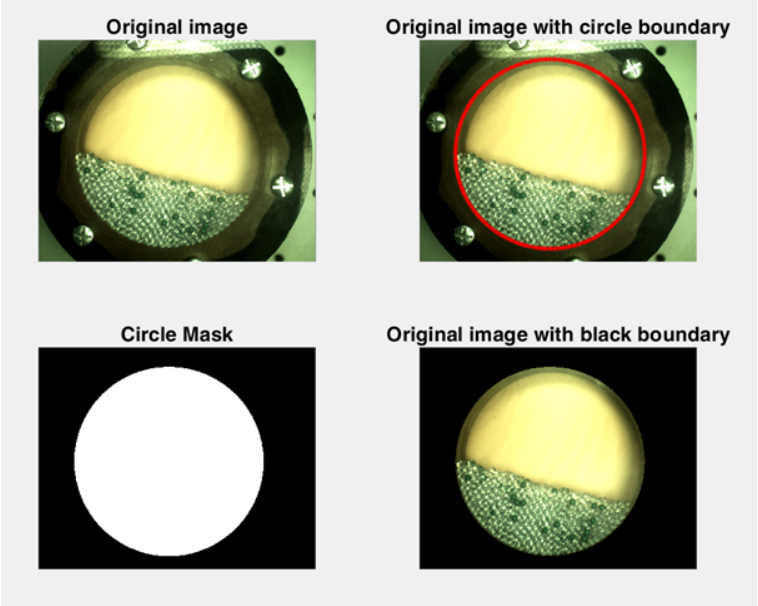
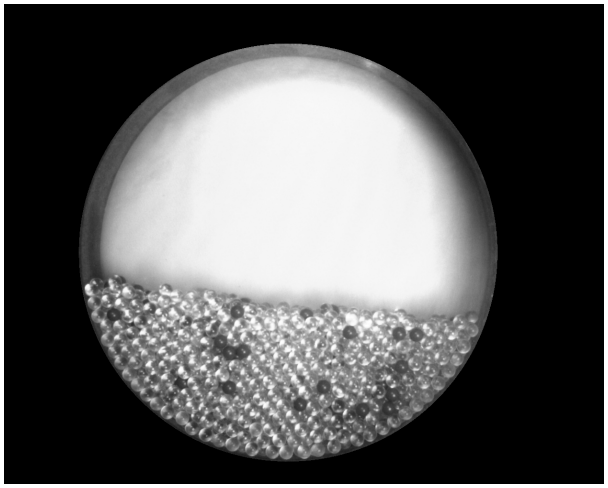
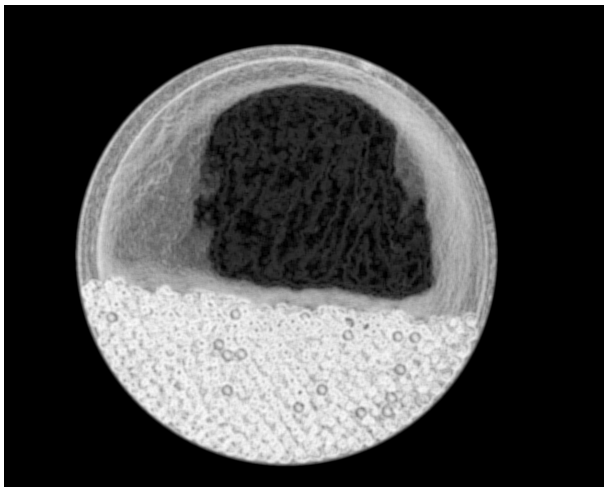


Figure 14: Drum boundary



(a) Gray scale image



(b) Entrophy filtered gray scale image

Figure 15: Drum boundary preparation

Figure 16: a)Black white particle segment b) close small object c) close small segments d) close holes

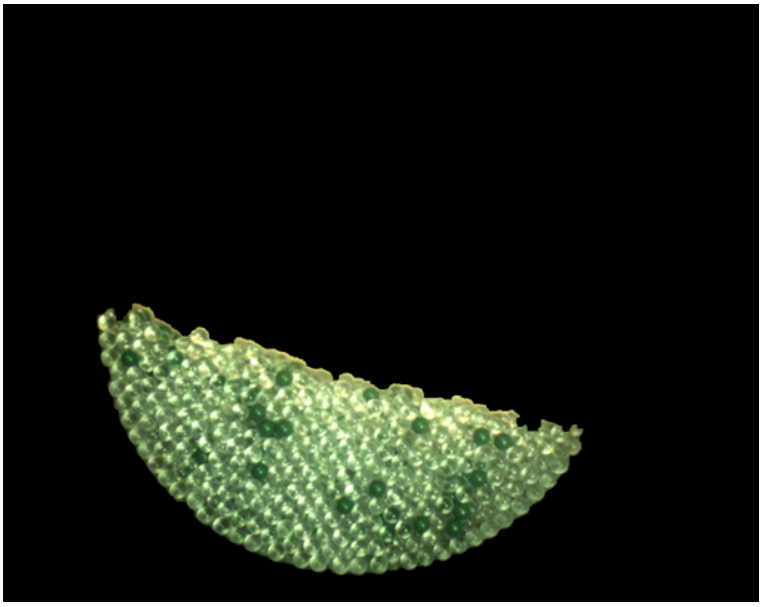


Figure 17: Particle segment

After processing the images, the detection of the particles is performed with the CHT. To use this method the images are converted to a grey scale. Because the radius of the particles is known, as previously explained the method only searches for circular objects with this radius in the image, thereby improving the precision of particle detection. The number of pixels representing the radius of the particles is calculated as:

$$\text{radius particle in number of pixels} = \text{radius drum in number of pixels} \cdot \frac{1}{2} \cdot \frac{\text{diameter particle}}{\text{diameter drum}}. \tag{45}$$

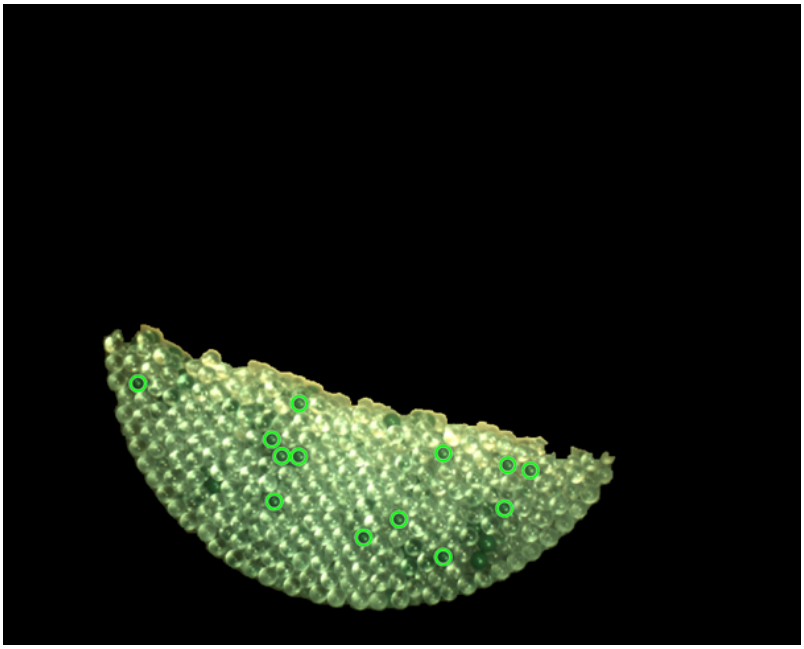
The number of pixels representing the diameter of the drum are known, this is found by detecting the drum radius and location as described in the beginning of this section. The calculated radius of the particles represented in pixels is in the range of 13 and 15 pixels.

Before detecting other particles, first the the tracer particles are detected using the CHT by using the number of pixels per particle radius as input. For this purpose the segmentation image shown in Figure 18a is used and the sensitivity of the pixel color intensity dark is chosen with a value of 0,981. This value is found by trail and error and gives the best results. The detected particles can be seen in Figure 18b. For the remaining particles the CHT particle detection the sensitivity of CHT for brightness is set with the threshold value 0,985. This value is found with trail and error and gives the best results for finding the contours of the particles. As shown in Figure 19a during the first particle detection round there are still many false detected and overlapping particles. These particles have to be labeled into two categories: particles in the first layer (front layer) and second layer (the layer behind the front layer). This is required to distinguish between overlapping and falsely detected particles. As all particles are known to have the same radius, the radius of all particles is set equal to the average radius of all detected particles. The pair correlation is used to test whether a particle is overlapping or non-overlapping. In Figure 19b the non-overlapping particles are shown in blue, the overlapping particles in red and the tracer particles in green colour. The results of the non-overlapping particles shown in blue and the tracer particles shown in green

are good, and thus accurately detected. To validate that the detection is accurate, it is necessary to distinguish the particles of the first layer, which are fully visible and therefore simple to validate. The overlapping particles shown in red are not yet good and need further processing. In Figure 20 the particles belonging to layer 2 are shown in red and the particles in white are the remaining not categorized particles. The remaining particles are tested again for non-overlapping particles now that the dominant overlapping particles have been removed.

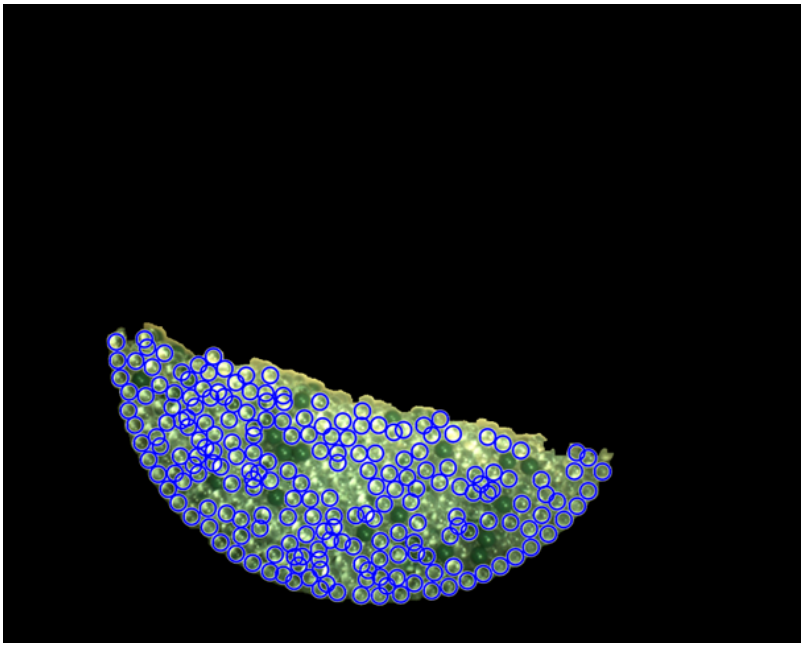
First the tracer particles are detected using the CHT by using the number of pixels per particle radius as input. The segmentation image wherein the holes are still visible (as shown in Figure 18a) are used for this calculation. However in this calculation the sensitivity of the pixel color intensity dark is chosen with a value of 0,981. This value is found by trail and error and gives the best results. The detected particles can be seen in Figure 18b.

(a) Tracer particles segmentation

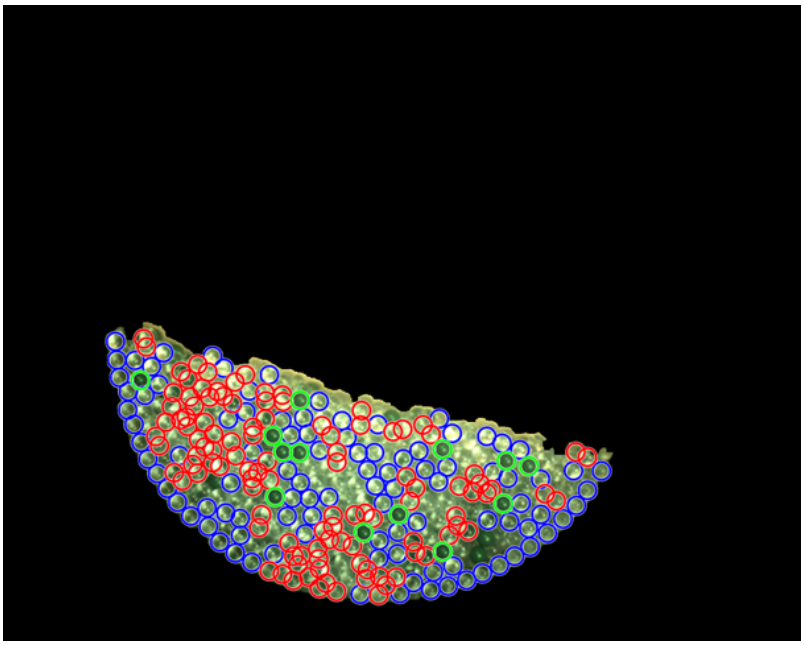


(b) Detected tracer particles

Figure 18: Tracer particles shown in green

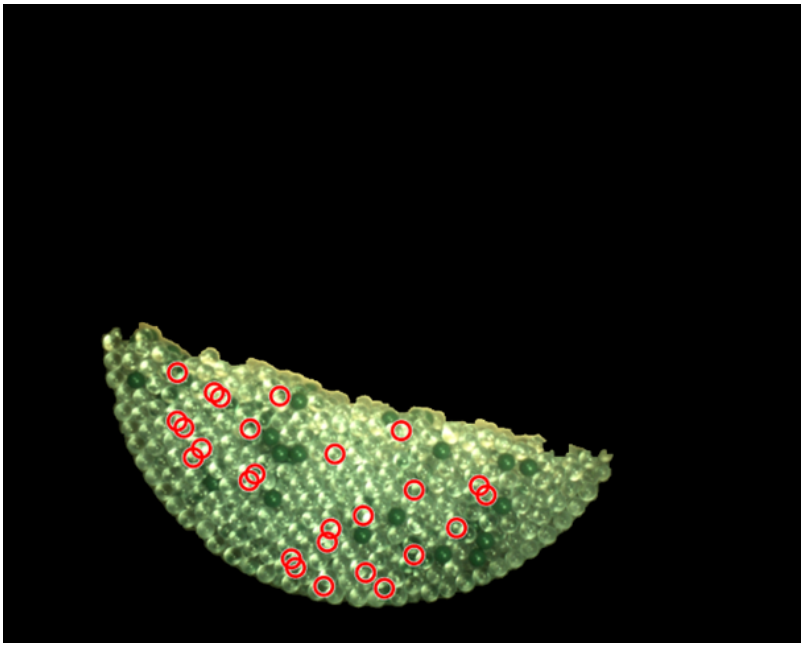


(a) Particle detection shown in blue



(b) Categorized particles, non overlapping shown in blue, overlapping shown in red, and tracer in green

Figure 19: First round particle detection



(a) Particles layer 2 shown in red

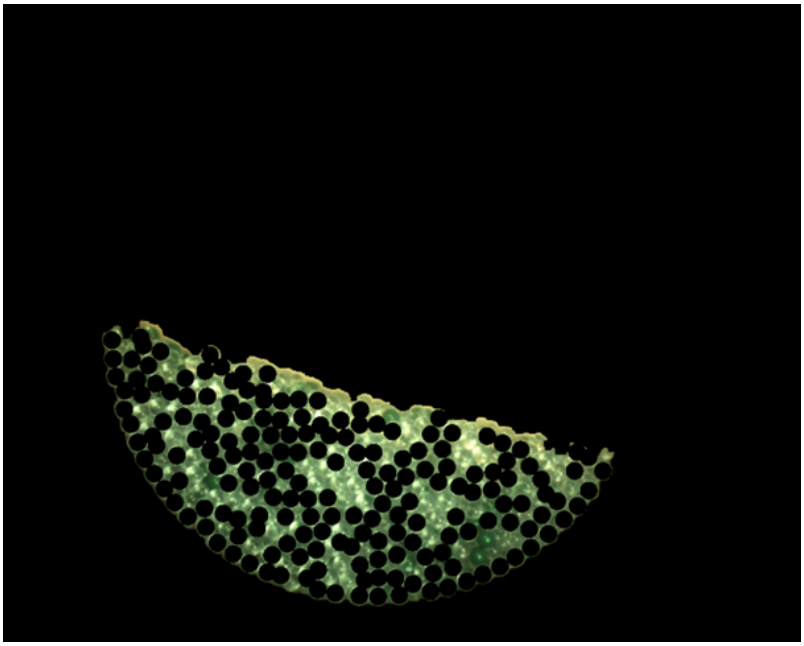
(b) remaining overlapping particles shown in white

Figure 20: Categorizing of overlapping particles

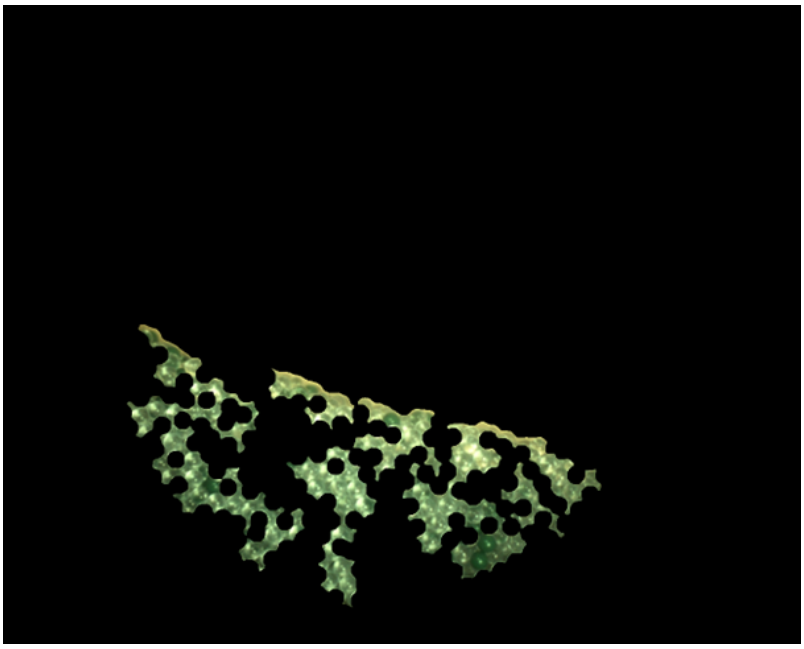
Once the particles are detected, they are removed from the image by same segmentation approach as described previously, see Figure 21. The remaining image is then processed again for the new round of detection's. After the scope of the search area is minimized a new CHT is performed. Now the bright threshold value is increased to 0.986 and the same radii range of 13-15 pixels is used. Hence, the previously undetectable particles are found. The disadvantage of changing the bright threshold value to a higher value is a higher false detection of particles and therefore this value is not used at the start of the experiment. After detecting new particles, one has to check for the overlap with previously detected particles. Those that have an overlap are removed and assigned to layer 2. The overlap is tested with a margin of $c = 0,85$ used in:

$$\text{if} \quad \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} < 2 \cdot r_{particle} \cdot c \quad \text{then the particle is overlapping.} \tag{46}$$

Here, X_i and Y_i are the centre of the particle and X_j and Y_j are the centre of the particle being tested and $r_{particle}$ is the radius of the particles.



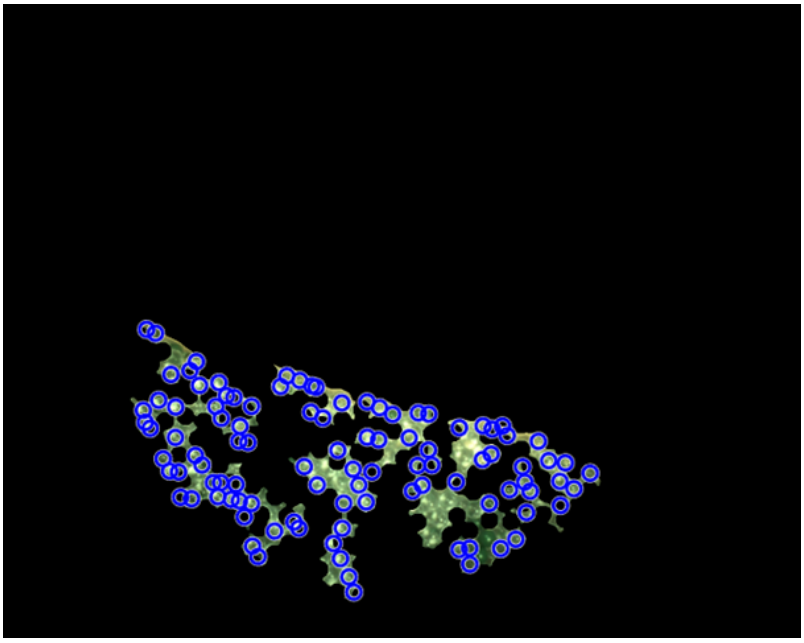
(a) black detected particles



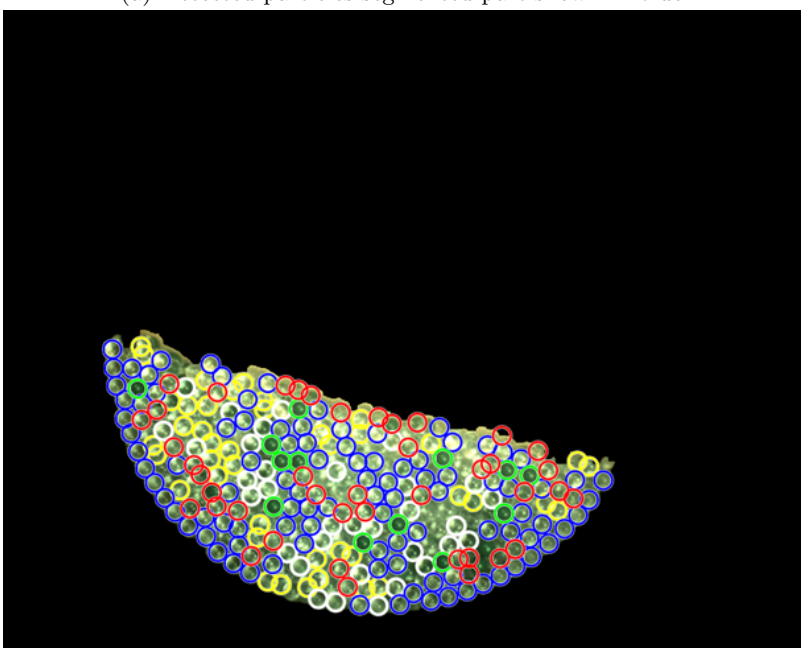
(b) Smooth black out

Figure 21: Black out detected particles

In Figure 22a the detected particles of the segmented part can be seen. Figure 22b shows all detected particles of layer 1. The red particles are those found in the segmented part. The accuracy of the particle detection is good for the particles shown in blue, white, and green. As the search for the other particles continues, it can be seen that this is becoming increasingly difficult and that this is at the expense of accuracy. This can be seen for the particles shown in red and yellow.



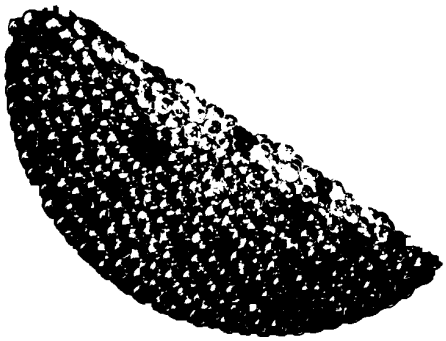
(a) Detected particles segmented part shown in blue



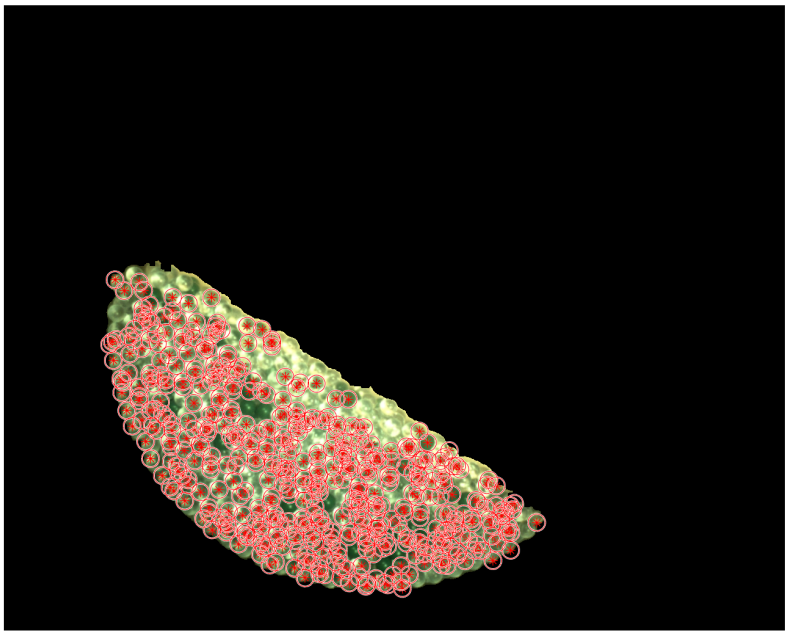
(b) All particles layer 1

Figure 22: Detected particles

In contrast to CHT method, one may also use the so-called blob detection as described in Chapter 2. Based on this detection method, the particle centers are detected on the basis of the light reflection. For this purpose the image of the particles is reduced to black and white as shown in Figure 23a. The particles found with blob detection are shown in Figure 23b. Here one may see the centers of the particles marked with a red marker, and the contours with red color.



(a) Particles in black and white



(b) Detected particles found with blob detection shown in red

Figure 23: Detected particles

In Figure 23b one can see that the detection of the particles with the blob detection does not give good results. This is because the light reflection is too strong on the transparent particles. As a result, too large surfaces reflect the light and the centers cannot be found correctly. For the particles from other experimental data in the color green and red, the light reflections are very well visible as small white light dots. However, the quality of these images is too poor to validate the particles and no tracer particles are used in this data set. To be able to use the blob detection method, a new experiment should be done with the green particles at a higher resolution and tracer particles should be used.

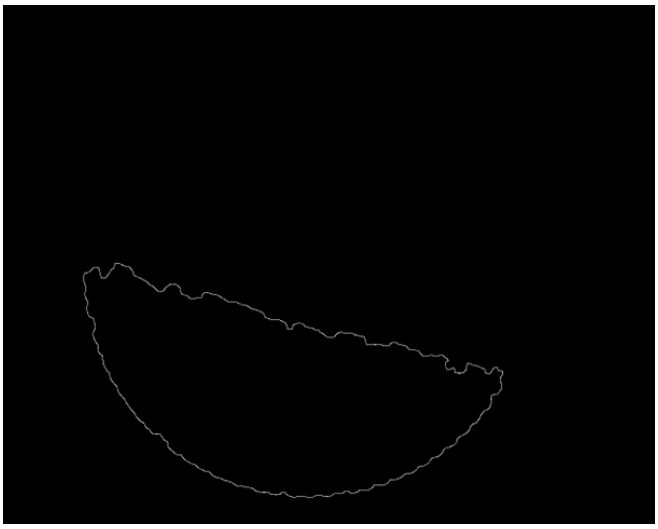
Detection of an avalanche slope angle

The slope angle provides valuable information about the development process of avalanches. To calculate the slope angle, one has to be able to trace the top line of the particles bulk. Thus, to find this contour the canny edge detection algorithm is used, and the drum contour is filtered out, because the drum outline is irregular. Now only the contour line of the slope is visible. This contour line is converted into point coordinates after which the trend

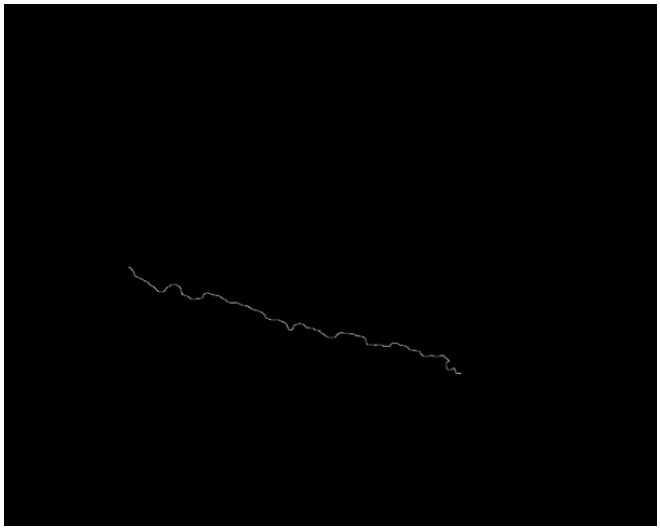
line of the slope is found with linear regression. With this trend line the angle is calculated as:

$$\text{slope angle} = tan^{-1}\frac{maxY - minY}{maxX - minX} \tag{47}$$

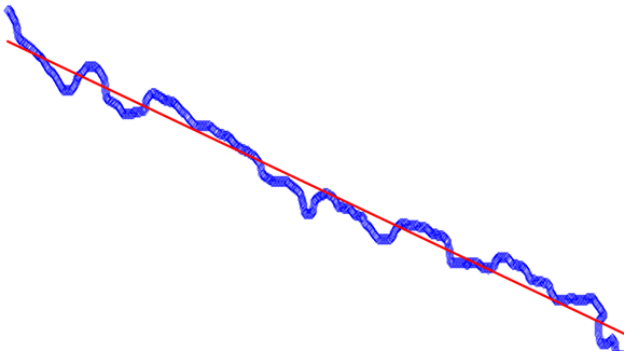
The steps are shown in Figure 24.



(a) canny edge detection



(b) Slope line



(c) Trend line

Figure 24: Slope angle detection

3.2 Tracing particles in subsequent images

In order to produce the trajectories of the particles, the nearest neighbour algorithm is utilized. In the subsequent images, the particles are paired by locating the particle with the shortest distance between them. A matrix is used to store the associated particles of a track, with each track being written in a row of the matrix. Thus, the matrix gives the coordinates of the particle, the IDs of the associated frames, the number of frames that are observed, and the track number. Only the tracks that are currently being observed are stored in this matrix so that the highest possible level of matrix efficiency can be maintained. If there is no possibility of linking a new particle to the track, then this particular row will be eliminated from the matrix in order to maintain its compact nature. During the process of storing a check is done to determine whether or not the minimum required number of frames are observed. A minimum of one hundred frames must be observed in order to generate a visible track; this is the minimum number of frames that is acceptable with a frame rate of 450. In the event that this is the scenario, the track will be saved on its own.

For the purpose of illustration, the tracks that have a continuous observation for a minimum of one hundred frames are plotted as shown in Figure 25. In this image one can see tracks depicted with different colors. The particles within the bulk are obviously moving according to the circular path, whereas the particles in the upper layer follow a more noisy path. This method and this data set allow for the creation of 2115 tracks. As the length of the track increases, more information is available about the behaviour of the specific particle track, which improves the ability to predict the tracks. For example when the arbitrary cut off point of continuous observation of at least 300 frames is chosen, the total number of created tracks is drastically reduced to 496. The reason is that with reduced number of frames more misdetections happen, and thus the track is stopped immediately. This makes Nearest Neighbour extremely sensitive to the number of observations as shown in Figure 26.

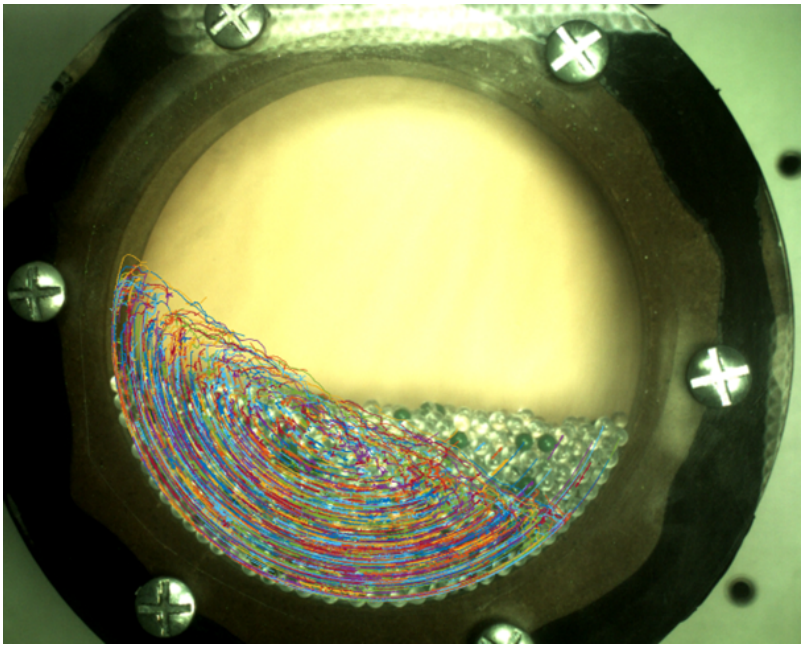


Figure 25: Tracks with an observation of at least 100 frames continuously

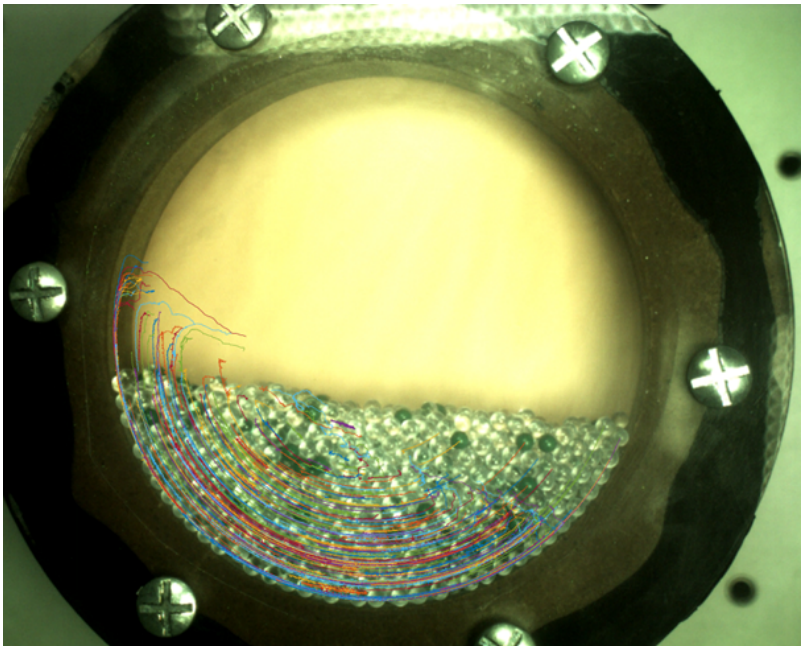
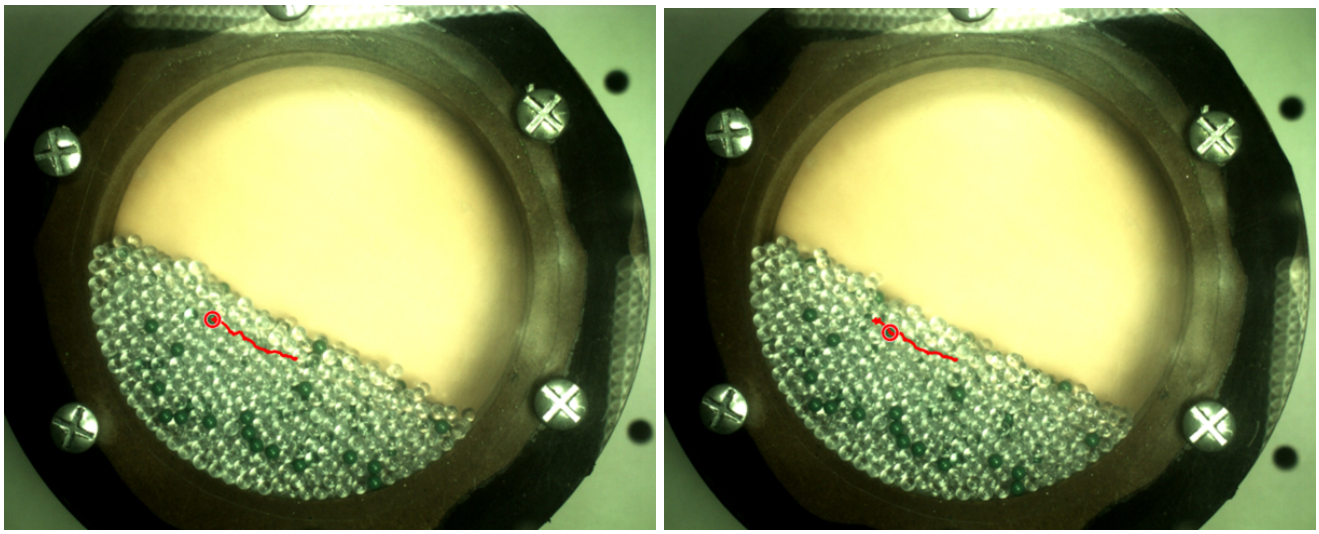


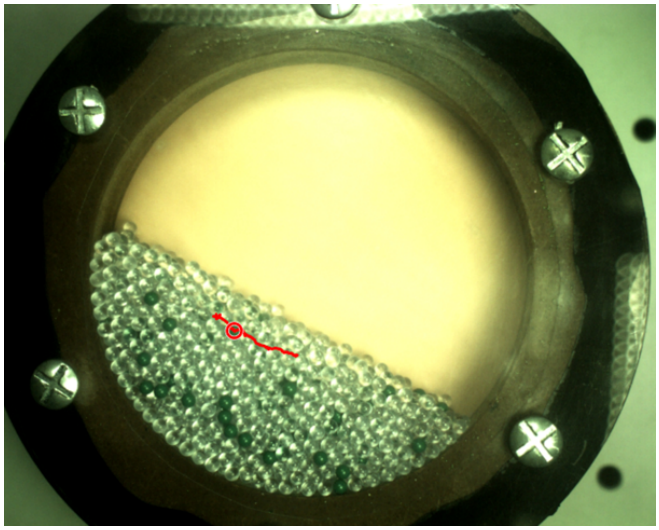
Figure 26: Tracks with an observation of at least 300 frames continuously

The accuracy of the nearest neighbour seems to be quite high, but this can only be verified by plotting the detected tracks in the corresponding frames and then visually validating the results. This is done for tracks with at least 300 frames of continuous observation. To validate the tracks, each track is displayed individually in a video with the specific particle circled. In each of these tracks, both the tracer particles and the transparent particles are detected accurately. The tracks of the tracer particles, which are denoted by the color green, are the ones that are the least difficult to validate. This is because the different color (transparent) of the surrounding particles makes it simple to determine whether or not the correct particle has been traced. As an illustration, a number of tracer tracks are shown below. The trajectory of a tracer particle is shown in Figure 27. This particular track is part of the avalanche's base layer. The red line represents the track, and the red circle represents the detected particle in the corresponding frame.

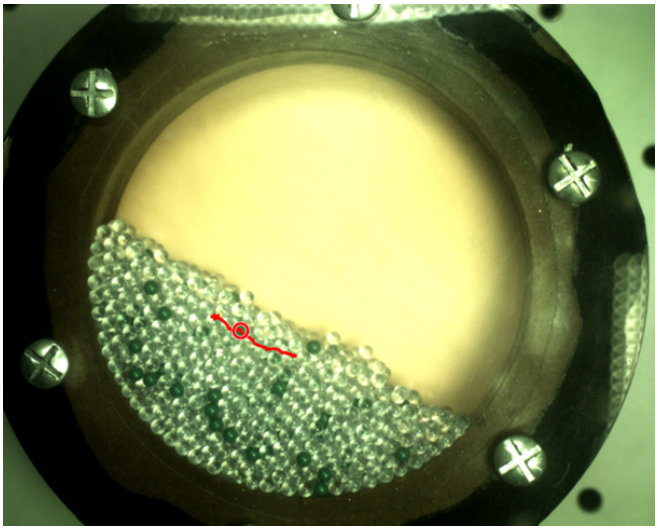


(a) frame 1

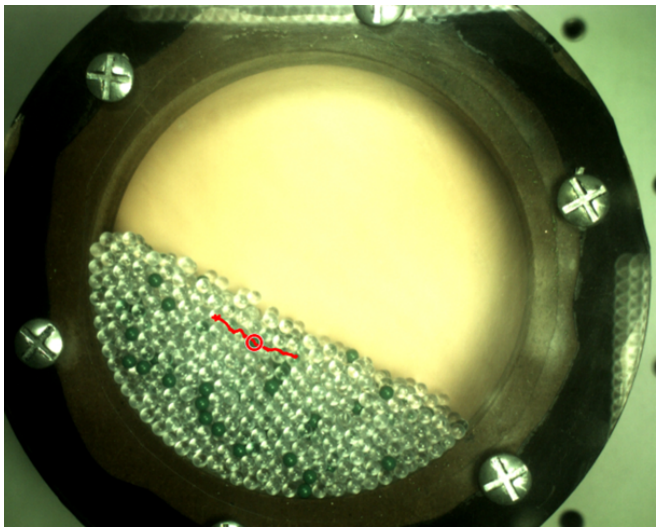
(b) frame 101



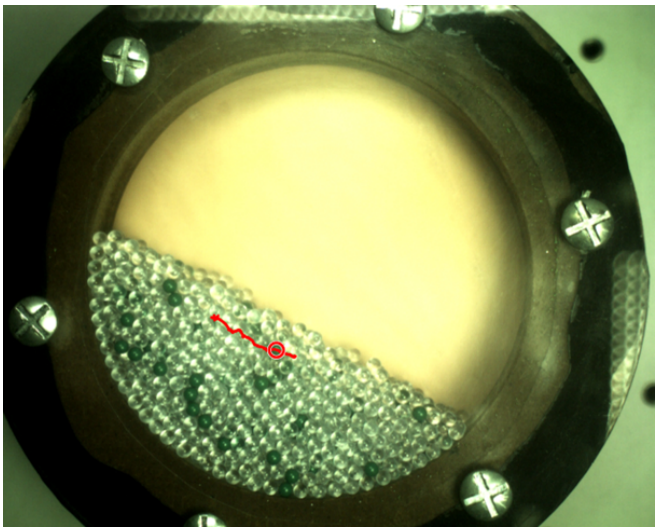
(c) frame 201



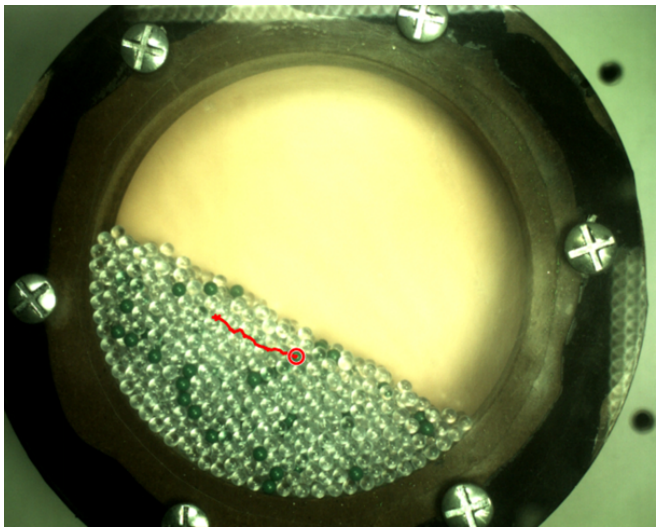
(d) frame 301



(e) frame 401



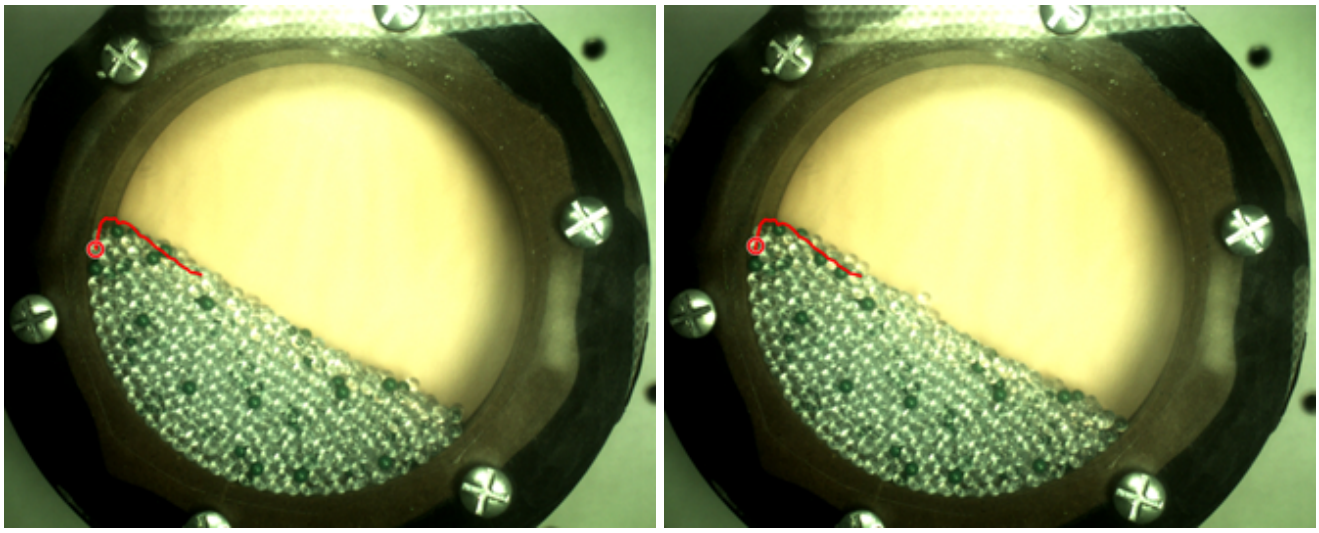
(f) frame 501



(g) frame 601

Figure 27: Tracer particle track over 601 frames, frame interval of 100, red line indicate the track and red circle show the corresponded detected particle

The top layer of particles moves more rapidly and more chaotically due to particles collision. This makes it more difficult to trace the top layer's particles. Due to the high frame rate of 450 frames per second, a part of the top layer of particles can be traced. Figure 28 illustrates the trajectory of a top layer particle spanning 331 frames with a frame interval of 30 frames. The red line represents the entire path of the particle, and the circled red dot indicates the particle detected in the particular frame. Initial observation reveals that the particle moves slowly and covers a small distance. As soon as the particle reaches the top layer and thus becomes a part of the avalanche, it accelerates to a greater speed, which is reflected in the distance it travels during the frame interval, as depicted in the final images of Figure 28. This track also demonstrates that the detection of transparent particles is effective.

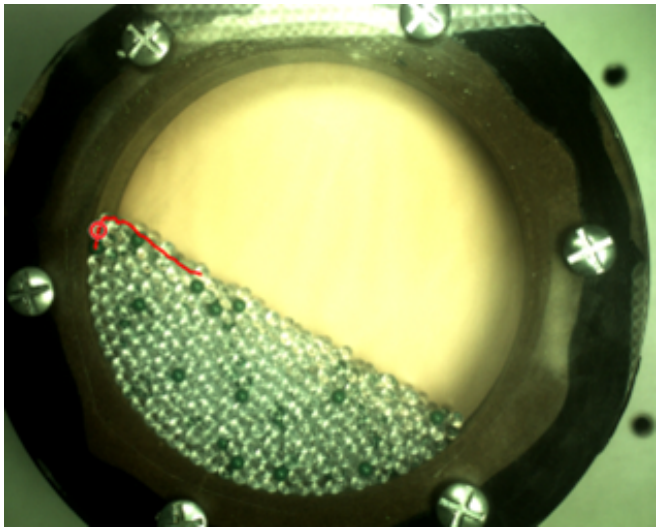


(a) frame 1

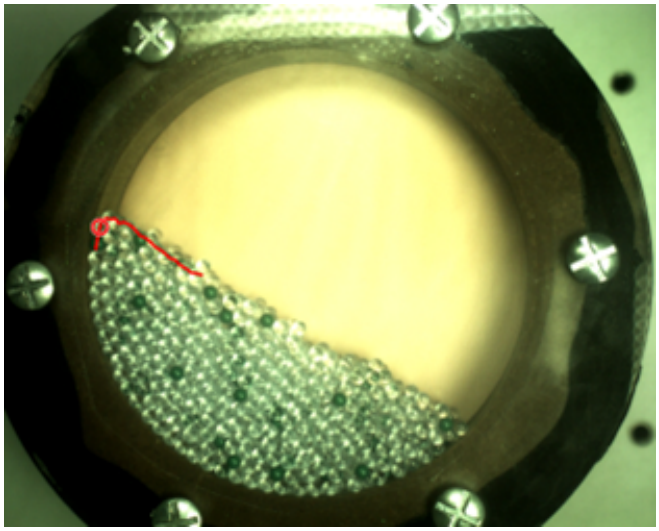
(b) frame 31

(c) frame 61

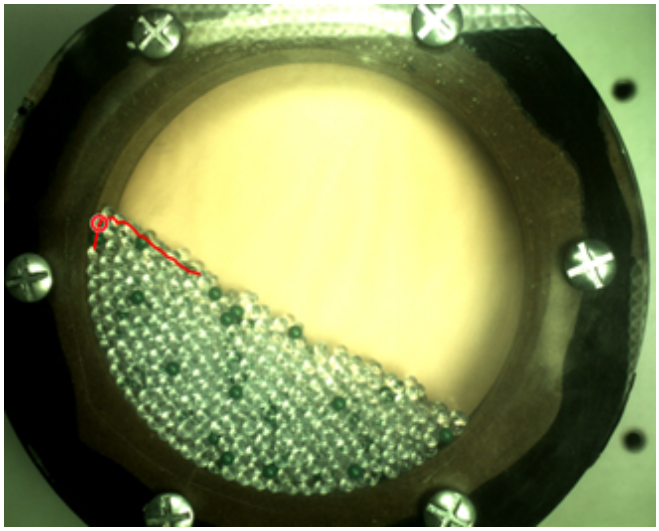
(d) frame 91



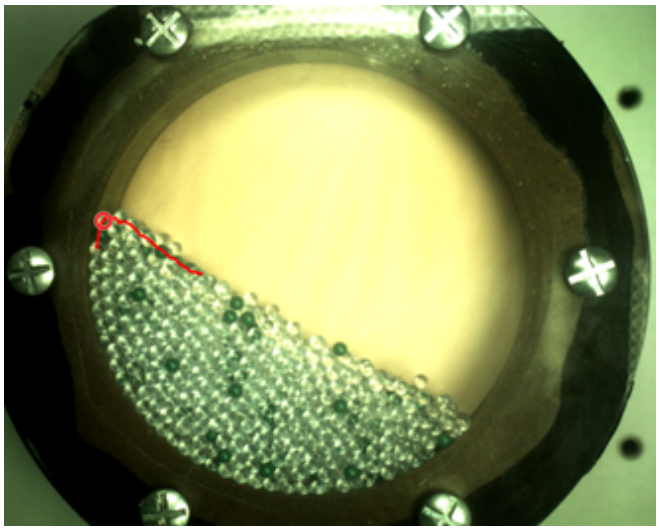
(e) frame 121



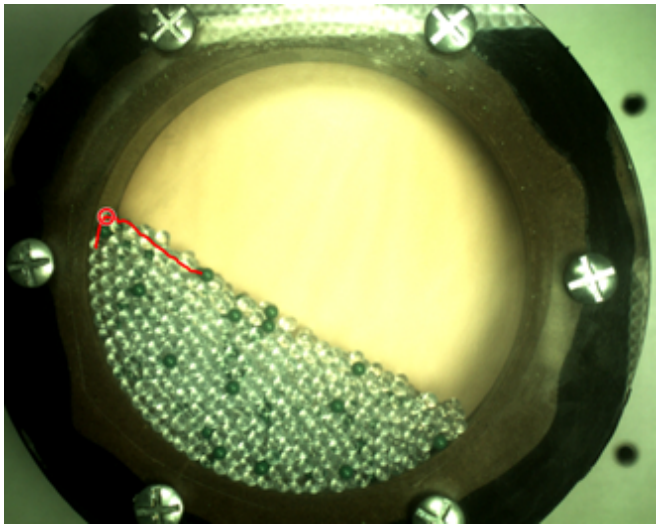
(f) frame 151



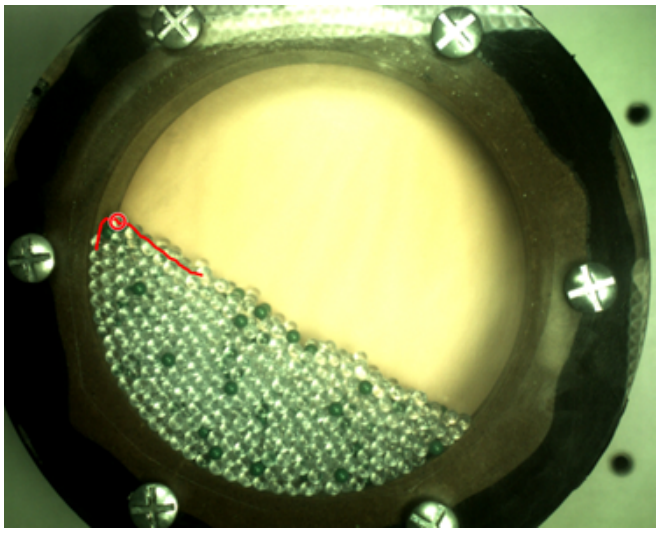
(g) frame 181



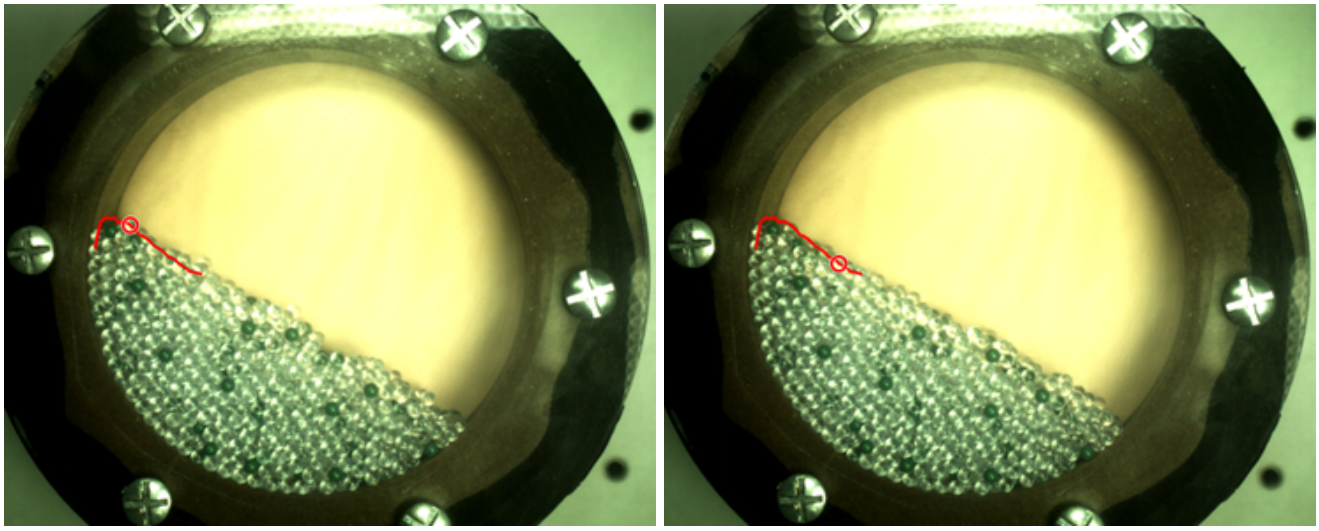
(h) frame 211



(i) frame 241



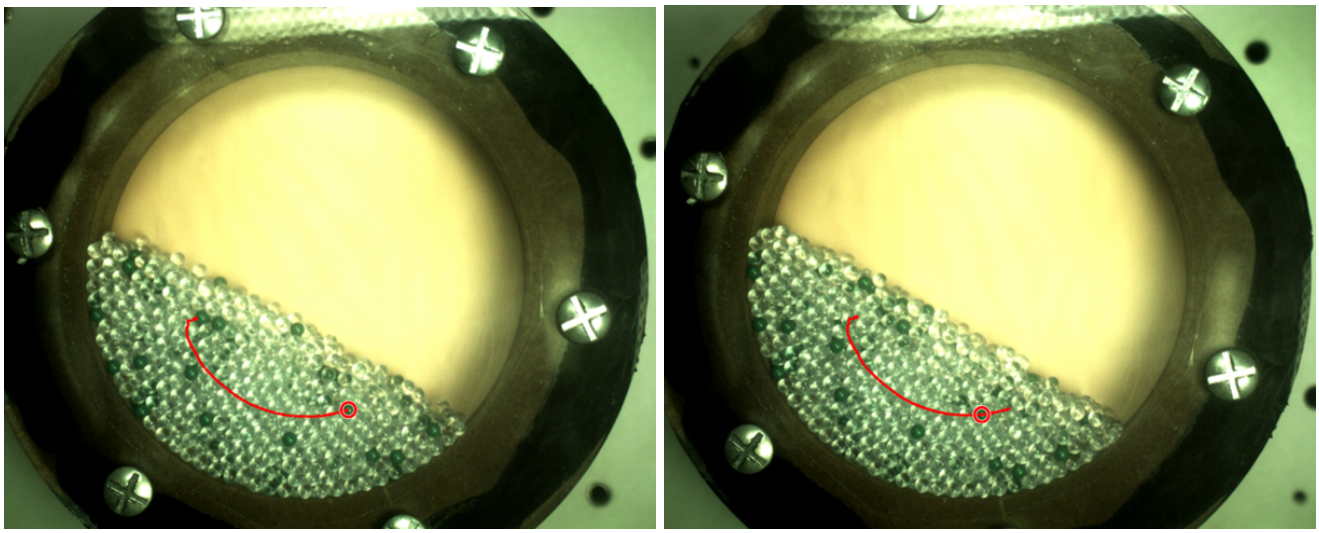
(j) frame 271



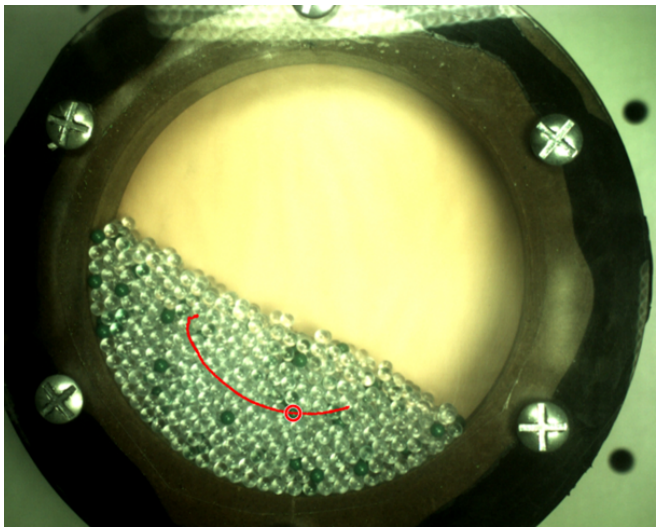
(k) frame 301 (l) frame 331

Figure 28: Particle track top layer spanning 331 frames, frame interval of 30, red line indicate the track and red circle show the corresponded detected particle

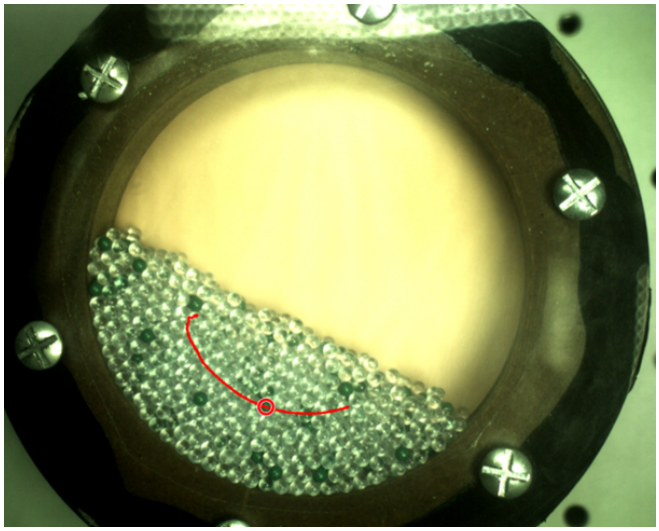
The other and final region where the particle's trajectory can be determined is the bottom of the drum. Due to the geometry and angular velocity of the drum, the particle's trajectory in this section consists of a circular upward motion. Figure 29 shows the circular trajectory of a particle that is located in this region. Similar to previous images, the trajectory and the specific particle are displayed in red colour. In the images below a frame interval of 250 is chosen to show that the specific particle is correctly detected during its track.



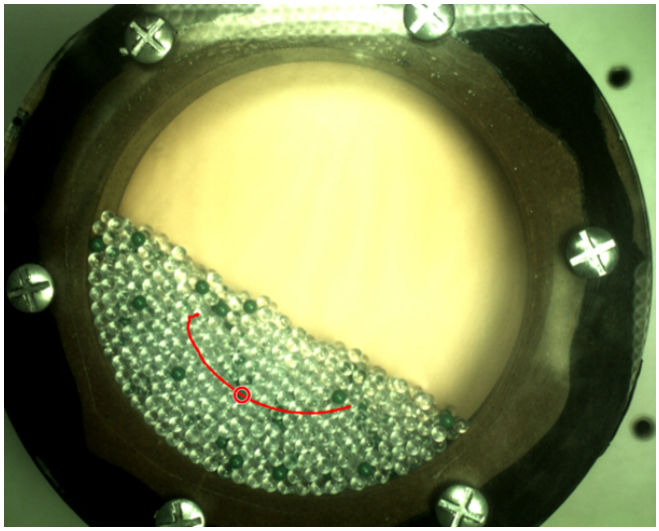
(a) frame 1 (b) frame 251



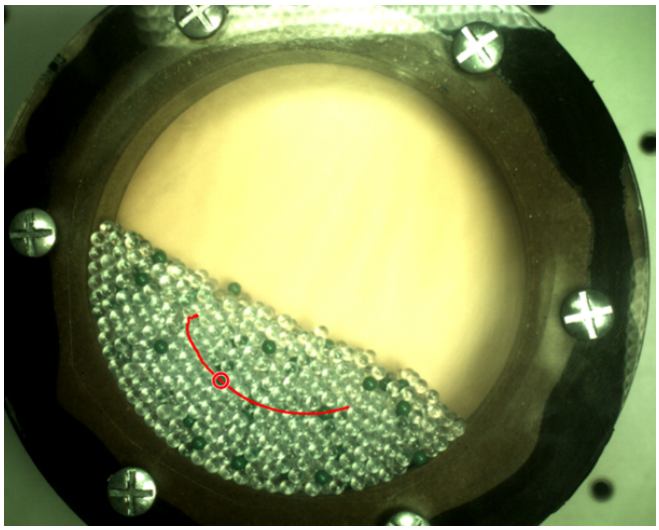
(c) frame 501



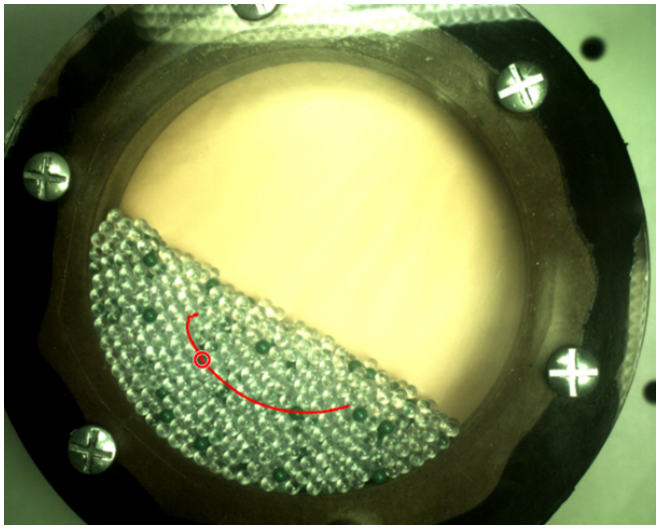
(d) frame 751



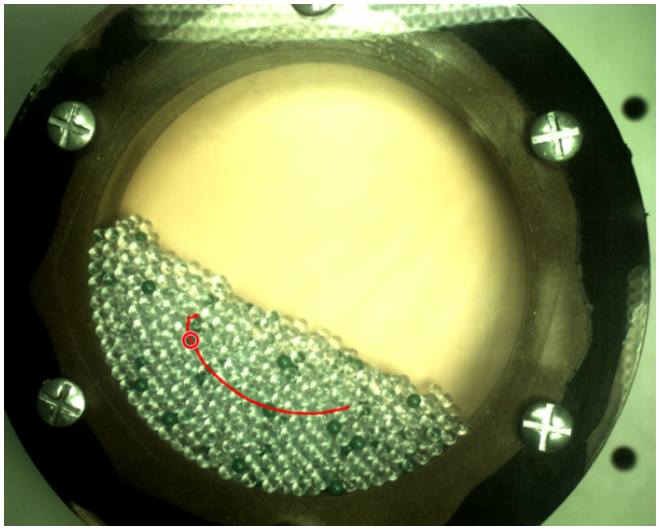
(e) frame 1001



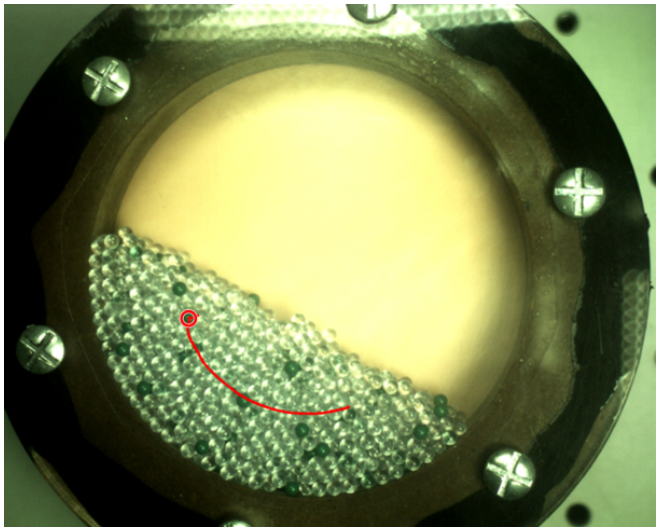
(f) frame 1251



(g) frame 1501



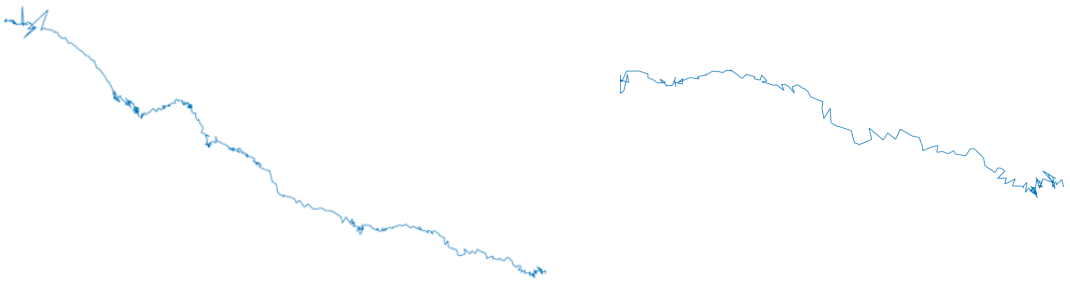
(h) frame 1751



(i) frame 2001

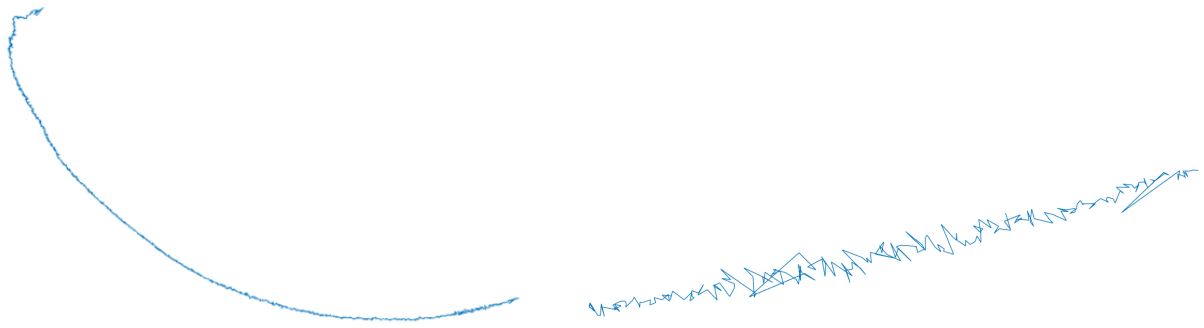
Figure 29: Particle track circular spanning 2001 frames, frame interval of 250, red line indicate the track and red circle show the corresponded detected particle

Zooming in on a single track as shown in Figure 30 demonstrates that the track fluctuates significantly. This fluctuation in the trajectory is due to the inaccuracy of the particle's detection. Using the unscented Kalman filter, the track is made more stable. Before it is used for multiple object tracking, which is discussed in the following section, the improvement of the various versions of the Kalman filter is compared using individual tracks..



(a) Single track avalanche base

(b) Single track avalanche base zoomed in



(c) Single track circular

(d) Single track circular zoomed in

Figure 30: single tracks

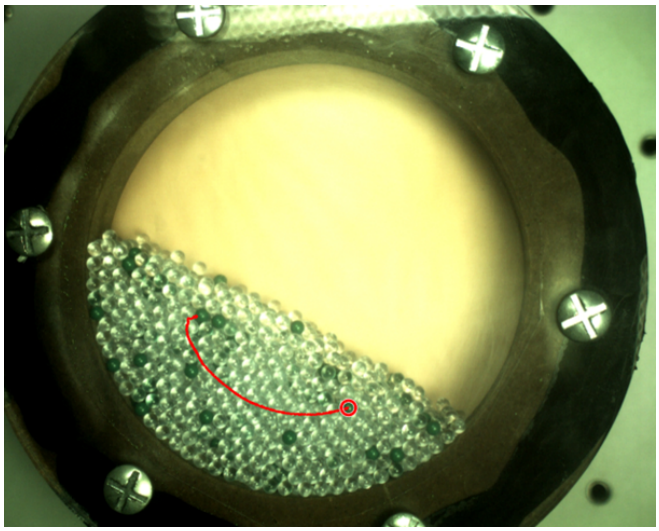
3.3 Particle tracking and predicting

To improve the track, the Kalman Filter (KF) and its variants are also applied. For the purpose of this study, the Unscented Kalman Filter (UKF) is compared to the standard linear KF. Research is also done into the Unscented Kalman Filter on Manifolds (UKF-M)[74], however in comparison with the UKF it has no benefits and therefore this method is excluded from the study. The theory of the KF and the UKF is described in the section 2.3.3. The objective is to improve the predictability and performance of the tracks. It is also investigated whether it is possible to reduce the frame rate (fps) of the video using the KF and its variants without sacrificing the quality or length of the particle tracks. Reducing the frame rate reduces the amount of data needed.

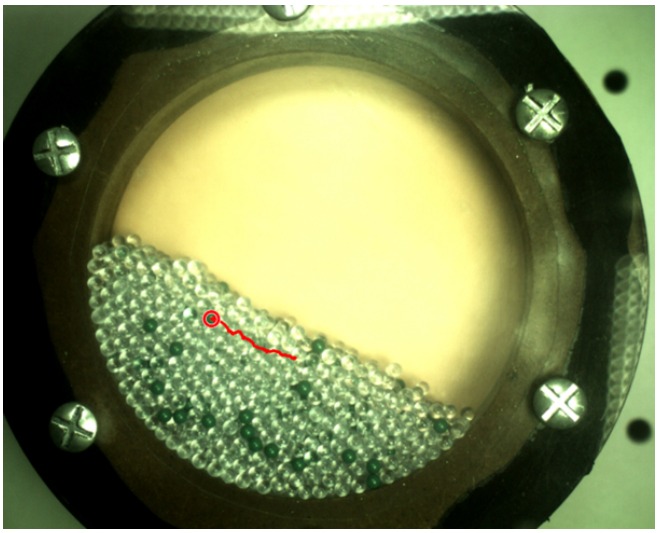
The goal is to predict for all particles, immediately after the particles are detected in a frame, where they will be in the next frame(s) so that the tracks can be created. If a particle is not detected in a subsequent frame, the prediction where the particle should have been is used until the particle is detected again within the number of allowed missing detection. The multiple tracking with the best KF variant must also ensure that there are no more mismatches so that there are no more unwanted outlier errors. Before the multiple tracking can be applied, it has first to be investigated which KF variant in combination with which model gives the best results to achieve the best results with the multiple object tracking. To evaluate the performance of the KF and the UKF, individual tracks are utilized. It is investigated how the performances are affected by missing data simulated by changing the measurement update interval. The performance of the KF and its variants are compared with a measurement update interval of the following interval sizes: one update per one frame, one update per ten frames and one update per twenty frames. It is also investigated how the tracks are predicted if there are no further measurement updates given. Different models are used to describe the motion of an object, in this case the motion of the particle, these models are described in section 2. For the linear KF, the linear model CV is used to describe the motion of a particle. The non-linear models CA and CT are used in combination with the UKF. In this chapter, the KF and the UKF with the associated models are tested for their performance, after which the method with the best performance is used for multiple tracking.

To test the performance of the different methods, they are tested on the basis of singular tracks. The singular tracks are generated with the Nearest Neighbour (NN). The disadvantage of the tracks created by the NN is that there are sometimes mismatches, resulting in relatively larger error spikes in the track. Another disadvantage is that there is not an infinite number of tracks available. The advantage of using the tracks created by the NN is that the stochastic process of the avalanche, such as the collisions between particles and the shifting of an entire layer of particles, is contained in the singular tracks. These disturbances are difficult to properly process in a simulation because they are random. For this reason it is decided to use the tracks created with the NN for testing the performance of the KF and the UKF with the corresponding models. As an illustration, three singular tracks are used

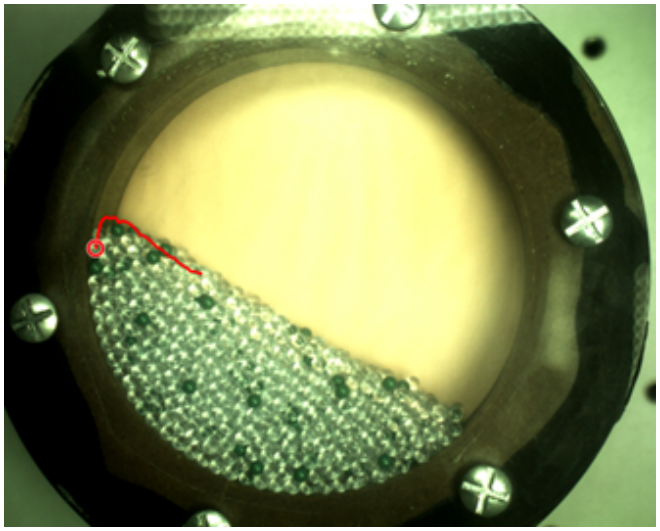
to demonstrate the performance of the KF's variants and models. The tracks are validated tracer particle tracks, so in the images below an individual particle during its track is shown (the red circle is the particle at the start of its track and the red line is the track of the particle). The three tracks indicate the different important locations: at the top, at the avalanche base and at the circular bottom. The first image shows the circular trajectory part, where the most particles are located. The second image shows the avalanche base part, this is the top layer of the particles where the particles slide down with very stochastic , this is the core of the avalanche. The third image shows the rapid change, which is when the particle moves from the circular path to the core of the avalanche. The tracks used to illustrate the performance are shown in Figure 31.



(a) Circulair track



(b) Avalanche base track



(c) Rapid change track

Figure 31: Three singular particle tracks for performance illustration, the red circle is the particle at the start of its track and the red line is the track of the particle

The methods are tested with the same singular tracks. To create images with a clear visible particle track and to depict the performance of the different KF variants the tracks are reduced to 350 observations. The effectiveness of reducing the measurement updates is explored. The goal is to test tracking accuracy when the frame rate is reduced, and to test the capacity of the performance of the KF and its variants when detection of particles is missing. Therefore several observation frequencies are tested: a measurement update for each frame, per ten frames and per twenty frames. The last fifty observations have no measurement update, so the performance of the predicting is tested. The prediction is shown with the confidence band that indicates the probability that the track is within the

predicted track. For this purpose a 95% confidence bounds is used.

3.3.1 Kalman Filter single track

The performance of the KF in combination with the CV model is described in this section. It is investigated whether the relatively simple linear KF is capable of tracing a non-linear model (the track of a particle) and correctly predicting the track.

Constant Velocity model

For the CV model there is an unknown parameter, the standard deviation of the process, the acceleration of the particle. Using trail and error it is found that a standard deviation of the acceleration with a value of $\sigma_a = 2000$ gives good results for particle tracking. First the performance of the circular trajectory is presented. Figure 32 shows the position of the particle against the KF-CV position. Here one may see the linear behavior of the KF-CV, in other words when steps are small, the filter is able to capture the particle track. When the observation increment becomes larger, the linear prediction starts deviating from the true one. This is clearly visible in the end part where a prediction of 50 observations is made.

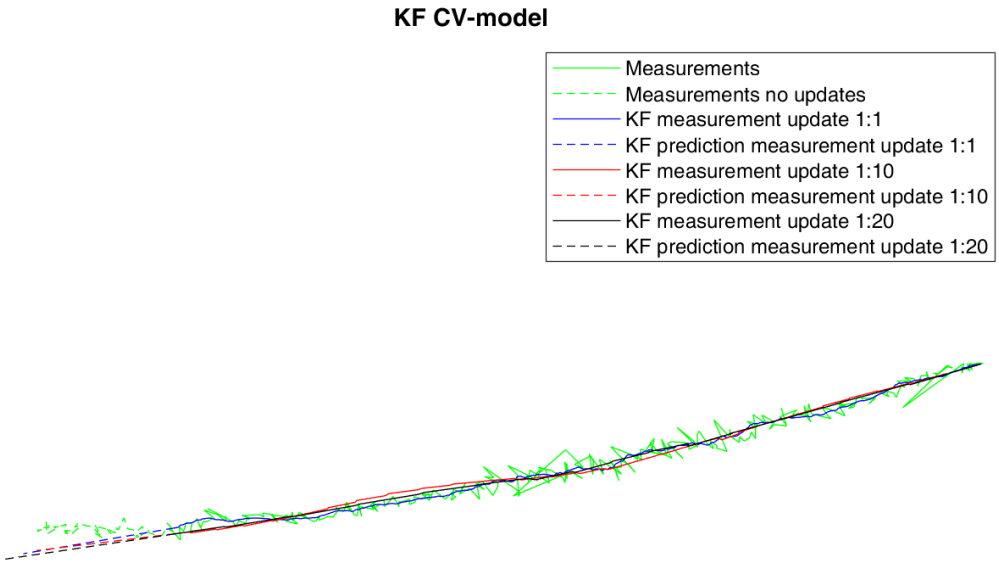


Figure 32: Performance Kalman Filter in combination with the Constant Velocity model for the circular track

Figure 33 shows the confidence band of two sigma for the X coordinates and the Y coordinates of the particle track for the KF-CV. Here it can be seen that the KF-CV has problems with keeping accurate track of nonlinear behavior, and its performance degrades as the measurement update steps become larger.

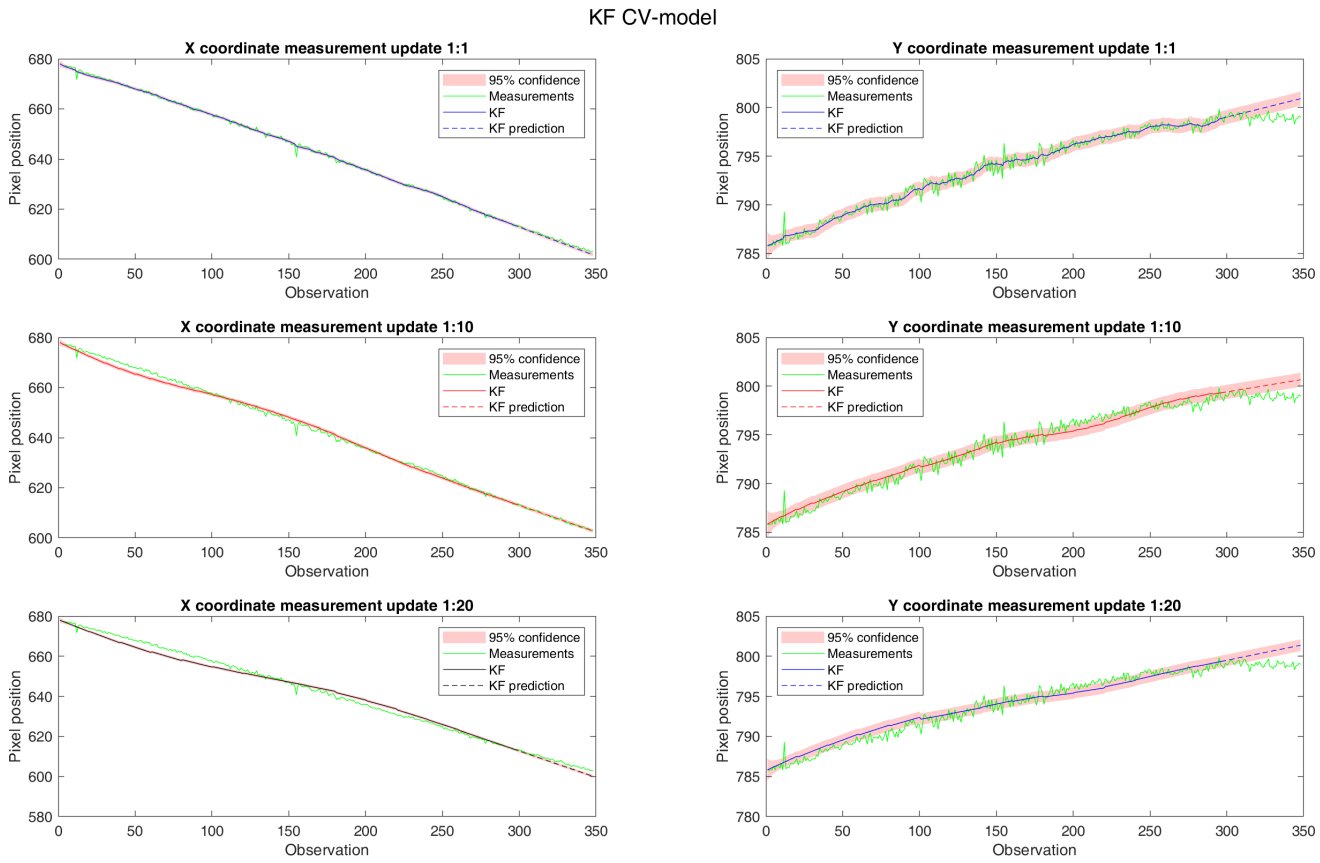


Figure 33: Confidence band X and Y position Kalman Filter in combination with the Constant Velocity model for the circular track

The relative error shows the distance between the observed position of the particle and the particle position found with the KF-CV. The distance is expressed in units of particle diameter and is shown in Figure 34

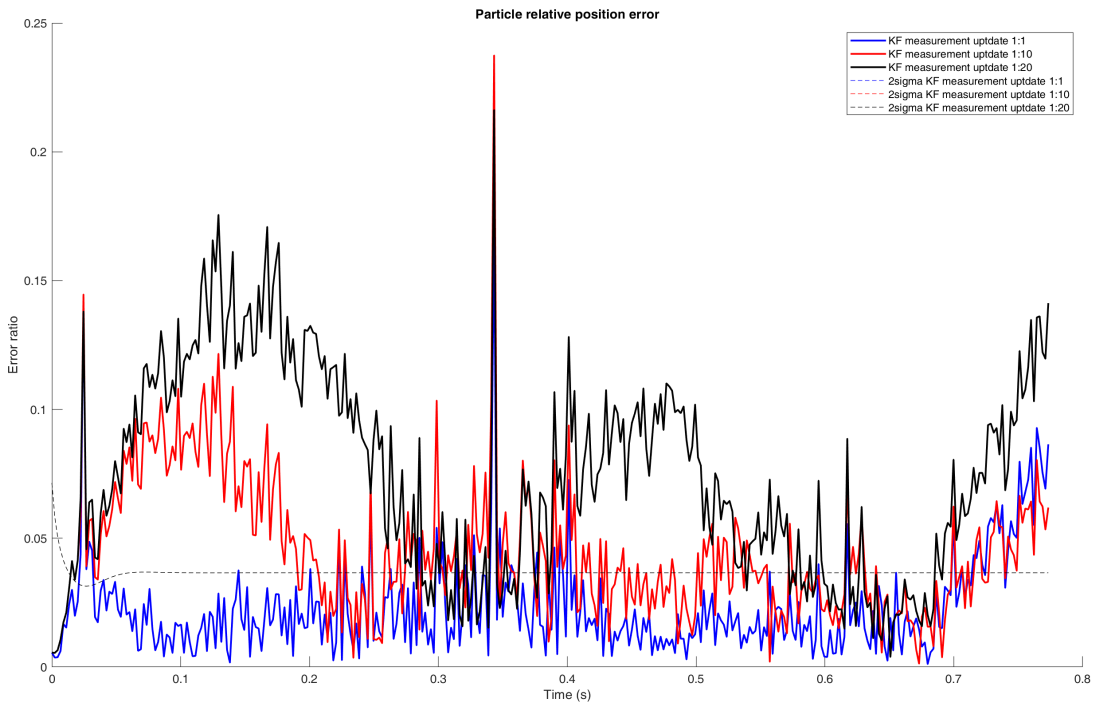


Figure 34: Relative error Kalman Filter in combination with the Constant Velocity model for the circular track

Second, the performance of the tracking and predicting for a particle in the top layer, the avalanche part, is presented. In the avalanche part, the particle moves more randomly. Figure 35 shows that the KF-CV is having trouble following the track. When the particle changes direction quickly, KF-CV loses the particle. If the KF-CV gets a measurement update every observation, it can be seen that this makes the track more stable. When predicting the last fifty observations, the KF-CV underperforms as this gives a linear prediction.

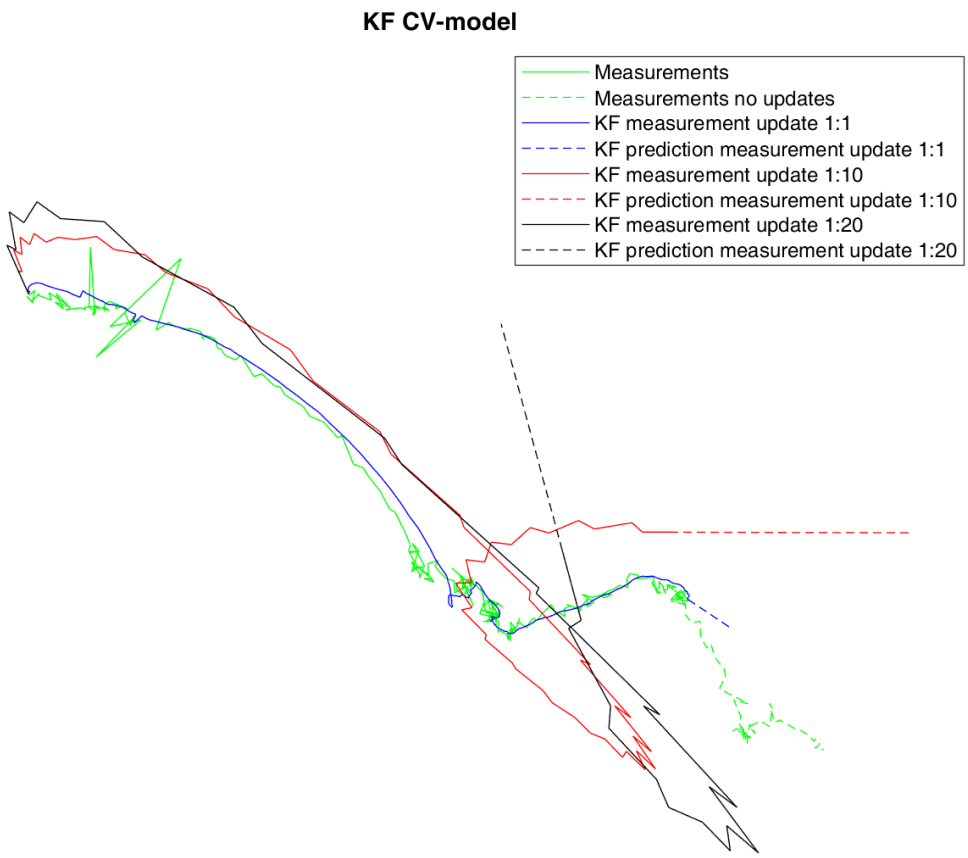


Figure 35: Performance Kalman Filter in combination with the Constant Velocity model for the avalanche track

Figure 36 shows the confidence band of two sigma for the X coordinates and the Y coordinates of the particle track for the KF-CV. It is clearly presented here that with a small step size without predictions the KF-CV has a good performance. As soon as the step size increases, the performance decreases dramatically.

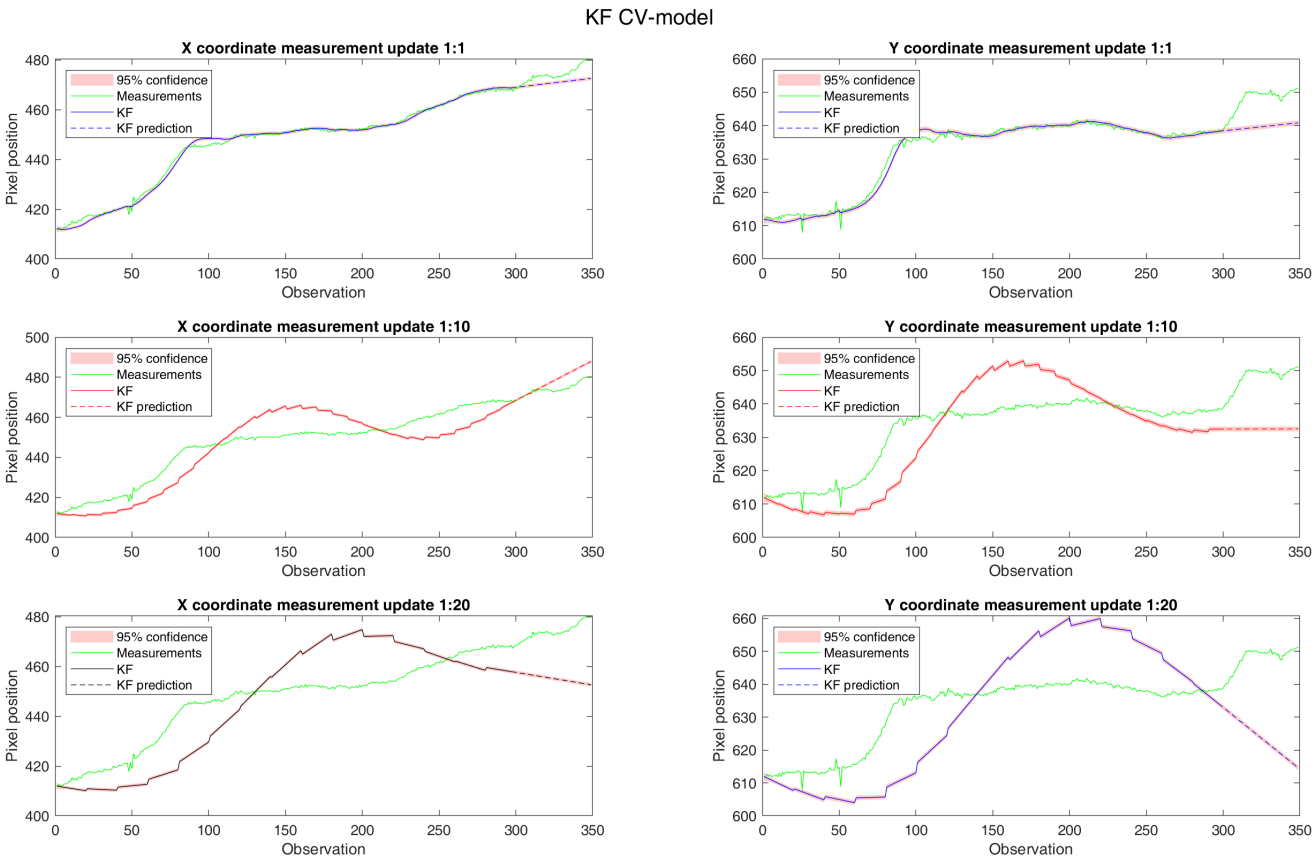


Figure 36: Confidence band X and Y position Kalman Filter in combination with the Constant Velocity model for the avalanche track

The relative error shows the distance between the observed position of the particle and the particle position found with the KF-CV. In Figure 37 it is clearly seen that as the measurement update increased the position error increases considerably which is not desirable.

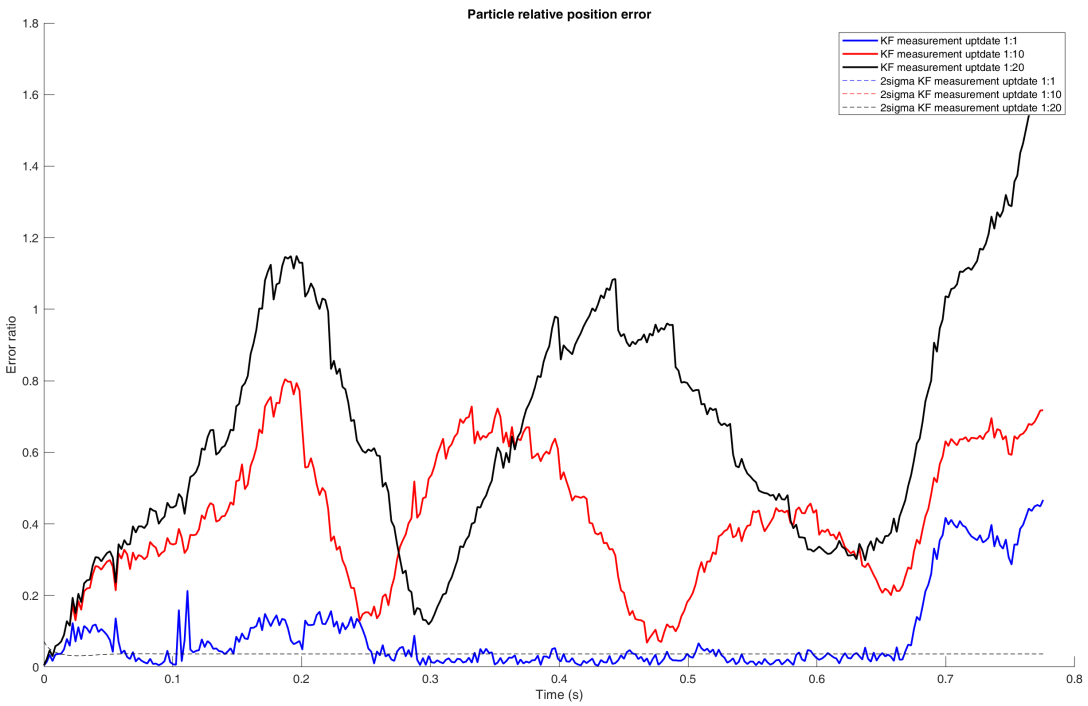


Figure 37: Relative error Kalman Filter in combination with the Constant Velocity model for the avalanche track

Finally, the results of the KF-CV for a rapid change in a track are shown. Figure 38 shows that in the initial part where the motion of the particle is quite constant, the KF tracks the particle well. When the rapid change in the track of the particle comes, the KF-CV has more difficulty keeping up with the track as the step size increases. When the model has to predict the last fifty observations, it can be seen that the KF-CV performs very poorly. This is because the particle accelerates rapidly here and makes another deflection of the track after the last measurement update.

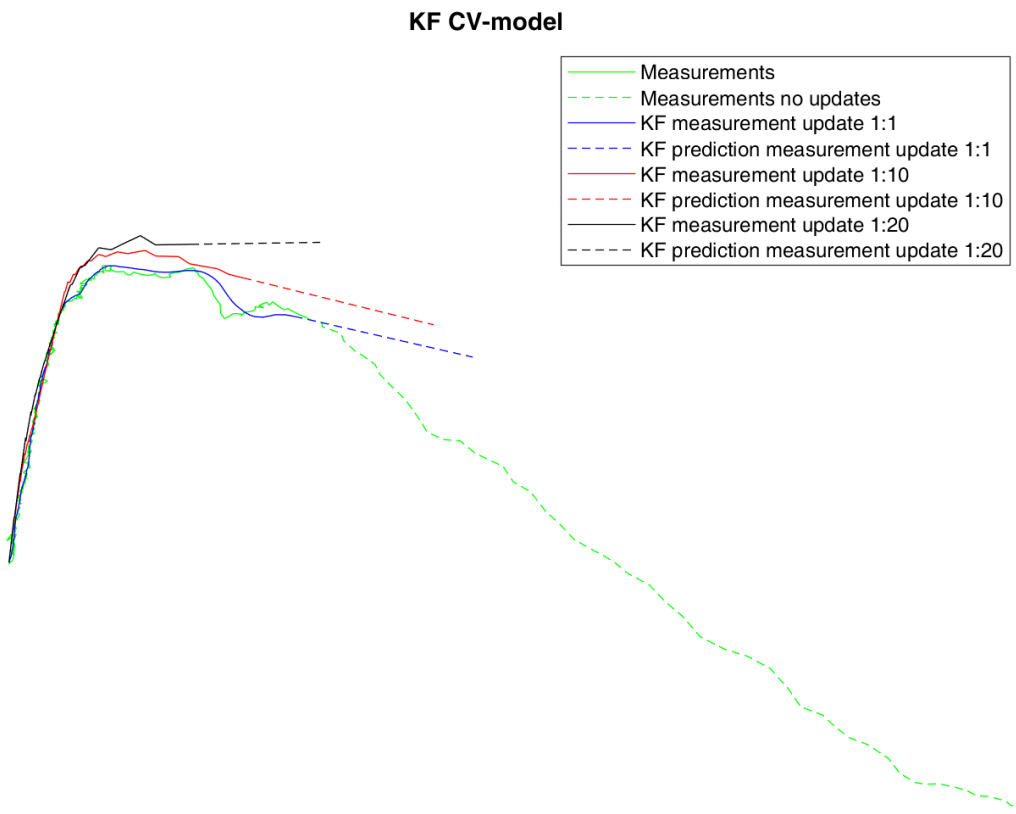


Figure 38: Performance Kalman Filter in combination with the Constant Velocity model for the rapid change track

Figure 39 shows, with the confidence band of two sigma for the X coordinates and the Y coordinates of the particle track, that the KF-CV underperforms. Especially at the end of the track where the KF-CV has to predict, the performance is very poor.

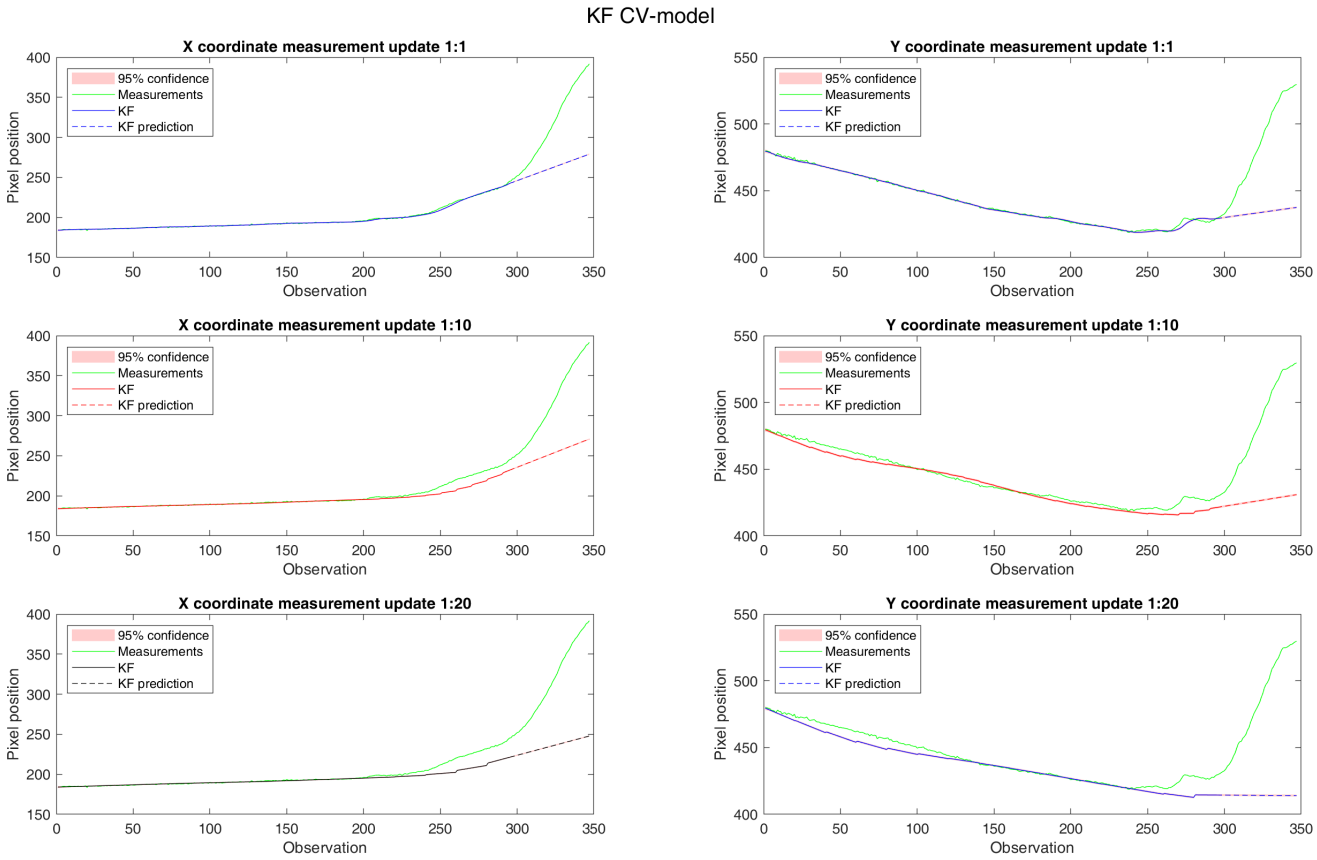


Figure 39: Confidence band X and Y position Kalman Filter in combination with the Constant Velocity model for the rapid change track

In Figure 40 the distance of the observed particle position is compared with the particle position compute with the KF-CV. The relative error is expressed in the ratio of the particle size. Figure 40 that the position error increases sharply and that the performance of the KF-CV is poor for rapid changes in the dynamics of the particle track.

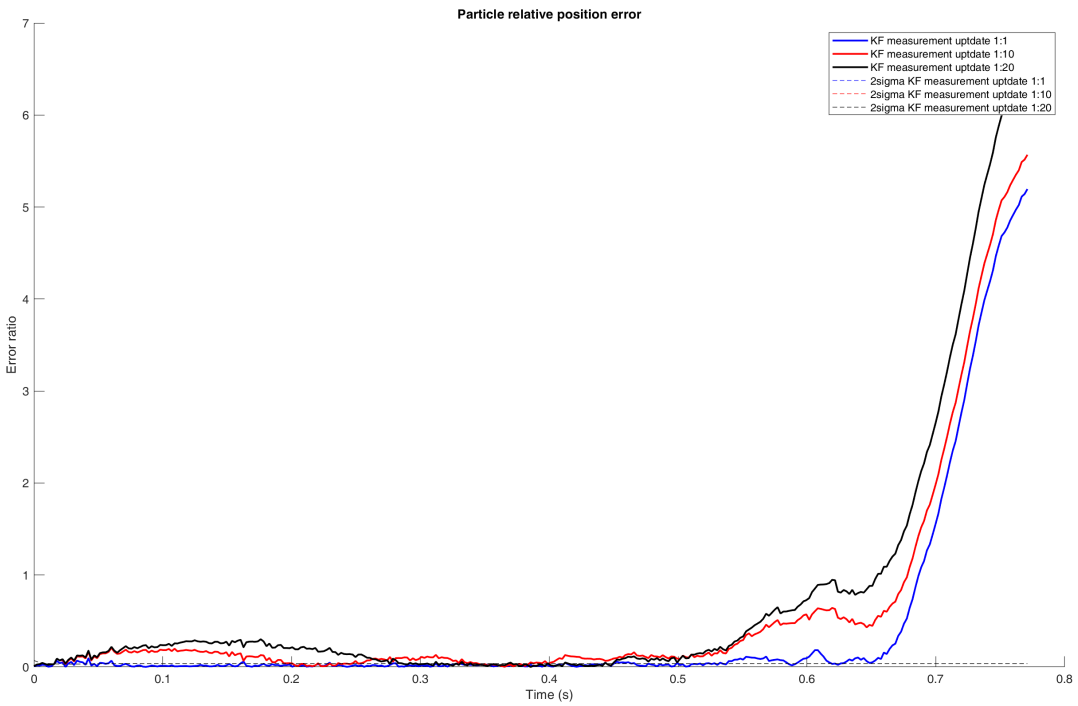


Figure 40: Relative error Kalman Filter in combination with the Constant Velocity model for the circular track

Following previous results, the KF is capable of enhancing the robustness of particle tracking. Nevertheless, the KF performs poorly when a part of the data is excluded. In addition, the linear filter does not cope well with missing data. Since the Kalman filter only makes linear predictions, there is a significant deviation when the track follows a circular path. Due to the linear behavior of the KF, it is not possible to have large distance steps between successive images, as the particle's trajectory cannot be determined just as found in [54]. Therefore, a linear approach does not adequately represent the tracks in the context of this study.

3.3.2 Unscented Kalman Filter single track:

This section describes how the UKF performed in combination with the CA and CT model.

Constant Acceleration model

For the CA model there is an unknown parameter; the standard deviation of the process, the jerk of the particle. Using trial and error it is found that a standard deviation of the jerk with a value of $\sigma_j = 5$ gives good results for particle tracking. First the performance of the UKF-CA of the circular trajectory is presented. In Figure 41 it is shown that the direction of the track is predicted correctly with the UKF-CA. It can be seen that from the point where the UKF-CA has to predict the track, the location is not correct. In the model with a measurement update of one on ten frames, it can be observed that the predicted end location is lagging behind, while in the measurement of one on one frame the end locations surpasses.

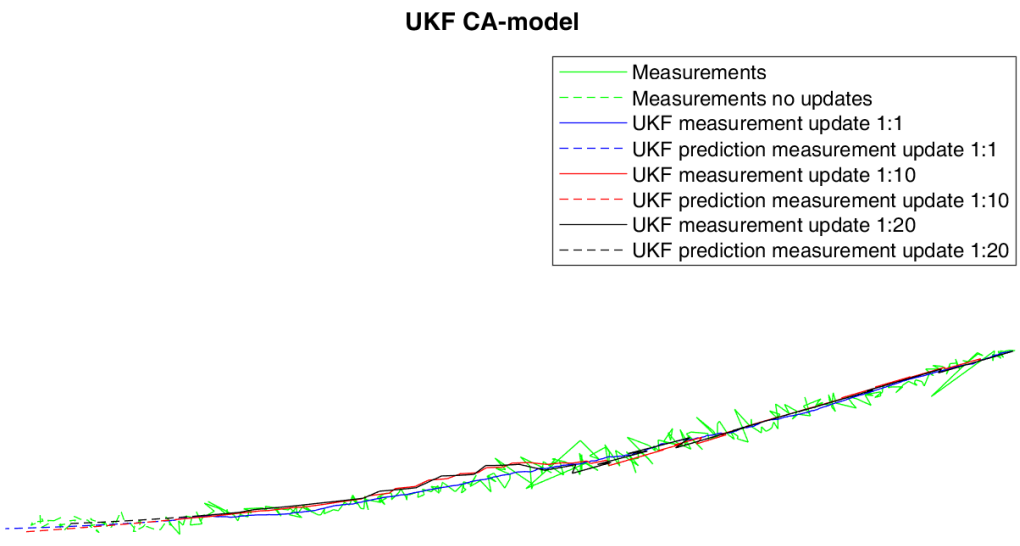


Figure 41: Performance Unscented Kalman Filter in combination with the Constant Acceleration model for the circular track

Figure 42 shows the confidence band of two sigma for the X coordinates and the Y coordinates of the particle track for the UKF-CA. It can be clearly seen that the prediction in the Y-direction performs well while the prediction in the X-direction deviates.

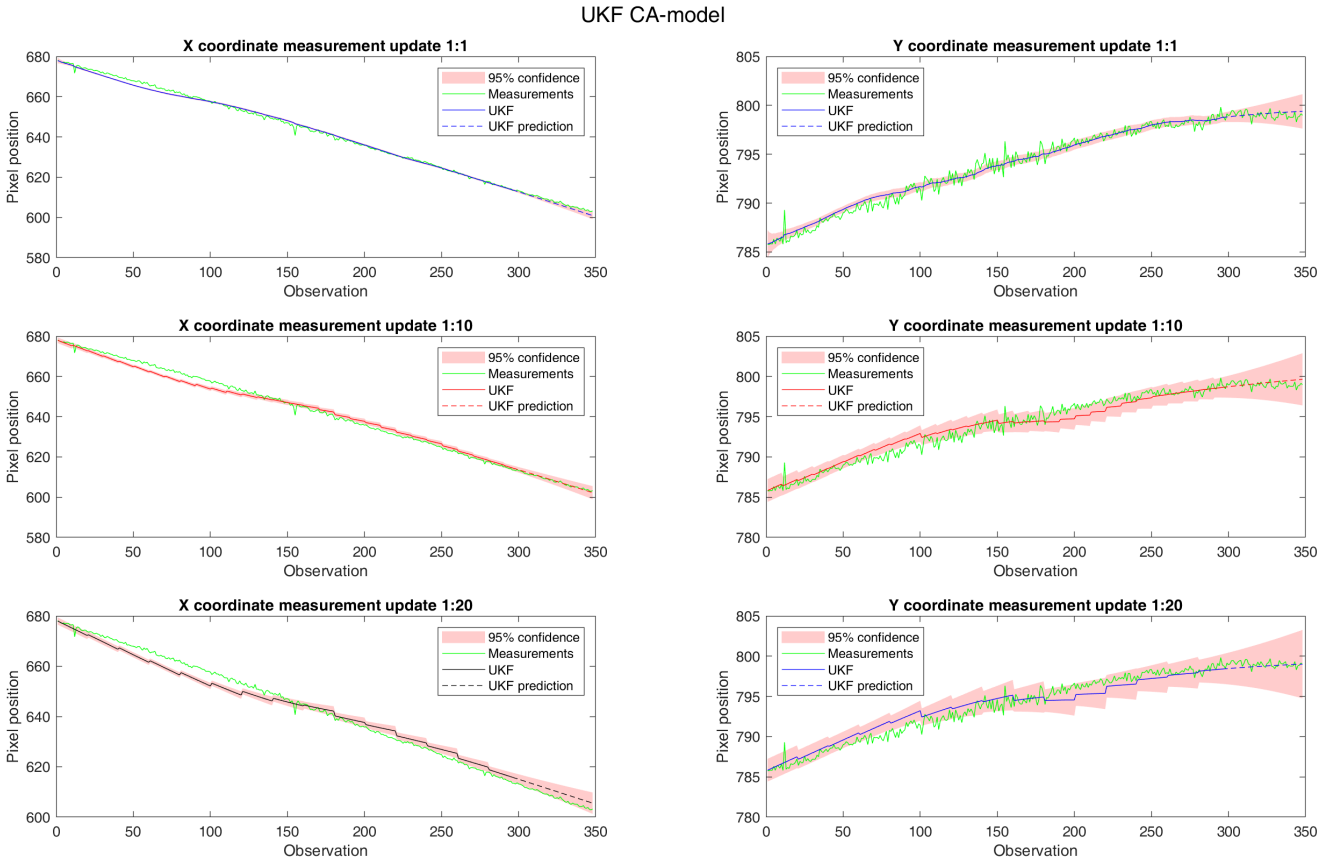


Figure 42: Confidence band X and Y position Unscented Kalman Filter in combination with the Constant Acceleration model for the circular track

The relative error shows the distance between the observed position of the particle and the particle position found with the UKF-CA. Figure 43 shows that especially in the beginning the UKF-CA thinks it is certain of the predicted position while the predicted position falls outside the confidence of 95% .

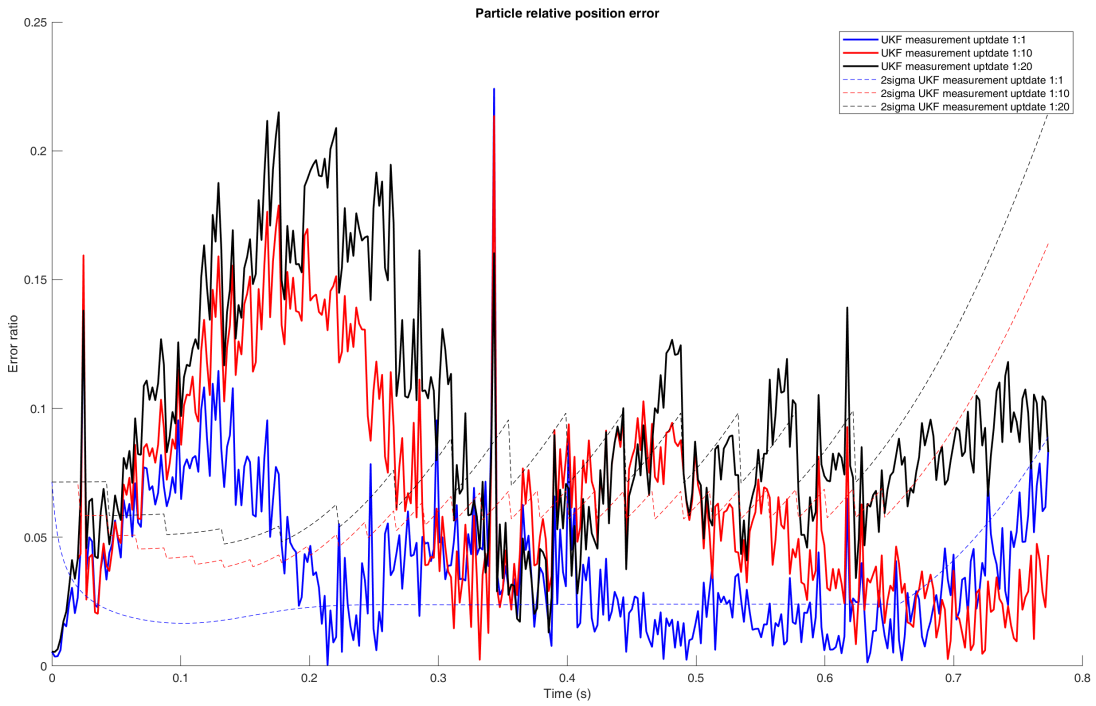


Figure 43: Relative error Unscented Kalman Filter in combination with the Constant Acceleration model for the circular track

Secondly, the performance of the tracking and predicting for a particle in the top layer, the avalanche part, is presented. Figure 44 shows that UKF-CA also has difficulties with tracking and predicting a particle that suddenly changes rapidly.

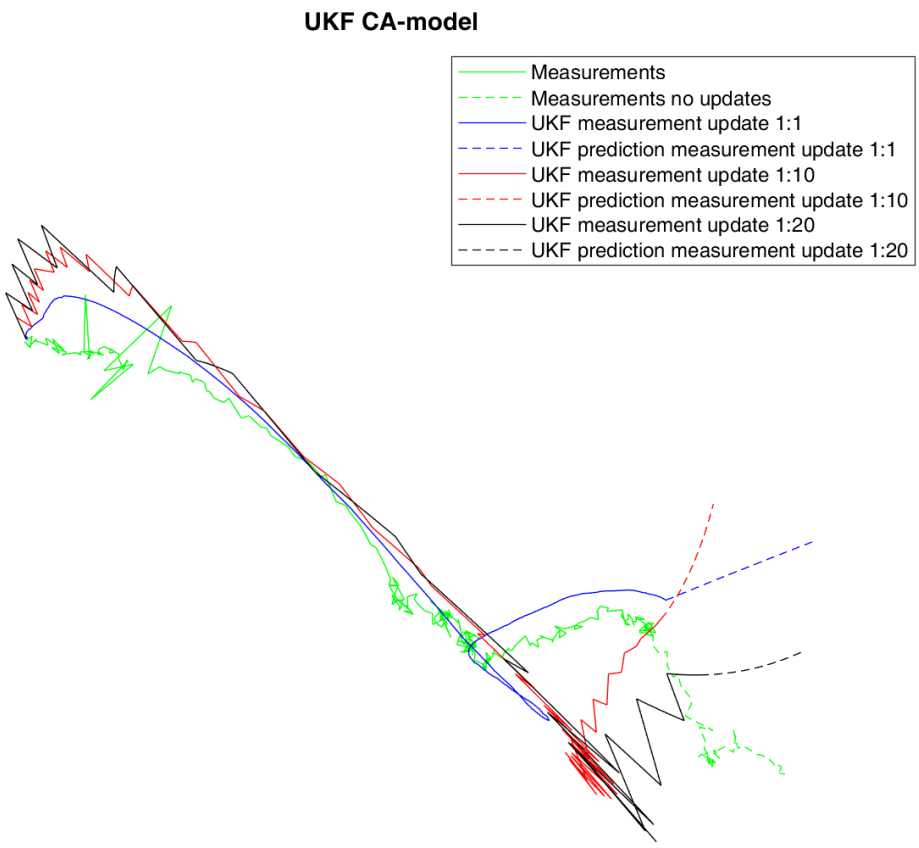


Figure 44: Performance Unscented Kalman Filter in combination with the Constant Acceleration model for the avalanche track

Figure 45 shows, with the confidence band for the X coordinates and Y coordinates of the particle track, that the performance of the UKF-CA is inadequate for a rapidly changing particle.

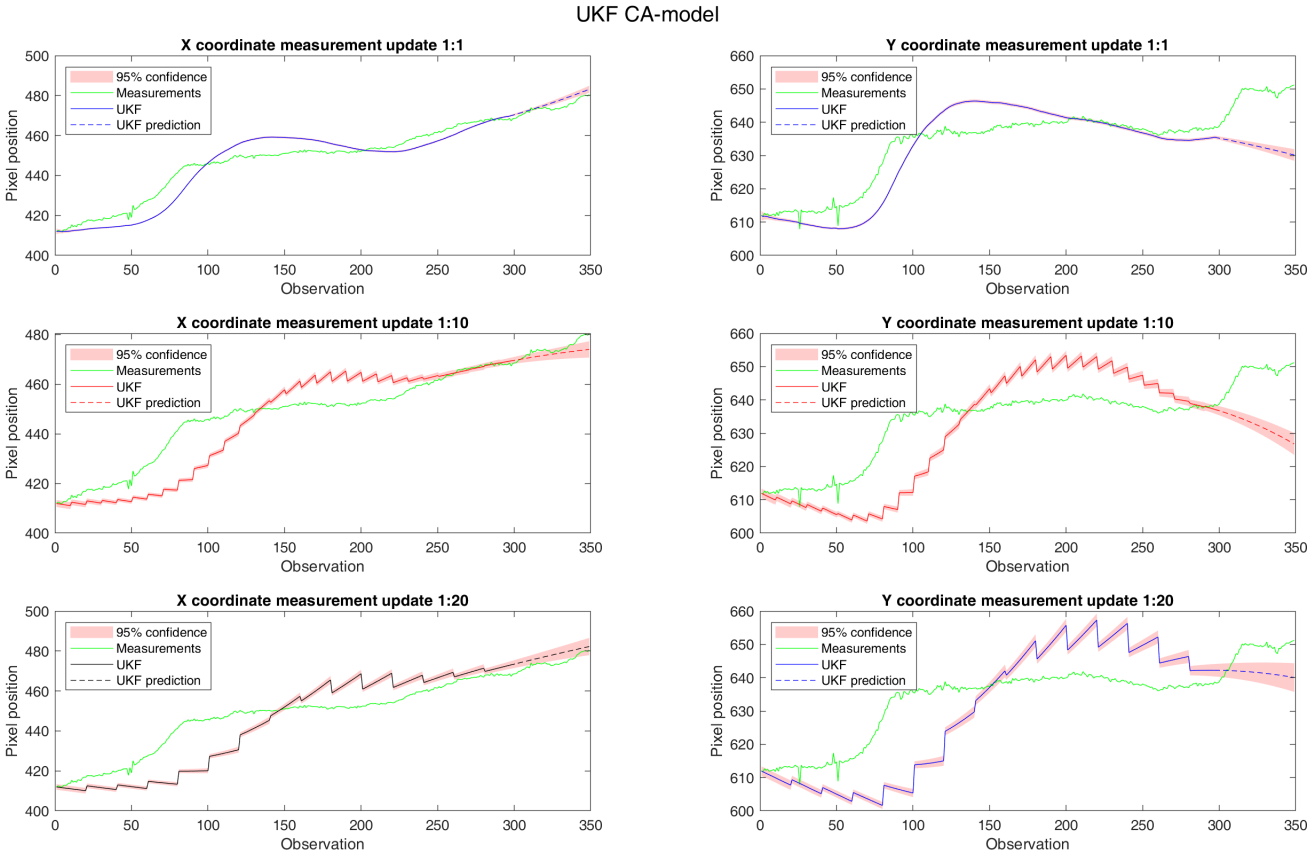


Figure 45: Confidence band X and Y position Unscented Kalman Filter in combination with the Constant Acceleration model for the avalanche track

The relative particle position error of the predicted particle position with the UKF-CA is shown in Figure 46 and confirms that the performance is poor.

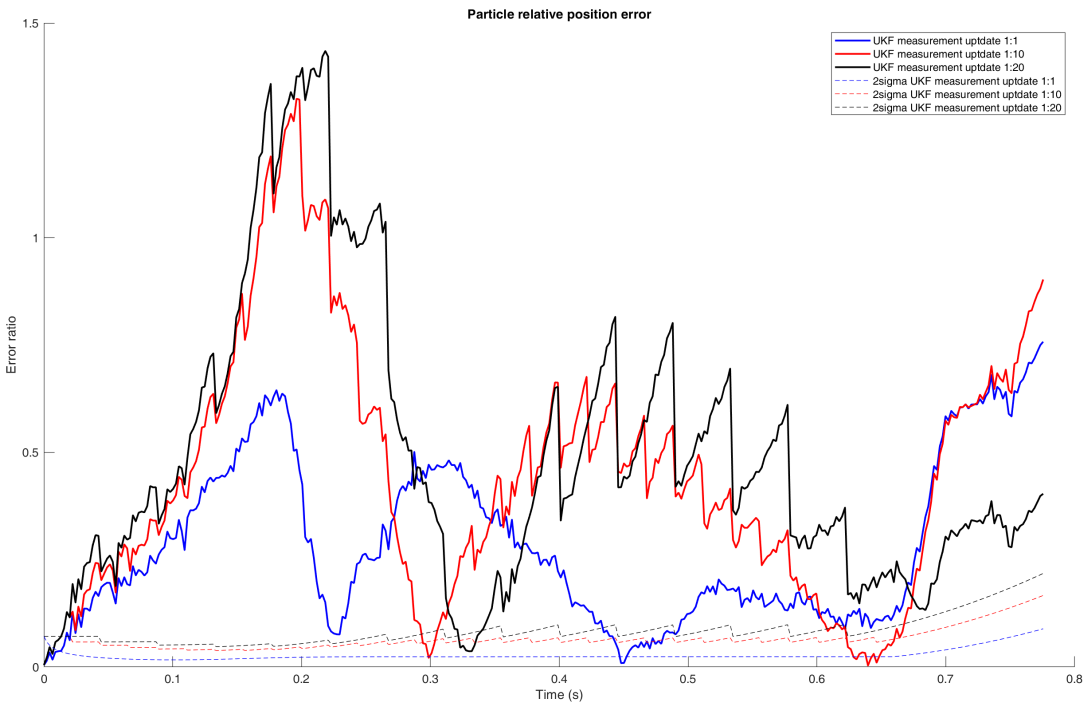


Figure 46: Relative error Unscented Kalman Filter in combination with the Constant Acceleration model for the avalanche track

Finally, the results of the UKF-CA for a rapid change in a track is shown. Figure 47 shows that the UKF-CA is capable of keeping up with the rapid track change as long as updates remain. When there are no measurement updates, while the particle changes a lot in acceleration and direction, it is not possible to continue to trace the particle with the previous knowledge.

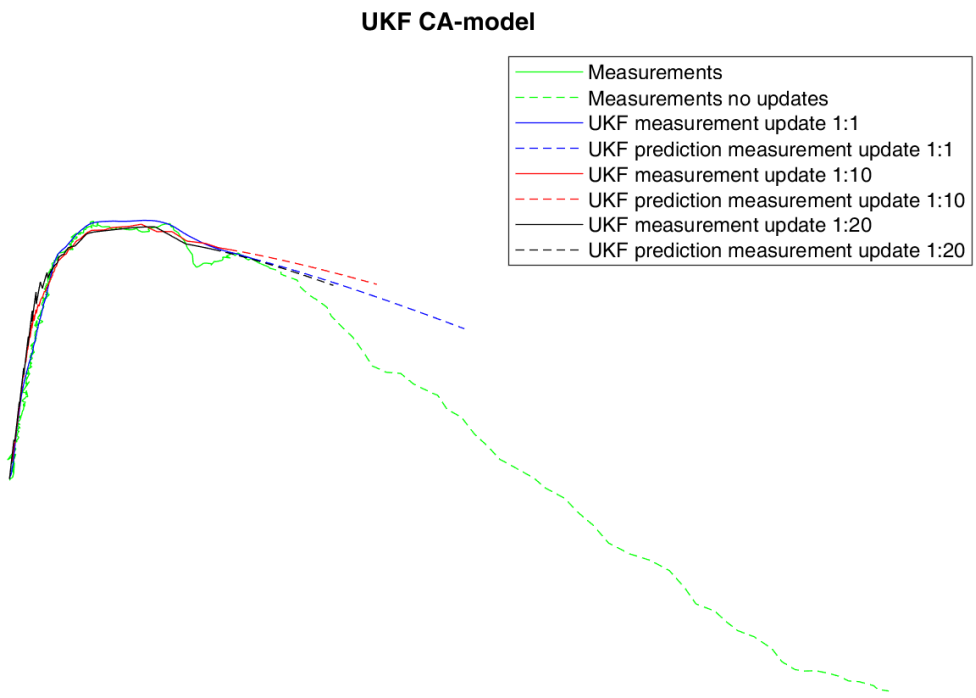


Figure 47: Performance Unscented Kalman Filter in combination with the Constant Acceleration model for the rapid change track

In Figure 48 it can be seen that the performance of the UKF-CA, when there are no more measurement updates, is poor for the rapidly changing track.

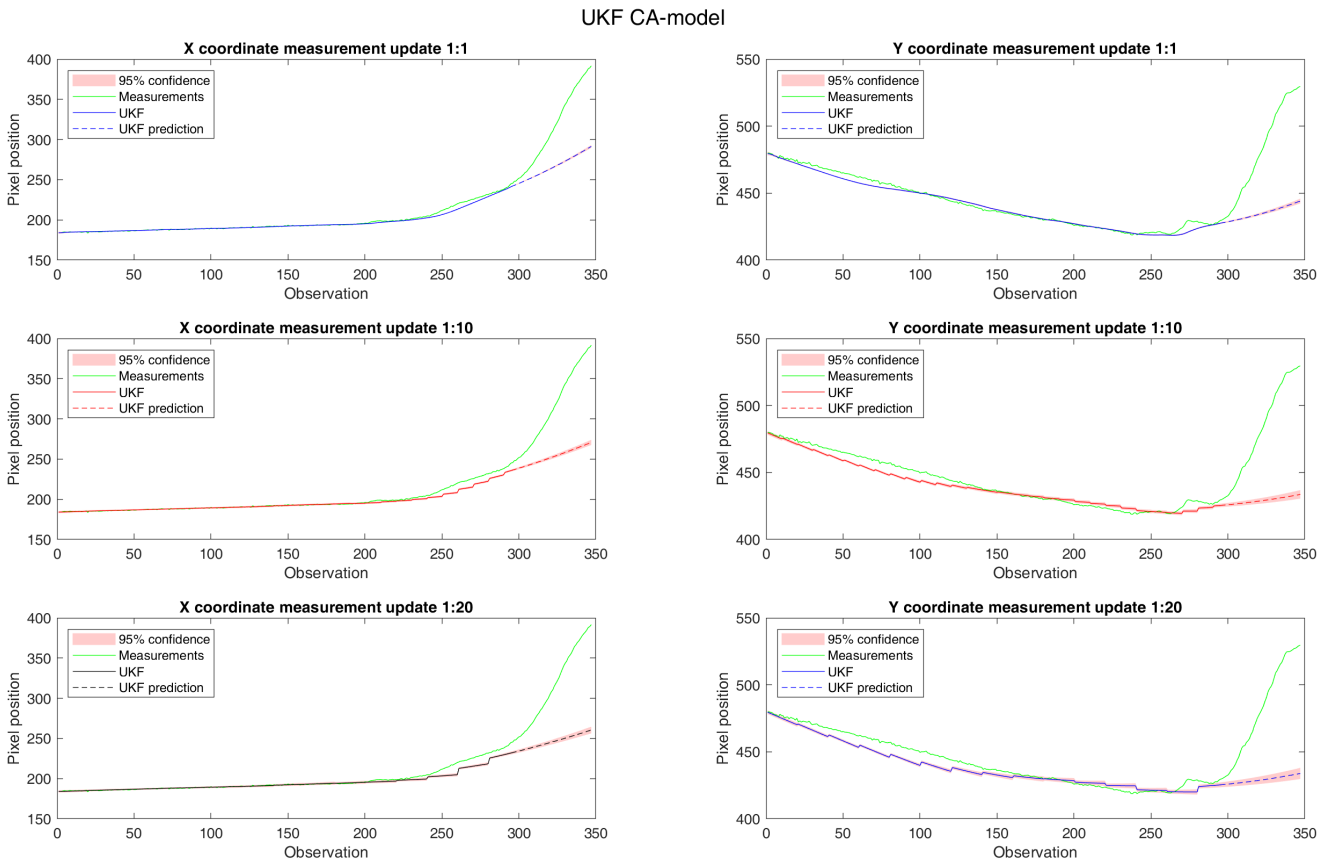


Figure 48: Confidence band X and Y position Unscented Kalman Filter in combination with the Constant Acceleration model for the rapid change track

The relative particle position error of the predicted particle position with the UKF-CA is shown in Figure 49.

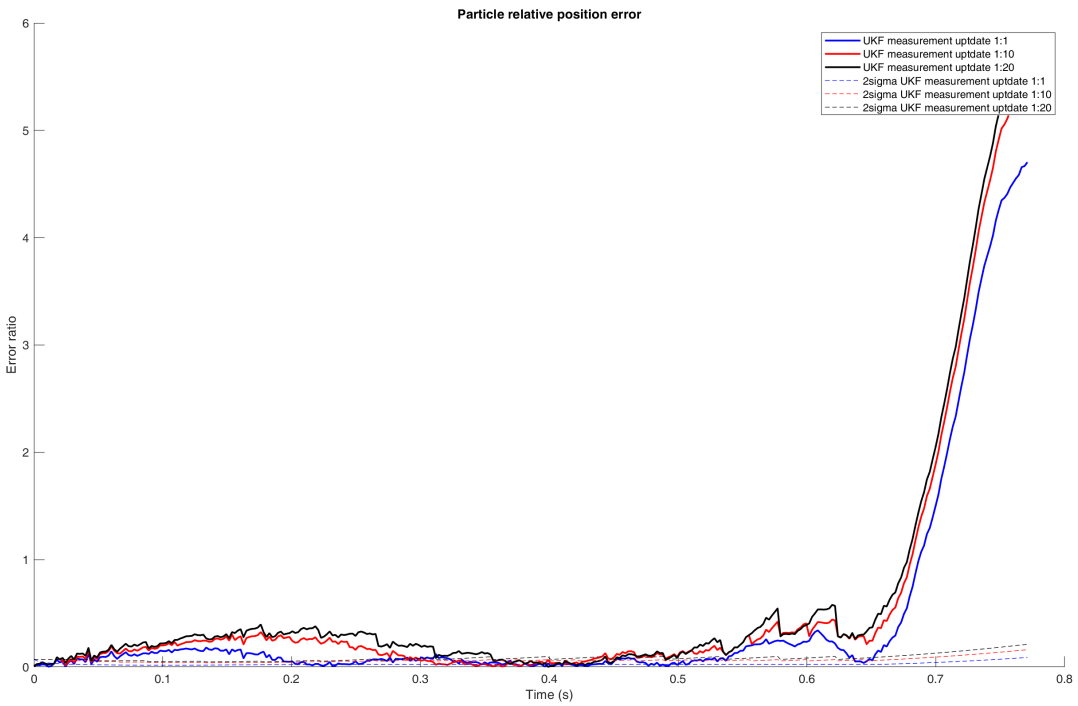


Figure 49: Relative error Unscented Kalman Filter in combination with the Constant Acceleration model for the circular track

Although the UKF-CA is capable of predicting non-linear tracks, its performance is not entirely satisfactory. The UKF-CA has difficulty predicting rapid track changes. Additionally, it has trouble with determining the correct speed of the particle, causing the predicted position to be either ahead or behind the actual position.

Coordinated Turn model

For the CT model, there are two unknown parameters, the standard deviation of the process, the acceleration of the particle and the angular acceleration. Using trail and error it is found that a standard deviation of the acceleration and the angular acceleration with a value of $\sigma_a = 2000$, $\sigma_\theta = 5$ gives good results for particle tracking.

Figure 50 presents the performance of the UKF-CT of the circular trajectory. The overall performance of the UKF-CT is good. This is clearly reflected in the predictive part of the last fifty observations. This shows that the deviation is minimal and that the UKF-CT model is also capable of predicting in a circular trajectory, which is highly desirable.

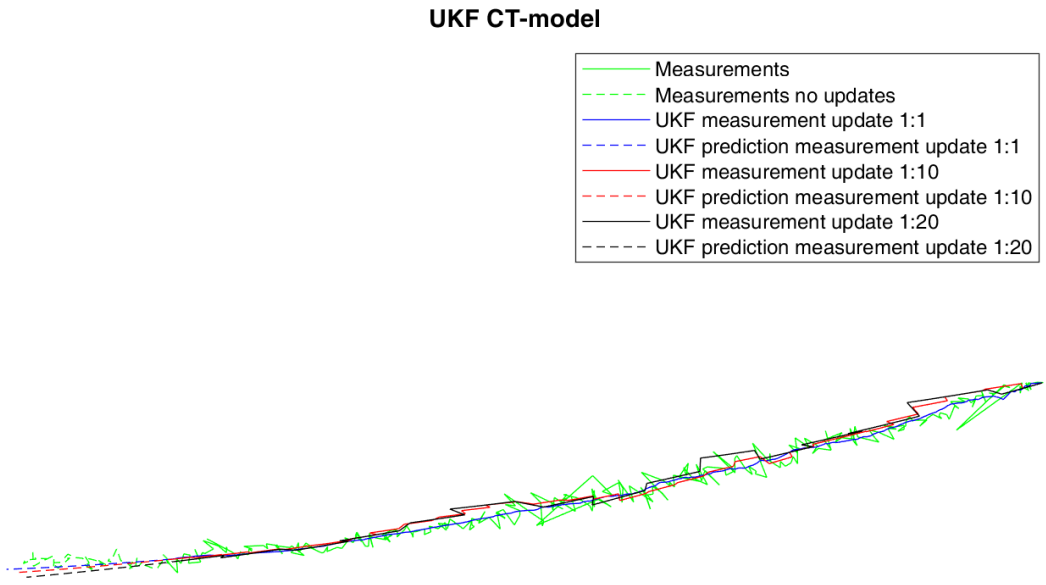


Figure 50: Performance Unscented Kalman Filter in combination with the Coordinated Turn model for the circular track

Figure 51 shows the confidence band of two sigma for the X coordinates and the Y coordinates of the particle position predicted with the UKF-CT. It can be seen clearly that the predicted track performed well.

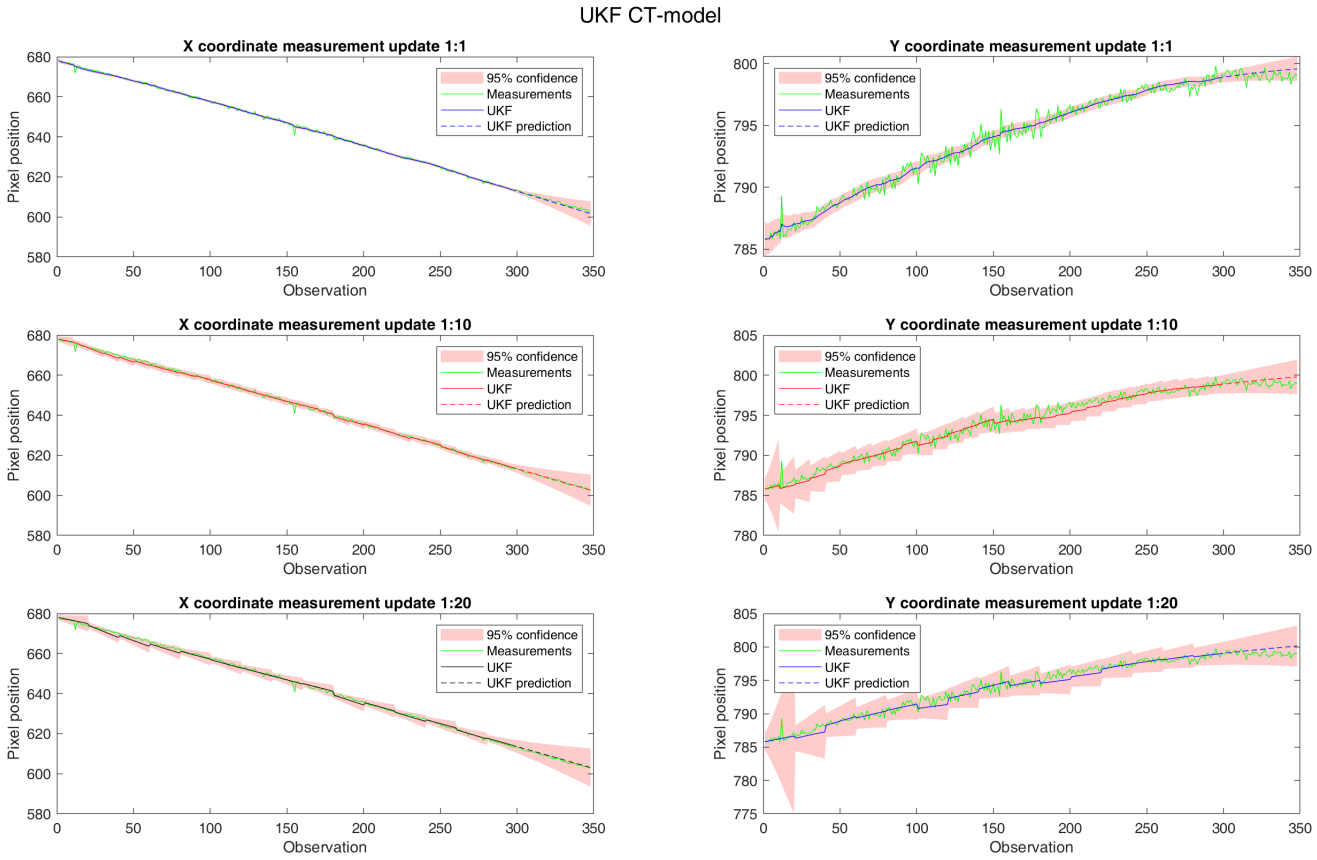


Figure 51: Confidence band X and Y position Unscented Kalman Filter in combination with the Coordinated Turn model for the circular track

The relative position error of the with the UKF-CT predicted particle position compared to the observed position is shown in Figure 52. It can be seen that the performance of the UKF-CT is very good even when measurement updates are not available for every frame. The relative position error also remains within 95% confidence. The few outliers in the position error are due to misdetections which are well filtered out by the UKF-CT.

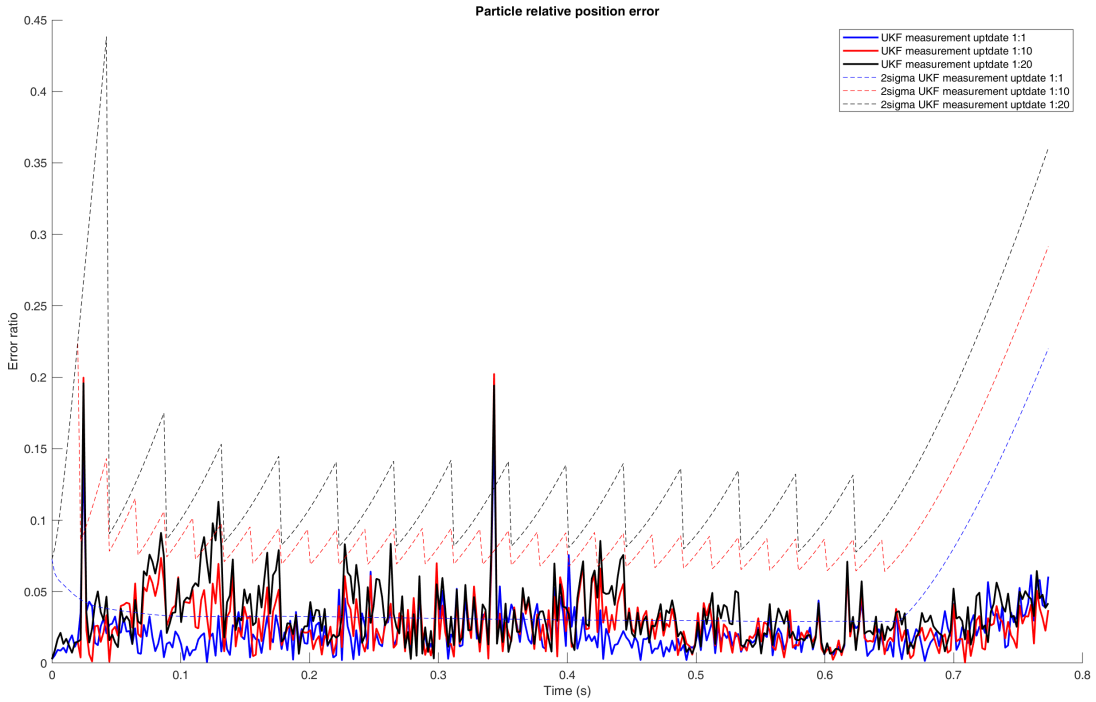


Figure 52: Relative error Unscented Kalman Filter in combination with the Coordinated Turn model for the circular track

The performance of the tracking and predicting for a particle in the top layer, the avalanche part, is presented for the UKF-CT in Figure 53. The performance is quite good, it can be seen that when the measurement update interval increases, the performance is less good in the beginning. This has to do with the fact that the mismatch of the track created with the NN is included in the measurement update, which has an adverse effect. When measurement updates are no longer available, it can be seen that the track of the particle is not predictable if there are rapid changes of track direction.

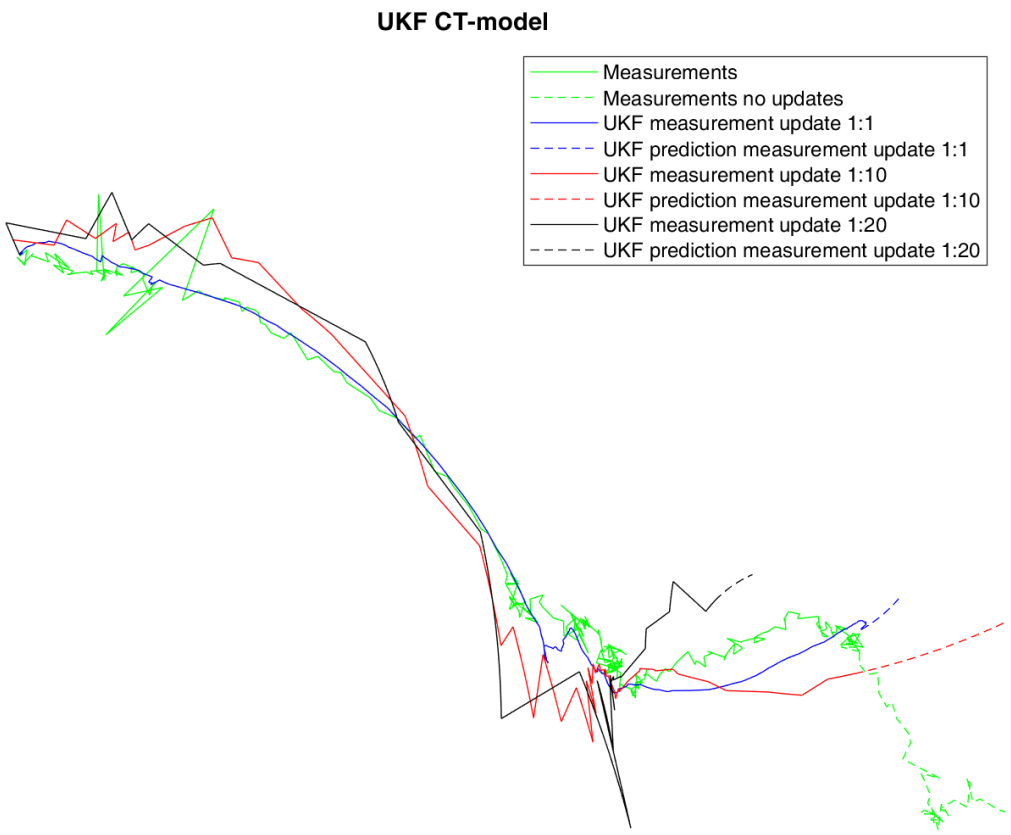


Figure 53: Performance Unscented Kalman Filter in combination with the Coordinated Turn model for the avalanche track

Figure 54 shows the confidence band of two sigma for the X coordinates and the Y coordinates of the particle track position for the UKF-CT.

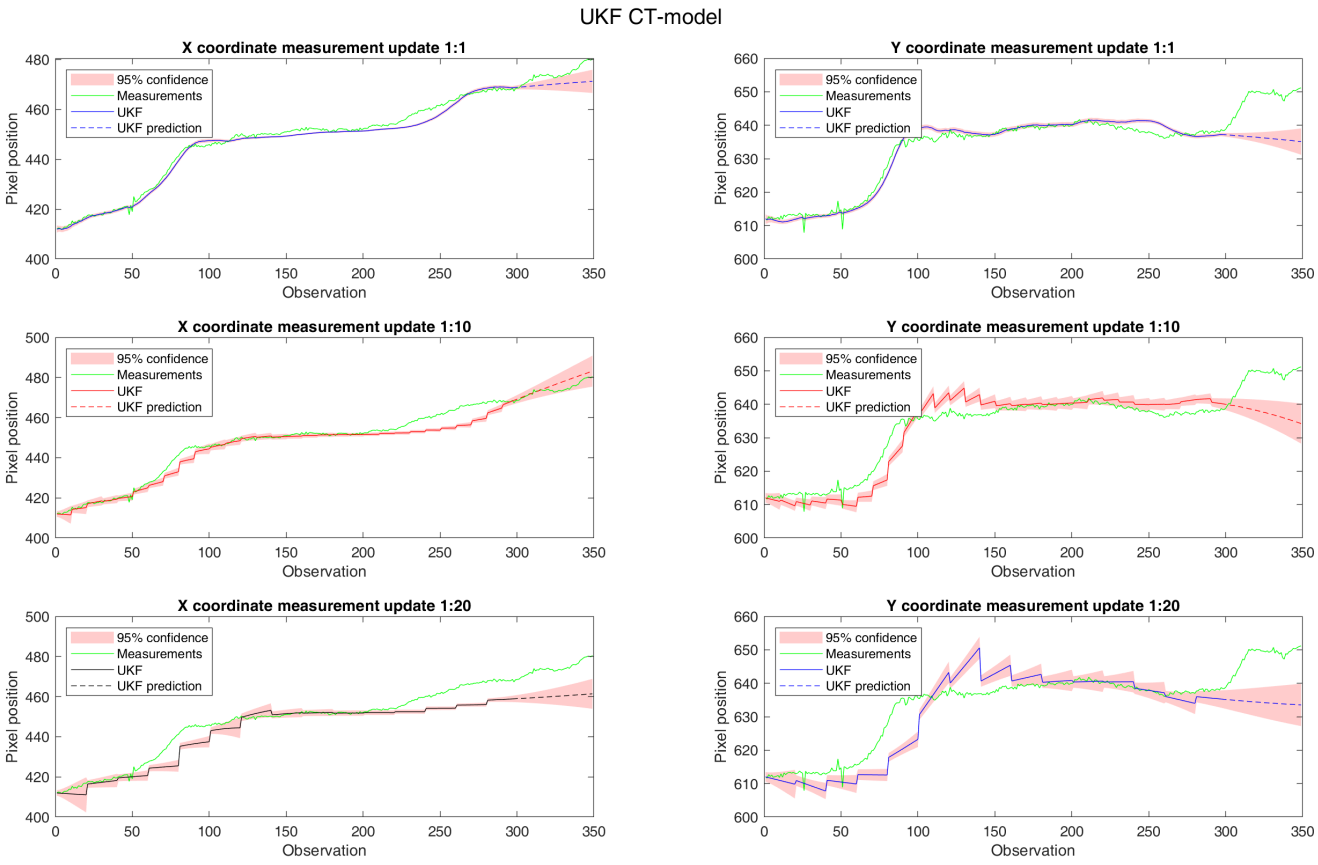


Figure 54: Confidence band X and Y position Unscented Kalman Filter in combination with the Coordinated Turn model for the avalanche track

The relative particle position error with the UKF-CT predicted position versus the observed particle position is shown in Figure 55. It can also be seen that for the UKF-CT it is difficult to trace rapidly changing dynamics of a particle, especially when fewer measurement updates are available.

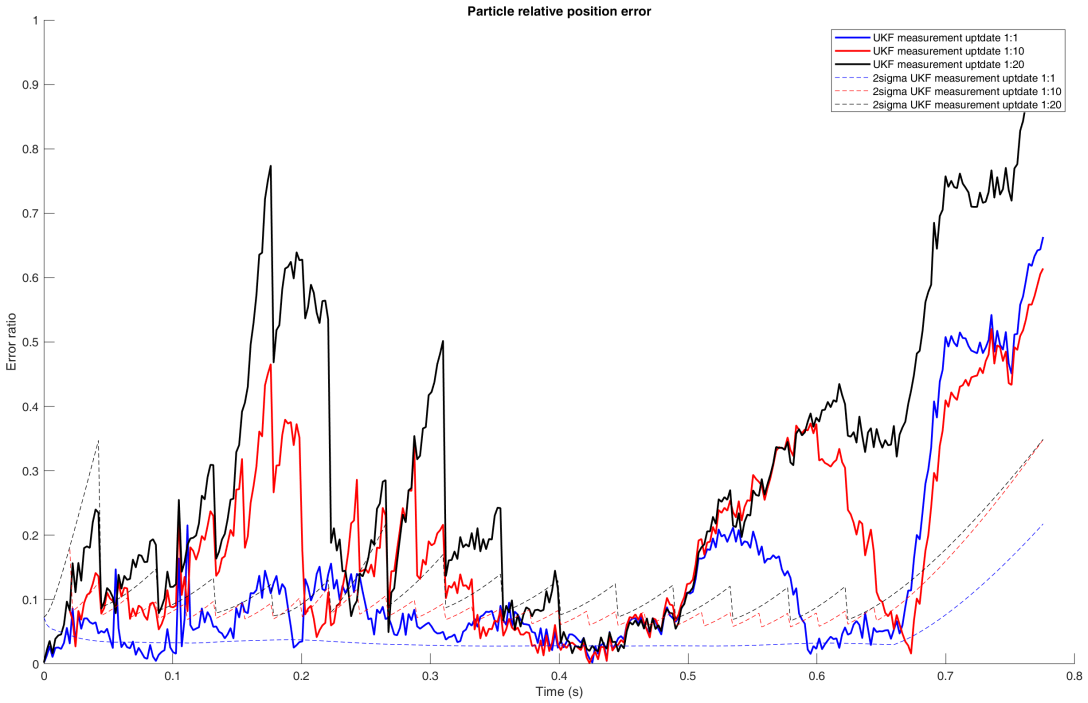


Figure 55: Relative error Unscented Kalman Filter in combination with the Coordinated Turn model for the avalanche track

Finally, the results of the UKF-CT for a rapid change in a track are shown in Figure 56. Here too it can be seen that a rapid change in the track is difficult to predict.

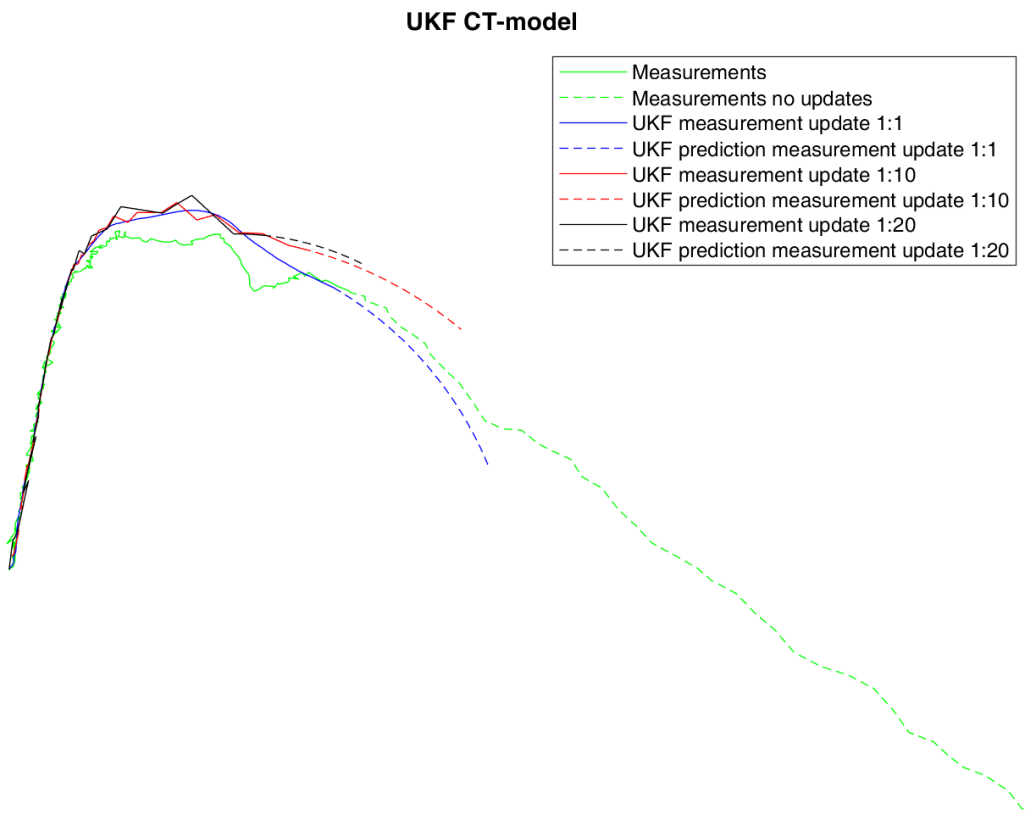


Figure 56: Performance Unscented Kalman Filter in combination with the Coordinated Turn model for the rapid change track

Figure 57 shows the confidence band of two sigma for the X coordinates and the Y coordinates of the particle track for the UKF-CT.

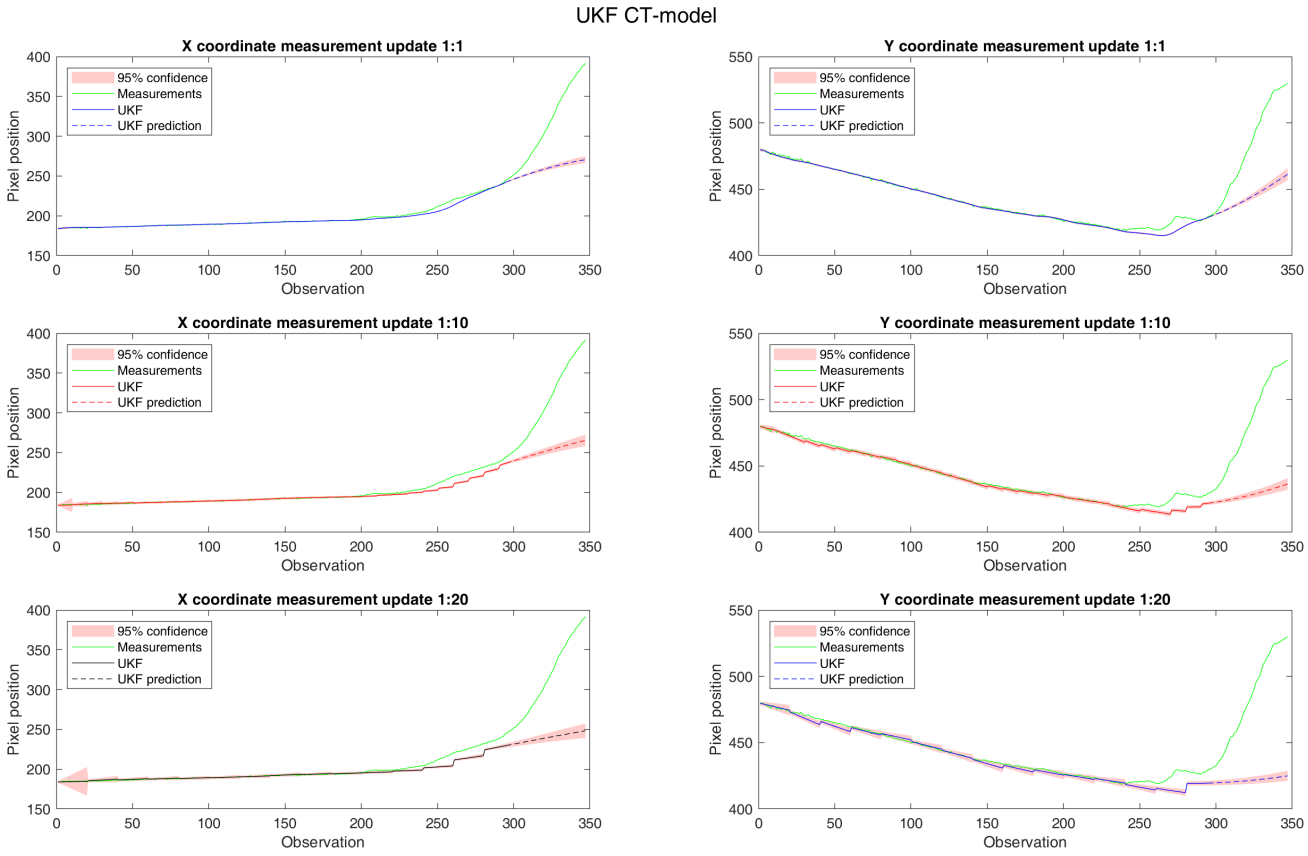


Figure 57: Confidence band X and Y position Unscented Kalman Filter in combination with the Coordinated Turn model for the rapid change track

The relative particle position error of the UKF-CT is shown in Figure 58. It is shown that in the first part the performance of particle tracking is good. However, for the last part where for 50 frames no measurement is available and therefore predictions have to be made which are not good. This has to do with the fact that the particle changes track direction here. This shows that when there is a rapid change in the particle dynamics it is important that a measurement update is available frequently.

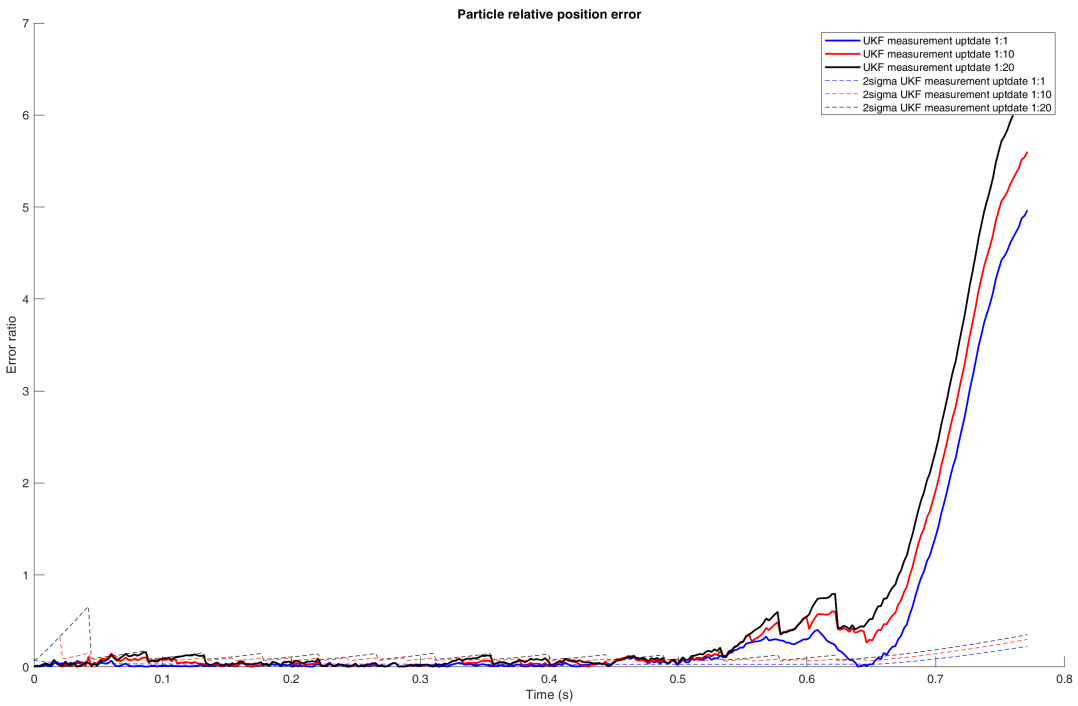


Figure 58: Relative error Unscented Kalman Filter in combination with the Coordinated Turn model for the circular track

The UKF-CT has a good performance for tracing and predicting the circular track. When rapid changes of the particle take place at a track, the UKF-CT can continue to track the particle reasonably well, but it is desirable that the measurement updates are frequent.

3.3.3 Comparison of the performance of the methods

In this section, the results of the KF-CV, UKF-CA, and UKF-CT are compared to determine which method should be utilized for multiple particle tracking. The performance of the methods is compared per track type, with the average error ratio expressing the performance. The singular track is produced using the NN as a reference. The performance is displayed in three tables, one for each measurement update interval size. The average error rate is an excellent indicator of performance as a whole. Nevertheless, mismatches in the singular tracks created by the NN have an effect on this. While performing multiple tracking, these mismatches should not occur, as the detected particle's position does not match the predicted position. This information is considered when selecting the method for tracking multiple particles.

With a measurement update for each observation and a frame rate of 450 (the highest available frame rate in the data set), the UKF-CA has the poorest performance. The KF-CV performs optimally on all tracks. When the time-steps are sufficiently small, with a frame rate of 450, the particle moves almost linearly between the observation steps. The UKF-CT performs marginally worse than the KF-CV, with the difference being particularly negligible in circular trajectory. Taking into account the section above, it is clear that the KF-CV performs better for the circular trajectory, but the UKF-CT performs significantly better for the predictive part. Therefore, the UKF-CT performed better than the KF-CV when no particle could be detected in the circular track for an extended period of time. In Table 1, the results of the average error rate can be found.

Table 1: Average particle relative position error, measurement update 1:1

	Circulair track	**Avalanche top**	**Rapid change**
KF-CV	0.0234	0.0854	0.4088
UKF-CT	0.0218	0.1281	0.4114
UKF-CA	0.0383	0.2825	0.4345

When the measurement update steps are increased to every tenth frame with a frame rate of 450, a shift in performance can be seen. Still the UKF-CA has the poorest performance for tracking the particle. Whereas the UKF-CT now outperforms the KF-CV, for all the track types. The results are listed in Table 2.

Table 2: Average particle relative position error, measurement update 1:10

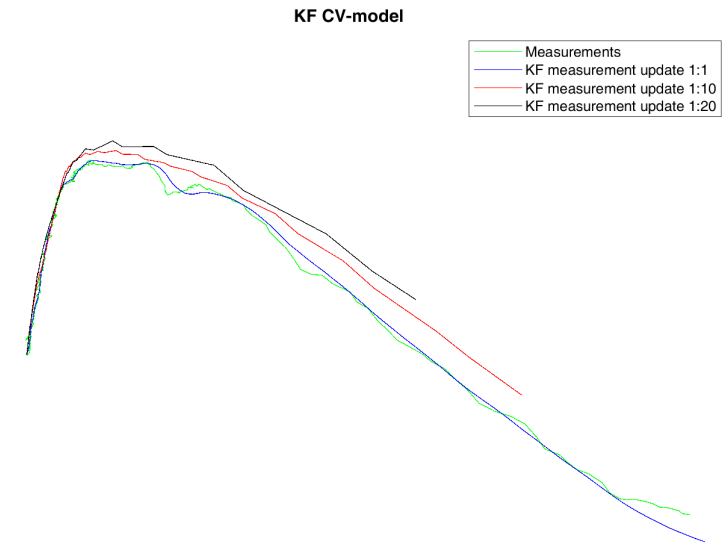
	Circulair track	**Avalanche top**	**Rapid change**
KF-CV	0.0451	0.4158	0.5824
UKF-CT	0.0267	0.1865	0.5311
UKF-CA	0.0669	0.4475	0.5705

Finally, the results of the average error ratio when the measurement is updated once every twenty observations with a frame rate of 450 are shown in Table3. Also now the performance of the UKF-CT is considerably the best.

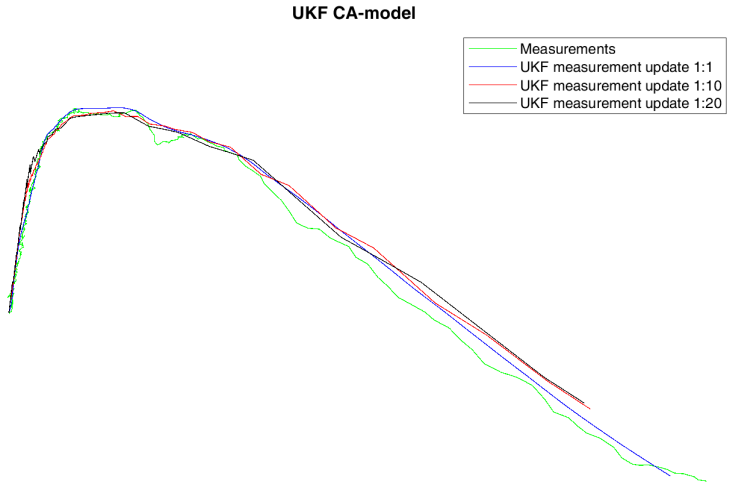
Table 3: Average particle relative position error, measurement update 1:20

	Circulair track	**Avalanche top**	**Rapid change**
KF-CV	0.0745	0.6627	0.7437
UKF-CT	0.0345	0.2979	0.6278
UKF-CA	0.0955	0.4939	0.6369

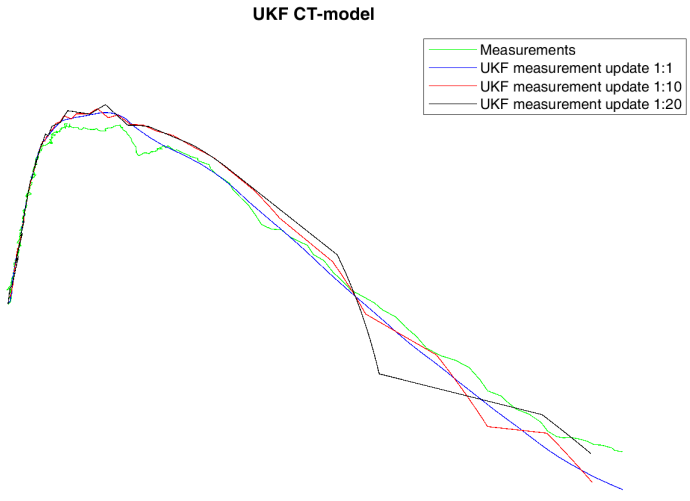
For a track with a rapid change, a prediction of fifty observations with a frame rate of 450 is too much because the track of the particle changes a lot in terms of acceleration. To better compare the methods, the results of the rapidly changing track are presented without the prediction part at the end. The results are shown in Figure 59.



(a) Rapid change track KF-CV



(b) Rapid change track UKF-CA



(c) Rapid change track UKF-CT

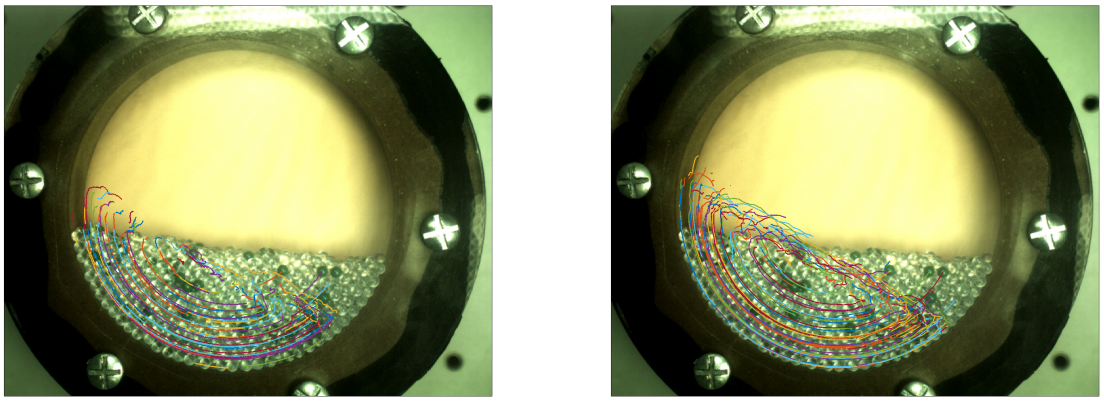
Figure 59: Performance rapid change track

Now that the rapid change track no longer has a separate prediction part, but is purely compared to only the step size of the measurement update, it shows that the KF-CV performs best when a measurement update is available for every observation. As the step size of the available measurement updates increases, it can be seen that the UKF-CT model is the only one still performing well. The goal is to be able to observe the tracks longer even when no detection is available in some frames. For this reason, the UKF-CT variant is used in combination with the NN for multiple particle tracking. Since the performance of the KF variant must be good for when no measurement updates are available, it is decided to use the UKF-CT for multiple particle tracking.

3.4 Multiple particle tracking

To improve the tracking of the particles compared to the NN, the tracks are now created with the NN in combination with the UKF-CT. This makes it possible to trace specific tracks when there is no detection of particles in some of the observation images. The advantage of this is that one is less dependent on the accuracy of the particle detection because the UKF-CT can predict the track of the particles. The multiple particle tracking algorithm is a combination of the NN and the UKF-CT. For every observation, the NN checks which particles of observation k versus $k-1$ have the smallest distance to each other. If a particle does not have a nearest neighbour, it is seen as a new track. Subsequently, for each track of $k-1$, a prediction is made with the UKF-CT as to what the position of the track will be in observation k. The location of the UKF-CT prediction and the nearest neighbour particle position found with the NN are compared. When these two positions per track fall within the allowed maximum distance, the position of the NN is considered the new location of the track. This location is then used to give the UKF-CT a measurement update. For this research, the maximum allowable distance between the location of the NN and the UKF-CT is set to a threshold of ten pixels, which corresponds to two-thirds of a particle's radius. If no particle within the threshold value can be linked to the UKF-CT predicted location, then the location of the UKF-CT is used as the new particle location of that track. However, this track will receive a penalty. The maximum allowable penalties per track are set to ten for this research. If a particle can be coupled within this value during an observation, the penalties are reset to zero. If the number of penalties exceeds the threshold value of ten, the track is removed. The tracks that have an observation length that meets the minimum desired length are saved. As the last position predicted with the UKF-CT is not saved, the last updated observed position ends the track.

For a good overview, first the tracks of the tracers are analyzed. The complete data set of 8741 frames is used for this with a frame rate of 450. In Figure60a the tracer tracks created by the NN are shown and in Figure 60b the tracer tracks are created with the NN in combination with the UKF-CT. As can be seen, the tracks of the NN in combination with the UKF-CT are improving. The tracks are improved for the bottom part of the particles where they follow a circular trajectory. However the largest improvement is in the part where the particles move from circular trajectory to the avalanche part, which is very desirable.

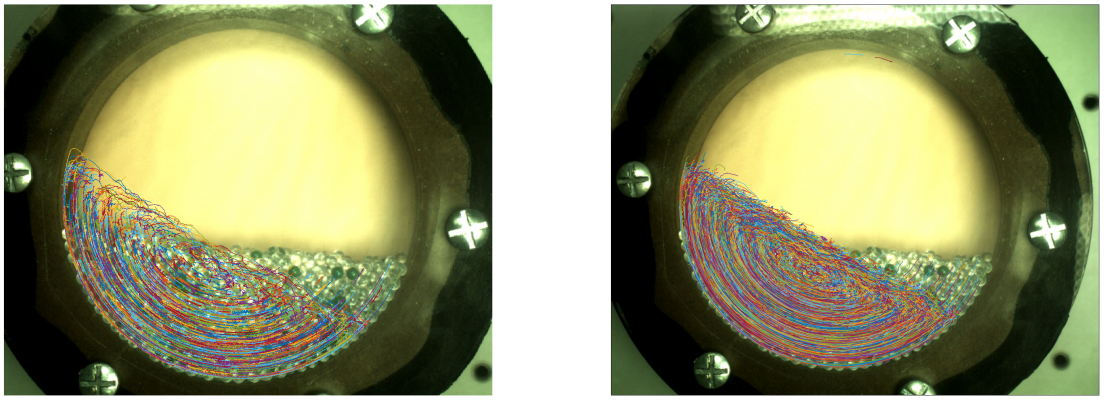


(a) Multiple tracer particle tracking NN

(b) Multiple tracer particle tracking NN UKF-CT

Figure 60: Multiple tracer particle tracking

In Figure 61a the track created with the NN for all particles are shown. Figure 61b shows the tracks for all particles created with the NN in combination with the UKF-CT. There is a clear improvement in the number of tracks and the length of the tracks.



(a) Multiple particle tracking NN

(b) Multiple particle tracking NN UKF-CT

Figure 61: Multiple particle tracking

For comparison, the tracks are ordered by length. The tracks of the NN are placed next to the tracks of the NN in combination with the UKF-CT for the comparison. The performance of the tracer tracks are shown in Figure 62. The number of tracer tracks has increased minimally for a track length of at least 100 observations. Creating tracer tracks with the NN yields 145 tracks, and the NN UKF-CT has 149 tracks. However, the tracer tracks lengths has improved considerably with the implementation of the UKF-CT. As an indication, the performance of the NN versus the NN UKF-CT is compared, the number of tracks compared is determined by the method with the fewest tracks. The average tracer track length of the NN is 480 observations and the average tracer track length of the NN UKF-CT is 738 observations. This is a performance increase with a factor of over 1.53. Mainly the longer tracer tracks have become even better, the tracer track length increases.

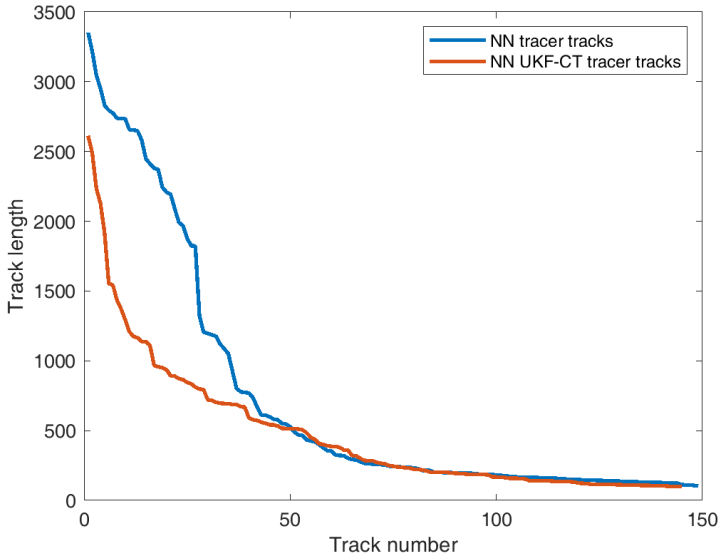


Figure 62: Performance comparison tracking tracer particle tracks NN versus NN-UKF-CT

The performance of all tracks can be seen in Figure 63. With the NN 2115 tracks were created with a minimum observation length of 100 observations while with the implementation of the UKF-CT 4429 tracks were created.

This is an improvement of more than 2.0. Furthermore, the length of the tracks are significantly increased. The NN has an average track length of 265 and the average track length of the NN UKF-CT is 885 observations. This is an 3.3 factor improvement.

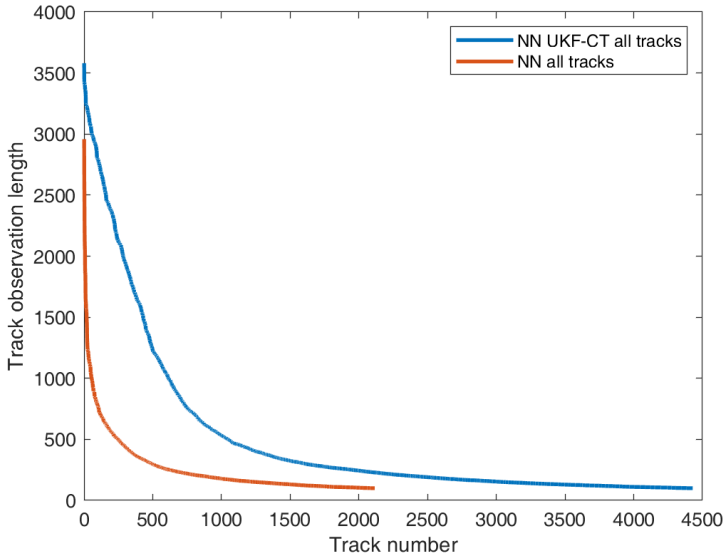


Figure 63: Performance comparison tracking all particle tracks NN versus NN-UKF-CT

The results presented show that the addition of the UKF-CT is advantageous to significantly improve the tracks length.

4 Conclusion

During particle detection a method is found to detect the particles in the video frames. This method is specially designed to detect the transparent dry particles and to distinguish the tracer particles. In the first step the image background is blacked out to prevent noise witch could interfere with the particle detection. In the next step the particles are detected with circular Hough transform, hereafter the particles are categorized. Pair correlation is used to place the particles in the front layer or the layer behind the front layer. To realize a higher accuracy of particle detection the described steps mentioned above are repeated, whereby previous categorized particles are blacked out. This method has some limitations that should be noted, such as the difficulty to determine false detected particles, particles belonging to the second layer and correctly detected particles. During the process of detection the remaining particles are more difficult to detect or categorize due to a loss of strong correlation between the particle and it's neighbour particles when the distance between them increases. When an individual particle is detected without other nearby particles it is very difficult to validate the detection because of the absence of direct correlations. Another limitation is validation, which is currently not possible as one only has unlabeled data set. The single method to validate the results is comparing the selected data by computer with the selection made by human eye. A mathematical validation score is thus lacking.

This research reveals it is possible to track the individual particles of an avalanche and predict the subsequent position. It is investigated whether the tracking of the particles can be improved in combination with a tracking algorithm. The research compared the performance of the KF-CV, the UKF-CA and the UKF-CT. The UKF-CT model gives the best results for tracking and predicting the track of a particle, because this model can predict circular movements and has the highest accuracy also when particles are not detected or frames are missing. This method makes it possible to trace the particles located in the circular trajectory. For this part it is also possible to reduce the frame rate and thus the required data because the UKF-CT is found to be very effective at predicting the circular trajectories. For the particles that enter the avalanche part, a high frame rate is necessary and it is not recommended to reduce the frame rate of 450. The reason for this is that the particle here behaves extremely stochastic and the speed is much higher than in the rest of the drum. The UKF-CT is able to track the particles in the avalanche part and make predictions for short periods. This is because the augmented Coordinated Turn model is applied in which the turn rate and velocity vector are not constant and therefore respond well when the particle changes direction and or acceleration.

The Nearest Neighbour in combination with the UKF-CT is shown to provide improved particle tracking performance. With the help of this method one may trace the particles for a long period of time (the particle track of the tracer became more than 1.5 as large and the particle tracks for the transparent particles become 3.3 as large) and trace more particles (there are more then double the number of tracks found). It is also possible to keep following the particles when they end up in the avalanche.

5 Recommendations

This research on detecting particles and predicting tracks of granular avalanches, lays the foundation for future research as the method used is unique and thus it was not possible to compare the results or method with other studies. To reduce the computational time of tracking, one may further train convolutional neural network as now data are labeled with the help of the proposed method in this thesis. So that in the future the particles can be detected with convolutional neural network.

The performance of the UKF for particle tracking can be improved by switching between different models. This makes it possible to use a different model and or different parameters when the particle moves to a different kind of maneuver such as from the circular track to the avalanche part. Further step is also to better tune the parameters of chaning models such as unknown process noise. This can be also achieved in KF setting by adding parameterized hierarchical prior on the noise, and learning its parameters.

6 Bibliography

References

[1] C. S. Campbell, "Rapid granular flows," *Annual Review of Fluid Mechanics*, vol. 22, pp. 57–92, 1990.

[2] K. Hutter and K. R. Rajagopal, "On flows of granular materials," *Continuum Mechanics and Thermodynamics*, vol. 6, pp. 81–139, 1994.

[3] H. J. Herrmann and S. Luding, "Modeling granular media on the computer," *Continuum Mechanics and Thermodynamics*, vol. 10, pp. 189–231, 1998.

[4] K. H. Shiva P. Pudasaini, *Avalanche Dynamics*. Springer, 2007.

[5] S. P. Pudasaini and K. Hutter, *Avalanche dynamics: dynamics of rapid flows of dense granular avalanches*. Springer Science & Business Media, 2007.

[6] I. Smalley, "Rockslides and avalanches," *Nature*, vol. 272, pp. 654–654, 1978.

[7] E. J. Hopfinger, "Snow avalanche motion and related phenomena," *Annual Review of Fluid Mechanics*, vol. 15, pp. 47–76, 1983.

[8] J. Copus, "Simpson, j. e. 1997. gravity currents: In the environment and the laboratory," *Geological Magazine*, p. 819–842, 1998.

[9] S. Havens, H.-P. Marshall, J. B. Johnson, and B. Nicholson, "Calculating the velocity of a fast-moving snow avalanche using an infrasound array," *Geophysical Research Letters*, vol. 41, no. 17, pp. 6191–6198, 2014.

[10] H. Brugger, P. Paal, and J. Boyd, "Prehospital resuscitation of the buried avalanche victim," *High Altitude Medicine & Biology*, vol. 12, no. 3, pp. 199–205, 2011.

[11] S. P. Pudasaini and K. Hutter, "Rapid shear flows of dry granular masses down curved and twisted channels," *Journal of Fluid Mechanics*, vol. 495, pp. 193 – 208, 2003.

[12] K. K. Rao, P. R. Nott, and S. Sundaresan, *An introduction to granular flow*, vol. 490. Cambridge university press Cambridge, 2008.

[13] A. Jarray, V. Magnanimo, M. Ramaioli, and S. Luding, "Scaling of wet granular flows in a rotating drum," in *EPJ web of conferences*, vol. 140, p. 03078, EDP Sciences, 2017.

[14] R. Han, J. Feng, Y. Zhang, H. Yang, V. Zivkovic, and R. Li, "Numerical simulation of avalanche propagation dynamics in a rotating drum," *Powder Technology*, vol. 380, p. 199 – 204, 2021.

[15] J. H. Kasper, V. Magnanimo, and A. Jarray, "Dynamics of discrete wet granular avalanches in a rotary drum," in *Proceedings of the 8th International Conference on Discrete Element Methods (DEM8)*, 2019.

[16] J. H. Kasper, V. Magnanimo, S. D. de Jong, A. Beek, and A. Jarray, "Effect of viscosity on the avalanche dynamics and flow transition of wet granular matter," *Particuology*, vol. 59, pp. 64–75, 2021.

[17] P. A. Cundall and O. D. L. Strack, "A discrete numerical model for granular assemblies," *Geotechnique*, vol. 29, pp. 47–65, 1979.

[18] R. Zhang, D. Su, Y. Jiang, H. Xiong, W. Han, and X. Chen, "Influence of topography on the impact mechanism of dry granular flow: A dem study," *Powder Technology*, vol. 416, p. 118216, 2023.

[19] K. Kihara and N. Okada, "Numerical simulation model of gas–liquid–solid flows with gas–liquid free surface and solid-particle flows," *Chemical Engineering Science*, vol. 270, p. 118507, 2023.

[20] A. Ahmadi, S. Larsson, and C. Wersäll, "Scaling granular material with polygonal particles in discrete element modeling," *Particuology*, vol. 75, pp. 151–164, 2023.

[21] S. Zhao, Z. Lai, and J. Zhao, "Leveraging ray tracing cores for particle-based simulations on gpus," *International Journal for Numerical Methods in Engineering*, 10 2022.

[22] H. Henein, J. Brimacombe, and A. Watkinson, "Experimental study of transverse bed motion in rotary kilns," *Metallurgical transactions B*, vol. 14, no. 2, pp. 191–205, 1983.

[23] Y. Ding, R. Forster, J. Seville, and D. Parker, "Segregation of granular flow in the transverse plane of a rolling mode rotating drum," *International Journal of Multiphase Flow*, vol. 28, no. 4, pp. 635–663, 2002.

[24] T. Loiseleux, P. Gondret, M. Rabaud, and D. Doppler, "Onset of erosion and avalanche for an inclined granular bed sheared by a continuous laminar flow," *Physics of fluids*, vol. 17, no. 10, p. 103304, 2005.

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[26] J. Lindberg, "Mathematical concepts of optical superresolution," *Journal of Optics*, 2012.

[27] T. Hao and D. Xu, "Circle area detection based on convolutional neural networks," p. 1123 – 1128, 2022.

[28] J. Du, "Understanding of object detection based on cnn family and yolo," *Journal of Physics: Conference Series*, vol. 1004, p. 012029, apr 2018.

[29] E. Cuevas, N. Ortega, D. Zaldivar, and M. Perez, "Circle detection by harmony search optimization," 2014.

[30] E. Cuevas, F. Wario, D. Zaldivar, and M. Cisneros, "Circle detection on images using learning automata," springerlink, 2013.

[31] E. Cuevas, F. Wario, V. Osuna, D. Zaldivar, and M. Perez, "Fast algorithm for multiple-circle detection on images using learning automata," 2014.

[32] T. M. H. N. R.-i. T. Roland Sireyjol, Atsushi Shimada, "How does cnn grasp transparent object features?," *CPE*, 2019.

[33] R. G. v. G. G. F. L. C. A. Tadros1, S. Drouyer, "Circular-shaped object detection in low resolution satellite images," *ISPRS Annals of the Photogrammetry*, 2020.

[34] R. Amirifar, K. Dong, and A. Yu, "Ordered packing of uniform spheres via random packing protocol," *Powder Technology*, vol. 409, p. 117853, 2022.

[35] F. H. S. R. C. Aleksandar Donev, Salvatore Torquato, "Jamming in hard sphere and disk packings," *JOURNAL OF APPLIED PHYSICS*, 2003.

[36] F. H. S. Steven Atkinson, Salvatore Torquato, "Existence of isostatic, maximally random jammed monodisperse hard-disk packings," *PNAS*, 2014.

[37] H. A. M. Yuliang Jin, James G. Puckett, "A statistical theory of correlations in random packings of hard particles," *Levich Institute and Physics Department*, 2014.

[38] J. Illingworth and J. Kittler, "A survey of the hough transform," *Computer vision, graphics, and image processing*, vol. 44, no. 1, pp. 87–116, 1988.

[39] S. M. M. Basu, "Gaussian-based edge-detection methods," *Science Publications*, pp. 252–1260, 2002.

[40] M. R. H. Y. P. Saad, "Object detection using circular hough transform," *Science Publications*, pp. 1606–1609, 2005.

[41] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, *et al.*, "Fiji: an open-source platform for biological-image analysis," *Nature methods*, vol. 9, no. 7, pp. 676–682, 2012.

[42] I. Grant, "Particle image velocimetry: a review," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 211, no. 1, pp. 55–76, 1997.

[43] K. Weise and W. Woger, "A bayesian theory of measurement uncertainty," *Measurement Science and Technology*, p. 11, jan 1993.

[44] R. Kalman, "A new approach to linear filtering and prediction problems," *Transaction of the ASME—Journal of Basic Engineering*, 1960.

[45] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.

[46] P. Del Moral, "Non linear filtering: Interacting particle solution," *Markov Processes and Related Fields*, vol. 2, pp. 555–580, 03 1996.

[47] A. Grootveld, V. I. Bugayev, L. Lackey, A. G. Klein, K. Vedula, and D. R. Brown, "Tracking of dynamical processes with model switching using temporal convolutional networks," pp. 1–9, 2021 IEEE Aerospace Conference (50100), 2021.

[48] J. Liu, Z. Wang, M. Xu, and J. Ren, "A deep neural network based maneuvering-target tracking algorithm," pp. 3117–3121, ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019.

[49] C. Gao, H. Liu, S. Zhou, H. Su, B. Chen, J. Yan, and K. Yin, "Maneuvering target tracking with recurrent neural networks for radar application," in *2018 International Conference on Radar (RADAR)*, pp. 1–5, 2018.

[50] G. Zhao, Z. Wang, Y. Huang, H. Zhang, and X. Ma, "Transformer-based maneuvering target tracking," *Sensors*, p. 8482, 2022.

[51] Y. Xia, S. Qu, S. Goudos, Y. Bai, and S. Wan, "Multi-object tracking by mutual supervision of cnn and particle filter," *Personal and Ubiquitous Computing*, vol. 25, 2021.

[52] Q. Xu, Z. He, Z. Chen, and Y. Jiang, "An optical flow based multi-object tracking approach using sequential convex programming," pp. 1216–1221, 12 2020.

[53] B. Du, S. Cai, and C. Wu, "Object tracking in satellite videos based on a multiframe optical flow tracker," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pp. 3043–3055, 2019.

[54] W. F. de Haas, "The application of machine learning models for the prediction of avalanche dynamics," *Universiteit Twente Applied Meachanics and Data Analysis*, pp. 1–38, 2021.

[55] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pp. 153–158, Ieee, 2000.

[56] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068, pp. 182–193, Spie, 1997.

[57] A. H. Jazwinski, "Stochastic processes and filtering theory," 1970.

[58] F. M. C. B. C. D. T. B. C. A. Virginie Hergault, Philippe Frey, "Image processing for the study of bedload transport of two-size spherical particles in a supercritical flow," *Experiments in Fluids*, 2010.

[59] T. Kumar and K. Verma, "A theory based on conversion of rgb image to gray image," *International Journal of Computer Applications*, vol. 7, no. 2, pp. 7–10, 2010.

[60] J. Canny, "A computational approach to edge detection," *IEEE*, pp. 679–698, 1986.

[61] I. Sobel and G. Feldman, "A 3×3 isotropic gradient operator for image processing," *Pattern Classification and Scene Analysis*, pp. 271–272, 01 1973.

[62] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3, pp. 850–855, 2006.

[63] P. Konstantinova, A. Udvarev, and T. Semerdjiev, "A study of a target tracking algorithm using global nearest neighbor approach," 01 2003.

[64] Q. Guo, C. Zeng, Z. Jiang, X. Hu, and X. Deng, "Application of unscented kalman filter in tracking of video moving target," in *Biometric Recognition: 14th Chinese Conference, CCBR 2019, Zhuzhou, China, October 12–13, 2019, Proceedings*, (Berlin, Heidelberg), p. 483–492, Springer-Verlag, 2019.

[65] J. C. Fei Cai, "Moving target tracking based on kalman algorithm," *Journal of Engineering Science and Technology Review*, 2014.

[66] V. P. J. X. Rong LI, "Survey of maneuvering target tracking. part 1: Dynamic models," *IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS*, 2003.

[67] X. R. Li and V. P. Jilkov, "A survey of maneuvering target tracking—part iv: Decision-based methods," *In Proceedings of the 2002 SPIE Conference on Signal and Data Processing of Small Targets*, 2002.

[68] F. G. Micheal Roth, Gustaf Hendeby, "Ekf/ukf maneuvering target tracking using coordinated turn models with polar/cartesian velocity," *Dept. of Sensor EW Systems, Swedish Defence Research Agency (FOI)*, 2011.

[69] F. Gustafsson, *Statistical Sensor Fusion*. Studentlitteratur, 2010.

[70] W. Zhang, X. Zhao, Z. Liu, K. Liu, and B. Chen, "Converted state equation kalman filter for nonlinear maneuvering target tracking," *Signal Processing*, vol. 202, p. 108741, 2023.

[71] S. Khobahi, "Object tracking using kalman filter," *github. com*, 2017.

[72] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proceedings of 1995 American Control Conference - ACC'95*, vol. 3, pp. 1628–1632 vol.3, 1995.

[73] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proceedings of 1995 American Control Conference-ACC'95*, vol. 3, pp. 1628–1632, IEEE, 1995.

[74] M. Brossard, A. Barrau, and S. Bonnabel, "A code for unscented kalman filtering on manifolds (ukf-m)," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5701–5708, 2020.