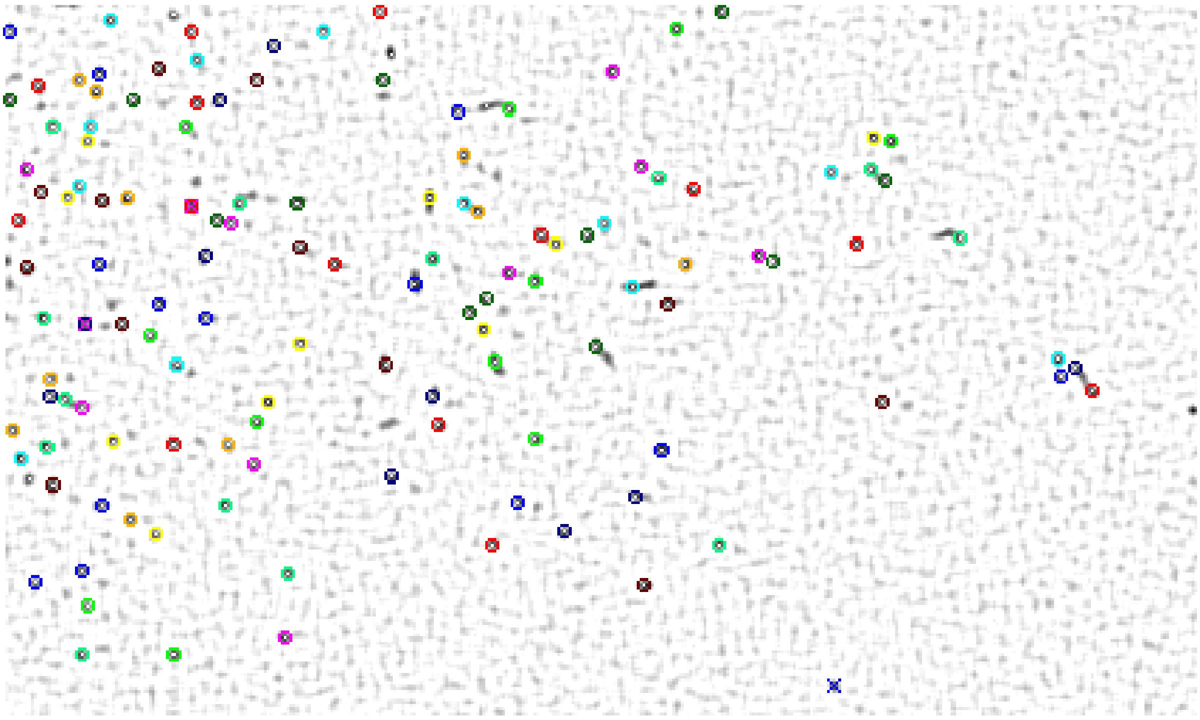
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

TP IV : Tracking and Analysis of Skyrmion Motion with matlab.

Mathilde Duchemin
mathilde.duchemin@epfl.ch
Professor : Henrik Ronnow
Supervisor : Thomas Schönenberger

June 8, 2018



Abstract

This TP IV follows the TP IV "Tracking of Skyrmion Motion with matlab" and is using new movie as material. This report presents a description of a code, including an identification and a tracking code, based on new motion hypotheses resulting from the previous study of Skyrmion motion, the main results and observations, finally a statistical analysis of the Skyrmion motion. This study is using a single movie showing current driven skyrmion motion.

Contents

1 Introduction

It is by looking and seeing that we come to know the world we live in. Exactly how the visual system works remains a mystery to be solved. Replacing the living creature with a computational instrument, we come to a computer vision which actually means the process of computers analyzing digital images or videos and obtaining an understanding from it. Because manually detecting and following large numbers of individual particles can't be digested by a human observer, automated computational method is necessary for these tasks. Computerized image analysis offers the potential to take advantage of available data in an efficient and reproducible manner. The importance of object tracking is reflected by the broad area of applications such as bioimaging, astrophysics, human behavior analysis, etc.

Technological developments in the past two decades have especially advanced the field of bioimaging and have enabled the investigation of dynamic processes in living cells at great spatial and temporal resolution. Achieving complete understanding of any living thing inevitably requires thorough analysis of both its anatomic and its dynamic properties. Examples include the study of cell membrane dynamics, cytoskeletal filaments, focal adhesion, viral infection, intracellular transport, gene transcription and genome maintenance [1]. Here, a 'particle' may be anything from a single molecule to a macromolecular complex, organelle, virus or microsphere, and the task of detecting and following individual particles in a time series of images is often referred to as 'single-particle tracking'.

Otherwise object tracking is an important part of a human-computer collaboration in a continuous environment, in the sense of allowing the computer to obtain a better model of the real world. For instance, in the application area of autonomous vehicles where it is not possible for a human to communicate the state of the environment accurately and quickly enough given the requirements of the agent [2].

In this report, the objects of interest are quasi-particles called *skyrmions*. A magnetic skyrmion is a topologically stable particle-like object that appears as a vortex-like spin texture [3]. Skyrmions are usually topologically protected so they are not supposed to appear, disappear, merge or split but we will see that those phenomenon will sometimes be recorded. A classical tracking process must be improved in order to consider those phenomenons. Those events frequency can then be studied.

Object tracking in video can be summarized as the task of finding the position of an object in every frame and is still considered as complex problem to solve. The difficulty comes from the fact that the behavior of the object is at first unknown and can be unpredictable. Moreover, in case of high density of objects or high velocity, the travelled distance between two time lapses is not so different from the distance between two particles. In case there is a clear flow in a specific direction, a favor displacement toward this direction might be necessary.

2 Motion hypotheses and description of the code

2.1 Motion hypotheses

The tracking process is to link the particles seen on one frame with the ones on the previous frame. By looking at the movie we can see fusion and fission phenomenons. The term "fusion" means the merge of two particles into one and the term "fission" means the split of one particle into two particles. Since skyrmions are supposed to be topologically protected, we consider that those particles are more likely to travel a long distance than to fuse with a closer particle and more likely to fuse with a close enough particle than to disappear. In the same way, a particle is more likely to have traveled a long distance than to come from the fission of a closer particle and more likely to come from the fission of a close enough particle than to appear. The movie also shows a net flux in a specific direction. Then we consider that the particles are more likely to follow this flow than to move against it or to go in a perpendicular direction. Sometimes the next position of a particle already appear on a current frame. Then instead of considering this phenomenon as one particle splitting into two particles and then recombining, it is considered as one only particle, the next position on the current frame isn't taken into account. Sometimes a particle seems to appear during one or two successive frames. This is taken as an error from the identification process and isn't taken as a real particle.

2.2 Identification

Our medium is a movie, therefore a preliminary step is to split it into a sequence of frames. The first image of this sequence will be used as an example to illustrate the different steps of the identification process. The raw image is shown in figure 1

The identification process is performed by an open source code developed by Daniel Blair and Eric Dufresne and consists in two main steps (functions). This code consider 255 gray scale images where particles should be bright spots on dark background. Asone an see in figure 1, the image has to be inverted and is shown in figure 2.

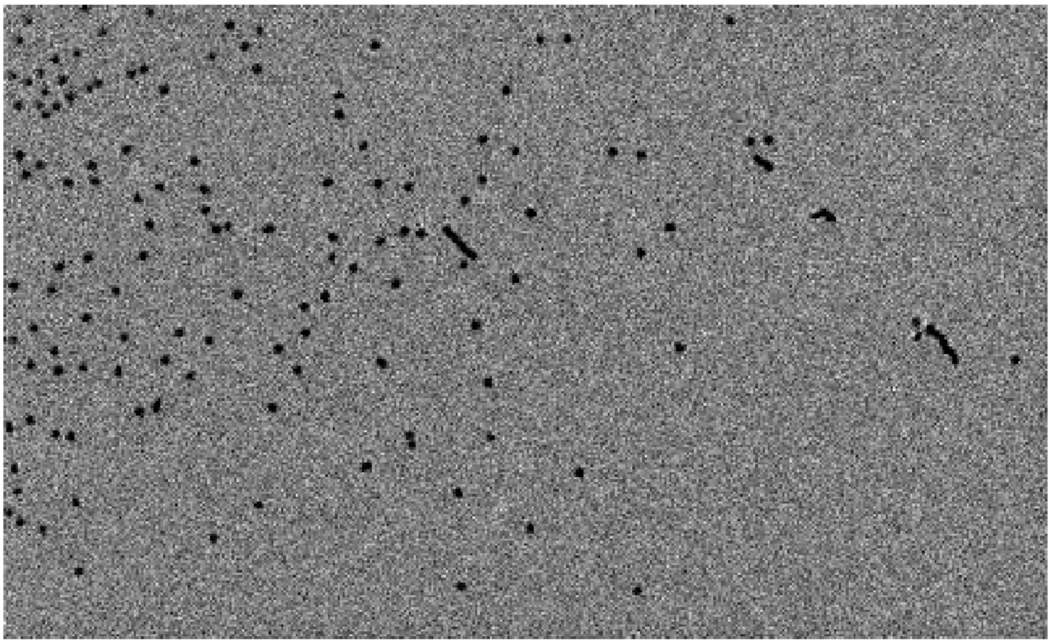


Figure 1: Raw image

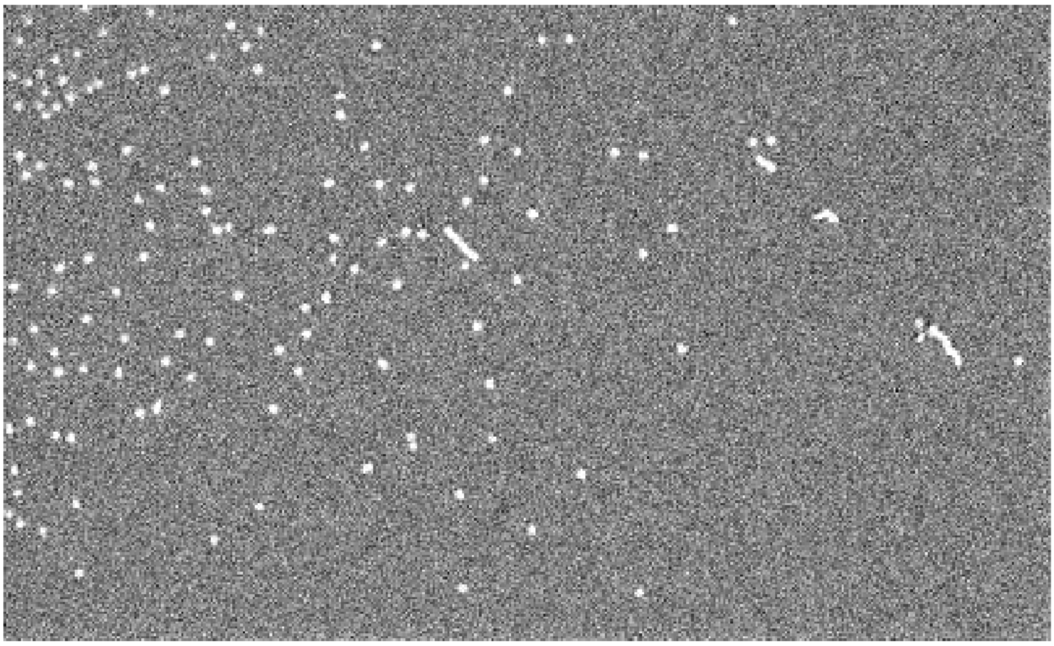


Figure 2: Inverted image

The first function *bpass* basically filter the image. It implements a real-space bandpass filter that suppresses pixel noise and long-wavelength image variations while retaining information of a characteristic size. The approximate size of the particles is given as input in the function. This size is directly measured on a random frame of the movie. The filtered is shown in figure 3.

The second function *pkfind* returns uses the filtered image and returns the localization of each particle. In detail, it finds local maximum in an image to pixel level accuracy as long as the brightness of the local maximum is above a threshold given as input to the function. Since the luminosity of each frame variate, the threshold is specific each frame : it depends on the average luminosity of the image. The approximate size of the particles is given as well : two local minimum that are closer that the expected size of a particle are considered as part of the same particle. The filtered image and the raw image with particle's location are shown respectively in figure 4 and 5.

This identification process is carried out over all the frames of the sequence.

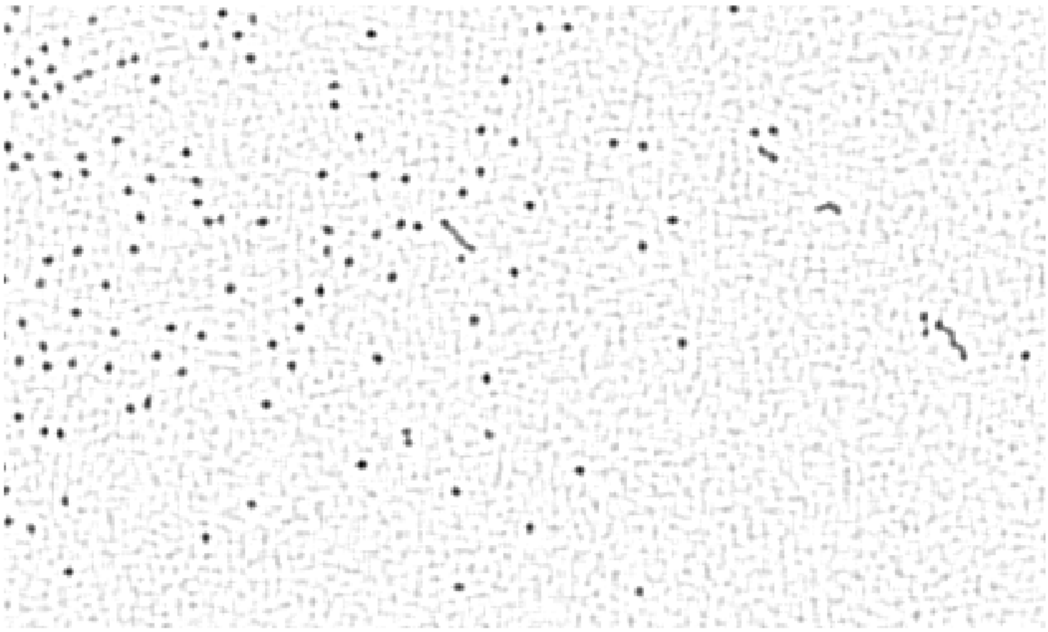


Figure 3: Filtered image

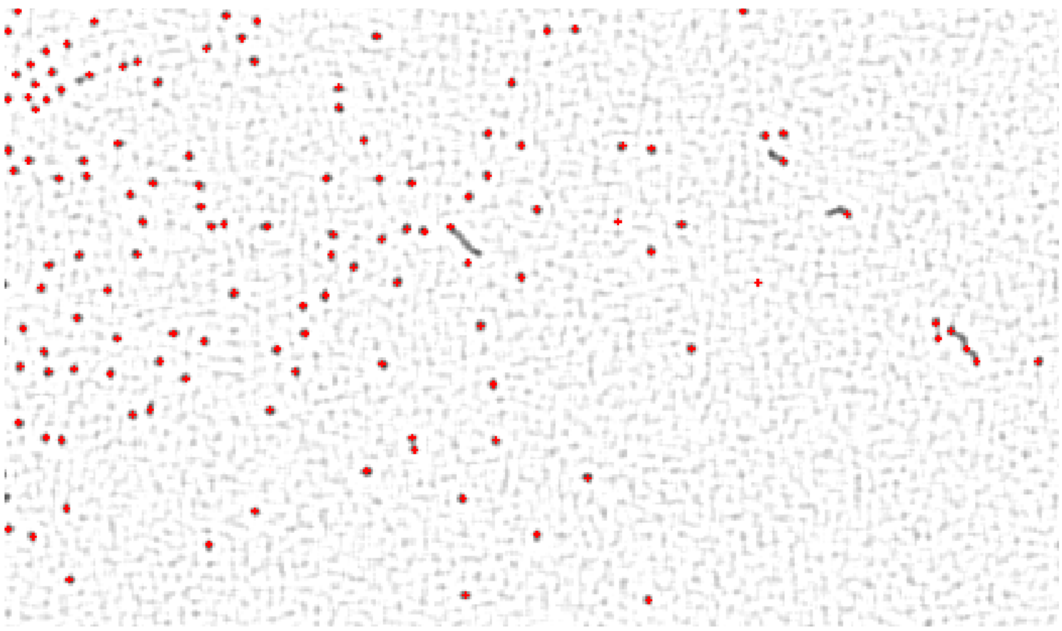


Figure 4: Filtered image with particle's location

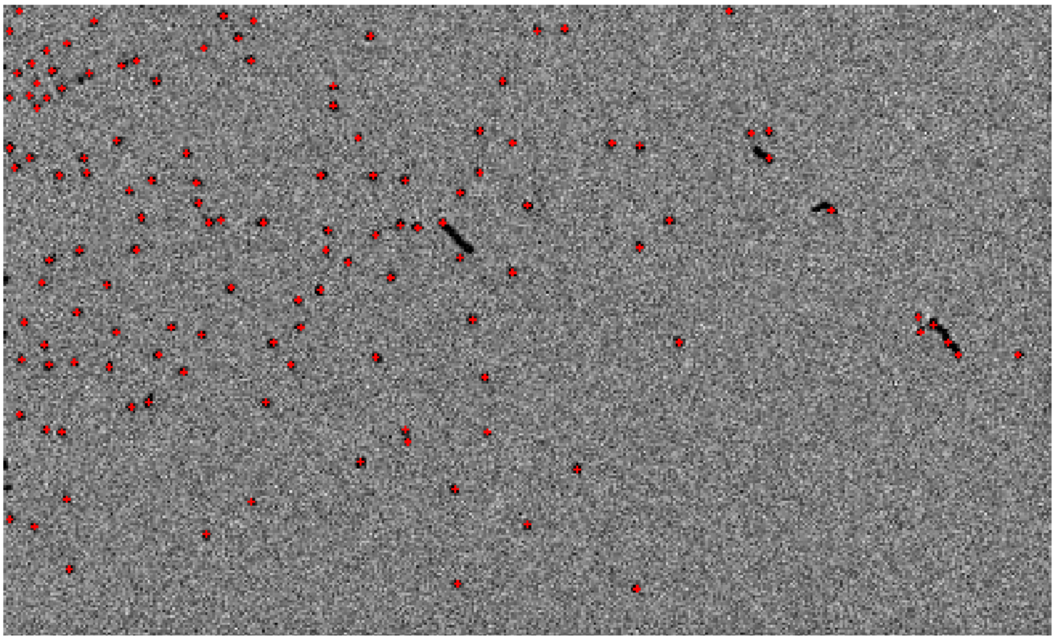


Figure 5: Raw image with particle's location

2.3 Tracking

Tracking particles means linking the position of a particle on a first frame to its future position on the following image. The tracking algorithm used for the TP4 "Tracking of Skyrmion Motion with matlab" has been modified following the motion hypotheses expounded in section 2.1. This linking process carried on all over all the frames of the sequence contains now 4 steps. Those four steps will be exposed bellow and will be illustrated by events from the movie studied. We will consider two frames: the i^{st} and the $(i+1)^{st}$. Once two particles from two different frames are linked, they form a vector containing the two locations. Those vetors are called "trajectories".

The first step is comes from the classical algorithm used in the previous TP4. Indeed, it uses the closest distance method where particles from the $i+1^{st}$ frame choose the closest particle from the i^{st}. In details, the "choosing" are all the particles from the $(i+1)^{st}$ frame. They choose the closest position among all the existing trajectories that end at frame i^{st} or frame $(i-1)^{st}$. In order to know the distance, a special distance function is used. This function favors displacement towards the positive X direction which is the direction of the flow. Therefore they are more likely to follow the flow. The $(i-1)^{st}$ frame is also considered in case the identification process missed the i^{st} position of a particle. Once all the particle from the $(i+1)^{st}$ frame have chosen its closest position, the matching process can start. If the distance to the chosen position is shorter than its distance to the closest particle from the same fame, then the position of the particle is added to the end of chosen trajectory. If not, it is saved for step 2. If two particles choose the same trajectory, the closest particle is the one added. The other one is saved for step 2 as well.

The second step is the same as the first step except that the "choosing" particles are the one saved from the first step. Therefore they all belong to the $(i+1)^{st}$ frame. They choose the closest position among all the existing trajectories that end at frame i or $i-1$ that is all the ones that hasn't be chosen during step 1. Here there is a match if distance is shorter than an arbitrary number a bit longer the one used for step 1. The particle from the $(i+1)^{st}$ frame that still don't match are saved for the 3^{rd} step.

This first two steps process is useful for the kinds of events illustrated in figure 6. Indeed, the light blue particle passes close by the dark blue one. One can see that both of the particles on the 3^{rd} frame are closer to the dark blue one from the 2^{nd} frame. This algorithm matches the dark one during step 1, saves the light one for step 2 and matches the latter with a bit further position from the 2^{nd} frame during step 2.

Figure 6: A particle passing by a non moving particle

The third step allows the "fission" of a particle into two particles. The considered particles are the one that couldn't match during step 2. Therefore they all belong to the $(i+1)^{st}$ frame. They might come from the fission of a previous particle. Here, they choose among all the existing trajectories that have already matched with a particle from the $(i+1)^{st}$ frame. They consider the position of those trajectories at the previous frame and choose the closest one. There is a match if distance is shorter than an arbitrary number. If so, the particle is considered as "being created from fission" and form a new trajectory that start from the position of the chosen trajectory at the i^{st} frame. The particles from the $(i+1)^{st}$ frame that still don't match are considered as "appearing" particles and form a new trajectory. It can happen that sometimes, two particles get so close that the identification process consider them as one only particle. Those particles move back away from each other again the next frame. If this case, we consider that particles didn't merge and split right away as the classical tracking process would do. In order to correct this phenomenon, the algorithm consider that if two particles that just merged want to split, then those particles never merged and continue there own way.

The figure 7 illustrates this phenomenon.

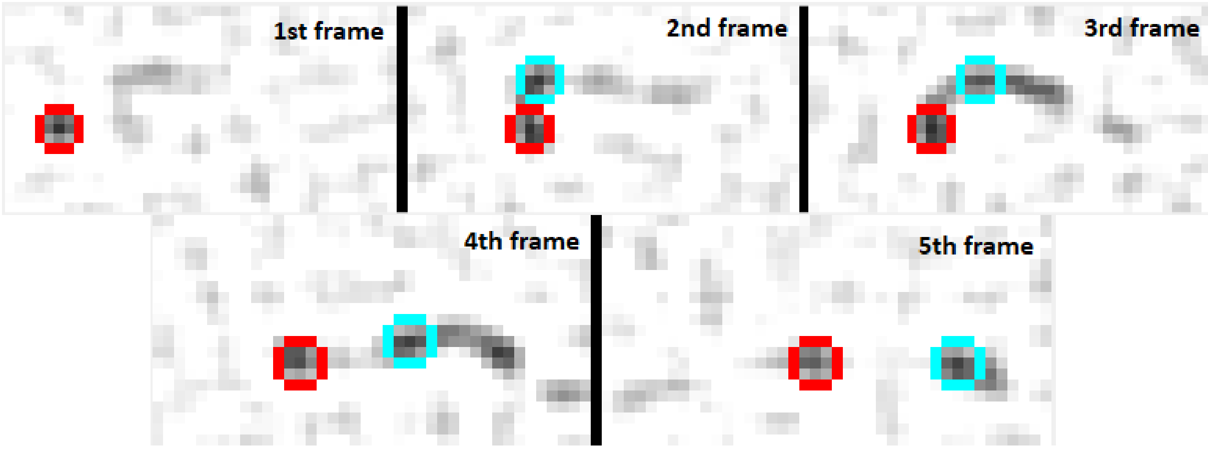


Figure 7: A particle splitting into two particles

The last step allows fusion of particles. This time, the particles considered are the trajectories from the i^{st} frame that didn't match with a particle from the $i+1^{st}$ frame. They can choose among all the trajectories that end at the $i+1^{st}$ frame. They choose the closer one and match under the condition that the distance is shorter than an arbitrary number. The figure 8 illustrates this phenomenon. It can happen that a particle split for only one frame and fusion back during the next frame. This phenomenon can happen during 2 frames as well. In this case, the particle is considered as one single particle all along. This 4^{th} step makes sure that if a particle fusion back to a particle that it left one or two frames before, this particle isn't a real one.

Figure 8: Two particles merging into one particle

3 Main observations

In this section will be presented some information about the movie and the main observations obtained from the algorithm. This movie shows current-driven Skyrmion motion [4]. The movie is captured by performing polar-MOKE imaging in the presence of a series of pulsed current, with a period of 1s. The pulse duration is 1ms, which is much smaller than the period. The pulse current density is $J_x = 2.1 \times 10^6$ A·cm^{-2} and the out-of-plane magnetic field is kept constant at $H_z = +5.0$ Oe. The movie lasts 8 seconds and contains 59 frames. The dimension of the observed area is $100\mu m \times 130\mu m$.

In order to know the main motion of those particles, some overview pictures will be presented.

The figure 9 shows the last frame with the all trajectories of all the particles that have appeared along the movie. One coloured circle represents one particle. Those are the particles still present at the last frame. The particles clearly move toward the X direction but there motion seems to concentrate into main axes. This is confirmed by the superposition of the positions over all the frames in figure 10. Figure 11 shows the map density with a grid unit of 2.3 μm^2.

The figure 12 shows the mean velocity within unit squares that is to say the sum of all velocities inside a unit square divided times the density of the square. Figure 12 shows the mean over the frames. One can see that the areas where there is higher density are the ones where the mean velocity is small.

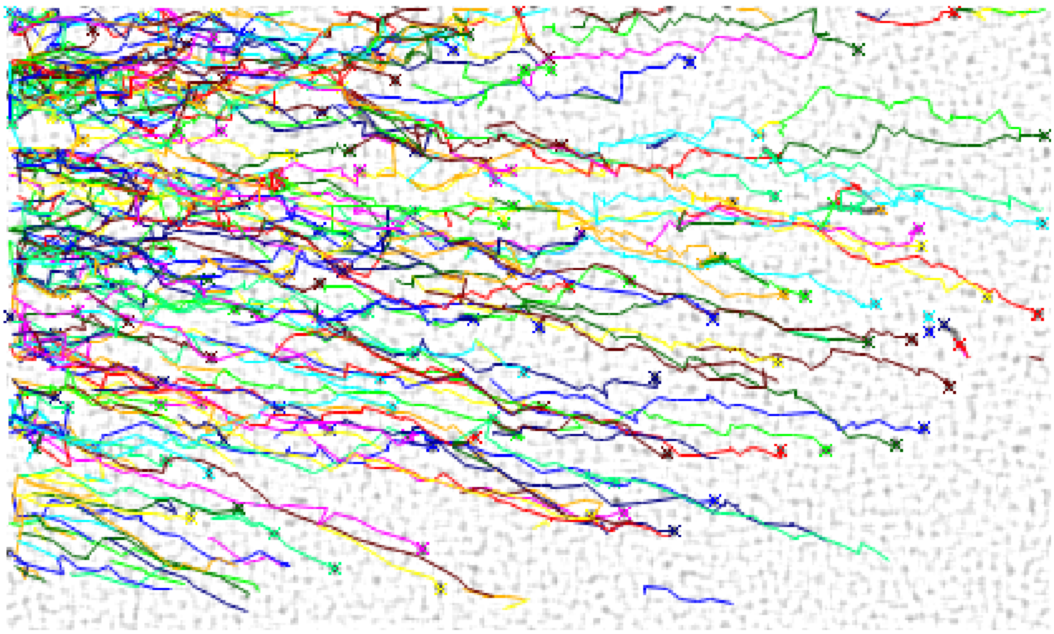


Figure 9: Trajectories of all the particles that have appeared along the movie

From the map density one can see that some small areas are highly dense. Those small areas could pin the particles down, the latter would be stock at the same place for a long time. The figure 13 represents the number of times that a particle has stayed inside the same unit square counting from the moment it stays inside the same unit square for 5 frames.

If we look at the total number of particles, the figure 14 shows that the total number of particles increases. Indeed, they are constantly coming from the left edge of the area and only a few reach the right edge to disappear.

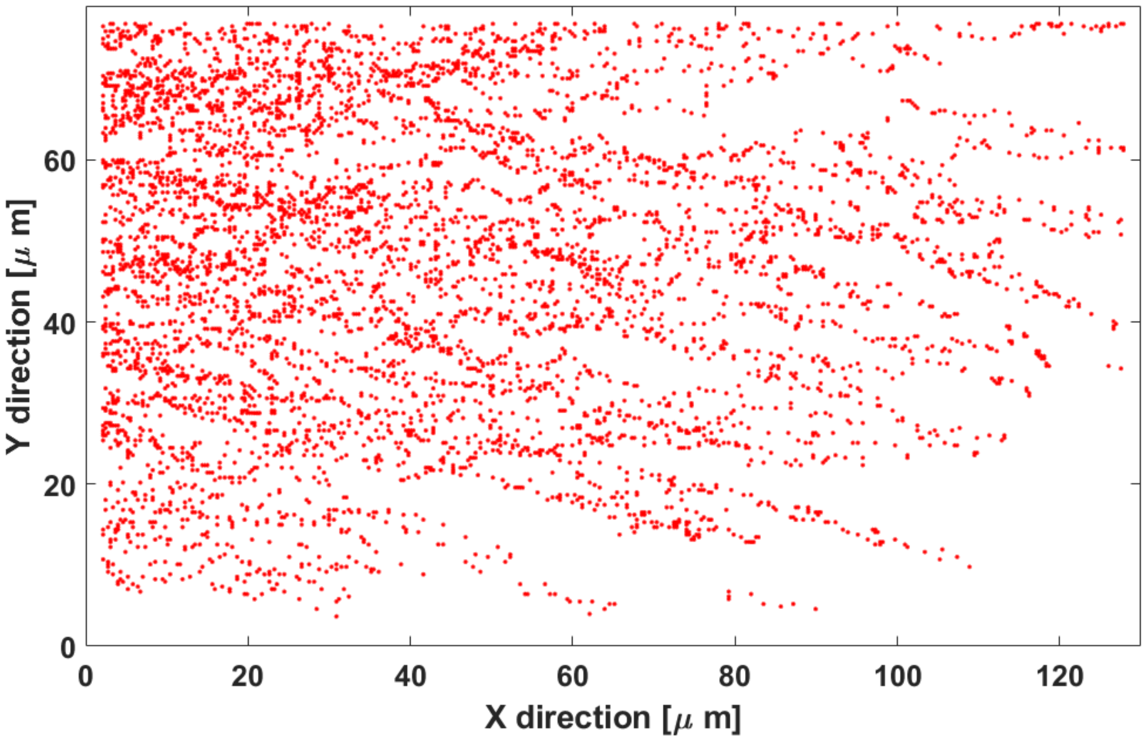


Figure 10: Superposition of the positions over the frames

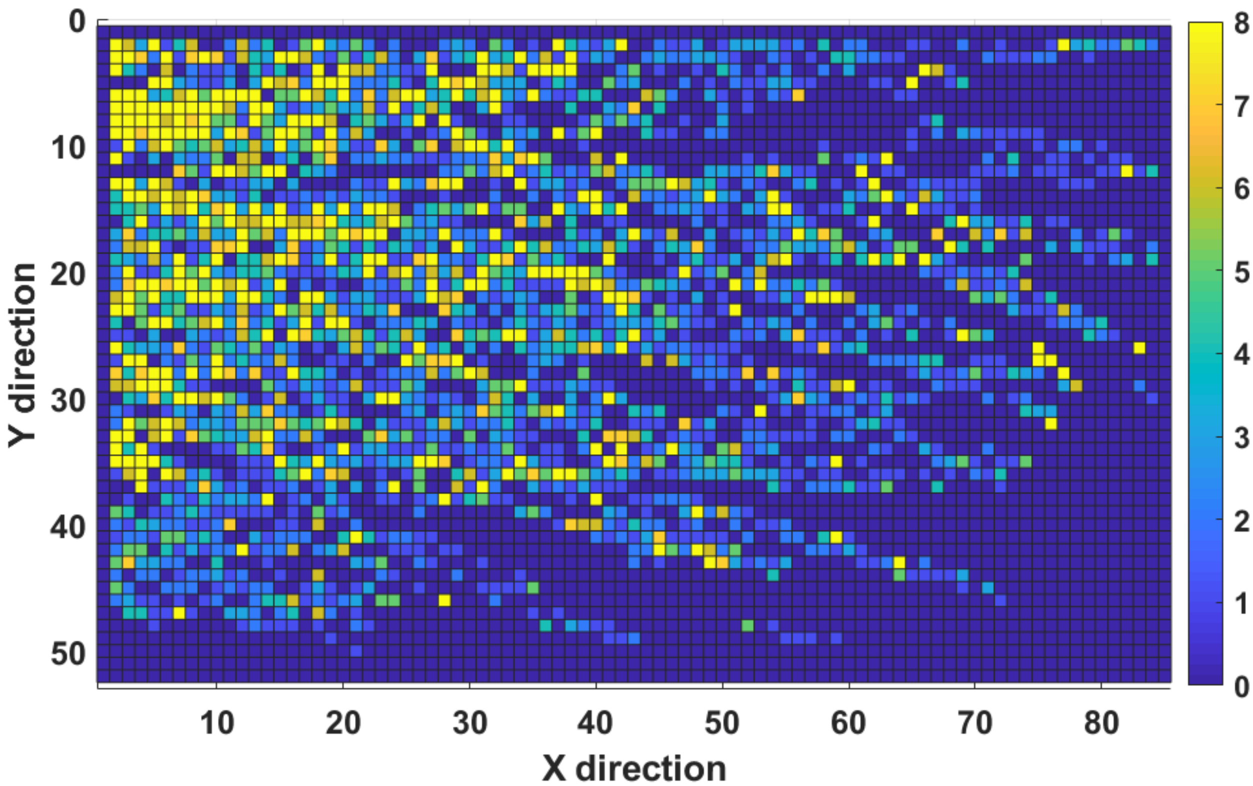


Figure 11: Density map

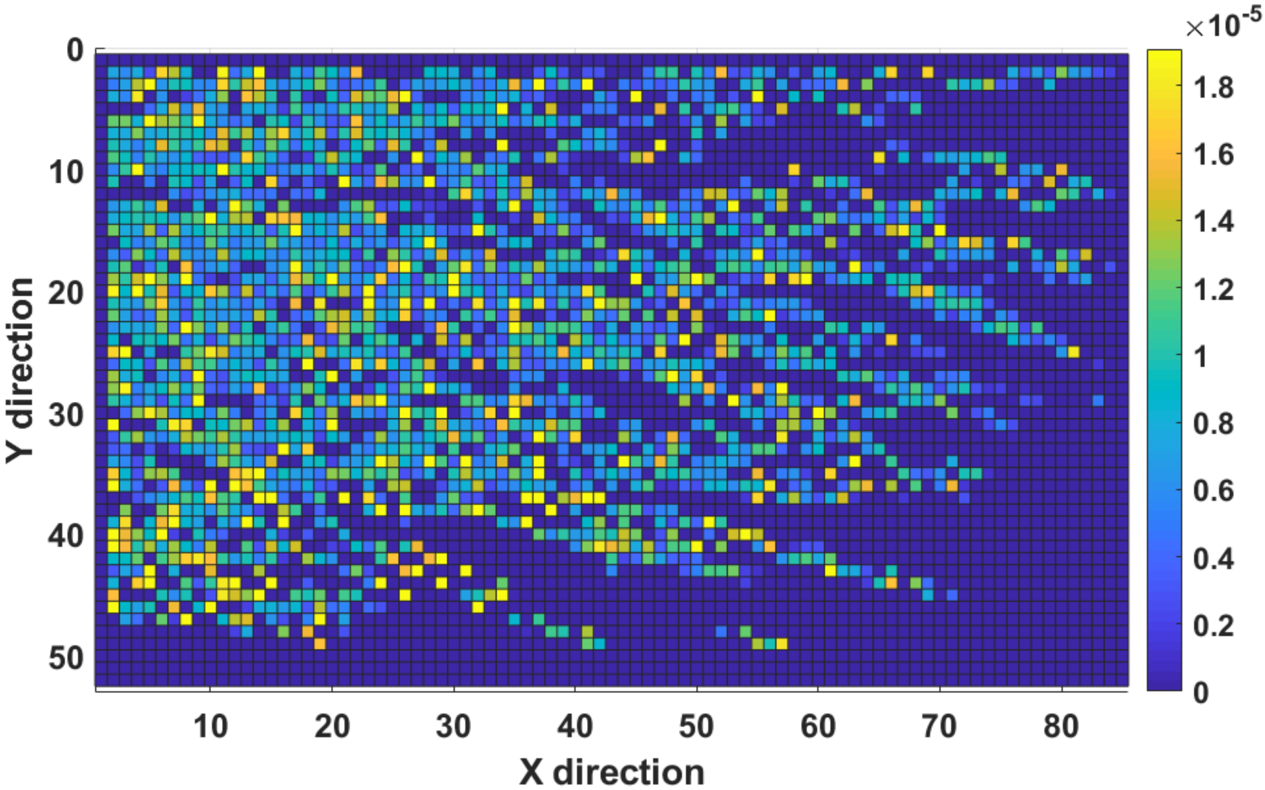


Figure 12: Mean over frames of the distribution of velocity

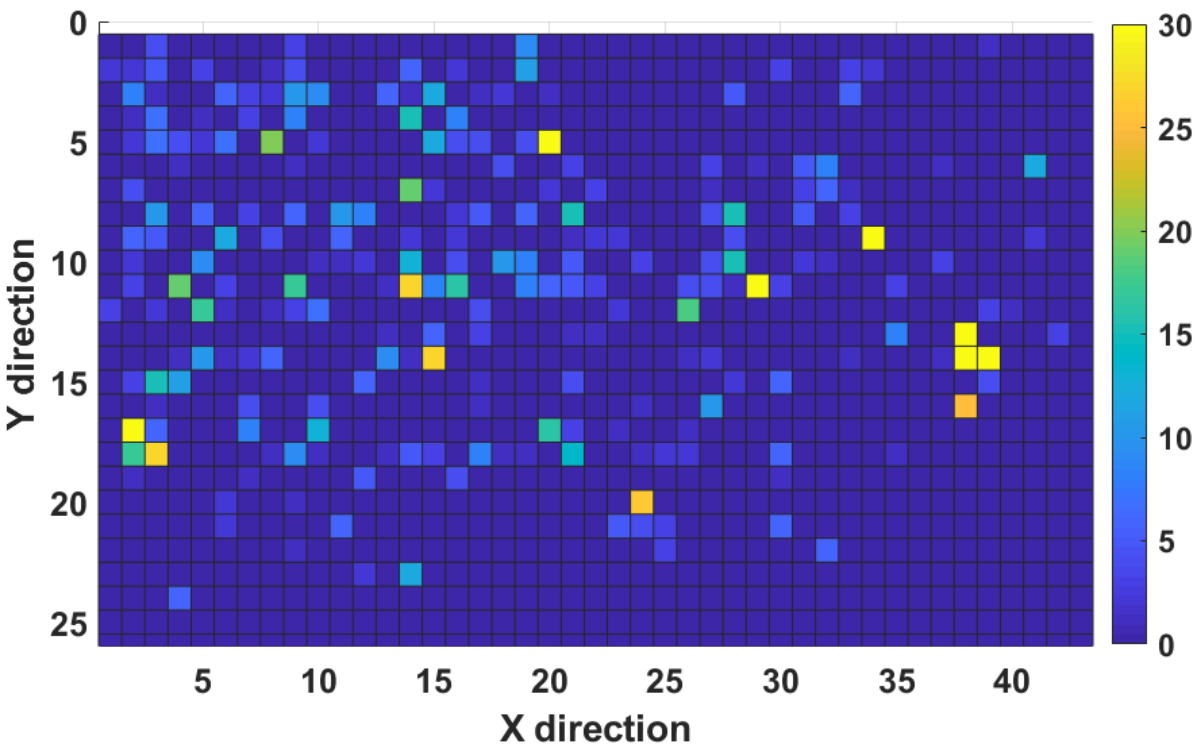


Figure 13: Pinning map

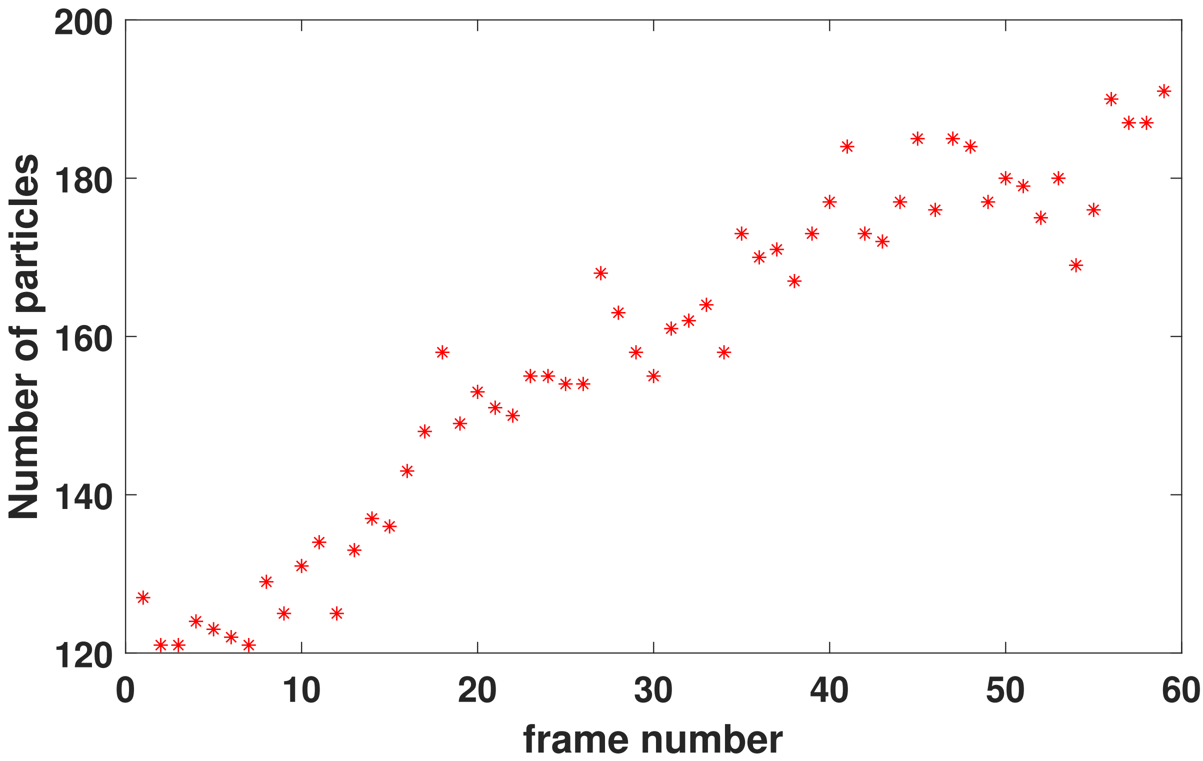


Figure 14: Total number of particles in each frame

4 Analysis

The aim of this TP4 is to collect statistical information about those particles. The first data of interest is the velocity. The figure 15 shows the average of the velocity of the particles of one frame for each frame. One can see that the averaged velocity is basically constant along the movie. At the beginning and at the end of the movie,the velocity is very low. Indeed, the particles almost don't move at those times. There might be no current pulse. Considering only the time during which there seems to be a current pulse, the averaged velocity over the frames is (8.4 ± 0.8) µm/s.

The figure 17 shows the velocity of the particles in function of there position along the X axis. One can see that there are more particles on the left of the movie, where the particles enter than on the right. There is no clear correlation between the velocity and the position on the X axis. The figure 18 shows the same data but the velocity is averaged over the Y axis. On can see that the distribution gets wider around the average because there are less particles in this area.

An histogram of all the velocities is shown in figure 16.

The figure 19 shows the mean velocity within a unit square of 2.3 µm^2 in function of the particle density of this square. Here a unit square contains maximum 3 particles. One can see that the global velocity inside a unit square decreases when the number of particles within the unit square increases. Figure 20 shows the same phenomenon but the density is here the mean density of a unit square of 2.3 μm^2 over the frames.

Some cases of fusion and fission of particles have been observed previously. As mentioned before, if two particles merge together, one of them is considered as "dead". In the same way, if one particle splits into two particles, one of the two is considered as a new particle. The figure 21 is an histogram recording such events. "Fusion" represents the particles ending due to there fusion with an other particle, "Disappearance" represents the particles ending just by disappearing, "Fission" represent particles that come from the split of a particle and "Appearance" represents the particles that spontaneously appear. Figures 22 and 23 show the locations where such events happen. One can see that most of the "starts" and "ends" of particles happen were there is the higher density except for "disappearance" of particles. Indeed, particles tend to disappear mostly at the edge of the area. The edges might be less stable areas for skyrmions.

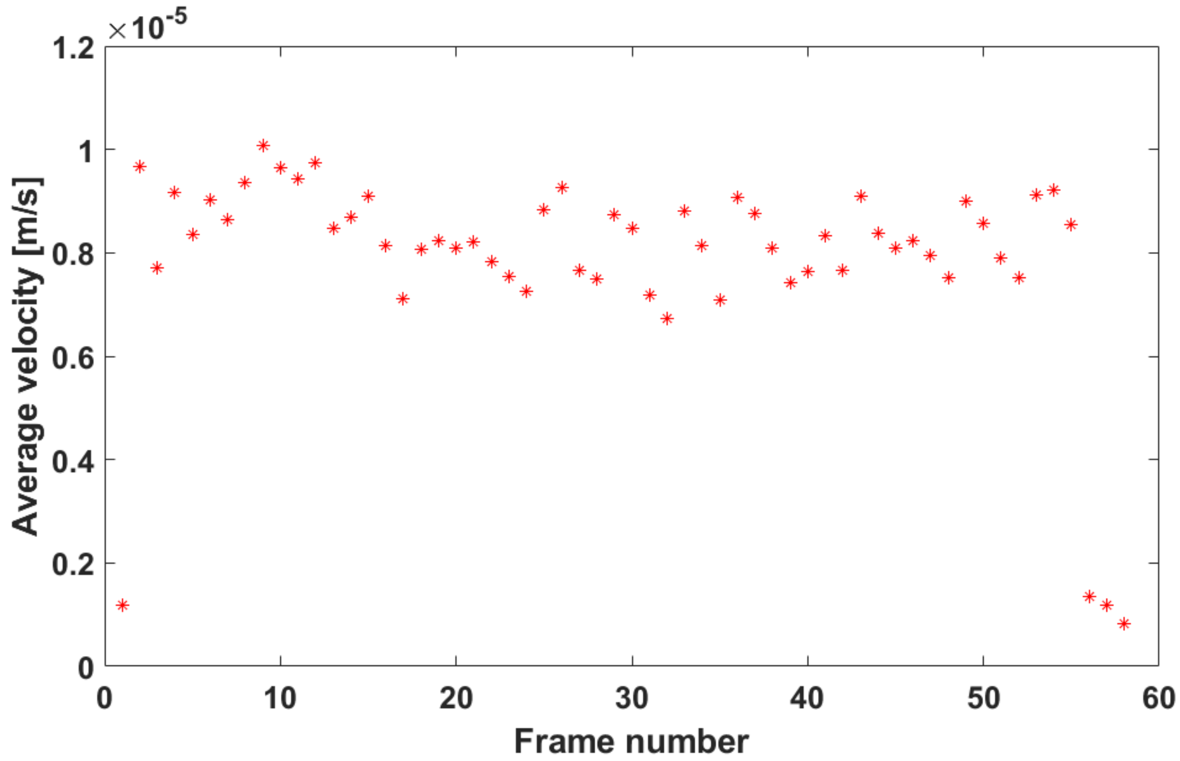


Figure 15: Average of the velocity of the particles of one frame for each frame

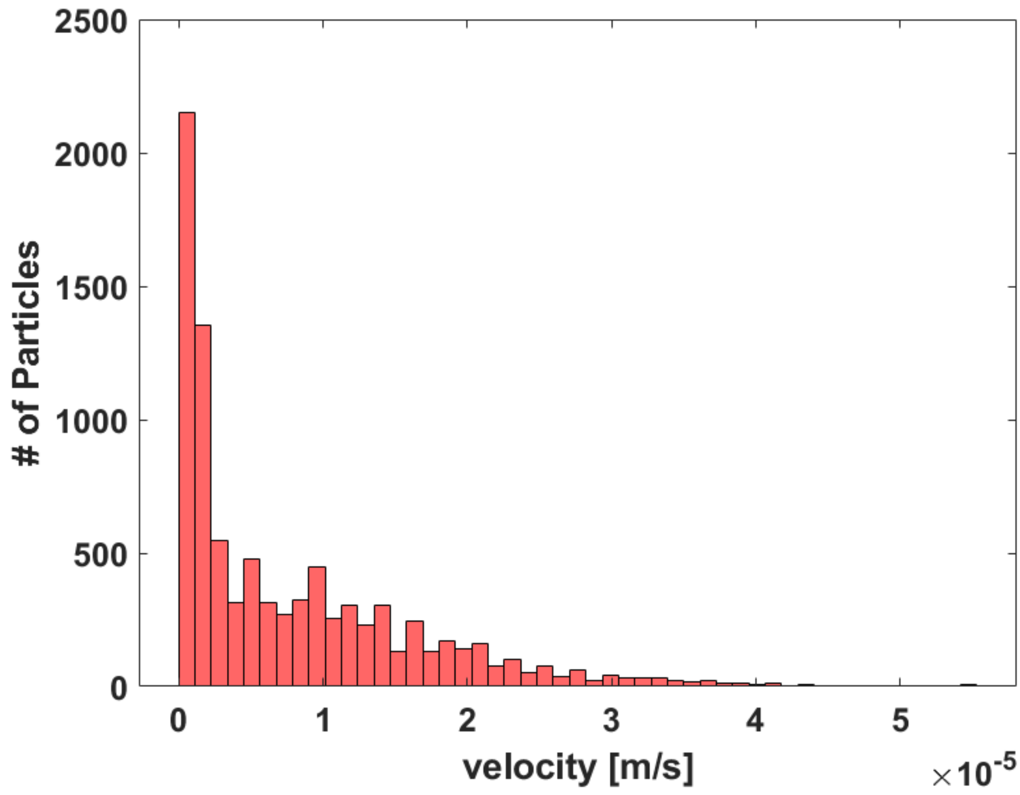


Figure 16: Histogram of the velocities over the frames.

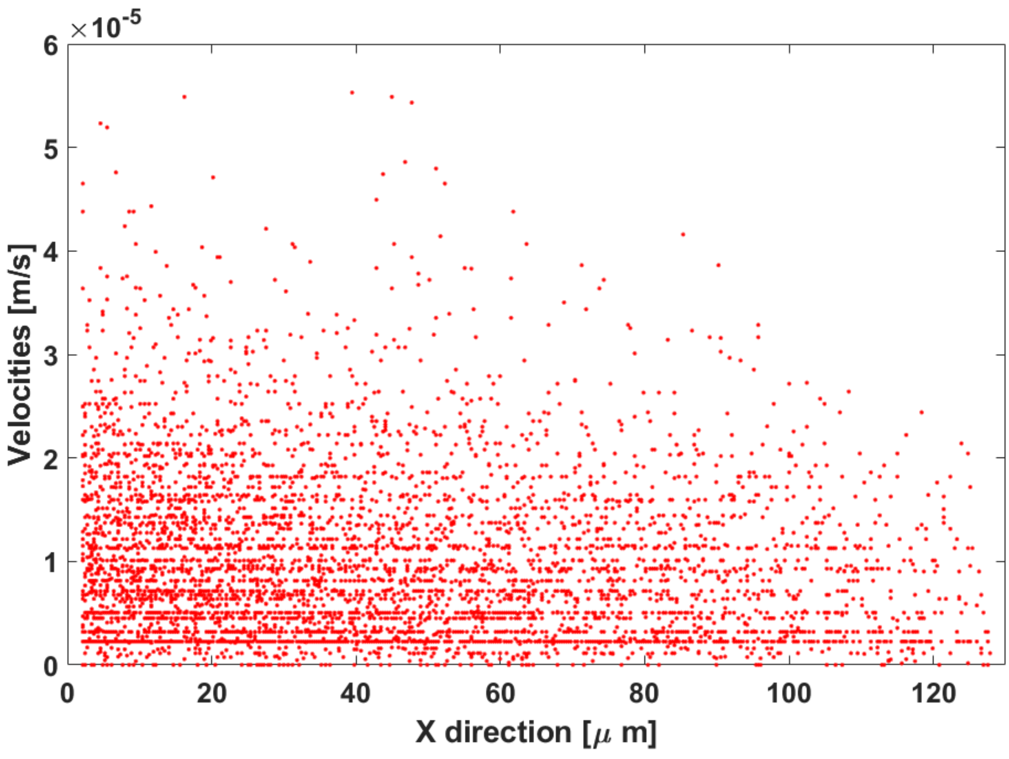


Figure 17: Velocity of the particles in function of there position along the X axis

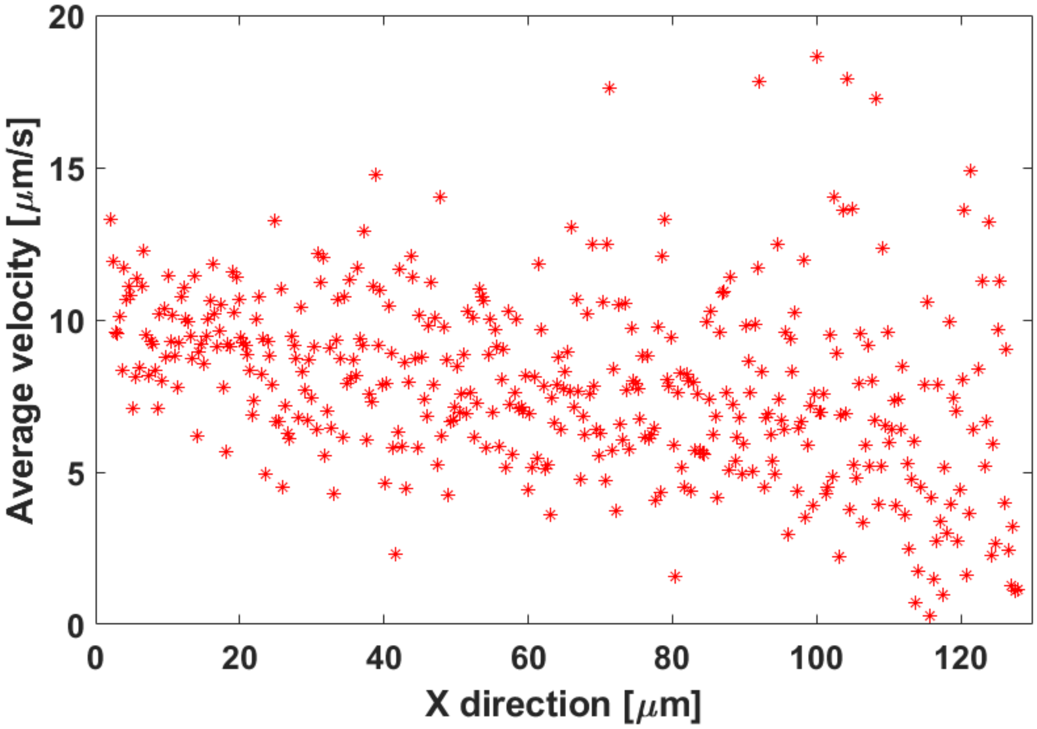


Figure 18: Average of the velocity over the Y axis of the particles in function of there position along the X axis

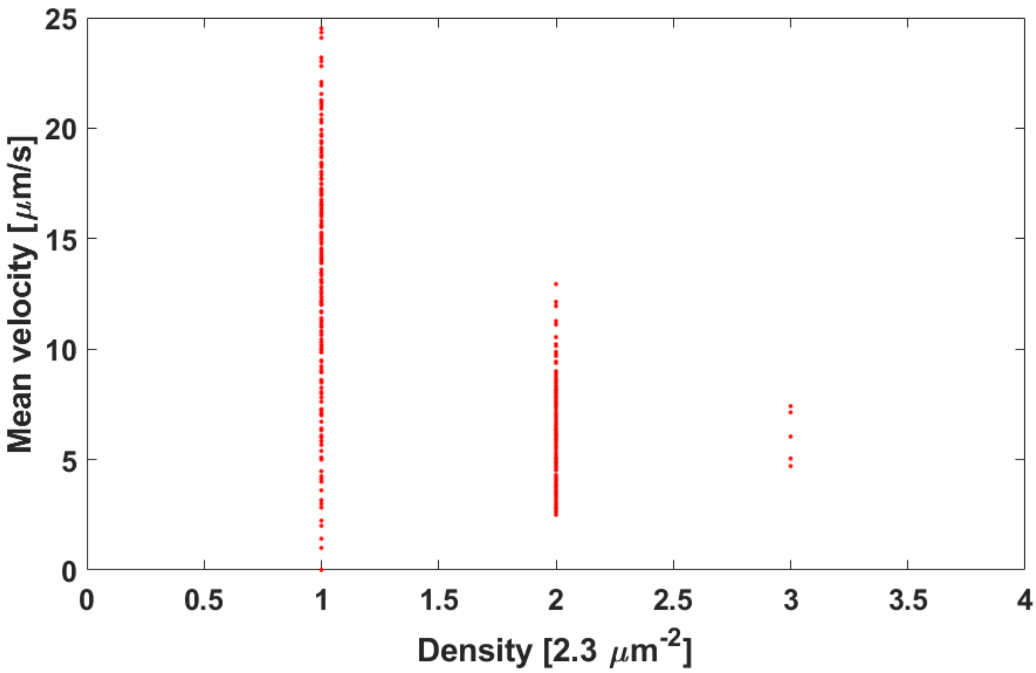


Figure 19: Mean velocity within a unit square of 2.3 μm^2 in function of the particle density of this square

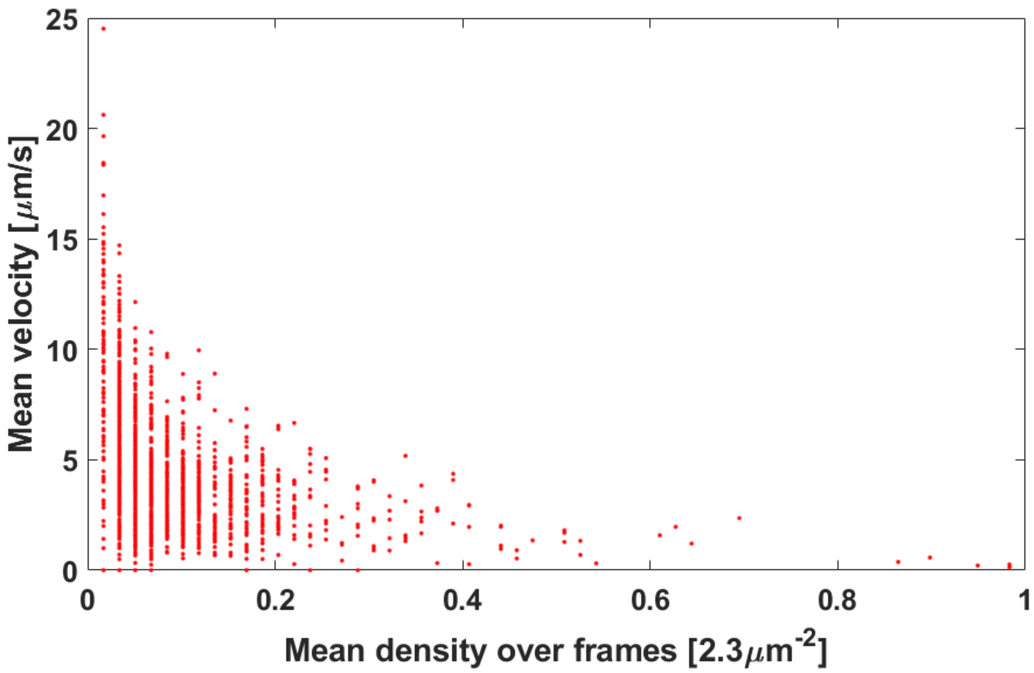


Figure 20: Mean velocity within a unit square of 2.3 µm^2 in function of the mean particle density over the frames of this square

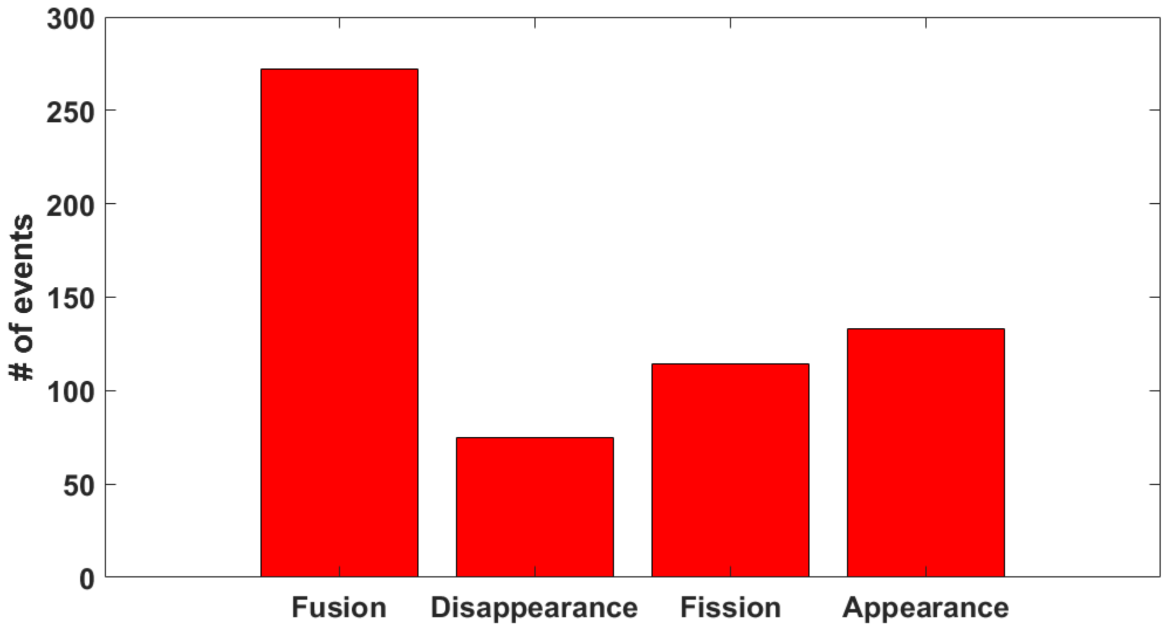


Figure 21: Histogram of the "starts" and "ends" of particles causes

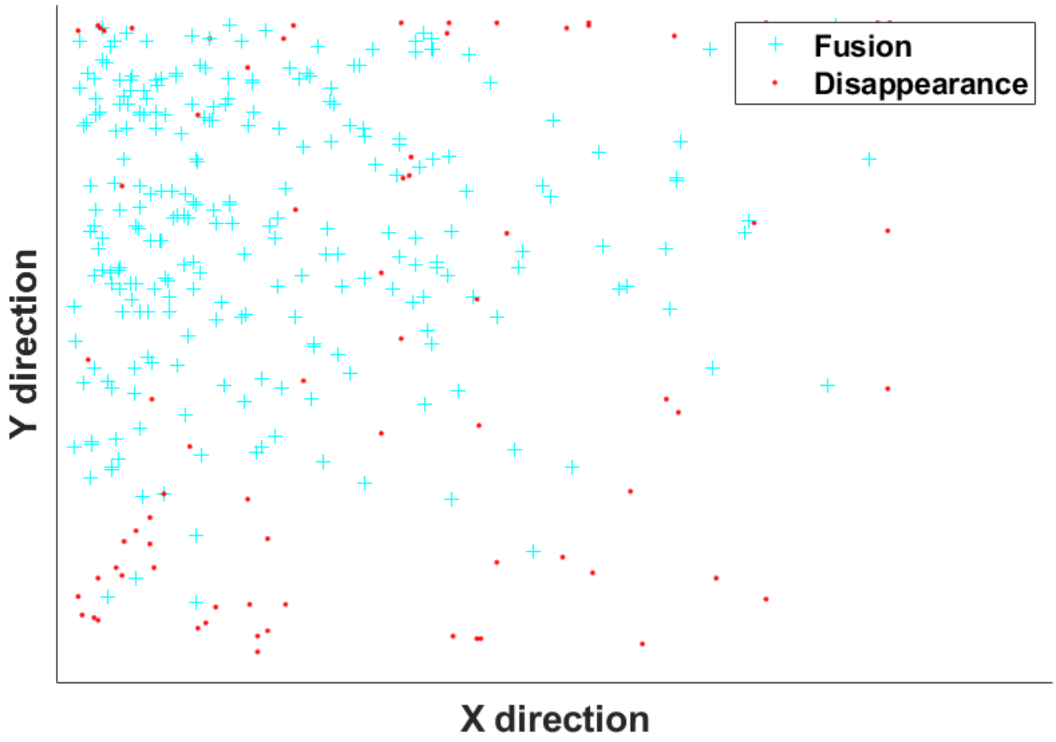


Figure 22: Location of "ends" of particles

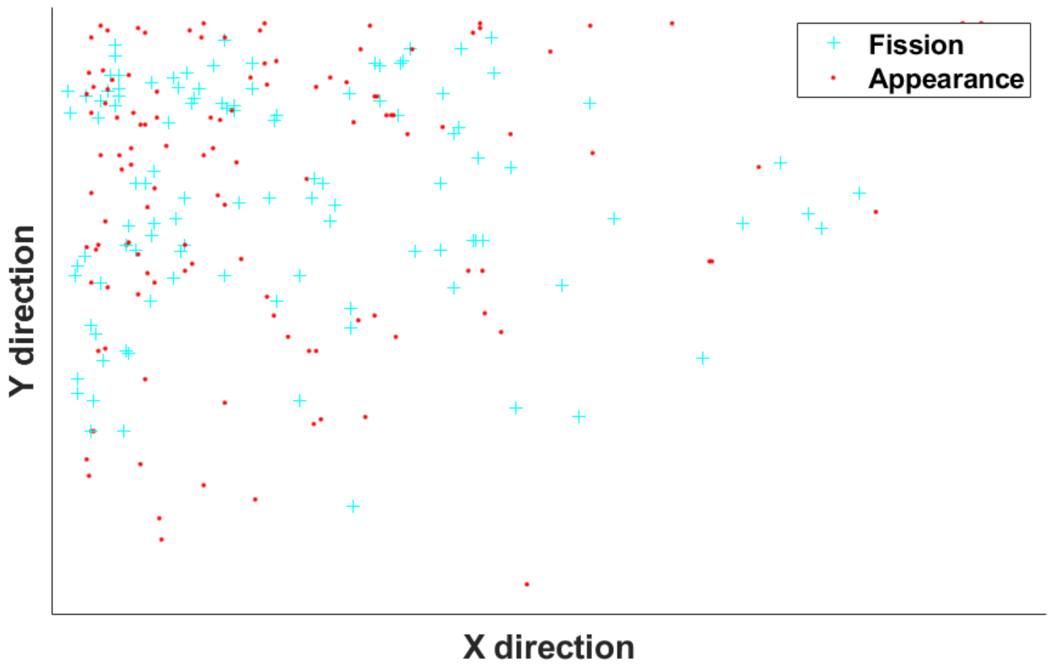


Figure 23: Location of "starts" of particles

5 Conclusion

In this TP IV, the tracking algorithm developed through the previous TP IV "Tracking of Skyrmion Motion with matlab" has been highly improved in order to be more specific to the Skyrmions. Anew and more efficient identification algorithm [4] has been used. The tracking process is now split into four successive steps allowing fusion and fission of particles. The global observations show that the particles tend to follow a same path but can't tell if a first particle created an attractive environment for the following particles or if the environment was already attractive. By repeating the experiment, we could determine if those paths that are often taken are at the same place If so, it means that this phenomenon is due to the material properties. If not it would be understood as a collective behaviour property.

References

[1] Nicolas Chenouard, Ihor Smal, 2014 : Objective comparison of particle tracking methods
https://www.nature.com/articles/nmeth.2808

[2] Sanna Agren : Object tracking methods and their areas of application: A meta-analysis
http://www8.cs.umu.se/education/examina/Rapporter/SannaAgrenFinal.pdf

[3] S. Seki, 2012: Observation of Skyrmions in a Multiferroic Material

[4] Daniel Blair and Eric Dufresne : The Matlab Particle Tracking Code Repository http://site.physics.georgetown.edu/matlab/