-TP4 Report-

Dynamics of Defects in Field-Induced Skyrmion Lattice Melting

Li Guanghao

Guanghao.li@epfl.ch

Prof.: Henrik Rønnow

Tutor: Schönenberger Thomas

Content

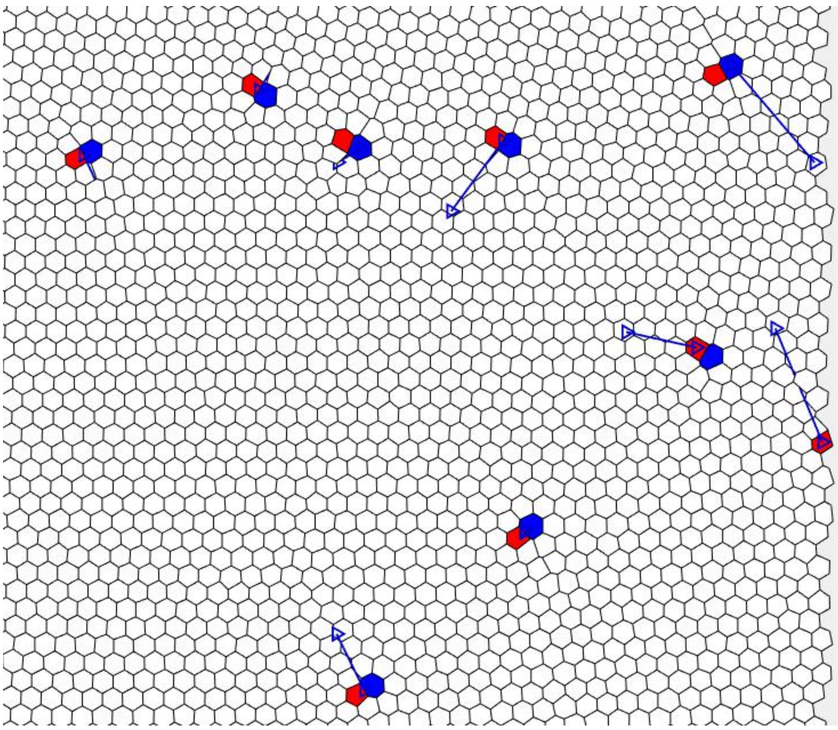

Figure 1: An example of tracking of defects

Introduction

What is a magnetic skyrmion?

A Skyrmion is a kind of quasi-particle named after the British physicist Tony Skyrme, who introduced the concept first in particle physics. [1]

In the magnetic sense, this quasi-particle is defined by the spin structure of the material. It presents a stability of structure and can be driven by electrical current in conductors. A skyrmion can be created, annihilated and driven by an external stimulus. Its stability is based on both Heisenberg model:

$$H = \sum_{ij} J_{ij} \vec{S}_i \cdot \vec{S}_j$$

and Dzyaloshinskii-Moriya interaction (DM interaction):

$$H_{DM} = \mathbf{D}_{ij} \cdot (\mathbf{S}_i \times \mathbf{S}_j)$$

with i and j representing the neighboring spins. To achieve the minimum energy, the former tends to form a parallel or anti-parallel arrangement of neighboring spins, and the latter tends to make them perpendicular one to another. Under these two interactions, neighboring spins finally form an angle between each other. The DM interaction usually occurs in system with a lack of inversion symmetry, hence skyrmions can exist in chiral materials. [2]

Besides, there are other mechanisms that may stabilize skyrmions: like the dipole-dipole interactions in conjunction with anisotropy. [3]

In our project, the material is Cu2OSeO3 single crystal, a chiral magnetic insulator. skyrmions are stabilized in the limited window of temperature (T) and applied magnetic field (H) over the helimagnetic ground state, and can appear as an independent quasiparticle or in hexagonal lattice form within a plane normal to H (figure 2). [4]

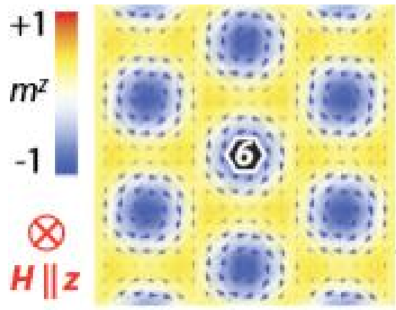

Fig.2: Magnetic skyrmion crystal formed within a Cu2OSeO3 plane normal to applied magnetic field (H). Background color represents the out-of-plane component of local magnetization vector. Extracted from [4].

Interests?

We are interested in skyrmions mainly because of its potential in information area, one example is the memory method called racetrack memory[5] in a conductor. Otherwise, it can be applied in logic devices[6].

What do I do?

To make use of skyrmions, one should first know its properties and how it behaves. We observe skyrmions in Cu_2OSeO_3 between the helimagnetic and ferrimagnetic phase, in a window of temperature T and field H. However, when we increase the magnetic field, some excitations, as we call defects, will form in the skyrmion lattice, whose amount keeps increasing, destroying the crystalline skyrmion order, till the material enters the ferrimagnetic phase. This process can be considered as the melting of a skyrmion lattice, which consists of a two-step phase transition. (figure 3) Experimentally we use Lorentz transmission electron microscopy (LTEM) to acquire images. [7]

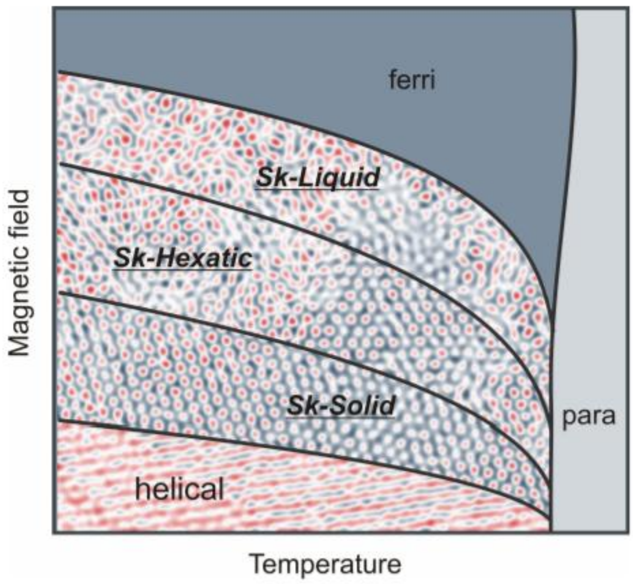

Fig.3: Phase diagram of Cu2OSeO3. The melting process from Sk-solid to Sk-hexatic phase and then from Sk-hexatic phase to Sk-liquid phase. Extracted from [7].

Starting at a certain field and temperature, we can either increase the temperature or field to melt the lattice. Experimentally, it is easier to control the field in our case, hence field melting. To understand the dependence of conditions on defects, it is helpful to know the dynamics and statistics of defects. In previous work of our group, the two-step melting process from crystal to liquid with an intermediate so called hexatic phase has been observed. (figure 4.b)

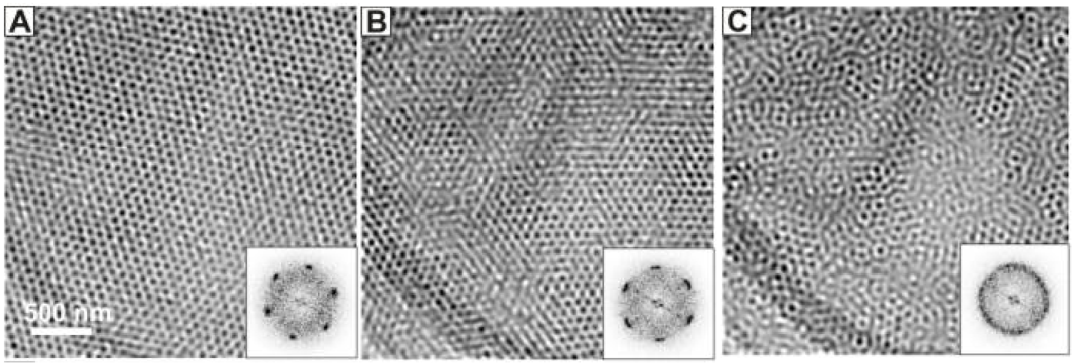

Fig.4: Microscopy image of (a)sk-solide phase, (b)sk-hexatic phase and (c)sk-liquid phase. In the corner of each is the Fourrier transform of image. Extracted from [7].

Since we can only work on 2D images, we select a sequence of consecutive frames performing the melting process to study. My work is mainly to achieve the identification, segmentation and tracking of defects and analyze their dynamics. In detail, the positions, the motion tracks, life expectancy, velocity distribution, etc. We will determine these observables among 488 frames while experimentally the field is ramped continuously from about 600 Oe to 1200 Oe.

More generally, the main part of this work - multiple object identification and tracking - can help do statistics in many other aspects. Since the identification and counting capacity of human visual system, especially facing large quantities of points, is strongly limited, we need automatic analyzing methods to accomplish some target. In this sense, in the domains like astrophysics, medical imaging, robots, action identification, etc, object tracking has its far-ranging application.

1. Pretreatment

Thanks to the pre-processing of images done by my tutor, the skyrmion positions in the original microscope image have been identified. Using Matlab, we can use the Voronoi diagram to visualize important information.

The Voronoi diagram is a topology of a point set, with each point called site, and for each site there is a corresponding area, which contains all the points closest to this site. As a result, a point set is converted into a plane segmented by the way the original points are placed. In this sense, the information of the image is conserved.

The Voronoi diagram allows us to determine some topological properties of the image, which is more obvious to our visual system. Instead of the individual skyrmion position we now display the coordination number of each skyrmion. Matlab will record positions of both sites and vertices of each site. Normal skyrmions are six-fold coordinated, while defects are 5-, 7- or other fold coordinated. After patching and colour coding the defects, they become visually easy to identify (compare to figure 4) and we obtain the figure 5.

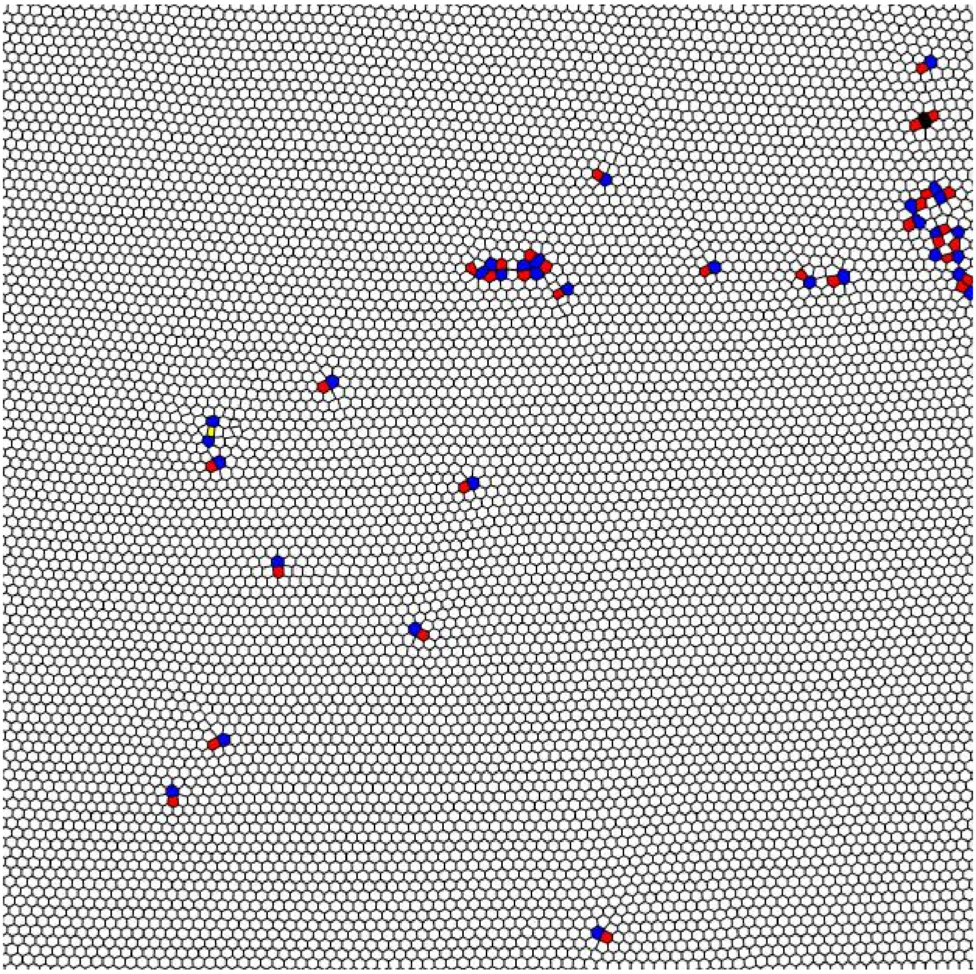

Fig.5: Defects in a Voronoi diagram. White dots represent normal skyrmions, others represent defects with different number of vertexes. (red for 5-defect, blue for 7-defect)

Since the Voronoi diagram is a topology of original image,each patch represents a skyrmion. The calculations we will do later are suitable both for Voronoi frames and original frames.

2. Statistics

2.1. Defect Identification

From figure 4 we can observe that isolated defects do not exist (which is true for other frames). Defects appear in pairs or as larger clusters. We want to first identify these clusters, with their size (or mass, a more particle-like saying) and mass centre. Again thanks to the Voronoi diagram, a defect cluster can be easily defined as a group of defects with vertexes in common (see figure 6). As vertexes of all sites are already determined once the Voronoi diagram is set, we can identify each cluster with its size. We can also calculate their average coordinate as so called mass centre. We number these clusters, knowing clusters is equal to knowing defects. From this step on we will consider the collective behavior of defects with cluster as unit, no longer single defect.

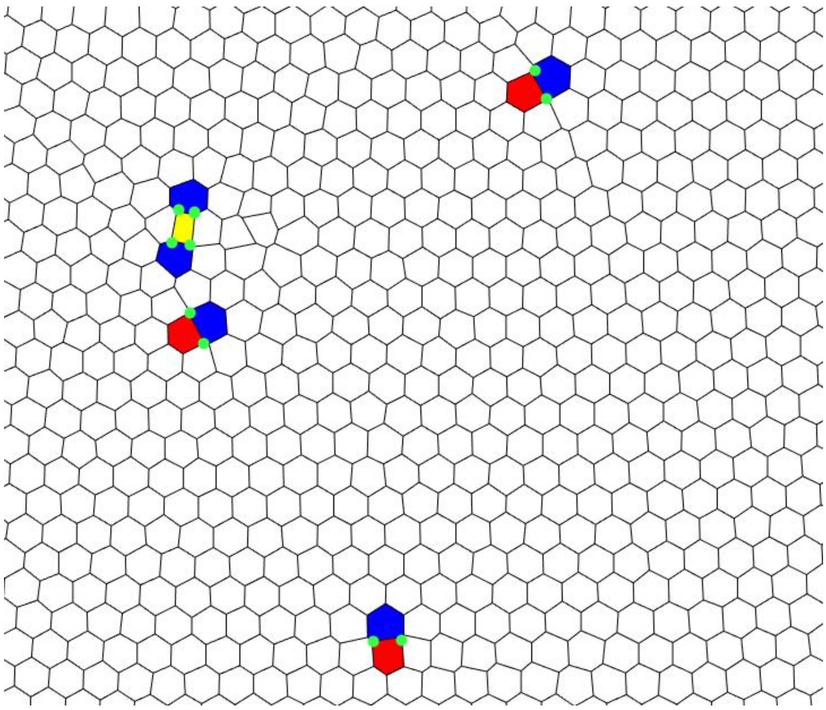

Fig.6: Cluster identification example, green points imply the shared vertexes

As a result, we can determine the number of clusters in each frame and do statistics. From figure 7 one can conclude that most defects appear as pairs (size 2) and the larger the size is, the more difficult it is to be found. Defects on boundaries are recorded, who have contributed to the group size=1.

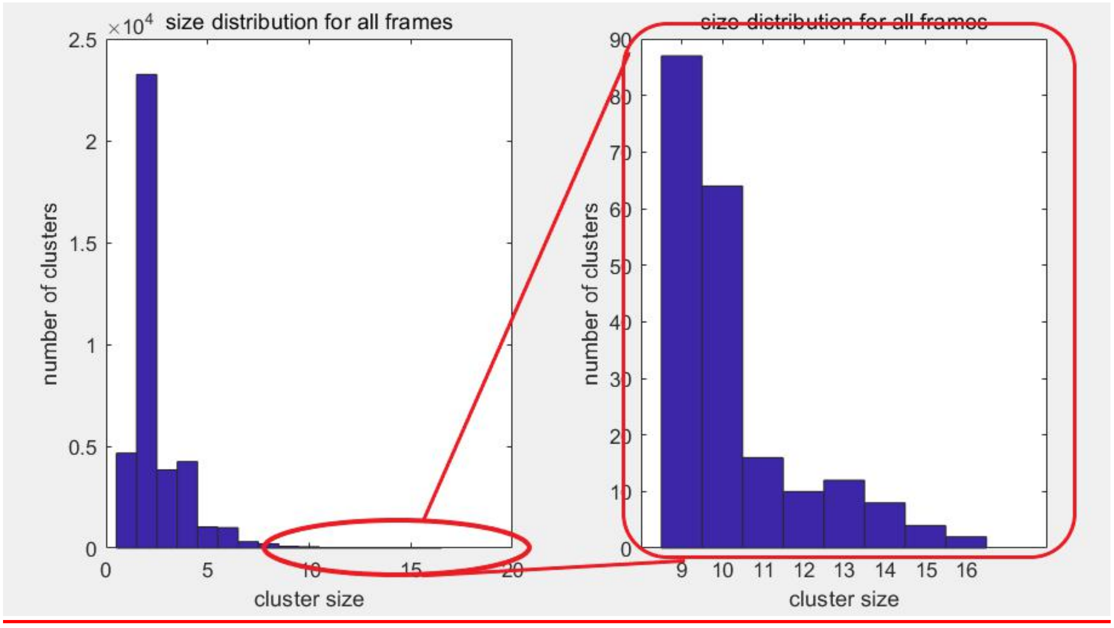

Fig.7: Distribution of cluster size for all frames

Furthermore, we determine the number of clusters as a function of the frame number.

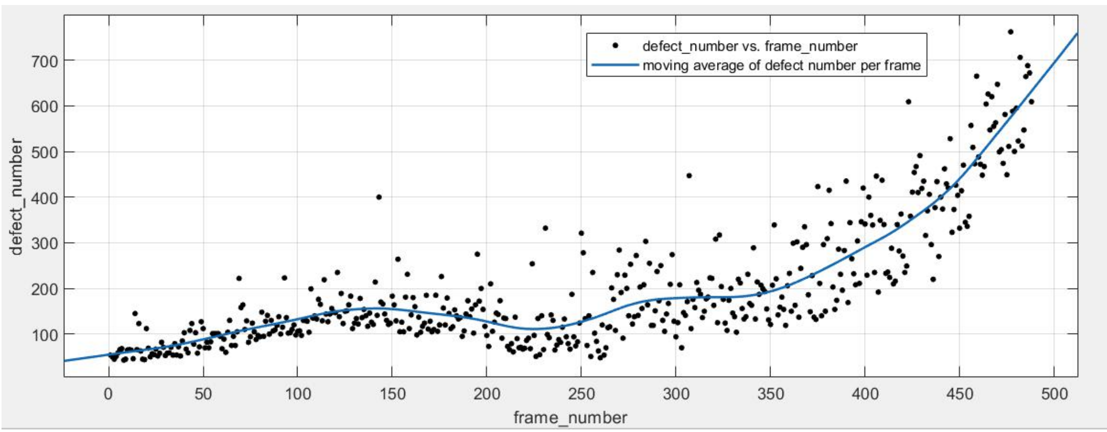

Fig.8: Moving average of evaluation of defect amount as a function of frame number. Number of defects increasing, corresponding to the melting process.

We can also determine the evolution of clusters of different sizes respectively. In figure 9, what could be interesting is that the evolution of 3-defect and 4-defect clusters has a similar evolution mode, which differs from other clusters.

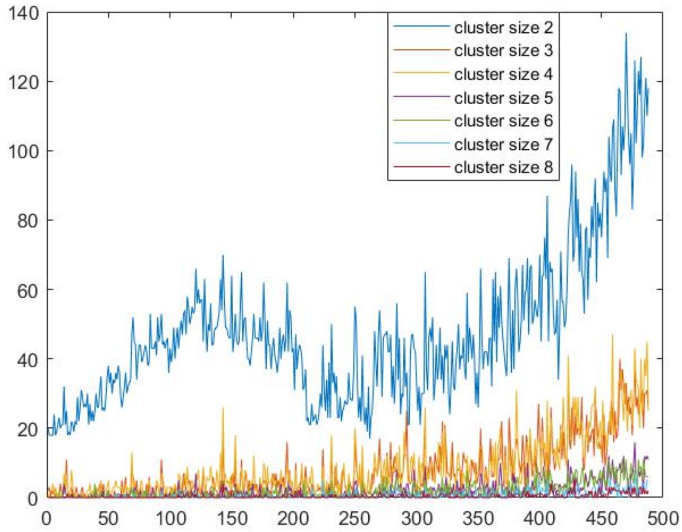

Fig.9: Evolution of number of clusters of different sizes.

2.2. Tracking of clusters

Now we are able to identify defect clusters and identify their mass center in one frame. Next step is to link clusters between a continuous sequence of frames and thus, to analyse the motion of clusters.

It is foreseeable that we will encounter a main obstacle for this, which is that we do not have information between two frames. Thus, we do not know exactly which cluster, in the next frame, is the successor of cluster in the previous frame. To solve this problem, we will pose two assumptions. By assuming that cluster motion is slow enough (velocity can be defined as displacement per transition of frame, hence pixel/frame) and their density low enough, we can consider two closest clusters respectively belonging to old/new frames, to be the same cluster. However, our assumptions are not necessarily true. We will first apply them and later on, we will use some results to ascend back to the assumptions.

As our reference remains static, one should first localize a cluster in one frame, and then search for the closest cluster in the next frame. In case that this cluster disappears and chooses another cluster as its successor, we can set a criterion radius of search, out of which no clusters can be chosen (In fact, this radius is related to defect velocity that we will show later on). We use its first position to determine its second position, then use its second position to determine the third position, and so on. We apply the same algorithm on consecutive frames, until the moment when we cannot find any cluster in the next frame within the chosen radius, which is when we lose the tracking of this cluster.

By linking its position in every moment, we can draw the trace of one cluster. In the algorithm, only clusters that last at least two frames will be recorded. After drawing that by arrows, we come to know that the motion of clusters is usually disordered (thus, without preference or trend of displacement). (figure 10)

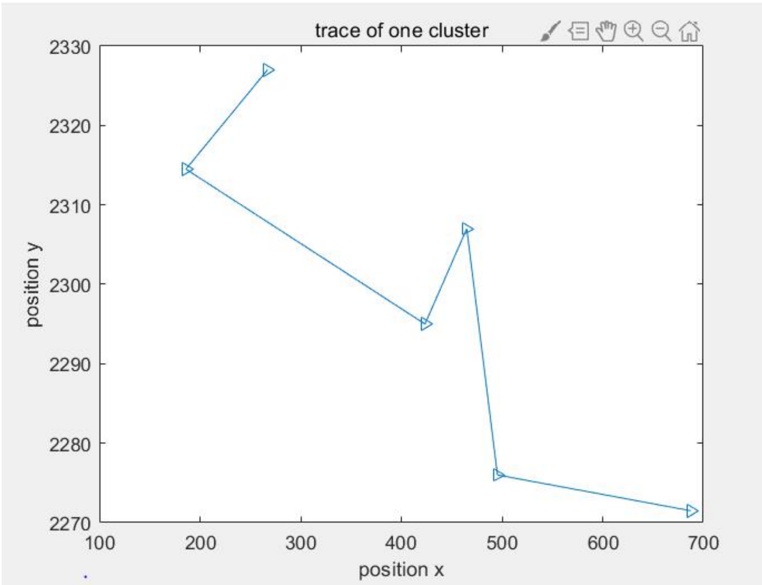

Fig.10: Example of cluster motion among several frames. The arrows show next positions of the cluster and the order of displacement. Positions are in units of pixel

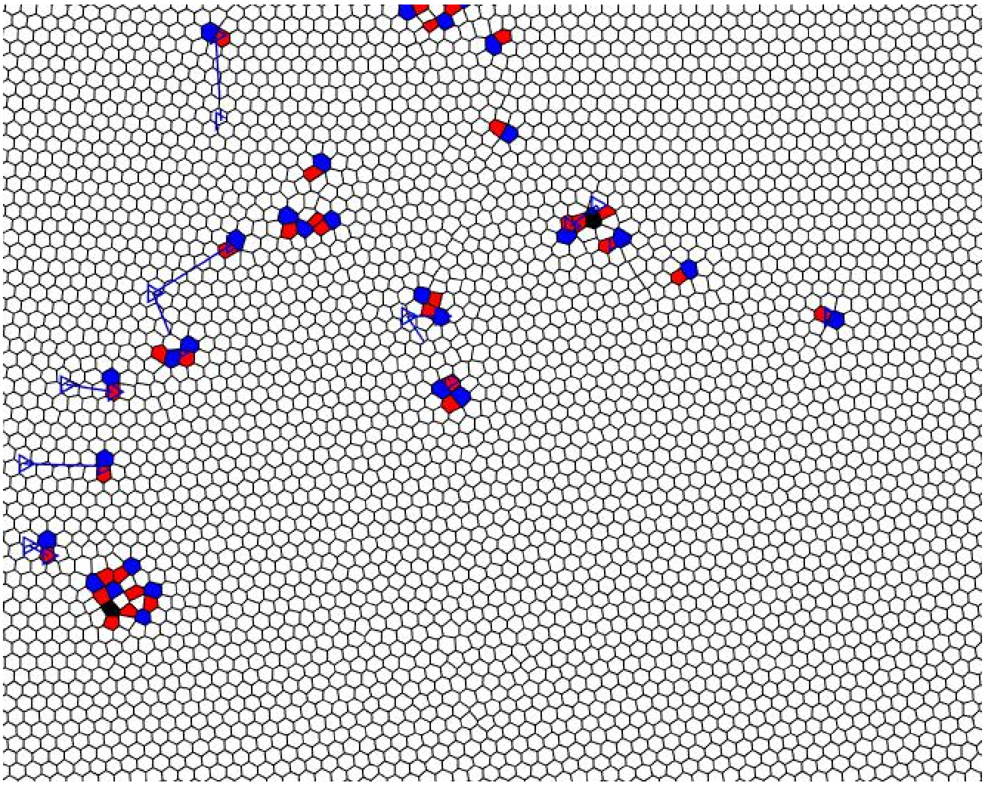

Fig.11: Following Movement of clusters in frame 71. Blue arrows show the next position of clusters

In this process, clusters are lost in almost every transition of frame (figure 12). Also, in each frame we got new clusters, which are not the successor of any cluster in the previous frame. We have also discovered some interesting phenomena, like that some clusters merge into one, which is quite rare (figure 13). Generally, the lost of defect might due to many reasons. It might move too quickly to be tracked, or merged with other defects, or simply disappears...

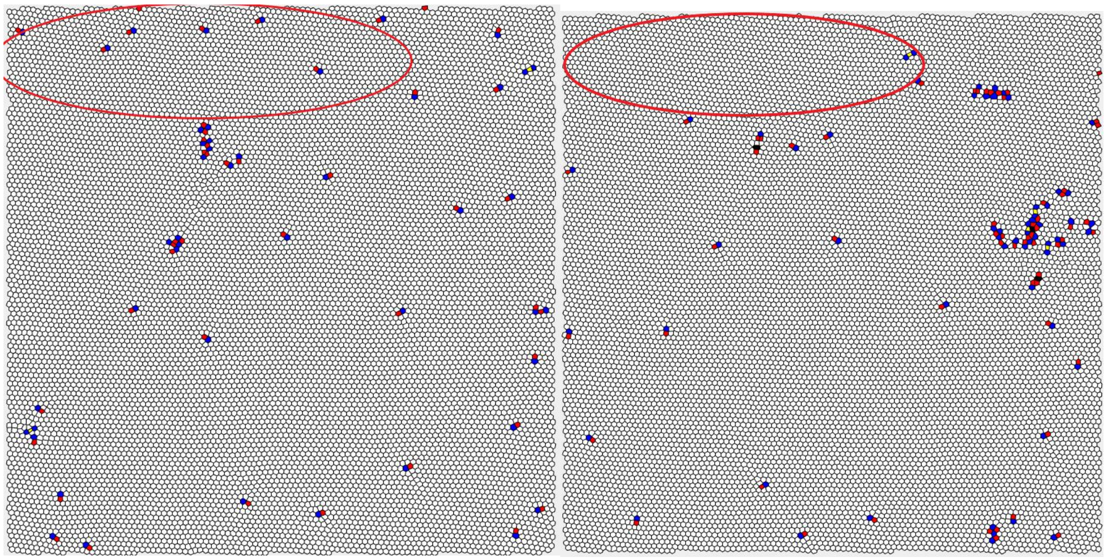

Fig.12: Clusters in top area of frame 300 disappear in frame 301

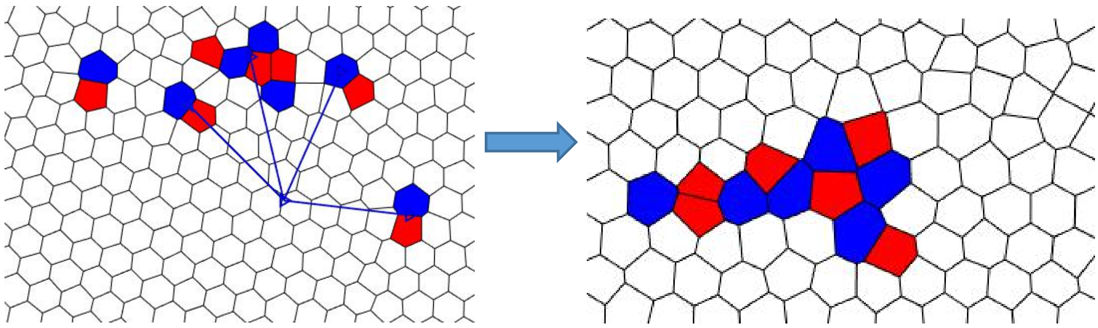

Fig.13: Cluster merging example. Different clusters point to same position

As we have different clusters in different frames (due to their disappearance and appearance), we can set a frame as initial (we set the first position of clusters) in which the clusters are our objects to trace. The result obtained is a structure (a "cell" in Matlab) containing information of all the clusters in the initial frame. The information contains cluster index, cluster size, trace (positions in every moment) and indices of defects in it. In the following part, we will call this structure "Trace".

Otherwise, if we want to achieve cluster tracking without looking into one particular frame, we can combine "Trace" in each frame and eliminate repeat data.

2.3. Life expectancy

Now we know obtaining Trace for all the frames. Generally speaking, among all the frames, some satisfy better our assumption that "cluster motion is slow enough and density is low enough", while some not. In general, the tracking of defects is considered "good" as long as it can last for a long time. As we want to study the collective behavior of defects, we want the clusters to behave "softly" and we want to trace clusters as long as possible. To this end, we want to study the "lifetime" of defects.

The lifetime of a defect is defined as the number of frames in which it exists, including the initial frame. Therefore, the definition "life expectancy" is for one frame, describes the average lifetime of all defects in it. A frame with longer defect life expectancy can be considered a system more stable and observable.

By looking into Trace, we can get the lifetime of a cluster in one frame. By calculating the average lifetime weighted by cluster size, the life expectancy of this frame is available.

One thing should be paid attention to, that is that calculating life expectancy of one frame should exclude the clusters in previous frames. That is to say, the life expectancy is for the "newly generated" defects in the initial frame, in this way the data does not possess any information in previous frames. The evolution of life expectancy vs. frame is shown below (figure 13).

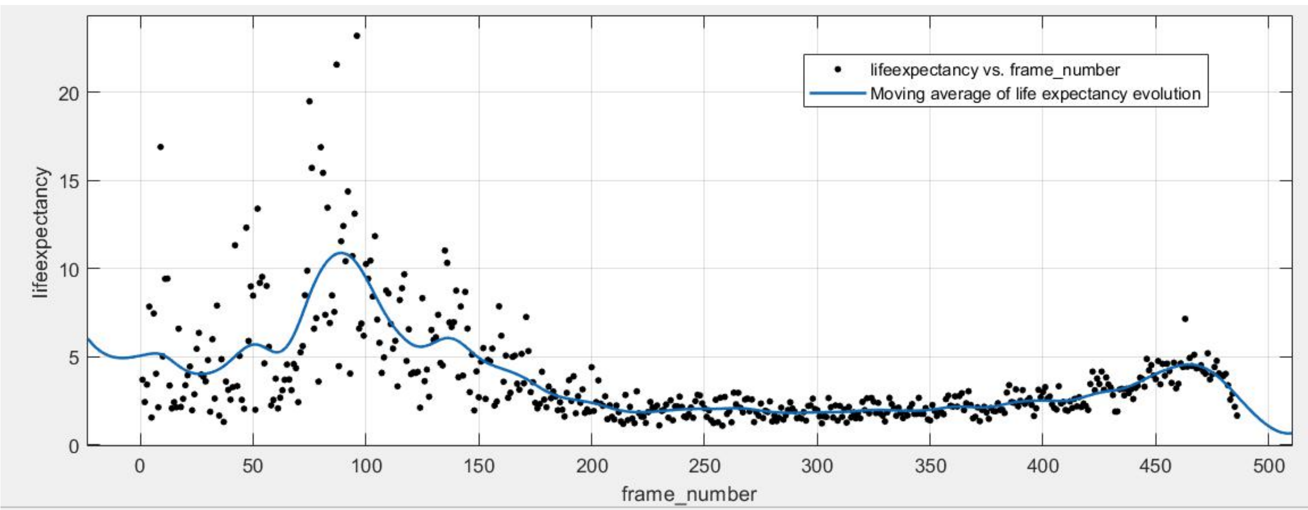

Fig.14: Moving average of life expectancy in units of frame (y) vs. frame number (x)

From the result obtained (figure 12), frame 1 to 150 have a respectively high life expectancy. Thus they are better objects for us to study. In other sense, for latter frames representing Sk-hexatic lattice and transition Sk-hexatic to Sk-liquid, the short lifetime of defects may also simply due to its disappearance. Combine with the result we obtained in figure 8, conclusion is that we have many defects appearing and disappearing at same time, their behavior is much more active than in Sk-solid.

Otherwise, instead of the statistics by frame, we can also look directly into the lifetime of all defects ever appear in the whole sequence (figure 14).

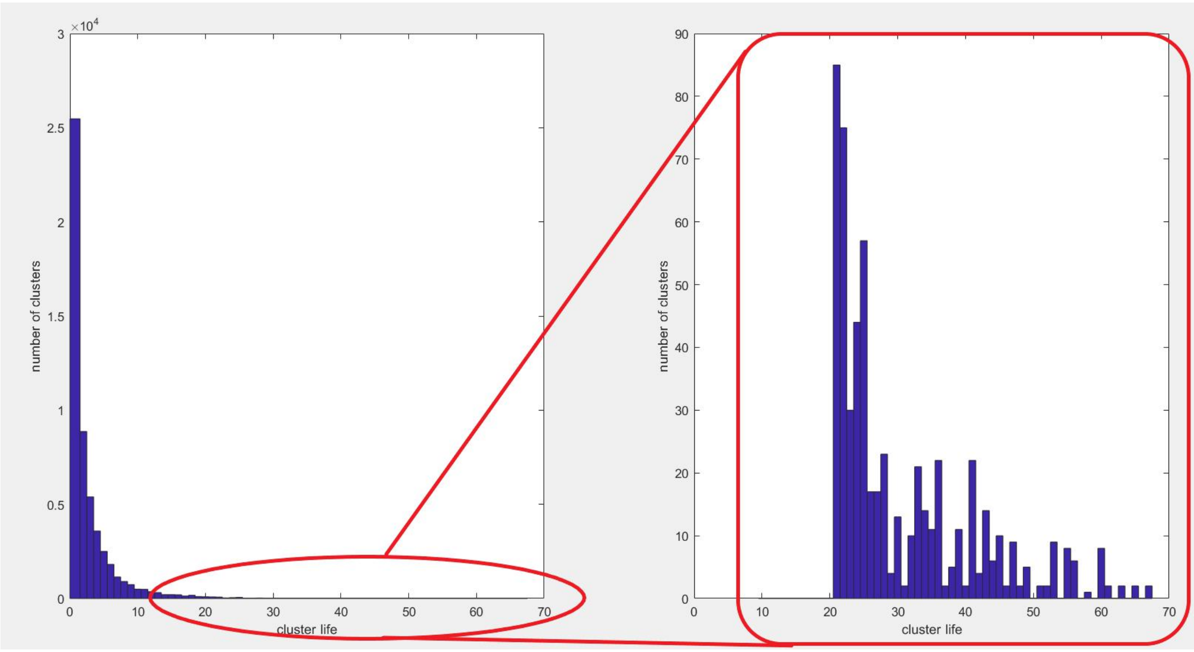

Fig.15: Evaluation of defect amount in function of defect life time

According to figure 15, this distribution is almost exponentially evanescent, which implies that most of defect cannot be tracked for a long term. By end, the average lifetime among the whole sequence is given at 3.78 ($\pm$2.98) frames, and in particular, for first 150 frames, average lifetime is calculated at 6.42 ($\pm$4.04) frames.

2.4. Velocity distribution

After calculating life expectancy of defects among all frames, conclusion is that defects in later frames can not be well tracked anymore. This may due to the high defect density in hexatic and liquid phase which disturbs the tracking, also, it is possible that defects disappear more often in later frames.

Now we want to go further into kinetics. We are going to determine the velocity distribution within frames, to see if the result agrees with previous part.

The method is similar to how we find life expectancy. According to the definition before, the velocity is in units of **[pixel/frame]**. So the velocity of one cluster is defined by average displacement from initial frame to last frame of its tracking. Similarly, as our object is defects, not clusters, the average velocity of one frame is weighted by size of each cluster (figure 16).

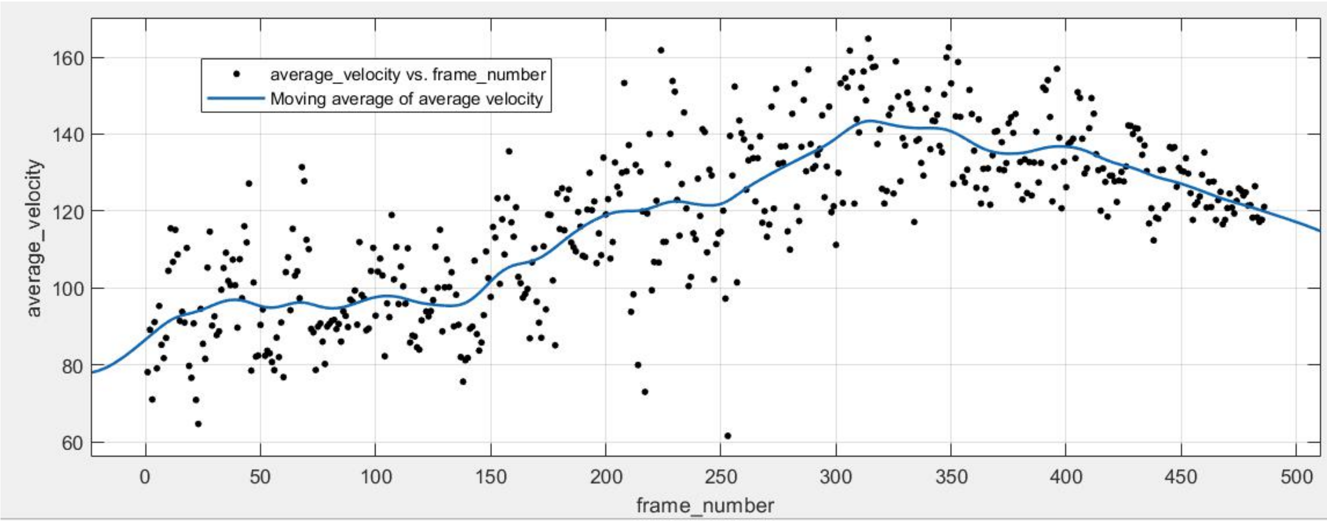

Fig.16: Moving velocity of average defect velocity (y) vs. frame number (x).

One can conclude from result (figure 16), that defects have an average velocity respectively low in first 150 frames. This accord with our assumption of low velocity, and yet defects have longer life expectancy in these frames. By contrast, in sk-hexatic and sk-liquid phase, the average velocity is respectively high, which reduces the reliability of tracking.

Otherwise, we can also give the velocity distribution of all defects ever appeared in the sequence. (See figure 17)

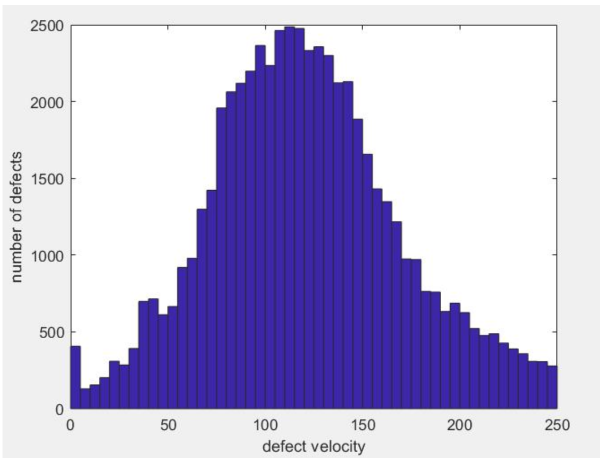

Fig.17: Distribution of velocity of all defects ever appeared

The result is appreciated because it has a form of gaussian, which is in accord with the velocity distribution of particles in perfect gas, and thus reveal a particle nature of skyrmion. The most probable velocity, as seen in figure 17, is around 120 pixels/frame. As mentioned in cluster tracing part (2.2), this result can contribute to the selection of the criterion radius, by considering that the criterion radius approximately equals to two times of the most probable velocity. In our program, the radius is chosen at 250.

3. String-like clusters

In a cluster, all kinds of defect combination are possible, but we are particularly interested in the way that defects form a string (figure 18). In this case, defects in a string are represented by side numbers 4, 5, 7, etc. (6 for normal skyrmions) (it's also vertices number)

Identifying a string is not like identifying defects or finding velocity because we don't have a clear definition for it, all we can use is its geographical property.

Basically, in a string we have two extremities, and between these extremities are some defects connected one by one. The way two defects are connected is that they have a side in common, hence share two vertexes. For extremities, they have 2 vertexes shared with other defects, and for the middle part each has 4 vertexes shared. That is to say, for the shared vertex, we have only choices of 2 and 4. In this sense, a string can be approximately defined as a cluster, containing 2 and 2 only defects with 2 shared vertexes for each. At the same time, it doesn't contain any defect with other than 2 or 4 shared vertexes. Meanwhile its size has to be larger than 2.

With this algorithm we can identify strings with their length in one frame. This helps us find quickly the strings and their size, arrangement, etc.

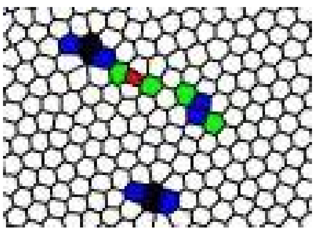

Fig.18: Typical examples of strings, they are 5-8-5-7-4-7, 7-5-5-7 and 5-8-5

Going through the result obtained in all frames, no matter what the length or structure of a string is, they usually obey one rule: Average side number of defects is 6 (for example 7+5+5+7=6×4), equals to that of normal skyrmions. This is understandable through geometric meaning. The physical reason is a local disturbance of the six-fold coordination of skyrmions instead of a global lattice distortion.

Like before, we can have statistics of number of strings in all frames (figure 19):

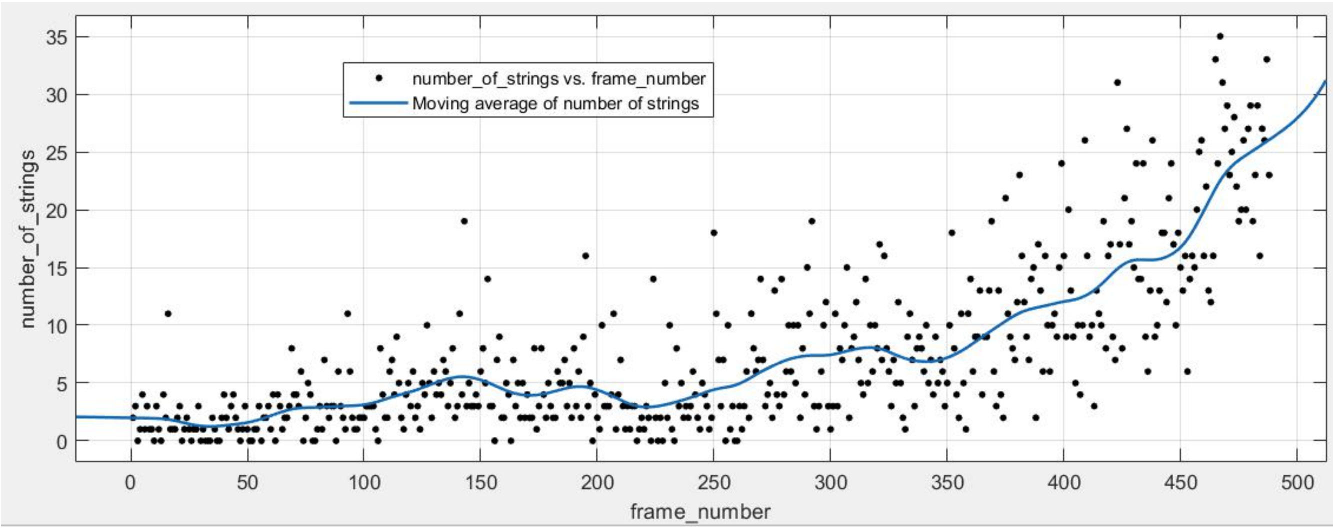

Figure 19: Number of strings as a function of frame number.

Number of strings increases as the lattice melts, but so are the defects. The proportion of string probably has some relations with melting process. To figure this relation out, we can show the evaluation of fraction strings over clusters. (figure 20)

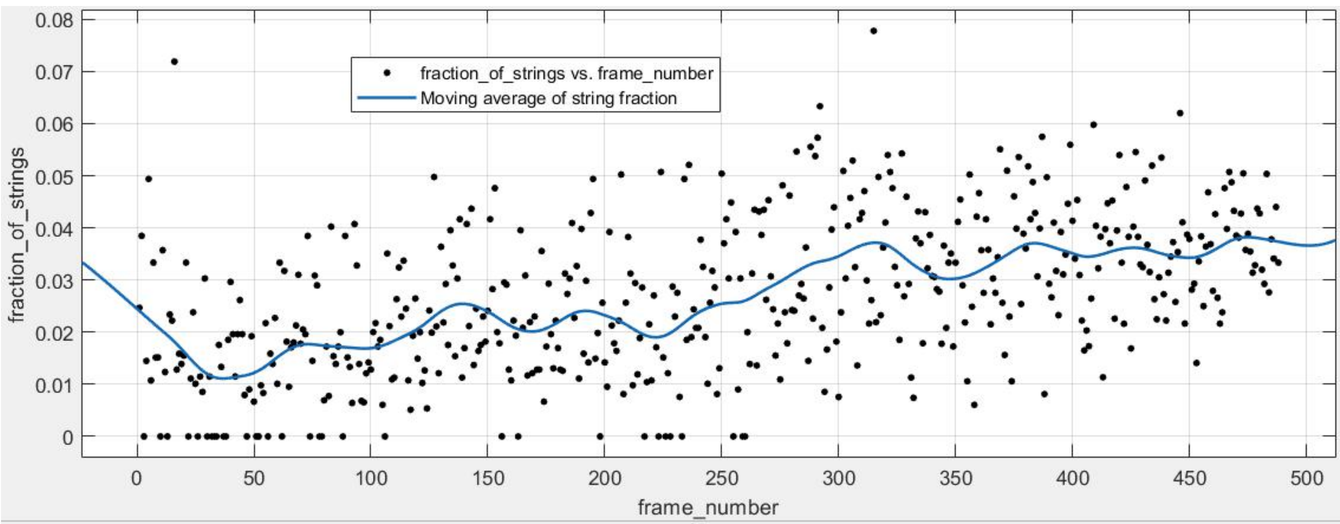

Figure 20: Moving average of proportion of strings as a function of frame number.

Through the moving average we can see a weak tendency of augmentation of the proportion in the process of melting, though the fluctuation still dominates.

4. Velocity map and melting

Till now, collective behavior of defects has been well studied. We can develop more details in skyrmion lattice phase transition process.

The melting process is highly related to the number of defects but we don't know at this point how the melting process is related to the defect concentration (it's increasing, of course). On the other hand, we have already determined the life time of a defect (how long we can track it) and the time it remains at one position or barely moves, which is related to its velocity. So we can make a velocity map of defects (might be related to some pinning potential).

Using data in velocity distribution program, linking with our Voronoi diagram, we can give value of velocity on positions where the defects are. For non defect-occupied position the velocity value is null. In order to describe the velocity map, we use matlab function "contourf" to draw equivelocity surfaces.

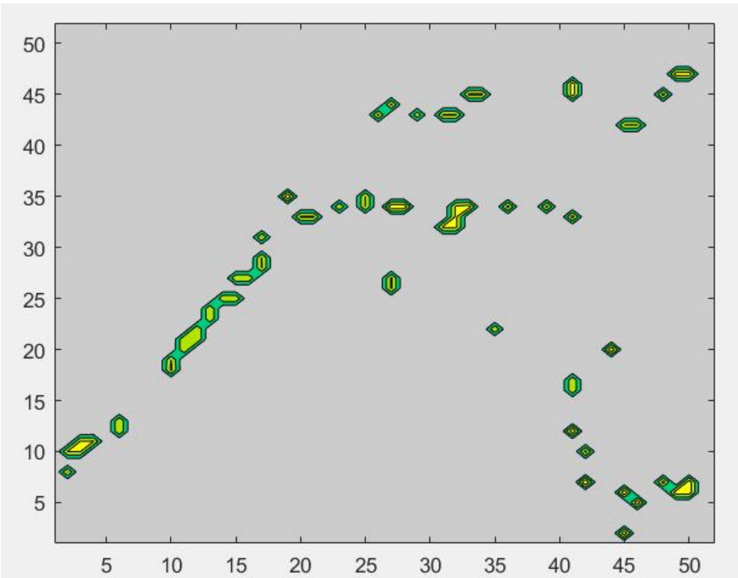

Fig.21 Velocity map of frame 109, colors from green to yellow indicate velocity from low to high. Grey part indicates background without defect.

Since the distribution of defects is quite discrete, the positions with non-zero velocity are shown by dots, thus we do not have a good vision for one single frame. A solution is that we can draw average velocity map of several consecutive frames (figure 22), which is feasible since velocity data of each frame are irrelevant, thus addable.

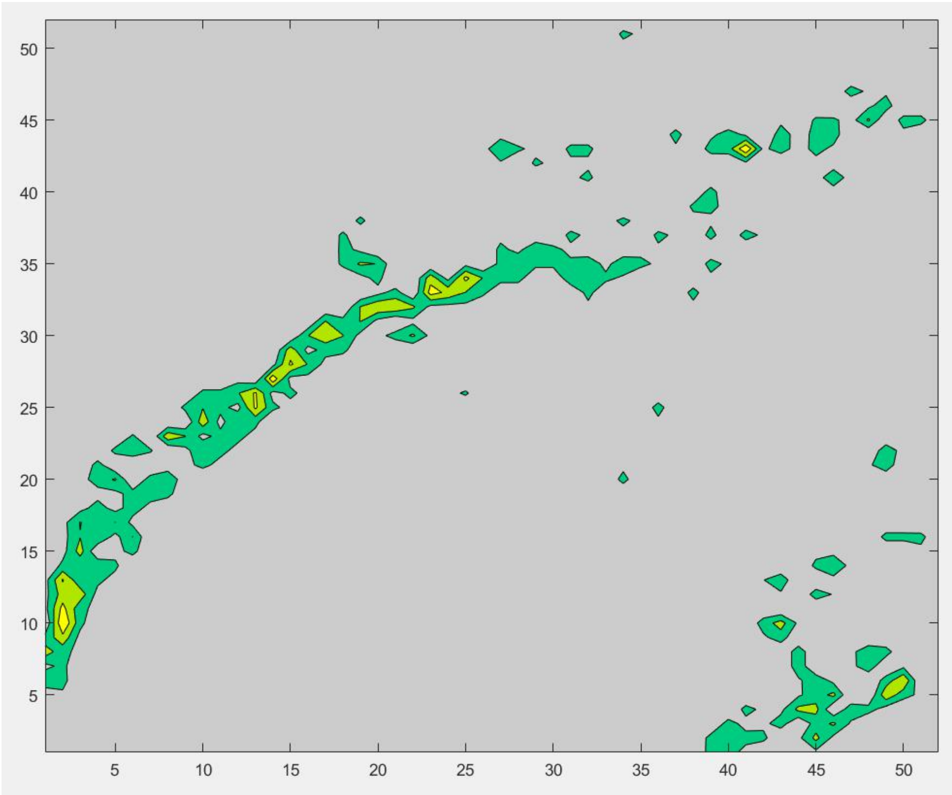

Fig.22 Average velocity map of consecutive frames 109-119. Colors from green to yellow indicate velocity from low to high. Grey part indicates background without defect.

In fact, in frames around 109 we happen to observe a skyrmion domain boundary in the diagram, which consists of many defect pairs moving slowly together, shown as the green band in figure 22.

We can compare between different skyrmion lattice melting process. (figure 23)

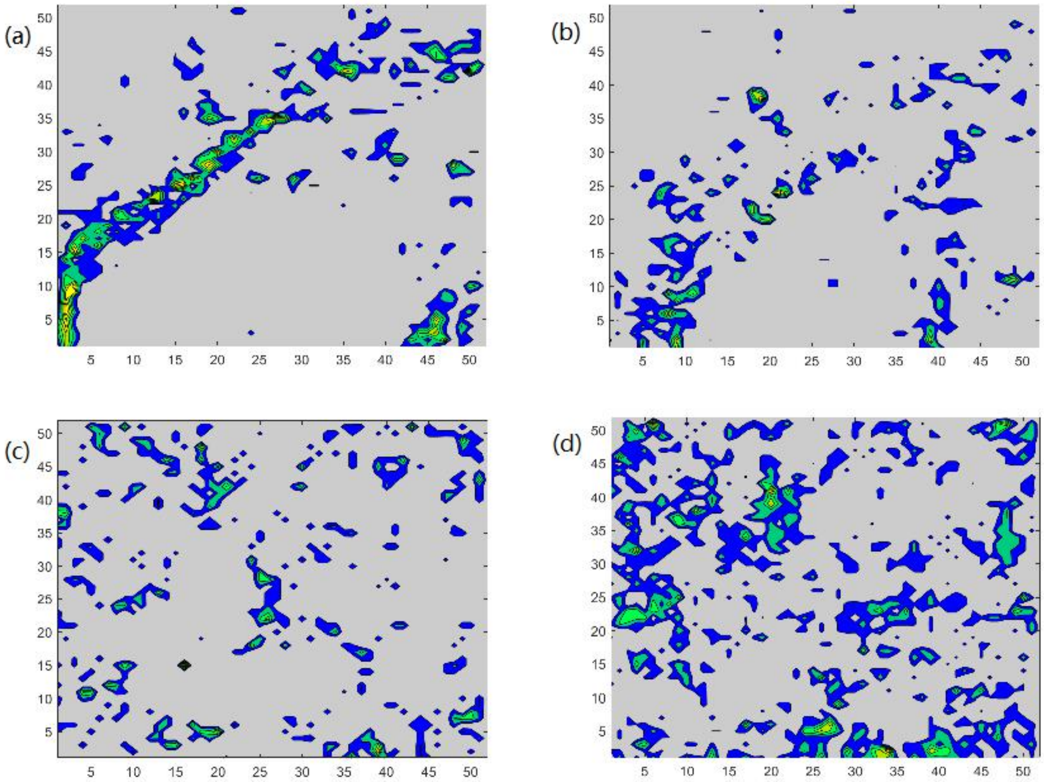

Fig.23 Average velocity map of consecutive frames. (a)frame 121-131, (b)frame 176-186, (c)frame 336-346, (d)frame 421-431. Colors from blue to yellow indicate velocity from low to high. Grey part indicates background without defect.

5. Conclusion

After our coding, identification of defects in a skyrmion lattice melting process is achieved. We can now track defect clusters in one frame, or any particular defect cluster we want in the whole process. According to our statistics, numbers of defects keeps increasing during the phase transition from Sk-solid to Sk-hexatic and from Sk-hexatic to Sk-liquid, but it retains its value when skyrmion lattice is in hexatic phase (figure 8).

For the kinetic part, we found that defects in hexatic and liquid phase usually have a active behavior, and are more difficult to get around. We still have the myth for strings and its relation with the melting process.

In this work we achieve the statistics of defects among consecutive frames, which may lead to some properties, and to optimize the visualization. For my personal part, this tp4 is more learning than working. I've learned the conception of skyrmion lattice and its phase transition process, also operations in matlab along with image analyzing and processing. Out of the framework of skyrmions, multi-object identification and tracking are still very useful.

6. Reference

[1] Tony Skyrme, "A unified field theory of mesons and baryons". Nuclear Physics. 31: 556–569.

[2] Kirsten von Bergmann and André Kubetzka: Skyrmions: a twisted future, Physics World, Volume 30, (2017, Number 5)

[3] Motohiko Ezawa: Giant Skyrmions Stabilized by Dipole-Dipole Interactions in Thin Ferromagnetic Films, Phys.Rev.Lett.105:197202 (2010)

[4] S. Seki, S. Ishiwata, and Y. Tokura, Magnetoelectric nature of skyrmions in a chiral magnetic insulator Cu2OSeO3, PHYSICAL REVIEW B 86, 060403(R) (2012)

[5] Albert Fert, Vincent Cros, João Sampaio, Skyrmions on the track, Nature Nanotechnology volume8, pages152–156 (2013).

[6] Xichao Zhang, Motohiko Ezawa, Yan Zhou, Magnetic skyrmion logic gates: conversion, duplication and merging of skyrmions, Scientific Reports volume5, Article number: 9400 (2015)

[7] Ping Huang, Marco Cantoni, Arnaud Magrez, Fabrizio Carbone, Henrik M. Rønnow, Melting a skyrmion lattice topologically: through the hexatic phase to a skyrmion liquid, arXiv:1807.08352v1 [cond-mat.str-el] (22 Jul 2018)