A close up of a logo

Description automatically generated A close up of a sign

Description automatically generated 

IMAGE PROCESSING AND DUST TRAJECTORY TRACKING IN TOKOMAKS

**By**

Arpan KHANDELWAL

**Tutor for Internship**

Didier VEZINET

**Tutor for IMT Atlantique**

Amanda PORTA

**Enterprise**

CEA Cadarache/IRFM

# **ABSTRACT**

Atomic Energy Commission is a French Government Research organization for research about energy, mainly focused on nuclear power (both fusion and fission). Tokamaks are an advanced kind of experimental fusion reactors. They are toroidal vacuum vessels containing a very hot (~10-100 M°C) plasma of hydrogen or its isotopes confined by means of a magnetic field.

In Tokomaks, dust particles are undesirables. An impact by runaways on first wall of the reactor vessel and cause huge energy flux to be directed to the walls of the vessel. This causes erosion of wall components, introducing dust into the tokomak leading to core plasma contamination. Dust can contain toxic elements such as Beryllium and can also be radioactive.This internship aims to develop a code which can perform image processing and dust tracking on video of plasma shots from various tokomaks, like performing a synthetic diagnostic. The code will be open source, will be optimized for tokomaks using a single camera for observation and it will try to reconstruct 3D trajectories of the dust particle**.**

# INDEX

1. Introduction 04
2. CEA Cadarache 05
3. Institut de Recherche sur la Fusion Par Confinement Magnétique 06
4. Context of Internship 07
5. My Integration into CEA and my Conditions of Work 07
6. The Tokomak WEST 08
7. Structure of Code 10
8. Pre-processing 10
9. Cluster Detection 12
10. Trajectory Tracking 16
11. Reliability
12. Appendix
13. Bibliography

# **INTRODUCTION**

I joined IMT Atlantique (Ex Ecole des mines de Nantes), Nantes, France for pursuing my studies in Nuclear Engineering in July 2018. I am part of a dual degree program between University of Petroleum and Energy studies, India and IMT Atlantique, France.

I was interested in Nuclear Energy, but I was not so sure about which aspects of nuclear energy I was interested in the most. Hence, I took up both fusion and fission studies during my masters. I was tutored in Plasma Physics and Tokamak Engineering by experts from Institute for Plasma Research (IPR), India which is the Indian Fusion Research Institute and is a part of ITER (The International Thermonuclear Experimental Reactor). ITER is an international nuclear fusion research and engineering megaproject, which will be the world’s largest magnetic confinement plasma physics experiment. My choice of internship also reflected my desire exploring the different aspects of nuclear energy. I have done my M1 internship from Nucleonix Systems Private Ltd., which is an Indian company dealing with the manufacturing of Radiation detectors. I spent my summer at IPR studying on industrial and medical usage of plasmas. These experiences helped me in understanding that my interests lie in fusion and hence I undertook this project as my M2 internship.

I worked on WEST, the French tokamak during my internship from 25th February to 30th August. A tokomak is a magnetic containment toroidal chamber intended to control plasma to study the possibility of energy production by means of nuclear fusion. My work is a part of a larger and more ambitious project which goes by the name TOFU1 (Tomography for Fusion). Tofu was developed by my tutor Didier VEZINET when he was undertaking his post graduate study at IPP, Germany. The subject of my internship was to perform image processing on films shot inside the tokamak and analyse them to recognize dust events and track them as they move around the tokamak. I wrote my entire code in python, utilizing already available open source packages, OpenCV, NumPy, SciPy etc to name a few. A list of all the resources used can be found in the appendix. I have had daily interactions with my tutor regarding the direction of the development of the code. He provided me with inputs to make my code more readable, efficient and faster.

Fusion is an extremely complex problem, one that has plagued humanity for decades. We have realized that it is too big of a problem to be solved a one single organization or entity governed by an individual company. This has resulted in the formation of ITER with 35 participating nations so that we can achieved our goal of harnessing the power of the sun, providing mankind with clean, safe and almost unlimited source of power. This can be achieved by sharing all our knowledge and resources together.

When I approached Didier VEZINET, he pointed out a very fundamental problem that the fusion community is faced with. When a software or a code is developed by a single lab, due to licensing issues the code is not distributed to other labs. This results in resources being spent in different lab to develop the same kind of code for achieving the same results. Instead if we develop an open source code which can be accessed by anybody at any time, modified according to their needs, improved via community feedback, it results in a code that is very robust efficient and economical. The best example that I can think of which illustrates this philosophy is Linux. Hence, we decided to develop DUMPRO, as an open source code. In this report I will try and present clearly the context of the internship in the organisation and the fusion community. I will try and present the philosophy of developing an open-source code and why it is becoming more and more important in the scientific community.

# **CEA Cadarache**

### Cadarache is one of the 9 research centres of the Commissariat à l’Energie Atomique et aux Energies Alternatives (​CEA). It is situated in the town of St Paul Lez Durance (Bouches-du-Rhone). It was created on 14th of October 1959 and is located about 40 kms north of Aix-en-Provence. CEA Cadarache is one of the most important research and technology development centre for energy in Europe. The activities of CEA are divided into several R & D platforms, mainly for nuclear energy (Fusion and Fission). There is research also into other forms of energy along with plant ecophysiology and microbiology.

### 

### Fig 1: Entry of CEA Cadarache

In support of these R & D activities, the Cadarache centre has a service platform bringing together the necessary resources:

* the management of nuclear material, waste and discharges from nuclear facilities and general means to ensure the monitoring of facilities and the environment, security.
* The proper functioning of research facilities (water treatment networks, water, electricity).

Several collaborations have been developed with regional companies, AMU - Aix-Marseille Université and research institutes [Institut français de recherche et exploitation de la mer (Ifremer)], the centre of research and geosciences education the environment [Cerege]) within the framework of scientific collaborations, teaching and the creation of mixed laboratories.

Cadarache employs more than 5000 people along with 2100 employees. Approximately 90 PhD students are hosted by the research teams at the Cadarache centre within the framework of conventions established with universities.

There are also numerous exchanges and collaborations within the framework of European and international research programs for nuclear energy (fusion and fission). Between 150 and 200 researchers from twenty different nationalities work each year on the Cadarache centre in the framework of European or international scientific collaborations.

# INSTITUT DE RECHERCHE SUR LA FUSION PAR CONFINEMENT MAGNÉTIQUE

IRFM is one of the 15 institutes that make up the fundamental research division in CEA (Direction de la Recherche Fondamentale). For almost 60 years, its responsibility has been to carry out research on thermonuclear magnetically confined fusion at the CEA in association with the Euratom Fusion Programme. Since the beginning of the Tore Supra programme in the late 80s, it has been located at the CEA Research Centre of Cadarache in the department of the Bouches-du-Rhône. To fulfil its missions, IRFM gathers three departments (and within them, groups), with various objectives expanding from engineering to physics to platform operation.

The IRFM activities are structured around three main areas:

- Contribute to the implementation of the ITER project and those of the “Broader Approach”,

- Prepare the scientific operation of ITER, through control and experimentation activities, and through theory and modelling,

- Establish the basis for future fusion reactor.

Those activities are closely linked to a special effort in education for the new physicist and engineers in fusion sciences. The IRFM is equipped with several R&D platforms, the best known is the Tore Supra tokamak which is becoming WEST (W – tungsten Environment Steady-state Tokamak) to test the ITER divertor.

The French Alternative Energies and Atomic Energy Commission (CEA) is a key player in research, development and innovation in four main areas:

* defence and security,
* low carbon energies (nuclear and renewable energies),
* technological research for industry,
* fundamental research in the physical sciences and life sciences.

Drawing on its widely acknowledged expertise, the CEA actively participates in collaborative projects with many academic and industrial partners. The CEA is established in nine centres spread throughout France. It works in partnership with many other research bodies, local authorities and universities. Within this context, the CEA is a stakeholder in a series of national alliances set up to coordinate French research in energy (ANCRE), life sciences and health (AVIESAN), digital science and technology (ALLISTENE), environmental sciences (AllEnvi) and human and social sciences (ATHENA)

# **Context of the Internship**

Plasma diagnostics are a collection of methods, instruments and experimental techniques used to measure properties of a plasma, such as plasma components’ density, distribution function over energy, spatial profiles and dynamics etc. ToFu is a synthetic diagnostic tool which enables a user to look at the film of a plasma shot inside a tokomak and then derive meaningful hypothesis or observations from it and use it as a base to improve plasma performance for the future or design experiments to study a particular problem.

One of the more complicated problems plaguing the plasma community is the problem of dust inside a tokamak. Dust happens as a result of impact of runaway electrons on the first wall of the tokamak. This results in material from the wall introduced into the plasma. This leads to the plasma core being contaminated as the elements introduced into the plasma (Iron, Tungsten, Beryllium) have a significantly higher atomic number compared to the fuel elements (Hydrogen, Deuterium, Tritium). This causes energy losses in form of radiative loss, destabilizes the plasma causing disruptions.

For sustained power generation through fusion, these energy losses will reduce the net output and hence increase the cost of producing energy. Tokamaks are extremely complicated and costly machines. To replace any major component takes a lot of time and this will require complete shutdown of the machine. The cost of material and manufacturing of the first wall of a tokamak is very high. If the erosion of the first wall is extreme then this posses a problem as the reactor will have to be shut down for most of the time. The cost of the reactor is also included in the cost of power generation. Hence the reactor components need to work at least for the lifetime of the reactor which can be anywhere between 40-60 years. Any additional replacements will increase the operation costs. Since dust is particles from the first wall, understanding its behaviour will help us in reducing the cost and making fusion power more feasible.

DUMPRO aims to study these dust particles inside the plasma by performing image processing on the tokamak shot films, hence studying their movement, their size, their distribution etc, providing scientist the base to design experiments to investigate the problem of dust. DUMPRO is designed to be integrated with ToFu and can run in any system such as Linux or Windows along with providing the user the flexibility to modify the code according to their personal needs.

# **My integration into CEA and My Conditions of Work**

I worked in building number 507. I found that the institute is quite independent of the rest of the centre, having its own administration. I have been brought into contact with many people working in the field of magnetic fusion, Didier Vezinet (my supervisor), Nicolas Fedorczak, Jim Gunn to name a few. M. Fedorczak provided me with multiple fast camera films from the tokamaks to work with.

I have had regular sessions with my supervisor Didier, to discuss the direction in which we would like the project to move. He helped me in optimising the code, taught me the usage of some of the open source packages required in my code. He has provided me a crash course on both ToFu and Git2.My tutor complemented my lack of programming experience with his expertise and I learned many ways which programmers use to make their code more readable, understandable, clean etc.

My tutor helped me get integrated well. On my very first day he took in charge of showing me around the research centre, introduced me to people who are working in fusion. He also game me the opportunity to visit ITER during a one-week coding camp. By the time I was starting my internship it was already late for registration, but he made a special request on my behalf to the ITER administration and I could participate in the coding camp. Didier also got me into a fusion summer school lecture on Plasma wall interactions where experts from ITER came and gave lectures. I was invited by him to attend the Scientific and Technical meeting (RST) which takes place each week and during which members of the group present their work and generate a debate on directions to follow. Through these I have been able to understand the administrative working aspects of CEA.

My tutor actively has helped me in pursing my goal of PhD by introducing me to members of the research centre who have proposed a topic to providing recommendation whenever I required them in my application. I have decided to continue the development of the code outside my internship as I am enjoying the working, I am doing. It is providing me the right amount of challenge with an opportunity to learn new topics.

My working hours was not binding, my time for work was 0730 to 1630 hours with lunch being from 1100 to 1400 hours. The research centre remained open till 1930 hours and I would sometimes stay back till 1830 because I enjoyed this relationship of trust with the centre. This atmosphere allowed me to keep working to the best of my abilities and has provided me the incentive to look for future employment at CEA.

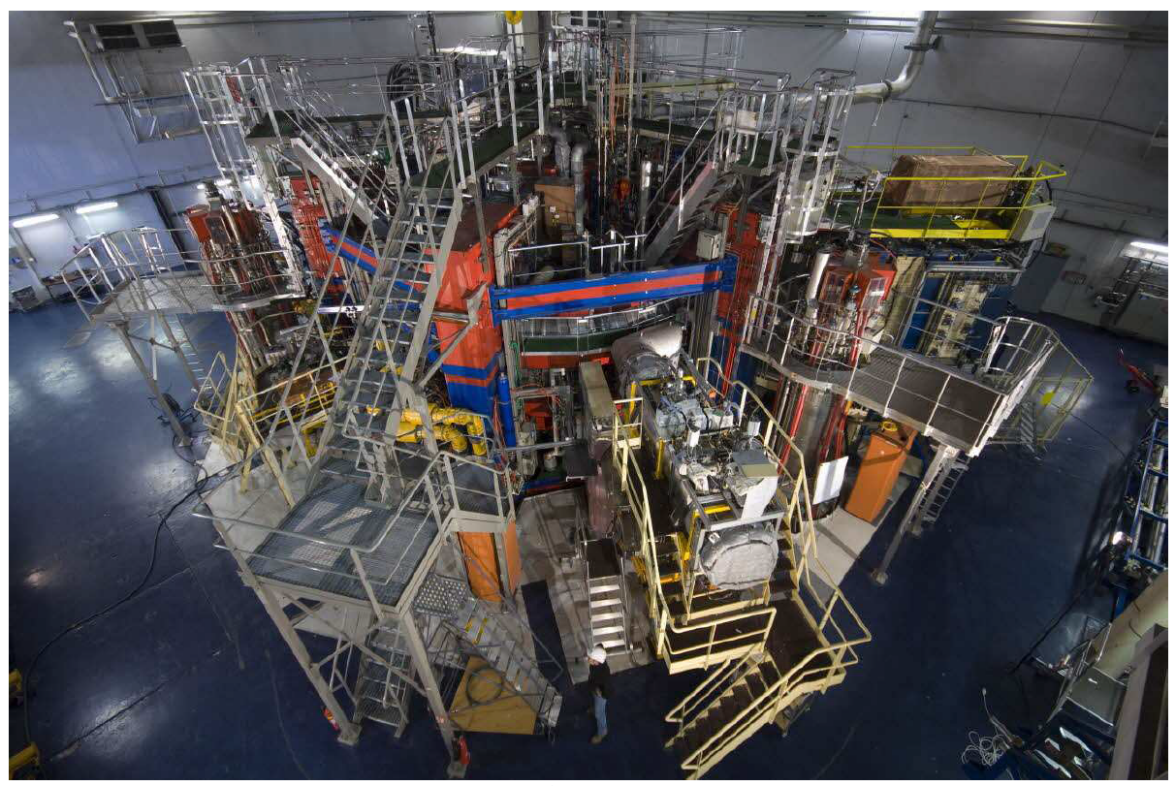
The CEA thus has a complex but ingenious organization that allows its employees to work in a pleasant way without feeling the weight of a huge company on their shoulders. This aspect made my internship very enjoyable, in addition to the exciting work that I did and that I'm going now detail.

# **The tokomak WEST**

A tokamak is a magnetic containment toroidal chamber intended to control a plasma to study the possibility of energy production by nuclear fusion. Invented at the beginning of the 1950s by the Russians Igor Tamm and Andrei Sakharov, the term tokamak comes from Russian "Тороидальная камера с магнитными катушками" (toroidalnaya kamera s magnitnymi katushkami: in French, toroidal chamber with magnetic coils).

WEST stands for tungsten(W) Environment in steady state tokamak is the transformation of the tore supra tokamak from a limiter to a divertor configuration. WEST involves testing a tungsten divertor for ITER. The divertor receives greater part of the heat fluxes and particles coming from the central plasma during experiments.

Tore Supra is a unique establishment due to its superconducting magnet and the fact that it is one of the few to be equipped with actively cooled plasma-facing components. To test the tungsten divertor for ITER the magnetic configuration of Tore Supra was modified from a circular form to assume one of a distorted-lens so that it can obtain plasmas with characteristics similar to those of ITER. WEST tokomak allows researchers to carry out a pertinent scientific programme focusing on the preparation of the experiments of ITER.



### Fig 1: WEST Tokamak. The image shows the entire machine with its various diagnostics, support structure, heating systems etc.

### The following above picture shows the tokamak and the one below shows the inside of the vessel and what we call ‘the first wall’ . The one on the left with a high curvature is called the inner wall and the one on the right is called outer wall. The flat surface running on top is the upper divertor and at the bottom we have the lower divertor and the baffle. The divertors are used to redirect impurities away from the plasma and hence have to be made with high heat conduction material, since the divertor receives an extremely high heat flux when the plasma impinges on it.

A picture containing indoor, object

Description automatically generated

### Fig 3: Inside WEST Tokamak. This is the first wall of the tokamak and the plasma is confined in this chamber

# **Structure of the Code**

To perform dust detection on movies, I must separate them from the rest of the movies. Since dust particles are light emitting particles and they are brighter than other objects present such as the plasma, the vacuum vessel wall. After their separation, the dust particles on all the frames need to be associated with each other. After the association we can track their trajectories and thus gather information such as dust velocity, distribution etc.

Hence this task can be divided into four parts:

1. Pre-processing: This involves removing all unnecessary information to reveal the dust particles. The associated steps involve grayscale conversion, denoising, smoothing, binary conversion. After this step only dust particles are left in the images.
2. Dust detection: This step involves detecting the dust particles present in the images and getting their centres, angular orientation, pixel area. These dust particles are then highlighted on the images.
3. Trajectory tracking: Each dust cluster is assigned a unique identifying number. This number is used to access the cluster while deciding on its parent and children. Using a unique id also allows us to preserve the time information for each cluster
4. Plotting trajectories: This last step plots all the trajectories, the dust distribution

The code is divided into 3 main modules which are further divided into classes and subroutines. Modularization helps with debugging the code and with improvements as specific parts can be worked on and improved without having to completely change the entire code.

1. Video\_comp
2. Image\_comp
3. Plotting

The three modules are called upon as required, by the core of the code which is stored as \_core.py. Video\_comp handles all operations related to video files. Image\_comp handles all operations related to image files and Plotting handles all operations for displaying trajectories and distributions. The following section will follow the logic and method of performing image processing and then we will look at how the packages implement it.

# **Pre-Processing:**

Pre-processing involves all the steps that lead to the preparing the videos or images for performing dust detection. To single out the dust, we need to remove the unnecessary information from the images step by step. Dust glows inside plasma due to constant collision with the ions and electrons present in the plasma. Hence colour information is not that important for us. All we need are the pixels with the highest intensities. Hence the first step involves conversion of film to grayscale. Then noise removal removes the unnecessary disturbances present in the film as a result of turbulence in plasma or radiation etc. After denoising the video is ready for background removal. After that we are left with only with setting up a threshold and then run the film through a contour detection algorithm.

* All films are made up of images. The images that we seen in electronic devices are not one image. They are made up of three images. These images are made of red, Blue, Green respectively. To convert an image to grayscale, we must take each image, add them to each other and then average them. The way this averaging is done as follows.

All images are made of small dots known as pixels. Each pixel has a value ranging from 0 to 255. To convert to grayscale, we take the first pixel from the 3 images and average them. This now becomes the first pixel of our grayscale image. Then we take the second pixel from the 3 images and average and so on. When we are done with all the pixels, we are left with an image made of the intensity values and we have our grayscale image

A picture containing screenshot

Description automatically generated

### Fig 4: Construction of a colour image

* Denoising involves taking and average of the values of surrounding pixels and then replacing the pixel in the middle with the average value. This removes information from the image and hence the number of pixels to consider for calculating the average is an important parameter. Noise is considered as a random variable with zero mean.

Let us consider a noisy pixel where p = p0 + n

n is the noise in the pixel while p0 is the true value.

To remove this noise, we take a large number of same types of pixel and compute their average. We should get the average value of the pixel as p = p0 since noise has zero mean. The denoising method used is a non-local denoising method. We take a patch of pixel and compare its vale to a similar patch of pixel, instead comparing it to the next image. This is done as there are lot of dust particles in some of the images and denoising might remove our dust particles.

* Background removal involves distinguishing the moving object of interest from the static background. There are two distinct ways of doing this.

1. Frame by Frame: If the dust particles are moving fast enough or if the camera is recording fast enough, we can use this method to separate the background. From each current frame the previous frame is subtracted to reveal the moving objects in the frame.
2. Background subtractor MOG: This is a Gaussian Mixture based background foreground segmentation method from OpenCV. It models each background pixel as a mixture of k gaussian distribution (k = 3 to 5). It has 4 different parameters that can be modified for better background removal.

* Thresholding is the next step which removes lighter features left out after the background removal. Thus, only the features which are the brightest such as dust or plasma features near the divertor region where it has reached saturation in terms of intensity, in the camera, are left behind in the image. After grayscale conversion, each pixel in a frame is made up of values ranging from 0 to 255. 0 is completely black while 255 is completely white. In thresholding operation, we set all values below a certain value to 0 and above the certain value to 255.

The preprocessing is complete after the following steps and now we are ready for cluster detection.

# **Cluster Detection:**

For detecting clusters, the images or film is run through a cluster detection algorithm. We used OpenCV find contours algorithm. Contour are a group of continuous points joined by a curve along the boundary. The points have all the same intensity, which was ensured by the last step of preprocessing. So after detecting all the boundary points, we fit a minimum area rectangle to each of these closed loops.

OpenCV has its own method called “minarearect()” which fits the rectangle with the least area to a given contour. From this we can get the orientation of the cluster, its center, area (in terms of number of pixels, this area is not the actual area but the area projected by a cluster to the plane perpendicular to the axis of the camera). The center is calculated by finding the centroid of the rectangle. This is done using “cv2.moments()” which returns the moment of each contour. Minarearect() also provides the side of the rectangle which helps in calculating the orientation of the cluster.

### Figure 5(a) shows the original frame of a tokomak shot. Some of the duct particles have been highlighted with a blue circle. These are the particles that we wish to detect and track. But there is also a bright crescent shaped plasma feature near the inner wall of the tokamak. This is something we want to avoid as it is part of the background and it would only provide us false results. Upon performing grayscale, we get the figure as shown in fig 5(b). The image is a representation of the intensity of the light coming from the plasma. 5(c) shows after denoising. Although it is not visible given the size of the image, but there are some significant changes to some of the local parts of the image. 5(d) shows what is left on the image after background removal. In this frame the background removal as done using background separator MOG from OpenCV. Fig 6 shows how it will look if a frame by frame subtraction method was used.

**** 

### Fig 5 (a) Original Frame of a shot. (b) Grayscale converted video. (c) After denoising. (d) After background removal

### 

### Fig 6: Frame by frame subtraction background removal.

### As the highlighted region shows there is some very faint plasma feature still remaining in the image. This will be removed in the thresholding operation of DUMPRO. Even after all of these treatments, sometimes some background plasma still persists in the image. These are further isolated by grouping clusters. Grouping is mainly done based on pixel size of each cluster. Figure 8 shows some of these plasma features on the final image. In 8(a) there was no dust in the actual frame as shown in 7(a) while in 8(b) there was a runaway impact which resulted in a saturation event for the camera as shown in 7(b) and hence it gets detected along with the impact hotspot

### Cluster groups are divided into two different categories: (i) small clusters (ii) big clusters. Based on the size difference between the largest and the smallest, there might also be a third group called “medium clusters”. When associating relation between clusters from two different frames, this grouping allows us to decide whether a cluster can have more than one child.

### After performing detection and extracting positional, angular etc information of the clusters, we are now ready to assign relation to clusters of different frames

### 

### Fig 7(a): Original frame for KSTAR\_1. (b) Original frame for KSTAR\_2.

### 

### Fig 8(a): foreground for KSTAR\_1. (b) Foreground frame for KSTAR\_2

# **Trajectory Tracking**

### For assigning relation to clusters we need to identify which clusters in two adjacent frames are a part of the same trajectory. Normally when we track a object we track some of its unique feature. For example, to track a car in a video, we track its color the shape of the car as these features are stationary features, which means that they do not change with time. But for dust we don’t have any such features. Dust particles inside plasma evolve with time. Their shape, size changes. A large dust particle can break into many smaller ones upon impact with the vessel wall. Hence, we use statistics and probability to find a dust particle. Every dust detected in a frame have the following properties:

### Size in terms of number of pixels

### Orientation which is measured by the angle between the longest side of the minimum area rectangle with the horizontal

### Position which is given by x and y co-ordinates which corresponds to its position on the frame.

### 

Frame t-1

Frame t-1

Frame t-1

Frame t

Frame t-1

### Fig 9: Picture of a cluster from frame t superimposed on frame t-1

### Using the 3 properties we can start assigning probabilities to all possible combination of clusters between two adjacent frames and the combinations with the highest probability is selected.

### For a dust at frame t, we are taking a look at all the dust present in frame t-1,

### The first parameter is the distance between the two clusters. Di,j represents the distance between the two clusters.

### 

Di,j

t-1

t

### The second parameter considers the history of the potential parent finds out where the probable position on the cluster in the current frame can be. This tracks the three most recent changes in x and y co-ordinates with time and uses curve fitting to find the curve that best describes their position. Using the curve, it can be estimated where the current position of the cluster can be. Then the distance between the probable position and the actual cluster position is calculated.

### 

D’i,j

t’

t-2

t-1

t

### D’I,j represents the distance between the probable position and the actual position. The advantage of using curve fitting is that it takes care of the direction of travel of the parent as well as the distance travelled.

### The third criteria is the history of evolution of the size of the cluster group. The change is size of the last 3 clusters are considered and a curve fitting is done to get a function which can estimate the change of size with time. Then the size of the current cluster and the estimated size is considered. It is represented by parameter t’I,J.

### Taking into account the history of size change can provide false data when the cluster is breaking up into smaller pieces. To counter act this the size difference between the current cluster in frame t and the size difference of cluster in frame t-1 is taken. This is represented by parameter t’I,J.

### Now that we have all the parameters that we need, we can assign a probability to the cluster in frame t-1

### The probability that the cluster being considered in frame t-1 is the parent of the cluster being considered in frame t is given by

## **Reliability**

### Reliability is dictated two prime aspects:

### Runtime

### Accuracy of data

# Runtime

### The runtime of the code depends on the number of frames it is processing, and the number of clusters present in those frames. The low run time is due to the usage of highly optimized subroutines available form opencv and scientific python package.

### All the frames are of size 610 x 640 pixels.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number of frames | Total clusters present | Background subtraction method | Image smoothening | Time(seconds) |
| 50 | 47 | Frame by frame | YES | 19.45876 |
| 50 | 293 | MOG | YES | 21.04874 |
| 100 | 70 | MOG | YES | 11.82930 |
| 100 | 759 | MOG | YES | 33.23481 |
| 100 | 81 | Frame by frame | YES | 25.20203 |
| 100 | 416 | Frame by frame | NO | 23.96531 |

# Accuracy of Data

### This parameter tries to quantify the accuracy of the code. It considers the number of clusters detected by the code and the number of clusters present in the code. The following figures displays the data for more than 300 frames which have undergone all the four possible pathways. The four possible pathways are determined by two variables: The background subtraction method used and whether image smoothing has been performed or not. The actual number of clusters per frame has been shown in white, while the path that has data closest to the actual number is highlighted in red.

### 

### 

### A picture containing text Description automatically generated

### 

# **Appendix**

### 1ToFu

ToFu is an open source python library with the objective of providing the fusion community with an object-oriented, transparent and documented tool for designing tomography diagnostics, computing synthetic signal (direct problem) as well as tomographic inversions (inverse problem). It gives access to a full 3D description of the diagnostic geometry, thus reducing the impact of geometrical approximations on the direct and, most importantly, on the inverse problem.

Tofu was created by Didier Vezinet at IPP (Max Plank Institute for Plasma Physics) in Garching, Germany, through the years 2014-2016. It is continuously maintained, debugged and upgraded to this day.

It is generic in sense that it has been developed to be machine independent and modular, thus guaranteeing that it can be used for arbitrary geometries and with an arbitrary number of apertures for each detector.

Tofu is distributed under MIT license, thus allowing free use, transparency, reproducibility and incremental improvements that guarantee quality on the long-term.

Tofu is hosted on GitHub.

More information on Tofu can be found at https://tofuproject.github.io/tofu/index.html

2GitHub

Git is an open source version control system started by the created of Linux, Linus Trovalds. Git is similar to other version control systems such as Subversion, CVS and Mercurial to name a few. When a developer creates something, they modify their code several times, making constant changes, releasing several versions.

Version control systems keep these revisions straight, storing the modifications in a central repository. This allows developers to easily collaborate, as they can download a new version of the software, make changes, and upload the newest revision. Every developer can see these new changes, download them, and contribute. People who are not associated with the project can download it, use the code, make another project based on it (fork) or make improvements and modification to the existing project and ask the developer to add those changes to the subsequent release.

Git is a command line-based tool. GitHub is the place where the developers can store their projects and collaborate with likeminded people.

More information can be found at <https://github.com/>

3OpenCV

OpenCV is an open source computer vision library and machine learning software library. The motive behind the creation of OpenCV was to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial product. It is distributed under BSD-licenses.

This code consists of more that 2500 optimised algorithms that tackle face detection, object tracking, camera movement tracking, stitching of images and many more. It has an estimated 18 million downloads and more that 47000 strong community.

It has C++, Java, Python and MATLAB interfaces and supports Windows, Linux, Android and MacOS. OpenCV leans mostly towards real time vision applications and takes advantages of MMX and SSE instructions wherever available. A fullfeature CUDA and OpenCL interfaces are being actively developed. OpenCV is natively written in C++.

For more information go to <https://opencv.org/>

# **Bibliography**

Code usage tutorials and function descriptions

1. OpenCV : <https://opencv.org/>
2. NumPy: <https://www.numpy.org/>
3. SciPy : <https://www.scipy.org/>
4. MatPlotlib : <https://matplotlib.org/>
5. Anaconda : <https://www.anaconda.com/>