**Particle Track Reconstruction from VELOPIX data**

**Semester:** 2020-2021  
**Supervisors:** Daniel Hugo Cámpora Pérez and Kurt Driessens  
**Students:**

* Maxime Duchateau
* Siemen Geurts
* Tjerk Kuil
* Jorrit Milan Nattebrede
* Marius Sommerfeld
* Carlos Tello Ordóñez

**Background and motivation:**

At the Large Hadron Collider beauty (LHCb) experiment at CERN bundles of particles collide at velocities close to the speed of light. In this collision new particles are created and scattered in all directions, leaving traces (hits) in the detectors placed in their way. Detecting these particles and their behaviour is key to adding to and testing our understanding of physics. To perform this detection, the LHCb experiment’s detector is composed of a series of subdetectors as shown in figure 1. The Vertex Locator subdetector (VELO) is the first subdetector at this experiment, and the focus of our project. The key role of the VELO within the LHCb experiment is to locate the particle collisions and find the trajectories (tracks) of the particles created in the collisions, which then helps to understand and explain the observations made in the later detectors. In our project we will exploit our knowledge about expected patterns in the data (tracks form straight lines) to reconstruct the trajectories of the particles created in the collision.

There are many techniques to do this reconstruction, and the current approach is already very good. However, due to a recent update, the amount of data that has to be processed will increase and the data will be processed on a different computer set up. For this reason, any new techniques that can be fast or that do well at the new set up would be very useful. The goal of this project is to identify approaches with the potential to compete with or compliment the current method for finding the particle trajectories. To this end, we will investigate the following techniques: Clustering, Hopfield Network, and/or Template Matching.

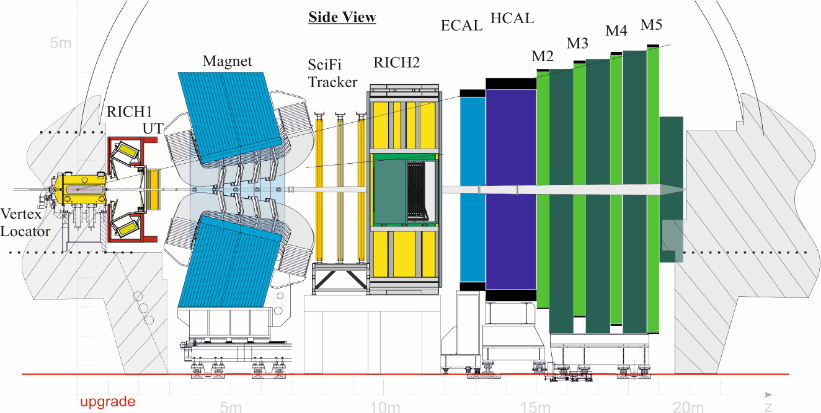


Fig 1: LHCb experiment

**The techniques explained:**

The VELO consists of several modules which are panels that the particle can hit without stopping but we can detect that hit. The set of hits that form a track can be thought of as a group or **cluster**. Clustering is a common technique in data science, and it means grouping hits together if they are near to each other based on some function or measurement. Pixel hits belonging to a track will share certain characteristics/features. We will investigate if it is possible to use such a characteristic to define a way to group together /cluster the hits that belong to the same track. Our second technique, **Template matching** also uses the fact that we are looking for lines, as it simply is comparing (matching) expected patterns (templates/lines) to the data. This requires some considerable fine tuning, as the number of possible templates is larger than the number of templates that could be computed fast enough. The third method we consider is the **Hopfield network /Boltzmann machine**: We define which pairs of hits could belong to the same track, creating a network of possible tracks. This will however render too many tracks, and many may not be possible. The Hopfield /boltzmann optimization method can be used to remove those connections between hits that do not lead to a larger track or if the hit has already been used.

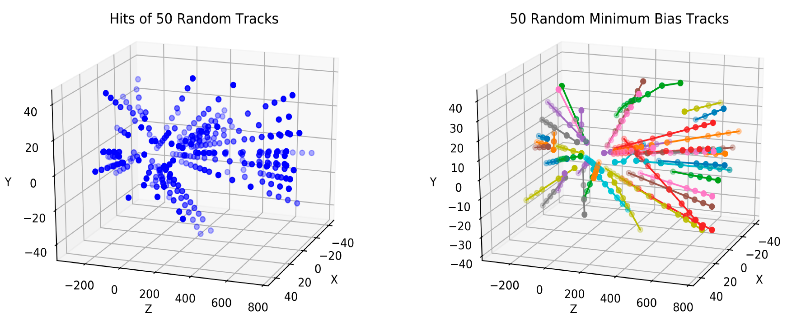


Fig 2: 3-dimensional image of 50 tracks

**Results thus far:**

For the Hopfield network / Boltzmann machine, we have an implementation working. It gives some strange connections, but in general it does quite well (See figure 3 and 4). In the next phase we will fine tune the model. For the template matching, we also have a simple implementation, which performs okay but not good enough to be competitive in the number of tracks it finds. We will work on a more sophisticated method. For clustering, we tried many representations, but did not find a suitable one. It did inspire us to investigate another method related to template matching: we sort the hits based on where they hit the panels in the detector, and then go over this sorted list to find the tracks. The benefit from this method is that it can be very quick, close to the theoretical minimum time required. We have spent some time on it, and the results are promising. This method will most likely be useful as a first step to find the easy tracks and remove these from the data. Later another technique can be applied to find harder tracks on the reduced data set.

We still have some work to do, but it seems that we can achieve our goal in making the case for and against our methods. We will create a report with our considerations & results, as well as provide the code to enable further research and replication.

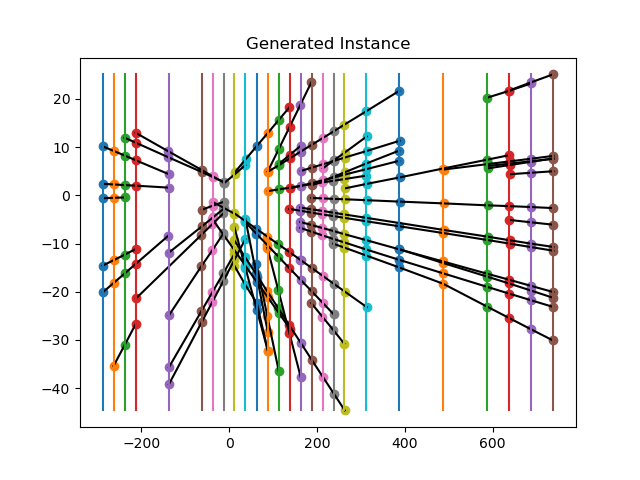


Fig 3: The real particle tracks

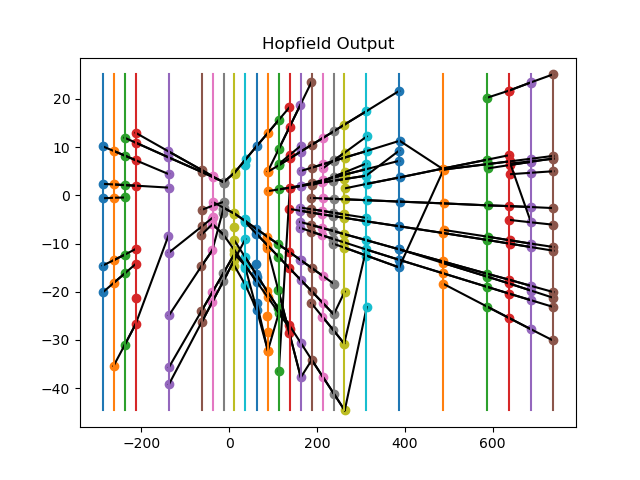


Fig 4: What the Hopfield / Boltzmann method returns.