UNIVERSIDADE FEDERAL DE JUIZ DE FORA

PPEE
Programa de Pós-Graduação em Engenharia Elétrica UFJF

Directional-iDBSCAN

a proposal to CYGNO

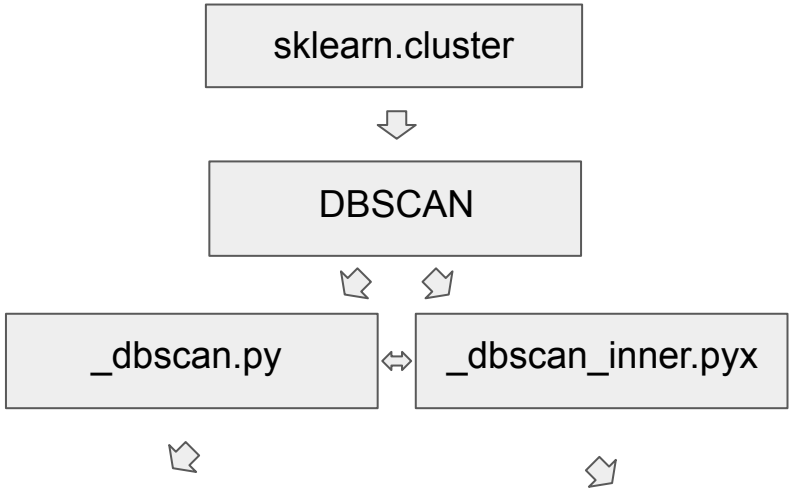
Igor Pains

Igor Abritta and Rafael A Nobrega

Last presentation

- The iDDBSCAN was updated with a new version of the RANSAC that is able to maintain a constant width in the fits.
- Some details in the iDDBSCAN needed to be adjusted in order to implement into the reconstruction algorithm.
    - These adjusts were made mostly to improve the time performance of the algorithm.

iDBSCAN - sklearn



Pure python algorithm responsible to extract information from the parameters, such as EPS and min_samples.

Algorithm written in cython responsible to do the iterative part of the DBSCAN with the information provided by the other part.

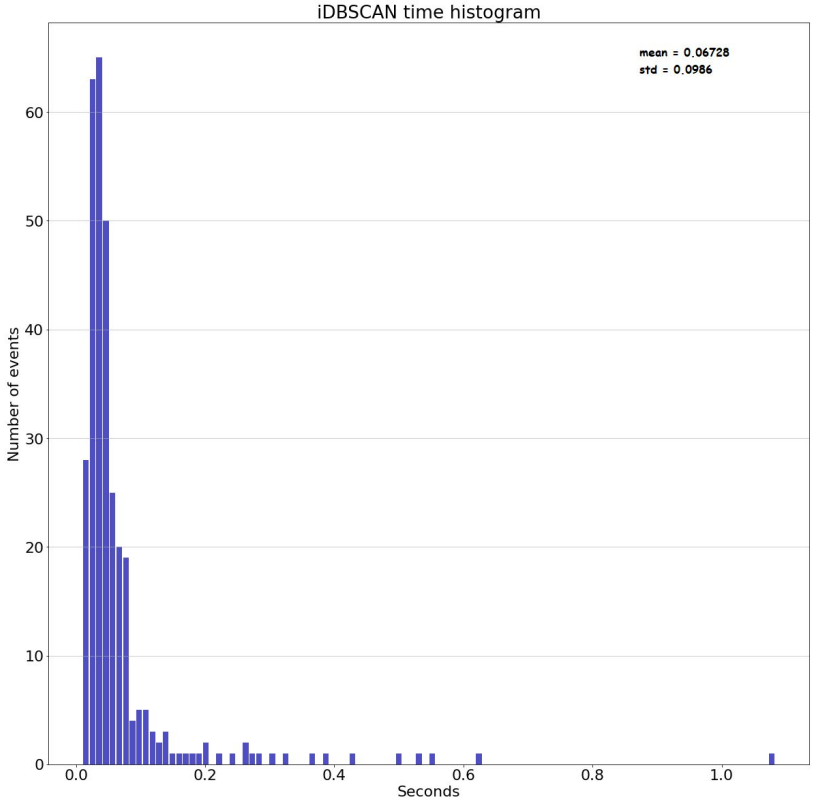
Python - Cython

- The python is one of the most used programming language in the world, due to the facility of implementing and being richly outfitted. But it is not the fastest one.
- The cython is a language that combines both python and C++, aiming the speed performance of the last keeping the facility of the first.

iDDBSCAN

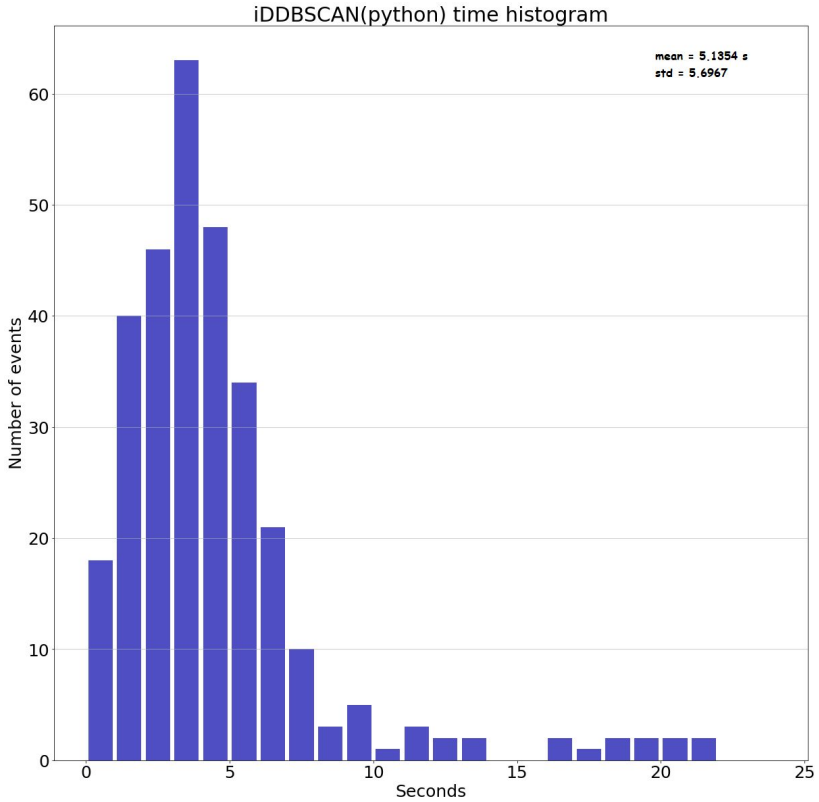
- The iDDBSCAN was written fully in python, due to its facility to develop and improve, but it was not as fast as the iDBSCAN.
- In order to speed it up, the algorithm was altered and part of it is now written in cython, just like the iDBSCAN.

Time efficiency - iDBSCAN



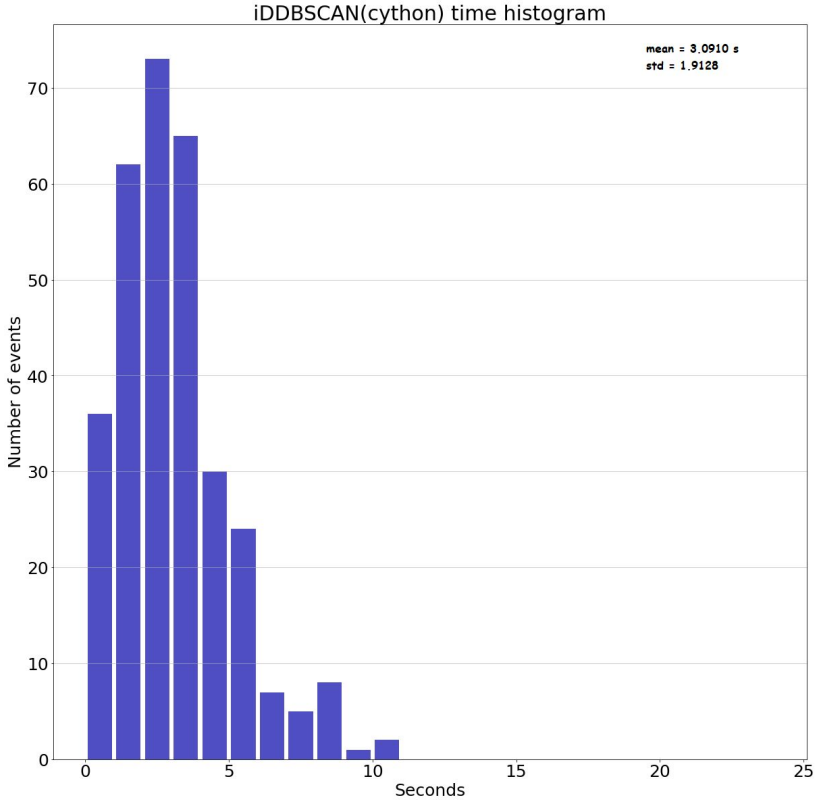
- The iDBSCAN (one iteration) needed less than one second to do the clusterization in most part of the events.
- It took less than 0.1 seconds for most part of the events.
- The slowest event needed 1.083 seconds

Time efficiency - iDDBSCAN (python)



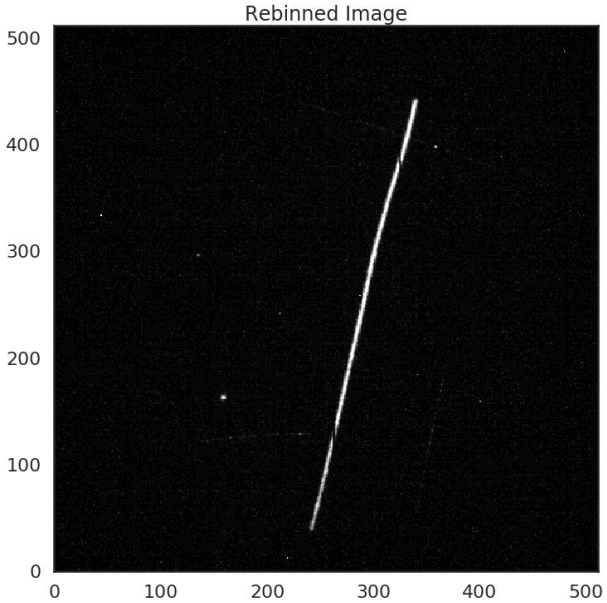
- The iDDBSCAN is slower due to the need to analyze the points of each cluster to perform the fit.
- It took less than 10 seconds for most part of the events.
- The slowest event needed 50.68 seconds

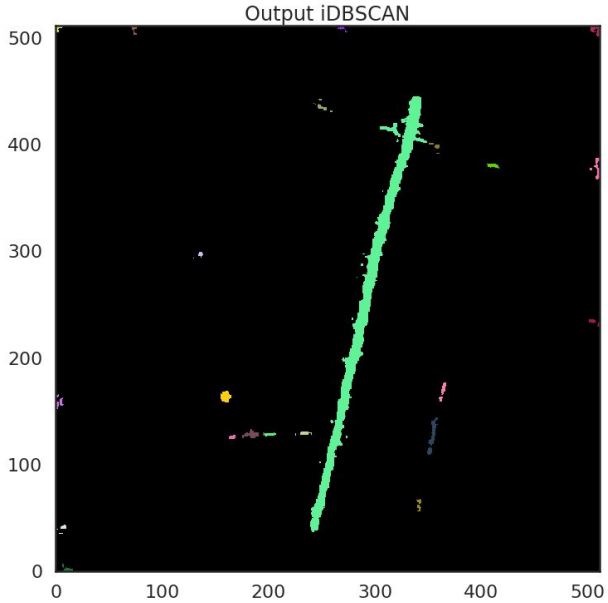
Time efficiency - iDDBSCAN (cython)

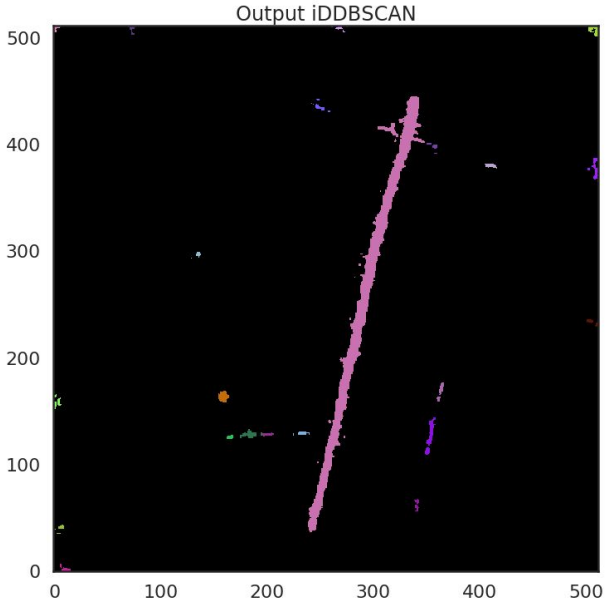


- The implementation with cython is quite faster than the one with python, but it still needs more time to do the clusterization than the iDBSCAN.
- It took less than 6 seconds for most part of the events.
- The slowest event needed 10.41 seconds

Run 2320 - Event 135

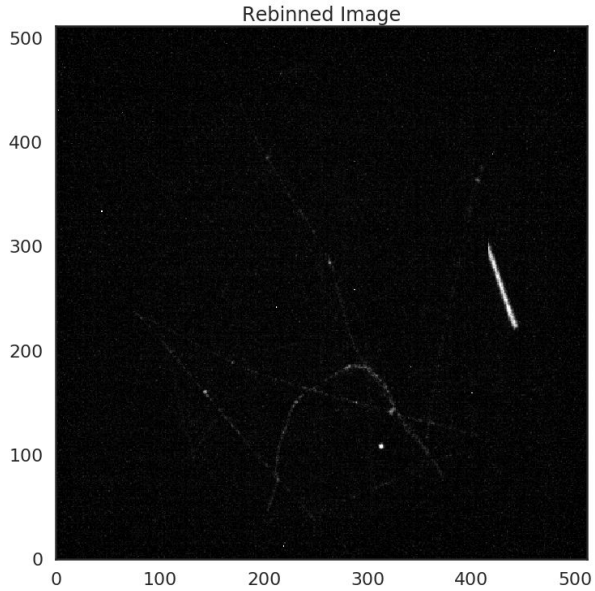


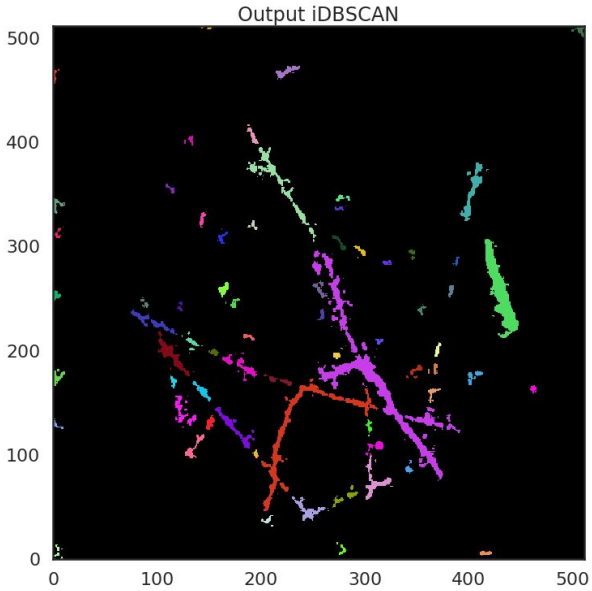


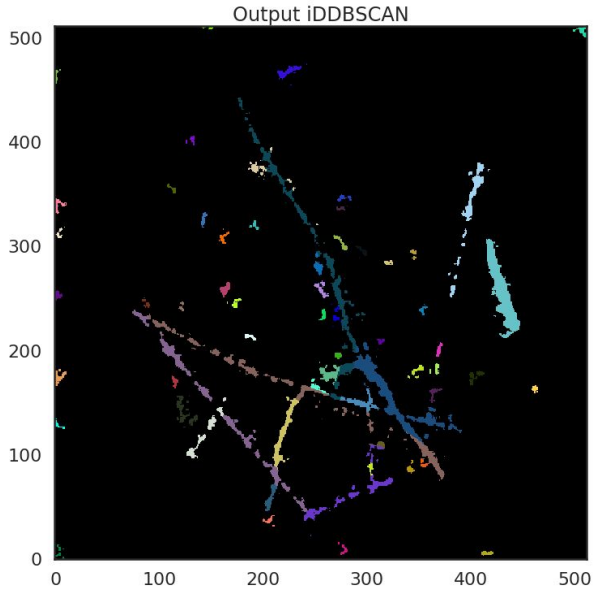


The slowest event with iDBSCAN and iDDBSCAN (python)

Run 2320 - Event 52







The slowest event with iDDBSCAN (cython)

Conclusions

- Converting part of the algorithm to cython was able to speed it up, making the implementation in reconstruction more viable.
  - from 5.1 seconds to 3.1 seconds
- The *maximum_time parameter* was not used in other to measure the time needed to process each image, but it can be used to stop the iDDBSCAN algorithm if it takes more than a certain time.