

# Vantage Point Trees

## Introduction

In this document I am going to compare two implementations of the vantage point trees (*VPT*), one written in *Python* [1] and the other one in *C++* [2]. With this, we can compare which of the programming languages perform better with this kind of data structure.

## Experiment

In order to compare both implementations, I designed the following experiment. First, in *C++* I generated randomly a vector of 3-dimensional points of size 10000, with real values between 100 and -100. Then, I created the *VPT* with the vector. Finally, I searched 100 neighbors of a random point, 100 times. I saved the vector of points, the searched points, the results, and the execution times in a file, to reproduce the same experiment with the other implementation and compare the results.

With the *Python* implementation I did the same, but instead of generating the data, I read from the saved files.

## Results

As a result of the experiment, I saw that the *C++* implementation performs much better than the *Python* one, in both time and found neighbors.

In the case of time, *Python* lasts 21.930 s to construct the tree, 0.0163 s to search 100 neighbors of a point on average, and 0.0234 s at most. In the other hand, *C++* lasts 2.831 s to construct the tree, 0.0004 s to search 100 neighbors of a point on average, and 0.0027 s at most.

In the other case, the *C++* implementation always returns the 100 closest neighbors to the searched point, while in some cases the *Python* implementation returns other ones or even less than 100 neighbors. Moreover, the *C++* implementation has more decimal precision. As a final comment, the *Python* implementation returned the correct neighbor points 97 of the 100 executions.

## References

- [1] Rickard Sjögren. vptree. <https://pypi.org/project/vptree/>, 2017.
- [2] (GitHub) pderkowski. vptree. <https://github.com/pderkowski/vptree>, 2017.