Report

Lab of ICP without acceleration and with ANN

Author: Zixin ZOU

1. Introduction

This lab is aim to realize the 3D point cloud registration between two dense sets of points. Through Iterative Closest Points (ICP) Algorithm, we can estimate transformation parameters. The points in {Set 1} using estimated parameters to close {Set 2}.

In the ICP algorithm, we need to search the nearest neighbor of each point. If without acceleration, it calculates the distance for each point in {Set 1} and {Set 2}. Time complexity is O(nm), n = number of {Set 1}, m = number of {Set 2}. So we have to find ways to reduce the running time.

ANN is a library to exact and approximate nearest neighbor searching. We can use ANN to build tree data structure. After that, nearest neighbor searching is will be easier. If with ANN, time complexity is O(N).

This lab shows two version codes. One is the basic version, one is the ANN acceleration version. And it shows their error, running time and average CPU. Through these, we can compare their complexity.

1. Version 1 : The basic version

General Seuil = 500;

1. **Sample 1 : Foot**
2. Error:

60.680233 29.576410 18.130400 11.958056 8.327651 6.344907 5.200370 4.471137 4.001554 3.688322 3.437684 3.112204 2.315095 0.391426 0.000000

Average : 10.775802

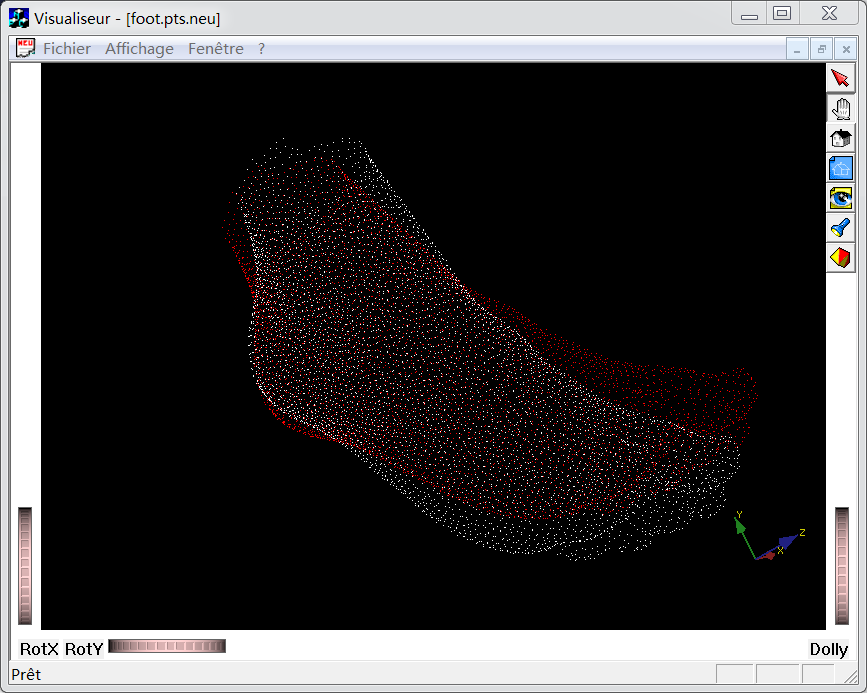
1. Running Time:

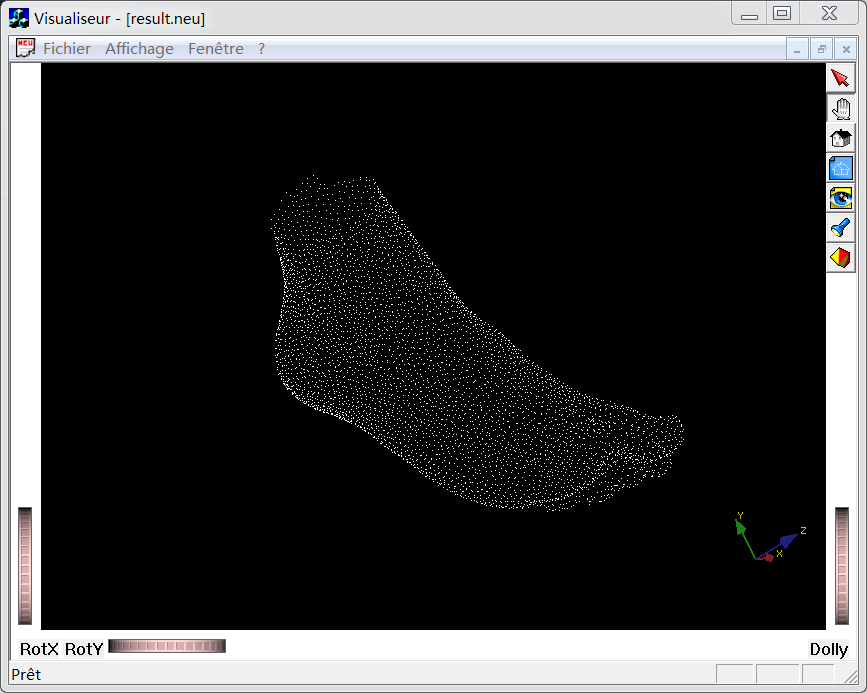
5146 5136 5047 4462 4930 4919 5328 5000 5329 4534 2089 1760 1721 1751 1782

Total : 58934ms

1. Results visualization:

Before:





1. **Sample 2 : Bunny**
2. Error:

0.000071

1. Running Time:

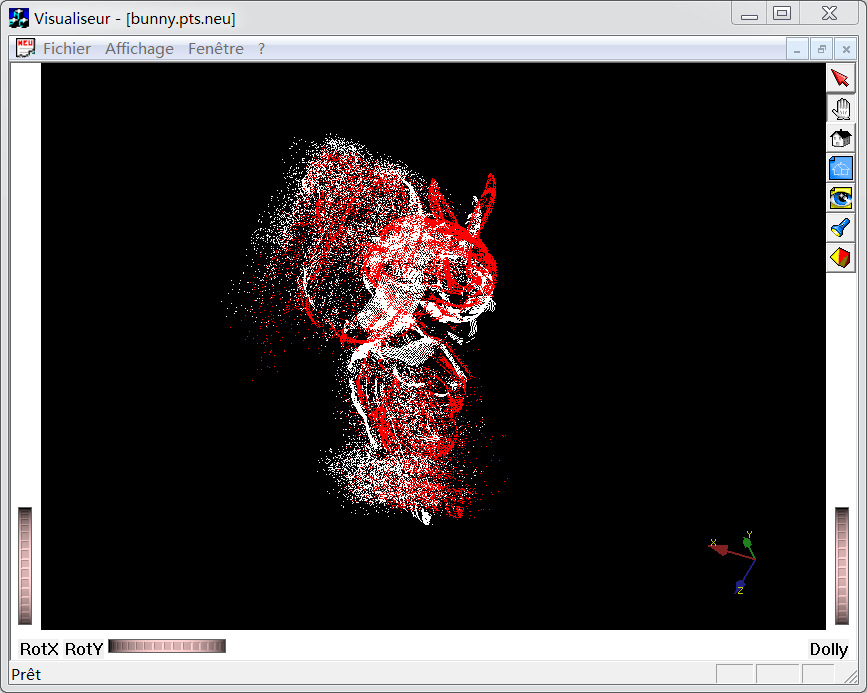
98646ms

1. CPU:

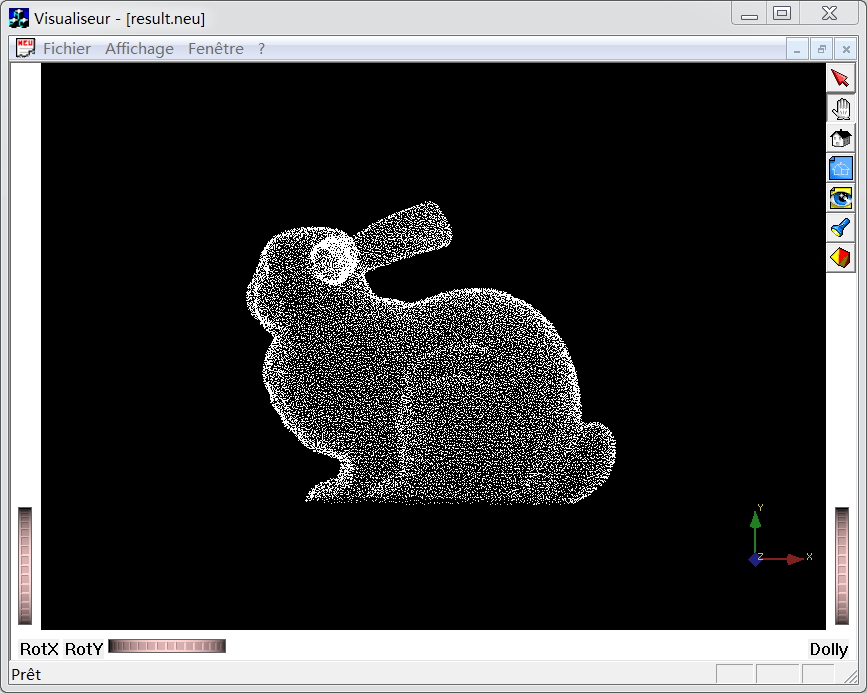
24.06

1. Results visualization:

Before:



After:



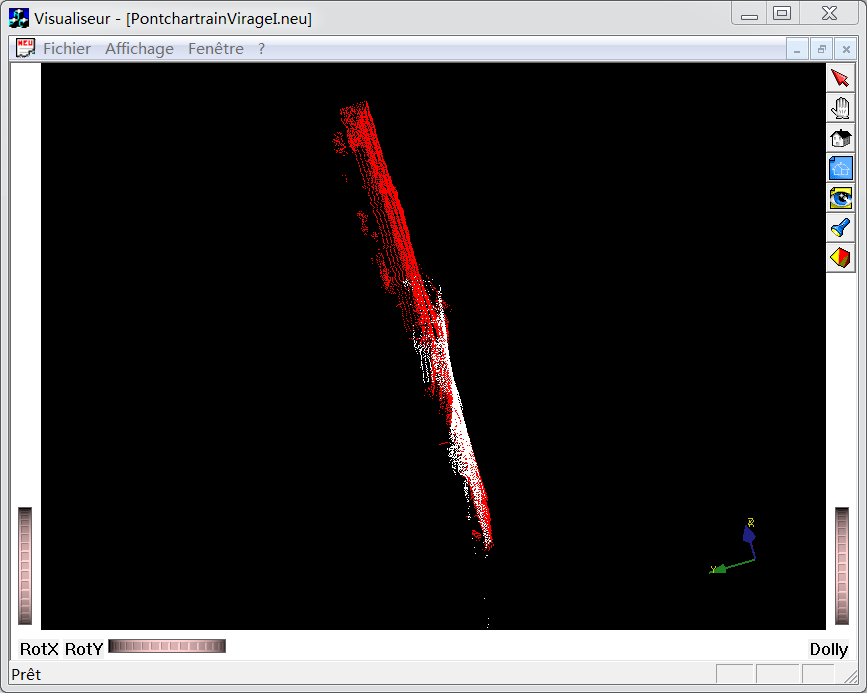
1. **Sample 3 : Virage**
2. Error:

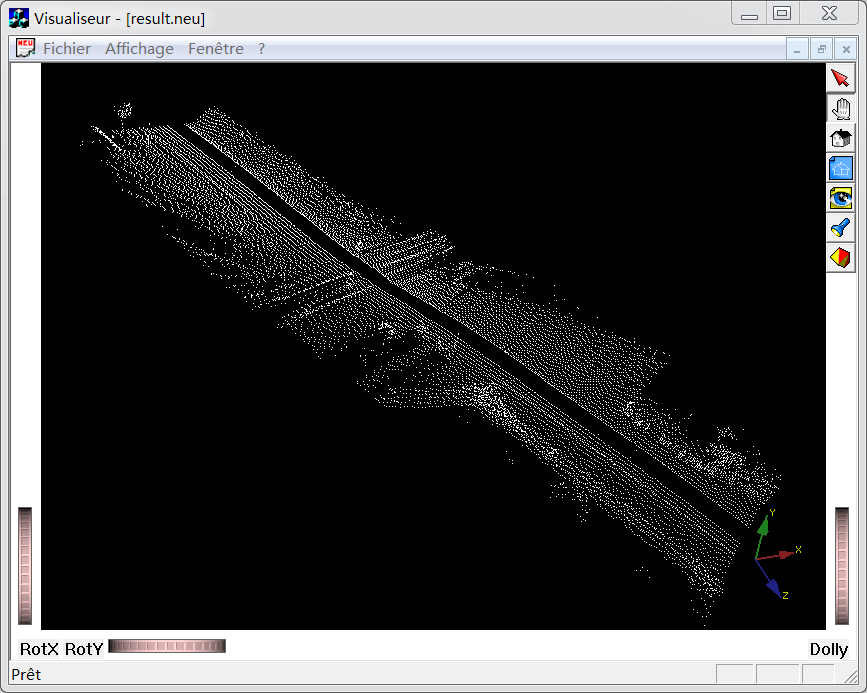
Average : 62753.81699

1. Running Time:

Total : 1371806ms

1. Results visualization:





1. Version 2 : The ANN acceleration version

General Seuil = 500;

1. Sample 1 Foot:

Average CPU : 19.78

Average Error: 10472.8513

Total time: 57532 ms

Compare with base version:

|  |  |  |
| --- | --- | --- |
|  | Base | With ANN |
| Average CPU | 21.06 | 19.78 |
| Average Error | 10.775802 | 106.4005234 |
| Total time (ms) | 58934 | 41438 |

1. Sample 2 Bunny

Average CPU : 21.78

Average Error: 0.000279

Total time: 6595ms

Compare with base version:

|  |  |  |
| --- | --- | --- |
|  | Base | With ANN |
| Average CPU | 24.06 | 21.78 |
| Average Error | 0.000071 | 0.000279 |
| Total time (ms) | 98646 | 6595 |

1. Sample 3 Virage

Average CPU : 3.56

Average Error: NaN

Total time: 23876

Compare with base version:

|  |  |  |
| --- | --- | --- |
|  | Base | With ANN |
| Average CPU | 24.85 | 5.86 |
| Average Error | 62753.81699 | NaN |
| Total time (ms) | 1371806 | 23876 |

1. Conclusion

Through comparison between the two versions, I find that Base Version uses more CPU resources and time generally. But error of ANN Version is greater than Base Version. I guess because calculation of nearest neighbor is more precise in Base Version.

In Sample Virage of ANN Version, the numbers of distance between points of {Set 1} and their nearest neighbor point in {Set 2} are large. In general, it’s 60000 to 70000. It far exceeds the threshold. So it hasn’t result. But in Base Version, there are a bit of number distance is less than threshold. So it has result. In this sample no result is correct.

In summary, I think ANN algorithm can reduce running time and CPU resources effectively without losing too much accuracy.