Université d'Ottawa
Faculté de génie

École de science
d'informatique
et de génie électrique

uOttawa

L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Electrical
Engineering
and Computer Science

On the Road!
Comprehensive Project
(32%)
CSI2120 Programming Paradigms

Winter 2023

This project is to be completed individually.

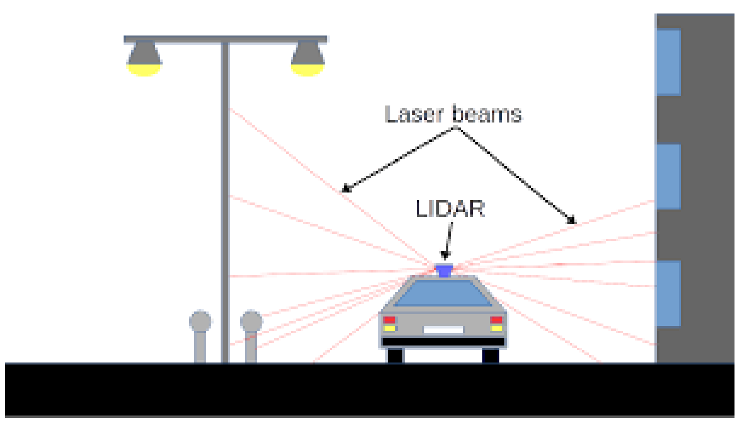
Part 1 due on 6 February at 23:30

Part 2 due on 27 February at 23:30

Part 3 due on 12 April at 23:30

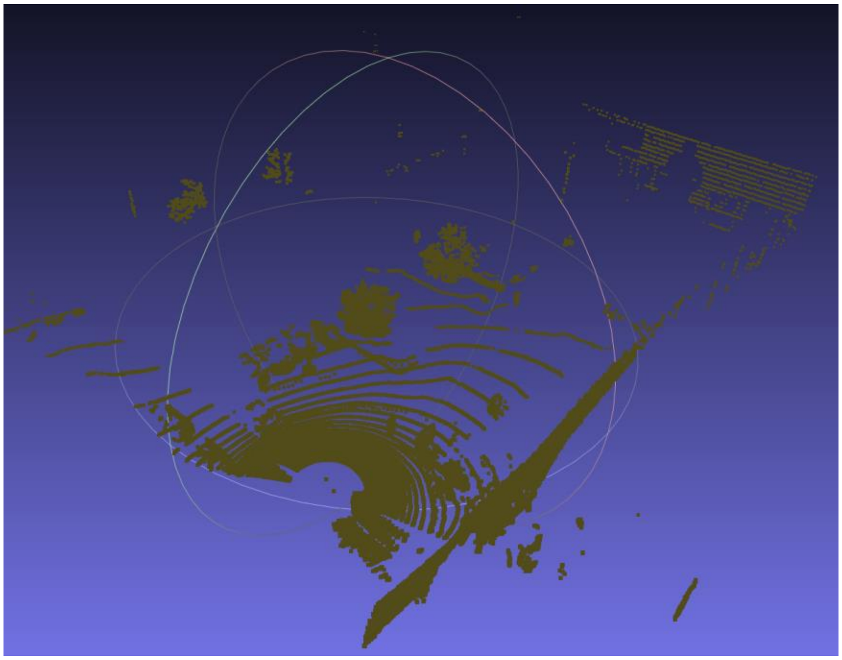
Problem to solve

The intelligent vehicles of the future will be equipped with a multitude of sensors in order to capture information about the surrounding scene and thus being able to autonomously navigate. One of these sensors is the Laser Scanner or LiDAR (Light Detection And Ranging). Using a LiDAR, the vehicle can scan the scene in front by sweeping few laser beams (typically between 8 to 64 lasers).



https://www.semanticscholar.org/paper/Ego-vehicle-localisation-using-LIDAR-and-compressed-Aronsson-Eriksson/010d3f269728a76ef62ead440541bc9481bc4a58

Each time a laser beam hit an object, the laser light bounces back to the LiDAR from which a precise distance can be estimated. A complete scan of the scene with these lasers will therefore generate a large set of 3D points (also called point cloud) that correspond to the structure of the scene. The figure below shows a typical point cloud captured by a car equipped with a LiDAR. A view of the same scene captured by a camera is also shown.



As it can be seen, the objects of the scene will be represented by clusters of 3D points. Some of the laser beams will also hit the road and the buildings' facades. These later are generally large planes. The objective of the project is to identify the main planar structures in the captured scene.

<u>Problem Description</u>

In this comprehensive assignment, you will implement an algorithm that will detect the dominant planes in a cloud of 3D points. The dominant plane of a set of 3D points is the plane that contains the largest number of points. A 3D point is contained in a plane if it is located at a distance less than eps (ε) from that plane. To detect these planes, you will use the RANSAC algorithm.

More specifically, three point clouds will be given to you. For each of them we ask you to find the three most dominant planes.

RANSAC : Random Sampling Consensus:

RANSAC is an iterative algorithm that is used to identify a geometric entity (or model) from a set of data that contains a large amount of outliers (data that does not belong to the model). It proceeds by randomly drawing the minimum number of samples required to estimate the parameters of a model instance and then validate it by counting the number of additional samples that support the computed model.

In our case, we are looking for a planar structure, made of several points, while the majority of the points in the set are outside that plane. The seek geometric entity is therefore a plane of the form: $ax+by+cz=d$. A minimum of 3 points is required to compute the equation of a plane.

Algorithm

RANSAC for the case of plane identification in 3D proceeds as follows :

1. *Initially, no dominant plane has been found, and the best support is set to 0 (see Step 5.)*
2. *Randomly draw 3 points from the point cloud.*
3. *Compute the plane equation from these 3 points.*
4. *Count the number of points that are at a distance less than eps (ε) from that plane. This number is the support for the current plane.*
5. *If the current support is higher than the best support value, then the current plane becomes the current dominant plane and its support is the new best support.*
6. *Repeat 2 to 5 until we are confident to have found the dominant plane.*
7. *Remove the points that belong to the dominant plane from the point cloud. Save these points in a new file.*

Step 6. raises the following question: how many iterations should we perform if we want to be almost certain (let's say at 99%) that we have found the dominant plane?

First, suppose that the percentage of points that support the dominant plane is p% of the total number of points in the cloud. The probability of randomly picking three points that belong to this plane is therefore p^3 %. We can then conclude that the probability of picking a set of random that contains at least one outlier is $(1- p^3)$%. If we pick k random triplets of points, the probability that these sets always contains an outlier is $(1- p^3)^k$ %. Consequently, the probability of finding at least one set made of 3 points that belongs to the dominant plane is $1-(1- p^3)^k$ %. We must therefore find the value of k that give us a confidence probability of, let's say, C= 99%.

$$k = \log(1 - C) / \log(1 - p^3)$$

To find the three most dominant plane, you then need to repeat the complete procedure 3 times.

Your task

We ask you to implement the RANSAC algorithm to identify the three most dominant planes in a cloud of 3D points. As input, we give you three point clouds, given in the xyz format, they are:

- *PointCloud1.xyz*
- *PointCloud2.xyz*
- *PointCloud3.xyz*

For each of these files, you must find the three dominant planes. Your ouput will therefore be, for each point cloud, three xyz files containing the list of points that belong to the dominant planes. You also provide the xyz file of the full point cloud in which you have removed the points belonging to the dominant planes. These files are named by appending pX to the point cloud filename. For example, for the first point cloud, the output files should be:

- *PointCloud1_p1.xyz*
- *PointCloud1_p2.xyz*
- *PointCloud1_p3.xyz*
- *PointCloud1_p0.xyz // this is the original cloud without the planes's points*

Note that xyz files can be easily visualized in a 3D space using two simple tools:

- *The first is a free simple online tool: https://imagetostl.com/view-xyz-online. You just upload the attached .xyz file and you can then visualize it nicely.*

- *The other tool is meshlab. It is a bit more advanced but is extremely easy to install and use. It is available for different OS at https://www.meshlab.net/#download.*

Programming

You have to write programs under different paradigms that solve different versions of this problem. You will receive specific instructions for each language.

Each program will be marked as follows:

Program produces the correct value	*[1.5 points]*
Program produces the correct set of items	*[1.5 points]*
Adherence to programming paradigm	*[3 points]*
Quality of programming (structures, organisation, etc)	*[1 point]*
Quality of documentation (comments and documents)	*[1 point]*

All your files must include a header showing student name and number. These files must be submitted in a zip file.