MTRN4010 – Lab Task #01

Feature Extraction from Laser Scanner Data

Part A.

Implement a function for processing an individual scan (generated by a laser scanner); in order to detect and classify objects of interest (OOI), from the raw measurements provided by the sensor. The OOIs are necessary by higher level applications in future projects. We consider as OOI any object that has a defined size: apparent diameter between 5cm and 20cm (e.g. a pole).

Input argument: assumed to be a vector, of size 361 x 1, class uint16 (16 bits unsigned int.). The data contained in the vector is the raw data associated to one laser scan; composed by a sequence of 361 "pixels". The function needs to extract range and intensity of each of the pixels, and properly scale the ranges.
Then, the data is processed in order to infer clusters (aka: segment, objects). The segments that seems to have certain size are selected and saved in a list of OOIs, whose properties are also recorded (e.g. position, size, color). The function must return an instance of a structure, designed to contain the following fields:

a) Center of geometry.
b) Size (approximate diameter).
c) "Color". Two possible values: Highly Reflective (HR=1) or Low Reflective (LR=0). We consider a pixel to be highly reflective if its associated intensity field is higher than zero. An OOI is said to be brilliant (HR) if at least one of its constituent pixels is HR.

For instance, the following fields are a good implementation of the required result:

N	: Scalar that represents the number of detected OOIs (in the current processed scan).
Colors	: Vector (of size 1 x N) having the "colors" of the N detected OOIs (see note 1).
Centers	: Matrix (of size 2 x N) for storing the X,Y coordinates of the OOIs' centers.
Diameters:	Vector (of size 1 x N) for storing the sizes of the OOIs.

Note 1: "Color" is a discrete property, indicating if the segment is opaque or brilliant. Opaque means that no pixel in the segment does have high intensity reflection index. Brilliant means that at least one of the pixels of the segment does have high intensity reflection index.

Consequently, object #i (provided that $1 \leq i \leq N$), would have its positon stored in OOI.Centers(:, i), its diameter in OOI.Diameters(1, i) and its intensity in OOI.Colors(1, i)

Part B

Implement a program for using your previous function (from part A), in a periodic fashion; for processing a sequence of images. The images are contained in a Matlab's data file (MAT file). The file contains real data (acquired from a laser scanner, according to a specified format). See the examples "ExampleProcessLaserData.m" and "ExampleUseLaserData.m" for details about reading those data files. The program will be implemented as a Matlab function, whose argument is assumed to be a string, for specifying the name of the data file.

Output of the program: The program will process a sequence of laser scans contained in the input data file, by applying the function developed in part A. It will process each individual laser scan separately, one by one. For each individual scan it will produce a list of OOIs and their associated characteristics:

The result, for each processed laser scan, will be shown dynamically in a figure. The plots will include the following features:

a) Raw points associated to the laser scan. These points will be shown by blue dots.
b) High intensity points will be shown as red dots.
c) The centers of the brilliant OOIs will be shown as green stars (opaque OOs do not need to be shown).

(All the plots must be presented in a Cartesian coordinate frame, not in the native polar representation).

The program will contain a main loop (based on 'while' or 'for' loops) for sequentially processing all the laser scans contained in the input data file. After processing an individual laser scan, the program will wait for an amount of time (e.g. ~ 100ms), before processing the next laser scan. The refreshing of the figures will be done efficiently, by using handles of the graphical objects (as shown in one of the provided examples)

Deadline for these tasks: Demonstration of this task will be on Week 5. In the demonstration, you will briefly show the tutors that your program is working. You may need to explain parts of your program, and answer questions, as part of the demonstration.

Report: There is not report associated to this task.

Note: Although you are not required to submit a report, this project involves implementing resources that you will use in subsequent projects; for that reason it is a good investment dedicating time in order to properly complete this task.
In addition, try to make the program efficient, because it will be used in subsequent projects whose programs will run in an on-line fashion. The processing time of this task (NOT including plotting) should be less than 70ms per laser scan, in the computers of the laboratory (A well implemented approach would take less than 5ms to process 1 scan).

Questions: Via Moodle Forum or email to lecturer (j.guivant@unsw.edu.au)
