laser3d

Michael Kaess

January 2002

Overview

laser3d is a visualization tool for data acquired by a laser range finder like a SICK LMS200 using MissionLab's hserver[1]. It allows the user to walk through a 3D model of the data.

This software was written as a tool for a student project[2]. It is published because it might be useful or interesting to other people, but it is not supported and most likely contains bugs.

Compiling

laser3d requires the development version of OpenGL/MESA to be installed (GLUT, GLU and GL). It comes with a simple *Makefile*. Calling *make* from the command line will create the executable *laser3d*. The program was tested with RedHat 7.1 Linux on Intel platforms. For other platforms or different configurations the *Makefile* has to be changed. For details see the comments in the *Makefile*.

In case the data file was created on a machine with different byte order than the machine used to run *laser3d*, the debug switch CHANGE_BYTE_ORDER has to be specified (see *Makefile*).

A sample data file with the name *SampleDataSet* is included. It was recorded in the third floor of the Georgia Tech MaRC building (for a photo of the hallway see http://www.cc.gatech.edu/ai/robot-lab/research/3d/) by driving the robot in a straight line through the hallway and using the millimeter resolution of the SICK LMS200 laser scanner. To view the sample data, call: *laser3d -I SampleDataSet*.

Parameters dependent on the configuration of the experiment (e.g. in which height the laser scanner is mounted) and constants influencing the behavior of the virtual world (e.g. step size when moving through the model) can be adjusted in *laser3d.h*. All parameters are explained by comments.

[1] see http://www.cc.gatech.edu/ai/robot-lab/research/MissionLab/

[2] see http://www.cc.gatech.edu/ai/robot-lab/research/3d/

Command Line Arguments

laser3d -h
laser3d -H
laser3d [*OpenGL-opts*] [-cmqrsv8CI] [-O *output-file* | -S *file.pgm*] {*input-file* | -R *file.pgm*}

-h Prints a list of acceptable command line options and exits.

-H Prints the key assignment for the 3D graphics window and exits.

OpenGL-opts Various OpenGL-Window options can be specified. See the documentation of OpenGL for details.

-c Assumes laser operated in centimeter (cm) mode when data was recorded.

-m Assumes laser operated in millimeter (mm) mode when data was recorded (default).

-q Quiet mode. Suppresses output of any non-error message.

-r Round Scan Mode. Assumes robot rotated in place and laser was mounted vertically.

-s Straight Scan Mode (default). Assumes robot drove in a straight line and laser was mounted facing upwards with the scan plane perpendicular to the movement direction of the robot.

-v Verbose mode (default). Displays information about the processed laser data and optional changes or corrections.

-8 Reads and writes .pgm files in 8 bit format (default is 16 bit). This may be helpful for visualization, since only few programs can process 16 bit pgms.

-C Cleans the data from unnessary scans in -r mode, that is it uses data from one complete rotation and omits any further scans.

-I Interpolates errors in the laser data. Such errors occur sometimes when the laser beam hits an edge. As a result the scanner will not sense any object and therefore return the highest possible distance value.

-O *output-file* Writes cleaned data to *filename* in the same format *laserfit/hserver* uses. Cleaned data means that illegal or redundant data at the start and end of the recording is removed, and further cleaning may be done depending on the presence of option -C.

-S *file.pgm* Writes raw data to *file.pgm* in pgm graphics format. In the resulting image each pixel corresponds to a single laser scan value, where darker pixels (lower values) represent shorter distances. Each horizontal line in the image

corresponds to a 180 degree scan, i.e. the resolution of the image is 361 times the number of complete 180 degree scans.
Please note that per default 16 bit pgm format is used, which is not supported by most programs. To visualize the images use -8 to write standard 8 bit pgm format instead.

-R	*file.pgm* Reads raw data from *file.pgm*. For a description of this format see option -S. Using 8 bit pgm format results in a significant loss of depth resolution. *laser3d* will print a warning message in this case.
input-file	Reads data from *input-file* created by *laserfit/hserver*.

3D Window

The 3D window displays the model generated from the laser data. It will not be displayed if the -O or -S options are used.

The initial image is generated using the viewpoint which corresponds to the starting position of the robot at the time of recording. The viewpoint can then be changed via keyboard (see table below). After each change a new image is calculated. Note that the time for this calculation depends on the complexity of the model (number of scans) and the video card.

Pressing the right mouse button over the 3D window opens a pop-up menu with additional options like display mode and floor mode, and the possibility to quit the program.

Keyboard mapping of the 3D window (case sensitive):

R	Resets viewpoint to the starting position
Q	Quits the program
y	Moves the viewpoint up by one step (0.1 m)
h	Moves the viewpoint down by one step (0.1 m)
i	Moves the viewpoint forward (0.1 m)
k	Moves the viewpoint backward (0.1 m)
u	Moves the viewpoint to the left (0.1 m)
o	Moves the viewpoint to the right (0.1 m)
j	Rotates the viewpoint to the left (6 degrees)
l	Rotates the viewpoint to the right (6 degrees)
r	Reverses the direction of view (180 degree turn)
p	Rotates the viewpoint upwards (6 degrees, maximum of 180 degrees)

;	Rotates the viewpoint downwards (6 degrees, minimum of -180 degrees)
a	Displays no floor
s	Displays a green floor
d	Displays a checkerboard floor with side length of 0.5 meters.
1	Displays the model in shaded mode using triangular interpolation of the scan points (most realistic).
2	Displays the model in triangle mode, i.e. triangular interpolation of the scan points with random color for each triangle.
3	Displays the model in quad strip mode, i.e. quadrangular interpolation of the scan points with random color for each quad strip.
4	Displays the model in quad mode, i.e. quadrangular interpolation of the scan points with random color for each quad.

The complete list of keys is printed to the terminal each time a 3D window is opened (unless the program is in quiet mode). Additionally it can be requested independently with the -H command line option.