The sensor used:

A picture containing electronics, electronic component, circuit component, passive circuit component

Description automatically generatedA picture containing loudspeaker, appliance

Description automatically generated

A close-up of a device

Description automatically generated with low confidence

From the company:

360degree view, maximum range 12 meters

Maximum sample frequency is 8K, with a scanning frequency of 5.5hz

A screenshot of a computer

Description automatically generated

A picture containing sketch, drawing, diagram, line art

Description automatically generated

Sensor was acquired based on availability and cost.

Scanning and sampling frequencies are could not be changed via the libraries used. Maximum range found through experimentation is ~13m, but with an object that is appropriately reflective, and with good environmental conditions (indoors, no direct sunlight on the sensor, white object, etc)

Code:

3 main functions: getScan, process scan, compare scans.

GetScan has no input and returns a 2D array of objects detected.

It has an iterator that makes the function return the second array, as the first array is not properly measured (cause unknown).

Process Scan has 4 input parameters: Matrix, a 2D array of [Object ID][object quality, object angle, object distance]. RatioVariable, which is a double that acts as the threshold for how close two points can be before they’re flagged. FlagNum, which is the number of flags needed for the cluster of close points to be considered an object. Lastly there’s QualityThres which is a double for the threshold for the quality of each point.

Matrix is typically a GetScan output.

RatioVariable was empirically found to be most appropriate at 0.015 (strictest) to 1.5 (loosest)

FlagNum was also empirically found to be most effective near 5.

QualityThres is 14 for most occasions as 15 is the figure that the sensor returns for all “hard” surfaces. Allowing for some discrepancy, 13-14 is ideal.

ProcessScan compares values or their derivatives with the thresholds and decides if points are close, at which point they’re labelled as spots. It then creates a new 2D array of [Object ID] [ MeanQuality, MeanAngle, MeanDistance, MeanLength, ShortestDistance, ShortestDistanceAngle], which is the output of the function.

CompareScans has 6 input parameters. Scan1, Scan2, QualityThres, AngleThres, DistThres, LengthThres. Scan1 and 2 are ProcessScan’s outputs.

Quality Thres is the absolute difference between two array cells of Scan1 and 2. Best value range 1-3, with 1 being the strictest and 3 the loosest.

AngleDistance is the absolute difference between two mean central angles. Best values are 5-15.

DistThres values are 100-200, measured in mm.

LengthThres values are 20-60, also measured in mm.

The function compares the variables of the two arrays passed, and outputs the Scan2 object that was found to be similar to a Scan1 object. (only scan2 is returned at this point).

If there is an output from CompareScans, the output is filtered for objects in the frontal 180 arc, and then divided into 30 degree sectors. If there’s an object in a 30 degree sector, the programme outputs a specific code for said sector, which allows the mission planner to take appropriate action.

0-30 = 1, 30-60 = 2, 60-90 = 3

330-360 = -1, 300-330 = -2, 270-300 = -3

There’s also the wall danger function, which outputs a code in case there’s a wall in our field of view. (wall is identified via length, as the maximum boat length for the competition is 2500mm, a threshold was put at 2600.) If anything larger than 2600mm is found, it must be avoided at all cost, so it has its own code situation.

Wall detected (angle is not of interest) = 4, wall detected in front = 5, wall detected in front, close = 6.