Obstacle Characterisation with Proximity Sensors

Tim Nguyen, Andrei Nitu, Ainsley Rutterford, and Faizaan Sakib

I. Introduction

This report documents an experiment conducted to test the capabilities of environment mapping within a robotic system. The task at hand is exploring the challenge of obstacle characterisation. A baseline solution is produced with the use of an infrared (IR) proximity sensor. Subsequently, an optimised solution using two proximity sensors is produced. The details of both implementations are reported. Finally, the two implementations are analysed, compared and evaluated using appropriate metrics. The goal is to understand how the optimisation influences the task of mapping. The experiment and its implementations have been carried out on a Pololu Romi 32U4.

II. Obstacle Characterisation

There is a wide range of domains to consider when attempting to map a robot's surrounding environment. However, this experiment focuses solely on the use of obstacle characterisation to investigate its effectiveness in addressing the mapping problem.

Obstacle characterisation is the task of identifying surrounding obstacles. A robot can use this information, along with its estimated pose, to create an internal understanding of the world around it. This is very useful as it opens up the possibility of truly autonomous navigation.

Simultaneous Localisation and Mapping (SLAM) [2] is a well-generalised and robust technique developed within this domain. It incorporates the use of cameras or infrared sensors, translating visual data into a map of a system's environment, while simultaneously inferring its location within the map. For instance, the introduction of SLAM in recently developed robot vacuum cleaners has led to superior performance to their predecessors, which would follow walls or randomised paths. By mapping the environment, the robots can traverse and clean rooms in a more systematic way which is faster and more efficient.

The implementation proposed for this experiment makes use of an IR proximity sensor in order to obtain distance measurements to obstacles in front of the robot. Since the information required to map the environment originates from within the robot itself, this would form the basis of a robot-centric mapping system, similar to SLAM. This proves to be a more autonomous system compared to a world-centric system, where the robot knows about the surrounding "world", most likely as a map provided as input. However, the experiment to follow will give an insight into how the use of sensors come with their own limitations which can restrict the performance of a system.

III. Hypothesis

A. Background

The accessibility of IR proximity sensors due to their relatively low cost and power consumption has introduced the possibility of beneficial distance measuring to the mass majority of systems.

However, the performance of this type of sensor is prone to multiple limitations. Firstly, IR proximity sensors exhibit non-linear behaviour, which is later discussed and shown in Figure 1. In addition, measurements of IR sensors are based on the intensity of IR light reflected by the surface in front of it. This means that the reliability of IR sensors is subject to the reflectance of surrounding objects and the angle of light intersection [1]. Finally, the accuracy of measurements from the sensor can be undermined by the heading estimation of the robot. This can lead to a distance measurement being attributed to the wrong angle relative to the robot.

Admittedly, these limitations put together can propagate a high level of error. As a result, this type of sensor is more commonly used for proximity detection using thresholding, rather than attempting to accurately map the environment with distance measurements. This task is better performed with the use of ultrasound sensors [1], or RGB cameras for more complex implementations such as SLAM [2].

B. Initial Hypothesis

The proximity sensor which will be used is the Sharp GP2Y0A60SZLF distance sensor [3]. This sensor is tested over a range of distances away from an obstacle. The relationship between the sensor voltage values produced and the distances is shown in Figure 1.

To gain a better understanding of the problem relating to proximity sensors, we can interpret their performance. The characteristics of these sensors will help to define an initial hypothesis forming the basis of the experiment.

Figure 1 provides a better understanding of the conditions that define the effectiveness of the sensor. For distances below 10cm, there is a spike and a subsequent fall in the voltage produced. This leads to voltage values that represent two separate distances. In order to maintain a continuous range of valid measurements, the values leading up to the peak of the spike can no longer be considered. Thus, distances below ∼10cm cannot be reliably measured. We will refer to these distances below the ∼10cm mark as the "blind spot" of the proximity sensors.

After the peak occurs, there is a high rate of decrease of voltage between ∼10-30cm, as shown by the steep gradient of the curve. Distances in this range can be expected to provide more accurate measurements. The high rate of change of voltage makes the measurements more distinguishable

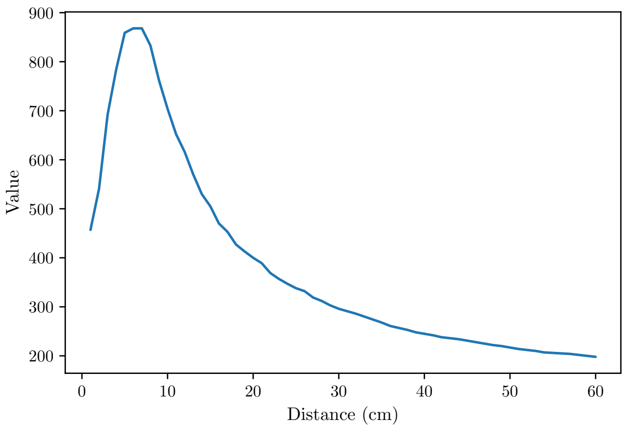


Fig. 1: The raw values in the range $[0, 1023]$ obtained from the Sharp GP2Y0A60SZLF proximity sensor when varying its distance from an obstacle.

between small changes in distance. Although the rate of change decreases towards 50cm, it should be possible to interpret the measurements produced as the rate of change should still be high enough. The curve tends towards a flat response across distances above 50cm, with the difficulty of obtaining accurate measurements increasing with distance.

With the preceding knowledge, we form the initial hypothesis:

> *"Infrared proximity sensors perform with a varying degree of accuracy, determined by the distance being measured. They are prone to a higher degree of error at the limits of their operation. External factors based on the environment and the system itself can introduce additional error. We hypothesise that the use of an IR proximity sensor will be able to provide sufficiently reliable measurements for distances in a certain range. Meanwhile, it will fail to give usable measurements for smaller distances."*

This leads to the definition of the "baseline" implementation of the experiment to test this hypothesis. This will involve a single IR proximity sensor placed at the front of a robot to measure distances of obstacle(s) around it. The experiment should contain an environment capable of testing the sensor over a broad range of distances to be able to test this hypothesis.

C. Final Hypothesis

As stated in Section III-B, there is an existence of a "blind spot" for distances measured by an IR proximity sensor below ∼10cm. This creates a non-continuous range of accurate measurements, caused by the non-linear behaviour of the sensor itself. It can be argued that this is a more severe constraint to the performance of the mapping system compared to the inaccuracy found in the higher levels of distance measurements. This is because for smaller distances, there may be false over-estimation of the obstacle in front. This could lead to obstacle collision, which can essentially be described as a system failure. On the contrary, measurements above 50cm, although inaccurate, do confirm the existence of an obstacle.

As such, we propose the introduction of a second sensor in an attempt to correct, or at least reduce the effects of this blind spot. Therefore, the variable that will be changed for the improved implementation is the number of proximity sensors. The second sensor would have to be placed at some offset behind the main sensor. This will in theory enable it to produce usable measurements for distances that the main sensor cannot.

Additionally, for readings above ∼10cm, the expectation is that the two proximity sensors will lead to improved performance. The distances in the accurate range and its fringes defined in Section III-B can be more confidently estimated by taking into account the measurements from both sensors. With a secondary sensor available, the effect of the error of one measurement can be reduced by the measurement from the other sensor.

The previous hypothesis can now be redefined:

> *"Infrared proximity sensors perform with a varying degree of accuracy, determined by the distance being measured. They are prone to a higher degree of error at the limits of their operation. External factors based on the environment and the system itself can introduce additional error. We hypothesise that with the introduction of a second sensor positioned appropriately, smaller distances can be reliably measured where previously not possible due to an increased combined range of distance covered. Furthermore, a combination of two sensor readings will help to produce better estimations."*

IV. Baseline Implementation

An IR proximity sensor is placed at the front of the robot. This ensures that distance measurements read by the sensor can most accurately describe the distance between the robot and a potential obstacle.

A. Calibration of the proximity sensors

In order to read from any sensors, an `analogRead()` call is used. Since the `analogRead()` call returns an 8-bit value, the value will be in the range of $[0, 1023]$. The values output by the proximity sensor are not linearly related to the actual distance measured by the sensor, and so these values must be calibrated. The original values output by the sensor at different distances from an object are shown in Figure 1.

Calibrating the proximity sensor allows the value produced by the sensor to be accurately mapped to the actual distance measured. Calculating this mapping function was achieved in multiple steps. First, an approximation of the function that maps the real distance to the value must be found. Then, the inverse of said function must be calculated. Finally, the raw value read can be input to the inverse function, and an accurate distance measurement should be returned.

Least squares fitting was used in order to approximate the function that maps the real distance to the raw value

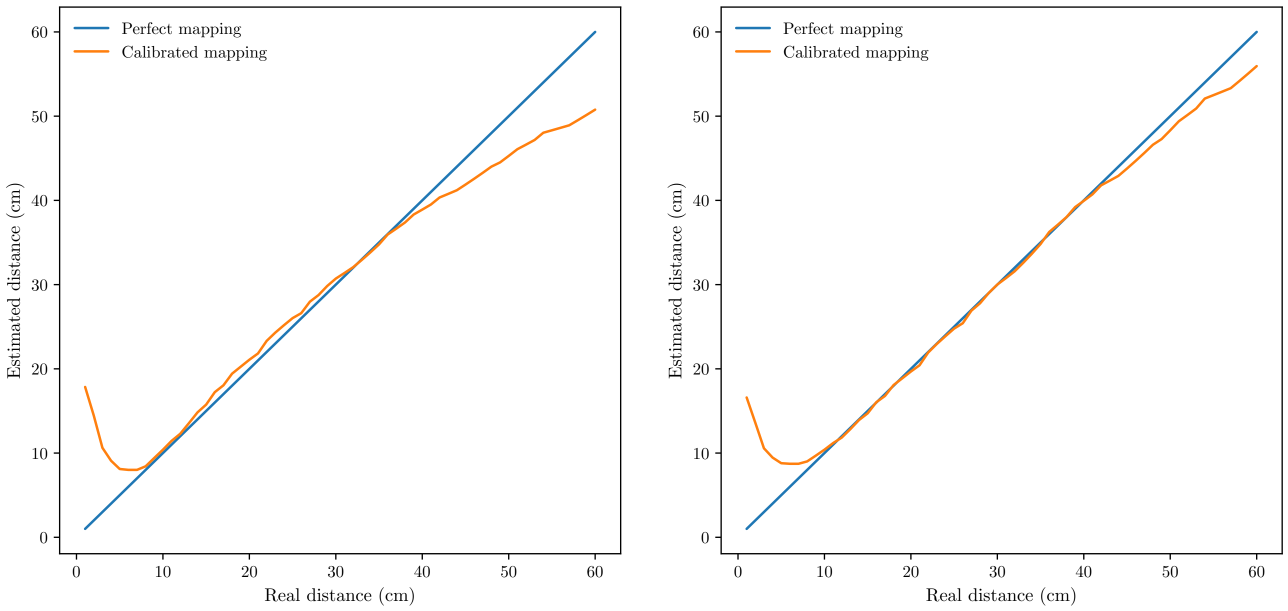


Fig. 2: The estimated distance calculated for raw `analogRead()` values against the actual distance measured using base functions of $f(x) = \left(\frac{x}{a}\right)^{-\frac{1}{b}}$ and $f(x) = \left(\frac{x^c}{a}\right)^{-\frac{1}{b}}$ respectively. Ultimately, the calibration shown on the right was used.

produced. Initially, the least squares algorithm was provided with a base function of the form

$$f(x) = \left(\frac{x}{a}\right)^{-\frac{1}{b}} \tag{1}$$

The least squares algorithm then returns the values a and b which best approximate the function that maps the real distance to the raw value produced. The raw values corresponding to the real distances in the range of 10-60cm were supplied. Once the parameters a and b were approximated, the inverse of the function can be used with these parameters in order to calculate the estimated distance for a given raw value. Equation 1 has an inverse in the form

$$f(x)^{-1} = \frac{a}{y^b} \tag{2}$$

Now that the inverse function has been calculated and the parameters have been approximated, the estimated distance calculated for the raw values measured can be plotted against the actual distance measured for each corresponding value. The results are shown in Figure 2 (left).

It can be seen that the resulting function does not perform well for values below ~10cm and above ~40cm as the estimated distance begins to diverge from the real distance measured. It is also clear that the distances in the range of ~10-40cm are over estimated. Looking at Figure 2 (left), the function that best fits the calibrated mapping is the radical function ($f(x) = \sqrt[c]{x}$). The inverse of the radical function is the power function ($f(x) = x^c$) so, in order to compensate for the undershoot of values lower than 40cm and the overshoot of values higher than 40cm, an exponent (that is >1) is added to the x factor of the equation.

$$f(x) = \left(\frac{x^c}{a}\right)^{-\frac{1}{b}} \tag{3}$$

with an inverse of

$$f(x)^{-1} = \sqrt[c]{\frac{a}{y^b}} \tag{4}$$

Plotting the estimated distance calculated for the raw values against the actual distance measured for each corresponding value then resulted in the graph shown in Figure 2 (right). The resulting mapping function is far more accurate, producing a sum of squares value of only 598cm from the 'perfect mapping' curve, compared to the 1193cm produced by the previous calibration. It is worth noting that even the improved calibrated mapping is only accurate in the range of ~10-40cm.

Although the Sharp GP2Y0A60SZLF proximity sensor advertises a detection range of 10-150cm [3], the produced mapping function only accurately maps `analogRead()` values to the estimated distance for distances in the range of ~10-40cm. When measuring distances outside of this range, the estimated distance produced cannot be assumed to be accurate.

V. Improved Implementation

A. *The second proximity sensor*

As mentioned in Section IV-A, the proximity sensor mounted at the front of the Romi is only accurate for distances in the range of ~10-40cm once calibrated. In order to obtain valid readings at very close range, while also maintaining the maximum range of the first sensor, we designed an improved implementation that makes use of a second proximity sensor mounted closer to the centre of the robot, at an offset from the first sensor. The implementation makes use of the information provided by both sensors in order to accurately cover a wider range of distances.

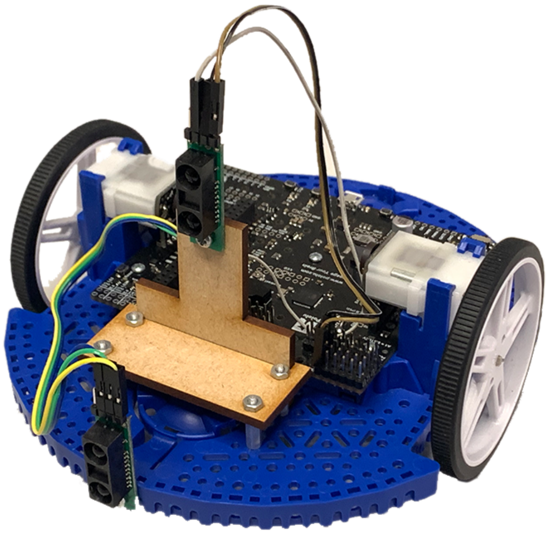


Fig. 3: An image of the Pololu Romi 32U4 used to carry out the experiment. The first proximity sensor is mounted on the front of the Romi chassis directly, whilst the second sensor is mounted using a laser cut mounting system. The second sensor is 46mm behind the first sensor.

In order to mount the second proximity sensor securely, a mounting system was designed and cut from a 3mm MDF sheet. The mounting system is shown in Figure 3. The second sensor was mounted above the first sensor to prevent the first sensor from blocking the infrared light produced by the second sensor. The sensors were both mounted in parallel to each other so as to ensure that they were both measuring the correct distance relative to each other. The second sensor was mounted closer to the centre of the Romi chassis, at an offset of 46mm from the first sensor.

B. Combining the Reported Distance Measurements

The improved implementation makes use of the distances provided by both sensors. Combining the measurements to provide the most accurate results required a heuristic of some sort to be decided upon.

The first case the heuristic had to cover was the first sensor being too close to an object. In this case, the distance that would be returned would simply be the distance measured by the second sensor. In order to detect when the first sensor was too close, the measurement from the second sensor was also taken into account. The sensors are exactly 46mm away from each other. Therefore, if the distance between the proximity sensors is reported to be less than 46mm (minus some tolerance to account for noise), the first sensor's value must be on the wrong side of the voltage peak that occurs at ∼10cm in Figure 1, and so is too close to be taken accurately. There is also the case that both of the sensors are too close to an object to read an accurate distance. In this case, the distance reported by the second sensor would be greater than the distance reported by the first sensor—which is not possible. Hence, the distance would simply be reported as zero.

```
$diff \leftarrow abs(ir0 - ir1)$
$offset \leftarrow 46$
$min\_dist \leftarrow 100$
$max\_dist \leftarrow 400$

if $diff < (offset - 5)$ then
    $distance \leftarrow ir1 - offset$
else if $ir0 + offset \geq max\_dist$ then
    $distance \leftarrow ir0$
else if $ir0 \geq ir1$ then
    $distance \leftarrow 0$
else
    $distance \leftarrow (ir0 + (ir1 - offset)/2)$
end if
```

Fig. 4: The pseudocode of the heuristic used to decide the extent to which each sensor measurement contributes to the final distance measured. $ir0$ and $ir1$ are the distance measurements in millimetres from the first and second sensors respectively. All variables are represented in millimetres.

The next case the heuristic had to cover was when the first sensor was reading a value that was within the ∼38-40cm range. In this range, the first sensor—which is mounted farther forward—is in its accurate range. However, the second sensor, which is at an offset of 46mm behind, would no longer be in its accurate range. In this case, the distance value reported by the first sensor would be the final value returned.

Finally, the ideal case is covered where both sensors are within their accurate ranges. The returned value is simply the average of the two distance values reported, with the offset being taken into account (i.e., $distance = (ir0 + (ir1 - offset))/2$). The implementation of the heuristic is summarised by the pseudocode shown in Figure 4.

VI. The Experiment

This section will define the methodology and the variables of the experiment. This is followed by an outline of the motivation for the choices made to ensure a robust and consistent experiment.

A. Methodology

As highlighted in Section III, the experiment focuses on improving the subsystem of obstacle characterisation. Specifically, the performance will be assessed between the implementations as described in Section IV and V. To acquire performance data for these two systems, the experiment will be carried out as follows:

1) The robot is placed in the centre of a square perimeter acting as a wall obstacle, with sides of length n cm.
2) Starting from the direction it is facing, the robot is switched on and rotates a full circle in a clockwise direction.
3) During the rotation, distance measurements from the proximity sensor are taken for the angle it is facing.
4) The resulting output is an array of distances with the index of each value corresponding to the angle it was measured at.

The primary independent variable in this experiment will be the number of proximity sensors used, as stated in Section III, whilst the sole dependent variable will be the distance measurements obtained (which are adjusted accordingly when using multiple sensors to take into account the offset between the two sensors).

The other independent variable will be the size of the square perimeter. This will be used to assess the system's performance for different distances. The lengths used for the square wall will be 33, 44 and 66 cm. This will be justified later in Section VI-E.

B. Recording Distance Measurements

A key part of the experiment is to measure distances using the calibrated proximity sensor. It is crucial to do this with a standardised approach throughout the experiment. The robot rotates clockwise on the spot at a constant speed. Kinematics is used to estimate the robot's current pose, allowing the robot to stop when it has successfully turned 360°. Multiple distance measurements are recorded at each degree, and the average value is taken in order to reduce the effect of the noise present in the sensor readings. Once the robot has turned 360°, an array containing the 360 distances measured is output, one for each of the heading angles of 0-359°. These values can then be used to calculate the metrics discussed later in Section VI-F, and can also be converted to coordinates in order to be visualised.

In order to rotate the robot at a constant speed, PID controllers are used to control the speed of both wheels separately. A constant rotation speed of 12 degree/s is requested. When rotating at this speed, an average of 185 measurements are taken per degree. When rotating at a speed slower than this, although the amount of readings per degree does increase which could lead to more robust results, the wheels are turning slow enough that the robot begins to turn one wheel at a time—i.e., the left wheel would turn a small amount, followed by the right, and so on. As a result of this, the robot can begin to deviate from its starting position, producing incorrect results. From 12 degree/s upwards, there is a trade off between the number of measurements taken per degree, and time taken to complete the task. Since the task is completed in $\sim$30 seconds at this speed, a faster speed was not required.

Although the PID controller is supposed to ensure that the robot rotates at a constant speed, it is not guaranteed. As a result of this variable speed, a variable number of distance measurements may be taken at each degree. In order to allow for these multiple readings per degree, the current reading is simply added to the existing value for the corresponding degree, and the result is divided by two. Effectively, the average is updated each time a value is read. While rotating clockwise, the pose is constantly checked. Once the reported pose is within 0.1° of 360°, the robot will pause and upon a button press, will print the values recorded so far.

C. Converting Distances to Coordinates

Once the array of 360 distance measurements was obtained, they could then be converted to coordinates in order to visualise the data. The following equations were used:

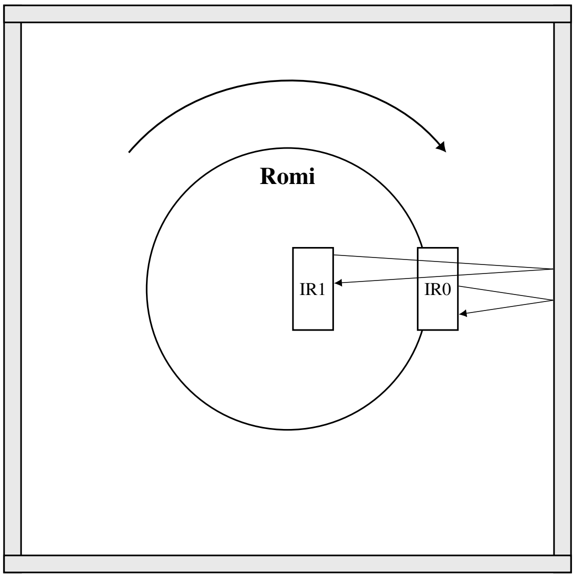


Fig. 5: A bird's-eye view of the experiment.

$$x_o = x_r + d\sin(\theta) \tag{5}$$

$$y_o = y_r + d\cos(\theta) \tag{6}$$

where x_o and y_o are the coordinates of the object that has been detected, x_r and y_r are the coordinates of the centre of the robot, d is the distance measured, and θ is the angle at which the distance was measured. Note that the distance d supplied to this equation must be the distance from the centre of the robot to the object, not from the sensor to the object. The first sensor is mounted on the front of the Pololu Romi chassis, which has a radius of 81.5mm. The sensor is mounted on a 2mm thick PCB, and has a thickness of 5mm itself. Thus, the distance d in the equation above is actually the distance measured plus an offset of $81.5 + 2 + 5 = 88.5$mm. The second sensor is mounted 46mm behind the first, and so this offset is also taken into account for its readings.

D. Control Variables

Below is a complete list of control variables that have been considered and kept constant to minimise external interference throughout the course of the experiment:

1) Hardware:

- *Robot:* Polulu Romi 32U4
- *IR proximity sensor:* Sharp GP2Y0A60SZLF
- *Sensor placement:* Specified in Section IV and V-A.
- *Sensor mounting:* Specified in Section V.
- *Batteries:* A fresh set of batteries were used before the start of the experiment.
- *Pin usage:* The same set of pins were used for each component throughout. Usage of only the un-reserved pins ensured that there was no interference from other components of the robot.

2) Software:

- *Turning speed:* 12 degrees per second. Specified in Section VI-B.
- *PID values:* $K_p = 12.85$, $K_i = 0.071$, $K_d = 0.00$. Based on encoder counts per second.
- *Number of readings for each angle:* Average of 185 readings. Specified in Section VI-B.

3) Environment:

- *Square perimeter:* Made of panels of MDF securely taped together and fastened with clips to ensure straightness. Tape covered with white paper to avoid interference from the variability of the reflectance of different materials (e.g. MDF, and the various types of tape used). The perimeter is formed as a square by measuring all 4 sides, the two diagonal lengths, and using a try-square.
- *Floor:* Flat and smooth surface ensuring minimum friction.

4) Procedure:

- *Rotation:* 0° to 359° in a clockwise direction.
- *Number of repeats:* 10 for each configuration.
- *Robot placement:* Centred within perimeter using taped guidelines on the floor. If not centred, the skew is adjusted post-result when plotting in Python. Initially set to 0 degrees facing the perimeter perpendicularly.
- *Data collection:* Results printed on button press at the end of rotation. This means the USB cable is not connected during the experiment run, avoiding movement obstruction which may affect the estimated pose/heading.

E. Rationale

A key aspect of this experiment is the use of the square perimeter. With the measurement of distance defined as the dependent variable, it is crucial to select an environment from which relevant results can be produced. Having to map the perimeter of a square from the inside means there are varying distances that the sensors have to perform against. Moreover, the environment tests the sensors against different angles of intersection. The robot has to measure distances to a perpendicular surface at the mid-point of each side, and at an angle of 45° at the corners. Thus, the environment can test the systems with rigour, and as a result, provide useful data to analyse.

In order to test a wide range of distance estimations, three problem sizes were chosen: square perimeters of side length 33, 44 and 66cm. Both the 44cm and 66cm square perimeters require the robot to map distances that are within the accurate range of the first proximity sensor. The range of values the sensor must detect are 14-22cm and 25-37cm for the 44 and 66cm boxes respectively. However, when mapping the 33cm box, the robot is required to detect distances in the range of 7-15cm, including values below a sensor's minimum accurate distance of ∼10cm.

The expected theme is that the addition of more hardware, in this case the second sensor, will improve performance. However, it is not possible to be certain about this without proof. Furthermore, Section III shows the performance of a single proximity sensor to be volatile across a range of distances. It will be interesting to see if the addition of a second sensor affects the performance uniformly across this range, as opposed to a varied performance according to which part of the range the measurement falls within (e.g. performing better for distances <10cm but worse for >30cm).

Although interpreting measurements from one sensor is trivial, the same cannot be said for when there are two sensors involved. In this case, there may be numerous possible approaches for getting a meaningful value. The heuristics used in the improvement implementation, explained in Section V, is the eventual approach taken for this experiment. Therefore, it must be noted that this experiment and its findings may not represent all systems where multiple proximity sensors are used. For example, an average of readings from the multiple sensors may produce different results.

F. Quantitative Metrics

In order to assess the performance of the robot during an experiment, three different quantitative metrics were used.

1) The sum of absolute differences: This metric is defined by the following equation

$$E = \frac{\sum_{i=0}^{359} |e_i - d_i|}{360} \tag{7}$$

where E is the final error value, e_i is the estimated distance measured at an angle of i, and d_i is the ground truth distance at an angle of i.

This metric essentially returns the average error of each distance measurement. The absolute value of the difference allows both over-estimated and under-estimated values to be taken into account. For example, if no absolute function was present, a result containing ten over-estimated values by 1cm as well as ten under-estimated values by 1cm would return an error of zero.

Whilst this metric can be useful to quantify the overall performance of the obstacle characterisation, due to the presence of the absolute function, the data concerning the amount of over-estimated and under-estimated differences is lost. If the error produced was 1cm, how much of this error was due to over-estimations, and how much of this error was due to under-estimations?

2) The sum of over-estimated differences and the sum of under-estimated differences: These metrics are defined by the following respective equations

$$E = \frac{\sum_{i=0}^{359} max(e_i - d_i, 0)}{360} \tag{8}$$

$$E = -\frac{\sum_{i=0}^{359} min(e_i - d_i, 0)}{360} \tag{9}$$

where E is the final error value, e_i is the estimated distance measured at an angle of i, and d_i is the ground truth distance at an angle of i.

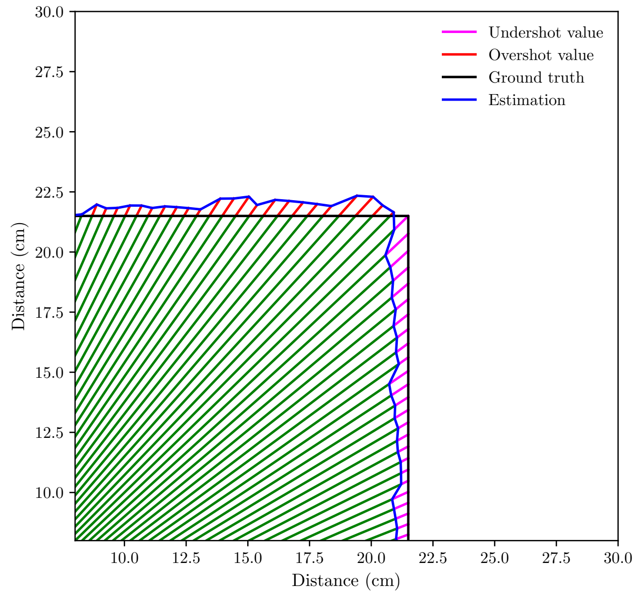


Fig. 6: A visualisation of the error metrics used. Red lines represent the over-estimation error, and magenta lines represent the under-estimation error. The sum of absolute differences is the sum of the lengths of the red and magenta lines divided by the total number of measurements. The sum of under-estimated differences is the sum of the lengths of the magenta lines divided by the total number of measurements. And finally, the sum of over-estimated differences is the sum of the lengths of the red lines divided by the total number of measurements.

The max and min functions mean that the resulting error value is correctly labelled as an over-estimate or an under-estimate. For example, if a value is under-estimated, $e_i - d_i$ would be negative, and thus the result of $min(e_i - d_i, 0)$ would be a negative result and would contribute to the under-estimation error. Since, in this case, the result of $max(e_i - d_i, 0)$ would be 0, the result would not contribute to the over-estimation error.

These metrics address the issues that arise with the sum of absolute differences metric. It is still important to note that these metrics alone only highlight either the over or under-estimation of distances, and thus should always be considered together when assessing the performance of the subsystem. In order to better explain the metrics used, the three metrics are visualised in Figure 6.

One feature of all of the above metrics that may be considered negative, is their sensitivity to outliers. Since each of the metrics are effectively some kind of mean, a small number of particularly high or low anomaly readings can affect the results more than they perhaps should.

G. Qualitative Metrics

The main qualitative metric used to assess the performance of the subsystem was plotting the estimated coordinate values produced by the robot, overlaid with the ground truth coordinates that represent the perimeter. Examples of this are shown in Figure 7. These plots allow us to visually inspect the results and make inferences about the performance that would otherwise have been hard to make.

VII. Baseline Subsystem Results and Analysis

The baseline and improved subsystems are assessed on their performances in increasing order of the size of the square perimeter.

A. 33×33cm

The baseline implementation making use of only one proximity sensor performed with varying degrees of accuracy for each problem size. When mapping the 33cm perimeter, the limitations of the single proximity sensor approach are highlighted by the areas of the perimeter closest to the robot. Looking at Figure 7a, it can be seen qualitatively that the corners of the square perimeter are mapped better than the edges, with each edge consistently being over-estimated. This is also visualised in Figure 8.

This phenomena can be explained by the calibration mapping that was implemented in Section IV-A. Recall that the optimal mapping found was only accurate within the range ∼10-40cm. The offset of the sensor from the centre of the Romi was calculated to be 88.5mm. When mapping the 33cm perimeter, the first sensor will be operating outside of its accurate range since $330/2 - 88.5 = 76.5$mm. Referring back to Figure 2, it can be seen that when reading a real distance of ∼7.5cm, the mapping function will over-estimate the distance measured. However, the over-estimation is relatively small, with the distance measurements close to the real distance. This is explained by the fact that at ∼7.5cm, the estimation is still close to the perfect mapping as shown in Figure 2. Any more decrease in distance required to measure would have a high rate of increase in over-estimation.

This phenomena is also reflected by the quantitative measurements. As seen in Table I, for this problem size when mapping with one sensor, the sum of over-estimate differences is 0.24cm, whilst the sum of under-estimated differences is only 0.18cm.

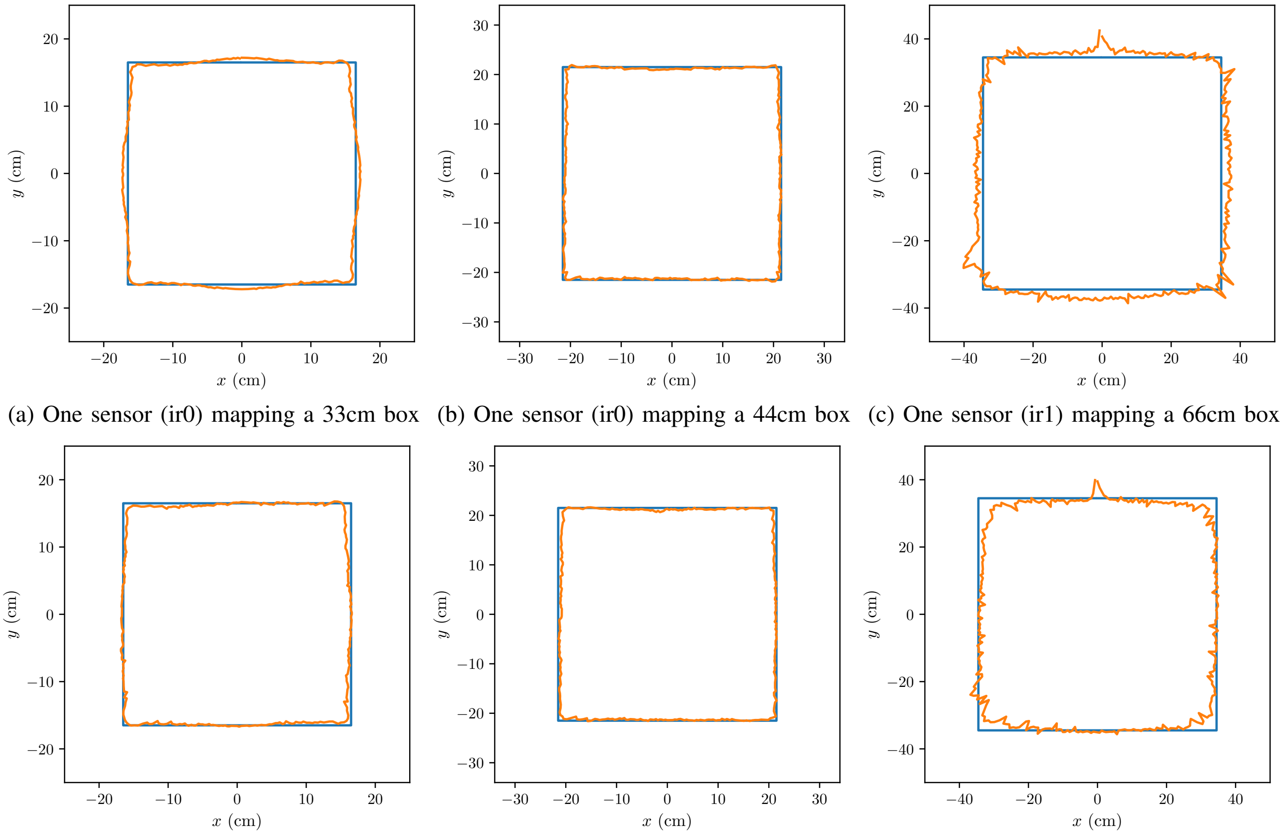
Although the resulting measurements produced have come close to the real distance of the perimeter, the aspects in which the experiment falls short is in line with the hypothesis stated in Section III-B. It produces a higher degree of error for the shorter distances measured. These distances are expected to have come in the range before the spike in voltage produced shown in Figure 1.

B. 44×44cm

When mapping the 44cm perimeter, the baseline implementation performed well, achieving a sum of absolute differences of only 0.38cm. The maximum distance the sensor was required to map was $\sqrt{(220^2 + 220^2)} - 88.5 =$ 222mm, whilst the minimum required distance to be mapped was 131mm. The measurement range for this perimeter (131-222mm) falls on the part of the curve of voltage readings where the gradient is steepest. Therefore, distances within this range would likely be represented over a larger range of voltage, leading to accurate measurements.

TABLE I: Performance of the two implementations in terms of the three quantitative metrics defined in Section VI-F.

Metric	One sensor			Two sensors		
	33 cm	44 cm	66 cm	33 cm	44 cm	66 cm
The sum of absolute differences (cm)	0.42	0.38	1.81	0.33	0.39	1.17
The sum of over-estimated differences (cm)	0.24	0.02	1.70	0.02	0.01	0.12
The sum of under-estimated differences (cm)	0.18	0.36	0.11	0.31	0.38	1.05



(a) One sensor (ir0) mapping a 33cm box (b) One sensor (ir0) mapping a 44cm box (c) One sensor (ir1) mapping a 66cm box

(d) Two sensors mapping a 33cm box (e) Two sensors mapping a 44cm box (f) Two sensors mapping a 66cm box

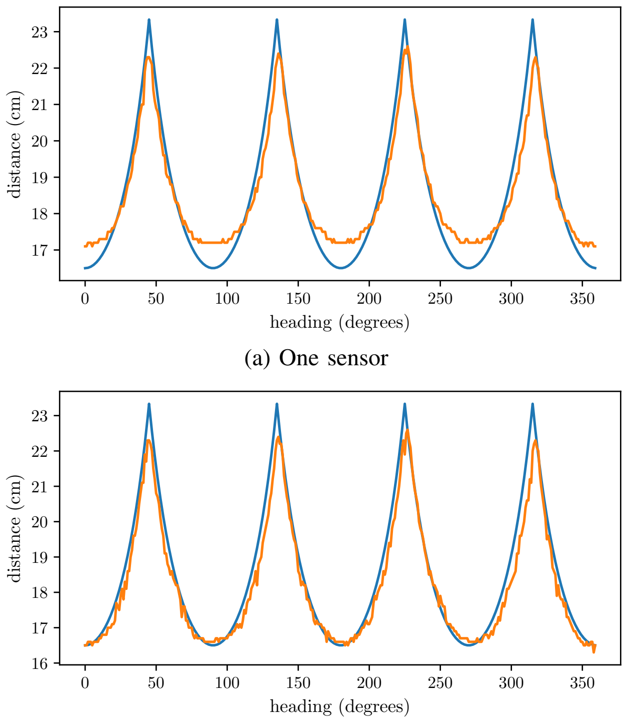
Fig. 7: Visualisations of each experiment scenario. In each plot, the orange line is the estimated box boundary, whilst the blue line is the ground truth boundary. The results shown for the 66cm box using one sensor were obtained with the ir1 sensor as opposed to the ir0 sensor in order to highlight the inaccuracies produced when measuring distances over the 40cm range.

When visually inspecting the results achieved when mapping the 44cm perimeter with one sensor, it is clear that the top left corner is not mapped as well as the rest since this corner is under-estimated noticeably more. We believe that this under-estimation occurs due to error accumulating in the pose estimation of the robot. Since the three other corners are consistently mapped better, and the box was ensured to be a square by measuring the two diagonal lengths, we reasoned that the final corner was incorrectly mapped since the robot it reporting a pose that is for example one degree further clockwise than it actually is, resulting in a skew of the distances taken.

C. 66×66cm

The performance achieved when mapping the 66cm perimeter is noticeably worse than the other two problem sizes. Not only do the estimated distance values contain more noise, they are also being consistently over-estimated as reflected by the sum of over-estimated differences being 1.70cm compared to the sum of under-estimated differences being only 0.11cm. This result is the opposite of what we were expecting. When looking at the Figure 2, it can be seen that when estimating farther distances past the range of 40cm, the values should be under-estimated as opposed to over-estimated. In order to investigate why these values were being over-estimated, we conducted a small experiment.

The experiment was to keep the robot and the object it was measuring a distance from still, and to record the distance measurements taken over time. With further investigation, it was found that when taking measurements at a distance of 20cm, the range in the raw value measured from `analogRead()` was 25, whereas when taking measurements at a distance of 40cm, a range of almost 200 was produced. We also found that the spikes of noise decreased the value obtained rather than increasing it. Since lower raw values correspond to higher distance measurements, these results helped explain why the sensor is consistently over estimating as opposed to under estimating.



(a) One sentsor

(b) Two sensors

Fig. 8: The distances measured over 360 degrees when mapping the 33cm box. In both plots, the blue line represents the ground truth distances, and the orange line represents the estimated distances. Here, it is obvious that the single sensor implementation consistently over-estimates the distances measured at the edge of the box, whilst the two sensor implementation did not.

Summarising the quantitative metrics produced for the baseline experiment in Table I, it can be seen that the problem size of 44cm produced the most accurate results, with a sum of absolute difference of 0.38cm. The next most accurate size is the smallest, of size 33cm, with a sum of absolute difference of 0.42cm. Finally, this is followed by the largest problem size of 66cm, with the least accurate performance with a sum of absolute differences of 1.81cm.

VIII. Improved Subsystem Results and Analysis

A. 33×33cm

When mapping the 33cm perimeter, the two sensor implementation performed better than the baseline implementation. This is reflected both by the qualitative metrics, and the quantitative metrics, with the sum of absolute differences being 0.33cm compared to the 0.42cm achieved by the baseline implementation. Inspecting the visualisations shown in Figure 7a and b, it is clear that the improvement in performance is due to the reduced over-estimation occurring when mapping the edges. This is also apparent when looking at the sum of over-estimated differences, with the value being 0.02cm for the baseline implementation, compared to 0.24cm for the improved implementation.

The better mapping of the edges is due to the second sensor and heuristic used, since the robot would have detected that the outer sensor is no longer within its accurate range, and would begin only taking the readings from the second sensor into account. This supports the main motivation of our hypothesis, where the introduction of the second sensor would provide accurate measurements for small distances which would be otherwise inaccurate with one IR proximity sensor.

B. 44×44cm

The improved implementation performs comparably to the baseline implementation when mapping the 44cm perimeter, achieving within 0.02cm of the baseline implementation for all three metrics. It is worth noting that the improved implementation still suffers from the accumulated error in the robots estimated pose that the baseline implementation also suffered from. This highlights the importance of an accurately estimated pose, as the resulting error is still present even with an increased number of sensors since the quality of the pose estimation is not reliant on the proximity sensors. In this case, there has not been a significant improvement in the accuracy of measurement as stated in the hypothesis.

C. 66×66cm

The estimated distances obtained by the improved implementation when mapping the 66cm perimeter are more accurate, producing a sum of absolute differences of 1.17cm compared to the 1.81cm produced by the baseline implementation. When inspecting the results visually, it can be seen that the distances measurements obtained are still noticeably more noisy than the measurements obtained when mapping the smaller problem sizes.

There is a noticeable increase in the under-estimation of measurement, with the total value being 1.05cm. Most of this is accumulated in the measurements of the corners of the perimeter. The effects of the large range of error found in the raw values for each distance measurement as stated in Section VII-C can be seen for this particular experiment too.

IX. Evaluation

By and large, the results obtained have supported the stated hypothesis. As highlighted in Sections VII-C and VIII-C, the results produced by the 66cm box contain a noticeable amount of noise that is not present in the results produced by the other two problem sizes. This is the case for both the baseline and improved implementations. This result corroborates the following claim made in the hypothesis: *"infrared proximity sensors perform with a varying degree of accuracy, determined by the distance being measured"*.

The optimal range of ∼10-40cm for distance measurement using an IR proximity sensor is stated in Section III-B. This is highlighted by the high rate of change in voltage values seen in Figure 1 and the accurate mapping shown in Figure 2. The results of the experiment based on the perimeter size of 44×44cm supports this prediction. Both the baseline and improved implementations produce very good results as the distances required to be measured fall well within the optimal range. There is no evidence of an improvement. Therefore, for this range of distances, the results do not support the claim made within the hypothesis stating: *"a combination of two sensor readings will help to produce better estimations."*

Figure 7a highlights the the following claim in the hypothesis: *"(IR proximity sensors) are prone to a higher degree of error at the limits of their operation"*. It is clear that the constant over-estimation occurring at the closest edges of the smaller box is due to the true distance lying below the accurate $\sim$10cm range which, as stated earlier in Section VII-A, results in an over-estimation as opposed to an under-estimation.

A common theme across the experiments was the relatively worse measurements of the last corner of the perimeter. This has been attributed to the accumulation of error produced by the kinematics-based heading estimation of the robotic system. This supports part of the claim in the hypothesis: *"External factors based on the environment and the system itself can introduce additional error."*. However, due to no concrete evidence, the effect of the environment on the performance of the system is inconclusive.

The key point of this experiment was to get an insight into exactly how the use of a second sensor affected the results produced by the baseline implementation. The two results that highlight the outcome of this attempted improvement were found in the experiments for problem sizes 33cm and 66cm. The second sensor was able to accurately measure the small distances, where were previously over-estimated. Finally, the performance for the larger perimeter, although not perfect, was significantly better using two proximity sensors. As such, it can be said that the main prediction of the hypothesis:

> *"...We hypothesise that with the introduction of a second sensor positioned appropriately, smaller distances can be reliably measured where previously not possible due to an increased combined range of distance covered. Furthermore, a combination of two sensor readings will help to produce better estimations."*

can be confidently supported by the findings of the experiment.

It is worth noting that the environments experimented with are relatively simple, and do not reflect the more complex environments that may need to be mapped in many applications of autonomous robots. Although the Sharp GP2Y0A60SZLF distance sensor performed well at smaller distances, the decrease in accuracy at larger distances was already becoming apparent. Furthermore, the overall range of distance tested within this experiment is very small. This experiment has highlighted the reasoning behind the fact that IR proximity sensors are more commonly used to characterise obstacles in the form of detection, rather than attempting to measure distance.

When mapping a large and complex environment, the use of ultrasound sensors, or RGB cameras may in fact be a better choice than an infrared proximity sensor.

References

[1] Benet Gilabert, G., Blanes, F., Simo, J., and Perez, P. Using infrared sensors for distance measurement in mobile robots. *Robotics and Autonomous Systems* (09 2002).

[2] Durrant-Whyte, H., and Bailey, T. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine 13*, 2 (2006), 99–110.

[3] Sharp. GP2Y0A60SZLF Datasheet. `https://www.pololu.com/file/0J812/gp2y0a60szxf_e.pdf`.