

Lab 5 – Searching for Objects (Group 51)

Alex Bhandari-Young – 260520610, Neil Edelman – 110121860

2013-10-20

Abstract

This lab has two parts. The first is recognizing different types of objects using the color sensor. The second is navigating the field while searching for blocks. These are combined with localization from the previous lab. First the robot localises in a set environment. It then searches for the foam object in a sea of wooden blocks, captures, and brings it to the set destination location in the allotted five minutes.

1 Data and Analysis

1.1 Calibrating

1.2 Object detection

1. “Perform at least 10 trials of object recognition using an object other than the Styrofoam block and note the number of false positives.[1]”

The other-object data is in Table 1. We tested the colour object recognition on wooden blocks (denoted "block" in the table) as well as on the floor and walls of the arena and empty space. The "recognizes" column refers to the assessment made by the algorithm, which reports whether the object in question is more likely wooden or styrofoam. The percentage calculated in the percent column refers to the certainty of the algorithm's assesment. It is apparent that the sensor is very effective, especially at close distances. The arena is made entirely of wood, and we can successfully recognize it as well as the blocks as wooden. There were no false positives when testing the wooden blocks as well as when testing the arena itself, as can be seen from the table. In open space the algorithm is designed to still detect wooden or styrofoam so there are two false positives. However, the sensor is only used when a block is within range as determined by the ultrasonic sensor so this should not be a problem.

Additionally, values with low certainty (close to 50 percent) can be filtered out.

object	cm	recognises	percent
space		wood	50%
space		wood	50%
space		equal	
space		equal	50%
block	50	wood	51%
block	45	wood	53%
block	40	wood	57%
block	35	wood	68%
block	30	wood	74%
block	25	wood	82%
block	20	wood	81%
block	15	wood	88%
block	10	wood	95%
block	5	wood	98%
floor	50	wood	52%
floor	20	wood	84%
floor	10	wood	94%
side	50	wood	76%
side	20	wood	84%
side	10	wood	94%
side	5	wood	99%

Table 1: Space is empty space giving (0,0,0) or very close; it has not been included. The differences can be explained by the orientation.

2. “Repeat the above step, but using the Styrofoam block each time, noting the total number of false negatives. Ideally both errors should be as small as possible.[1]”

The styrofoam data is in Table 2. We used an even greater range of distances when testing on the styrofoam blocks. From the data we can conclude that our algorithm is reliably able to differentiate the styrofoam from about 30 cm. It starts to recognize at distances less than 70cm. Again at close distances the sensor is extremely accurate. There are no false positives at distances less than 70cm. And within the required operating range of this sensor, which is around 30cm and below, the certainty is high.

object	cm	recognises	percent
styrofoam	90	wood	50%
styrofoam	90	styrofoam	50%
styrofoam	90	wood	50%
styrofoam	90	styrofoam	50%
styrofoam	90	wood	50%
styrofoam	90	wood	50%
styrofoam	80	styrofoam	50%
styrofoam	70	styrofoam	51%
styrofoam	60	styrofoam	54%
styrofoam	50	styrofoam	53%
styrofoam	40	styrofoam	59%
styrofoam	30	styrofoam	71%
styrofoam	20	styrofoam	84%
styrofoam	10	styrofoam	97%
styrofoam	5	styrofoam	98%

Table 2: Distance that the colour sensor can differentiate a styrofoam block.

1.3 Searching for Objects

3. “Run through the search program at least 5 times, recording the average time taken to localize, to find a block, and then to travel to the destination. In the final competition, you must complete localization in less than 30 seconds. Also, estimate your localization and final destination errors for each trial.[\[1\]](#)”

Table 3 is from Lab 4[\[2\]](#). It gives the error incurred from in-place localisation. The robot was placed in the first tile facing different angles and localized. The error in theta is calculated. Table 4 gives the time for each step (localising, finding the block, traveling to the destination) and the total time taken. It also gives an estimate on the error when reaching the end; this error is not normally distributed, it is a function of how many collisions the robot took.

θ_{start} ($^{\circ}$)	θ_{final} ($^{\circ}$)	θ_{reported} ($^{\circ}$)	θ_{error} (cm)
45	359	1.8366	-2.8366
0	359	0.9664	-1.9664
315	0	1.6627	-1.6627
270	0	1.8364	-1.8364
225	359	1.4885	-2.4885
180	359	1.9662	-2.9662
135	0	1.8364	-1.8364
90	359	1.0112	-2.0112
345	0	1.0104	-1.0104
15	359	2.1846	-3.1846

Table 3: The error mean is -2.18 cm, variance is 0.46 cm^2 , and the corrected sample standard deviation is 0.67 cm.[\[2\]](#)

localising (s)	finding (s)	travelling (s)	total (s)	error (cm)
15	180	4	199	30
16	220	4	240	5
15	360	7	382	40
14	120	6	140	1
15	320	7	342	3
16	290	4	310	30

Table 4: The error mean is 18 cm, variance is 291 cm^2 , and the corrected sample standard deviation is 17 cm.

2 Error Calculations

The localisation error in Table 3.[2]

Calculate the differences in Equation 1–10.

$$\begin{aligned} d_1 &= ((359) - (1.8366))_{360[-180,180]} \\ &= -2.8366 \end{aligned} \quad (1)$$

$$\begin{aligned} d_2 &= ((359) - (0.9664))_{360[-180,180]} \\ &= -1.9664 \end{aligned} \quad (2)$$

$$\begin{aligned} d_3 &= ((0) - (1.6627))_{360[-180,180]} \\ &= -1.6627 \end{aligned} \quad (3)$$

$$\begin{aligned} d_4 &= ((0) - (1.8364))_{360[-180,180]} \\ &= -1.8364 \end{aligned} \quad (4)$$

$$\begin{aligned} d_5 &= ((359) - (1.4885))_{360[-180,180]} \\ &= -2.4885 \end{aligned} \quad (5)$$

$$\begin{aligned} d_6 &= ((359) - (1.9662))_{360[-180,180]} \\ &= -2.9662 \end{aligned} \quad (6)$$

$$\begin{aligned} d_7 &= ((0) - (1.8364))_{360[-180,180]} \\ &= -1.8364 \end{aligned} \quad (7)$$

$$\begin{aligned} d_8 &= ((359) - (1.0112))_{360[-180,180]} \\ &= -2.0112 \end{aligned} \quad (8)$$

$$\begin{aligned} d_9 &= ((0) - (1.0104))_{360[-180,180]} \\ &= -1.0104 \end{aligned} \quad (9)$$

$$\begin{aligned} d_{10} &= ((359) - (2.1846))_{360[-180,180]} \\ &= -3.1846 \end{aligned} \quad (10)$$

Calculate the sum of the differences (Equation 1–10) in Equation 11.

$$\begin{aligned} \text{sum} &= \sum_{i=1}^{10} d_i \\ &= (-2.8366) + \\ &\quad (-1.9664) + \\ &\quad (-1.6627) + \\ &\quad (-1.8364) + \\ &\quad (-2.4885) + \\ &\quad (-2.9662) + \\ &\quad (-1.8364) + \\ &\quad (-2.0112) + \\ &\quad (-1.0104) + \\ &\quad (-3.1846) \\ &= -21.7994 \end{aligned} \quad (11)$$

Calculate the sum of the differences (Equation 1–10) squared in Equation 12.

$$\begin{aligned}
\text{ssq} &= \sum_{i=1}^{10} d_i^2 \\
&= (-2.84)^2 + \\
&\quad (-1.97)^2 + \\
&\quad (-1.66)^2 + \\
&\quad (-1.84)^2 + \\
&\quad (-2.49)^2 + \\
&\quad (-2.97)^2 + \\
&\quad (-1.84)^2 + \\
&\quad (-2.01)^2 + \\
&\quad (-1.01)^2 + \\
&\quad (-3.18)^2 \\
&= 51.62
\end{aligned} \tag{12}$$

Calculate the mean from Equation 11 in Equation 13.

$$\begin{aligned}
\text{mean} &= \frac{\text{sum}}{N} \\
&= \frac{-21.7994}{10} \\
&= -2.179940
\end{aligned} \tag{13}$$

Calculate the variance from Equation 11 and 12 in Equation 14.

$$\begin{aligned}
\sigma^2 &= \frac{\text{ssq} - \frac{\text{sum}^2}{N}}{N - 1} \\
&= \frac{51.6208 - \frac{(-21.7994)^2}{10}}{10 - 1} \\
&= 0.455492
\end{aligned} \tag{14}$$

Calculate the corrected sample standard deviation from the variance (Equation 14) in Equation 15.

$$\begin{aligned}
\sigma &= \sqrt{\sigma^2} \\
&= \sqrt{0.455492} \\
&= 0.674902
\end{aligned} \tag{15}$$

This is the error on Table 4.

Calculate the sum of the error in Equation 16.

$$\begin{aligned}
\text{sum} &= \sum_{i=1}^6 d_i \\
&= (30) + \\
&\quad (5) + \\
&\quad (40) + \\
&\quad (1) + \\
&\quad (3) + \\
&\quad (30) \\
&= 109
\end{aligned} \tag{16}$$

Calculate the sum of the error squared in Equation 17.

$$\begin{aligned}
\text{ssq} &= \sum_{i=1}^6 d_i^2 \\
&= (30.00)^2 + \\
&\quad (5.00)^2 + \\
&\quad (40.00)^2 + \\
&\quad (1.00)^2 + \\
&\quad (3.00)^2 + \\
&\quad (30.00)^2 \\
&= 3435
\end{aligned} \tag{17}$$

Calculate the mean from Equation 16 in Equation 18.

$$\begin{aligned}
\text{mean} &= \frac{\text{sum}}{N} \\
&= \frac{109}{6} \\
&= 18.166667
\end{aligned} \tag{18}$$

Calculate the variance from Equation 16 and 17 in Equation 19.

$$\begin{aligned}
\sigma^2 &= \frac{\text{ssq} - \frac{\text{sum}^2}{N}}{N - 1} \\
&= \frac{3435.0000 - \frac{109.0000^2}{6}}{6 - 1} \\
&= 290.966667
\end{aligned} \tag{19}$$

Calculate the corrected sample standard deviation from the variance (Equation 19) in Equation 20.

$$\begin{aligned}
\sigma &= \sqrt{\sigma^2} \\
&= \sqrt{290.966667} \\
&= 17.057745
\end{aligned} \tag{20}$$

3 Observations and Conclusions

We did a literature search and discovered that some variant of Kalman filter is used almost exclusively for robot navigation; for example in [3]. We did not use this because of time constraints, our robot following a heuristic path. We implemented a PID-control class. We have a light sensor and a colour sensor, so we could implement odometer correction; however, we wanted our robot to go at all angles anywhere in the field. Odometer correction was not tested with this relaxed constraint.

1. “What differences, if any, did you observe in the behavior/performance of your earlier code (i.e. localization, odometry, navigation) when combined in a larger system? Explain any discrepancies. If it turns out that things worked out pretty much as expected, explain how the design of your code contributed.”

We spent a large amount of time on the structure of our code in previous labs, and sorted out a lot of problems with integrating the odometry and navigation with localisation in Lab 4. Because of this, when we implemented these parts into this lab, things went rather smoothly. However, we were still not happy with the structure of the code, and started on a redesign which is much more intuitive, but we could not get it working reliably in time for this lab so we used the old code from lab 4 that we know worked. In terms of the structure of this code, we solved a lot of our problems from previous labs by abstracting all of the required methods from labs 2 and 3 into one class, the odometer, which contains navigation. Calling `odometer.travelTo()` calls the `travelTo` method in navigation. This made using the code much simpler, but it is still not a very intuitive solution. In the redesign, which we hope to use in the final project, we would like everything contained in a robot controller class, making the code much cleaner.

2. “How reliable was your object detection? What factors influence the reliability of object detection? Where would you expect your code to break down? What steps can you take to make detection more robust?”

The object detection was very solid. We spent a lot of time making it robust and independent of environmental factors such as lighting. We used a method that is $\mathcal{O}(2^{n-1})$, but n is constant and two (styrofoam and wood.) We normalise our colours to make it lighting-independent. We compare with experimental values for the different substances, and pick the closest Cartesian distance to normalised colour values.

The normalisation projects the HSL values onto $L = 0.5$ so it isn’t affected by natural light. The barycentric coordinates are the square of these values. We could have multiplied the eigenvalues by their relative sensitivities, but we didn’t know them; the blue was suspected to be lower when we tested it.

The comparison introduces an error since we are working in normalised spherical co-ordinates. It’s monotonic, which is all we care about when comparing. The approximate error is maximally seen in Equation 21.

$$\int_{x=0}^{\sqrt{2}} \left(\sqrt{1-y^2} - (1-y) \right) dy \approx 0.19 \tag{21}$$

While there are not many situations where the code would break down within the limitations of Lab 5 (short of very drastic changes in lighting, which were not tested), in the final project the introduction of an unexpected object, namely the opposing teams robot, could effect the algorithm. This could be accounted for by testing the sensor on the expected colors of the NXT robot. This should not be a problem as the sensor looks for blue or wood colors, both of which are unlike the colors of the NXT. Unless the opposing team purposely uses blue or wood colors to confuse our robot our code should function properly. Additional thought could be put into dealing with a blue robot, but this seems very unlikely.

We could try turning the flood light on while determining colour.

3. “What aspect of this lab did you find most difficult? What aspect of this lab did you find most surprising or unexpected?”

We had the most trouble in trying to redesign our current code structure to make it easier to use. In transferring the code of lab 1–4[4, 5, 6, 2] to the new design a lot of stuff just didn’t work. We decided to use a different coordinate system from that provided in the previous lab[7]. Because the odometer and navigation were designed to use this system, we had some difficulty and were not able to complete the redesign, and ended up reverting to the old design used in lab 4. This had already been tested, and was more robust. We would like to use the redesigned code because it is more intuitive and thus easier to work with and much smaller code size. The mechanical planning of the robot was fun.

- [5] A. Bhandari-Young and N. Edelman, “Lab 2 odometry (group 51),” *McGill*, 2013.
- [6] A. Bhandari-Young and N. Edelman, “Lab 3 navigation (group 51),” *McGill*, 2013.
- [7] “Iso 80000-2:2009,” International Organization for Standardization, 2009.

4 Further Improvements

The colour detection worked very well. Also the front bumper on the robot designed as a capture system for styrofoam blocks was simple and worked well. These systems could be refined, but we are confident the overall design is good enough to be used in the competition. As mentioned before, we would like a very robust and intuitive robot class that consolidates all of the previous lab code. This will be extremely important when the code gets more complex for the final project. The current version of this code redesign was buggy. However, this is expected to happen when multiple sources are merged. Hopefully with a little more work we can get it reliable.

References

- [1] McGill, 304-211, *Lab 5: Searching for Objects*.
- [2] A. Bhandari-Young and N. Edelman, “Lab 4 – localisation (group 51),” *McGill*, 2013.
- [3] A. J. Davison, “Real-time simultaneous localisation and mapping with a single camera,” in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pp. 1403–1410, IEEE, 2003.
- [4] A. Bhandari-Young and N. Edelman, “Lab 1 wall follower (group 51),” *McGill*, 2013.