Lab: Search and Localize

Submission instructions: Students will work for the first time in their design teams (team list posted on MyCourses). Similar to prior labs, this lab consists of three components: demonstration, code submission, and lab report. Demonstration instructions can be found here. Lab reports and code submission must follow the guidelines established in this handout and for the course. For more information, see the ECSE211SubmissionInstructions.pdf on MyCourses. ***Note*** that as you are now working in design teams, code and report submissions are expected to follow the naming convention Team#_Lab#.zip or Team#_Lab#.pdf as appropriate.

Design objectives

1. Implement a robot with ultrasonic and color sensors for determining the precise location of a 220 ml aluminum can without physical contact.
2. Starting in a known corner, localize to the grid, and perform a search in a prescribed area for a can of specified color.

Design requirements

The following design requirements must be met:

- The system must satisfy the design requirements from Lab 4 with respect to localization, with the following exceptions:
    - You must follow the (**X**,**Y**,**θ**) convention specific to this lab (refer to **Figure 2**).
    - You are free to use either rising or falling edge localization.
    - You are **not** required to provide a way of selecting rising or falling edge.
    - You do **not** have to wait for user input after completing ultrasonic or light localization.
- You must allow for the following data to be entered using either the buttons and LCD display OR an easily modifiable section in your code:
    1. The coordinates of the Lower-Left hand corner of the search region, (**LLx,LLy**).
    2. The coordinates of the Upper-Right hand corner of the search region, (**URx,URy**).
    3. The number defining the color of the target can, **TR**.
    4. The starting corner, **SC**.
    5. **Note**: **LLx**, **LLy**, **URx**, and **URy** are in the range of [0,8], **TR** is in the range [1,4], and **SC** is in the range [0,3].
- Your robot must be able to localize in the starting corner, navigate to the Lower-Left hand corner of the search region, search for and identify the target can, and then navigate to the Upper-Right hand corner of the search region.
- Your robot must complete its demonstration procedure in a maximum of 5 minutes.

Color Calibration

The color assignments of the target can variable, **TR**, are 1: Blue, 2: Green, 3: Yellow, 4:Red Sample images of the 4 can varieties are shown in the Appendix.

Part of the preparation for this experiment is calibrating your color sensor against the sample cans in the lab. Each color corresponds to a distribution of Red, Green, and Blue values. For each can, you need to take several measurements of the RGB values returned by the sensor. These measurement values will be used to build a statistical model for each can's color. Note that since these cans do not have solid colors, you must measure carefully (i.e. avoid labeling and decorative graphics) to obtain a correct sample of the dominant color. Similarly, when you later probe a can to determine its color, you will have to scan at several locations and obtain multiple "hits" of the same color to correctly classify the color.

A particularly simple approach is to model each RGB channel as a Gaussian distribution, i.e. empirically compute the mean and standard deviation for the Red, Green, and Blue channels for each can from the measurement samples. Therefore, each can can be represented by its mean RGB values. A simple means of classifying each can is to compute the Euclidean distance, d, between the RGB values of a measured sample, s_R, s_G, and s_B, and a target can's mean RGB values, μ_R, μ_G and μ_B, and. A can is then deemed to be the match when the minimum Euclidean distance value is met.

$$d = \sqrt{(s_R - \mu_R)^2 + (s_G - \mu_G)^2 + (s_B - \mu_B)^2}$$

However, in a real-world scenario, this approach may not work accurately. Given the Gaussian distribution for each channel, an easy way to distinguish can colors from each other is to use their standard deviations, i.e. σ_R, σ_G, and σ_B, precomputed from the measurement samples. For example, if a sample distance is greater than 2 or 3 standard deviations from its mean, that can is probably not the target. This method uses a probabilistic measure by introducing a level of uncertainty when dealing with the color sensor samples.

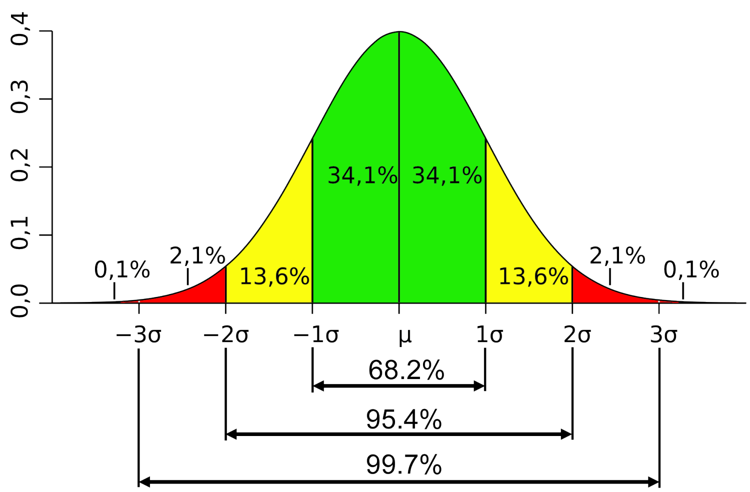

Figure 1. Normalized Gaussian distribution: 68% within ± 1σ, 95.4% within ± 2σ.

In computing your mean statistics, i.e., R_{mi}, G_{mi}, B_{mi}, R_s, G_s, and B_s, variations will creep in due to changes in ambient lighting. One way of mitigating this is to normalize the mean in a particular channel by the average over all 3 channels, shown for R_s, G_s, and B_s below,

$$\hat{R}_s = \frac{R_s}{\sqrt{R_s^2 + G_s^2 + B_s^2}}, \hat{G}_s = \frac{G_s}{\sqrt{R_s^2 + G_s^2 + B_s^2}}, \hat{B}_s = \frac{B_s}{\sqrt{R_s^2 + G_s^2 + B_s^2}},$$

and similarly for R_{mi}, G_{mi}, and B_{mi}.

Demonstration (30 points)

The design must satisfy the requirements by completing the demonstration outlined below.

Design presentation (5 points)

Before demoing the design, your group will be asked some questions for approximately 5 minutes. You will present your design (hardware and software) and answer questions to test your understanding of the lab concepts. Grades will apply to the entire group, although TAs reserve the right to grade individually if they deem it necessary.

You must present your workflow, an overview of the hardware design, and an overview of the software functionality. Visualizing software with graphics such as flowcharts is valuable.

Color Classification (5 points)

The TA will check whether your robot can correctly classify the Blue, Green, Yellow and Red cans. When any object is brought close to your sensor(s), the LCD must display "*Object*

Detected" on the first line. The next line should read "Blue", "Green", "Yellow" and "Red" for a target can. Nothing should be displayed on the LCD if no object is detected. You can decide the distance between the tested can and the sensor. This color classification should not take more than **10 seconds per trial** for **5 successive trials** using the cans in the lab. False positives and negatives are considered as incorrect. Hence, the following points are awarded:

- **1 point** for each time the robot correctly classifies a can (i.e. "*Object Detected*" and *color*).

Field Test (20 points)

The field test will be set up on an 8x8 grid as shown below in **Figure 2**. Note that as of Lab 5 we are using a coordinate system with origin as shown (i.e. shifted left and down by one square from the one used in the previous experiments).

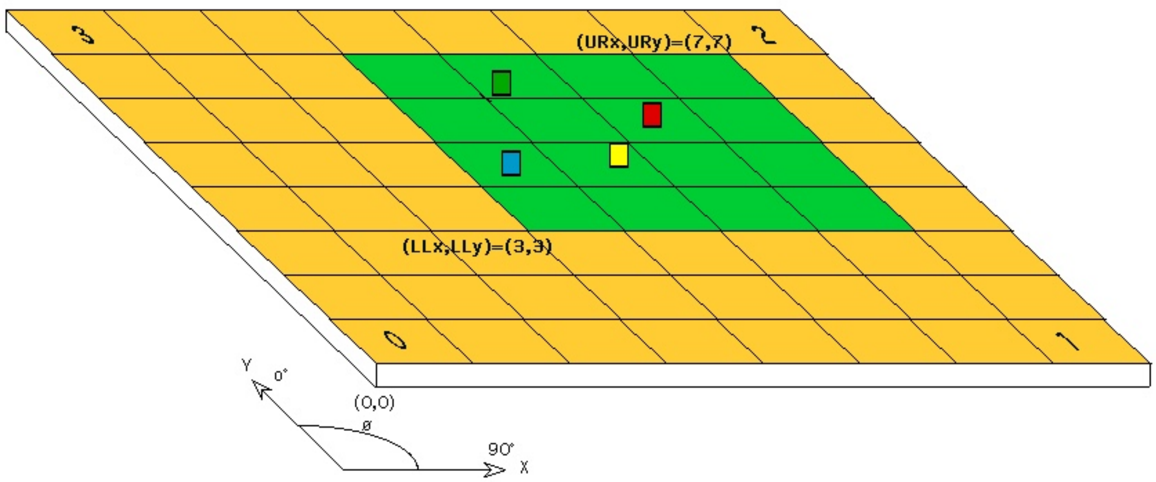

Figure 2. Search and Localize demo setup.

For a successful field test, the following steps must be followed:

- Enter the parameters given to you, i.e. (**LLx,LLy**), (**URx,URy**), **TR**. **SC** is set to 0.
- Place your robot in a starting corner according to **SC** similar to Lab 4.
- Start your robot upon a button push.
- Your robot must proceed to localize, navigate to the Lower-Left hand corner of the search zone, i.e. (**LLx**,**LLy**), and BEEP when it reaches there.
- The robot must then search for the target can specified by **TR**.
  - If the target can is found, the robot stops and BEEPS ONCE.
  - If a non-target can is found, the robot stops and BEEPS TWICE.

  - All the cans in the search region should be identified until the target is found.
  - There may be more than one can in the search region.
- The robot then proceeds to (**URx**,**URy**) and stops.
- All these steps must be completed within 5 minutes.

Hence, the following points are awarded for the field test:

- **5 points** are awarded for navigating to (**LLx,LLy**) within approximately half a tile.
- **10 points** are awarded for correctly identifying the target can and not misidentifying other cans. **5 points** are awarded for partial success as deemed by the TA.
- **5 points** are awarded for navigating to (**URx,URy**) within approximately half a tile.

We recommend referring to the **FAQ** section for details on some specific limitations.

Provided materials

Sample code

No sample code is provided for this lab.

Physical material

The four target cans in Blue, Green, Yellow, and Red. An 8x8 grid floor will be used for this demonstration.

Implementation instructions

The implementation of this lab is at your discretion. Since this lab will be done in conjunction with the entire design team, you have the opportunity to explore different solution approaches.

Report Requirements

The following sections must be included in your report. Answer all questions in the report and copy them into your report. For more information, refer to ECSE211SubmissionInstructions.pdf. Always provide justifications and explanations for all your answers.

Section 1: Design Evaluation

You should concisely explain the overall design of your software and hardware. You must present your workflow, an overview of the hardware design, and an overview of the software functionality. You must briefly talk about your design choices before arriving at your final design. Visualizing hardware and software with graphics (i.e. flowcharts, class diagrams) is valuable.

Section 2: Test Data

This section describes what data must be collected to evaluate your design requirements. Collect the data using the methodology described below and present it in your report.

Model Acquisition *(4x10 independent trials)*

- Collect a minimum of 10 RGB color values for each **TR** (i.e. Blue, Green, Yellow and Red).
- The can-to-sensor distance should be within the working range of the sensor (which you need to determine). You should ensure to sample different points on the surface.

Color and Position Identification *(4 independent runs)*

- Set the following parameters: (**LLx**,**LLy**) = (3,3), (**URx**,**URy**) = (7,7), **SC** = 0 as in **Figure 2**.
- Place all 4 cans in the search region.
- Run the field test 4 times for 4 different **TR** values:
    - There are 1 target and 3 non-target cans in each run.
    - In each run, record the sample RGB values (s_R, s_G, s_B) for all the identified cans (target or non-target). These are used to evaluate your classifier's performance.
    - In each run, record the closest grid intersection (**TPEx**,**TPEy**) to the target can as estimated by your code. Note the actual (real) position (**TPRx**,**TPRy**) in tile units.
    - Hint: you can write your measurements to the LCD or remote console program.

Section 3: Test Analysis

Model Acquisition

- Compute the RGB mean and standard deviation values for each of the 4 sample **TR** sets.
- For each sample **TR** set, plot the corresponding RGB normalized Gaussian distributions and draw vertical lines at μ, $\pm 1\sigma$ from μ, and $\pm 2\sigma$ from μ.
- Ensure that your plots are labelled and captioned in sufficient detail.

Color and Position Identification

- For each field test run, compute the Euclidean distance, d, of all recorded RGB values. This should result in 4 distance values for each recorded sample. Then, rank order the distances for each can index in ascending order. For each entry in the rank order table, list the classification that your algorithm returns (i.e. was a Red can detected as Red?).
- Precisely describe the method you used to determine can position (**TPEx**,**TPEy**).
- For each classification, regardless if it was correct or not, calculate the Euclidean distance between the estimated position (**TPEx**,**TPEy**) and real position (**TPRx**,**TPRy**).

Section 4: Observations and Conclusions

- Are rank-ordecan Euclidean distances a sufficient means of identifying can colors? Explain in detail why or why not.

- Is the standard deviation a useful metric for detecting false positives? In other words, if the can color determined using the Euclidean distance metric, d, is incorrect, can this false positive be detected by using $\mu \pm 1\sigma$ or $\mu \pm 2\sigma$ values instead?
- Under what conditions does the color sensor work best for correctly distinguishing colors?

Section 5: Further Improvements

- Depending on how you implemented your color classifier, can your results be improved by using one or more of the noise filtecan methods discussed in class?
- How could you improve the accuracy of your target can's position identification?

Frequently asked questions (FAQ)

1. **Can we demo the each separately if our localization or navigation is inaccurate?**
   No, all parts must be performed in succession for the field test run.

2. **Do we have to follow the (X,Y,θ) convention specified in Figure 1?**
   Yes. Also, note that the bottom-left corner of **Figure 2** is specified as (0,0).

3. **How large can the search region be?**
   Each side of the search region has a range of [2,6] tiles. The smallest search region is 2x2 tiles and the largest is 6x6. The search region can be positioned anywhere on the 8x8 grid.

4. **How many cans will be there in the search region?**
   There will be at least 2 cans and at most 5 cans in the search region. Only 1 can will have the target color in the search region.

5. **What is the minimum separation distance between two cans in the search region?**
   2 cans are separated by a distance of at least 1 can width (i.e. 10 cm).

6. **What happens if the field test run takes longer than 5 minutes?**
   The TA will conclude your demo at the 5-minute mark and will award 0 points for all the remaining, incomplete parts.

7. **Can the cans be placed anywhere within the search region?**
   No, cans will be placed only at grid intersections.

8. **Do we have to demo position identification as part of the demo?**
   No, but you will have to report this in your write-up.

Appendix: Images of 220 ml cans used in this experiment

GREEN

BLUE

YELLOW

RED