Spring 2023

Mid-Term Exam Report

Perception for autonomous robots

March 16 2023

Student:
Rishikesh Jadhav

Instructors:
Dr. Samer Charifa

Course code:
ENPM 673

Contents

1 Problem 1

This problem consists of 2 tasks -

1. 1.A. List of perception sensors used for a wall painting robot with the descriptions.
2. 1.B. List of perception sensors used for an underwater robot used for fixing a broken pipe with the descriptions.

1.1 1.A

To design a robot for indoor wall painting tasks, it is crucial to include perception sensors that can provide the robot with important information about its surroundings. These sensors are necessary for enabling the robot to navigate its environment, avoid obstacles, and carry out painting tasks efficiently. According to me the following sensors are required for the task provided:

1. **LIDAR**: This sensor uses laser light to create 3D maps of the environment and measure distances, enabling obstacle detection and mapping.
2. **RGB-D camera**: The RGB-D camera can be a useful sensor for a wall painting robot, providing valuable information for navigation, obstacle avoidance, object detection, and color sensing.
3. **Ultrasonic sensors**: These sensors use high-frequency sound waves to detect the distance to objects in the environment, enabling obstacle detection and navigation.
4. **Pressure sensors**: These sensors can detect the pressure applied by the robot's end effector to the wall, allowing the robot to adjust its position and pressure to ensure a consistent paint job.
5. **Force/torque sensors**: These sensors can detect the force and torque applied by the robot's end effector to the wall, enabling the robot to apply the correct amount of force and adjust its position to ensure a consistent paint job.

1.2 1.B

When designing a robot for fixing an underwater broken pipe, it is essential to consider the challenges posed by the underwater environment.The robot needs to be able to operate effectively in any underwater water environment and carry out repairs with precision. To accomplish this,the following perception sensors are necessary to help the robot navigate, avoid obstacles, and perform its tasks successfully.

1. **Depth sensor:** Measures the water pressure and depth, enabling the vehicle to maintain a desired depth and avoid obstacles.
2. **Altitude and forward-looking sonar:** Detects the presence of obstacles and distance from the seafloor, providing important information for navigation and obstacle avoidance.
3. **Doppler velocity log (DVL):** Estimates the vehicle velocity relative to the seafloor and relative water motion, enabling precise navigation and control.
4. **Vision systems (cameras):** Performs tasks such as visual tracking of pipelines, station keeping, visual serving, or image mosaicking. Cameras provide visual information to the vehicle's control system, enabling it to perform a range of tasks such as monitoring the surrounding environment and performing inspection and maintenance tasks.

5. **Magnetic sensors:** Can locate metal pipes or parts of pipes that may have come loose or fallen off. They can also be used to attach the robot to the pipe or hold it in place while performing repairs.
6. **Tactile sensors:** Can detect the texture and hardness of the pipe's surface, which can help the robot determine the extent of the damage and the best way to fix it.
7. **Pressure sensors:** Can detect water depth and changes in pressure, which could indicate the presence of a leak or crack in the pipe. This information is crucial when positioning the robot.
8. **Chemical sensors:** Can detect hazardous materials, such as oil or gas, that may have leaked from the broken pipe. This information can help the robot avoid danger and determine the best way to fix the pipe.

2 Problem 2

In this question the objective is to detect a ball in a video using the Hough transform. The width of the ball is given to be around 11 pixels. The Hough transform is a popular technique for detecting shapes in images and videos, and it is particularly useful for detecting circular shapes such as the ball in this scenario. The objective was to automatically detect the ball in the video using the Hough transform, which involved identifying the parameters that best described the circular shape of the ball in the video. The solution required careful selection of parameters and fine-tuning of the Hough transform algorithm to achieve accurate detection of the ball in the video.

2.1 Pipeline

1. Read video frame by frame
2. Grayscale the frame.
3. Apply median blur.
4. Blur the image using gaussian blur.
5. perform Hough transform using HoughCircles function.
6. Detect and draw circles on the images for visualization.
7. Store the x,y co-ordinates of the ball center.
8. Plot trajectory of the ball.

2.2 Results

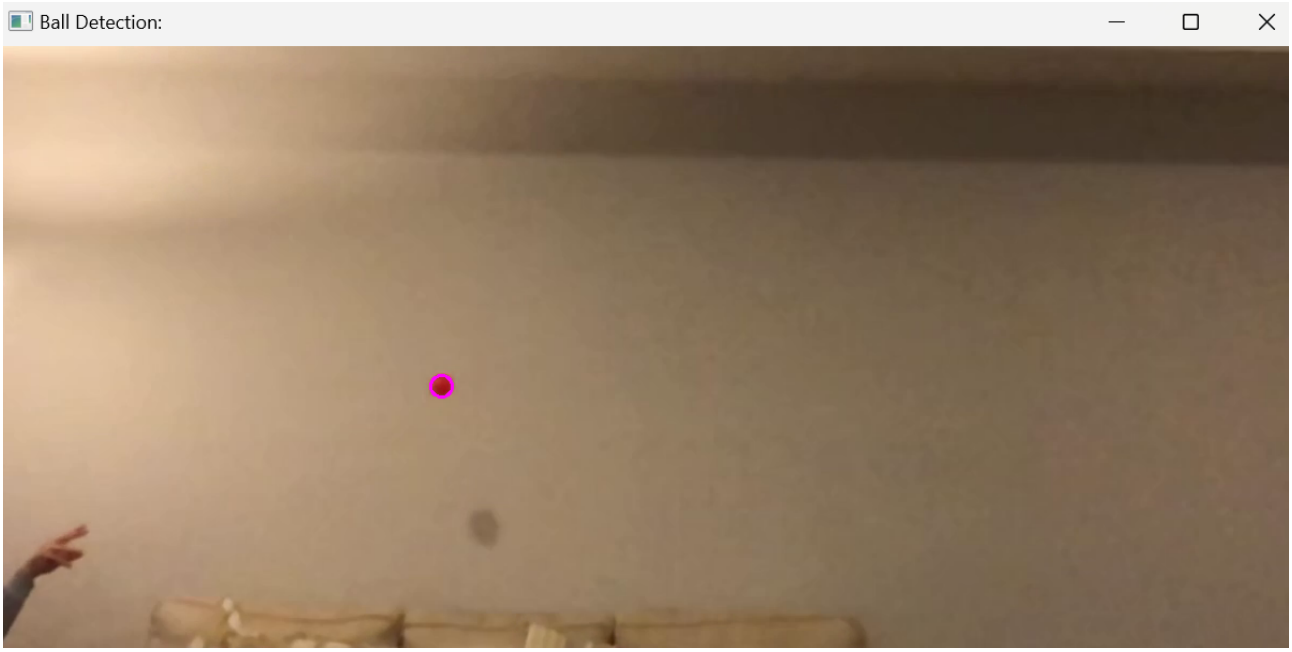

Figure 1: Ball Tracking

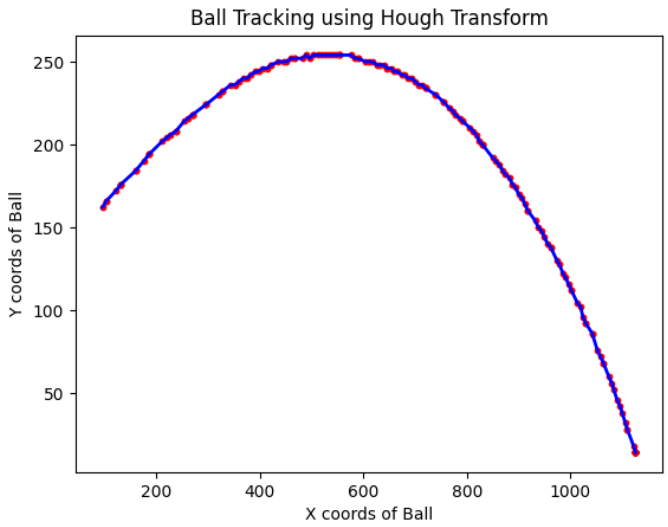

Figure 2: Trajectory PLot

3 Problem 3

The problem required transforming an image of a train track to obtain a top-down view, followed by measuring the distance between the tracks in each row of the transformed image and calculating the average distance between them.

The solution involved using image processing techniques like perspective transformation, edge detection, and Hough transform to achieve the desired result.

The intermediate results such as the warped image with edges and the detected lines are also displayed to provide insight into the process.

3.1 Pipeline

1. Read the input image.
2. Define the source and destination points for perspective transformation to get bird eyes view.
3. Compute the transformation matrix using cv2.getPerspectiveTransform().
4. Warp the image perspective using cv2.warpPerspective().
5. Convert the warped image to grayscale
6. Apply a Gaussian blur using cv2.GaussianBlur().
7. Apply a median blur using cv2.medianBlur() to further reduce noise.
8. Detect edges in the blurred image using cv2.Canny().
9. Use the Hough transform to detect lines in the edge-detected image using cv2.HoughLines().
10. Iterate over the detected lines to calculate the distance between parallel lines using their respective rho and theta values.
11. Finally calculate the average distance between the parallel lines.

3.2 Results

Average distance between the train tracks:295.39999728215537

Here is the transformation matrix:

$$\begin{bmatrix} -2.64187867 \times 10^{-1} & -1.52641879 \times 10^{-1} & 5.45694716 \times 10^{2} \\ -9.60733840 \times 10^{-16} & -1.07436399 & 1.18180039 \times 10^{3} \\ -1.15653902 \times 10^{-18} & -1.03718200 \times 10^{-3} & 1.00000000 \end{bmatrix}$$

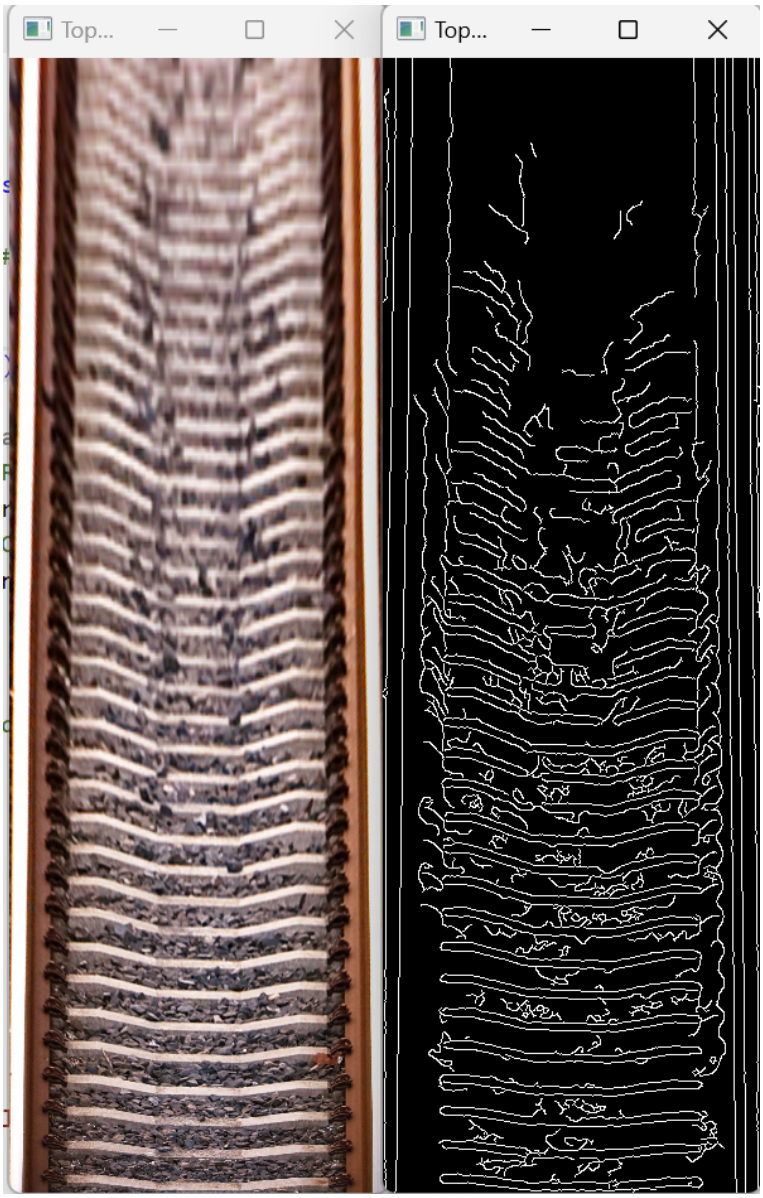

Figure 3: Left : warped and straightened Right : Warped image after canny edge detection

4 Problem 4

The given problem is to detect each hot air balloon in a given image of hot air balloons and to find the number of balloons automatically. The end result should displays each hot air balloon labeled with different color bounded rectangles.

This problem involved several steps, including image preprocessing, contour detection, and object labeling. The objective was to automatically detect and label each balloon in the image, which can be a challenging task due to variations in color, size, and shape of the balloons.

The solution involves using various image processing techniques to preprocess the image and detect the balloons, followed by labeling them with different colors for visualization purposes as described in the pipeline above.

4.1 Pipeline

1. Read the image.
2. Grayscale the image.
3. Apply median blur.
4. Blur the image using gaussian blur.
5. Perform binary thresholding to obtain a binary image.
6. Find the contours in the binary image using the findContours() function.
7. Retrieve all of the contours and compress horizontal, vertical, and diagonal segments and leaves only their end points.
8. For each contour, compute a bounding rectangle using the boundingRect() function.
9. Check if the bounding rectangle has a width and height greater than 100 pixels. If it does, draw a rectangle around the contour using the rectangle() function with a randomly selected color and a thickness of 4 pixels.
10. Count the number of rectangles drawn to obtain the number of balloons in the image.
11. Print the number of balloons in the terminal.
12. Save the resulting image using the imwrite() function.

4.2 Results

The number of balloons that we detected :- 16

Figure 4: Hot Air Balloon Detection

5 Problem 5

The problem at hand is to apply the k-means clustering algorithm to a given set of numbers and obtain three clusters. In this scenario, we have eight numbers, and we want to group them into three distinct clusters. The objective is to apply the k-means algorithm to these numbers and obtain three distinct groups or clusters, with each number assigned to one of the clusters.

The solution involves several steps, starting with initializing the k-means algorithm with a predefined number of clusters (in this case, three). We then need to assign each of the data points to one of the three clusters based on their proximity to the cluster center. We repeat this process of assigning data points to clusters and updating the cluster centers until convergence, which occurs when the assignments no longer change.

5.1 Explanation

1. Given points = $[25, 13, 2, 11, 4, 6, 15, 22]$
2. Choose 3 random centers for 3 clusters ($c1 = 7, c2 = 10, c3 = 14$)
3. Calculate Distance of all points from centers. $Dist.c1 = [18, 6, 5, 4, 3, 1, 8, 15]$, $Dist.c2 = [15, 3, 8, 1, 6, 4, 5, 12]$, $Dist.c1 = [11, 1, 12, 3, 10, 8, 1, 8]$
4. Assign points based on distance to nearest center. $c1 = [2, 4, 6]$, $c2 = [11]$, $c3 = [25, 13, 15, 22]$
5. Calculate new centers from the new lists by taking mean. $c1.new = (2+4+6)/3 = 4$, $c2.new = 11$, $c3.new = (25 + 13 + 15 + 22)/ = 18.75$
6. Repeat step 3 and assign points based on nearest center.
7. New Assignment list = $c1.new(4) = [2, 4, 6]$, $c2.new(11) = [13, 11, 15]$, $c3.new(18.75) = [25, 22]$
8. Take average of the new assignments to get new centers. $c1.new2 = (2 + 4 + 6)/3 = 4$, $c2.new2 = (13 + 11 + 15)/3 = 13$, $c3.new2 = (25 + 22)/2 = 23.5$
9. After Repeating the above steps of Finding distance to centers, assignment to groups and finding new centers again, we get the same centers.
10. Hence the final Center values are $C1 = 4$, $C2 = 13$, $C3 = 23.5$.

6 Problems Encountered and solutions

Following road blocks were found during the entire process of completing the project:

1. 1 - In the first question finding the correct set of combinations of sensors was a bit tricky, after reading multiple blog posts and revisiting the lecture slides I was able to find a right set of Sensors to do the jobs.
2. 2 - This Problem was fairly simple as we had worked on this in Project1.
3. 3 - This problem was by far the toughest one of the lot even though the pipeline seemed easy but I face issues during the implementation of the pipeline as I faced warping issues and finding the correct set of source and destination points and average distance between parallel lines was tough.
4. 4 - In this problem the tricky part was the overlapping hot balloons part but the process was fairly straight forward
5. 5 - I was able to solve this problem very quickly as Professor had taken a similar example in class and solving this Problem clarified the complete concept of k Means