

ME 4451 Lab 4 Handout

Machine Vision

Anirban Mazumdar, Bryan Blaise, Nikhil Nandikanti
(1 Session Lab)

Spring 2024

1 Objectives

1. Experience the capabilities and limitations of simple web camera in the use of robotics.
2. Make yourself familiar enough with the vision system, so that you could use it for future projects.

2 Prelab Assignment

- Read the Lab 4 Handout.
- Familiarize yourself with the concept of circularity.
- Create a completed script and accompanying functions to execute the tasks of this lab.
- Ensure necessary Matlab packages are installed as defined in Section 5. Attempt to use your laptop's webcam to ensure it works during your lab section.

3 Instructions and Tasks for This Lab

The focus of this lab is the development of a “blob” analysis algorithm, using functions built into the Image Processing Toolbox (IPT) in Matlab. Since Matlab can take a long time to perform calculations, one must try to reduce the processing time through smart and efficient code. The script should read in an image, identify shapes, display their properties, and track and plot them.

4 Vision Algorithm Design

Your first task is to come up with the outline of your vision algorithm script. This can be hand written or done using Matlab comments in a shell script you create.

- Read the “List of Functions” section of this manual, and use the “help” feature in Matlab to understand what each function does. For specific examples and usages, Google them online for more detail.
- Determine which functions to use, and in what order, to create a blob analysis program. See the “Blob Identification Algorithm” section for a generalized algorithm.
- Don’t try to manipulate the actual image matrices manually, as the output may not be in the correct number format. Use built in Matlab functions instead.
- Brainstorm as a team and outline your algorithm.
- Show the algorithm outline to the TA for preapproval before proceeding. Be prepared to explain the method you used for distinguishing each shape

5 Camera Setup/Image Capture

- You will be using your own webcam on your laptops. Make sure you have MATLAB Support Package for USB Webcams installed and ready to use.
- If you don’t have access to your webcam, make sure you let the TA known 24 hours before your lab section. You will use the webcam on the remote desktop.
- Once you have MATLAB Support Package for USB Webcams installed, check to see if you can connect to your webcam: [Click Here](#).
- Feel free to play around with the functions to get comfortable. Detailed guide: [Click Here](#).

6 Algorithm Implementation

Now you will actually write and test your algorithm.

6.1 Blob Identification

- Download the test blob images (Blobs.zip) from Canvas. This file contains pictures of various shapes. You do not need to print these. You can load the images directly to Matlab as jpegs.
- Write a function (“blob_id”) that will read in the blob image and detect one of the shapes in the image. Focus on identifying the circle and move on to the next section. Once you have completed the lab, you can come back to identify other shapes.
- Design your algorithm and check with a TA before writing too much code. Test your code in the smallest possible chunks.

- Your algorithm should:
 - Identify shapes by drawing boundaries around them in the image image. Draw a different boundary color for different shapes (i.e. yellow for circles, pink for rectangles).
 - Plot the centroid of the shape in the same color as the boundary
 - Be able to do this reliably across multiple images. Your TA will specify which images you need to identify shapes in at the beginning of your lab session.s
 - You cannot simply hard code a measured property (i.e. number of pixels in the blob = x, therefore this must be a triangle) to identify the shape. Your methods should be analytical. But you may add a reasonable tolerance to calculations since pixellated shapes are not perfect.
 - Make whatever method you choose both scale and rotation invariant

6.2 Object Tracking

- Once blob recognition works, implement blob tracking. This involves following and highlighting a specific object using color identification as you move it in front of the camera.
- Use the Color Thresholder app in the Image Processing toolbox to track a color live by grabbing images from the camera.
- Step 1: draw a boundary around some object of a certain color in a still frame.
- Step 2: draw a boundary around an object of a certain color in continuous frames.
- Step 3: make sure the boundary you draw is around the *same* object from frame to frame.
- You'll likely be most successful thresholding for the color in the HSV color space

Show the outputs of your code to the TA.

7 List of Possibly Useful Functions

The following is a list of functions (or code statements) from Matlab and the Image Processing Toolbox that should be useful to you for this lab.

Note: Because most of these functions can accept multiple combination of inputs, including optional inputs, this list does not contain input syntax. Some suggested inputs are provided. There are other functions that may be useful as well.

- `bwboundaries()`
- `bwlabel(—, 8)`
- `im2bw()`
- `imcomplement()`
- `imfill()`
- `imread()`
- `imshow()`
- `regionprops—, 'all'` (Useful properties: 'Centroid', 'Area', 'Perimeter', 'PixelIdxList', 'Circularity', 'MajorAxisLength', 'MinorAxisLength', 'MaxFeretProperties', 'MinFeretProperties')
- `rgb2gray()`
- `clc`
- `clear all`
- `figure`
- `hold on`
- `hold off`
- `size()`
- `length`