

Homework 2

16-311: Introduction to Robotics

Contents

1	Learning Objectives	1
2	Filters	2
3	Convolution	2
4	Determining Distance Geometrically	2
5	Stereo Vision	3
6	Neural Net	4
7	What To Submit	5

1 Learning Objectives

1. Gain experience creating and using filters.
2. Improve understanding of convolution.
3. Practice geometric techniques for finding distance using similar triangles and stereo vision.
4. Use a neural net to classify images.

2 Filters

In this section, we will be discussing filters (also known as masks or kernels). Filtering an image can be useful to average the value of a group of pixels and reduce noise or missing data in the image file.

1. Come up with a three-by-three filter that can take the average of the nine pixels it overlaps with.
2. Come up with a three-by-three filter that takes the average of the neighbors of the center pixel and the center pixel by the 4-point connectivity definition of neighbor.
3. Come up with a slightly more trusting three-by-three filter that gives a weighted average of the 9 pixels that it would overlap with, with more weight on the center pixel. Please make sure all the weights sum to 1.
4. Is it possible to recover the original image if we know the filter that was used?

3 Convolution

For this problem you are going to convolve a small 2x2 image with a 3x3 filter by hand. Please do not use something like the MATLAB function `convolve`. You want the output image to be the same dimensions as the input image. In order to do this, reflect the edges of the image so that the image is extended. Then convolve the image with the given filter.

Image:

0	200
100	10

Filter:

0	0.1	0
0.1	0.6	0.1
0	0.1	0

4 Determining Distance Geometrically

Suppose you have a robot with a camera that you need to characterize. This robot has been in the lab for a while so there is little documentation available and the sensing box has been welded shut so you cannot measure the camera's geometry.

You want to find the distance between the lens and image sensor (some sources call this the focal length) so you can use that for future experiments to judge distance.

You take a picture of a one-inch square cube from a precise distance of one foot with the cube centered in the image. You threshold and segment the image and count the number of pixels that represent the cube. You repeat this process 10 times and get an average of 47 pixels for the width of the one-inch cube. You read someone's well-documented code for displaying the images from this camera and see that the camera's resolution is 100 ppi (pixels per inch).

Using this information, what is the distance between the lens and image sensor? Please include at least one symbolic equation that you used with your answer.

5 Stereo Vision

The goal of this section is to practice implementing the geometry involved in determining depth from an image and position from stereo cameras. The image below shows a robot with two forward-facing cameras. The cameras are 10 cm apart and you can simplify the cameras as pinhole cameras with 50 mm between the camera opening and image sensor.

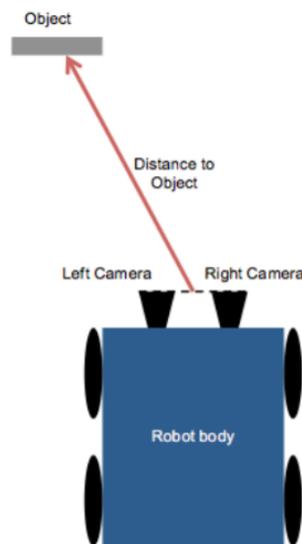


Figure 1: Robot and object. Not to scale.

Two frames from the camera pair are shown below. The object is 30 pixels left

of the center of the image taken by the left camera and 50 pixels to the left of the center of the image taken from the right camera. The resolution of the camera is 20 pixels per cm.

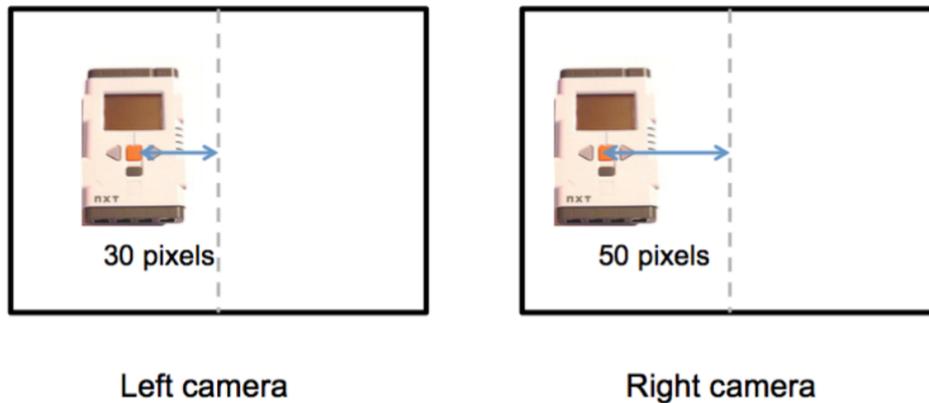


Figure 2: Images from left and right cameras.

In your writeup, find the distance from the center of the front of the lenses to 4 the center of the front of the object (realize that this is not just a single-dimensional problem). Show your work. At a minimum, write down two different symbolic equations showing the geometry used to determine the distance.

6 Neural Net

The explosion of machine leaning has lead to the creation of dozens of machine learning tools. In this question, you will use MATLAB's Deep Learning Designer (pre-installed in MATLAB 2018b).

The goal of this question is to create training data, implement a neural net, and explore the effect of changing some of the parameters for classifying 5x5 images as Waldos or boxes.

Below are larger sample images of each.



Figure 3: Sample images: Waldo and box.

1. First, create training/testing data in a folder called 'images'. About half of your images should be based off the simplified Waldo and about half should be based on the box. You can use any functions you want to create these images. Divide these images in two folders, one labeled 'waldos' and one labeled 'boxes'. The names of the individual images doesn't matter. We will use your images as training data with test images that are based off the example images. The test images will be based off the two sample images but may be burred or have noise. You should not need to make more than 20 images of each category. You will zip up your images folder and submit to Autolab.
2. Next, create a simple neural net to classify the images. Start from this code and specify the layer characteristics: <http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/16311/www/current/hw/hw2/basicNeuralNet.m> Record your accuracy with the given parameters in the .pdf submission.
3. Next, note the accuracy with 7 epochs, instead of 4.
4. Does increasing the number of epochs always help increase accuracy? Explain in one sentence.
5. Choose 5x5 for the filter size. How does this affect accuracy?
6. Finally, try an activation function other than ReLU. How does this affect accuracy?

This tutorial: <https://www.mathworks.com/help/deeplearning/examples/create-simple-dee.html> provides a good overview of the process.

7 What To Submit

Submissions are due on Autolab/Gradescope by the date specified in the Syllabus.

1. Create a .pdf file with the written answers ALL THE SECTIONS named hw2.pdf and submit to Gradescope.
2. Zip your images folder from Section 6 and submit to Autolab.