CSci 4270 and 6270
Computational Vision,
Spring Semester, 2025
Lecture 04 - 06 Practice Exercises

Overview

1. Please see the lecture 01 and 02 practice problems for an overview.

2. Some of these questions were on Quiz 1 last spring.

Problems

1. Convert a line in slope-intercept ($y = mx + b$) form to the implicit form and use this to compute the minimum Euclidean distance of point (x_0, y_0) to the line, expressed in terms of m and b.

2. Given an $N \times 2$ array of points, write code to find and print the values of a, b and c for the orthogonal least squares best fitting line. Resolve the sign ambiguity of (a, b) by making sure that a is positive. (For the sake of time, don't worry about the possibilty that a is 0). Please, no for loops.

3. If $\mathbf{H}$ is a 3x3 transformation matrix representing a 2d transformation and $\tilde{\mathbf{x}} = (u, v, 1)^\top$ is the homogeneous representation of a point, show that $s\mathbf{H}$ maps $\tilde{\mathbf{x}}$ to the same point location for all $s \neq 0$.

4. Why does smoothing require the sum of the weights of the convolution kernel to be 1.0?

5. Consider boundary effects during convolution. When computing a derivative — for example with the Sobel kernels as we did in HW 2 — what happens when we consider all values outside the image as 0? Why might it be better to just assign the value of the convolution to be 0 at boundary pixels? (Note, if I asked this on the quiz I would give you the Sobel kernels. You don't need to memorize them.)

6. Given a 3x3 array how can you determine the simplest type of 2d transformation it describes? Is it rigid? Is it a similarity? Is it affine? Or, is it projective? Write a short python script that takes a 3x3 array and outputs one of five different words: `rigid`, `similarity`, `affine`, `homography` or `none`. You may assume that the last entry (lower right corner) of the array is 1. Ignore numerical issues such as round-off errors. Note that if the matrix is not full rank then `none` is the correct answer. You can use the determinant (`np.linalg.det`) to check rank.

   As examples, consider for these five example arrays

```
1.4  0.6 -1244
-0.6 1.4 989
0  0  1

2.6  0.6 244
-0.6 1.4 90
0  0  1
```

```
 0   0   114
 0   0   255
0.1 0.4    1

 0.82533561  0.56464247  163
-0.56464247  0.82533561  -45
  0          0            1


 1    0   114
 0    0   255
0.01 0.04    1
```

the outputs should be, in order,

```
similarity
affine
none
rigid
homography
```

7. Intuitively, why do the Sobel kernels smooth the image in addition to computing the derivative?

8. A camera is in the world at location $X = 0$, $Y = 1$, $Z = 2$ (in meters), its optical axis is aligned with the Z axis, and its focal length is 50 mm (0.05 meters). Consider a point `p` at location $X = -1$, $Y = 3$, $Z = 7$ (in meters). Showing your work on this problem is very important.

   (a) Find the projection of `p` onto the image plane in millimeters.

   (b) If the optical axis intersects the image plane in the center of a `4000 x 6000` (rows x columns) pixel image and the pixels are 20 microns (0.02 millimeters) on a side, what is the pixel location of `p` in the image?

9. Given are the following binary image (left) and structuring element (right), where blank pixels and pixels outside the shown grid have value 0.

		1	1						
	1					1	1		
	1	1					1		
	1		1				1	1	
	1	1	1						
			1	1	1				

	-1	0	1
-1	1	1	1
0	1	1	1
1	1	1	1

(a) How many 8-connected components are in the binary image?

(b) Show the result of morphological closing (dilation followed by erosion) with the given structuring element. To do this, simply put an **X** through the pixels **(in the figure above!)** that change from 0 to 1 or 1 to 0 as a result of the closing (you don't need show the dilation result).

(c) How many 8-connected components are there after the closing is applied? As part of your answer, circle each of the connected components in the above figure.

(d) Can a closing **ever** increase the number of connected components? Why (briefly)?

10. Give two reasons why the change detection example from the end of Lecture 6 is too naive for non-artificial change detection problems.

11. Write NumPy code that, given a gray scale image `im`, computes the following:

(a) Sorts the intensities. Call the resulting array `val` — short for "values". (Note that `np` has a sort function.)

(b) Finds the grayscale values `g0` and `g1` such that 1/3 the pixels have gray scale values less than or equal to `g0` and 2/3 of the pixels have gray scale values less than or equal to `g1`.

(c) Creates a new image called `im_n` where all intensities less than or equal to `g0` are mapped to 0, all intensities greater than `g1` are mapped to 255, and every other intensity is unchanged.

To illustrate, in the example below, the image `im` is shown on the left, `g0 == 21` and `g1 == 41`, and the new image `im_n` is shown on the right.

15	21	30	17	11	12
45	44	23	60	22	33
98	41	61	10	39	42

0	0	30	0	0	0
255	255	23	255	22	33
255	41	255	0	39	255

12. The RANSAC objective function has been criticized for being a simple count of points whose distance is less than a threshold τ from a line. Instead, some researchers use a weighted objective function that gives more weight for smaller distances:

$$\begin{cases} 1 - d^2/\tau^2 & \text{if } \|d\| \leq \tau \\ 0 & \text{otherwise.} \end{cases}$$

Write code to evaluate this new objective function on the N data points represented as a $2 \times N$ NumPy array called `pts`. Assume a hypothesized line is given to you, represented in implicit form by variables `a`, `b`, `c`. Also assume variable `tau` has been assigned to store the value of τ. Assign the final result to a variable called `obj`.