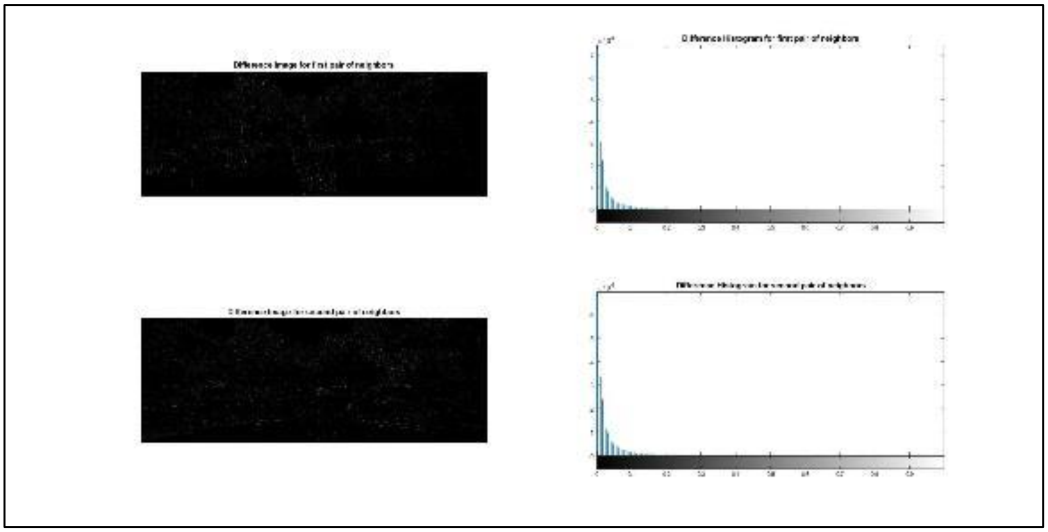
ECE 558 – Digital Imaging Systems
Project 01
mmsardes@ncsu.edu

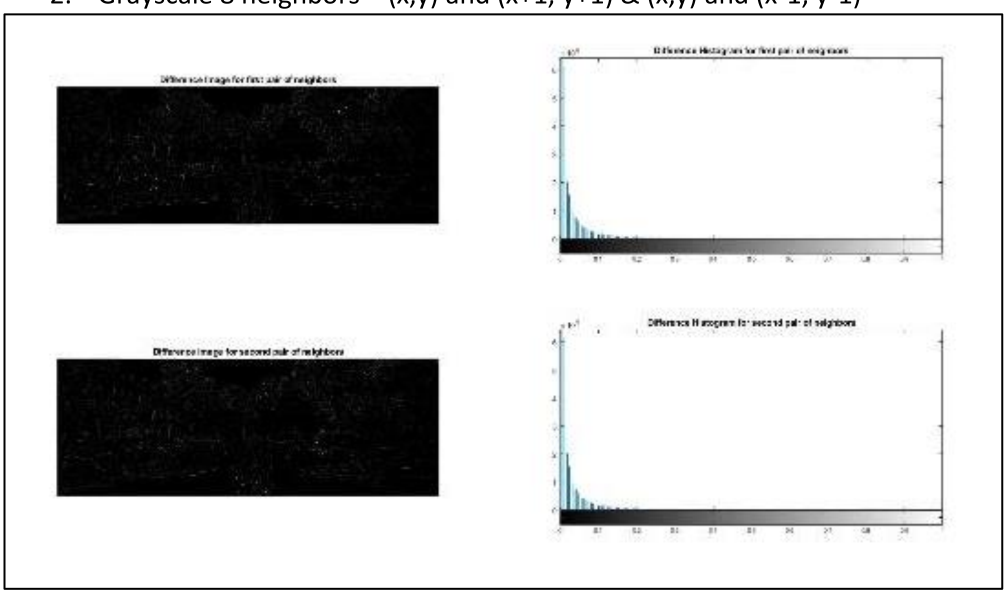
Question 1.
Visualizing and checking the smoothness prior of an image.
Input
1. Image
2. User input
    a. Selection of color space
    b. Selecting neighbor type – 4 and 8/d

Output
1. Histogram of squared difference of neighboring pixels

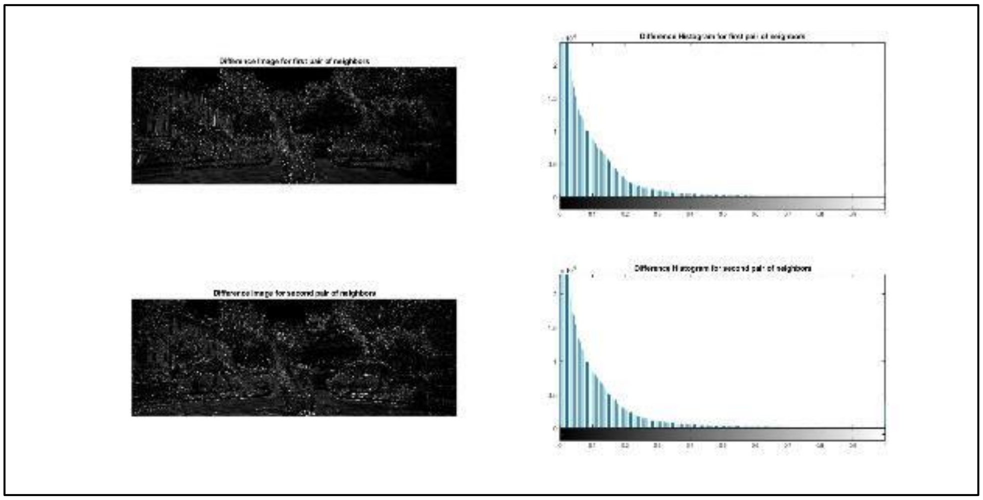
1.Grayscale 4 neighbors – (x,y) and (x+1, y) & (x,y) and (x, y+1)



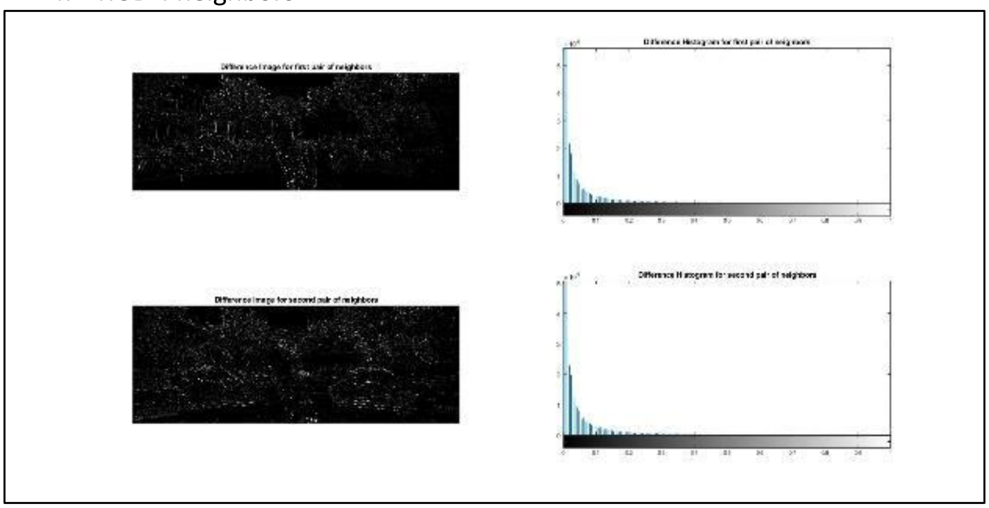
2. Grayscale 8 neighbors – (x,y) and (x+1, y+1) & (x,y) and (x-1, y-1)



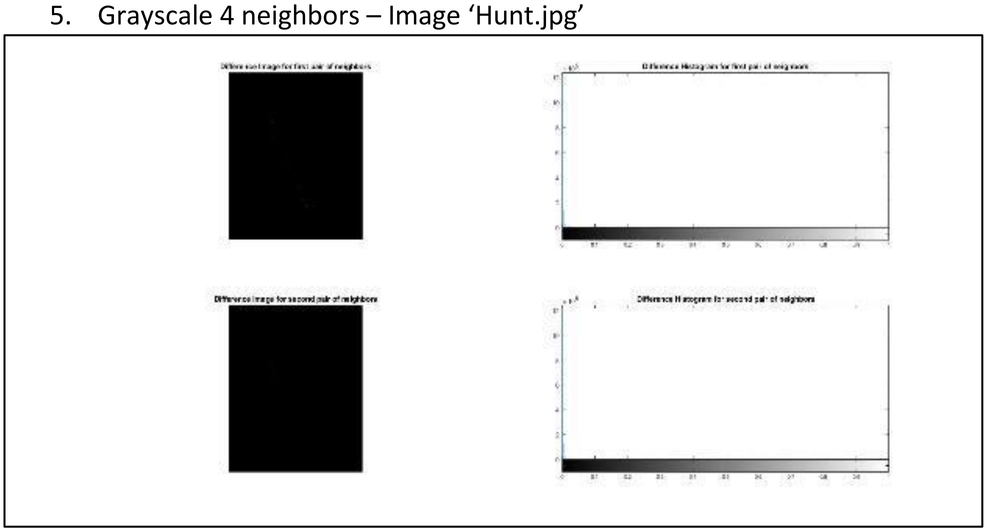
3. HSV 4 neighbor



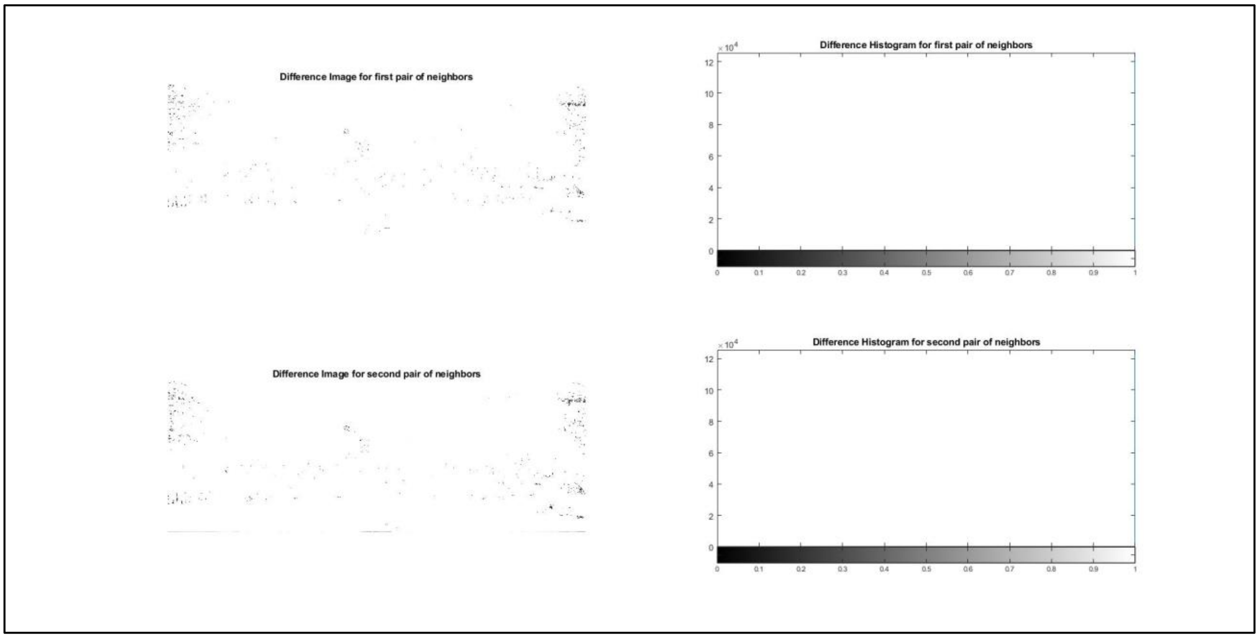
4. RGB 4 neighbors



5. Grayscale 4 neighbors – Image ‘Hunt.jpg’



6. La*b* - 4 neighbors – both pairs.




```
Which color space you want to consider
a. RGB
b. Grayscale
c. HSV
d. La*b*
RGB
Which neighbor type you want to consider
a. 4 - Neighbors -(x,y) and (x+1,y) , (x,y) and (x,y+1)
b. 8/d - Neighbors (x,y) and (x+1, y+1) ,(x,y) and (x-1, y-1)
4
Elapsed time is 0.712301 seconds.
```

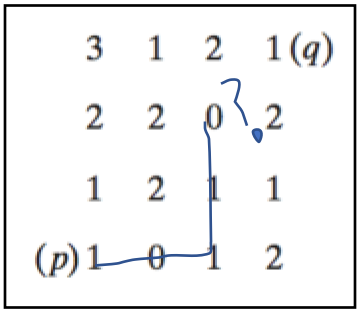
Fig. Recording the Runtime of the code.

Question 2.

Find the shortest path

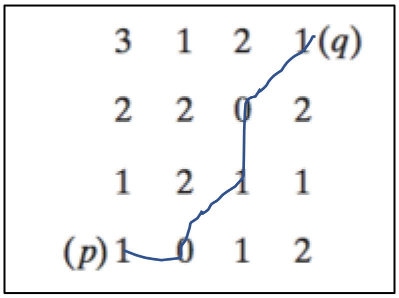
$$\begin{array}{ccccc} & 3 & 1 & 2 & 1\,(q) \\ & 2 & 2 & 0 & 2 \\ & 1 & 2 & 1 & 1 \\ (p) & 1 & 0 & 1 & 2 \end{array}$$

a) Find the shortest 4, 8 and m-paths between p and q for V = {0,1}
b) Repeat the same for V = {1,2}
c) Write code to implement shortest distance function.



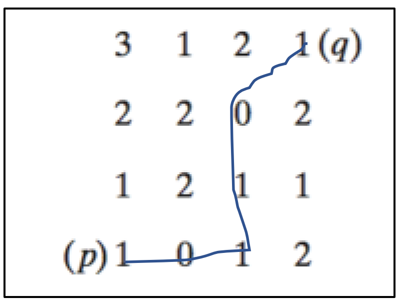
a) 1. 4 path -

We start from p and move towards coordinate (3,2) -> (2,2) -> (1,2)
But beyond that we do not have any other 4 neighbors.
That's why we do not have any valid 4 path

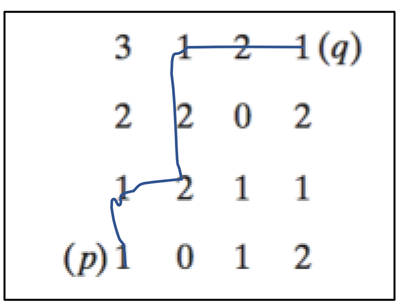


2. 8 path -

We start at p and follow the traced path as shown in the corresponding figure by following diagonal elements wherever possible.
Path length – 4

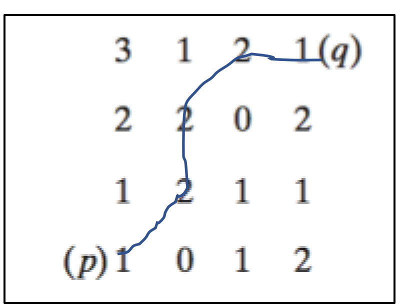


3. m – path
m path is used to solve ambiguities than come up in 8 neighbor adjacencies.
Valid connections for m adjacency are
1. N4 – 4 neighbors
2. Nd – Only if N4(P) and N4(q) have no common neighbors in V-set
Here, we trace the path as shown in the figure. ***Path length – 5***

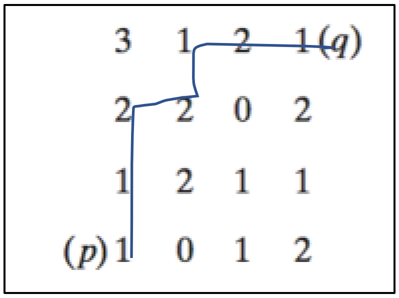


b) V = {1,2}

1. 4 path –
As we have a valid path with elements contained in the V set this time, we can trace our 4 path from p to q as shown.
Path length – 6



2. 8 path –
8 path from p to q can be traced as shown in the figure.
Path length – 4



3. m – path

As there are valid 4 neighbors throughout the path in the corresponding figure for this V set, the m path is same as the 4-path.

c) Python code for shortest path implementation.

The python code first accepts the following inputs from the user.

1. Input Image/ Graph from part a and b
2. Predefined V set
3. Start and end points
4. Path type to be traced (4, 8, m path)

We get the following outputs

1. Shortest path – Coordinate sequence followed.
2. Length of the path

Algorithm

- The image is first split into individual planes for different channels. Every plane can be treated as a separate graph.
- The image is traversed to find the elements which belong to the V set.
- If a pixel value is from the V set it is added as a vertex to the adjacency list. If the pixel value already exists in the keys of the dictionary, the next coordinates are added as neighbors to that vertex.
- Such an adjacency list is created after examining all conditions for the specified path 4/ 8 path.
- The created adjacency list is given as an input to the Breadth first search algorithm which returns the shortest path from pixel p and q/
- Breadth first search is implemented as the weights of the edges of the graph can be considered as 1 for any given type of adjacency. The main goal is to find at least one shortest path. Thus, just a BFS algorithm would work in that case.

  Breadth first search
  1. Add node to the queue of visited nodes. And to working queue
  2. Set current node and empty path by popping them from the queue
  3. While queue is not empty
     a. Visit current node and all its neighbors
     b. Check if neighbors are already visited.
     c. Add neighboring nodes till no visiting nodes are left.
     d. Remove visited nodes from the queue as you go.
- The code includes the given testing example and another custom testing example along with an image input which tests the shortest path for 2 pixels in the image.
- There’s a .html of the python console output which logs the runtime of the code.