LAB # 03: Connected Component Analysis

Lab Objective:

The objective of this lab is to understand concept of connected component analysis algorithm and apply on different images to count objects.

Lab Description:

1. Connected Component Analysis

Connected Component Analysis or Labelling enables us to detect different objects from a binary image. Once different objects have been detected, we can perform a number of operations on them: from counting the number of total objects to counting the number of objects that are similar, from finding out the biggest object of the bunch to finding out the smallest and from finding out the closest pair of objects to finding out the farthest etc.
Connected Component labelling procedure is as follows:

- Process the image from left to right, top to bottom:

  If the next pixel to process is 1

  i.) If only one from top or left is 1, copy its label.

  ii.) If both top and left are one and have the same label, copy it.

  iii.) If top and left they have different labels

  Copy the smaller label

  Update the equivalence table.

  iv.) Otherwise, assign a new label.
- Re-label with the smallest of equivalent labels

Video explanation: https://www.youtube.com/watch?v=ticZclUYy88

Some Useful Commands:

i. arr = np.array([[1, 2, 3] , [6, 5, 4]])
   a. arr + 2 will add 2 in each element of arr

ii. arr = np.array([[1, 2, 3] , [6, 5, 4]])
   a. arr == 2 will return following bolean array
      i. array([[False, True, False] , [False, False, False]])

iii. **_min = numpy.amin(my_array)**

iv. To calculate the power of an array using NumPy: array_power = **numpy.power (my_array, power)**

v. To obtain percentile value. **percentile_array = numpy.percentile(my_array, percentile)**

vi. To change data type of array. **my_array = my_array.astype(numpy.uint16)**

Distance Measures:

If we have 3 pixels p, q and z respectively p with (x,y) ,q with (s,t) and z with (v,w)

Then D is a distance function or metric if

a. $D(p,q) \geq 0$,$D(p,q) = 0$ iff $p = q$

b. $D(p,q) = D(q,p)$

c. $D(p,z) \leq D(p,q) + D(q,z)$

Euclidean distance between p and q is defined as:

$$De(p,q) = [(x-s)^2 + (y-t)^2]^{1/2}$$

The pixels having distance less than or equal to some value r from (x,y) are the points contained in a disk of radius r centered at (x,y).

D4 distance (also called **city-block distance**):

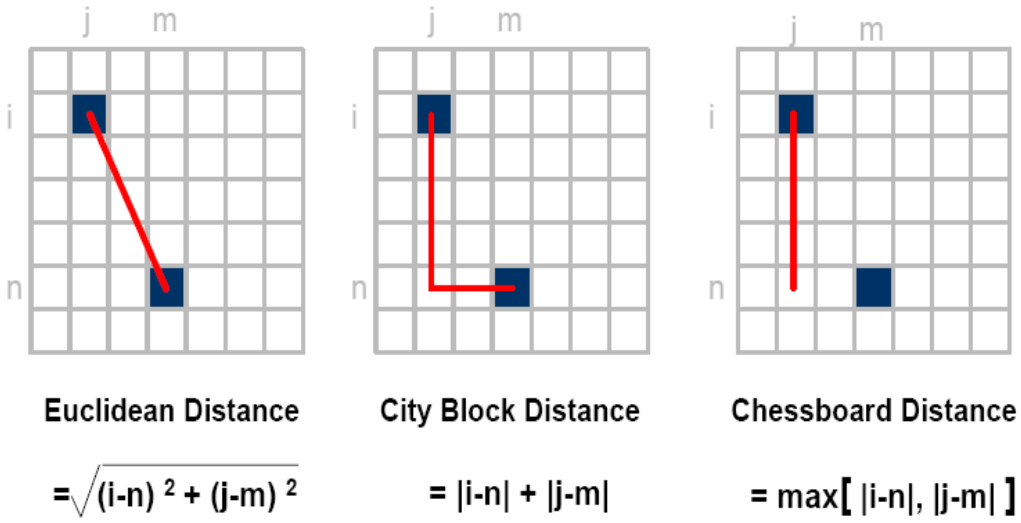
$$D4(p,q) = |x-s| + |y-t|$$

The pixels having a D4 distance less than some r from (x,y) form a diamond centered at (x,y) .

D8 distance (also called **chessboard distance**):

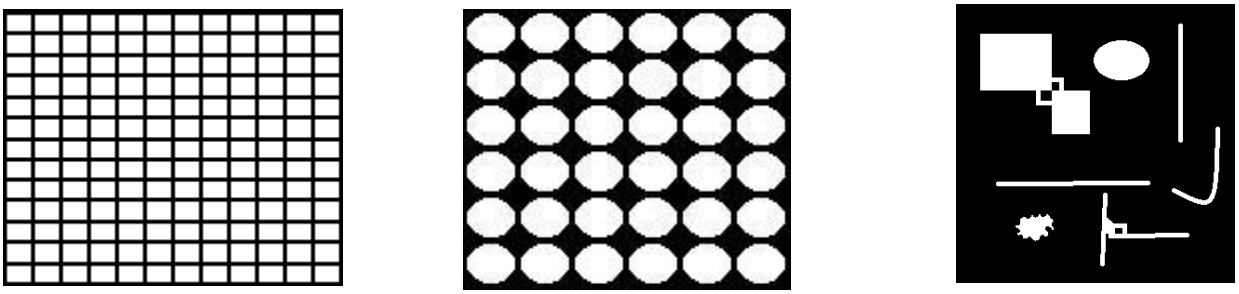
$$D8(p,q) = \max(|x-s| , |y-t|)$$

The pixels having a D8 distance less than some r from (x,y) form a square centered at (x,y)



Lab Tasks:

1. For the image given below (provided with the lab handout), apply the connected component labelling and count the total number of white objects. First threshold the images and then perform connected component analysis algorithm.



(a) (b) (c)

2. Write a function which takes two point of image as input argument and calculate above distances.

Calculate distance(p1,p2,choice). Choice variable is input integer. If its value is 1 calculate Euclidean distance , 2 for city block distance and 3 for chessboard distance.

Applications of Connected Component Analysis:

Following are some applications of connected component analysis. We can use this analysis in many real problems such as,

I. Binary object classification
II. Detect multiple objects in image

References:

- ✓ https://homepages.inf.ed.ac.uk/rbf/HIPR2/label.htm
- ✓ https://www.pyimagesearch.com/2016/10/31/detecting-multiple-bright-spots-in-an-image-with-python-and-opencv/