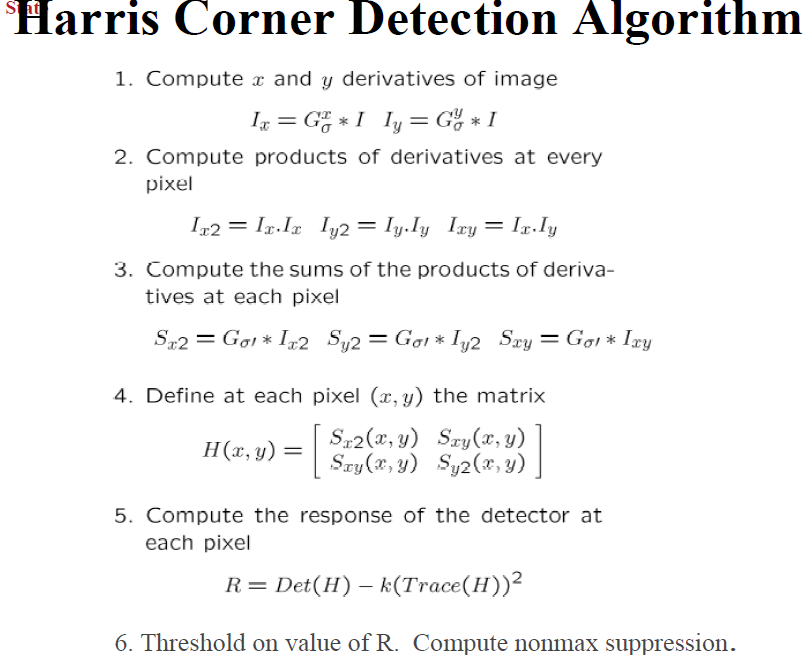
**Cs 512 Assignment 2: report**

**Abstract –** This program is for corner detection and using Harris corner detection algorism and localization algorism. Meanwhile, in order to feature vector detect, we increase the gradient vector degree histogram to do the feature detection.

1. **Problem statement**
   1. Do not know how to realize Harris corner detection algorism.
   2. What is the meaning of result of Localization algorism?
   3. How to do feature vector for each corner point
   4. Cannot understand the meaning of ‘the variance of the Gaussian (scale)’, ‘the neighborhood size for computing the correlation matrix’, ‘the weight of the trace in the Harris corner detector’.
   5. Using cv2.connectedComponentsWithStats function to find the center of the corner but the center of image also appears in the result.
   6. How to create a histogram for gradient degree?
   7. feature vector cannot detect same corner in two image.
2. **Proposed solution**
   1. After checking opencv source code and material from network.
   2. Checking with TA to understand the meaning of those parameter.
   3. Ignore the first center as that is the center of the image.
   4. Create 9 column which count the number of -180, -135, -90, -45, 0, 45, 90, 135, 180 degree vectors.
   5. Calculate the distance between each corner with histogram value and then to do the feature detection.
3. **Implementation details**



**Localization**:

1. ***P = inverse(C) \* Σ (∇I(x) \* Ttranspose (∇I(x) \* x)***

Σ (∇I(x) \* Ttranspose (∇I(x) \* x) = (sum(Ixx) \* x + sum(Ixy) \* y, sum(Ixy \* x + sum(Iyy) \* y)

Scale = 1 / (det(C))

1. ***inverse(C) \* Σ (∇I(x) \* Ttranspose (∇I(x) \* x) ：***

#c-1 \* sum I(xi)I(xi)t\*xi

det = a\*c - b\*b

scale = 1/det

out[i-1][0] = centroids[i][0] + c\*scale\* np.sum(bb1,None) - b\*scale \* np.sum(bb2,None)

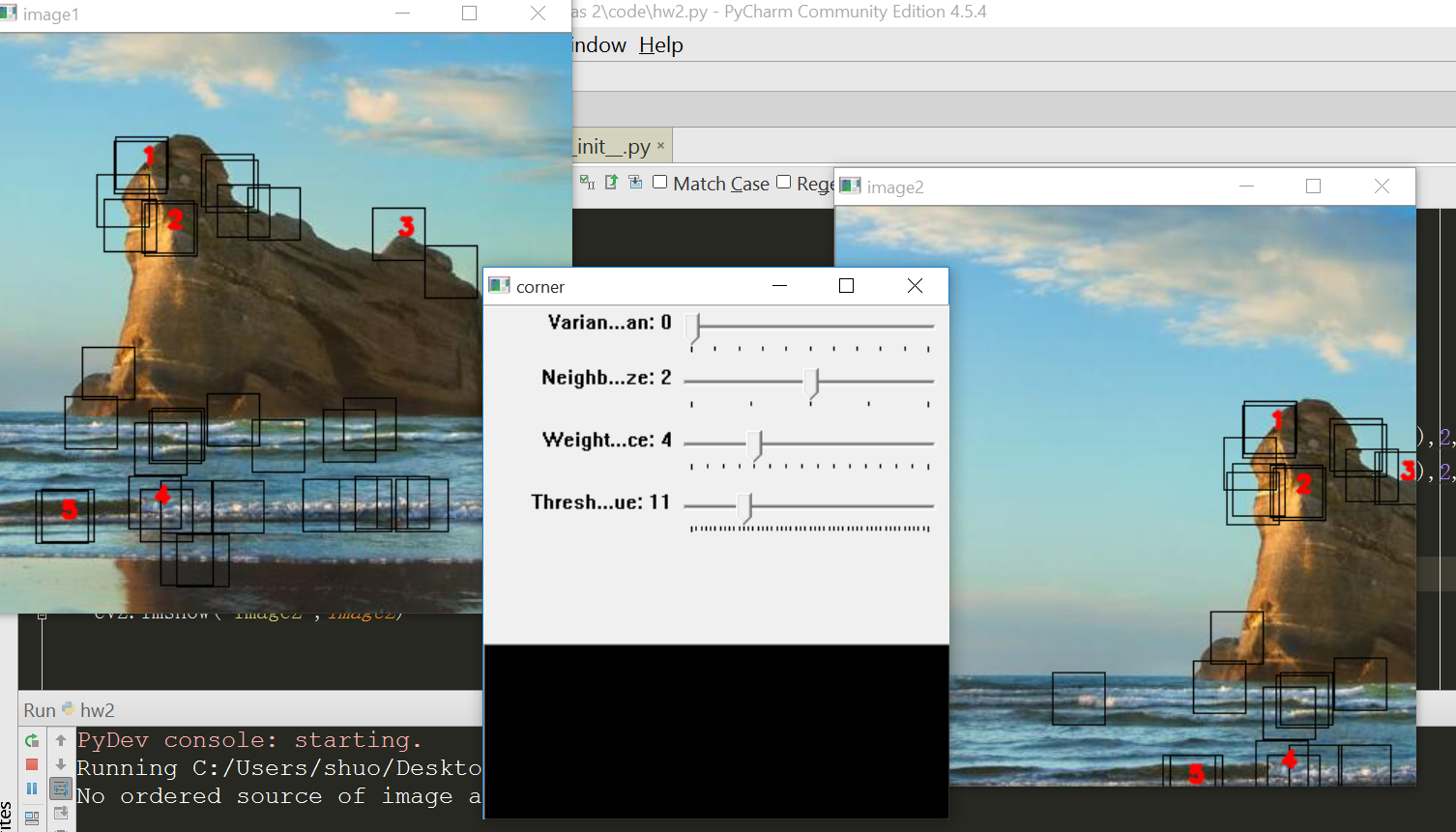
out[i-1][1] = centroids[i][1] - b\*scale\* np.sum(bb1,None) + a\*scale \* np.sum(bb2,None)

**Feature vector**

Filter each corner`s histogram, for each histogram in image 1, it has a smallest distance between the corner in image 2, that seems feather matched successfully.

cno = 1  
*for* i *in* range(0, len(*corners1*)):  
 *for* j *in* range(0, len(*corners2*)) :  
 # compare histogram for each corner  
 *if*( np.array\_equal(*histogram1*[i], *histogram2*[j]) ):  
 cv2.putText(*image1*,str(cno), (*corners1*[i,0],*corners1*[i,1]), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5,(0,0,255),2,cv2.LINE\_AA)  
 cv2.putText(*image2*,str(cno), (*corners2*[j,0],*corners2*[j,1]), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5,(0,0,255),2,cv2.LINE\_AA)  
 cno = cno +1  
  
cv2.imshow('image1',*image1*)  
cv2.imshow('image2',*image2*)

1. **Results and discussion**



1. **References**

[1] <http://docs.opencv.org/> opencv website.