

Assignment #2

(Due on 30th of November)

The main aim of this assignment is to analyze the interaction between various scoring functions and automatic threshold detection. In more specific terms, scoring functions (corner detections) are to be applied to the image, then the resulting gradient magnitudes are to be subjected to automatic threshold detection. As per that, the following components are to be implemented:

1. Corner detection:
 - a. Harris.
 - b. SUSAN.
2. Automatic threshold detection:
 - a. Calculate convex hull.

Assignment #2

(Due on 30th of November)

Harris and Susan corner detection

In this part, you are asked to implement two functions as follows:

1. Harris:

- **Input:**
 - i. **The input image** (2D array or the original image)
 - ii. **Tunable parameter k** (for simplicity equate it to **0.04** for testing purposes).
- **Output:**
 - i. **2D array** representing the matrix (the resulting image) after corner detection.
 - ii. **Min** which represents the min value of the resulted values after applying harris on the input image
 - iii. **Max** which represents the max value of the resulted values after applying harris on the input image
- **Description:**
 - Implements the Harris corner detection technique as discussed in the tutorial using **a 3 x 3 window** around each pixel and use the following kernels to compute the derivative in the **x** and **y** directions:

df/dx :

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

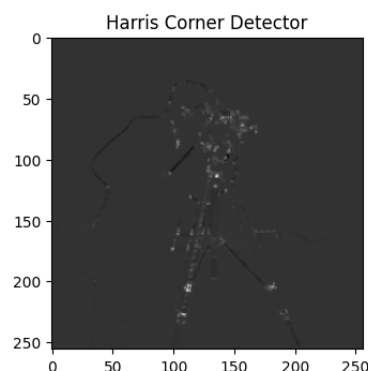
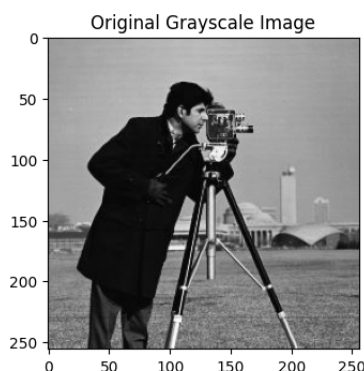
df/dy :

-1
0
1

- **Example for the output:**

1. First Example:

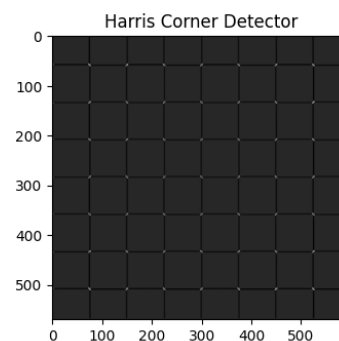
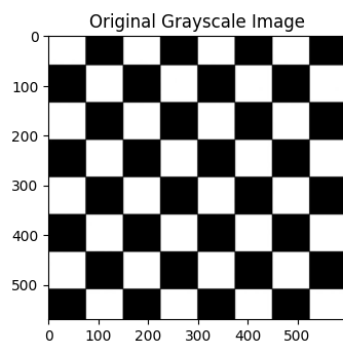
- Min = -4756062904.36
- Max = 19205376969.44



Assignment #2

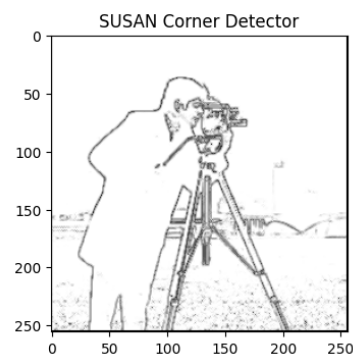
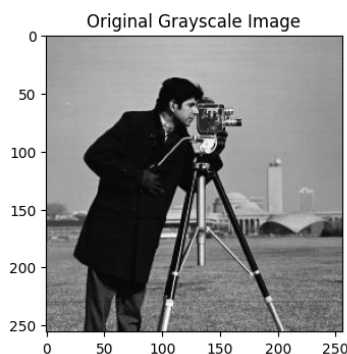
(Due on 30th of November)

- Min = -4678122250.24
- Max = 25481986947.04



2. Susan:

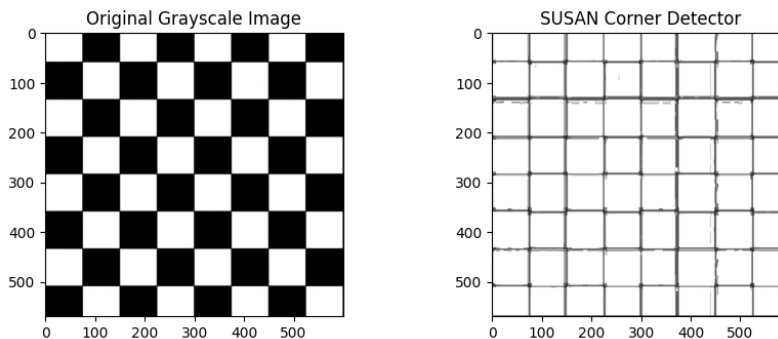
- **Input:**
 - i. **The Original Image** (It is advisable to not turn the image to 2D array to give more accurate results)
 - ii. **Tunable Parameter t** (which will differ from one image to the other)
- **Output:**
 - i. **2D array** representing the resulting image.
- **Description:**
 - Implement the Susan corner detection using the equation discussed in the tutorial and a kernel size which is equal to **3x3**.
- **Example for the output:**
 1. **First Example with $t = 50$:**



Assignment #2

(Due on 30th of November)

2. Second Example with $t = 2$:



Getting Data Ready for Convex Hull

(Given)

In order to get the data ready for the convex hull, we need to get the **histogram** of the resulting images. **Susan** is fine to get a histogram while **Harris** is not !!. Why is this? Because the resulted values vary from the negative to positive values. Hence, we get the max and min value resulted in Harris. So we need to handle it first by performing **contrast stretching** on the image before getting the histogram so that the values are mapped between **[0-255]**.

1. Contrast Stretching (given):

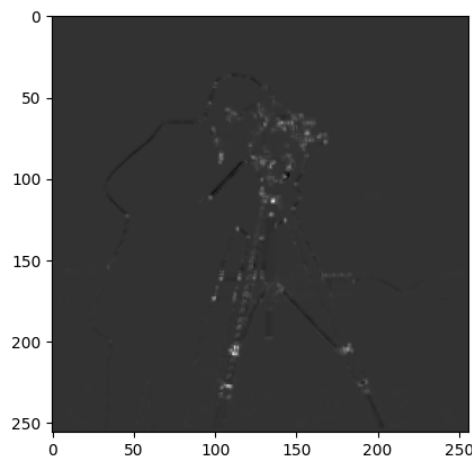
- **Input:**
 - i. **2D array representing the image** (the output of harris)
 - ii. **a,b,c,d** (the variables required for contrast stretching)
- **Output:**
 - i. **2D array representing the output after applying contrast stretching.**

Assignment #2

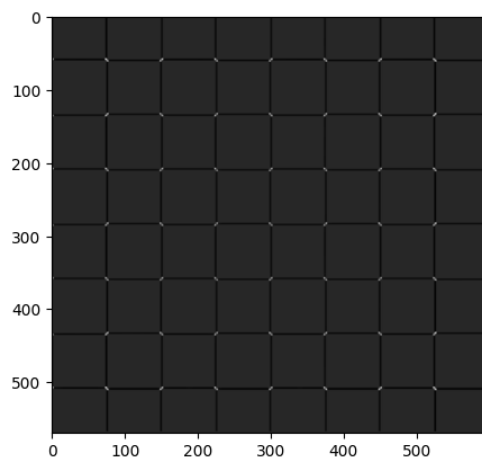
(Due on 30th of November)

- **Example:**

1. **On the camera man image:**



2. **On the second example:**



2. Calculate Histogram for Susan (given):

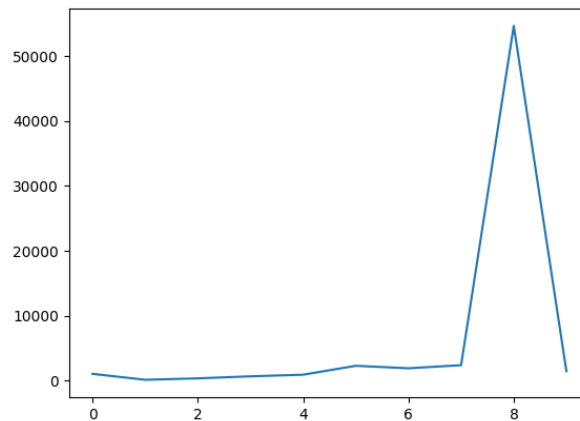
- **Input:**
 - i. 2D array representing the image (resulted from susan)
- **Output:**
 - i. an array representing the histogram of the image

Assignment #2

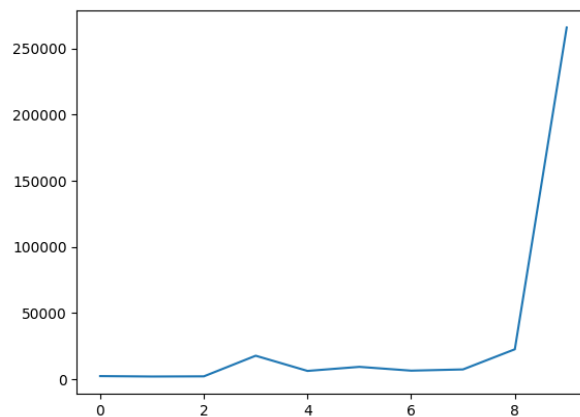
(Due on 30th of November)

- **Example:**

- i. **For first image:**



- ii. **For second image:**



3. Calculate Histogram for harris (given):

- **Input:**

- i. **2D array representing the image** (resulted from harris after doing the necessary modification)

- **Output:**

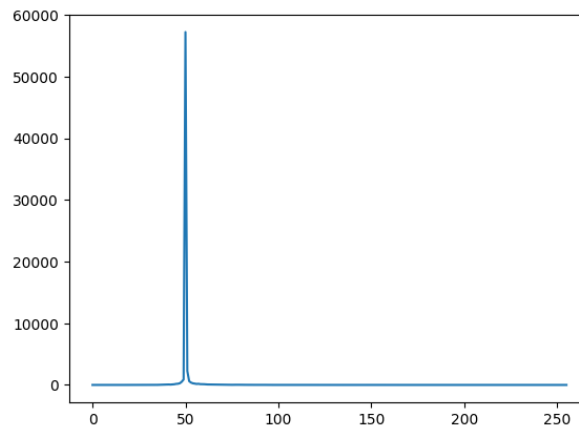
- i. **an array representing the histogram of the image**

Assignment #2

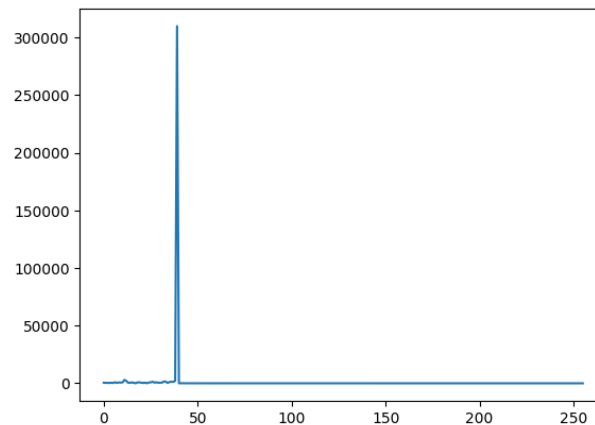
(Due on 30th of November)

- **Example:**

- i. **For first image:**



- ii. **For second image:**



Convex Hull:

As gradient magnitudes are calculated out of edges and corners detectors, a threshold is traditionally used in order to allow only strong features to be returned as a result. Instead of assigning static thresholds, an automated approach is to be followed, which is the

Assignment #2

(Due on 30th of November)

convex hull. Hence, in this part you are asked to implement the following functions:

1. GetHull:

- **Input:**
 - i. **1D array** representing the histogram.
 - **Output:**
 - ii. **1D array of tuples** representing the start point and the end of each region in the convex hull.
 - **Description:**
 - As discussed in the tutorial, we start the convex hull by getting the maximum slope that will represent each region and hence we want to know the start point of the region (x,y) and the end point of it (x,y).
 - **Example:**
 - i. **For Susan:**
 - **For the first image:**
 - a. [(0, 1020), (8, 54633), (9, 1423)]
 - **For the Second Image:**
 - a. [(0, 2336), (9, 265732)]
 - ii. **For Harris:**
 - **For the first image:**
 - a. [(0, 1), (50, 57206), (255, 0)]
 - **For the Second Image:**
 - a. [(0, 504), (39, 309797), (255, 0)]
-

Assignment #2

(Due on 30th of November)

2. GetHullValues:

- **Input:**
 - i. **1D array of tuples** representing the convex hull points..
- **Output:**
 - i. **1D array** representing the hull values for each color.
- **Description:**
 - Here we get the hull values according to the equation of region which you can get using the equation of the straight line: **$y=mx+c$** .
 - Get the **hull value** of each intensity by substituting in the line equation to get the **y** value (*representing the hull value*)
- **Example:**
 - i. **For Susan:**
 - **For the first image:**
 - a. [1020.0, 7721.625, 14423.25, 21124.875, 27826.5, 34528.125, 41229.75, 47931.375, 54633.0, 1423.0]
 - **For the second image:**
 - a. [2336.0, 31602.222222222223, 60868.444444444445, 90134.666666666667, 119400.88888888889, 148667.11111111112, 177933.33333333334, 207199.55555555556, 236465.77777777778, 265732.0]

3. GetThreshold:

- **Input:**
 - i. **1D array** representing the histogram.
 - ii. **1D array** representing the hull values.
- **Output:**
 - i. **The threshold** according to the convex hull criteria

Assignment #2

(Due on 30th of November)

- **Description:**

- i. Here you take the hull values and the histogram values and apply the last step in convex hull algorithm to get the threshold

- **Example:**

- i. **For Susan:**

- 1. **First Image:**

- 7

- 2. **Second Image:**

- 8

- ii. **For Harris:**

- 1. **First Image:**

- 52

- 2. **Second Image:**

- 40

4. ApplyThresh

- **Input:**

- i. **1D array** representing the histogram
 - ii. **The Threshold** value that we already got.

- **Output:**

- i. **2 1D arrays** representing the histogram after splitting it below and above the threshold.

- **Description:**

- The main aim of this function is to split the histogram into two arrays representing the histogram of values below threshold and histogram of values above threshold so that we can re-apply the thresholding once again on the resulted histograms

- **Example:**

- **For Susan:**

- **First Image:**

- [1020, 107, 324, 641, 890, 2261, 1881]

DMET 901 – Computer Vision

Assignment #2

(Due on 30th of November)

- [2356, 54633, 1423]
- Second Image:
 - [2336, 2020, 2147, 17732, 6258, 9327, 6458, 7387]
 - [22603, 265732]

Plotting Results (Given):

1. drawImages:

- **Input:**
 - i. **First Threshold** representing the lower threshold.
 - ii. **Second Threshold** representing the higher threshold.
 - iii. **Image** the image resulted from the corner detectors.
- **Output:**
 - i. **3 images** representing the original image after splitting it into three regions which are:
 - **lower region** which is the image below the threshold.
 - **mid region** which is the image between the 2 thresholds.
 - **higher region** which is the image above the higher threshold.
- **Description:**
 - The main purpose of this image is to split the image into 3 images after getting all the work done and getting 2 thresholds from the resulted image of the corner detection so that we can divide the image into corners, detectors and flat regions.

Final Goal:

The final goal is to apply Harris and Susan corner detection to the image, afterwards you apply Convex hull to the result so that you can get a specific value corresponding to the threshold. Based on this threshold, you will split the histogram of the resulting image into 2

DMET 901 – Computer Vision

Assignment #2

(Due on 30th of November)

parts and then take the first part and apply on it the Convex Hull thresholding again resulting in 2 thresholds so that we can divide the image based on them.

Code Template:

Please use the following link as a template for your code.

<https://colab.research.google.com/drive/1AkAKlwE4SJNu4AfDbs9u88Fauv3j8o9i?usp=sharing>

First of all copy it to your google drive so that you can edit it and save your results. You will find the imports for the libraries and the images in the template, together with the given function that were highlighted earlier. Also, you will find the final result that you will get evaluated based on it at the end of the template with the plotting template and separate cells for each of the 4 test images.

Please use the indicated tunable parameters as indicated in the template with its corresponding image to get the right results.

Please Note: you are not allowed to use any built in method except the one for the conversion of the image to array!!

The details for the submission will be send later