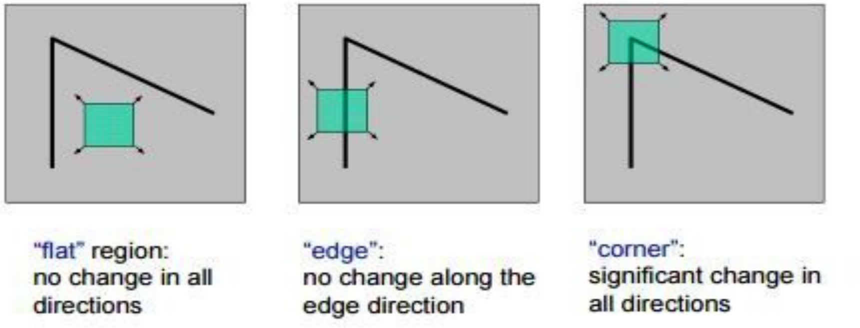
Harris Corner Detection

Name : Aditi Sawant
Roll No: 193310008

Introduction:

- A corner is a point whose local neighborhood stands in two dominant and different edge directions
- Corners are the important features in the image, and they are generally termed as interest points which are invariant to translation, rotation, and illumination.



Source: https://medium.com/data-breach/introduction-to-harris-corner-detector-32a88850b3f6

Theory:

Algorithm basically finds the difference in intensity for a displacement of (u,v) in all directions. This is expressed as follows:

$$E(u,v) = \sum_{x,y} \underbrace{w(x,y)}_{\text{window function}} \, \underbrace{[I(x+u,y+v)}_{\text{shifted intensity}} - \underbrace{I(x,y)]^2}_{\text{intensity}}$$

Window function is either a rectangular window or gaussian window which gives weights to pixels underneath.

We need to maximize function E(u,v) for corner detection. We need to maximize second term in equation.

Applying Taylor Expansion to above equation and using some mathematical steps we get:

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

Where,

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

Here I_x and I_y are image derivatives in x and y directions respectively.

Below equation will determine if window can contain a corner or not.

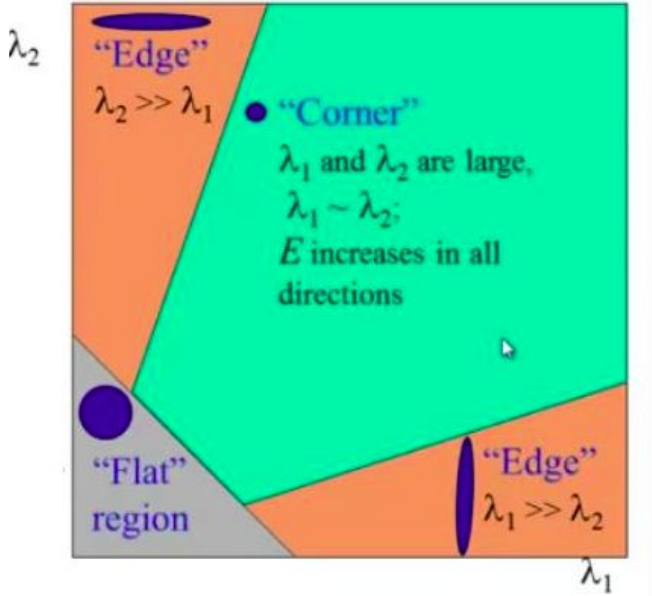
$$R = det(M) - k(trace(M))^2$$

Where,

- $det(M) = \lambda_1 \lambda_2$
- $trace(M) = \lambda_1 + \lambda_2$
- Eigenvalues of M are λ_1 and λ_2

So the values of these eigen values decide whether a region is corner, edge or flat.

- When |R| is small, which happens when λ_1 and λ_2 are small, the region is flat
- When R < 0, which happens when $\lambda_1 >> \lambda_2$ or vice versa,the region is edge
- When R is large, which happens when λ_1 and λ_2 are large and $\lambda_1 \sim \lambda_2$ the region is a corner.



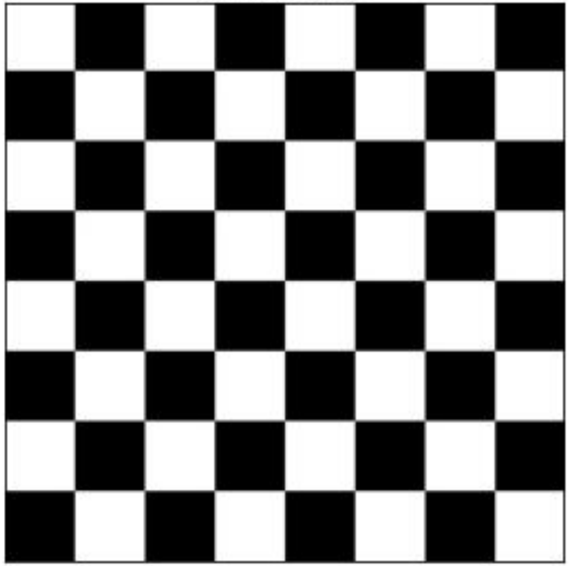
Source: https://opencv-python-tutroals.readthedocs.io/en/latest/_images/harris_region.jpg

Pseudocode:

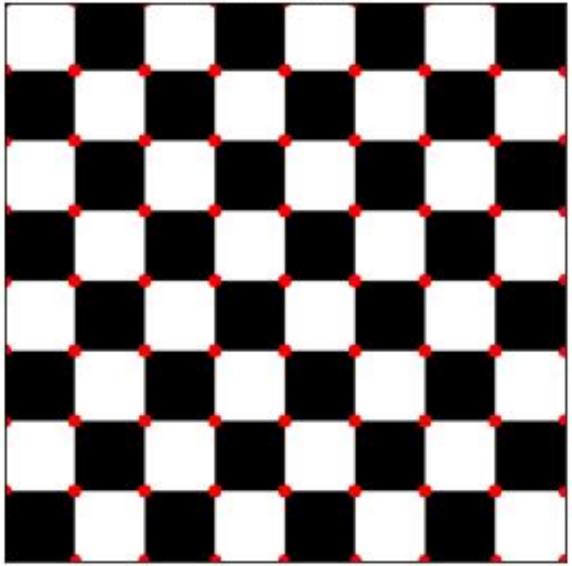
- Take the grayscale of the original image
- Apply a Gaussian filter to smooth out any noise
- Apply Sobel operator to find the x and y gradient values for every pixel in the grayscale image
- For each pixel p in the grayscale image, consider a 3×3 window around it and compute the corner strength function. Call this its Harris value.
- Find all pixels that exceed a certain threshold and are the local maxima within a certain window (to prevent redundant dupes of features)
- For each pixel that meets the criteria in 5, compute a feature descriptor.

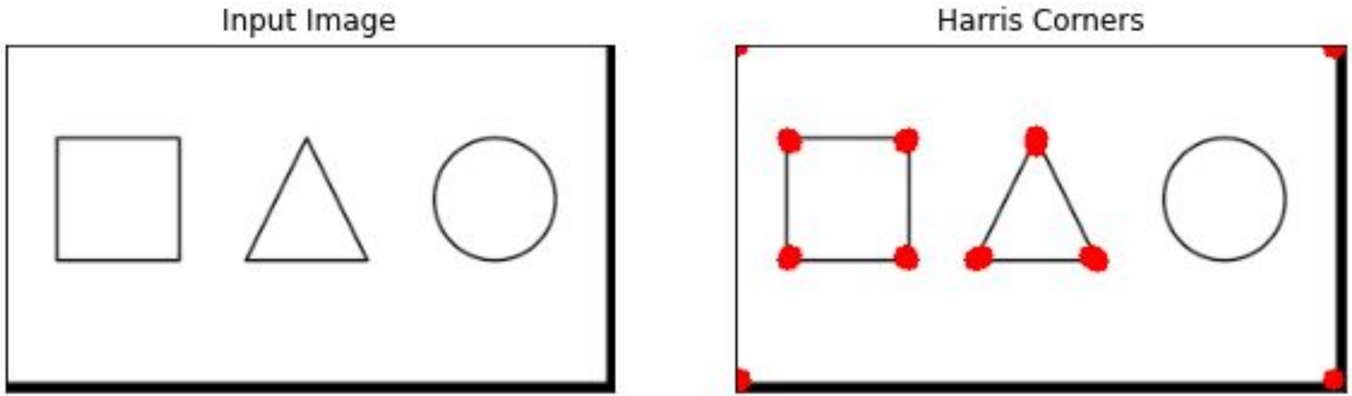
Outputs:

Input Image



Harris Corners




Input Image
Harris Corners

References:

- https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_features_harris/py_features_harris.html
- https://medium.com/data-breach/introduction-to-harris-corner-detector-32a88850b3f6
- https://en.wikipedia.org/wiki/Harris_Corner_Detector