HARRIS CORNER DETECTION ALGORITHM

Mainak Dutta
IRO2017009
Mtech Robotics, 1st year

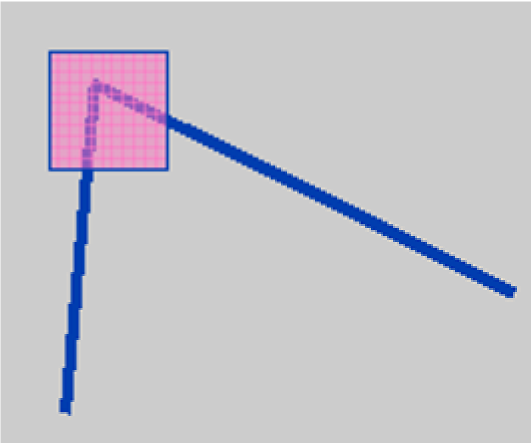
HARRIS CORNER DETECTION

In computer vision, usually we need to find matching points between different frames of an environment. Why? If we know how two images relate to each other, we can use *both* images to extract information of them.When we say matching points we are referring, in a general sense, to characteristics in the scene that we can recognize easily. We call these characteristics features.

1. Corners are generally intuitively junction of contours.
2. Generally more stable changes over changes of viewpoint
3. Intuitively large variation in the neighbourhood of the point in all directions
4. They are good features to match.

Corner points:

1. We should easily recognize the point by looking at intensity values within a small window.
2. Window change in any direction will yield great change in the appearance.



HARRIS CORNER DETECTION BASIC IDEA:

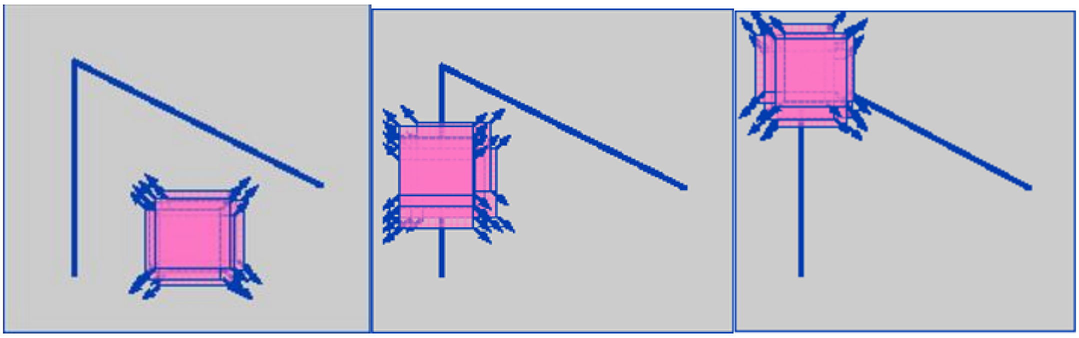


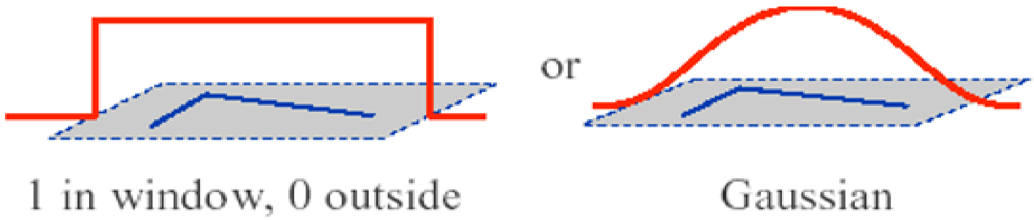
Fig 1: flat region no change in all directions Fig 2: 'edge' no change in edge directions Fig 3 : 'corner' significant change in all directions

Harris corner detection gives a mathematical model to implement all the cases.

We know that corners are regions in the image with large variation in intensity in all the directions. One early attempt to find these corners was done by Chris Harris & Mike Stephens in their paper A Combined Corner and Edge Detector in 1988, so now it is called Harris Corner Detector. He took this simple idea to a mathematical form. It basically finds the difference in intensity for a displacement of (u,v)in all directions. This is expressed as below

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

Window function can be of two types:



For nearly constant patches,

$[\underbrace{I(x+u,y+v)} - \underbrace{I(x,y)}]^2$ will be zero. For very distinctive patches,this will be larger. Hence, we want patches where E(u,v) is maximum.

First order Taylor's Expansion is given by:

$$f(x+u,y+v) \approx f(x,y) + uf_x(x,y) + vf_y(x,y)$$

For maximizing the E(u,v) we need to maximize the second term and doing so we get,

$$\sum [I(x+u,y+v) - I(x,y)]^2$$

(First order approximation)

$$\approx \sum [I(x,y) + uI_x + vI_y - I(x,y)]^2$$

$$= \sum u^2 I_x^2 + 2uvI_xI_y + v^2 I_y^2$$

Rewriting as Matrix equation,

$$= \sum \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$= \begin{bmatrix} u & v \end{bmatrix} \left(\sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}$$

We get the final equation as :

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

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

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

Classification using Eigenvalues:

After this, they created a score, basically an equation, which will determine if a window can contain a corner or not.

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

Where, det(M) = $\lambda_1 \lambda_2$

Trace(M) = $\lambda_1 + \lambda_2$

λ_1 and λ_2 are the eigen values of M

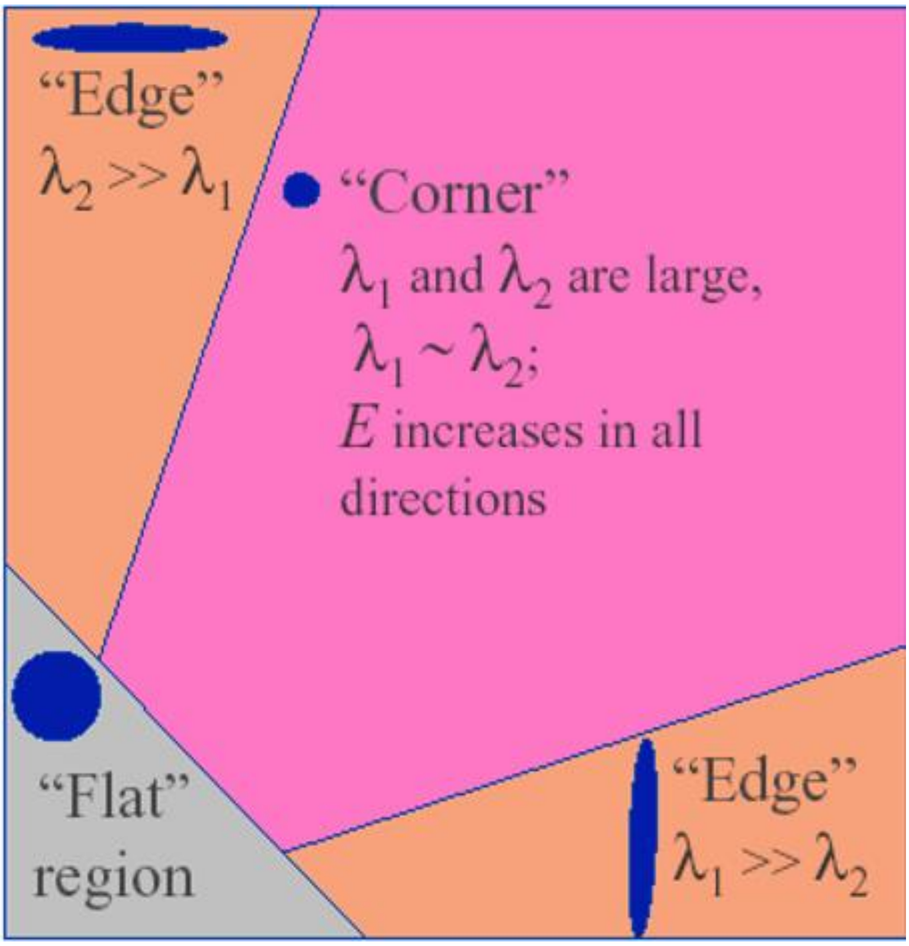
So, the values of these eigen values decide whether a region is corner,edge or flat.

When |R| is small,which happens when λ_1 and λ_2 are small; flat region

When R < 0, which happens when $\lambda_1 >> \lambda_2$ or vice versa, region is a 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.

The above theory is summed up in one picture as:



Complete Algorithmic Steps:

1. Compute x and y derivatives of image

$$I_x = G_\sigma^x * I \quad I_y = G_\sigma^y * I$$

2. Compute products of derivatives at every pixel

$$I_{x2} = I_x . I_x \quad I_{y2} = I_y . I_y \quad I_{xy} = I_x . I_y$$

3. Compute the sums of the products of derivatives at each pixel

$$S_{x2} = G_{\sigma'} * I_{x2} \quad S_{y2} = G_{\sigma'} * I_{y2} \quad S_{xy} = G_{\sigma'} * I_{xy}$$

4. Define at each pixel (x, y) the matrix

$$H(x,y) = \begin{bmatrix} S_{x2}(x,y) & S_{xy}(x,y) \\ S_{xy}(x,y) & S_{y2}(x,y) \end{bmatrix}$$

5. Compute the response of the detector at each pixel

$$R = Det(H) - k(Trace(H))^2$$

6. Threshold on value of R. Compute nonmax suppression.

References:

1. Harris Corner Detection, Robert Collins, Penn state
2. Local feature detectors and descriptors, Computer Vision,IIT Madras
3. Analysis of Harris Corner Detection for Color Images, Parvathy Ram&Dr. S. Padmavathi