

Harris Corner Detection

Mickey Smith

September 2019

1 Background

Corners are one of the most important features that can be detected within an image. Similar to an edge detector, methods exist for detecting the regions within an image where a corner is likely to exist. Also similarly to an edge detector, an important characteristic of the image used to gain information to predict corners is the set of directional brightness gradients within the image.

2 Corner Detection Algorithm

The Harris Corner Detection Algorithm can be split into the following computation steps:

1. Compute the x and y derivatives of the image using a Sobel operator
2. Iterate over the image using a window of size (u, v)
3. At each window location compute dx^2 , dy^2 , and $dx dy$
4. Create a $u \times v$ matrix, M , containing the sum of each derivative product within the window
5. Using the eigenvalues of the matrix, λ_1 and λ_2 , determine if a region contains a corner.

2.1 Image Derivatives

The derivative of an image can be taken extremely quickly and simply by convolving the image with a Sobel edge-detection operator, to determine the gradient of the image in a specified direction. The Sobel operators used to determine dx and dy are shown in figure 1

$$\begin{array}{cc} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \\ \text{(a) } dx \text{ Sobel Operator} & \text{(b) } dy \text{ Sobel Operator} \end{array}$$

Figure 1: Sobel Operators

2.2 Corner Detection/ M Calculation

Now the hunt for corners begins. Using a window of size (u, v) , and iterating through the image, at each point it is possible to create an M matrix that is specific to each window. Equation set 2 demonstrates how to derive M for an image. In equation set 2, I_x and I_y represent dx and dy , respectively.

$$E(u, v) = \sum_{x,y}^{u,v} w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

Using Taylor Expansion:

$$E(u, v) \approx \sum_{x,y}^{u,v} [I(x, y) + uI_x + vI_y - I(x, y)]^2$$

Cancelling and distributing:

$$E(u, v) \approx \sum_{x,y}^{u,v} [u^2 I_x^2 + 2uv I_x I_y + v I_y^2]$$

Using the window function and grouping into a matrix, we can get M :

$$M = w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Figure 2: M matrix calculation

2.3 Thresholding Image

Once M is determined for an image, it is still necessary to determine whether or not the region contains a corner. The eigenvalues of M are critical for this. Since we know that $\det(M) = \lambda_1 \lambda_2$, and that $\text{trace}(M) = \lambda_1 + \lambda_2$, we can just take use the determinant and trace of M 's relationship to create a score for a given window (equation ??).

$$r = \det(M) - k(\text{trace}(M))^2$$

Here, k , is an empirically determined constant that is between 0.04 and 0.06. If r is above some threshold then, we positively identify a pixel that contains a corner. For the purpose of the programs created here, the 99th percentile is used as the value to threshold r , so that 1% of pixels in the image contain a corner. This is empirically determined and different for other applications.