Winter 2019

DMET 901 – Computer Vision

Local Feature Extraction (1)

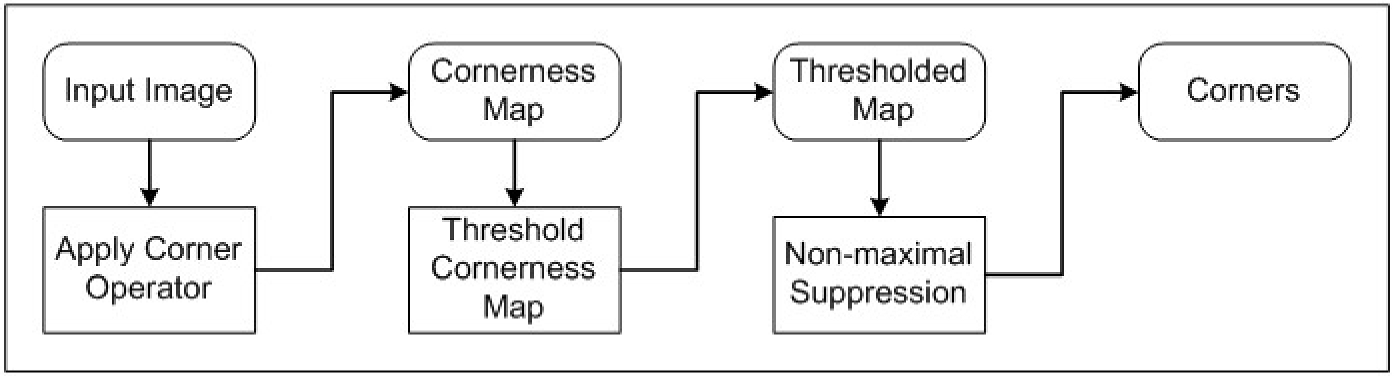
Seif Eldawlatly

Corner Detectors

- It is always useful to find pairs of corresponding points in two similar images
- This could be used in the analysis of moving images
- This could be done by comparing all possible pairs of pixels in the two images. However, it is computationally intensive
- This process might be simplified by comparing interest points only such as corners
- A corner can be defined as a pixel in its small neighborhood where two dominant and different edges meet

Moravec Operator

- General structure of corner detectors



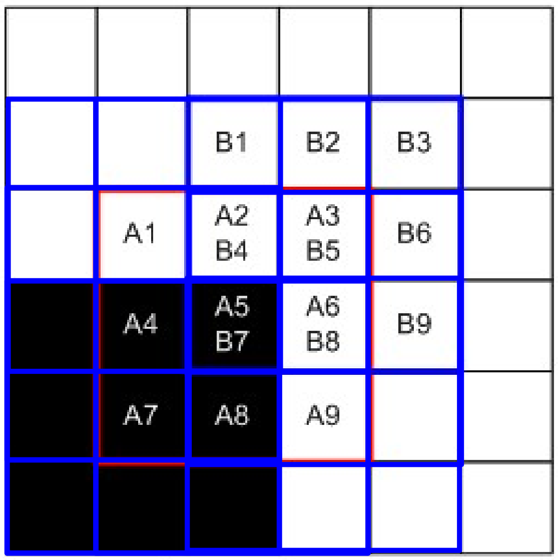
http://kiwi.cs.dal.ca/~dparks/CornerDetection/algorithms.htm

- Moravec operator estimates the *cornerness* of a point by computing a measure of intensity variation in a given neighborhood w with a shift of u and v

$$V(x,y)_{w(u,v)} = \sum_{i=-1}^{1} \sum_{j=-1}^{1} \left(f(x+i, y+j) - f(x+u+i, y+v+j) \right)^2$$

Moravec Operator

- It measures the cornerness by shifting a window around the considered pixel by 1 pixel in each of the 8 principal directions and calculating the corresponding V



$$V = \sum_{i=1}^{9} (A_i - B_i)^2 = 3 * 255^2$$

http://kiwi.cs.dal.ca/~dparks/CornerDetection/algorithms.htm

$$V = \begin{array}{cc} & \textbf{Window } \boldsymbol{B} \textbf{ Centered at} \\ \begin{bmatrix} 3*255^2 \\ 2*255^2 \\ 3*255^2 \\ 2*255^2 \\ 5*255^2 \\ 2*255^2 \\ 3*255^2 \\ 2*255^2 \end{bmatrix} & \begin{array}{c} \text{A3} \\ \text{A2} \\ \text{A1} \\ \text{A4} \\ \text{A7} \\ \text{A8} \\ \text{A9} \\ \text{A6} \end{array} \end{array}$$

- The cornerness at a pixel (x, y)

$$C(x, y) = \min_{u,v} V(x, y)_{w(u,v)}$$

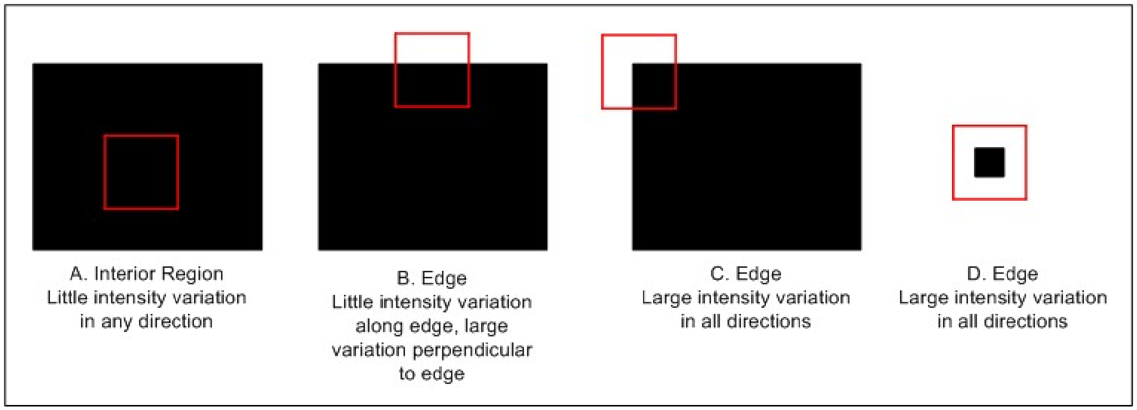
Moravec Operator

- Example (using a 3 x 3 window)

X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	0	0	0	0	0	0	0	0	0	1	1	1	X	X
X	X	0	0	0	0	0	1	1	0	0	1	2	1	X	X
X	X	0	0	0	0	0	2	1	0	0	1	1	1	X	X
X	X	0	0	0	0	0	0	0	0	0	0	0	0	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Moravec Operator

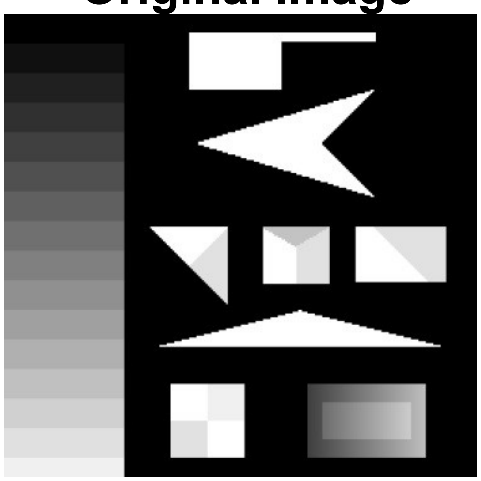
- Why would Moravec operator work?



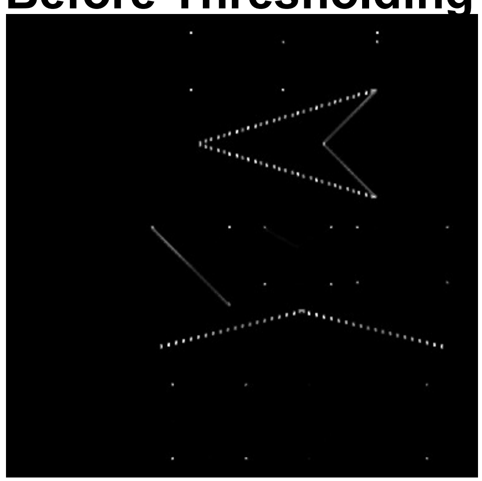
http://kiwi.cs.dal.ca/~dparks/CornerDetection/algorithms.htm

- By setting all points with cornerness below a threshold T to 0, corner points can be detected

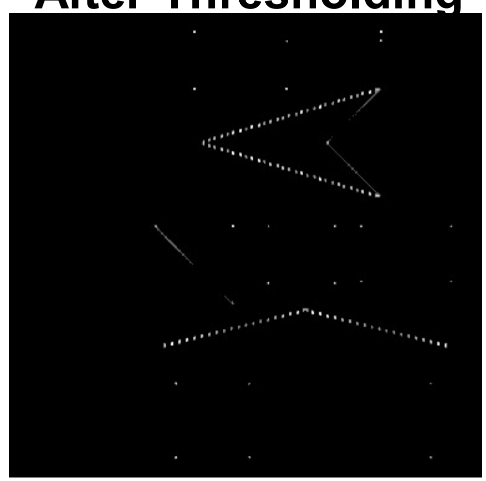
Original Image



Before Thresholding



After Thresholding

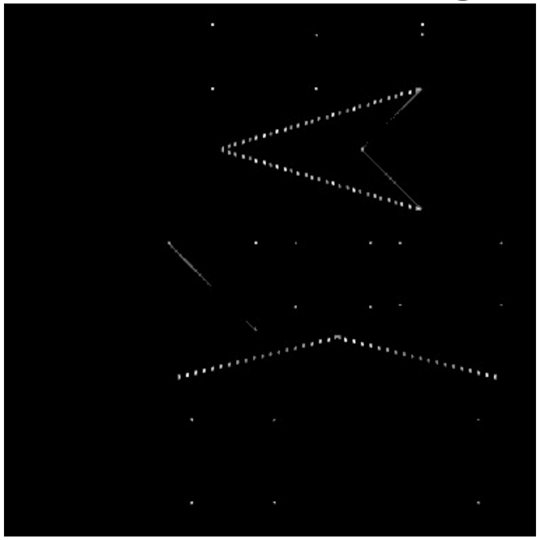


Moravec Operator

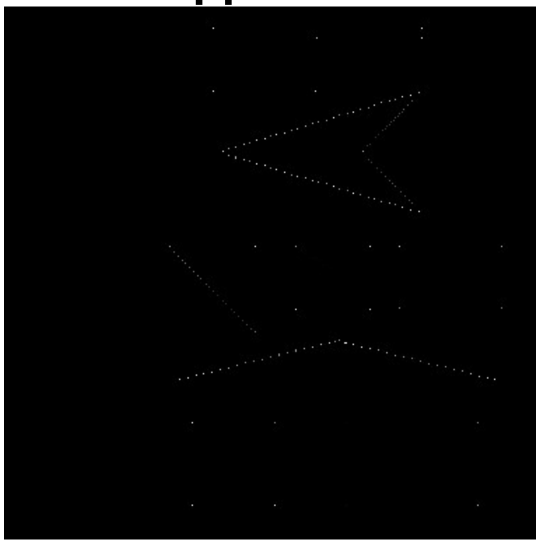
- Finally, non-maximal suppression can be applied

Original Image

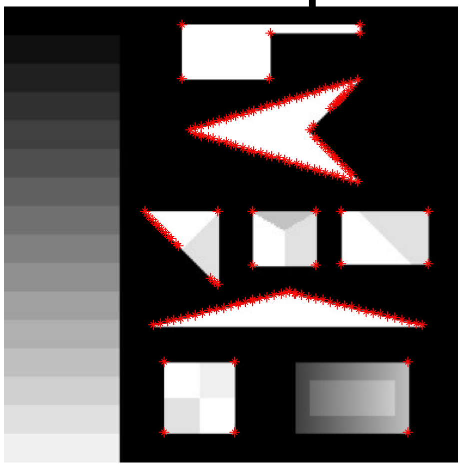
After Thresholding



After Non-maximal Suppression



Final Output



Cornerness of any pixel is set to 0 if it is not larger than the cornerness of all 8-neighbors

Harris Corner Detector

- Improved upon Moravec operator

$$S_W(\Delta x, \Delta y) = \sum_{x_i \in W} \sum_{y_i \in W} \left(f(x_i, y_i) - f(x_i - \Delta x, y_i - \Delta y)\right)^2$$

- Similar to the goal of Moravec operator, we try to find the minimum value of S_W

- This could be found analytically if the shifted image patch is approximated by the first-order Taylor expansion

$$f(x_i - \Delta x, y_i - \Delta y) \approx f(x_i, y_i) + \left[\frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y}\right] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

- Substituting in the expression for S_W

$$S_W(\Delta x, \Delta y) = \sum_{x_i \in W} \sum_{y_i \in W} \left(f(x_i, y_i) - f(x_i, y_i) - \left[\frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y}\right] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}\right)^2$$

Harris Corner Detector

$$
\begin{aligned}
S_W(\Delta x, \Delta y) &= \sum_{x_i \in W} \sum_{y_i \in W} \left(f(x_i, y_i) - f(x_i, y_i) - \left[\frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y} \right] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\
&= \sum_{x_i \in W} \sum_{y_i \in W} \left(- \left[\frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y} \right] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\
&= \sum_{x_i \in W} \sum_{y_i \in W} \left(\left[\frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y} \right] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\
&= \sum_{x_i \in W} \sum_{y_i \in W} [\Delta x, \Delta y] \left(\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix} \right) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\
&= [\Delta x, \Delta y] \left(\sum_{x_i \in W} \sum_{y_i \in W} \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix} \right) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\
&= [\Delta x, \Delta y] \, A_W(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix},
\end{aligned}
$$

Harris Corner Detector

- The goal now is to minimize

$$S_W(\Delta x, \Delta y) = [\Delta x, \Delta y]\, A_W(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

- This is equivalent to finding the eigenvector of A_w corresponding to the minimum eigenvalue
- What are the eigenvectors and eigenvalues?

  For any matrix L, the eigenvectors and eigenvalues are defined as

$$Lf = \lambda f \;\rightarrow\; f^T L f = \lambda$$

f is an eigenvector of L

λ is the eigenvalue corresponding to f

- For A_w, the eigenvector is $\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$ and the eigenvalue is the corresponding S_w

Harris Corner Detector

- Therefore, finding the eigenvalues of A_w would be sufficient to solve the minimization problem where

$$A(x,y) = \begin{bmatrix} \sum_{x_i \in W} \sum_{y_i \in W} \frac{\partial^2 f(x_i,y_i)}{\partial x^2} & \sum_{x_i \in W} \sum_{y_i \in W} \frac{\partial f(x_i,y_i)}{\partial x} \frac{\partial f(x_i,y_i)}{\partial y} \\ \sum_{x_i \in W} \sum_{y_i \in W} \frac{\partial f(x_i,y_i)}{\partial x} \frac{\partial f(x_i,y_i)}{\partial y} & \sum_{x_i \in W} \sum_{y_i \in W} \frac{\partial^2 f(x_i,y_i)}{\partial y^2} \end{bmatrix}$$

- Instead of computing the eigenvalues, Harris suggested using the following approximation

$$R(A) = \det(A) - \kappa\, trace^2(A)$$

where det(A) is the determinant of the local structure matrix A

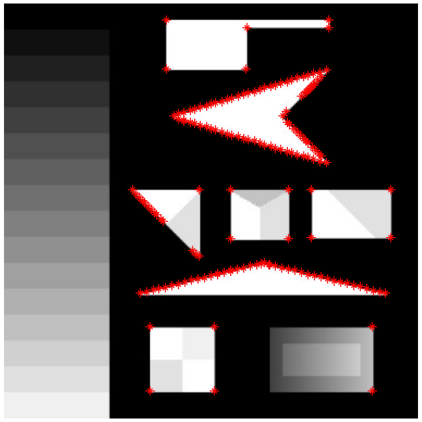
$trace\,(A)$ is the trace of matrix A (sum of elements on the diagonal)

κ is a tunable parameter

Harris Corner Detector

- Algorithm
  1. Filter the image with a Gaussian filter
  2. Estimate intensity gradient in 2 perpendicular directions for each pixel
     $\frac{\partial f(x,y)}{\partial x}$ and $\frac{\partial f(x,y)}{\partial y}$
  3. For each pixel and a given neighborhood window
     - Calculate the local structure matrix A
     - Evaluate the response function $R(A)$
  4. Set all pixels with response less than a threshold T to 0 and perform non-maximal suppression

Moravec Operator



Harris Detector