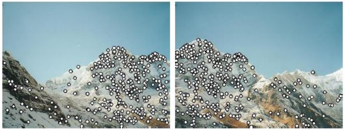
Interest Point Detection
CS-477 Computer Vision

Dr. Mohsin Kamal
Associate Professor
dr.mohsinkamal@seecs.edu.pk

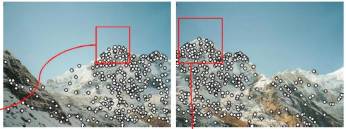
School of Electrical Engineering and Computer Science (SEECS)
National University of Sciences and Technology (NUST), Pakistan

Local features: main components

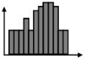
1) **Detection:**
Find a set of distinctive key points.



2) **Description:**
Extract feature descriptor around each interest point as vector.

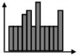


x_1



$$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$$

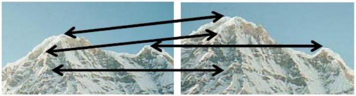
x_2



$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$

3) **Matching:**
Compute distance between feature vectors to find correspondence.

$$d(\mathrm{x}_1, \mathrm{x}_2) < T$$



Use cases

- Automate object tracking
- Stereo calibration
  - Estimation of fundamental matrix
- Motion based segmentation
- Recognition
- 3D object reconstruction
- Robot navigation
- Image retrieval and indexing

Goal: interest operator **repeatability**

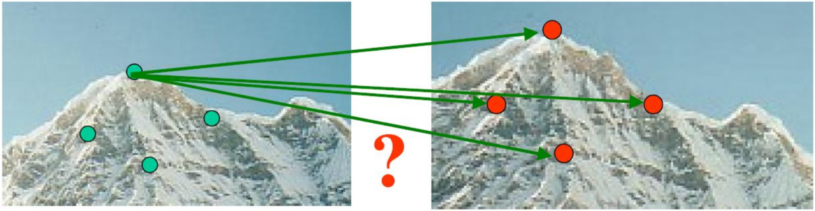
- We want to detect (at least some of) the same points in both images.

No chance to find true matches!

- Yet we have to be able to run the detection procedure independently per image.

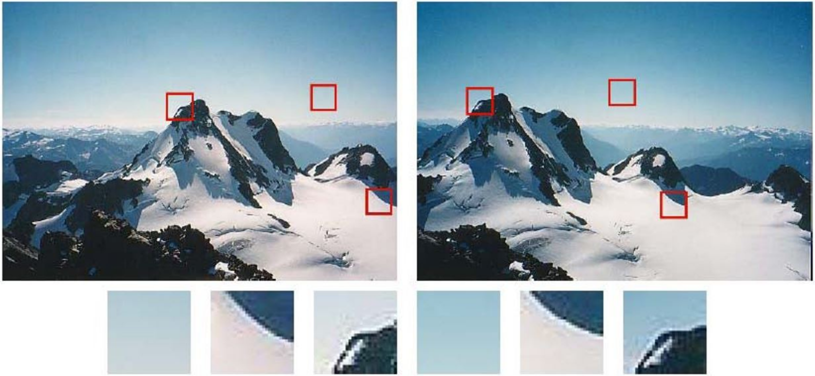
Goal: descriptor distinctiveness

- We want to be able to reliably determine which point goes with which.



- Must provide some invariance to geometric and photometric differences between the two views.

- Some patches can be localized or matched with higher accuracy than others.

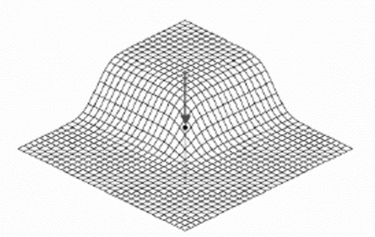


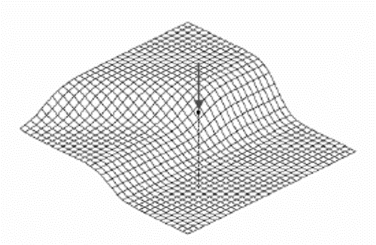
Characteristics of good features

- Repeatability
  - The same feature can be found in each view independently, despite geometric and photometric transformations
- Saliency
  - Each feature is unique/distinctive/unusual, leads to unambiguous matches
- Compactness and efficiency
  - Considerably fewer feature points than image pixels; real-time performance possible
- Locality
  - A feature occupies a relatively small area of the image; robust to clutter and occlusion

What is an interest point

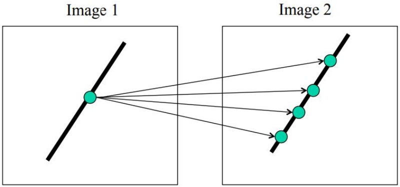
- Expressive texture
  - The point at which the direction of the boundary of object changes abruptly
  - Intersection point between two or more edge segments



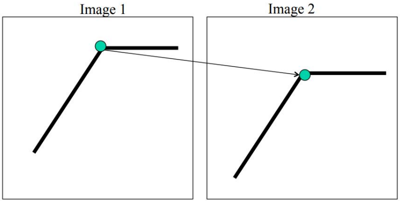


Corners contain more information than lines (edges)

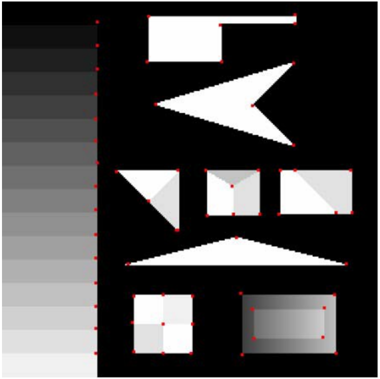
- A point on a line is hard to match



- A corner is easier to match



Synthetic and Real Interest Points



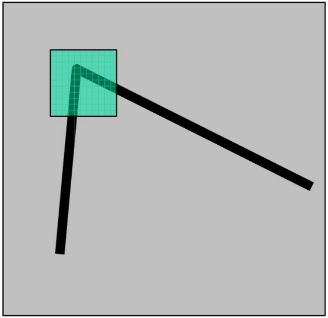
Properties of Interest Point Detectors

- Detect all (or most) true interest points
- No false interest points
- Well localized.
- Robust with respect to noise.
- Efficient detection

Possible Approaches to Corner Detection

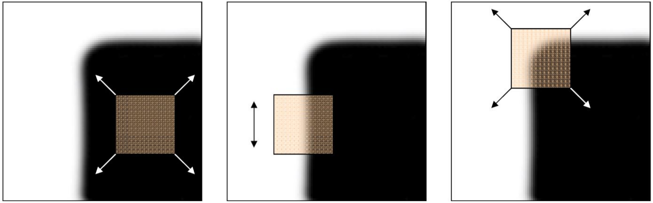
- Based on brightness of images
  - Usually image derivatives
- Based on boundary extraction
  - First step: edge detection
  - Curvature analysis of edges

- Corner point can be recognized in a window
- Shifting a window in any direction should give a large change in intensity



Basic idea

- A good feature point should exhibit gradients in more than one dominant direction.
- Intuitively, when seen through a small window, a small shift in any direction should result in large changes in intensity.



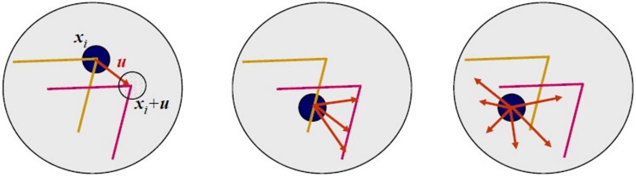
"Flat" region: no change in all directions

"Edge": no change along the edge direction

"Corner": significant change in all directions

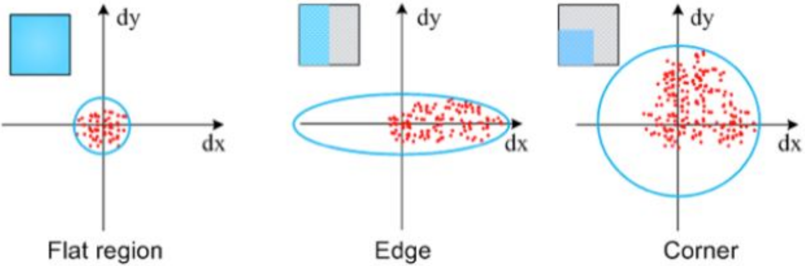
Aperture problem

The reason why corner points are good interest points because they do not have any aperture problem.



(a) (b) (c)

Distribution of image gradients



Derivative distribution of different regions

Correlation vs. Sum of Squared Difference (SSD)

$$\underset{min}{\text{SSD}} = \sum_k \sum_l (f(k,l) - h(i+k, j+l))^2$$

$$\underset{min}{\text{SSD}} = \sum_k \sum_l \left(f(k,l)^2 - 2h(i+k, j+l)f(k,l) + h(i+k, j+l)^2\right)$$

$$\underset{min}{\text{SSD}} = \sum_k \sum_l (-2h(i+k, j+l)f(k,l))$$

$$\underset{max}{\text{SSD}} = \sum_k \sum_l (2h(i+k, j+l)f(k,l))$$

The above equation represents the correlation (studied previously). Hence,

$$\underset{max}{\text{Correlation}} = \sum_k \sum_l (h(i+k, j+l)f(k,l))$$

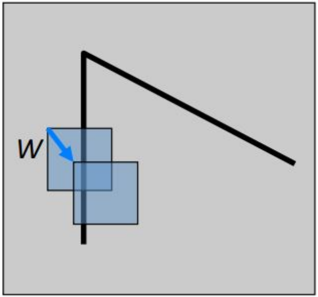
Auto-correlation can be represented as,

$$f \bigotimes f = \sum_k \sum_l (f(k,l)f(i+k, j+l))$$

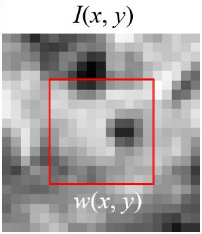
- In Harris detector, we want to find the change of intensity for the shift/displacement in (u,v)
- Change in appearance of window w(x,y) for shift [u,v], measured by a weighted SSD as:

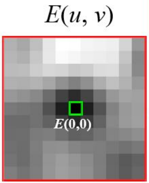
$$E(u,v) = \sum_{(x,y)\in W} w(x,y)\,[I(x+u,y+v) - I(x,y)]^2$$

- $I(x,y)$ = Intensity at x,y
- $I(x,y)$ = Shifted intensity
- $w(x,y)$ = window function
  - Uniform
  - Gaussian



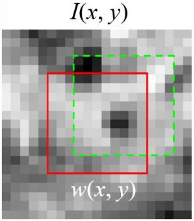
w(x,y) here is 11x11-pixel sized region. Thus E(u,v) is also 11x11, since we can only shift the center of the red window 5 positions upward and downward and similarly leftward and rightward

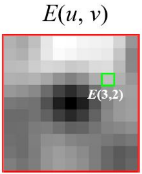




E(0,0) is always black (since there's no change in the window when it hasn't yet moved)

NOTE: E(u,v) is essentially constructed for a *given local* image region (i.e., window centered at a given pixel x,y), for a *defined* shift range (u,v = -5,-4, …,0, 1, …, +4,+5)





For,

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

We want to find out how this function behaves for small shifts (u,v) for a window (x,y) centered at each pixel

But this is very slow to compute naively:

$$\textit{no of mults} = (\text{window\_width}^2 \times \text{shift\_range}^2 \times \text{image\_width}^2)$$
$$= (11^2 x 11^2 x 480 x 640) = \text{ 5 billion multiplications}$$

i.e., 14.6 thousand per pixel

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

Taylor Series[1] approximation of $I(x+u, y+v)$ is applied as

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

$$E(u,v) = \sum_{x,y} w(x,y)\left[uI_x + vI_y\right]^2$$

$$E(u,v) = \sum_{x,y} w(x,y)\left[\begin{bmatrix} u & v \end{bmatrix}\begin{bmatrix} I_x \\ I_y \end{bmatrix}\right]^2$$

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

$$E(u,v) = \begin{bmatrix} u & v \end{bmatrix}\left[\sum_{x,y} w(x,y)\begin{bmatrix} I_x \\ I_y \end{bmatrix}\begin{bmatrix} I_x & I_y \end{bmatrix}\right]\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

[1] https://www.cs.cornell.edu/courses/cs4670/2016sp/lectures/lec10_features2_web.pdf

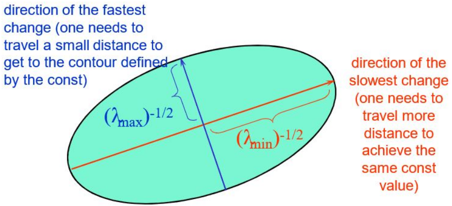
$$E(u,v) = \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

By solving *M* in previous slide, we get:

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

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

- $E(u,v)$ is an equation of an ellipse, where M is the covariance
- let λ_1 and λ_2 be the eigenvalues of M



Eigen values and Eigen vectors

- An eigenvector corresponds to the real non zero eigenvalues which point in the direction stretched by the transformation whereas eigenvalue is considered as a factor by which it is stretched. In case, if the eigenvalue is negative, the direction of the transformation is negative.
- To find eigen values of a matrix A first find the roots of:
$$\det(A - \lambda I) = 0$$
- Then solve the following linear system for each eigen value to find corresponding eigen vector
$$(A - \lambda I)x = 0$$

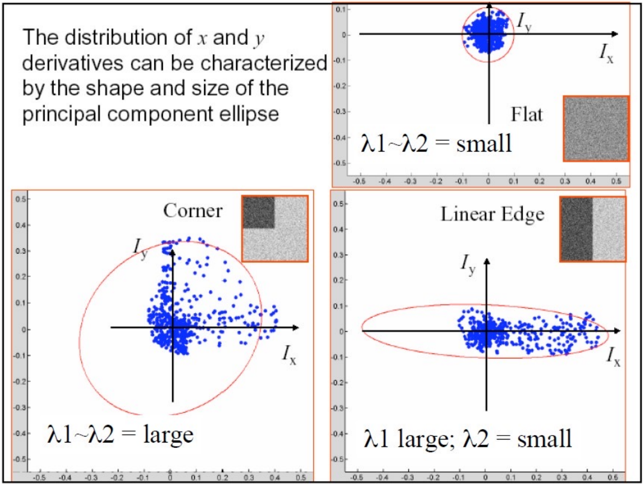
Do it!

$$A = \begin{bmatrix} -1 & 2 & 0 \\ 0 & 3 & 4 \\ 0 & 0 & 7 \end{bmatrix}$$

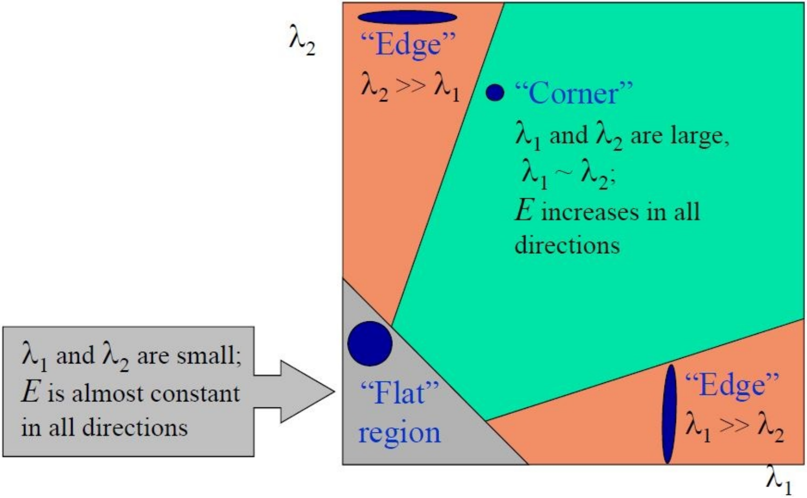
Eigen Values: $\lambda_1 = 7, \lambda_2 = 3, \lambda_3 = -1$

Eigen vectors: $x_1 = \begin{bmatrix} 1 \\ 4 \\ 4 \end{bmatrix}, x_2 = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}, x_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

Fitting Ellipse to Each Set of Points



Classification of image points using eigenvalues of M



Cornerness response measure

Cornerness:

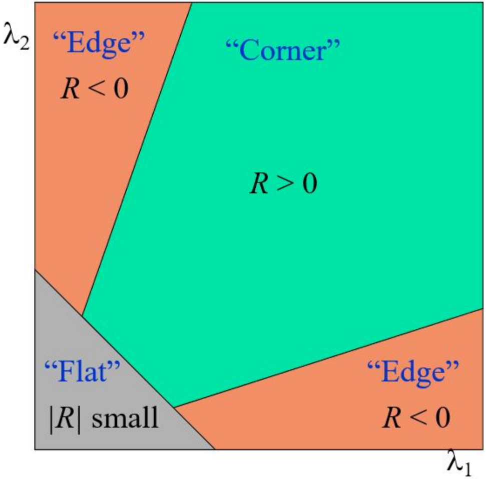
$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$

where k is a constant with values ranging between 0.04 to 0.06

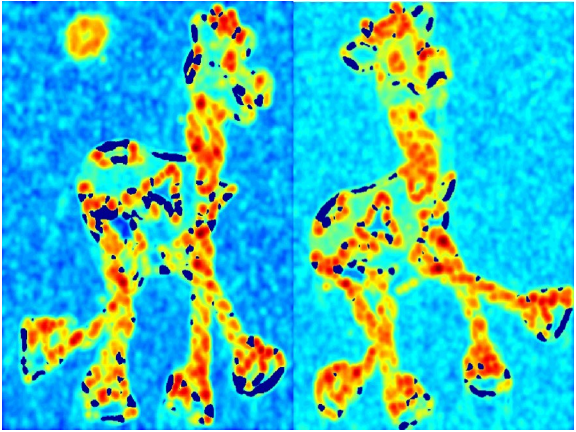
Instead of computing eigenvalues, we can use determinant and trace of M i.e.,

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

- R depends only on eigenvalues of M
- R is large for a corner
- R is negative with large magnitude for an edge
- $|R|$ is small for a flat region

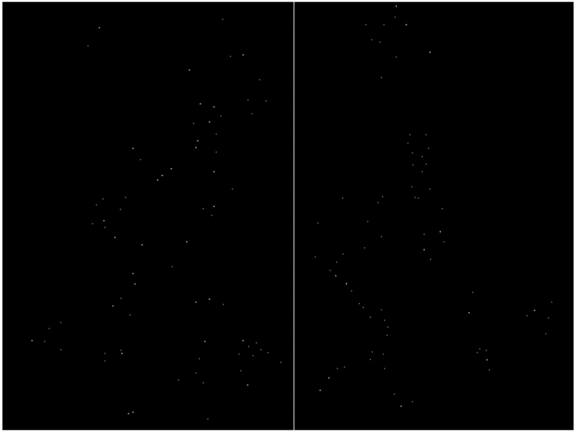


Compute corner response



Find points with large corner response i.e., $R >$Threshold

Take only the points of local maxima of R



If pixel value is greater than its neighbors then it is a local maxima.

Final result

Example of Harris detector

Consider the following image:

I

0	0	1	4	9
1	0	5	7	11
1	4	9	12	16
3	8	11	14	16
8	10	15	16	20

d/dx

-1	0	1

d/dy

-1
0
1

Compute the Harris matrix

$$H = \sum_{(x,y)\in W} \begin{bmatrix} I_x(x,y)^2 & I_x(x,y)I_y(x,y) \\ I_x(x,y)I_y(x,y) & I_y(x,y)^2 \end{bmatrix}$$

for the 3 by 3 highlighted window. In the above formula $I_x = dI/dx$, $I_y = dI/dy$, and W is the window highlighted in the image.

A) First, compute the derivatives using the differentiation kernels shown above. No normalization (division by 2) is needed. (5 points).

$I_x = dI/dx$

X	X	X	X	X
X	4	7	6	X
X	8	8	7	X
X	8	6	5	X
X	X	X	X	X

$I_y = dI/dy$

X	X	X	X	X
X	4	8	8	X
X	8	6	7	X
X	6	6	4	X
X	X	X	X	X

Example of Harris detector

$$H = \sum_{(x,y)\in W} \begin{bmatrix} I_x(x,y)^2 & I_x(x,y)I_y(x,y) \\ I_x(x,y)I_y(x,y) & I_y(x,y)^2 \end{bmatrix}$$

$$\sum_{(x,y)\in W} I_x(x,y)^2 = 4^2+7^2+6^2+8^2+8^2+7^2+8^2+6^2+5^2 = 403$$

$$\sum_{(x,y)\in W} I_y(x,y)^2 = 4^2+8^2+8^2+8^2+6^2+7^2+6^2+6^2+4^2 = 381$$

$$\sum_{(x,y)\in W} I_x(x,y)\,I_y(x,y) = 4*4+7*8+6*8+8*8+8*6+7*7+8*6+6*6+5*4 = 385$$

$$H = \begin{matrix}[403 & 385] \\ [385 & 381]\end{matrix}$$

C) Compute the Harris cornerness score $C = det(H) - k\, trace(H)^2$ for $k = 0.04$. What do we have here? A corner? An edge? Or a flat area? Why? (5 points)

$$C = det(H) - K\, trace(H)^2 = 5318 - 0.04 * (784)^2 = -19268.24$$

A negative Harris score indicates an edge.

Other versions of Harris detector

- Triggs: $R = \lambda_1 - \alpha\lambda_2$
- Szeliski (Harmonic mean): $R = \frac{\det(M)}{trace(M)} = \frac{\lambda_1\lambda_2}{\lambda_1+\lambda_2}$
- Shi-Tomasi: $R = \lambda_1$