ARAVIND
2019102024

Lecture 18
Image Segmentation

Harris Corner:

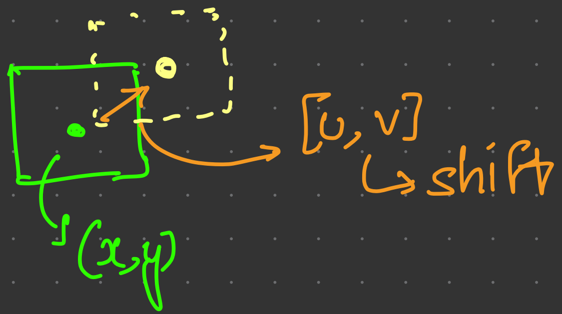
→ How intensity changes as we slide across a window.

→ In flat region, no change in any direction

→ In edge region, no change in average intensity

→ In we have window centered in corner, change in gradients is larger

Math behind its:



So,
change of intensity:

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

↓ Weight [window fn]

↳ change of intensity as function of shift

Example: ① 1 - in window
0 - outside window

② Gaussian weights

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

u,v is very small

So,

$I(x+u, v+y) = I(x,y)$
$u I_x + v I_y$
$+$
$uv I_x I_y$

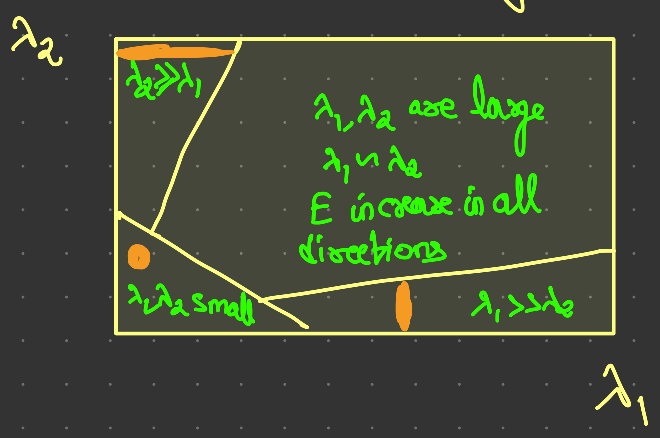
Using bilinear approximation,

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

→ 2x2 matrix computed from image derivatives

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

M is basis for how we classify corners



If λ_1, λ_2 are small, shift is constant ⇒ Flat region

If $\lambda_2 >> \lambda_1$, shift is along λ_2 axis ⇒ Edge

If $\lambda_1 >> \lambda_2$, shift is along λ_1 axis ⇒ Edge

If λ_1 and λ_2 are large, $\lambda_1 \sim \lambda_2$ shift causes increase in intensity in all directions

⇒ Corner

$$\left.\begin{aligned} \lambda_1 + \lambda_2 &= tr(M) \\ \lambda_1 \lambda_2 &= det(M) \end{aligned}\right\}$$ Quantity in terms of image derivative [intensity gradients

To give a "score"

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

→ k = Empirical constant is 0.04 – 0.06

↳ Measure of corner response

→ R depends only on eigenvalues

Edge: -ve with large magnitude
Corner: large for a corner
Flat Region: $|R|$ is small

Overall Algorithm:
→ Find points with large corner response R (R > threshold)
→ Take points of local maxima of R

→ Applications -
→ Finding corresponding points using set matching algorithms

→ Properties:
→ Invariant to rotation [as they're based on eigenvalues which stay same]

→ Partial invariance to affine intensity change
↳ Reason: Only derivatives are used ⇒ Invariant to intensity shift
Shift: $I \leftarrow a + I$
Intensity scale: $I \rightarrow Ia$

→ Non-invariant to image size
↳ Can't accurately localise
↳ Can be solved by SIFT and other descriptors

IMAGE SEGMENTATION

→ Segmentation is partition of image I into set of regions S satisfying:

→ $\cup S_i = S$ ⟶ Partition covers whole image

→ $S_i \cap S_j = \phi$ ⟶ No regions intersect

→ $\forall S_i, P(S_i) = True$ ⟶ Homogenity predicate

→ $P(S_i \cup S_j) = false$ ⟶ Union of adjacent regions

→ $i \neq j$, S_i adjacent to S_j ⟶ Doesn't satisfy homogenity

→ Threshold based approach

→ Separate pixel associated with object of interest from background

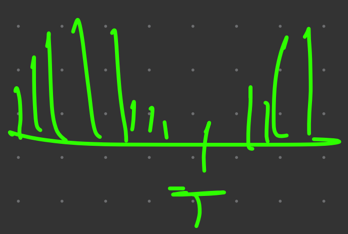
→ Given $f(x,y)$, segment $g(x,y)$

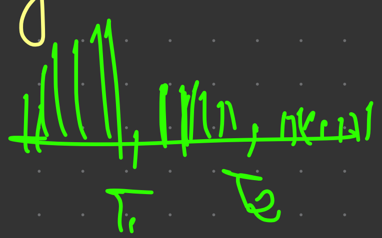
$$g(x,y) = \begin{cases} 1, & f(x,y) > T \\ 0, & f(x,y) \leq T \end{cases}$$

T → Constant over entire image [Global Thresholding]
Variable " " " [Variable Thresholding]

How to find T?

→ Use image histogram / Check for some intensity that separates bg & fg

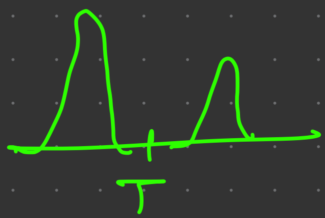




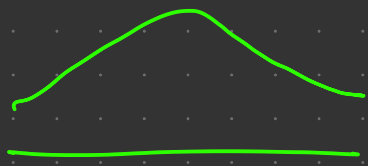
Role of noise:

No noise

Little noise



Most noise



So, need to filter out noise

Role of illumination and reflectances

After multiplication with a ramp, won't get good output

Finding T:

Iterative approach:

→ Select initial estimate of gt T

→ Segment using T

→ Produces two groups of pixels (G_1, G_2)

→ Compute average intensity values m_1, m_2 for pixels in G_1 and G_2

→ Compute new threshold value $T_{new} = \frac{(m_1 + m_2)}{2}$

→ If $|T_{new} - T| <$ epsilon, stop

Else set $T = T_{new}$

Usually located at valley (one of the valleys)

If bg I is very close to Region of Interest (RoI), can't separate (no valley)

Otsu's method:

→ Based on histograms
→ Automatically find optimal threshold maximising b/w class & variance
→ Relate the scatter of an image

Intra and Inter class variance:

$\sigma_A^2 > \sigma_B^2$

We want high inter-class variance and low intra-class variance

Variance = Measure of region homogenity
Regions with high homogenity ⟹ Low variance

Otsu's method minimise intra-class variance

Algorithm:

→ Consider possible thresholds T (0, 255)
→ For each T,
1. Find intra-class variance for class 1 (< T), class 2 (≥ T)

MINIMISING INTRA-CLASS VARIANCE ⟹ MAXIMISING BETWEEN CLASS VARIANCE

Intra-class variance:

$$\sigma^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)$$

Between class variance ← $\sigma_B^2(k) = P_1(k)\left[m_1(k) - m_G\right]^2 + P_2(k)\left(m_2(k) - m_G\right)^2 \}$

Otsu's limitation:

→ Using global thresholding but can handle multiple thresholds

Per pixel variable thresholdings

↳ Not fully gone over in class. To be referred from slides

Choosing thresholding Algorithms

→ Based on size of fg/AOI:

↳ Small ⟹ Adaptive/Local

↳ Large ⟹ Global