Kanade-Lucas-Tomasi (KLT) Tracker

Computer Vision (CS0029)

Motivation

KLT Tracker

An iterative image registration technique
1981

Detection and tracking of feature points
1991

Good features to track
1994

Two Questions

- How should we track them from frame to frame? (Lucas and Kanade)
  - Method for aligning (tracking) an image patch
- How should we select features? (Tomasi and Kanade)
  - Method for choosing the best feature (image patch) for tracking

What Are Good Feature to Track?

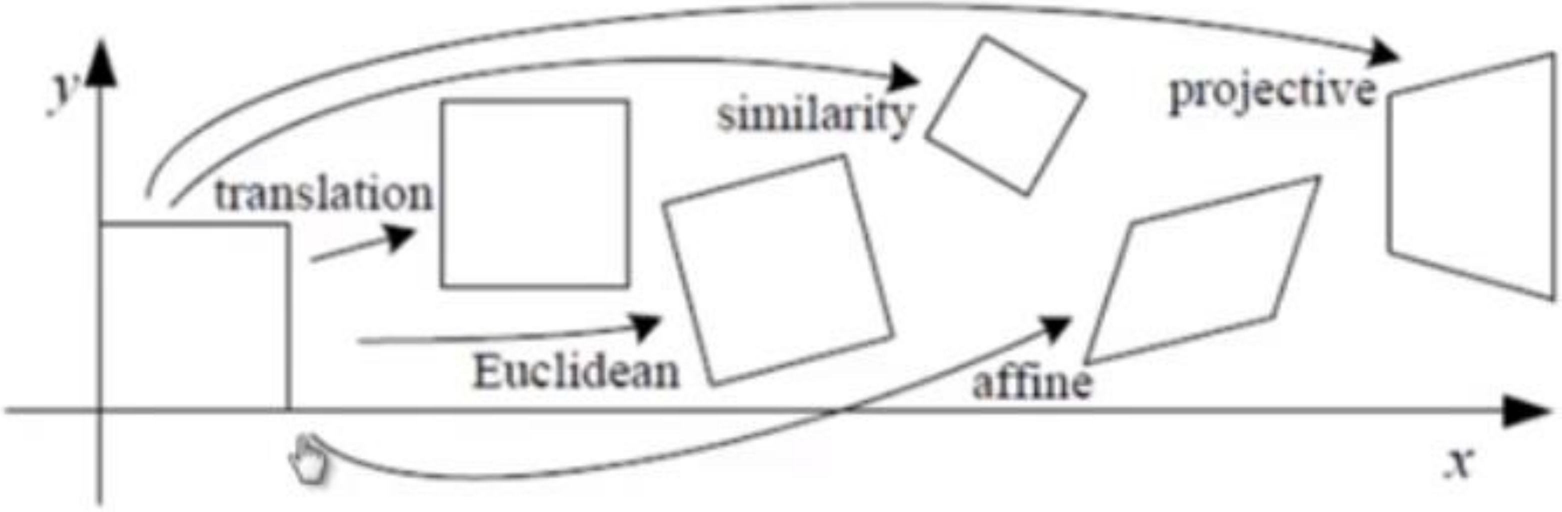
- Intuitively, we want to avoid smooth regions and edges. But is there a more is principled way to define good features?

What Are Good Feature to Track?

- Intuitively, we want to avoid smooth regions and edges. But is there a more is principled way to define good features?

- Can be derived from the tracking algorithm
  - A feature is good if it can be tracked well

Basic 2D Transformation



2D Transformation

Translation

$$x' = x + b_1$$
$$y' = y + b_2$$

$$W(\boldsymbol{x};\boldsymbol{p}) = (x + b_1, y + b_2)$$

$$W(\boldsymbol{x};\boldsymbol{p}) = \begin{bmatrix} 1 & 0 & b_1 \\ 0 & 1 & b_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rigid (rotation)

$$x' = x\cos\theta - y\sin\theta + b_1$$
$$y' = x\,sin\,\theta - y\,cos\,\theta + b_2$$

$$W(\boldsymbol{x};\boldsymbol{p}) = (x\cos\theta - y\sin\theta + b_1, x\,sin\,\theta - y\,cos\,\theta + b_2)$$

$$W(\boldsymbol{x};\boldsymbol{p}) = \begin{bmatrix} cos\theta & -sin\theta & b_1 \\ sin\theta & cos\theta & b_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$= [R|t]_{2\times 3} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

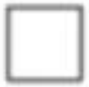
Affine

$$x' = a_1x + a_2y + b_1$$
$$y' = a_3x + a_4y + b_2$$

$$W(\boldsymbol{x};\boldsymbol{p}) = (a_1x + a_2y + b_1, a_3x + a_4y + b_2)$$

$$W(\boldsymbol{x};\boldsymbol{p}) = \begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2D Transformation

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\left[\, \boldsymbol{I} \mid \boldsymbol{t} \,\right]_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\left[\, \boldsymbol{R} \mid \boldsymbol{t} \,\right]_{2\times 3}$	3	lengths	
similarity	$\left[\, s\boldsymbol{R} \mid \boldsymbol{t} \,\right]_{2\times 3}$	4	angles	
affine	$\left[\, \boldsymbol{A} \,\right]_{2\times 3}$	6	parallelism	
projective	$\left[\, \tilde{\boldsymbol{H}} \,\right]_{3\times 3}$	8	straight lines	

Derivative & Gradient

- Function : $f(x)$

- Derivative: $f'(x) = \frac{df}{dx}$

- Function: $f(x_1, x_2, \dots, x_n)$

- Gradient: $\nabla f(x_1, x_2, \dots, x_n) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3}, \dots, \frac{\partial f}{\partial x_n}\right)$

Jacobian

$$F(x_1, x_2, \dots, x_n) = (f_1(x_1, x_2, \dots, x_n), f_2(x_1, x_2, \dots, x_n), \dots, f_m(x_1, x_2, \dots, x_n))$$

Vector valued Function

Derivative?

$$J(F) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \cdots & \cdots & \frac{\partial f_2}{\partial x_n} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ \frac{\partial f_m}{\partial x_1} & \cdots & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Displacement of Model Jacobians

Translation

$$x' = x + b_1$$
$$y' = y + b_2$$
$$W(\boldsymbol{x};\boldsymbol{p}) = (x + b_1, y + b_2)$$

$$\frac{\partial W}{\partial P} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Rigid (rotation)

$$x' = x\cos\theta - y\sin\theta + b_1$$
$$y' = x\,sin\,\theta - y\,cos\,\theta + b_2$$
$$W(\boldsymbol{x};\boldsymbol{p}) = (x\cos\theta - y\sin\theta + b_1, x\,sin\,\theta - y\,cos\,\theta + b_2)$$

$$\frac{\partial W}{\partial P} = \begin{bmatrix} 1 & 0 & -xsin\theta - ycos\theta \\ 0 & 1 & xcos\theta - ysin\theta \end{bmatrix}$$

Affine

$$x' = a_1x + a_2y + b_1$$
$$y' = a_3x + a_4y + b_2$$
$$W(\boldsymbol{x};\boldsymbol{p}) = (a_1x + a_2y + b_1, a_3x + a_4y + b_2)$$

$$\frac{\partial W}{\partial P} = \begin{bmatrix} 1 & 0 & x & y & 0 & 0 \\ 0 & 1 & 0 & 0 & x & y \end{bmatrix}$$

Finding Alignment

- Find $\boldsymbol{p}$ s.t. following is minimized
  - $\sum_x \left[I\big(W(\boldsymbol{x};\boldsymbol{p})\big) - T(x)\right]^2$
- Assume initial estimate of p is known, find $\Delta\boldsymbol{p}$
  - $\sum_x \left[I\big(W(\boldsymbol{x};\boldsymbol{p}+\Delta\boldsymbol{p}\,)\big) - T(x)\right]^2$
- Find Taylor Series
  - $\sum_x \left[I\big(W(\boldsymbol{x};\boldsymbol{p}\,)\big) + \nabla I\,\frac{\partial W}{\partial_P}\boldsymbol{\Delta p} - T(x)\right]^2$
  - $\nabla_I = \begin{bmatrix} I_x \\ I_y \end{bmatrix}$

Finding Alignment

- $\sum_x \left[I\big(W(\boldsymbol{x};\boldsymbol{p}\,)\big) + \nabla I \frac{\partial W}{\partial_P} \boldsymbol{\Delta p} - T(x) \right]^2$
- Differentiate wrt $\boldsymbol{\Delta p}$ and equate it to zero
    - $2\sum_x \left[\nabla I \frac{\partial W}{\partial_P} \boldsymbol{\Delta p} \right]^T \left[I\big(W(\boldsymbol{x};\boldsymbol{p}\,)\big) + \nabla I \frac{\partial W}{\partial_P} \boldsymbol{\Delta p} - T(x) \right]$
- And equate it to zero to find
    - $2\sum_x \left[\nabla I \frac{\partial W}{\partial_P} \boldsymbol{\Delta p} \right]^T \left[I\big(W(\boldsymbol{x};\boldsymbol{p}\,)\big) + \nabla I \frac{\partial W}{\partial_P} \boldsymbol{\Delta p} - T(x) \right] = 0$
    - $\boldsymbol{\Delta p} = H^{-1} 2\sum_x \left[\nabla I \frac{\partial W}{\partial_P} \right]^T \left[T(x) - I\big(W(\boldsymbol{x};\boldsymbol{p}\,)\big) \right] = 0$
    - $H^{-1} = \sum_x \left[\nabla I \frac{\partial W}{\partial_P} \right]^T \left[\nabla I \frac{\partial W}{\partial_P} \right]$

Summary of LKT Tracker

$$\boldsymbol{\Delta p} = H^{-1} 2 \sum_{x} \left[\nabla I \frac{\partial W}{\partial_P} \right]^T \left[T(x) - I\big(W(\boldsymbol{x}; \boldsymbol{p}\,) \big) \right] = 0$$

- Warp I with $W(x;p)$
- Subtract I from T
- Compute image gradient ∇I
- Evaluate the Jacobian descent $\frac{\partial W}{\partial_P}$
- Compute steepest descent $\nabla I \frac{\partial W}{\partial_P}$
- Compute inverse Hessian $\mathrm{H} = \sum_x \left[\nabla I \frac{\partial W}{\partial_P} \right]^T \left[\nabla I \frac{\partial W}{\partial_P} \right]$
- Multiply steepest descent with error $\sum_x \left[\nabla I \frac{\partial W}{\partial_P} \right]^T \left[T(x) - I\big(W(\boldsymbol{x}; \boldsymbol{p}\,) \big) \right]$
- Compute $\boldsymbol{\Delta p}$
- Update parameters $\boldsymbol{p} \rightarrow \boldsymbol{p} + \boldsymbol{\Delta p}$

Stability of Gradient Descent iterations

- It depends on
  - $\boldsymbol{\Delta p} = H^{-1} 2 \sum_x \left[\nabla I \frac{\partial W}{\partial_P}\right]^T [T(x) - I(W(\boldsymbol{x}; \boldsymbol{p}))] = 0$
- Inverting the H
  - $\mathrm{H} = \sum_x \left[\nabla I \frac{\partial W}{\partial_P}\right]^T \left[\nabla I \frac{\partial W}{\partial_P}\right]$
- When does the inversion fail?
  - H is singular
  - What does it means?

Eigenvalue of Singular Matrix

- H is singular
- $\det(H) = 0$
- $\det(H - 0 * I) = 0$
- 0 is eigenvalue of A

- How is that related to image content?

Hessian for Translation Motion

$$H^{-1} = \sum_x \left[\nabla I \frac{\partial W}{\partial P}\right]^T \left[\nabla I \frac{\partial W}{\partial P}\right]$$

Jacobian of translation transformation

$$\frac{\partial W}{\partial P} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$H^{-1} = \sum \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} = \sum \begin{bmatrix} (I_x)^2 & I_x I_y \\ I_x I_y & (I_y)^2 \end{bmatrix}$$

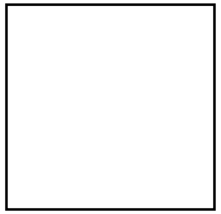
H^{-1} is calculated from all points in a small image patch

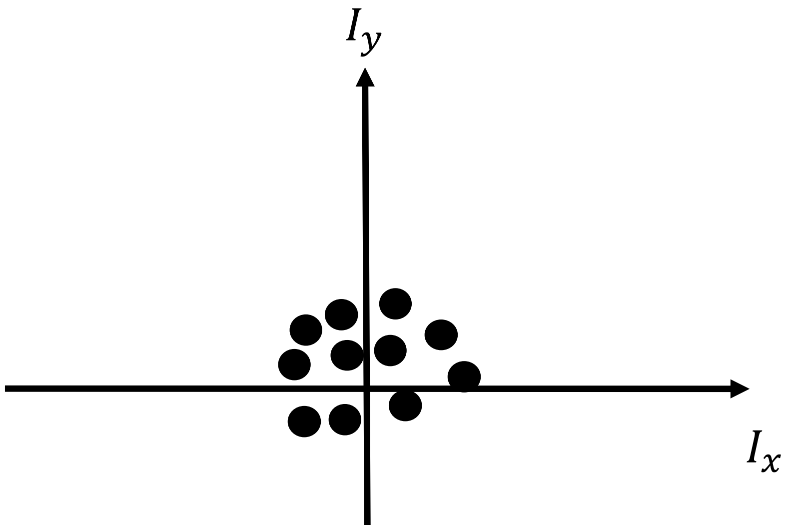
Find Good Features

- Good feature = track well = we must be able to compute H^{-1}
- Intuitively, good features is not a smooth region and a edge
  - corner
- For a small image patch, we can calculate I_x and I_y on every pixel
- Covariance Matrix from all I_x and I_y in the image patch
  - $AA^T = \begin{bmatrix} I_x^1 & I_x^2 & \dots & I_x^N \\ I_y^2 & I_y^2 & \dots & I_y \end{bmatrix} \begin{bmatrix} I_x^1 & I_y^1 \\ I_x^2 & I_y^2 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ I_x^N & I_y^N \end{bmatrix} = \sum \begin{bmatrix} (I_x)^2 & I_x I_y \\ I_x I_y & (I_y)^2 \end{bmatrix} = \sum \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} = H^{-1}$
- If we run eigen decomposition for the above covariance matrix, what we will get?

Find Good Features

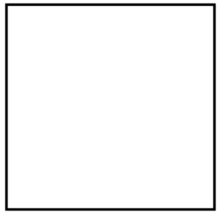
- If we have this image patch (smooth region), what is the distribution of all I_x and I_y in the patch

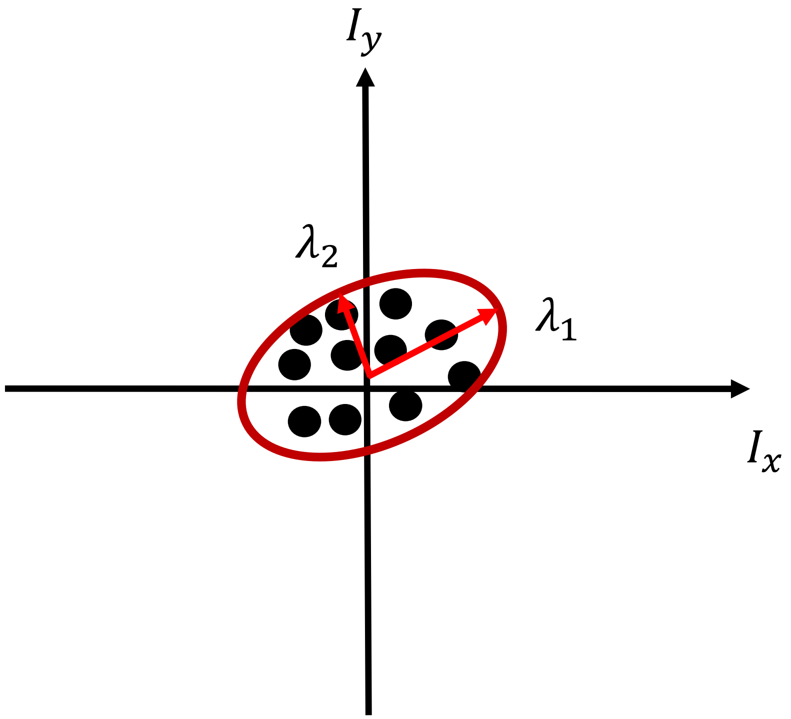




Find Good Features

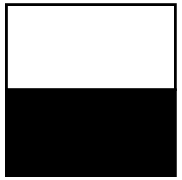
- If we have this image patch (smooth region), what is the distribution of all I_x and I_y in the patch
  - λ_1 and λ_2 are small in this case

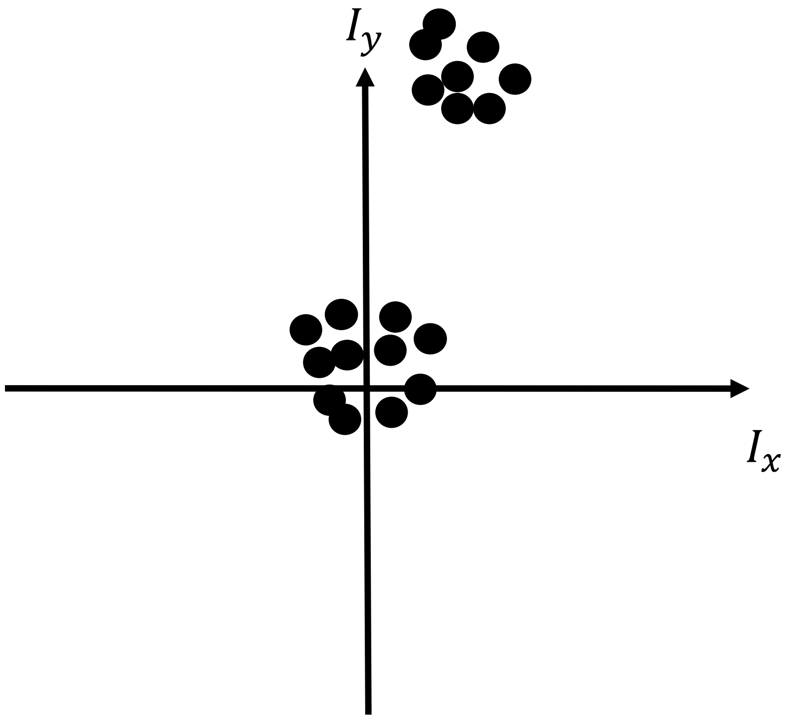




Find Good Features

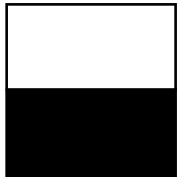
- If we have this image patch (edge), what is the distribution of all I_x and I_y in the patch

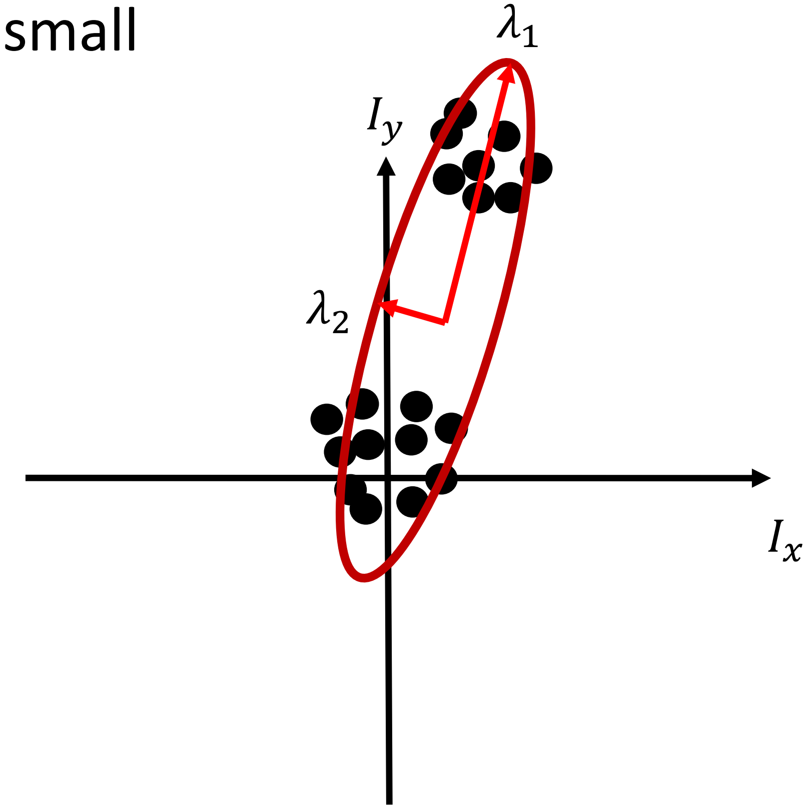




Find Good Features

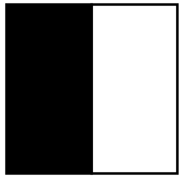
- If we have this image patch (edge), what is the distribution of all I_x and I_y in the patch
  - λ_1 is large but λ_2 is small

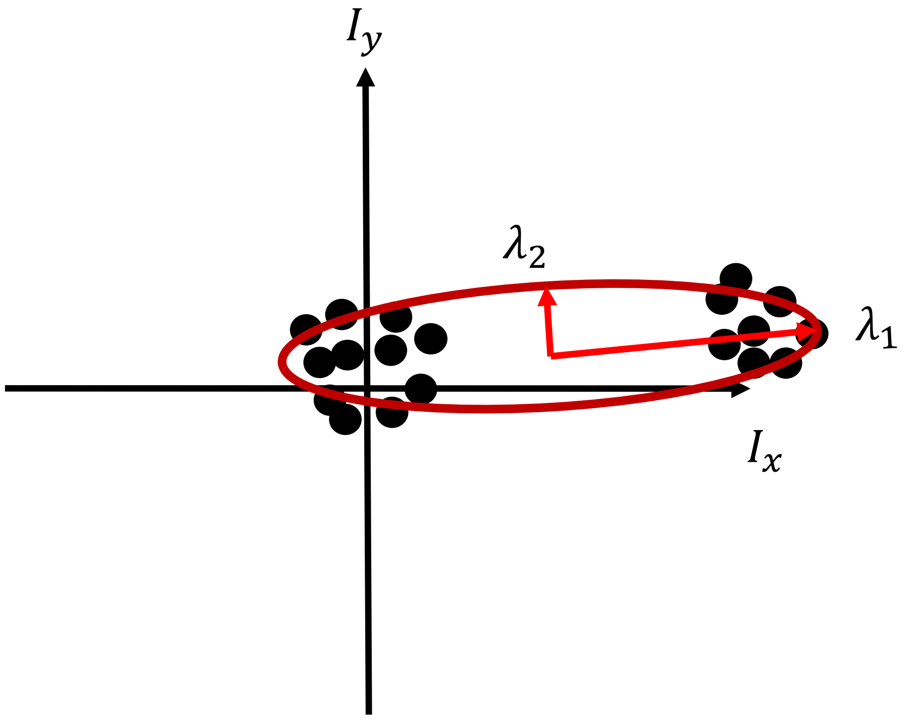




Find Good Features

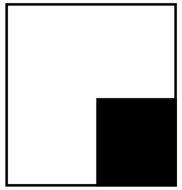
- If we have this image patch (edge), what is the distribution of all I_x and I_y in the patch
  - λ_1 is large but λ_2 is small

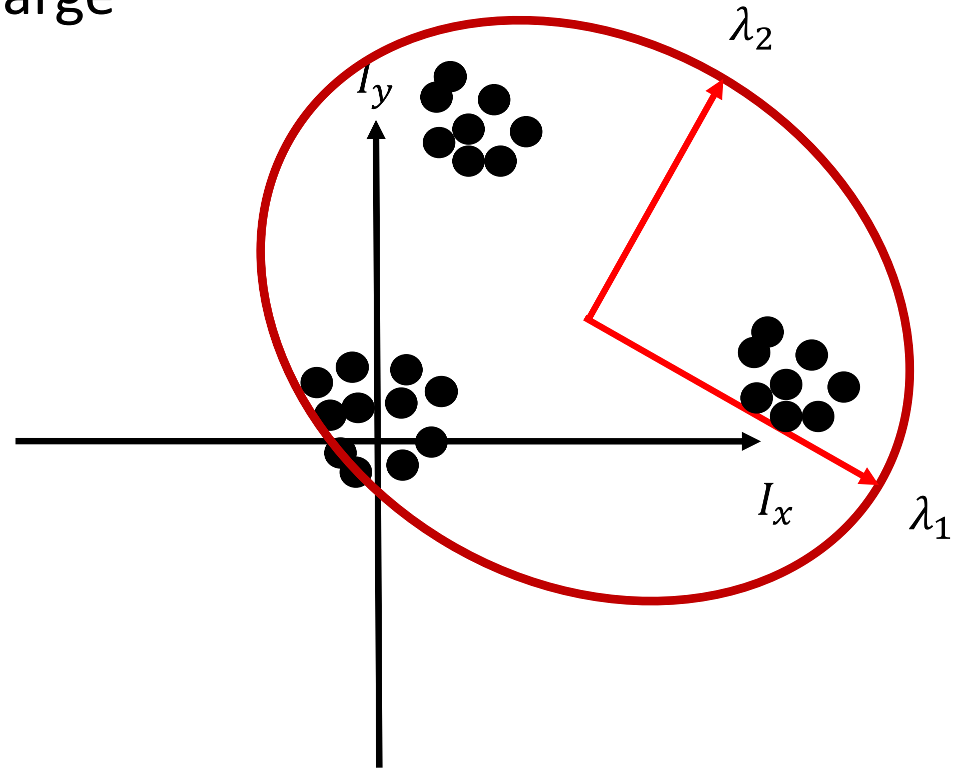




Find Good Features

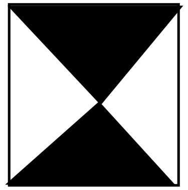
- If we have this image patch (corner), what is the distribution of all I_x and I_y in the patch
  - Both λ_1 and λ_2 are large

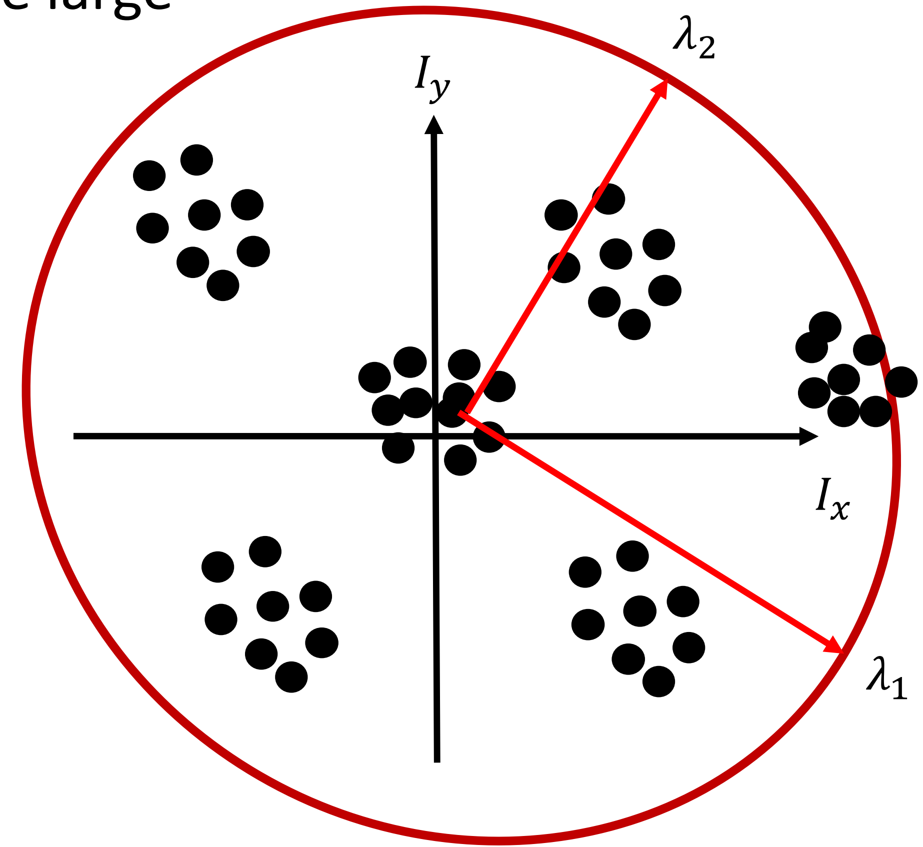




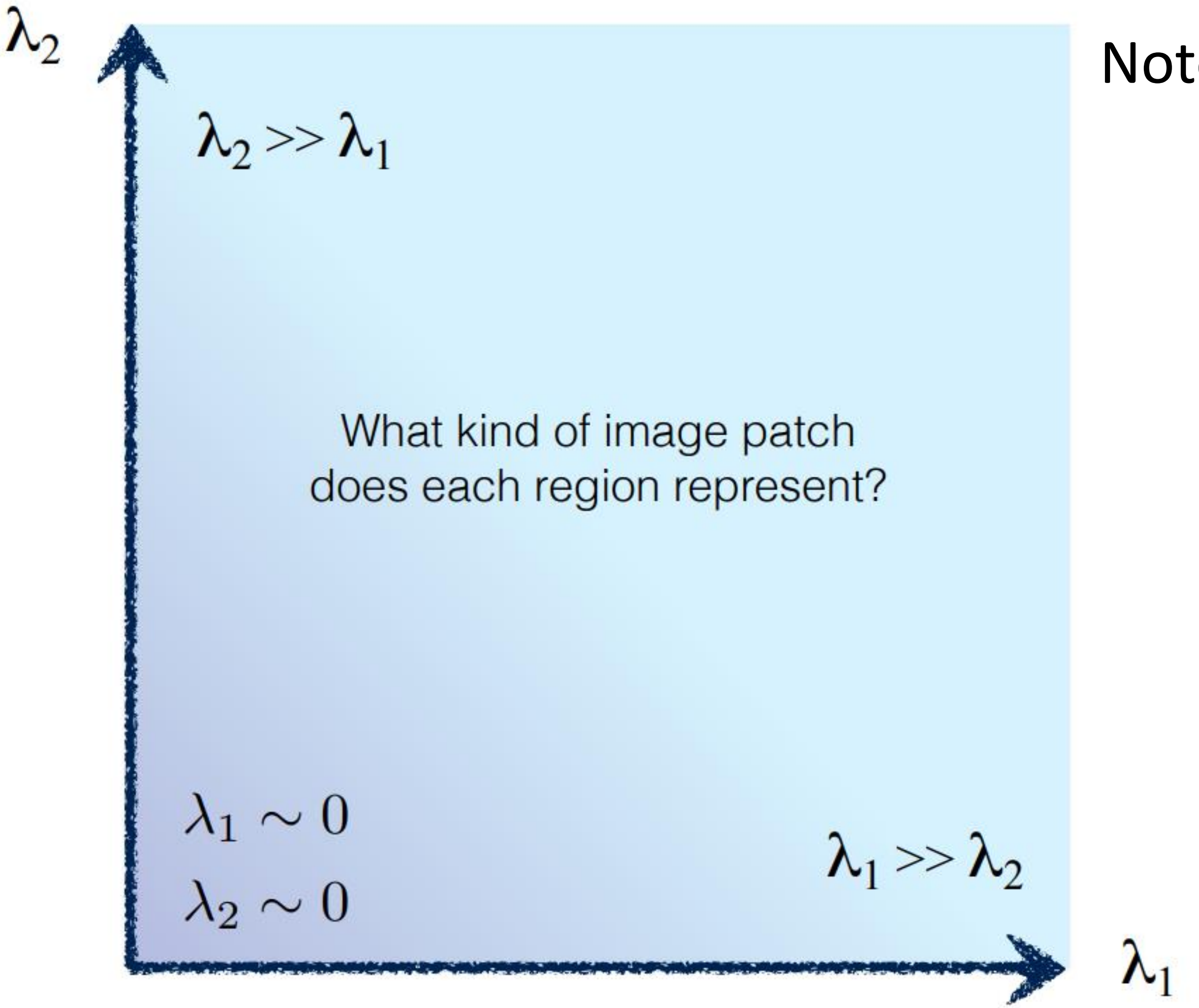
Find Good Features

- If we have this image patch (corner), what is the distribution of all I_x and I_y in the patch
  - Both λ_1 and λ_2 are large



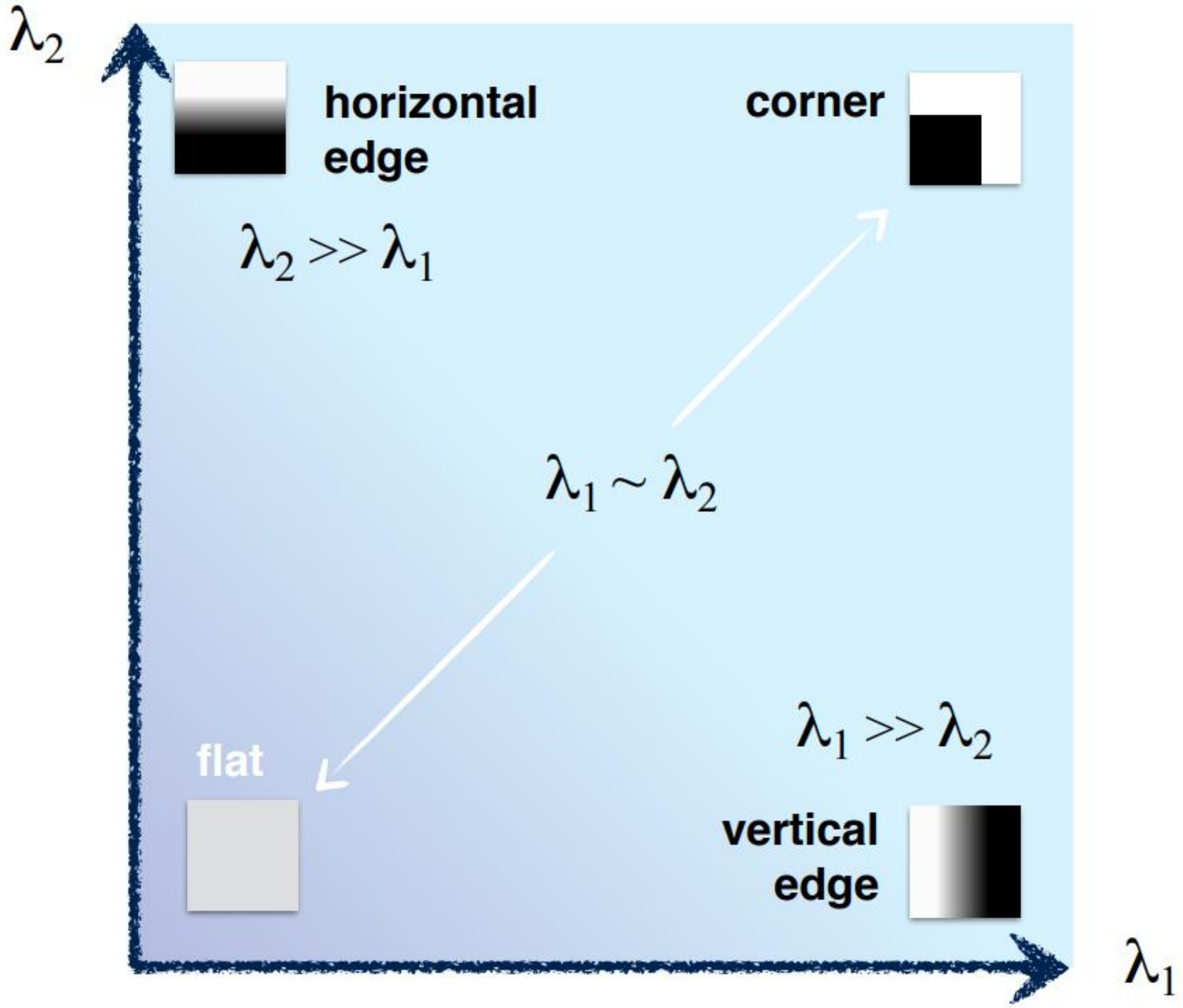


Interpreting Eigenvalues



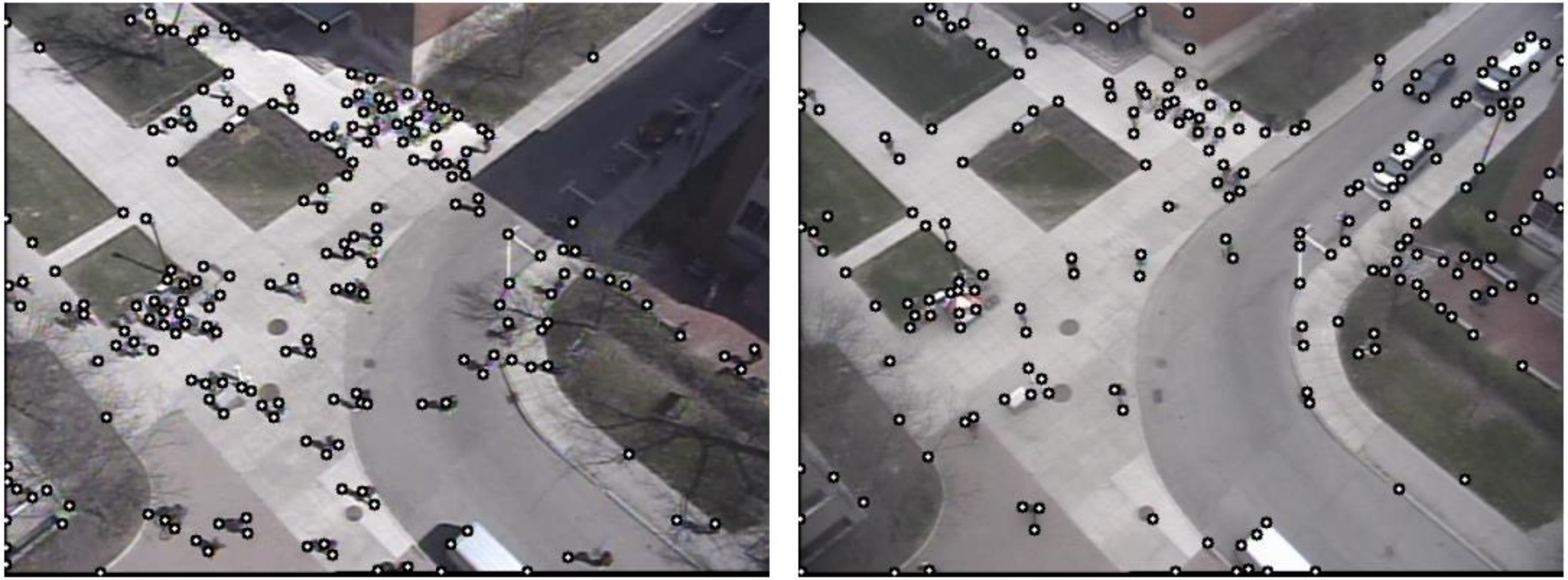
Note: here is the λ_1, λ_2 space

Interpreting Eigenvalues



Good feature for tracking? $\boldsymbol{min(\lambda_1, \lambda_2) > \tau}$

Good Features



Multiple good features
(at least 10 pixels apart between features)

Tracking Good Features

- https://www.youtube.com/watch?v=6B_PNDCWtz4

- https://www.youtube.com/watch?v=pmKtNQphq1E

OpenCV

- KLT example on Moodle

Summary

- KLT = Kanade-Lucas-Tamosi
  - Not track everywhere. Find good features (corners) to track
  - When tracking, find the best transformation (parameters) of a patch from time $t-1$ to t

- An image patch is a good feature?
  - Compute H^{-1} matrix of the patch
  - H^{-1} of a patch is computed from gradient X and Y of all pixels in the patch $H^{-1} = \sum \begin{bmatrix} (I_x)^2 & I_x I_y \\ I_x I_y & (I_y)^2 \end{bmatrix}$
  - If both eigen values (λ_1 , λ_2) are greater than a threshold. This patch is good feature.

- Track a patch between frames
  - Define a transformation $W(x;p)$
  - Find the parameters of the transformation to minimize $\sum_x \left[I\left(W(\boldsymbol{x};\boldsymbol{p})\right) - T(x)\right]^2$
  - Iterative process to find a best p