Lecture:
Interest Point Detection

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

What we will learn today?

- Local invariant features
  - Motivation
  - Requirements, invariances
- Keypoint localization
  - Harris corner detector

Some background reading:
Rick Szeliski, Chapter 4.1.1; David Lowe, IJCV 2004

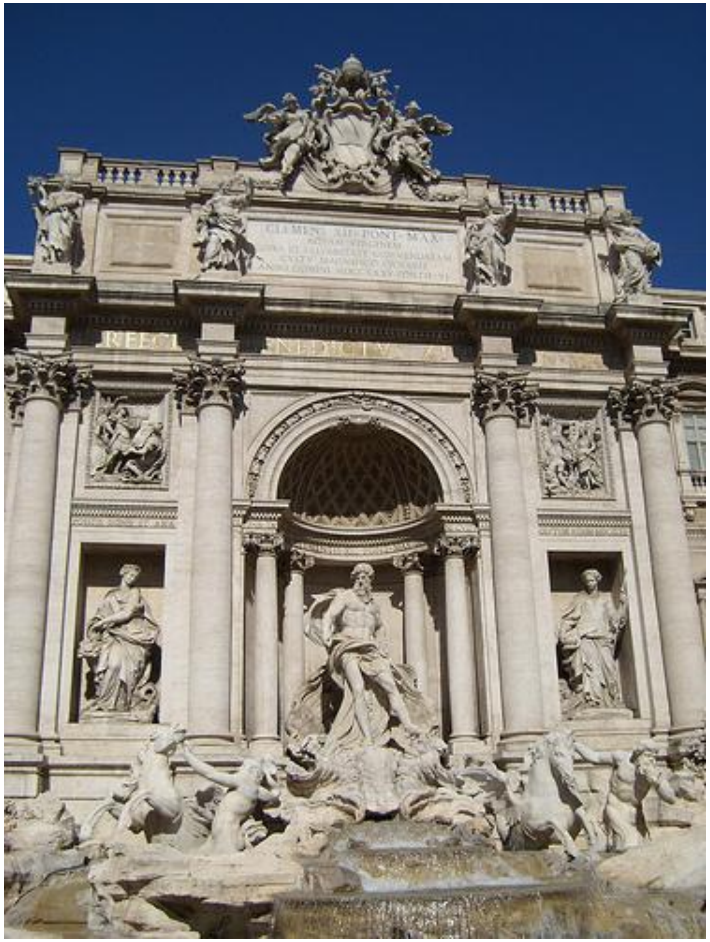
Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Image matching: a challenging problem

Adapted from slides by

Image matching: a challenging problem

by [Diva Sian]



by [swashford]

Slide credit: Steve Seitz

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Harder Case

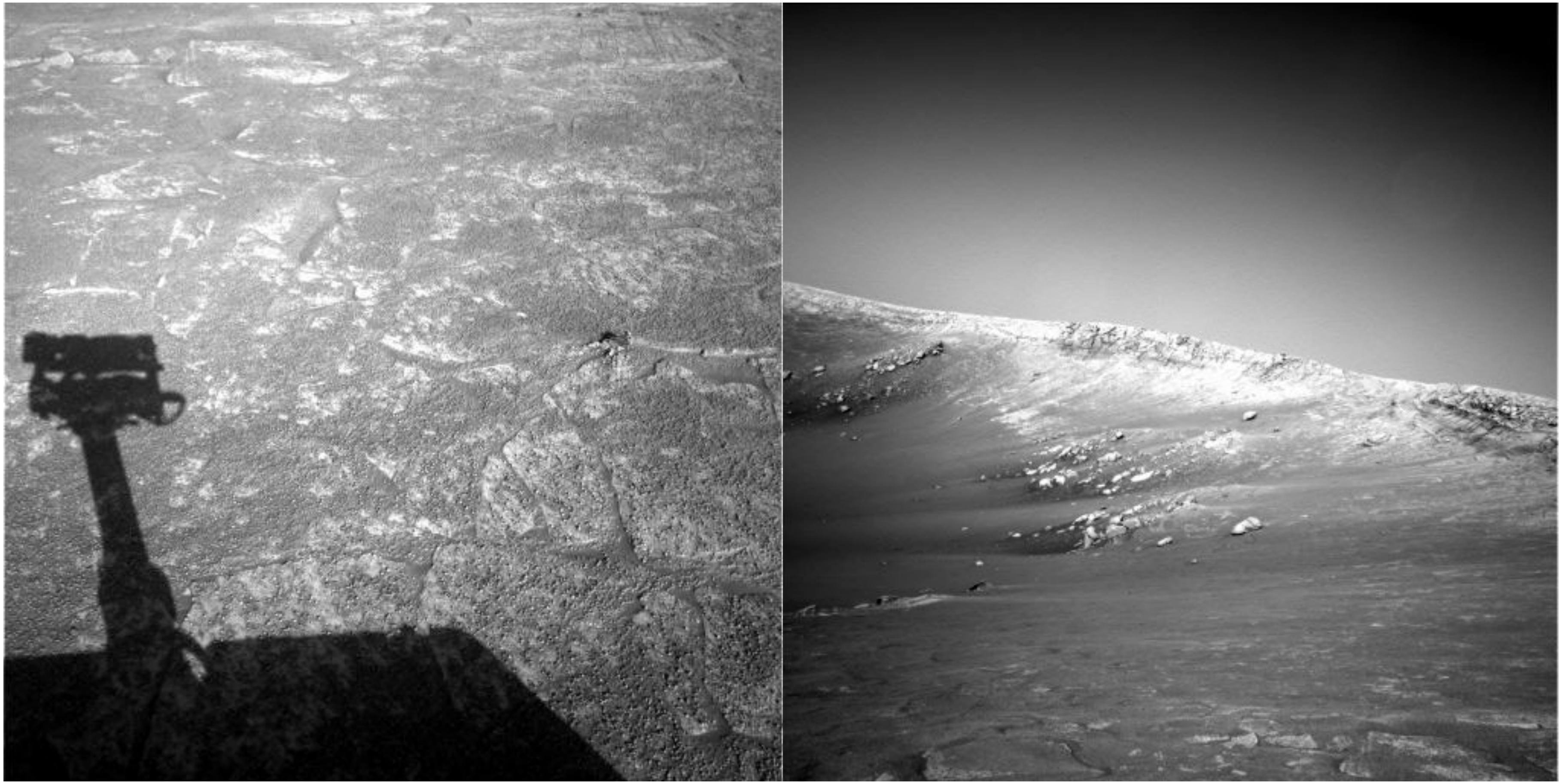
by Diva Sian

by scgbt

Slide credit: Steve Seitz

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Harder Still?

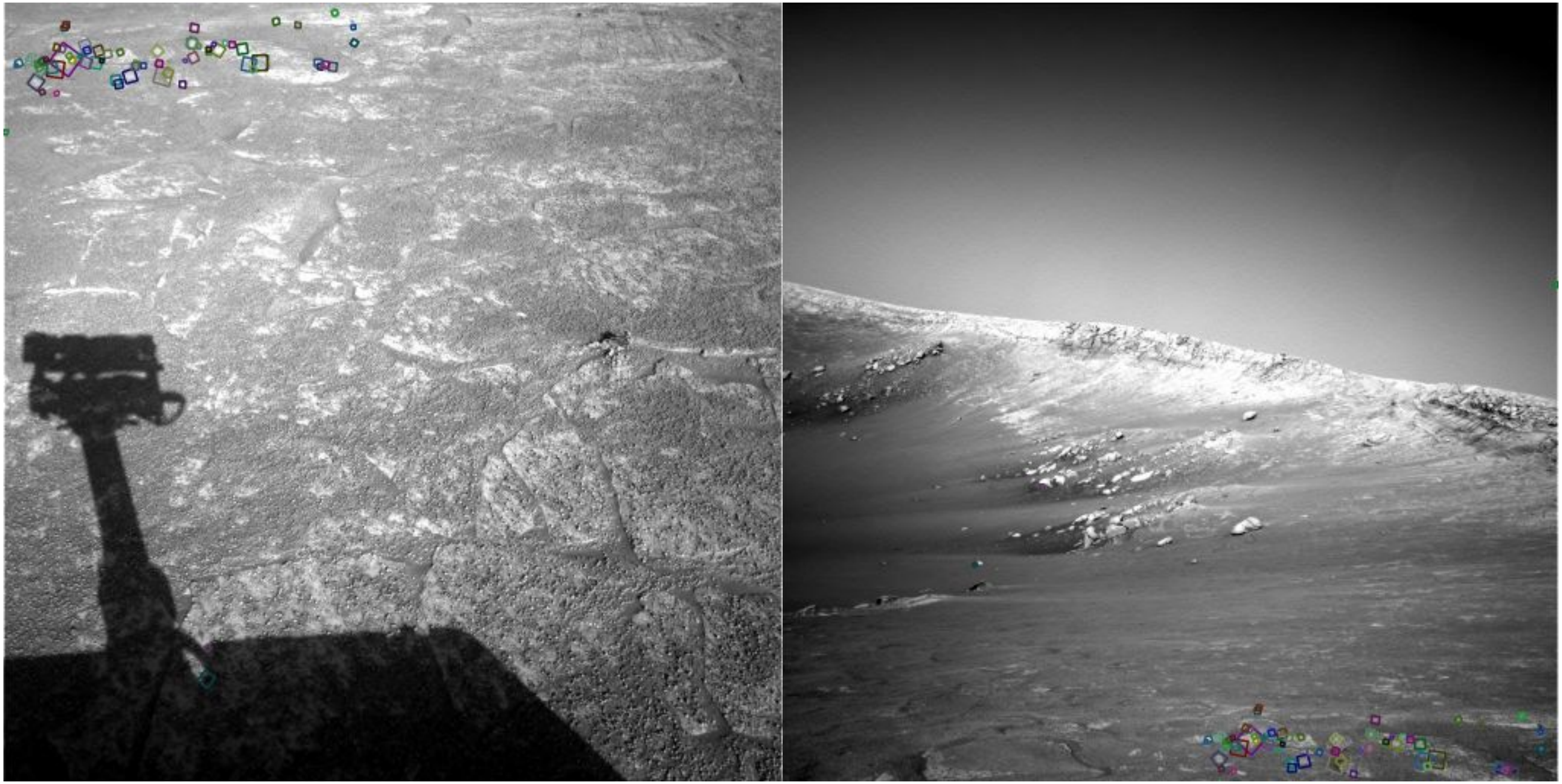


NASA Mars Rover images

Slide credit: Steve Seitz

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Answer Below (Look for tiny colored squares)



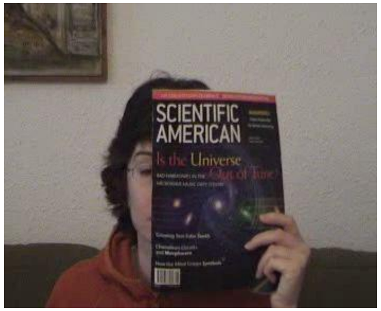
NASA Mars Rover images with SIFT feature matches
(Figure by Noah Snavely)

Slide credit: Steve Seitz

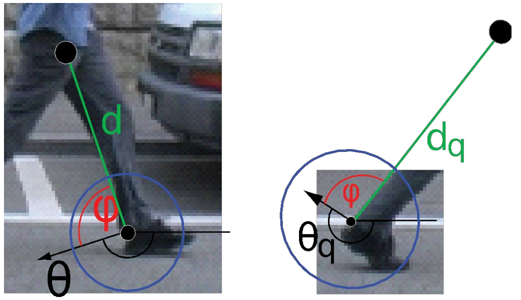
Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Motivation for using local features

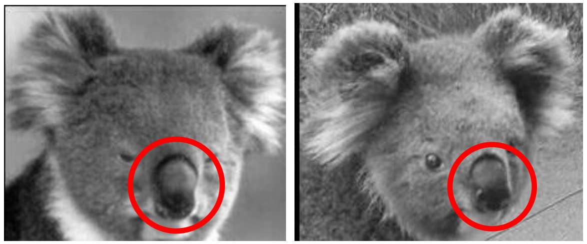
- Global representations have major limitations
- Instead, describe and match only local regions
- Increased robustness to
  - Occlusions



  - Articulation

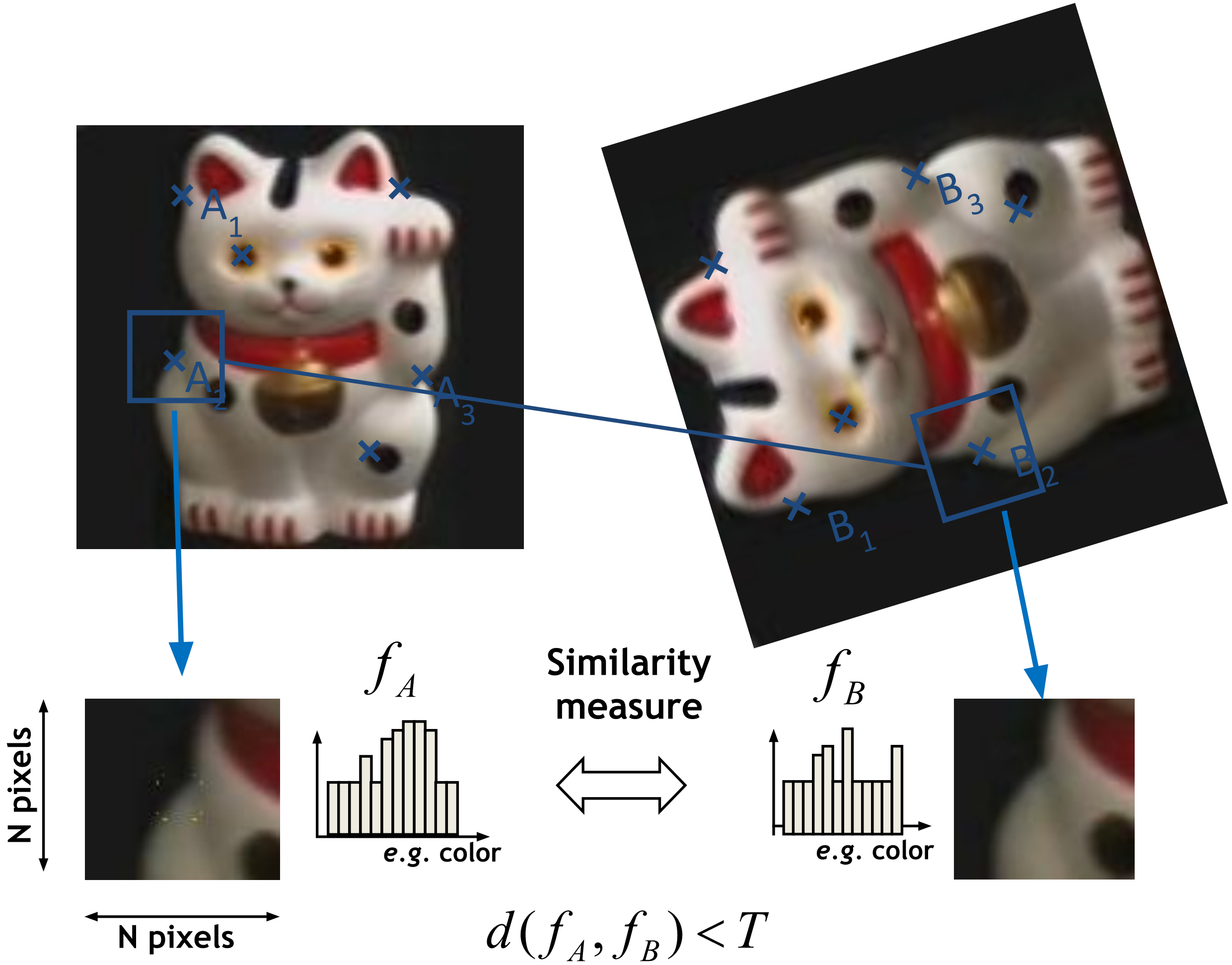


  - Intra-category variations



Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

General Approach



1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

Slide credit: Bastian Leibe

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Common Requirements

- Problem 1:
  - Detect the same point *independently* in both images

No chance to match!

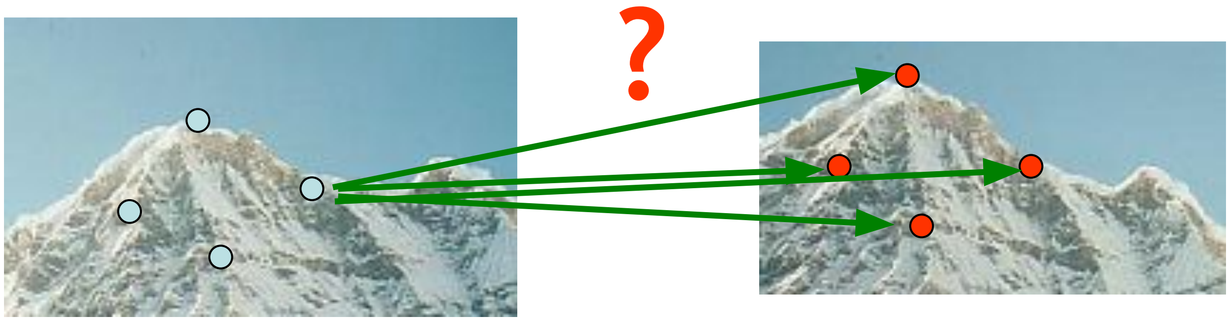
We need a repeatable detector!

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Slide credit: Darya Frolova, Denis Simakov

Common Requirements

- Problem 1:
  - Detect the same point *independently* in both images
- Problem 2:
  - For each point correctly recognize the corresponding one



We need a reliable and distinctive descriptor!

Slide credit: Darya Frolova, Denis Simakov

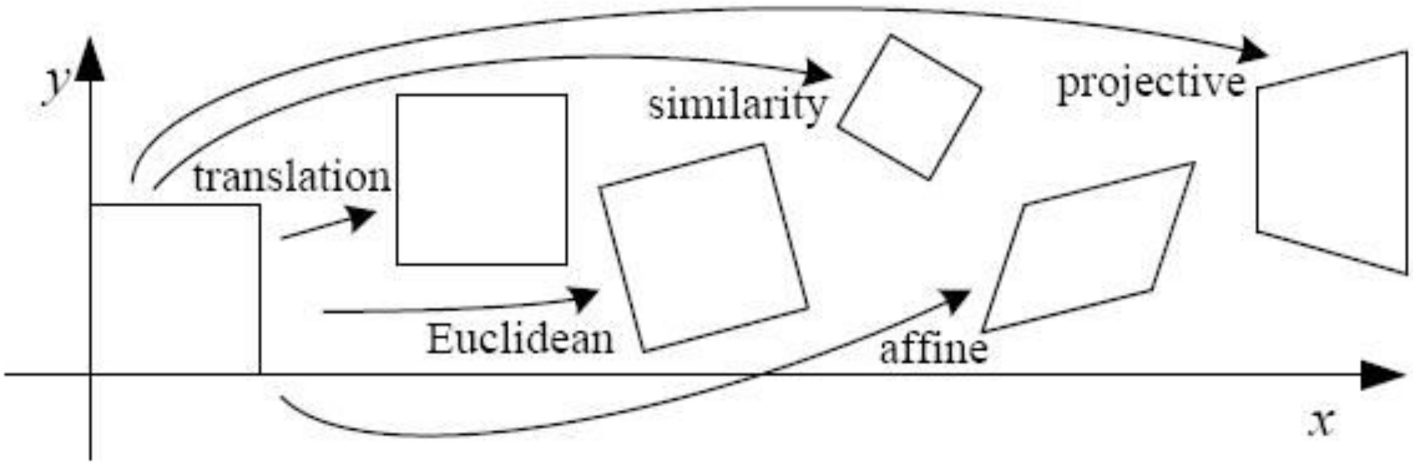
Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Invariance: Geometric Transformations

Slide credit: Steve Seitz

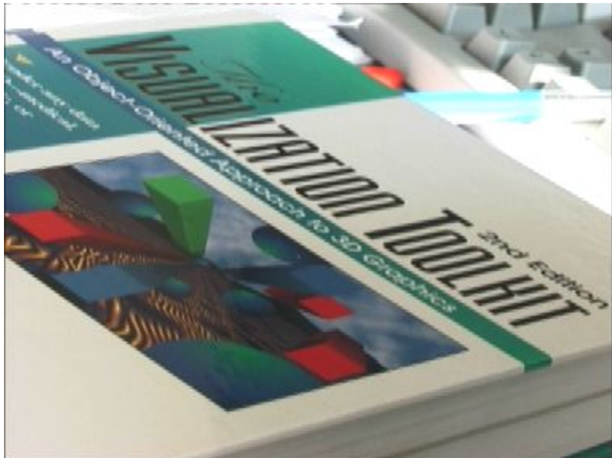
Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

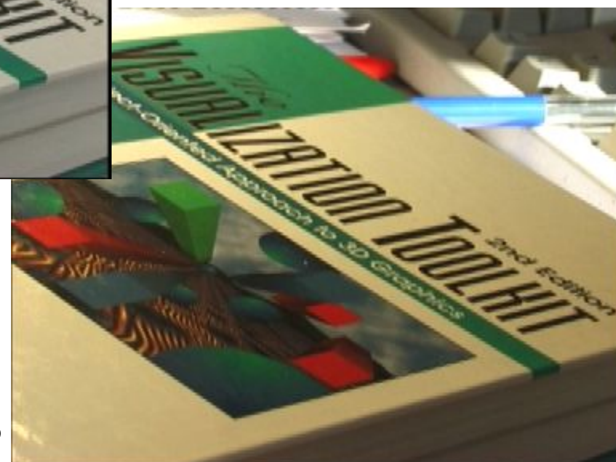
Levels of Geometric Invariance

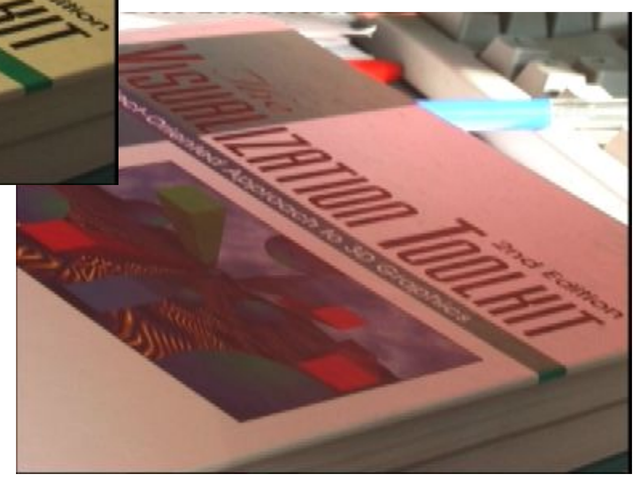


Slide credit: Bastian Leibe

Invariance: Photometric Transformations







- Often modeled as transformation:
  - Scaling + Offset

Slide credit: Tinne Tuytelaars

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Requirements

- Region extraction needs to be repeatable and accurate
  - Invariant to translation, rotation, scale changes
  - Robust or covariant to out-of-plane (≈affine) transformations
  - Robust to lighting variations, noise, blur, quantization
- Locality: Features are local, therefore robust to occlusion and clutter.
- Quantity: We need a sufficient number of regions to cover the object.
- Distinctivenes : The regions should contain "interesting" structure.
- Efficiency: Close to real-time performance.

Slide credit: Bastian Leibe

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Many Existing Detectors Available

- Hessian & Harris [Beaudet '78], [Harris '88]
- Laplacian, DoG [Lindeberg '98], [Lowe '99]
- Harris-/Hessian-Laplace [Mikolajczyk & Schmid '01]
- Harris-/Hessian-Affine [Mikolajczyk & Schmid '04]
- EBR and IBR [Tuytelaars & Van Gool '04]
- MSER [Matas '02]
- Salient Regions [Kadir & Brady '01]
- Others…

- *Those detectors have become a basic building block for many pre-deep learning approaches in Computer Vision.*
- *They are still very relevant & powerful tools*

Slide credit: Bastian Leibe

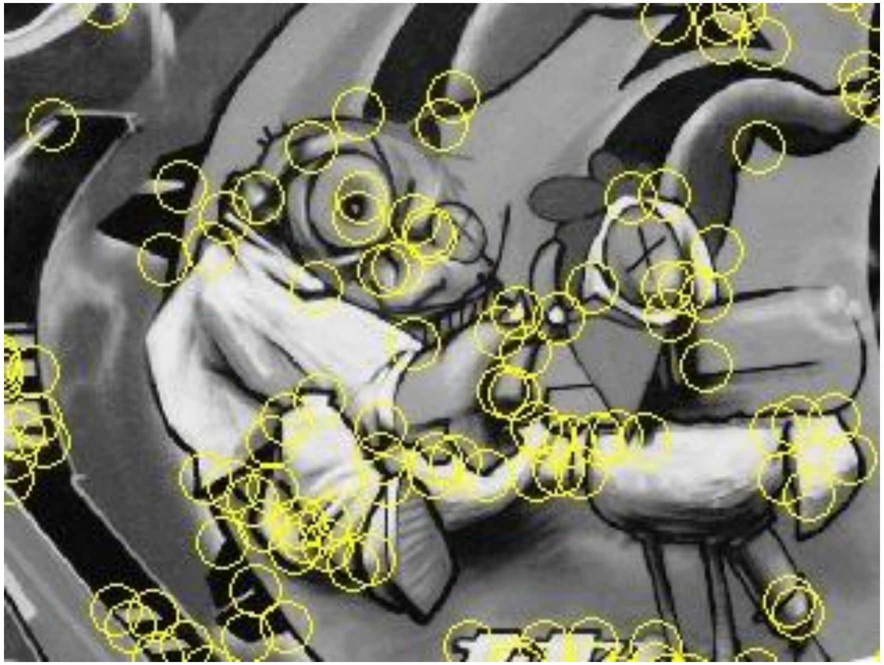
Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

What we will learn today?

- Local invariant features
  - Motivation
  - Requirements, invariances
- Keypoint localization
  - Harris corner detector

Some background reading:
Rick Szeliski, Chapter 4.1.1; David Lowe, IJCV 2004

Keypoint Localization



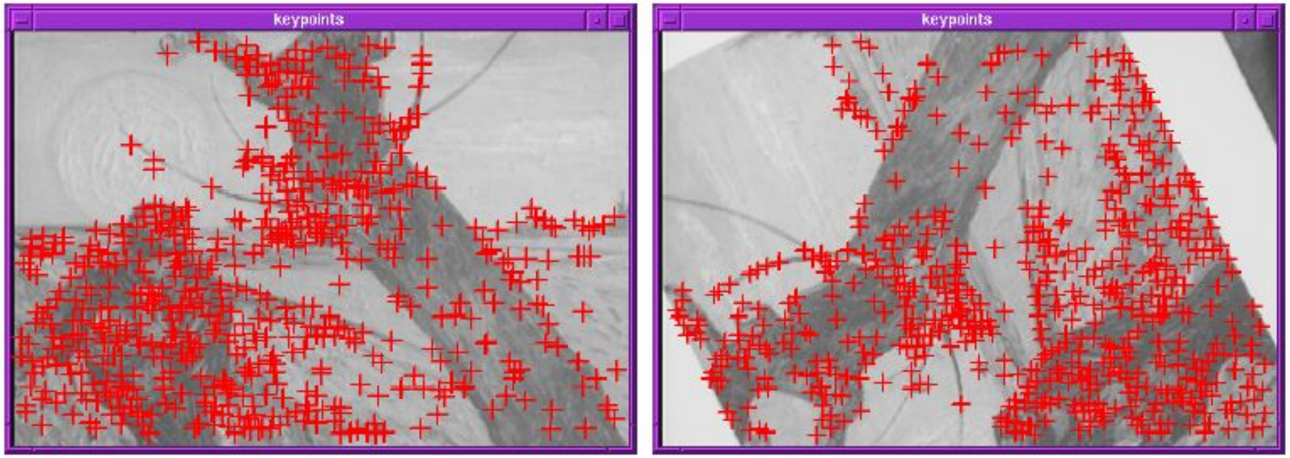
- Goals:
  - Repeatable detection
  - Precise localization
  - Interesting content

⇒ *Look for two-dimensional signal changes*

Slide credit: Bastian Leibe

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Finding Corners



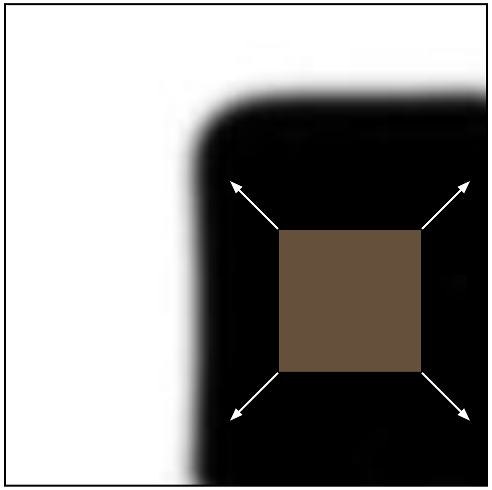
- Key property:
  - In the region around a corner, image gradient has two or more dominant directions
- Corners are *repeatable* and *distinctive*

C.Harris and M.Stephens. ["A Combined Corner and Edge Detector."]
Proceedings of the 4th Alvey Vision Conference, 1988.

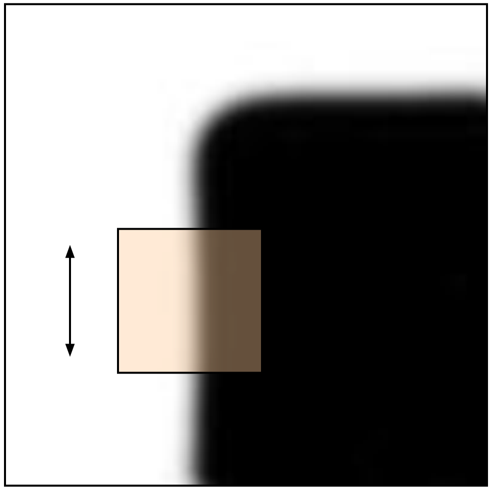
Slide credit: Svetlana Lazebnik

Corners as Distinctive Interest Points

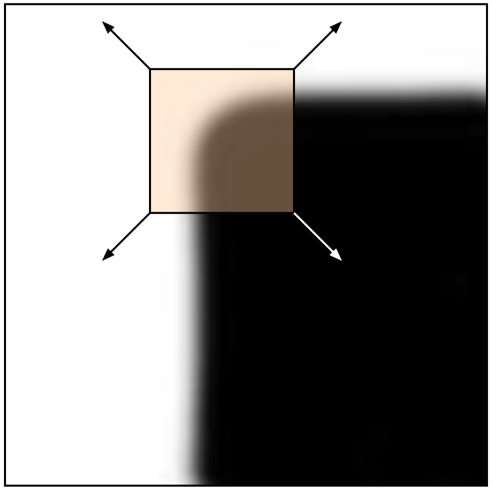
- Design criteria
  - We should easily recognize the point by looking through a small window (***locality***)
  - Shifting the window in *any direction* should give *a large change* in intensity (***good localization***)



"flat" region:
no change in all directions



"edge":
no change along the edge direction

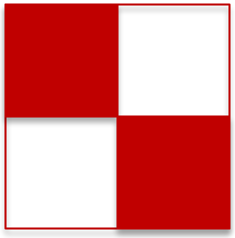


"corner":
significant change in all directions

Slide credit: Alyosha Efros

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Corners versus edges



$\sum I_x^2$ → Large

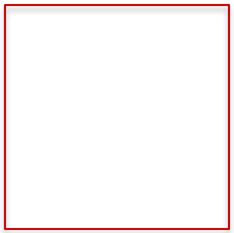
$\sum I_y^2$ → Large

Corner

$\sum I_x^2$ → Small

$\sum I_y^2$ → Large

Edge



$\sum I_x^2$ → Small

$\sum I_y^2$ → Small

Nothing

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Corners versus edges

$$\sum I_x^2 \longrightarrow ??$$

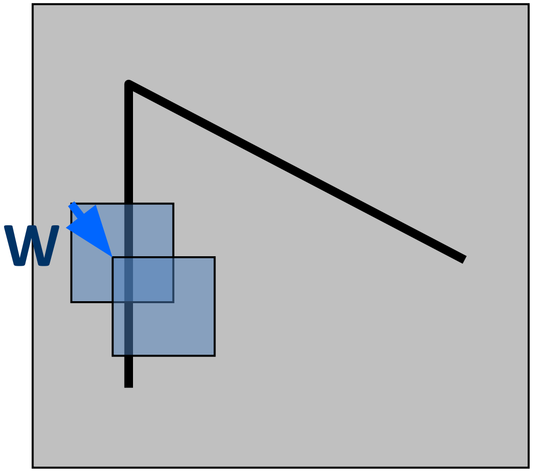
$$\sum I_y^2 \longrightarrow ??$$

Corner

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Harris Detector Formulation

- Consider shifting the window W by (u,v):
  - How do the pixels in W change?
  - Auto-correlation function measures the self similarity of a signal and is related to the sum-of-squared difference.
  - Compare each pixel before and after the shift by summing up the squared differences (SSD).
  - $$E(u,v) = \sum_{(x,y) \in W} [I(x+u, y+v) - I(x,y)]^2$$



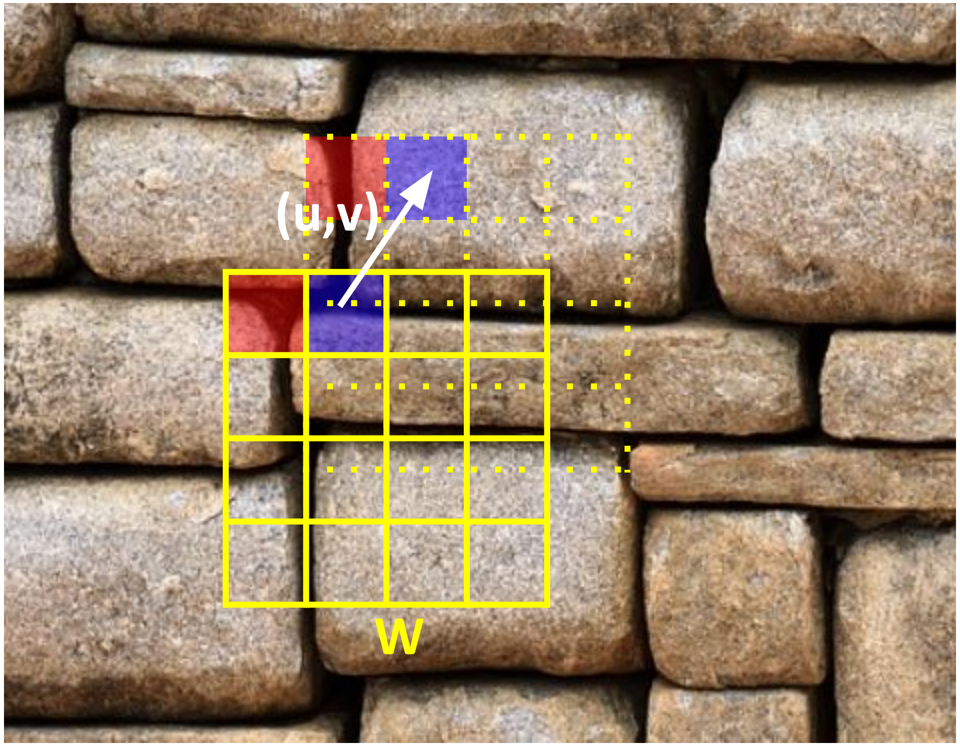
SSD for (u,v) direction of a single region

Slide partially from Selim Aksoy.

Harris Detector Formulation

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

- <u>Specific to window W</u>
- W = set of pixel coordinates of the box
- Measure the total difference when the window W moved by (u,v) pixels
    - Sum of squared L2 distances between each pixel and its (u,v) translated version.



Wall image source:
https://www.maxpixel.net/Background-Backdrop-Wall-Stone-Wall-Old-Closeup-1530682

Harris - weighted window

- Change of intensity for the shift [u,v]:

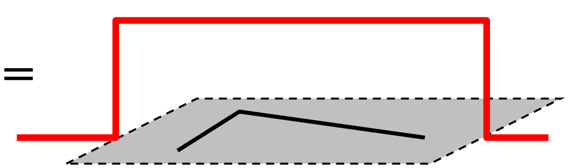
$$E(u,v)=\sum_{x,y} w(x,y)\left[I(x+u,y+v)-I(x,y)\right]^2$$

Window function

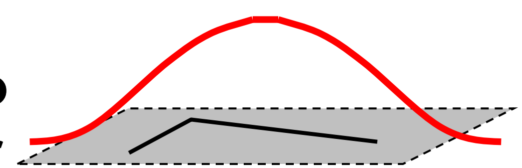
Shifted intensity

Intensity

Window function $w(x,y) =$



or



1 in window, 0 outside

Gaussian

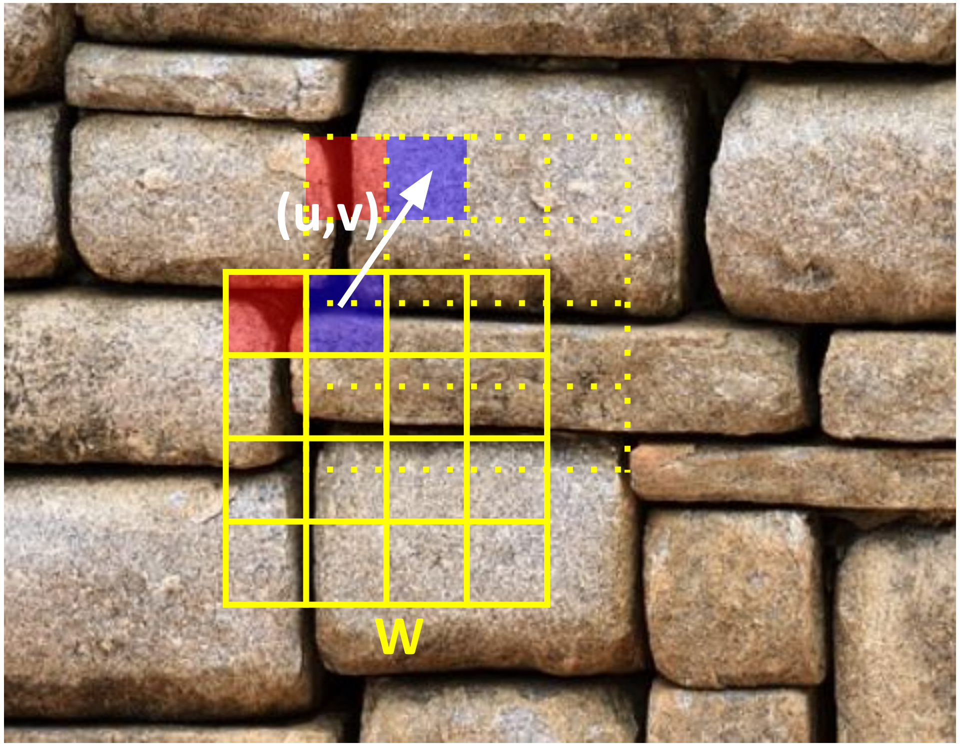
Slide credit: Rick Szeliski

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Harris - compute cost blues

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

- Think of the computational cost over the whole image. (Observe: *weighted windowing* is not the root of the problem)
- How can you speed up the process?



Wall image source:
https://www.maxpixel.net/Background-Backdrop-Wall-Stone-Wall-Old-Closeup-1530682

Harris - Approximation

- Taylor Series expansion of I:

$$I(x+u, y+v) = I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \text{higher order terms}$$

- If the motion (u,v) is assumed to be small, then first order approximation is good.

$$\begin{aligned} I(x+u, y+v) &\approx I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v \\ &\approx I(x,y) + [I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix} \end{aligned}$$

shorthand: $I_x = \frac{\partial I}{\partial x}$

- Plugging this into the formula on the previous slide...

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Slide partially from Selim Aksoy

Harris Detector Formulation

- Sum-of-squared differences error E(u,v):

$$
\begin{aligned}
E(u,v) &= \sum_{(x,y)\in W} \left[\boxed{I(x+u,y+v)} - I(x,y)\right]^2 \\
&\approx \sum_{(x,y)\in W} \left[\boxed{\cancel{I(x,y)} + [I_x \;\; I_y]\begin{bmatrix} u \\ v \end{bmatrix}} - \cancel{I(x,y)}\right]^2 \\
&\approx \sum_{(x,y)\in W} \left[[I_x \;\; I_y]\begin{bmatrix} u \\ v \end{bmatrix}\right]^2
\end{aligned}
$$

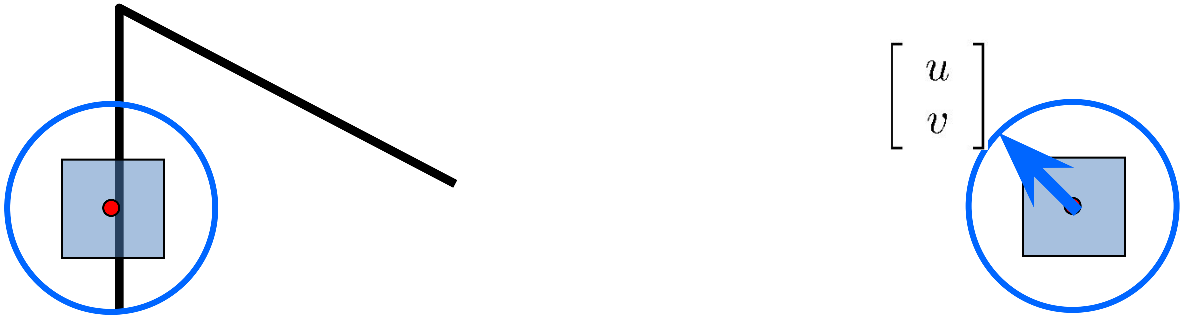
Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Slide partially from Selim Aksoy

Harris Detector Formulation

- This can be rewritten (*self-study homework*):

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



- For the example above:
  - You can move the center of the green window to anywhere on the blue unit circle.
  - Which directions will result in the largest and smallest E values?
- Faster? A bit better but not much yet -- we'll revisit this in a few slides.

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Slide partially from Selim Aksoy

Moving the summation into H

- This can be rewritten:

$$E(u,v) = \sum_{(x,y)\in W} [u \ v] \underbrace{\begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}}_{H} \begin{bmatrix} u \\ v \end{bmatrix}$$

[sum over all (x,y)]

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Moving the summation into H

- This measure of change can be further simplified as:

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

where H is a 2×2 matrix computed from image derivatives:

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

Gradient with respect to x, times gradient with respect to y

Sum over image region – the area we are checking for corner

Faster? Still not there yet -- we'll revisit this in a few slides.

$$= \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ \ I_y]$$

Slide credit: Rick Szeliski

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Harris Detector Formulation

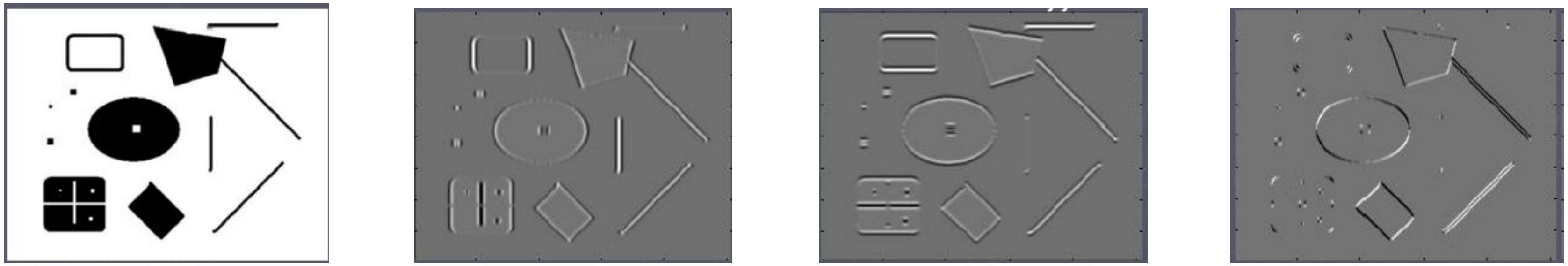


Image I $\quad I_x \quad I_y \quad I_x I_y$

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

Slide credit: Rick Szeliski

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Harris Detector Formulation

- We want to find (u,v) such that E(u,v) is maximized or minimized:

$$\mathrm{E}(u,v) = \begin{bmatrix} u \\ v \end{bmatrix}^T \mathbf{H} \begin{bmatrix} u \\ v \end{bmatrix}$$

– By definition, we can find these directions by looking at the eigenvectors of **H**.

– First eigenvector of **H** is an unit vector that maximizes E(u,v).

– Second eigenvector of **H** is an unit vector that minimizes E(u,v)

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Slide partially from Selim Aksoy

Quick eigenvector/eigenvalue review

- Relevant theorem:

THEOREM 2.5 If $\mathcal{A}$ and $\mathcal{B}$ are symmetric and $\mathcal{B} > 0$, then the maximum of $x^\top \mathcal{A} x$ under the constraints $x^\top \mathcal{B} x = 1$ is given by the largest eigenvalue of $\mathcal{B}^{-1}\mathcal{A}$. More generally,

$$\max_{\{x: x^\top \mathcal{B} x = 1\}} x^\top \mathcal{A} x = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p = \min_{\{x: x^\top \mathcal{B} x = 1\}} x^\top \mathcal{A} x,$$

where $\lambda_1, \ldots, \lambda_p$ denote the eigenvalues of $\mathcal{B}^{-1}\mathcal{A}$. The vector which maximizes (minimizes) $x^\top \mathcal{A} x$ under the constraint $x^\top \mathcal{B} x = 1$ is the eigenvector of $\mathcal{B}^{-1}\mathcal{A}$ which corresponds to the largest (smallest) eigenvalue of $\mathcal{B}^{-1}\mathcal{A}$.

http://fedc.wiwi.hu-berlin.de/xplore/tutorials/mvahtmlnode16.html

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Quick eigenvector/eigenvalue review

- The eigenvectors of a matrix **A** are the vectors **x** that satisfy:

$$Ax = \lambda x$$

- The scalar λ is the eigenvalue corresponding to **x**
  - The eigenvalues are found by solving:

$$det(A - \lambda I) = 0$$

  - In our case, **A** = **H** is a 2x2 matrix, so we have

$$det \begin{bmatrix} h_{11} - \lambda & h_{12} \\ h_{21} & h_{22} - \lambda \end{bmatrix} = 0$$

  - The solution:

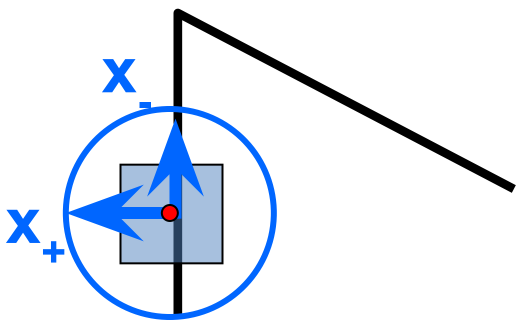
$$\lambda_{\pm} = \frac{1}{2}\left[(h_{11} + h_{22}) \pm \sqrt{4h_{12}h_{21} + (h_{11} - h_{22})^2}\right]$$

- Once you know λ, you find **x** by solving

$$\begin{bmatrix} h_{11} - \lambda & h_{12} \\ h_{21} & h_{22} - \lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0$$

Local features: detection

$$E(u,v) = \begin{bmatrix} u \\ v \end{bmatrix}^T \mathbf{H} \begin{bmatrix} u \\ v \end{bmatrix}$$



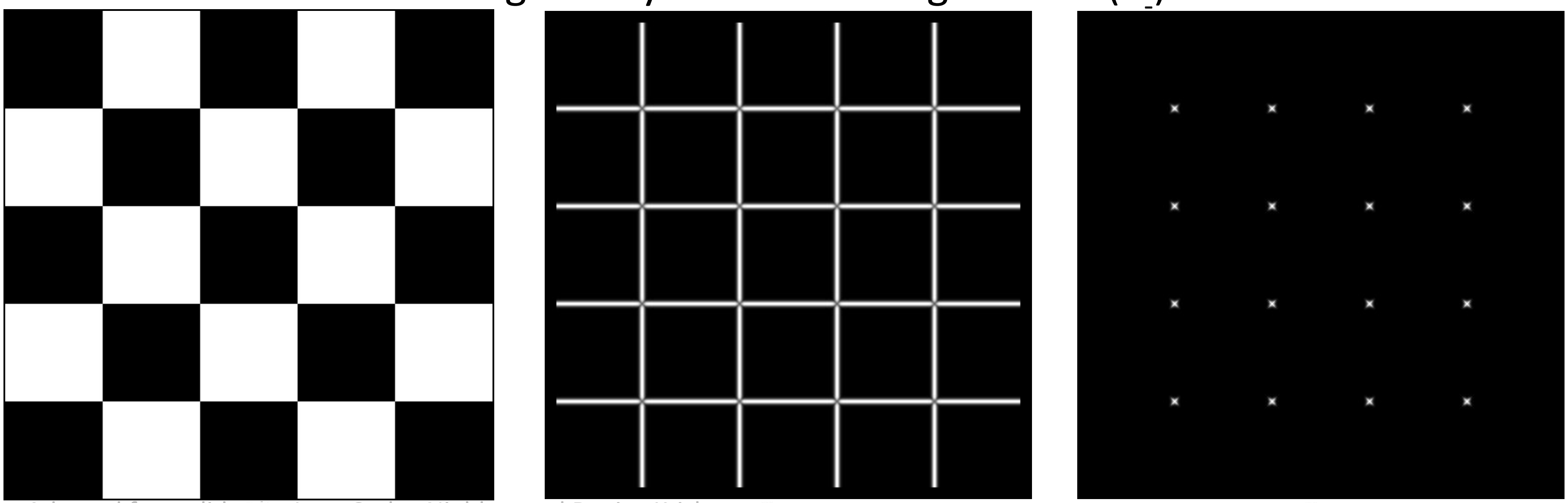
- Eigenvalues and eigenvectors of **H**:
  - Define shifts with the smallest and largest change (E value).
  - x_+ = direction of largest increase in E.
  - λ_+ = amount of increase in direction x_+.
  - x_- = direction of smallest increase in E.
  - λ_- = amount of increase in direction x_-.

$$\begin{aligned} Hx_+ &= \lambda_+ x_+ \\ Hx_- &= \lambda_- x_- \end{aligned}$$

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Local features: detection

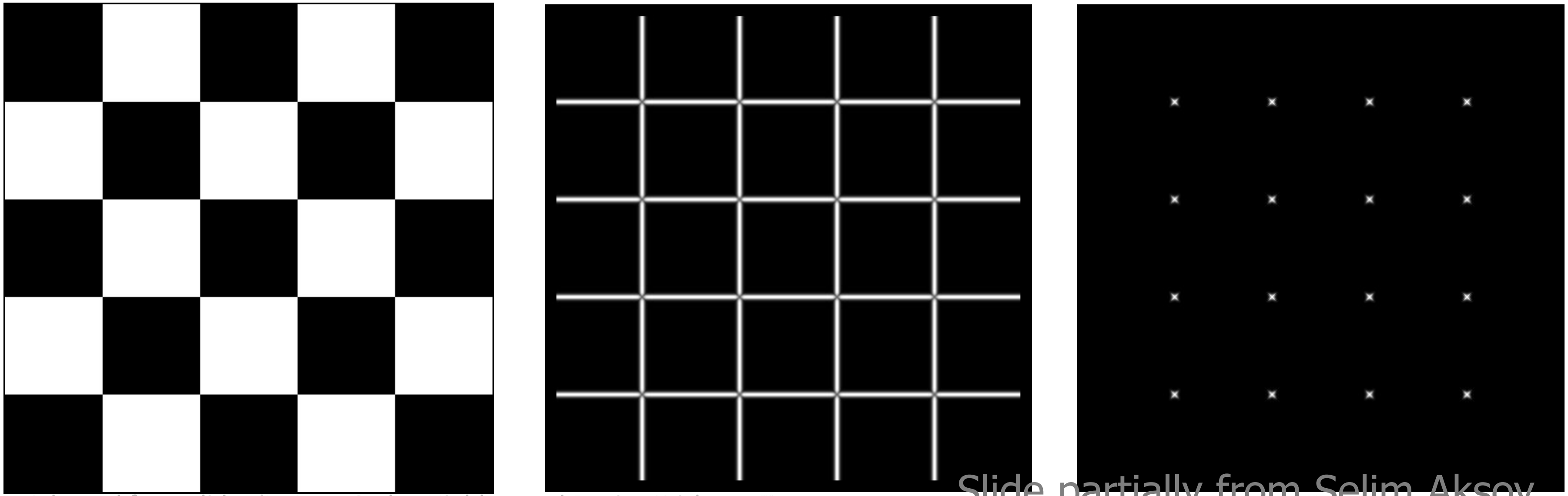
- How are λ_+, x_+, λ_-, and x_- relevant for feature detection?
  - What's our feature scoring function?
- Want E(u,v) to be large for small shifts in all directions.
  - The minimum of E(u,v) should be large, over all unit vectors [u v].
  - This minimum is given by the smaller eigenvalue (λ_-) of **H**.



I λ_+ λ_-

Local features: detection

- Here's what you do:
  - Compute the gradient at each point in the image.
  - Create the **H** matrix from the entries in the gradient, at each point.
  - Compute the eigenvalues.
  - Find points with large response (λ_- > threshold).
  - Choose those points where λ_- is a local maximum as features.



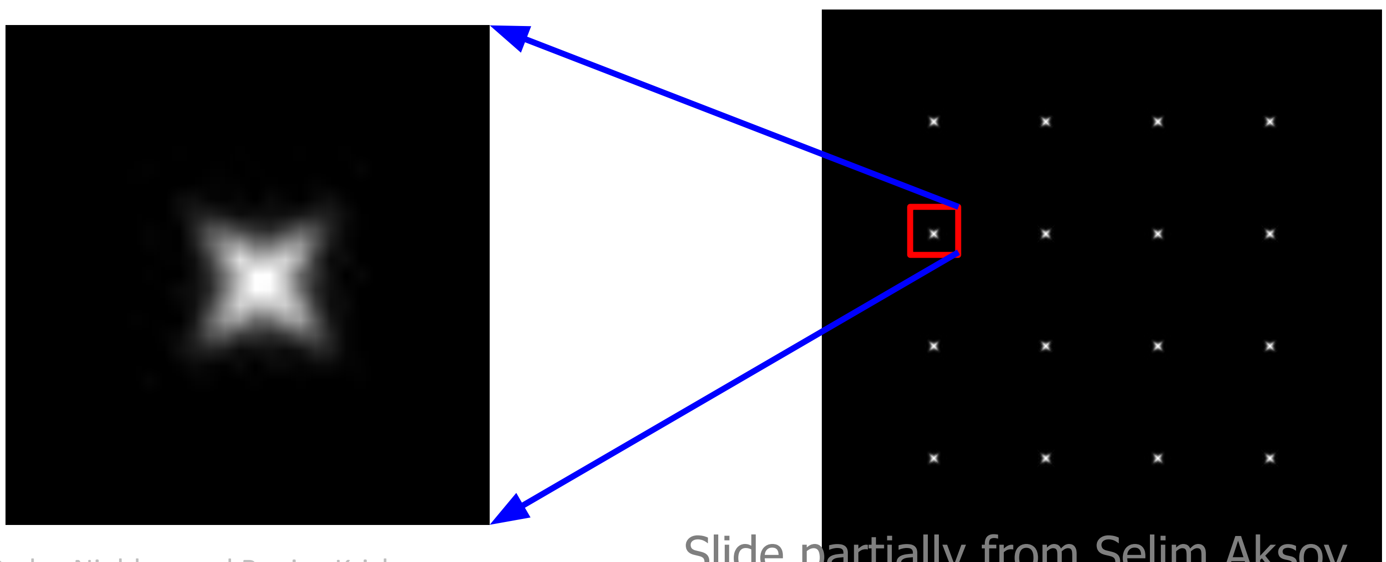
I λ_+ λ_-

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Slide partially from Selim Aksoy

Local features: detection

- Here’s what you do:
  - Compute the gradient at each point in the image.
  - Create the **H** matrix from the entries in the gradient.
  - Compute the eigenvalues.
  - Find points with large response (λ_- > threshold).
  - Choose those points where λ_- is a local maximum as features.



Slide partially from Selim Aksoy

λ_-

Harris detector

- To measure the corner strength:

$$R = \det(H) - k(\mathrm{trace}(H))^2$$

  where

$$\mathrm{trace}(H) = \lambda_1 + \lambda_2$$
$$\det(H) = \lambda_1 \times \lambda_2$$

  (λ_1 and λ_2 are the eigenvalues of H).

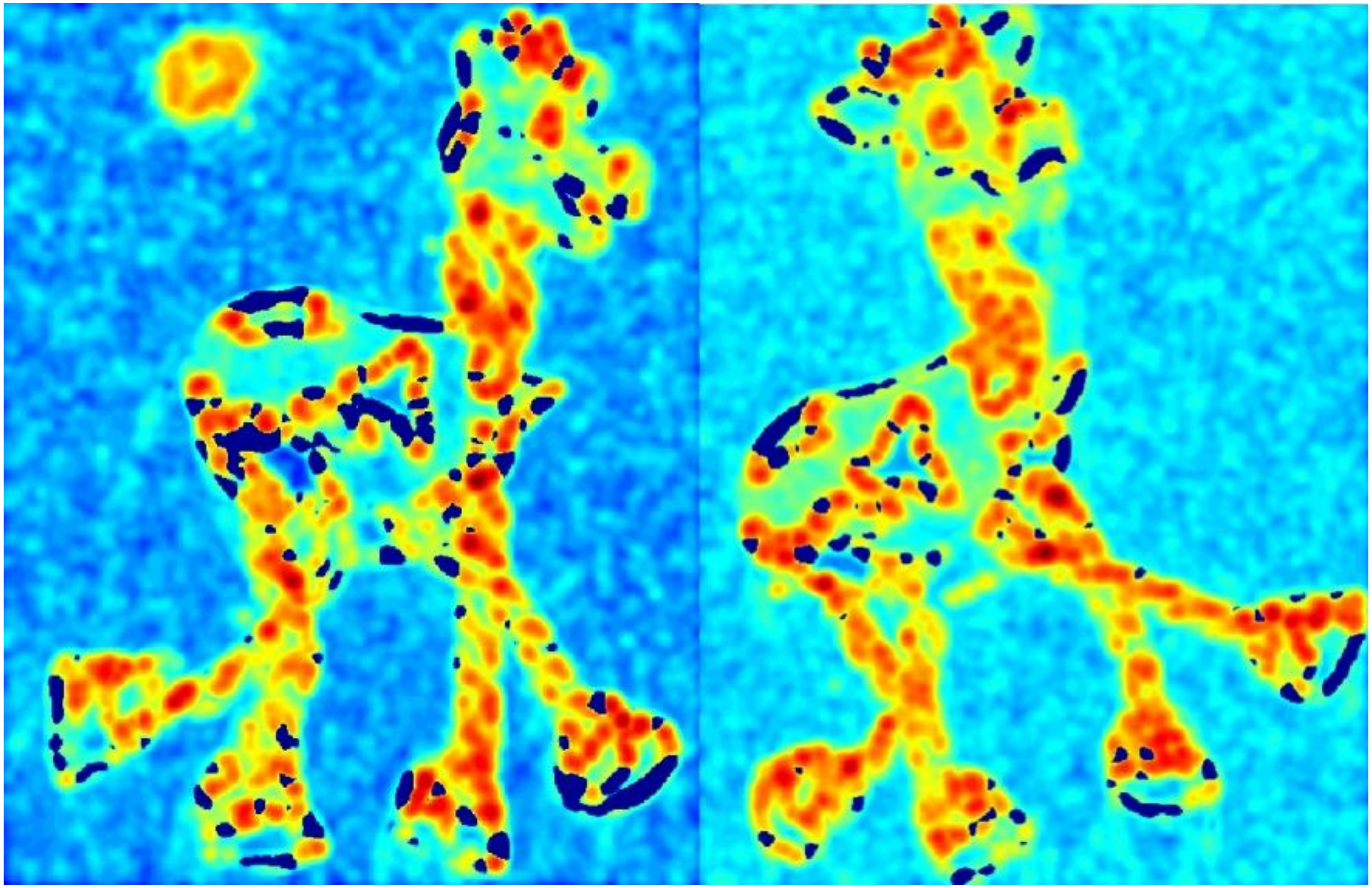
- R is positive for corners, negative in edge regions, and small in flat regions.
- Very similar to λ_- but **less expensive** (no square root).
- Also called the "Harris Corner Detector" or "Harris Operator".
- Lots of other detectors, this is one of the most popular.

Slide partially from Selim Aksoy

Harris detector example

Slide partially from Selim Aksoy

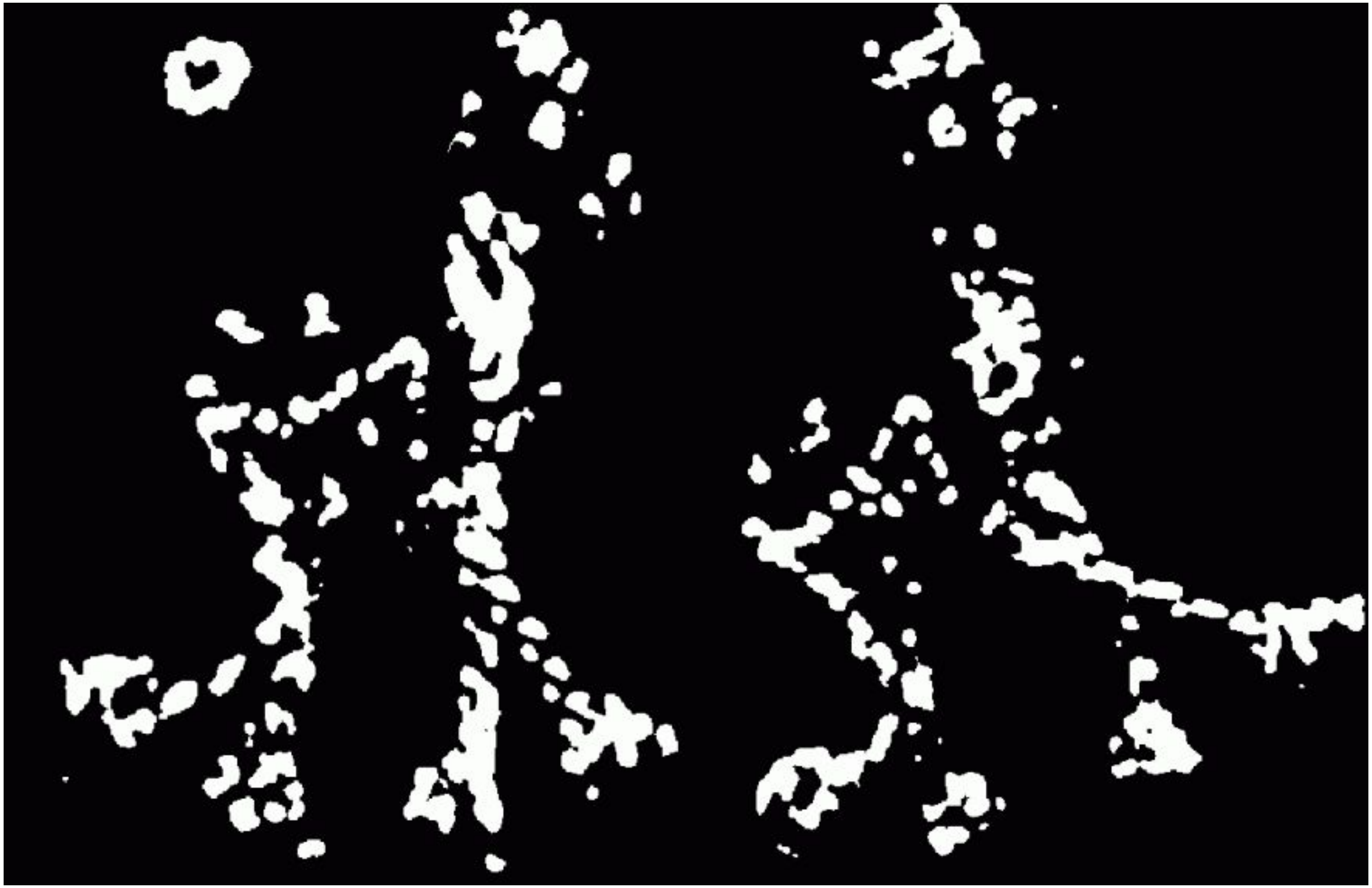
Harris detector example



R values (red high, blue low)

Slide partially from Selim Aksoy

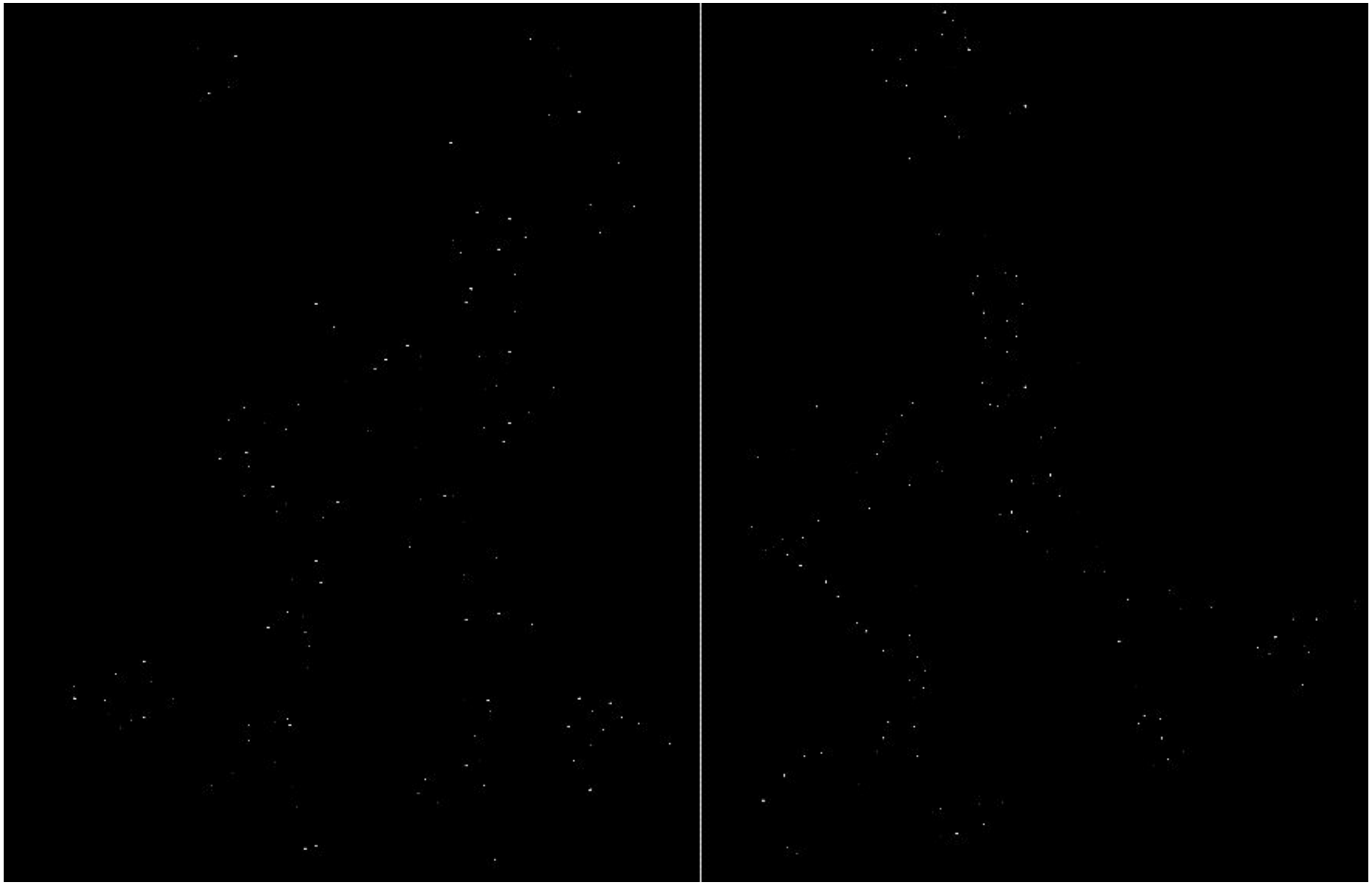
Harris detector example



Threshold (R > value)

Slide partially from Selim Aksoy

Harris detector example



Local maxima of R

Slide partially from Selim Aksoy

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Harris detector example

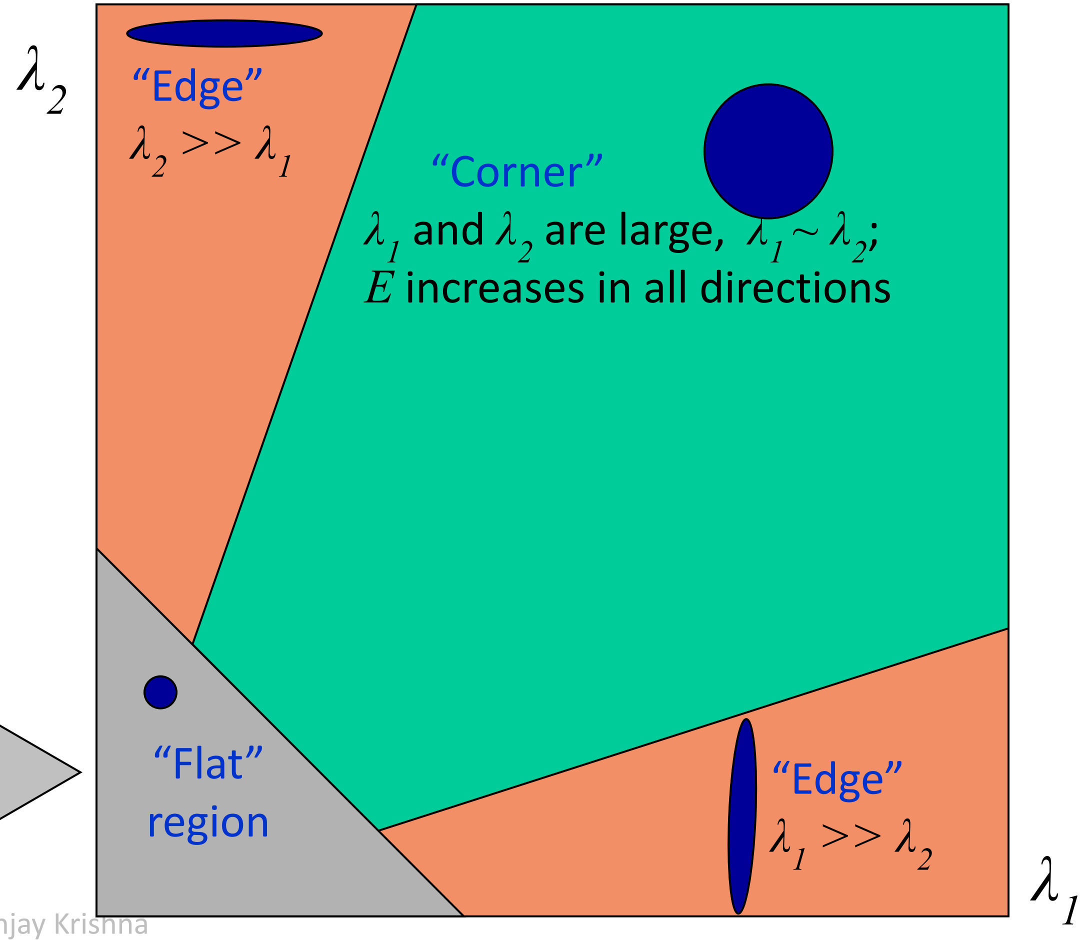
Harris features (red)

Slide partially from Selim Aksoy

Interpreting the Eigenvalues

- Classification of image points using eigenvalues of M:

λ_1 and λ_2 are small;
E is almost constant in all directions

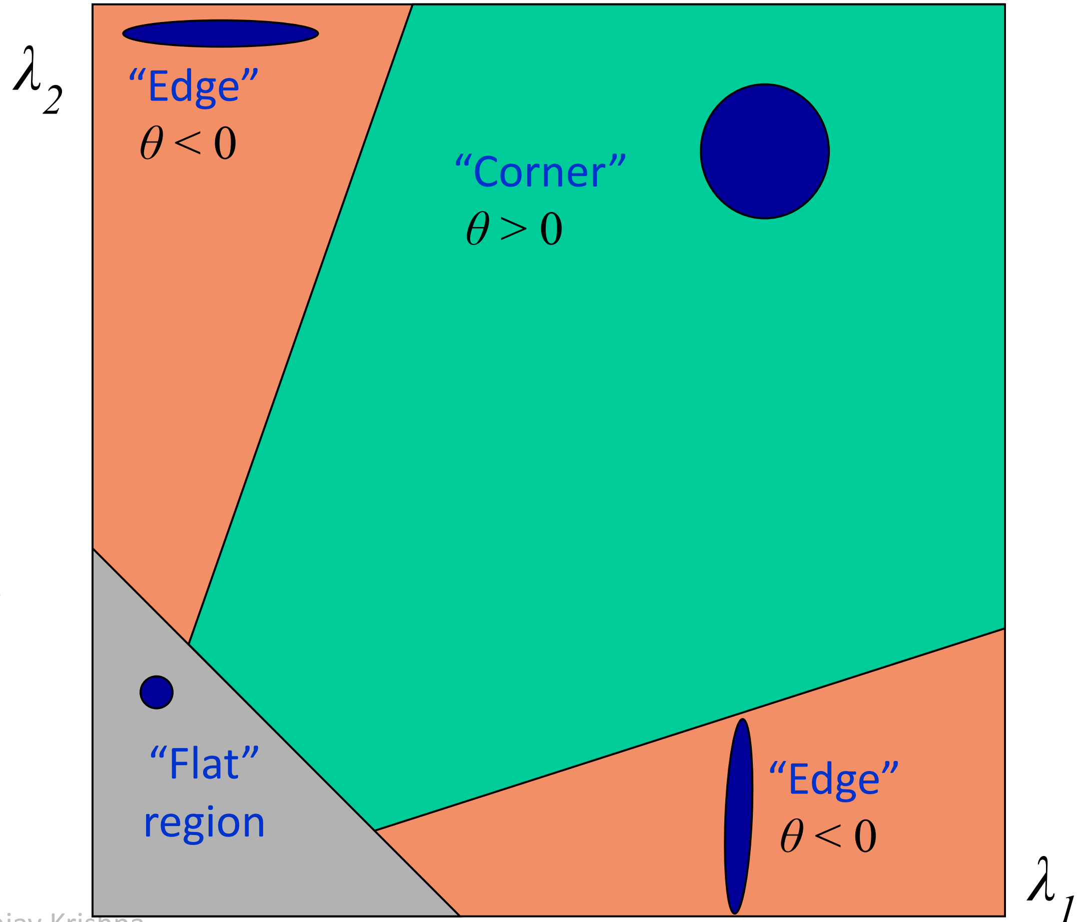


Slide credit: Kristen Grauman

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Corner Response Function

$$\theta = \det(M) - \alpha\, \mathrm{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$



- Fast approximation
  - Avoid computing the eigenvalues
  - α: constant (0.04 to 0.06)

Slide credit: Kristen Grauman

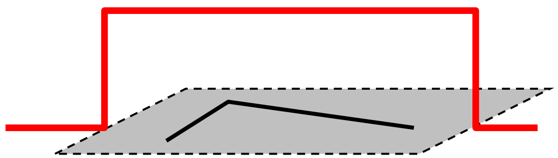
Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Harris - FAST IMPLEMENTATION

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

- Option 1: uniform window
  - Sum over square window (*box filter,* which is *separable*)

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

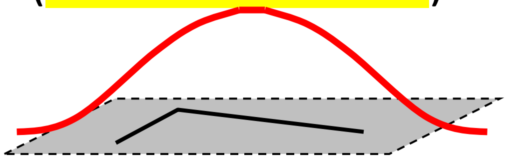


1 in window, 0 outside

  - Problem: not rotation invariant

- Option 2: Smooth with Gaussian
  - Gaussian already performs weighted sum (also see next slide)

$$M = g(\sigma) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$



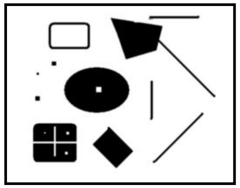
Gaussian

  - Result is rotation invariant

Slide credit: Bastian Leibe

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

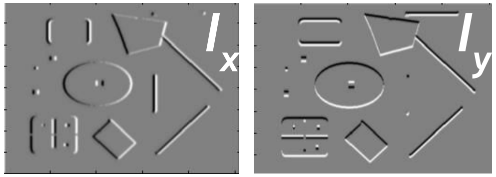
Summary: Harris Detector [Harris88]



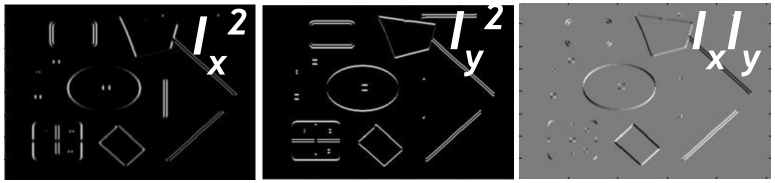
- Compute second moment matrix (autocorrelation matrix)

$$M(\sigma_I,\sigma_D)=g(\sigma_I)*\begin{bmatrix} I_x^2(\sigma_D) & I_xI_y(\sigma_D) \\ I_xI_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

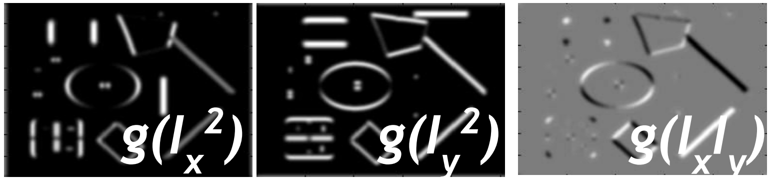
1. Image derivatives



2. Square of derivatives



3. Gaussian filter $g(\sigma_I)$



4. Cornerness function – two strong eigenvalues

$$\theta=\det[M(\sigma_I,\sigma_D)]-\alpha[\mathrm{trace}(M(\sigma_I,\sigma_D))]^2$$
$$=g(I_x^2)g(I_y^2)-[g(I_xI_y)]^2-\alpha[g(I_x^2)+g(I_y^2)]^2$$

5. Perform non-maximum suppression

Slide credit: Krystian Mikolajczyk

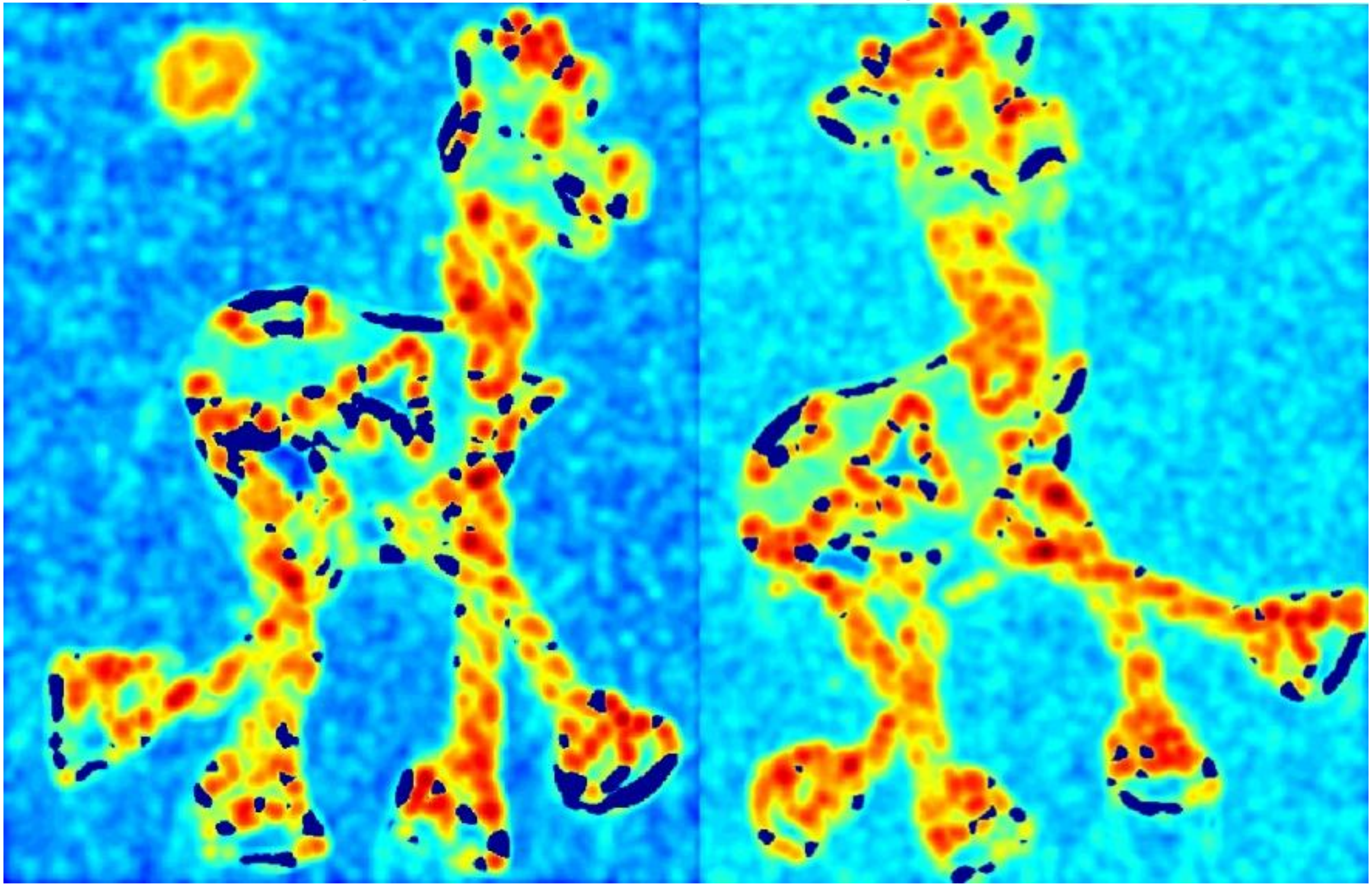
Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Harris Detector: Workflow

Slide adapted from Darya Frolova, Denis Simakov

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Harris Detector: Workflow
- computer corner responses θ

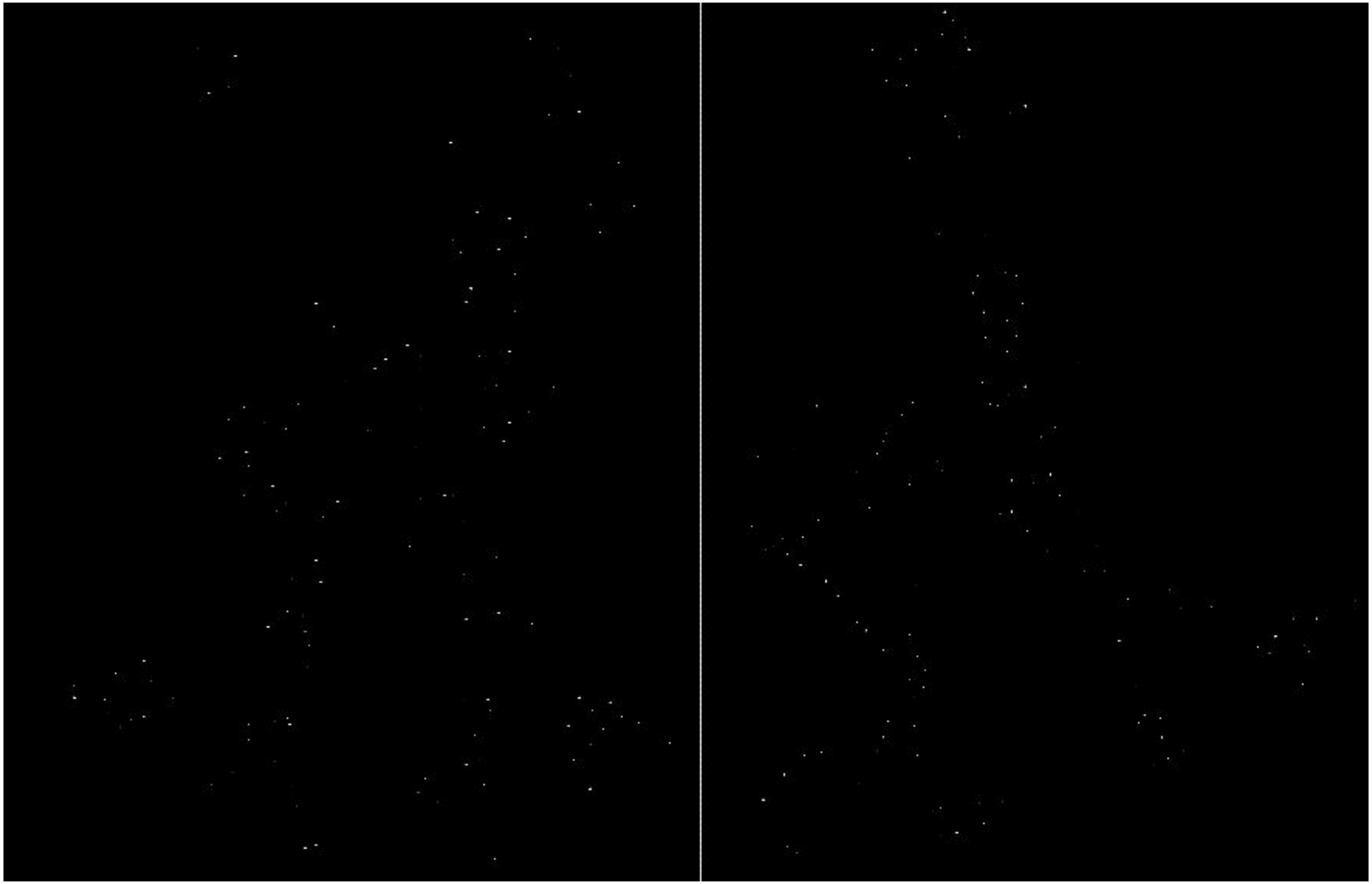


Slide adapted from Darya Frolova, Denis Simakov

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Harris Detector: Workflow

- Take only the local maxima of θ, where θ>threshold



Slide adapted from Darya Frolova, Denis Simakov

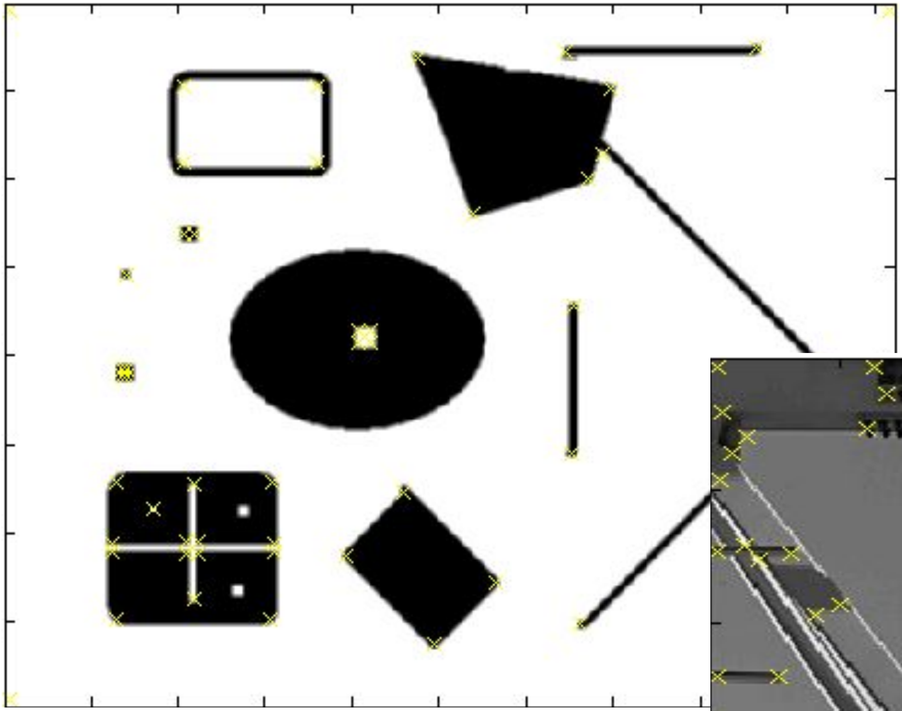
Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

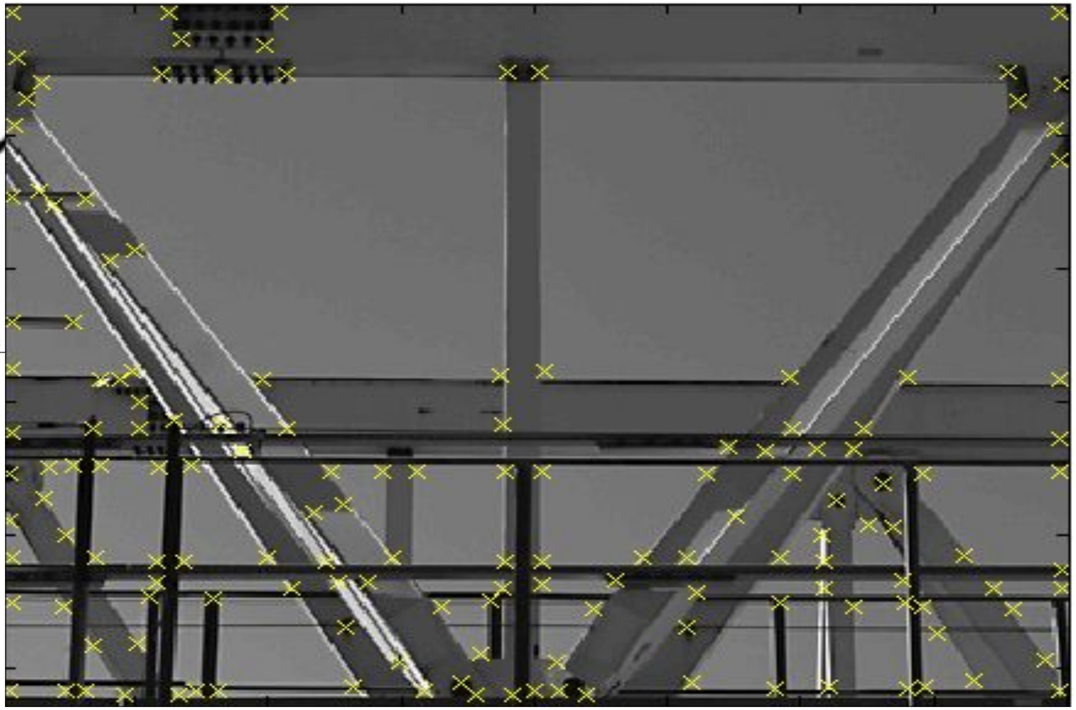
Harris Detector: Workflow
- Resulting Harris points

Slide adapted from Darya Frolova, Denis Simakov

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Harris Detector – Responses [Harris88]





Effect: **A very precise corner detector.**

Slide credit: Krystian Mikolajczyk

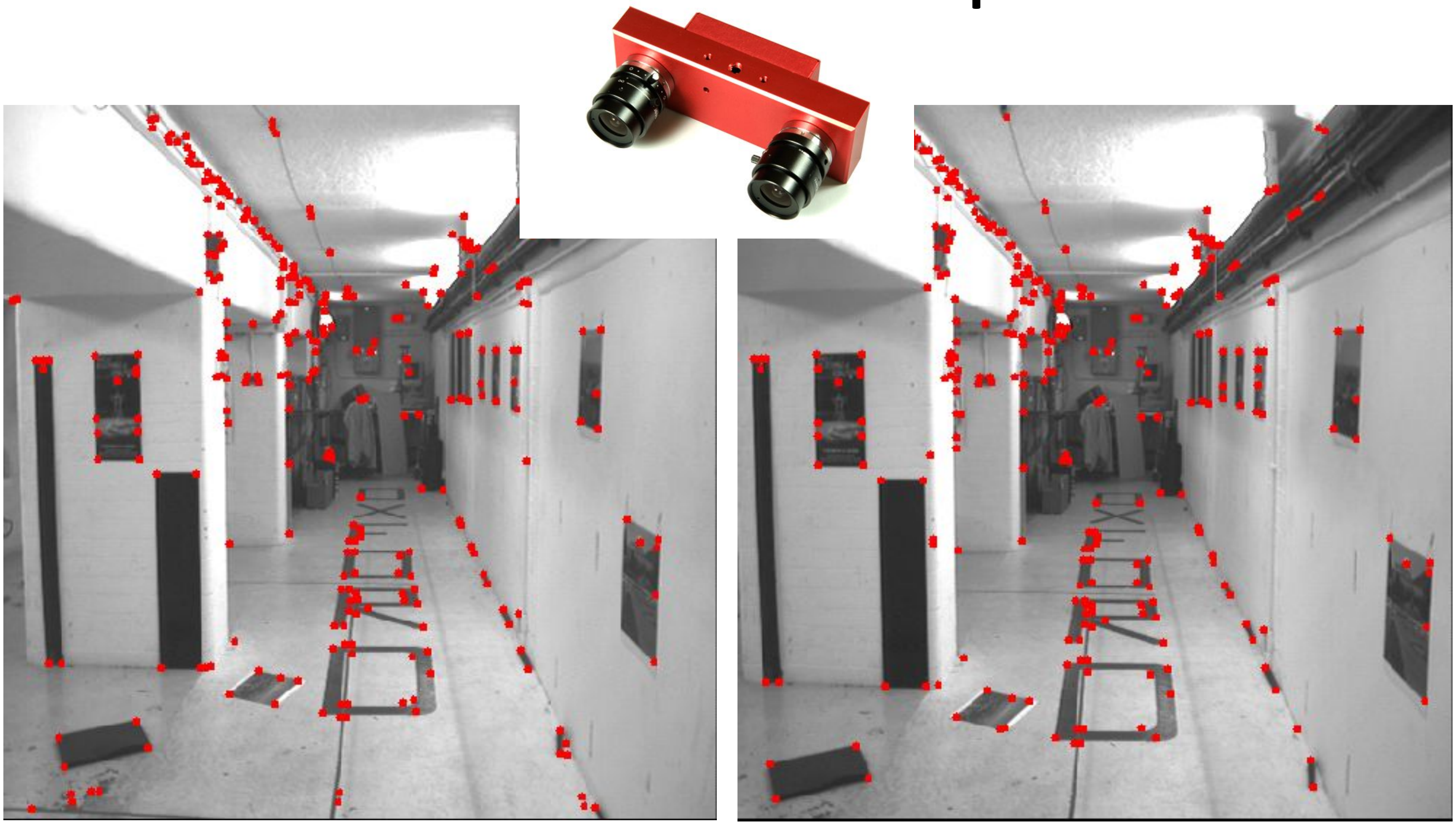
Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Harris Detector – Responses [Harris88]

Slide credit: Krystian Mikolajczyk

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Harris Detector – Responses [Harris88]



- Results are well suited for finding stereo correspondences

Slide credit: Kristen Grauman

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

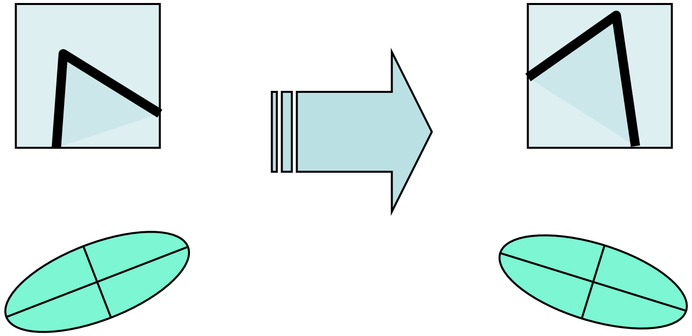
Harris Detector: Properties

- Translation invariance?

Slide credit: Kristen Grauman

Harris Detector: Properties

- Translation invariance
- Rotation invariance?



Ellipse rotates but its shape (i.e. eigenvalues) remains the same

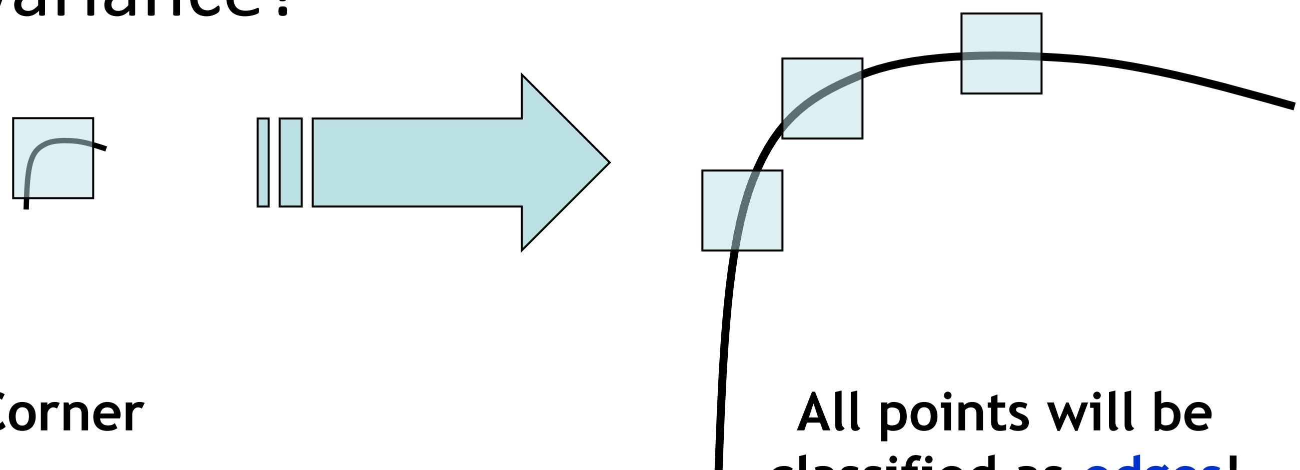
***Corner response θ* is invariant to image rotation**

Slide credit: Kristen Grauman

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Harris Detector: Properties

- Translation invariance
- Rotation invariance
- Scale invariance?



Corner

All points will be classified as edges!

***Not* invariant to image scale!**

(Also remember the relative discussion at edge detection)

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Slide credit: Kristen Grauman

What we learned so far?

- Local invariant features
  - Motivation
  - Requirements, invariances
- Keypoint localization
  - Harris corner detector

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna