Lecture: RANSAC and feature detectors

Juan Carlos Niebles and Ranjay Krishna
Stanford Vision and Learning Lab

What we will learn today?

- A model fitting method for edge detection
  - RANSAC
- Local invariant features
  - Motivation
  - Requirements, invariances
- Keypoint localization
  - Harris corner detector

What we will learn today

- A model fitting method for line detection
  - RANSAC
- Local invariant features
  - Motivation
  - Requirements, invariances
- Keypoint localization
  - Harris corner detector

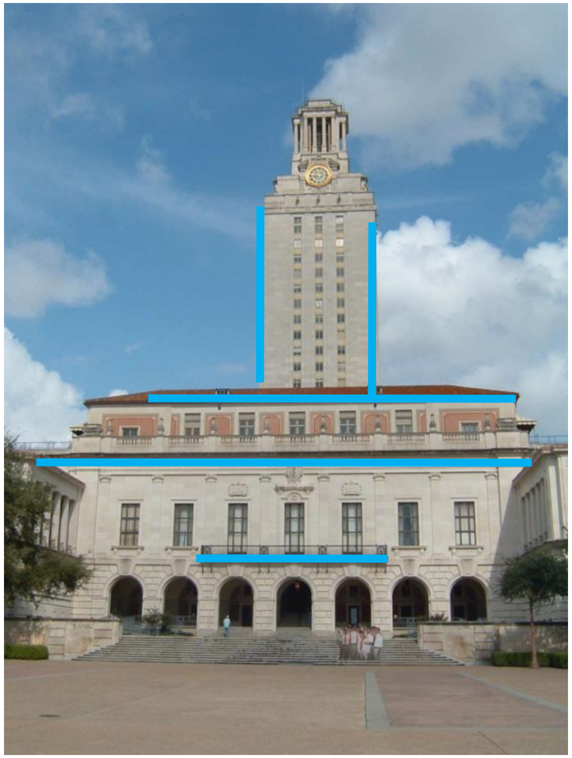
Fitting as Search in Parametric Space

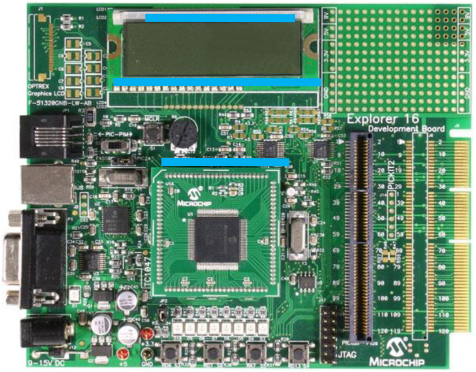
- Choose a parametric model to represent a set of features
- Membership criterion is not local
  - Can't tell whether a point belongs to a given model just by looking at that point.
- Three main questions:
  - What model represents this set of features best?
  - Which of several model instances gets which feature?
  - How many model instances are there?
- Computational complexity is important
  - It is infeasible to examine every possible set of parameters and every possible combination of features

Source: L. Lazebnik

Example: Line Fitting

- Why fit lines?
  Many objects characterized by presence of straight lines

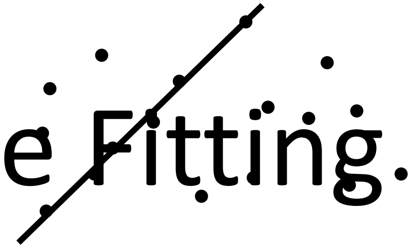


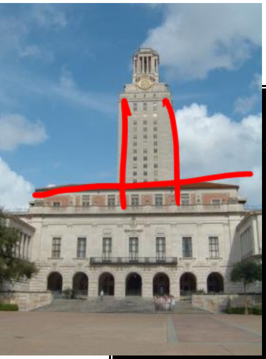


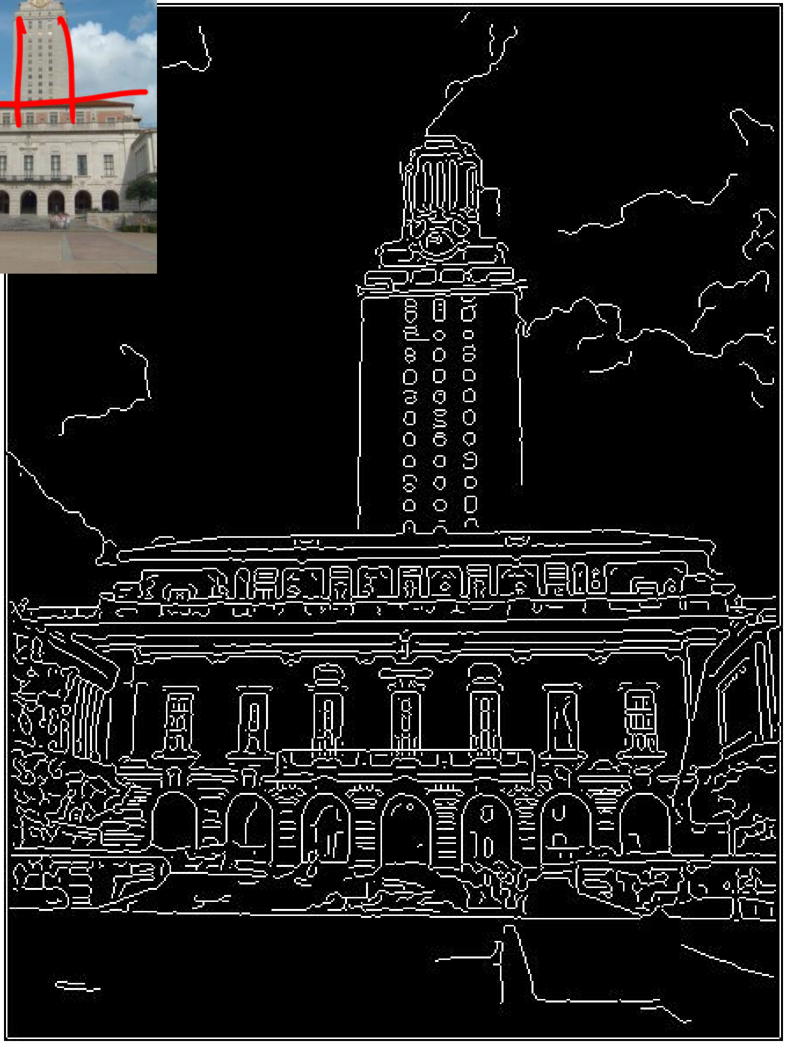
- Wait, why aren't we done just by running edge detection?

Slide credit: Kristen Grauman

Difficulty of Line Fitting







- Extra edge points (clutter), multiple models:
  - Which points go with which line, if any?
- Only some parts of each line detected, and some parts are missing:
  - How to find a line that bridges missing evidence?
- Noise in measured edge points, orientations:
  - How to detect true underlying parameters?

Slide credit: Kristen Grauman

Voting

- It’s not feasible to check all combinations of features by fitting a model to each possible subset.
- Voting is a general technique where we let the features vote for all models that are compatible with it.
  - Cycle through features, cast votes for model parameters.
  - Look for model parameters that receive a lot of votes.
- Noise & clutter features will cast votes too, *but* typically their votes should be inconsistent with the majority of “good” features.
- Ok if some features not observed, as model can span multiple fragments.

Slide credit: Kristen Grauman

RANSAC [Fischler & Bolles 1981]

- RANdom SAmple Consensus

- Approach: we want to avoid the impact of outliers, so let’s look for “inliers”, and use only those.

- Intuition: if an outlier is chosen to compute the current fit, then the resulting line won’t have much support from rest of the points.

Slide credit: Kristen Grauman

RANSAC [Fischler & Bolles 1981]

RANSAC loop:

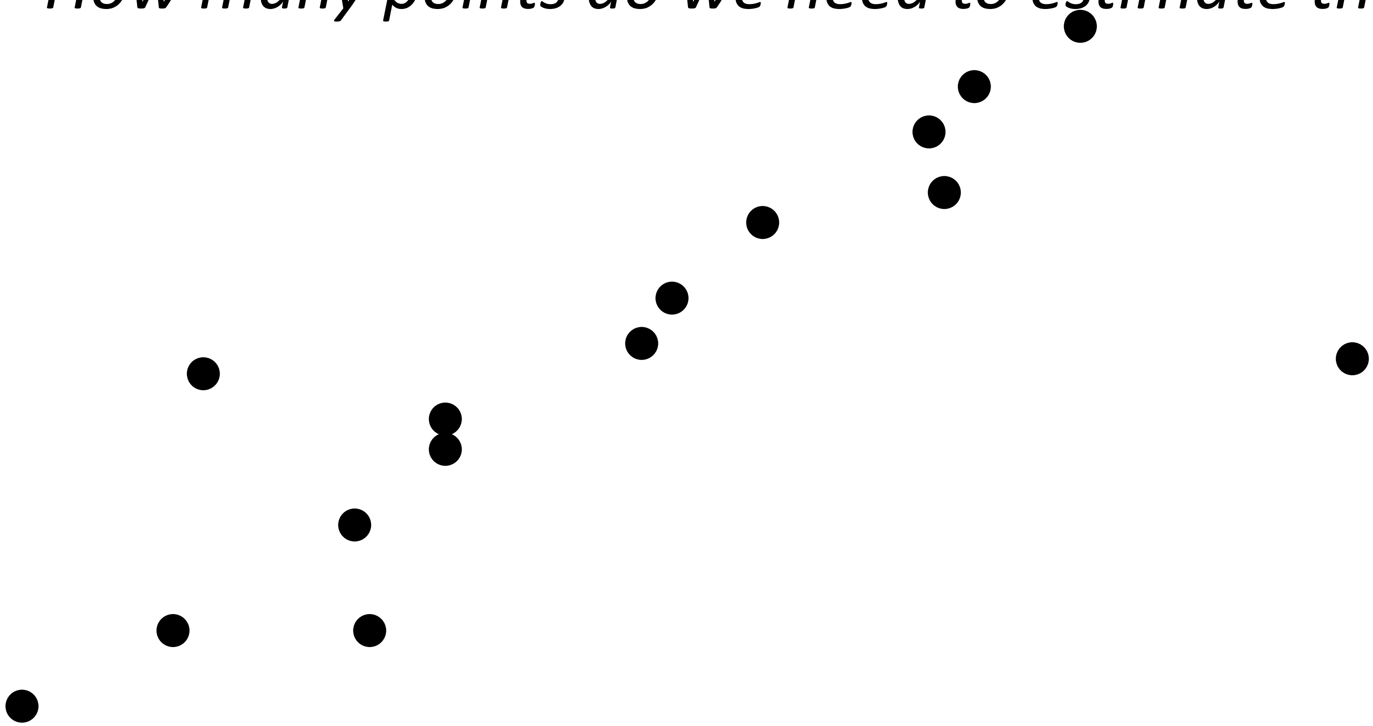
1. Randomly select a *seed group* of points on which to base transformation estimate (e.g., a group of matches)
2. Compute transformation from seed group
3. Find *inliers* to this transformation
4. If the number of inliers is sufficiently large, re-compute least-squares estimate of transformation on all of the inliers

- Keep the transformation with the largest number of inliers

Slide credit: Kristen Grauman

RANSAC Line Fitting Example

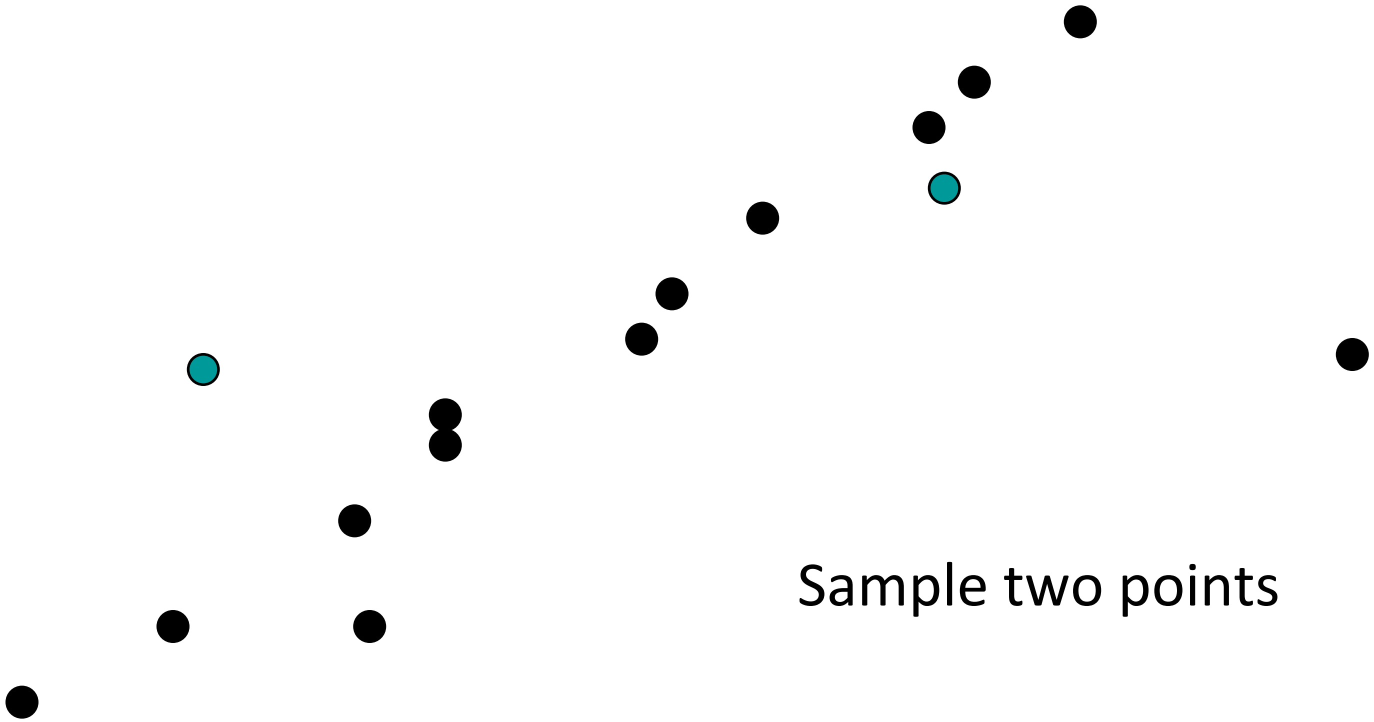
- Task: Estimate the best line
  - *How many points do we need to estimate the line?*



Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

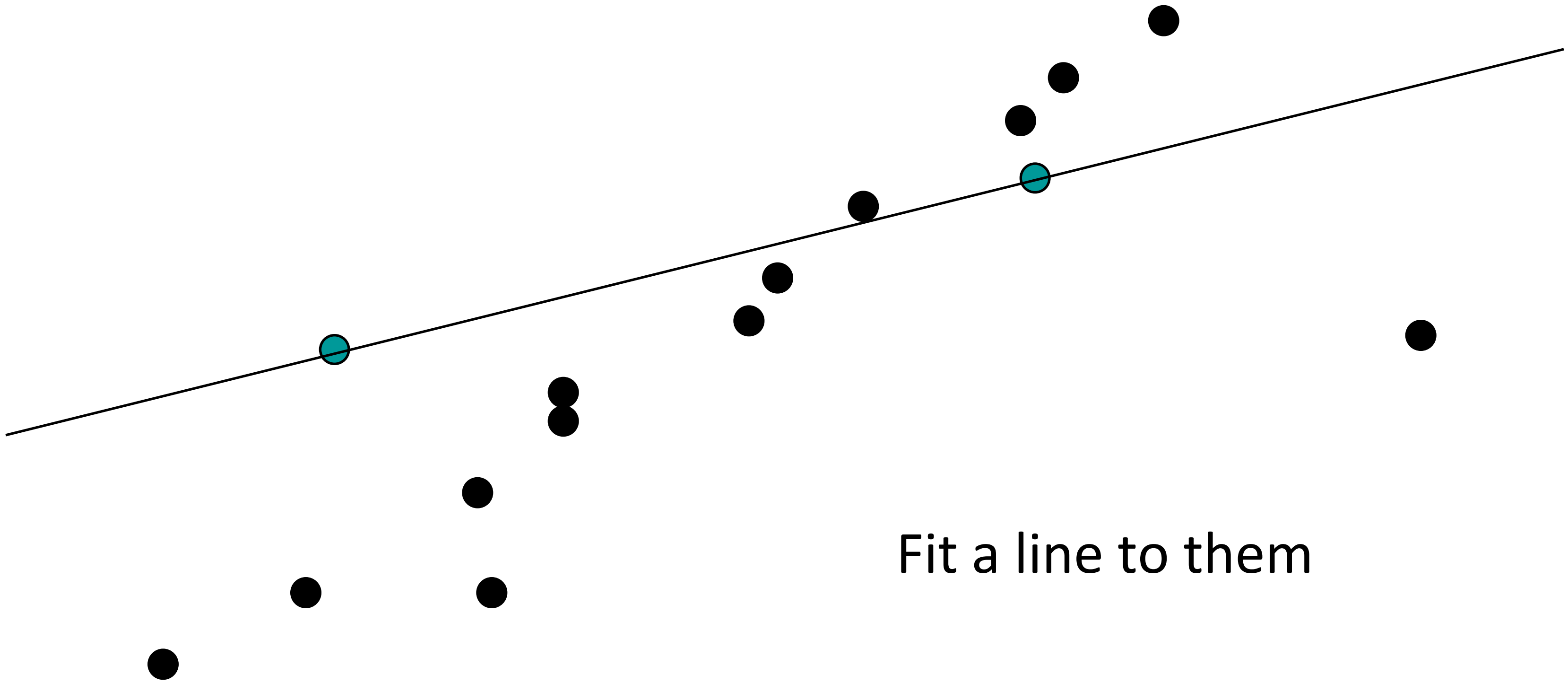
- Task: Estimate the best line



Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

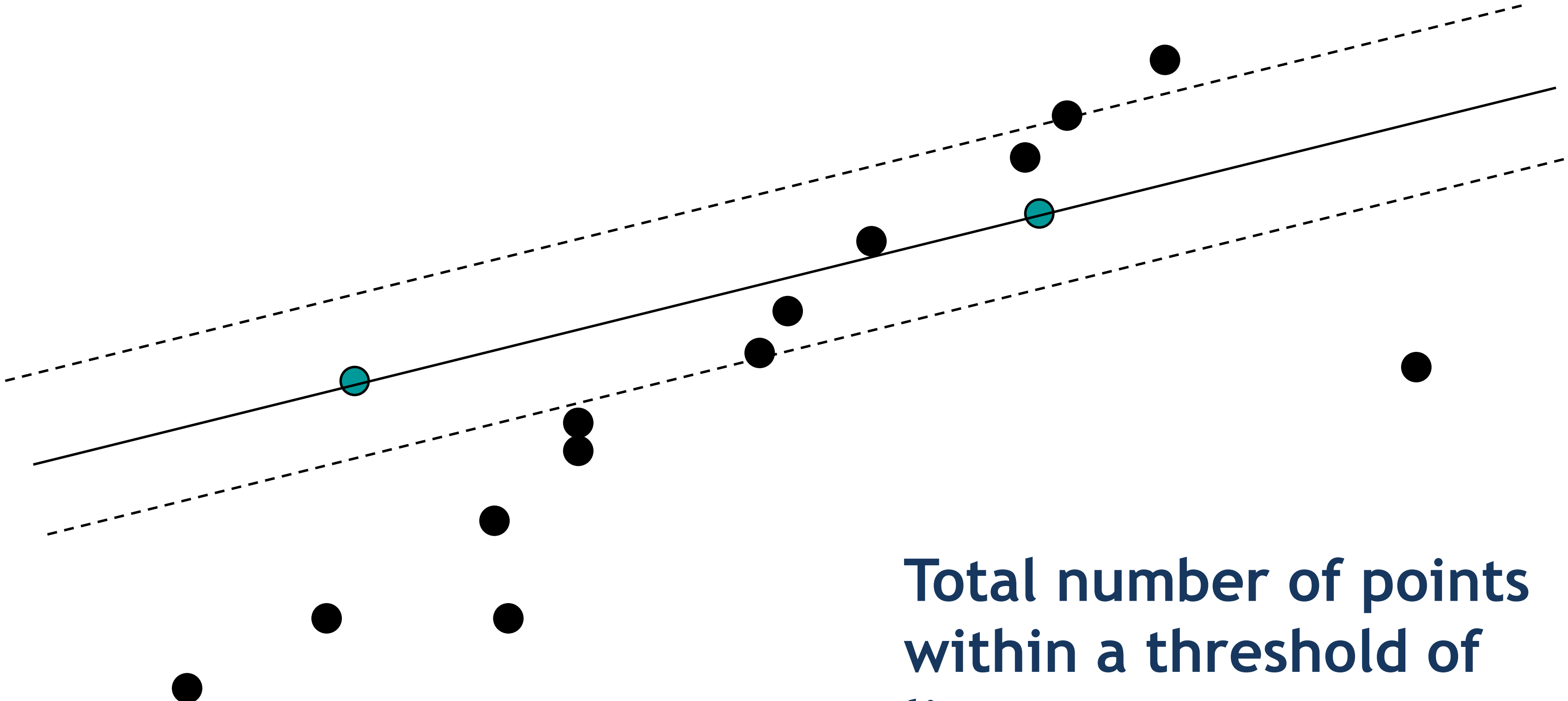
- Task: Estimate the best line



Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

- Task: Estimate the best line

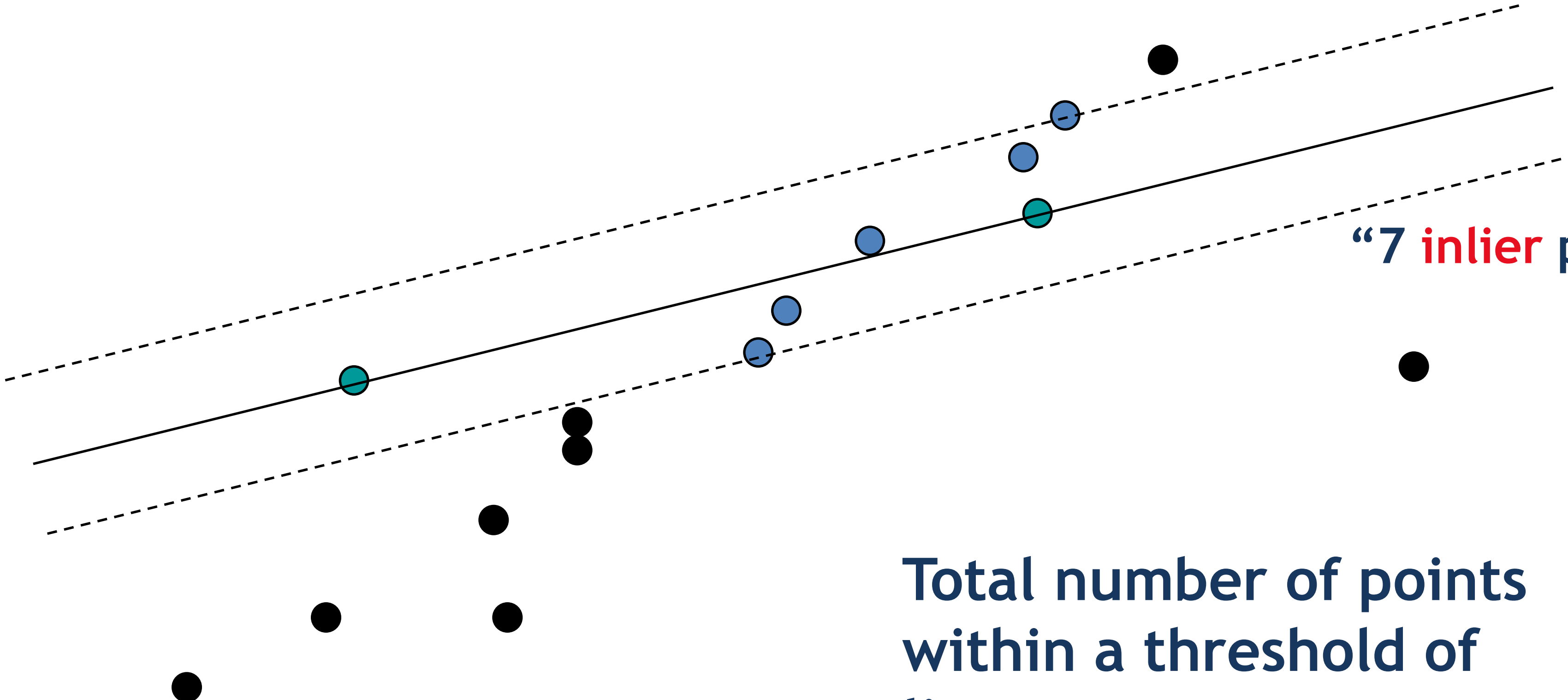


Total number of points within a threshold of line.

Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

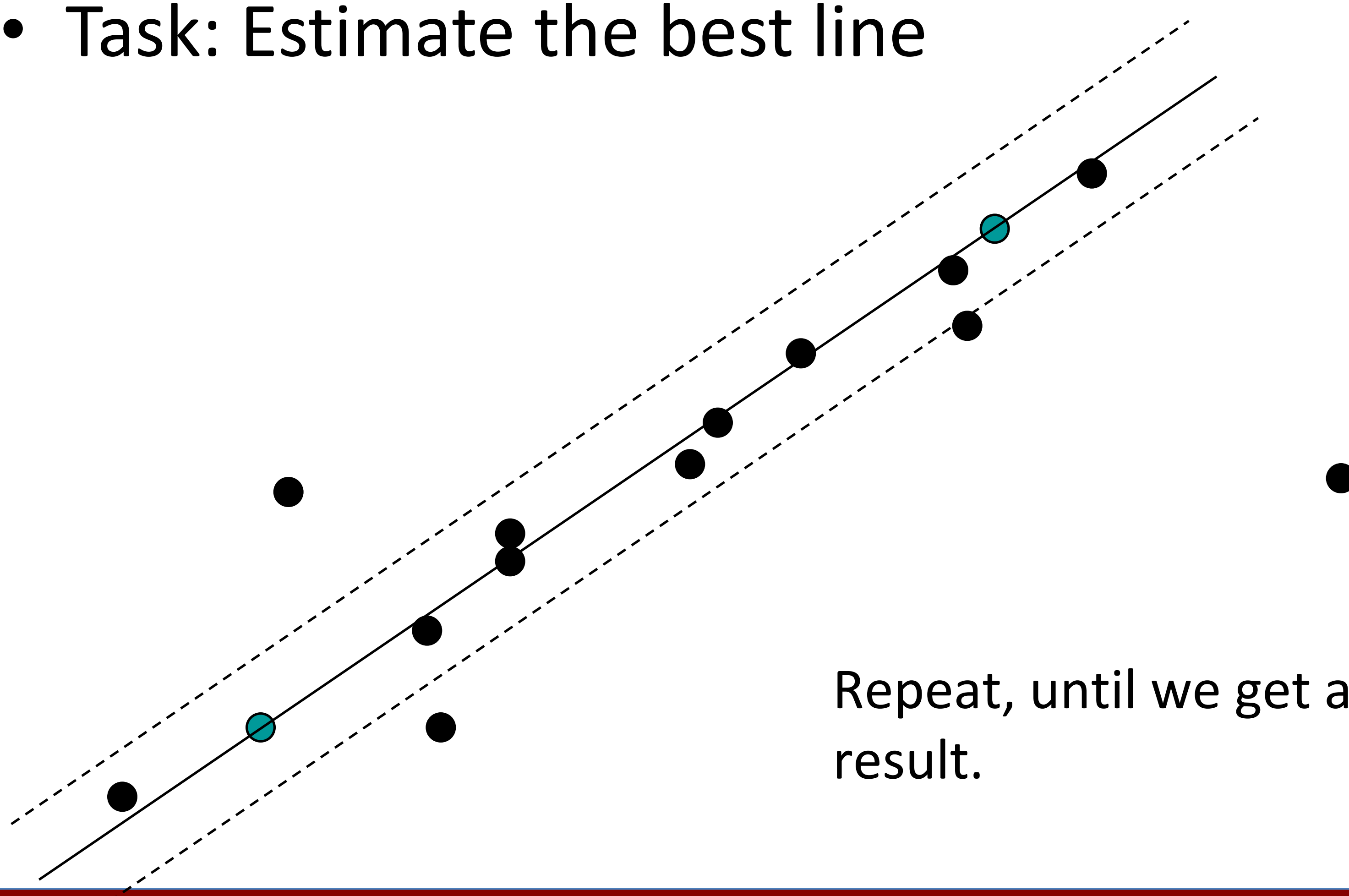
- Task: Estimate the best line



Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

- Task: Estimate the best line

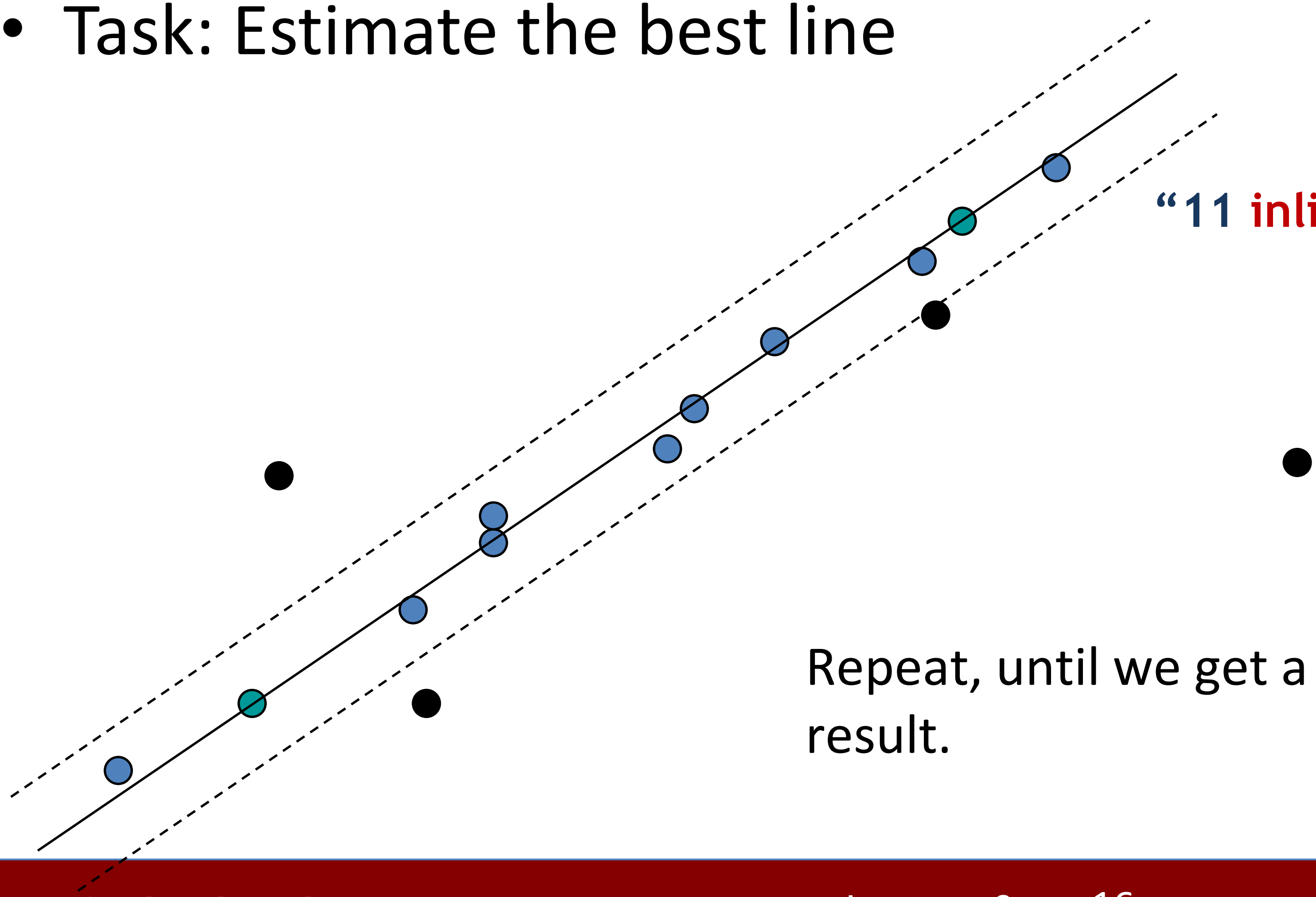


Repeat, until we get a good result.

Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

- Task: Estimate the best line



Repeat, until we get a good result.

Slide credit: Jinxiang Chai

Algorithm 15.4: RANSAC: fitting lines using random sample consensus

Determine:
    n — the smallest number of points required
    k — the number of iterations required
    t — the threshold used to identify a point that fits well
    d — the number of nearby points required
        to assert a model fits well
Until k iterations have occurred
    Draw a sample of n points from the data
        uniformly and at random
    Fit to that set of n points
    For each data point outside the sample
        Test the distance from the point to the line
            against t; if the distance from the point to the line
            is less than t, the point is close
    end
    If there are d or more points close to the line
        then there is a good fit. Refit the line using all
        these points.
end
Use the best fit from this collection, using the
    fitting error as a criterion

RANSAC: How many samples?

- How many samples are needed?
  - Suppose w is fraction of inliers (points from line).
  - n points needed to define hypothesis (2 for lines)
  - k samples chosen.
- Prob. that a single sample of n points is correct: w^n
- Prob. that all k samples fail is: $(1-w^n)^k$

$\Rightarrow$ Choose k high enough to keep this below desired failure rate.

Slide credit: David Lowe

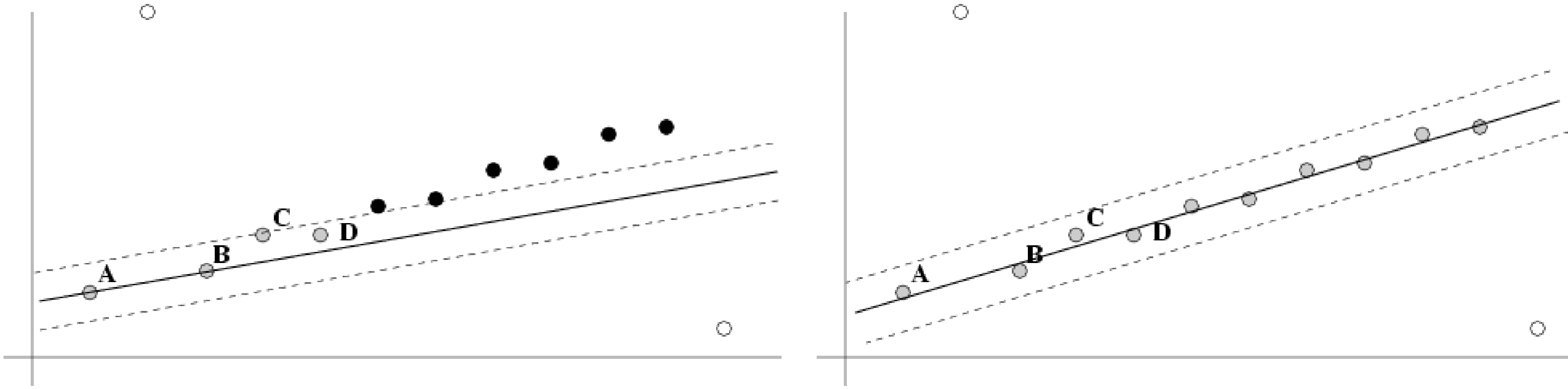
RANSAC: Computed k (p=0.99)

Sample size n	Proportion of outliers						
	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Slide credit: David Lowe

After RANSAC

- RANSAC divides data into inliers and outliers and yields estimate computed from minimal set of inliers.
- Improve this initial estimate with estimation over all inliers (e.g. with standard least-squares minimization).
- But this may change inliers, so alternate fitting with re-classification as inlier/outlier.



Slide credit: David Lowe

RANSAC: Pros and Cons

- **<u>Pros</u>:**
  - General method suited for a wide range of model fitting problems
  - Easy to implement and easy to calculate its failure rate
- **<u>Cons</u>:**
  - Only handles a moderate percentage of outliers without cost blowing up
  - Many real problems have high rate of outliers (but sometimes selective choice of random subsets can help)
- A voting strategy, The Hough transform, can handle high percentage of outliers

What we will learn today?

- A model fitting method for edge detection
  - RANSAC
- Local invariant features
  - Motivation
  - Requirements, invariances
- Keypoint localization
  - Harris corner detector

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

Image matching: a challenging problem

Image matching: a challenging problem

by Diva Sian

by swashford

Slide credit: Steve Seitz

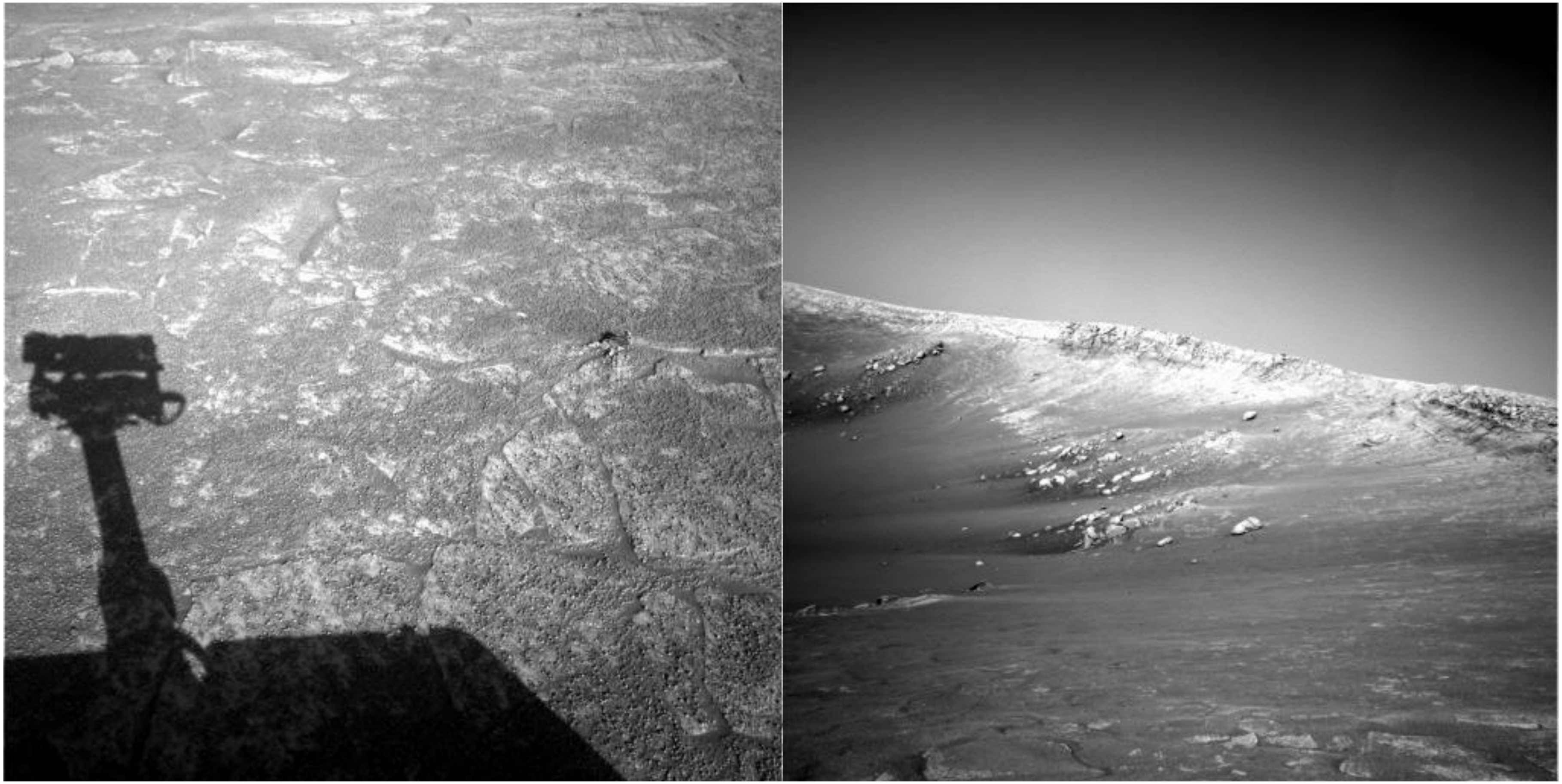
Harder Case

by Diva Sian

by scgbt

Slide credit: Steve Seitz

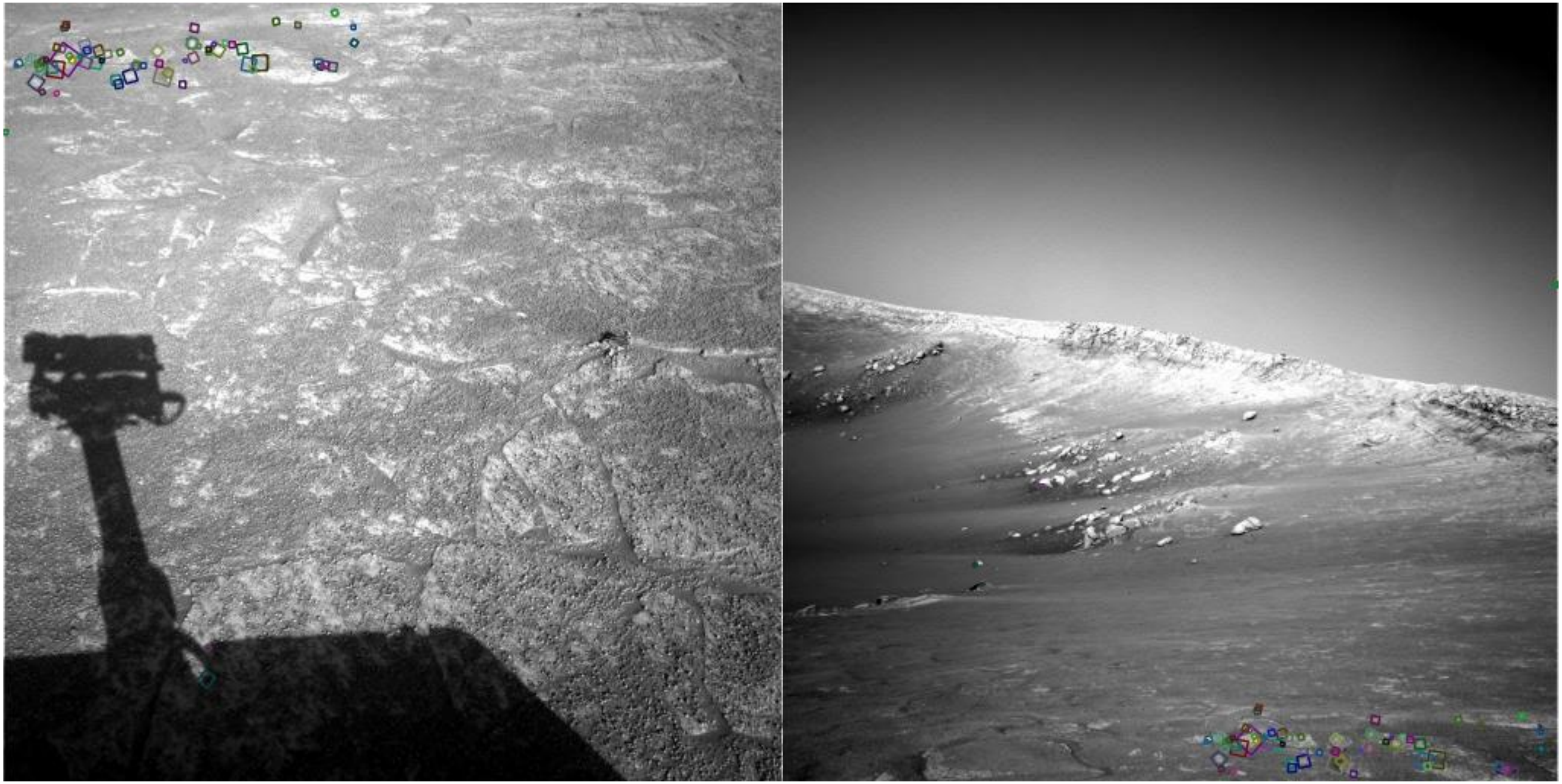
Harder Still?



NASA Mars Rover images

Slide credit: Steve Seitz

Answer Below (Look for tiny colored squares)

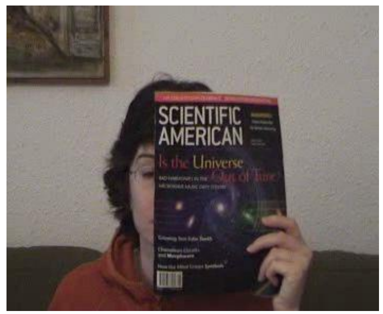


NASA Mars Rover images with SIFT feature matches
(Figure by Noah Snavely)

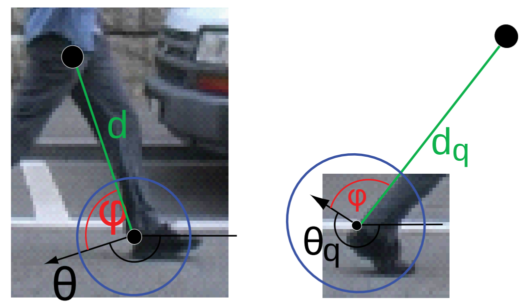
Slide credit: Steve Seitz

Motivation for using local features

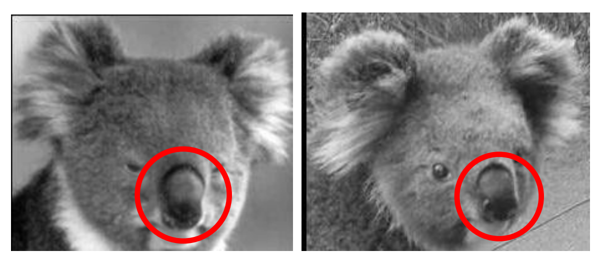
- Global representations have major limitations
- Instead, describe and match only local regions
- Increased robustness to
  - Occlusions



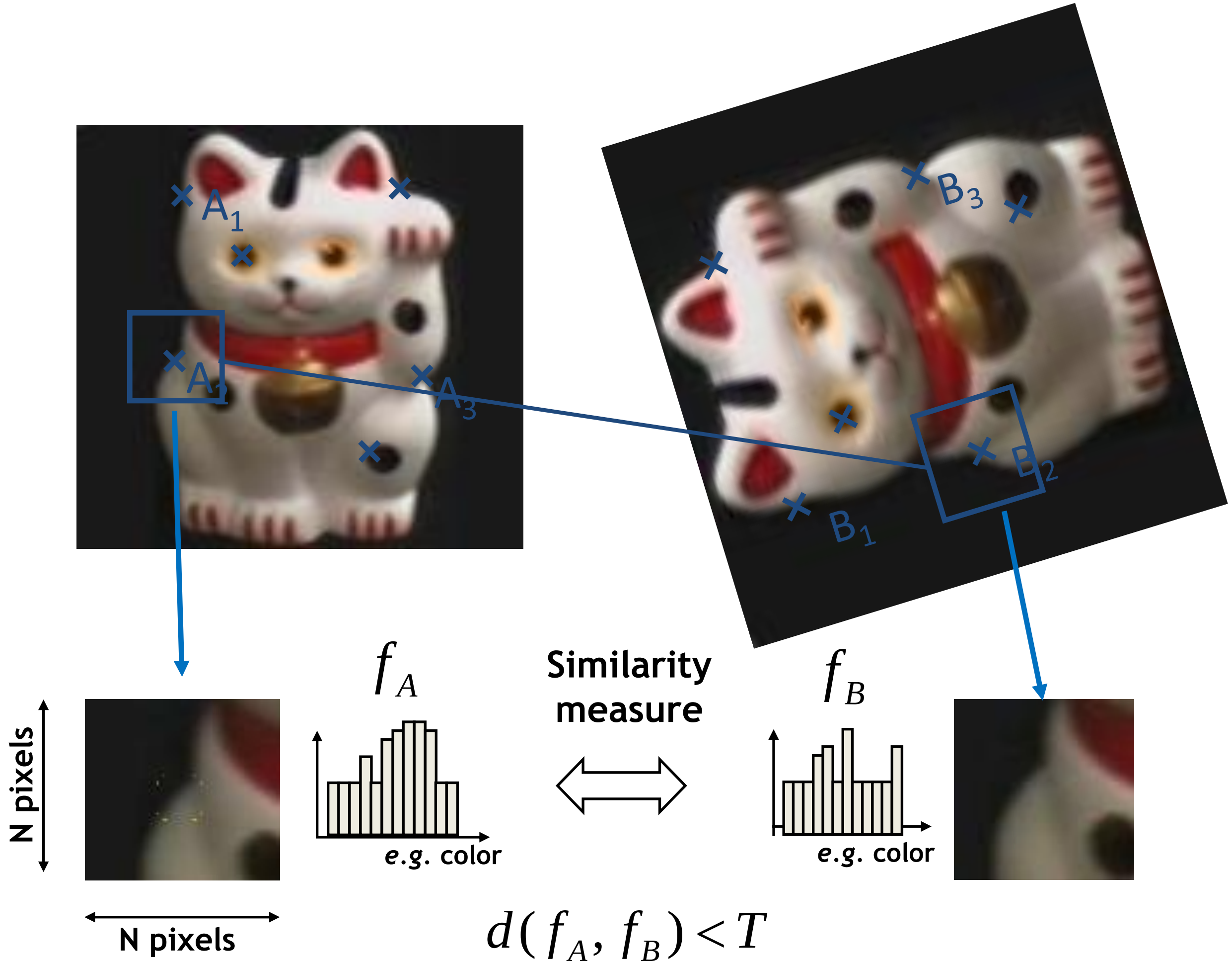
  - Articulation



  - Intra-category variations



General Approach



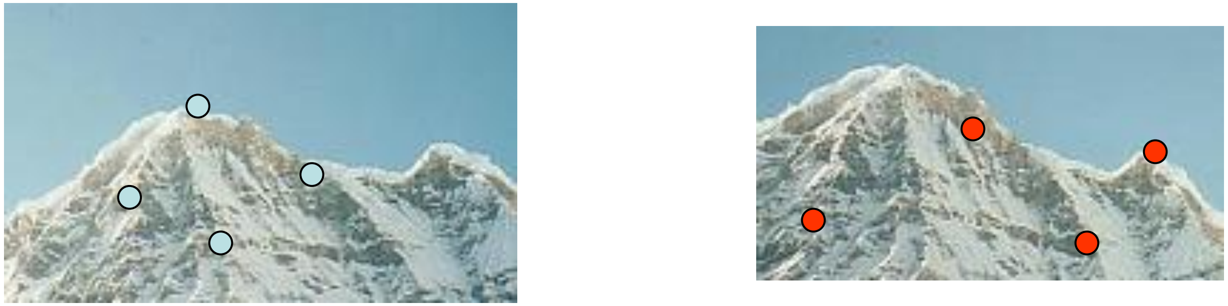
$$d(f_A, f_B) < T$$

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

Common Requirements

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



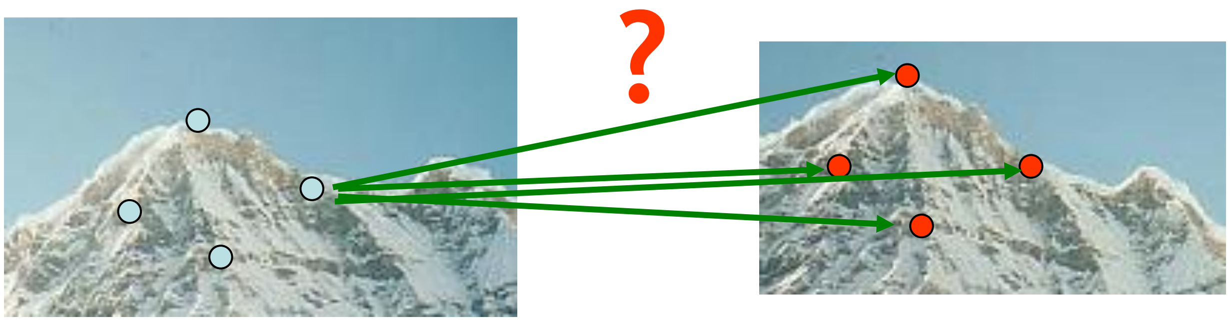
No chance to match!

We need a repeatable detector!

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



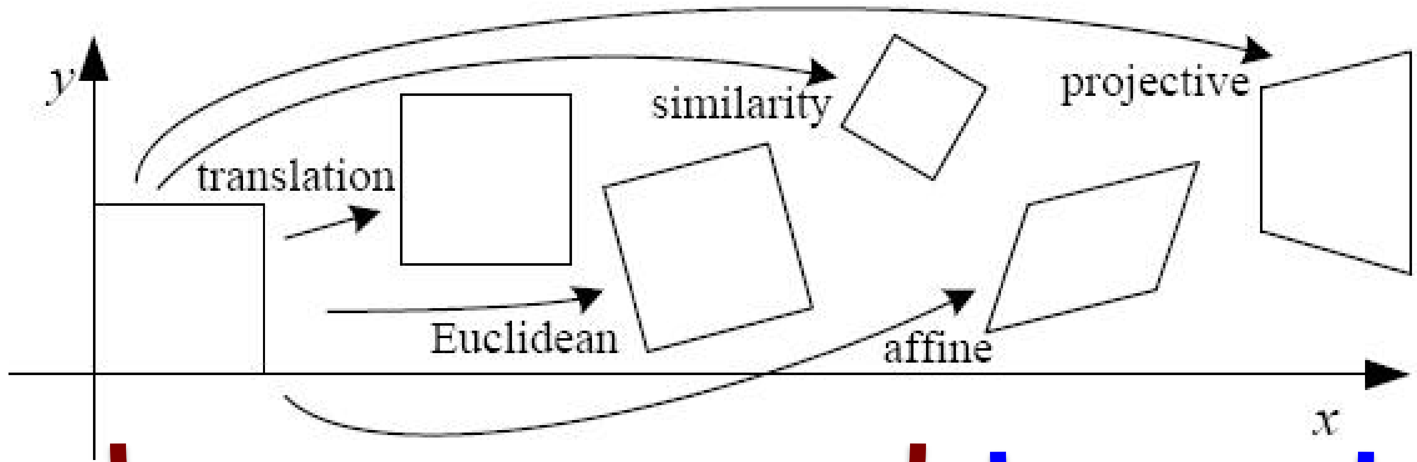
We need a reliable and distinctive descriptor!

Slide credit: Darya Frolova, Denis Simakov

Invariance: Geometric Transformations

Slide credit: Steve Seitz

Levels of Geometric Invariance



CS131

CS231a

Slide credit: Bastian Leibe

Invariance: Photometric Transformations

- Often modeled as a linear transformation:
  - Scaling + Offset

Slide credit: Tinne Tuytelaars

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

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 recent applications in Computer Vision.*

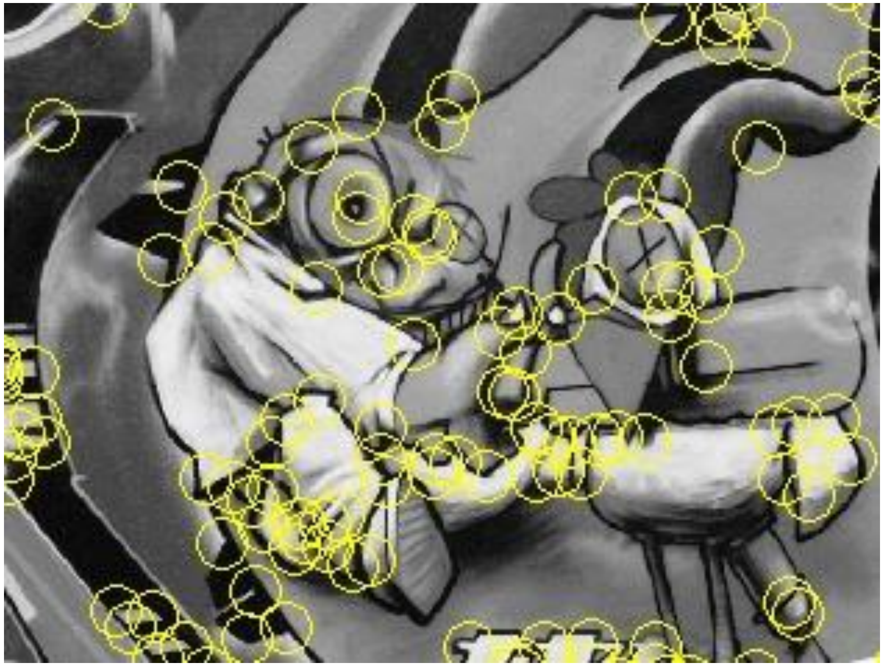
Slide credit: Bastian Leibe

What we will learn today?

- A model fitting method for edge detection
  – RANSAC
- 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

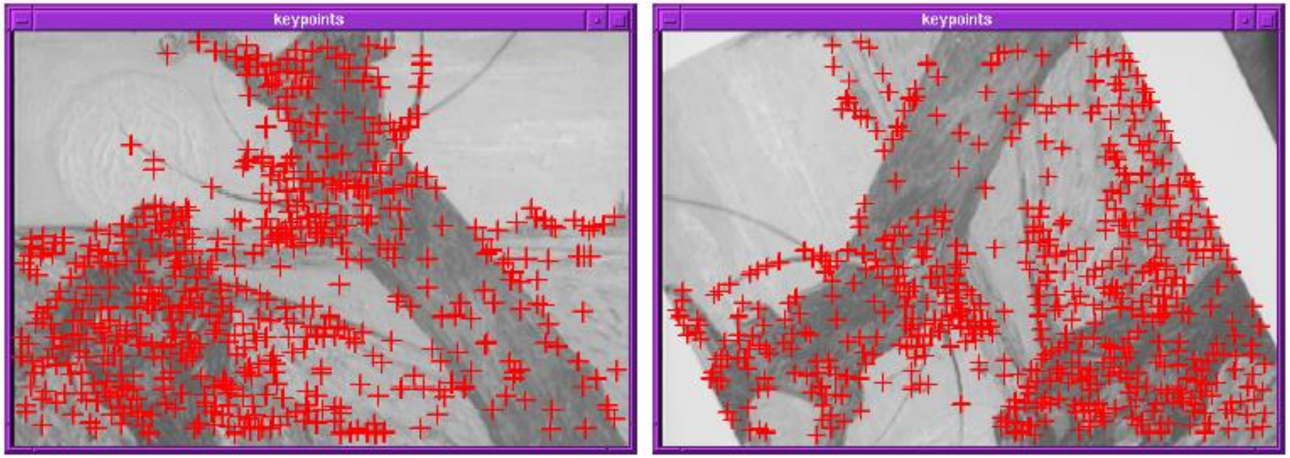


- Goals:
  - Repeatable detection
  - Precise localization
  - Interesting content

$\Rightarrow$ *Look for two-dimensional signal changes*

Slide credit: Bastian Leibe

Finding Corners



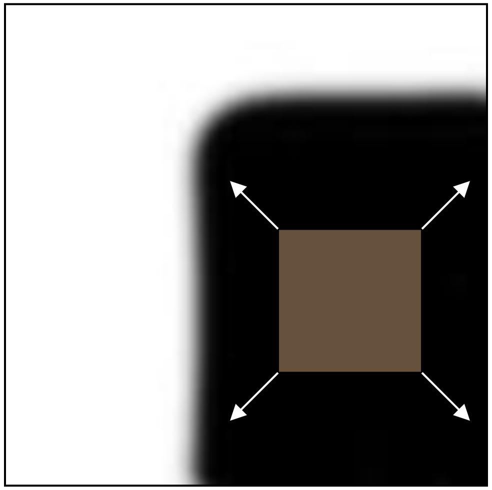
- Key property:
  - In the region around a corner, image gradient has two or more dominant directions
- Corners are *repeatable* and *distinctive*

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

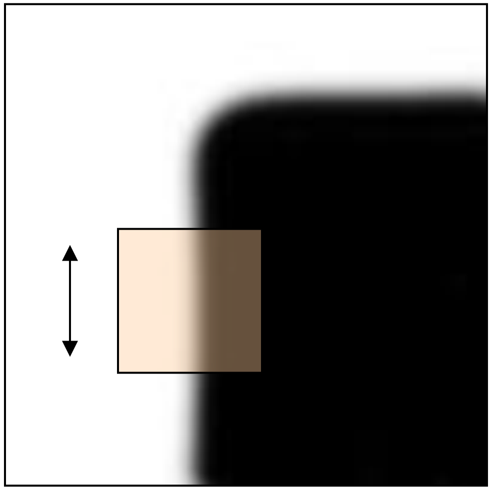
Slide credit: Svetlana Lazebnik

Corners as Distinctive Interest Points

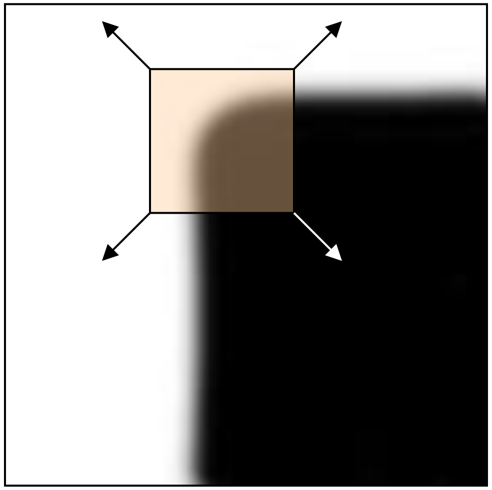
- Design criteria
  - We should easily recognize the point by looking through a small window (*locality*)
  - Shifting the window in *any direction* should give *a large change* in intensity (*good localization*)



"flat" region:
no change in all directions



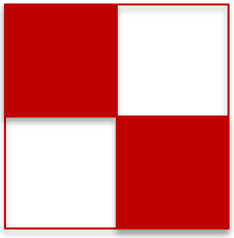
"edge":
no change along the edge direction



"corner":
significant change in all directions

Slide credit: Alyosha Efros

Corners versus edges



$\sum I_x^2$ → Large

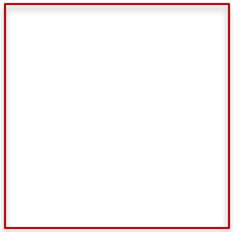
$\sum I_y^2$ → Large

Corner

$\sum I_x^2$ → Small

$\sum I_y^2$ → Large

Edge

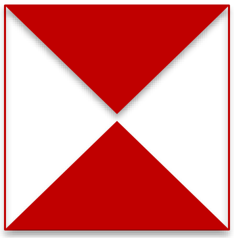


$\sum I_x^2$ → Small

$\sum I_y^2$ → Small

Nothing

Corners versus edges



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

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

Corner

Harris Detector Formulation

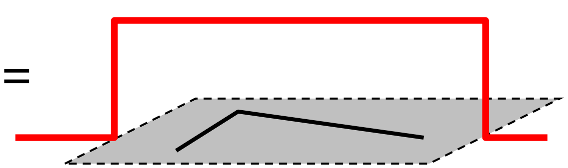
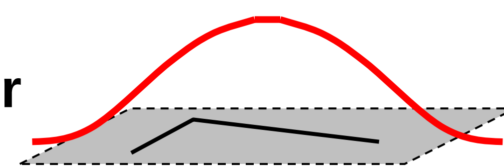
- Change of intensity for the shift [u,v]:

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

Window function

Shifted intensity

Intensity

Window function $w(x,y)$ =  or 

1 in window, 0 outside

Gaussian

Slide credit: Rick Szeliski

Harris Detector Formulation

- This measure of change can be approximated by:

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

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

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

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

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

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

Slide credit: Rick Szeliski

Harris Detector Formulation

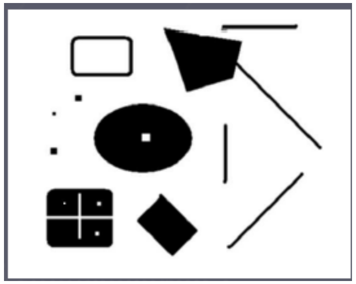
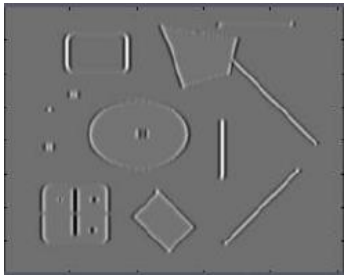
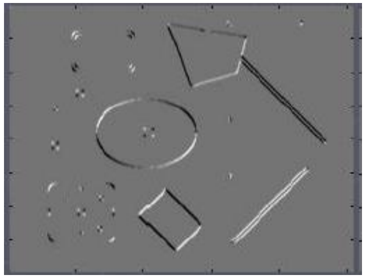


Image I



I_x

I_y



$I_x I_y$

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

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

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

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

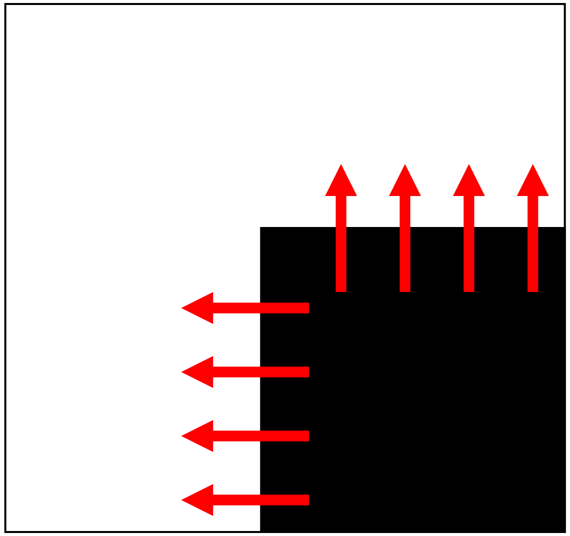
$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y]$$

Slide credit: Rick Szeliski

What Does This Matrix Reveal?

- First, let's consider an axis-aligned corner:

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

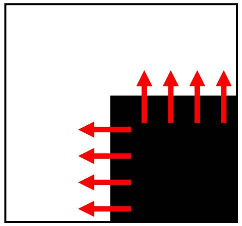


Slide credit: David Jacobs

What Does This Matrix Reveal?

- First, let's consider an axis-aligned corner:

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



- This means:
  - Dominant gradient directions align with x or y axis
  - If either λ is close to 0, then this is not a corner, so look for locations where both are large.
- What if we have a corner that is not aligned with the image axes?

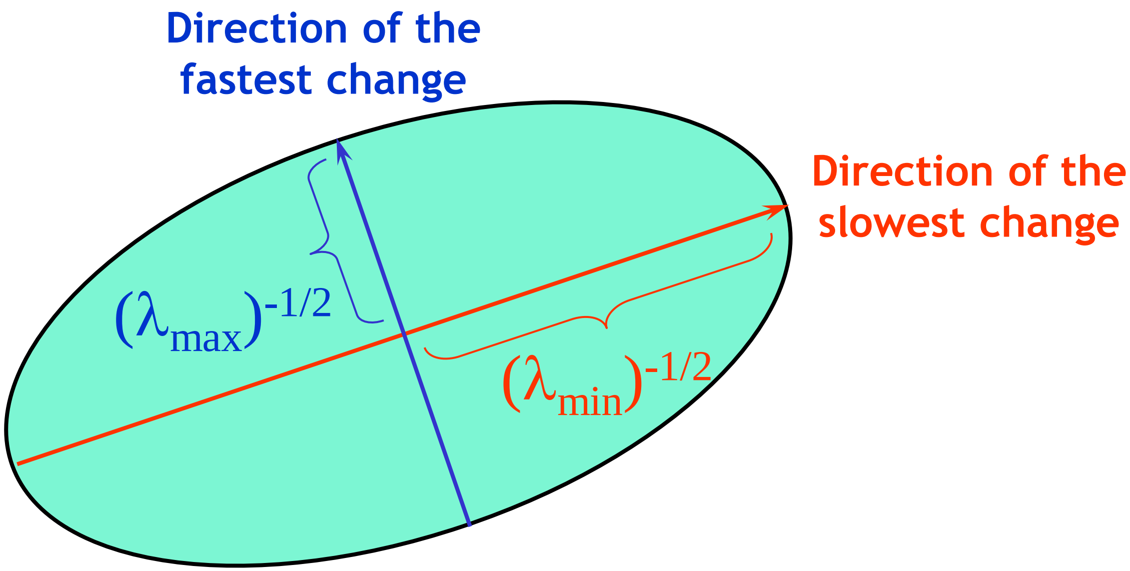
Slide credit: David Jacobs

General Case

- Since M is symmetric, we have $$M = R^{-1}\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

  (Eigenvalue decomposition)

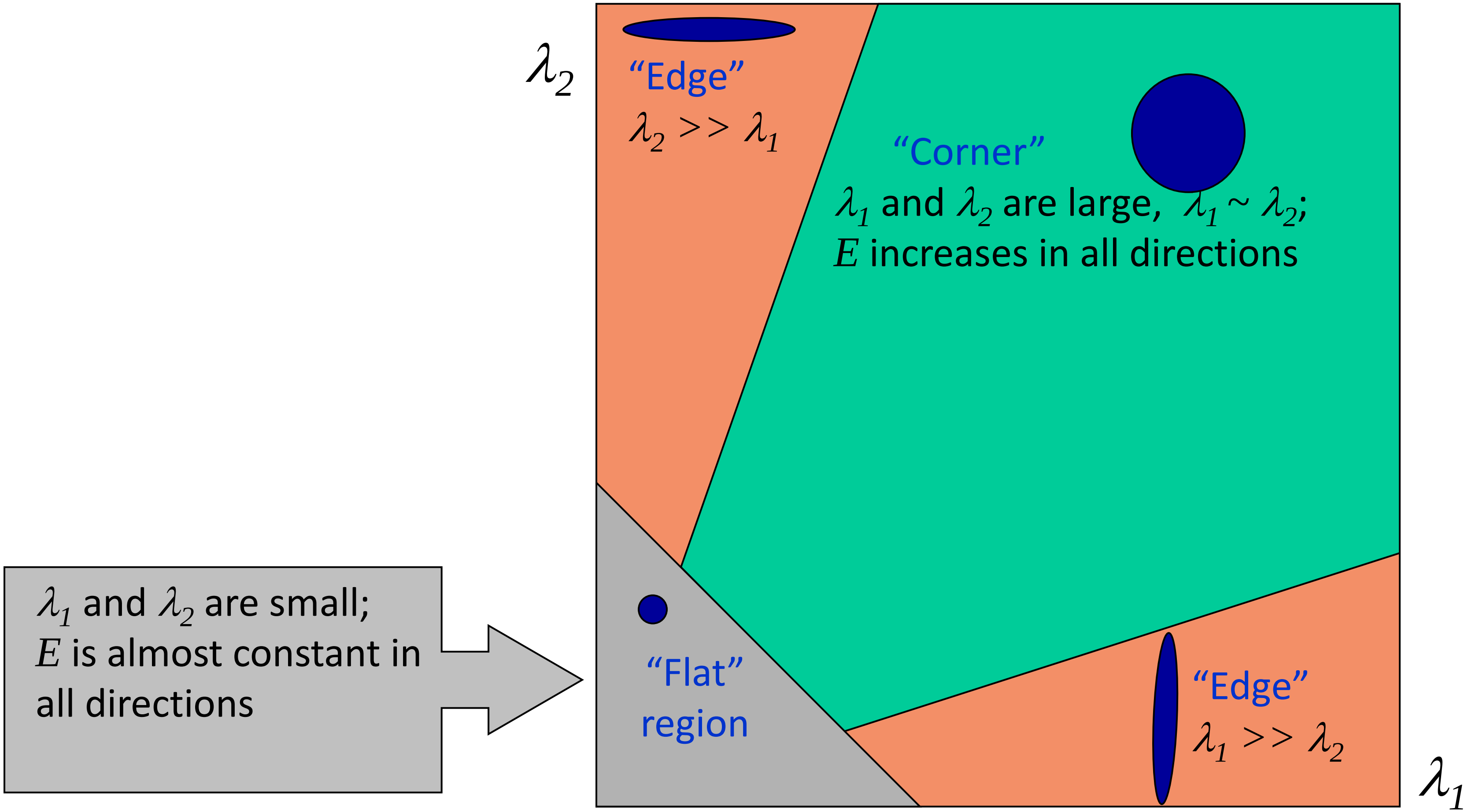
- We can visualize M as an ellipse with axis lengths determined by the eigenvalues and orientation determined by R



adapted from Darya Frolova, Denis Simakov

Interpreting the Eigenvalues

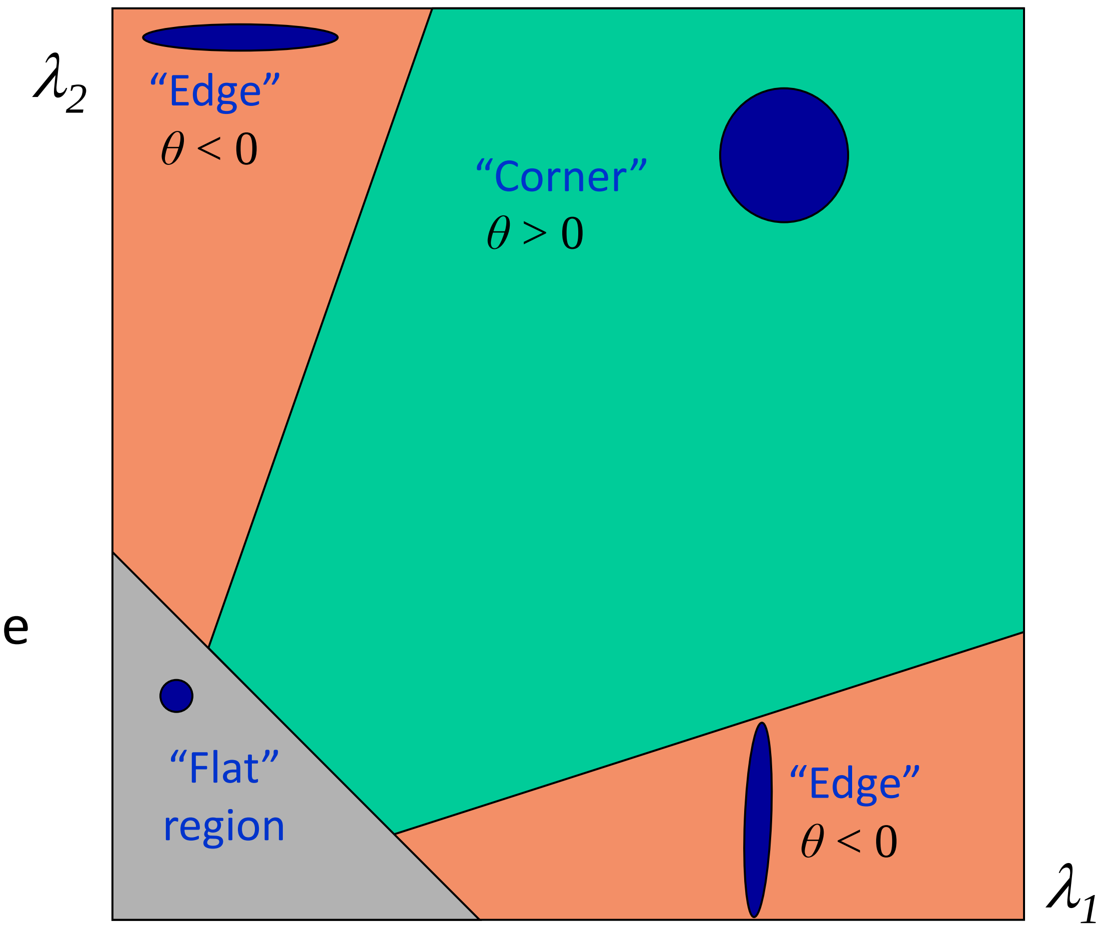
- Classification of image points using eigenvalues of M:



Slide credit: Kristen Grauman

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

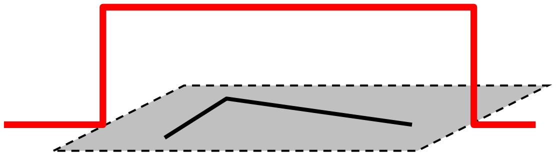
Window Function $w(x,y)$

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

- Option 1: uniform window
  - Sum over square window

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

  - Problem: not rotation invariant

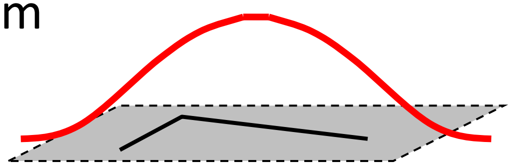


1 in window, 0 outside

- Option 2: Smooth with Gaussian
  - Gaussian already performs weighted sum

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

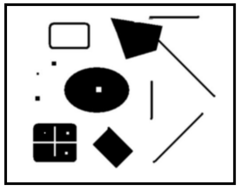
  - Result is rotation invariant



Gaussian

Slide credit: Bastian Leibe

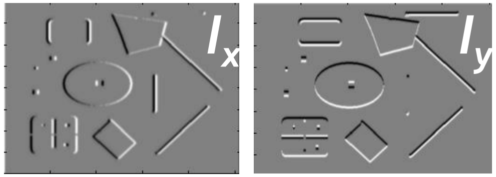
Summary: Harris Detector [Harris88]



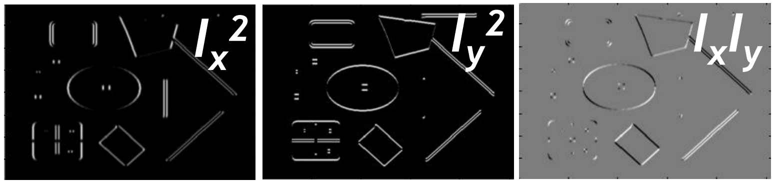
- Compute second moment matrix (autocorrelation matrix)

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

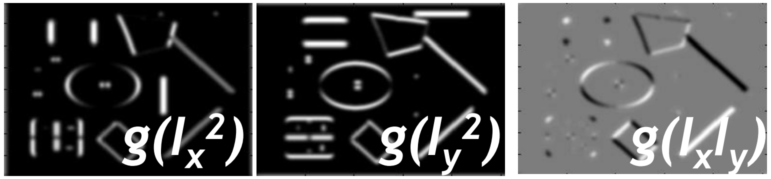
1. Image derivatives



2. Square of derivatives



3. Gaussian filter *g(σ_I)*



4. Cornerness function - two strong eigenvalues

$$q=\det[M(s_I,s_D)]-a[\mathrm{trace}(M(s_I,s_D))]^2$$

$$=g(I_x^2)g(I_y^2)-[g(I_xI_y)]^2-\alpha[g(I_x^2)+g(I_y^2)]^2$$

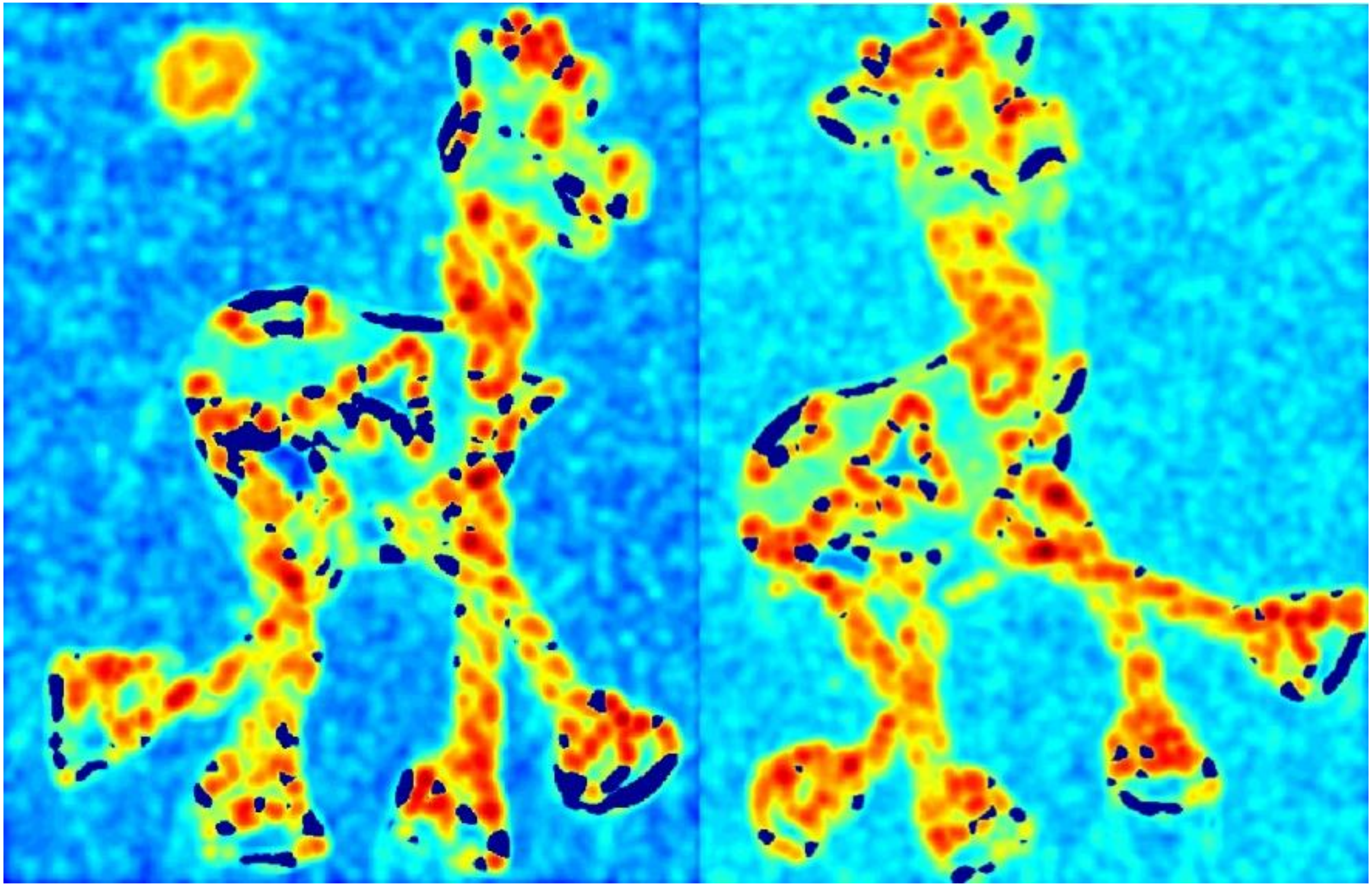
5. Perform non-maximum suppression

Slide credit: Krystian Mikolajczyk

Harris Detector: Workflow

Slide adapted from Darya Frolova, Denis Simakov

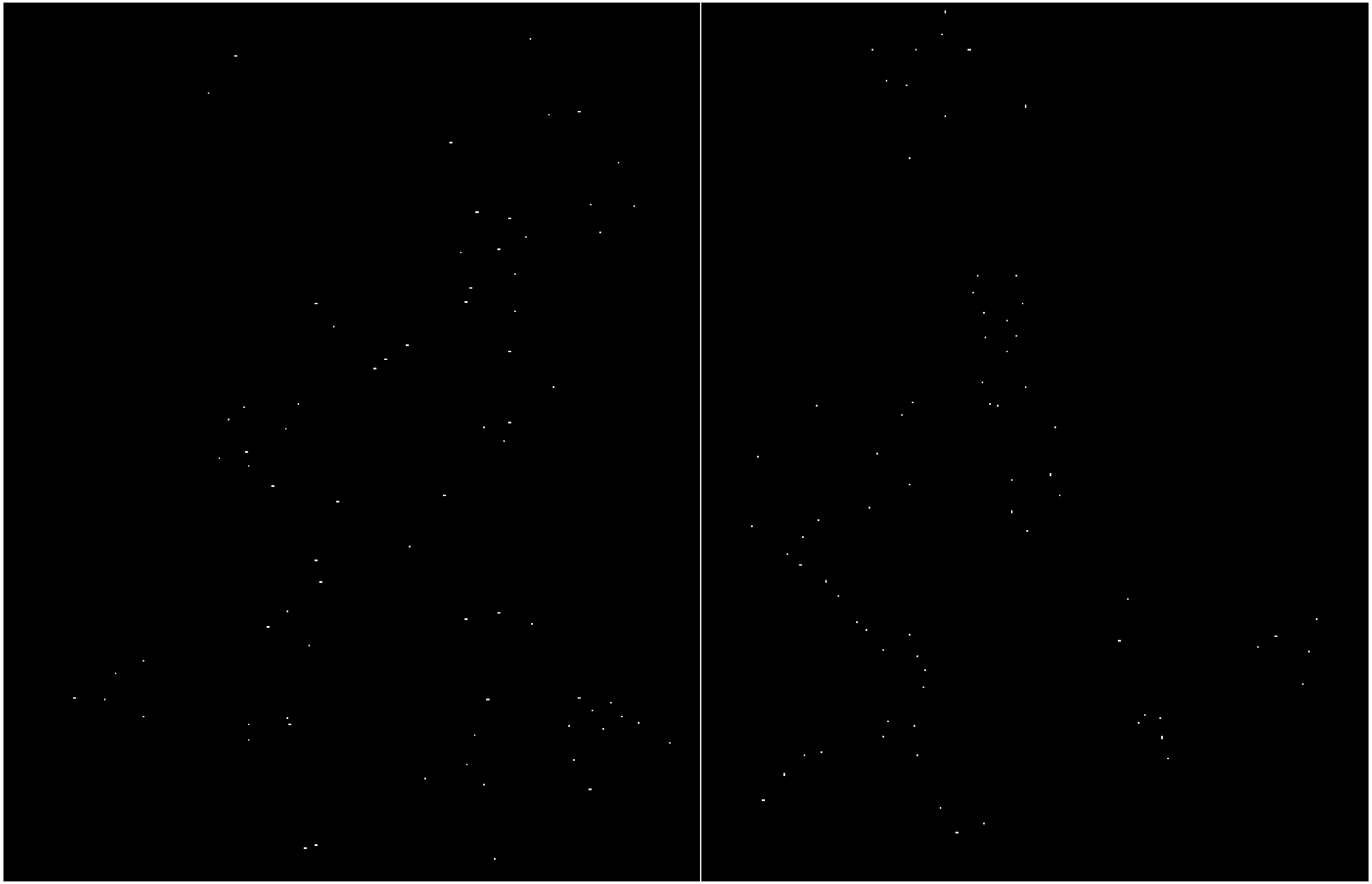
Harris Detector: Workflow
- computer corner responses θ



Slide adapted from Darya Frolova, Denis Simakov

Harris Detector: Workflow

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

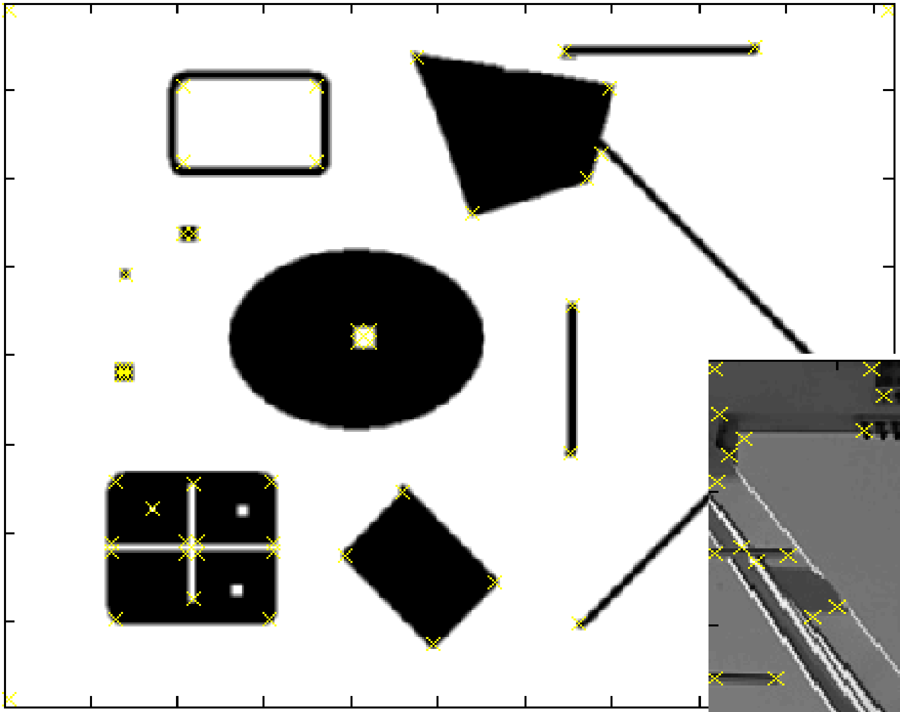


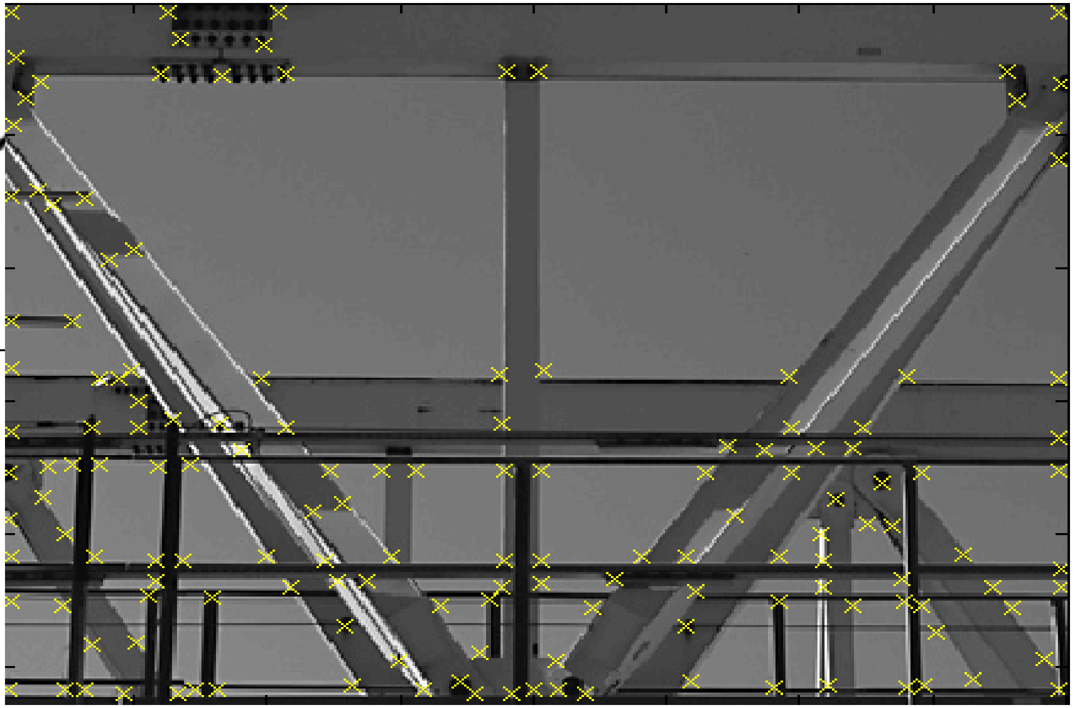
Slide adapted from Darya Frolova, Denis Simakov

Harris Detector: Workflow
- Resulting Harris points

Slide adapted from Darya Frolova, Denis Simakov

Harris Detector – Responses [Harris88]





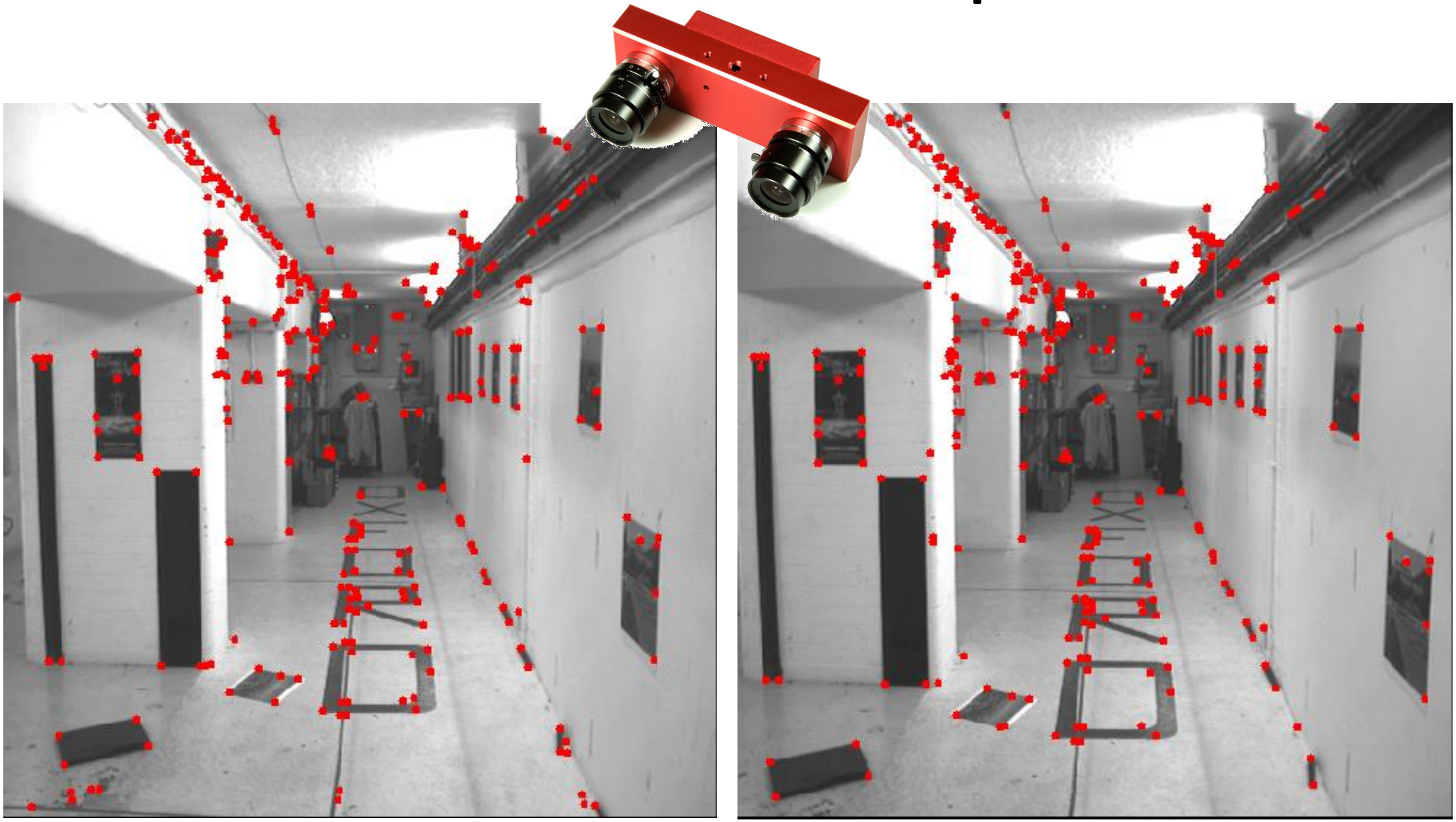
Effect: **A very precise corner detector.**

Slide credit: Krystian Mikolajczyk

Harris Detector – Responses [Harris88]

Slide credit: Krystian Mikolajczyk

Harris Detector – Responses [Harris88]



- Results are well suited for finding stereo correspondences

Slide credit: Kristen Grauman

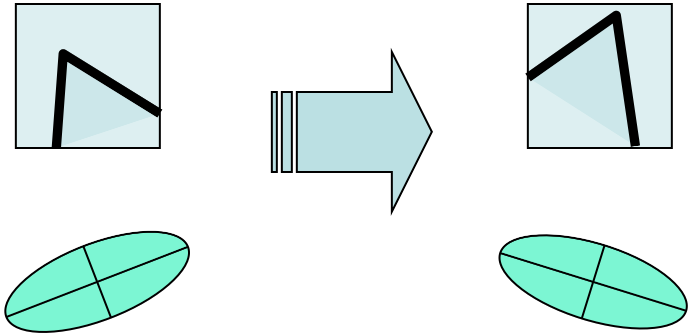
Harris Detector: Properties

- Translation invariance?

Slide credit: Kristen Grauman

Harris Detector: Properties

- Translation invariance
- Rotation invariance?



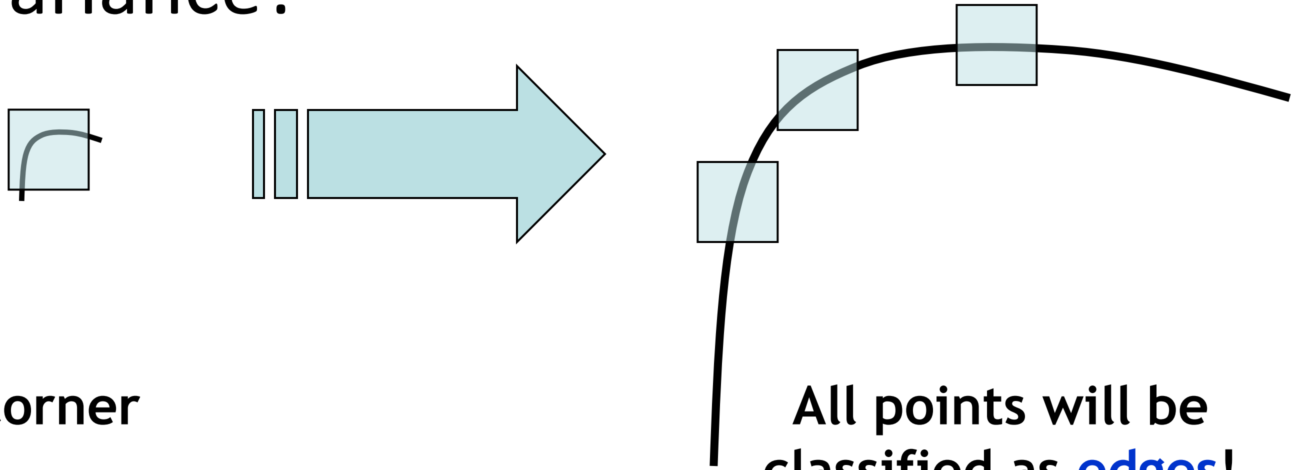
Ellipse rotates but its shape (i.e. eigenvalues) remains the same

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

Slide credit: Kristen Grauman

Harris Detector: Properties

- Translation invariance
- Rotation invariance
- Scale invariance?



Corner

All points will be classified as edges!

Not invariant to image scale!

Slide credit: Kristen Grauman

What we are learned today?

- A model fitting method for edge detection
  - RANSAC
- Local invariant features
  - Motivation
  - Requirements, invariances
- Keypoint localization
  - Harris corner detector