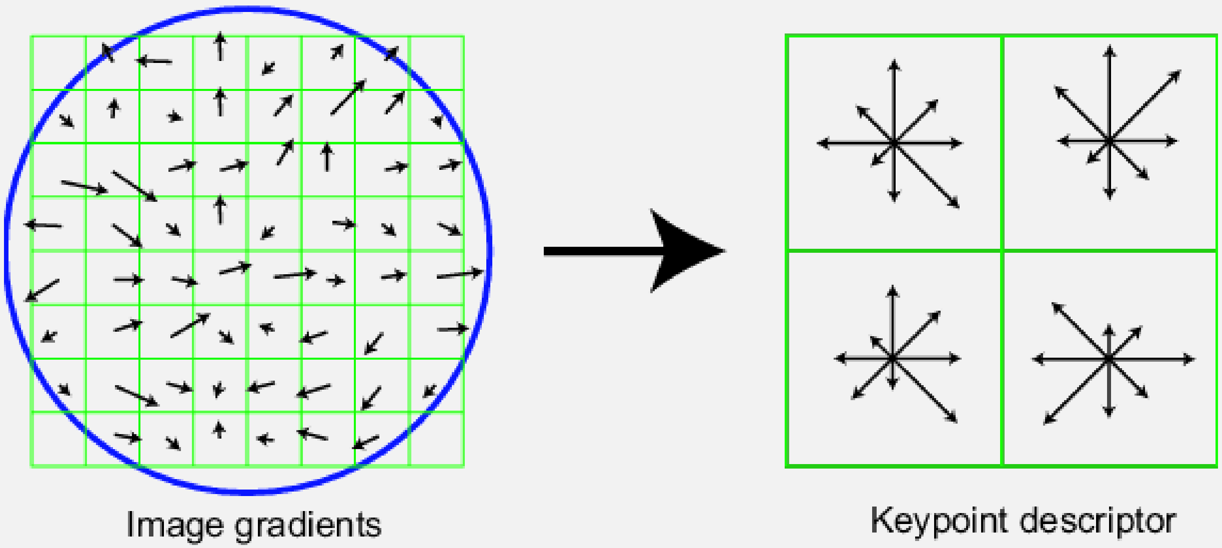


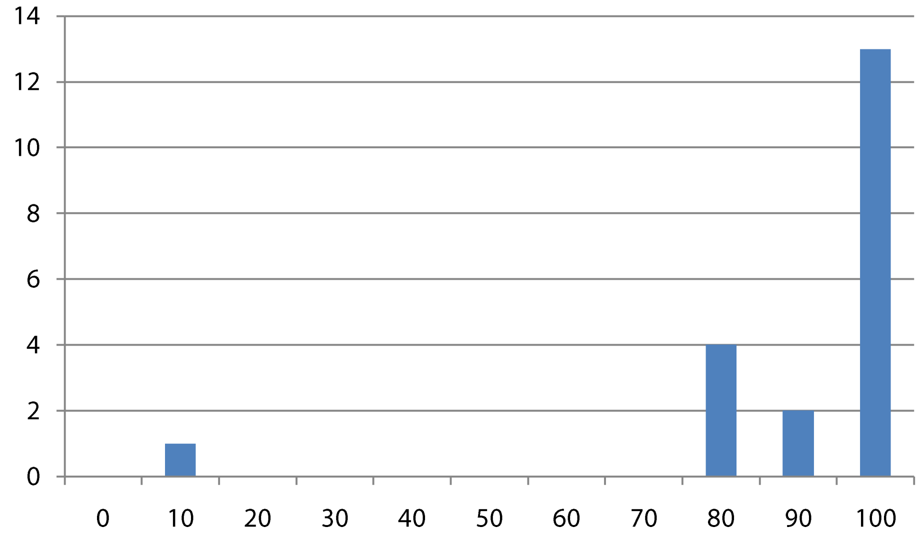
SIFT

CSE 455, Winter 2010

February 3, 2010

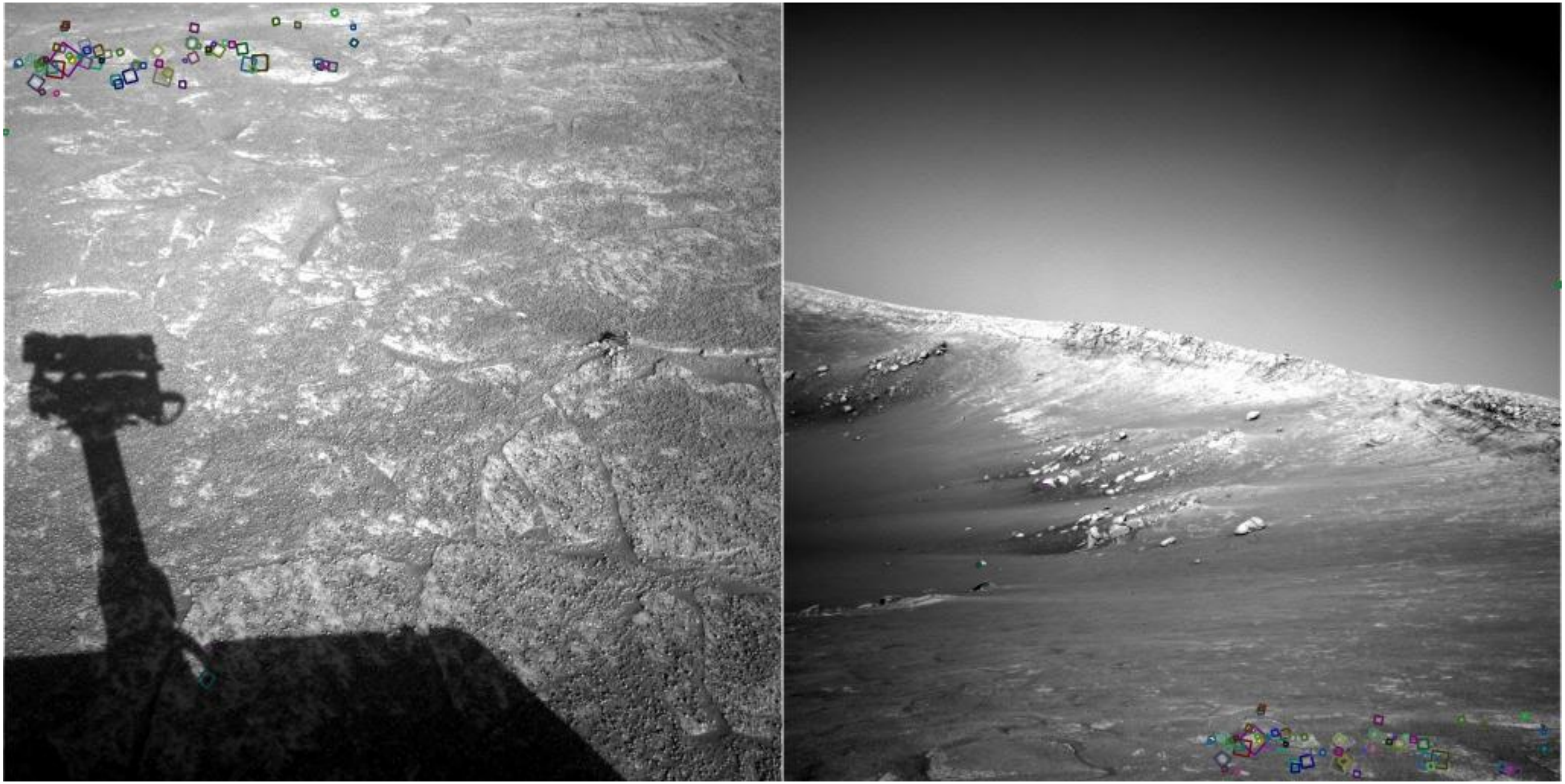
Announcements

- Project 1b grades out
- https://catalysttools.washington.edu/gradebook/iansimon/17677
- Statistics
  - Mean 88.95
  - Median 100
  - Std. Dev. 20.97



Review From Last Time

Image Matching



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

What makes a good feature?

Snoop demo

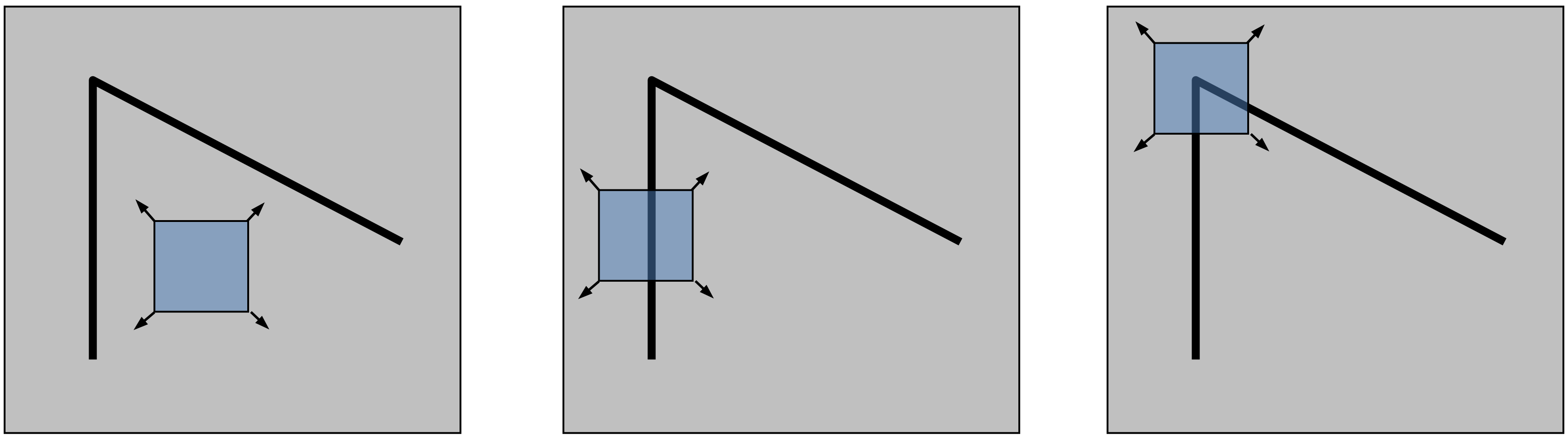
Want uniqueness

- Look for image regions that are unusual
  - Lead to unambiguous matches in other images

- How to define "unusual"?

Feature detection

- Local measure of feature uniqueness
  - How does the window change when you shift it?
  - Shifting the window in *any direction* causes a *big change*



"flat" region:
no change in all directions

"edge":
no change along the edge direction

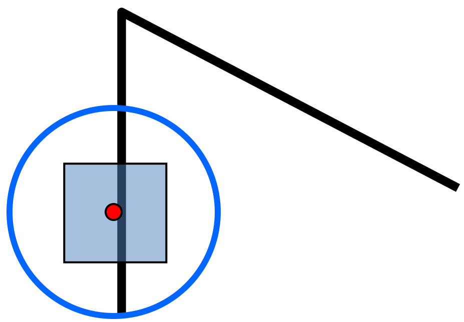
"corner":
significant change in all directions

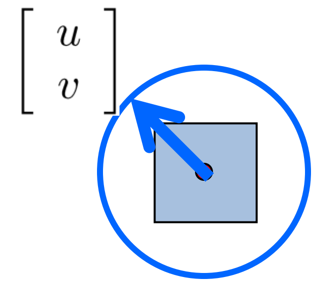
Slide adapted form Darya Frolova, Denis Simakov, Weizmann Institute

Feature detection: the math

This can be rewritten:

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





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?
- We can find these directions by looking at the eigenvectors of *H*

Today

- Invariant Features

Readings

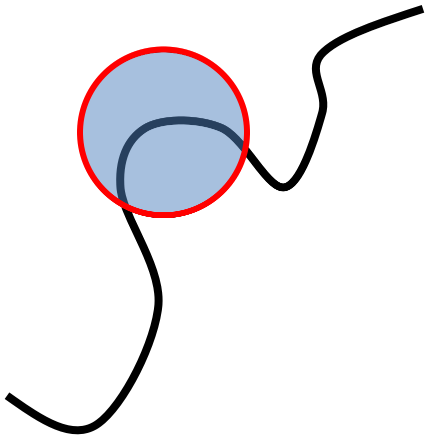
- M. Brown et al. Multi-Image Matching using Multi-Scale Oriented Patches, CVPR 2005

Invariance

- Suppose you **rotate** the image by some angle
  - Will you still pick up the same features?
- What if you change the brightness?
- Scale?

Scale invariant detection

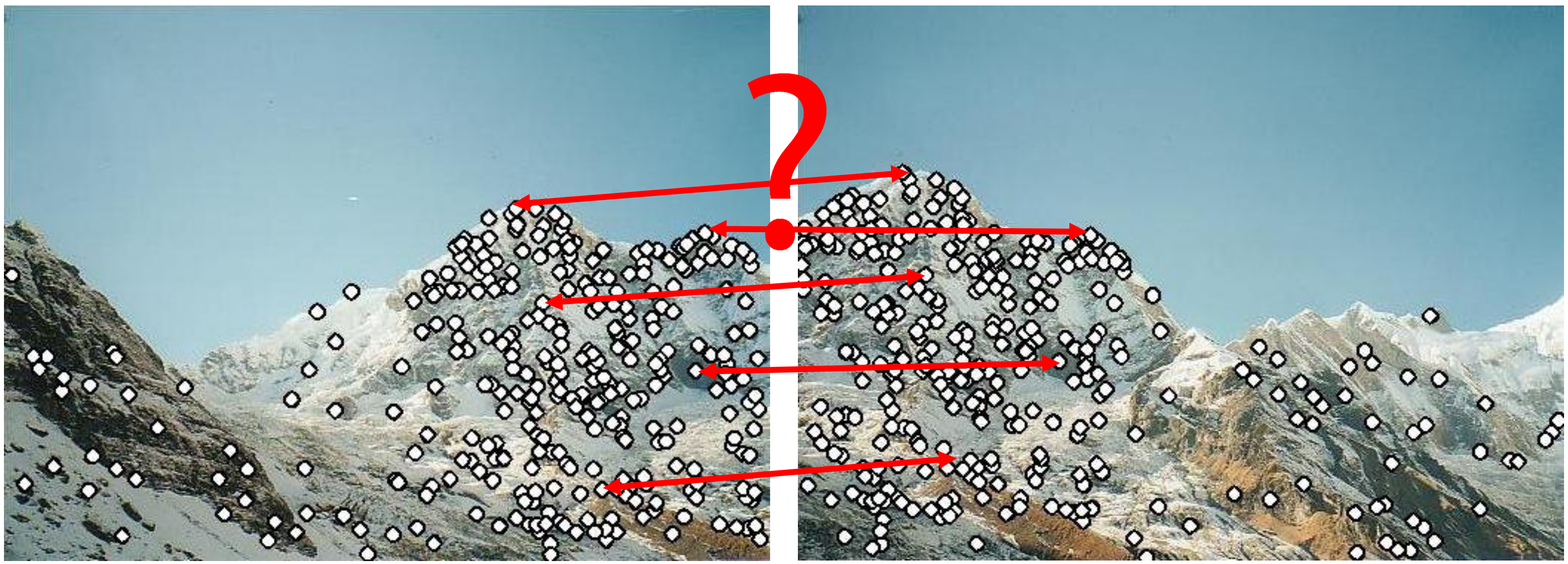
- Suppose you’re looking for corners



- Key idea: find scale that gives local maximum of f
  - f is a local maximum in both position and scale

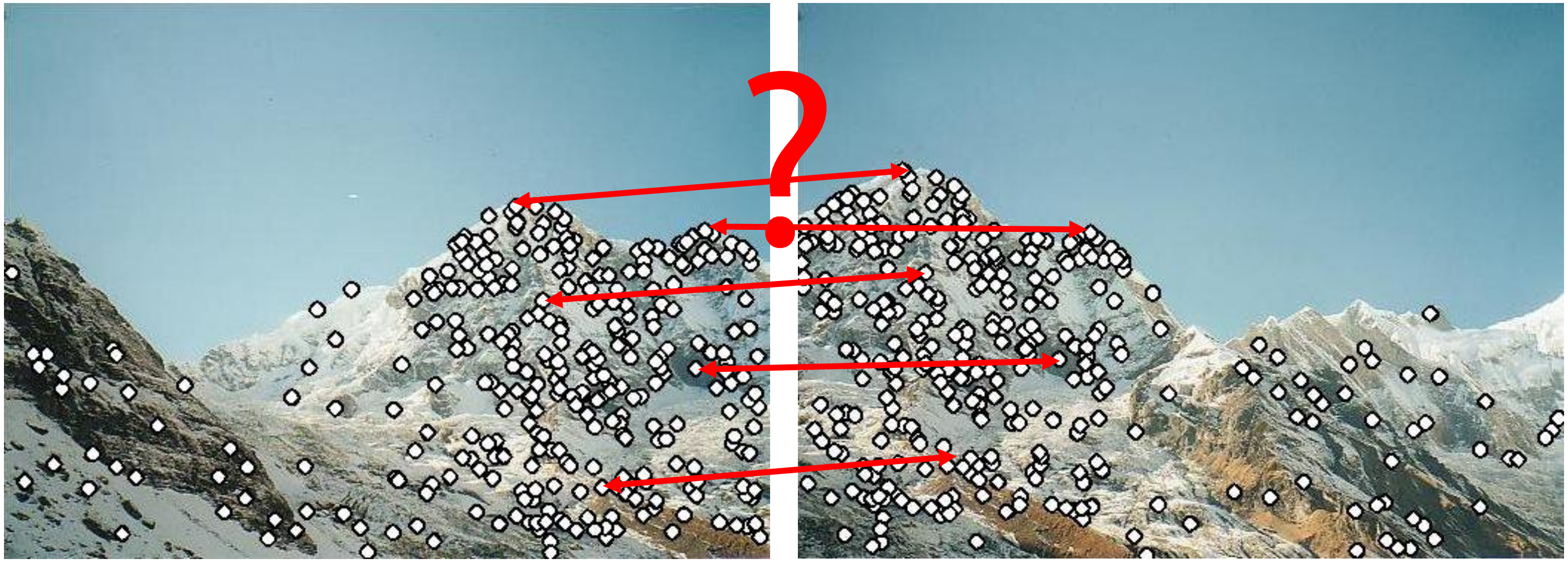
Feature descriptors

- We know how to detect good points
- Next question: **How to match them?**



Feature descriptors

- We know how to detect good points
- Next question: **How to match them?**



- Lots of possibilities (this is a popular research area)
  - Simple option: match square windows around the point
  - State of the art approach: SIFT
    - David Lowe, UBC http://www.cs.ubc.ca/~lowe/keypoints/

Invariance

- Suppose we are comparing two images I_1 and I_2
  - I_2 may be a transformed version of I_1
  - What kinds of transformations are we likely to encounter in practice?

- We'd like to find the same features regardless of the transformation
  - This is called transformational ***invariance***
  - Most feature methods are designed to be invariant to
    - Translation, 2D rotation, scale
  - They can usually also handle
    - Limited 3D rotations (SIFT works up to about 60 degrees)
    - Limited affine transformations (some are fully affine invariant)

How to achieve invariance

- Need both of the following:

1. Make sure your detector is invariant
   - Harris is invariant to translation and rotation
   - Scale is trickier
     - common approach is to detect features at many scales using a Gaussian pyramid (e.g., MOPS)
     - More sophisticated methods find “the best scale” to represent each feature (e.g., SIFT)

- 2. Design an invariant feature *descriptor*
  - A descriptor captures the information in a region around the detected feature point
  - The simplest descriptor: a square window of pixels
    - What’s this invariant to?
  - Let’s look at some better approaches…

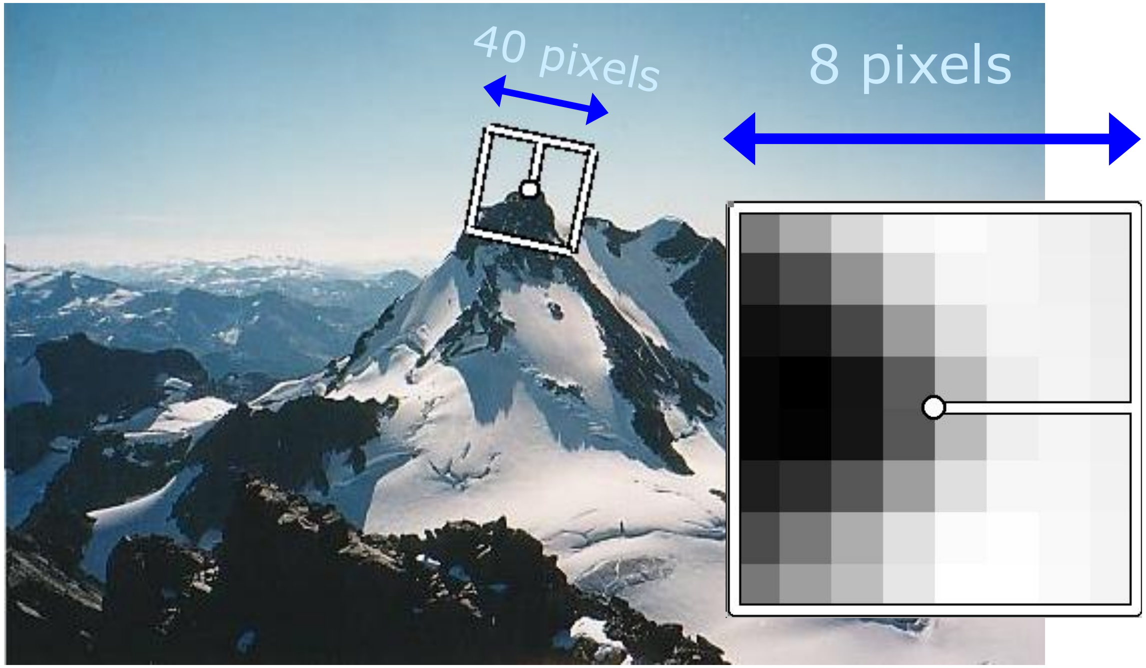
Rotation invariance for feature descriptors

- Find dominant orientation of the image patch
  - This is given by $\mathbf{x}_+$, the eigenvector of $\mathbf{H}$ corresponding to λ_+
    - λ_+ is the *larger* eigenvalue
  - Rotate the patch according to this angle

Figure by Matthew Brown

Multiscale Oriented PatcheS descriptor

- Take 40x40 square window around detected feature
  - Scale to 1/5 size (using prefiltering)
  - Rotate to horizontal
  - Sample 8x8 square window centered at feature
  - Intensity normalize the window by subtracting the mean, dividing by the standard deviation in the window



Adapted from slide by Matthew Brown

Detections at multiple scales

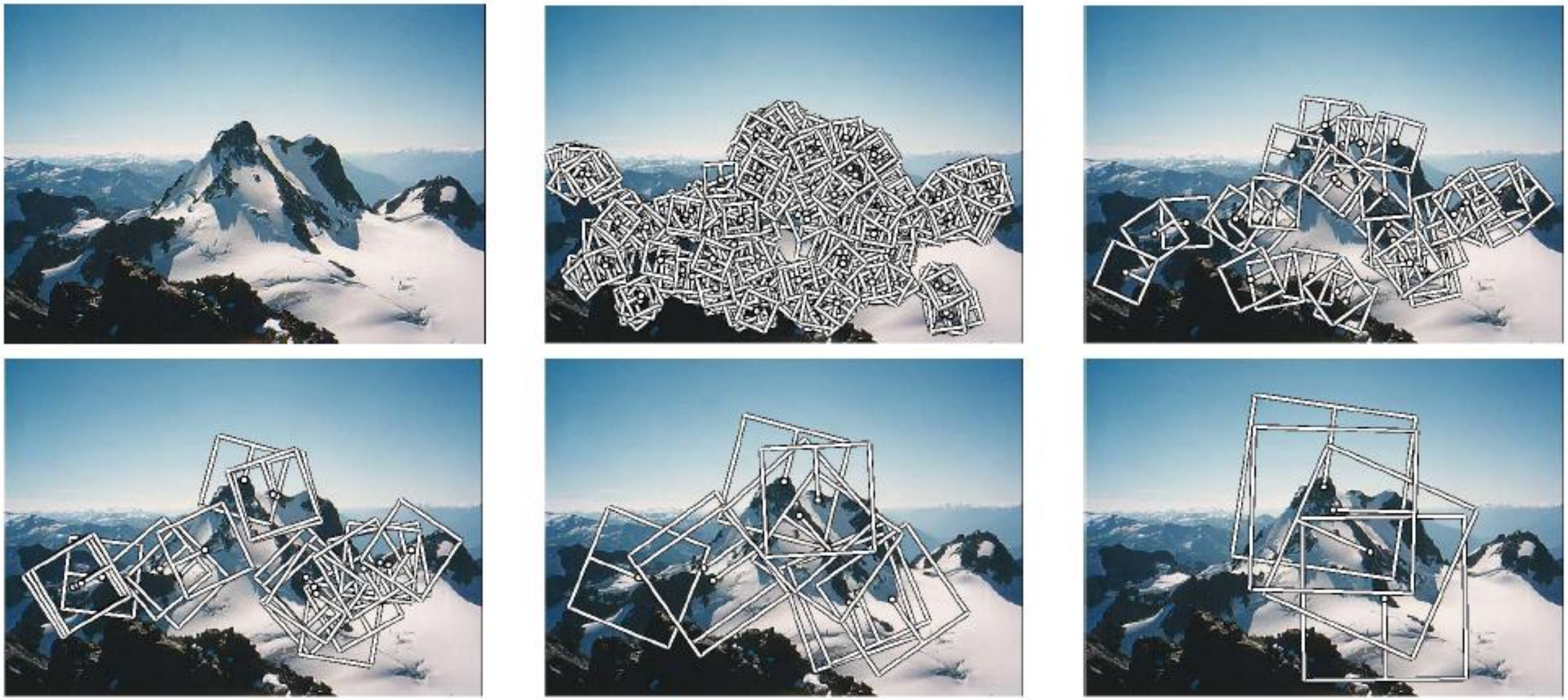
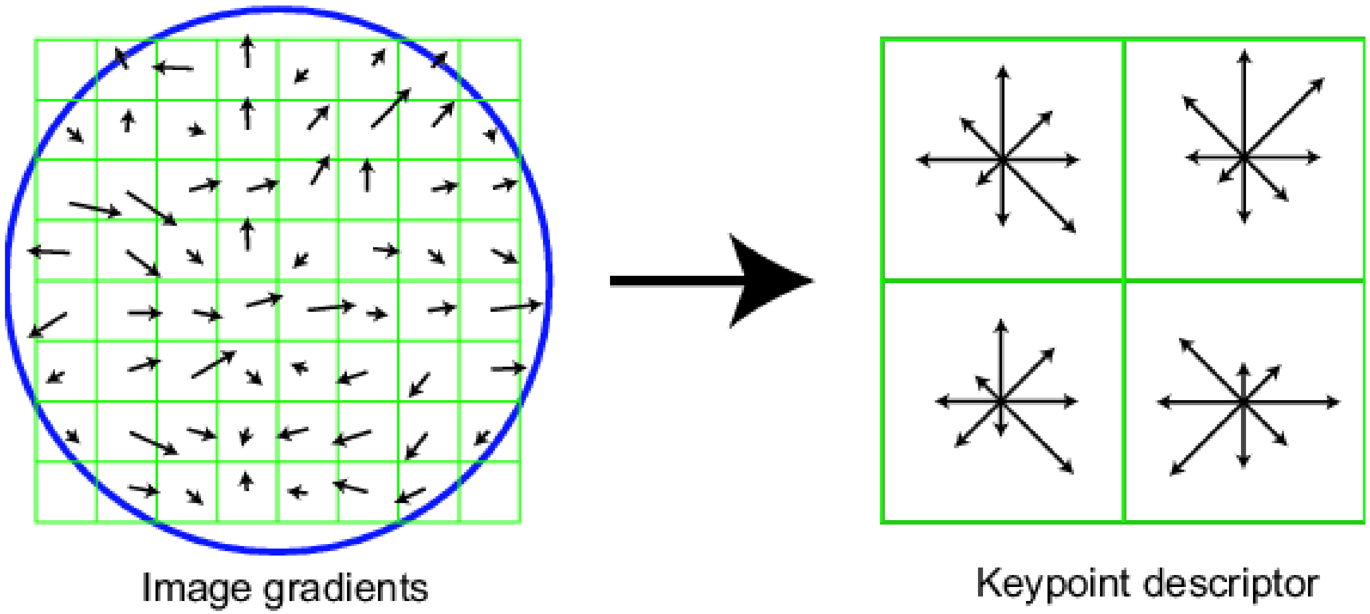


Figure 1. Multi-scale Oriented Patches (MOPS) extracted at five pyramid levels from one of the Matier images. The boxes show the feature orientation and the region from which the descriptor vector is sampled.

SIFT descriptor

Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor

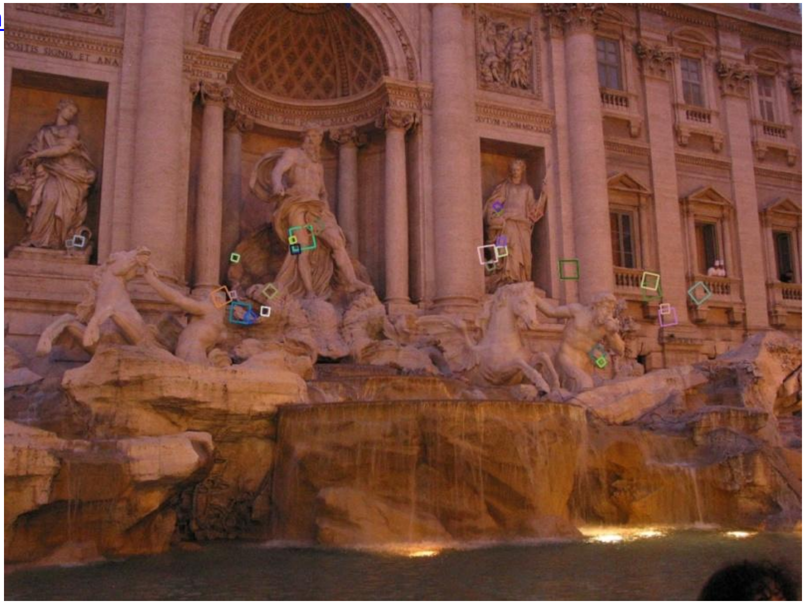


Adapted from slide by David Lowe

Properties of SIFT

- Extraordinarily robust matching technique
  - Can handle changes in viewpoint
    - Up to about 60 degree out of plane rotation
  - Can handle significant changes in illumination
    - Sometimes even day vs. night (below)
  - Fast and efficient—can run in real time
  - Lots of code available

n

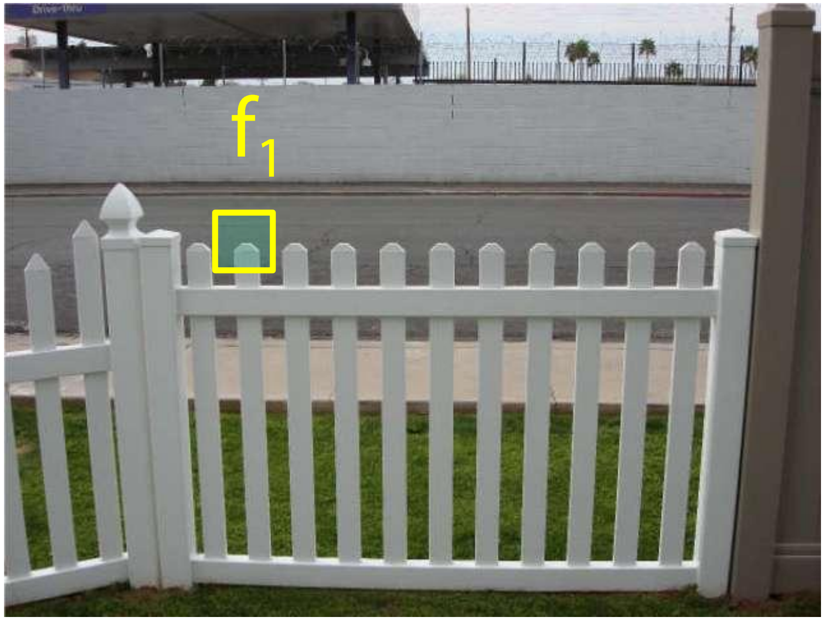


Feature matching

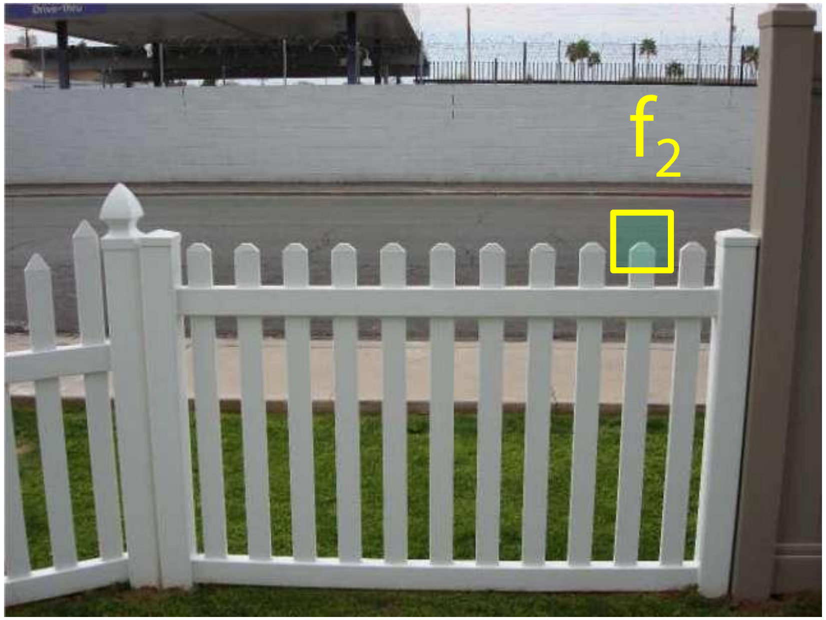
- Given a feature in I_1, how to find the best match in I_2?
  1. Define distance function that compares two descriptors
  2. Test all the features in I_2, find the one with min distance

Feature distance

- How to define the difference between two features f_1, f_2?
  - Simple approach is SSD(f_1, f_2)
    - sum of square differences between entries of the two descriptors
    - can give good scores to very ambiguous (bad) matches



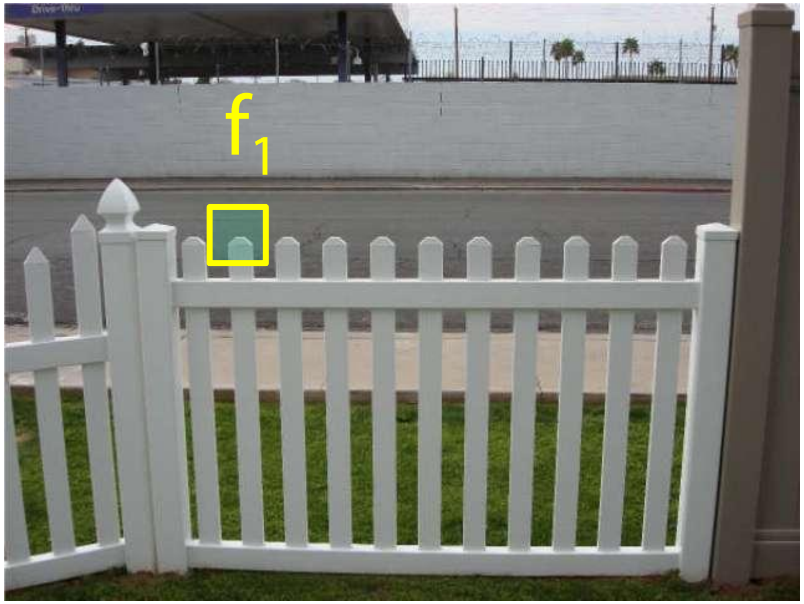
I_1



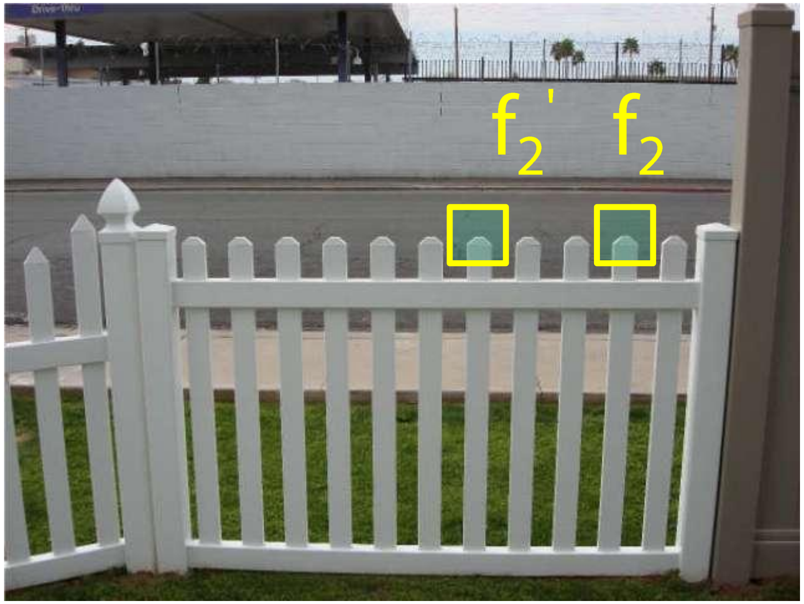
I_2

Feature distance

- How to define the difference between two features f_1, f_2?
  - Better approach: ratio distance = $SSD(f_1, f_2) / SSD(f_1, f_2')$
    - f_2 is best SSD match to f_1 in I_2
    - f_2' is 2^{nd} best SSD match to f_1 in I_2
    - gives small values for ambiguous matches



I_1

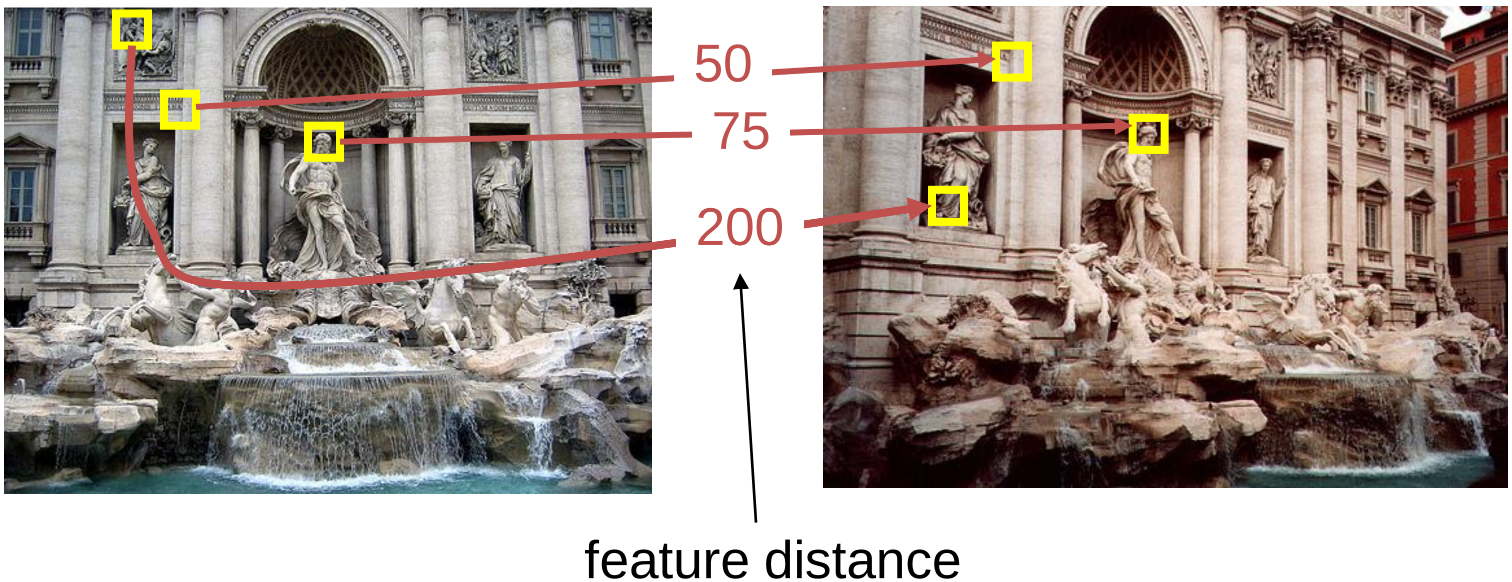


I_2

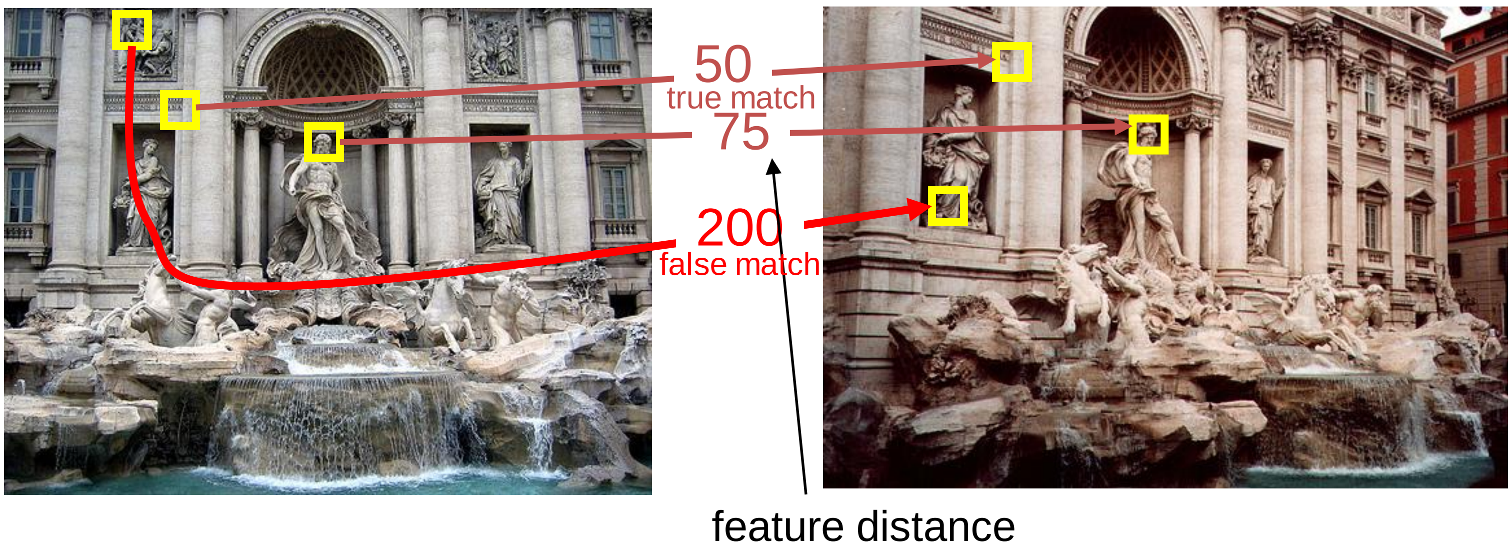
Neel Joshi, CSE 455, Winter 2010

Evaluating the results

- How can we measure the performance of a feature matcher?



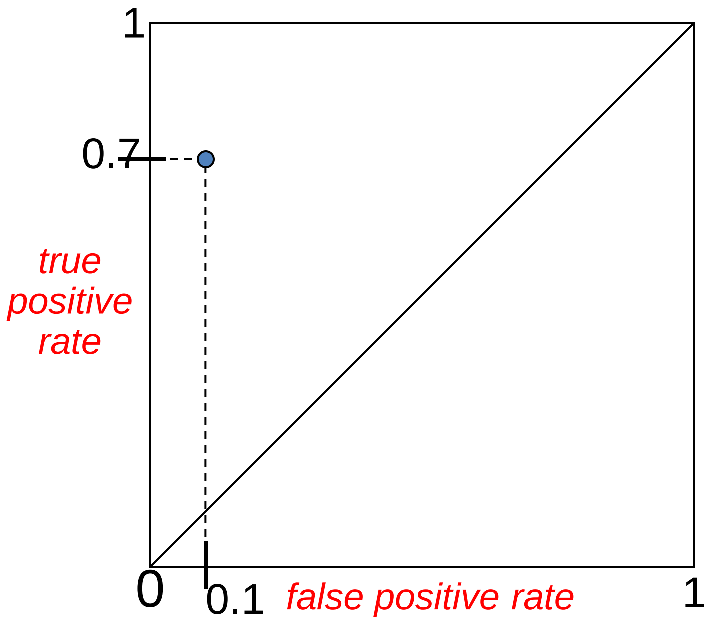
True/false positives



- The distance threshold affects performance
  - True positives = # of detected matches that are correct
    - Suppose we want to maximize these—how to choose threshold?
  - False positives = # of detected matches that are incorrect
    - Suppose we want to minimize these—how to choose threshold?

Evaluating the results

- How can we measure the performance of a feature matcher?



$$\text{true positive rate} = \frac{\text{\# true positives}}{\text{\# matching features (positives)}}$$

$$\text{false positive rate} = \frac{\text{\# false positives}}{\text{\# unmatched features (negatives)}}$$

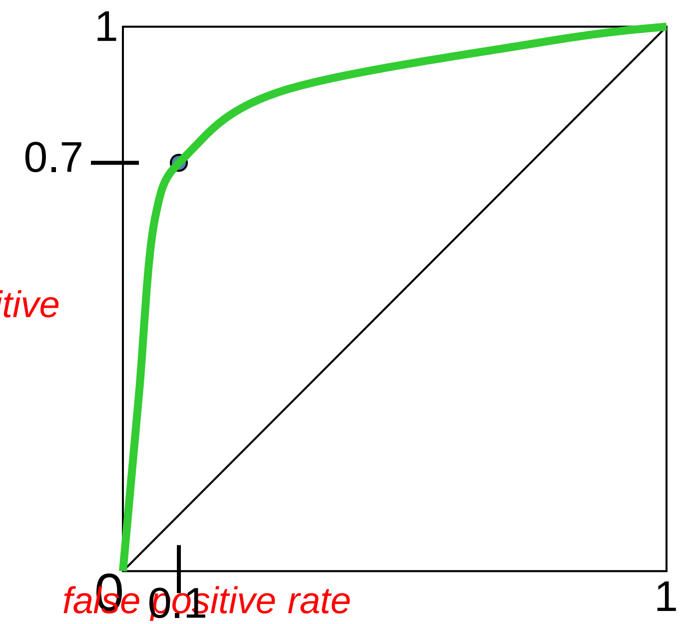
Evaluating the results

- How can we measure the performance of a feature matcher?

ROC curve ("Receiver Operator Characteristic")

$$\frac{\#\text{ true positives}}{\#\text{ matching features (positives)}}$$

true positive rate

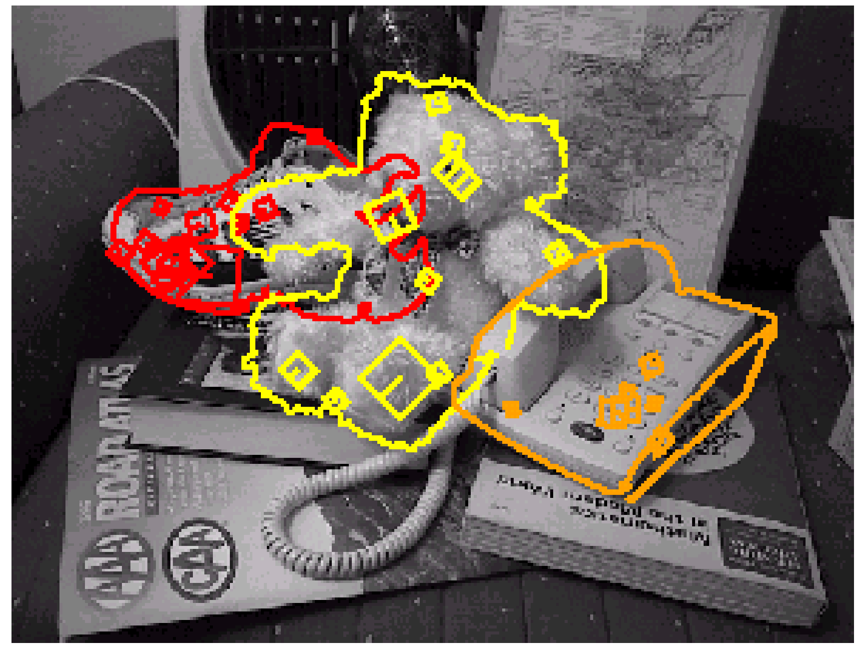


$$\frac{\#\text{ false positives}}{\#\text{ unmatched features (negatives)}}$$

ROC Curves

- Generated by counting # current/incorrect matches, for different threholds
- Want to maximize area under the curve
- Useful for comparing different feature matching methods
- For more info: http://en.wikipedia.org/wiki/Receiver_operating_characteristic

Object recognition (David Lowe)



Aibo

Sony Aibo

SIFT usage:

- Recognize charging station
- Communicate with visual cards
- Teach object recognition

Demos

- Gigapixel Panoramas
- PhotoTourism

Midterm

- Topics:
  - Filtering
  - Edge + Corner Detection
  - Resampling
  - Interpolation
  - Image Gradients, Energies
  - Recognition, Eigenfaces
  - Camera, Projective Geometry
  - Single View Modeling
  - Features
  - SIFT

Midterm

- Conceptual Questions
  - E.G. Designing a filter, interpreting a filter
  - Types of transformations
- Mathematical Questions
  - Measuring distances
  - Lens properties
- Algorithm Design
- Stepping through an Algorithm