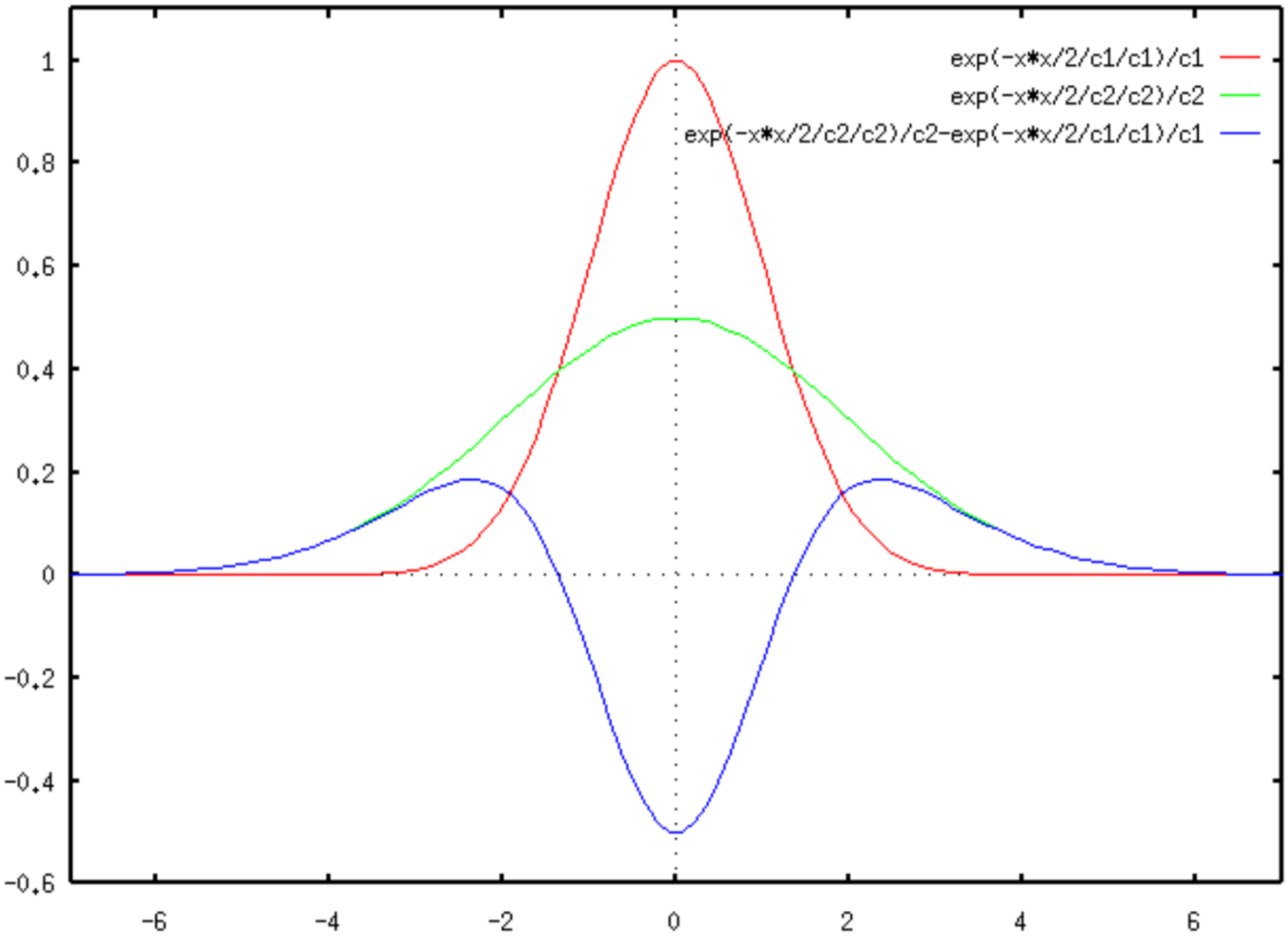
LoG-DoG (Difference of Gaussian)



LoG can be approximated by a difference of two Gaussians (DoG) at different scales

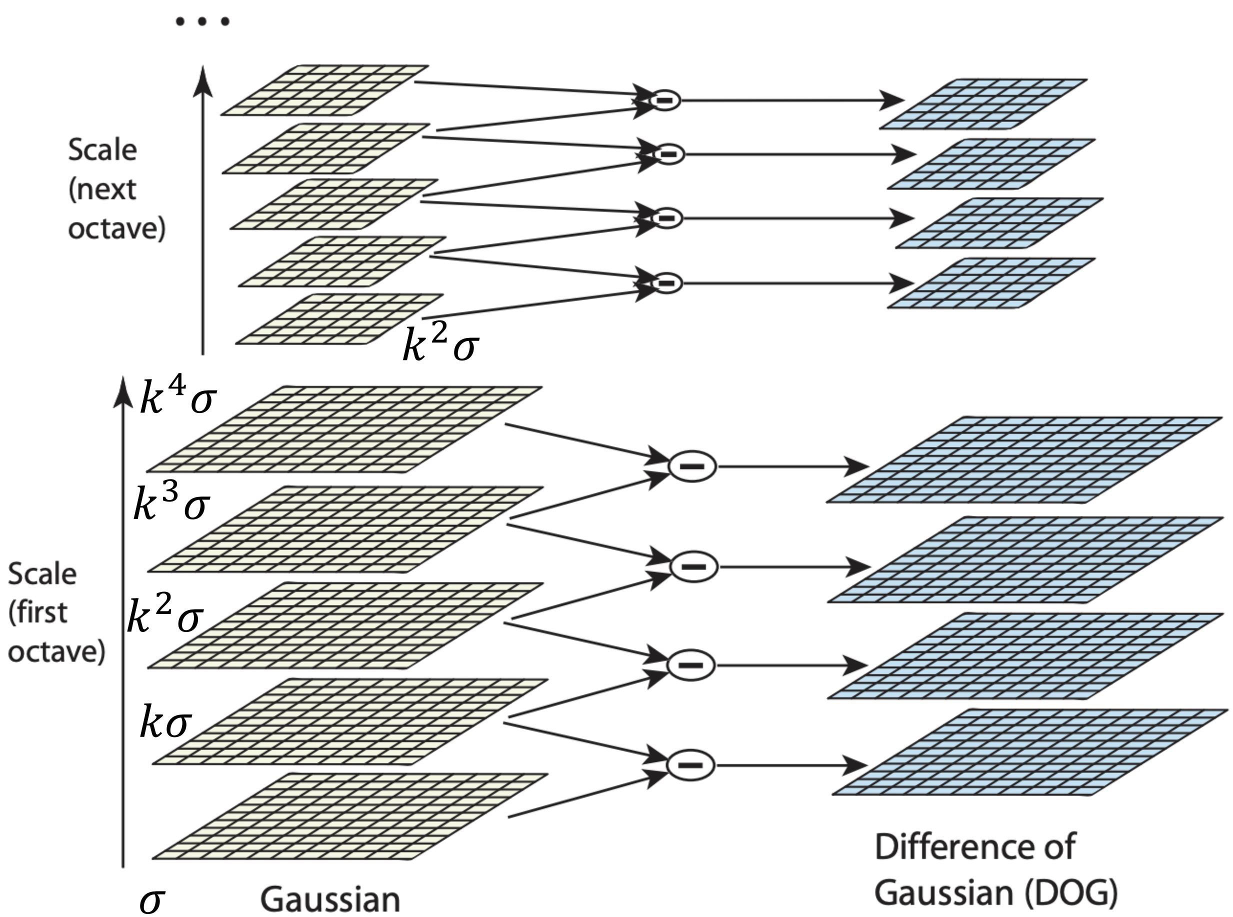
Adapted from Wiki

Scale Invariant Feature Transform

- Scale space peak selection
  - Potential locations for finding features
- Key point localization
  - Accurately locating the feature key points
- Orientation assignment
  - Assigning orientation to the key points
- Key point descriptor
  - Describing the key point as a high dimensional vector (128) (SIFT Descriptor)

Adapted from slide by Mubarak Shah

Building the Scale Space

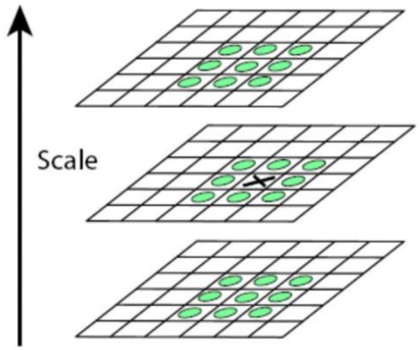


Adapted from IJCV 2004 by David Lowe

Peak Detection

Compare a pixel (**X**) with 26 pixels in current and adjacent scales (Green Circles)

Select a pixel (**X**) if larger/smaller than all 26 pixels



Adapted from slide by Mubarak Shah

Assignment of the Orientation

For the rotation invariance, compute the gradient magnitude and direction at the scale of the keypoint.

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y),$$

Adapted from IJCV 2004 by David Lowe

Assignment of the Orientation

- An orientation histogram is formed from the gradient orientations of sample points within a region around the keypoint.

- The orientation histogram has 36 bins

- The samples added to the histogram is weighted by the gradient magnitude.

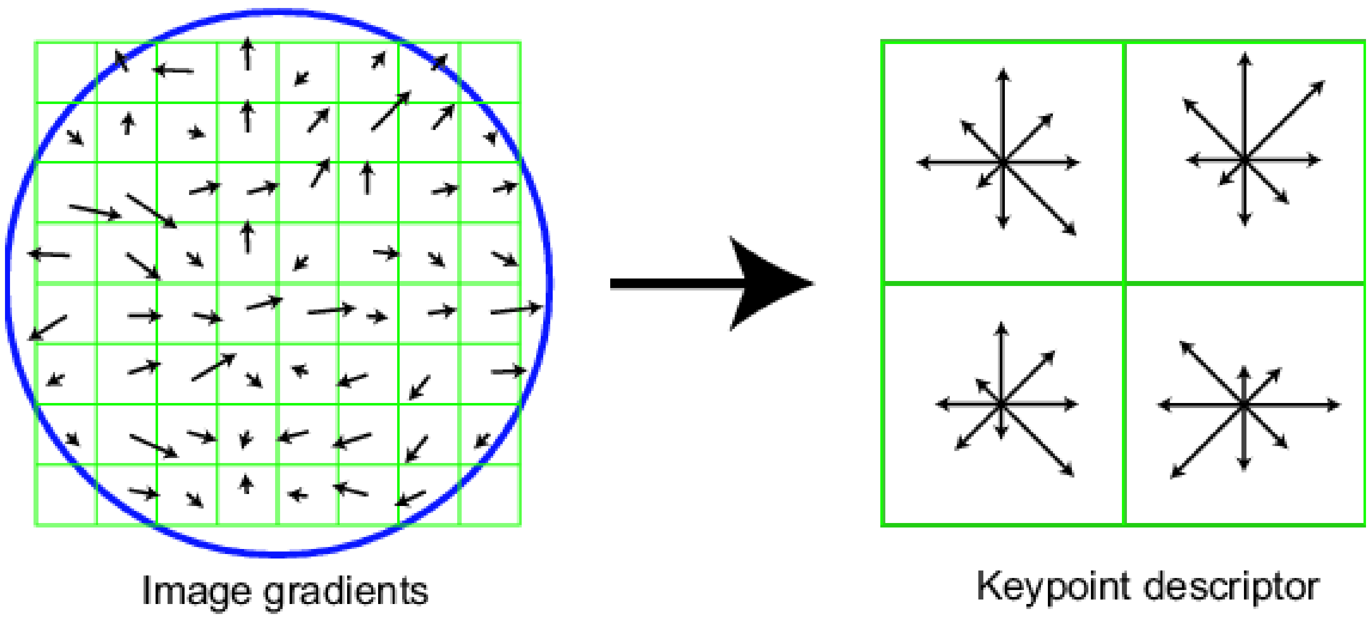
- The dominate direction is the peak in the histogram.

Adapted from IJCV 2004 by David Lowe

SIFT descriptor

Full version

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

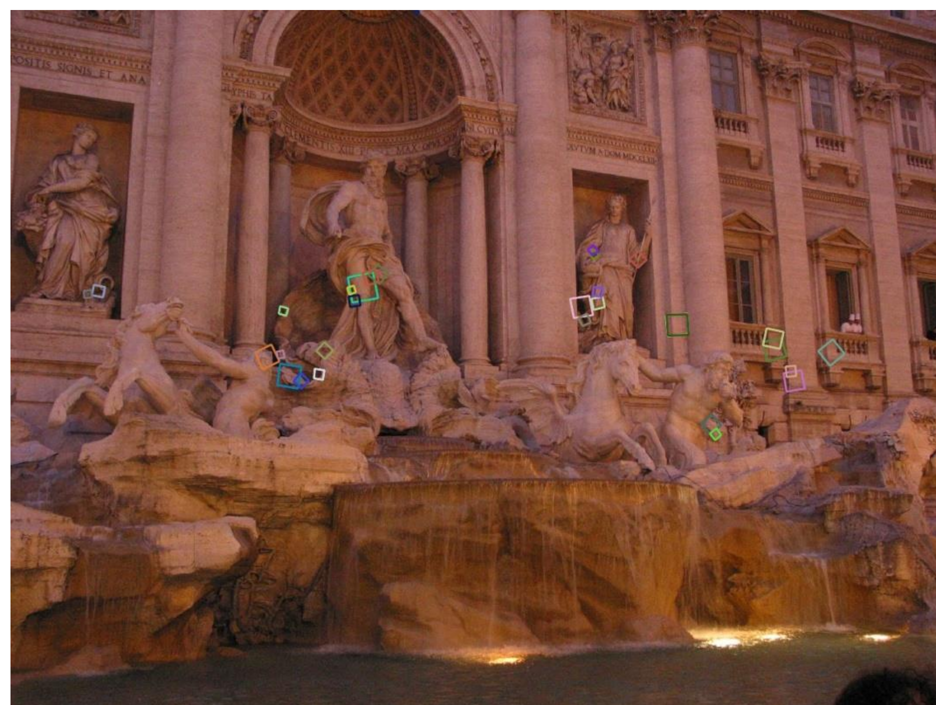


Adapted from slide by David Lowe

Properties of SIFT

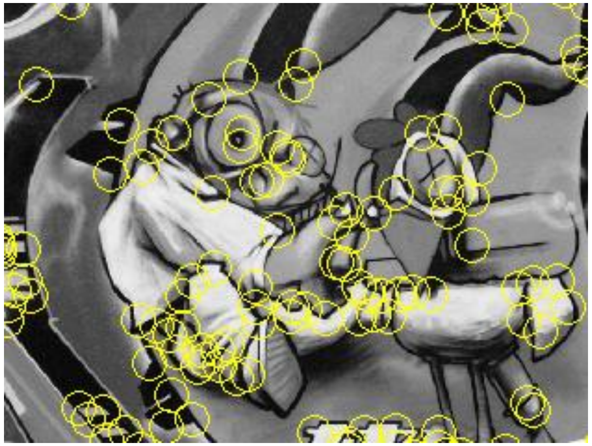
Robust matching technique

- Can handle changes in viewpoint
- Can handle significant changes in illumination
- Fast and efficient—can run in real time
- Lots of code available

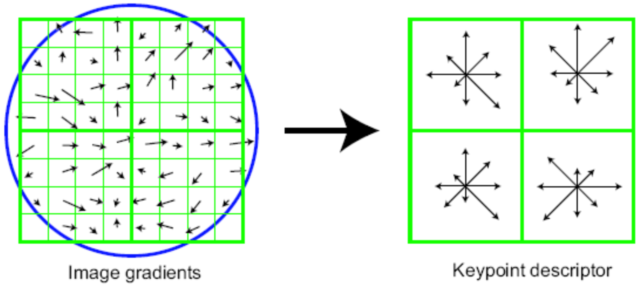


Summary

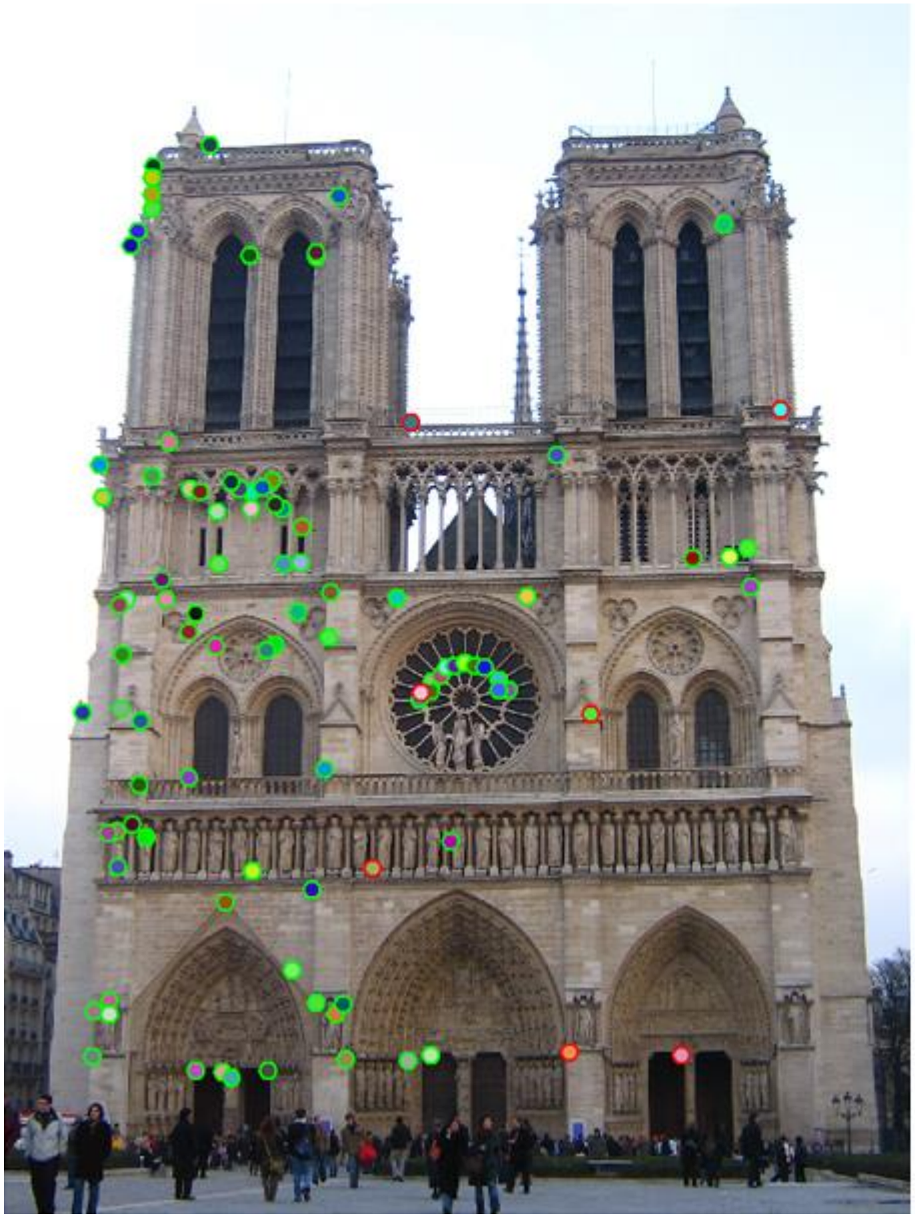
- Keypoint detection: repeatable and distinctive
  - Corners, blobs
  - Harris, LoG

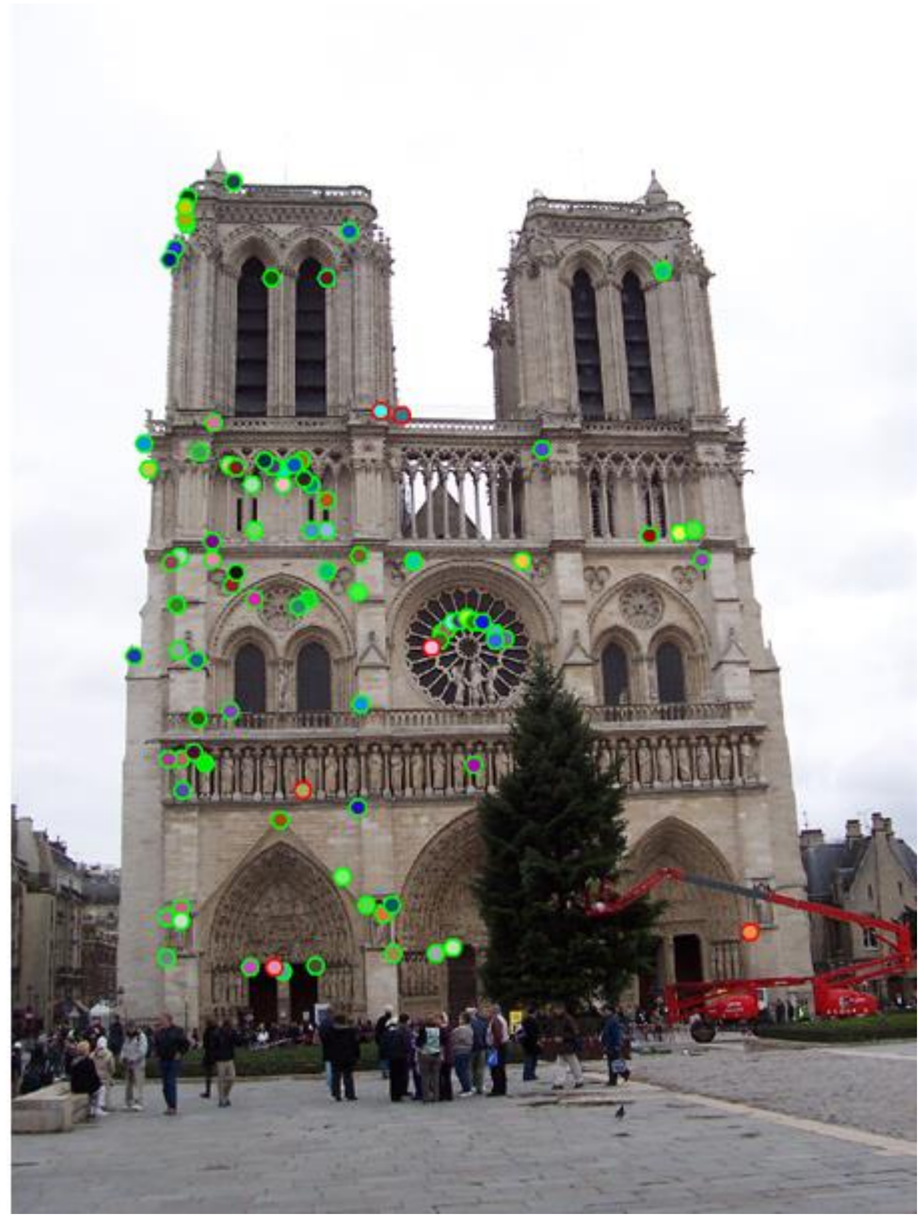


- Descriptors: robust
  - spatial histograms of orientation
  - SIFT and variants are typically good for stitching and recognition



Which features match?





Feature matching

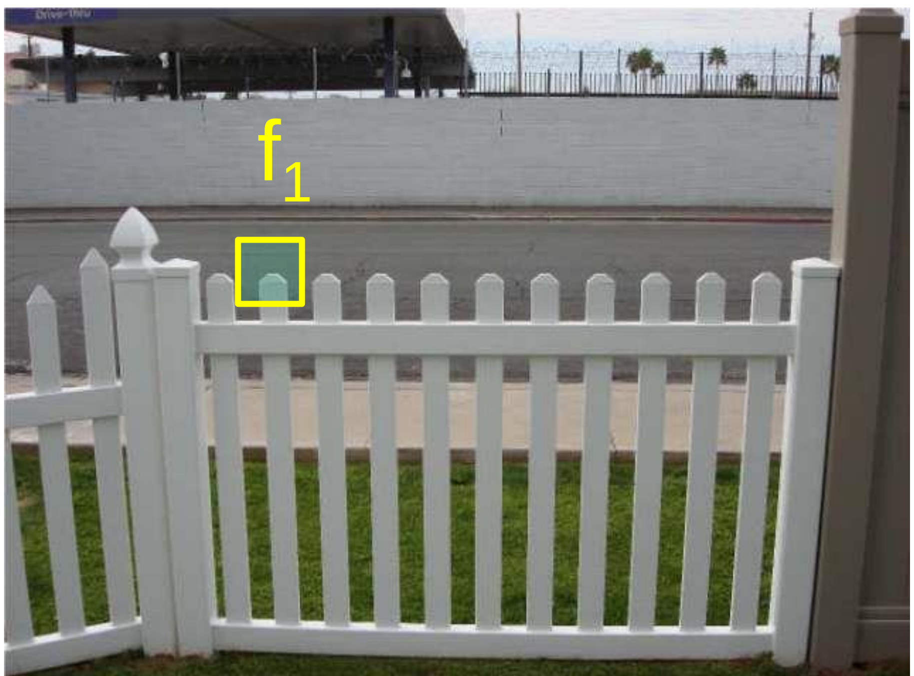
Given a feature in I_1, how to find the best match in I_2?

1. Define distance function that compares two descriptors
2. Test all the features in I_2, find the one with min distance

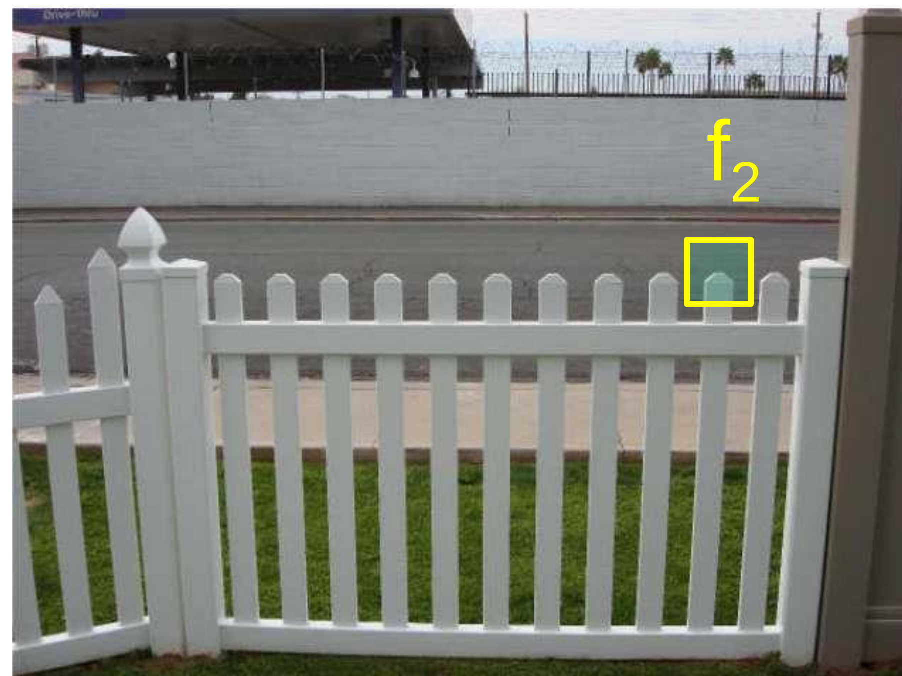
Feature distance

How to define the difference between two features f_1, f_2?

- Simple approach: L_2 distance, $||f_1 - f_2||$ (aka SSD)
- can give good scores to ambiguous (incorrect) matches



I_1

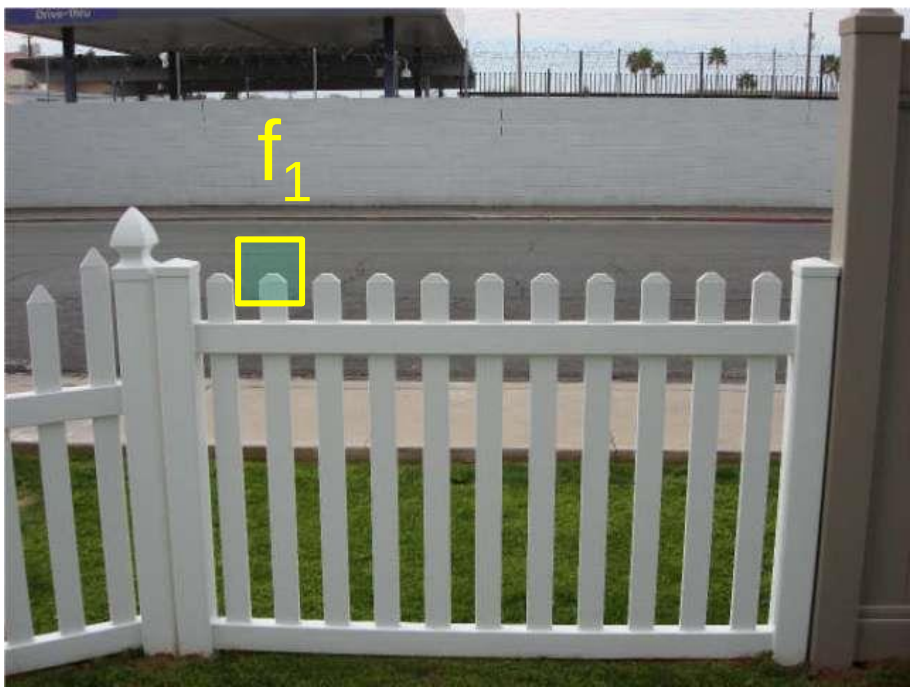


I_2

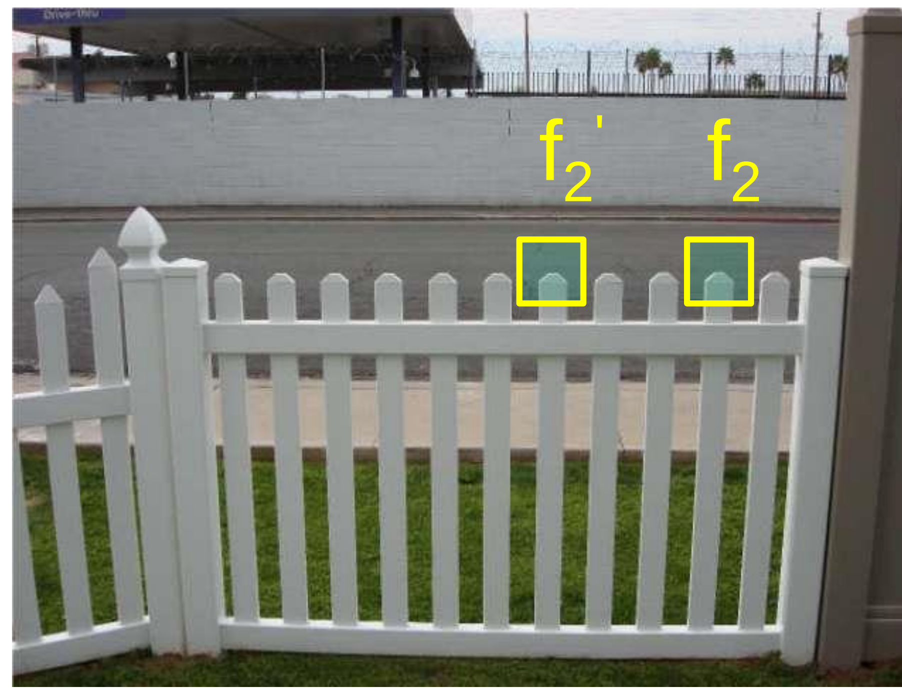
Feature distance

How to define the difference between two features f_1, f_2?

- Better approach: ratio distance = $||f_1 - f_2|| \, / \, || f_1 - f_2' ||$
  - 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 large values for ambiguous matches

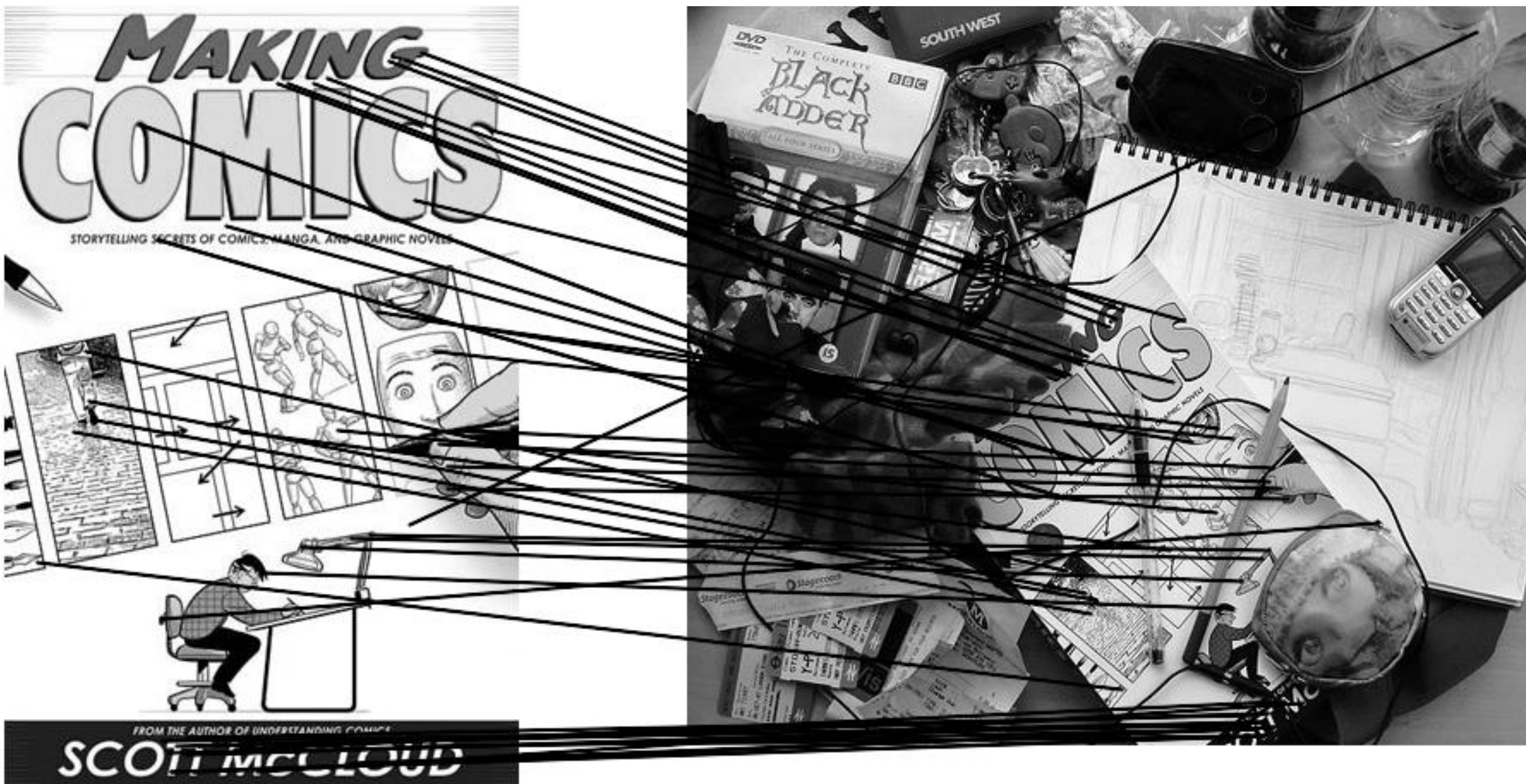


I_1



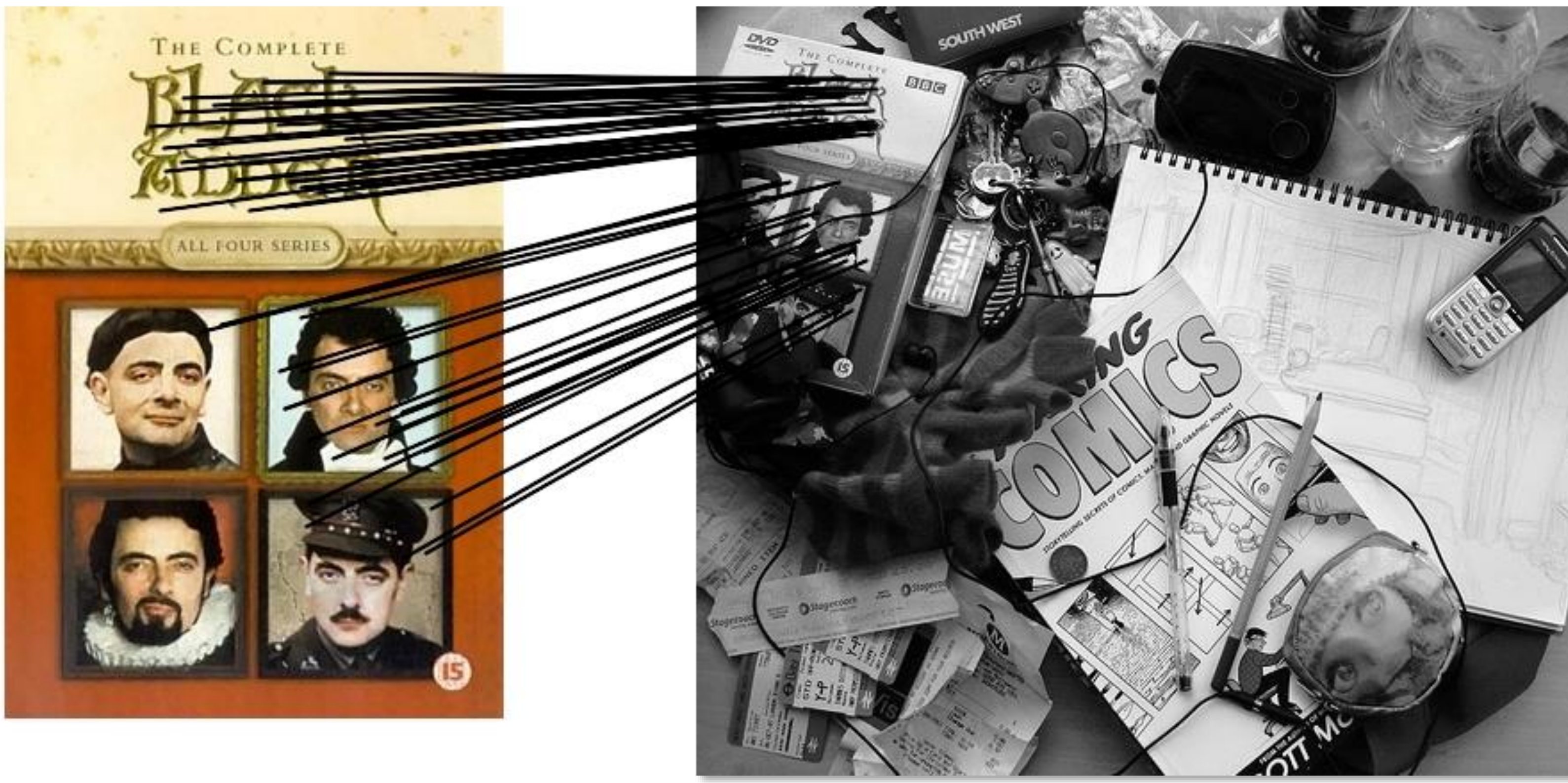
I_2

Feature matching example



51 matches (thresholded by ratio score)

Feature matching example



58 matches (thresholded by ratio score)