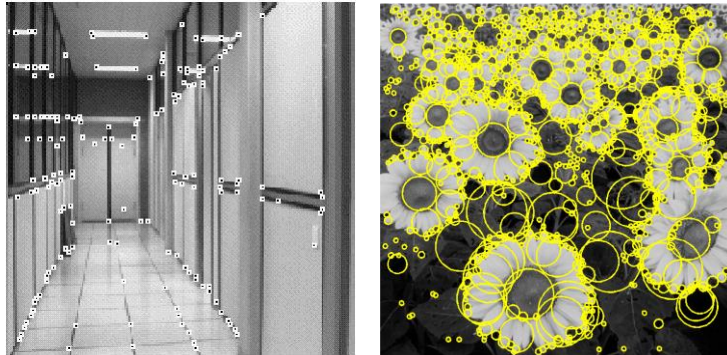


# CS4670/5670: Computer Vision

Noah Snavely

## Lecture 7: Blob detection and SIFT

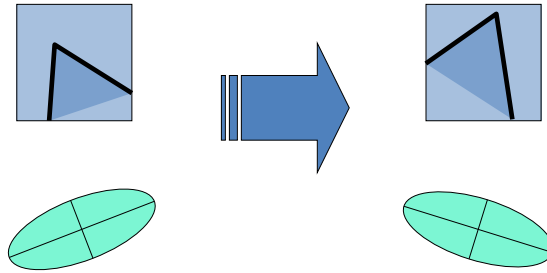


## Reading

- Szeliski: 4.1

## Harris Detector: Invariance Properties

- Rotation



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

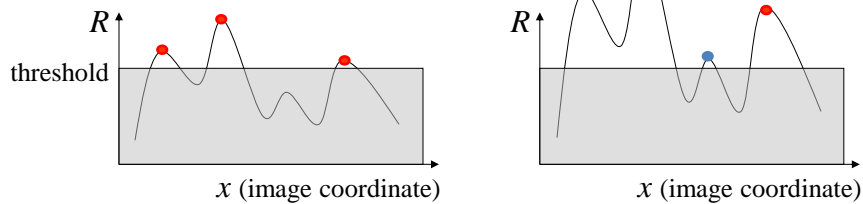
Corner response is invariant to image rotation

## Harris Detector: Invariance Properties

- Affine intensity change:  $I \rightarrow aI + b$

✓ Only derivatives are used =>  
invariance to intensity shift  $I \rightarrow I + b$

✓ Intensity scale:  $I \rightarrow aI$



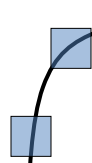
Partially invariant to affine intensity change

## Harris Detector: Invariance Properties

- Scaling

Corner



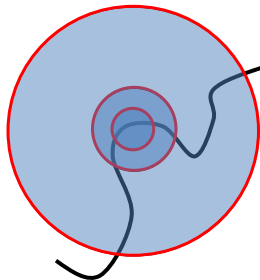


All points will be classified as *edges*

*Not invariant to scaling*

## Scale invariant detection

Suppose you're looking for corners

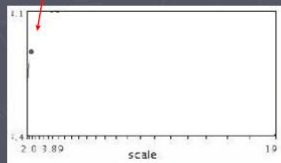


Key idea: find scale that gives local maximum of  $f$

- in both position and scale
- One definition of  $f$ : the Harris operator

# Automatic scale selection

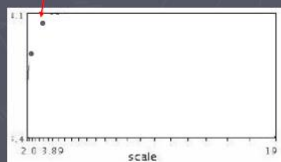
Lindeberg et al., 1996



$f(U_{k+1}(x, \sigma))$

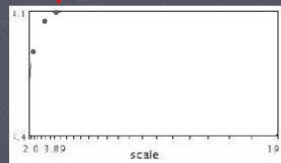
Slide from Tinne Tuytelaars

# Automatic scale selection



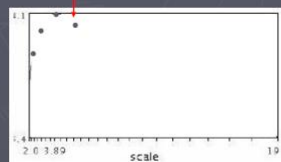
$f(U_{k+1}(x, \sigma))$

## Automatic scale selection



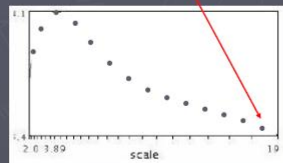
$$f(U_{k,m}(x, \sigma))$$

## Automatic scale selection



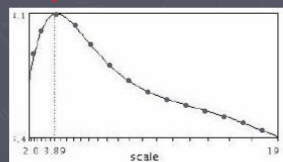
$$f(U_{k,m}(x, \sigma))$$

## Automatic scale selection



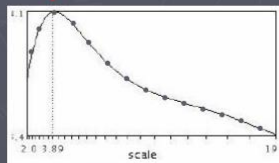
$f(U_{k,m}(x, \sigma))$

## Automatic scale selection

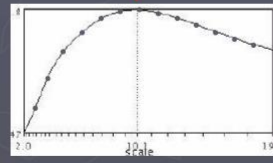


$f(U_{k,m}(x, \sigma))$

## Automatic scale selection



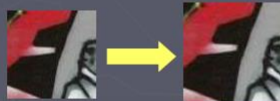
$$f(I_{h...m}(x, \sigma))$$



$$f(I_{h...m}(x', \sigma'))$$

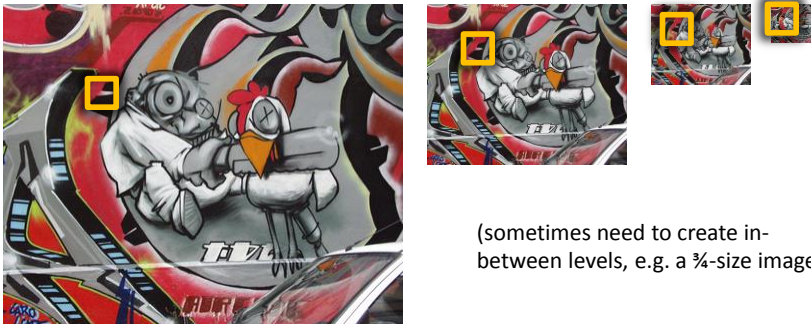
## Automatic scale selection

Normalize: rescale to fixed size



## Implementation

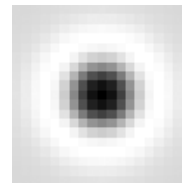
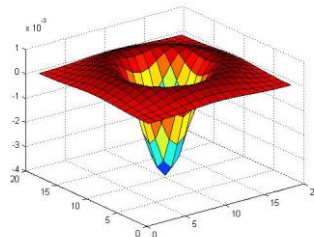
- Instead of computing  $f$  for larger and larger windows, we can implement using a fixed window size with a Gaussian pyramid



(sometimes need to create in-between levels, e.g. a  $\frac{3}{4}$ -size image)

## Another common definition of $f$

- The *Laplacian of Gaussian (LoG)*



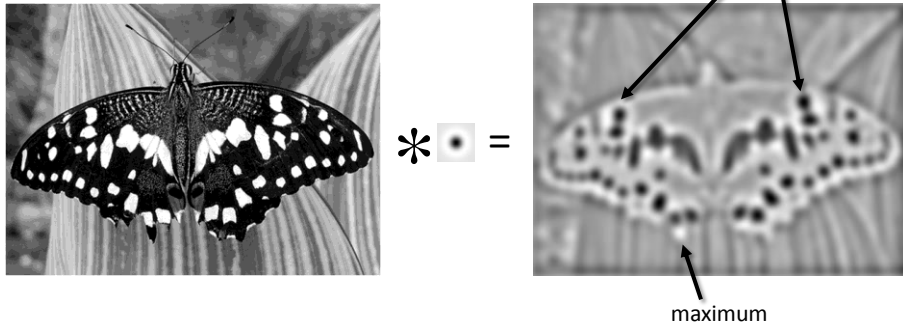
$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

(very similar to a Difference of Gaussians (DoG) – i.e. a Gaussian minus a slightly smaller Gaussian)



## Laplacian of Gaussian

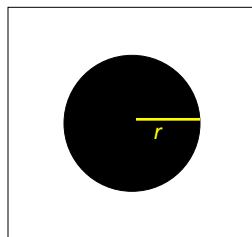
- “Blob” detector



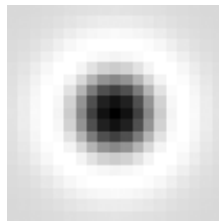
- Find maxima *and minima* of LoG operator in space and scale

## Scale selection

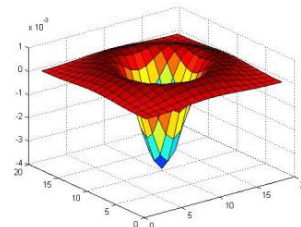
- At what scale does the Laplacian achieve a maximum response for a binary circle of radius  $r$ ?



image

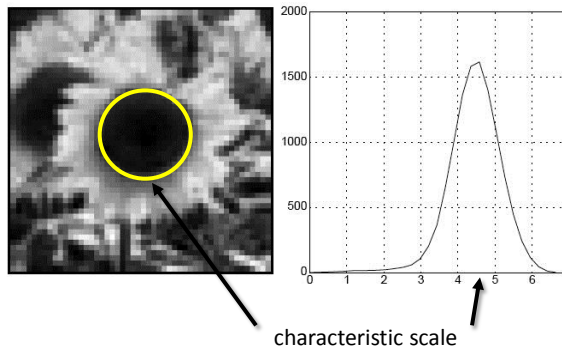


Laplacian



## Characteristic scale

- We define the characteristic scale as the scale that produces peak of Laplacian response



T. Lindeberg (1998). "[Feature detection with automatic scale selection.](#)"  
*International Journal of Computer Vision* **30** (2): pp 77--116.

## Scale-space blob detector: Example

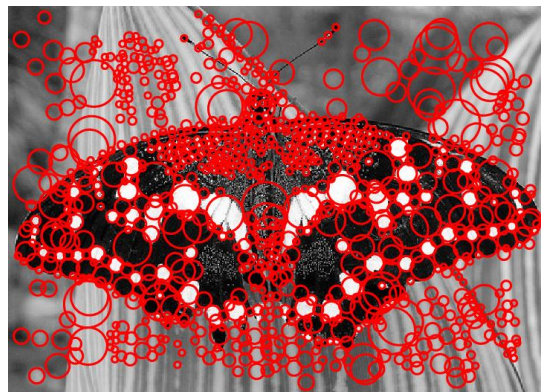


## Scale-space blob detector: Example



sigma = 11.9912

## Scale-space blob detector: Example

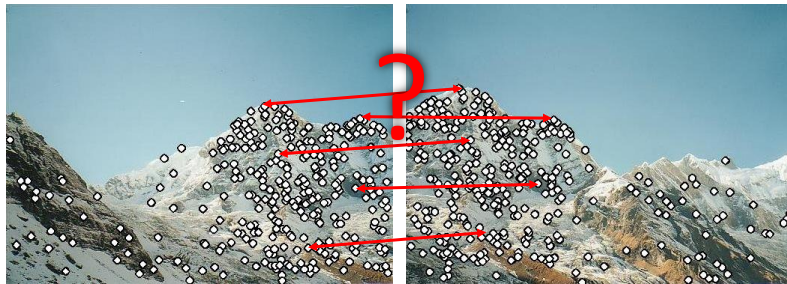


# Questions?

## Feature descriptors

We know how to detect good points

Next question: **How to match them?**

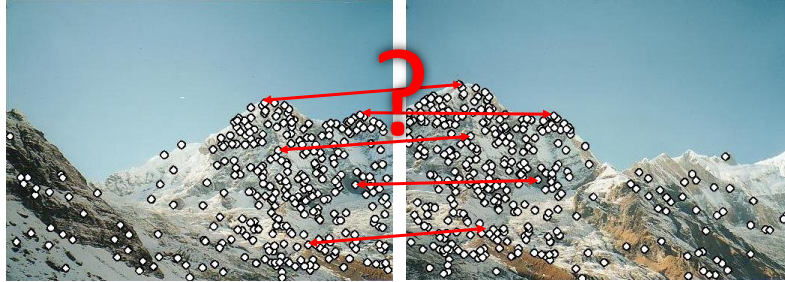


**Answer:** Come up with a *descriptor* for each point,  
find similar descriptors between the two images

## 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 vs. discriminability

- Invariance:
  - Descriptor shouldn't change even if image is transformed
- Discriminability:
  - Descriptor should be highly unique for each point

## Invariance

- Most feature descriptors 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)
  - Limited illumination/contrast changes

## How to achieve invariance

Need both of the following:

1. Make sure your detector is invariant
2. Design an invariant feature descriptor
  - Simplest descriptor: a single 0
    - What's this invariant to?
  - Next 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 could be computed in several ways:
    - $\mathbf{x}_{\max}$ , the eigenvector of  $\mathbf{H}$  corresponding to  $\lambda_{\max}$  (what could go wrong?)
    - Smoothed gradient direction at the center point
  - Rotate the patch according to this angle

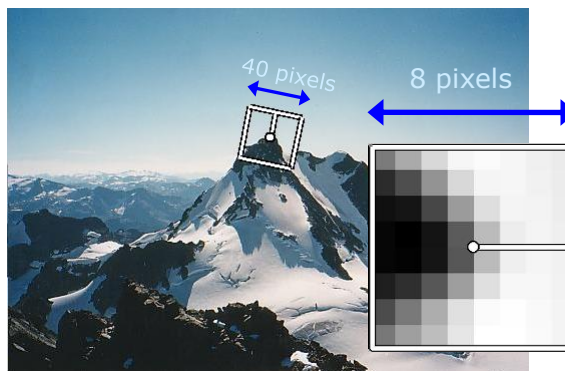


Figure by Matthew Brown

## Multiscale Oriented PatchS 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

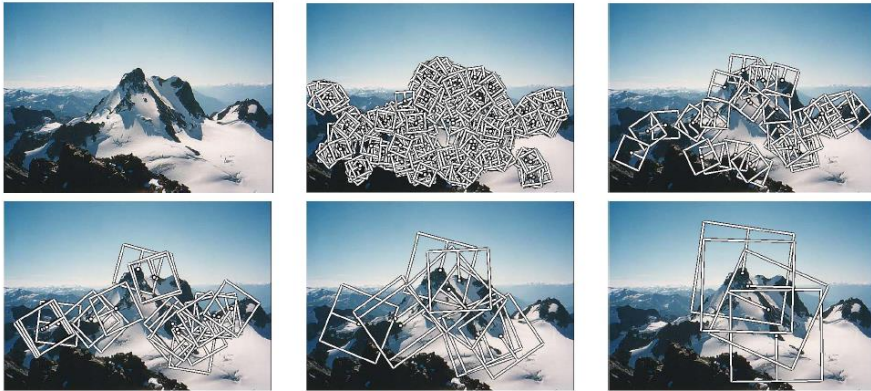
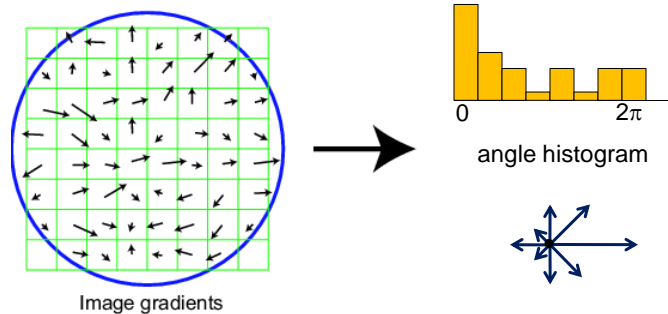


Figure 1. Multi-scale Oriented Patches (MOPS) extracted at five pyramid levels from one of the Matier images. The boxes show the feature orientation and the region from which the descriptor vector is sampled.

## Scale Invariant Feature Transform

Basic idea:

- Take 16x16 square window around detected feature
- Compute edge orientation (angle of the gradient -  $90^\circ$ ) for each pixel
- Throw out weak edges (threshold gradient magnitude)
- Create histogram of surviving edge orientations



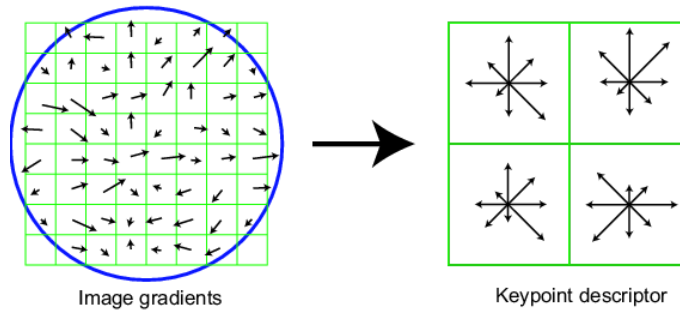
Adapted from slide by David Lowe



# SIFT descriptor

Full version

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

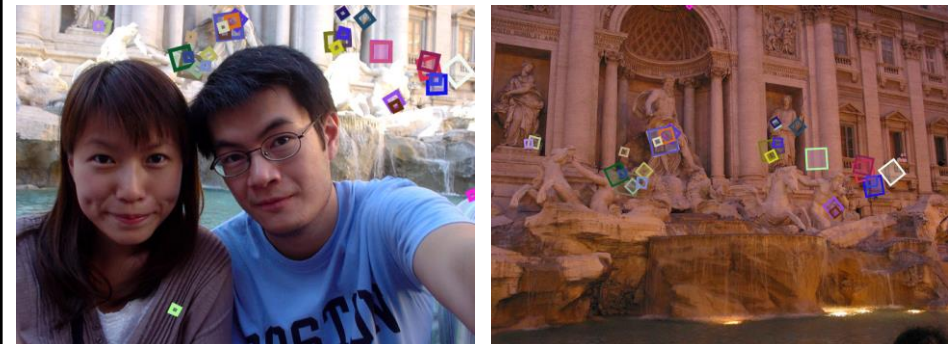


Adapted from slide by David Lowe

# 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
  - [http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known\\_implementations\\_of\\_SIFT](http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT)

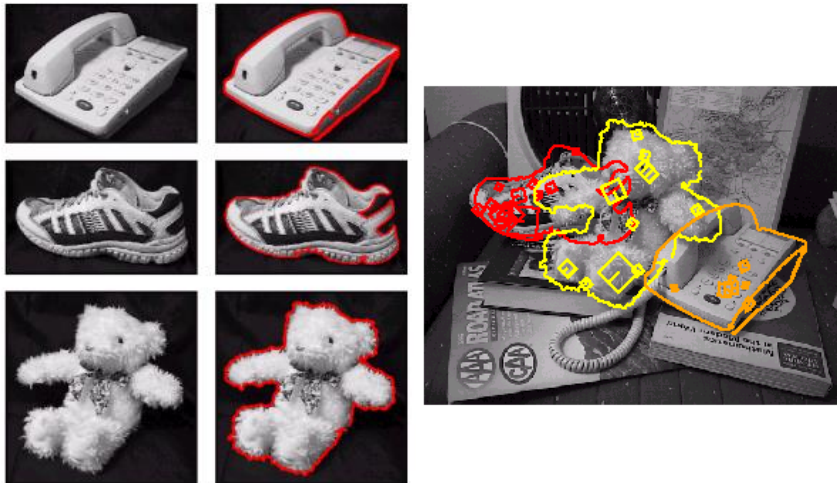


## Lots of applications

Features are used for:

- Image alignment (e.g., mosaics)
- 3D reconstruction
- Motion tracking
- Object recognition (e.g., **Google Goggles**)
- Indexing and database retrieval
- Robot navigation
- ... other

## Object recognition (David Lowe)



## 3D Reconstruction



Internet Photos ("Colosseum")



Reconstructed 3D cameras  
and points

## Sony Aibo

### SIFT usage:

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

**AIBO® Entertainment Robot**  
Official U.S. Resources and Online Destinations

**ERS-7**  
Entertainment Robot AIBO

ERS-7 with:  
Wireless LAN  
AIBO MIND software  
Energy Station  
AIBOne  
Pink Ball  
AIBO Cards (15)  
WLAN Manager CD  
Battery & AC Adapter

**3rd Generation**  
Pre-order Now!

Questions?