Lecture: Feature Descriptors

Juan Carlos Niebles and Ranjay Krishna

Stanford Vision and Learning Lab

CS 131 Roadmap

Pixels	Segments	Images	Videos	Web
Convolutions	Resizing	Recognition	Motion	Neural networks
Edges	Segmentation	Detection	Tracking	Convolutional
Features	Clustering	Machine learning		neural networks

What we will learn today?

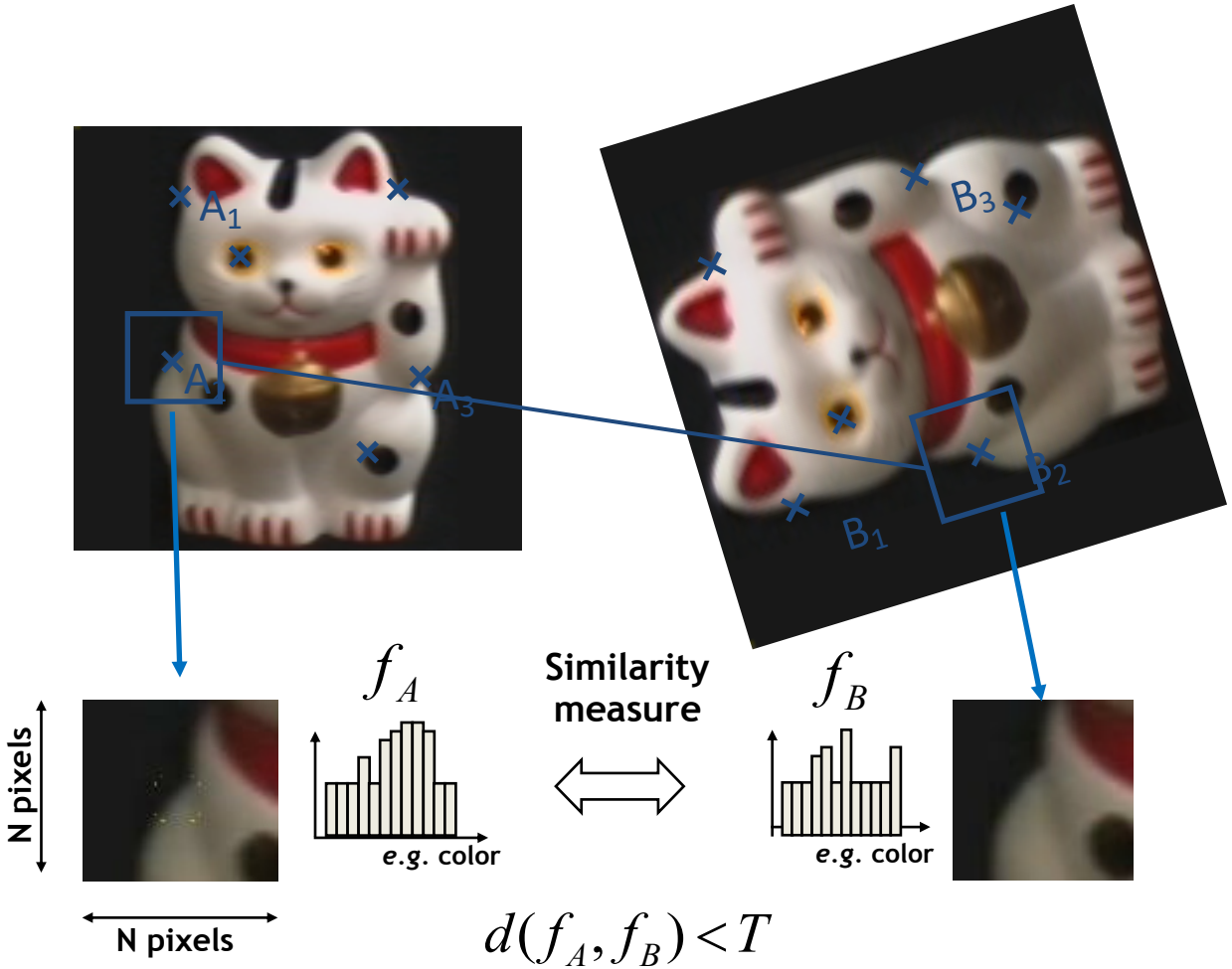
- Scale invariant region selection
  - Automatic scale selection
  - Difference-of-Gaussian (DoG) detector
- SIFT: an image region descriptor
- HOG: another image descriptor
- Application: Panorama

Some background reading: David Lowe, IJCV 2004

A quick review

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

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

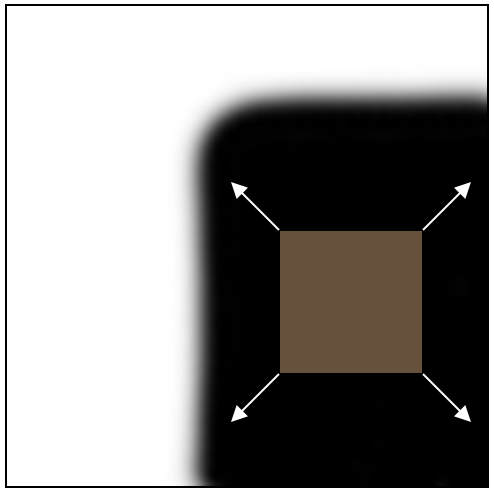
5. Match local descriptors

Slide credit: Bastian Leibe

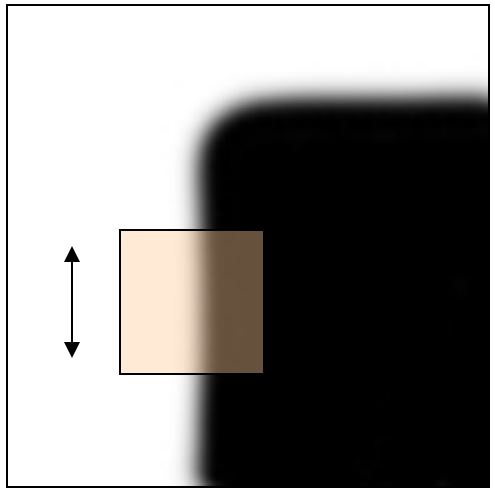
A quick review

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

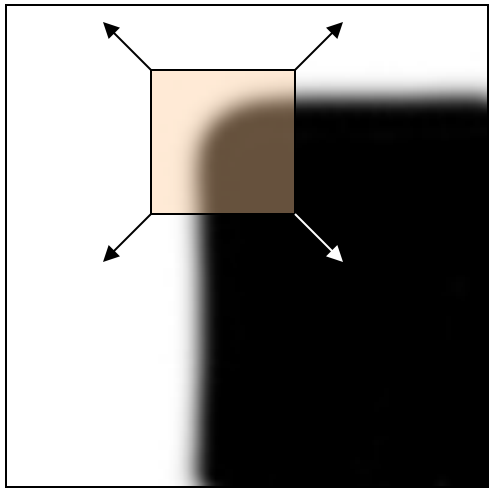
Quick review: Harris Corner Detector



“flat” region:
no change in all directions



“edge”:
no change along the edge direction



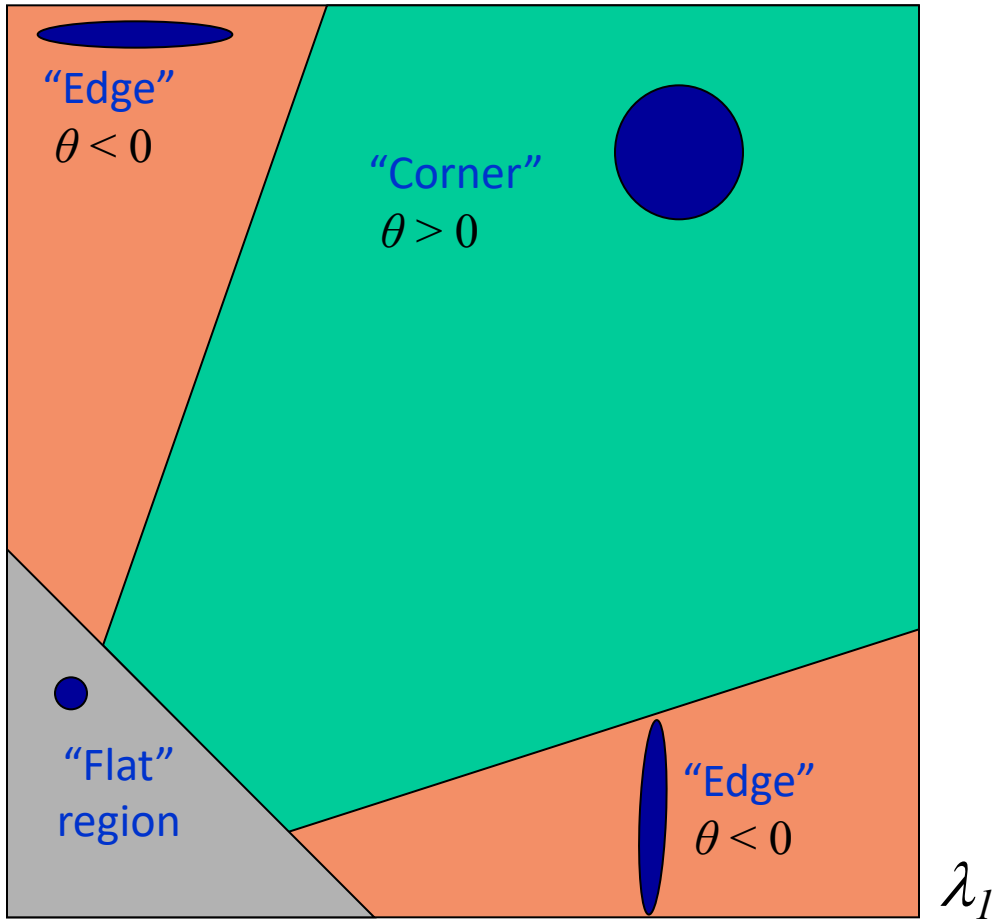
“corner”:
significant change in all directions

Slide credit: Alyosha Efros

Quick review: Harris Corner Detector

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

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



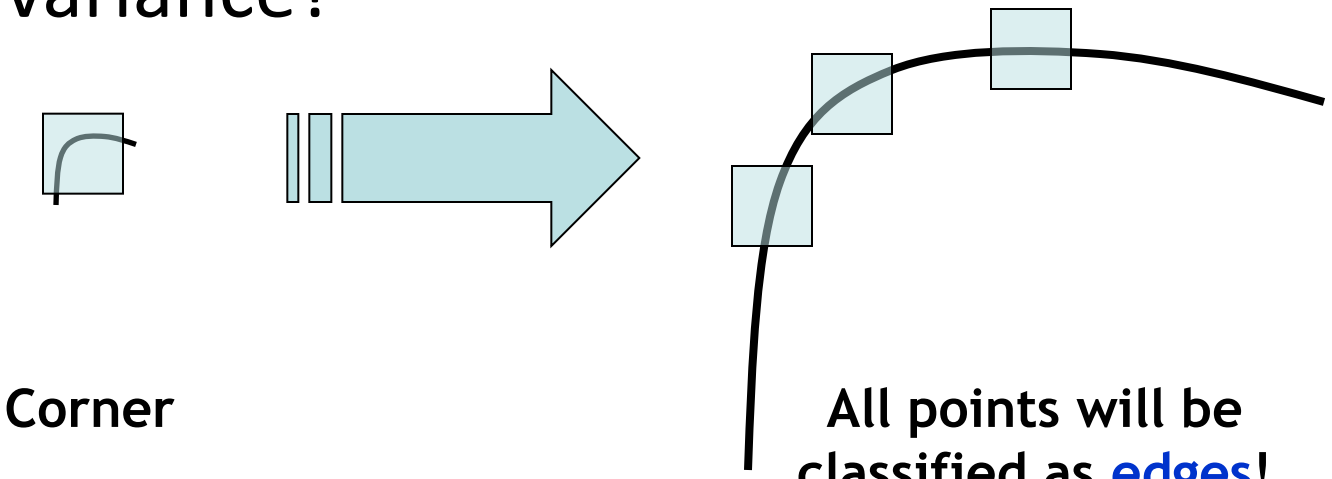
Slide credit: Kristen Grauman

Quick review: Harris Corner Detector

Slide adapted from Darya Frolova, Denis Simakov

Quick review: Harris Corner Detector

- 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 will learn today?

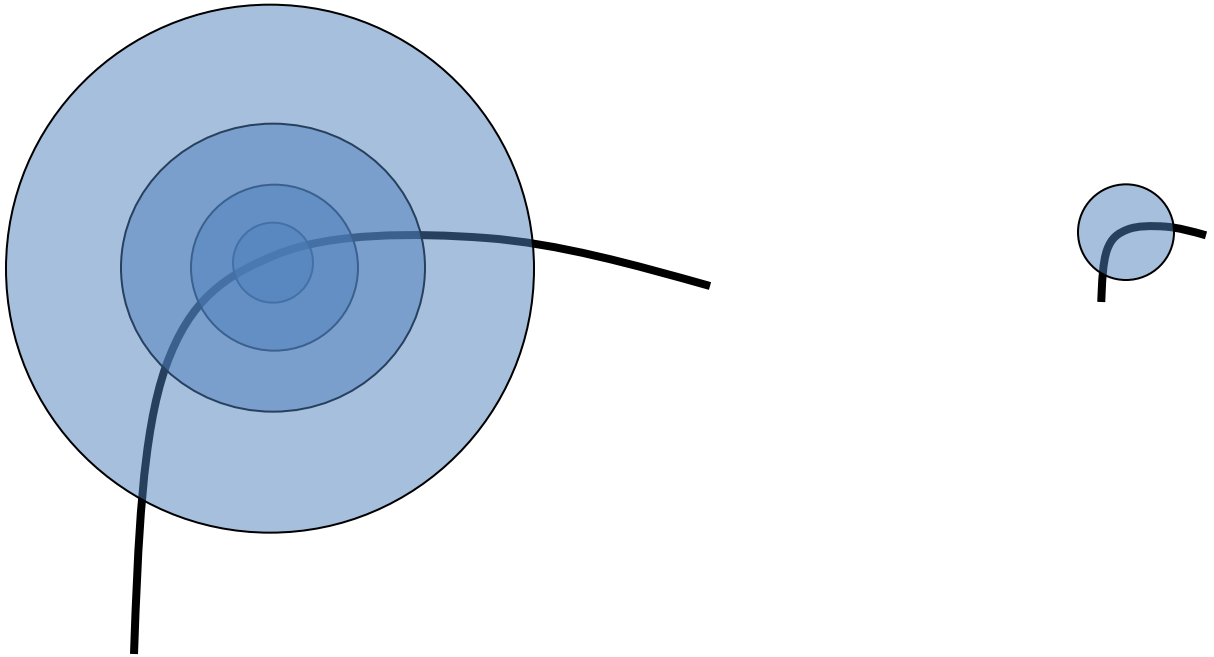
- Scale invariant keypoint detection
  - Automatic scale selection
  - Difference-of-Gaussian (DoG) detector
- SIFT: an image region descriptor
- HoG: another image region descriptor
- Application: Panorama

Does scale matter?

- When detecting corners, the `scale` of the window you use can change the corners you detect.

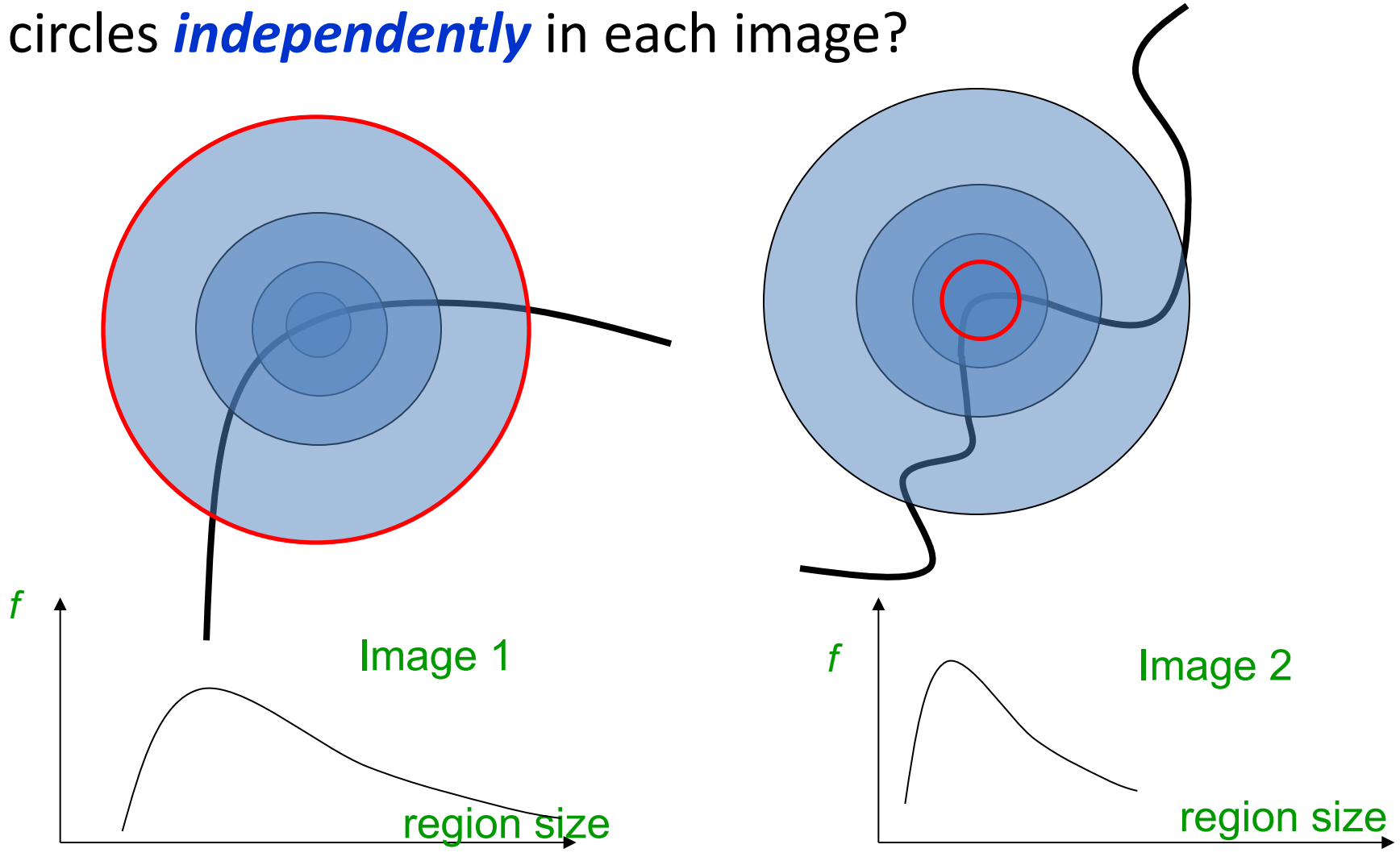
Scale Invariant Detection

- Consider regions (e.g. circles) of different sizes around a point
- What region size do we choose, so that the regions look the same in both images?



Scale Invariant Detection

- The problem: how do we choose corresponding circles ***independently*** in each image?

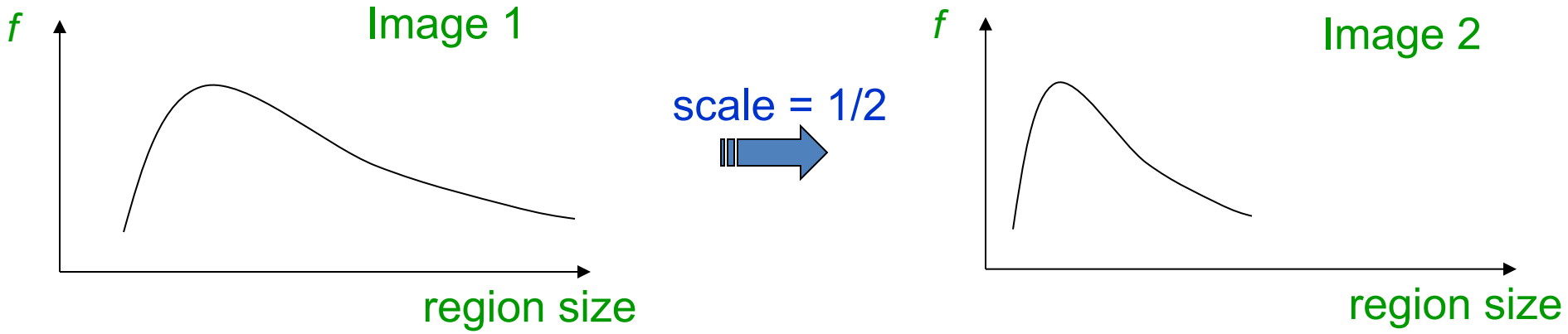


Scale Invariant Detection

- Solution:
  - Design a function on the region (circle), which is "scale invariant" (the same for corresponding regions, even if they are at different scales)

Example: average intensity. For corresponding regions (even of different sizes) it will be the same.

  - For a point in one image, we can consider it as a function of region size (circle radius)

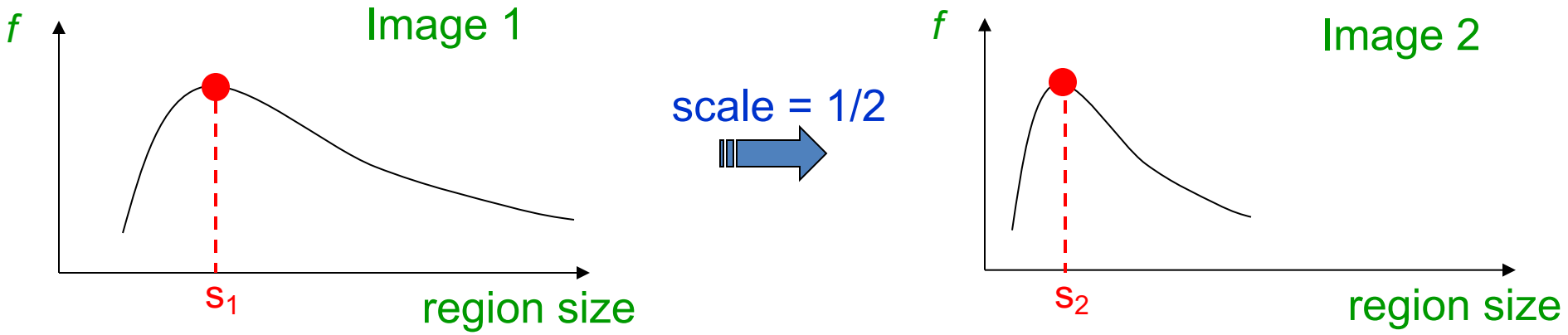


Scale Invariant Detection

- Common approach:
  Take a local maximum of this function

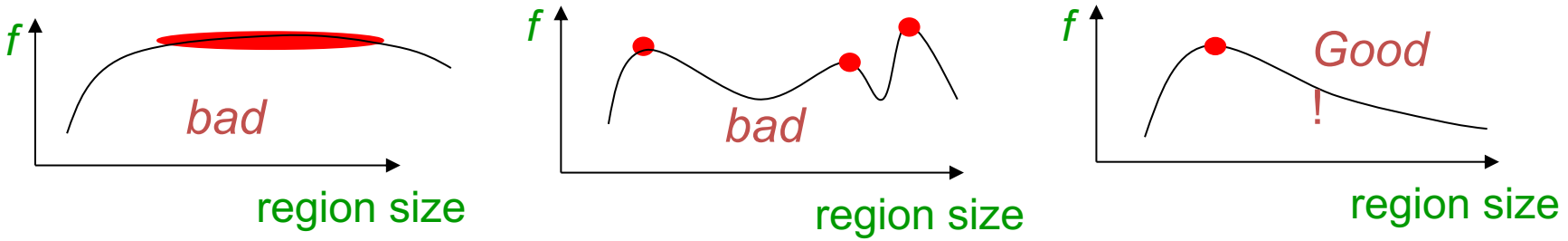
- Observation: region size, for which the maximum is achieved, should be *co-variant* with image scale.

Important: this scale invariant region size is found in each image independently!



Scale Invariant Detection

- A “good” function for scale detection:
  has one stable sharp peak



- For usual images: a good function would be one which responds to contrast (sharp local intensity change)

Scale Invariant Detection

- Functions for determining scale $f = \text{Kernel} * \text{Image}$

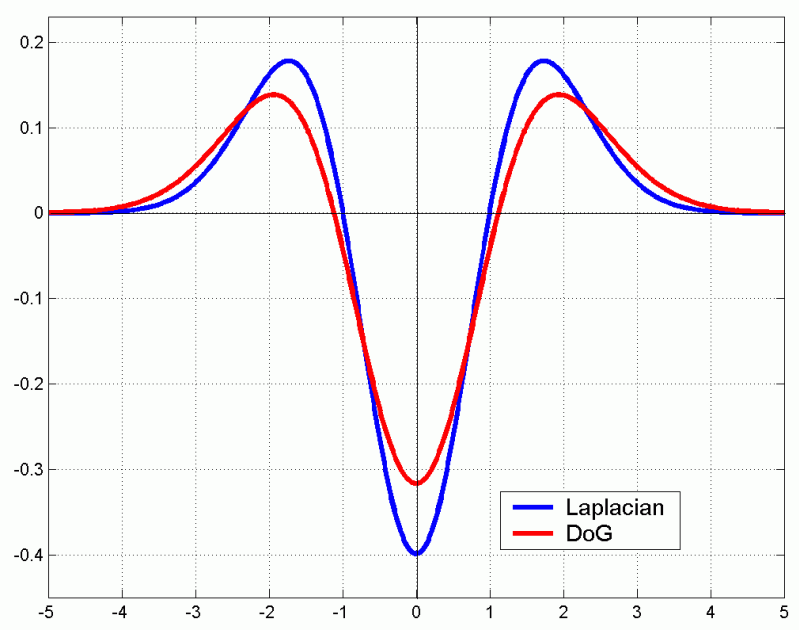
Kernels:

$$L = \sigma^2 \left(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

(Laplacian)

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)

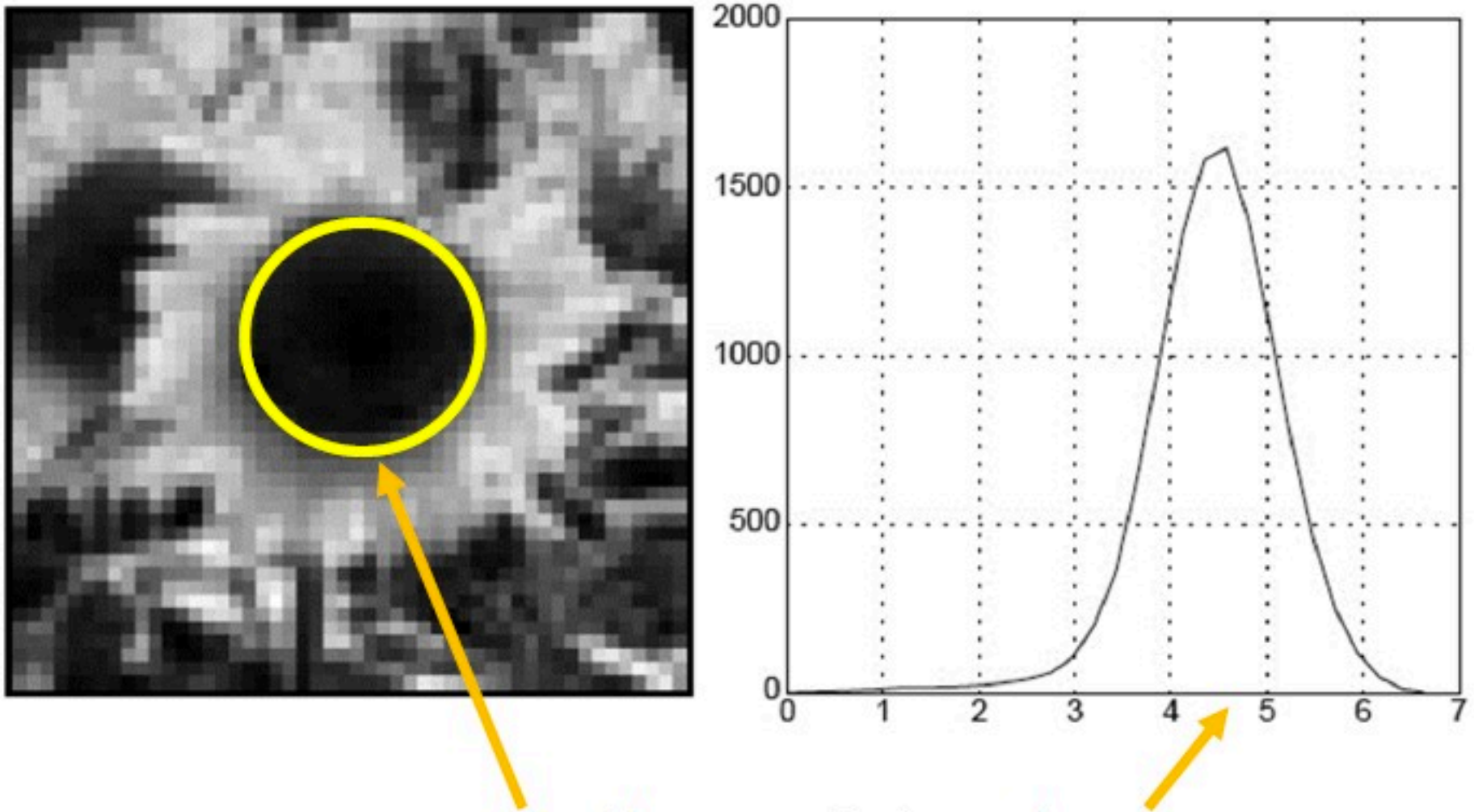


where Gaussian

$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

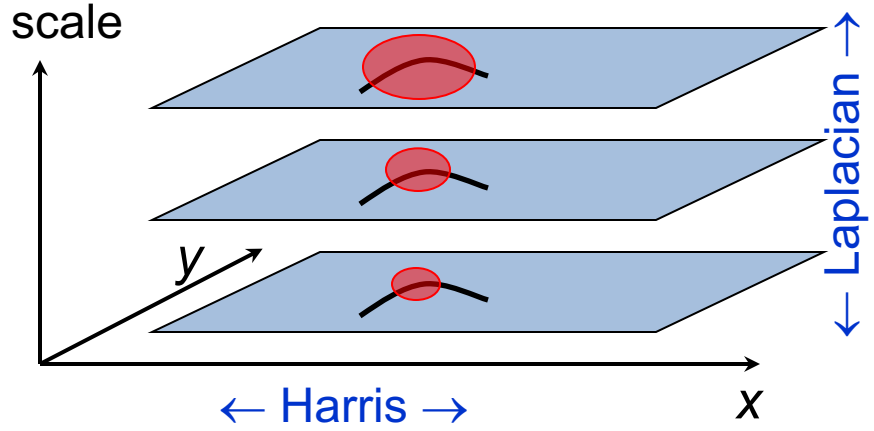
Note: both kernels are invariant to *scale* and *rotation*

Laplacian



Scale Invariant Detectors

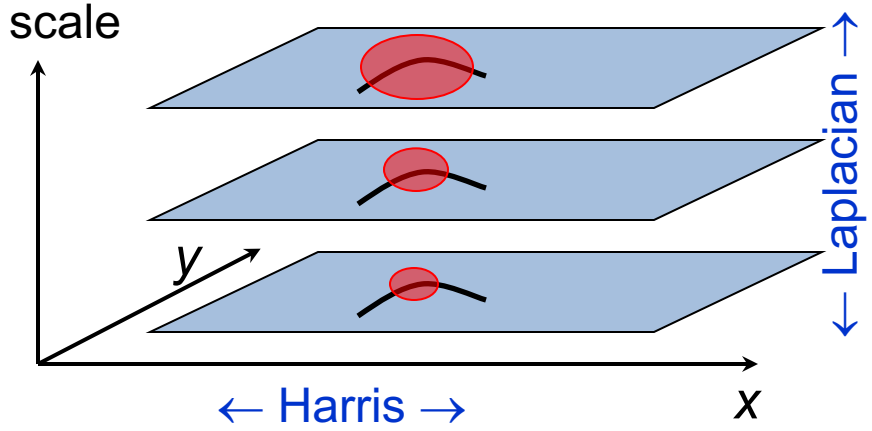
- Harris-Laplacian[1]
  Find local maximum of:
  - Harris corner detector in space (image coordinates)
  - Laplacian in scale



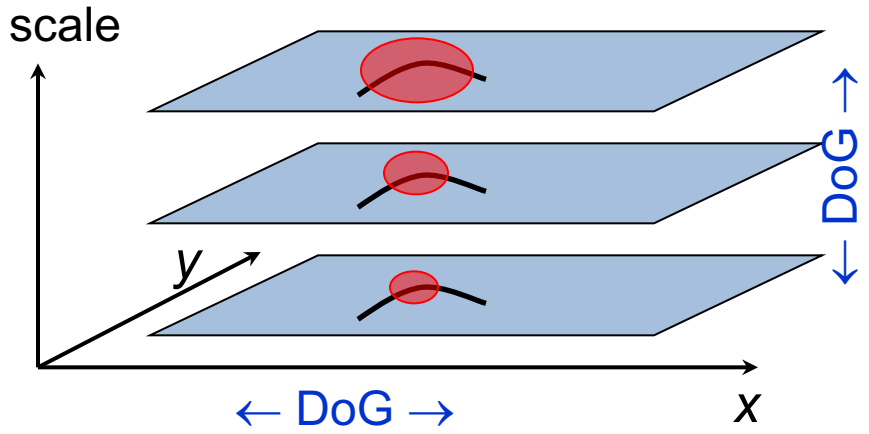
[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

Scale Invariant Detectors

- Harris-Laplacian[1]
  Find local maximum of:
  - Harris corner detector in space (image coordinates)
  - Laplacian in scale



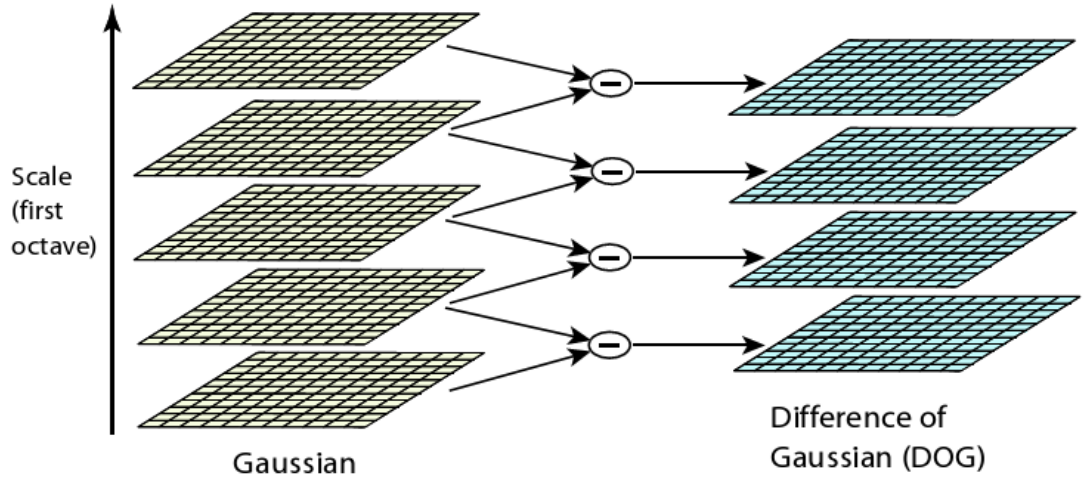
- DoG (from SIFT by Lowe)[2]
  Find local maximum of:
  - Difference of Gaussians in space and scale



[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

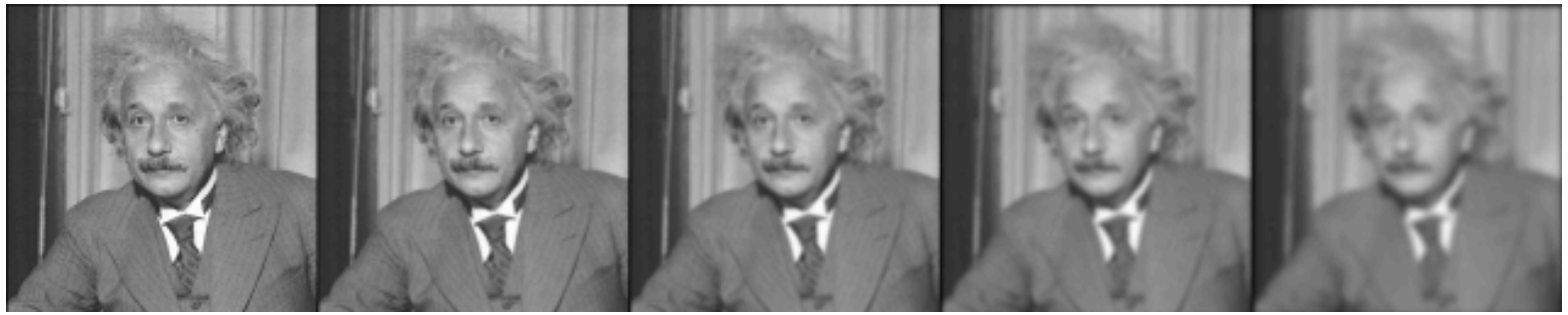
Difference-of-Gaussians

$G(k^2\sigma)*$
$G(k\sigma)*I$
$G(\sigma)*I$



$$D(\sigma) \equiv (G(k\sigma) - G(\sigma)) * I$$

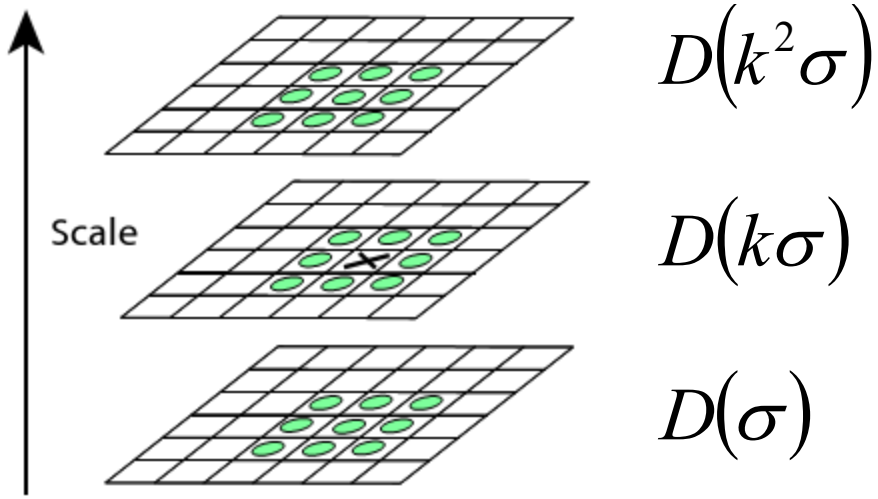
Gaussian:



DoG:

Scale-Space Extrema

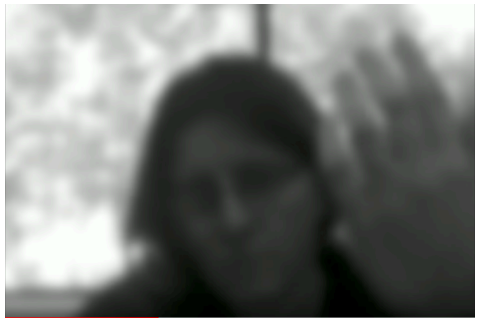
- Choose all extrema within 3x3x3 neighborhood.



X is selected if it is larger or smaller than all 26 neighbors

Example of Gaussian kernel

Original video



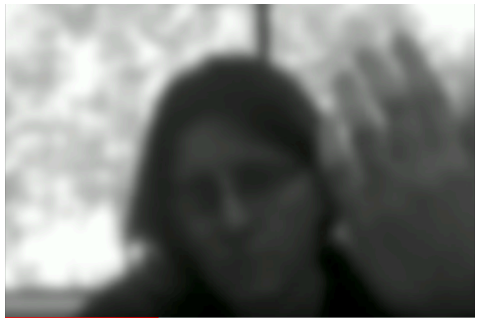
Blurred with a gaussian kernel

Blurred with a different gaussian kernel

source

Example of Gaussian kernel

Original video



Blurred with a gaussian kernel

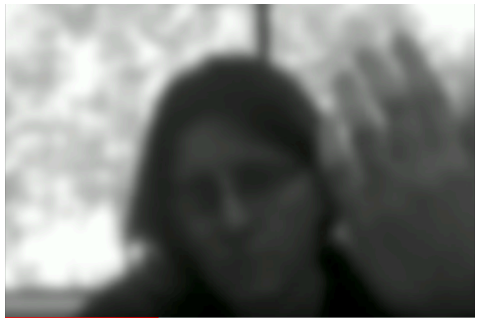
Blurred with a different gaussian kernel

What happens if you subtract one blurred image from another?

source

Difference of Gaussians (DoG)

Original video

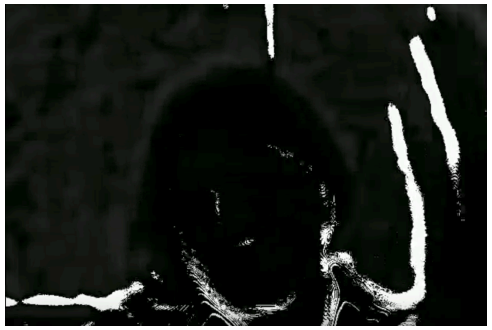


Blurred with a gaussian kernel: k_1

Blurred with a different gaussian kernel: k_2

DoG: $k_1 - k_2$

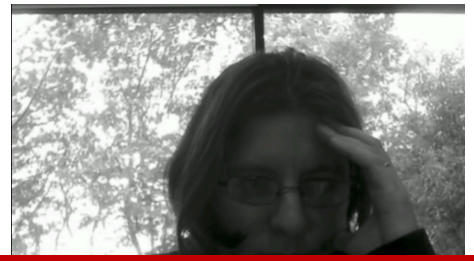
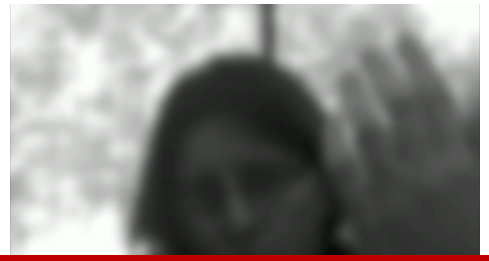
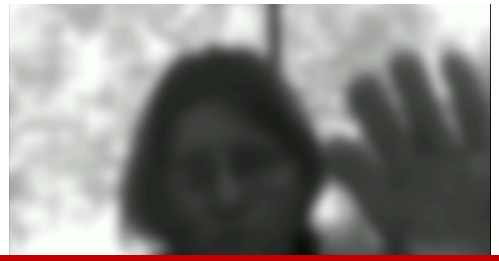
DoG: $k_1 - k_3$



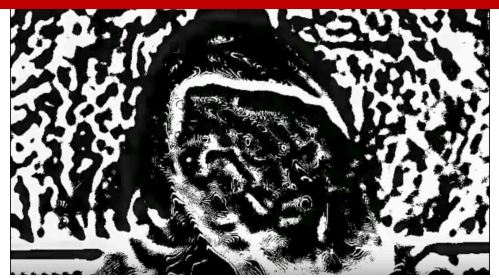
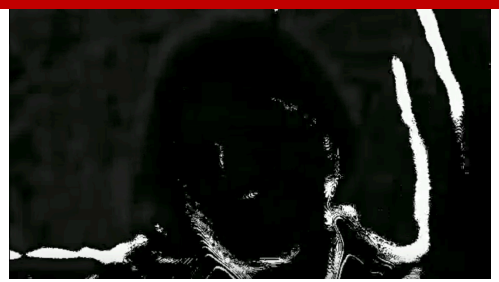
DoG: $k_1 - k_4$

source

Difference of Gaussians (DoG)

At different resolutions of kernel size, we see different fine details of the image. In other words, we can capture keypoints at varying scales.

DoG: $k_1 - k_2$ DoG: $k_1 - k_3$ DoG: $k_1 - k_4$

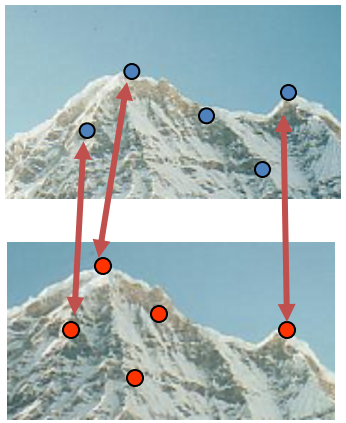
source

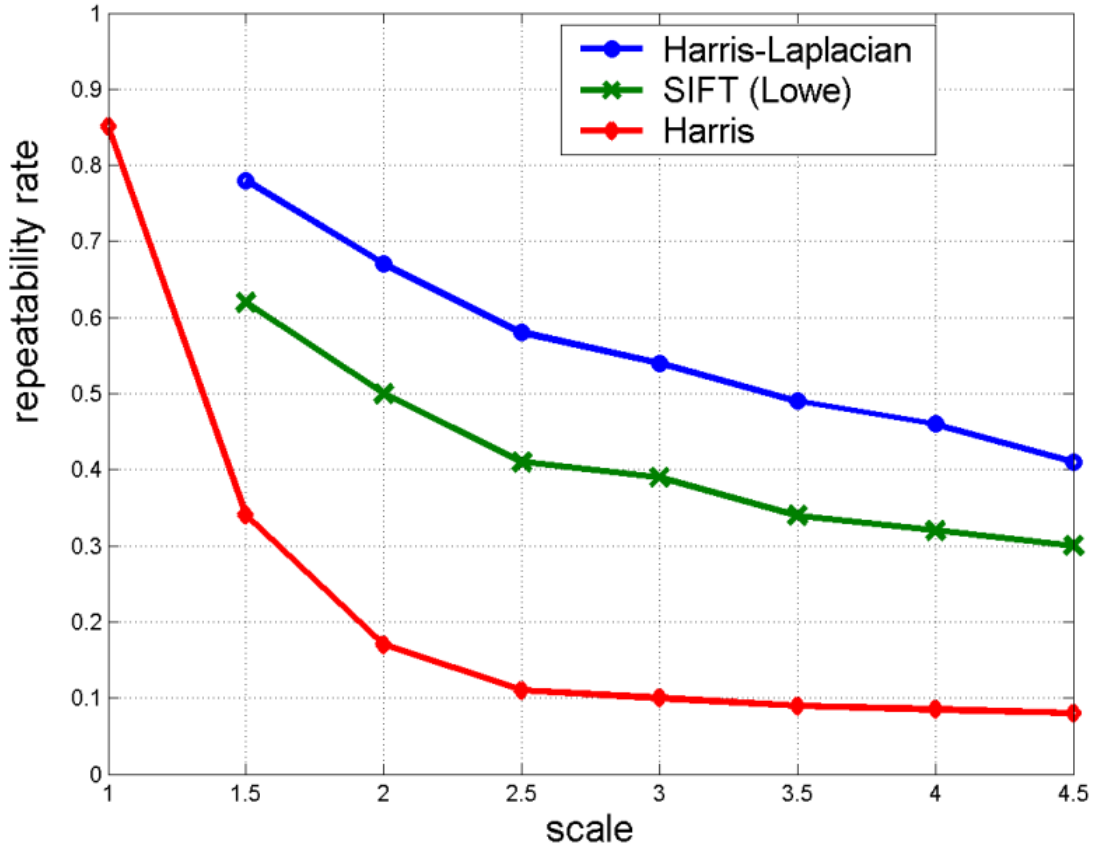
Scale Invariant Detectors

- Experimental evaluation of detectors w.r.t. scale change

Repeatability rate:

$$\frac{\#\text{ correspondences}}{\#\text{ possible correspondences}}$$





K.Mikolajczyk, C.Schmid. “Indexing Based on Scale Invariant Interest Points”. ICCV 2001

Scale Invariant Detection: Summary

- Given: two images of the same scene with a large *scale difference* between them
- Goal: find *the same* interest points *independently* in each image
- Solution: search for *maxima* of suitable functions in *scale* and in *space* (over the image)

Methods:

1. Harris-Laplacian [Mikolajczyk, Schmid]: maximize Laplacian over scale, Harris' measure of corner response over the image
2. SIFT [Lowe]: maximize Difference of Gaussians over scale and space

What's next?

So we have can detect keypoints at varying scales. But what can we do with those keypoints?

Things we would like to do:

- Search:
  - We would need to find similar key points in other images
- Panorama
  - Match keypoints from one image to another.
- Etc…

For all such applications, we need a way of `describing` the keypoints.

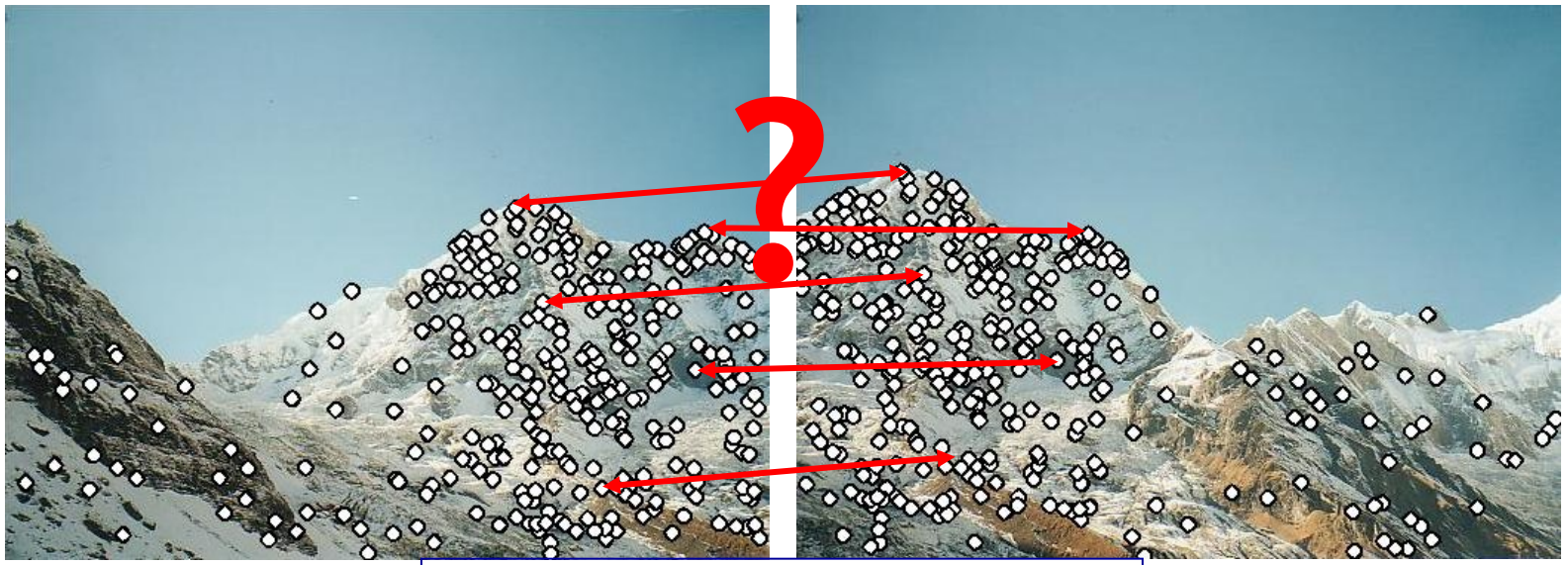
What we will learn today?

- Scale invariant region selection
  - Automatic scale selection
  - Difference-of-Gaussian (DoG) detector
- SIFT: an image region descriptor
- HoG: another image region descriptor
- Application: Panorama

Local Descriptors

- We know how to detect points
- Next question:

How to *describe* them for matching?



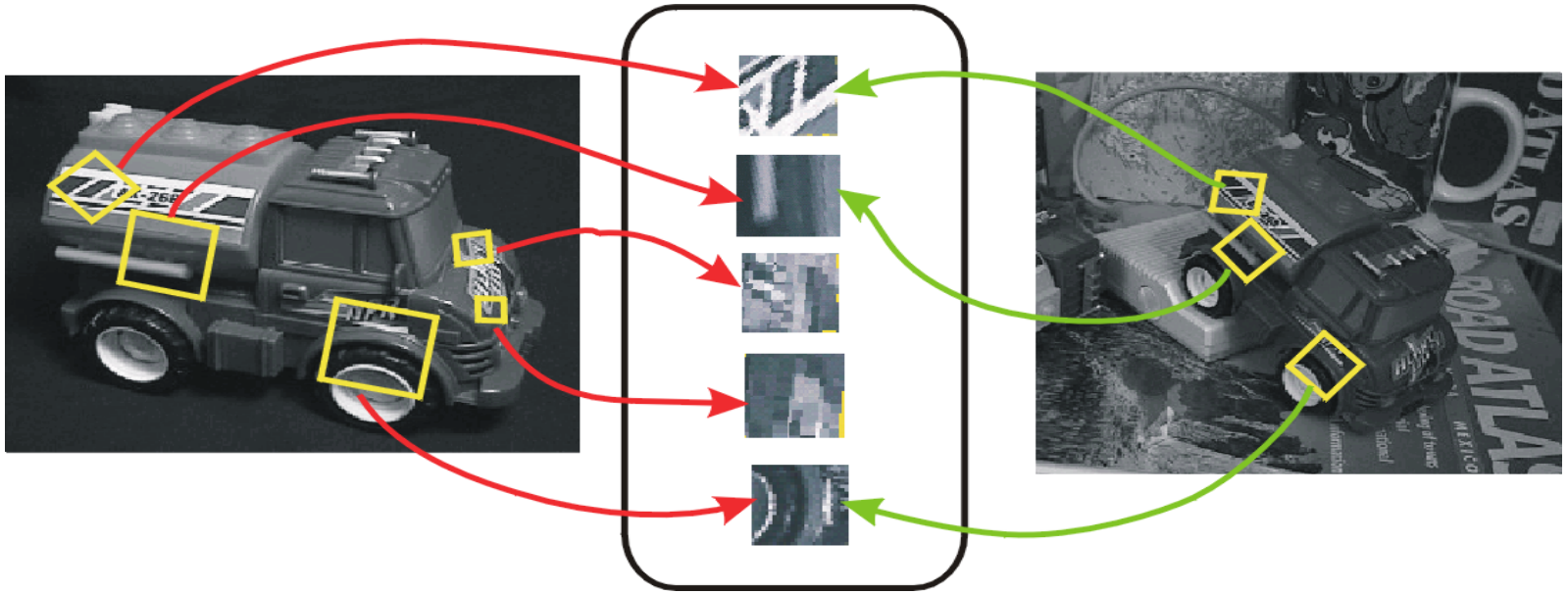
Point descriptor should be:

1. **Invariant**
2. **Distinctive**

Slide credit: Kristen Grauman

Invariant Local Features

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters

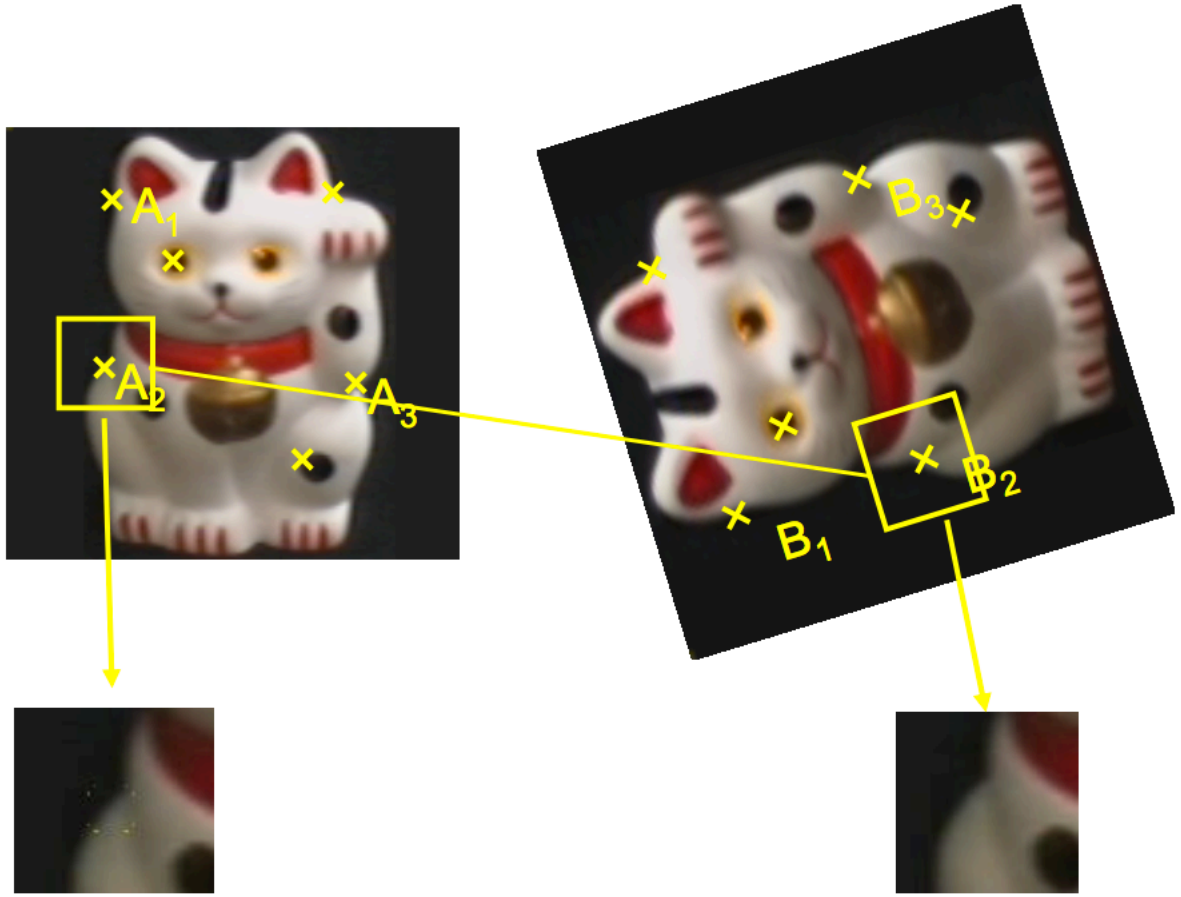


Following slides credit: CVPR 2003 Tutorial on **Recognition and Matching Based on Local Invariant Features** David Lowe

Advantages of invariant local features

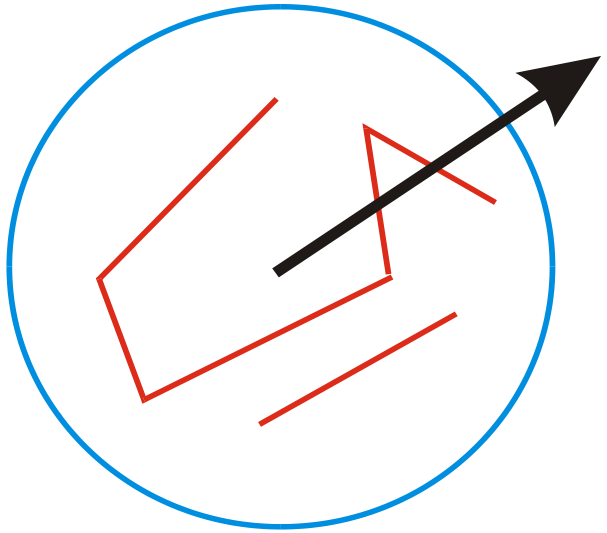
- Locality: features are local, so robust to occlusion and clutter (no prior segmentation)
- Distinctiveness: individual features can be matched to a large database of objects
- Quantity: many features can be generated for even small objects
- Efficiency: close to real-time performance
- Extensibility: can easily be extended to wide range of differing feature types, with each adding robustness

Becoming rotation invariant

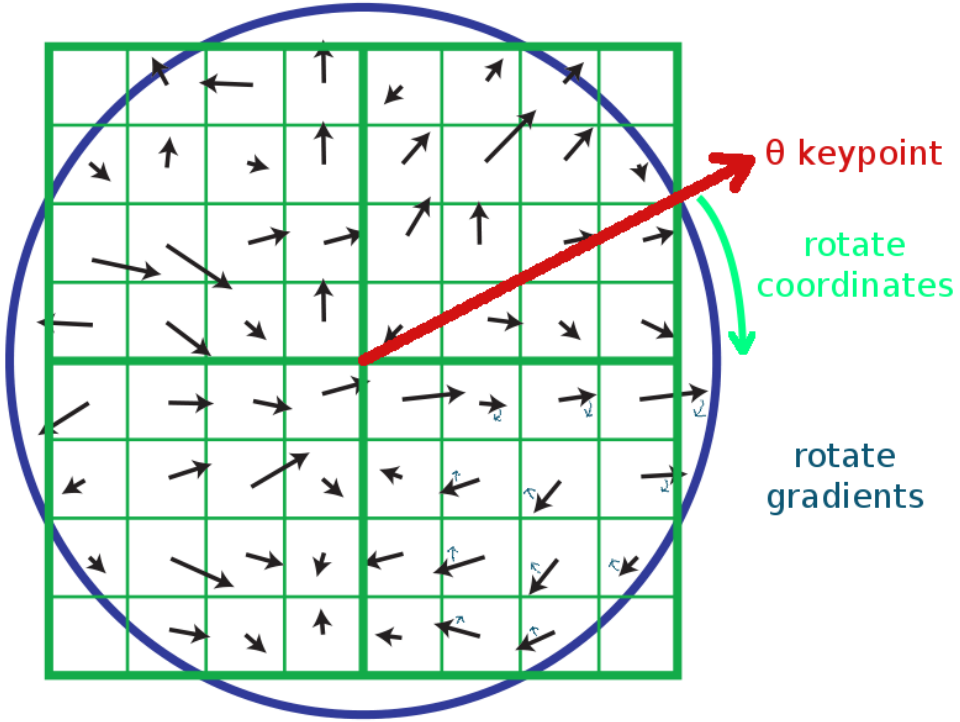


Becoming rotation invariant

- We are given a keypoint and its scale from DoG
- We will select a characteristic orientation for the keypoint (based on the most prominent gradient there; discussed next slide)
- We will describe all features **relative** to this orientation
- Causes features to be rotation invariant!
  - If the keypoint appears rotated in another image, the features will be the same, because they're **relative** to the characteristic orientation

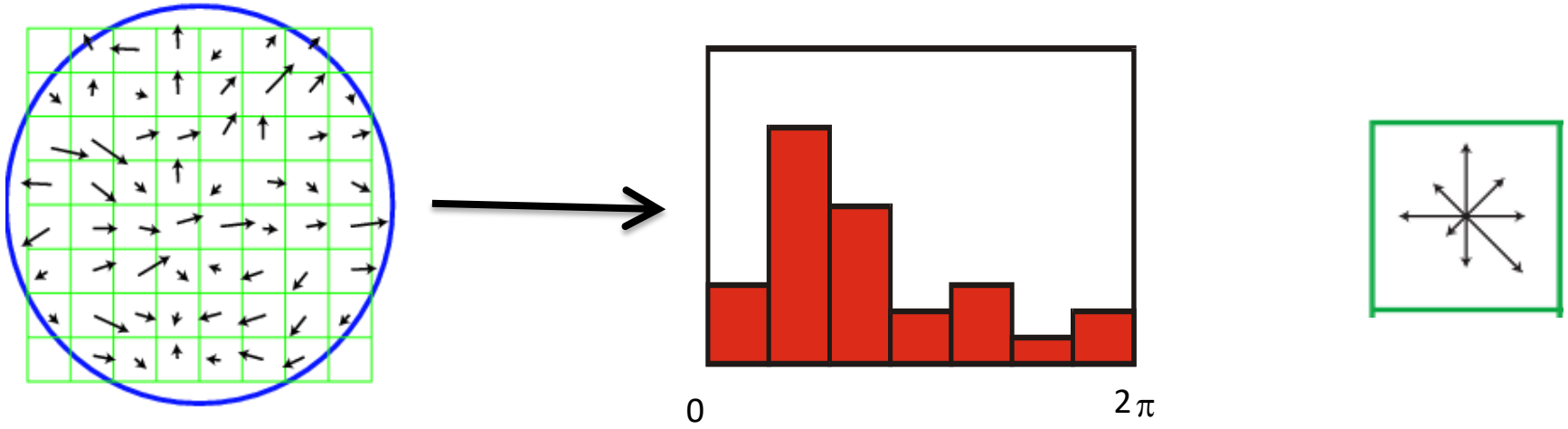


SIFT descriptor formation



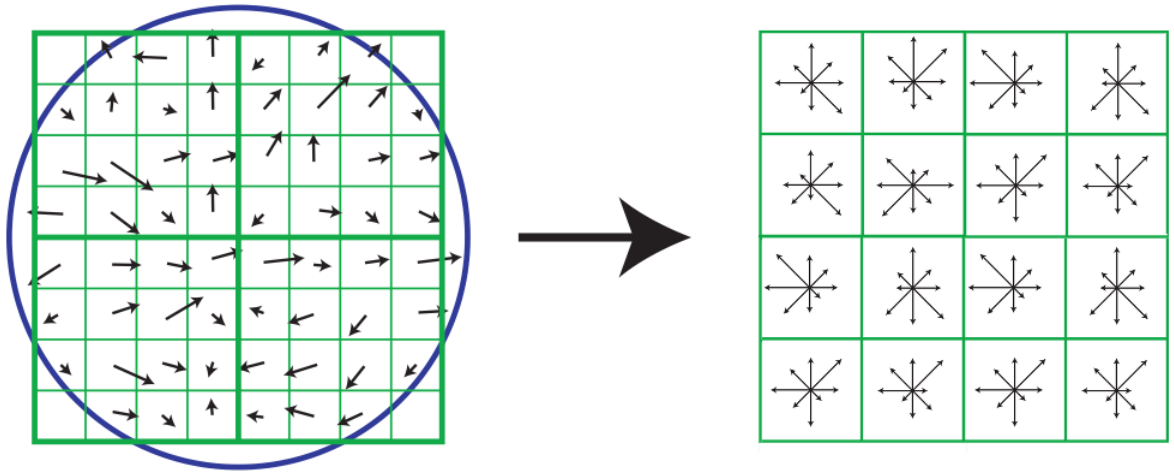
- Use the blurred image associated with the keypoint's scale
- Take image gradients over the keypoint neighborhood.
- To become rotation invariant, rotate the gradient directions AND locations by (-keypoint orientation)
  - Now we've cancelled out rotation and have gradients expressed at locations **relative** to keypoint orientation θ
  - We could also have just rotated the whole image by -θ, but that would be slower.

SIFT descriptor formation



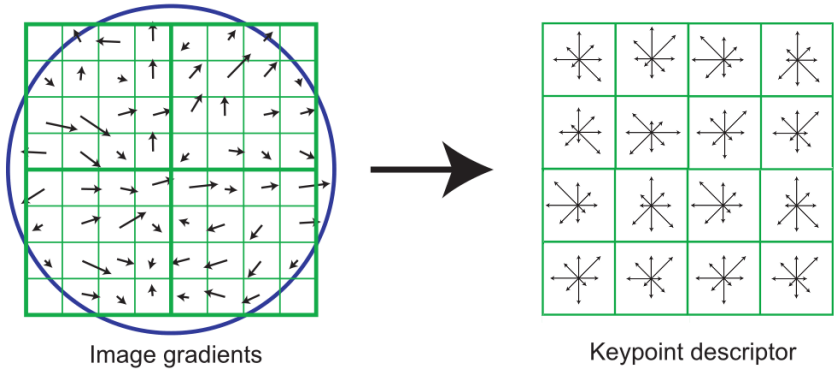
- Using precise gradient locations is fragile. We'd like to allow some "slop" in the image, and still produce a very similar descriptor
- Create array of orientation histograms (a 4x4 array is shown)
- Put the rotated gradients into their local orientation histograms
  - A gradients's contribution is divided among the nearby histograms based on distance. If it's halfway between two histogram locations, it gives a half contribution to both.
  - Also, scale down gradient contributions for gradients far from the center
- The SIFT authors found that best results were with 8 orientation bins per histogram.

SIFT descriptor formation



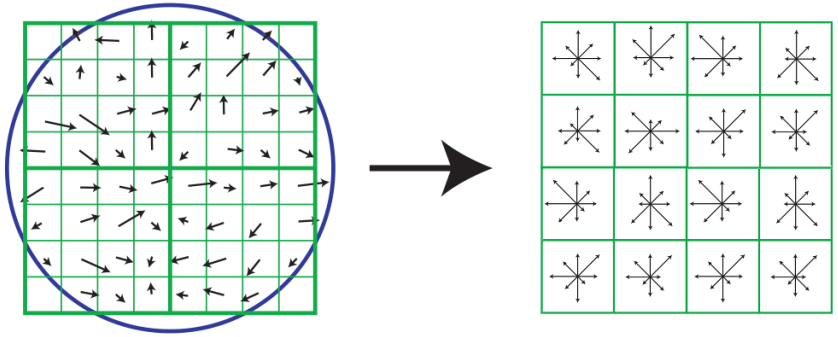
- Using precise gradient locations is fragile. We'd like to allow some "slop" in the image, and still produce a very similar descriptor
- Create array of orientation histograms (a 4x4 array is shown)
- Put the rotated gradients into their local orientation histograms
  - A gradients's contribution is divided among the nearby histograms based on distance. If it's halfway between two histogram locations, it gives a half contribution to both.
  - Also, scale down gradient contributions for gradients far from the center
- The SIFT authors found that best results were with 8 orientation bins per histogram, and a 4x4 histogram array.

SIFT descriptor formation



- 8 orientation bins per histogram, and a 4x4 histogram array, yields 8 x 4x4 = 128 numbers.
- So a SIFT descriptor is a length 128 vector, which is invariant to rotation (because we rotated the descriptor) and scale (because we worked with the scaled image from DoG)
- We can compare each vector from image A to each vector from image B to find matching keypoints!
  - Euclidean “distance” between descriptor vectors gives a good measure of keypoint similarity

SIFT descriptor formation



- Adding robustness to illumination changes:
- Remember that the descriptor is made of gradients (differences between pixels), so it's already invariant to changes in brightness (e.g. adding 10 to all image pixels yields the exact same descriptor)
- A higher-contrast photo will increase the magnitude of gradients linearly. So, to correct for contrast changes, normalize the vector (scale to length 1.0)
- Very large image gradients are usually from unreliable 3D illumination effects (glare, etc). So, to reduce their effect, clamp all values in the vector to be ≤ 0.2 (an experimentally tuned value). Then normalize the vector again.
- Result is a vector which is fairly invariant to illumination changes.

Sensitivity to number of histogram orientations

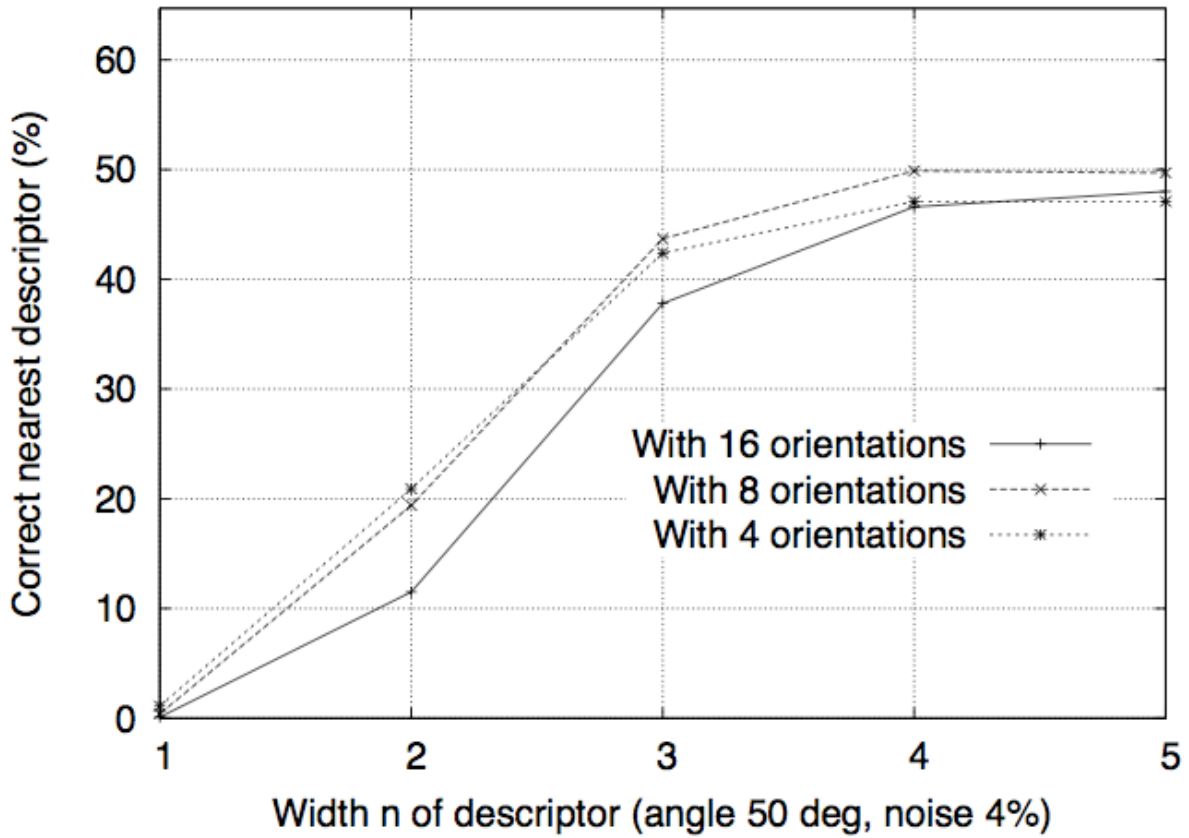
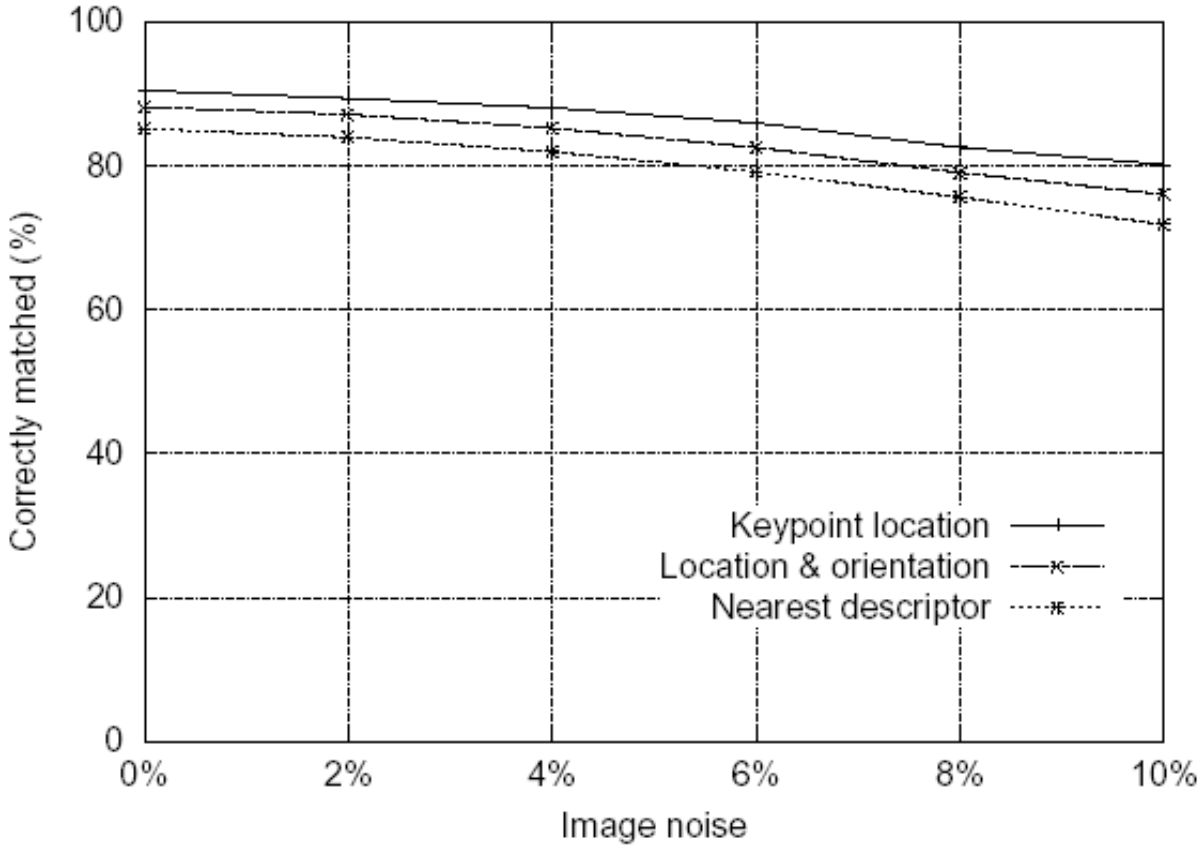


Figure 8: This graph shows the percent of keypoints giving the correct match to a database of 40,000 keypoints as a function of width of the $n \times n$ keypoint descriptor and the number of orientations in each histogram. The graph is computed for images with affine viewpoint change of 50 degrees and addition of 4% noise.

David G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004), pp. 91-110

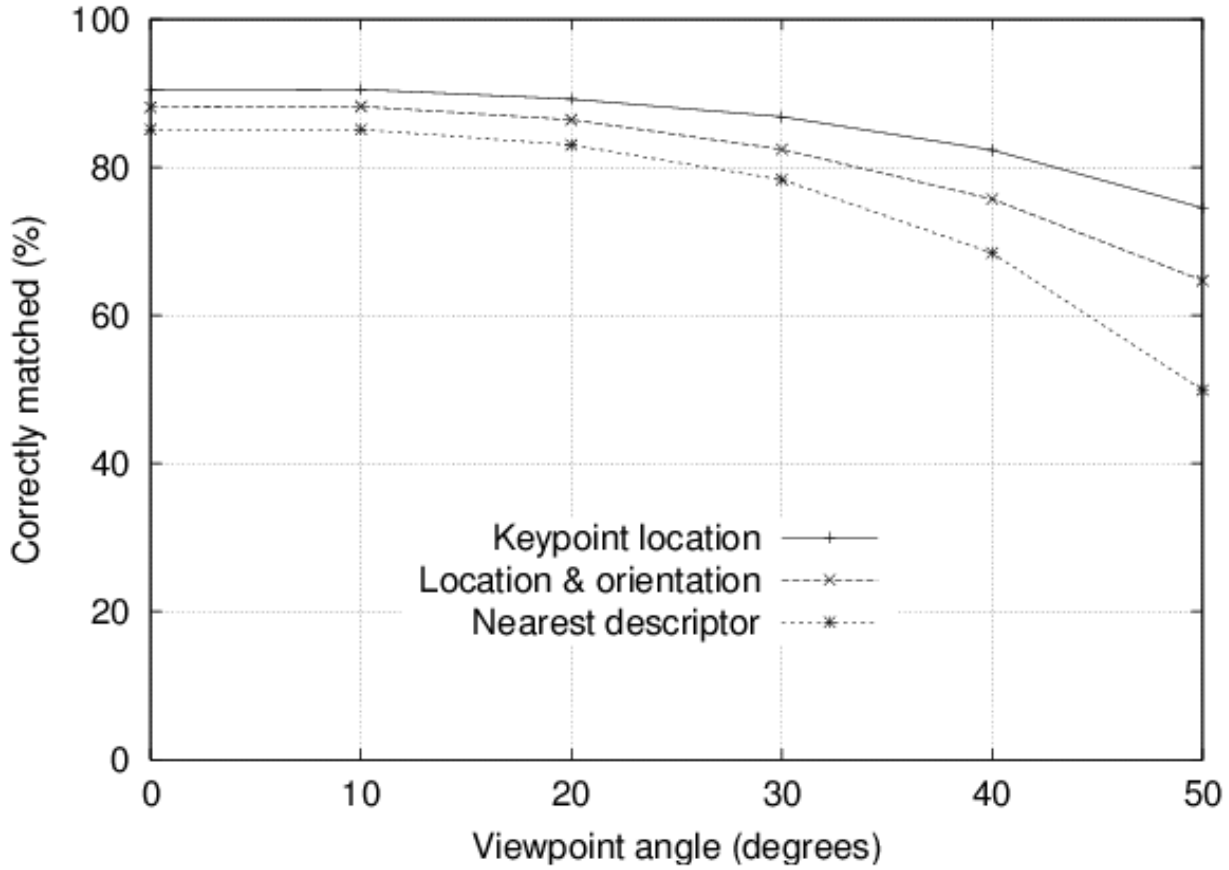
Feature stability to noise

- Match features after random change in image scale & orientation, with differing levels of image noise
- Find nearest neighbor in database of 30,000 features



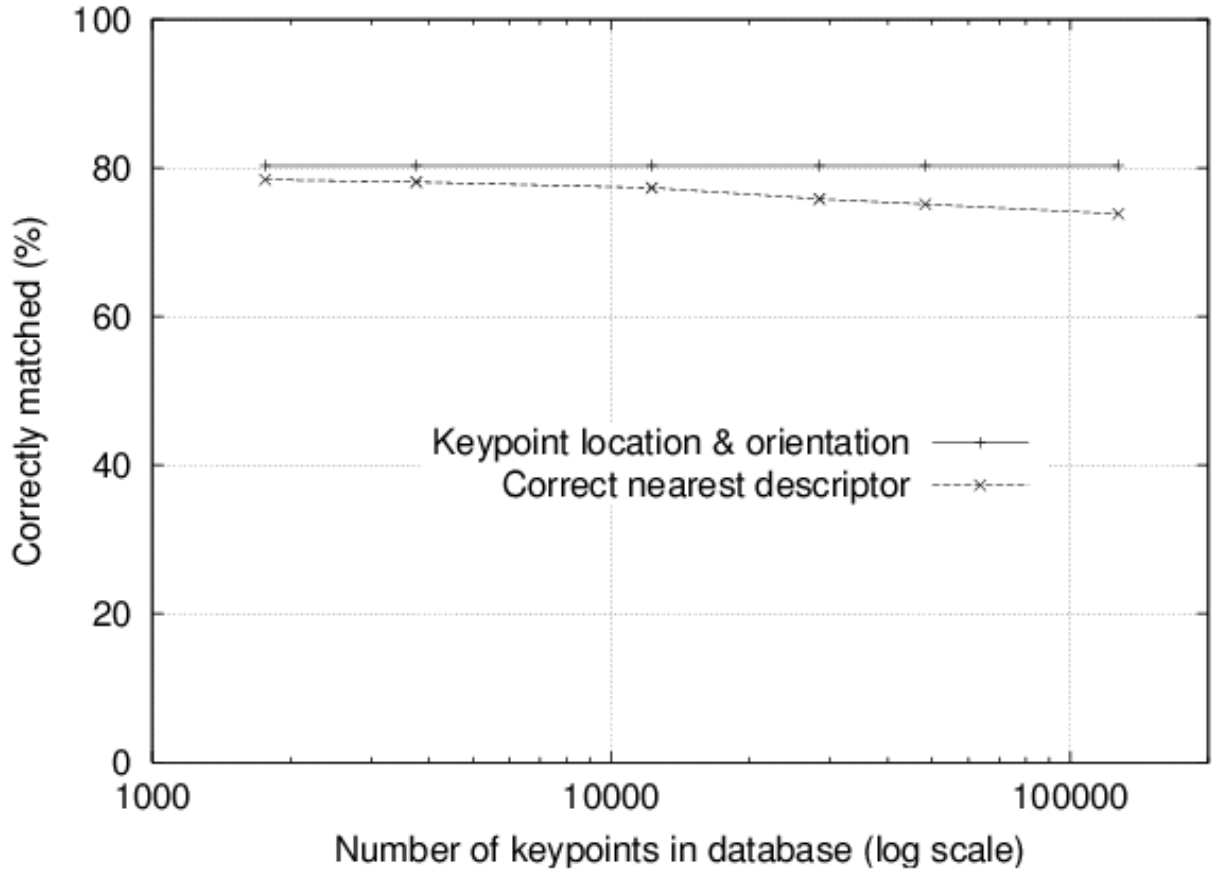
Feature stability to affine change

- Match features after random change in image scale & orientation, with 2% image noise, and affine distortion
- Find nearest neighbor in database of 30,000 features



Distinctiveness of features

- Vary size of database of features, with 30 degree affine change, 2% image noise
- Measure % correct for single nearest neighbor match



Ratio of distances reliable for matching

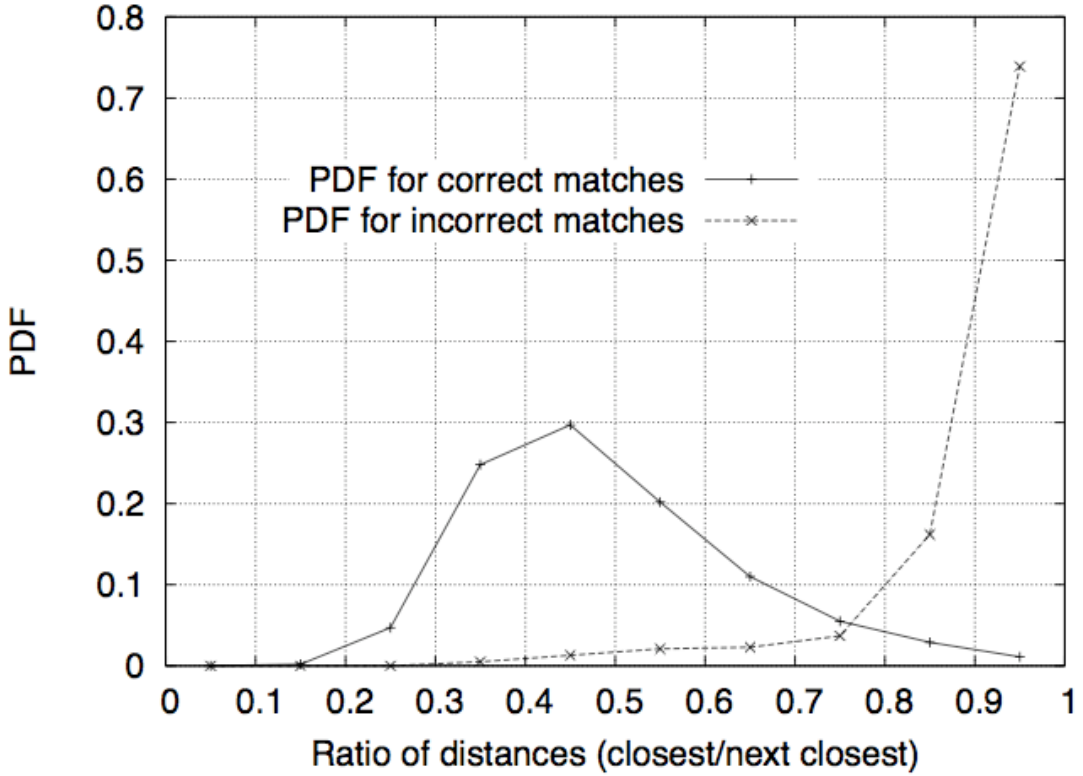


Figure 11: The probability that a match is correct can be determined by taking the ratio of distance from the closest neighbor to the distance of the second closest. Using a database of 40,000 keypoints, the solid line shows the PDF of this ratio for correct matches, while the dotted line is for matches that were incorrect.

Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affi ne transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.

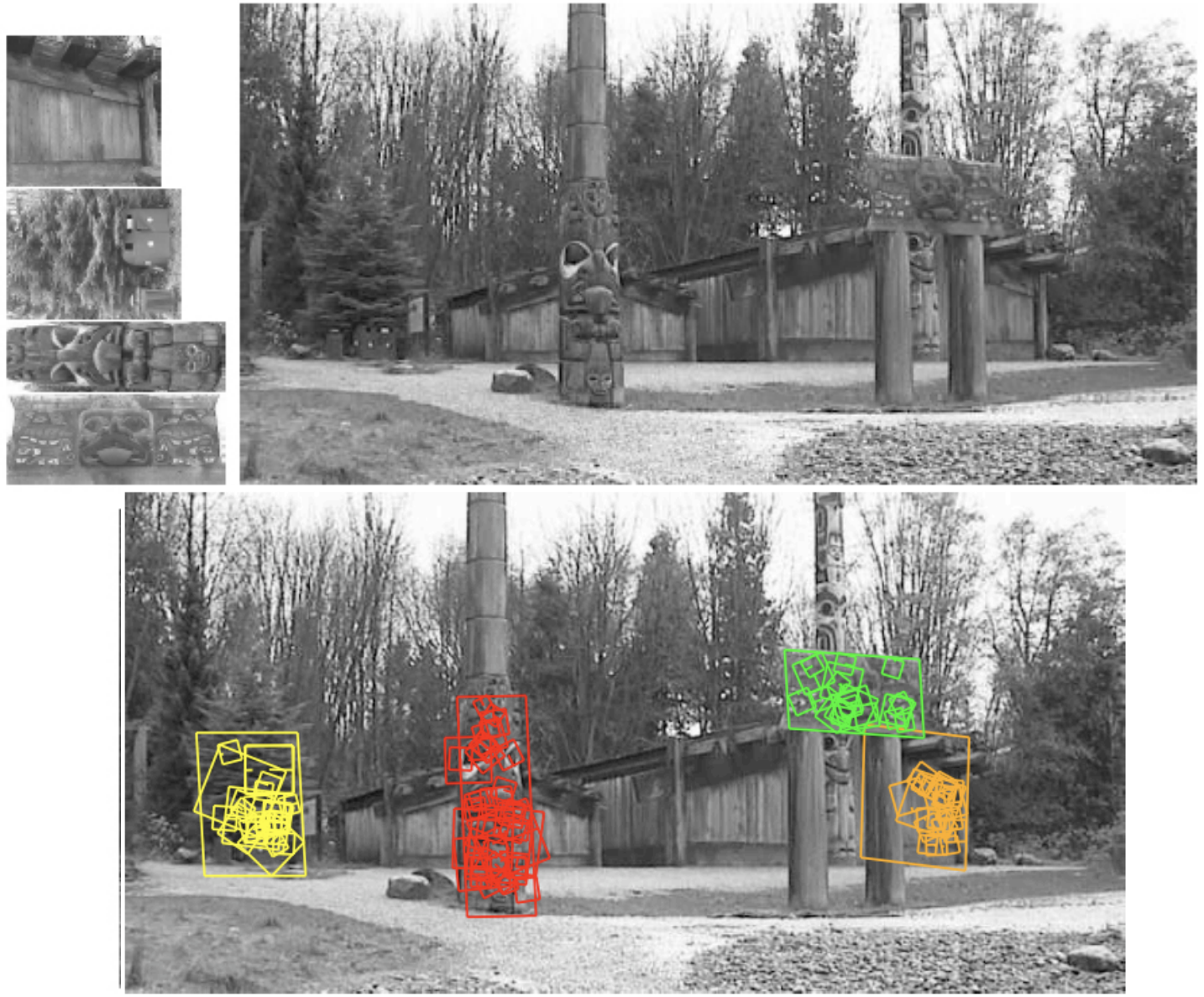
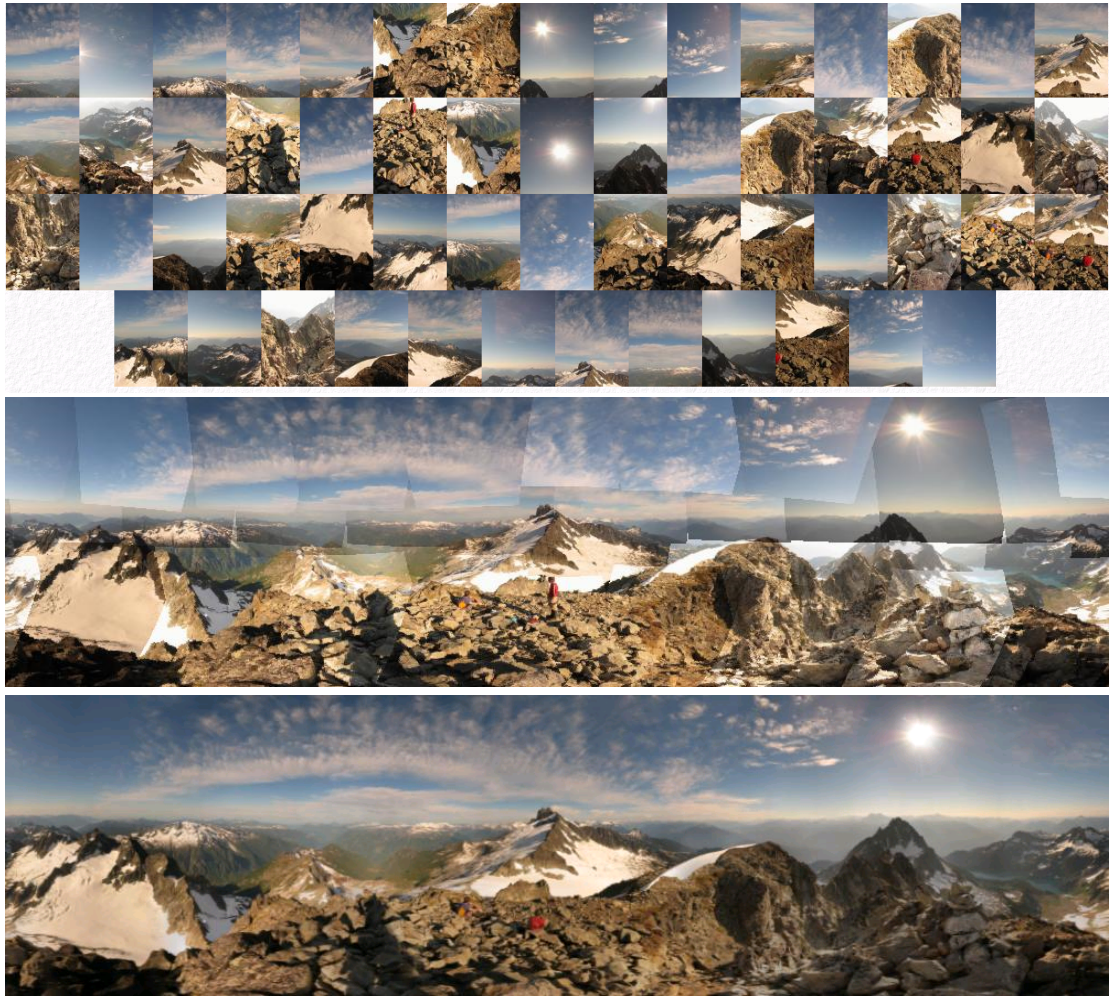


Figure 13: This example shows location recognition within a complex scene. The training images for locations are shown at the upper left and the 640x315 pixel test image taken from a different viewpoint is on the upper right. The recognized regions are shown on the lower image, with keypoints shown as squares and an outer parallelogram showing the boundaries of the training images under the affi ne transform used for recognition.

Nice SIFT resources

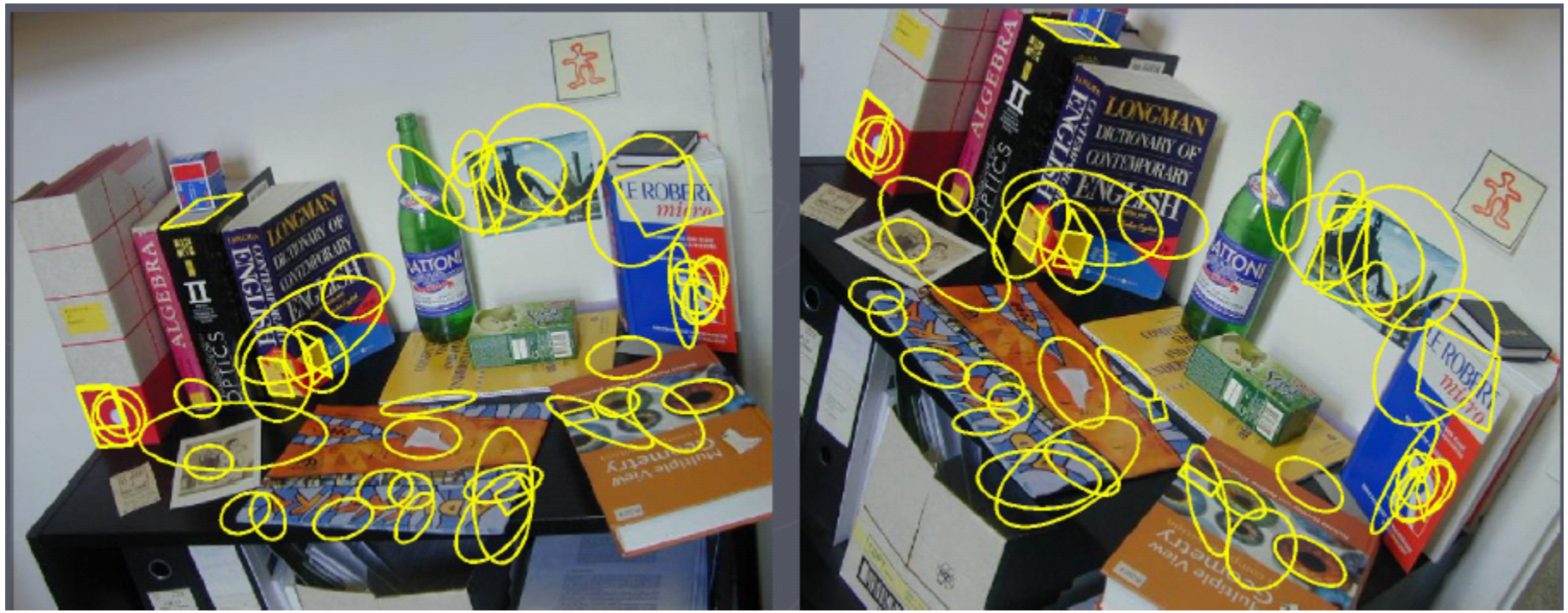
- an online tutorial:http://www.aishack.in/2010/05/sift-scale-invariant-feature-transform/
- Wikipedia:http://en.wikipedia.org/wiki/Scale-invariant_feature_transform

Automatic mosaicing



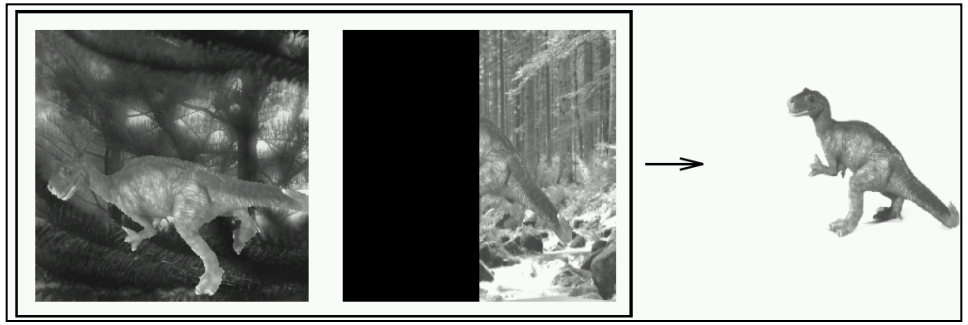
http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

Wide baseline stereo



[Image from T. Tuytelaars ECCV 2006 tutorial]

Recognition of specific objects, scenes



Schmid and Mohr 1997

Sivic and Zisserman, 2003

Rothganger et al. 2003

Lowe 2002

Applications of local invariant features

- Wide baseline stereo
- Motion tracking
- **Panoramas**
- Mobile robot navigation
- 3D reconstruction
- Recognition
- …

What we will learn today?

- Scale invariant region selection
  - Automatic scale selection
  - Difference-of-Gaussian (DoG) detector
- SIFT: an image region descriptor
- HoG: another image descriptor
- Application: Panorama

Histogram of Oriented Gradients

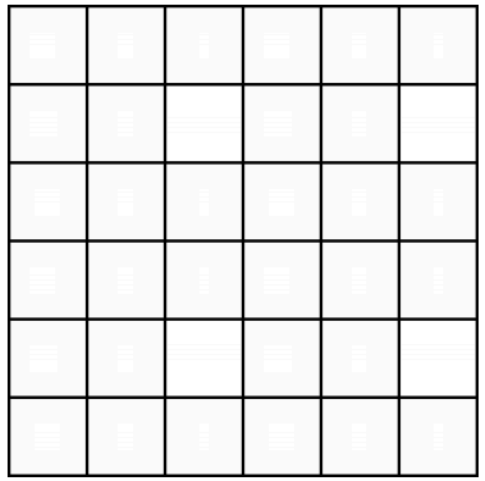
- Find robust feature set that allows object form to be discriminated.
- Challenges
  - Wide range of pose and large variations in appearances
  - Cluttered backgrounds under different illumination
  - “Speed” for mobile vision
- References
  - [1] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In CVPR, pages 886-893, 2005
  - [2] Chandrasekhar et al. CHoG: Compressed Histogram of Gradients - A low bit rate feature descriptor, CVPR 2009

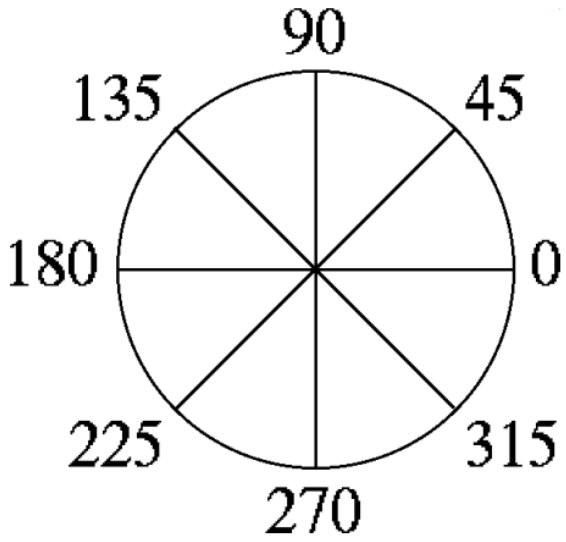
Histogram of Oriented Gradients

Local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions.

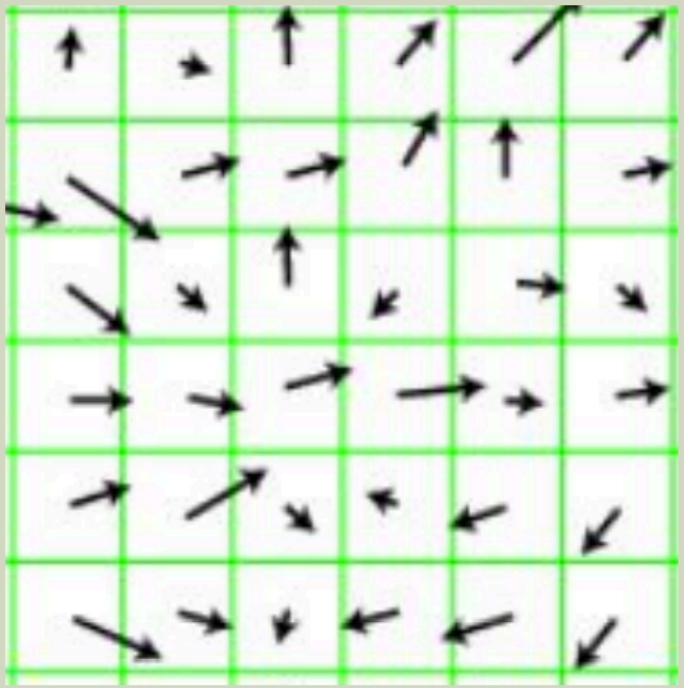
Histogram of Oriented Gradients

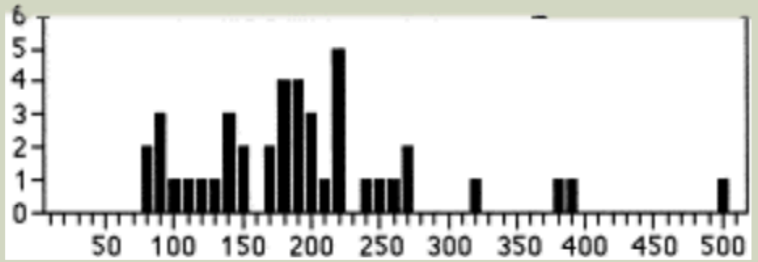
- Dividing the image window into small spatial regions (cells)
- Cells can be either rectangle or radial.
- Each cell accumulating a weighted local 1-D histogram of gradient directions over the pixels of the cell.



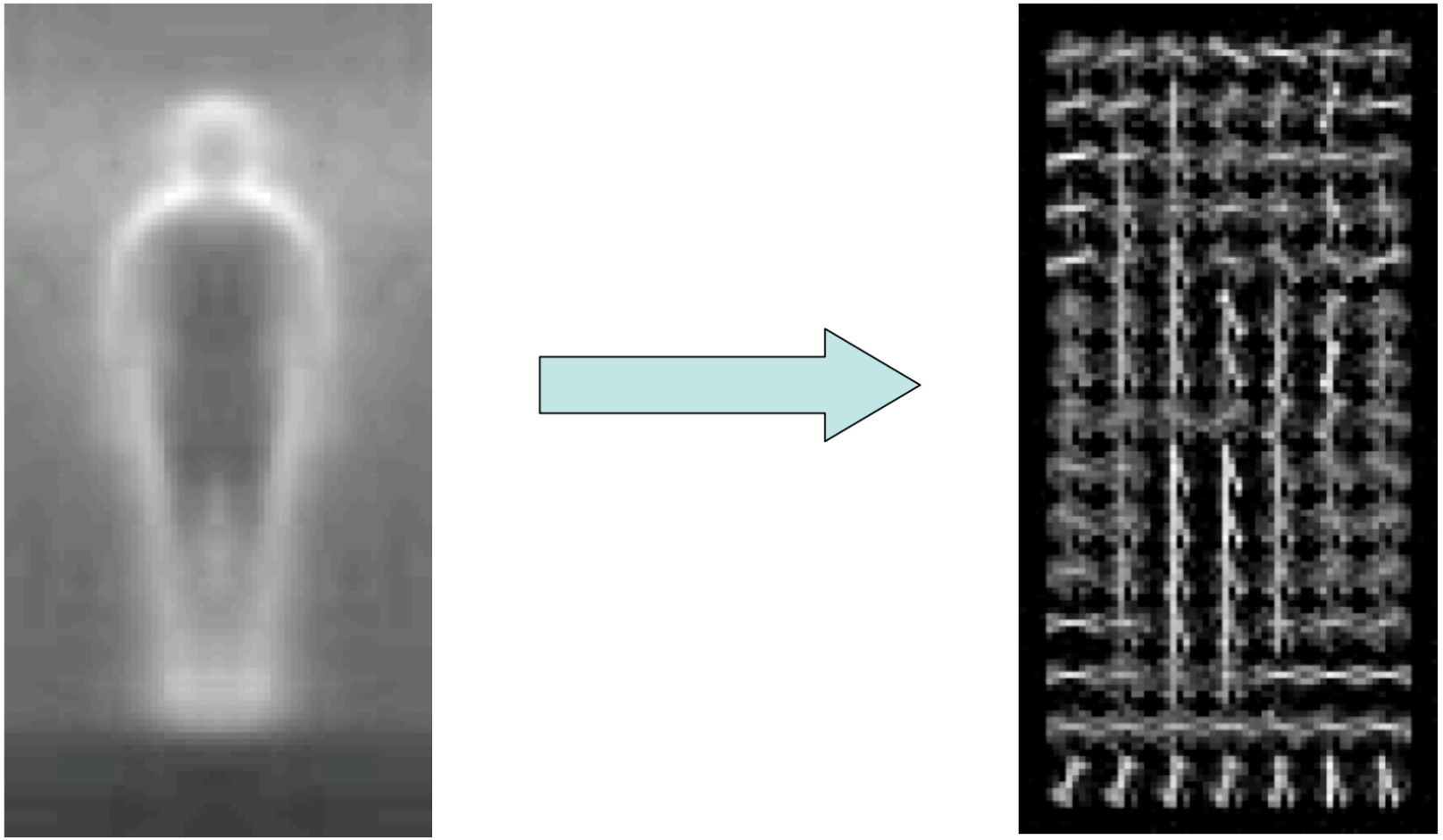


Histogram of Oriented Gradients



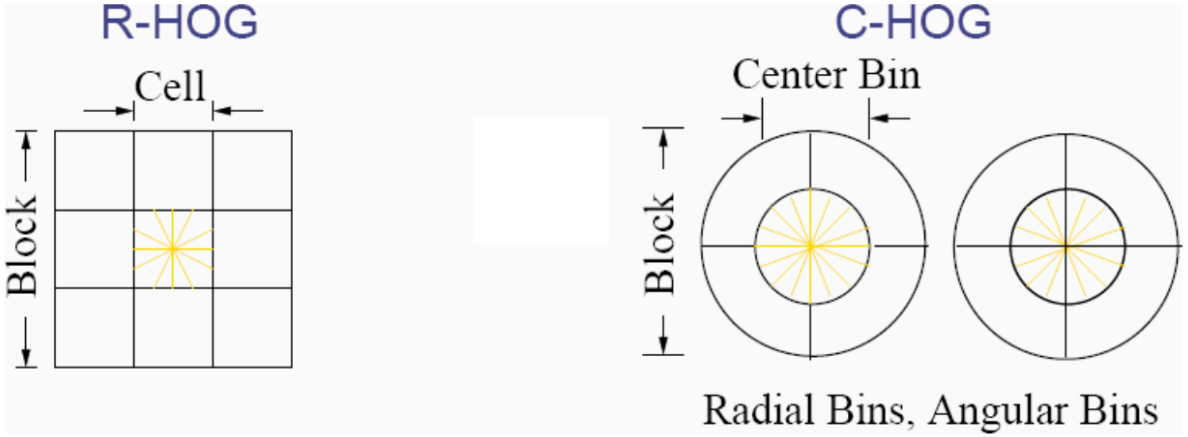


Histogram of Oriented Gradients

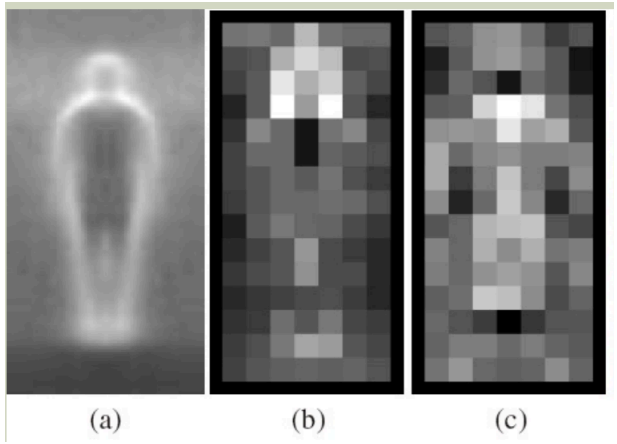


Normalization

- For better invariance to illumination and shadowing. it is useful to contrast-normalize the local responses before using them.
- Accumulate local histogram “energy” over a larger regions (“blocks”) to normalize all of the cells in the block.

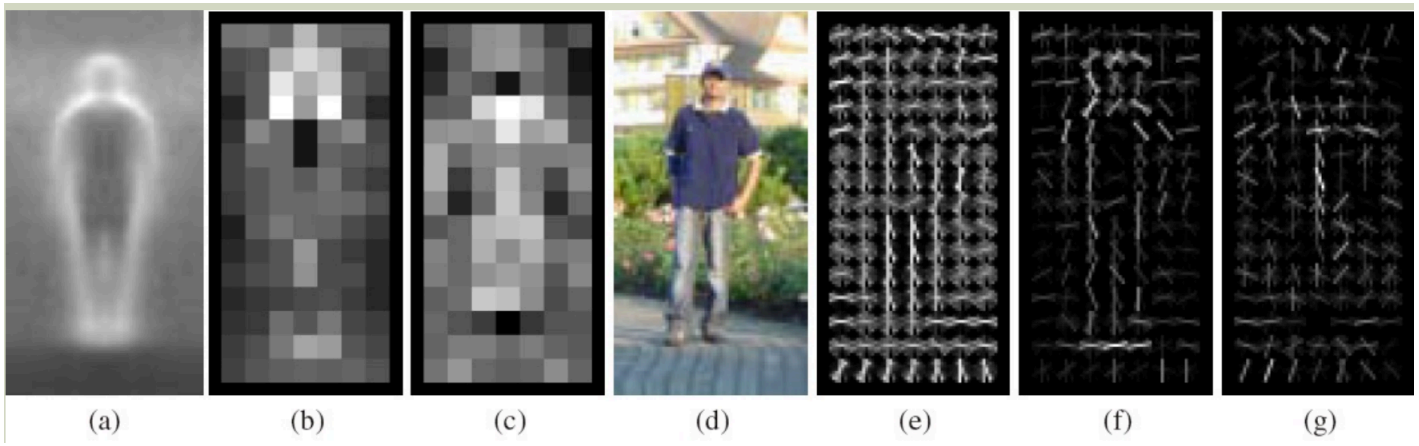


Visualizing HoG



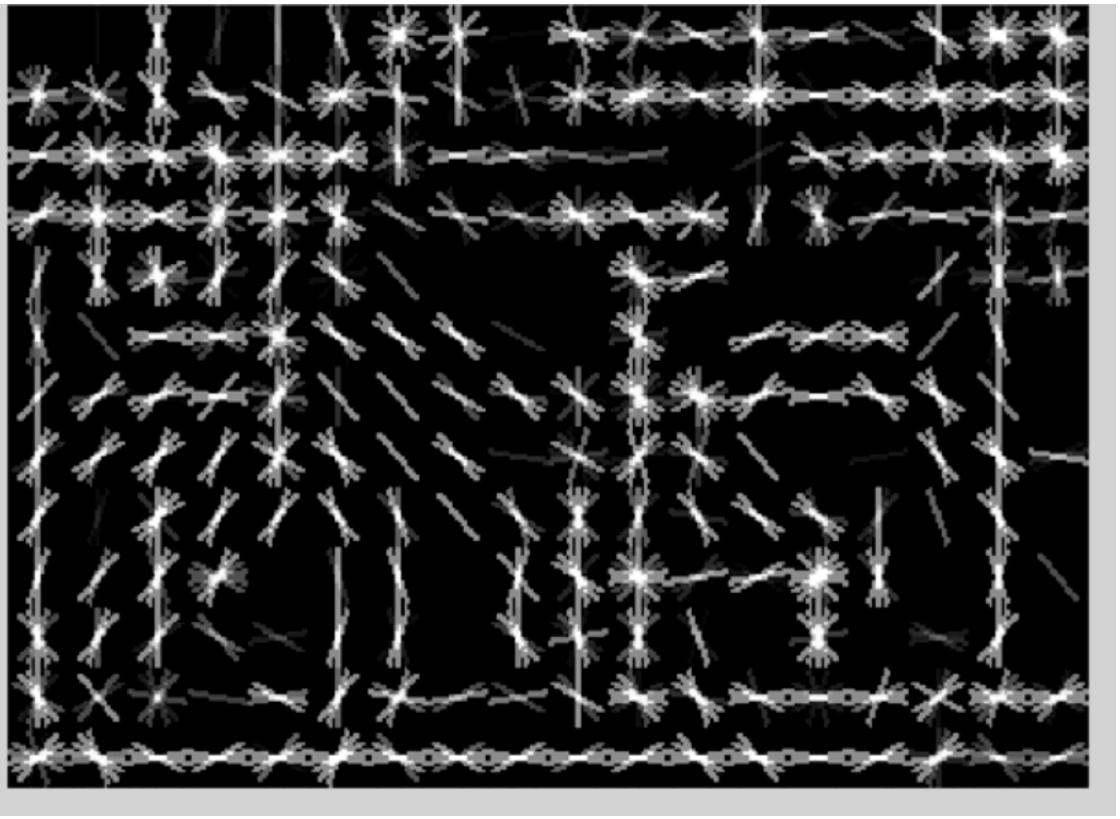
a. Average gradient over positive examples
b. Maximum positive weight in each block
c. Maximum negative weight in each block
d. A test image
e. It's R-HOG descriptor
f. R-HOG descriptor weighted by positive weights
g. R-HOG descriptor weighted by negative weights

Visualizing HoG



a. Average gradient over positive examples
b. Maximum positive weight in each block
c. Maximum negative weight in each block
d. A test image
e. It's R-HOG descriptor
f. R-HOG descriptor weighted by positive weights
g. R-HOG descriptor weighted by negative weights

Visualizing HoG



Difference between HoG and SIFT

- HoG is usually used to describe entire images. SIFT is used for key point matching
- SIFT histrograms are oriented towards the dominant gradient. HoG is not.
- HoG gradients are normalized using neighborhood bins.
- SIFT descriptors use varying scales to compute multiple descriptors.

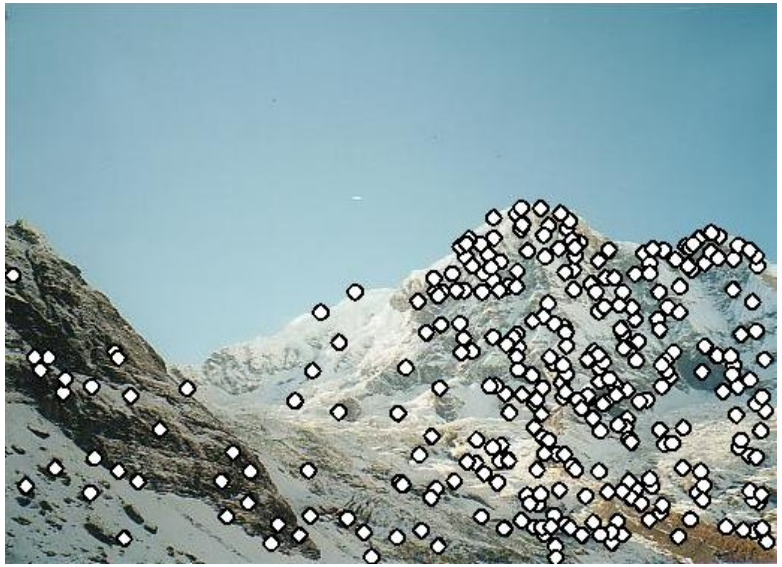
What we will learn today?

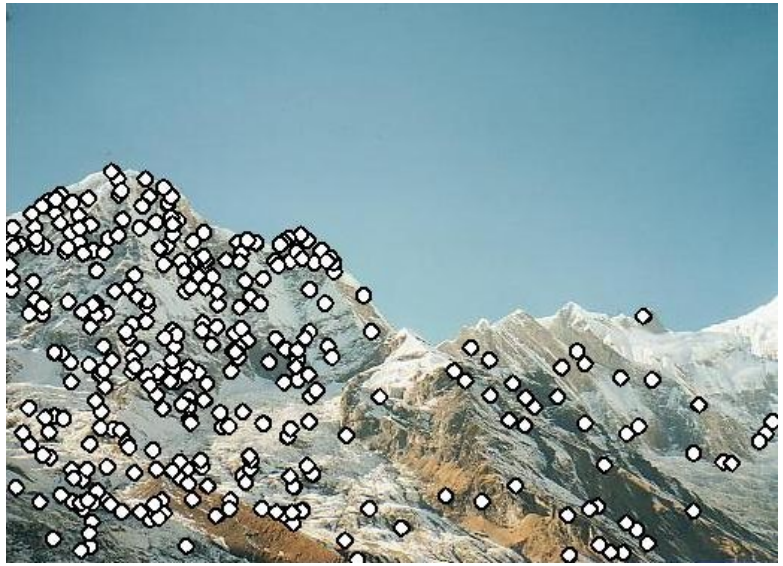
- Scale invariant region selection
  - Automatic scale selection
  - Difference-of-Gaussian (DoG) detector
- SIFT: an image region descriptor
- HoG: another image descriptor
- Application: Panorama

Application: Image Stitching

Slide credit: Darya Frolova, Denis Simakov

Application: Image Stitching

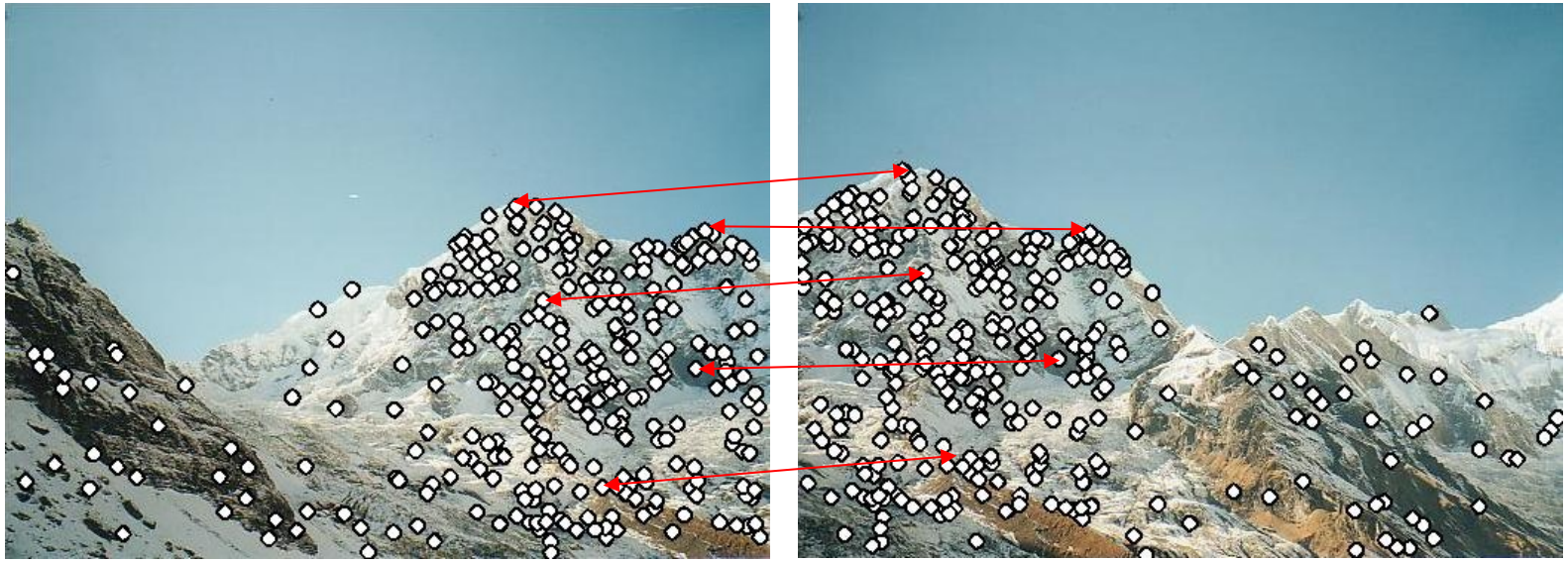




- Procedure:
  - Detect feature points in both images

Slide credit: Darya Frolova, Denis Simakov

Application: Image Stitching



- Procedure:
  - Detect feature points in both images
  - Find corresponding pairs

Slide credit: Darya Frolova, Denis Simakov

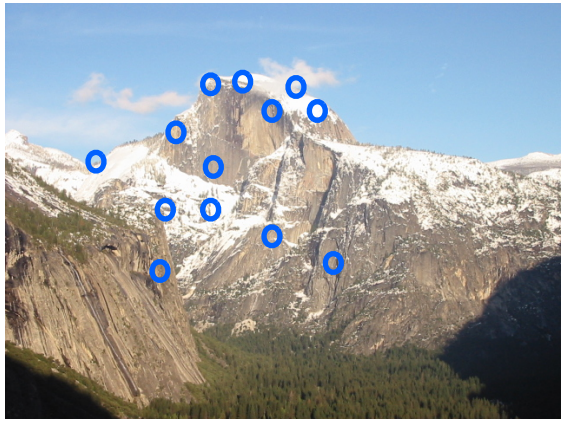
Application: Image Stitching

- Procedure:
  - Detect feature points in both images
  - Find corresponding pairs
  - Use these pairs to align the images

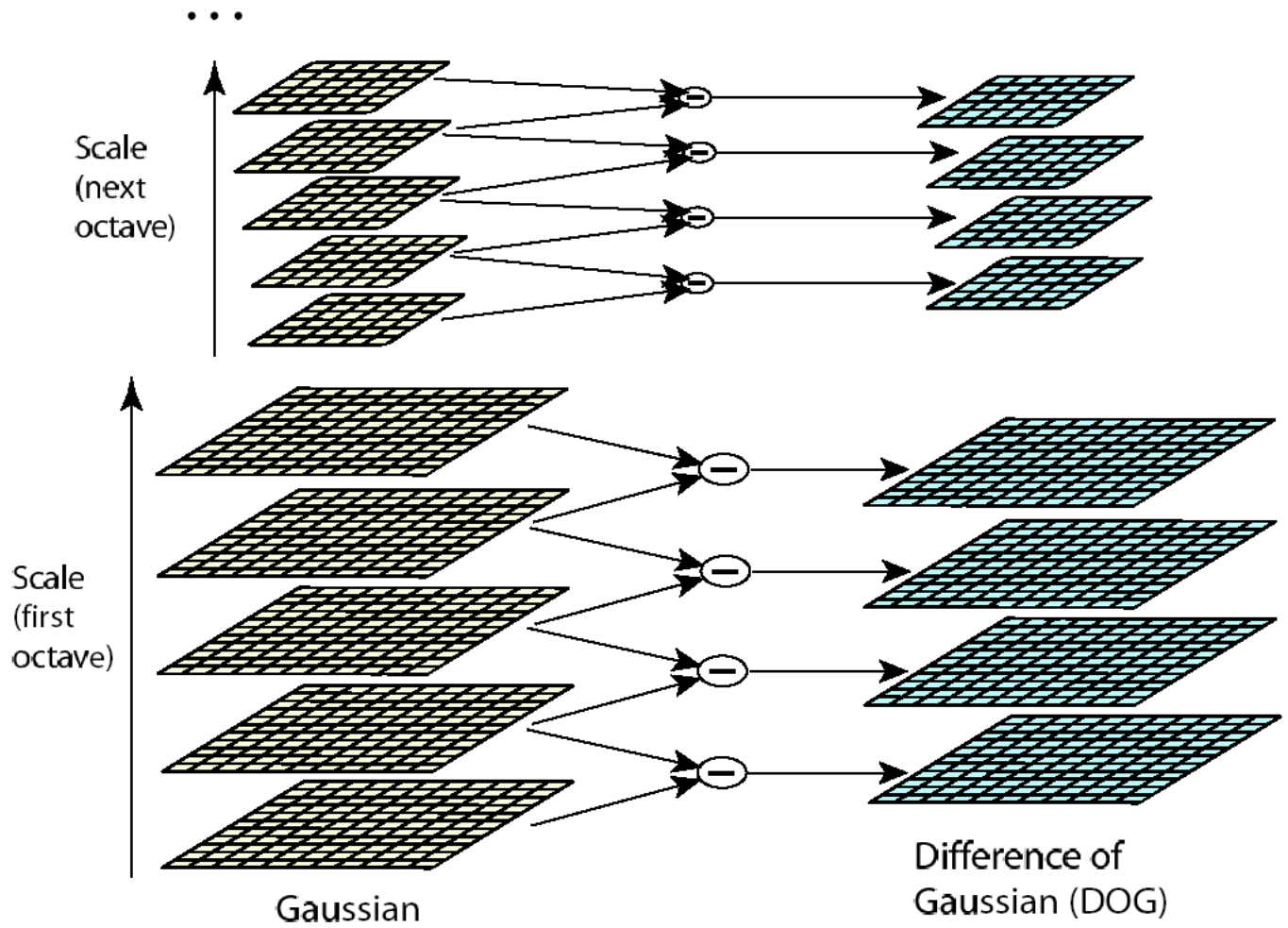
Slide credit: Darya Frolova, Denis Simakov

Main Flow

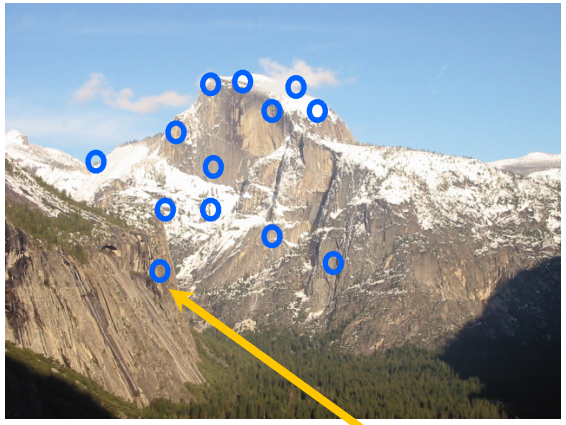
- Detect key points



Detect Key Points



Main Flow



- Detect k
- Build the SIFT descriptors

$(a_1, a_2, \ldots, a_{128})$

$(v_1, v_2, \ldots, v_{128})$

Build the SIFT Descriptors

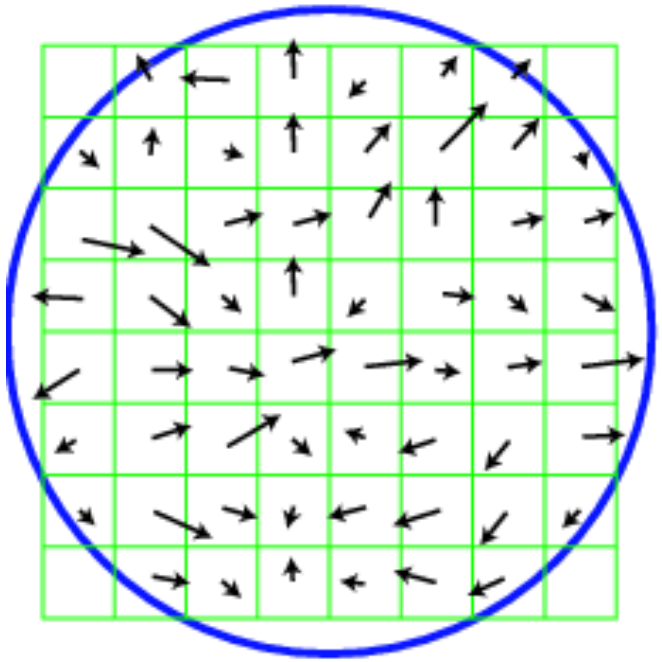
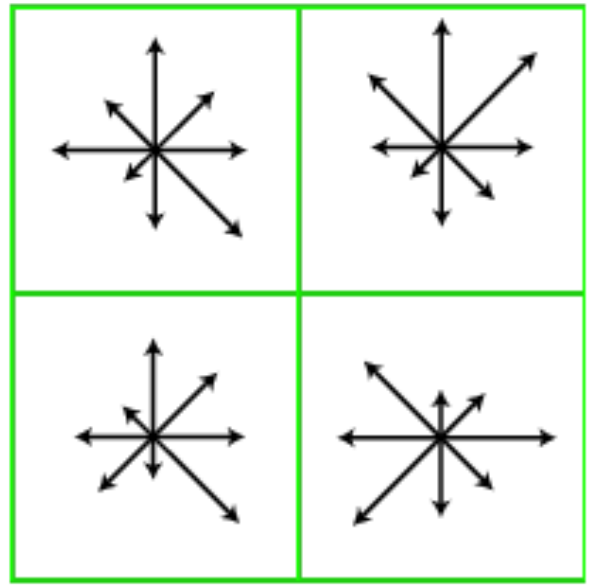
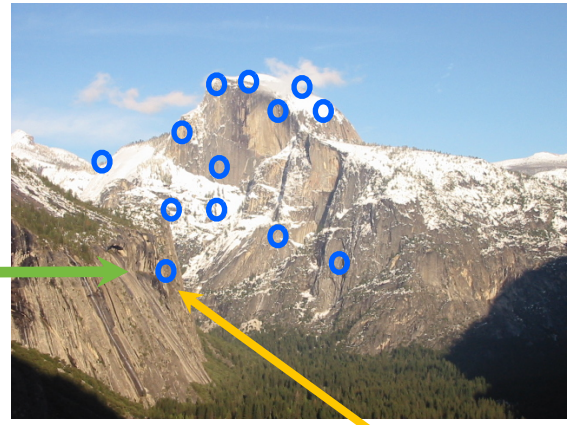


Image gradients



Keypoint descriptor

Main Flow



- Detect k
- Build the SIFT descriptors

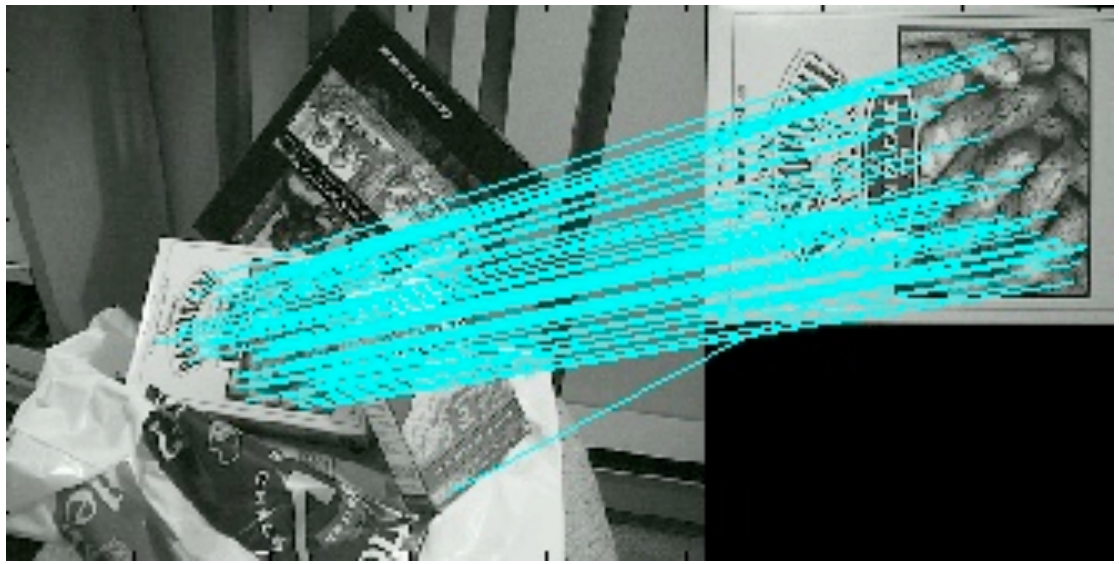
$(u_1, u_2, \ldots, u_{128})$

$(v_1, v_2, \ldots, v_{128})$

- Match SIFT descriptors

Match SIFT Descriptors

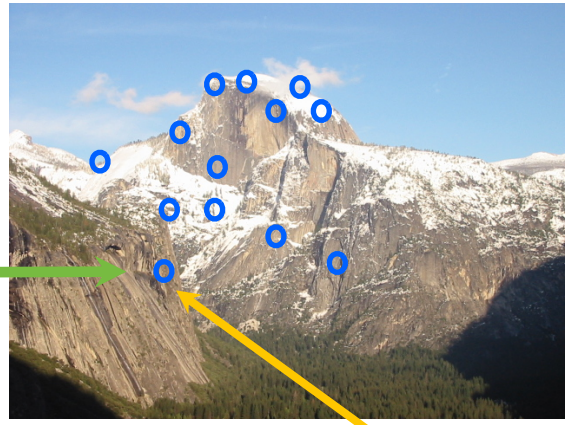
- Euclidean distance between descriptors



Skeleton Code

- Match SIFT descriptors (6 lines of code)
  - Input: D1, D2, thresh (default 0.7)
  - Output: match [D1's index, D2's index]
  - Try to use *one* *for* loop

Main Flow



$(u_1, u_2, \ldots, u_{128})$

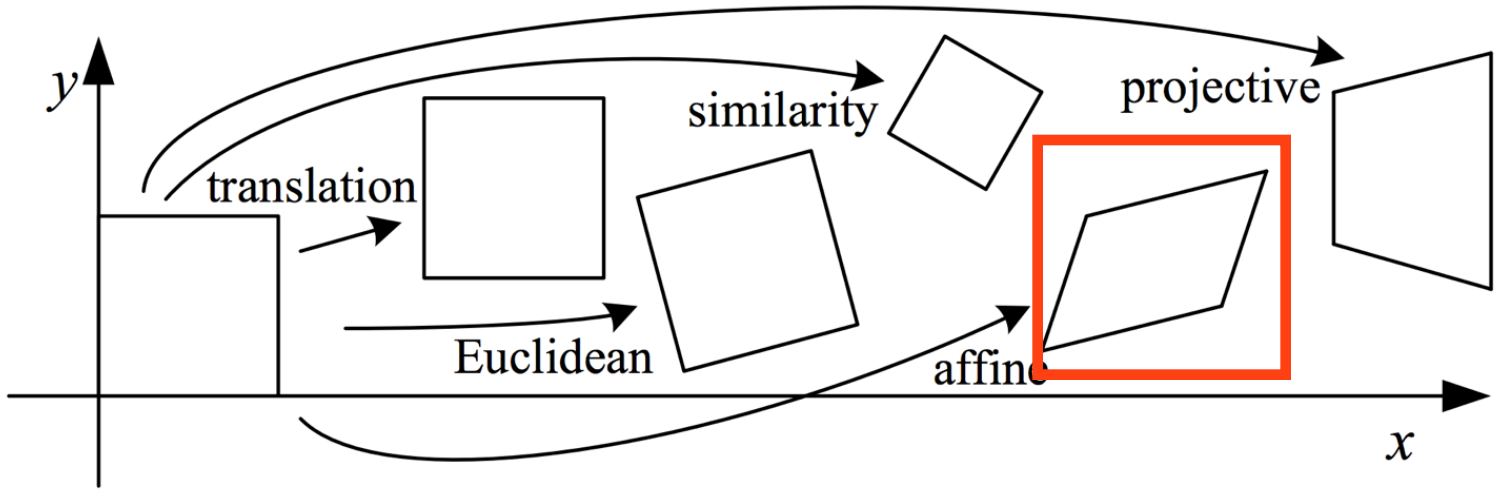
$(v_1, v_2, \ldots, v_{128})$

- Detect k
- Build the SIFT descriptors
- Match SIFT descriptors
- Fitting the transformation

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

Fitting the transformation

- 2D transformations



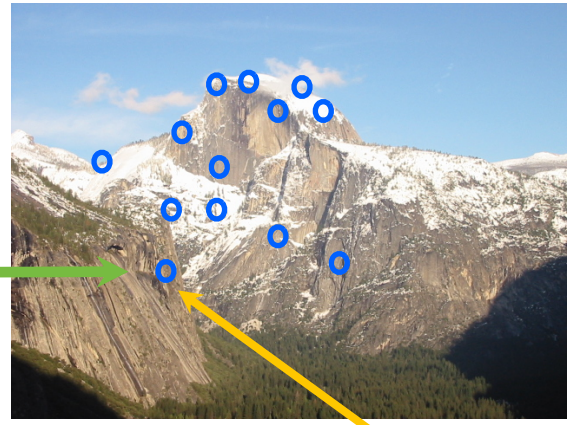
Skeleton Code

- Fit the transformation matrix

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

- Six variables
  - each point give two equations
  - at least three points
- Least squares

Main Flow



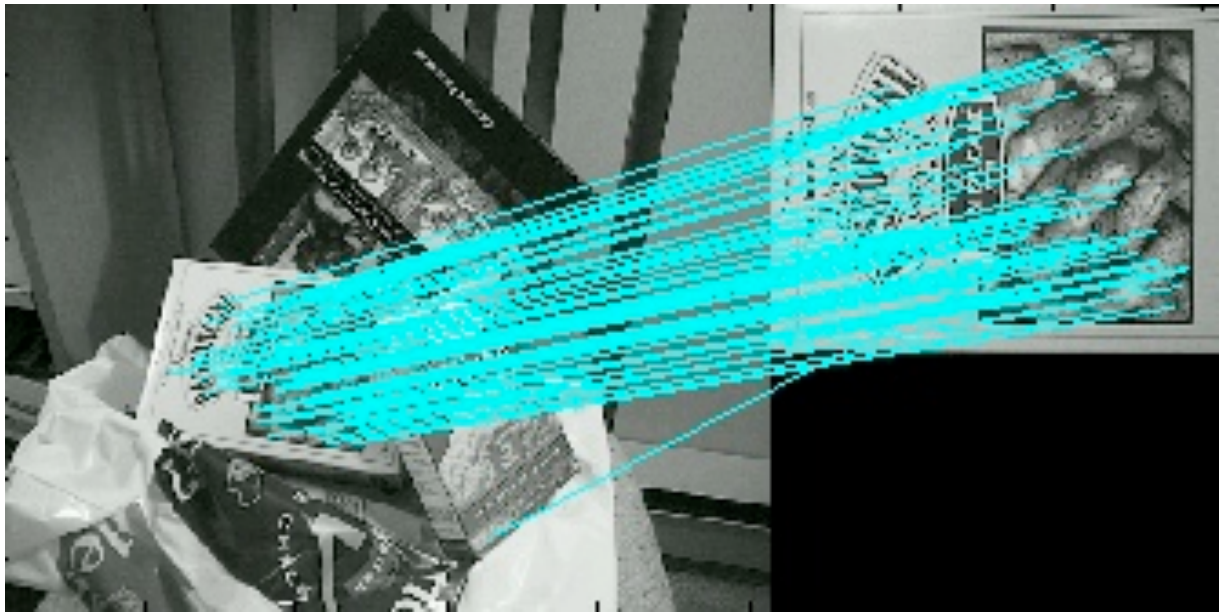
$(u_1, u_2, \ldots, u_{128})$

$(v_1, v_2, \ldots, v_{128})$

- Detect k
- Build the SIFT descriptors
- Match SIFT descriptors
- Fitting the transformation
- RANSAC

RANSAC

- A further refinement of matches

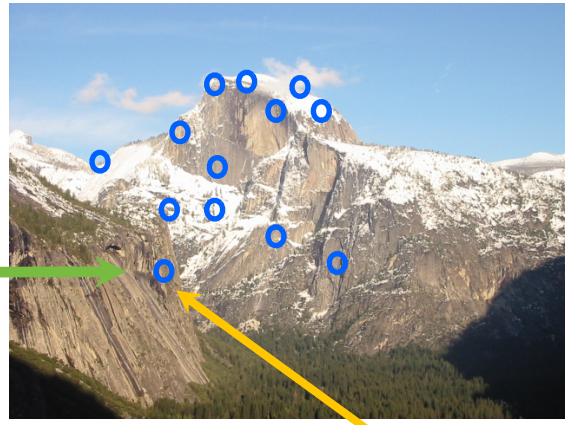


Skeleton Code

- RANSAC
  - ComputeError

$$\left\| \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} - H \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \right\|_2$$

Main Flow

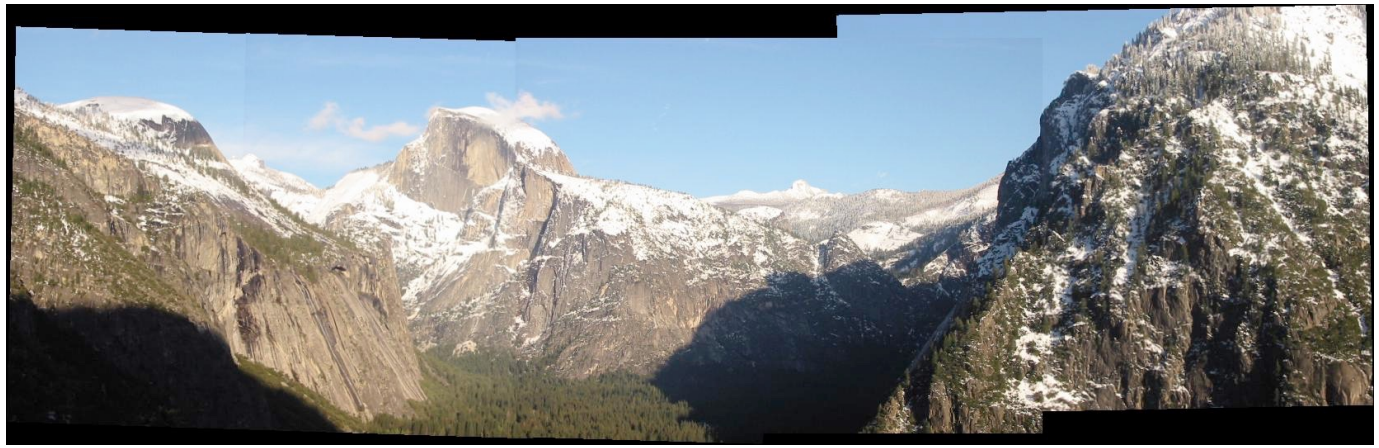


$(u_1, u_2, \ldots, u_{128})$

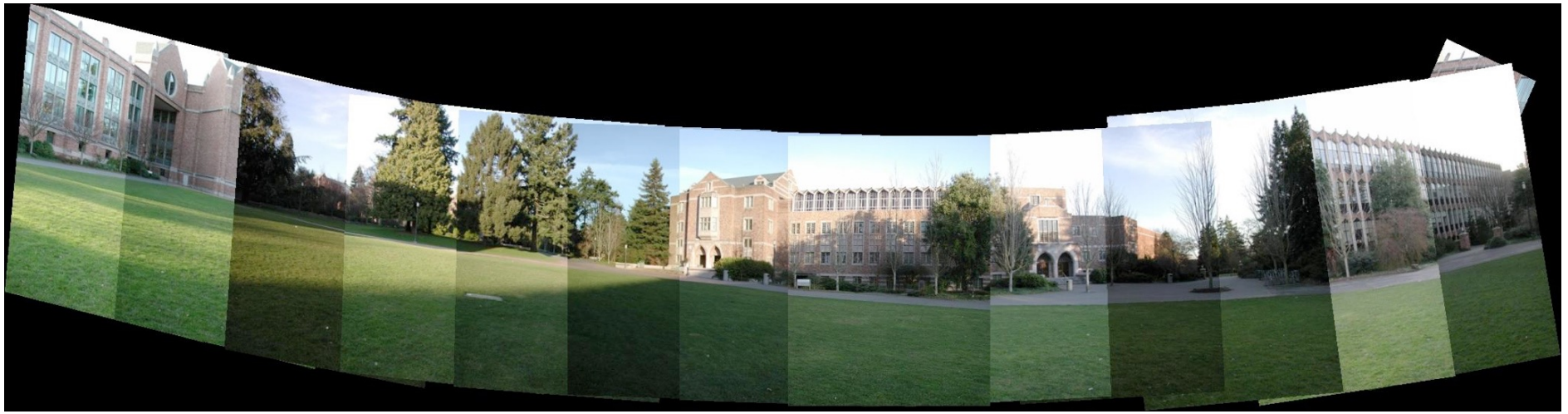
$(v_1, v_2, \ldots, v_{128})$

- Detect key points
- Build the SIFT descriptors
- Match SIFT descriptors
- Fitting the transformation
- RANSAC

Results



Results



What we have learned this week?

- Scale invariant region selection
  - Automatic scale selection
  - Difference-of-Gaussian (DoG) detector
- SIFT: an image region descriptor
- Application: Panorama

Some background reading: R. Szeliski, Ch 4.1.1; David Lowe, IJCV 2004