



MIT CSAIL

6.869: Advances in Computer Vision

Antonio Torralba, 2013

MIT
COMPUTER
VISION

Lecture 13

Image features

With some slides from
Darya Frolova, Denis Simakov, David Lowe, Bill Freeman

Finding the “same” thing across images

Categories Find a bottle:



Can't do
unless you do not
care about few errors...

Instances Find these two objects



Can nail it



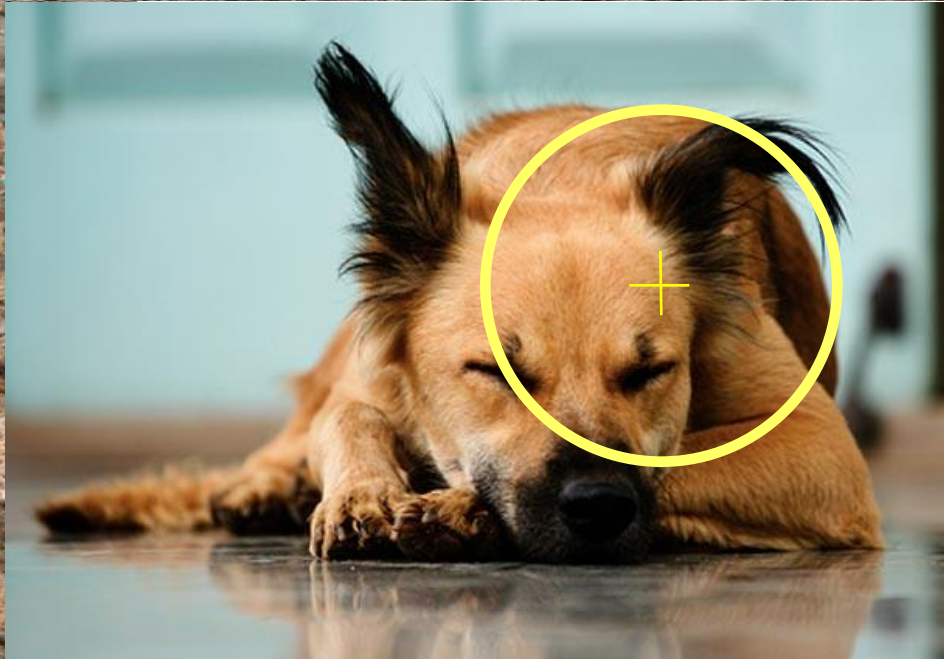
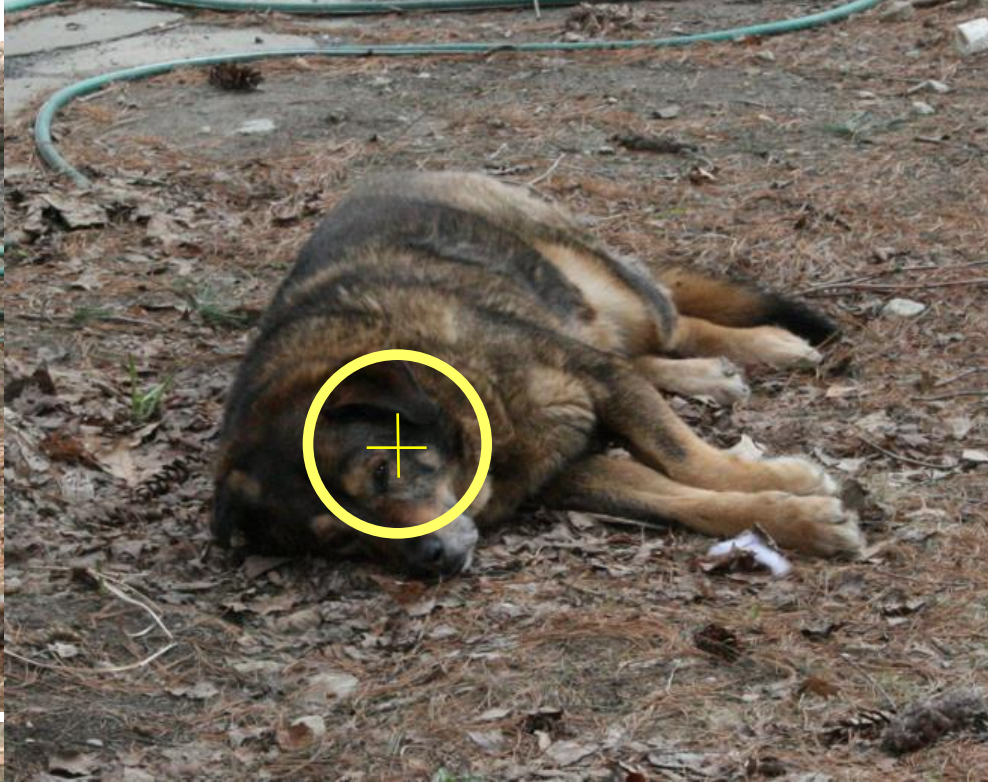
Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original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



But where is that point?

Building a Panorama



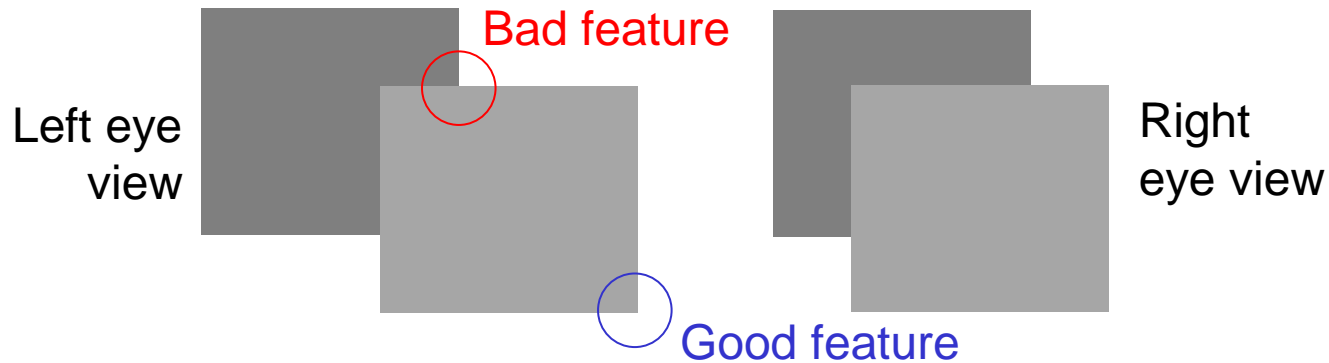


Uses for feature point detectors and descriptors in computer vision and graphics.

- Image alignment and building panoramas
- 3D reconstruction
- Motion tracking
- Object recognition
- Indexing and database retrieval
- Robot navigation
- ... other

Selecting Good Features

- What’s a “good feature”?
 - Satisfies brightness constancy—looks the same in both images
 - Has sufficient texture variation
 - Does not have too much texture variation
 - Corresponds to a “real” surface patch—see below:



- Does not deform too much over time

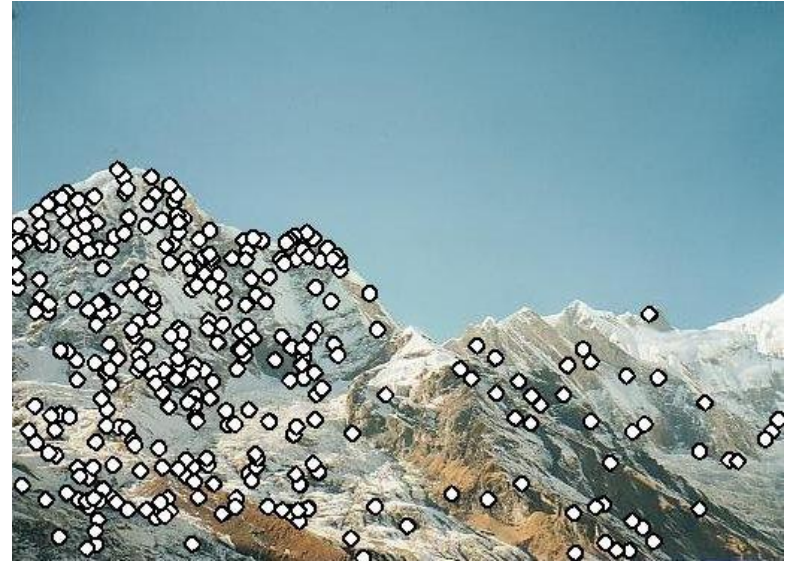
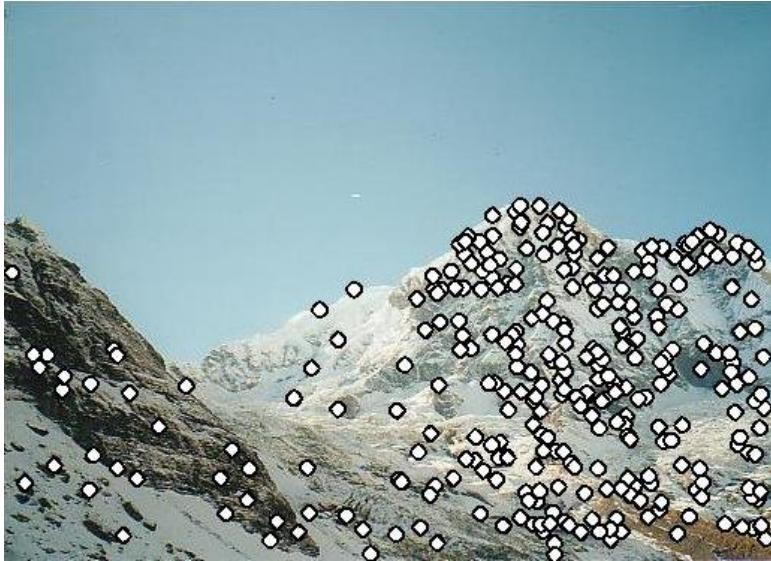
How do we build a panorama?

- We need to match (align) images



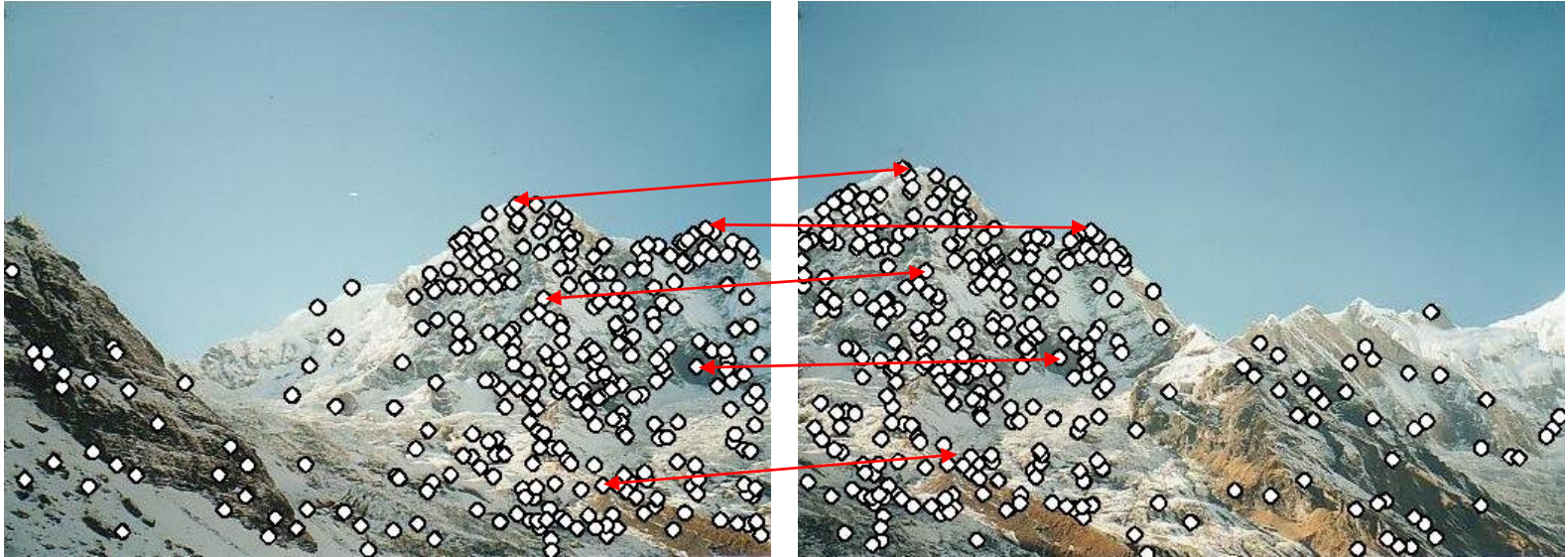
Matching with Features

- Detect feature points in both images



Matching with Features

- Detect feature points in both images
- Find corresponding pairs



Matching with Features

- Detect feature points in both images
- Find corresponding pairs
- Use these matching pairs to align images - the required mapping is called a homography.



Matching with Features

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

counter-example:

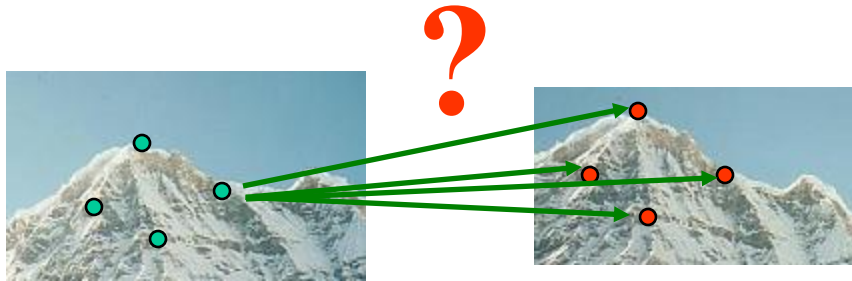


no chance to match!

We need a repeatable detector

Matching with Features

- Problem 2:
 - For each point correctly recognize the corresponding one



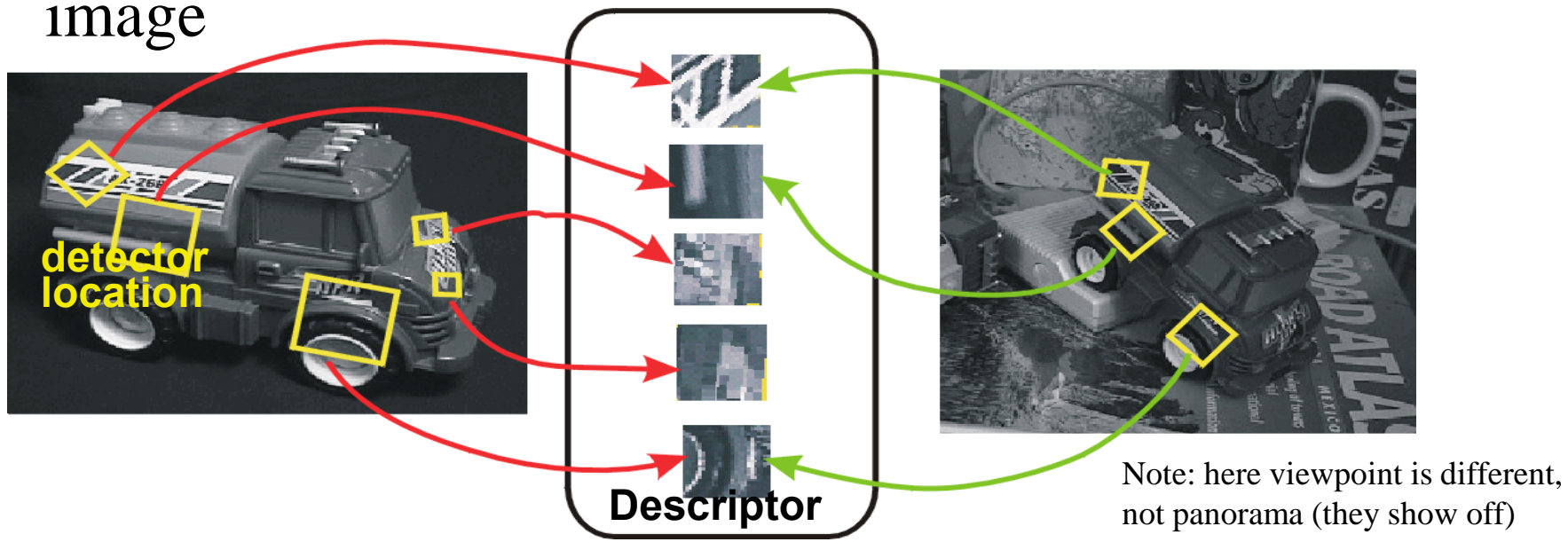
We need a reliable and distinctive **descriptor**

Building a Panorama



Preview

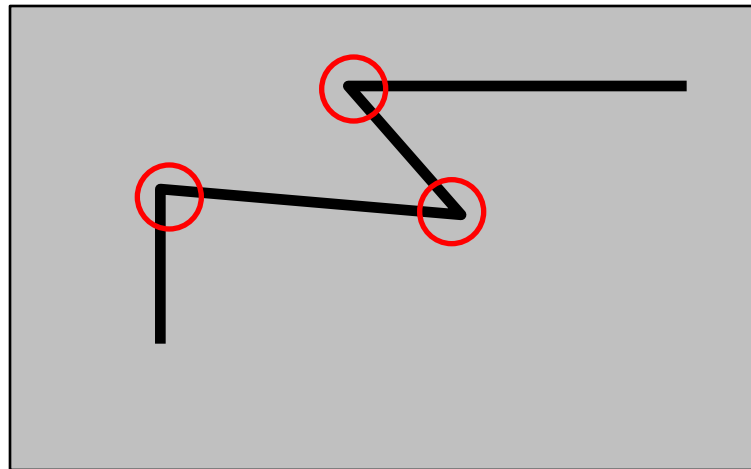
- **Detector:** detect same scene points independently in both images
- **Descriptor:** encode local neighboring window
 - Note how scale & rotation of window are the same in both image (but computed independently)
- **Correspondence:** find most similar descriptor in other image



Outline

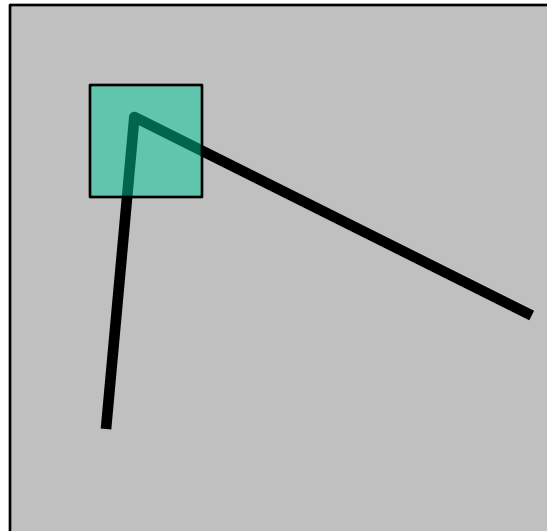
- Feature point detection
 - Harris corner detector
 - finding a characteristic scale: DoG or Laplacian of Gaussian
- Local image description
 - SIFT features

Harris corner detector

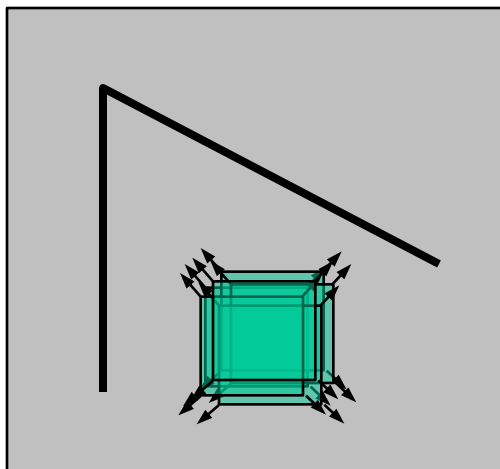


The Basic Idea

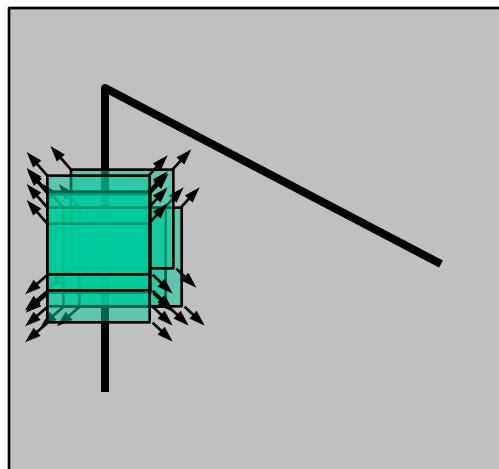
- We should easily localize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in pixels intensities in window
 - makes location precisely define



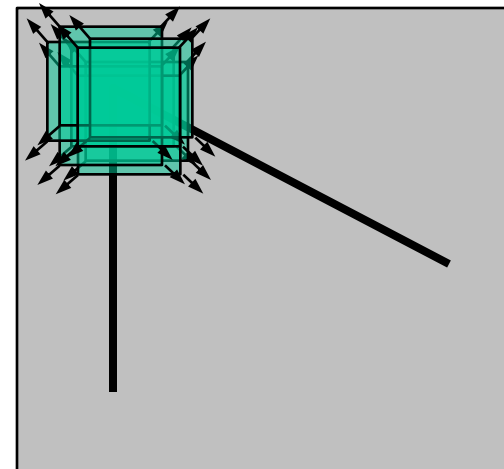
Corner Detector: Basic Idea



“flat” region:
no change in all
directions



“edge”:
no change along
the edge direction



“corner”:
significant change
in all directions

Harris Detector: Mathematics

Window-averaged squared change of intensity induced by shifting the image data by $[u, v]$:

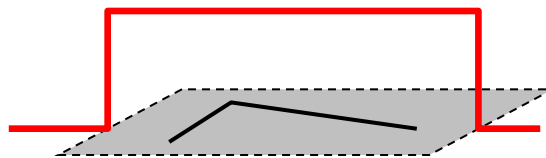
$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

Window function

Shifted intensity

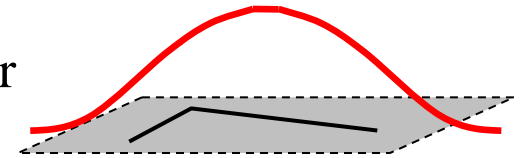
Intensity

Window function $w(x, y) =$



1 in window, 0 outside

or



Gaussian

Taylor series approximation to shifted image gives quadratic form for error as function of image shifts.

$$\begin{aligned} E(u, v) &\approx \sum_{x,y} w(x, y) [I(x, y) + uI_x + vI_y - I(x, y)]^2 \\ &= \sum_{x,y} w(x, y) [uI_x + vI_y]^2 \\ &= (u \quad v) \sum_{x,y} w(x, y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \end{aligned}$$

Harris Detector: Mathematics

Expanding $I(x,y)$ in a Taylor series expansion, we have, for small shifts $[u, v]$, a *quadratic* approximation to the error surface between a patch and itself, shifted by $[u, v]$:

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

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

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

M is often called structure tensor

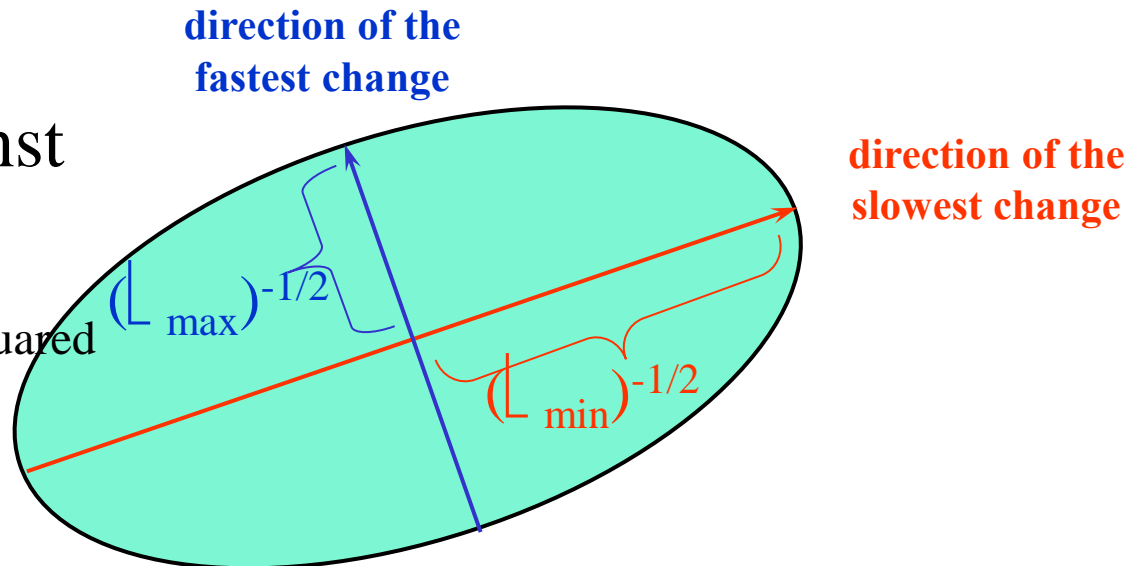
Harris Detector: Mathematics

Intensity change in shifting window: eigenvalue analysis

$$E(u, v) \cong [u, v] M \begin{bmatrix} u \\ v \end{bmatrix} \quad \lambda_1, \lambda_2 - \text{eigenvalues of } M$$

Ellipse $E(u, v) = \text{const}$

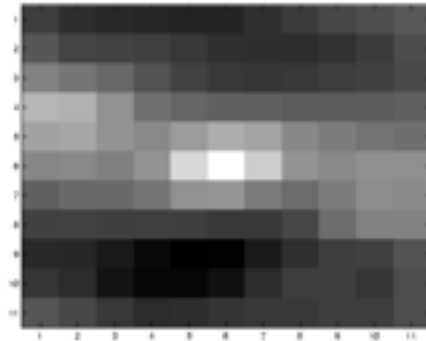
Iso-contour of the squared error, $E(u, v)$



Selecting Good Features

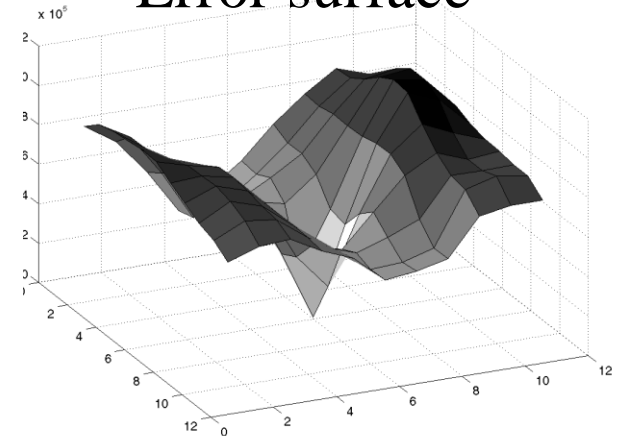


Image patch



12×10^5

Error surface



L_1 and L_2 are large

Selecting Good Features

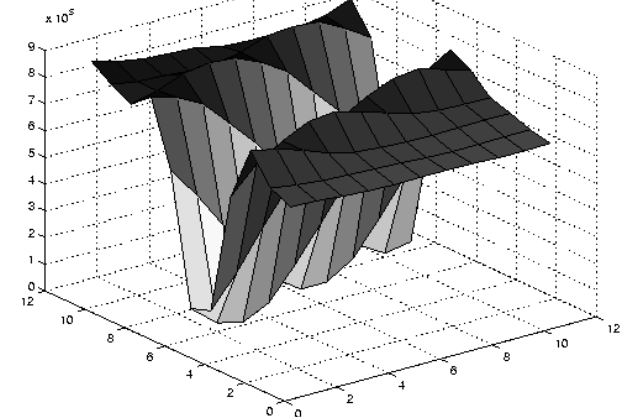


Image patch



9×10^5

Error surface



large L_1 , small L_2

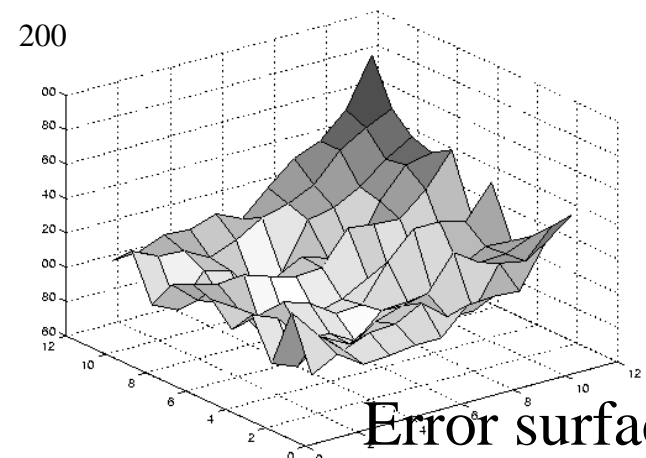
Selecting Good Features



Image patch



(contrast auto-scaled)

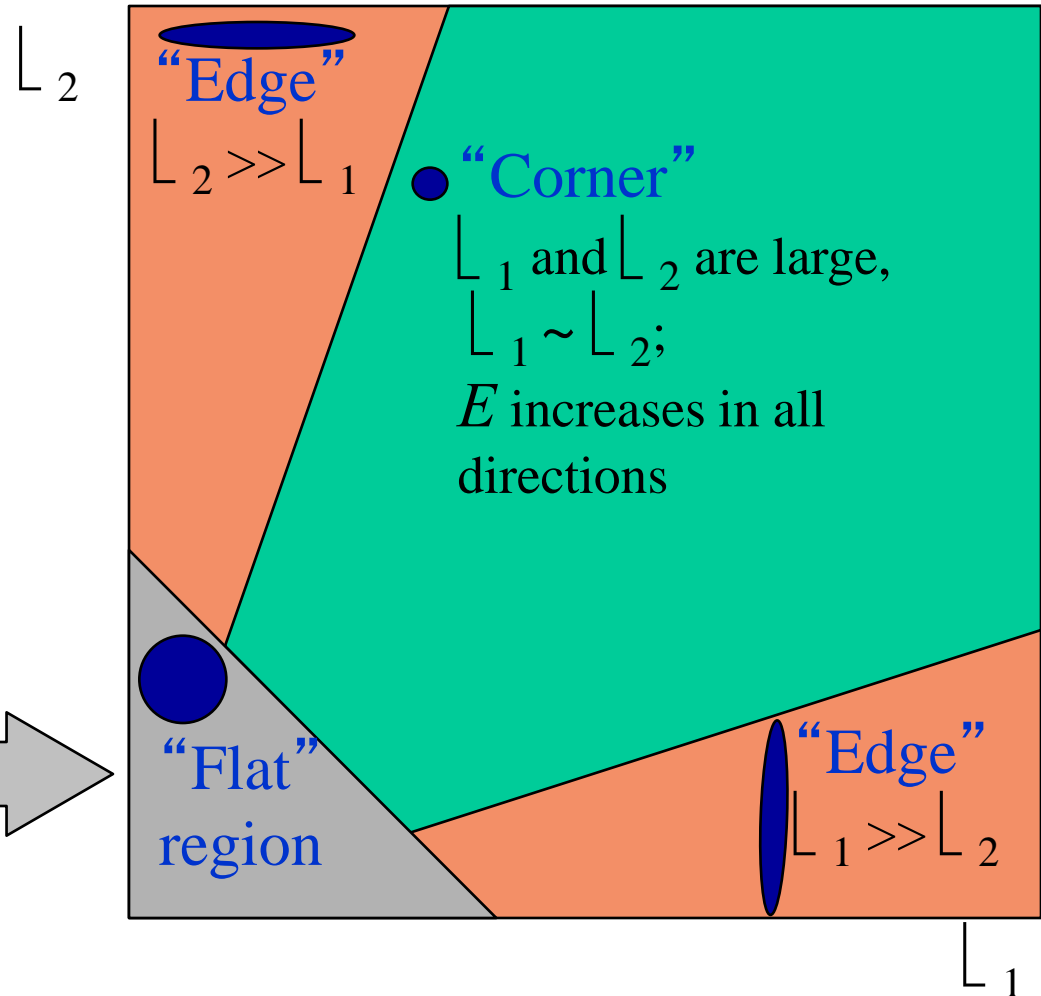
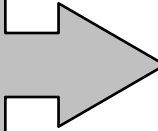


(vertical scale exaggerated relative to previous plots)
small L_1 , small L_2

Harris Detector: Mathematics

Classification of image points using eigenvalues of M :

L_1 and L_2 are small;
 E is almost constant
in all directions



Harris Detector: Mathematics

Measure of corner response:

$$R = \det M - k (\text{trace } M)^2$$

$$\det M = \lambda_1 \lambda_2$$

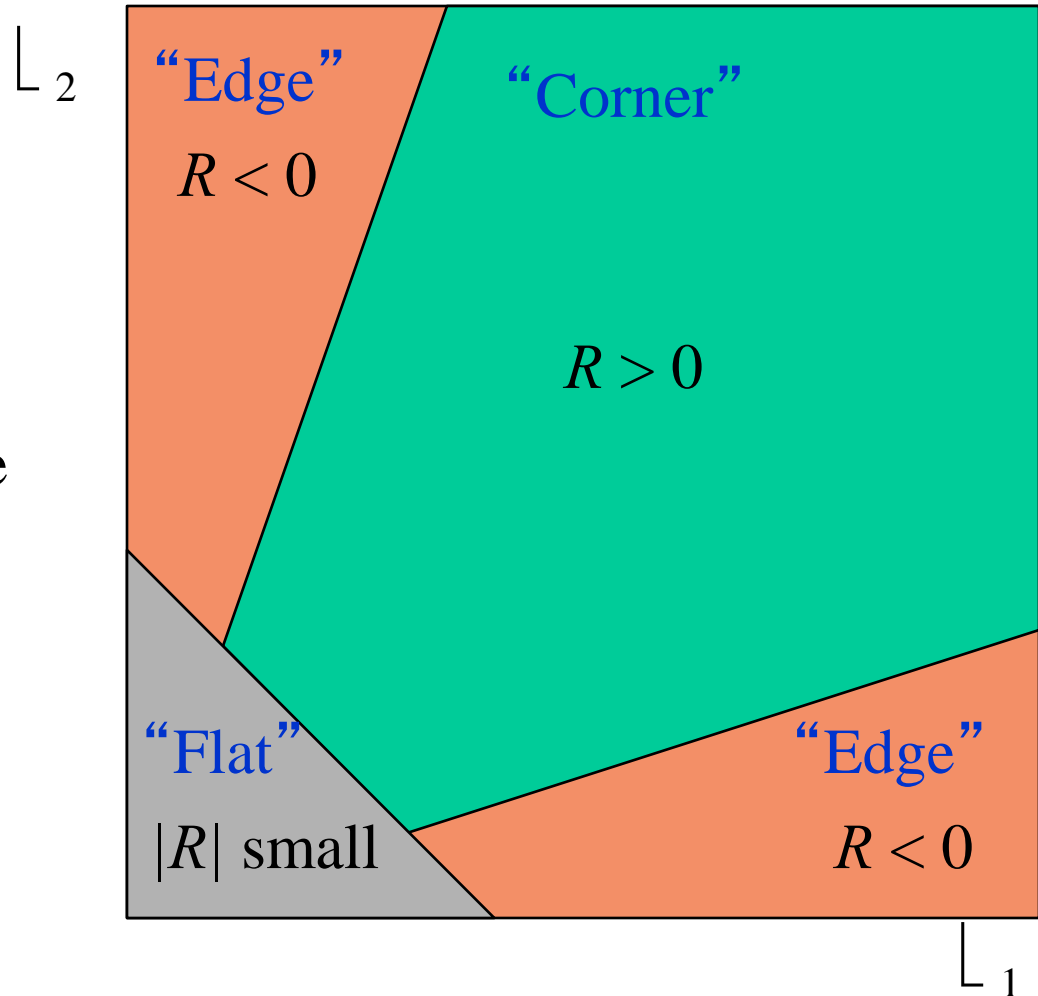
$$\text{trace } M = \lambda_1 + \lambda_2$$

(k – empirical constant, $k = 0.04-0.06$)

(Shi-Tomasi variation: use $\min(\lambda_1, \lambda_2)$ instead of R)

Harris Detector: Mathematics

- R depends only on eigenvalues of M
- R is large for a **corner**
- R is negative with large magnitude for an **edge**
- $|R|$ is small for a **flat** region



Harris Detector

- The Algorithm:
 - Find points with large corner response function R ($R > \text{threshold}$)
 - Take the points of *local maxima* of R

Harris corner detector algorithm

- Compute image gradients I_x I_y for all pixels
- For each pixel

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

by looping over neighbors x,y

- compute
$$R = \det M - k (\text{trace } M)^2$$

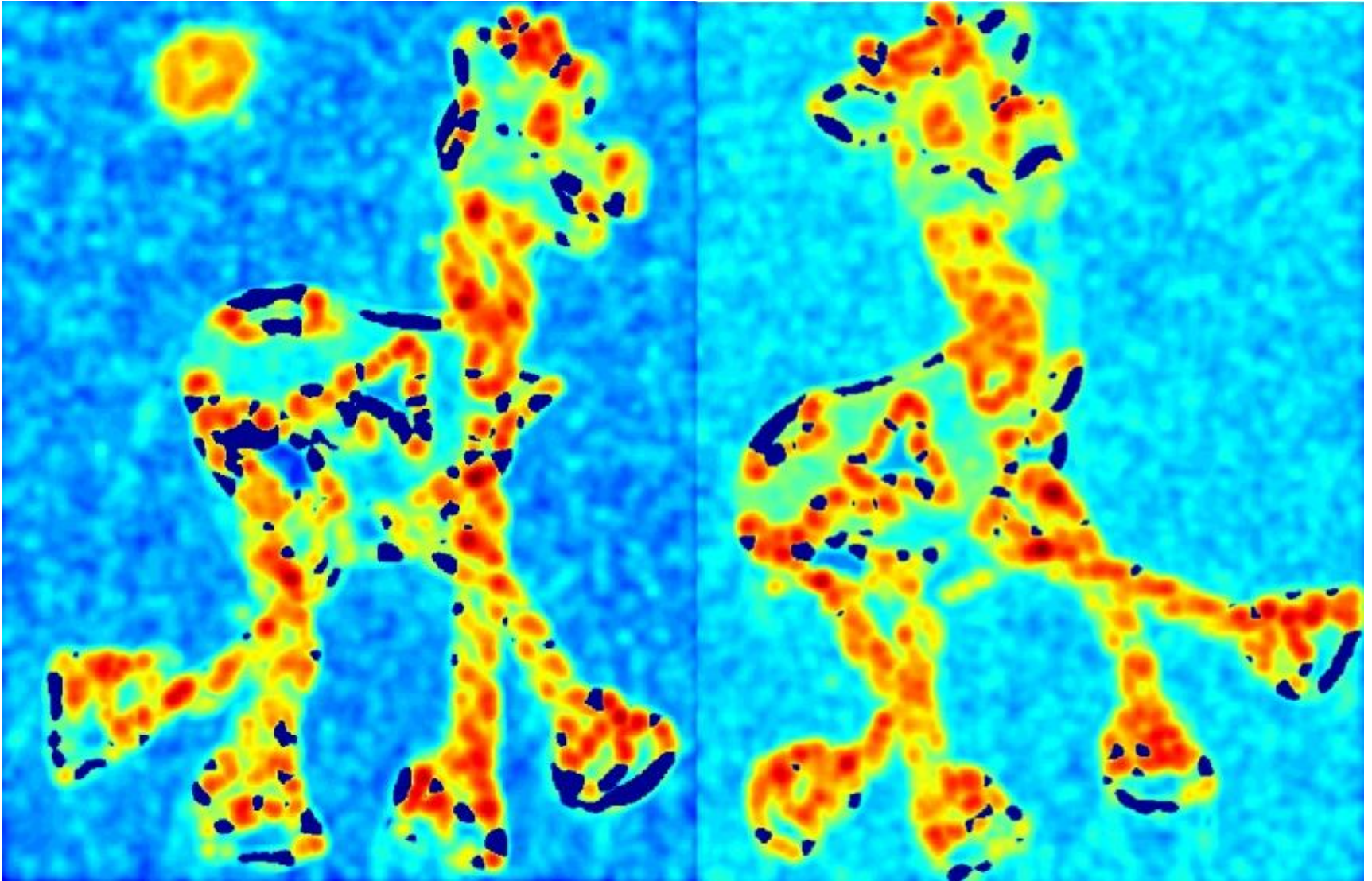
- Find points with large corner response function R ($R > \text{threshold}$)
- Take the points of locally maximum R as the detected feature points (ie, pixels where R is bigger³² than for all the 4 or 8 neighbors).

Harris Detector: Workflow



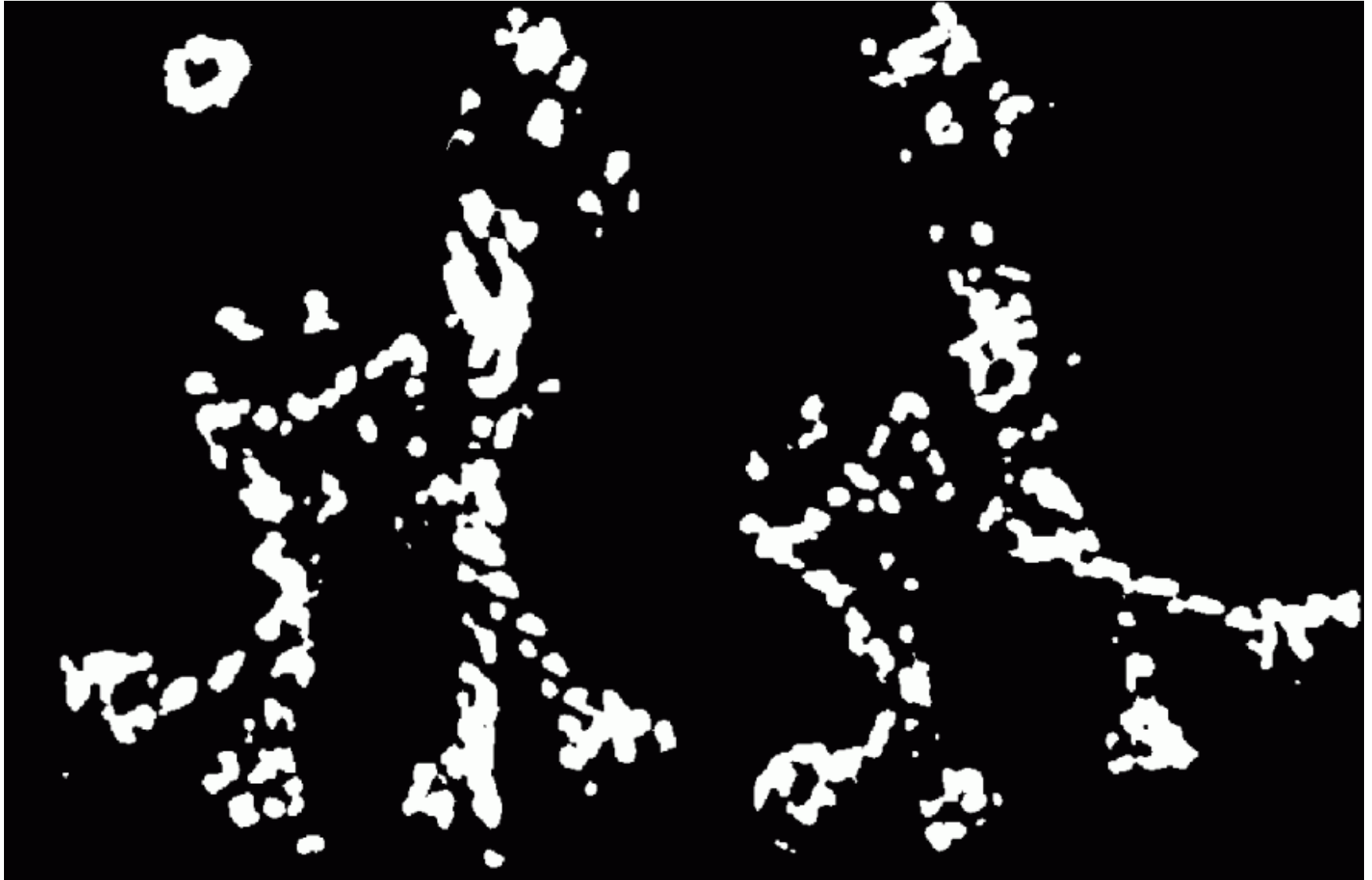
Harris Detector: Workflow

Compute corner response R



Harris Detector: Workflow

Find points with large corner response: $R > \text{threshold}$



Harris Detector: Workflow

Take only the points of local maxima of R



Harris Detector: Workflow



Analysis of Harris corner detector invariance properties

- Geometry
 - rotation
 - scale
- Photometry
 - intensity change

Evaluation plots are from this paper



International Journal of Computer Vision 37(2), 151–172, 2000
© 2000 Kluwer Academic Publishers. Manufactured in The Netherlands.

Evaluation of Interest Point Detectors

CORDELIA SCHMID, ROGER MOHR AND CHRISTIAN BAUCKHAGE

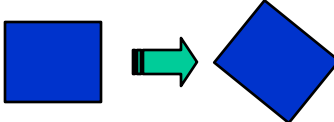
INRIA Rhône-Alpes, 655 av. de l'Europe, 38330 Montbonnot, France

Cordelia.Schmid@inrialpes.fr

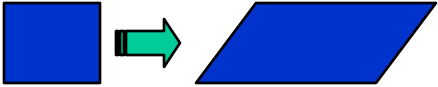
Abstract. Many different low-level feature detectors exist and it is widely agreed that the evaluation of detectors is important. In this paper we introduce two evaluation criteria for interest points: repeatability rate and information content. Repeatability rate evaluates the geometric stability under different transformations. Information content measures the distinctiveness of features. Different interest point detectors are compared using these two criteria. We determine which detector gives the best results and show that it satisfies the criteria well.

Models of Image Change

- Geometry

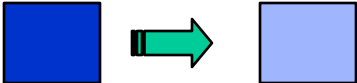
- Rotation 

- Similarity (rotation + uniform scale) 

- Affine (scale dependent on direction) 

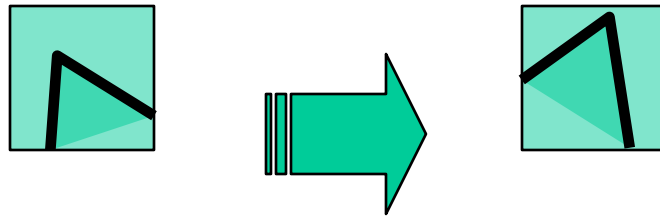
valid for: orthographic camera, locally planar object

- Photometry

- Affine intensity change ($I \mapsto aI + b$) 

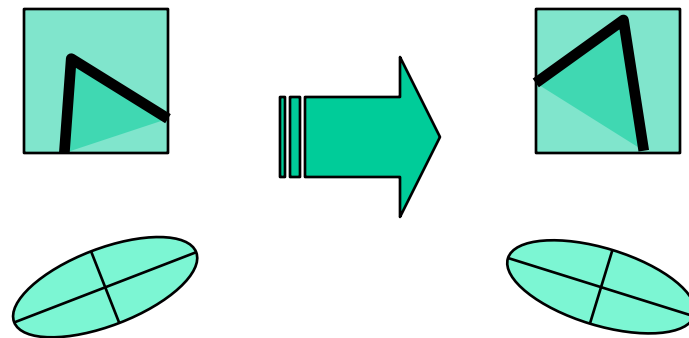
Harris Detector: Some Properties

- Rotation invariance?



Harris Detector: Some Properties

- Rotation invariance



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

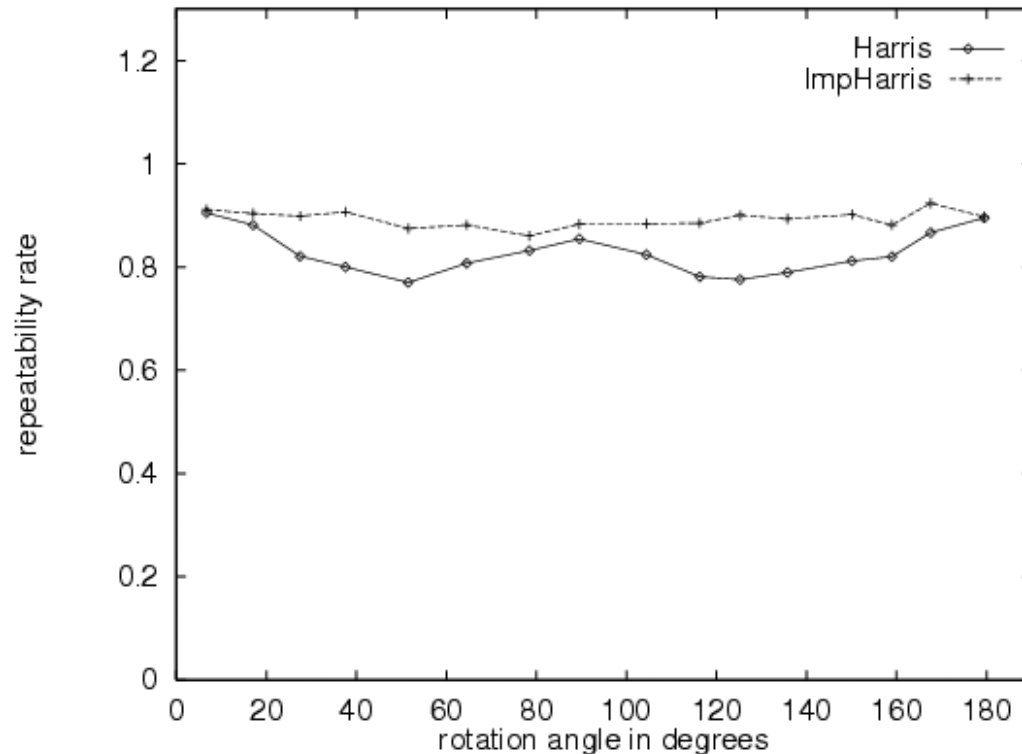
Corner response R is invariant to image rotation

Eigen analysis allows us to work in the canonical frame of the linear form.

Rotation Invariant Detection

ImpHarris: derivatives are computed more precisely by replacing the $[-2 -1 0 1 2]$ mask with derivatives of a Gaussian ($\sigma = 1$).

Harris Corner Detector



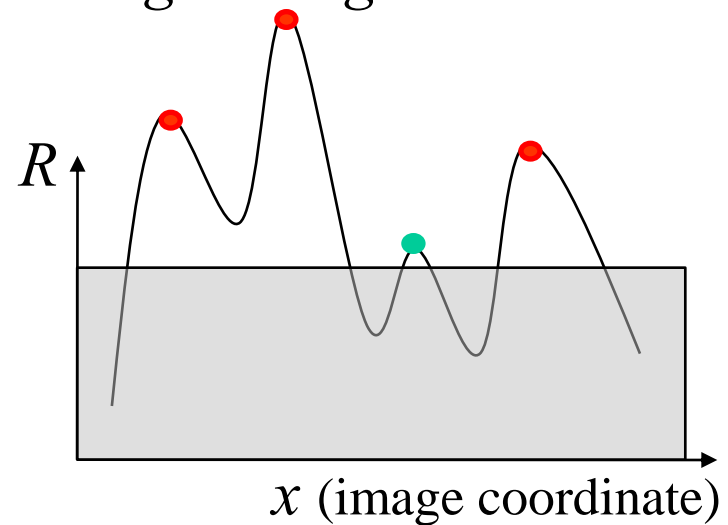
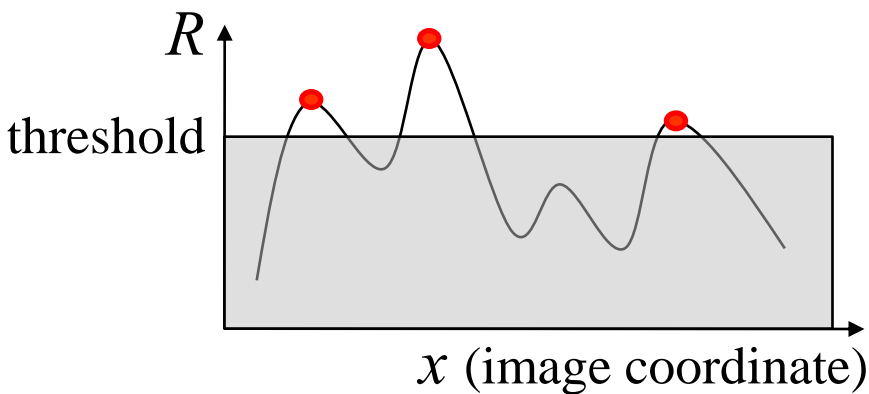
This gives us rotation invariant *detection*, but we'll need to do more to ensure a rotation invariant *descriptor*...

Harris Detector: Some Properties

- Invariance to image intensity change?

Harris Detector: Some Properties

- Partial invariance to additive and multiplicative intensity changes
 - ✓ Only derivatives are used \Rightarrow invariance to intensity shift $I \square I + b$
 - ✓ Intensity scaling: $I \square a I$ fine, *except for the threshold that's used to specify when R is large enough.*



Harris Detector: Some Properties

- Invariant to image scale?



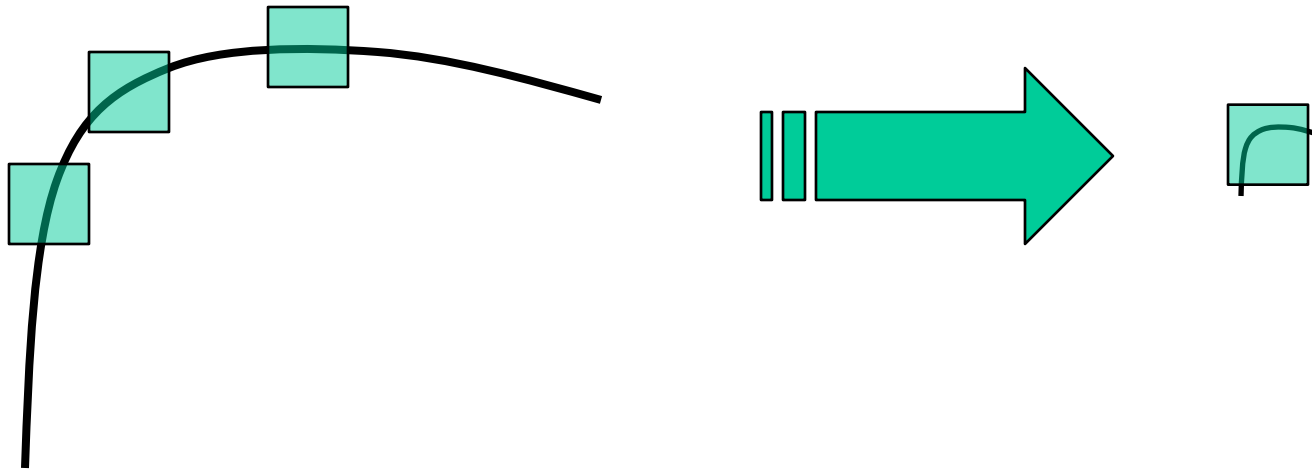
image



zoomed image

Harris Detector: Some Properties

- Not invariant to *image scale*!



All points will be classified as **edges**

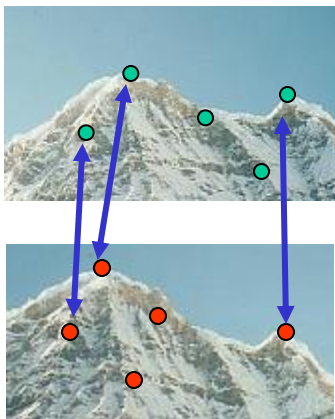
Corner !

Harris Detector: Some Properties

- Quality of Harris detector for different scale changes

Repeatability rate:

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

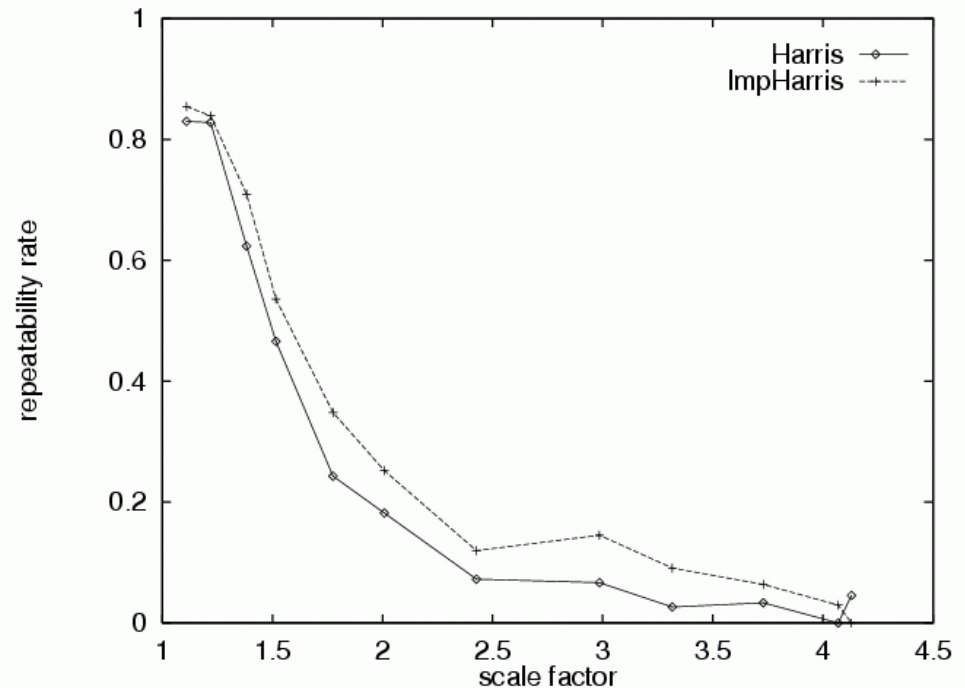
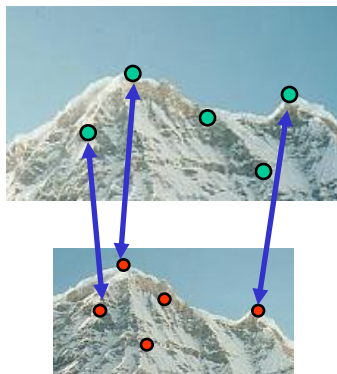


Harris Detector: Some Properties

- Quality of Harris detector for different scale changes

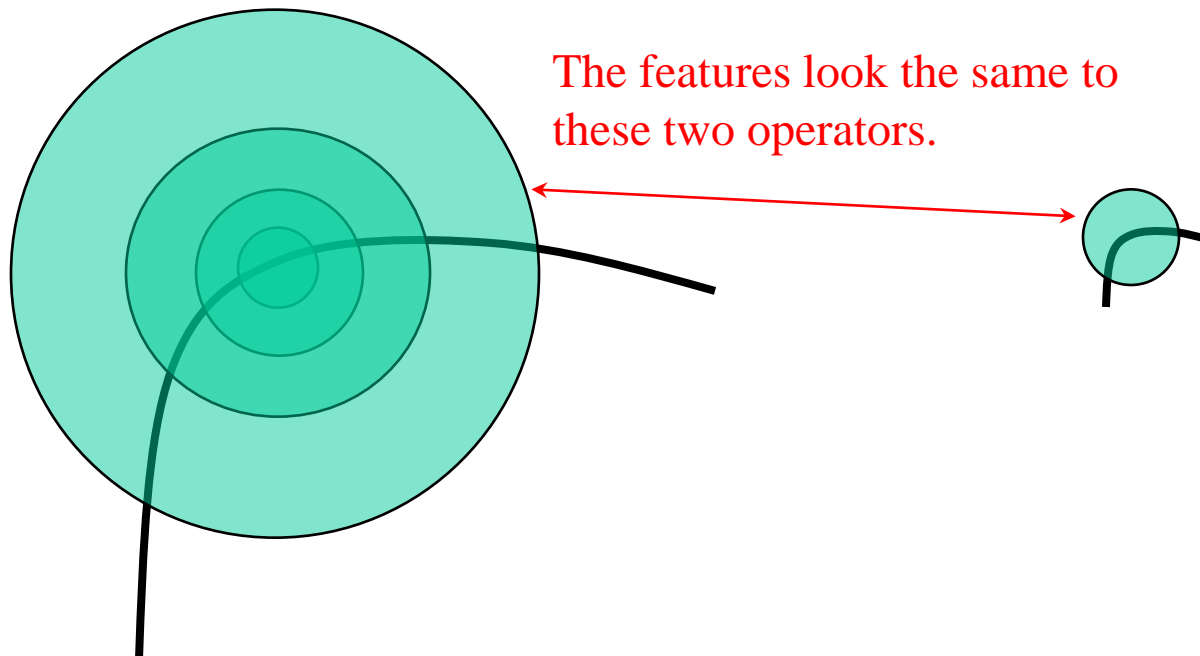
Repeatability rate:

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



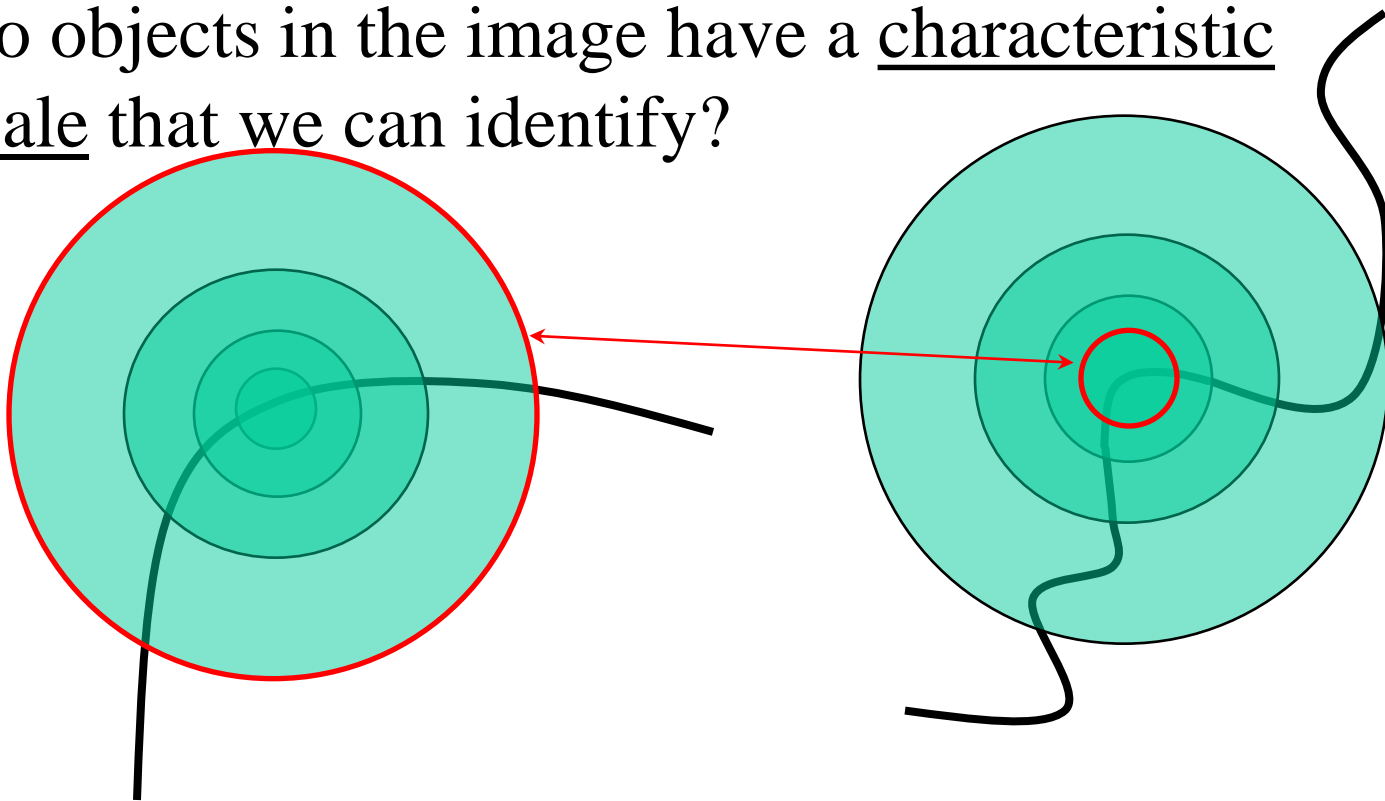
Scale Invariant Detection

- Consider regions (e.g. circles) of different sizes around a point
- Regions of corresponding sizes will look the same in both images



Scale Invariant Detection

- The problem: how do we choose corresponding circles *independently* in each image?
- Do objects in the image have a characteristic scale that we can identify?

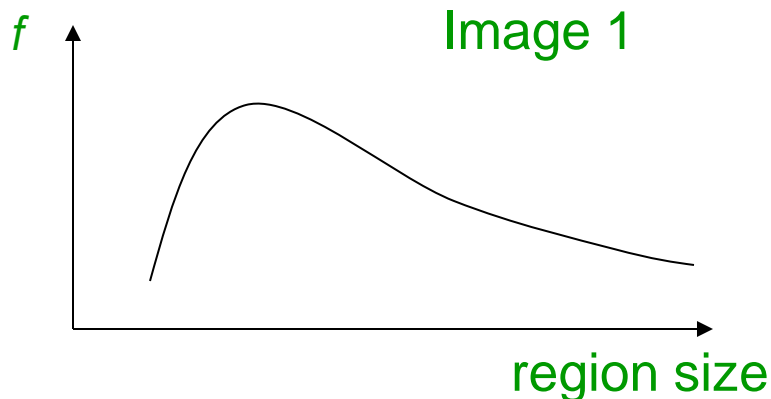


Scale Invariant Detection

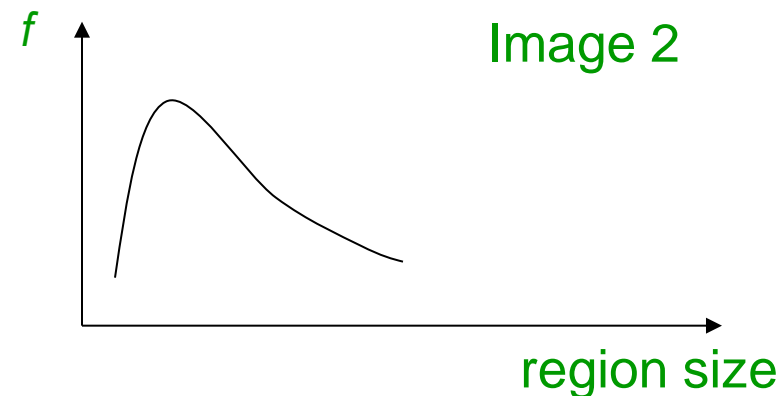
- Solution:
 - Design a function on the region (circle), which is “scale invariant” (the same for corresponding regions, even if they are at different scales)

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 = 1/2
→



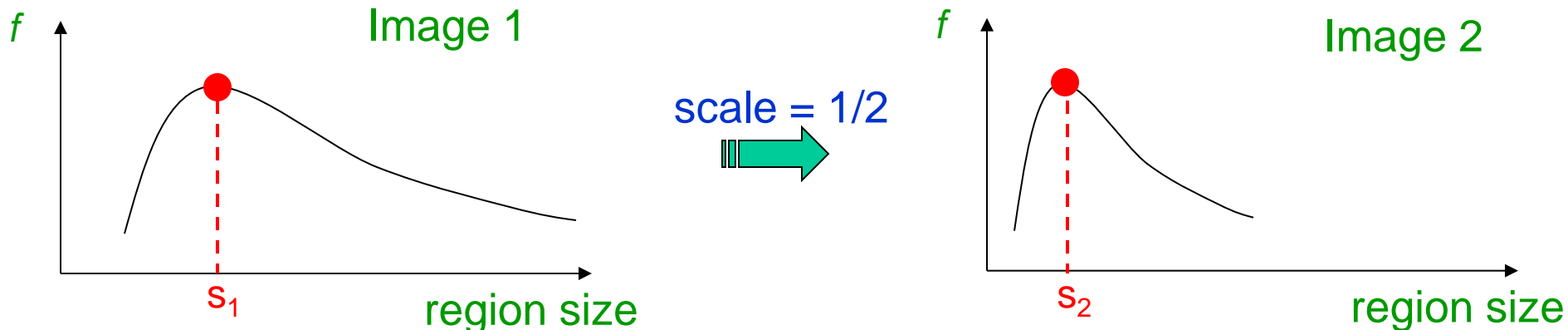
Scale Invariant Detection

- Common approach:

Take a local maximum of this function

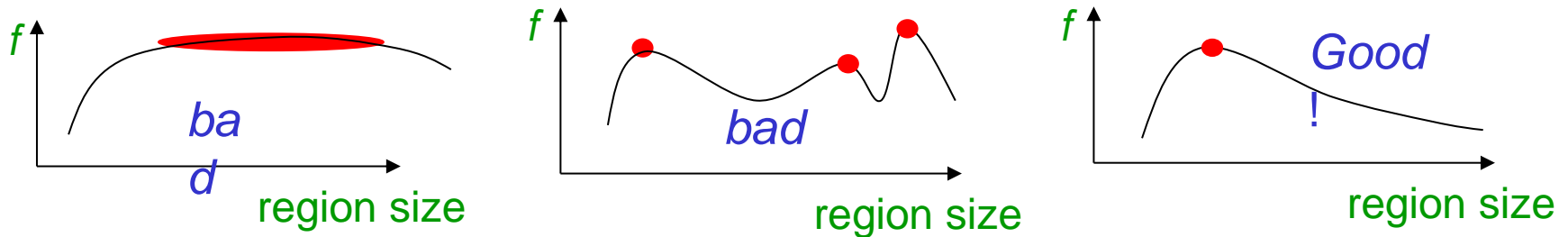
Observation: region size, for which the maximum is achieved, should be *invariant* to image scale.

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 a one which responds to contrast (sharp local intensity change)

Detection over scale

Requires a method to repeatably select points in location and scale:

- The only reasonable scale-space kernel is a Gaussian (Koenderink, 1984; Lindeberg, 1994)
- An efficient choice is to detect peaks in the difference of Gaussian pyramid (Burt & Adelson, 1983; Crowley & Parker, 1984 – but examining more scales)
- Difference-of-Gaussian with constant ratio of scales is a close approximation to Lindeberg's scale-normalized Laplacian (can be shown from the heat diffusion equation)

Scale Invariant Detection

- Functions for determining scale

$$f = \text{Kernel} * \text{Image}$$

Kernels:

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

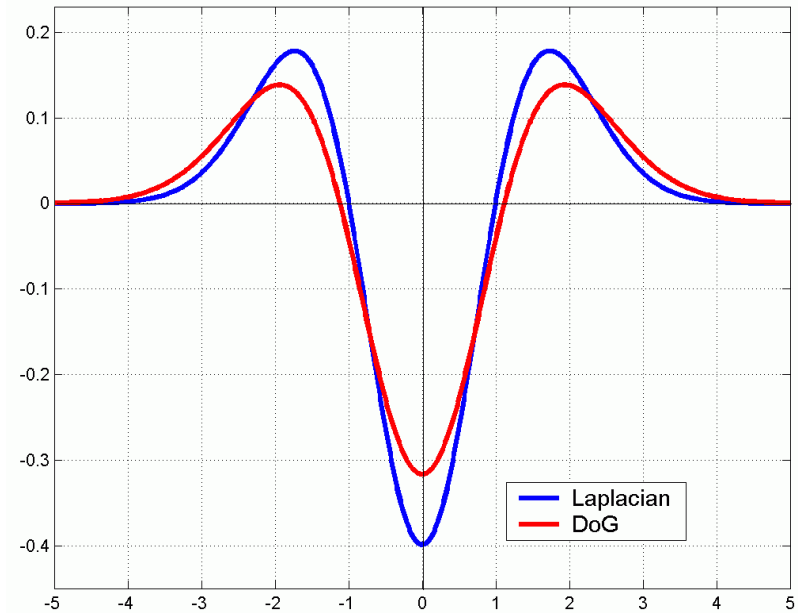
(Laplacian: 2nd derivative of Gaussian)

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

(Difference of Gaussians)

where Gaussian

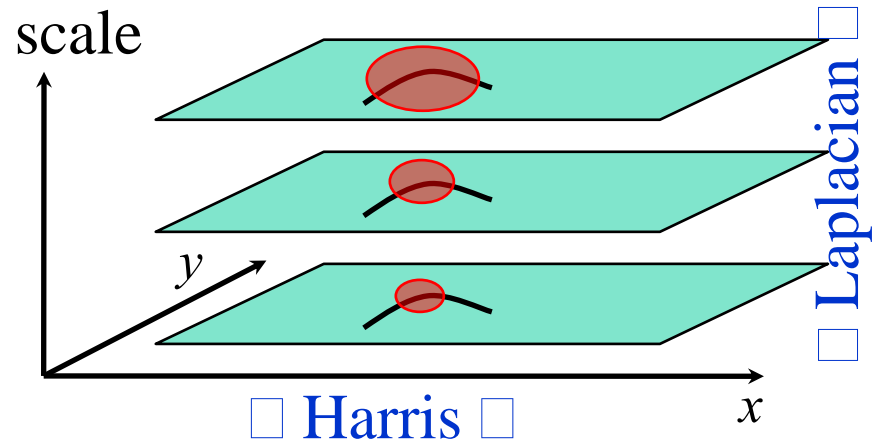
$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2 + y^2}{2\sigma^2}}$$



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

Scale Invariant Detectors

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



Laplacian of Gaussian for selection of characteristic scale

http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/mikolajczyk_ijcv2004.pdf

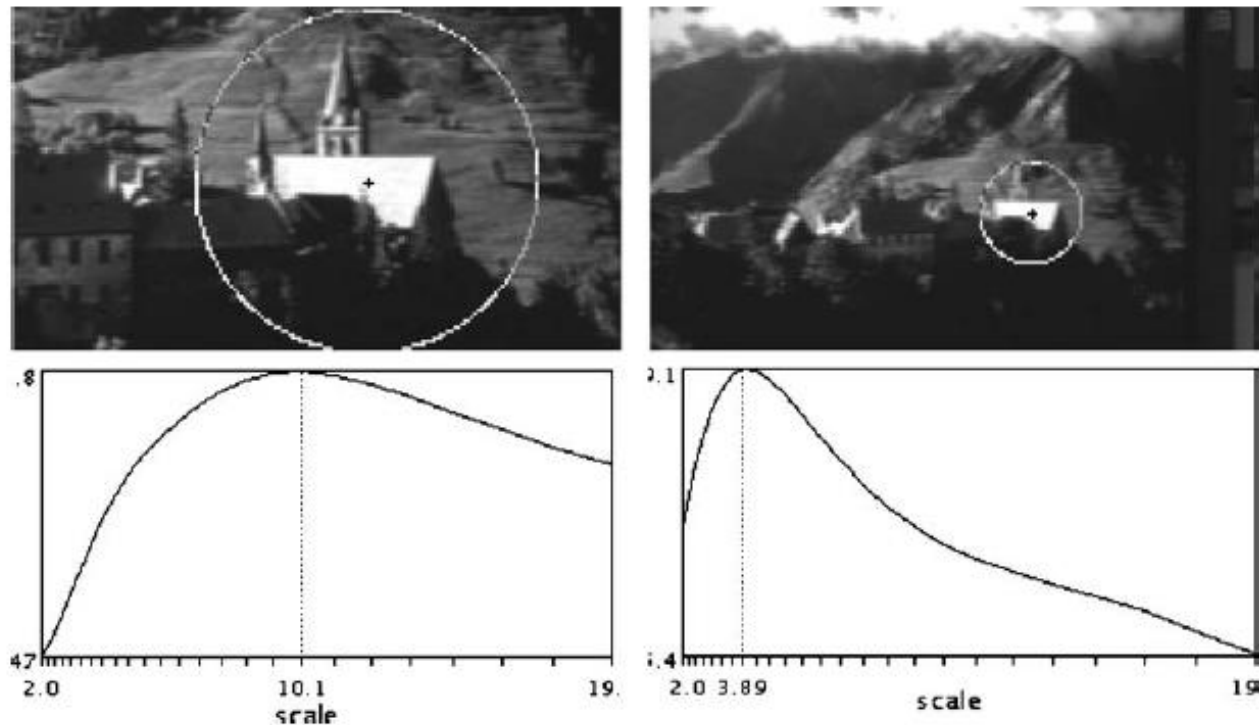


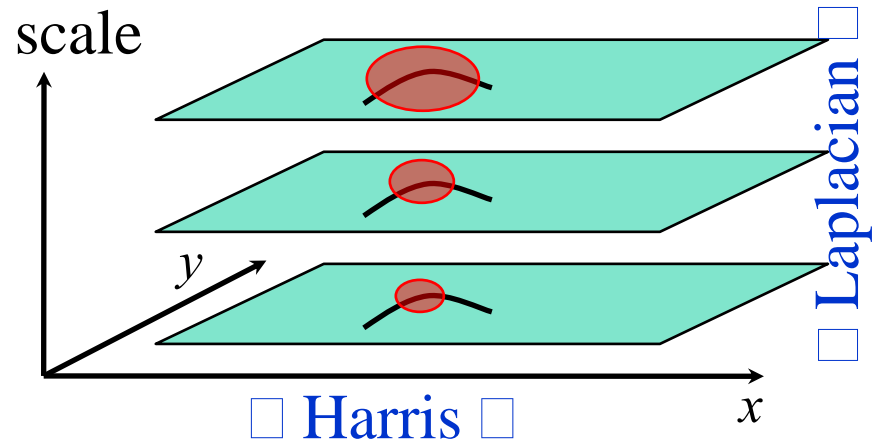
Figure 1. Example of characteristic scales. The top row shows two images taken with different focal lengths. The bottom row shows the response $F_{\text{norm}}(x, \sigma_n)$ over scales where F_{norm} is the normalized LoG (cf. Eq. (3)). The characteristic scales are 10.1 and 3.89 for the left and right image, respectively. The ratio of scales corresponds to the scale factor (2.5) between the two images. The radius of displayed regions in the top row is equal to 3 times the characteristic scale.

Scale Invariant Detectors

- **Harris-Laplacian**¹

Find local maximum of:

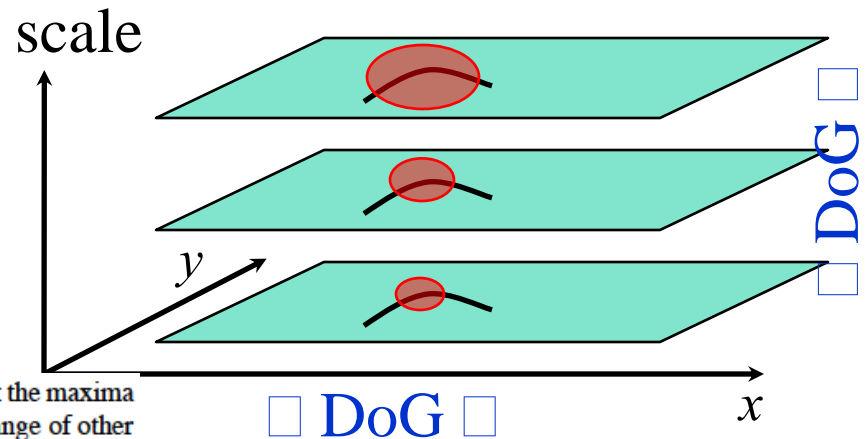
- Harris corner detector in space (image coordinates)
- Laplacian in scale



- **SIFT (Lowe)**²

Find local maximum (minimum) of:

- Difference of Gaussians in space and scale



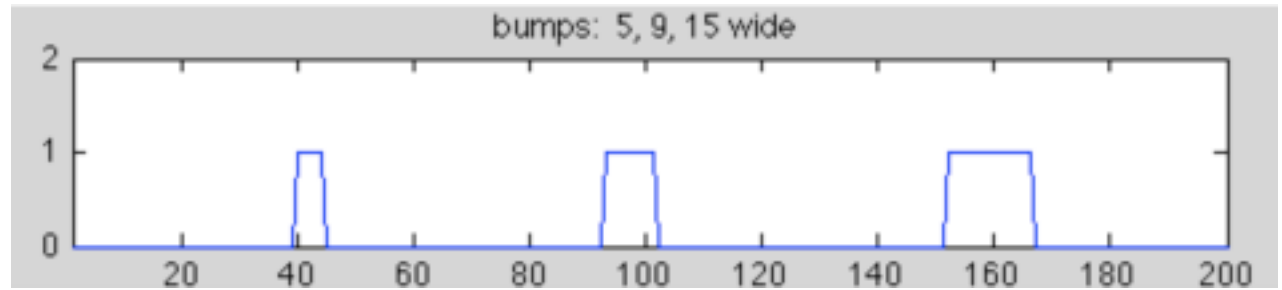
In detailed experimental comparisons, Mikolajczyk (2002) found that the maxima and minima of $\sigma^2 \nabla^2 G$ produce the most stable image features compared to a range of other possible image functions, such as the gradient, Hessian, or Harris corner function.

¹ K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001

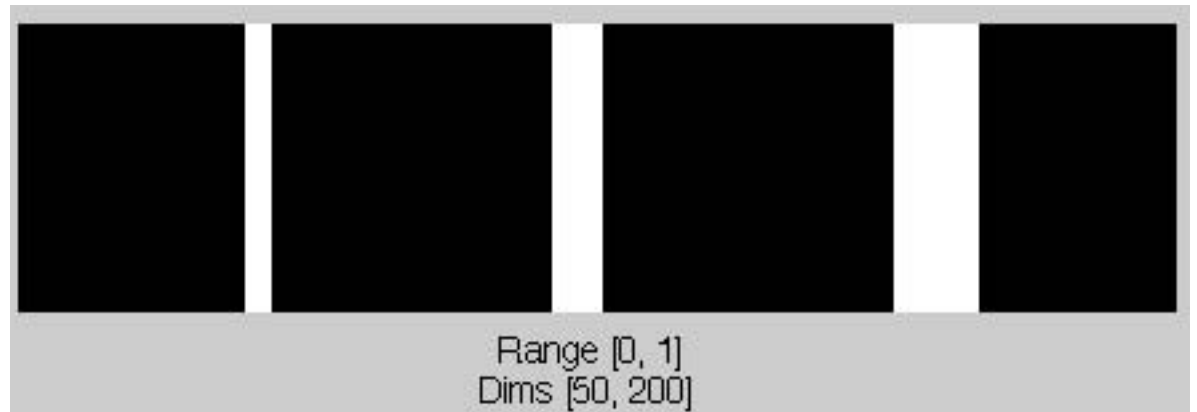
² D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". Accepted to IJCV 2004

Scale-space example: 3 bumps of different widths.

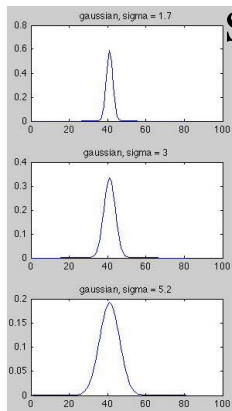
1-d bumps



display as an image



blur with
Gaussians of
increasing
width



scale

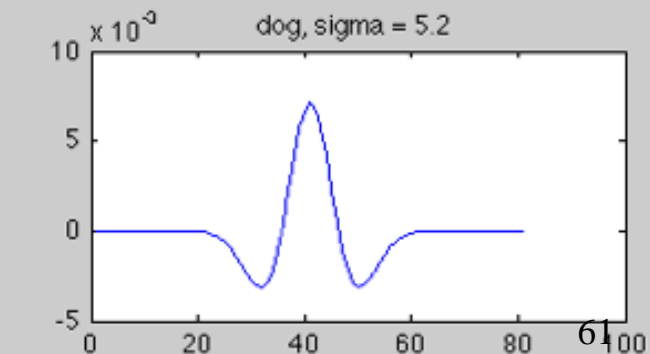
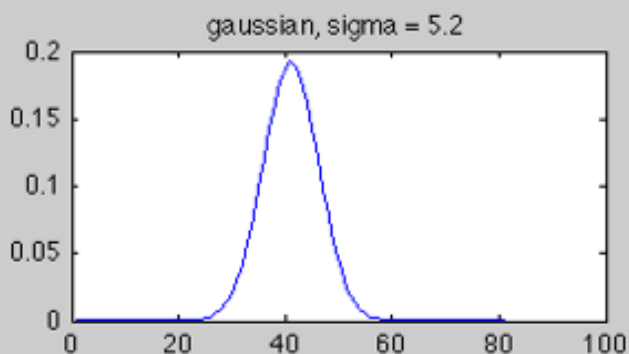
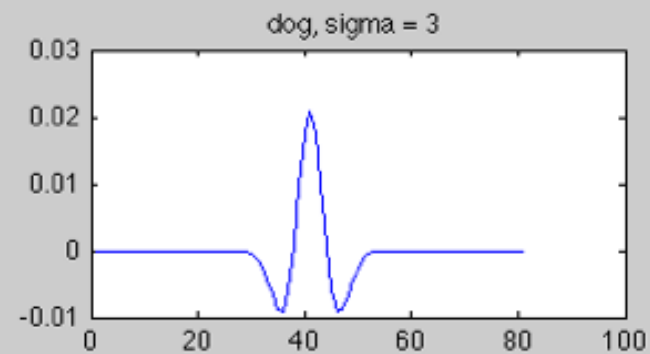
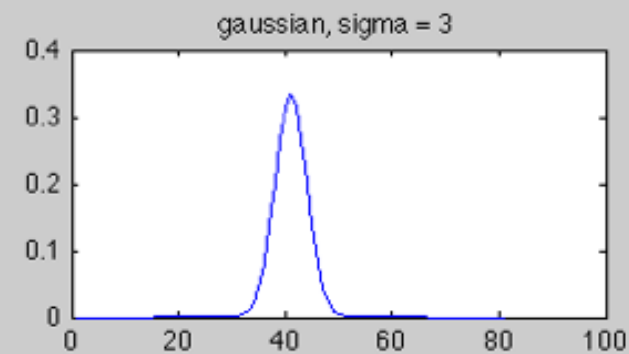
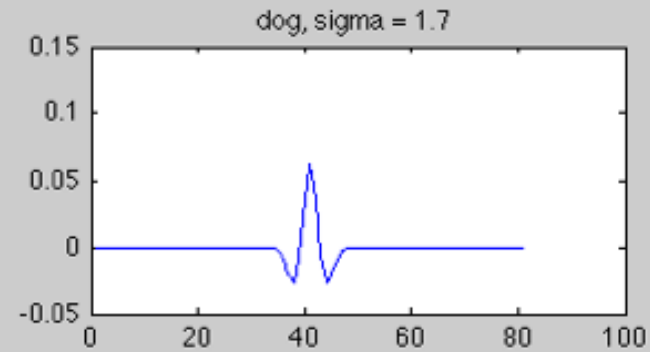
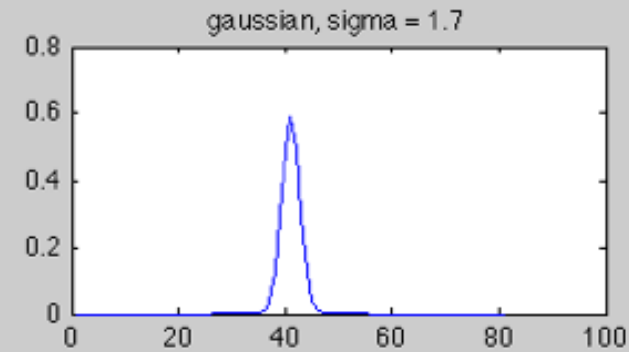


space

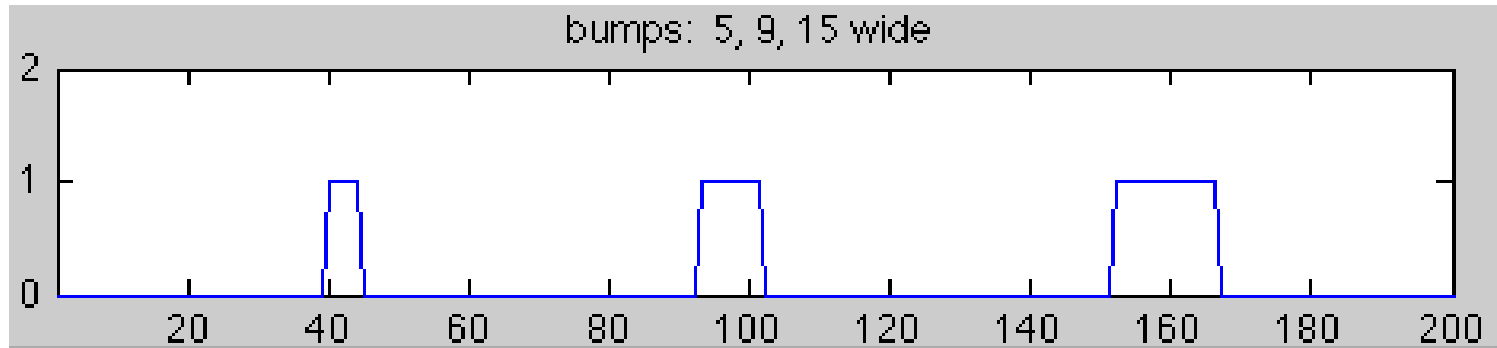


Range [0, 2.51]
Dims [50, 200]

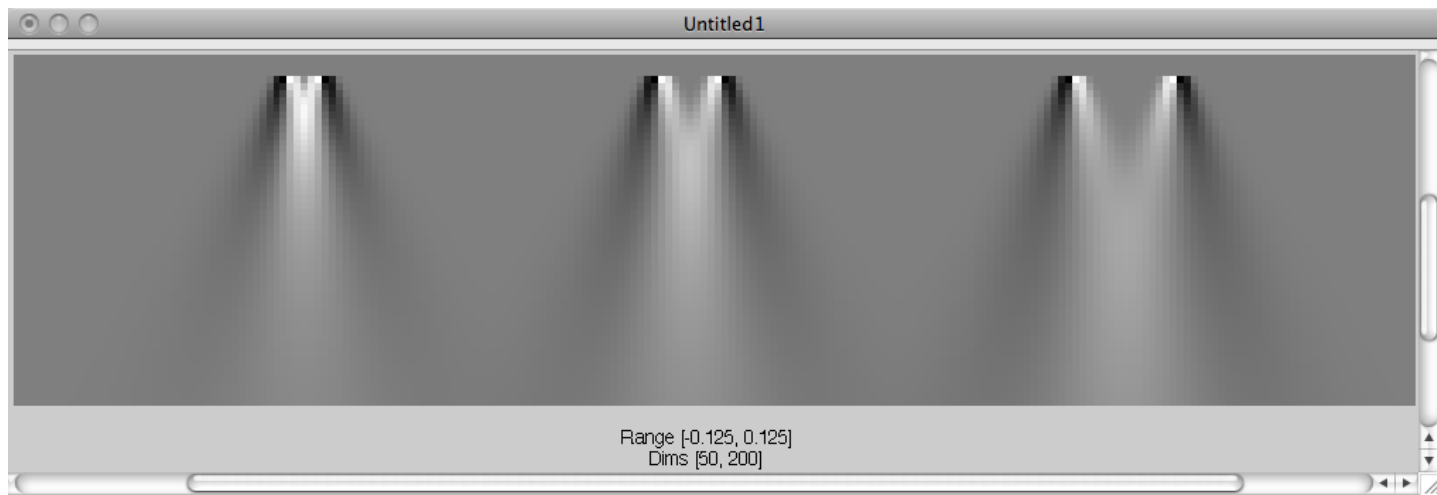
Gaussian and difference-of-Gaussian filters



The bumps, filtered by difference-of-Gaussian filters

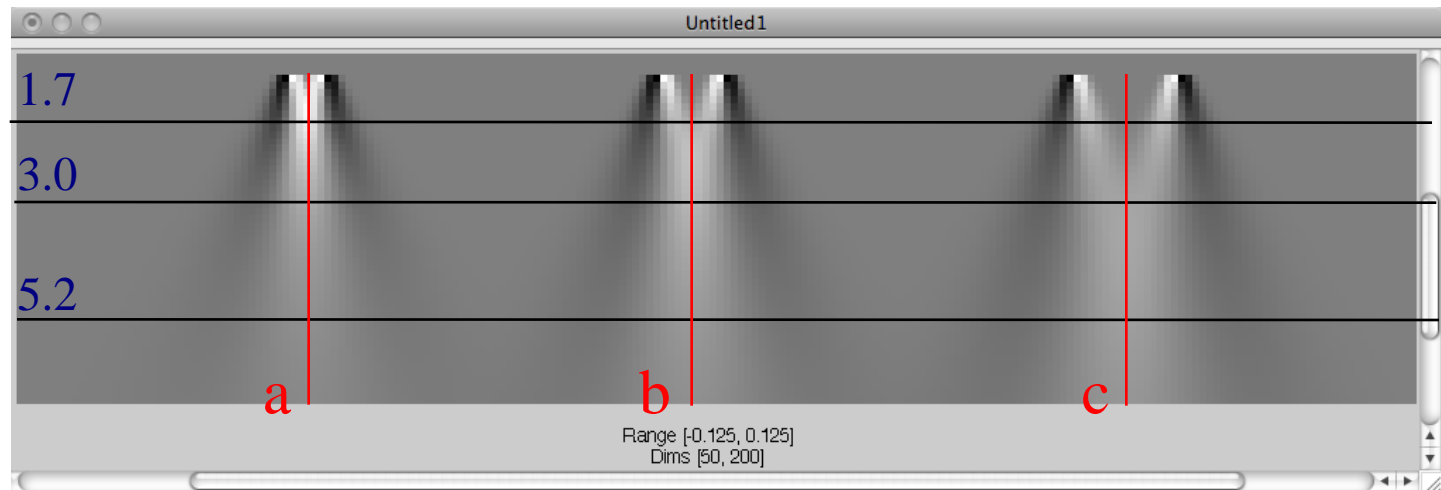
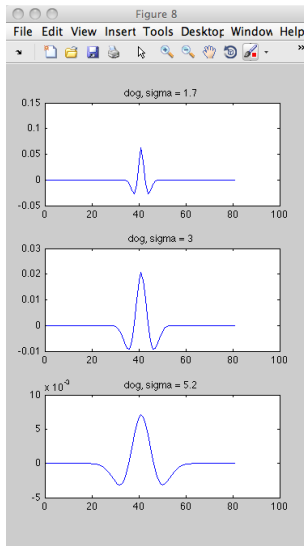
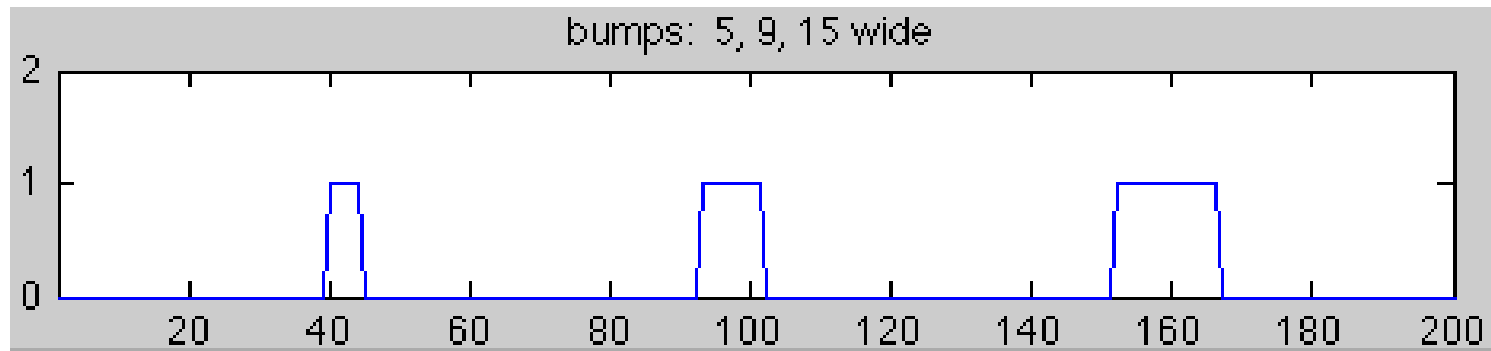


scale



space →

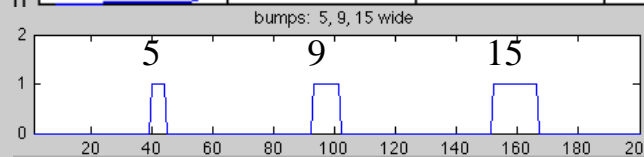
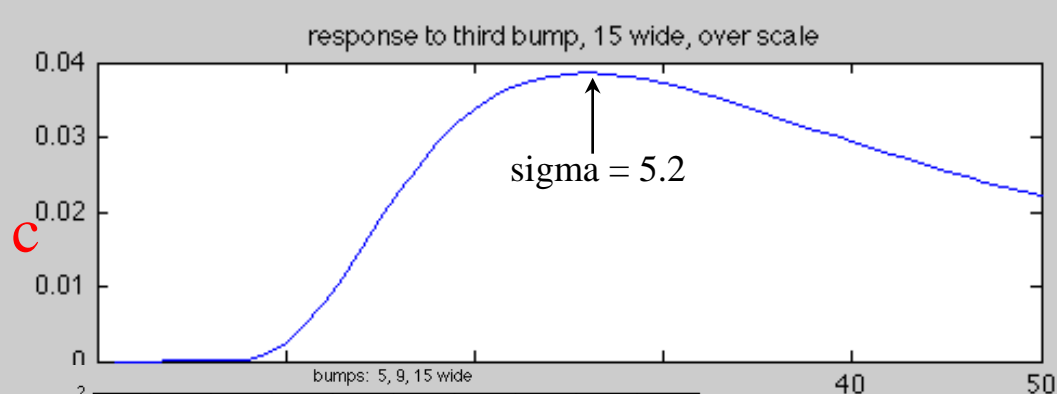
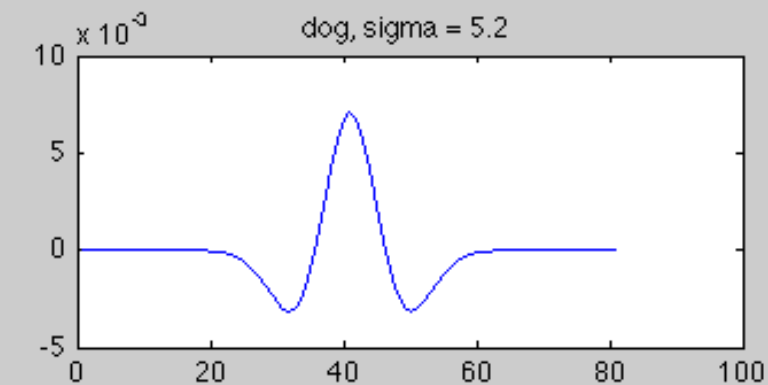
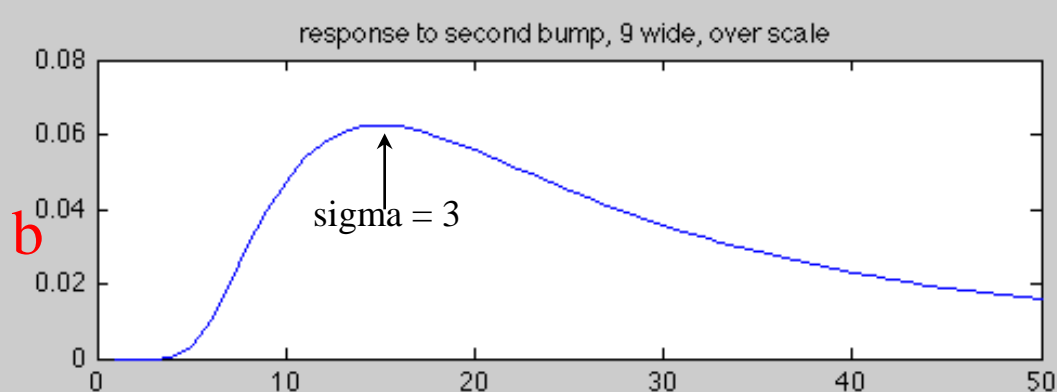
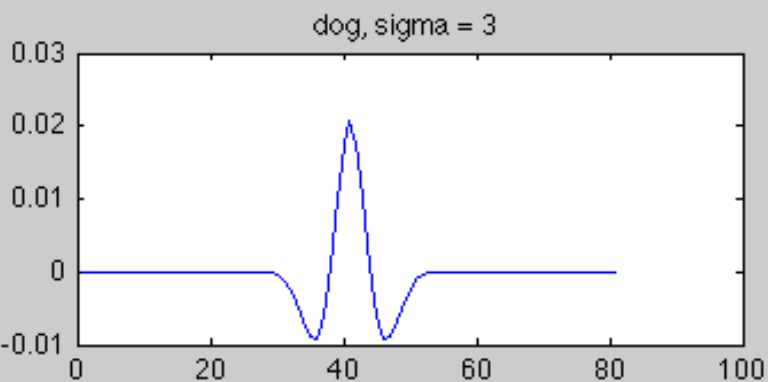
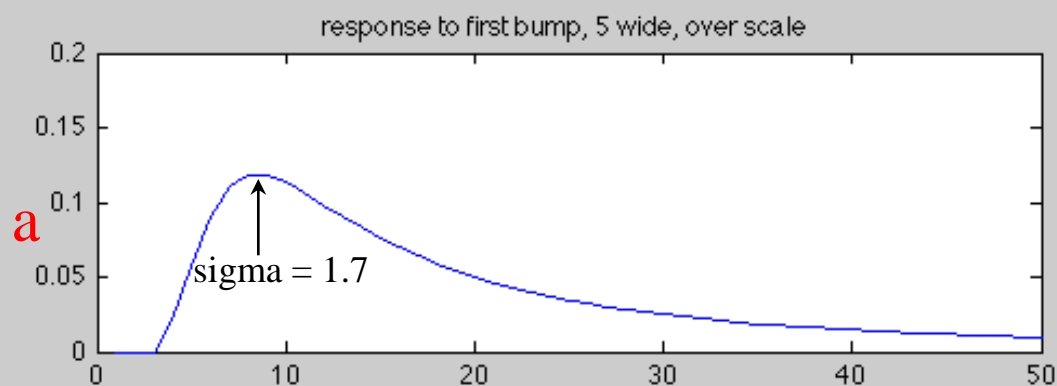
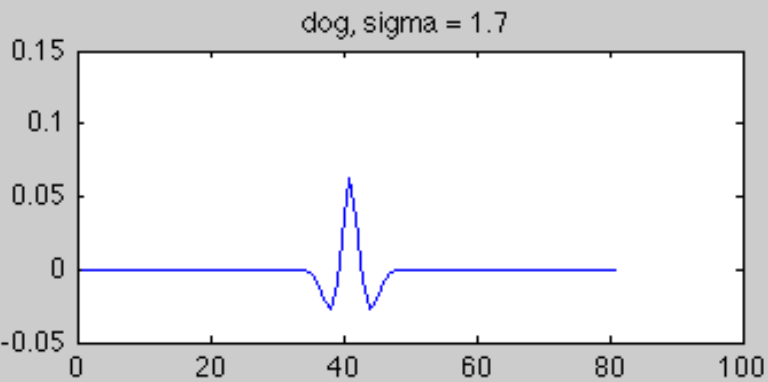
The bumps, filtered by difference-of-Gaussian filters



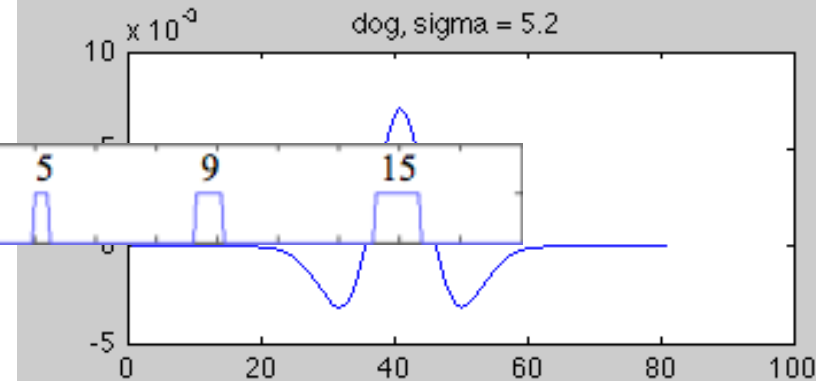
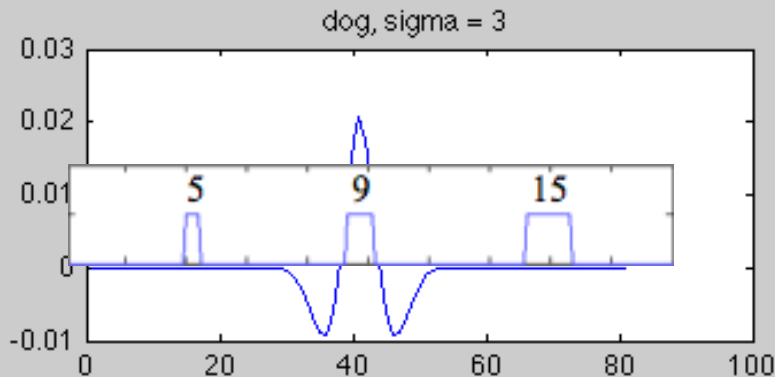
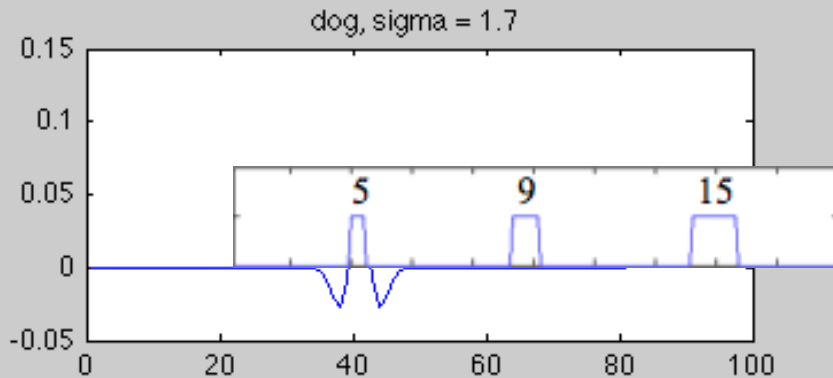
cross-sections along red lines plotted next slide

Scales of peak responses are proportional to bump width (the characteristic scale of each bump):

$$[1.7, 3, 5.2] ./ [5, 9, 15] = 0.3400 \quad 0.3333 \quad 0.3467$$



Diff of Gauss filter giving peak response



Scales of peak responses are proportional to bump width (the characteristic scale of each bump):

$$[1.7, 3, 5.2] ./ [5, 9, 15] = 0.3400$$

$$0.3333 \quad 0.3467$$

Note that the max response filters each has the same relationship to the bump that it favors (the zero crossings of the filter are about at the bump edges). So the scale space analysis correctly picks out the “characteristic scale” for each of the bumps.

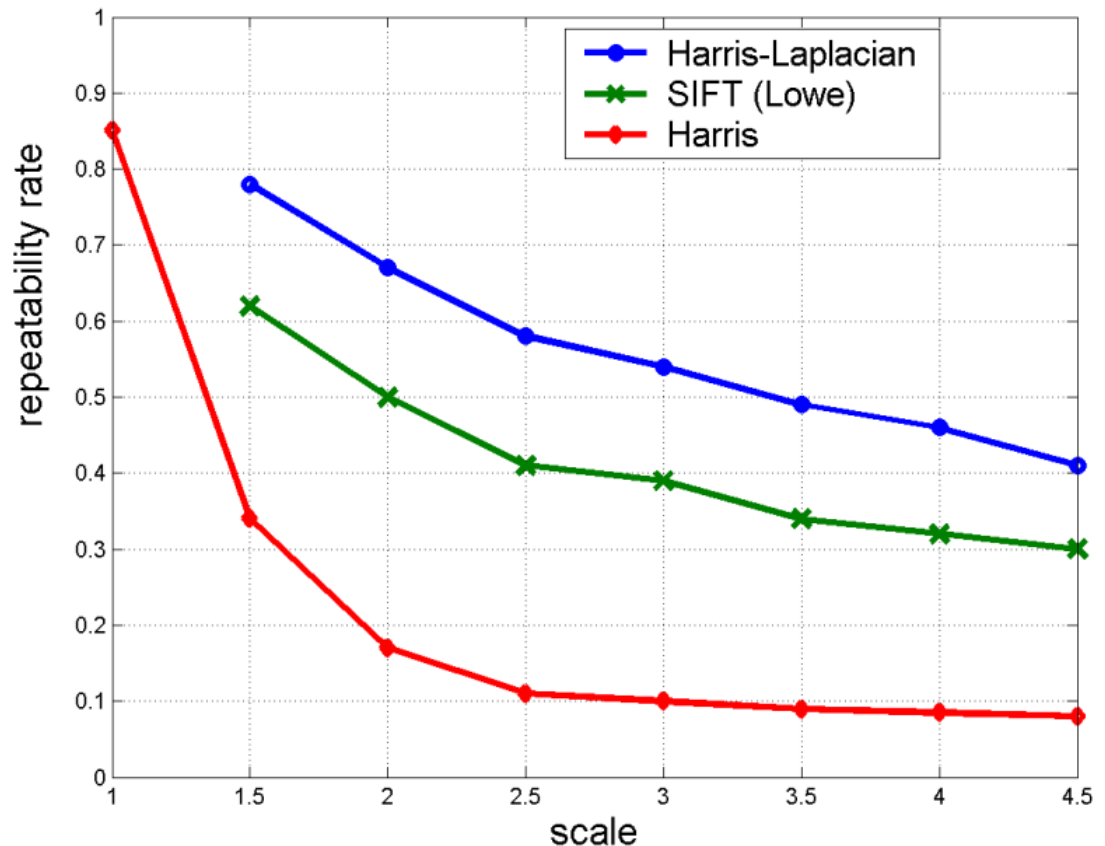
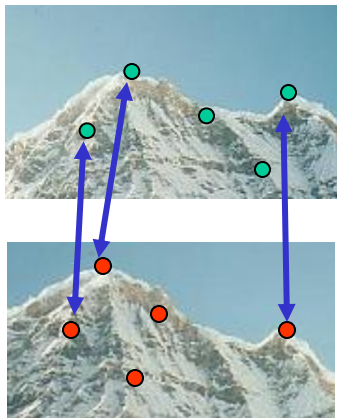
More generally, this happens for the features of the images we analyze.

Scale Invariant Detectors

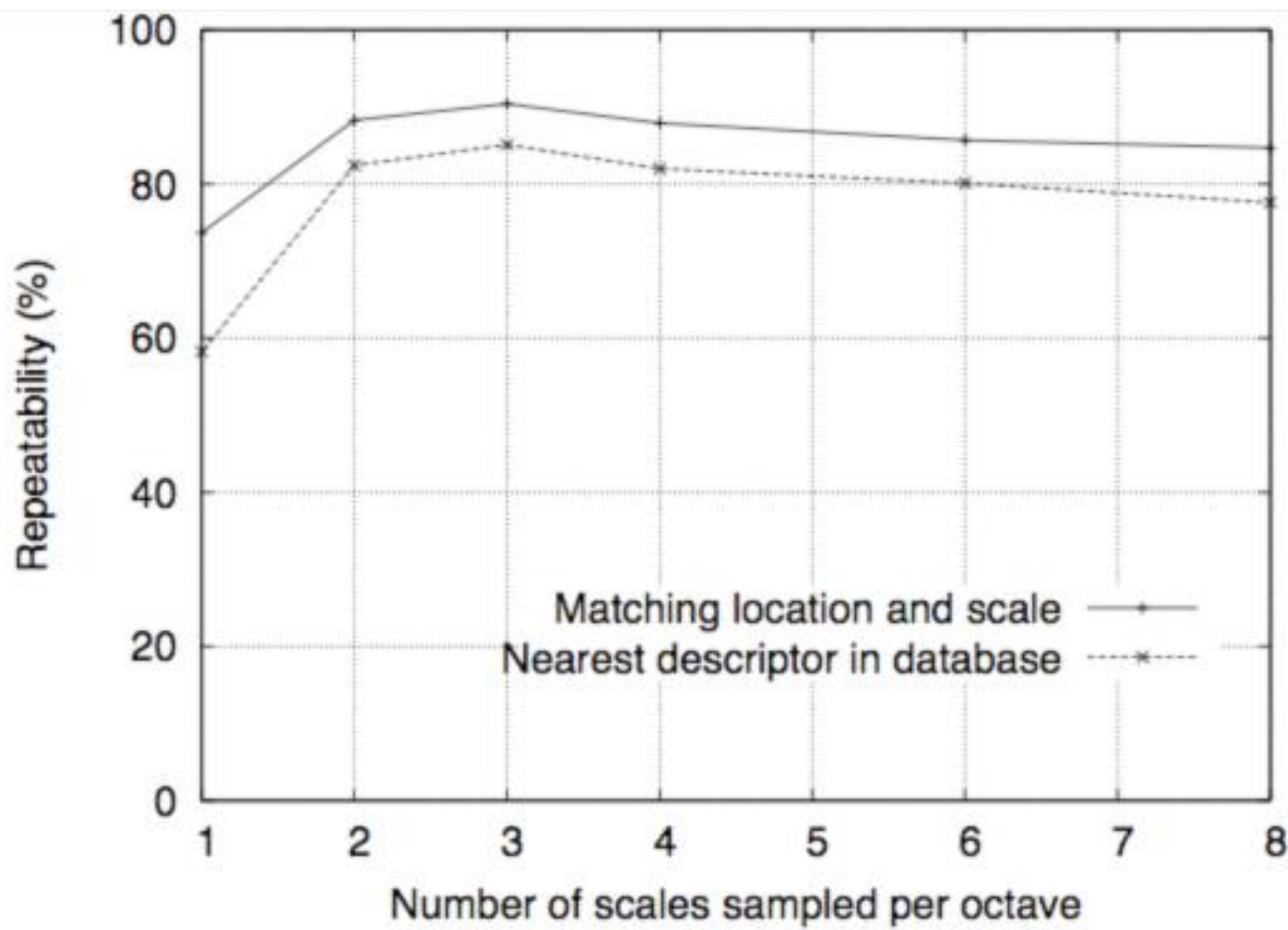
- Experimental evaluation of detectors w.r.t. scale change

Repeatability rate:

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



Repeatability vs number of scales sampled per octave



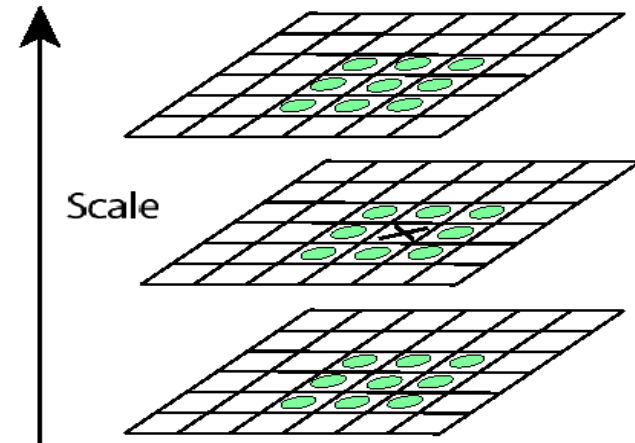
Some details of key point localization over scale and space

- Detect maxima and minima of difference-of-Gaussian in scale space
- Fit a quadratic to surrounding values for sub-pixel and sub-scale interpolation (Brown & Lowe, 2002)
- Taylor expansion around point:

$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}$$

- Offset of extremum (use finite differences for derivatives):

$$\hat{\mathbf{x}} = - \frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}$$



Scale and Rotation Invariant Detection: Summary

- **Given:** two images of the same scene with a large *scale difference and/or rotation* 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). Also, find characteristic *orientation*.

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

Example of keypoint detection



(c)

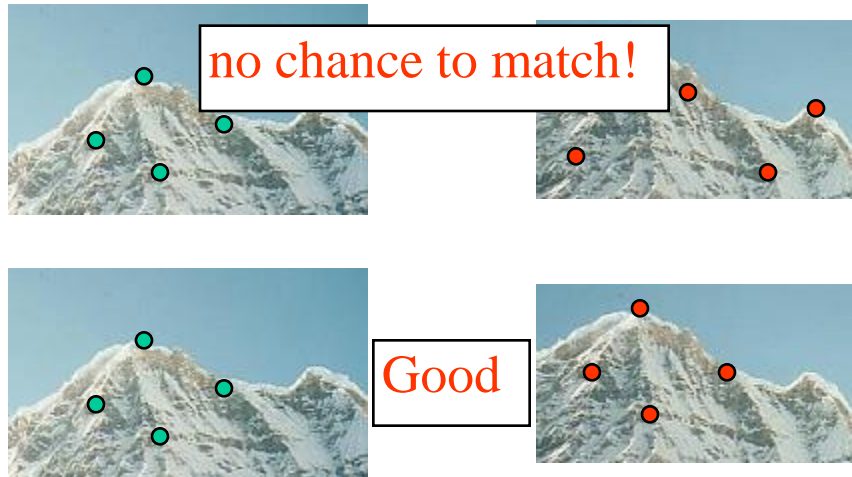
Figure 12. Robust matching: Harris-Laplace detects 190 and 213 points in the left and right images, respectively (a). 58 points are initially matched (b). There are 32 inliers to the estimated homography (c), all of which are correct. The estimated scale factor is 4.9 and the estimated rotation angle is 19 degrees.

Outline

- Feature point detection
 - Harris corner detector
 - finding a characteristic scale
- Local image description
 - SIFT features

Recall: Matching with Features

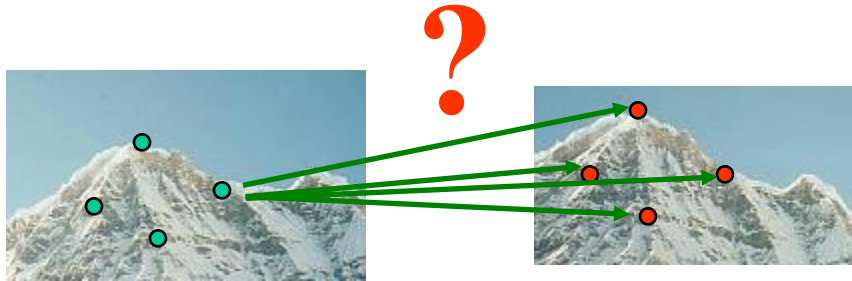
- Problem 1:
 - Detect the *same point independently* in both images



We need a repeatable detector

Recall: Matching with Features

- Problem 2:
 - For each point correctly recognize the corresponding one



We need a reliable and distinctive descriptor

CVPR 2003 Tutorial

Recognition and Matching Based on Local Invariant Features

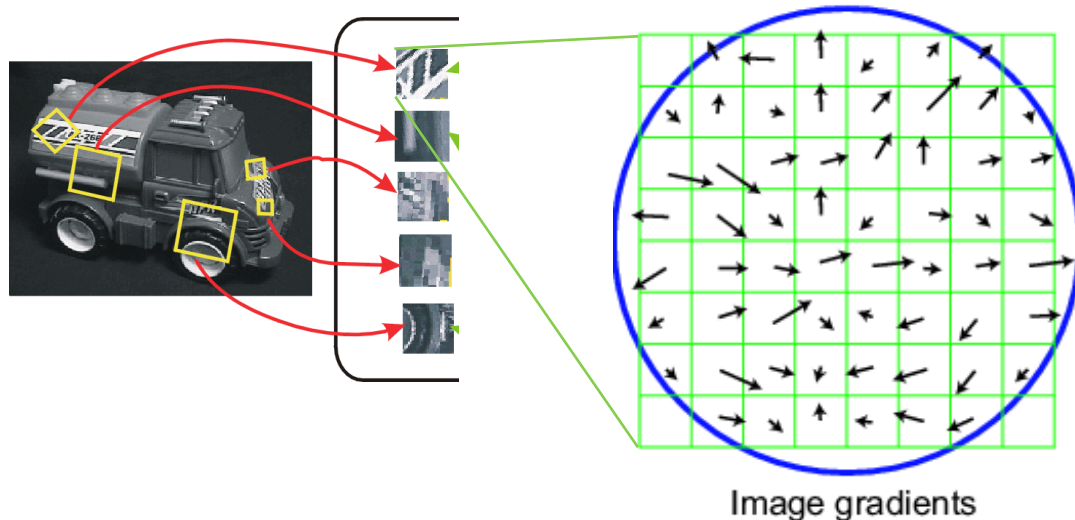
David Lowe

Computer Science Department
University of British Columbia

<http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>

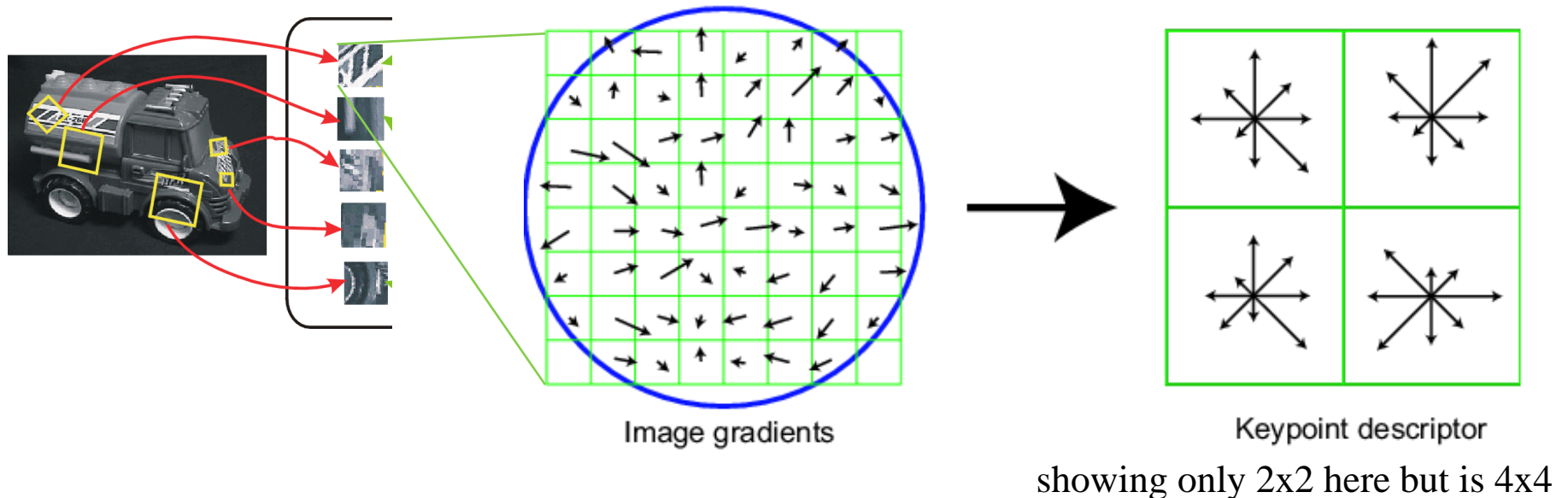
SIFT vector formation

- Computed on rotated and scaled version of window according to computed orientation & scale
 - resample the window
- Based on gradients weighted by a Gaussian of variance half the window (for smooth falloff)



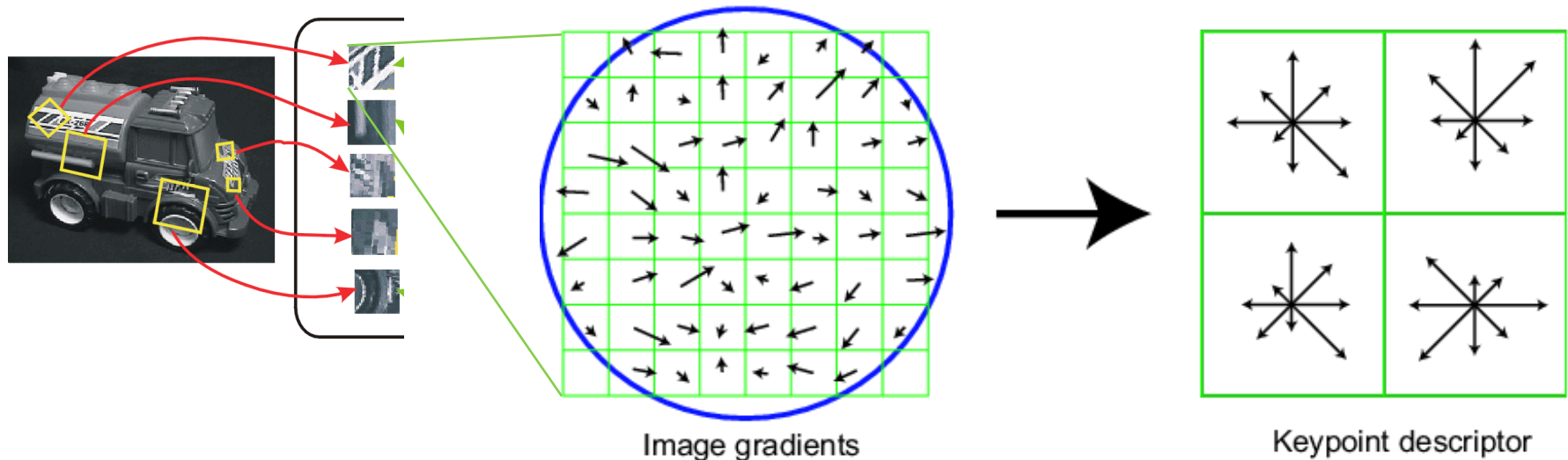
SIFT vector formation

- 4x4 array of gradient orientation histograms
 - not really histogram, weighted by magnitude
- 8 orientations x 4x4 array = 128 dimensions
- Motivation: some sensitivity to spatial layout, but not too much.



Reduce effect of illumination

- 128-dim vector normalized to 1
- Threshold gradient magnitudes to avoid excessive influence of high gradients
 - after normalization, clamp gradients >0.2
 - renormalize



Tuning and evaluating the SIFT descriptors

Database images were subjected to rotation, scaling, affine stretch, brightness and contrast changes, and added noise. Feature point detectors and descriptors were compared before and after the distortions, and evaluated for:

- Sensitivity to number of histogram orientations and subregions.
- Stability to noise.
- Stability to affine change.
- Feature distinctiveness

Sensitivity to number of histogram orientations and subregions, n .

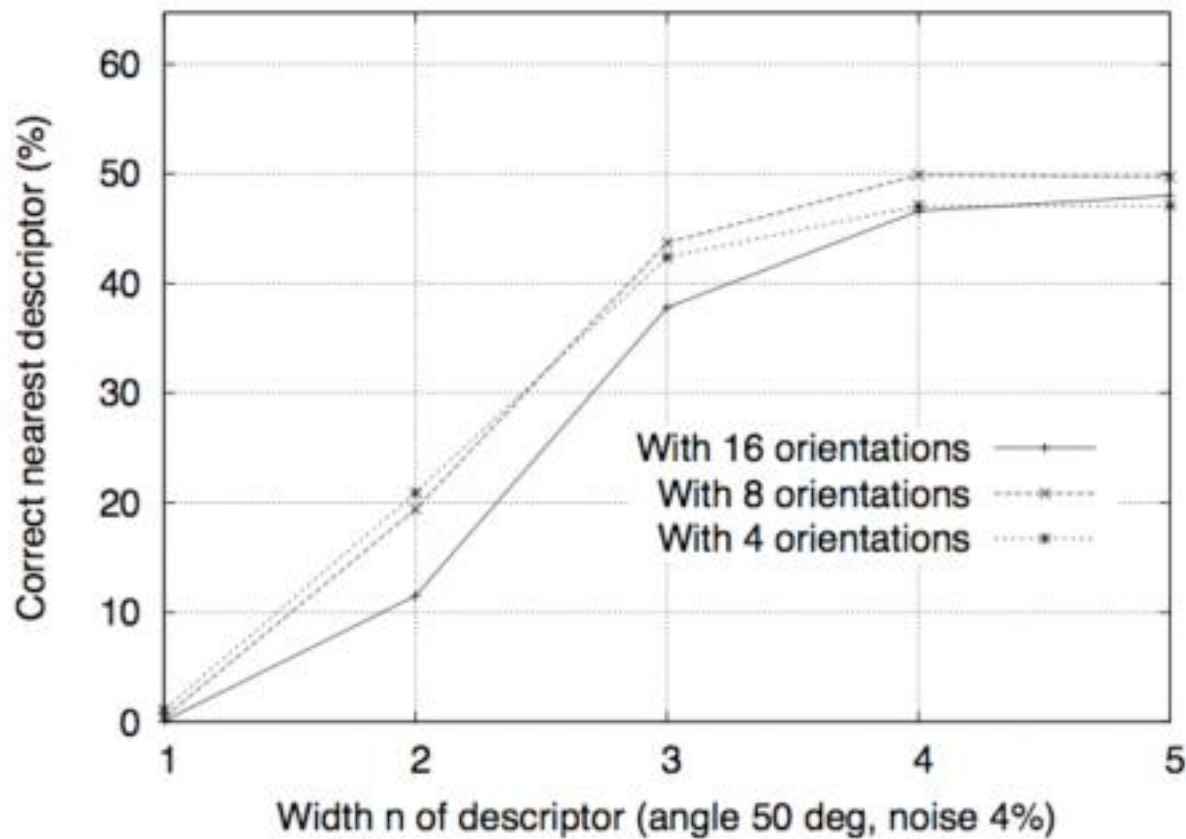
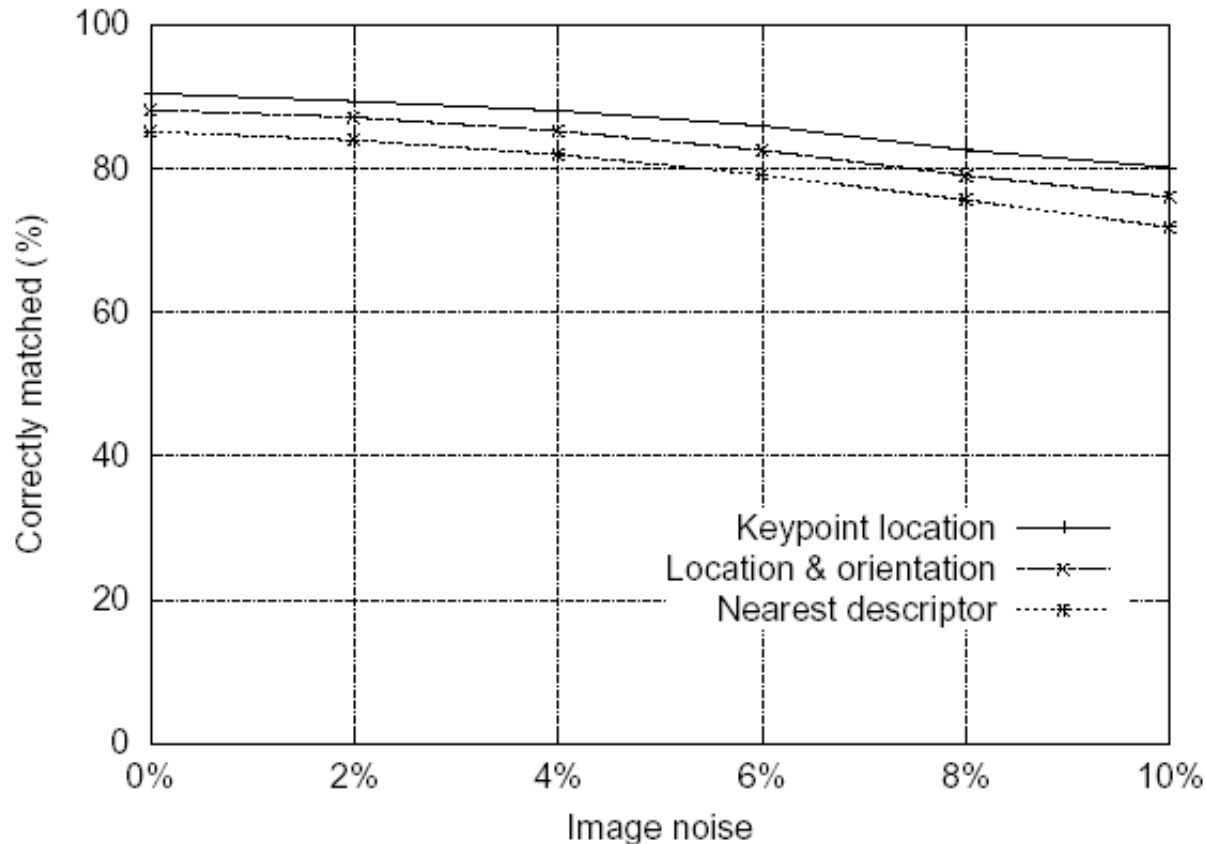


Figure 8: This graph shows the percent of keypoints giving the correct match to a database of 40,000 keypoints as a function of width of the $n \times n$ keypoint descriptor and the number of orientations in each histogram. The graph is computed for images with affine viewpoint change of 50 degrees and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

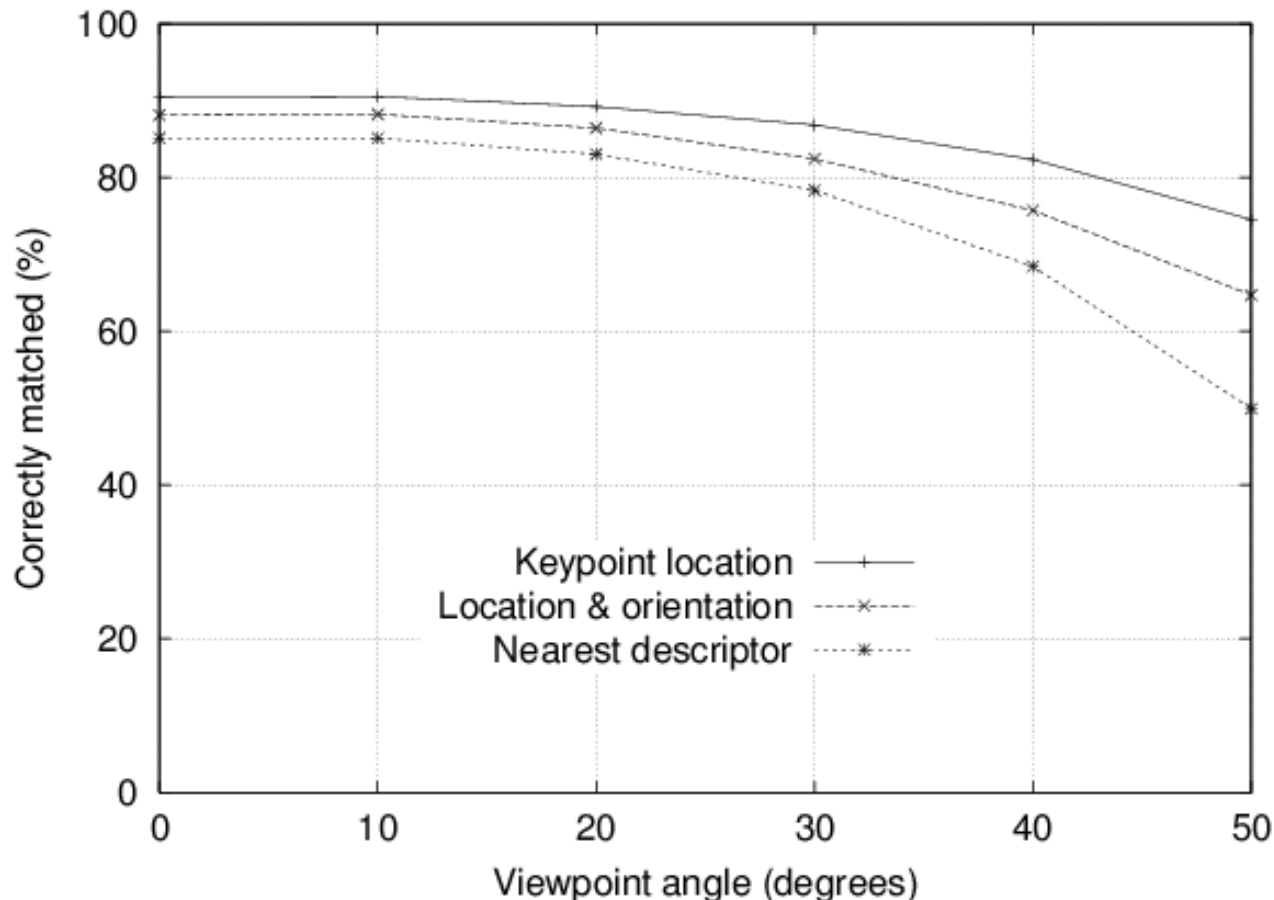
Feature stability to noise

- 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

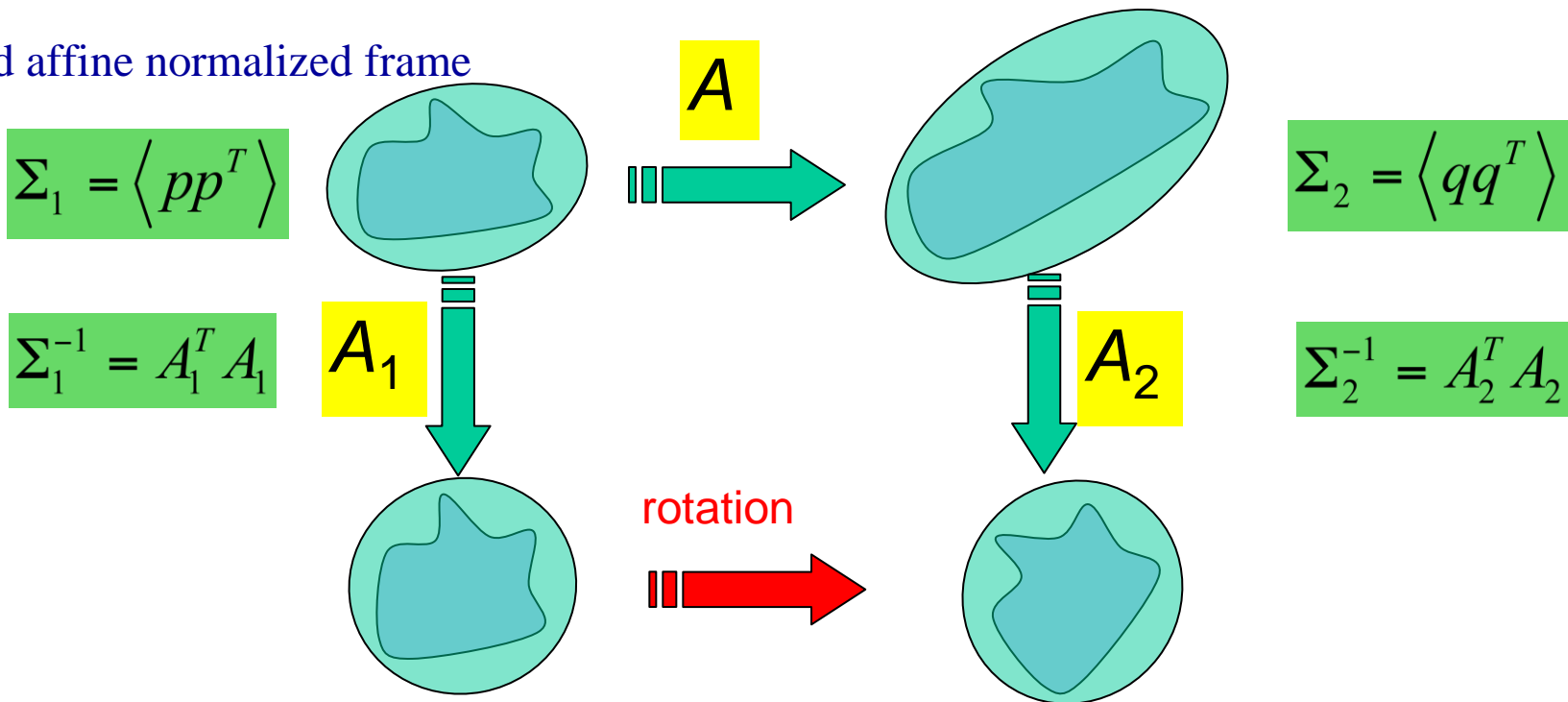
- Match features after random change in image scale & orientation, with 2% image noise, and affine distortion
- Find nearest neighbor in database of 30,000 features



Affine Invariant Descriptors

If a wide range of affine invariance is desired, such as for a surface that is known to be planar, then a simple solution is to adopt the approach of Pritchard and Heidrich (2003) in which additional SIFT features are generated from 4 affine-transformed versions of the training image corresponding to 60 degree viewpoint changes. This allows for the use of standard SIFT features with no additional cost when processing the image to be recognized, but results in an increase in the size of the feature database by a factor of 3.

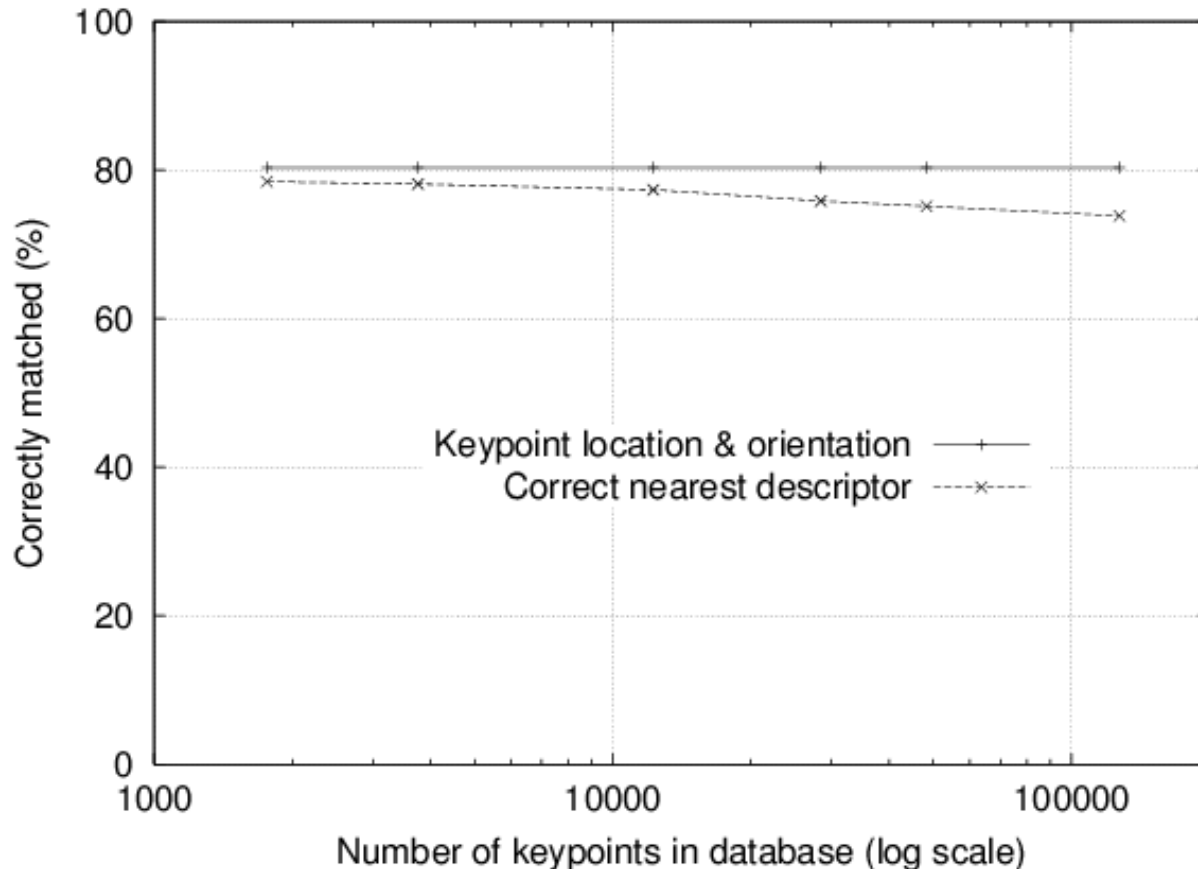
Find affine normalized frame



Compute rotational invariant descriptor in this normalized frame

Distinctiveness of features

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



Application of invariant local features to object (instance) recognition.

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

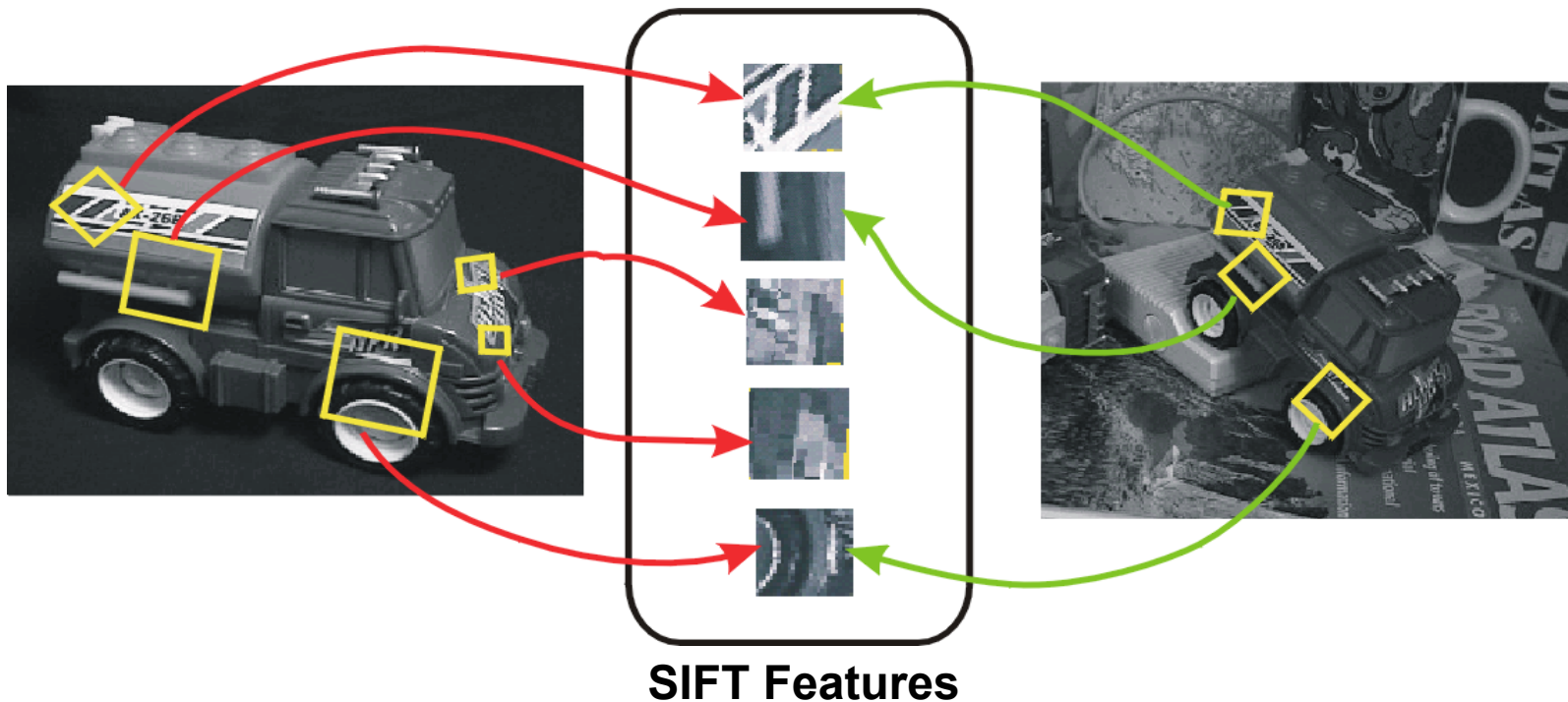




Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original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

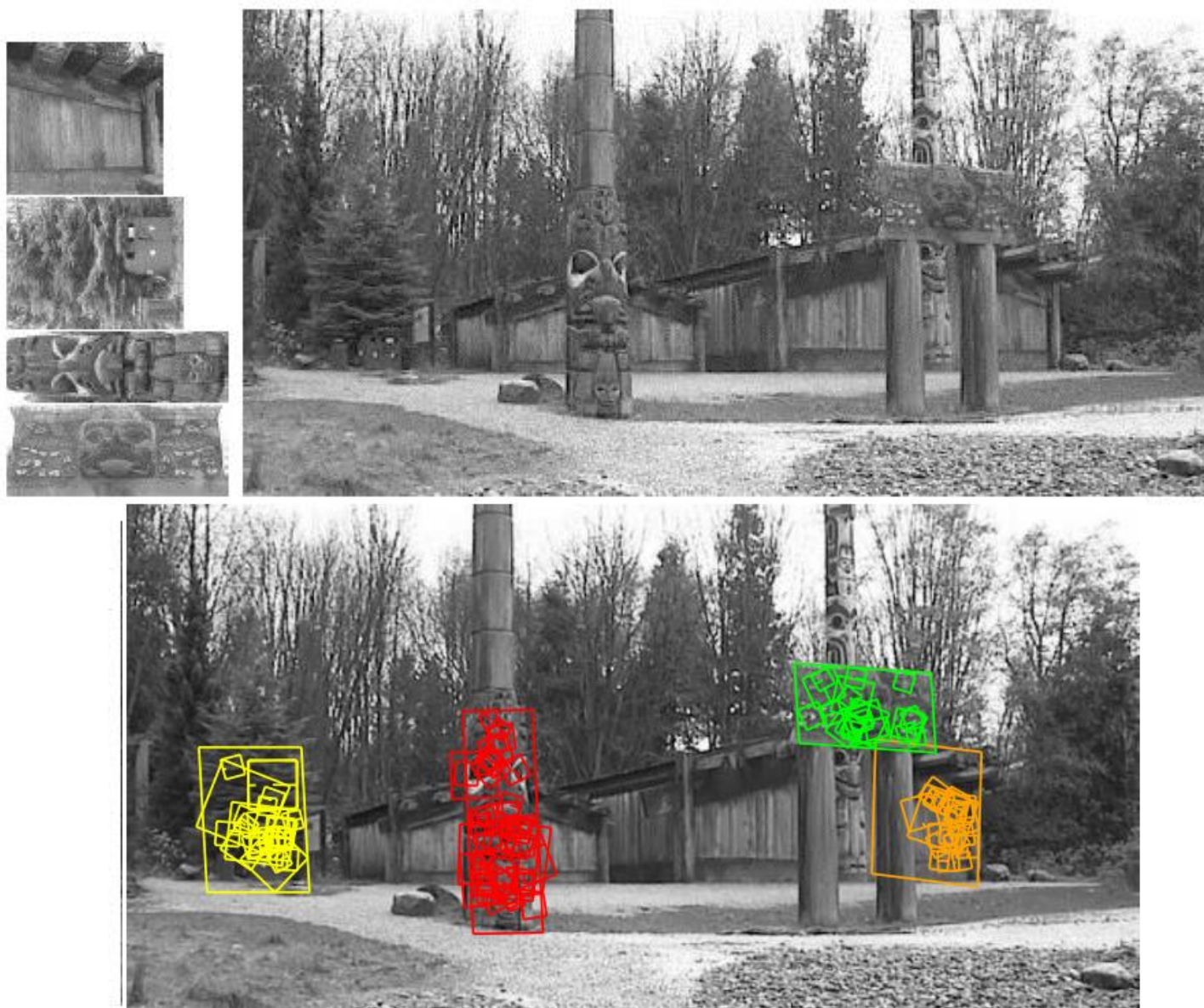


Figure 13: This example shows location recognition within a complex scene. The training images for locations are shown at the upper left and the 640x315 pixel test image taken from a different viewpoint is on the upper right. The recognized regions are shown on the lower image, with keypoints shown as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

SIFT features impact

SIFT feature paper citations:

Distinctive image features from scale-invariant keypoints DG Lowe -
International journal of computer vision, 2004 - Springer
International Journal of Computer Vision 60(2), 91–110, 2004 cc
2004 Kluwer Academic Publishers. Computer Science Department,
University of British Columbia ... **Cited by 16232 (google)**

A good SIFT features tutorial:

<http://www.cs.toronto.edu/~jepson/csc2503/tutSIFT04.pdf>

By Estrada, Jepson, and Fleet.

The original SIFT paper:

<http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>

Now we have

- Well-localized feature points
- Distinctive descriptor
- Now we need to
 - match pairs of feature points in different images
 - Robustly compute homographies
(in the presence of errors/outliers)