6.819 / 6.869: Advances in Computer Vision

MIT COMPUTER VISION

Image Features:

Harris detector & SIFT

Instructor: Aude Oliva

Lecture TR 9:30AM – 11:00AM
(Room 34-101)

Website:
http://6.869.csail.mit.edu/fa15/

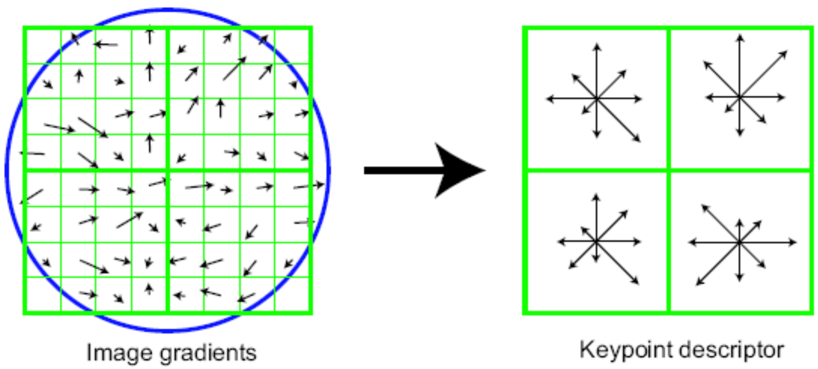
Finding the same thing across images

Instances Find these two objects

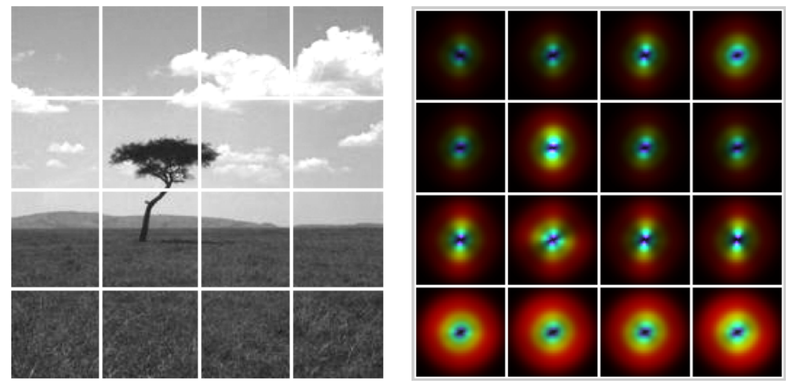
Categories Find a bottle:

Finding similar instances

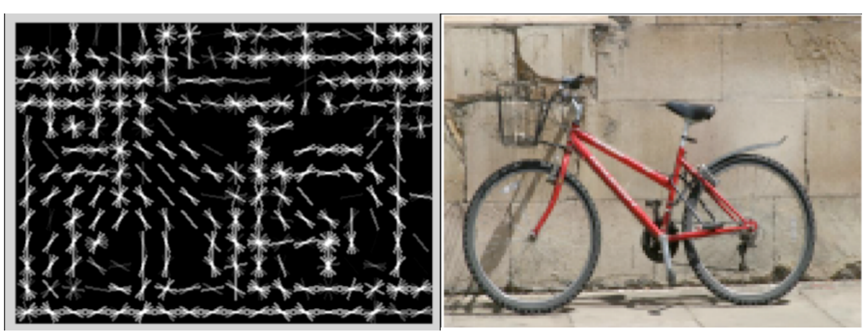
SIFT: Scale-Invariant Feature Transform
(Lowe, 1999)



Gist: Grid of gabors
(Oliva & Torralba, 2001)

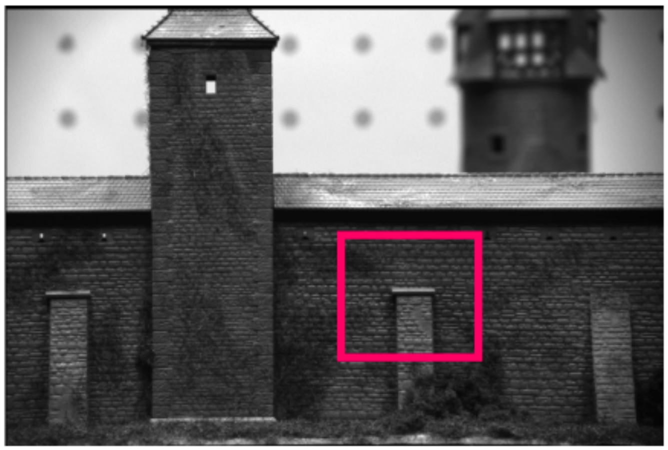


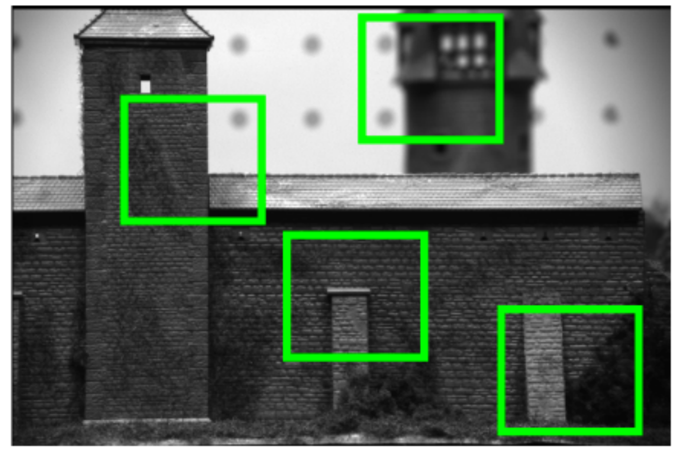
HOG: Histograms of oriented gradients
(Dalal & Triggs CVPR 05)



DPM: Deformable Part Models
(Felzenszwalb, McAllester, Ramanan, 2008)

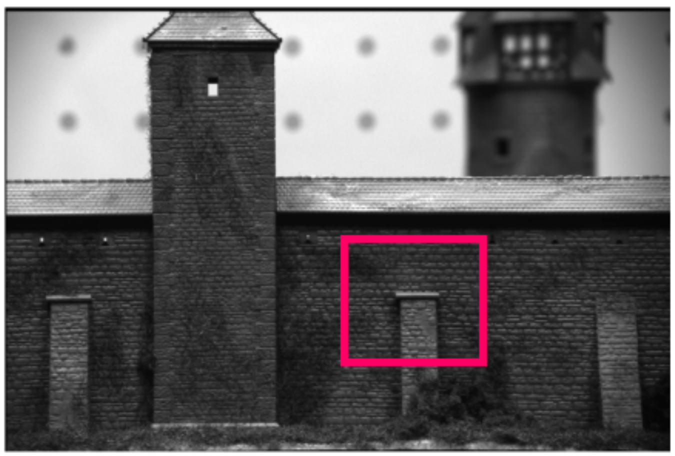
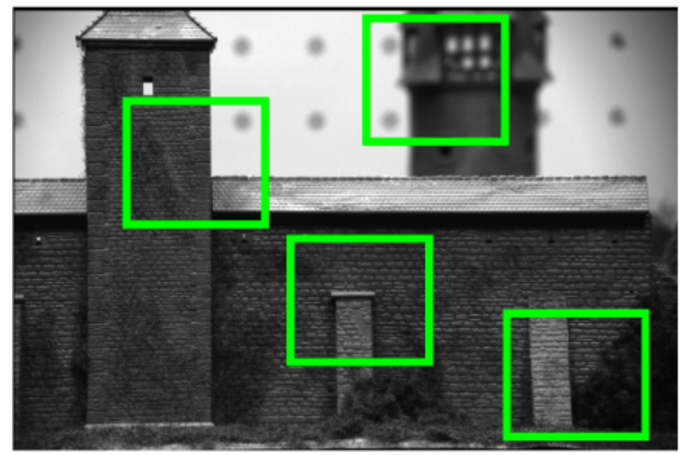
Goal: Find the same patch





Task: find the most similar patch in a second image

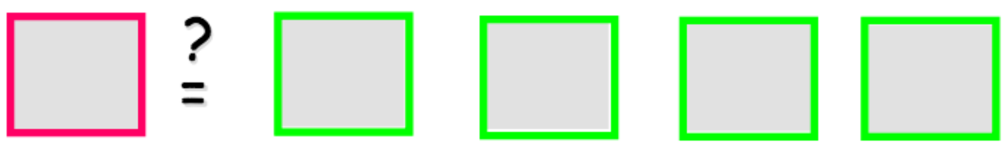
Not all patches are created equal

Intuition: this would be a good patch for matching, since it is very distinctive (there is only one patch in the second frame that looks similar)

Not all patches are created equal

Intuition: this would be a bad patch for matching, since it is **not** very distinctive (there are many similar patches in the second frame)



Building a Panorama

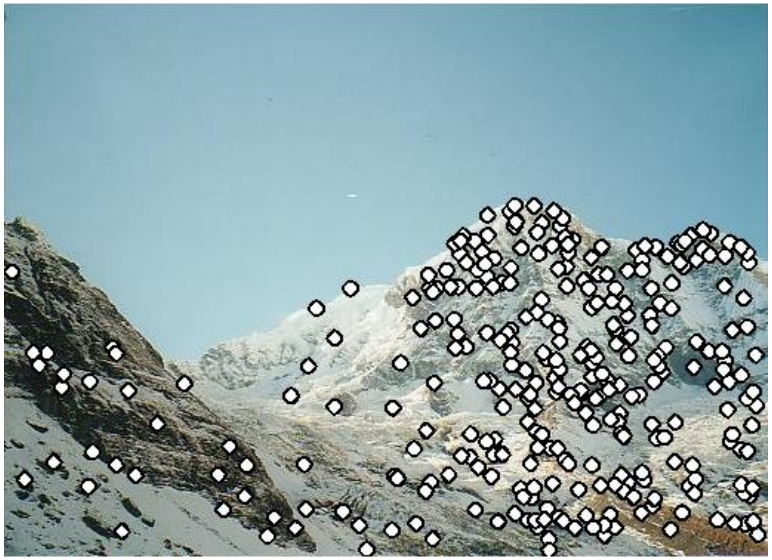
M. Brown and D. G. Lowe. Recognising Panoramas. ICCV 2003

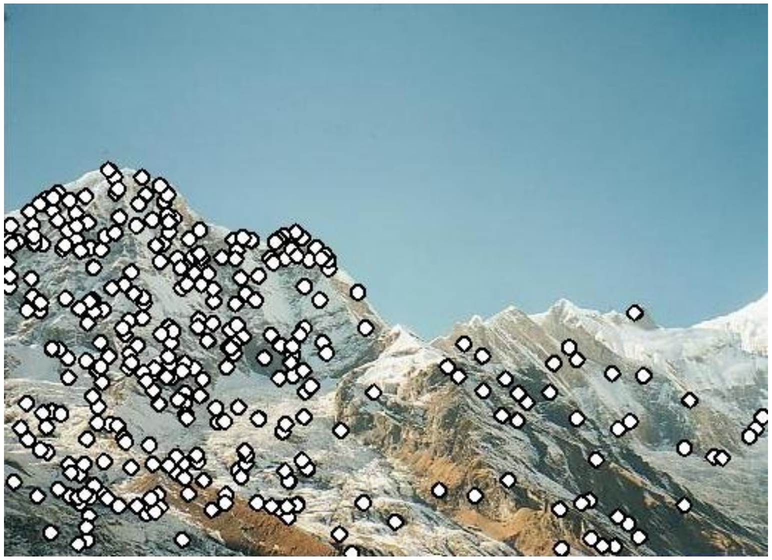
How do we build a panorama?

- We need to match (align) images

Matching with Features

Detect feature points in both images

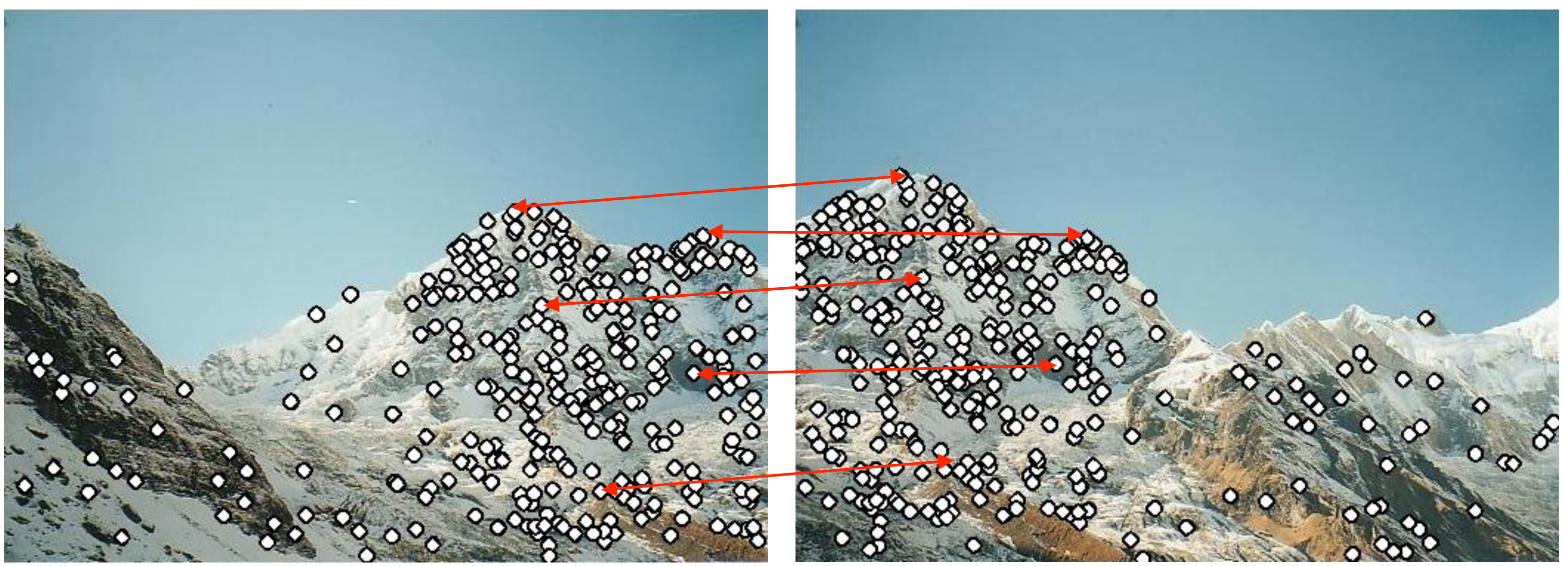




Matching with Features

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

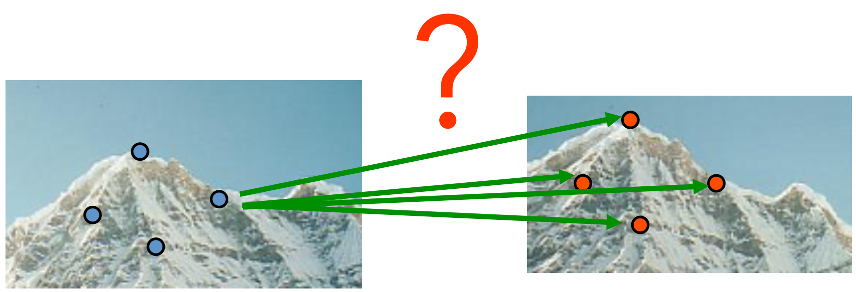
counter-example:


no chance to match!

We need a repeatable detector

Matching with Features

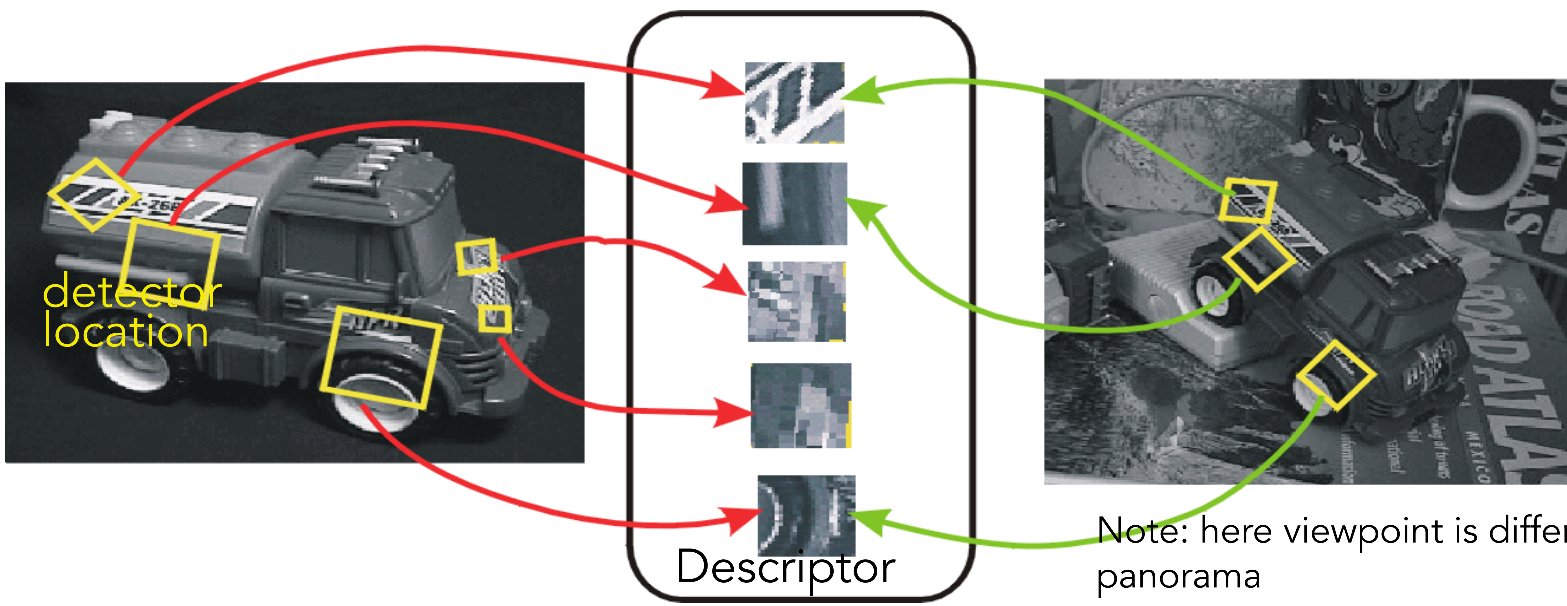
- Problem 2:
  - For each point correctly recognize the corresponding one



We need a reliable and distinctive descriptor

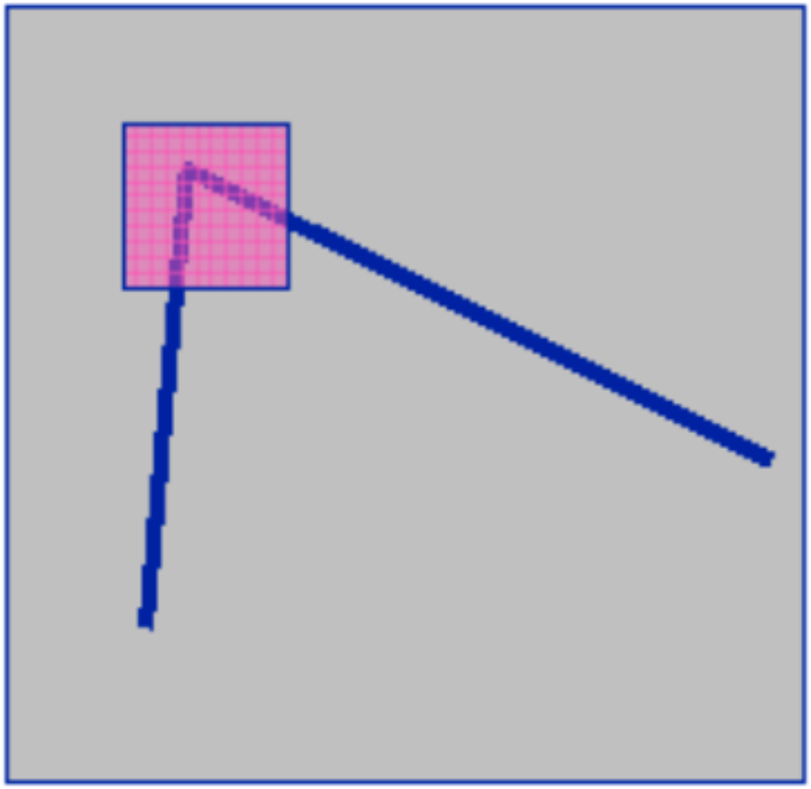
Preview

- Detector: detect same scene points independently in both images
- Descriptor: encode local neighboring window
- Correspondence: find most similar descriptor in other image



Note: here viewpoint is different, not panorama

Basic Ideas of Corner Points



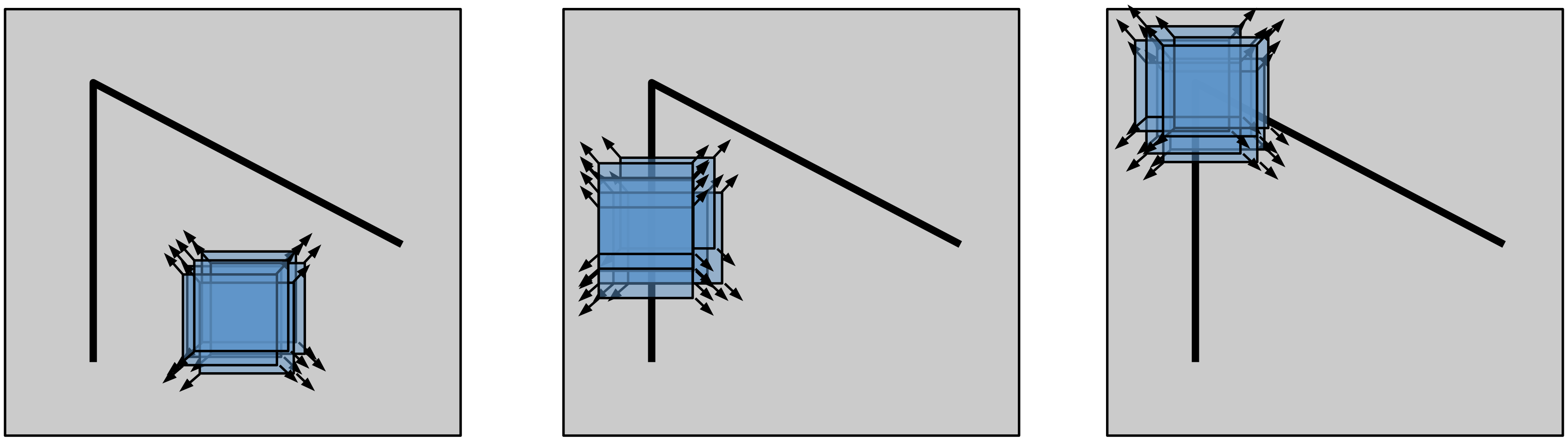
Junctions and Corners: they are the most stable features over changes in viewpoint

Intuitively, there is a large variations in the neighborhood of the point in all directions

We should easily recognize the point by looking at intensity values within a small window

Shifting the window in any direction should yield a large change in appearance.

Corner Detector: Basic Idea



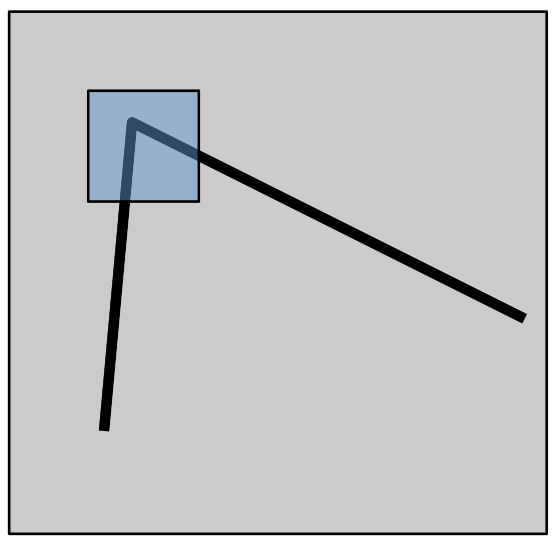
"flat" region:
no change in all directions

"edge":
no change along the edge direction

"corner":
significant change in all directions

Harris 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



Harris Detector: Maths & Intuition

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

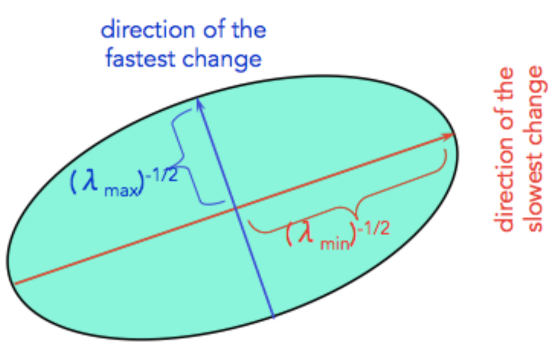
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

Ellipse

Window function

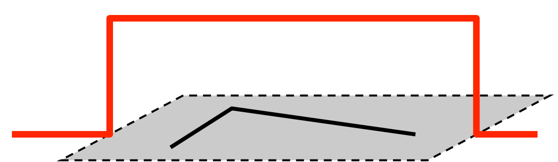
Shifted intensity

Intensity



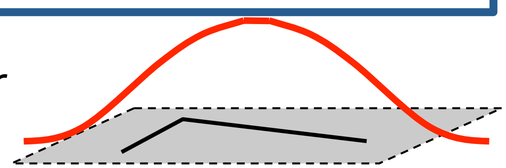
For nearly constant patches, this will be near 0.
For very distinctive patches, this will be larger.
Hence... we want patches where E(u,v) is LARGE.

Window function w(x,y) =



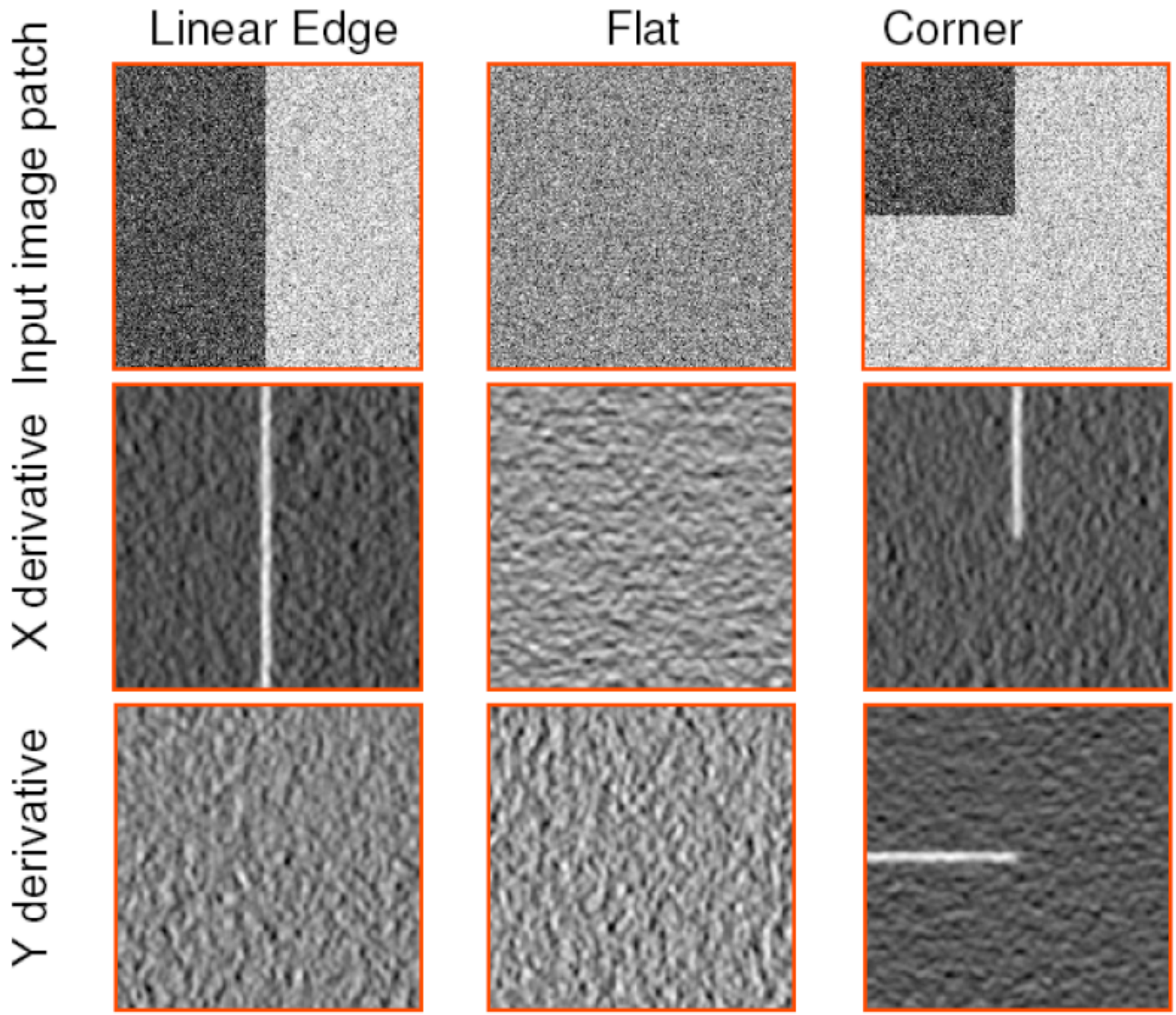
1 in window, 0 outside

or



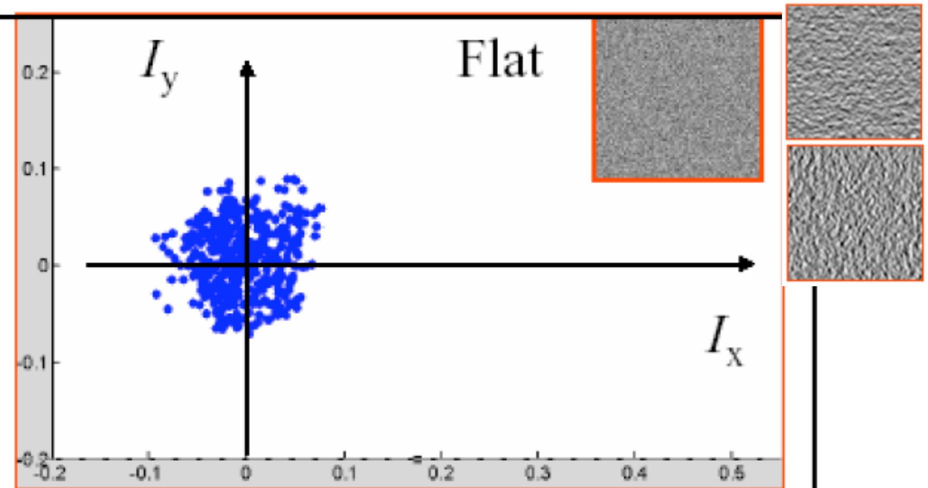
Gaussian

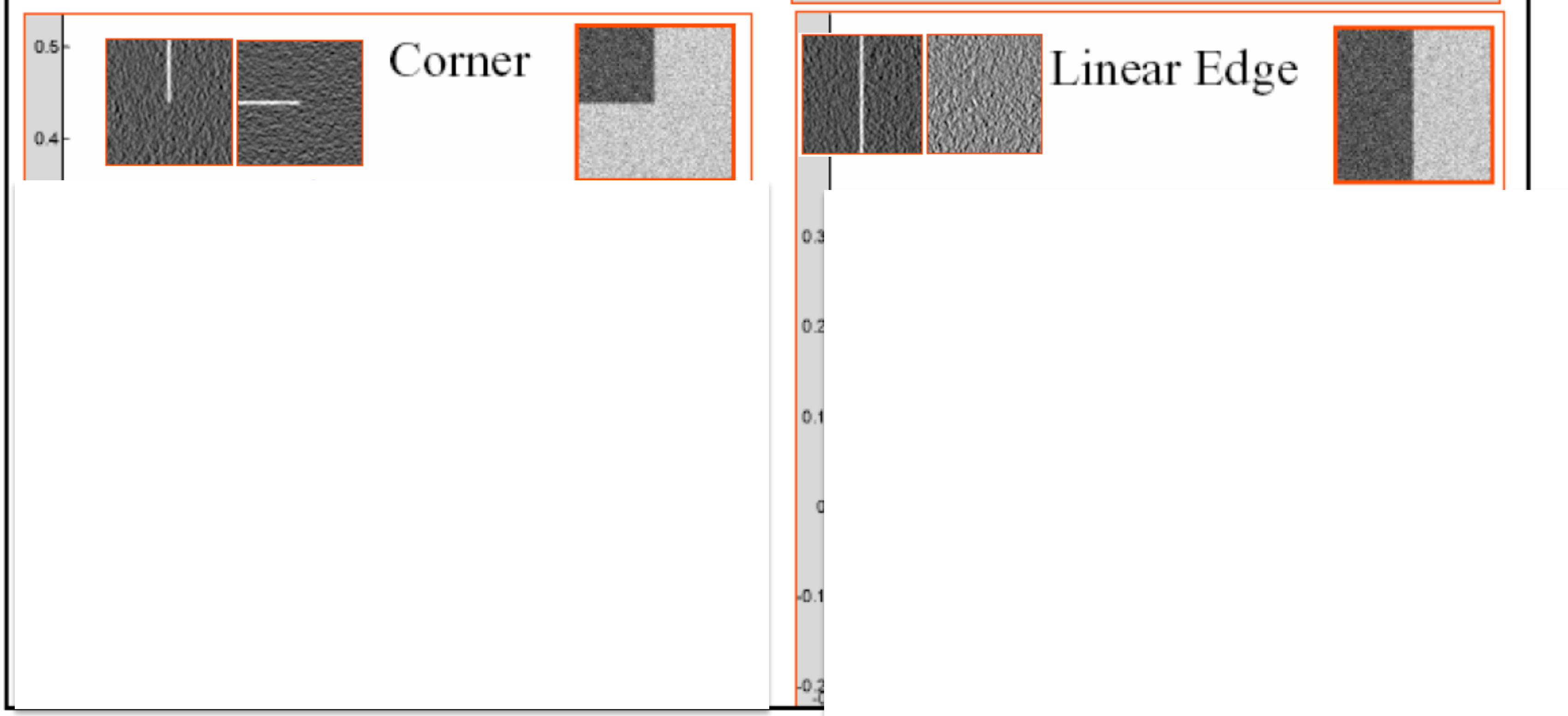
Intuitive way to understand Harris



Plotting Derivatives as 2D points

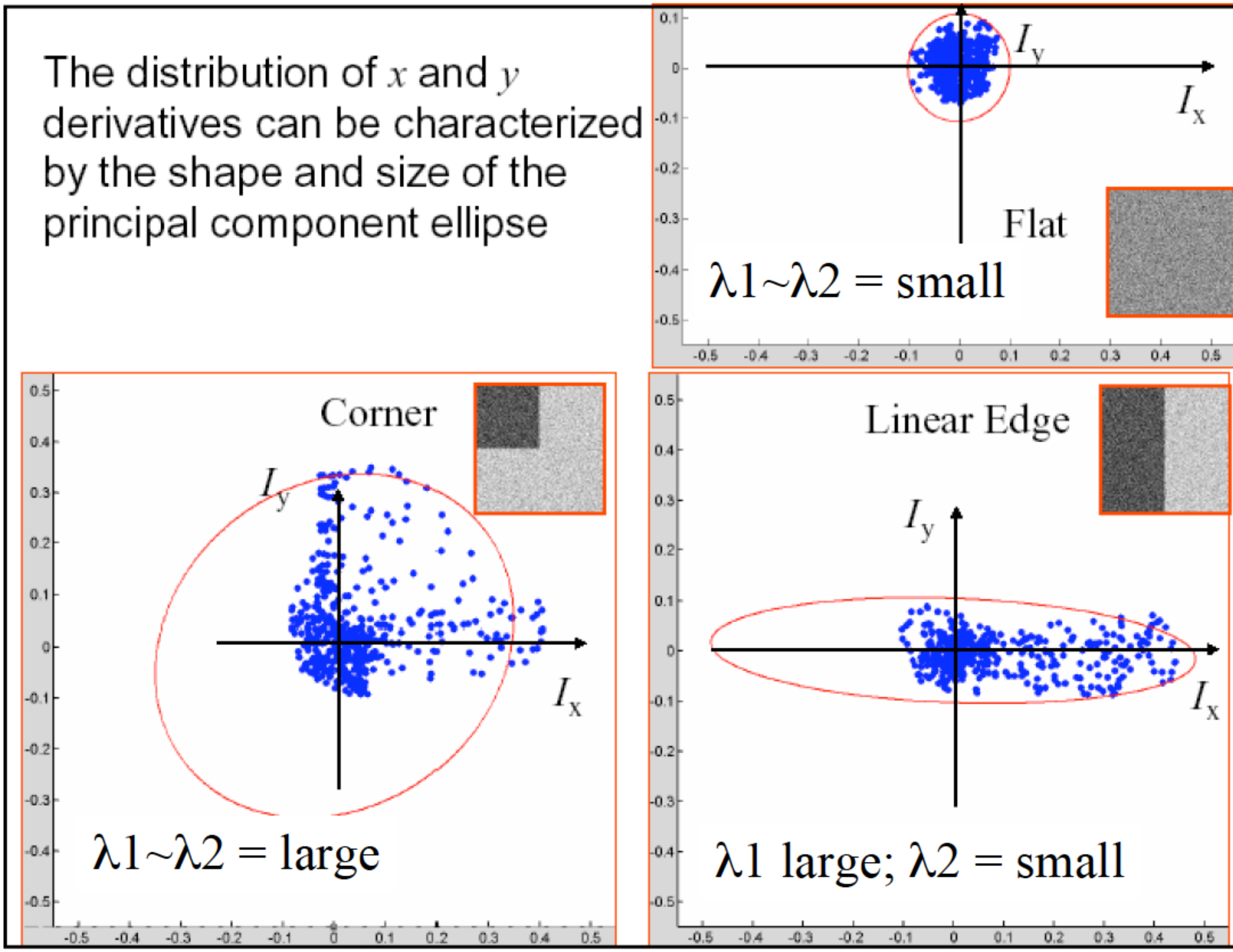
The distribution of the x and y derivatives is very different for all three types of patches





Fitting ellipses to each set of points

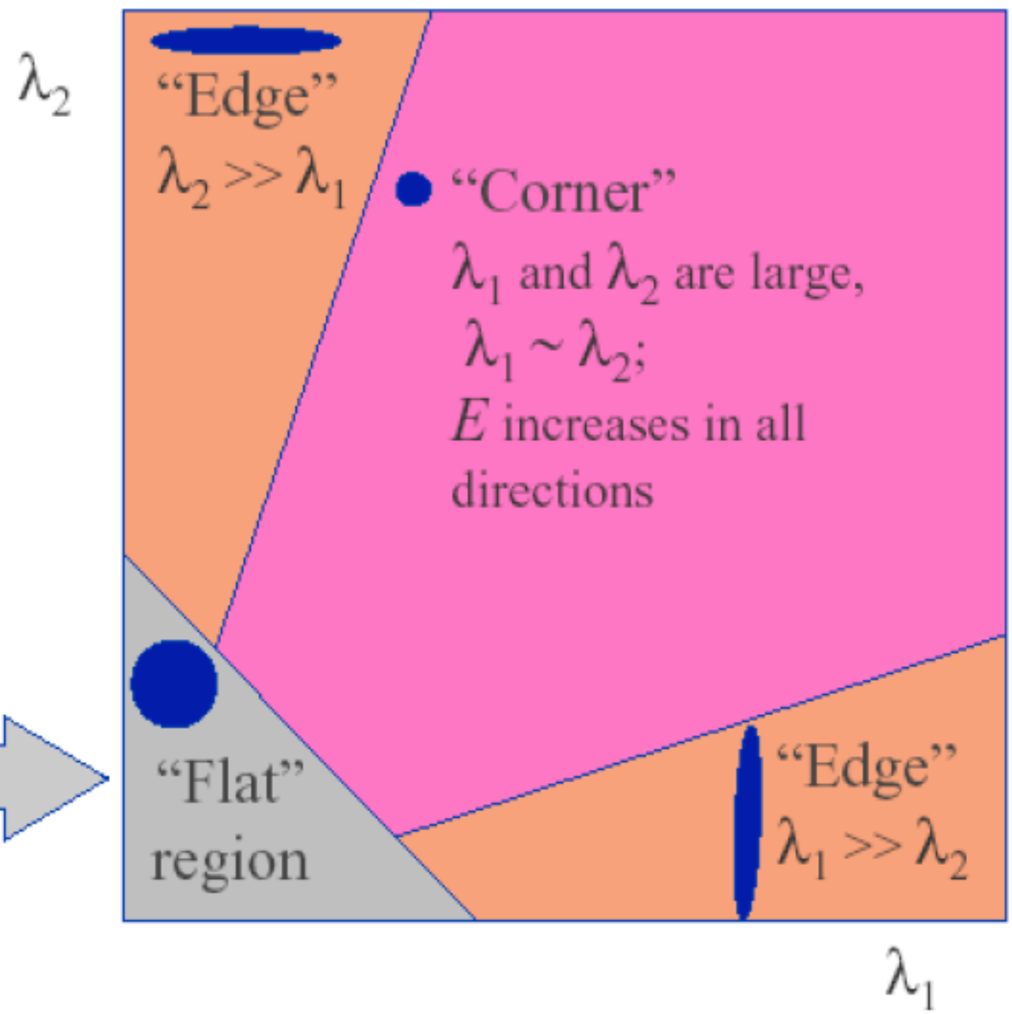
The distribution of x and y derivatives can be characterized by the shape and size of the principal component ellipse



Classification via Eigenvalues

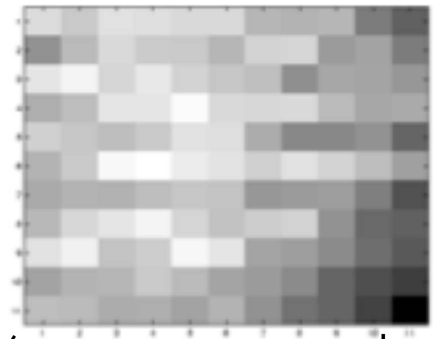
Classification of image points using eigenvalues of M:

λ_1 and λ_2 are small; E is almost constant in all directions

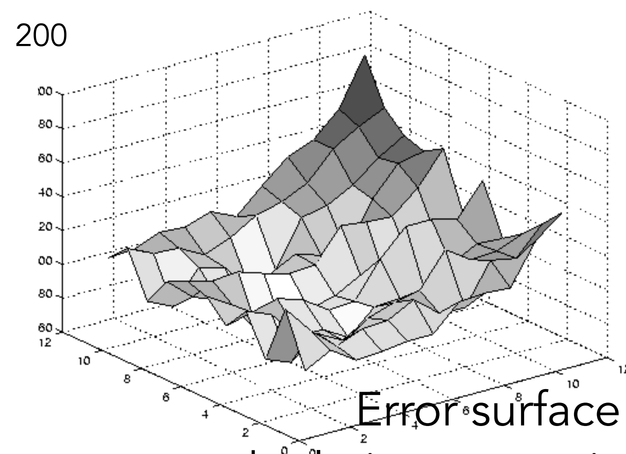


Selecting Good Features

Image patch



(contrast auto-scaled)



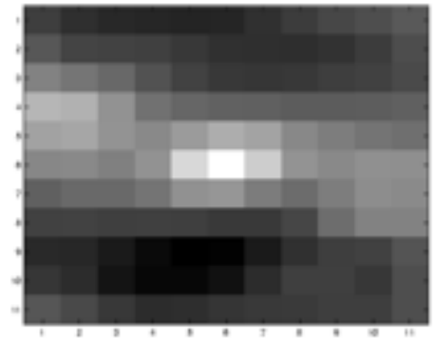
Error surface

(vertical scale exaggerated relative to previous plots)

small λ_1, small λ_2

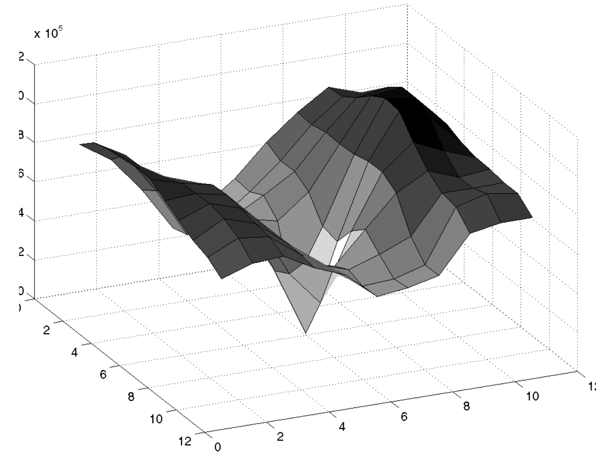
Selecting Good Features

Image patch



Error surface

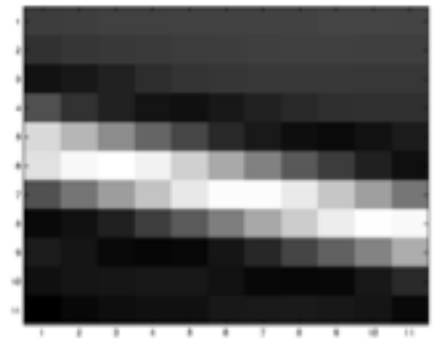
12x10^5



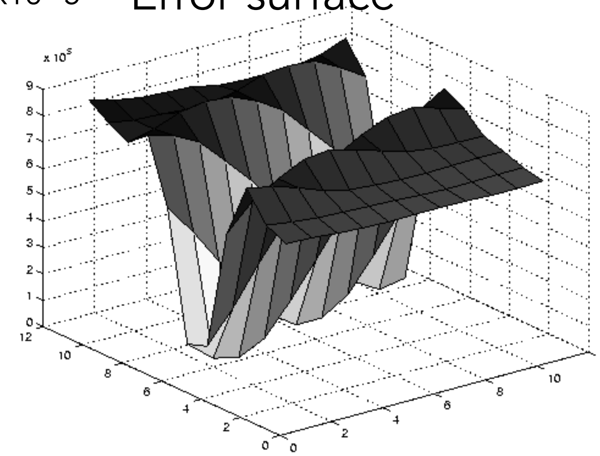
λ_1 and λ_2 are large

Selecting Good Features

Image patch



9x10^5 Error surface

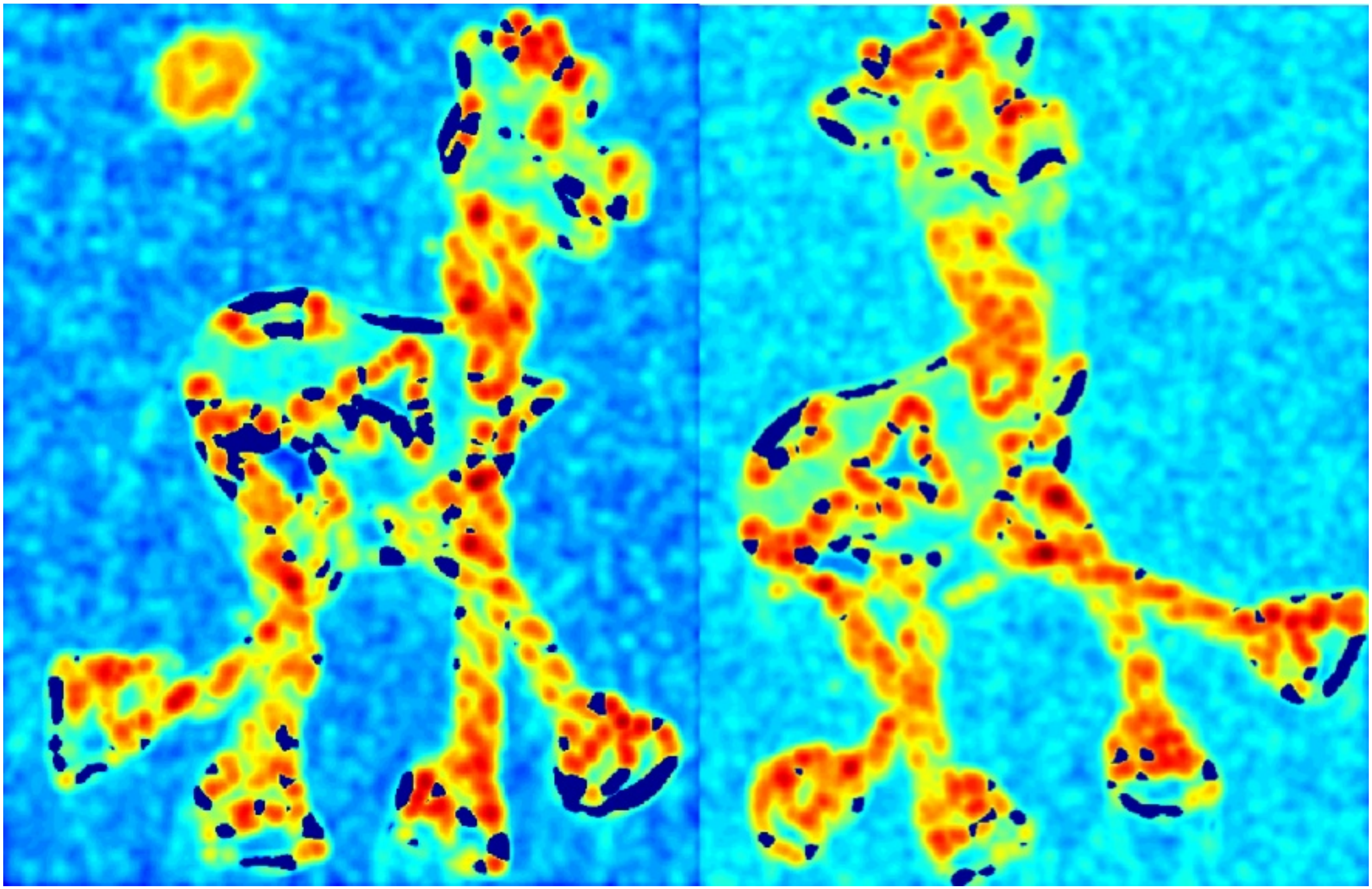


large λ_1, small λ_2

Harris Detector: Workflow

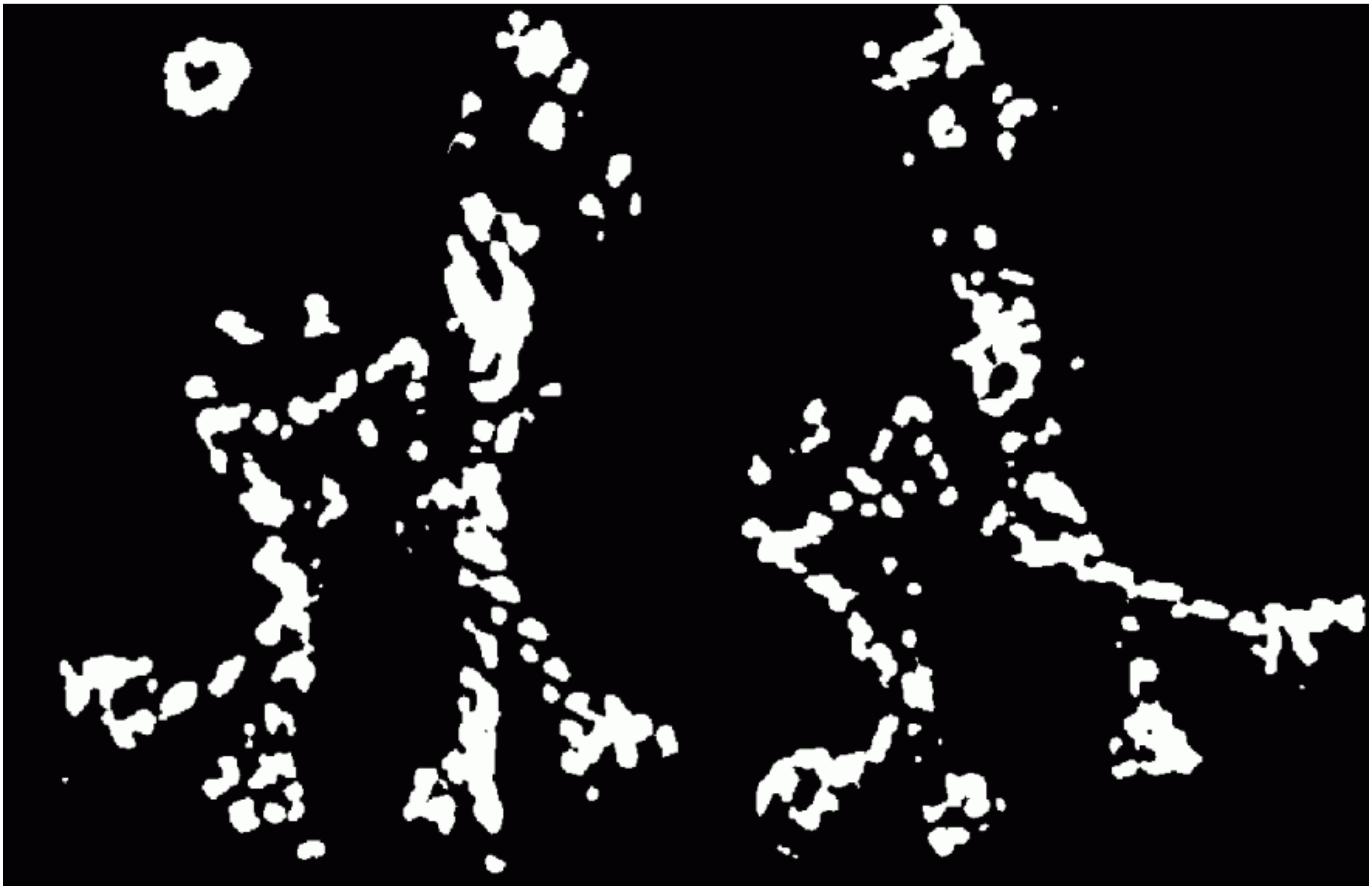
Harris Detector: Workflow

Compute corner response R



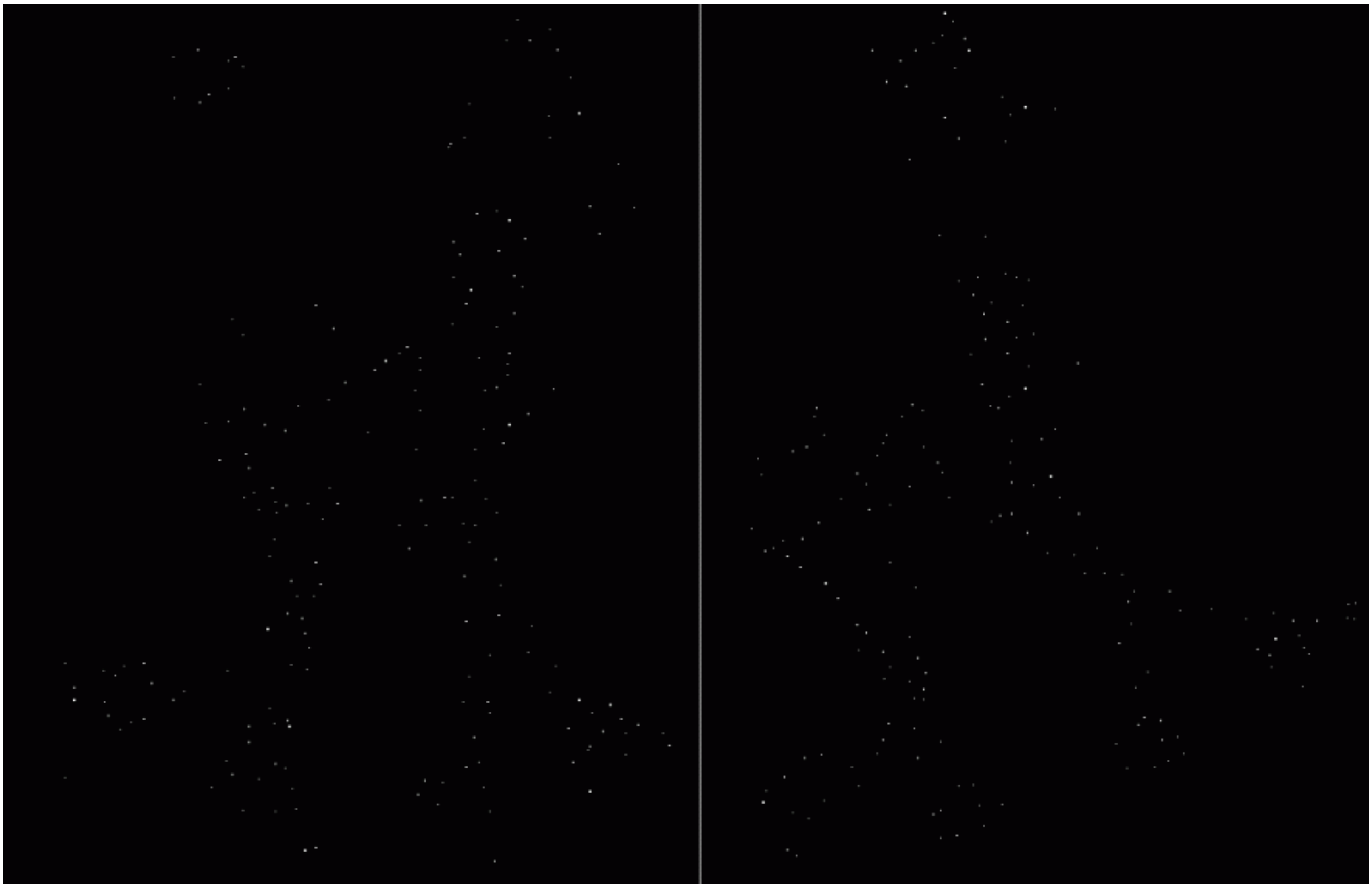
Harris Detector: Workflow

Find points with large corner response: R>threshold



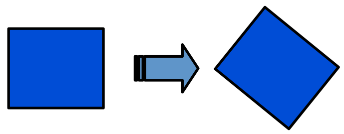
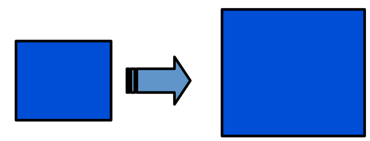
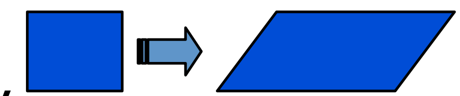
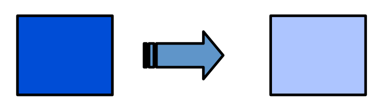
Harris Detector: Workflow

Take only the points of local maxima of R



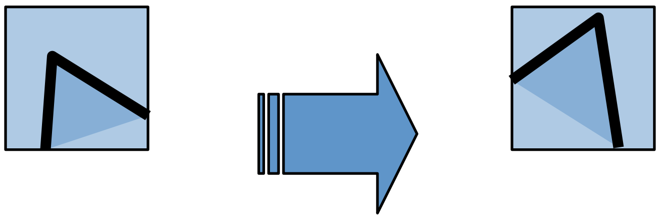
Harris Detector: Workflow

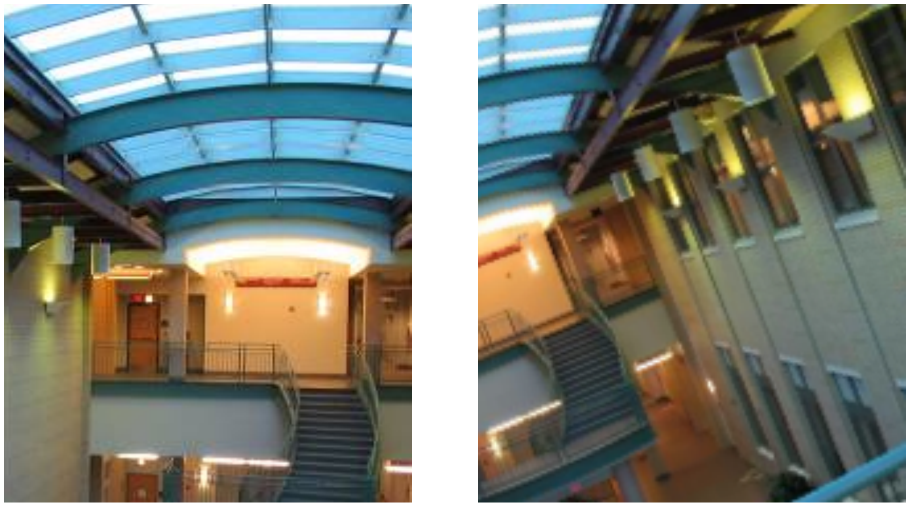
Models of Image Change

- Geometry
  - Rotation 
  - Similarity (rotation + uniform scale) 
  - Affine (scale dependent on direction) valid for: orthographic camera, locally planar object 
- Photometry
  - Affine intensity change ($I \rightarrow a\, I + b$) 

Harris Detector: Some Properties

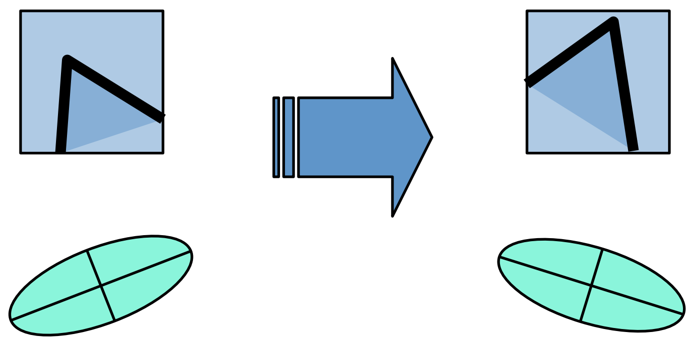
- Rotation invariance?





Harris Detector: Some Properties

- Rotation invariance



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

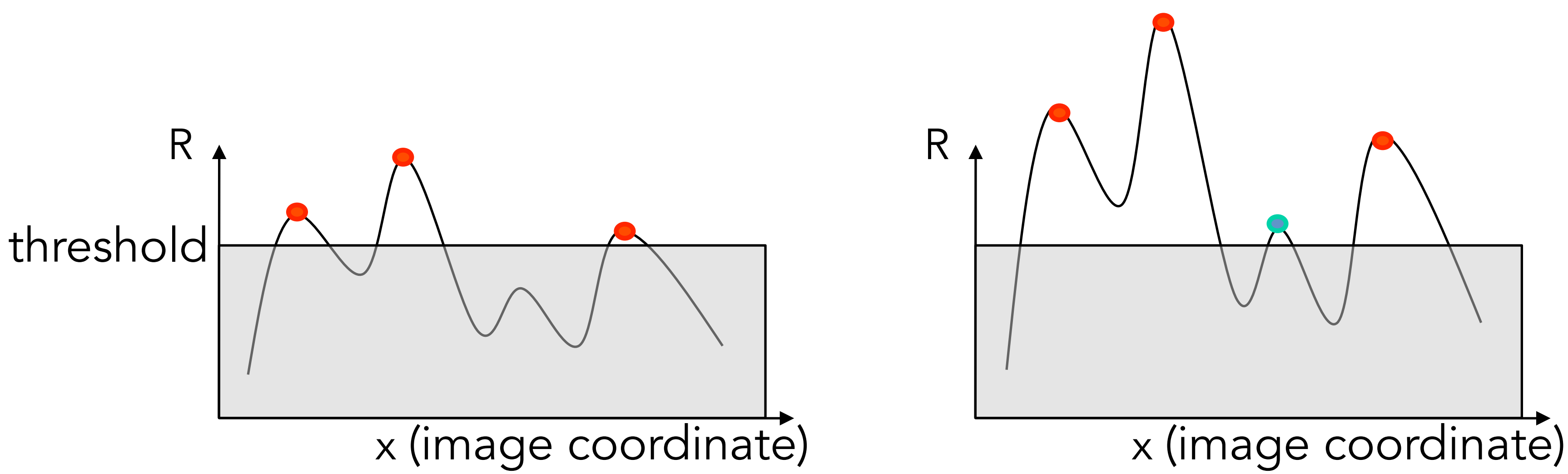
Corner response R is invariant to image rotation

Harris Detector: Some Properties

- Partial invariance to additive and multiplicative intensity changes

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

  ✓ Intensity scaling: $I \rightarrow a\,I$ fine, except for the threshold that's used to specify when R is large enough.



Harris Detector: Some Properties

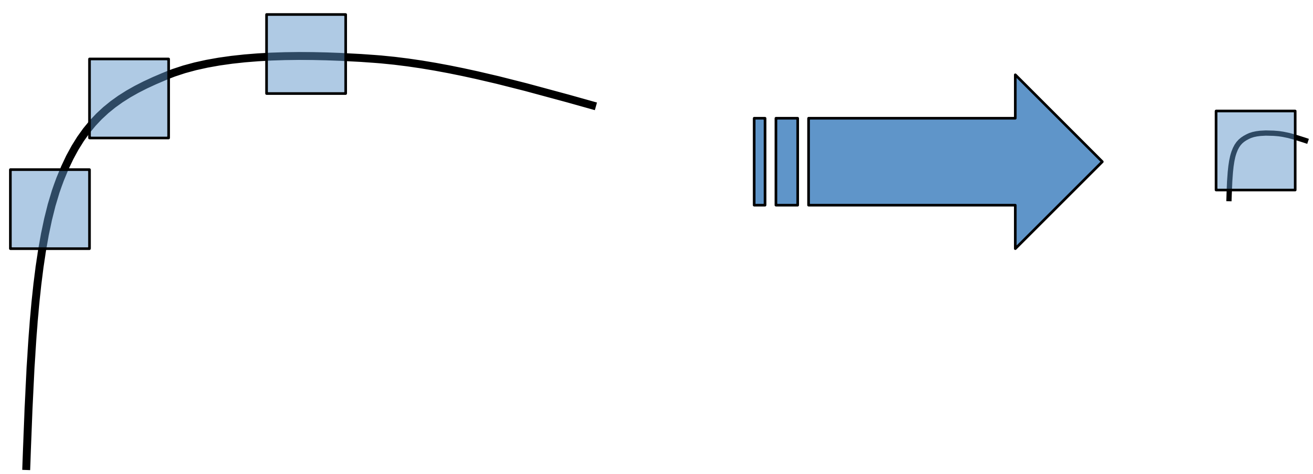
- Invariant to image scale?

image

zoomed image

Harris Detector: Some Properties

- Not invariant to image scale!

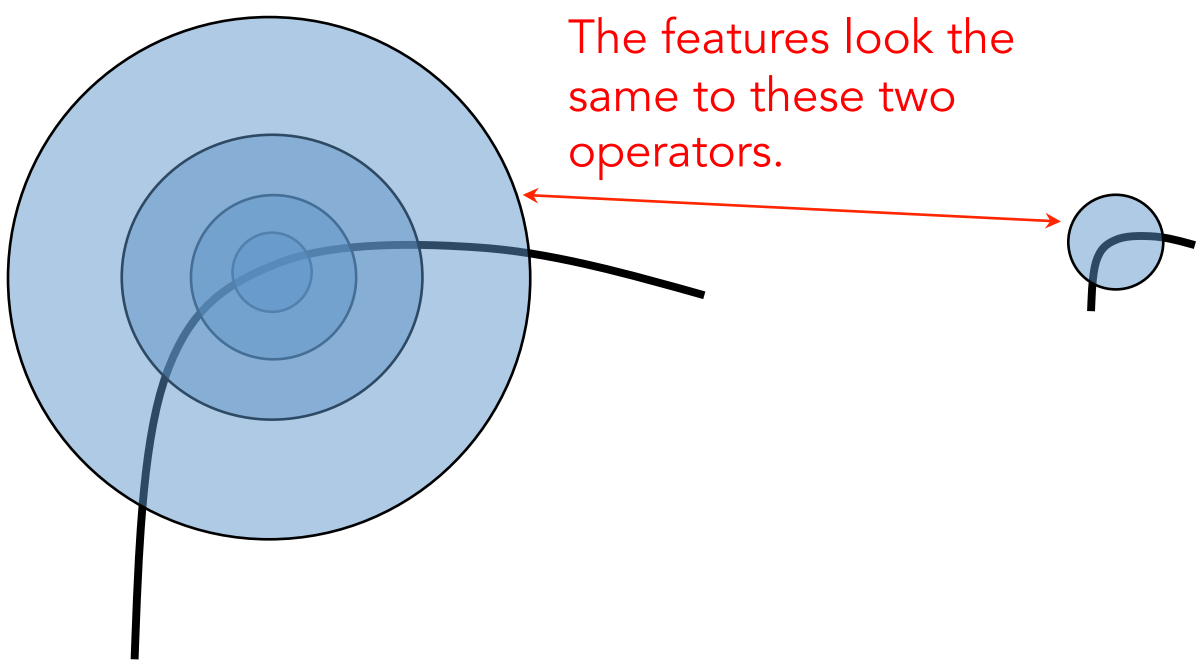


All points will be classified as edges

Corner !

Scale Invariant Detection

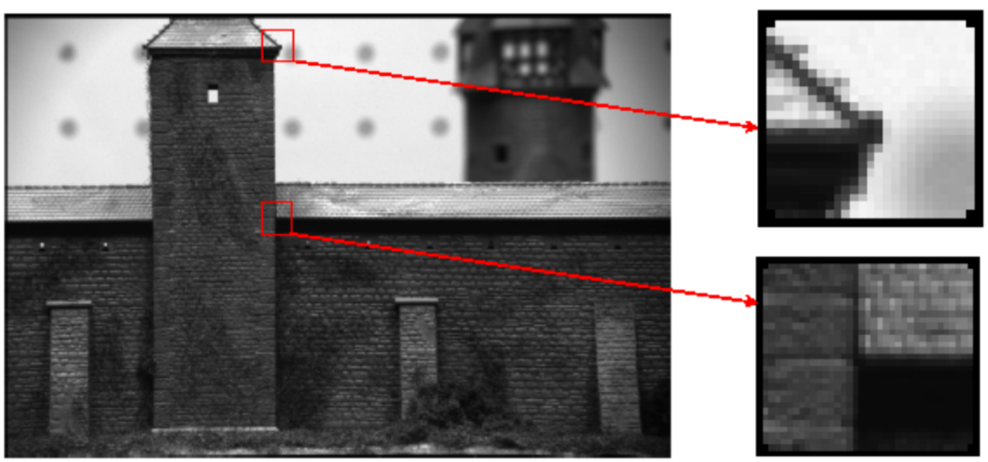
- Consider regions (e.g. circles) of different sizes around a point
- Regions of corresponding sizes will look the same in both images



Finding same instance

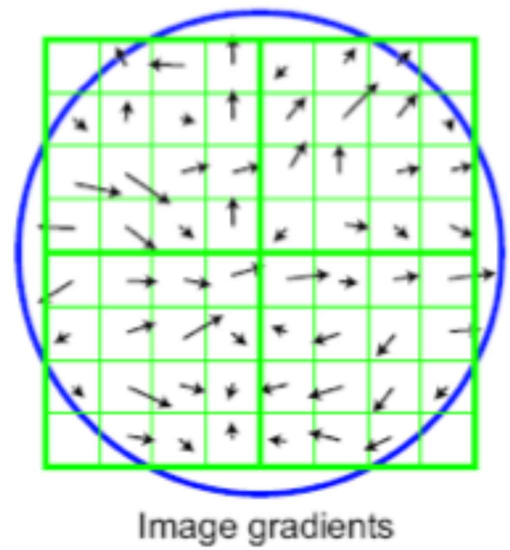
Vision tasks such as stereo and motion estimation require finding corresponding features across two or more views.

Feature point detection



- Harris corner detector
- finding a characteristic scale: DoG or Laplacian of Gaussian

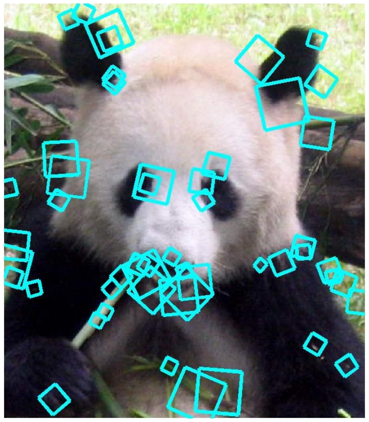
Local image description



SIFT: Scale-Invariant Feature Transform

Scale-Invariant Feature Transform

- Generates image features, “keypoints”
  - invariant to image scaling and rotation
  - partially invariant to change in illumination and 3D camera viewpoint
  - many can be extracted from typical images
  - highly distinctive



SIFT [Lowe]

SIFT Algorithm Stages

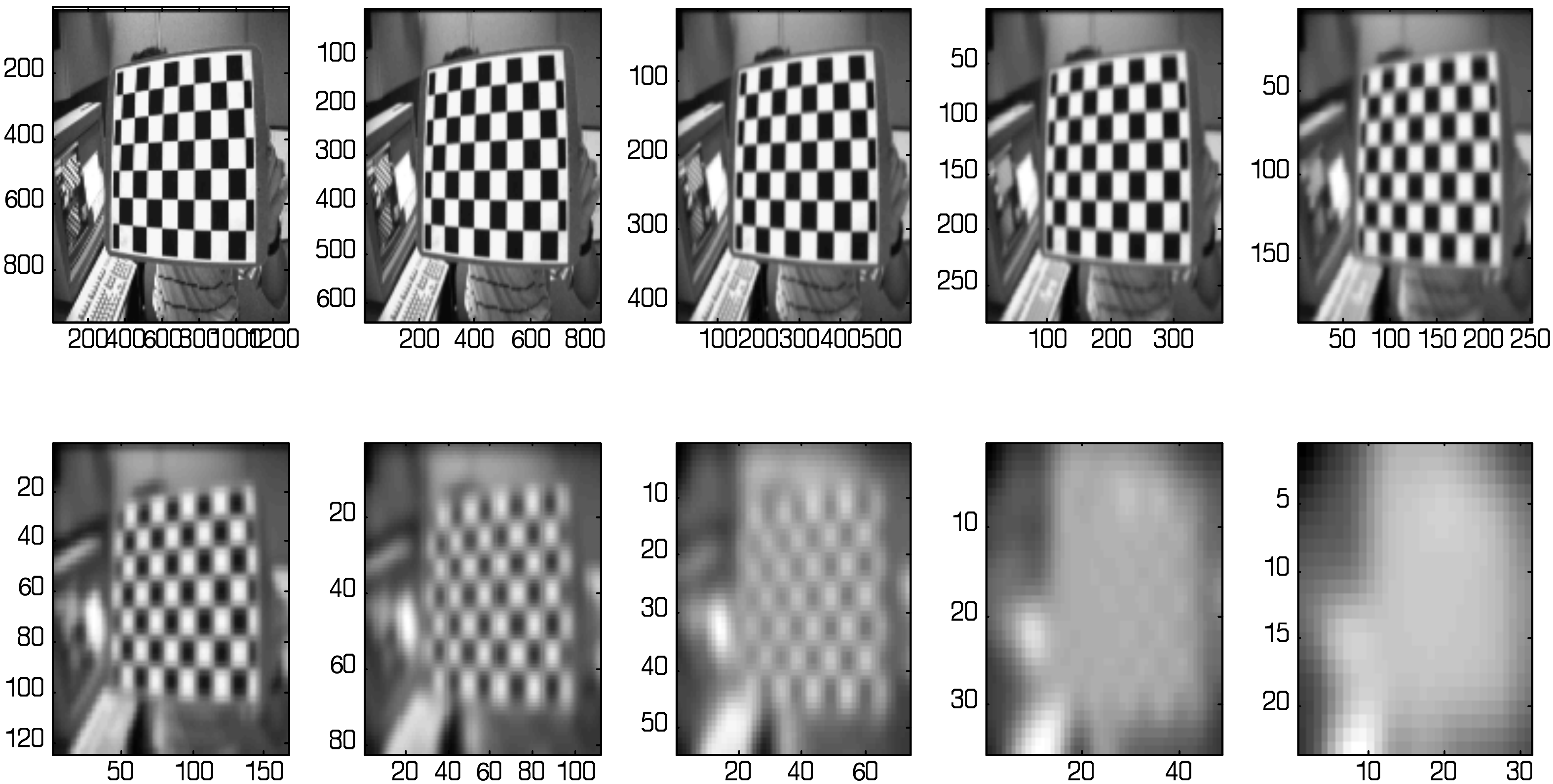
- Scale-space Extrema Detection
  - Uses difference-of-Gaussian function
- Keypoint Localization
  - Sub-pixel location and scale fit to a model
- Orientation assignment
  - 1 or more for each keypoint
- Keypoint descriptor
  - Created from local image gradients

See document: SIFT-tutorial.pdf

Scale Space

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

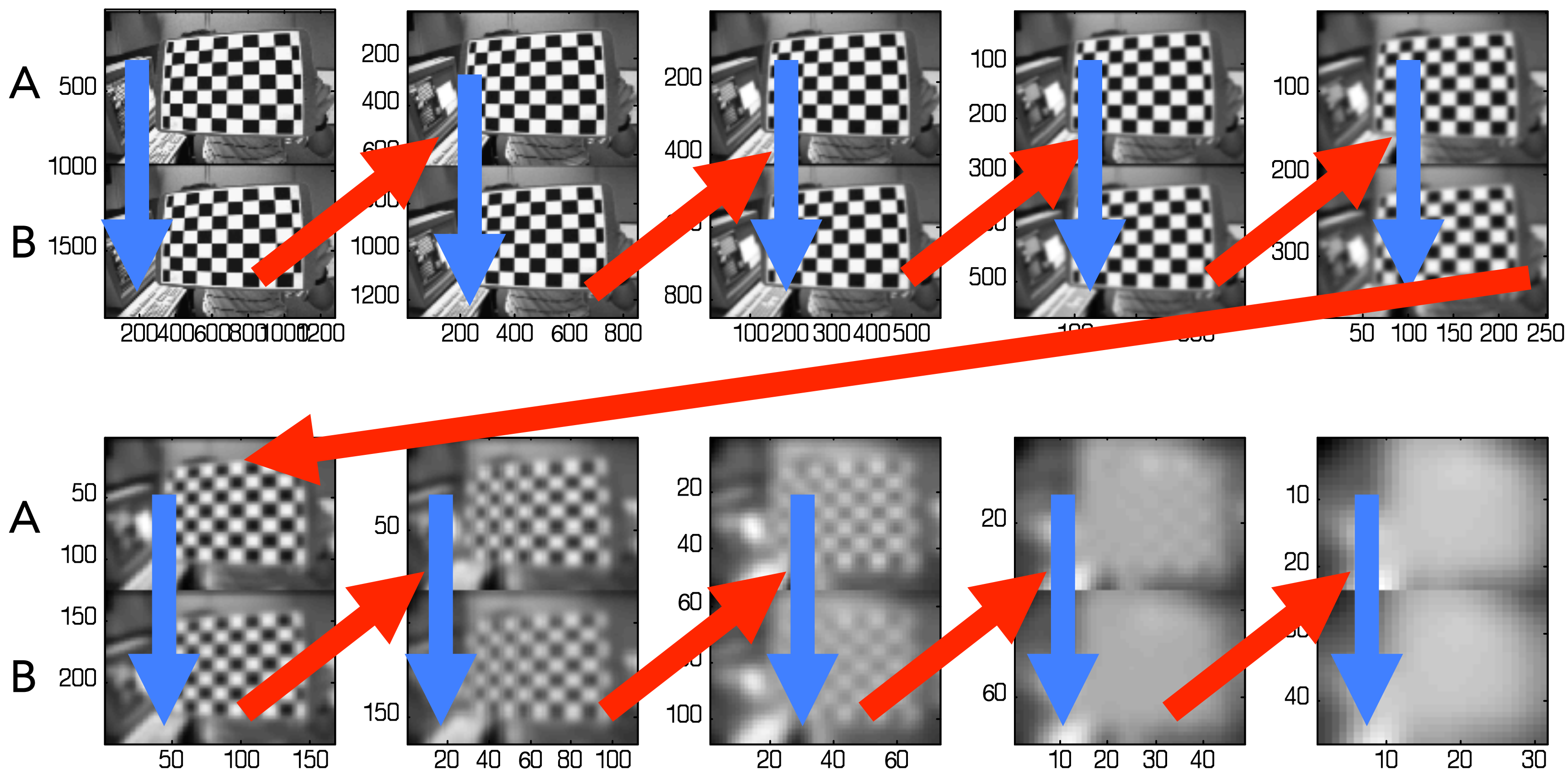
$$G(x, y, \sigma) = \frac{1}{2\pi} e^{-(x^2+y^2)/2\sigma^2}$$



Difference Of Gaussian Pyramid

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

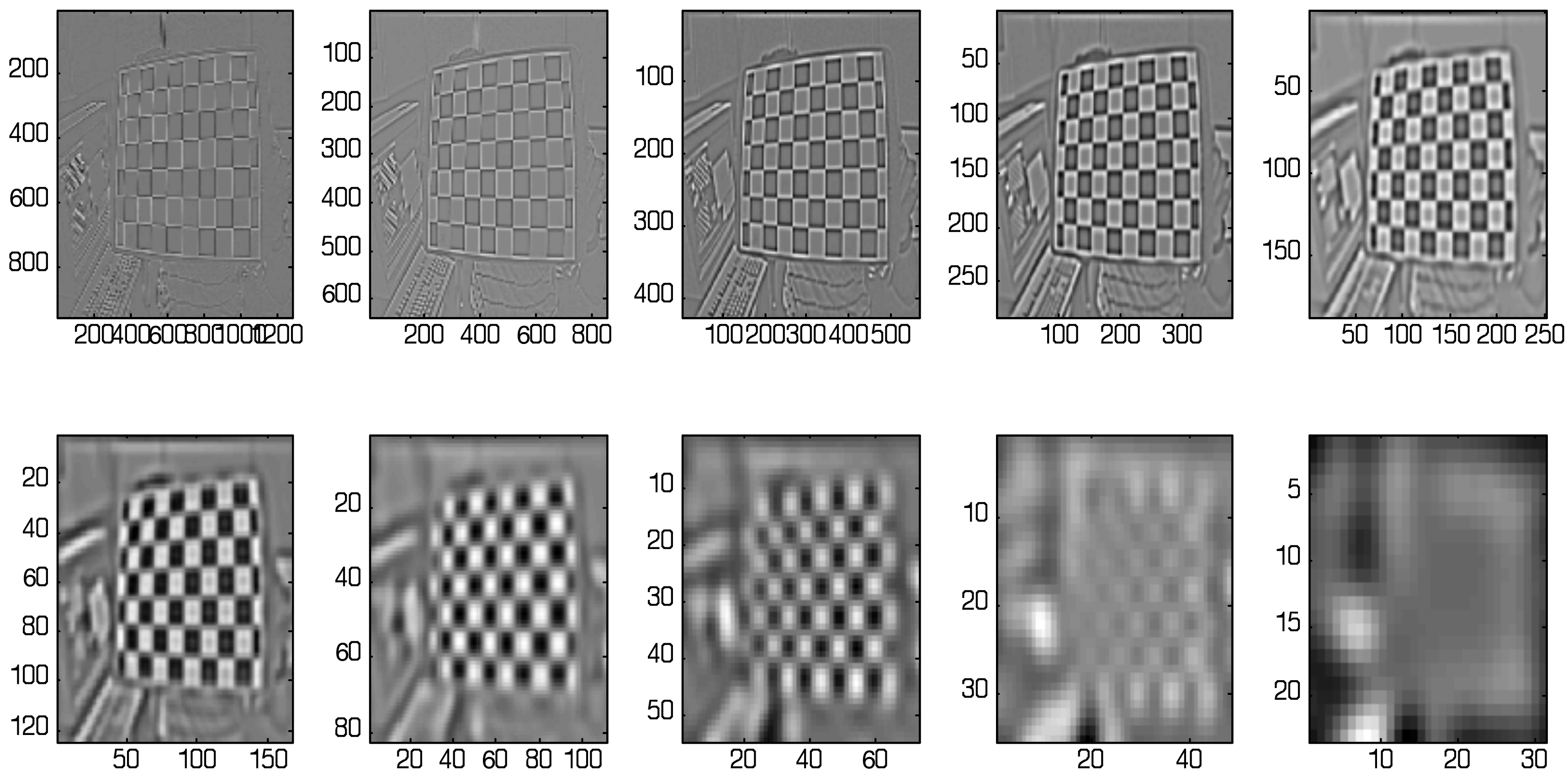
Blur & Resample



Difference Of Gaussian Pyramid

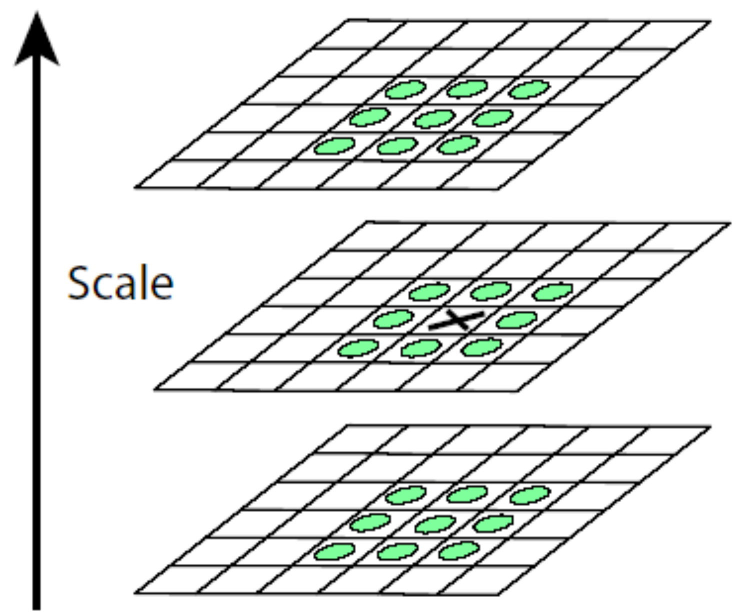
$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma) * I(x, y)$$
$$= L(x, y, k\sigma) - L(x, y, \sigma)$$

A- B

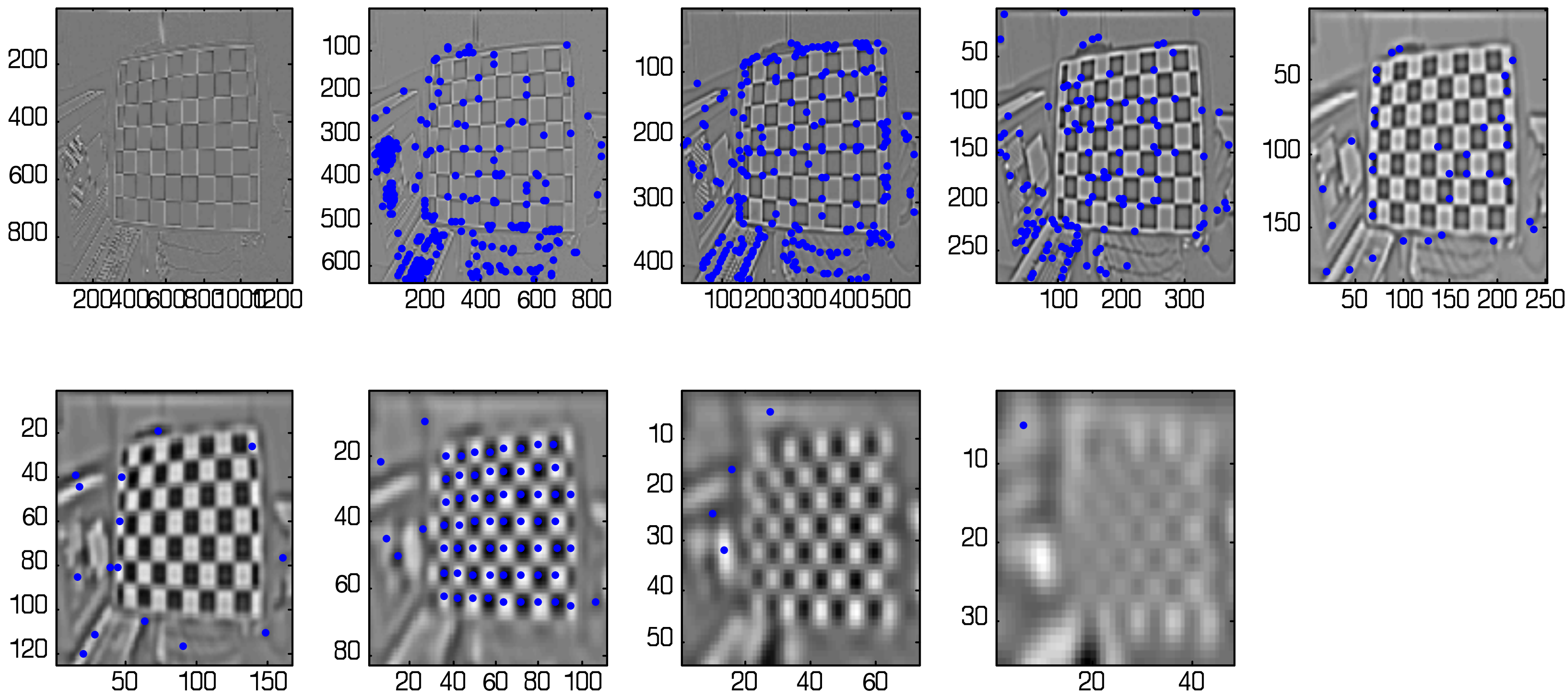


Extrema Detection

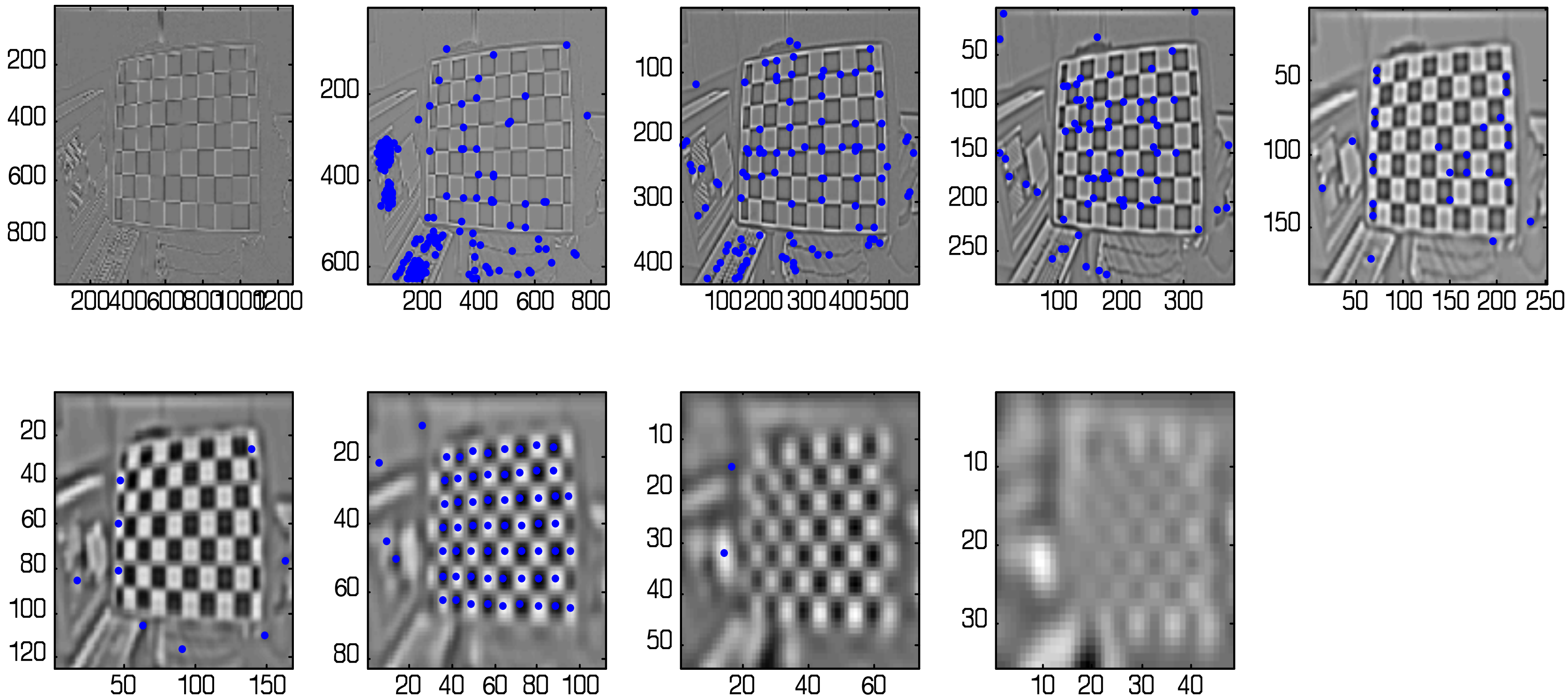
- Keypoint must be a minima or maxima of its 8 neighbors at it's scale and the 9 neighbors above and 9 below.



Extrema Detection

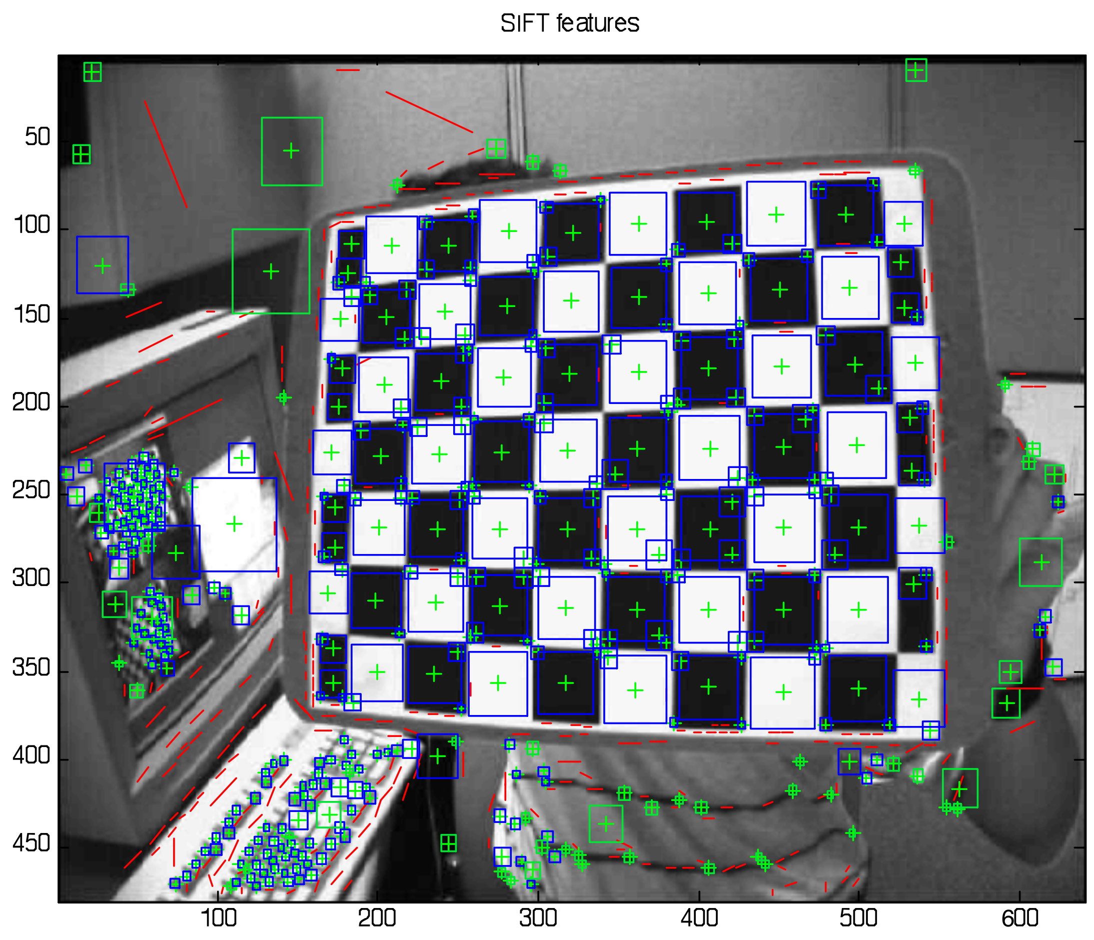


Keypoint Localization and Refinement

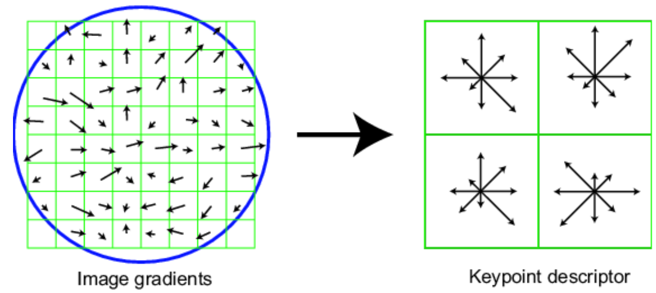


Throw out points that have low contrast
Remove points that are too “edgy”.

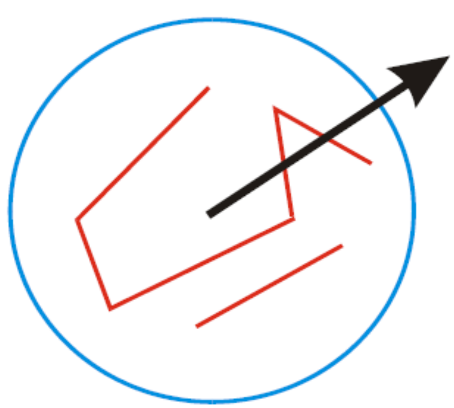
Keypoint Localization and Refinement

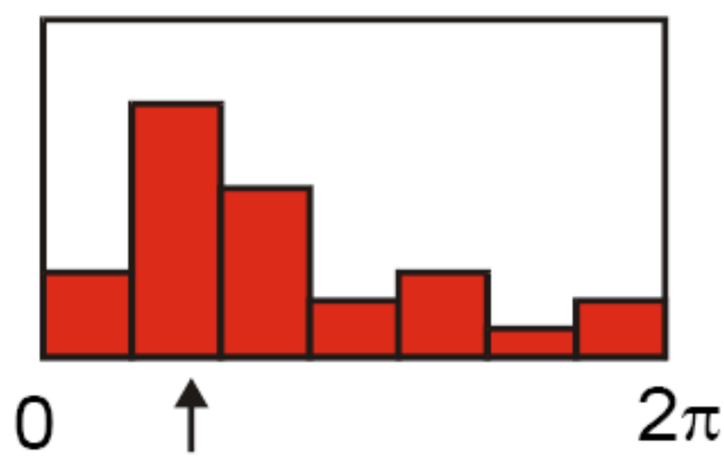


Orientation Assignment

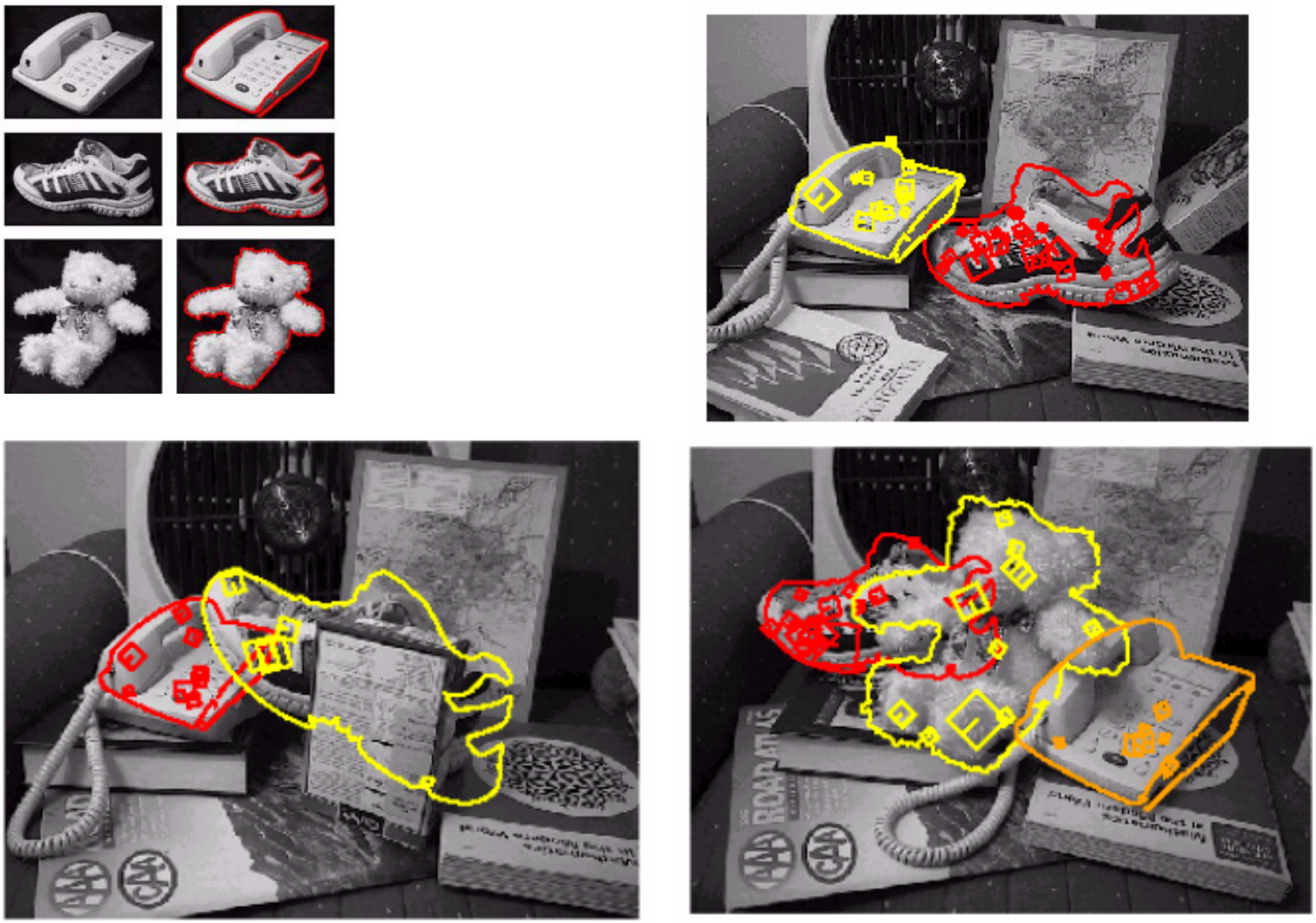


- Create histogram of local gradient directions computed at selected scale
- Assign canonical orientation at peak of smoothed histogram
- Each keypoint specifies stable 2D coordinates (x, y, scale, orientation)





3D object recognition example



SIFT Review

- Generates image features, “keypoints”
  - invariant to image scaling and rotation
  - partially invariant to change in illumination and 3D camera viewpoint
  - many can be extracted from typical images
- Each “keypoint” has an associated **descriptor** that is
  - Relative to keypoint orientation and scale
  - Is robust to small affine transformations.

Download MATLAB Toolbox for the LabelMe Image Database

The LabelMe Matlab toolbox is designed to allow you to download and interact with the images and annotations in the LabelMe database. The toolbox contains functions for plotting and querying the annotations, computing statistics, dealing with synonyms, etc. This page gives a step-by-step overview of the main toolbox functionalities.

Download

There are two ways to download the Matlab toolbox

1. Github repository

We maintain the latest version of the toolbox on gitl
then run "git clone https://github.com/CSAILVision,
version by running "git pull" from inside the project

2. Zip file

The zip file is a snapshot of the latest source code or

SIFT descriptor

Here we provide a function to compute dense SIFT features as described in:

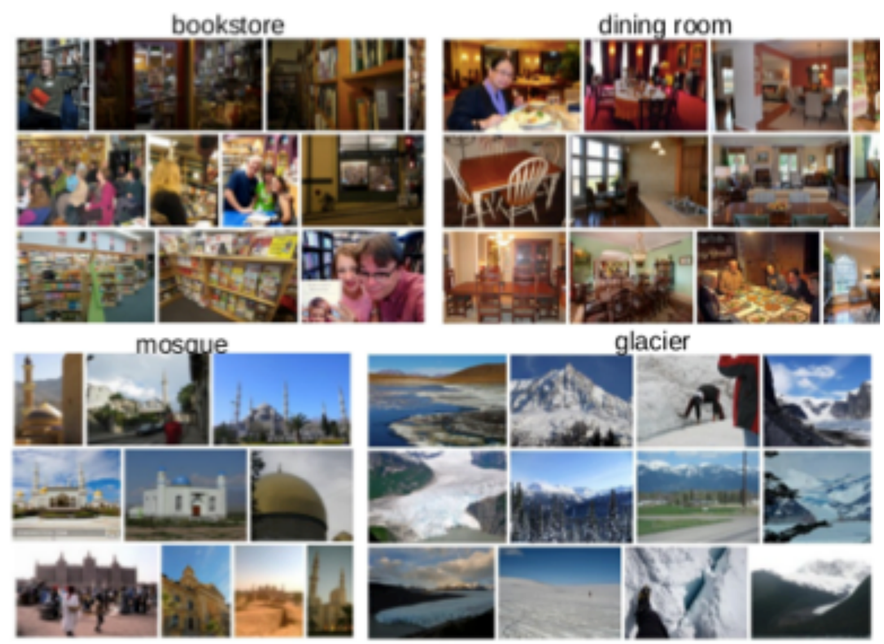
- S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, CVP 2006.
- C. Liu, J. Yuen, A. Torralba. Dense scene alignment using SIFT Flow for object recognition. CVPR 2009.

The function LMdenseSift.m computes a SIFT descriptor at each pixel location (in this implementation there is no ROI detection as in the original definition by D. Lowe). This function is a modification of the code provided by S. Lazebnik. The current implementation uses convolutions. Here there is an example of how to compute the dense SIFT descriptors for an image and to visualize the descriptors as described in Liu et al 09.

```
% demo SIFT using LabelMe toolbox

img = imread('demo1.jpg');
img = imresize(img, .5, 'bilinear');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SIFT parameters:
SIFTparam.grid_spacing = 1; % distance between grid centers
SIFTparam.patch_size = 16; % size of patch from which to compute SIFT descriptor (it has to be a factor of 4)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% CONSTANTS (you can not change this)
w = SIFTparam.patch_size/2; % boundary

% COMPUTE SIFT: the output is a matrix [nrows x ncols x 128]
SIFT = LMdenseSift(img, '', SIFTparam);

figure
subplot(121)
imshow(img(w:end-w+1,w:end-w+1,:))
title('cropped image')
subplot(122)
showColorSIFT(SIFT)
title('SIFT color coded')
```

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

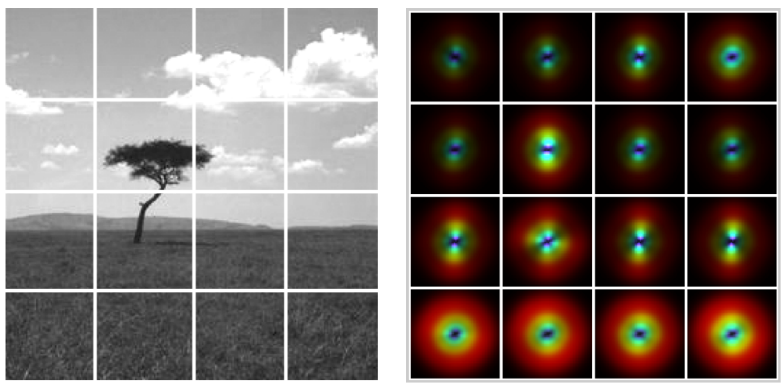
– Image alignment and building panoramas
– 3D reconstruction
– Motion tracking
– Object and scene recognition
– Indexing and database retrieval
– Robot navigation
– … other



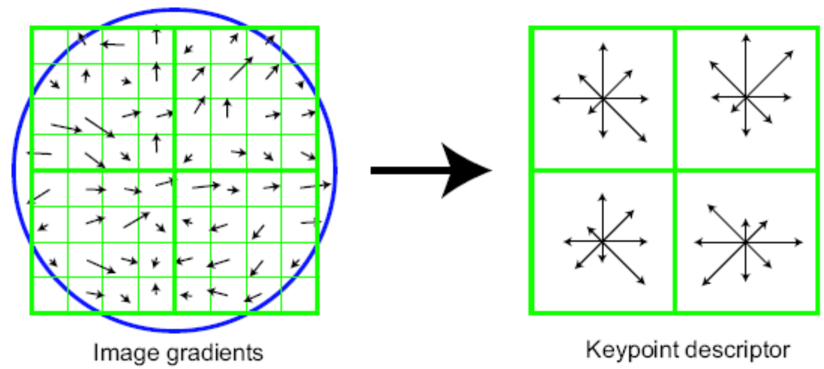

a) Camera translation
Original images
Aligned and composite image
Masks and seams between images
Final sequence
b) Camera rotation
Original images
Final sequence

Finding similar instances

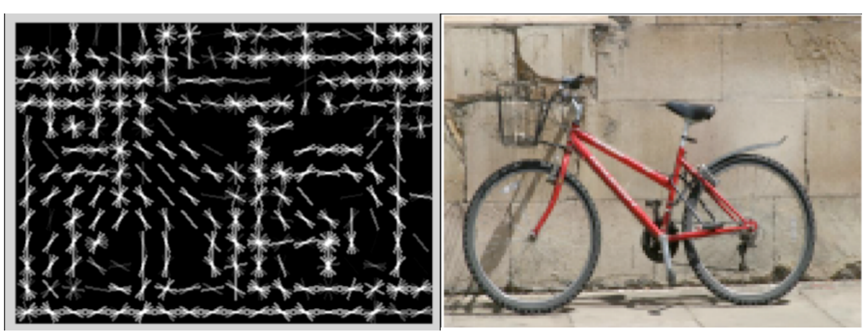
Gist: Grid of gabors
(Oliva & Torralba, 2001)



SIFT: Scale-Invariant Feature Transform
(Lowe, 1999)



HOG: Histograms of oriented gradients
(Dalal & Triggs CVPR 05)



DPM: Deformable Part Models
(Felzenszwalb, McAllester, Ramanan, 2008)