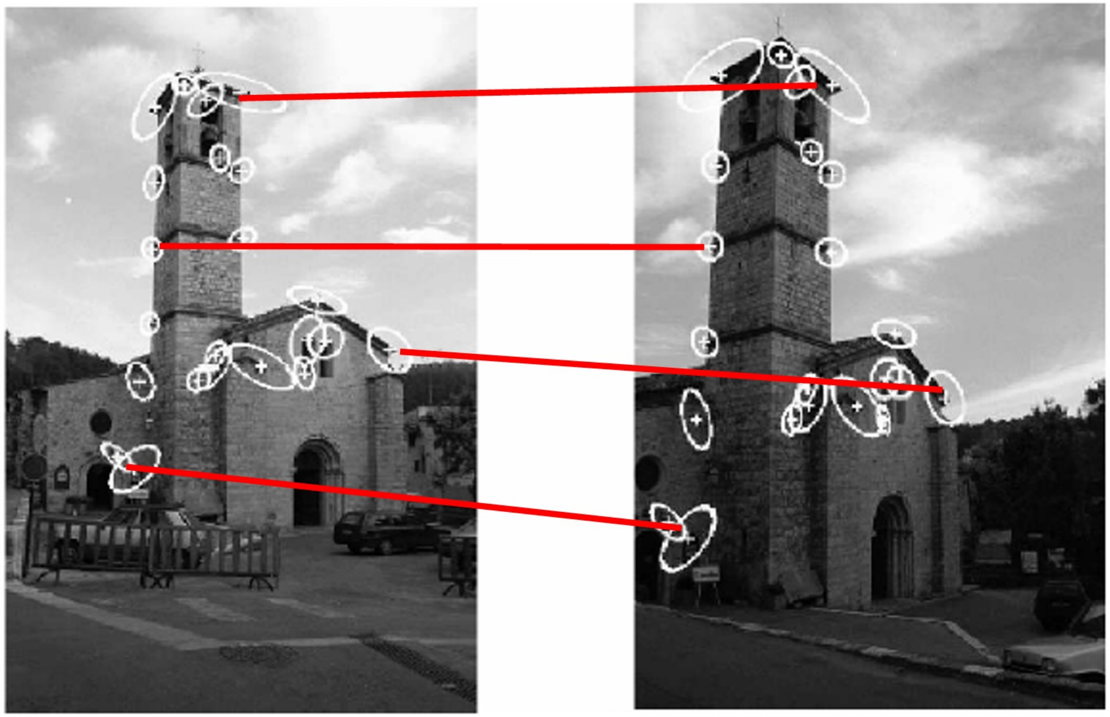
CSE 185 Introduction to Computer Vision

Lecture 5: Interest Points

Slides credit: Yuri Boykov, Ming-Hsuan Yang, Robert Collins, Richard Szeliski, Steve Seitz, Alyosha Efros, Fei-Fei Li, etc.

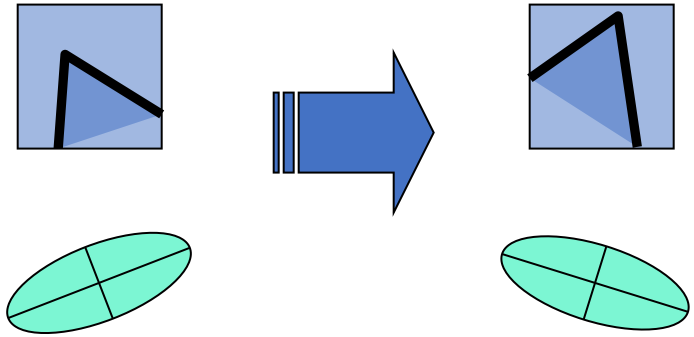
Motivation for feature points

Many applications require
generic "discriminant" feature points with
<u>identifiable appearance and location</u>
(so that they can be matched across multiple images)



Harris detector: some properties

- Rotation invariance

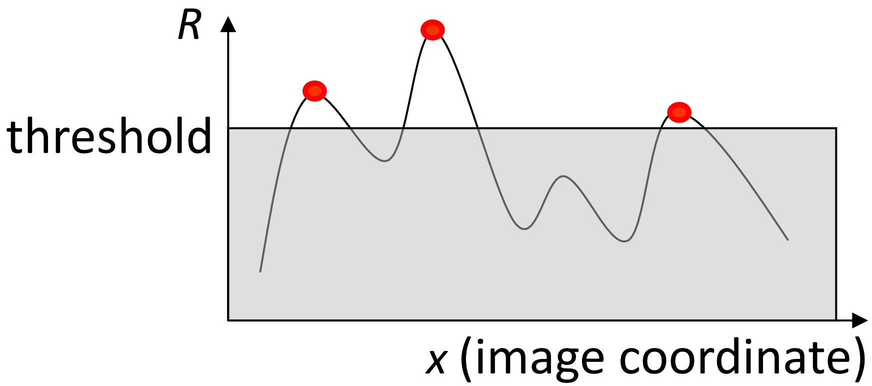


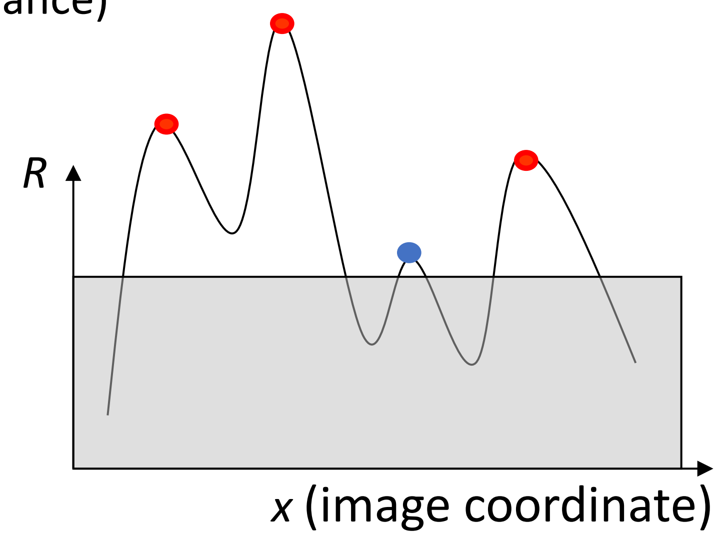
Ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner response R is invariant to image rotation

Harris detector: some properties

- Partial invariance to affine intensity change
  - ✓ Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$ ("**bias**" invariance)
  - ✓ Intensity scale: $I \rightarrow a\, I$ ("**gain**" invariance)

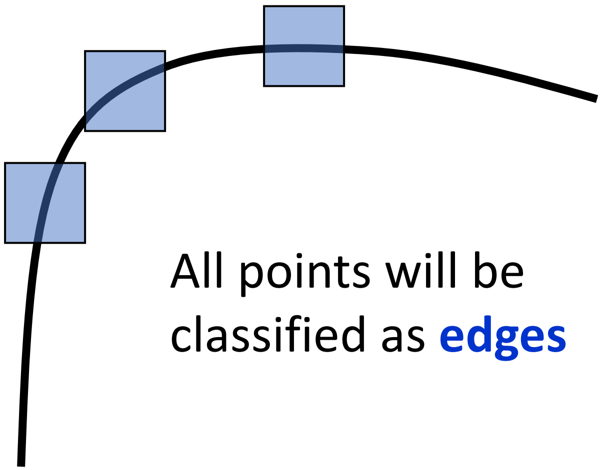


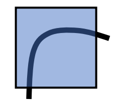


features locations stay the same,
but some may appear or disappear depending on gain a

Harris detector: some properties

- ❑ Not invariant to image scale

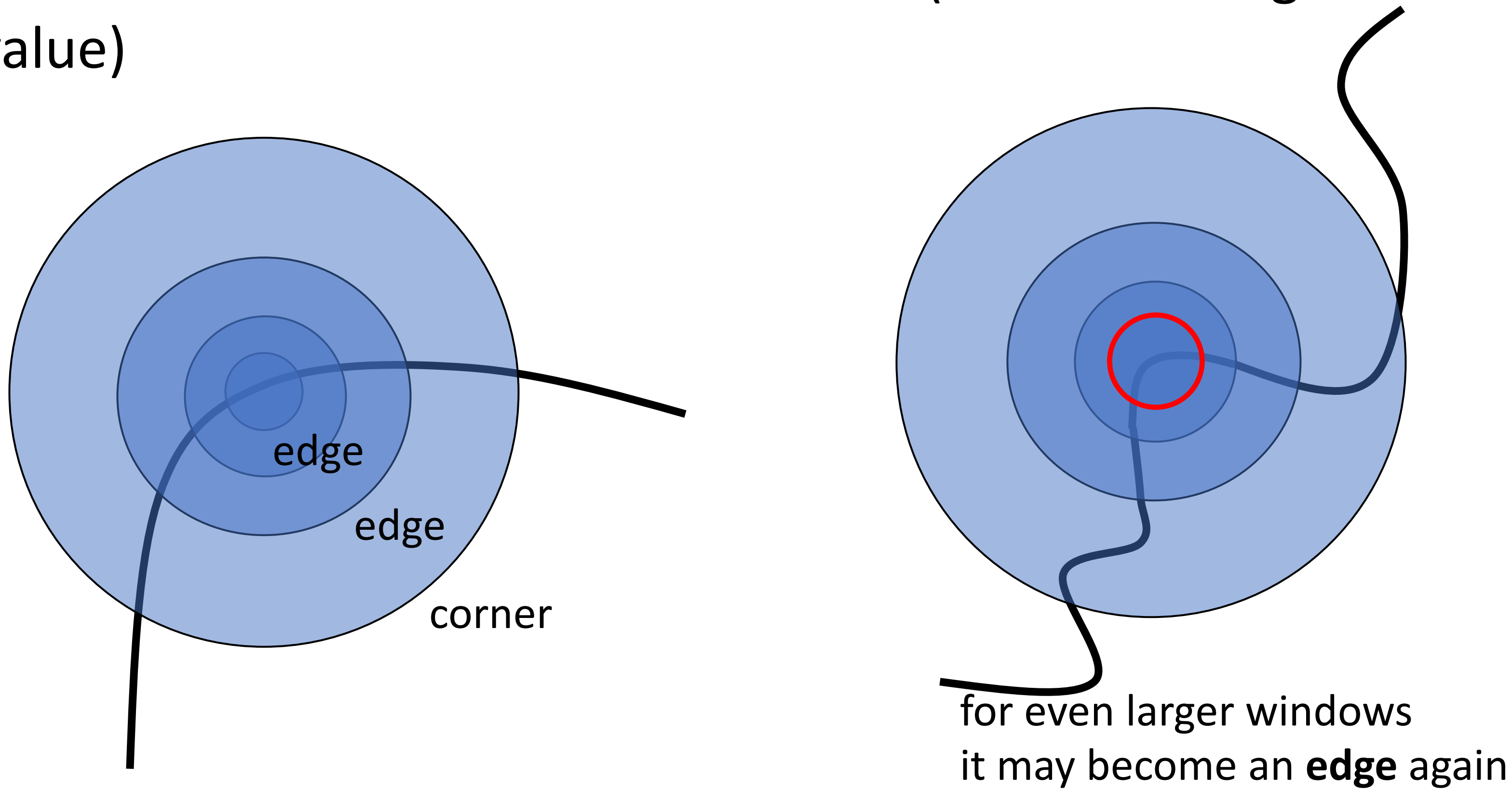




Two images of the same object taken at different scales (e.g. zoom settings)

Scale invariant detection

- Consider windows (circles) of different sizes around a point
- At some scale it looks like a corner.
- Choose the scale of the "best" corner (scale with largest R value)

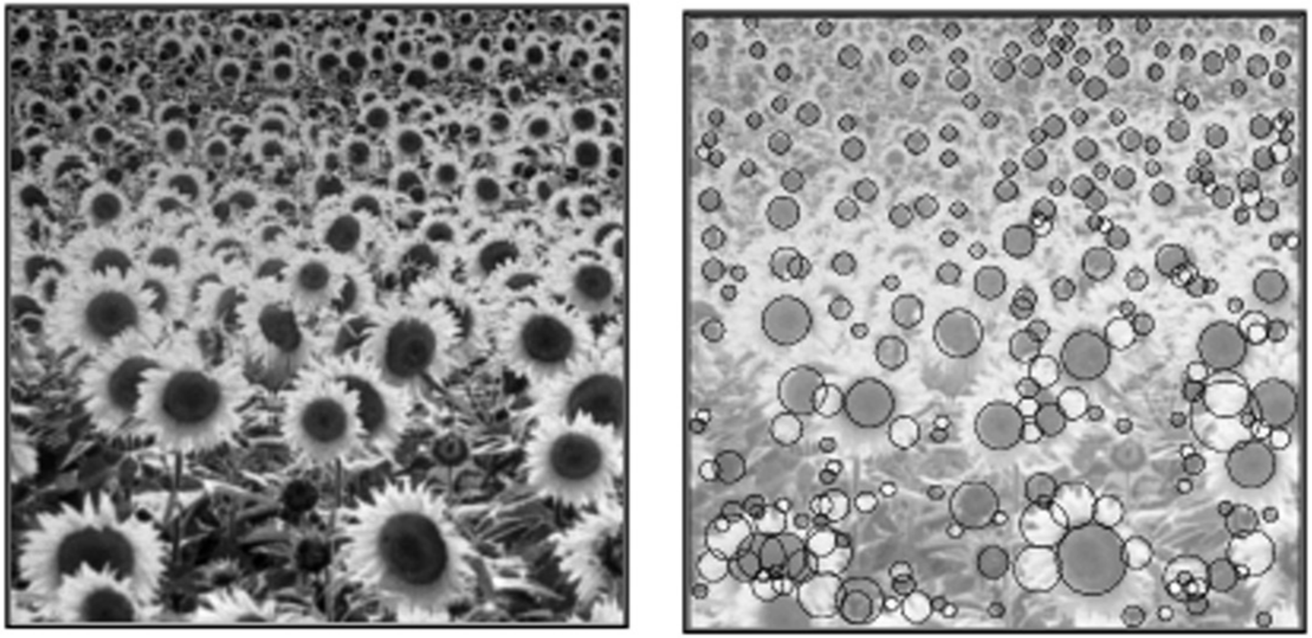


Laplacian of Gaussian (LoG) filter

- First, smooth image (Gaussian filtering)
- Second, find zero-crossings (Laplacian operator)

$$\underbrace{\nabla^2(f(x,y)\otimes G(x,y))}_{\text{Laplacian of Gaussian-filtered image}} = \underbrace{\nabla^2 G(x,y)\otimes f(x,y)}_{\text{Laplacian of Gaussian (LoG)-filtered image}}$$

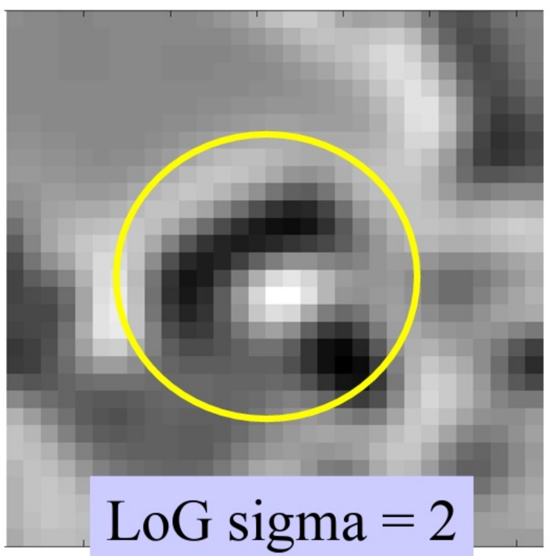
Other use of LoG: Blob detection

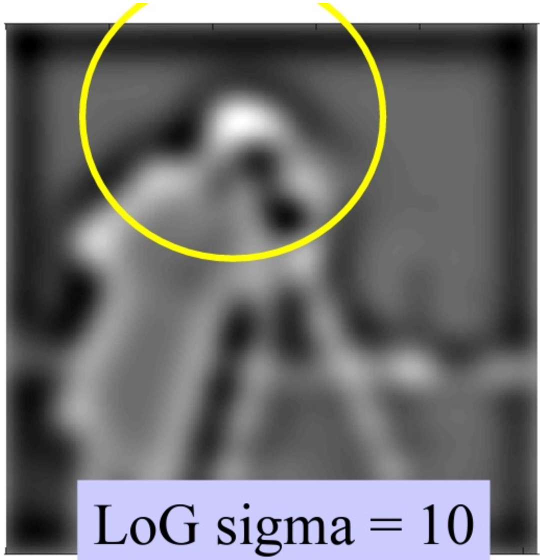


Lindeberg: ``Feature detection with automatic scale selection''. International Journal of Computer Vision, vol 30, number 2, pp. 77--116, 1998.

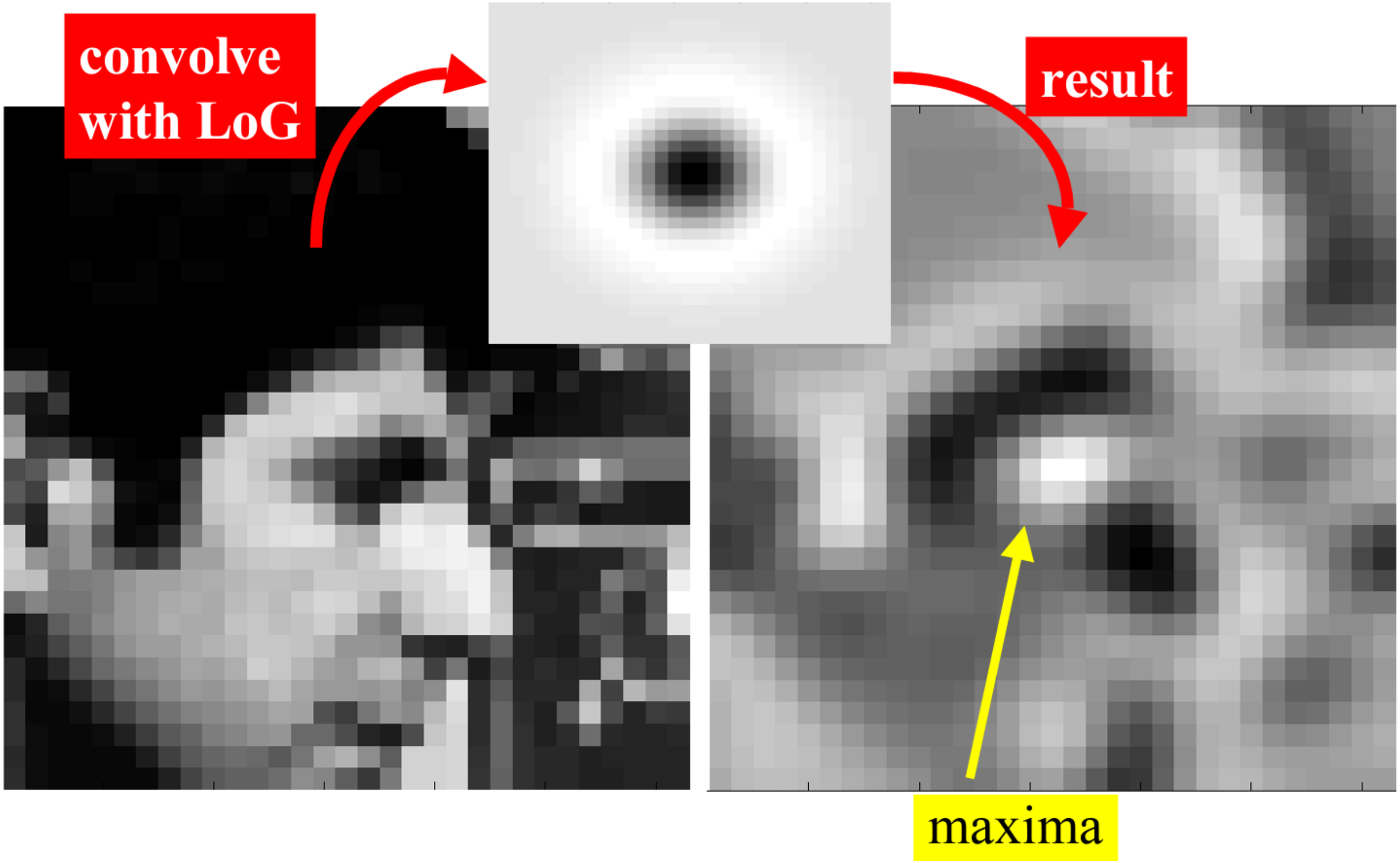
LoG blob finding

- LoG filter extrema locates “blobs”
  - maxima = dark blobs on light background
  - minima = light blobs on dark background
- Scale of blob (size ; radius in pixels) is determined by the sigma parameter of the LoG filter.



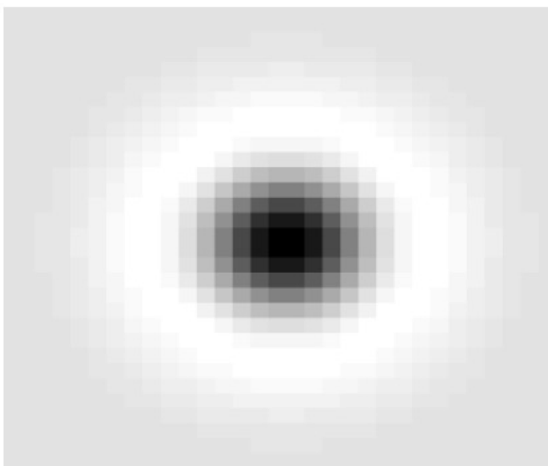


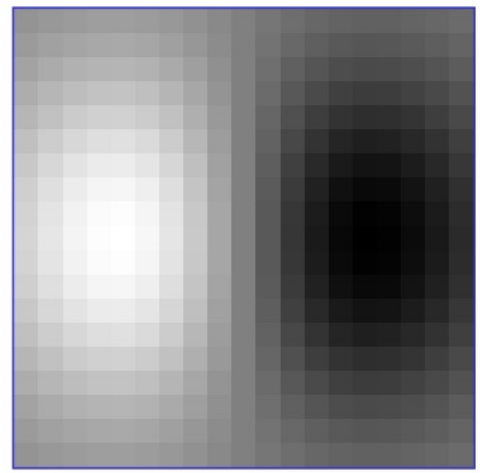
LoG blob finding



LoG blob finding

- Key idea: Cross correlation with a filter can be viewed as comparing a little "picture" of what you want to find against all local regions in the image.





Maximum response:
 dark blob on light background
Minimum response:
 light blob on dark background

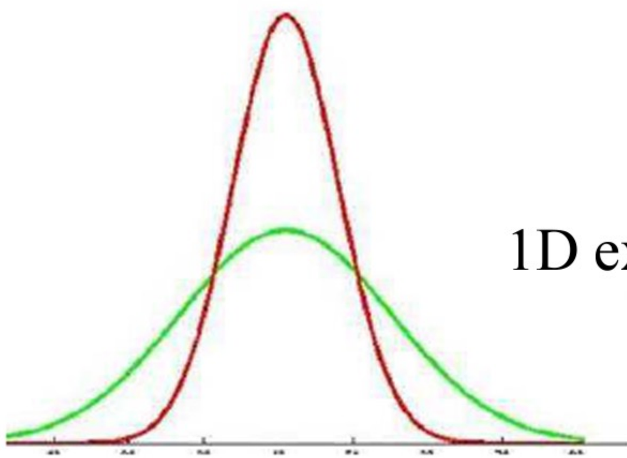
Maximum response:
 vertical edge; lighter on left
Minimum response:
 vertical edge; lighter on right

LoG and DoG

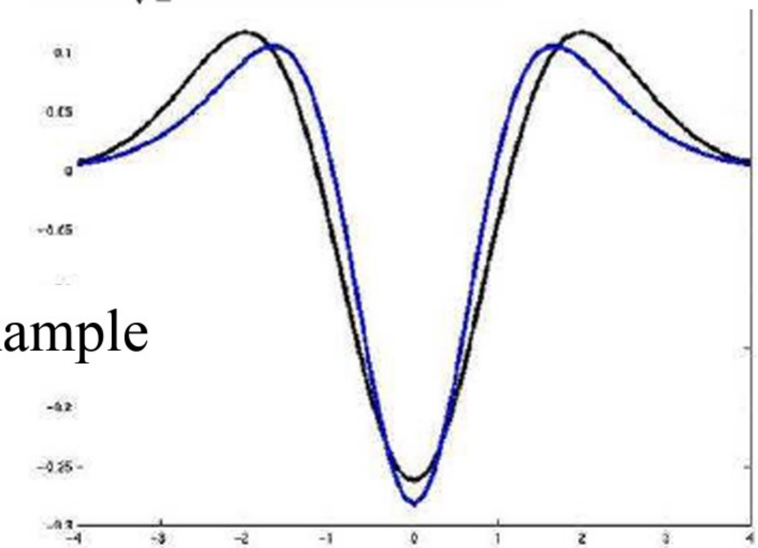
- ❑ LoG can be approximated by difference of Gaussian

$$\nabla^2 G_\sigma \approx G_{\sigma_1} - G_{\sigma_2}$$

Best approximation when:
$\sigma_1 = \frac{\sigma}{\sqrt{2}}, \ \sigma_2 = \sqrt{2}\sigma$

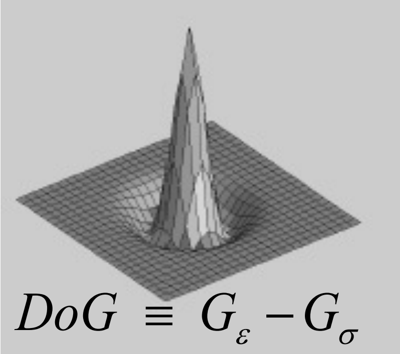


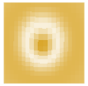
1D example

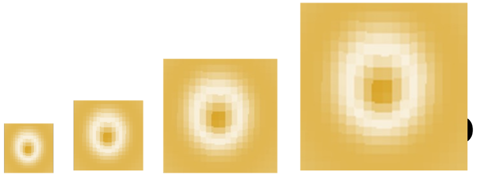


Blob-like discriminant feature points

- *DoG* (or a similar *LoG*) kernels are used to detect blob-like features



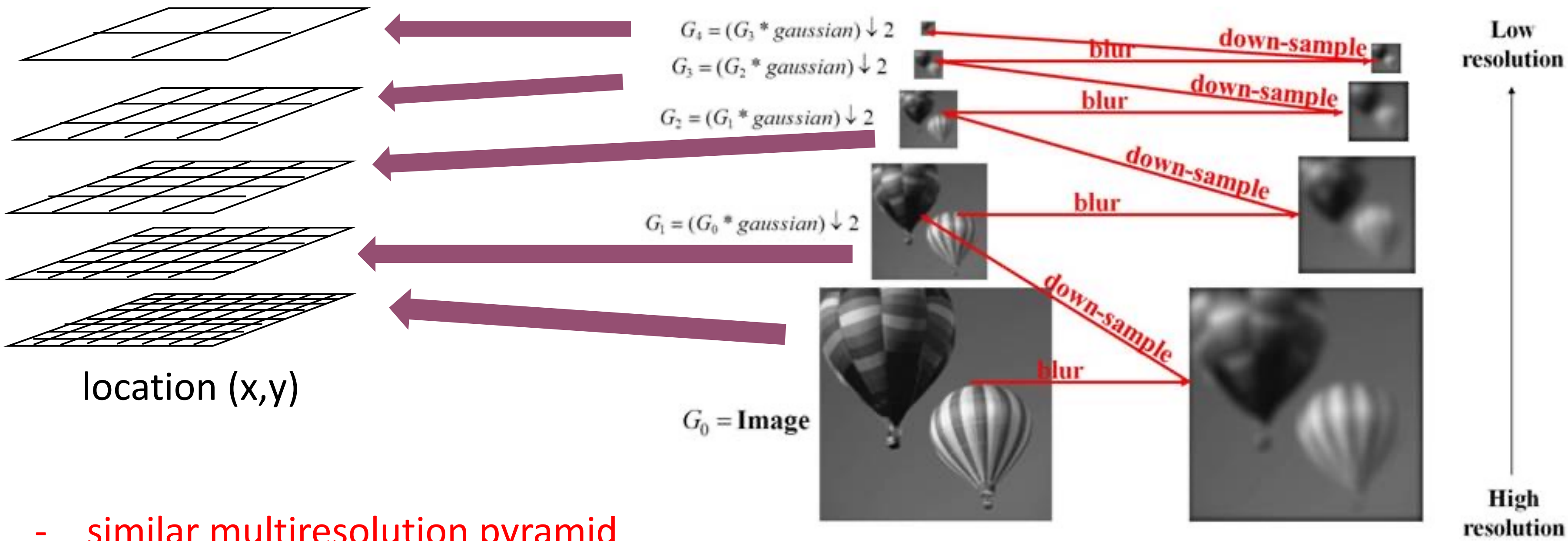
Feature locations: extrema points for convolution with 

Feature scale is still not known: 

How to find the **right scale**?

Gaussian pyramid

- *Gaussian pyramid* helps to find "optimal scale" for features

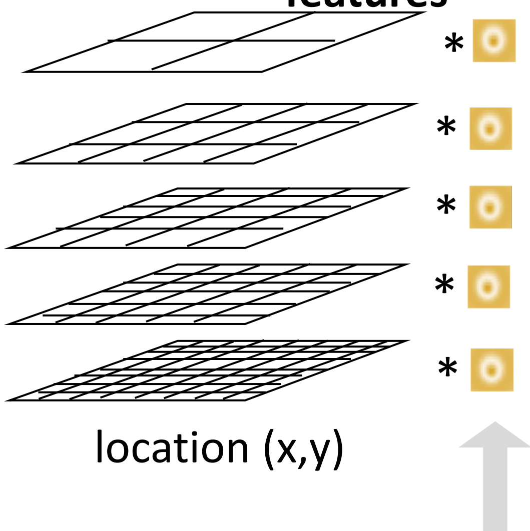


- similar multiresolution pyramid also appears in the "*encoder*" part of common CNNs

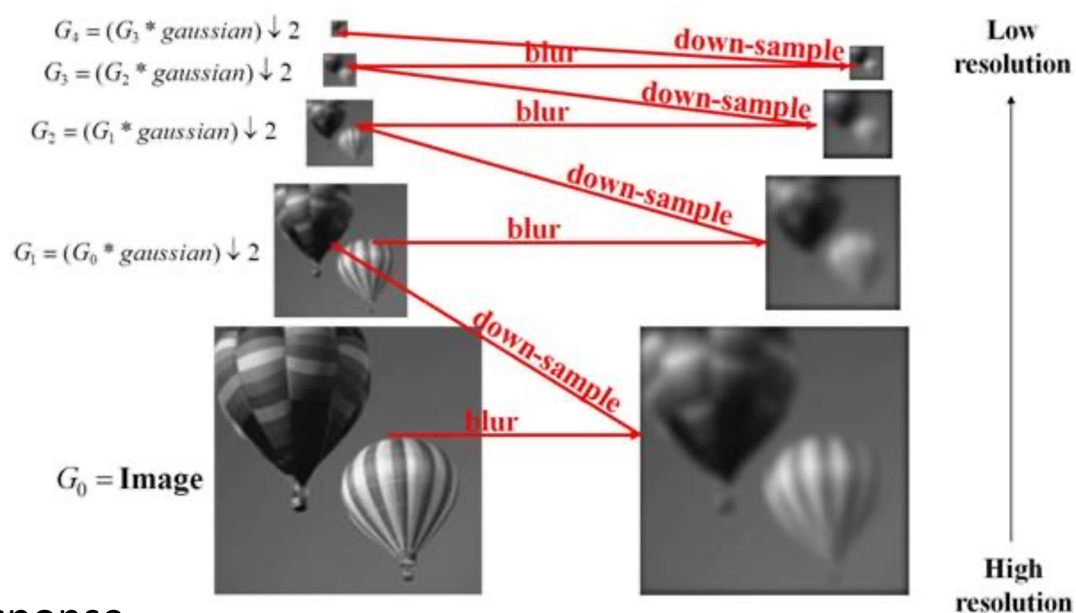
Gaussian pyramid

- *Gaussian pyramid* helps to find "optimal scale" for features

e.g. consider **DoG features**



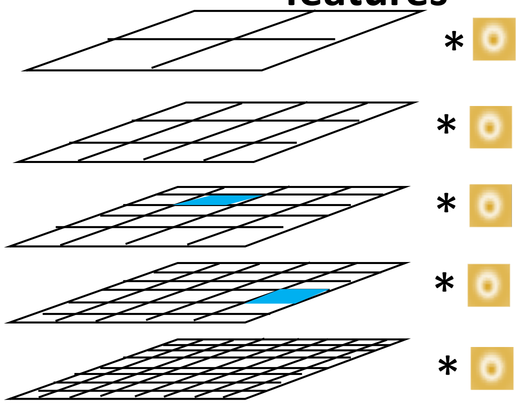
compute feature response
(*e.g.* convolve w. kernel)
with image **at each scale**



Gaussian pyramid

- *Gaussian pyramid* helps to find "optimal scale" for features

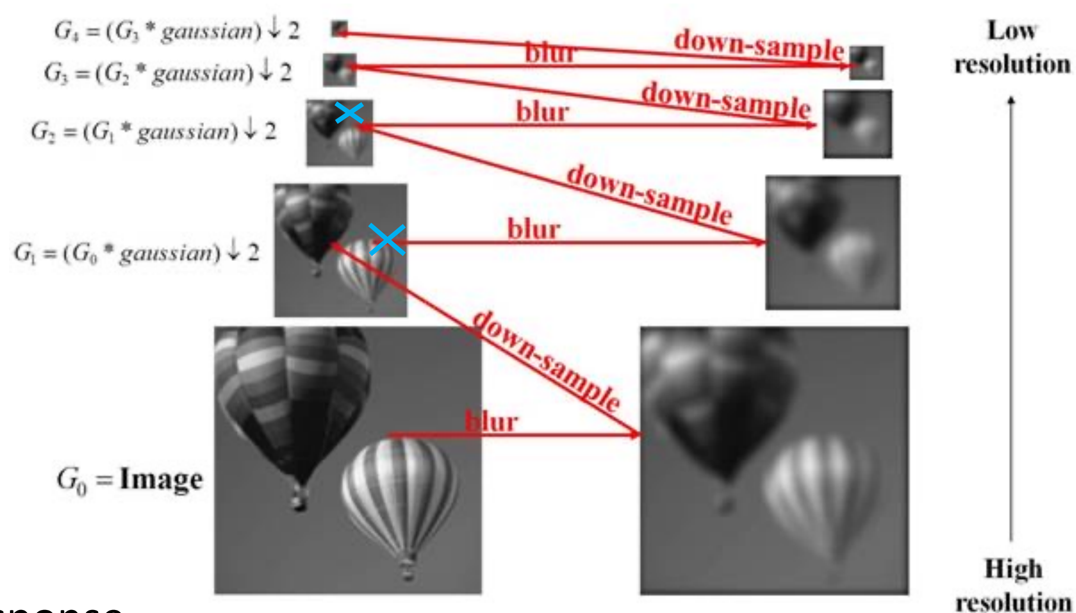
e.g. consider **DoG features**



location (x,y)

find local maxima response in volume (x,y,s)

compute feature response (*e.g.* convolve w. kernel) with image **at each scale**



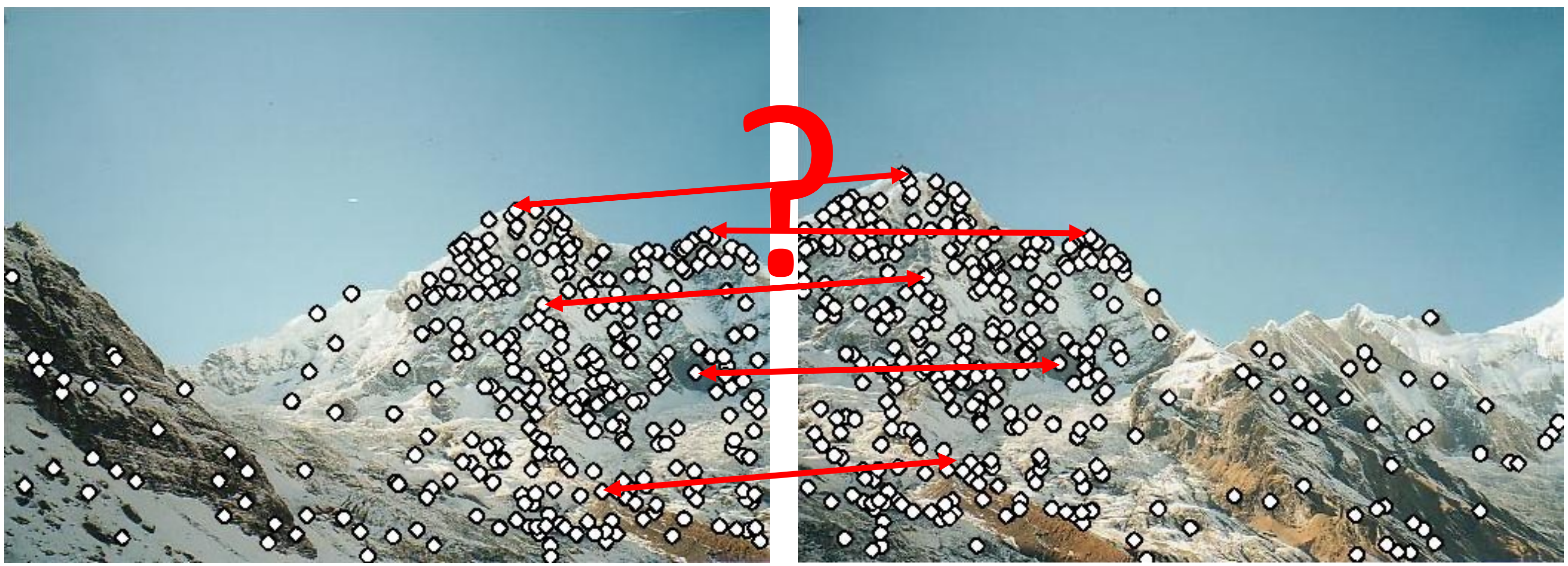
Example

Dog

circle center -> feature location
circle radius -> feature scale

Features: location + descriptor

- Now we know how to detect (locate) interest points or features
- Next question: **How to match them?**



Besides location each feature point should have its signature or **descriptor**

Point descriptor should be: **invariant** (stable to illumination and view point changes)

distinctive (discriminant)

Common generic feature points

- MOPS, Hog, SIFT, …

Features are characterized by	**location** and	**descriptor**
color	any pixel	RGB vector
edge	local extrema of $\|\nabla f\|$	∇f
MOPS	corners	normalized intensity patch
HOG SIFT	DOG or LOG extrema points or other interest points	gradient orientation histograms

more below

highly discriminative

(see Szeliski, Sec. 4.1.2)

Multi-scale oriented patches (MOPS)

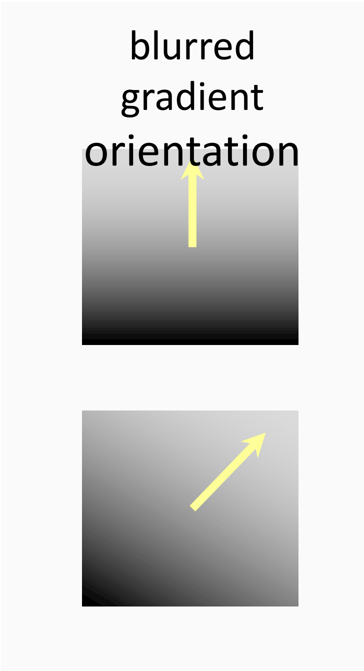
Summary of main ideas:

- Patch location and orientation
  - Multi-scale Harris corners
  - Orientation from blurred gradient => invariant to rotation
- Descriptor vector
  - Sampling of intensities in a local 8x8 patch
  - Bias/gain normalization => invariance to affine intensity changes

[Brown, Szeliski, Winder, CVPR'2005]

MOPS: Patch location and orientation

- Location and Scale – Harris corner
- Orientation - blurred gradient
- **Rotation Invariant Frame**
  - Scale-space position (x, y, s) + orientation (θ)



Detection at multiple scales

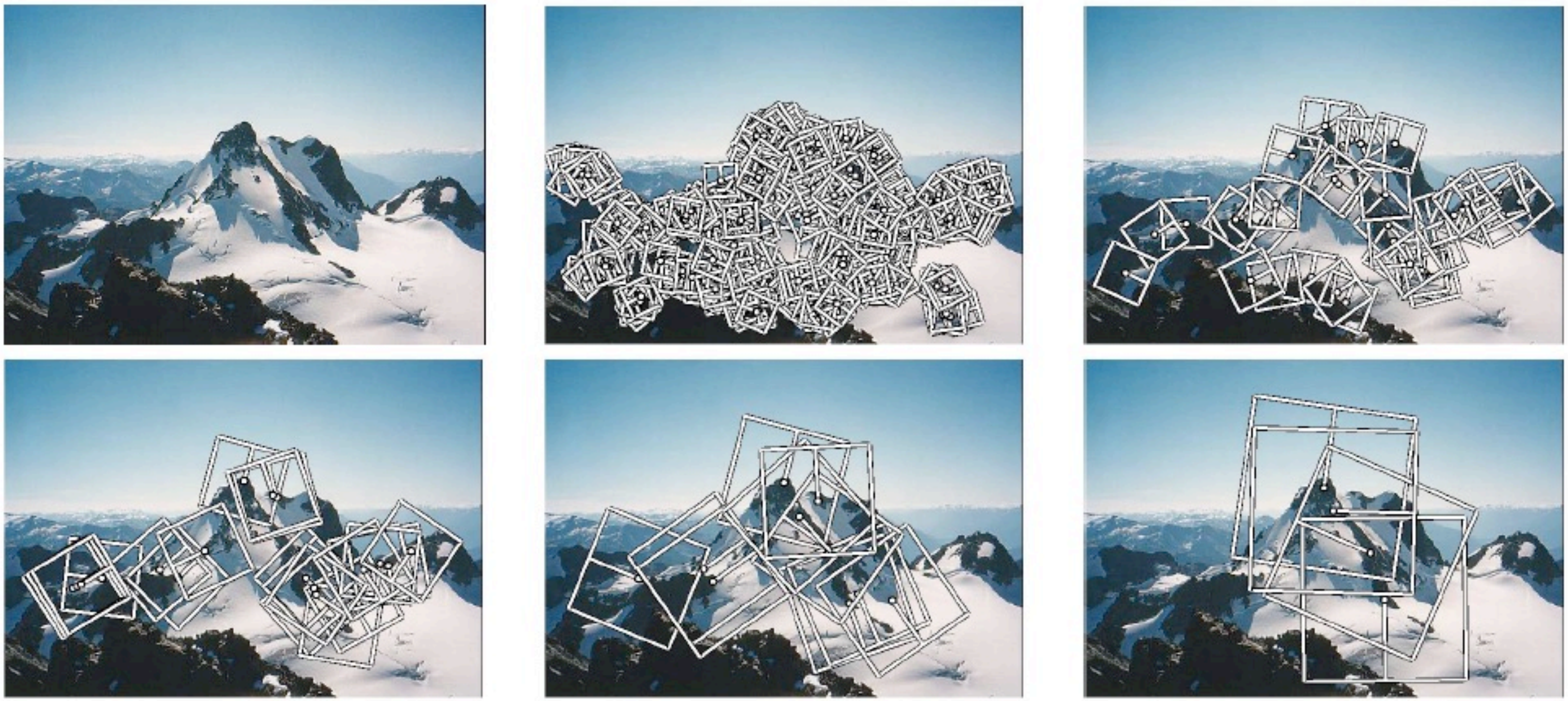
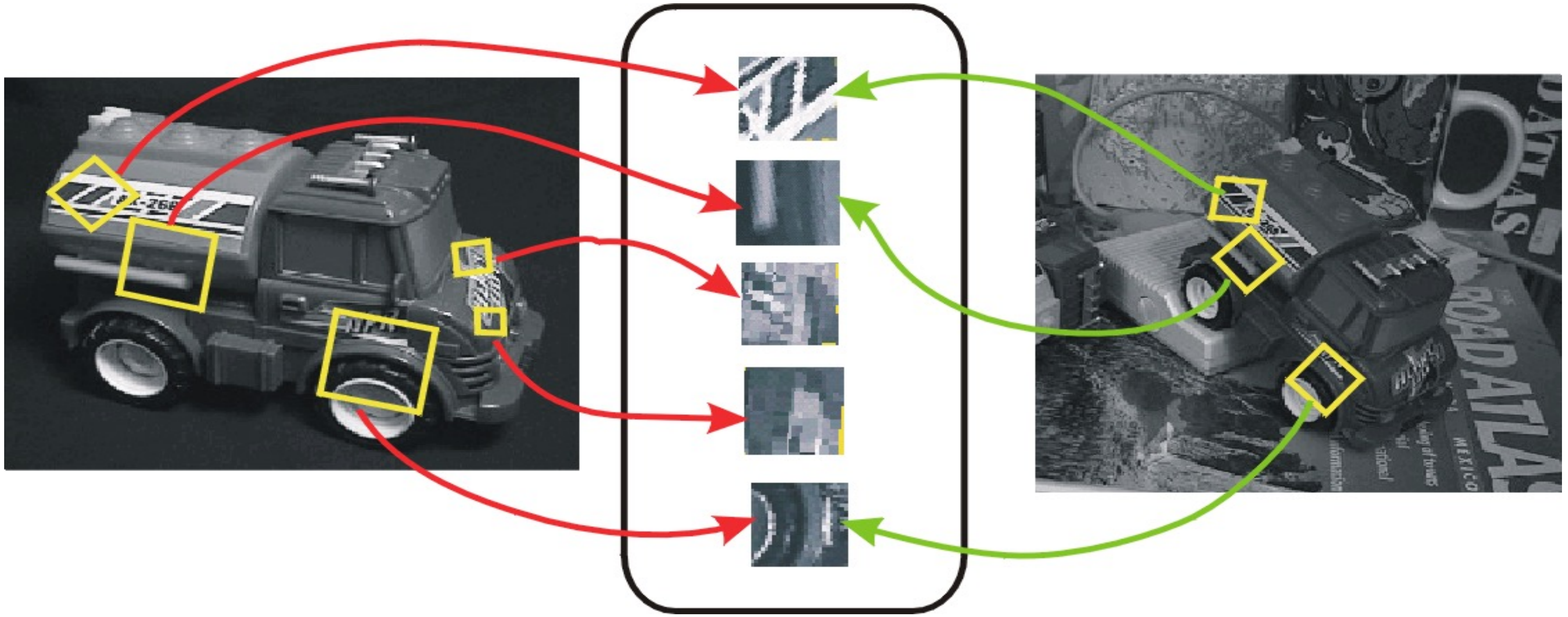


Figure 1. Multi-scale Oriented Patches (MOPS) extracted at five pyramid levels from one of the Matier images. The boxes show the feature orientation and the region from which the descriptor vector is sampled.

SIFT

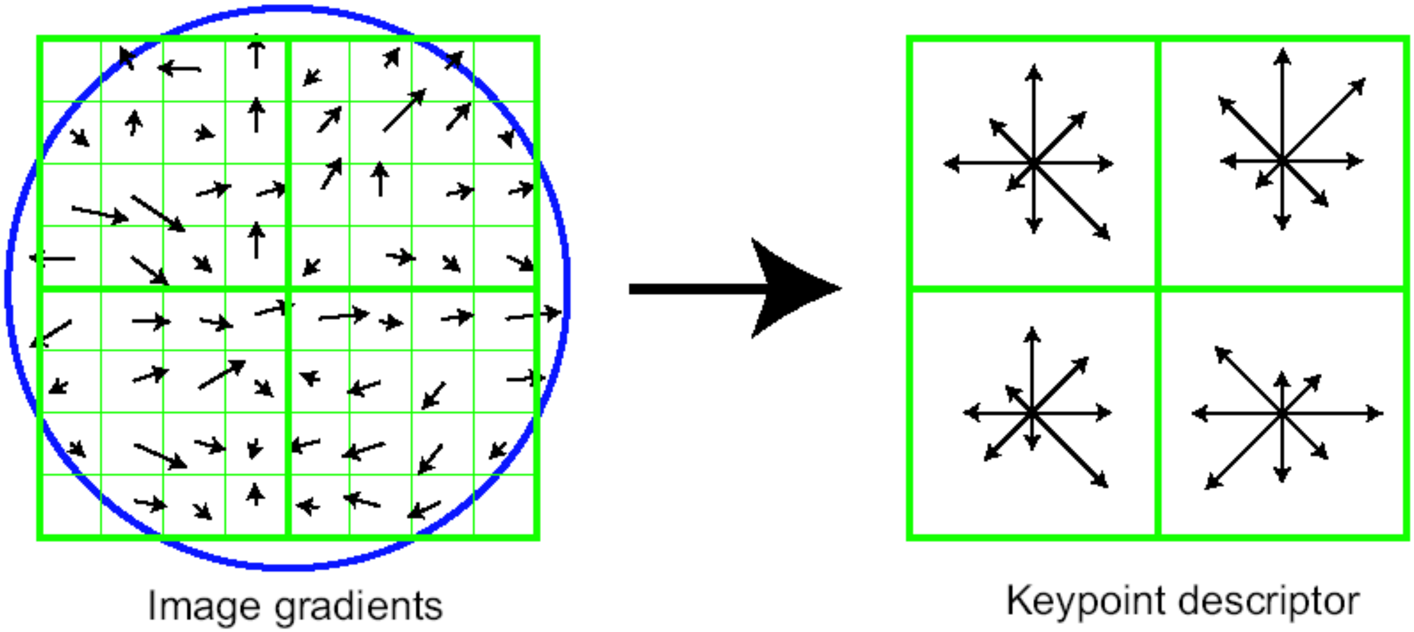
- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



SIFT Keypoint descriptor

- Each keypoint has
  - location
  - scale
  - orientation
- Next is to compute a descriptor for the local image region about each keypoint that is
  - highly distinctive
  - invariant as possible to variations such as changes in viewpoint and illumination

Lowe's keypoint descriptor
(shown with 2 X 2 descriptors over 8 X 8)



In experiments, 4x4 arrays of 8 bin histogram is used, a total of 128 features for one keypoint

Lowe’s keypoint descriptor

- ❑use the normalized region about the keypoint
- ❑compute gradient magnitude and orientation at each point in the region
- ❑weight them by a Gaussian window overlaid on the circle
- ❑create an orientation histogram over the 4 X 4 subregions of the window
- ❑4 X 4 descriptors over 16 X 16 sample array were used in practice. 4 X 4 times 8 directions gives a vector of 128 values.

Use for SIFT

- Feature points are used also for:
  - Image alignment (homography, fundamental matrix)
  - 3D reconstruction (e.g. Photo Tourism)
  - Motion tracking
  - Object recognition
  - Indexing and database retrieval
  - Robot navigation
  - … many others

[Photo Tourism: Snavely et al. SIGGRAPH 2006]