Image Features (Part III)

Goal: We introduce and motivate several types of image features. These are each designed for specific vision tasks.

We consider features to support the following tasks:

I. Matching image patches between images with significantly different viewpoints,

II. Extracting image landmarks; a) their (x, y) position,

III. Extracting image landmarks; b) their scale, and c) their orientation. $\Leftarrow$ **Today**

Readings: Szeliski, Section 4.1 and 4.2.

Part III: Scale and Orientation of Image Landmarks

Recall an **image landmark** is meant to help orient a vision system with respect to the image content.

As such, a landmark should be repeatably identifiable as a roughly corresponding image patch across a range of different views of the same object. Here the image patch specification must include image location, scale and orientation.

In the previous lecture we discussed one method for determining landmark **locations**, as spatially local maxima of the Harris operator. These are distinct and (somewhat) repeatably identifiable locations in an image.

The **scale** of an image landmark is its (rough) diameter in the image. It is denoted by σ, which is measured in pixels.

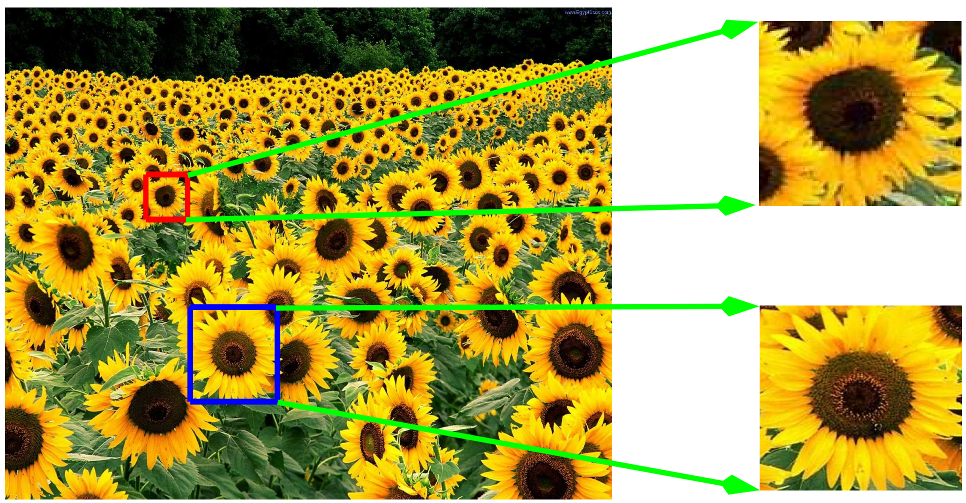
The **orientation** of an landmark specifies its rotation within the image plane.

We begin by considering image scale.

Matching Image Patches at Different Scales

We begin the story about extracting image scale at the end, namely by thinking about how we could match two image patches corresponding to successfully extracted landmarks.

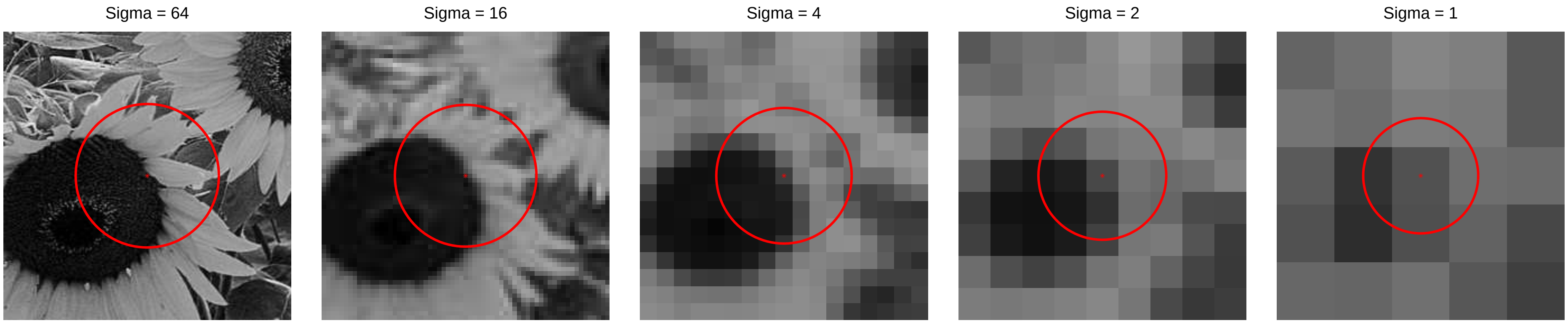
Consider the two landmarks indicated by the red and blue boxes below. Each landmark specifies an image patch location, orientation (i.e., these boxes are upright in the image) and scale (i.e., the size of the box).



How can these image patches be compared? One (inefficient) approach would be to resample both image patches to a single size (above right). Then these resized image patches could be compared using a technique that is tolerant of some misalignment, such as HoG. We would prefer to avoid this image resizing.

Image Scale and Pixel Resolution

We consider how image features can be directly compared across different scales.



The red circles above denote the same (i.e., definitely corresponding) image patch for different pixel resolutions. Note the resolution is extremely coarse for σ equal to 1 or 2 pixels.

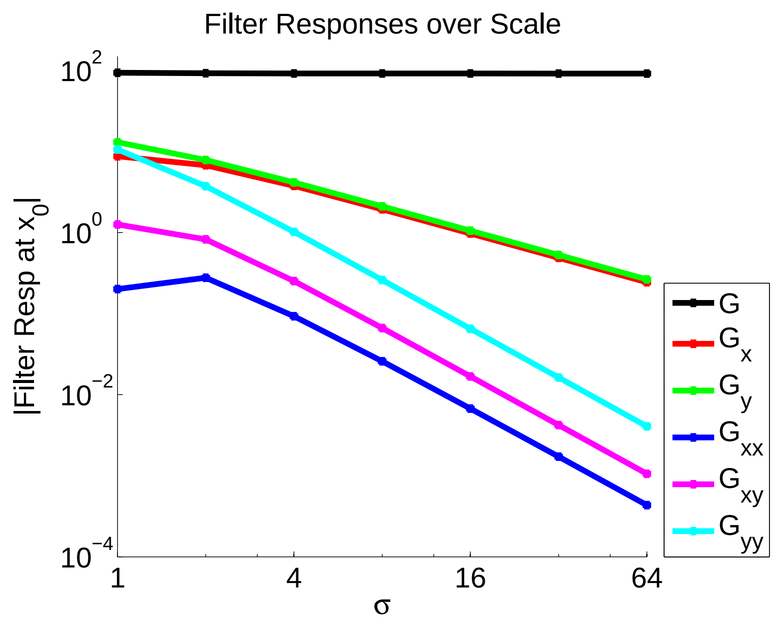
Using the clipped Gaussian kernel, $g(\vec{x}, \sigma)$, we compute the blurred image $r(\vec{x}, \sigma)$ along with its first and second derivatives in $\vec{x}$, and sample the result at the center pixel $\vec{x}_0$ (marked with the red asterisk above). This forms $r(\vec{x}_0, \sigma)$, which is a function of σ.

Boundary effects are avoided by doing all the filtering in a much larger image.

The filter responses $r(\vec{x}_0, \sigma)$ for $\sigma \in [1, 64]$ are shown on the next slide.

Image Derivatives at Different Resolutions

Below we show a log-log plot of the absolute values of $r(\vec{x}, \sigma)$ along with its first and second derivatives (all sampled at $\vec{x} = \vec{x}_0$) versus the scale parameter σ.



This plot exhibits clear linear relationships between the log-absolute responses and $\log(\sigma)$.

As we show next, the slope of these lines depend on the order of the derivative of r that is being plotted (i.e., a derivative of order $p = 0, 1$, or 2 produces a line with slope $-p$).

The Intuition of Scale-Normalized Derivatives

Suppose the region of interest is a disk of radius σ centered on the current pixel.

Note the derivative $r_x(\vec{x}_0, \sigma) = [\partial r / \partial x](\vec{x}_0, \sigma)$, equals the rate of increase in the graylevel r, *per pixel step in* x. It follows that these derivatives are inversely proportional to the number of pixels across the region of interest.

A scale-invariant choice for the spatial coordinates is to scale them by the radius of the region of interest, i.e., $(u, v) = (x/\sigma, y/\sigma)$.

Then $\partial r / \partial u = (dx/du)(\partial r / \partial x) = \sigma r_x$, which provides the rate of increase in the graylevel r, *per step of size* σ. Similarly, $\partial r / \partial v = \sigma r_y$.

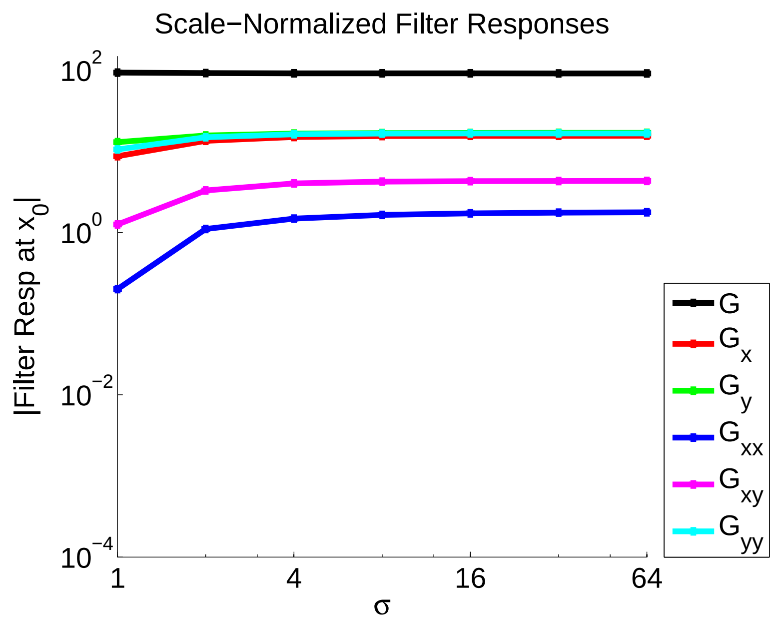
The second derivatives r_{uu}, r_{uv} and r_{vv} are equal to $\sigma^2 r_{xx}$, $\sigma^2 r_{xy}$ and $\sigma^2 r_{yy}$, respectively.

This scaling of the first and second derviatives by σ and σ^2, respectively, provides *scale-normalized derivatives*.

More generally, we scale-normalize a p^{th} order image derivative (expressed in terms of x and y) by multiplying it by σ^p.

Scale-Normalized Image Derivatives

For $\sigma \geq 4$ (pixels) these *scale-normalized derivatives* are nearly invariant of the pixel resolution. Discretization effects cause the results to deviate from the ideal for small σ (cf., p.4).



In practice, we typically downsample images and only use σ in the range of about 2 to 4 pixels. We include much larger values of σ here to illustrate the point that *scale-normalized derivative filters are insensitive to pixel resolution*, except at coarse spatial resolutions.

Even for $\sigma \in [2, 4]$, the first-order (un-normalized) image derivatives vary by a factor of 2, and this factor increases to 4 for second-order derivatives.

Scale-Normalized Image Scale

The only parameter that we have not scale-normalized is σ itself (which is still measured in pixels).

What could it mean to scale-normalize the scale parameter itself?

Consider changing σ by 5 pixels, i.e., $\sigma \to \sigma + 5$:

- If the original σ was 5, then this increment doubles σ to 10.
- If the original σ was 100, then this increment increases σ by 5%.

This motivates choosing a "scale-invariant scale parameter" such that the patch radius σ is increased by a *constant percentage* for any unit step in this parameter.

This can be arranged by defining $\sigma = 2^s$, where s is the new (scale-normalized) scale parameter.

Note that a unit step from s to $s + 1$ corresponds to doubling σ, no matter what magnitude σ has in pixels. This is exactly as desired.

Estimating Canonical Image Scales

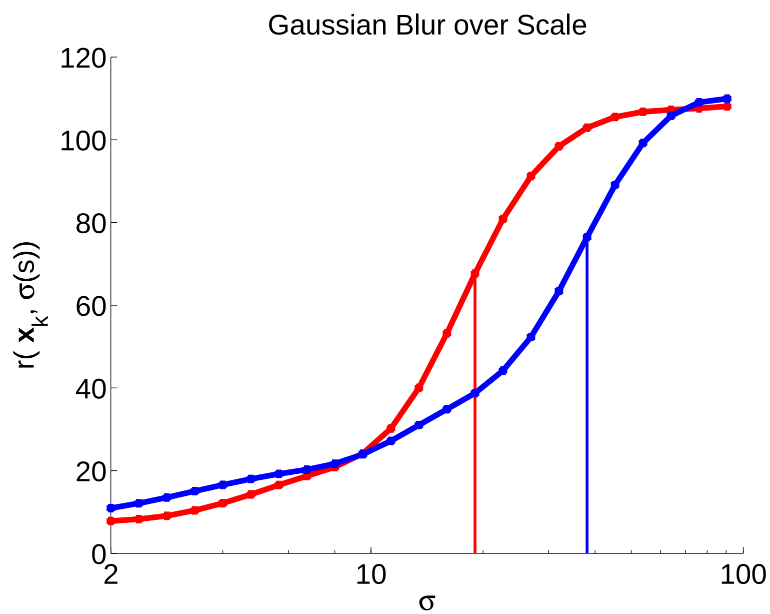
Given an image point $\vec{x}_0$, we wish to use the Gaussian blurred response $r(\vec{x}_0, \sigma(s))$ to estimate the canonical scale(s) for patches centered at $\vec{x}_0$.

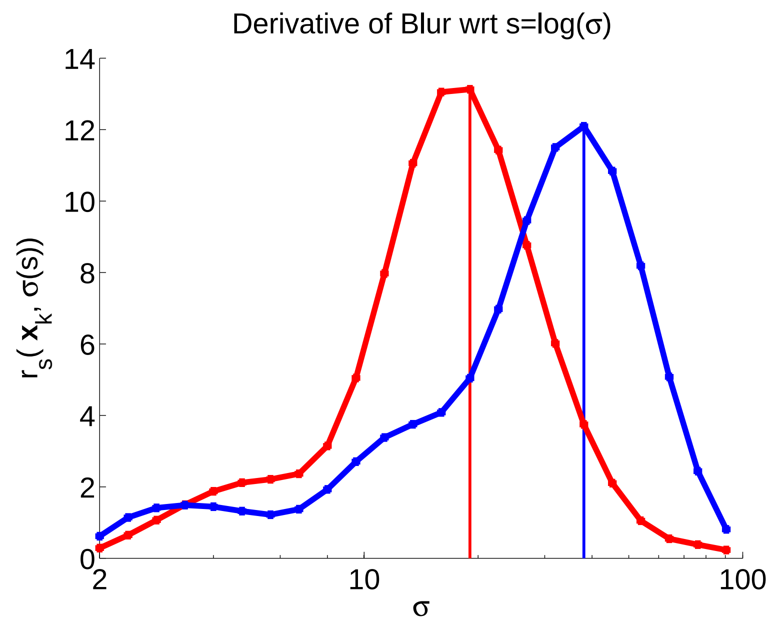
For the following demonstration we have chosen the red and blue points shown above, in the center two sunflowers, as two possible choices for $\vec{x}_0$.

For each $\vec{x}_0$, we consider the blurred response $r(\vec{x}_0, \sigma(s))$ as a function of the scale parameter s.

Estimating Canonical Image Scales

The Gaussian blurred responses $r(\vec{x}_0, \sigma(s))$ at the red and the blue points are shown below left:





The derivative of the blurred response $r(\vec{x}_0, \sigma(s))$ with respect to s is shown on the right above. Strong positive maxima (or negative minima) in this derivative indicate significant scales for patches centered at $\vec{x}_0$. (The derivative responses above could also be interpolated to find the peaks more accurately.)

The scales that have been identified are shown with the vertical lines above, and correspond to radii $\sigma(s) = 19$ and 23 pixels for the red and blue circles shown on p.9. (Note a bias in the estimated sizes.)

Scale-Normalized Laplacian of Gaussian

We argued above that strong peaks/pits in the derivative $dr/ds(\vec{x}_0, \sigma(s))$ can be used for scale selection. Moreover, on the following page we show

$$\frac{dr}{ds}(\vec{x}, \sigma(s)) \;=\; \log(2)\, \sigma^2 \; [(\triangle g) * I]\, (\vec{x}, \sigma), \tag{1}$$

where $\sigma(s) = 2^s$ and

- $\triangle = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ denotes the Laplacian differential operator,
- $\triangle g(\vec{x}, \sigma)$ is called the Laplacian of a Gaussian (LoG),
- $\sigma^2 \triangle$ denotes the scale-normalized Laplacian (see pp.6,7),
- $\sigma^2 \triangle g(\vec{x}, \sigma)$ is the normalized LoG (nrmLoG), which can be used for scale selection (see eqn (1).

In view of equation (1), it is equivalent to look for strong peaks/pits in the image convolved with the scale-normalized LoG filter, $\sigma^2 \triangle g(\vec{x}, \sigma)$.

LoG, DoG, and the Gaussian Derivative Filter in Scale

For the image $I(\vec{x})$, define $\sigma(s) = 2^s$ and $r(\vec{x}, \sigma) = (g * I)(\vec{x})$, where $g(\vec{x}, \sigma)$ is the 2D Gaussian kernel

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

(For fine resolutions, the normalization by $1/(2\pi\sigma^2)$ in equation (2) approximates the discrete sum we used previously to normalize the clipped Gaussian kernel.) It then follows that:

$$\begin{aligned} \frac{dr}{ds}(\vec{x}_0, \sigma(s)) = \left[\frac{\partial g}{\partial s} * I\right](\vec{x}_0) &= \frac{d\sigma}{ds}(s)\left[\frac{\partial g}{\partial \sigma} * I\right](\vec{x}_0) \\ &= \log(2)\sigma(s)\left[\frac{\partial g}{\partial \sigma} * I\right](\vec{x}_0). \end{aligned} \tag{3}$$

By (2), the (normalized) derivative of the Gaussian filter $\sigma \partial g/\partial\sigma$ is given by

$$\begin{aligned} \sigma\frac{\partial g}{\partial \sigma}(\vec{x}, \sigma) &= \left[\frac{(x^2+y^2)}{\sigma^2} - 2\right] g(\vec{x}, \sigma) \\ &= \sigma^2 \left[\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}\right] \equiv \sigma^2 \bigtriangleup g(\vec{x}, \sigma). \end{aligned} \tag{4}$$

Here $\bigtriangleup = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ denotes the Laplacian differential operator. Together equations (3) and (4) imply (1) as desired.

A difference of Gaussian kernel (in 2D) is defined to be

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

where $\rho > 1$. For ρ close to 1, it follows that

$$\begin{aligned} \frac{-1}{(\rho-1)} DoG(\vec{x}, \sigma, \rho) &= \frac{g(\vec{x}, \sigma) - g(\vec{x}, \rho\sigma)}{\sigma - \rho\sigma}\sigma \\ &\approx \sigma\left[\frac{\partial g}{\partial \sigma}(\vec{x}, \sigma)\right] = \sigma^2 \bigtriangleup g(\vec{x}, \sigma). \end{aligned}$$

Thus a DoG filter can be used to approximate the normalized LoG.

A DoG approximation is convenient because it can be implemented using two separable (i.e., Gaussian) filter kernels.

Local Maxima of Normalized LoG

Suppose we successively increase σ by a factor of ρ (e.g., $\rho = 2^{1/k}$, where k equals the number of samples desired per octave in scale).

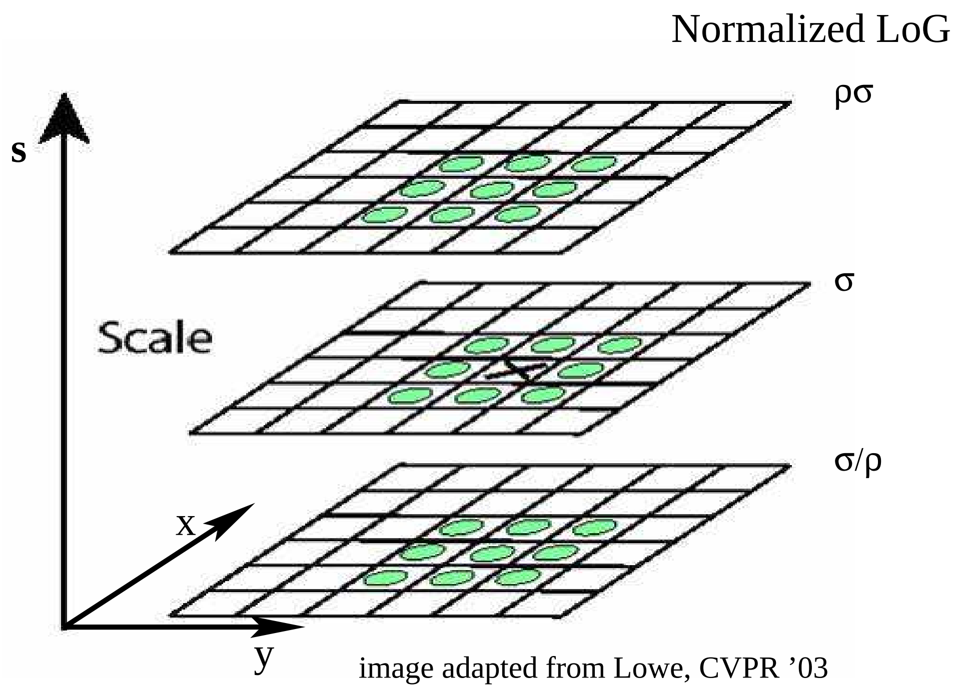


image adapted from Lowe, CVPR ’03

The strong positive local maxima, and negative local minima, (i.e. peaks and pits) can be found in the normalized LoG response images. These are extrema in both the spatial coordinates x, y and the scale coordinate s.

Scale Selection in Practice

A monochrome brightness image $I(\vec{x})$ was formed for the sunflower image on p.3 (repeated again on the next slide).

The normalized LoG filter was applied to $I(\vec{x})$ at scales σ ranging from 2 to 64 pixels, with σ increasing by a factor of $\rho = 2^{1/4}$ each step.

Local maxima were identified in these scale-space (i.e., x, y, and $\sigma(s)$) response images. We kept the local maxima larger than a fixed threshold T, i.e., with $\sigma^2[\triangle g * I](\vec{x}) \geq T > 0$. These maxima correspond to relatively dark regions in the image (as compared to their surrounding regions).

Relatively bright regions could also have been found by finding local minima such that $\sigma^2[\triangle g * I](\vec{x}) \leq -T < 0$. We omit these regions to avoid clutter in the displayed responses.

See `nrmLoG_Sunflowers` movie.

Sunflower Image

We include the sunflower image below to allow for comparison with the subsequent results.

Image is `Field_of_Sunflowers_Kentucky.jpg` from picasaweb.google.com.

Scale Selection in Practice

All the local maxima (in $\vec{x}$ and $\sigma(s)$) for which $\sigma^2[\triangle g * I](\vec{x}) \geq T$ are shown in the image below.

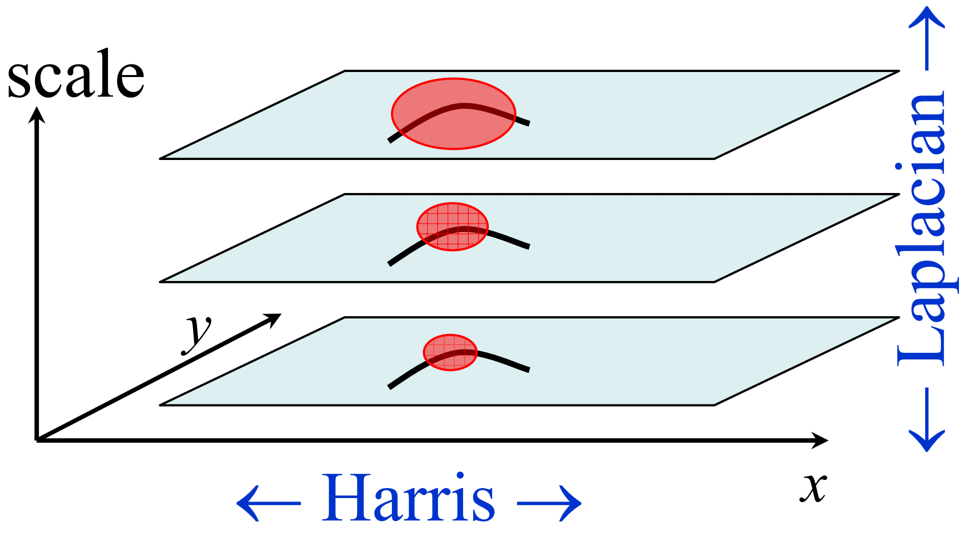
The circles are centered at each strong local maxima, (x_k, y_k), with radii equal to $\sigma(s_k)$. Note that relatively dark regions are successfully identified in both position and scale. Non-circular regions can also be seen to lead to one or more responses.

Scale Selection in Practice

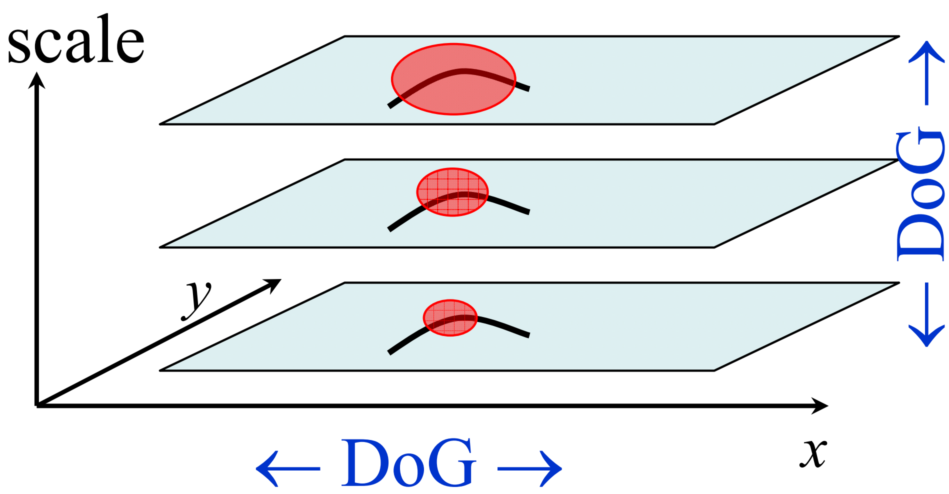
The following three slides are from Darya Frolova and Denis Simakov, of the Weizmann Institute.

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 of:
  - Difference of Gaussians in space and scale



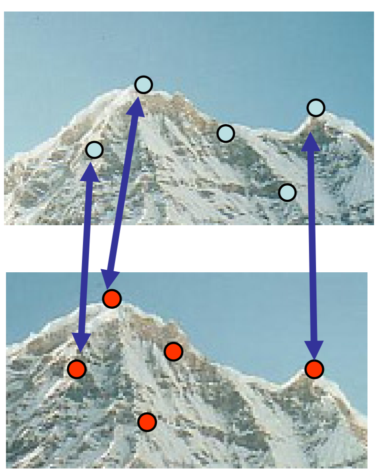
[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". Accepted to IJCV 2004

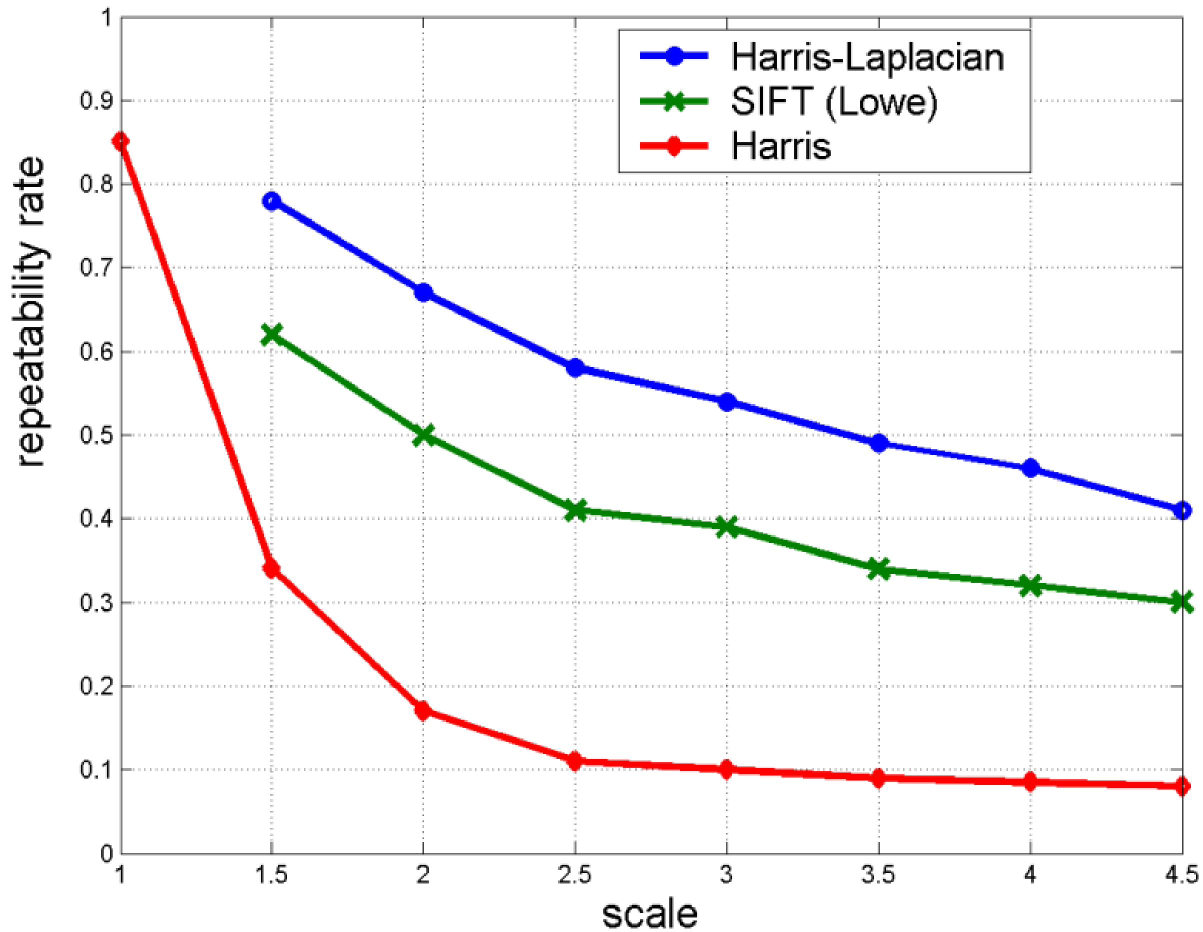
Scale Invariant Detectors

- 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)

Possible methods include:

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

Canonical Orientation(s) of an Image Patch

Given a scale-invariant image patch, as detected by one of the previous approaches, we consider how to obtain a canonical image orientation for this patch.

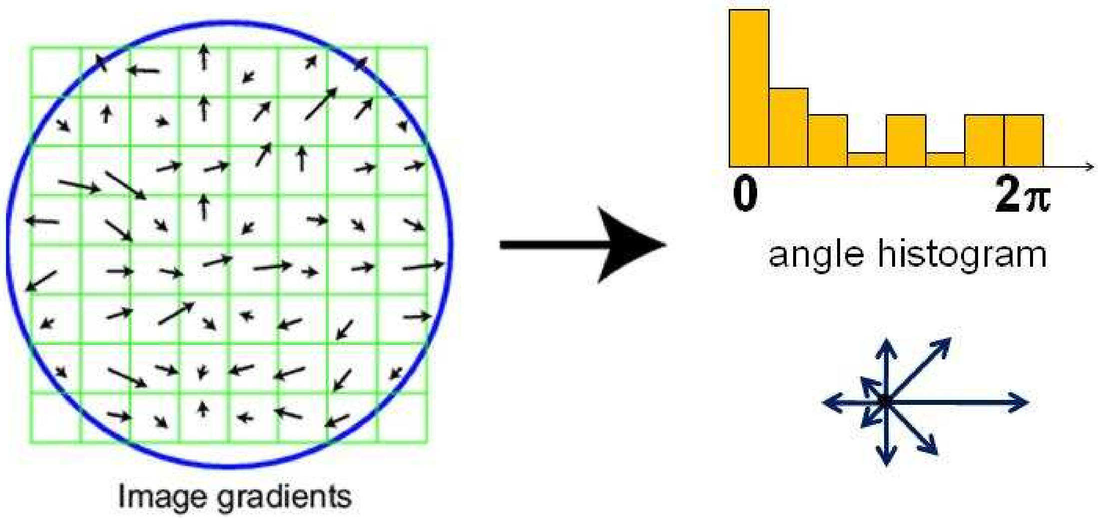


Image from D. Lowe, CVPR 2003 tutorial.

The direction and magnitude of image gradients within a scaled neighbourhood of the center of the detected patch are used to form a gradient orientation histogram.

The significant peak(s) of this histogram are used to define the canonical orientation(s) of the patch.

Summary: Image Landmarks

The detection of the image position, scale and orientation of image landmarks is done *independently* across each image. An abstract representation of the image patch (such as a HoG model) is stored for each landmark.

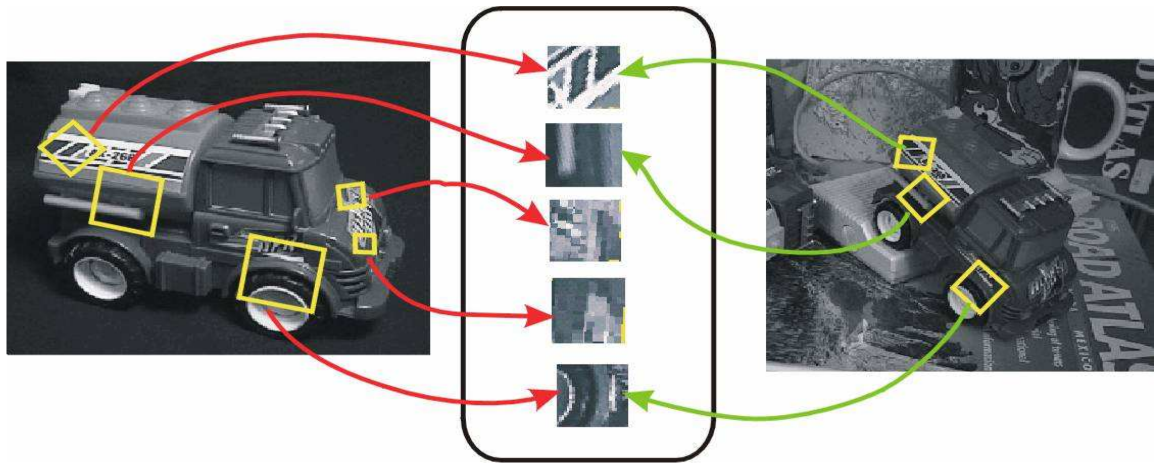


Image from D. Lowe, CVPR 2003 tutorial.

The same landmark can be identified in multiple images by comparing the image patch descriptors. We use such features later in the course to:

- infer the 3D geometry of the scene from multiple viewpoints;
- do view-based object recognition;
- begin object category recognition (e.g., cows or bicycles).

References

A few papers on local image features in Computer Vision are listed below:

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

Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, SURF: Speeded Up Robust Features, Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346–359, 2008.

T. Tuytelaars and K. Mikolajczyk , Local Invariant Feature Detectors - Survey, Foundations and Trends in Computer Graphics and Vision , 3(3):177-280, 2008.

K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors. In PAMI 27(10), pp.1615-1630, 2005.

The following paper uses HoG for the purpose of detecting pedestrians in images:

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, *International Conference on Computer Vision and Pattern Recognition*, Vol.2, June 2005, pp.886-893.

The following papers discuss a roughly similar abstraction to HoG, that is proposed to be used by the human visual cortex:

Riesenhuber, M., and Poggio, T., Hierarchical models of object recognition in cortex, Nature Neuroscience, 2 (11), 1999, p. 1019-1025.

Shimon Edelman, Nathan Intrator and Tomaso Poggio, Complex Cells and Object Recognition, NIPS 1997.