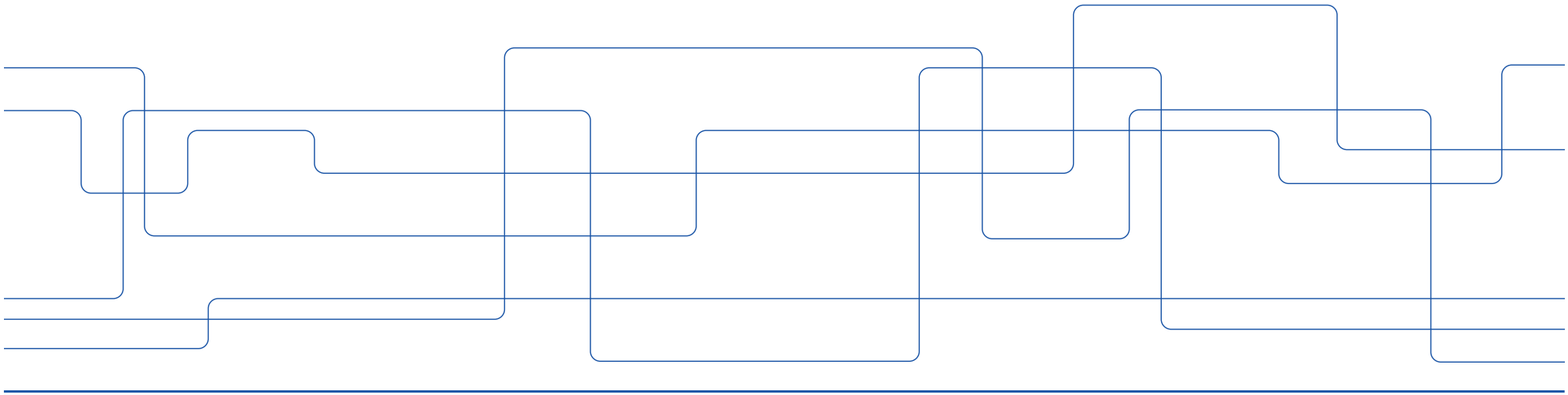
Image features II

Mårten Björkman

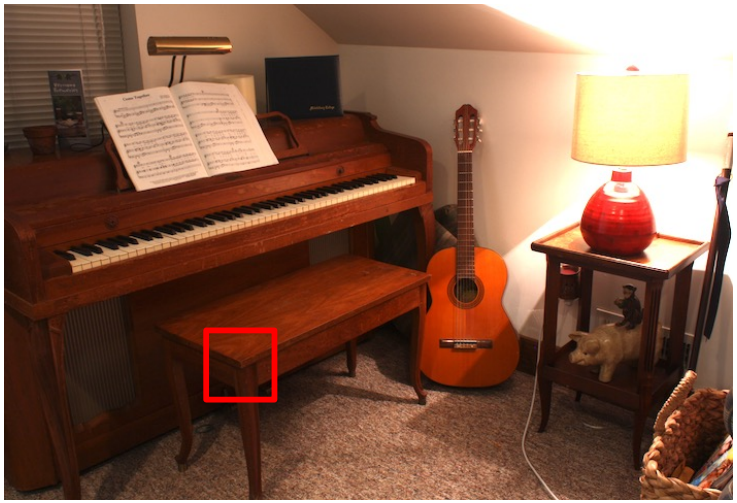


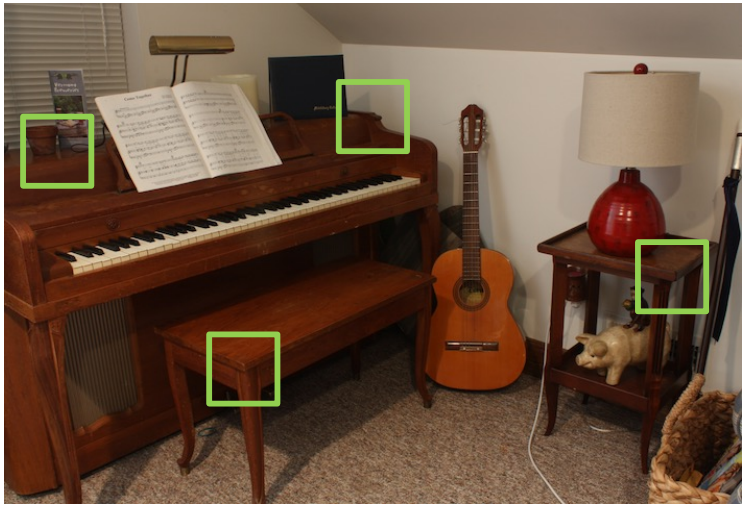
Motivation

Feature-based methods for e.g. stereo and motion estimation typically require determination of correspondences between features in several images.

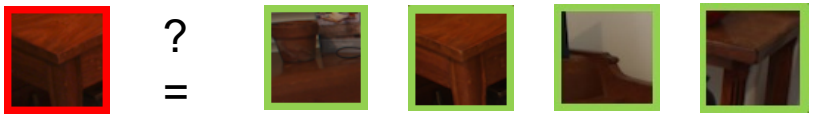
Matching image patches

Assume we want to match an image patch of fixed size.





Task: find the best (most similar) patch in the second image.

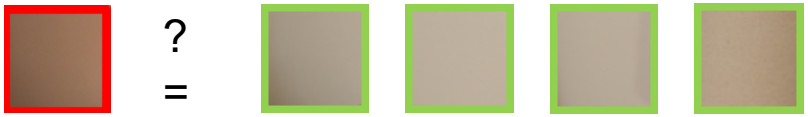


Intuition: this would be a GOOD patch to match, since it looks different from other patches.

Matching image patches

Assume we want to match an image patch of fixed size.

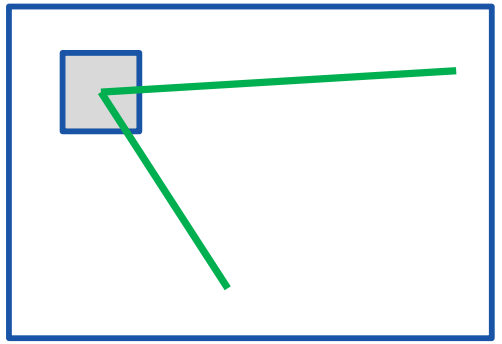
Task: find the best (most similar) patch in the second image.



Intuition: this would be a BAD patch to match, since it looks similar to many other patches.

Interest points: Intuitive ideas

- We should easily recognize the point by looking at the pixels in a small window.
- Shifting the window in any direction should yield a large change in appearance.



- A good interest point should:
  - have a clear mathematically well-founded *definition*
  - have a well-defined *position* in image-space
  - be rich in *information content* in a local neighbourhood such that the interest points can be reliably matched
  - be *stable* under natural image transformations
  - be sufficiently *distinct* such that interest points corresponding to different physical points can be kept separate

Second-moment matrix

- Idea: Accumulate statistics of local directions in regional neighbourhood around every image point.
  - With vector notation:

$$\mu(x) = \sum_{q \in N_x} \nabla L(q) \nabla L^T(q) \; g(x - q; s)$$

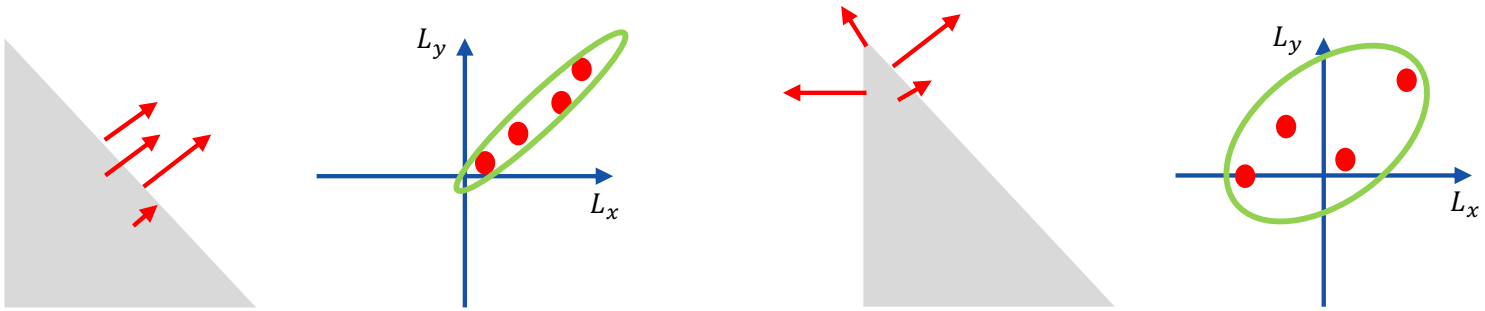
  - In terms of coordinates:

$$\mu(x) = \sum_{q \in N_x} \begin{pmatrix} L_x^2(q) \; g(x - q; s) & L_x(q) L_y(q) \; g(x - q; s) \\ L_y(q) L_x(q) \; g(x - q; s) & L_y^2(q) \; g(x - q; s) \end{pmatrix}$$

- Images are prefiltered $L(x; t) = g(x; t) * f(x)$ where t is the local scale.
- The scale s determines the size of the window in which statistics is collected.
- Sometimes $g(x; s)$ is ignored and you have a summation with uniform weights.

Second-moment matrix

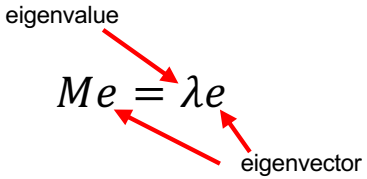
- The second-moment matrix can be viewed as a covariance matrix of gradients.



- In the first location the gradients are spread along a line. BAD!
- In the second location the gradients are spread in more directions. GOOD!

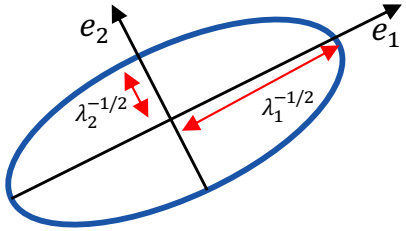
Compute eigenvalues and eigenvectors

Assume we have a matrix M and want to find its eigenvalues and eigenvectors



1. Compute the determinant of $M - \lambda I$
2. Find the roots λ_i of the polynomial $\det(M - \lambda I) = 0$
3. For each eigenvalue λ_i, solve $(M - \lambda_i I)e_i = 0$

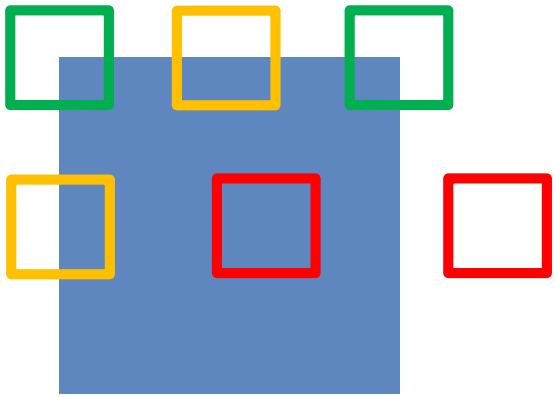
If M is symmetric, e.g. a covariance matrix, it can be visualized as an ellipse.



Properties of the second-moment matrix

Eigenvalues $\lambda_1 \geq \lambda_2 \geq 0$ of the second-moment matrix μ:

- Smooth image region without strong edge structures:
  - Both eigenvalues λ_1 and λ_2 small.
- Along a straight edge:
  - One eigenvalue much larger than the other, $\lambda_1 >> \lambda_2$.
- At a corner with two or more dominant directions:
  - Both eigenvalues λ_1 and λ_2 large.



The Harris operator (Harris & Stephens 1988)

- Given the second-moment matrix μ with eigenvalues $\lambda_1, \lambda_2 \geq 0$, compute the Harris measure

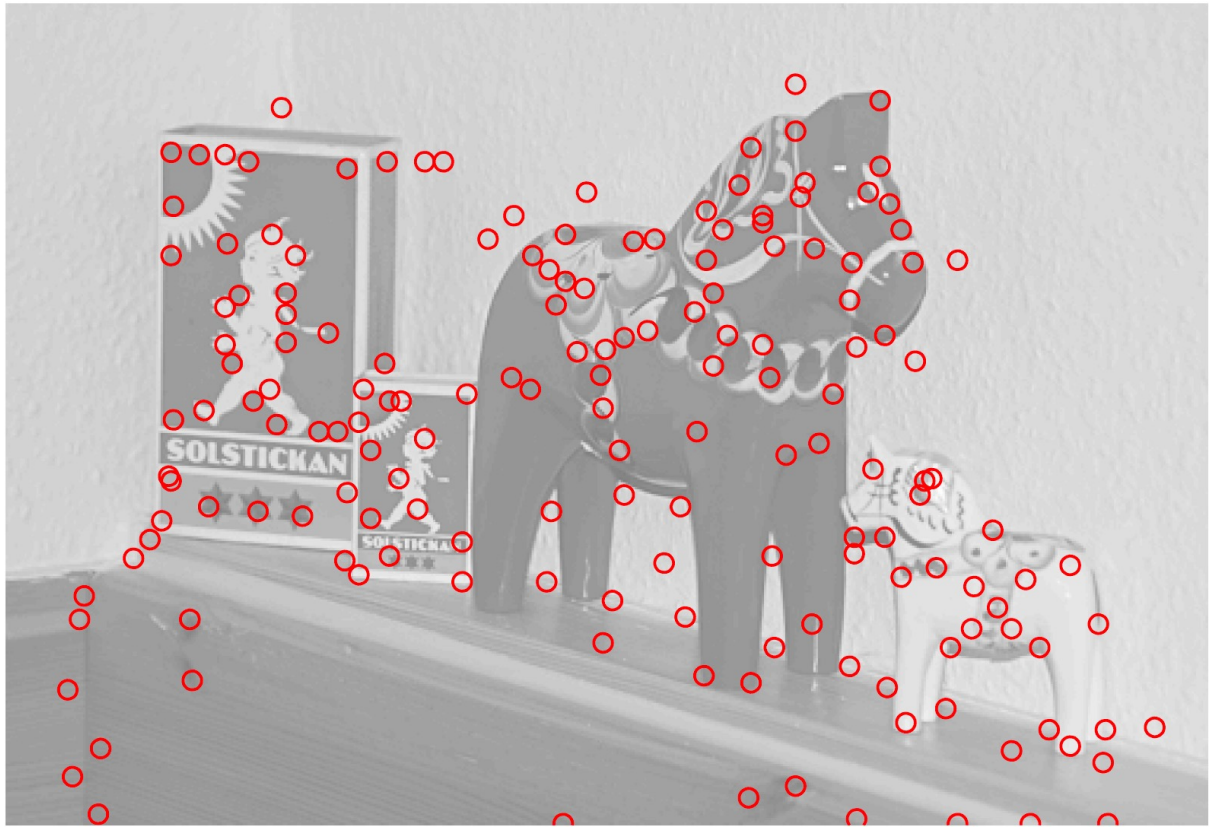
$$H = \det(\mu) - k\,\mathrm{trace}^2(\mu) = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$$

  where k is a constant in the range $[0,1/4]$, but typically $k \in [0.04,0.06]$.

- Then, detect thresholded local maxima of H with $H \geq H_0 > 0$.

- What does this mean?
  - Both eigenvalues λ_1, λ_2 have to be large enough for $H \geq H_0$.
  - The difference between eigenvalues should be sufficiently small (controlled by k).
  - The ellipses on the earlier slide should be large, but not elongated.

The Harris operator



Algorithmic steps: Harris corner detection

1. Smooth the image using a Gaussian, $L(x;t) = g(x;t) * f(x)$
2. Compute partial derivatives L_x and L_y at some scale t.
3. Compute products L_x^2, $L_x L_y$ and L_y^2 at every image point.
4. Compute weighed sums of products $E[L_x^2]$, $\mathrm{E}[L_x L_y]$ and $\mathrm{E}[L_y^2]$ using window function $g(x;s)$ of scale s.
5. At every image point compute

$$H = \det(\mu) - k\,\mathrm{trace}^2(\mu) = E[L_x^2]\mathrm{E}[L_y^2] - E\left[L_x L_y\right]^2 - k\left(E[L_x^2] + \mathrm{E}[L_y^2]\right)^2$$

6. Detect local extrema of H that satisfy the thresholding criterion $H \geq H_0$.

Laplacian blob detection

- Given a scale-space representation $L(x;t)$ of an image $f(x)$ obtained by Gaussian smoothing

$$L(x;t) = g(x;t) * f(x)$$

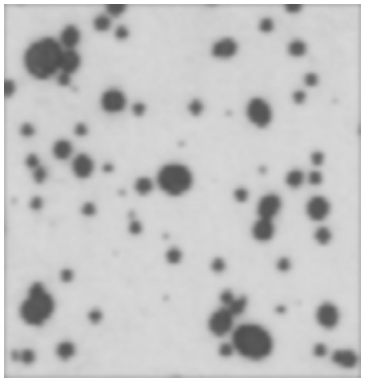
compute the Laplacian operator

$$\nabla^2 L = L_{xx} + L_{yy}$$

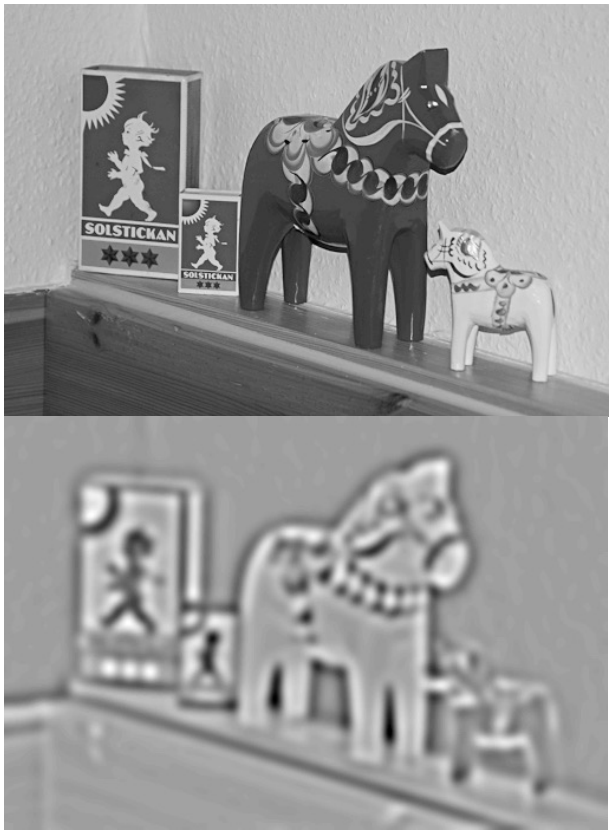
- Find spatially local extrema of $\nabla^2 L$ and regard these as blob responses with

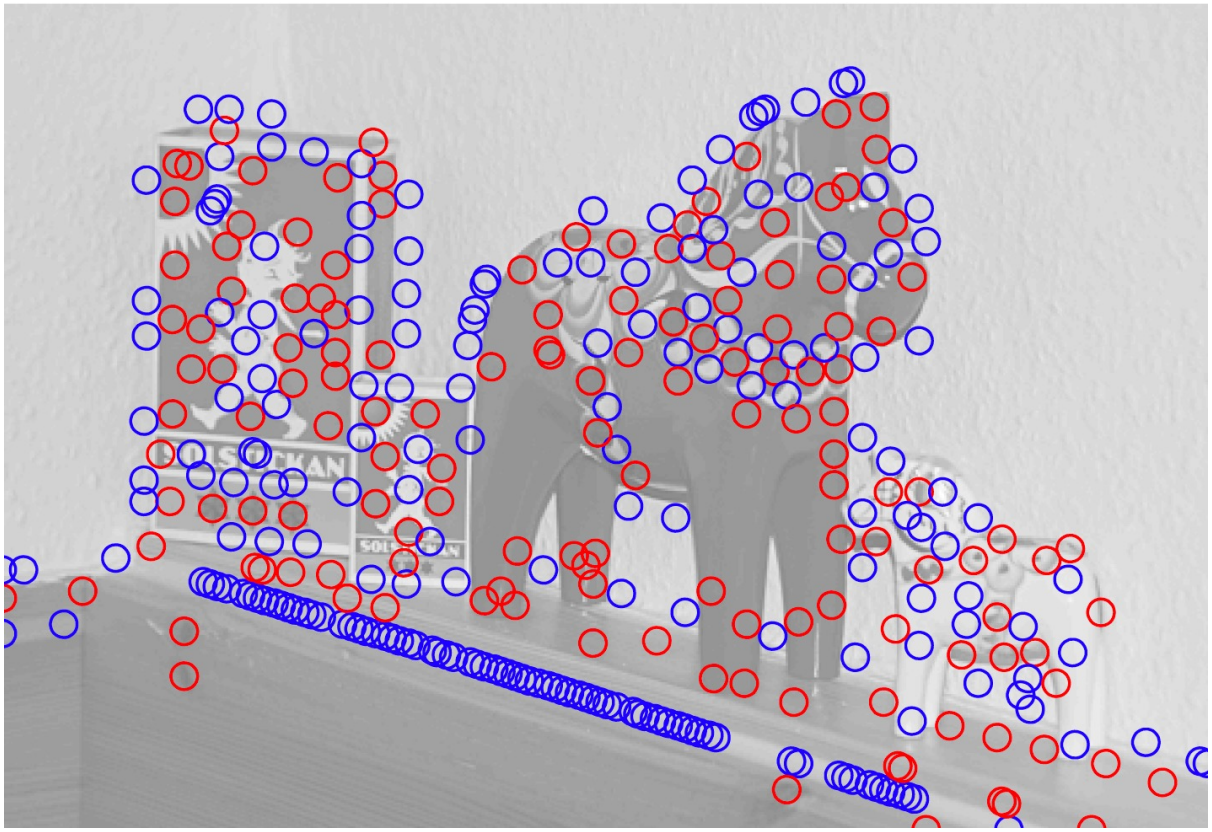
$\nabla^2 L < 0 \implies$ "bright blob"

$\nabla^2 L > 0 \implies$ "dark blob"



The Laplacian operator





Determinant of the Hessian blob detection

- Given a scale-space representation $L(x;t)$ of an image $f(x)$ obtained by Gaussian smoothing

$$L(x;t) = g(x;t) * f(x)$$

compute the determinant of the Hessian matrix

$$\det \mathcal{H}L = L_{xx}L_{yy} - L_{xy}^2$$

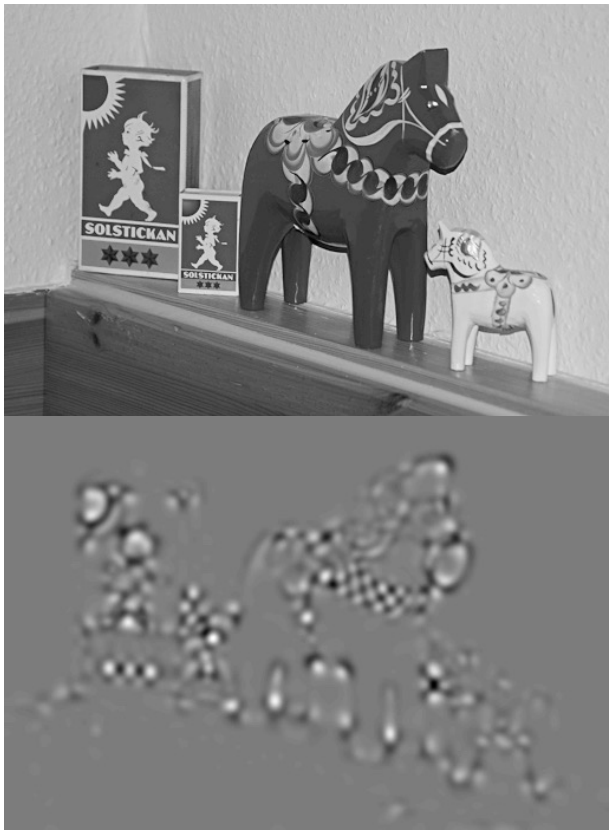
- Find positive local maxima of $\det \mathcal{H}L$ and regard these as blob responses with

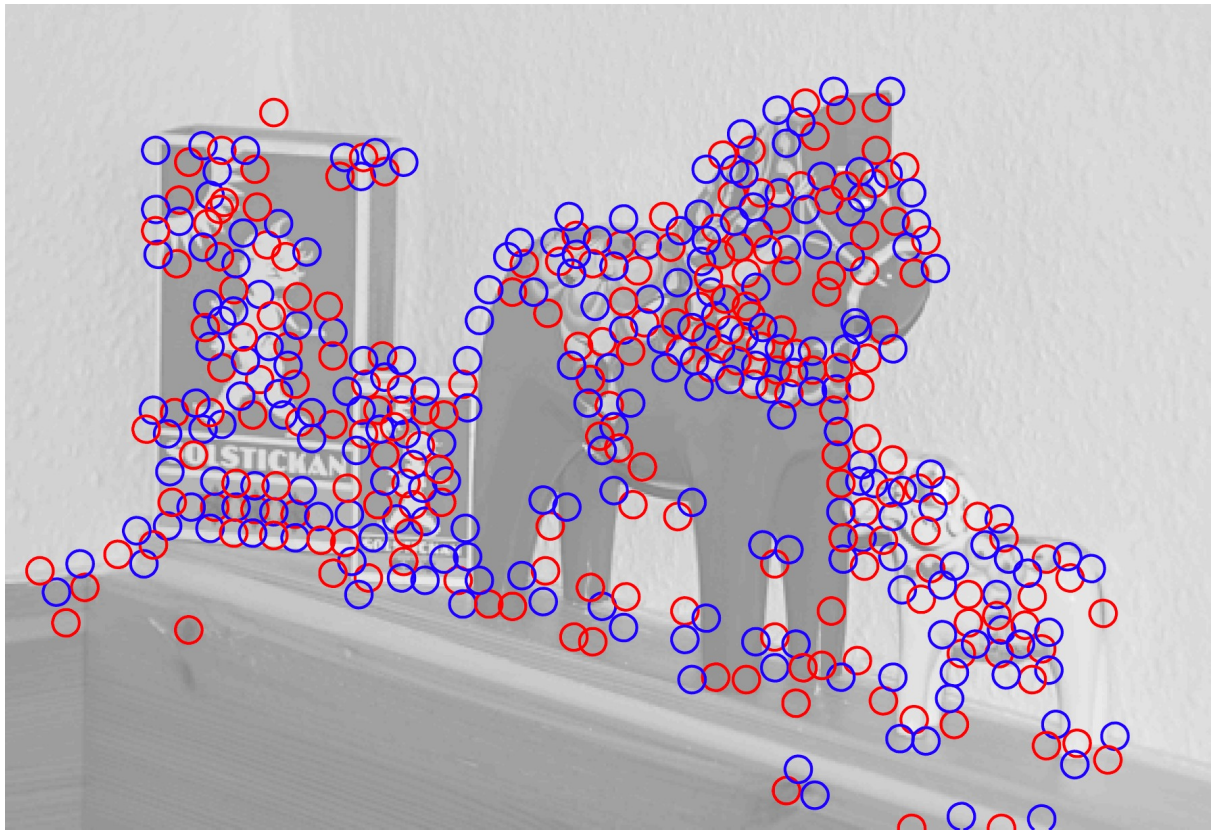
$$\nabla^2 L < 0 \implies \text{“bright blob”}$$

$$\nabla^2 L > 0 \implies \text{“dark blob”}$$

- Negative local minima of $\det \mathcal{H}L$ can be regarded as saddle-like features.

The determinant of the Hessian operator





Laplacian vs. determinant of the Hessian

- In a coordinate frame aligned to the eigendirections of the Hessian matrix

$$\mathcal{H}L = \begin{pmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

- The Laplacian and determinant of the Hessian operators correspond to

$$\nabla^2 L = L_{xx} + L_{yy} = \lambda_1 + \lambda_2$$

$$\det \mathcal{H}L = L_{xx}L_{yy} - L_{xy}^2 = \lambda_1 \lambda_2$$

Thus,

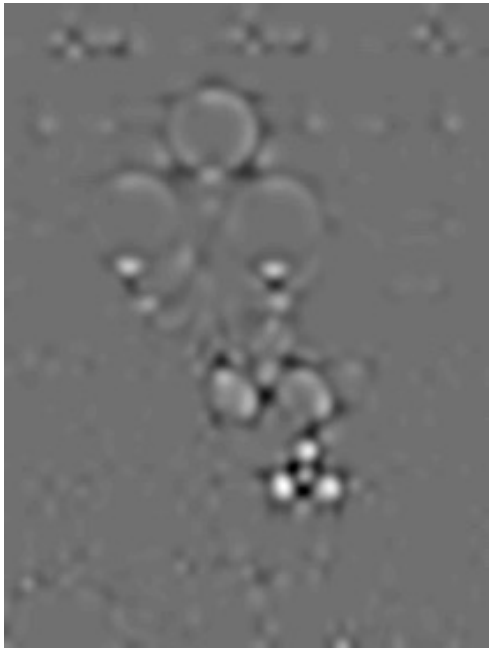
- For the Laplacian, it is enough that the response is strong in either one of the directions, i.e. either λ_1 or λ_2 is large enough.
- For the determinant of the Hessian, the response should be strong in both directions, i.e. either λ_1 or λ_2 are large enough.
- This is why spurious responses are seen along edges for the Laplacian, but not for the determinant of the Hessian.

Blob detection from differential invariants

- Strong blob responses can be obtained provided that the scale level is adapted to the size of the image structures in the image domain

Original image

$\nabla^2 L$ at $t = 16$



$\det \mathcal{H} L$ at $t = 16$

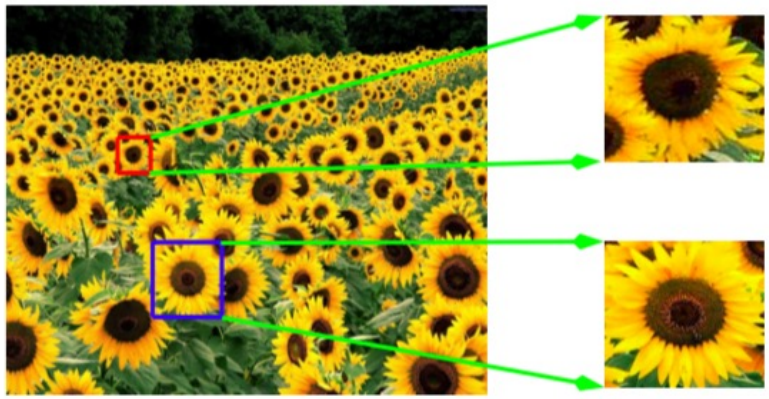
Need for scale selection

Major problems when applying interest point detectors at a single fixed scale:

- A single scale level may not be sufficient to capture the relevant image structures in a given image.
- Resulting image features may depend strongly on the imaging conditions, e.g., if the same object is seen from two different distances.

$\Longrightarrow$ Desirable to include a mechanism for:

- selecting scale levels automatically and
- computing scale-invariant image features.



Scale selection from normalized derivatives

- Observation: Image derivatives decrease for increasing scale and so does the maximum responses of the feature detectors just described.

  How can we correct for this and make responses at different scale comparable?

- Solution: Use scale-normalized derivatives

$$\delta_\xi = t^{1/2}\,\delta_x,\ \delta_\eta = t^{1/2}\,\delta_x$$

  and the detectors with their scale-normalized counterparts, e.g.

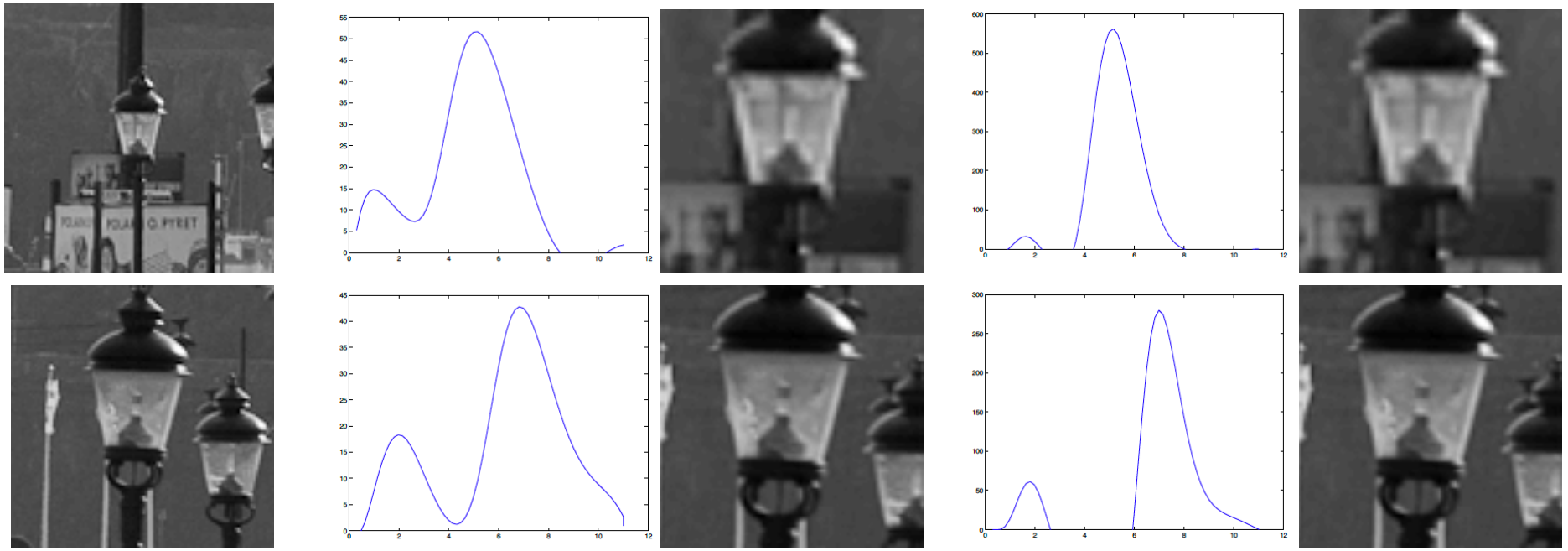
$$\nabla^2_{norm} L = t\,(L_{xx} + L_{yy})$$

$$\det \mathcal{H}_{norm} L = t^2 (L_{xx} L_{yy} - L^2_{xy})$$

- Detect scale-space extrema, i.e. not only over image space, but also over scale.
- This allows for the inherent size of local image structures to determined and a scale-invariant reference frame to be created.

(Lindeberg 1998)

Scale-invariant reference frame



original | scale estimate | scale normalized | scale estimate | scale normalized

$\nabla^2_{norm} L$ $\det \mathcal{H}_{norm} L$

Scale selection applied to Gaussian blob

Input image = Gaussian blob:

$$f(x,y) = g(x,y;t_0) = \frac{1}{2\pi t_0} e^{-\frac{x^2+y^2}{2t_0}}$$

Scale-space representation by semi-group property:

$$L(x,y;t) = g(x,y;t) * g(x,y;t_0) = g(x,y;t+t_0) = \frac{1}{2\pi(t+t_0)} e^{-\frac{x^2+y^2}{2(t+t_0)}}$$

Scale-normalized Laplacian at origin, $(x,y) = (0,0)$:

$$\nabla^2_{norm} L = t\left(L_{xx} + L_{yy}\right) = t\left(\left(\frac{x^2 - t_0 + t}{(t_0+t)^2}\right)L + \left(\frac{y^2 - t_0 + t}{(t_0+t)^2}\right)L\right) = -\frac{t}{\pi(t_0+t)^2}$$

Differentiate with respect to scale:

$$\delta_t \nabla^2_{norm} L = 0 \Longrightarrow \hat{t} = t_0$$ ”measures size of Gaussian blob”

Scale selection applied to sine wave

Input signal = 1D sine wave:

$$f(x) = \sin\omega_0 x$$

Scale-space representation obtained using a Fourier transform:

$$L(x;t) = g(x;t) * f(x) = \ e^{-\omega_0^2 t/2}\sin\omega_0 x$$

Amplitude of scale-normalized second order spatial derivative:

$$L_{\xi\xi,max} = t\omega_0^2 e^{-\omega_0^2 t/2}$$

Differentiate with respect to scale, maximize, and express in terms of standard deviation $\sigma = \sqrt{t}$ and wavelength $\lambda_0 = 2\pi/\omega_0$:

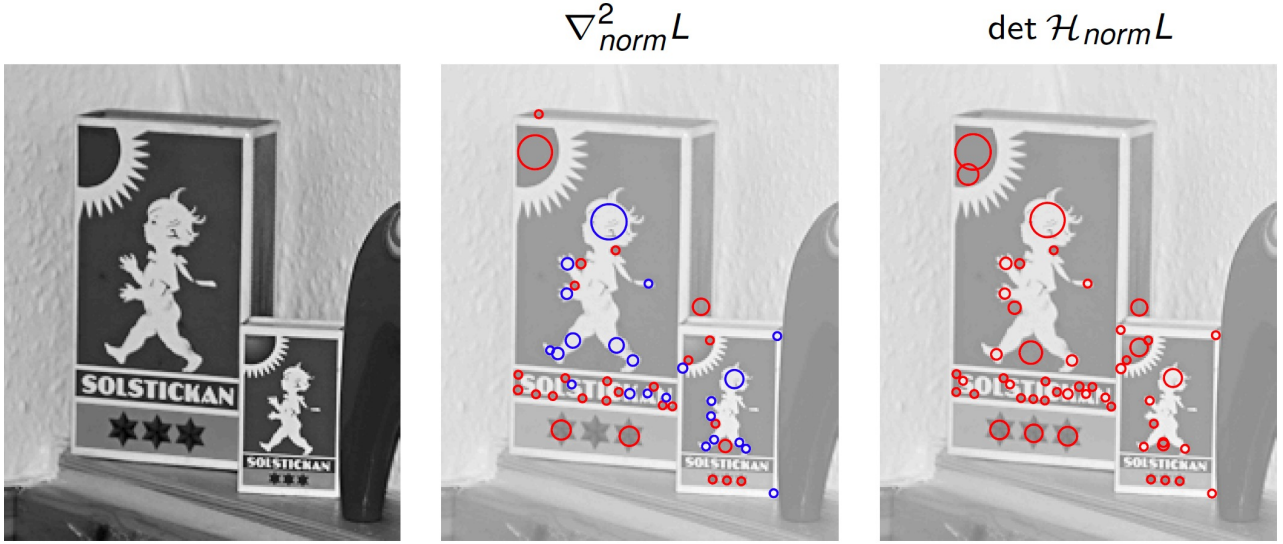
$$\hat{\sigma} = \frac{1}{\sqrt{2}\pi}\lambda_0 \quad \Longrightarrow \text{ “scale estimate proportional to wavelength”}$$

Scale invariance

General scaling property: If the original image f is rescaled by uniform scaling factor $f'(x', y') = f(sx, sy)$, then

- a scale-space extremum in f at $(x_0, y_0; t_0)$ is transformed to a scale-space extremum in f' at $(x_0', y_0'; t_0') = (sx_0, sy_0; s^2 t_0)$

$\Longrightarrow$ selected scale levels automatically adapt to scaling variations.



Algorithmic steps

Scale-space extrema detection algorithm:

1. Given a scale-range $[t_{min}, t_{max}]$, distribute a set of scale levels uniformly in terms of effective scale $\tau = \log t$.
2. Convolve image by a Gaussian kernel to each scale level.
3. For every image point, compute approximations of necessary derivatives and combine these into the desired measures, e.g. $\nabla^2_{norm} L$.
4. Detect local extrema over scale and space in $3 \times 3 \times 3$ neighbourhoods and by thresholding on the magnitude of the response.
5. Optionally, sort the interest points in decreasing order with respect to their scale-normalized magnitude values.

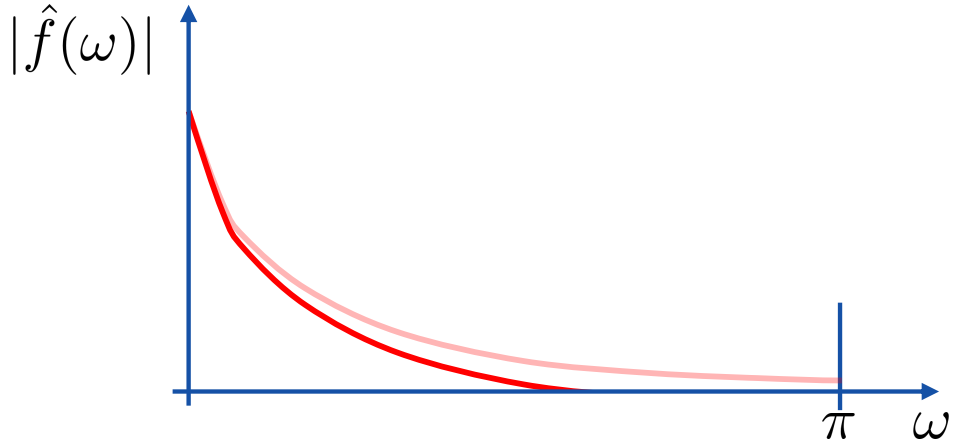
Real-time implementation with pyramids

Motivation:

- A regular scale-space representation uses the same resolution at all scales.
- But… with increasing scales, small-scale image structures will be suppressed.

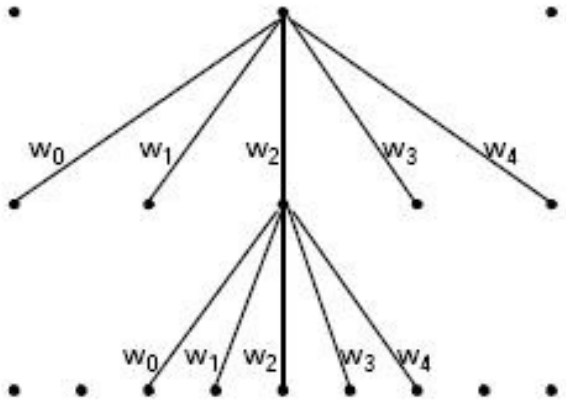
$\Longrightarrow$ It should be possible to subsample coarser scale representations to improve the computational efficiency at coarser scales.

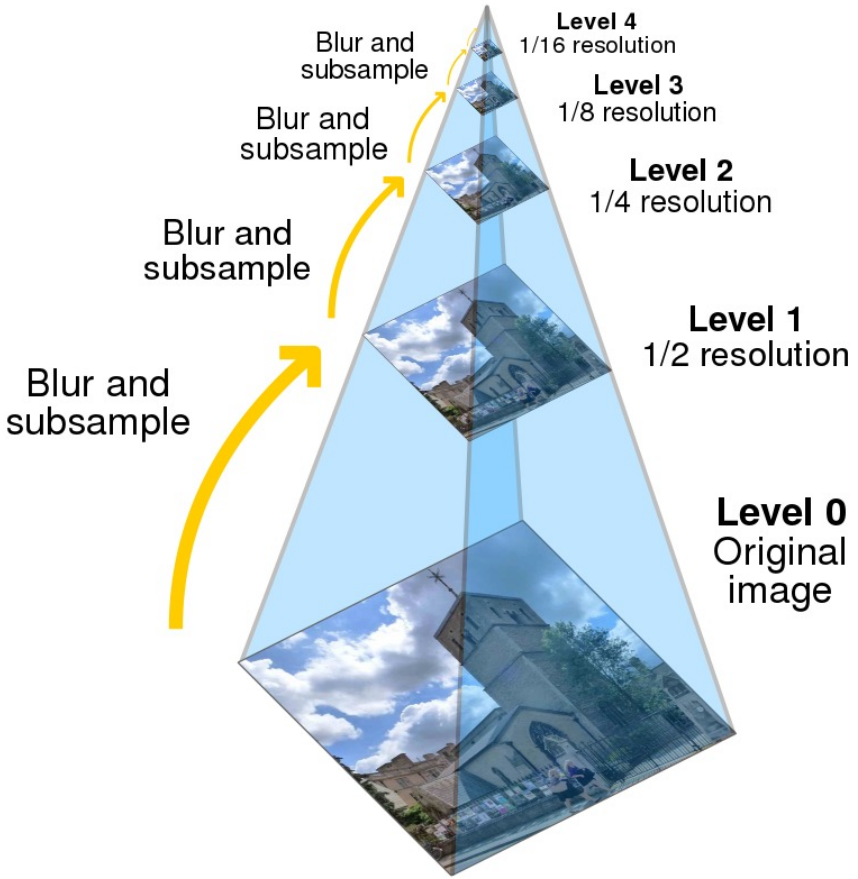
With enough blurring, you can subsample without losing any information.



Pyramid representation

- Basic idea: combine successive smoothing and subsampling.
- Usually: from $2^n w \times 2^n h$ image compute $2^{n-1} w \times 2^{n-1} h$ image.





Pyramid construction

In the 1D case (extended to 2D be reparable filtering) we have

$$L^{k-1} = \text{REDUCE}\left(L^k\right), \qquad L^{k-1}(x) = \sum_n c(n) L^k(2x - n)$$

where the filter coefficients should satisfy

- positivity: $c(n) \geq 0$
- symmetry: $c(-n) = c(n)$
- unimodality: $c(n) \geq c(n+1), \text{ for } n \geq 0$
- normalization: $\sum_n c(n) = 1$
- equal contribution: $\sum_n c(2n) = \sum_n c(2n+1)$

Binomial kernels

For kernels with 5 coefficients (c, b, a, b, c):

- the normalization condition implies: $a + 2b + 2c = 1$
- the equal contribution condition implies: $a + 2c = 2b$

$$\Longrightarrow \begin{cases} a \geq 1/4 & \text{(by unimodality } a \geq b\text{)} \\ b = 1/4 & \\ c = 1/2 - a/4 & \end{cases}$$

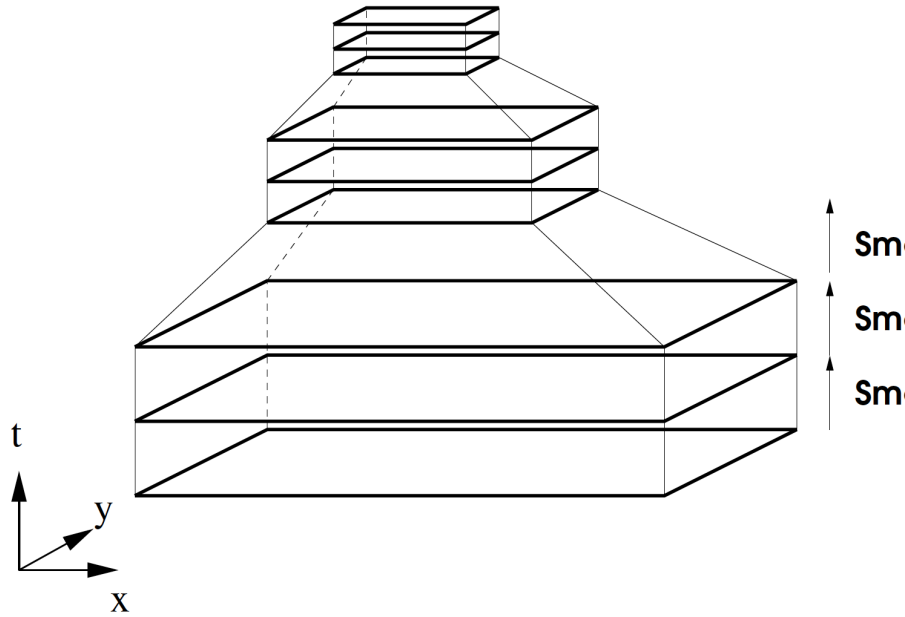
Empirically, Burt & Adelson (1983) suggested $a \approx 0.4$.

The value $a = 3/8 = 0.375$ corresponds to the binomial kernel

$$\frac{1}{16}(1, 4, 6, 4, 1) = \frac{1}{16}(1, 2, 1)^2$$

Hybrid pyramids

- Allow for intermediate scale levels between each subsampling operation
- Allow for different trade-offs between computational accuracy and efficiency.



Usually, you have 5-6 scales per subsampling, known as an octave.

Trade-offs in hybrid pyramids

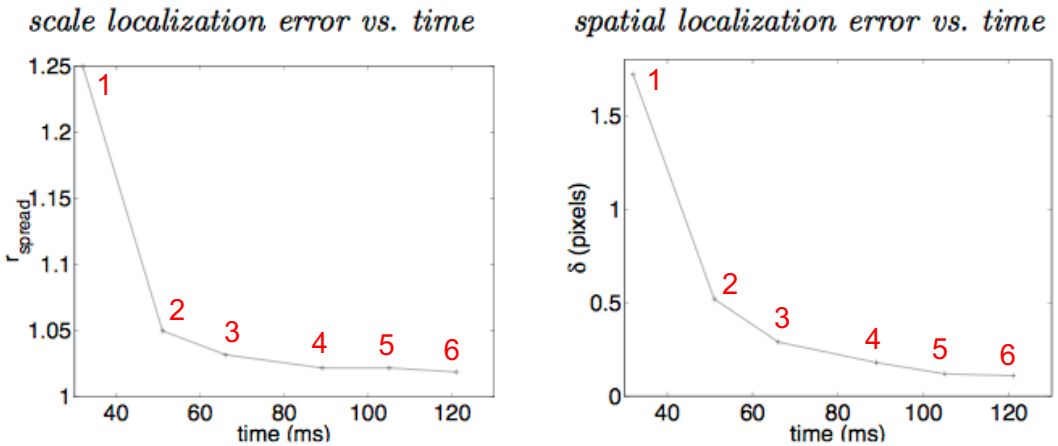


Fig. 4: Trade-offs between the localization error (vertical axis) and the computation time (horizontal axis) for hybrid pyramids with different values of ρ: (left) scale localization error, (right) spatial localization error.

Figure from Lindeberg and Bretzner (2003) “Real-time scale selection in hybrid multi-scale representations”, Proc. Scale-Space Methods in Computer Vision, Springer LNCS 2695: 148–163.

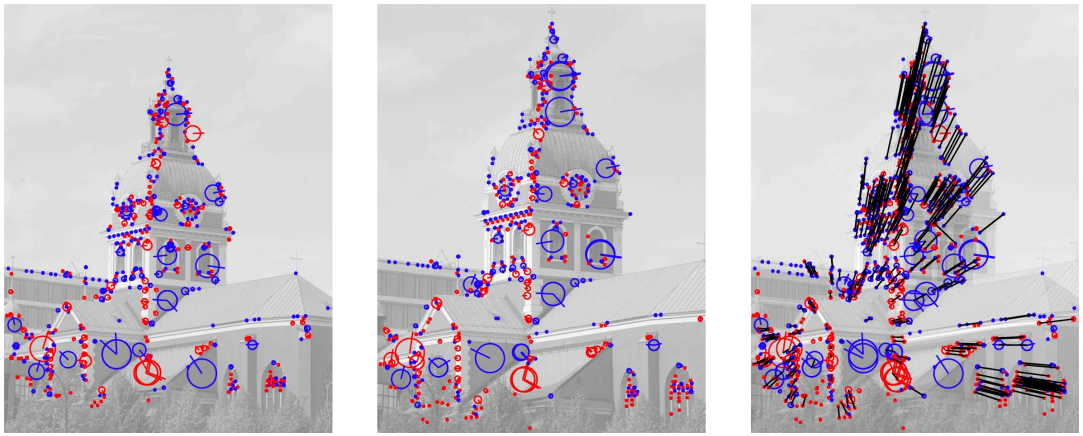
Image-based matching and recognition

Common approach to image-based matching and recognition:

1. Detect interest points.
2. Compute image descriptors around the interest points.
3. Match the image descriptors.

SIFT (Lowe 2004), SURF (Bay et al 2008) and related approaches.

Applications: multi-view matching, object recognition, video tracking, gesture recognition, panorama stitching, robot localization and matching.

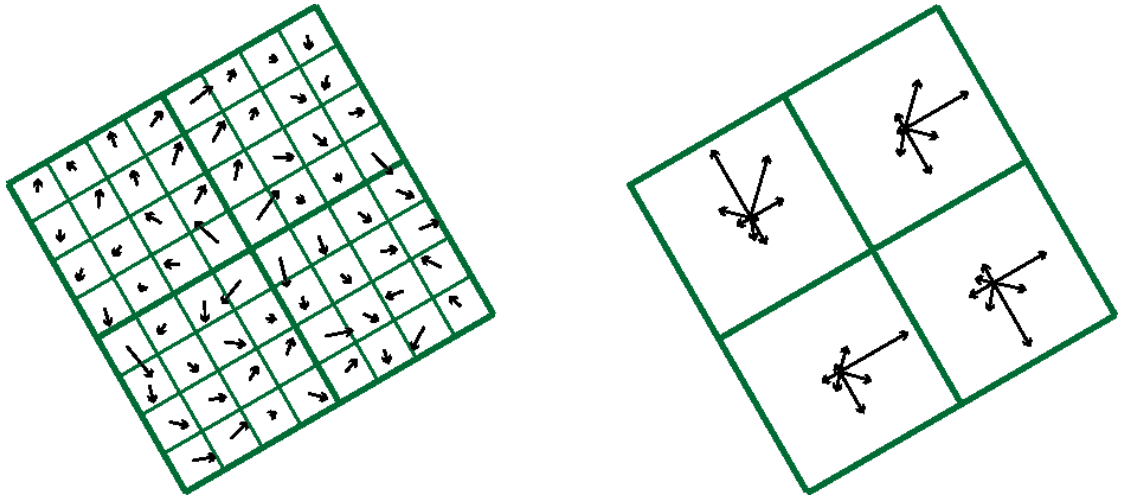


The SIFT descriptor

- Compute gradients ∇L around (x_o, y_0) at scale $\hat{t}$ of interest point.
- Estimate the orientation of the interest point by computing a 36-bin histogram of local gradient directions. Create an orientation normalized image frame.
  - Strongest peak(s) $\Longrightarrow$ orientation estimates
- Define 4×4 grid around interest point with spacing proportional to scale in orientation normalized image frame.
  - Accumulate histogram of gradient directions quantized into 8 bins.
  - This histogram is computed in an orientation normalized image frame.
  - Weighted accumulation using overlapping Gaussian window functions.
- Trilinear interpolation for distributing weights between adjacent bins.

The SIFT descriptor

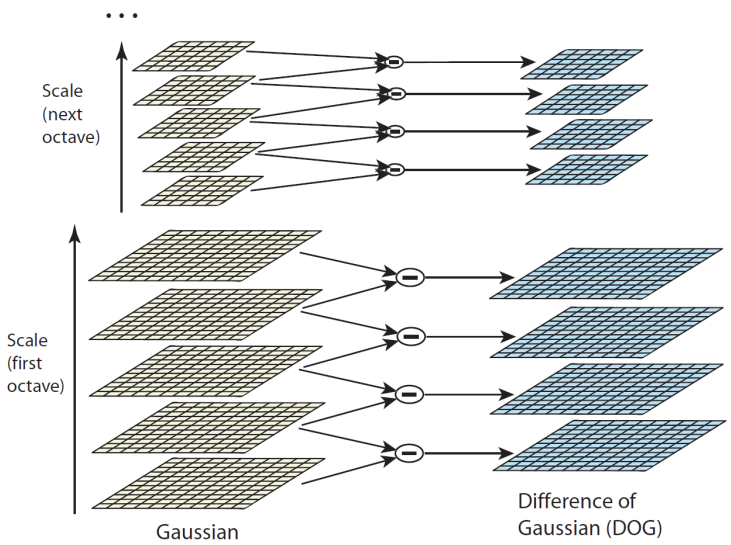
- Place a window around the interest point based on the predicted orientation.
- Collect gradient directions in a grid of histograms.
- Here, schematically illustrated for a 2×2 instead of a 4×4 grid:

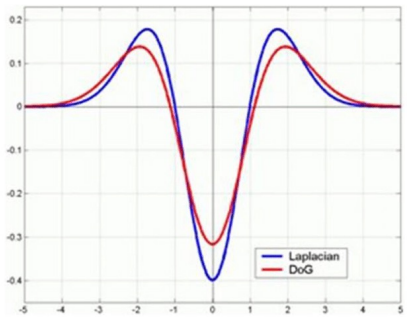


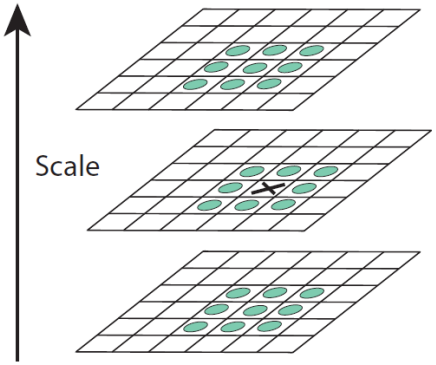
4x4 grid of histograms with 8 bins for gradient directions $\Longrightarrow$ 128D descriptor

SIFT keypoint detection

- Previously, we talked about the Laplacian being used for blob detection.
- SIFT uses a faster approximation based of Differences of Gaussians (DoG).
  Interpretation: “whatever was removed when going from one scale to the next”.







Difference of Gaussians vs Laplacian

- Detect scale-space extrema in difference-of-Gaussians pyramid.
- Approximation of scale-space extrema of Laplacian in scale-space

$$\frac{1}{2}\nabla^2 L(x;t) = \delta_t L(x;t) \approx \frac{L(x;t+\Delta t) - L(x;t)}{\Delta t} = \frac{\mathrm{DOG}(x;\ t,\Delta t)}{\Delta t}$$

- With self-similar scale sampling $\sigma_{i+1} = k\sigma_i$ we have $t_{i+1} = k^2 t_i$

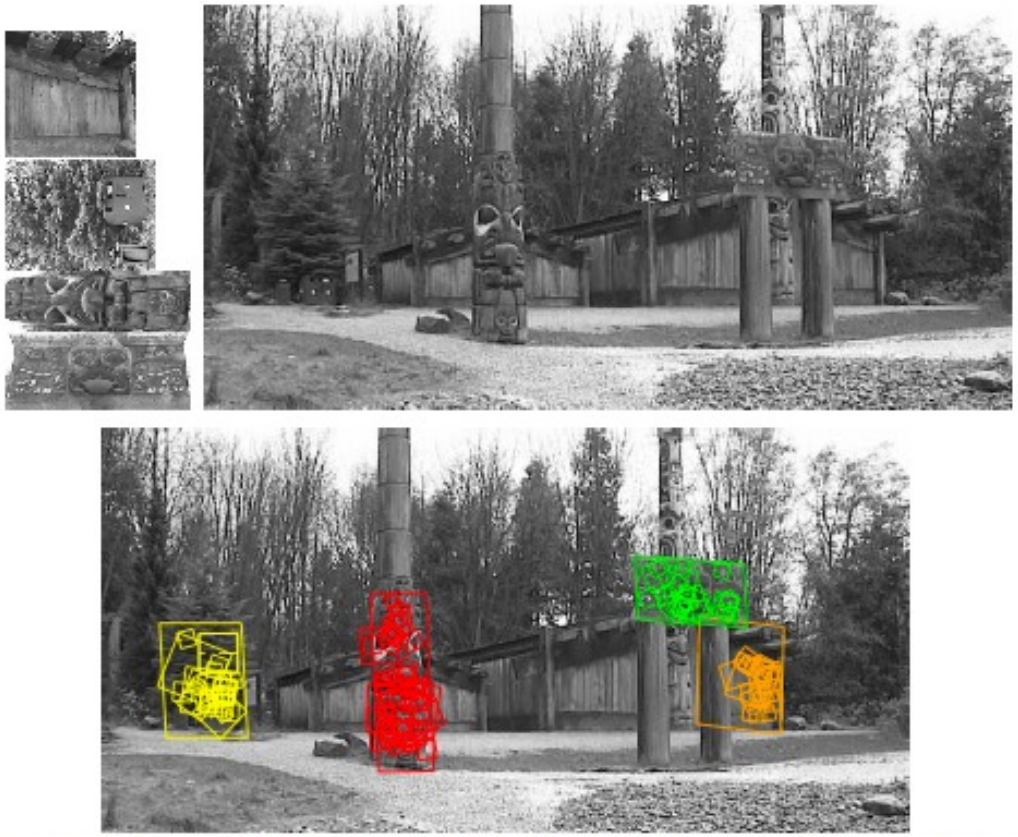
$$\Delta t \nabla^2 L = (k^2 - 1) t \nabla^2 L = (k^2 - 1)\nabla^2_{norm} L$$

which implies

$$\mathrm{DOG}(x;\ t,\Delta t) \approx \frac{(k^2 - 1)}{2}\nabla^2_{norm} L(x;t)$$

Recognition based on SIFT descriptors

Training images



A new image

Figure from Lowe (2004) “Distinctive image features from scale-invariant keypoints”, Int. J. of Computer Vision 60(2): 91–110.

The SURF descriptor

Faster alternative to SIFT, suitable for low-end devices.

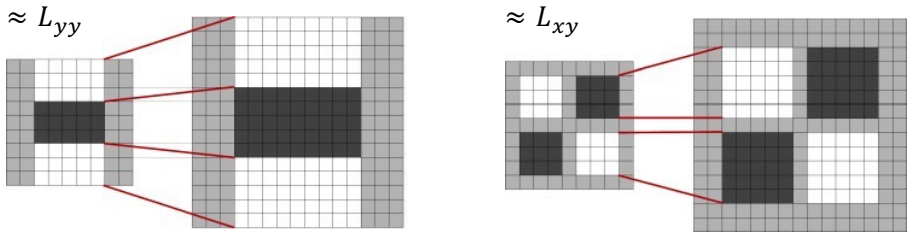
- Compute derivatives L_x and L_y around (x_0, y_0) at scale $\hat{t}$ of the interest point.
- Orientation normalization similar to SIFT.
- Sums of derivatives $\sum L_x$, $\sum |L_x|$, $\sum L_y$, $\sum |L_y|$ over 4×4 over sub-windows around the interest point.

SURF interest point detector

- Interest point detection by determinant of Hessian

$$\det \mathcal{H}L = L_{xx}L_{yy} - L_{xy}^2$$

- Derivatives approximated by Haar wavelets (box filters) to allow for fast computations by integral images. (Bay at al. 2008)



Based on fast computations of pixels over rectangles.

- Fast, but not as accurate as SIFT:
  - Less invariant to rotations
  - Ringing phenomena due to box filters
  - Worse scale-space properties over scale

Matching of image descriptors

- Given two images f_A and f_B, compute sets of interest points $A = \{A_i\}$ and $B = \{B_i\}$ from each image.
- Compare interest points in the two domains by computing the Euclidean difference between their image descriptors.
- Accept match between pair of interest points (A_i, B_j) only if:
  - A_i is the best match for B_j in relation to all the other, points in A.
  - B_j is the best match for A_i in relation to all the other points in B.
- To suppress ambiguous matches, also require the ratio between the distances to the nearest and the next nearest matches to be less than $r = 0.9$.

Experimental protocol

For each pair of images with different scale or viewing variations:

1. Compute interest points from each image separately.
2. Transform interests points to the other image using homography, and scale of interest points by local scale factor of homography.
3. Represent each interest point by circle of size proportional to scale.
4. Accept match if ratio between intersection and union of circles above threshold:

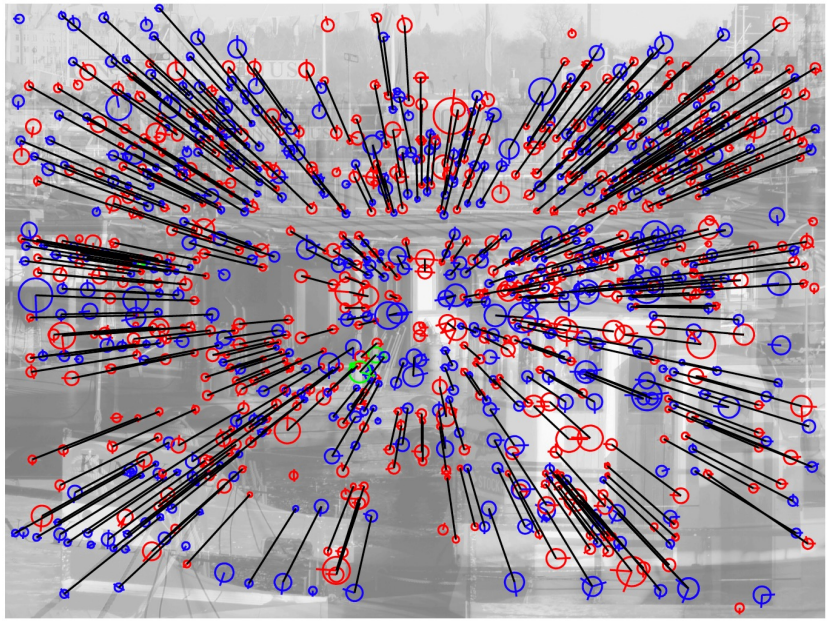
$$m(C_A, C_B) = \frac{|\cap(C_A, C_B)|}{|\cup(C_A, C_B)|} \geq m_0$$

5. Measure performance of interest point detector by:

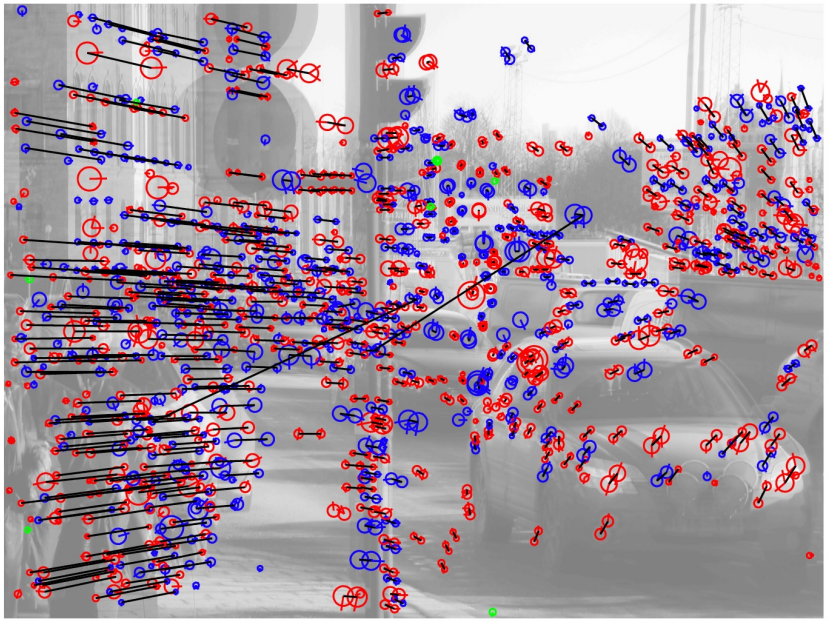
$$\text{efficiency} = \frac{\#(\text{accepted interest point matches})}{\#(\text{interest points})}$$

Examples of matching results

Scaling transformation



Foreshortening transformation



Improved interest point detector with Gauss-SIFT descriptors

Figures from Lindeberg (2015) “Image matching using generalized scale-space interest points”, Journal of Mathematical Imaging and Vision 52(1): 3-36.

Experimental results

- Interest points and image descriptors ranked on matching efficiency:

Interest points	*Descriptor*	*Efficiency*
$\det \mathcal{H}_{norm} L$	*SIFT*	*0.7678*
$\det \mathcal{H}_{norm} L$	*SURF*	*0.7529*
$\nabla^2_{norm} L$	*SIFT*	*0.7498*
$\nabla^2_{norm} L$	*SURF*	*0.7352*
Harris-Laplace	*SIFT*	*0.7024*
Harris-Laplace	*SURF*	*0.6836*

- Conclusions (with Gaussian derivatives for both features and descriptors):
  - $\det \mathcal{H}_{norm} L$ is a better interest point detector than $\nabla^2_{norm} L$.
  - SIFT is a better image descriptor than SURF, but not as fast.
  - both $\det \mathcal{H}_{norm} L$ and $\nabla^2_{norm} L$ are much better detectors than Harris-Laplace.
- Harris-Laplace: Harris for detection, Laplace for scale selection.

Summary of good questions

- What are the motivations for computing interest points? What are they typically used for?
- Describe three common interest point detectors including their mathematical definitions.
- Why is scale selection an important operation?
- Describe how scale selection can be performed in practice.
- What is the motivation for using image pyramids in computer vision?
- How are image pyramids computed from image data?
- Describe a basic trade-off issue that arises in hybrid pyramids.
- What is the purpose of computing image descriptors at interest points?
- How is the SIFT descriptor defined from image data?
- How is the SURF descriptor defined from image data?
- Outline the basic steps in an algorithm that matches interest points with associated image descriptors between two images of the same scene.

Recommended reading

- Gonzalez & Woods: Chapters 11.6-11.7
- Szeliski: Chapters 6.1, 7.1