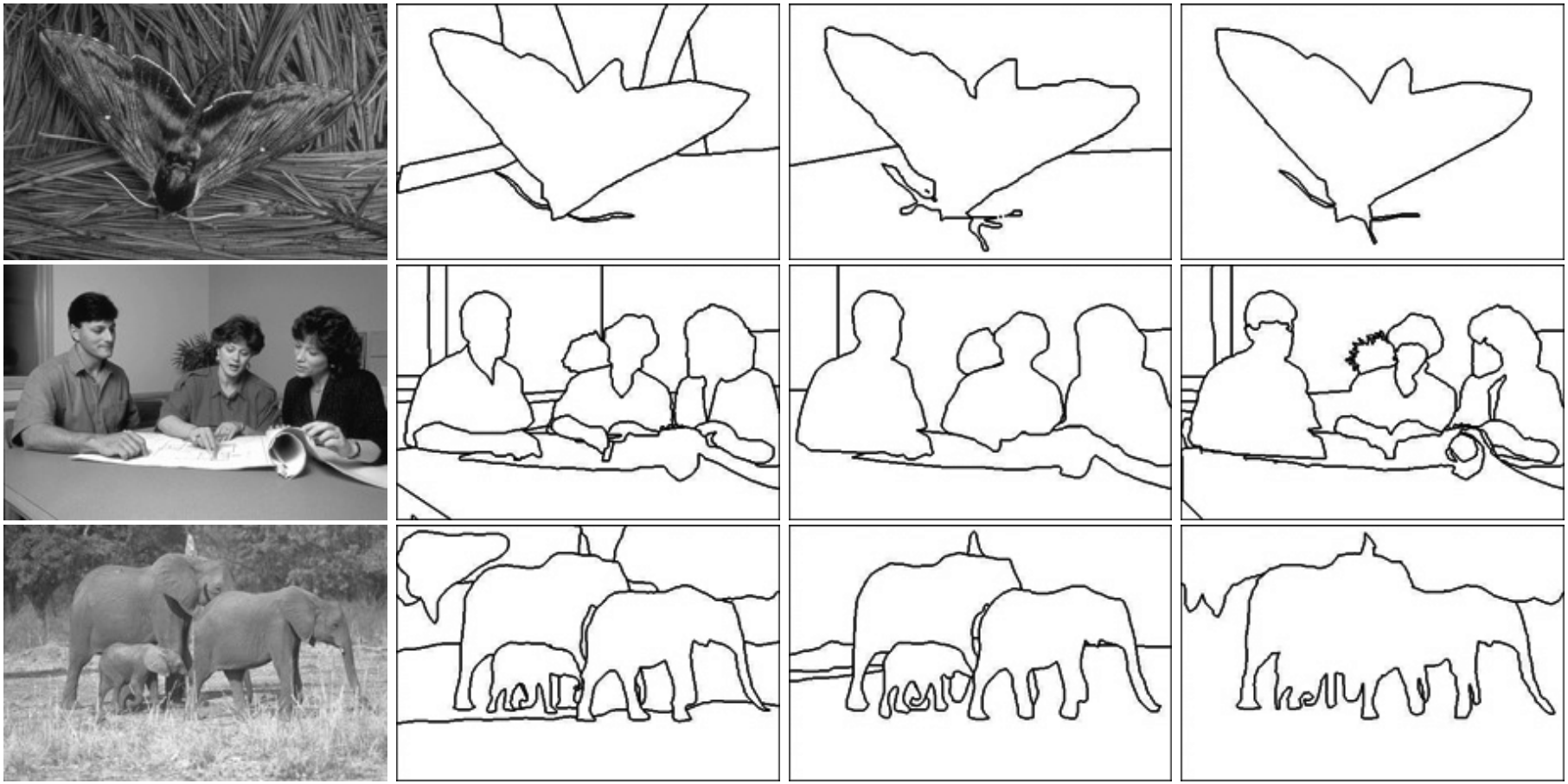
Computer vision description and embeddings

Ronan Sicre
Credits to Yannis Avrithis https://sif-dlv.github.io/

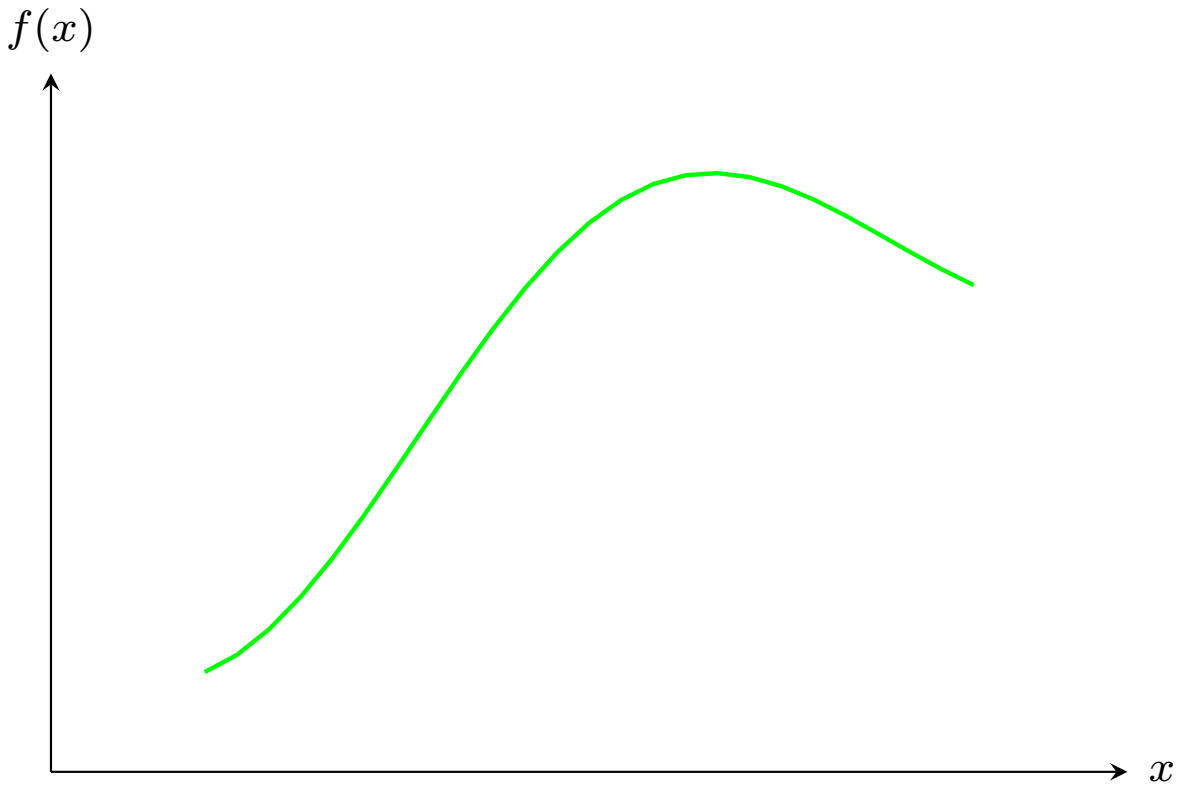
edges



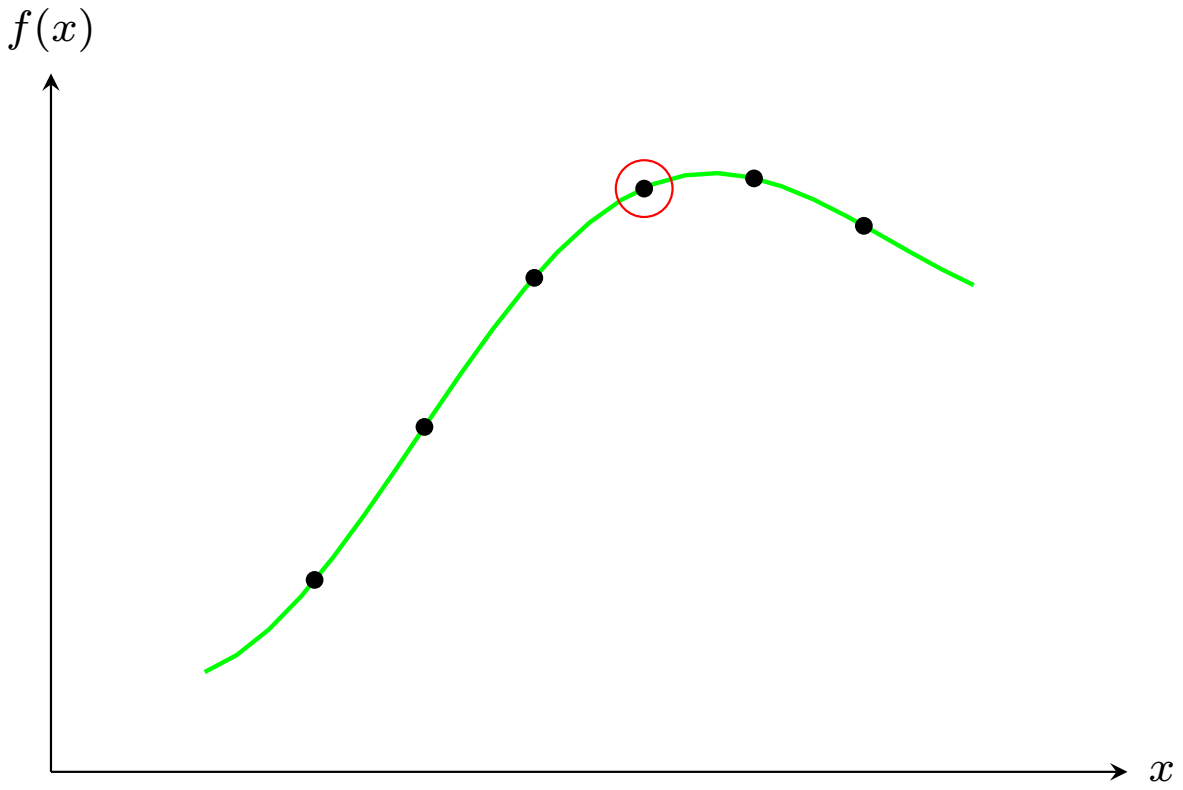
- connection between image recognition and segmentation
- database of human ‘ground truth’ to evaluate edge detection

Martin, Fowlkes, Tal, Malik. ICCV 2001. A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics.

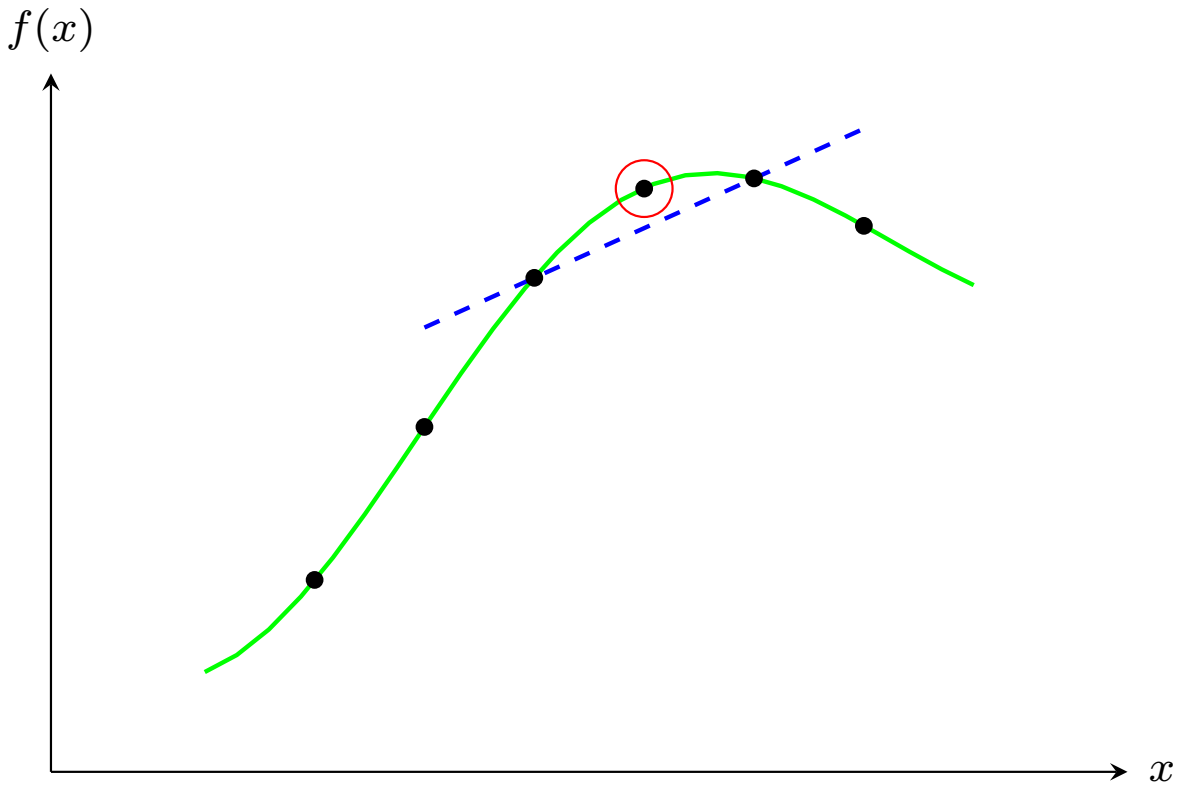
discrete derivative approximation



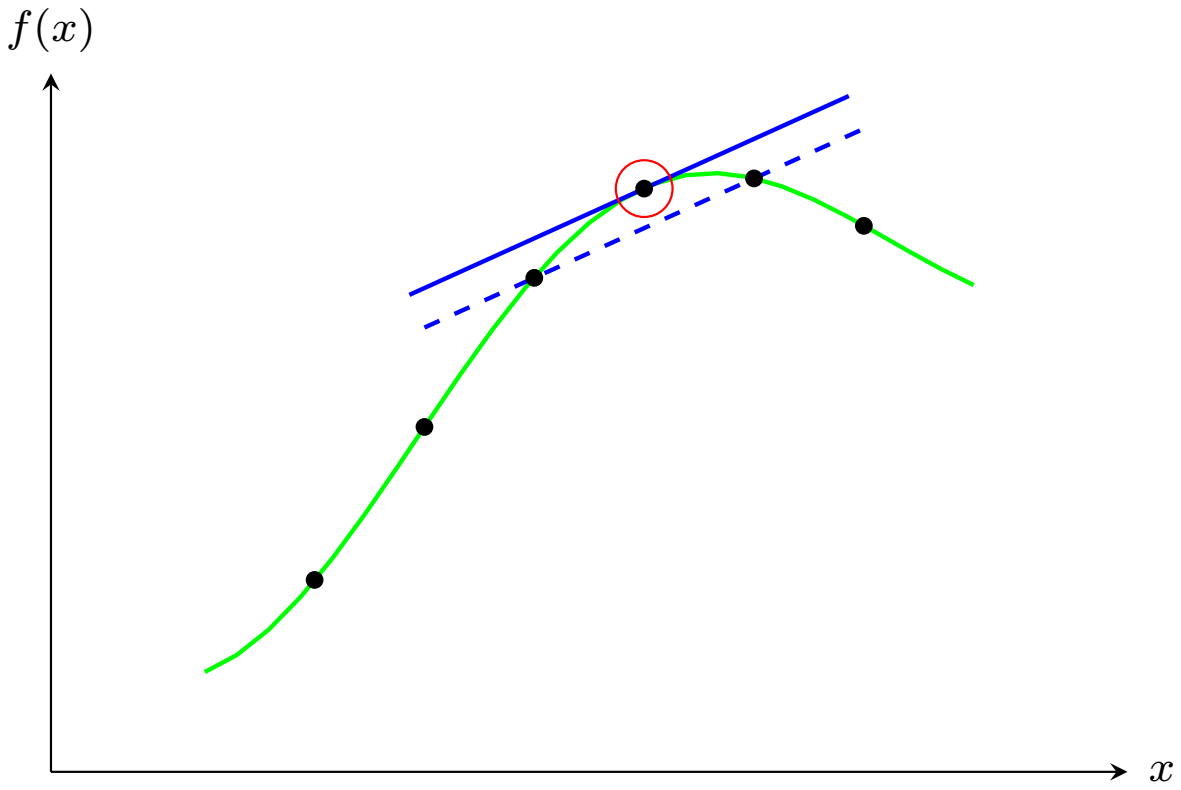
discrete derivative approximation



discrete derivative approximation

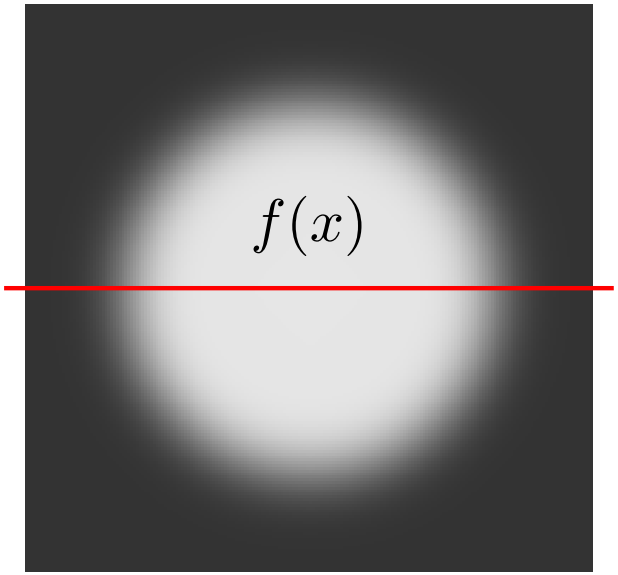


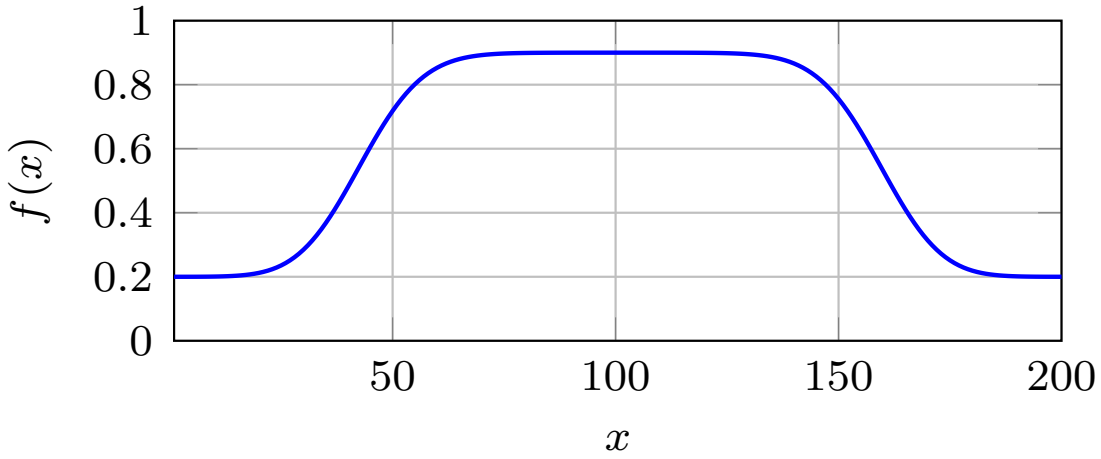
discrete derivative approximation



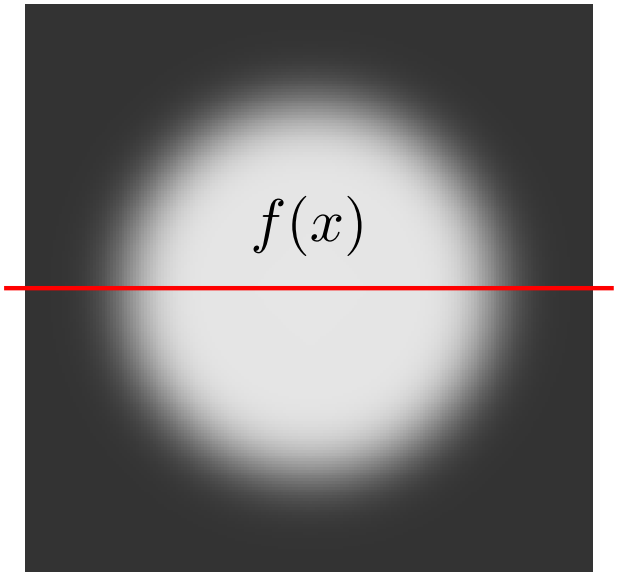
$$\frac{df}{dx}(x) \approx \frac{f(x+1) - f(x-1)}{2}$$

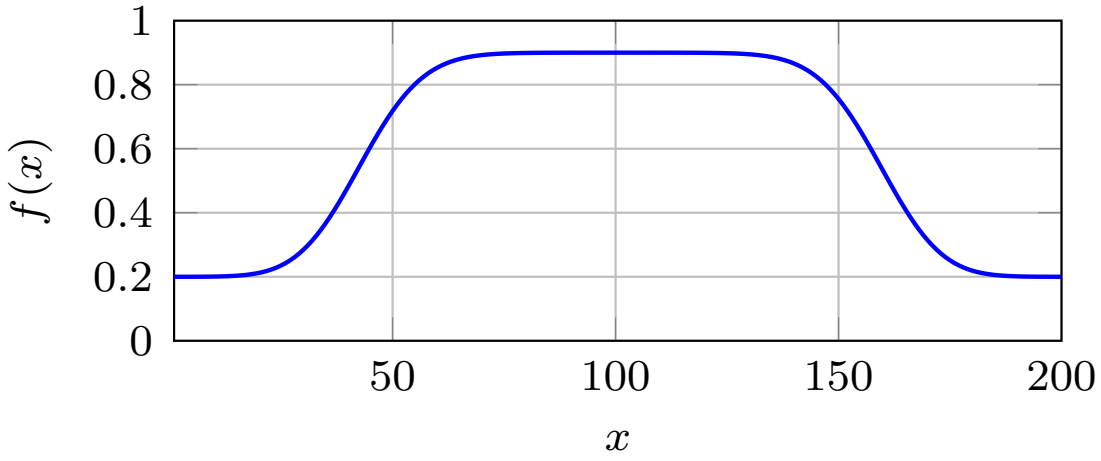
derivative in one dimension

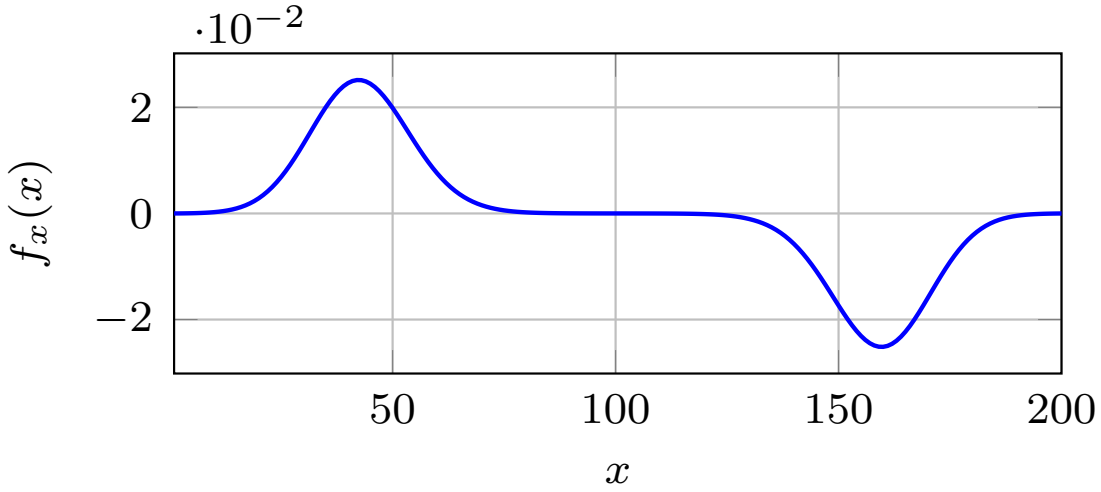




derivative in one dimension







$$f_x(x) := \frac{f(x+1) - f(x-1)}{2} = h * f, \quad h := \frac{1}{2}[1 \ 0 \ -1]$$

derivative in two dimensions: gradient

f

$f_x := h_x * f$

$h_x := \frac{1}{2}[1 \; 0 \; -1]$

$f_y := h_y * f$

$h_y := \frac{1}{2}[1 \; 0 \; -1]^\top$

derivative in two dimensions: gradient

f

$\|(f_x, f_y)\|$

$f_x := h_x * f$

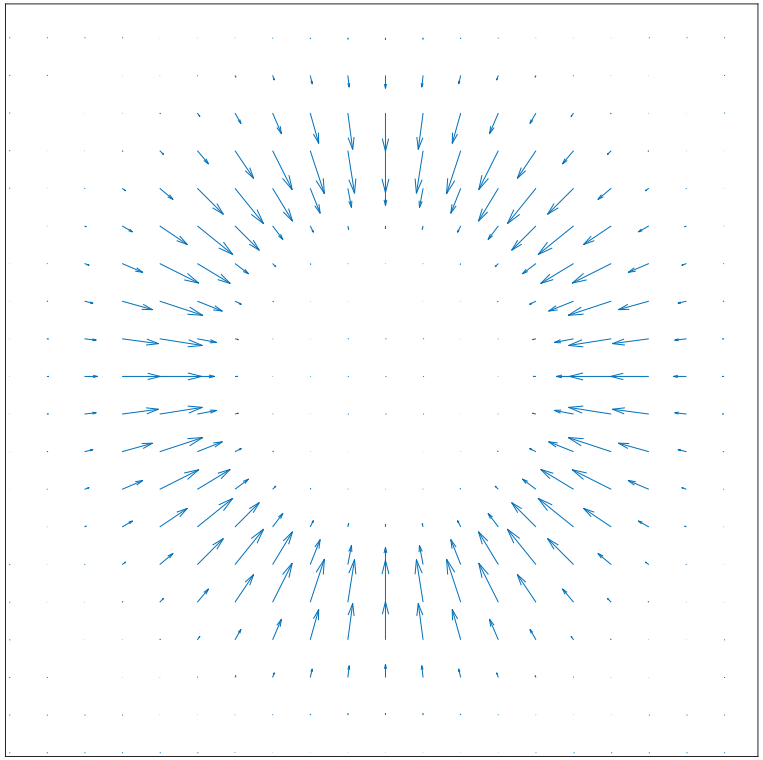
$h_x := \frac{1}{2}[1 \; 0 \; -1]$

$f_y := h_y * f$

$h_y := \frac{1}{2}[1 \; 0 \; -1]^\top$

gradient: magnitude and orientation

$\|(f_x, f_y)\|$

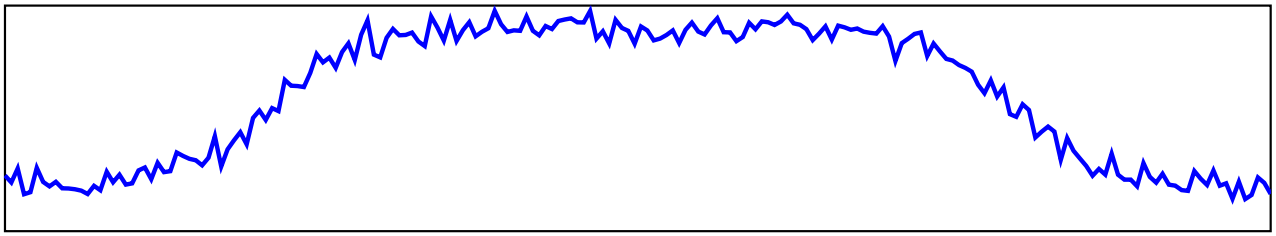


(f_x, f_y)

$$\nabla f(\mathbf{x}) := \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)(\mathbf{x}) \approx (h_x * f, h_y * f)(\mathbf{x}) = (f_x, f_y)(\mathbf{x})$$

noise

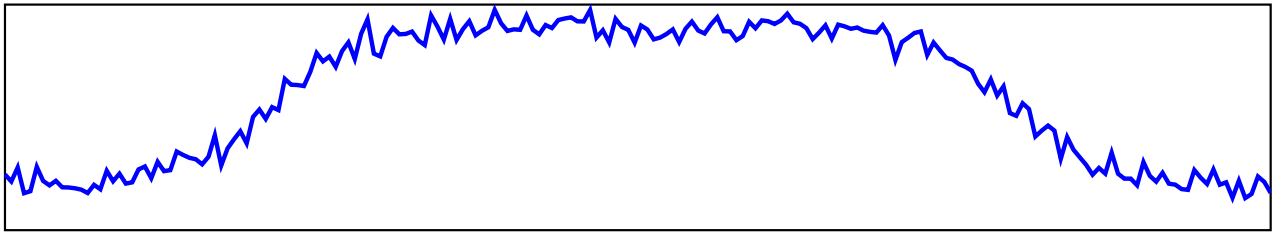
f



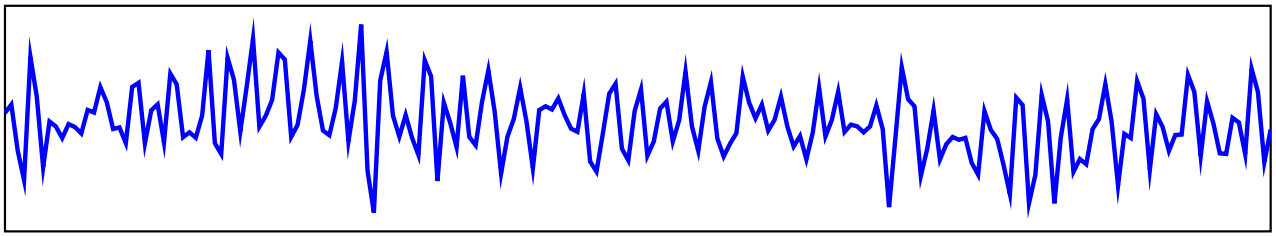
- Q: what happened to the edges?
- derivative is a high-pass filter: signal vanishes, noise remains

noise

f



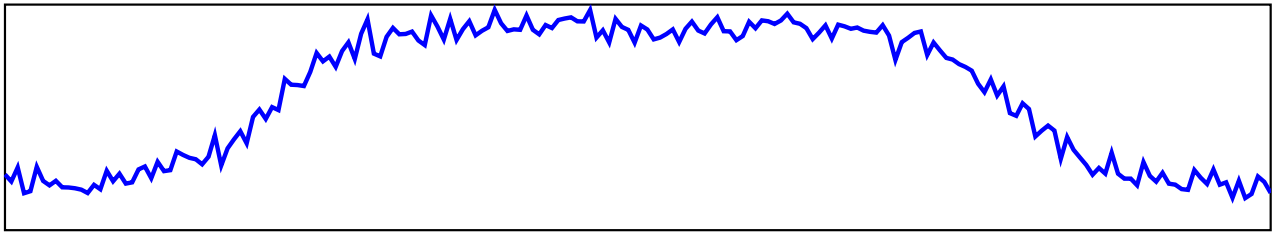
$\frac{d}{dx}(f)$



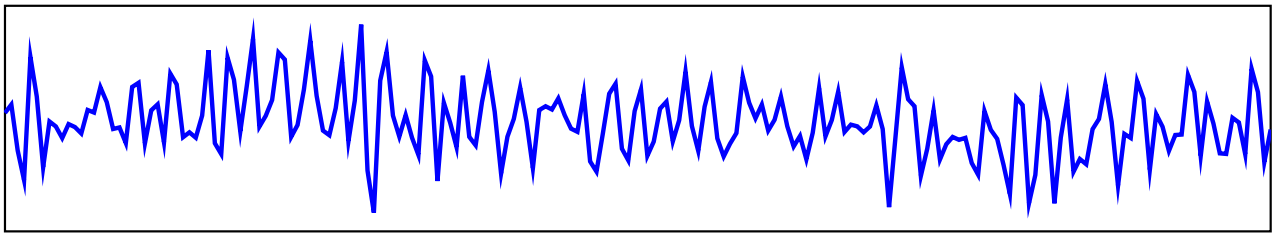
- Q: what happened to the edges?
- derivative is a high-pass filter: signal vanishes, noise remains

noise

f



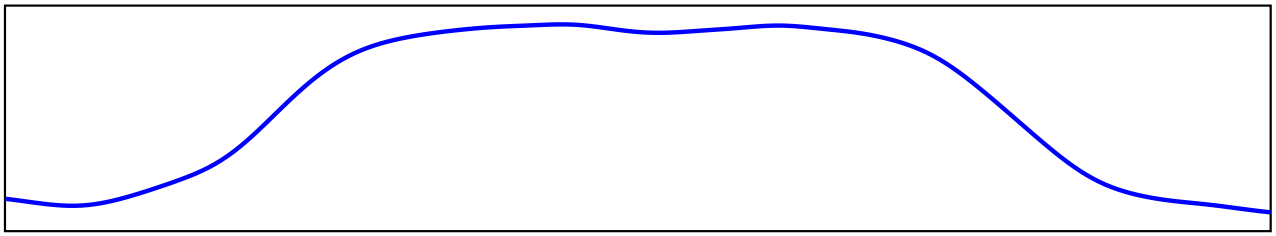
$\frac{d}{dx}(f)$



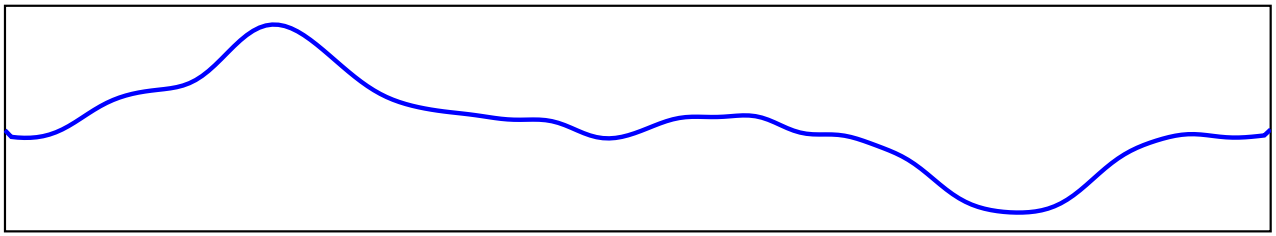
- Q: what happened to the edges?
- derivative is a high-pass filter: signal vanishes, noise remains

smoothing

$g * f$



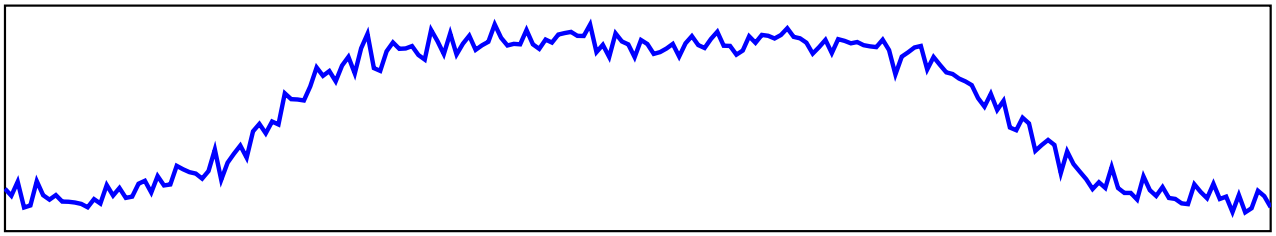
$\frac{d}{dx}(g * f)$



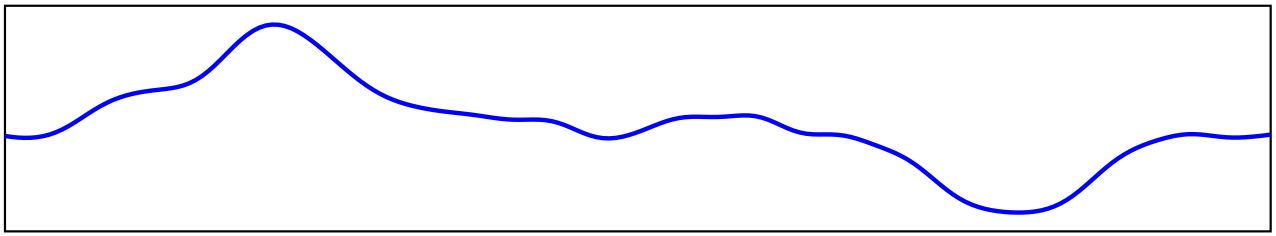
- smooth signal first
- that's better: edges recovered

filter derivative

f

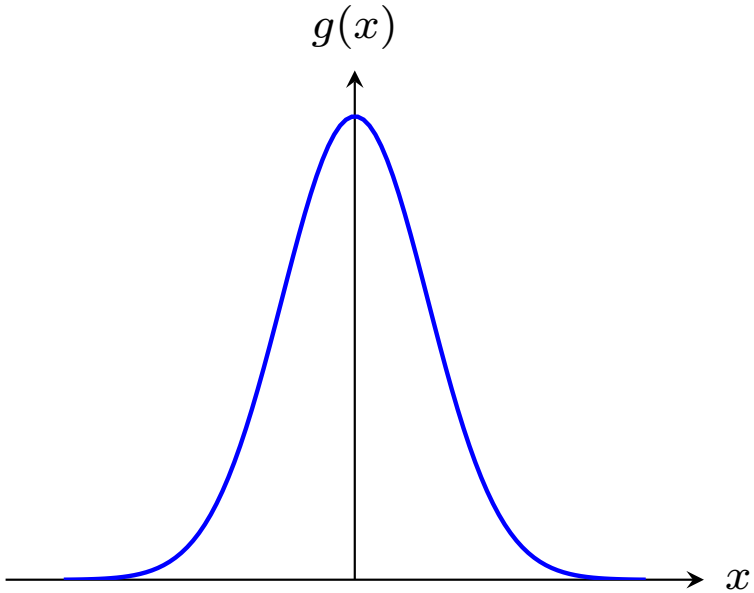


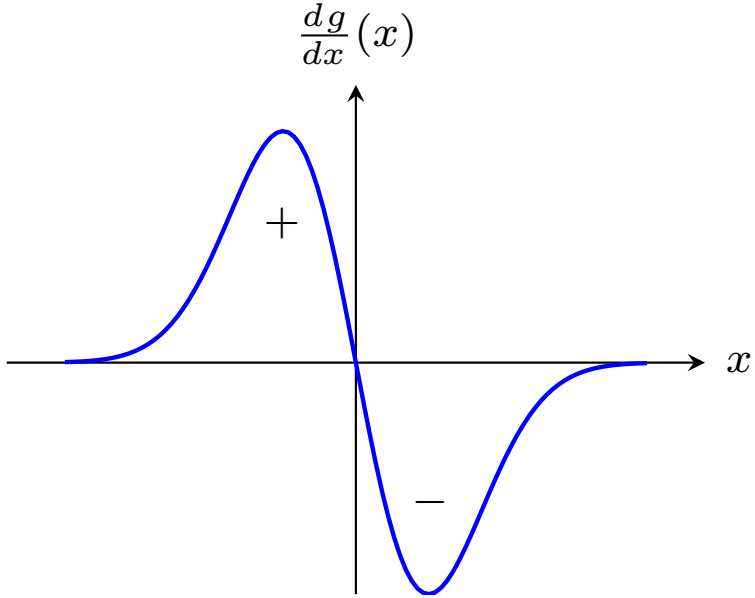
$\frac{d}{dx}(g) * f$



- this is equivalent to convolution with the filter derivative
- that's even better: filter is known in analytic form

1d Gaussian derivative



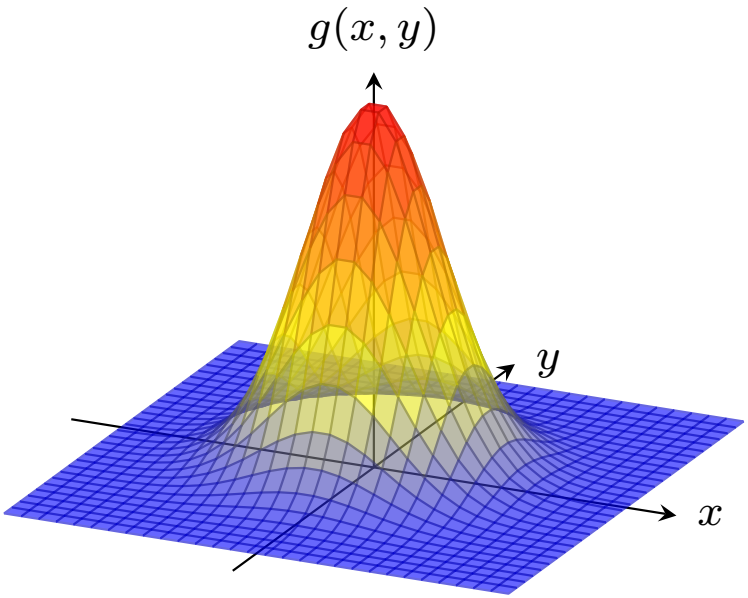


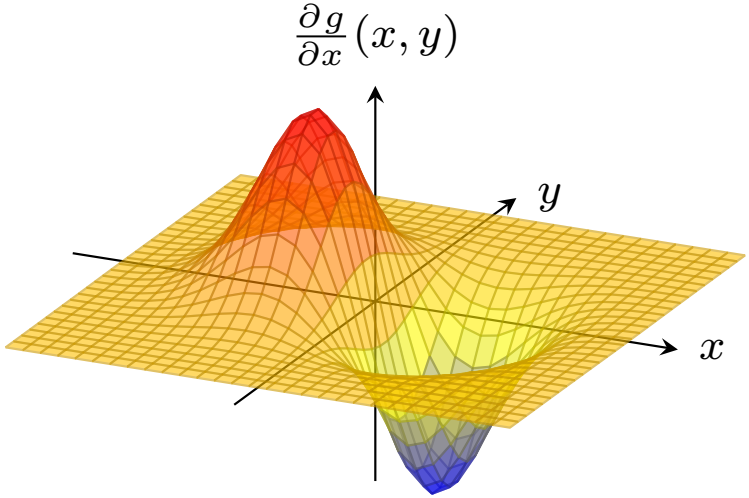
$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

$$\frac{dg}{dx}(x) = -\frac{x}{\sigma^2} g(x)$$

- performs derivation and smoothing at the same time
- σ : "derivation scale"

2d Gaussian derivative





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

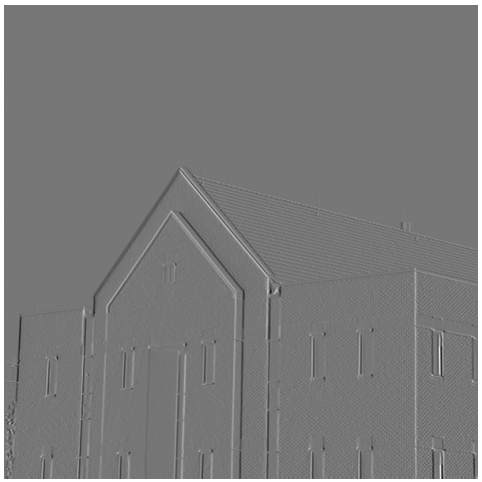
$$g_x(x,y) := \frac{\partial g}{\partial x}(x,y) = -\frac{x}{\sigma^2} g(x,y)$$

- derivation in one direction, smoothing in both
- "derivative = convolution"

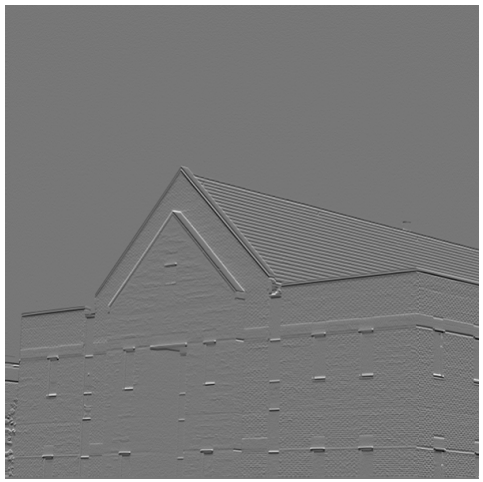
2d gradient

f

$\|(f_x, f_y)\|$



$f_x := h_x * f$



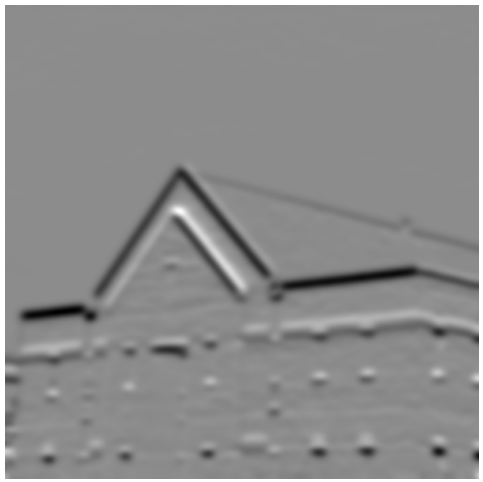
$f_y := h_y * f$

2d gradient by Gaussian derivative

f

$\|\nabla g * f\|$

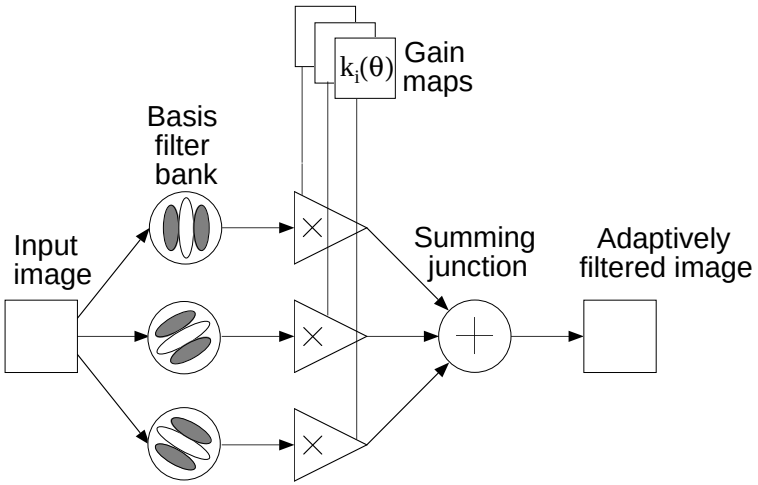
$g_x * f$



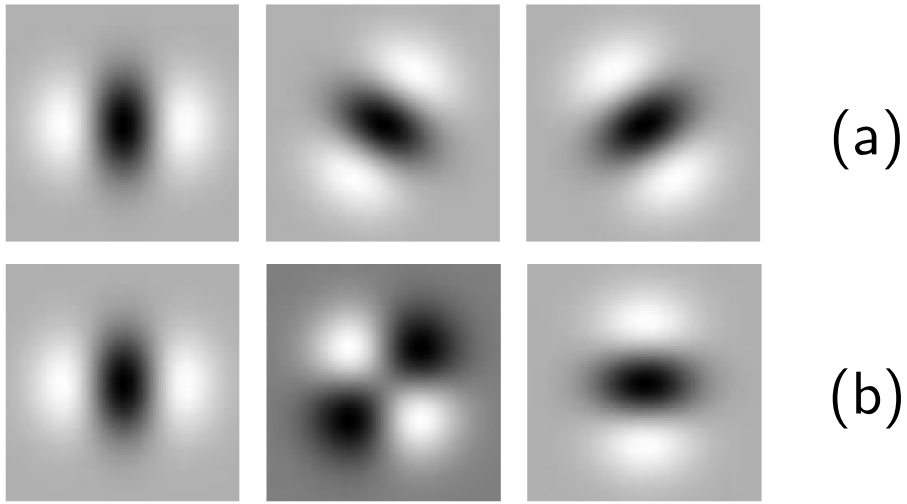
$g_y * f$

steerable filter

[Freeman and Adelson 1991]



architecture

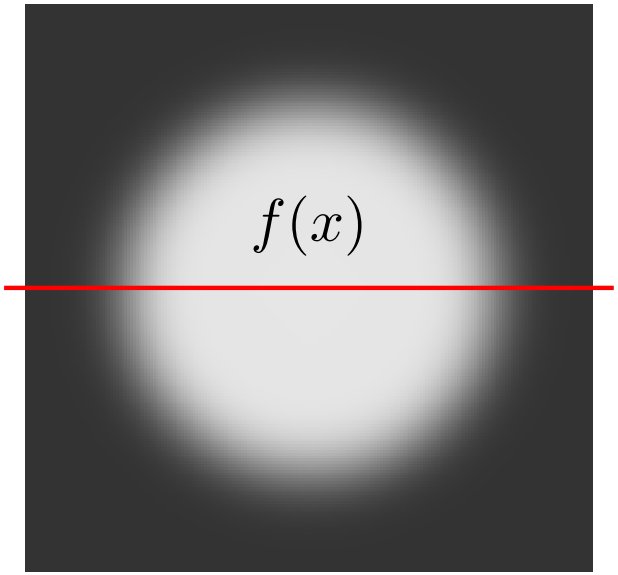


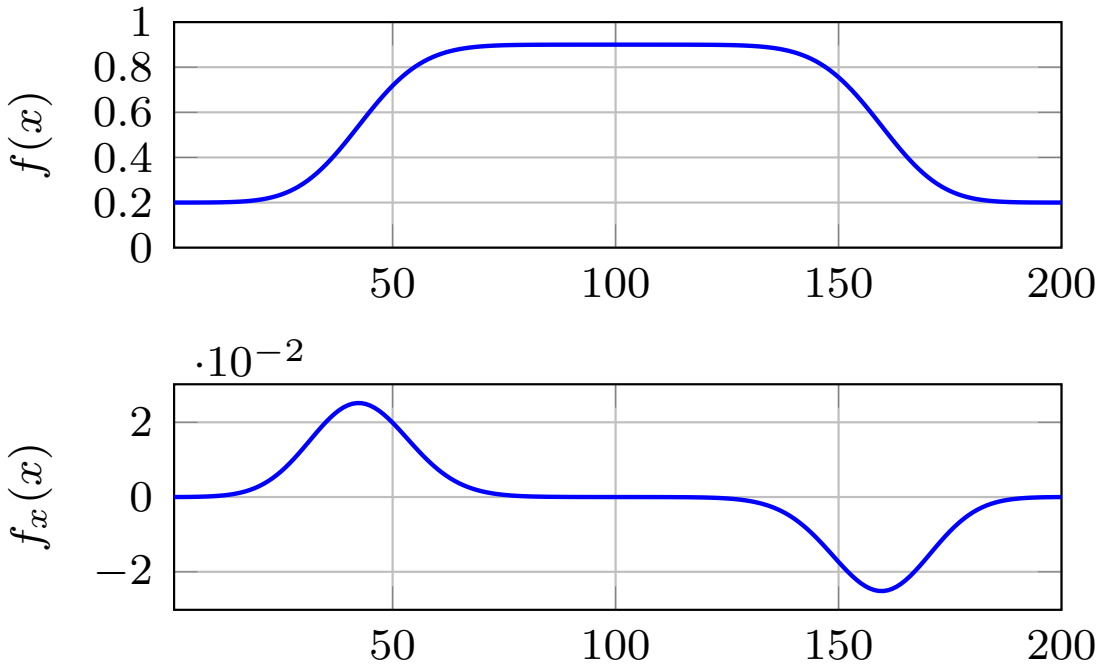
basis sets of $\frac{\partial^2 g}{\partial x^2}$

- an orientation-selective filter that can be expressed as a linear combination of a small basis set of filters
- the basis set can be (a) a set of rotated versions of itself, or (b) a set of separable filters

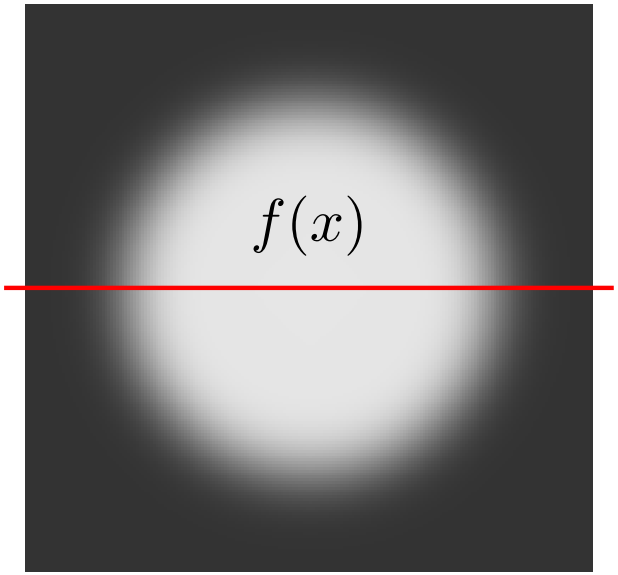
Freeman and Adelson. PAMI 1991. The Design and Use of Steerable Filters.

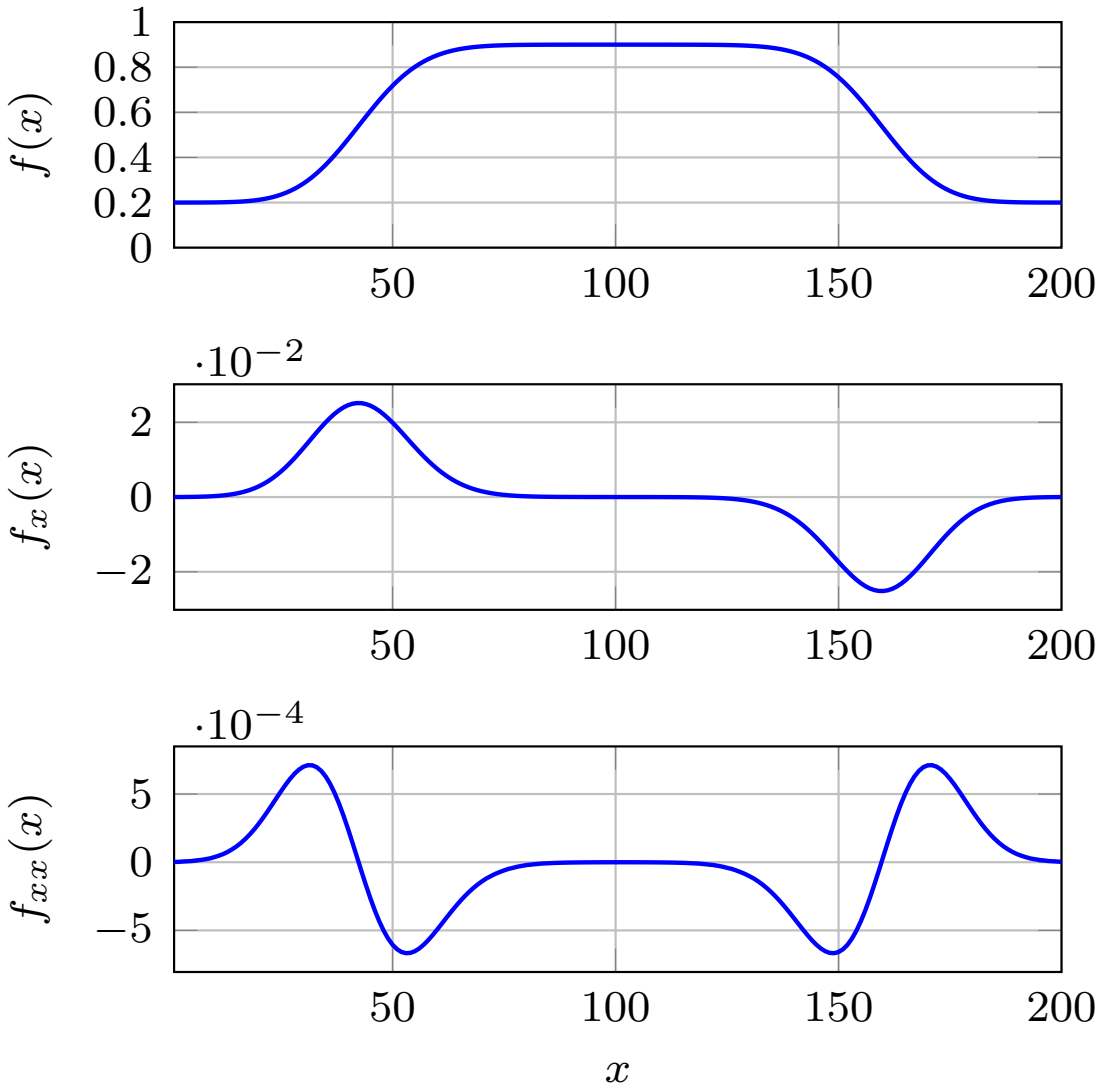
second derivative in one dimension





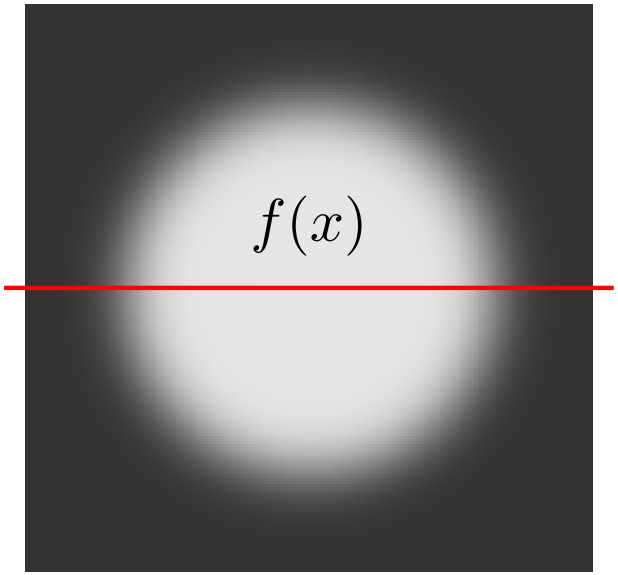
second derivative in one dimension

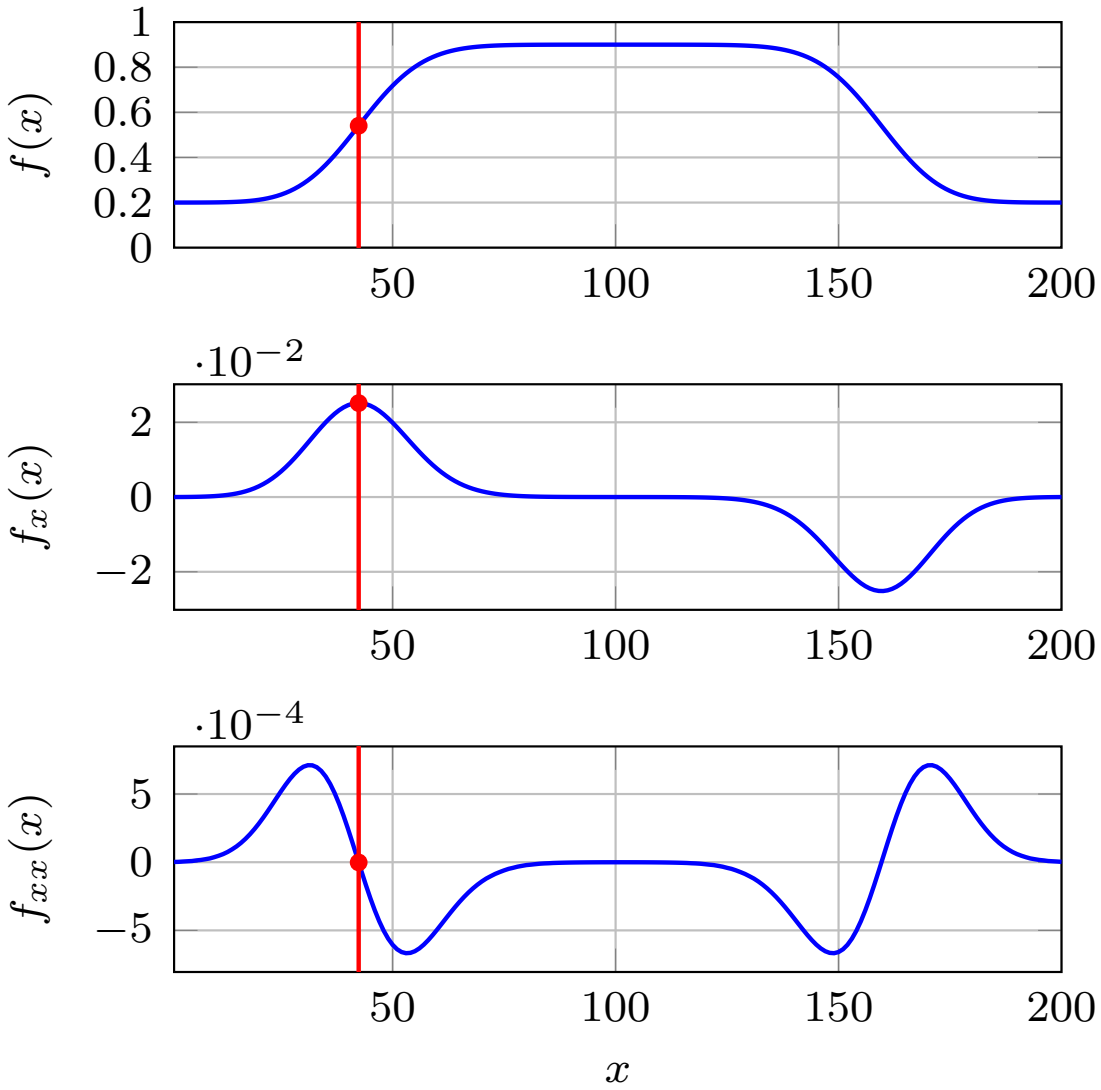




$$f_{xx}(x) := \frac{f(x-1) - 2f(x) + f(x+1)}{4} = h * f, \quad h := \frac{1}{4}[1 \;\; -2 \;\; 1]$$

second derivative in one dimension





$$f_{xx}(x) := \frac{f(x-1) - 2f(x) + f(x+1)}{4} = h * f, \quad h := \frac{1}{4}[1 \;\; -2 \;\; 1]$$

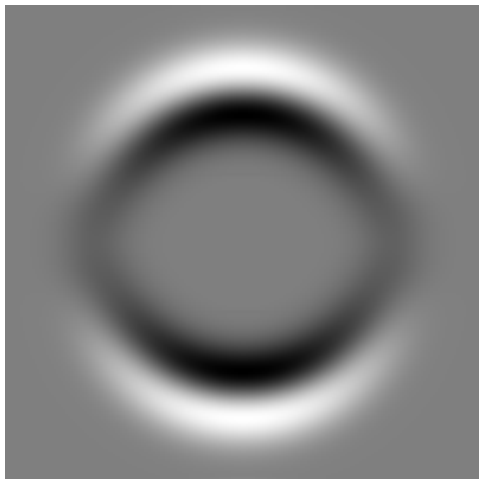
second derivative in two dimensions: Laplacian

f

$f_{xx} + f_{yy}$

$f_{xx} := h_{xx} * f$

$h_{xx} := \frac{1}{4}[1 \;\; -2 \;\; 1]$



$f_{yy} := h_{yy} * f$

$h_y := \frac{1}{4}[1 \;\; -2 \;\; 1]^\top$

Laplacian operator

- discrete approximation

$$h_{xx} := \frac{1}{4}[1 \;\; -2 \;\; 1]$$

$$h_{yy} := \frac{1}{4}[1 \;\; -2 \;\; 1]^\top$$

$$h_L := h_{xx} + h_{yy} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

- differential operator

$$\nabla^2 f(\mathbf{x}) := \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\right)(\mathbf{x})$$

$$\approx (h_{xx} * f + h_{yy} * f)(\mathbf{x}) = (f_{xx} + f_{yy})(\mathbf{x})$$

Laplacian operator

- discrete approximation

$$h_{xx} := \frac{1}{4}[1 \;\; -2 \;\; 1]$$

$$h_{yy} := \frac{1}{4}[1 \;\; -2 \;\; 1]^\top$$

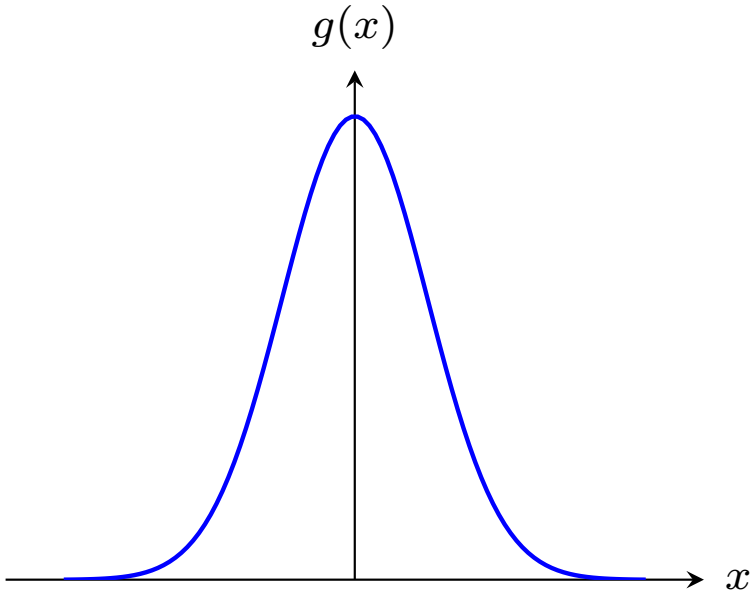
$$h_L := h_{xx} + h_{yy} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

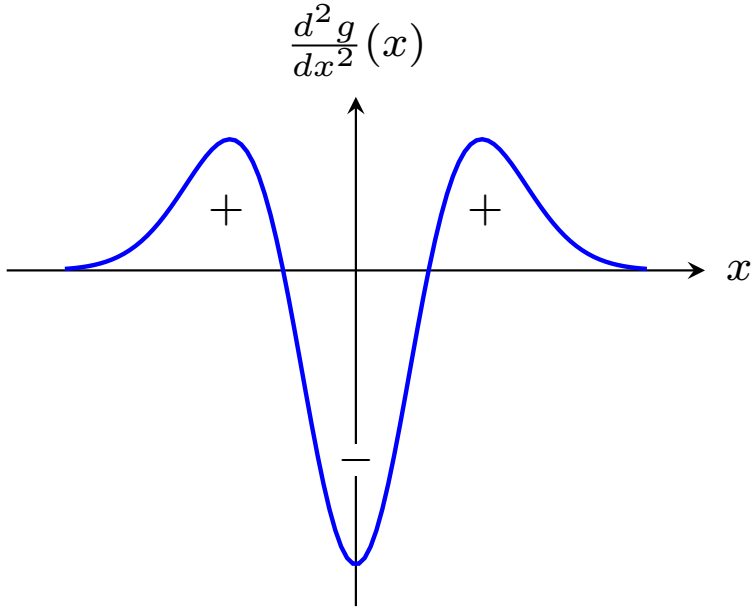
- differential operator

$$\nabla^2 f(\mathbf{x}) := \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right)(\mathbf{x})$$

$$\approx (h_{xx} * f + h_{yy} * f)(\mathbf{x}) = (f_{xx} + f_{yy})(\mathbf{x})$$

1d Gaussian second derivative



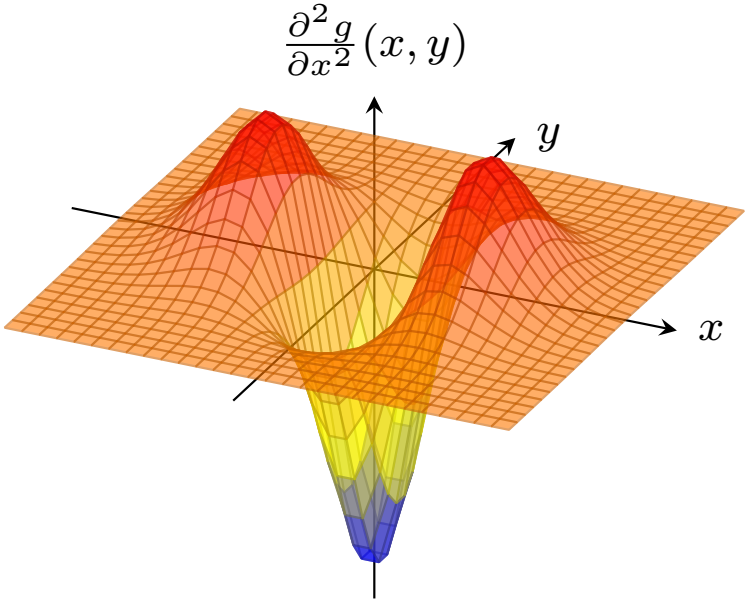


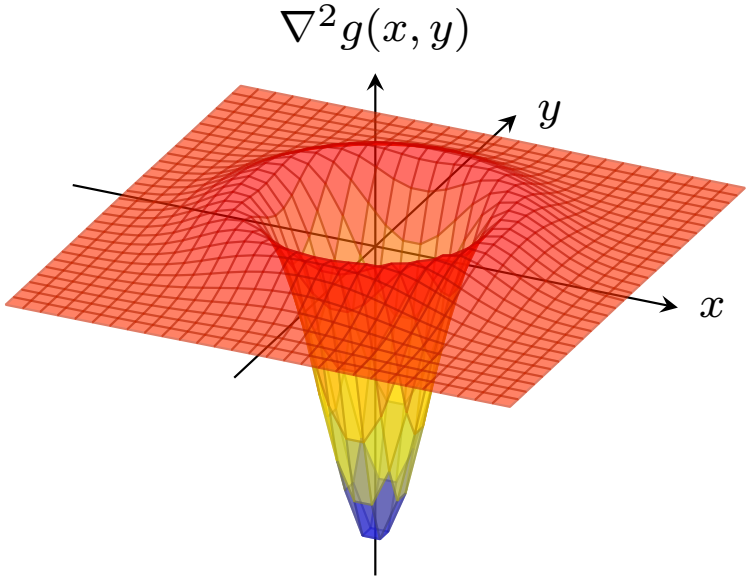
$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

$$\frac{d^2g}{dx^2}(x) = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right) g(x)$$

- "center-surround" operator

2d Laplacian of Gaussian (LoG)





$$\frac{\partial^2 g}{\partial x^2}(x,y) = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right) g(x,y)$$

$$\nabla^2 g(x,y) := \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}\right)(x,y)$$

- rotationally symmetric
- “mexican hat”

edge detection

f

$L_0(\nabla^2 g * f)$

$\|\nabla g * f\|$

$\nabla^2 g * f$

edge detection

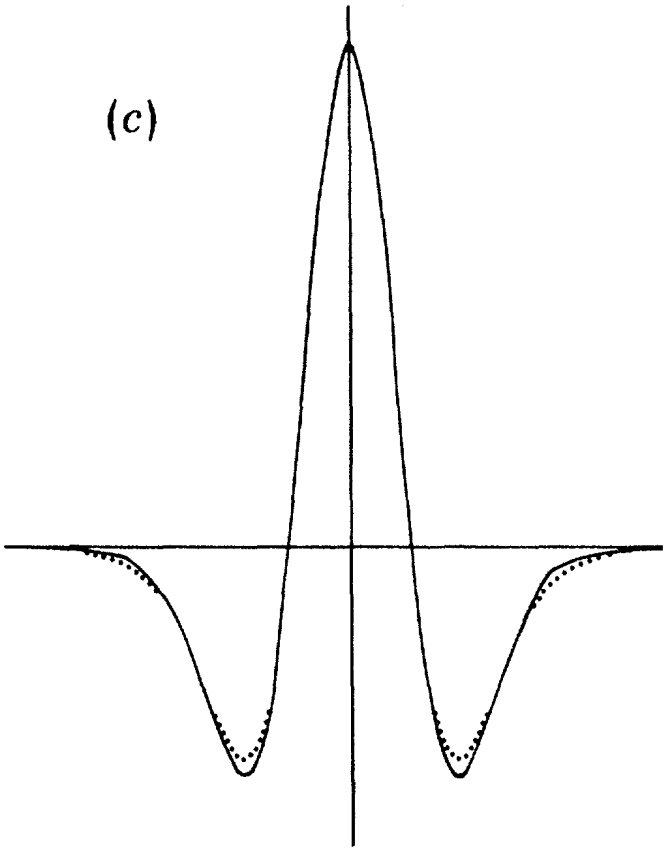
$L_0(\nabla^2 g * f)$

edge detection

$$L_0(\nabla^2 g * f)\|\nabla g * f\|$$

difference of Gaussians (DoG)

[Marr and Hildreth 1980]

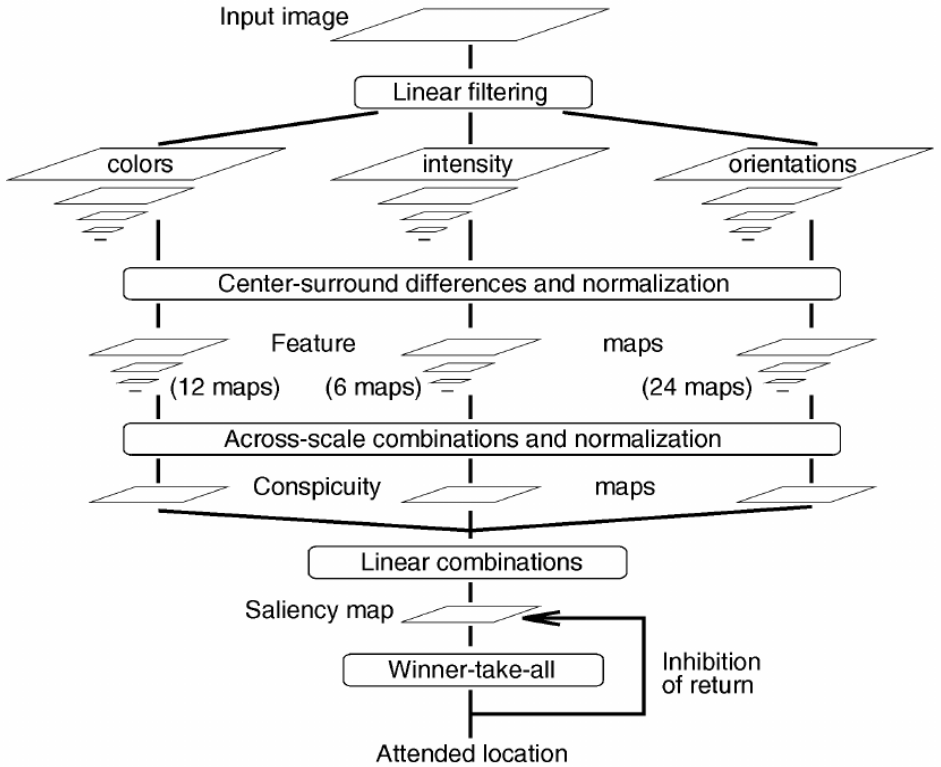


- studied the $\nabla^2 g$ operator as a model of retinal X-cells
- popularized it as a computational theory of edge detection
- hypothesized a biological implementation as a difference of Gaussians with $\sigma_1/\sigma_2 \approx 1.6$

feature detection

saliency and visual attention

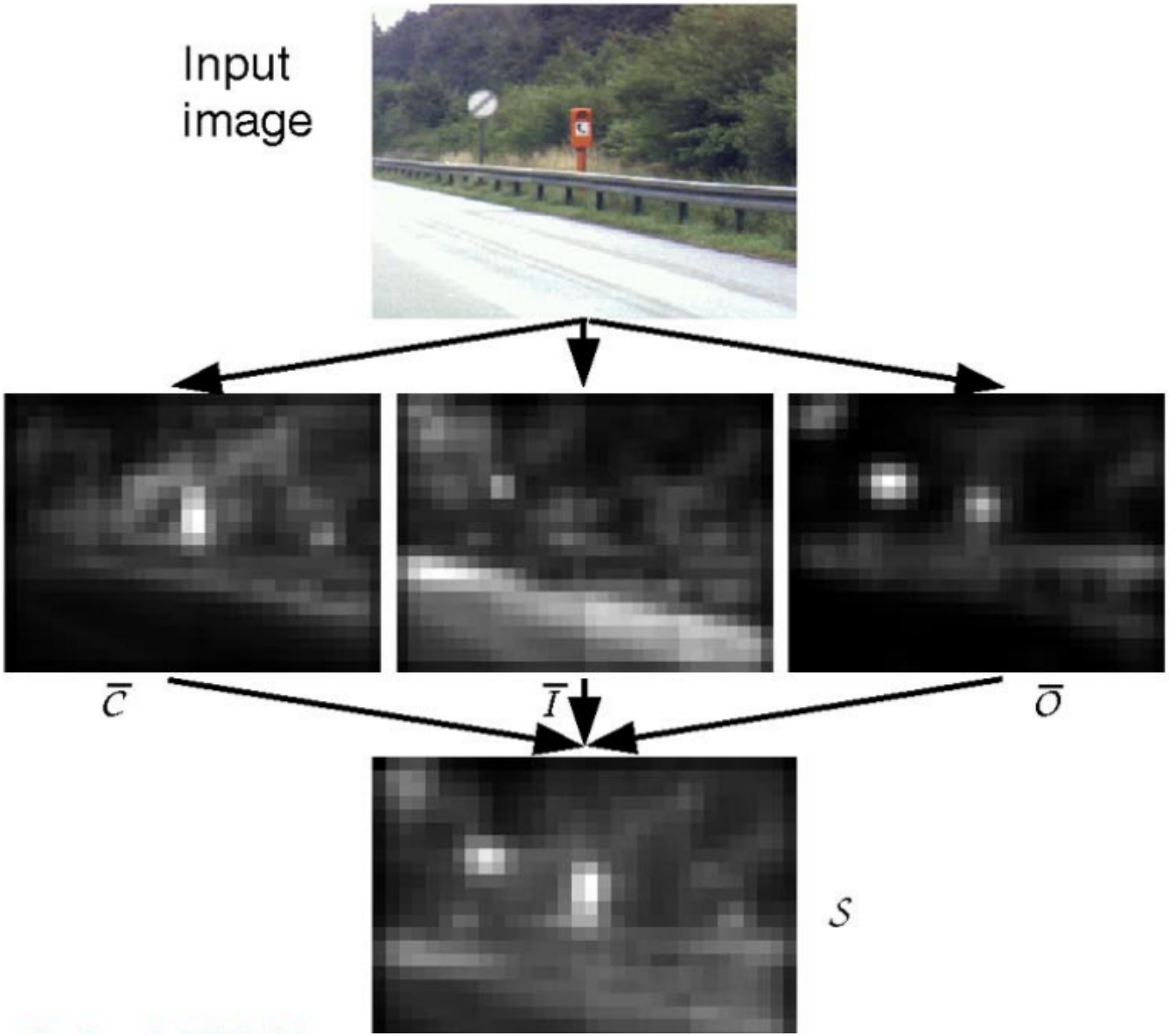
[Itti et al. 1998]



- visual attention system, inspired by the early primate visual system
- multiple scales, multiple features, center-surround, normalization and winner-take-all operations

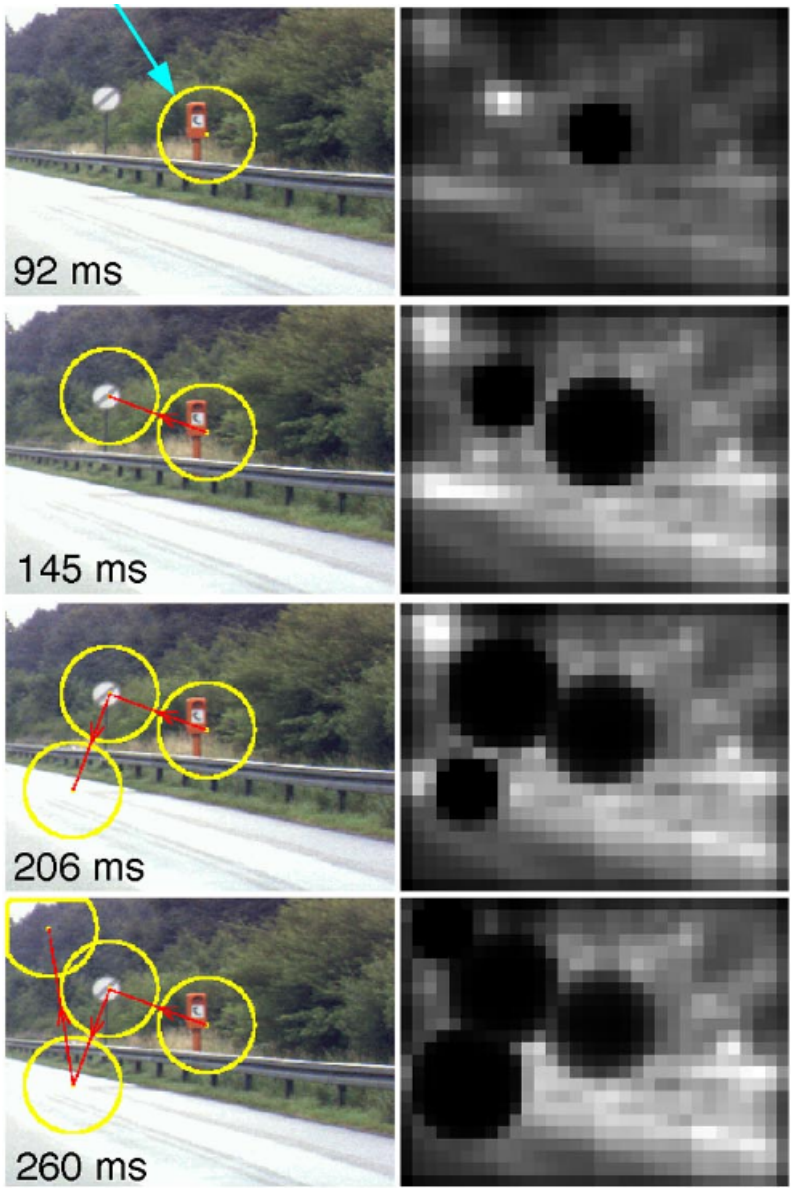
Itti, Koch and Niebur. PAMI 1998. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis.

saliency and visual attention



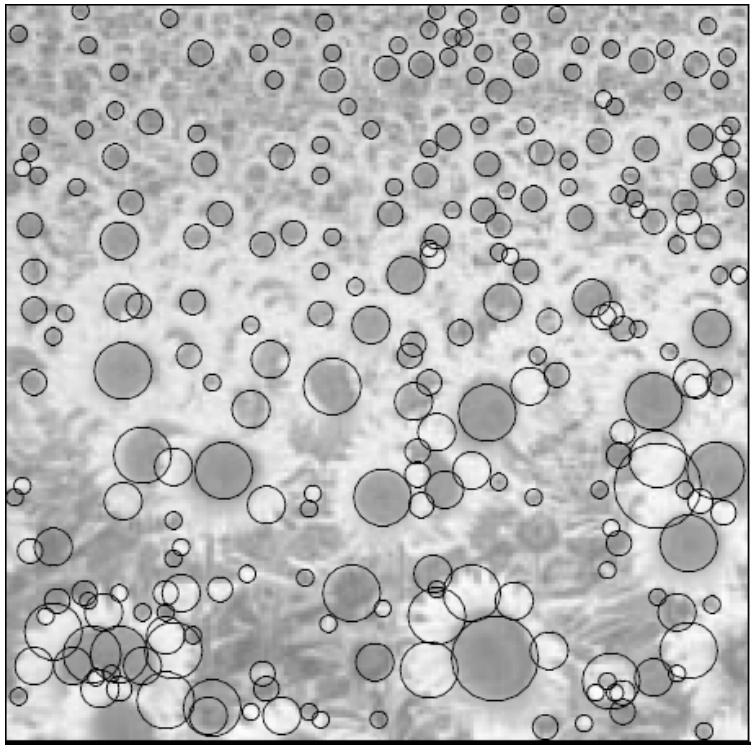
Itti, Koch and Niebur. PAMI 1998. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis.

saliency and visual attention



Itti, Koch and Niebur. PAMI 1998. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis.

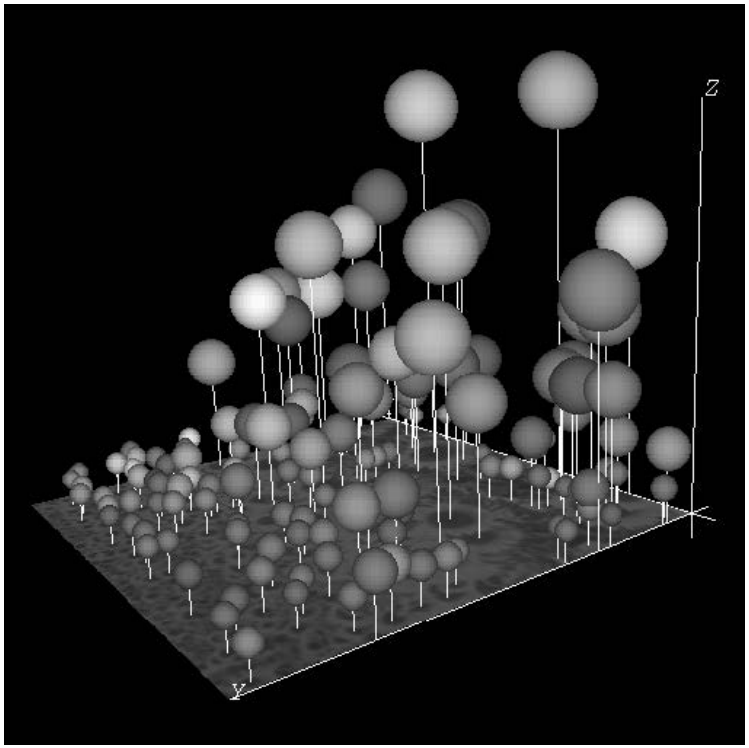
blob detection



- convolution with a circular symmetric center-surround pattern in scale-space
- local maxima in scale-space yield positions and scales of blobs

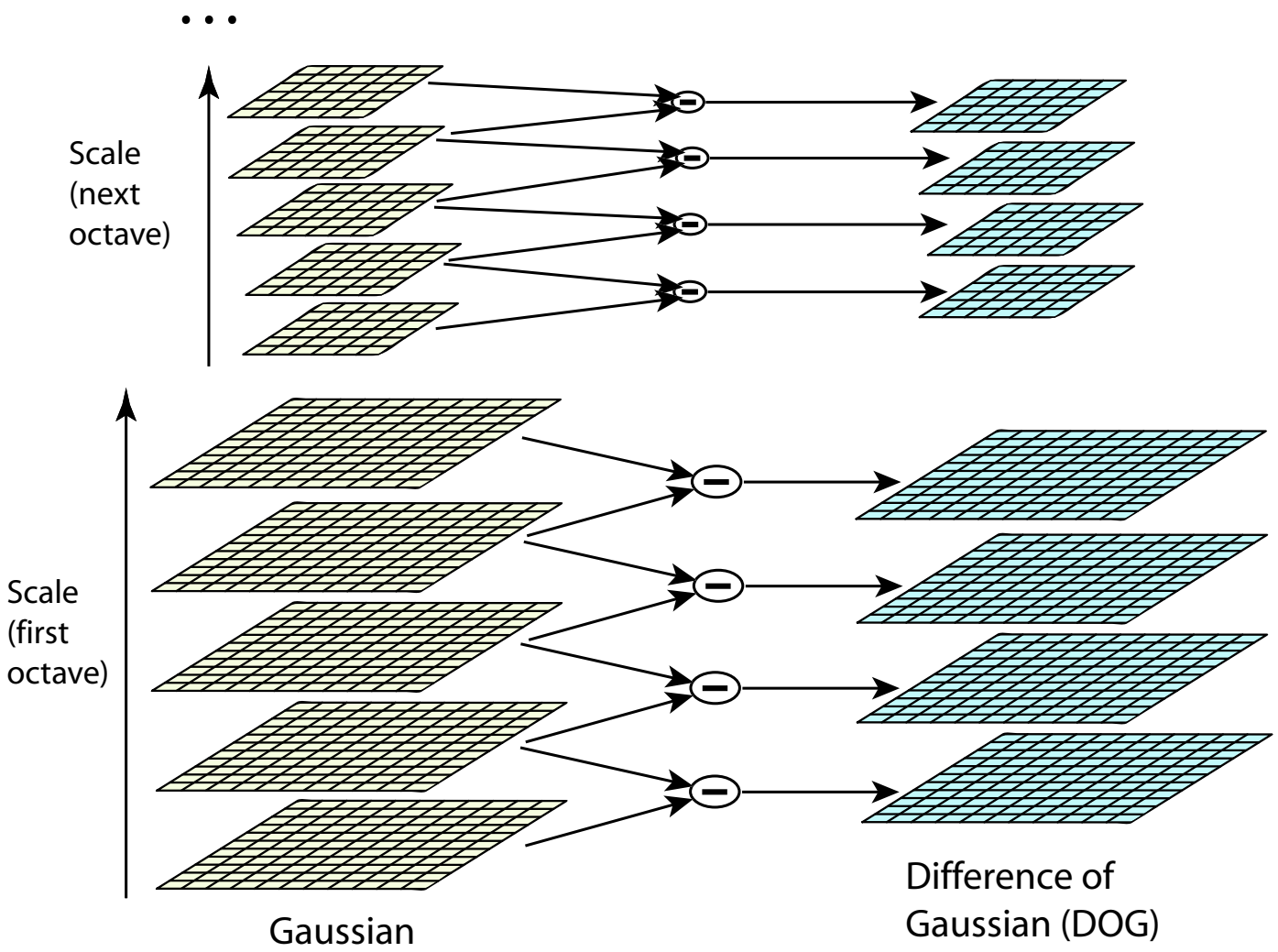
Lindeberg. IJCV 1998. Feature Detection with Automatic Scale Selection.

blob detection



- convolution with a circular symmetric center-surround pattern in scale-space
- local maxima in scale-space yield positions and scales of blobs

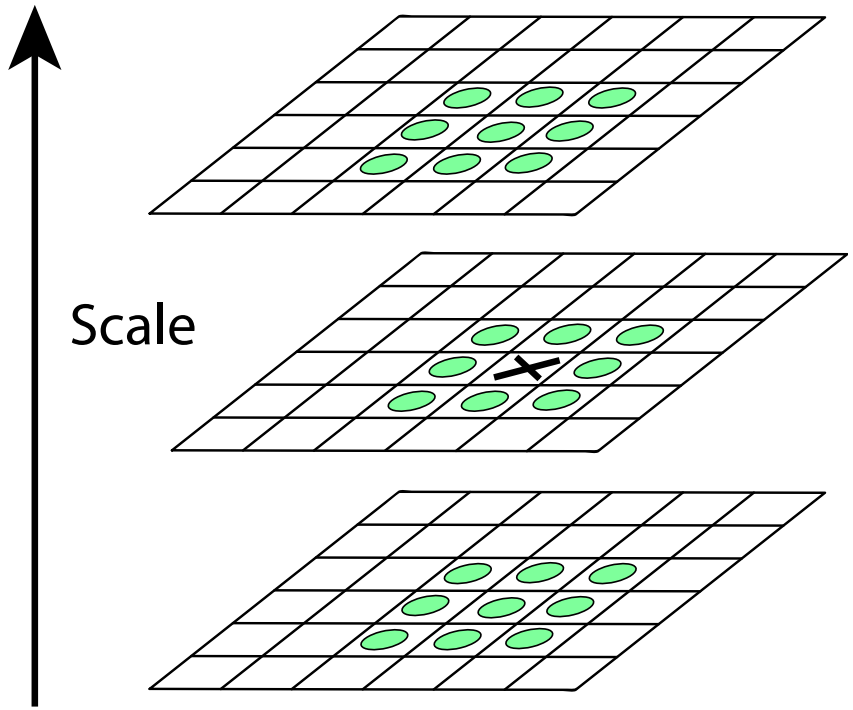
scale-space computation



- incrementally convolve with Gaussian, subsample at each octave

Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

scale-space local extrema

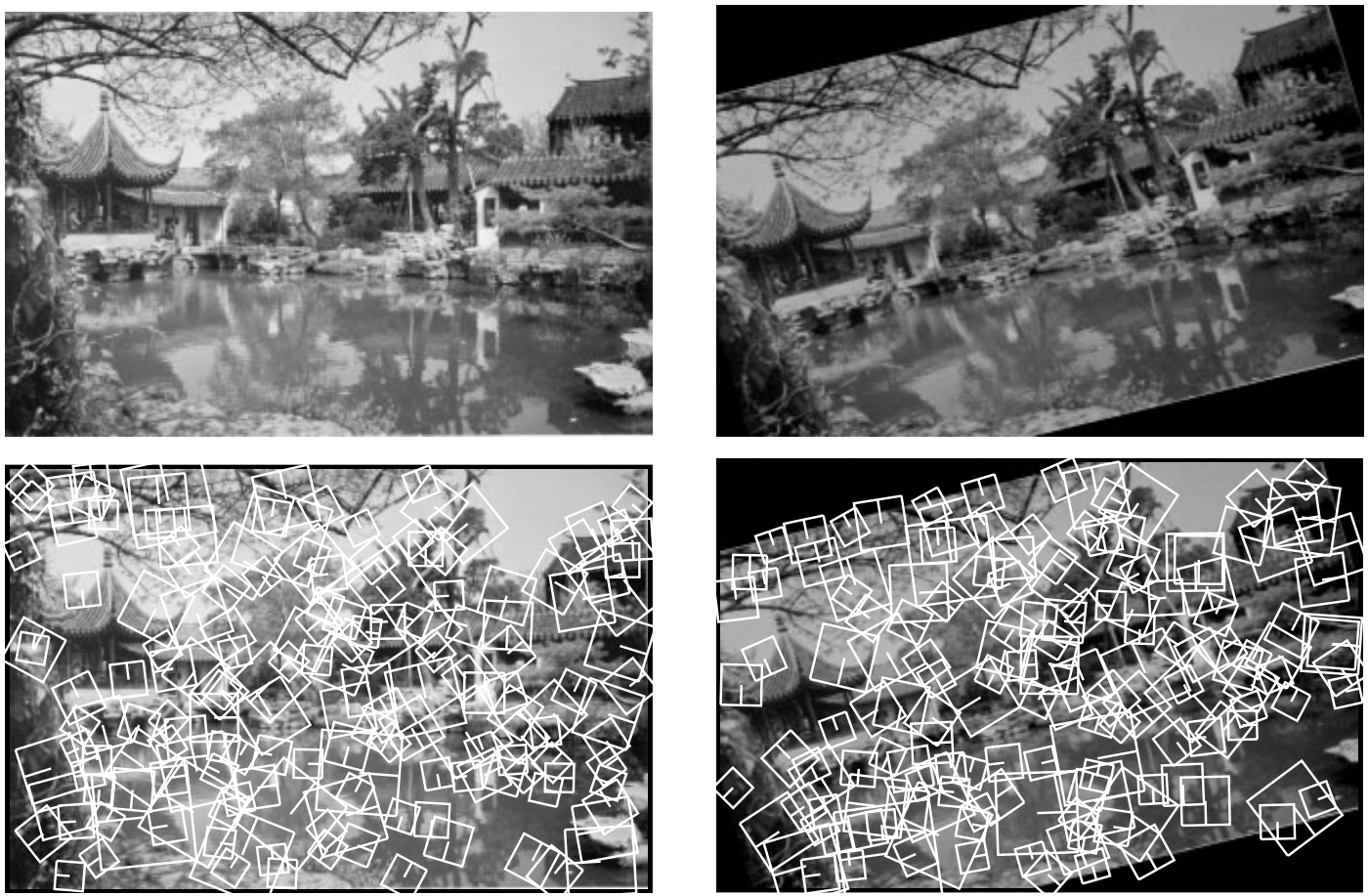


- local maxima among 26 neighbors selected
- accurately localized, edge responses rejected, orientation normalized

Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

scale-invariant feature transform (SIFT)

[Lowe 1999]



- detected patches equivariant to translation, scale and rotation

Lowe. ICCV 1999. Object recognition from local scale-invariant features.

desired properties of local features

- **repeatable**: in a transformed image, the same feature is detected at a transformed position
- **distinctive**: different image features can be discriminated by their local appearance
- **localized**: relatively small regions, robust to occlusion

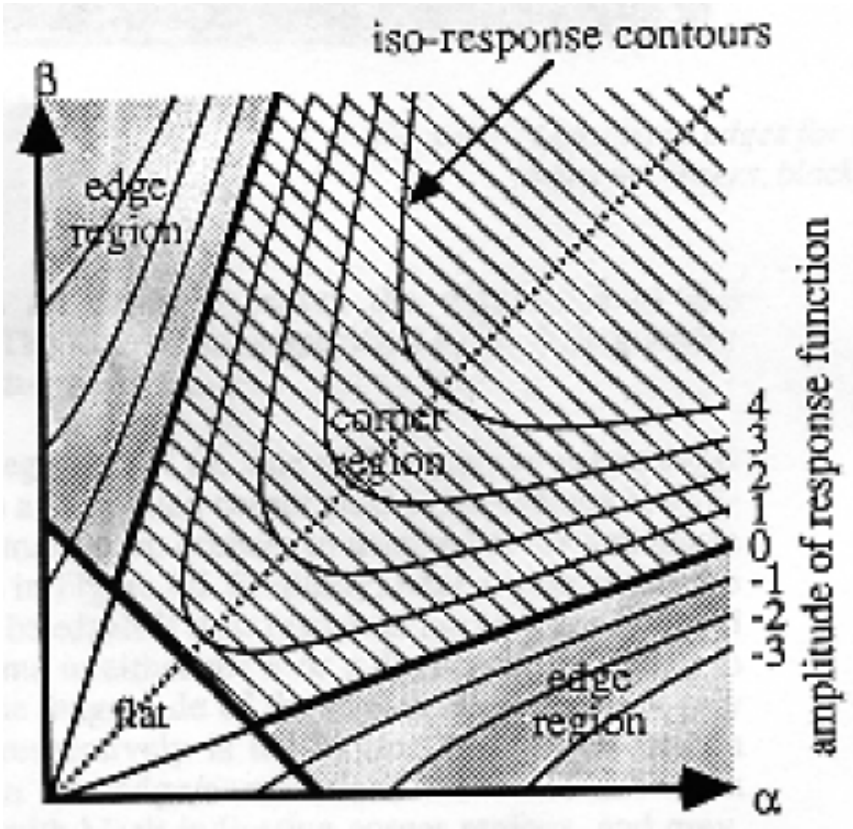
– *elongated*: edges, ridges

\+ *isotropic*: blobs, extremal regions

\+ *points*: corners and junctions

Harris corners

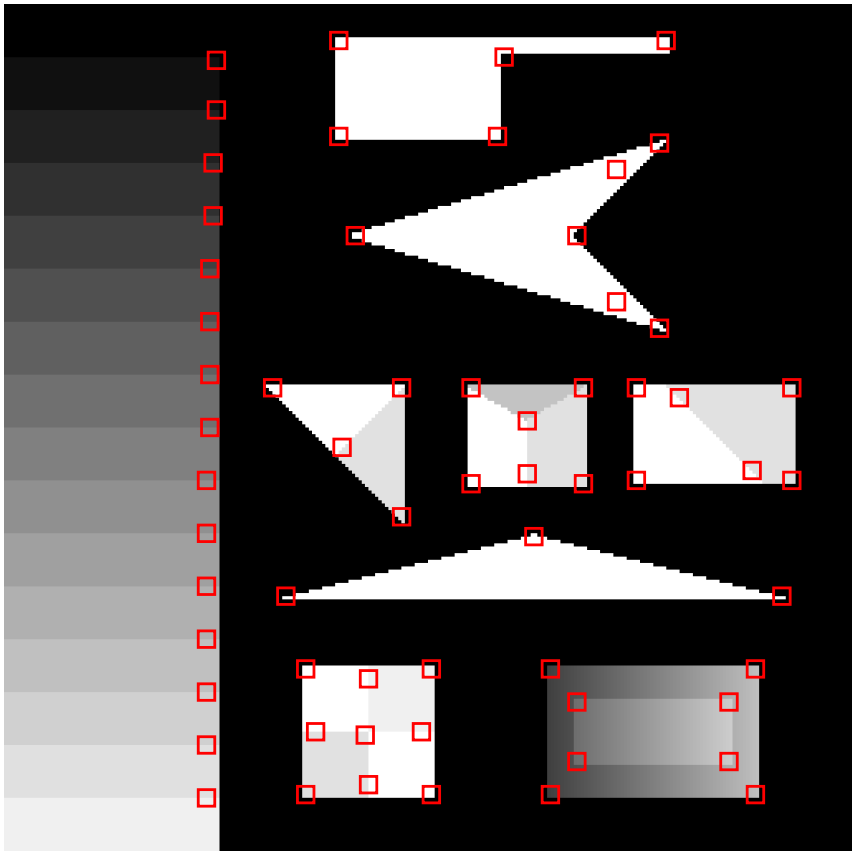
[Harris and Stevens 1988]



- if trace $\lambda_1 + \lambda_2$ is too low $\rightarrow$ flat
- if condition number λ_1/λ_2 is too high $\rightarrow$ edge
- response function $r(\mu) = \det \mu - k \operatorname{tr}^2 \mu$

Harris and Stephens AVC 1988. A Combined Corner and Edge Detector.

Harris corners (and junctions)



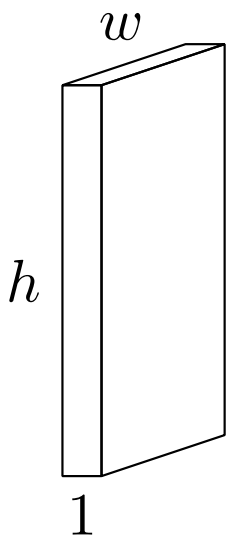
corners

response

- response: positive on corners, negative on edges, zero otherwise
- detection: non-maxima suppression and thresholding

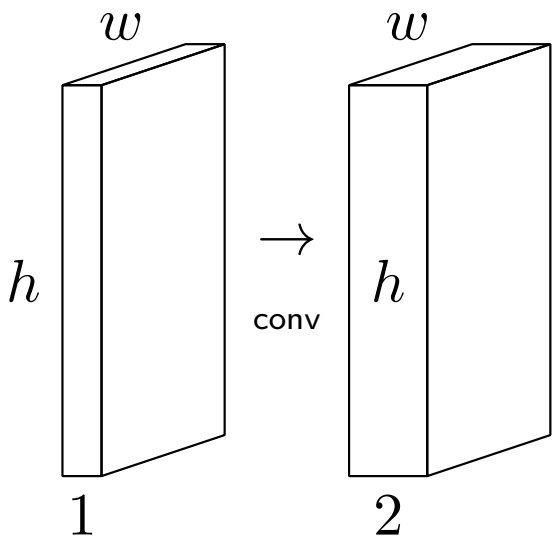
Harris and Stephens AVC 1988. A Combined Corner and Edge Detector.

Harris pipeline



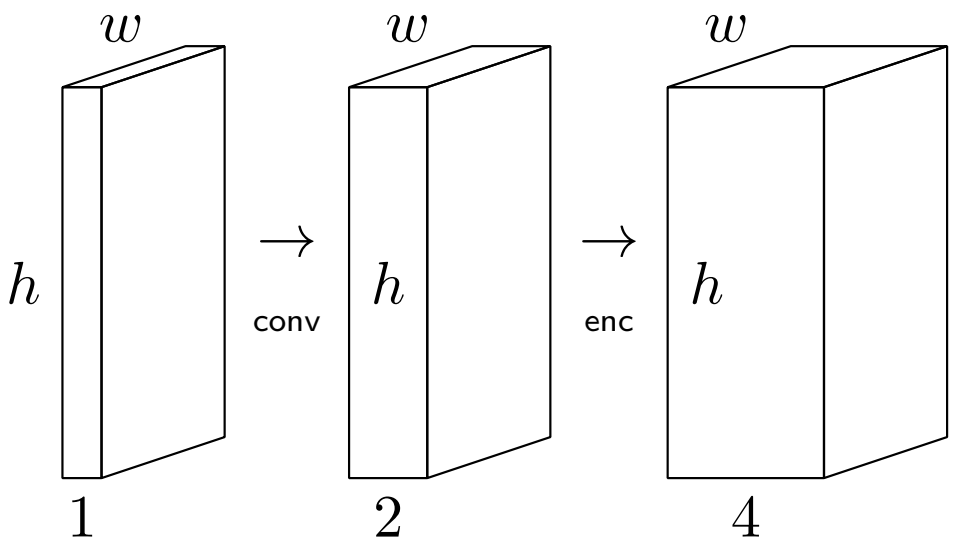
- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- compute gradient $\nabla F = (F_x, F_y)$ at derivation scale
- encode into tensor product $\nabla F \otimes \nabla F = (F_x^2, F_x F_y, F_x F_y, F_y^2)$
- average pooling by window w at integration scale
- compute point-wise nonlinear response function r

Harris pipeline



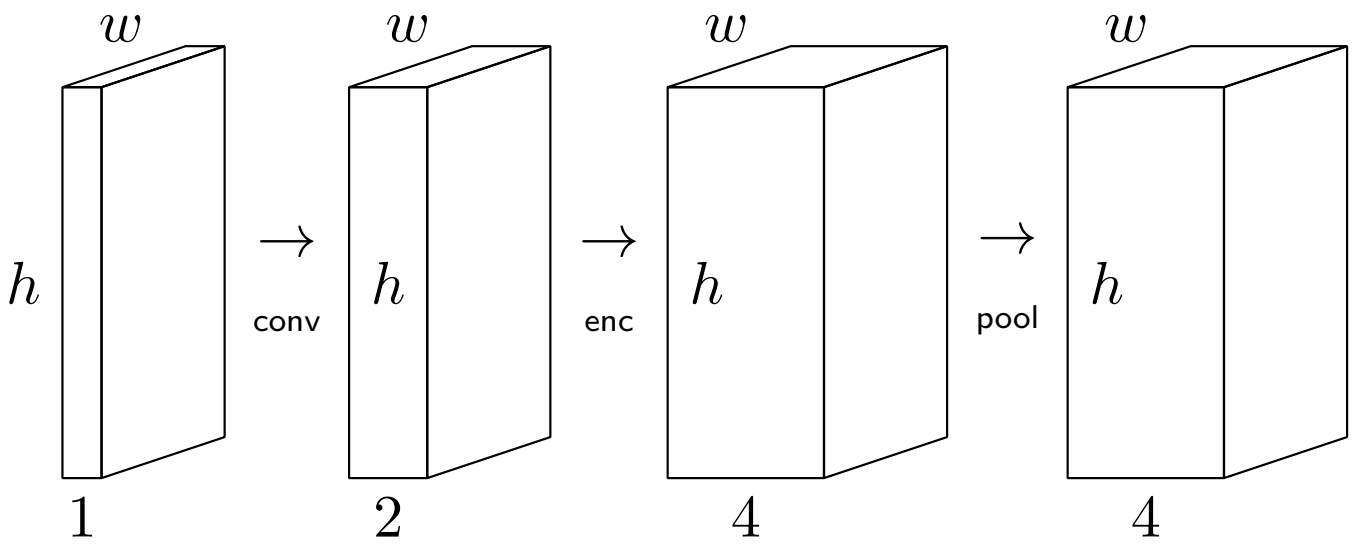
- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- compute gradient $\nabla F = (F_x, F_y)$ at derivation scale
- encode into tensor product $\nabla F \otimes \nabla F = (F_x^2, F_x F_y, F_x F_y, F_y^2)$
- average pooling by window w at integration scale
- compute point-wise nonlinear response function r

Harris pipeline



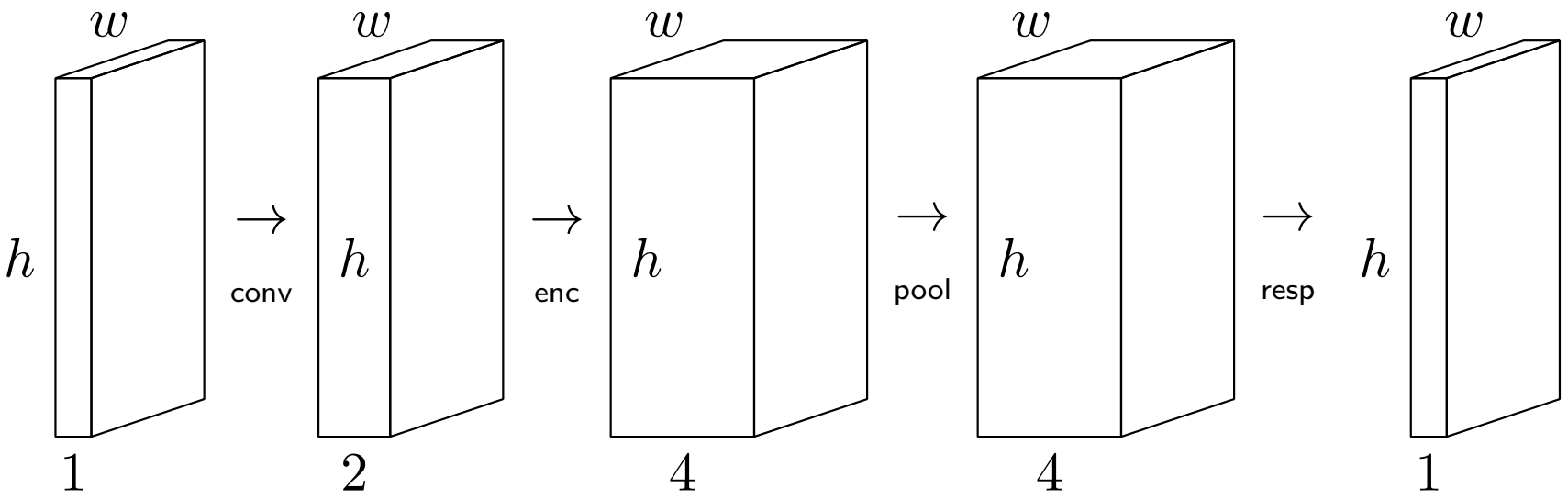
- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- compute gradient $\nabla F = (F_x, F_y)$ at derivation scale
- encode into tensor product $\nabla F \otimes \nabla F = (F_x^2, F_x F_y, F_x F_y, F_y^2)$
- average pooling by window w at integration scale
- compute point-wise nonlinear response function r

Harris pipeline



- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- compute gradient $\nabla F = (F_x, F_y)$ at derivation scale
- encode into tensor product $\nabla F \otimes \nabla F = (F_x^2, F_x F_y, F_x F_y, F_y^2)$
- average pooling by window w at integration scale
- compute point-wise nonlinear response function r

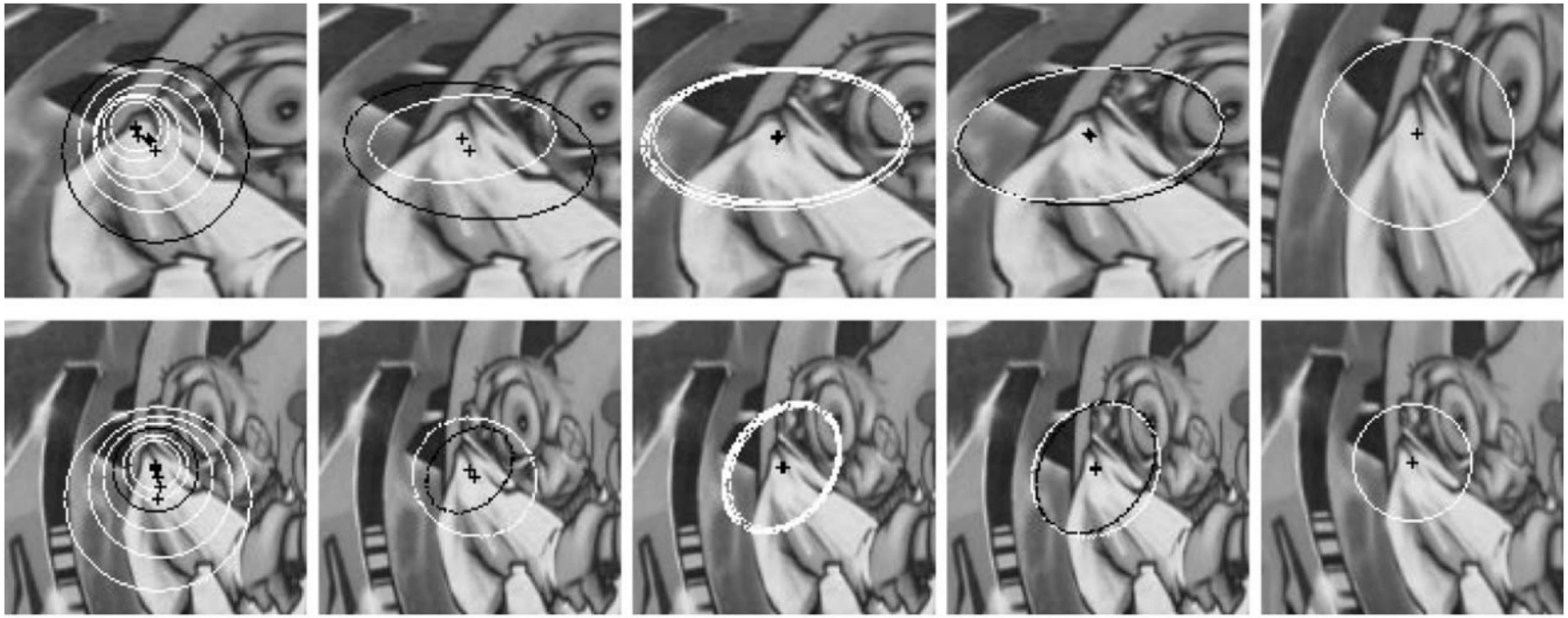
Harris pipeline



- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- compute gradient $\nabla F = (F_x, F_y)$ at derivation scale
- encode into tensor product $\nabla F \otimes \nabla F = (F_x^2, F_x F_y, F_x F_y, F_y^2)$
- average pooling by window w at integration scale
- compute point-wise nonlinear response function r

Harris affine & Hessian affine

[Mikolajczyk and Schmid 2004]



- multi-scale Harris or Hessian detection, Laplacian scale selection
- iterative affine shape adaptation, based on Lindeberg
- Hessian-affine *de facto* standard on image retrieval for several years

Mikolajczyk and Schmid IJCV 2004. Scale & Affine Invariant Interest Point Detectors.

spatial matching

dense registration

[Lucas and Kanade 1981]

- for each location in an image, find a displacement with respect to another reference image
- appropriate for small displacements, *e.g.* stereopsis or optical flow

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

dense registration

[Lucas and Kanade 1981]

- for each location in an image, find a displacement with respect to another reference image
- appropriate for small displacements, *e.g.* stereopsis or optical flow

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

dense registration

[Lucas and Kanade 1981]

- for each location in an image, find a displacement with respect to another reference image
- appropriate for small displacements, *e.g.* stereopsis or optical flow

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

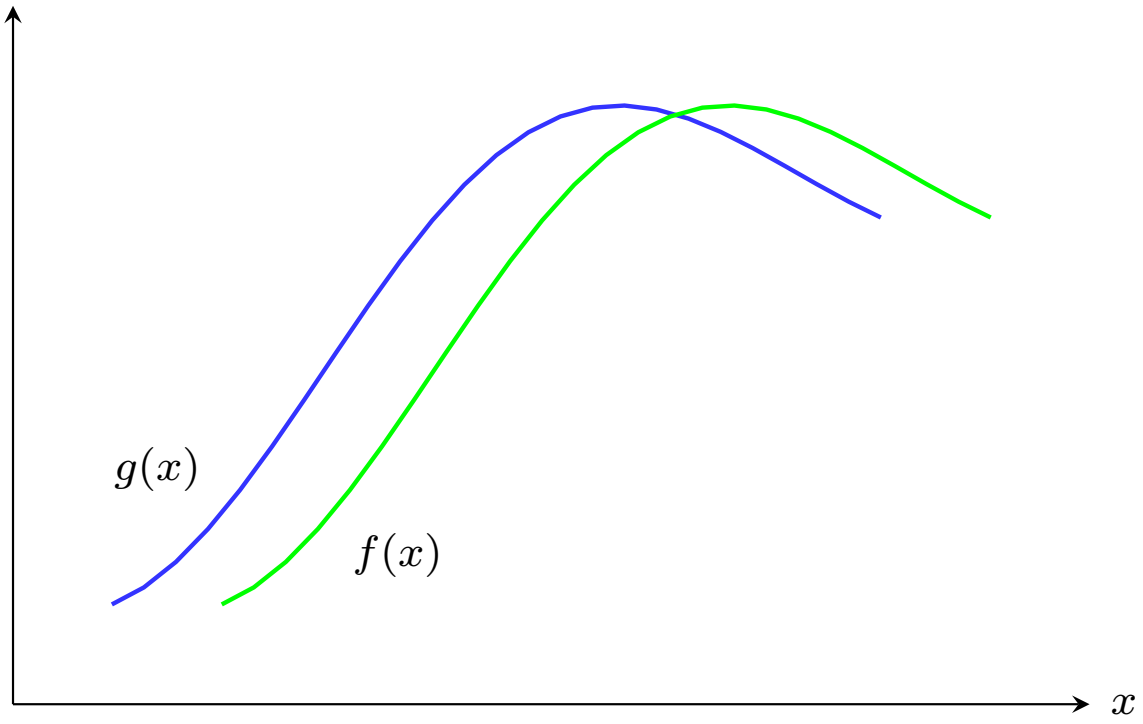
dense registration

[Lucas and Kanade 1981]

- for each location in an image, find a displacement with respect to another reference image
- appropriate for small displacements, *e.g.* stereopsis or optical flow

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

one dimension

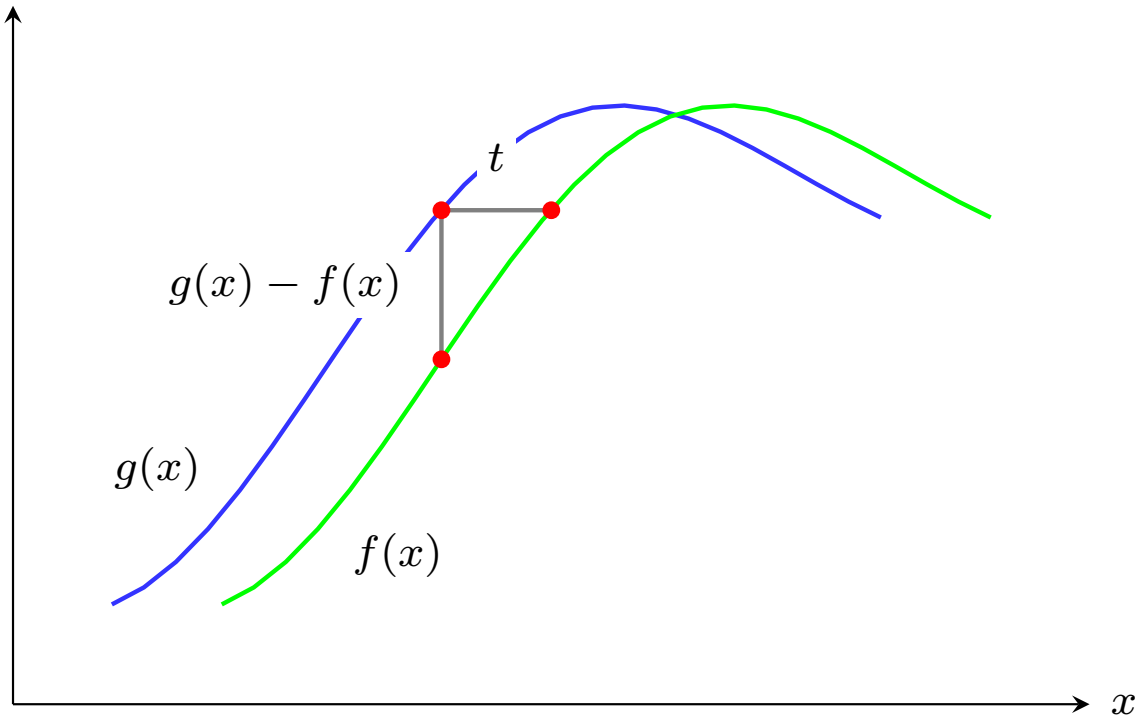


- assuming $g(x) = f(x + t)$ and t is small,

$$\frac{df}{dx}(x) \approx \frac{f(x+t) - f(x)}{t} = \frac{g(x) - f(x)}{t}$$

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

one dimension

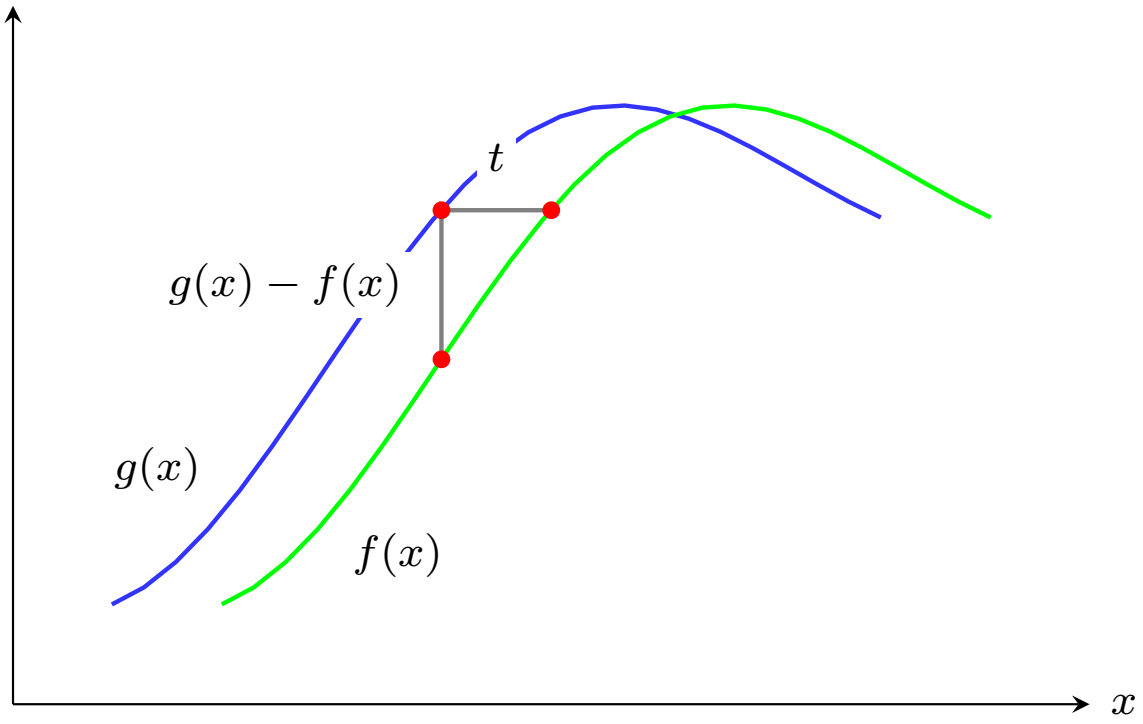


- assuming $g(x) = f(x + t)$ and t is small,

$$\frac{df}{dx}(x) \approx \frac{f(x+t) - f(x)}{t} = \frac{g(x) - f(x)}{t}$$

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

one dimension



- assuming $g(x) = f(x + t)$ and t is small,

$$\frac{df}{dx}(x) \approx \frac{f(x+t) - f(x)}{t} = \frac{g(x) - f(x)}{t}$$

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

two dimensions: least squares

- again, assume an image patch defined by window w; what is the error between the patch shifted by $\mathbf{t}$ in reference image f and a patch at the origin in shifted image g?

$$E(\mathbf{t}) = \sum_{\mathbf{x}} w(\mathbf{x})(f(\mathbf{x}+\mathbf{t}) - g(\mathbf{x}))^2$$

- error minimized when gradient vanishes

$$\mathbf{0} = \frac{\partial E}{\partial \mathbf{t}} = \sum_{\mathbf{x}} w(\mathbf{x}) 2 \nabla f(\mathbf{x})(f(\mathbf{x}) + \mathbf{t}^\top \nabla f(\mathbf{x}) - g(\mathbf{x}))$$

- least-squares solution

$$\left(w * (\nabla f)(\nabla f)^\top\right) \mathbf{t} = w * ((\nabla f)(g - f))$$

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

two dimensions: least squares

- again, assume an image patch defined by window w; what is the error between the patch shifted by $\mathbf{t}$ in reference image f and a patch at the origin in shifted image g?

$$E(\mathbf{t}) = \sum_{\mathbf{x}} w(\mathbf{x})(f(\mathbf{x}+\mathbf{t}) - g(\mathbf{x}))^2$$
$$\approx \sum_{\mathbf{x}} w(\mathbf{x})(f(\mathbf{x}) + \mathbf{t}^\top \nabla f(\mathbf{x}) - g(\mathbf{x}))^2$$

- error minimized when gradient vanishes

$$\mathbf{0} = \frac{\partial E}{\partial \mathbf{t}} = \sum_{\mathbf{x}} w(\mathbf{x}) 2 \nabla f(\mathbf{x})(f(\mathbf{x}) + \mathbf{t}^\top \nabla f(\mathbf{x}) - g(\mathbf{x}))$$

- least-squares solution

$$\left(w * (\nabla f)(\nabla f)^\top\right) \mathbf{t} = w * ((\nabla f)(g - f))$$

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

two dimensions: least squares

- again, assume an image patch defined by window w; what is the error between the patch shifted by $\mathbf{t}$ in reference image f and a patch at the origin in shifted image g?

$$E(\mathbf{t}) = \sum_{\mathbf{x}} w(\mathbf{x})(f(\mathbf{x}+\mathbf{t}) - g(\mathbf{x}))^2$$
$$\approx \sum_{\mathbf{x}} w(\mathbf{x})(f(\mathbf{x}) + \mathbf{t}^\top \nabla f(\mathbf{x}) - g(\mathbf{x}))^2$$

- error minimized when gradient vanishes

$$\mathbf{0} = \frac{\partial E}{\partial \mathbf{t}} = \sum_{\mathbf{x}} w(\mathbf{x}) 2 \nabla f(\mathbf{x})(f(\mathbf{x}) + \mathbf{t}^\top \nabla f(\mathbf{x}) - g(\mathbf{x}))$$

- least-squares solution

$$\left(w * (\nabla f)(\nabla f)^\top\right) \mathbf{t} = w * ((\nabla f)(g - f))$$

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

two dimensions: least squares

- again, assume an image patch defined by window w; what is the error between the patch shifted by $\mathbf{t}$ in reference image f and a patch at the origin in shifted image g?

$$E(\mathbf{t}) = \sum_{\mathbf{x}} w(\mathbf{x})(f(\mathbf{x}+\mathbf{t}) - g(\mathbf{x}))^2$$
$$\approx \sum_{\mathbf{x}} w(\mathbf{x})(f(\mathbf{x}) + \mathbf{t}^\top \nabla f(\mathbf{x}) - g(\mathbf{x}))^2$$

- error minimized when gradient vanishes

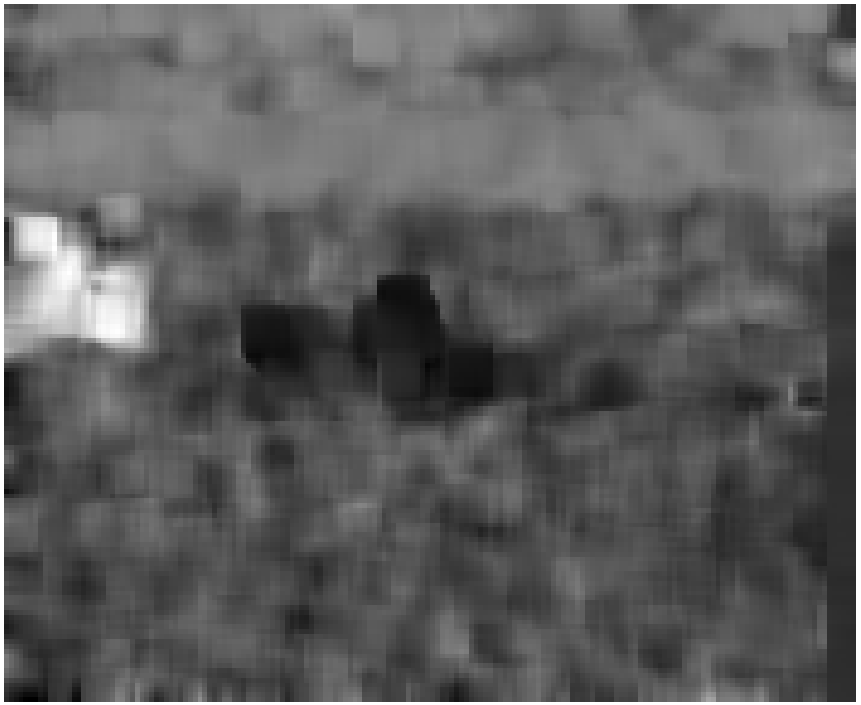
$$\mathbf{0} = \frac{\partial E}{\partial \mathbf{t}} = \sum_{\mathbf{x}} w(\mathbf{x}) 2 \nabla f(\mathbf{x})(f(\mathbf{x}) + \mathbf{t}^\top \nabla f(\mathbf{x}) - g(\mathbf{x}))$$

- least-squares solution

$$\left(w * (\nabla f)(\nabla f)^\top\right) \mathbf{t} = w * ((\nabla f)(g - f))$$

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

dense optical flow



- camera follows background, two objects at opposite horizontal directions
- motion noisy on uniform regions

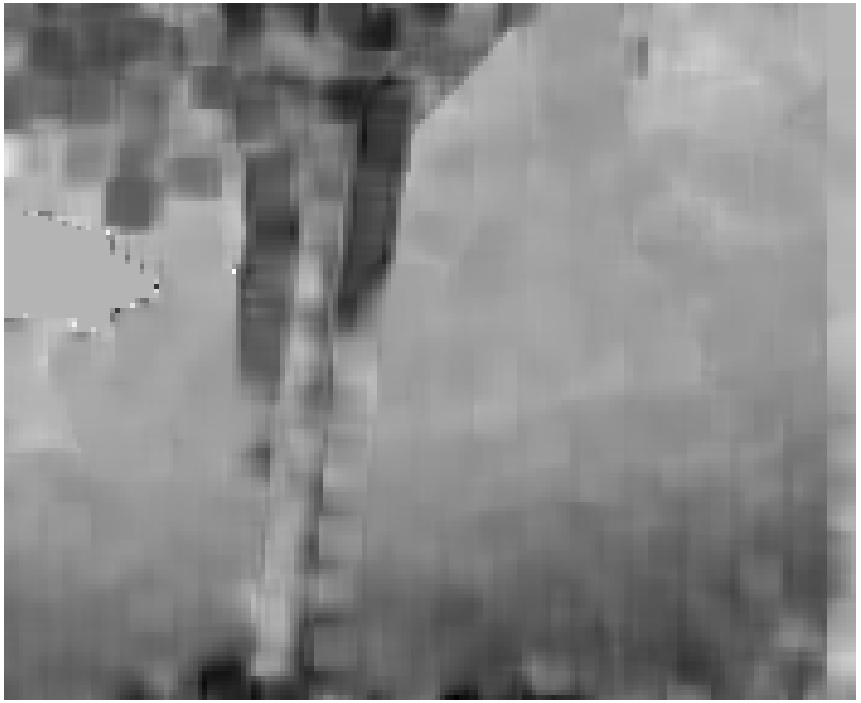
Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

dense optical flow

- camera follows background, two objects at opposite horizontal directions
- motion noisy on uniform regions

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

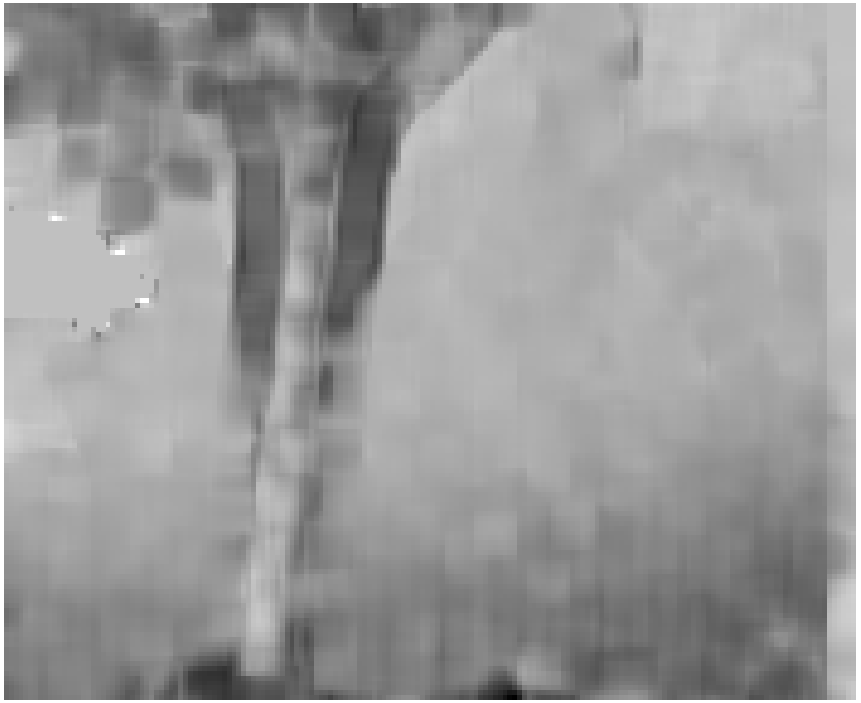
dense optical flow



- parallax: tree closer to viewer than background
- stable on textured regions
- window size visible on edges

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

dense optical flow



- parallax: tree closer to viewer than background
- stable on textured regions
- window size visible on edges

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

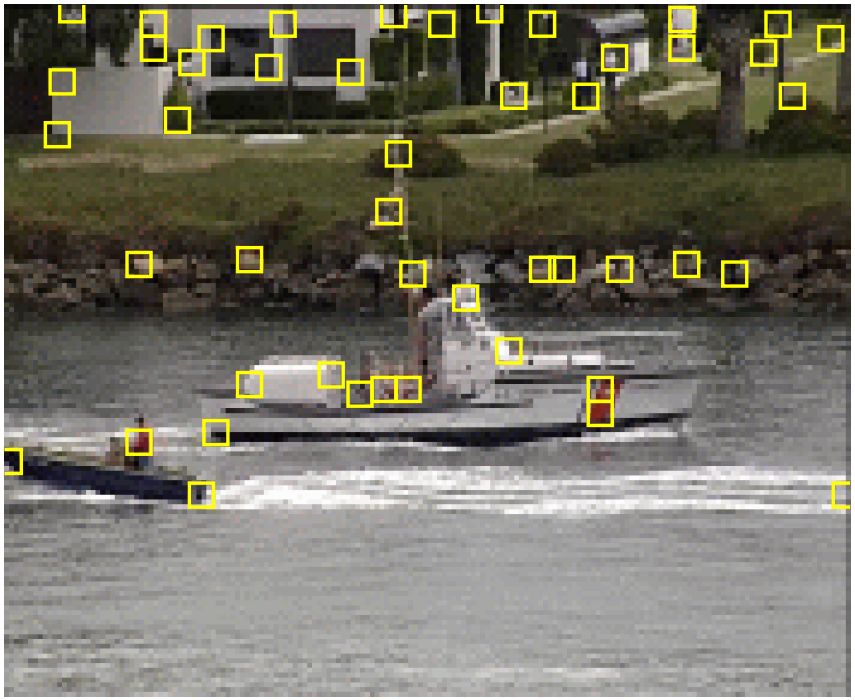
feature point tracking

[Tomasi and Kanade 1991]

- linear system can be solved reliably if matrix μ is well-conditioned: λ_1/λ_2 is not too large
- detect feature points at local maxima of response $\min(\lambda_1, \lambda_2)$

Tomasi and Kanade 1991. Detection and Tracking of Point Features.

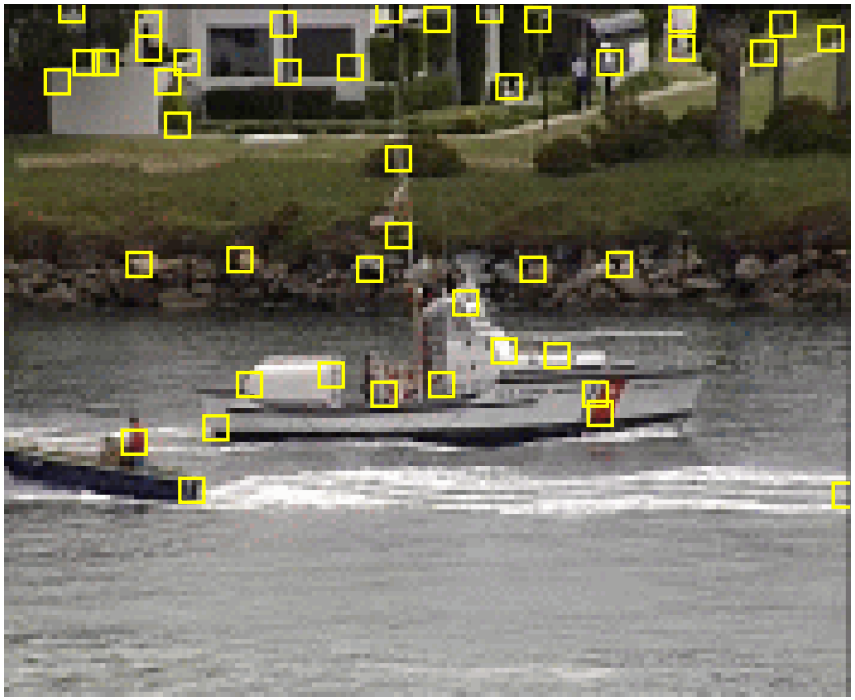
feature point tracking



- uniform regions are not tracked now
- nearly same response as Harris corners
- Q: why do we need the window? what should the size be?

Tomasi and Kanade 1991. Detection and Tracking of Point Features.

feature point tracking



- uniform regions are not tracked now
- nearly same response as Harris corners
- Q: why do we need the window? what should the size be?

Tomasi and Kanade 1991. Detection and Tracking of Point Features.

wide-baseline matching

- in dense registration, we started from a local "template matching" process and found an efficient solution based on a Taylor approximation
- both make sense for small displacements
- in wide-baseline matching, every part of one image may appear anywhere in the other
- we start by pairwise matching of local descriptors without any order and then attempt to enforce some geometric consistency according to a rigid motion model

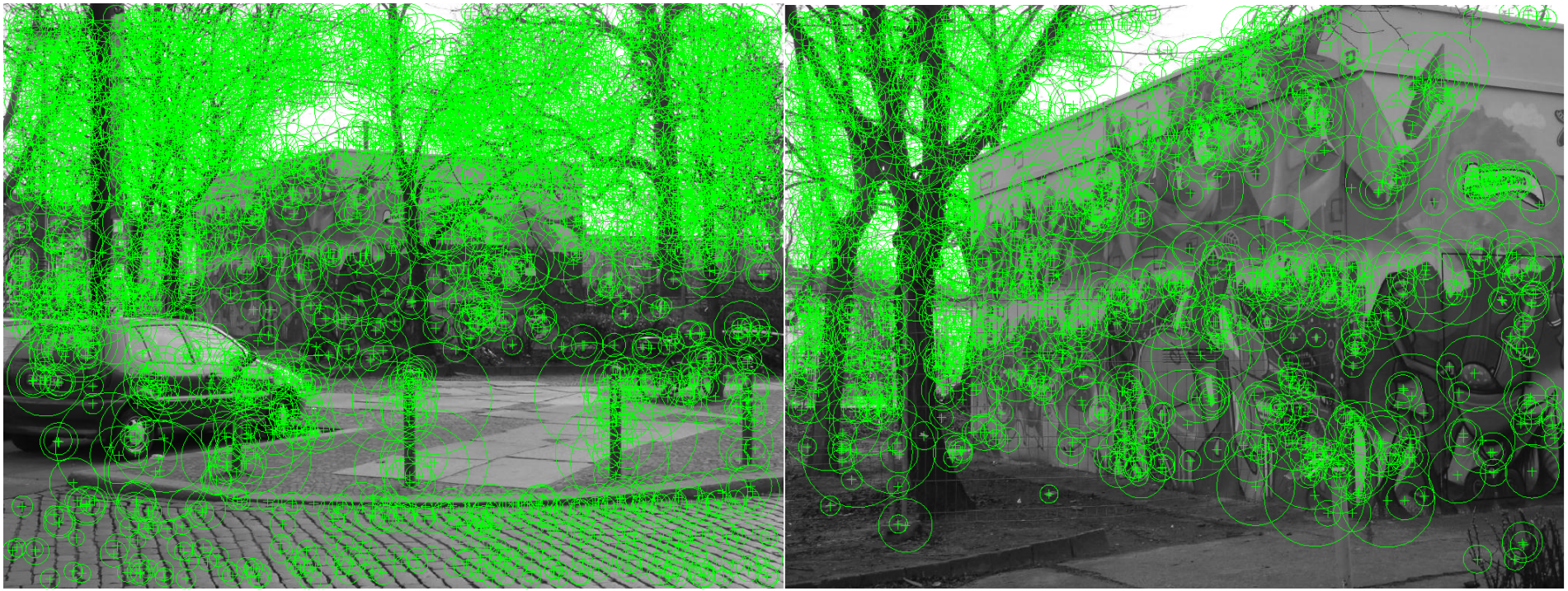
wide-baseline matching

- in dense registration, we started from a local "template matching" process and found an efficient solution based on a Taylor approximation
- both make sense for small displacements
- in wide-baseline matching, every part of one image may appear anywhere in the other
- we start by pairwise matching of local descriptors without any order and then attempt to enforce some geometric consistency according to a rigid motion model

wide-baseline matching

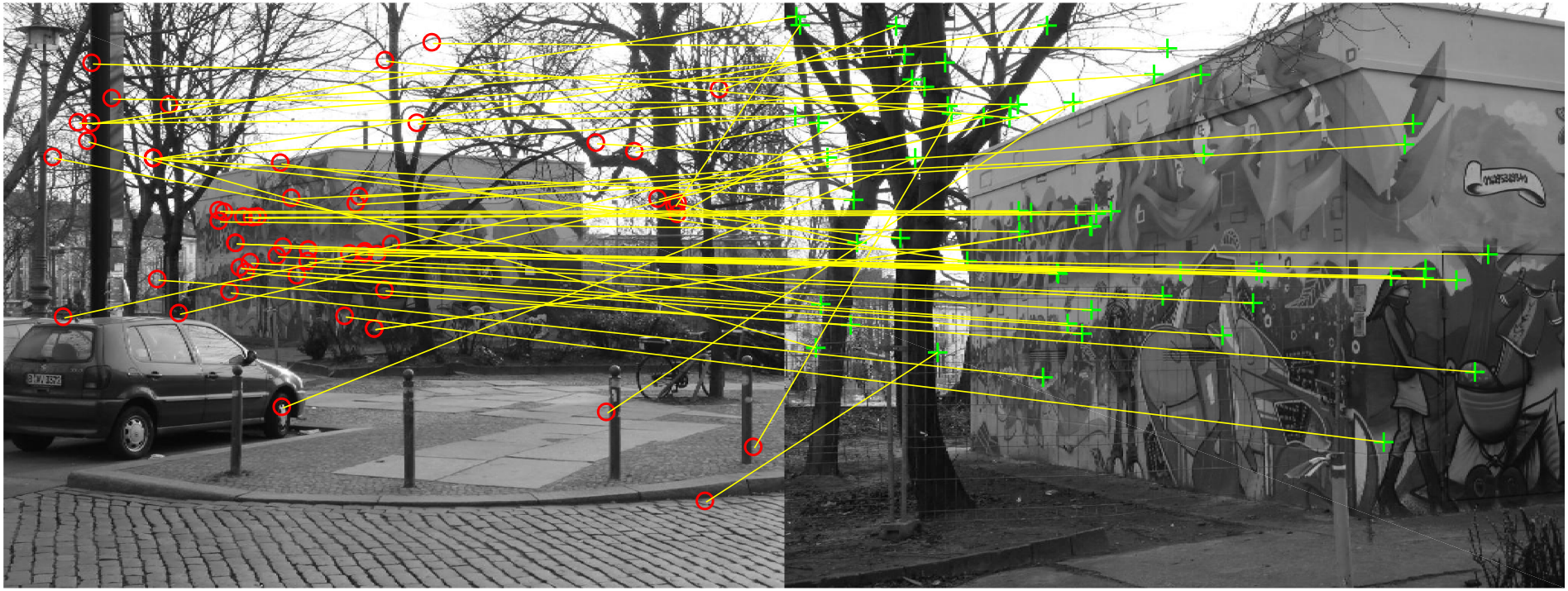
- a region in one image may appear anywhere in the other

wide-baseline matching



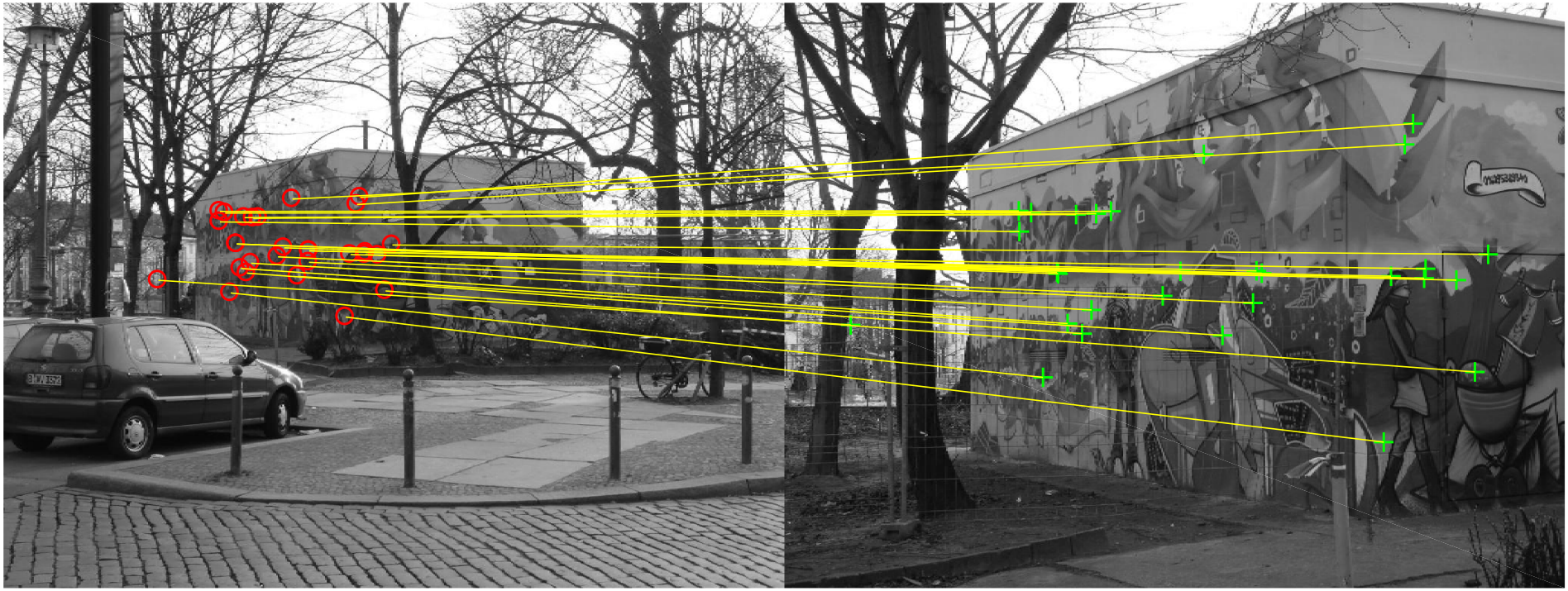
- features detected independently in each image

wide-baseline matching



- tentative correspondences by pairwise descriptor matching

wide-baseline matching



- subset of correspondences that are ‘inlier’ to a rigid transformation

descriptor extraction

for each detected feature in each image

- construct a local histogram of gradient orientations
- find one or more dominant orientations corresponding to peaks in the histogram
- resample local patch at given location, scale, affine shape and orientation
- extract one descriptor for each dominant orientation

Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

descriptor matching

Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

descriptor matching

- detect features

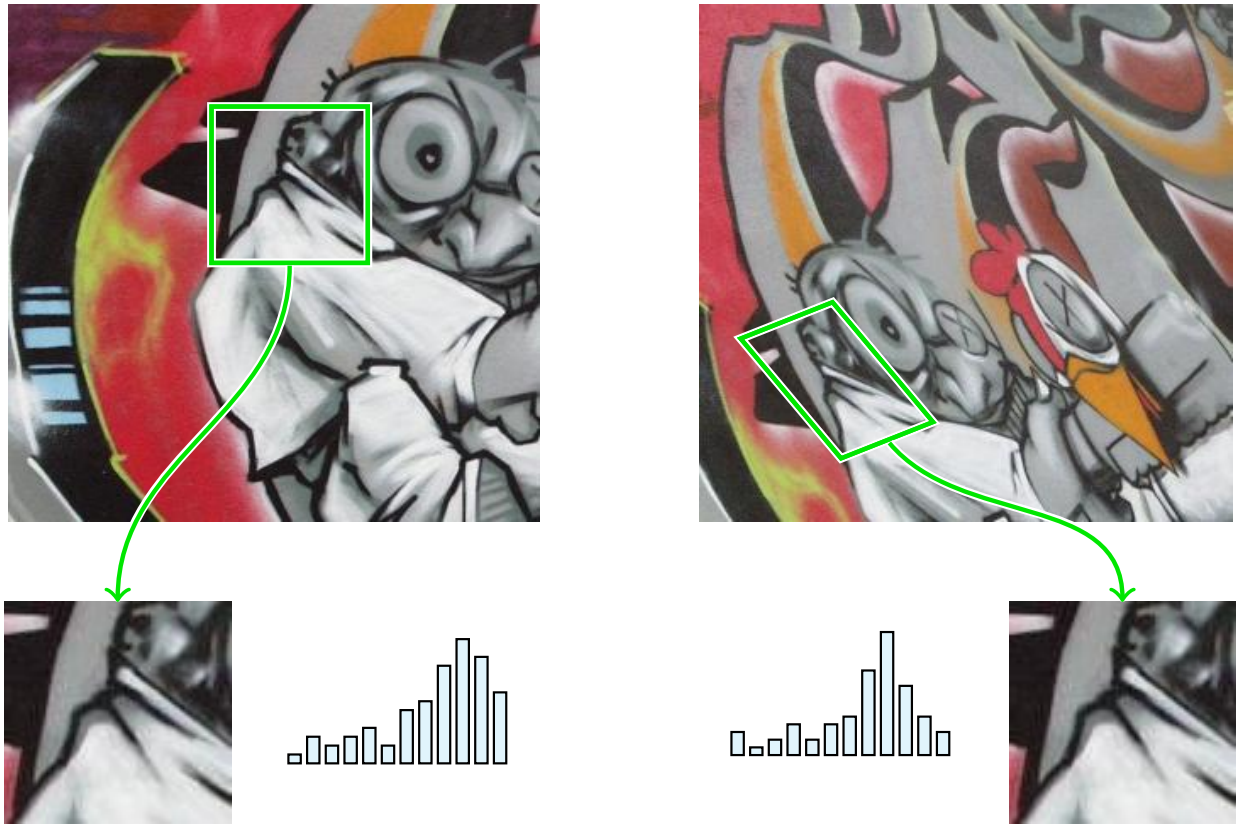
Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

descriptor matching

- detect features - find dominant orientation, resample patches

Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

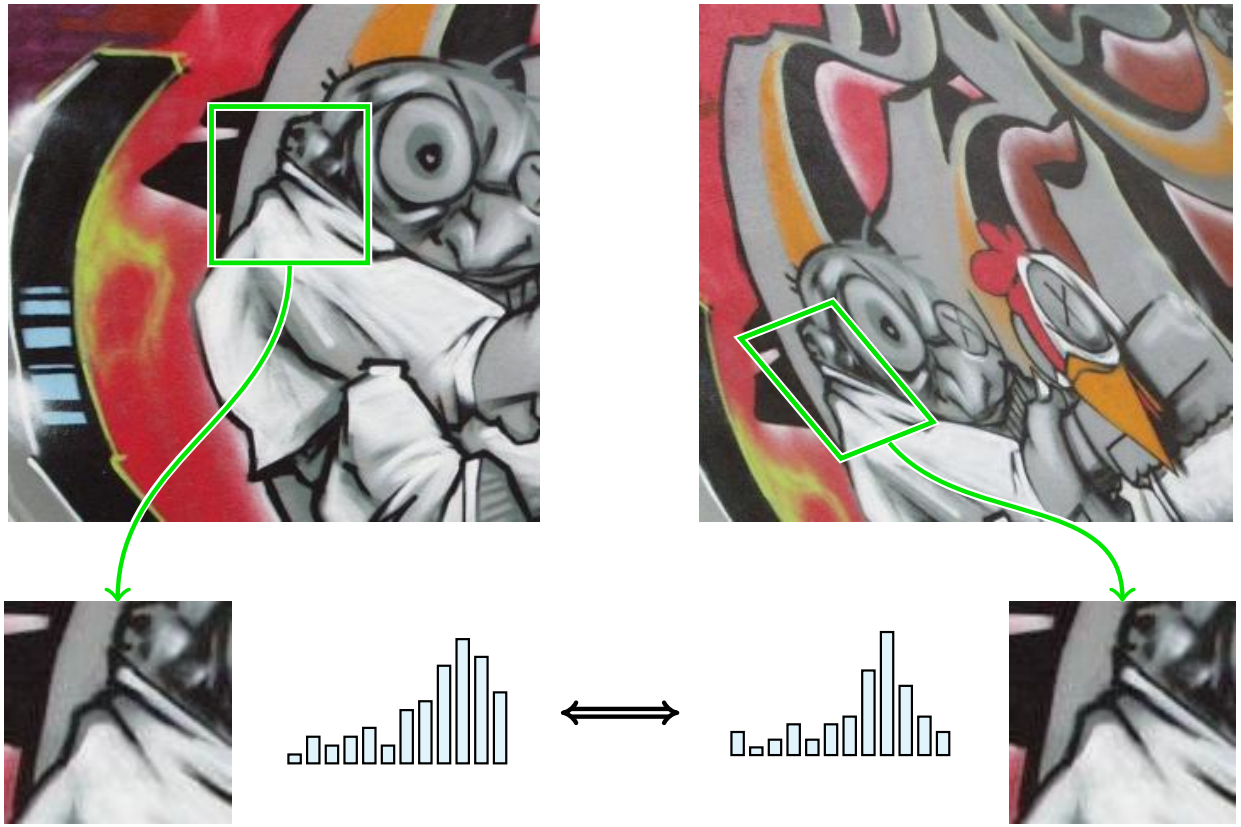
descriptor matching



- detect features - find dominant orientation, resample patches - extract descriptors

Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

descriptor matching



- detect features - find dominant orientation, resample patches - extract descriptors - match pairwise

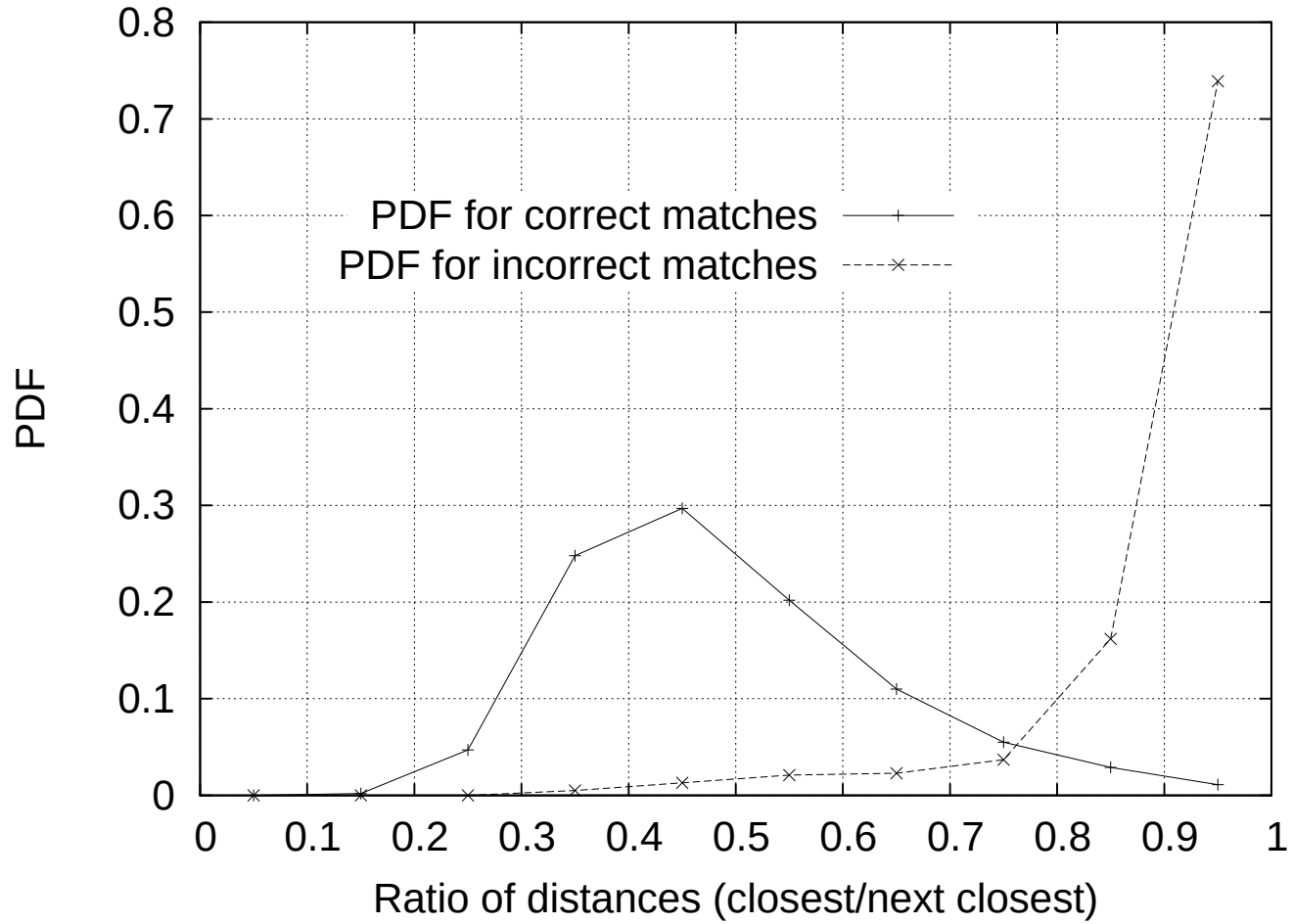
Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

descriptor matching

- for each descriptor in one image, find its two nearest neighbors in the other
- if ratio of distance of first to distance of second is small, make a correspondence
- this yields a list of tentative correspondences

Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

ratio test



- ratio of first to second nearest neighbor distance can determine the probability of a true correspondence

Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

spatial matching

why is it difficult?

- should allow for a geometric transformation
- fitting the model to data (correspondences) is sensitive to outliers: should find a subset of *inliers* first
- finding inliers to a transformation requires finding the *transformation* in the first place
- correspondences have gross error
- inliers are typically less than 50%

geometric transformations

- two images f, f' are equal at points x, x'

$$f(\mathbf{x}) = f'(\mathbf{x}')$$

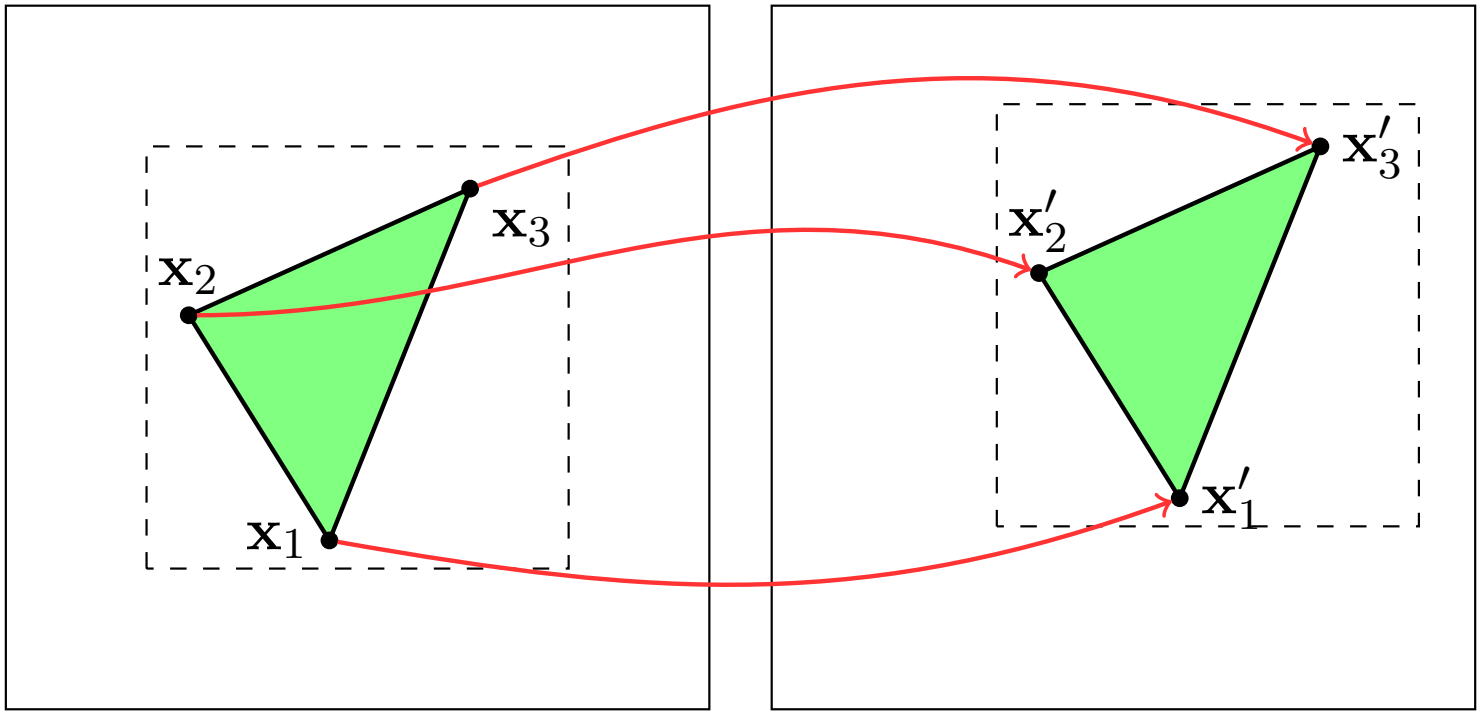
- $\mathbf{x}$ is mapped to $\mathbf{x}'$

$$\mathbf{x}' = T(\mathbf{x})$$

- T is a bijection of $\mathbb{R}^2$ to itself:

$$T : \mathbb{R}^2 \to \mathbb{R}^2$$

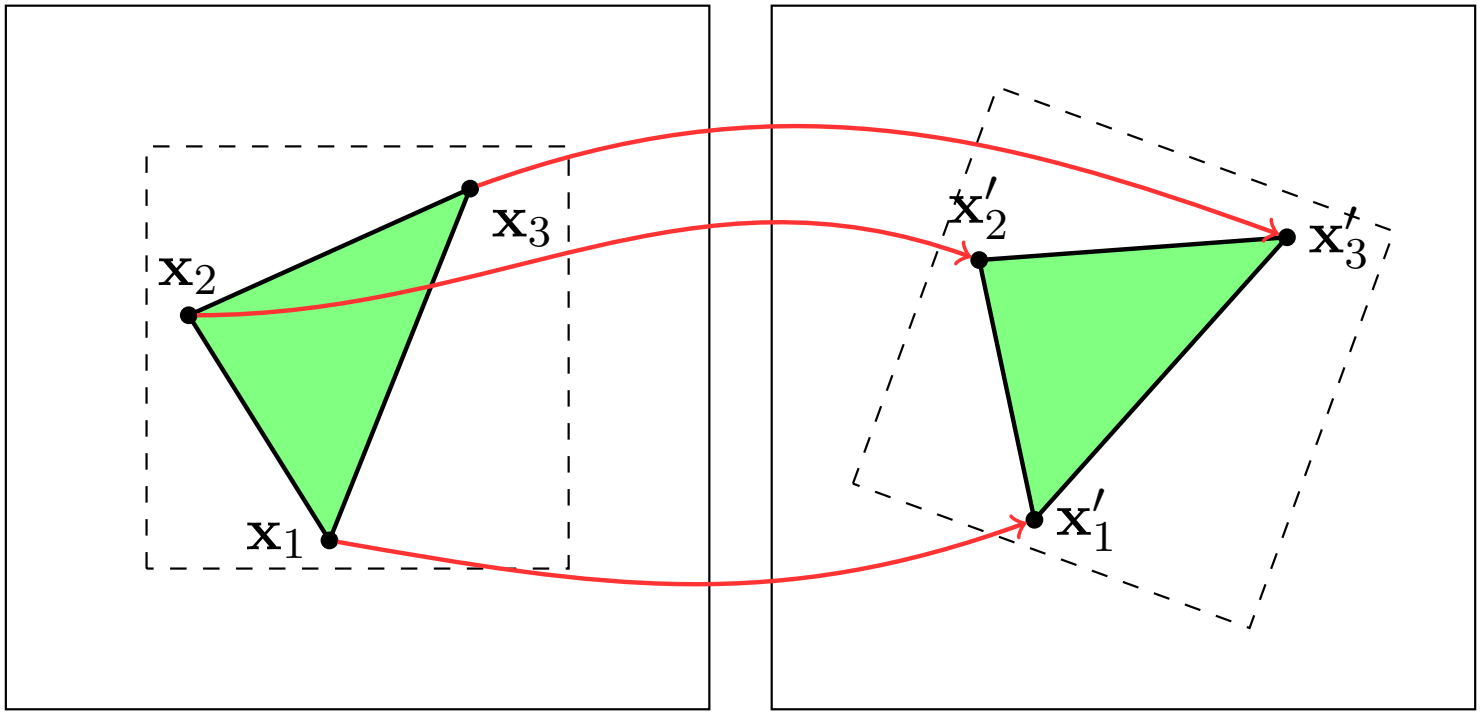
geometric transformations



- translation: 2 degrees of freedom

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

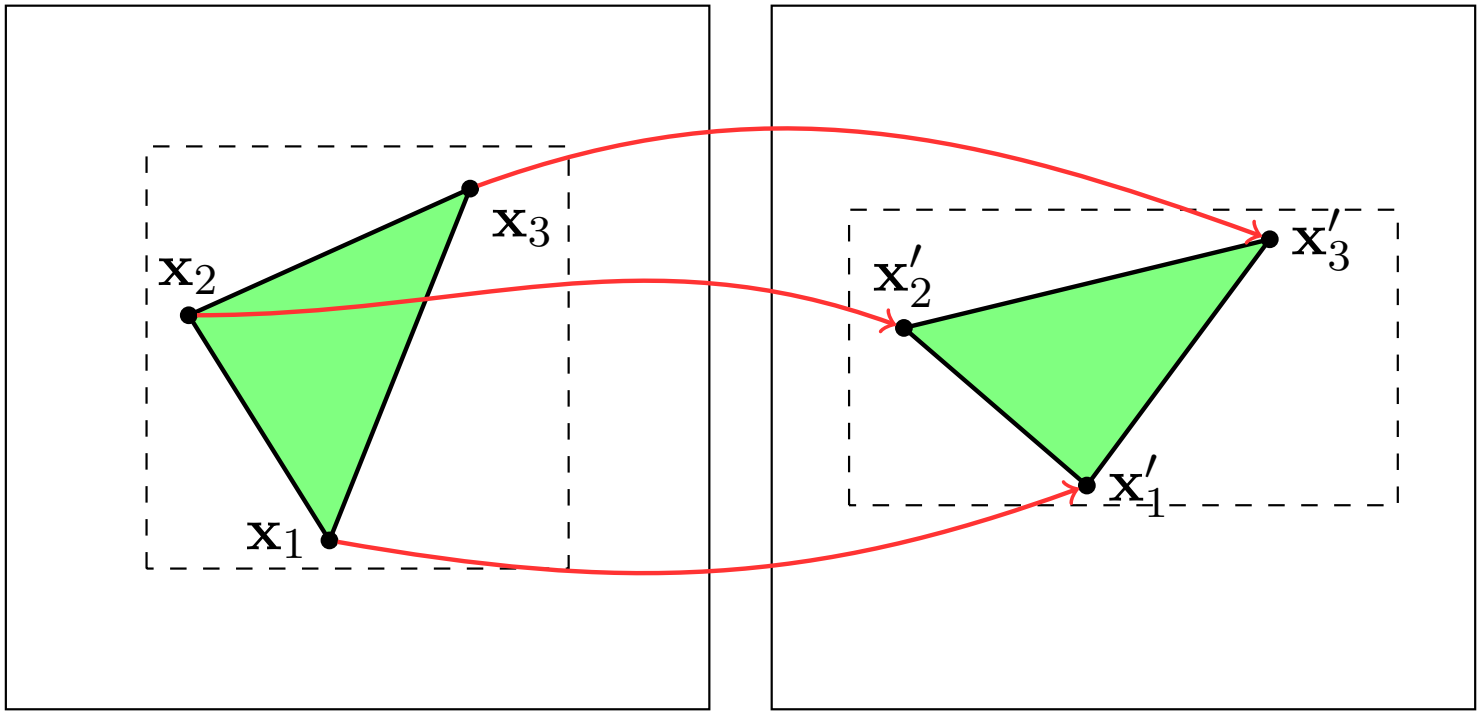
geometric transformations



- rotation: 1 degree of freedom

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

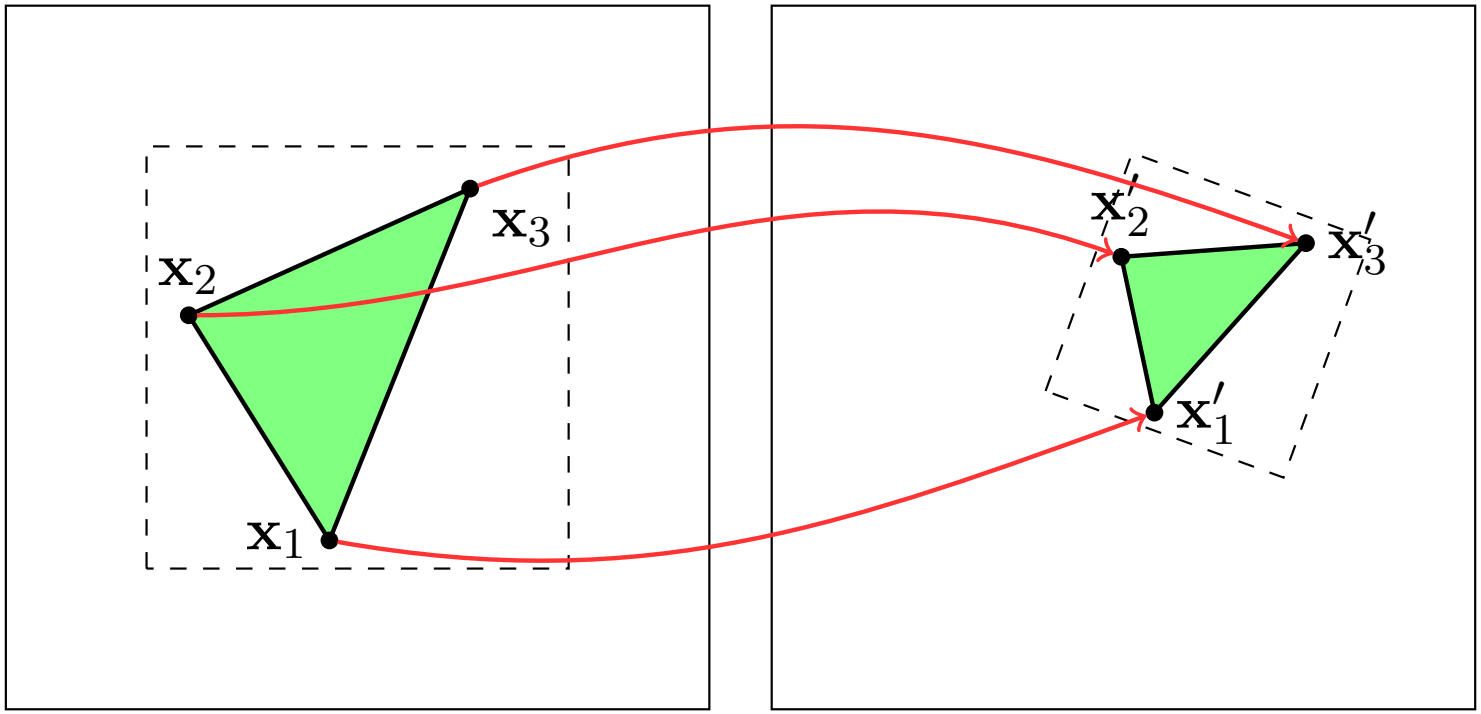
geometric transformations



- scale: 2 degrees of freedom

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

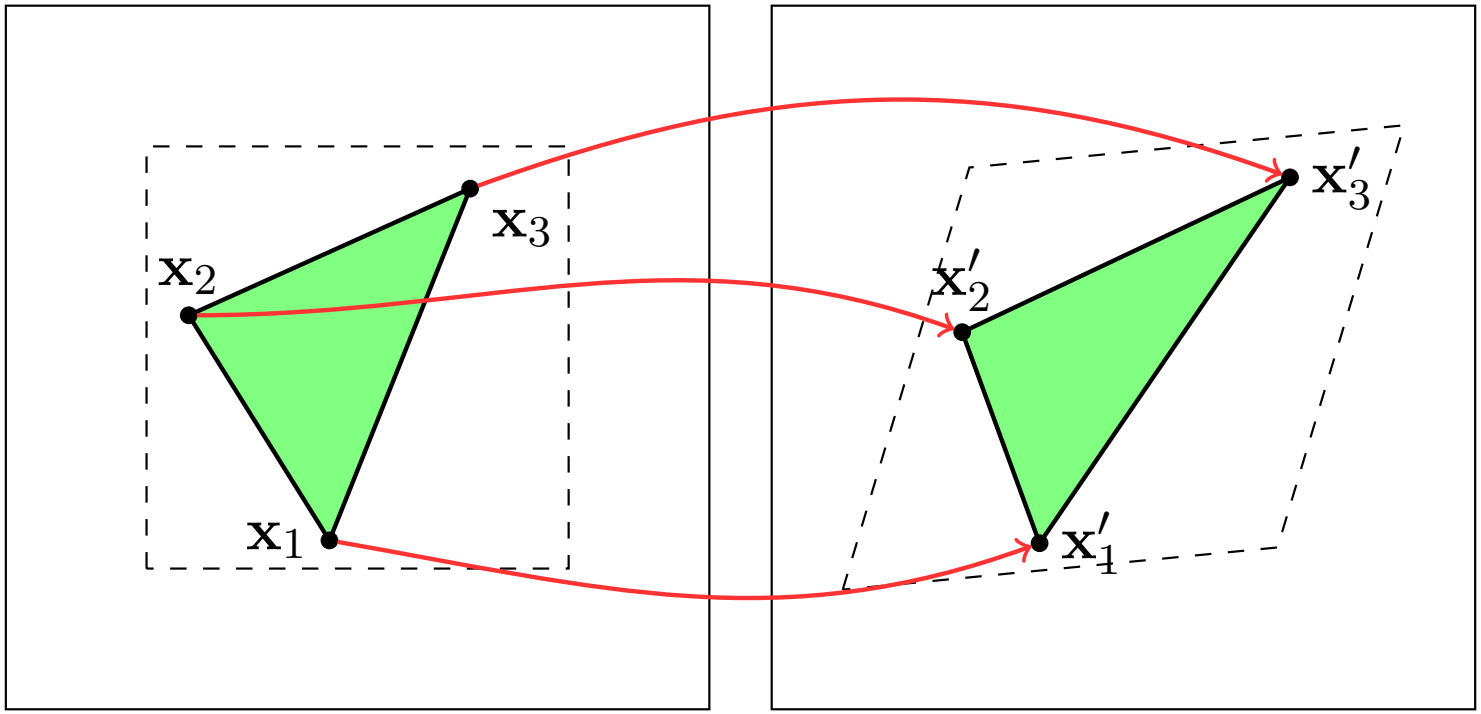
geometric transformations



- similarity: 4 degrees of freedom

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} r\cos\theta & -r\sin\theta & t_x \\ r\sin\theta & r\cos\theta & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

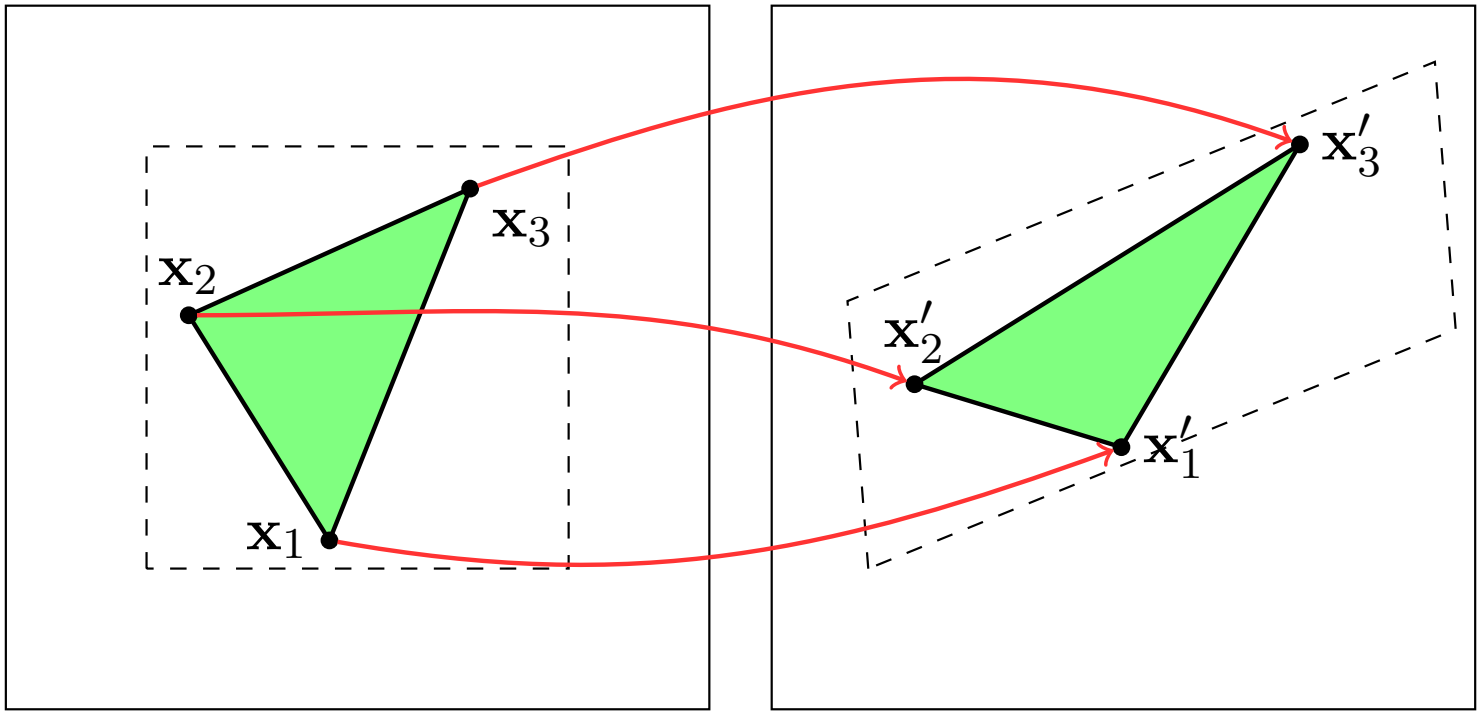
geometric transformations



- shear: 2 degrees of freedom

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & b_x & 0 \\ b_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

geometric transformations



- affine: 6 degrees of freedom

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

however

- details don't matter; in all cases, the problem is transformed to a linear system (why?)

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

  where $\mathbf{A}, \mathbf{b}$ contain coordinates of known point correspondences from images f, f' respectively, and $\mathbf{x}$ contains our model parameters

- we need $n = \lceil d/2 \rceil$ correspondences, where d are the degrees of freedom of our model
- let's take the simplest model as an example: fit a line to two points

however

- details don't matter; in all cases, the problem is transformed to a linear system (why?)

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

  where $\mathbf{A}, \mathbf{b}$ contain coordinates of known point correspondences from images f, f' respectively, and $\mathbf{x}$ contains our model parameters
- we need $n = \lceil d/2 \rceil$ correspondences, where d are the degrees of freedom of our model
- let's take the simplest model as an example: fit a line to two points

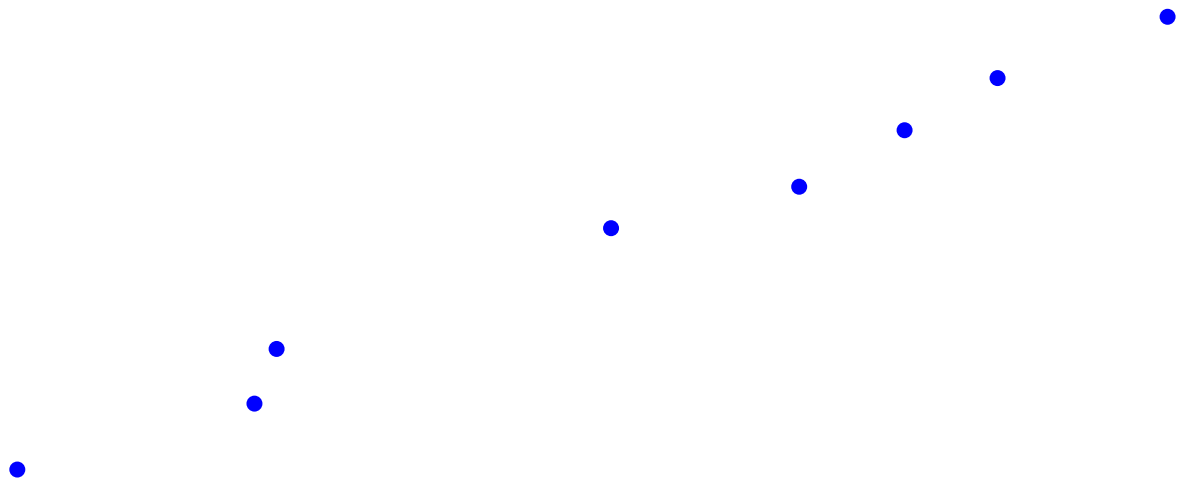
however

- details don't matter; in all cases, the problem is transformed to a linear system (why?)

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

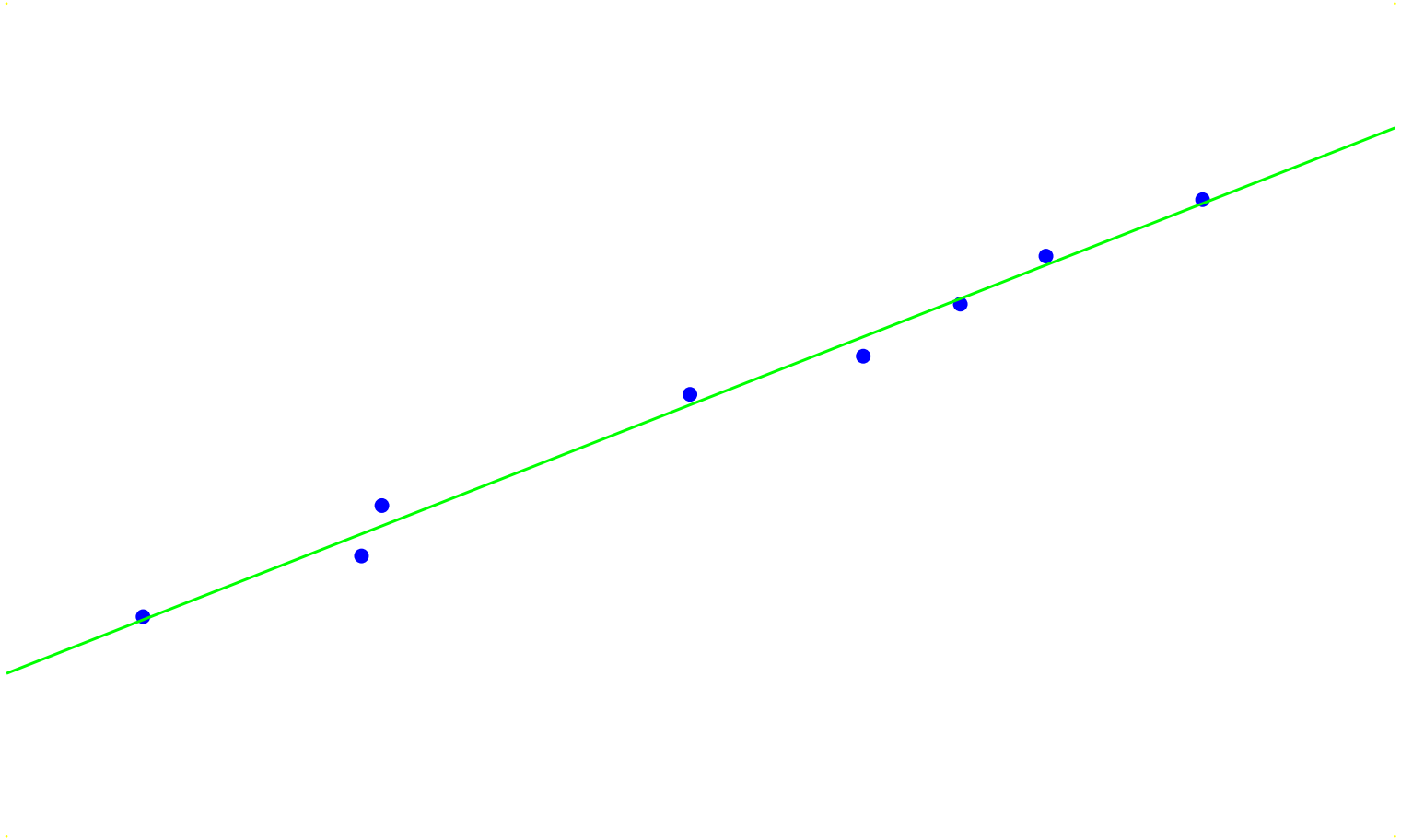
  where $\mathbf{A}, \mathbf{b}$ contain coordinates of known point correspondences from images f, f' respectively, and $\mathbf{x}$ contains our model parameters
- we need $n = \lceil d/2 \rceil$ correspondences, where d are the degrees of freedom of our model
- let's take the simplest model as an example: fit a line to two points

least squares and gross outliers



- clean data, no outliers : least squares fit ok

least squares and gross outliers

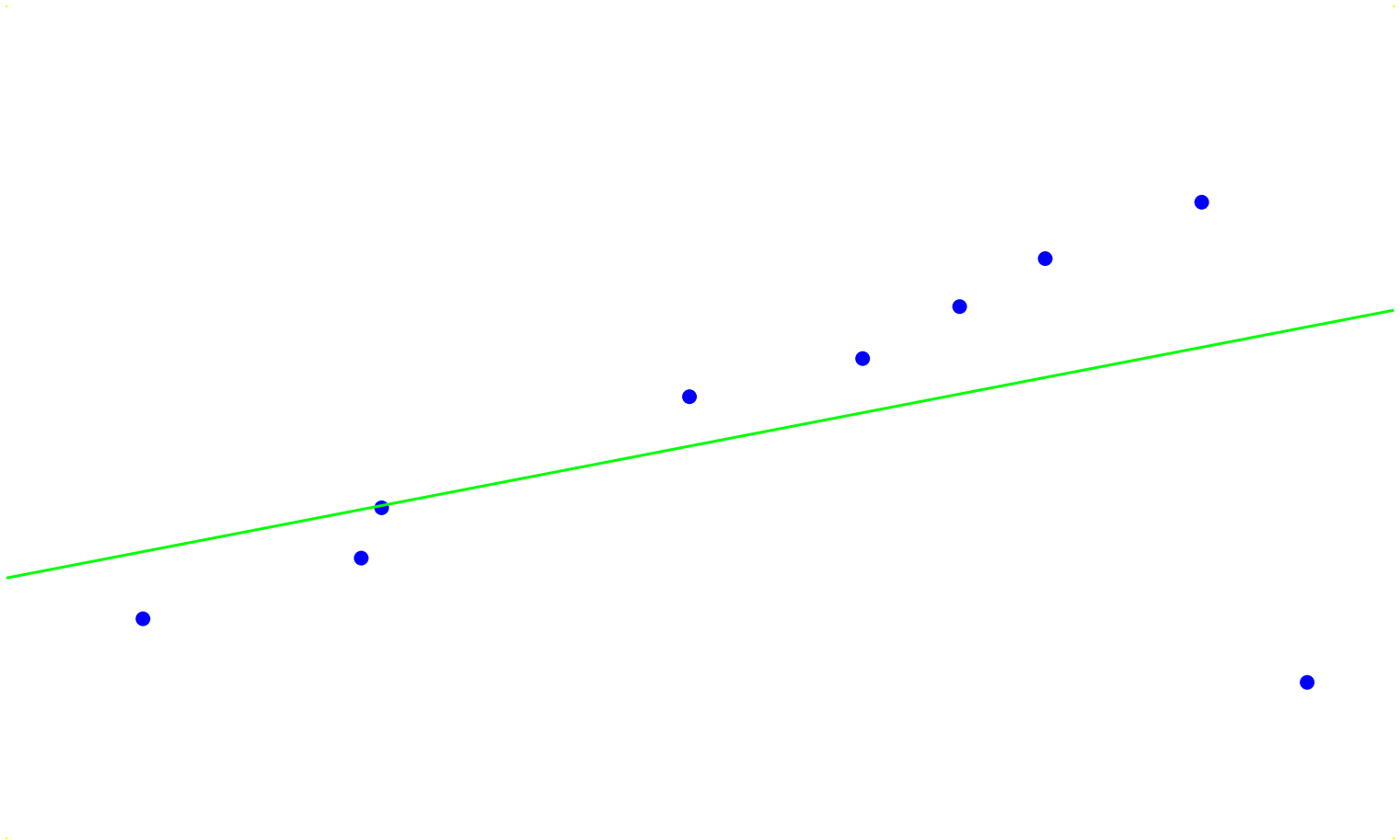


- clean data, no outliers : least squares fit ok

least squares and gross outliers

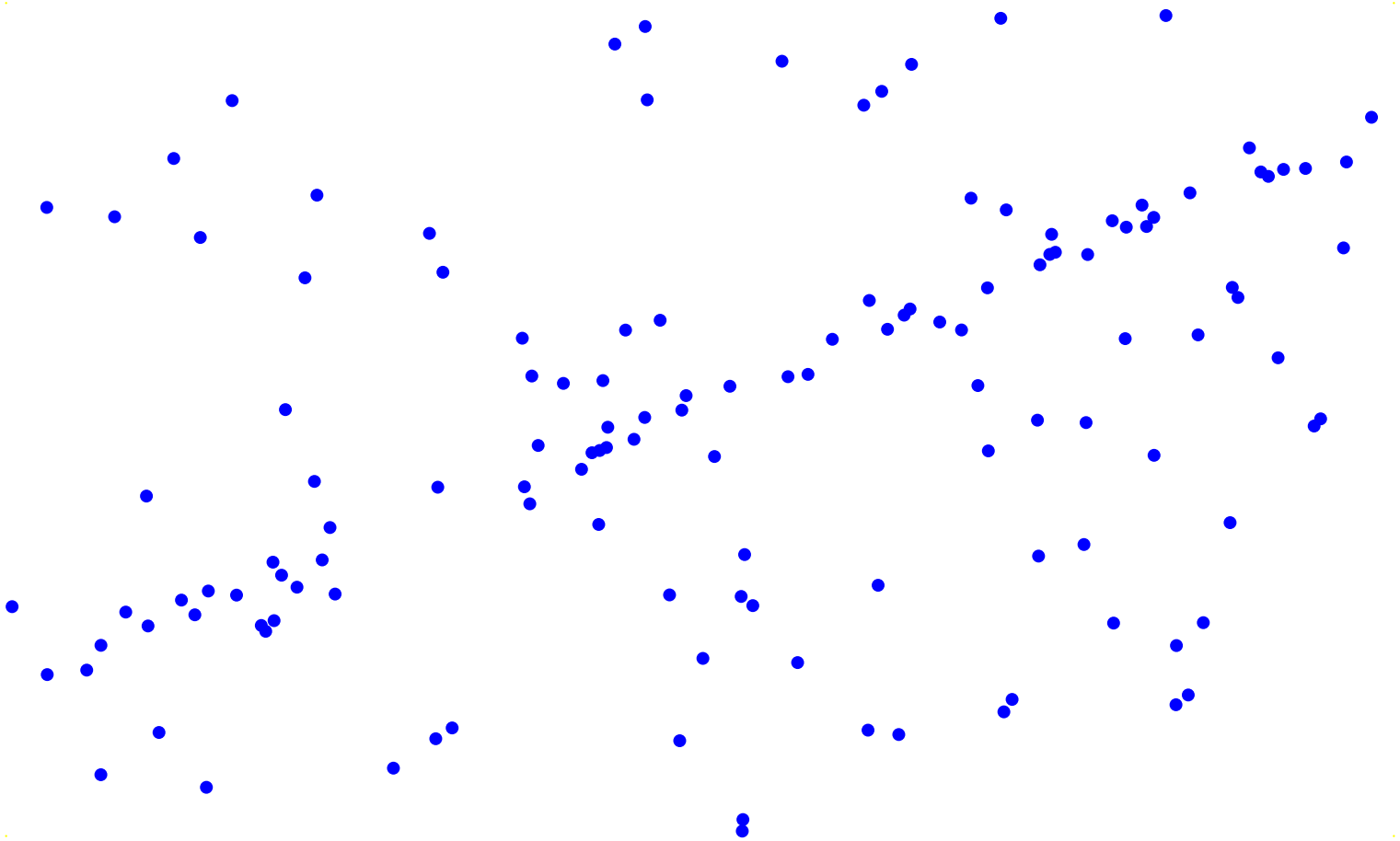
- one gross outlier : least squares fit fails

least squares and gross outliers



- one gross outlier : least squares fit fails

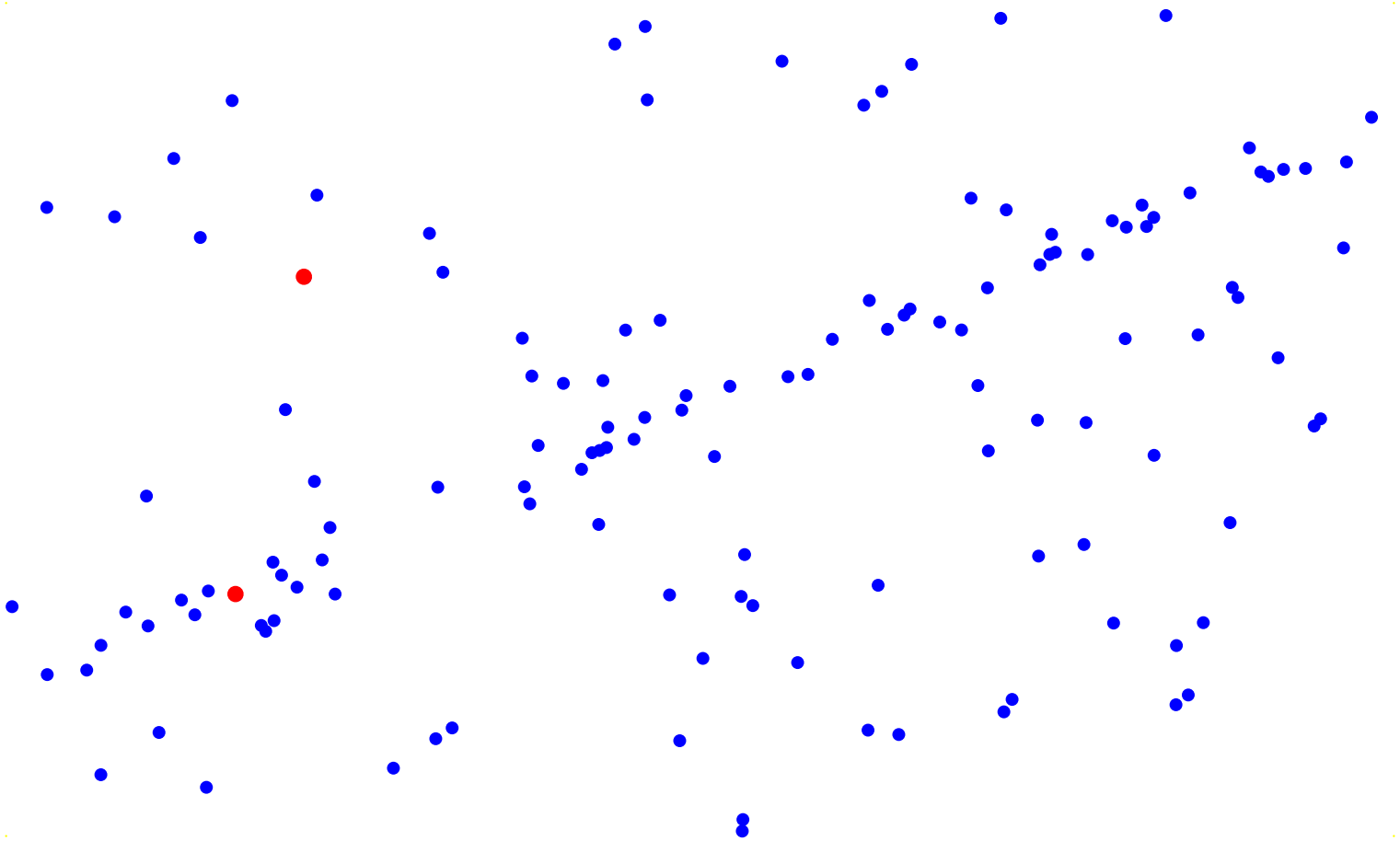
random sample consensus (RANSAC)



- data with outliers - pick two points at random - draw line through them - set margin on either side - count inlier points

Fischler and Bolles. CACM 1981. Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography.

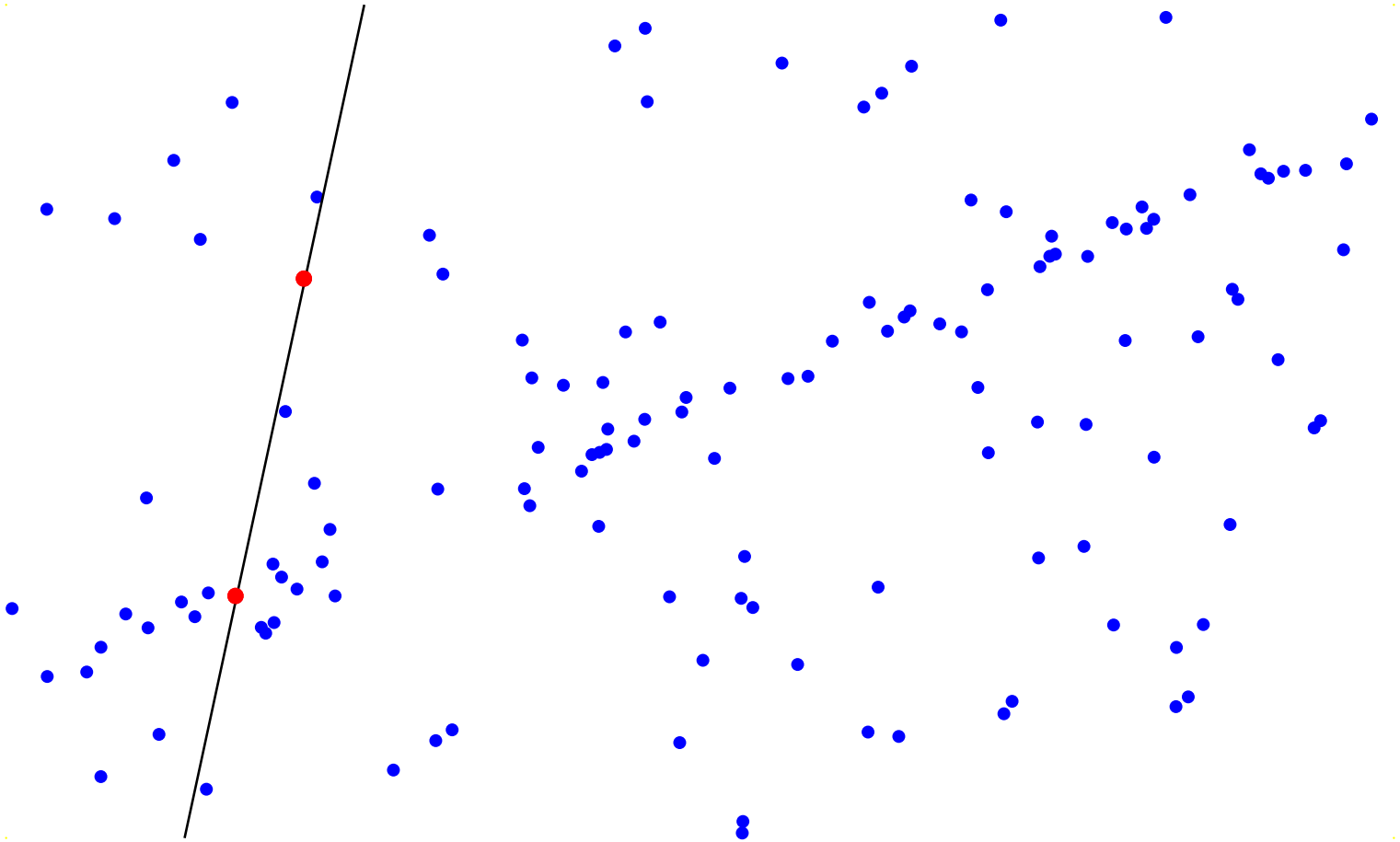
random sample consensus (RANSAC)



- data with outliers - pick two points at random - draw line through them - set margin on either side - count inlier points

Fischler and Bolles. CACM 1981. Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography.

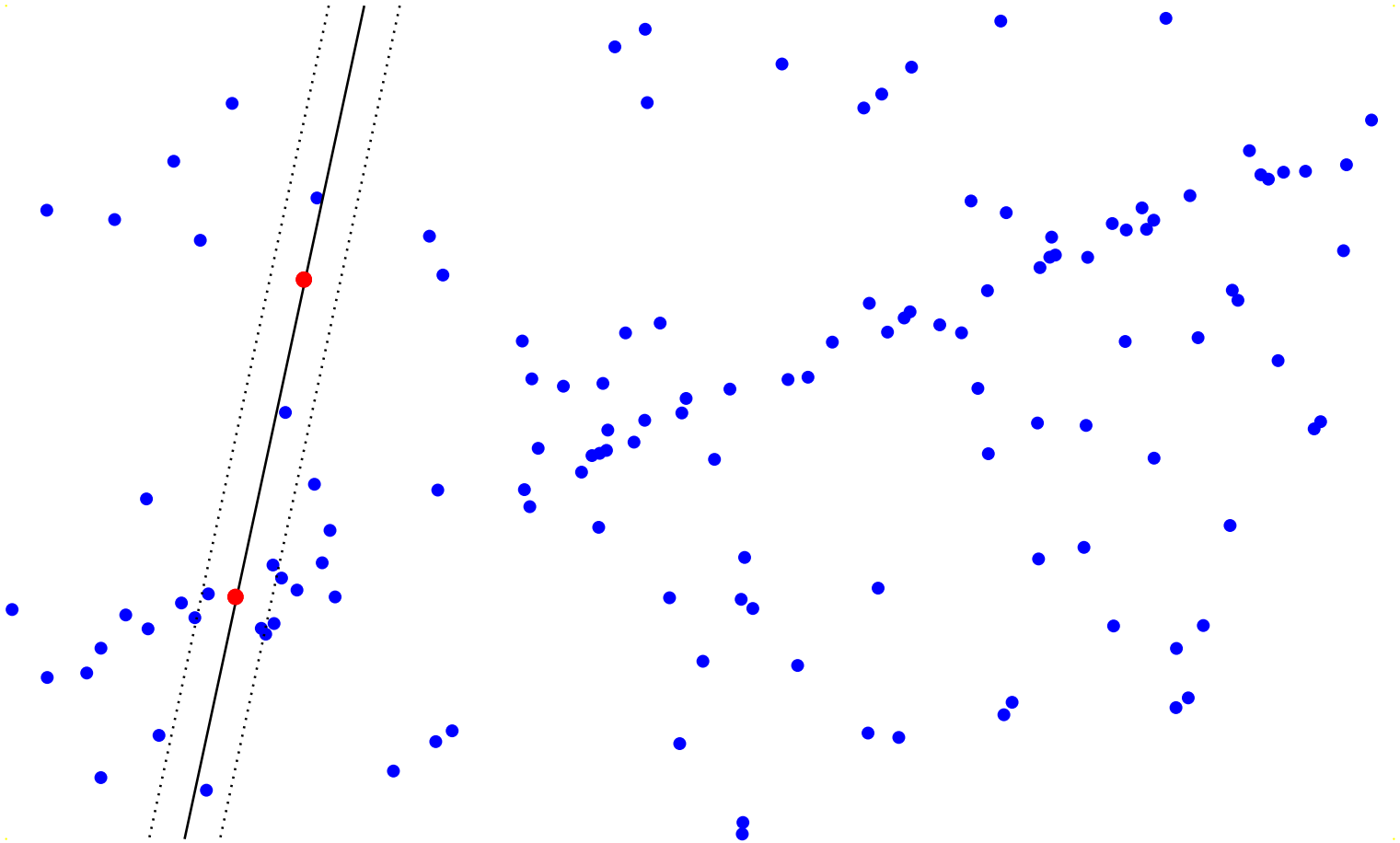
random sample consensus (RANSAC)



- data with outliers - pick two points at random - draw line through them - set margin on either side - count inlier points

Fischler and Bolles. CACM 1981. Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography.

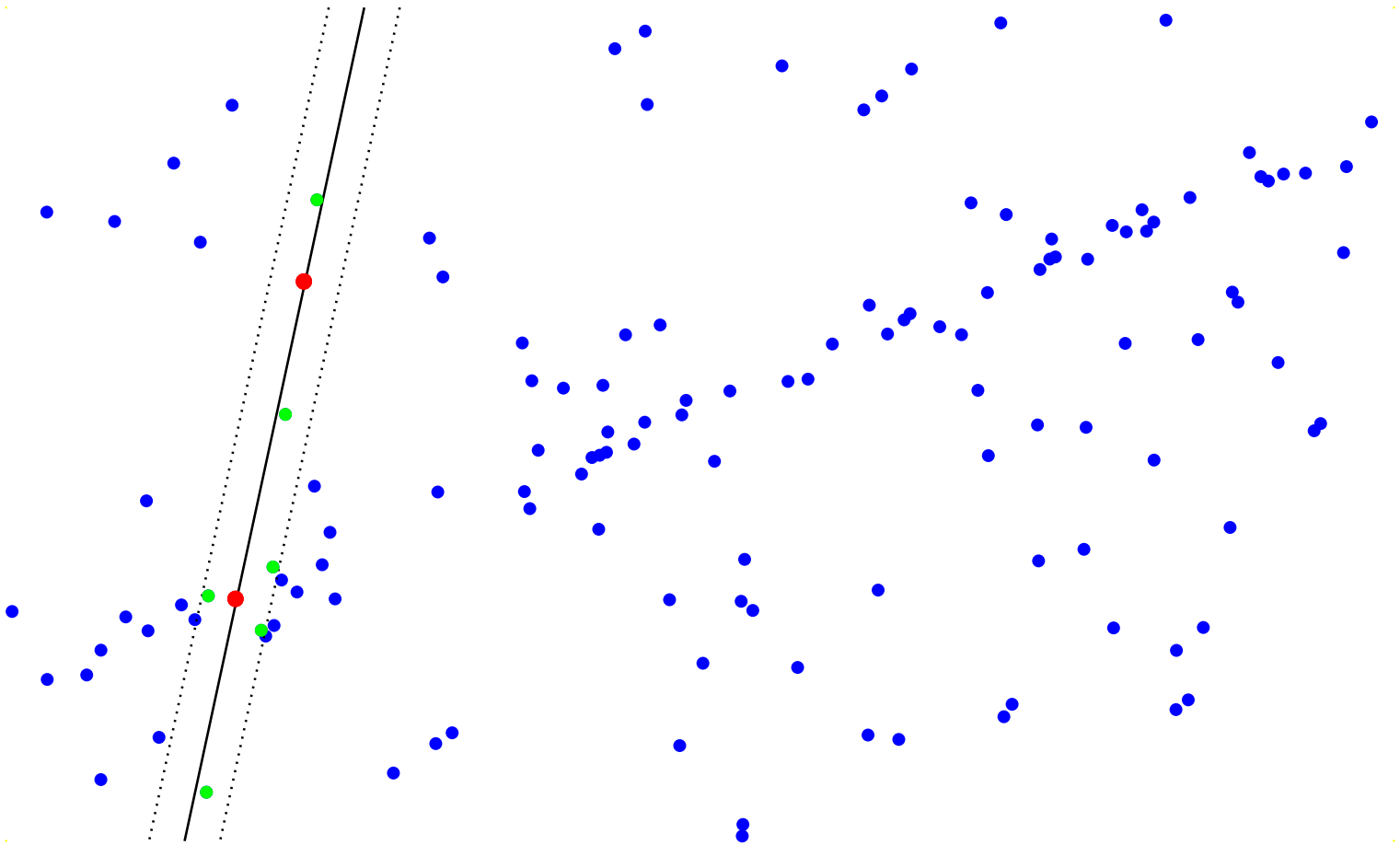
random sample consensus (RANSAC)



- data with outliers - pick two points at random - draw line through them - set margin on either side - count inlier points

Fischler and Bolles. CACM 1981. Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography.

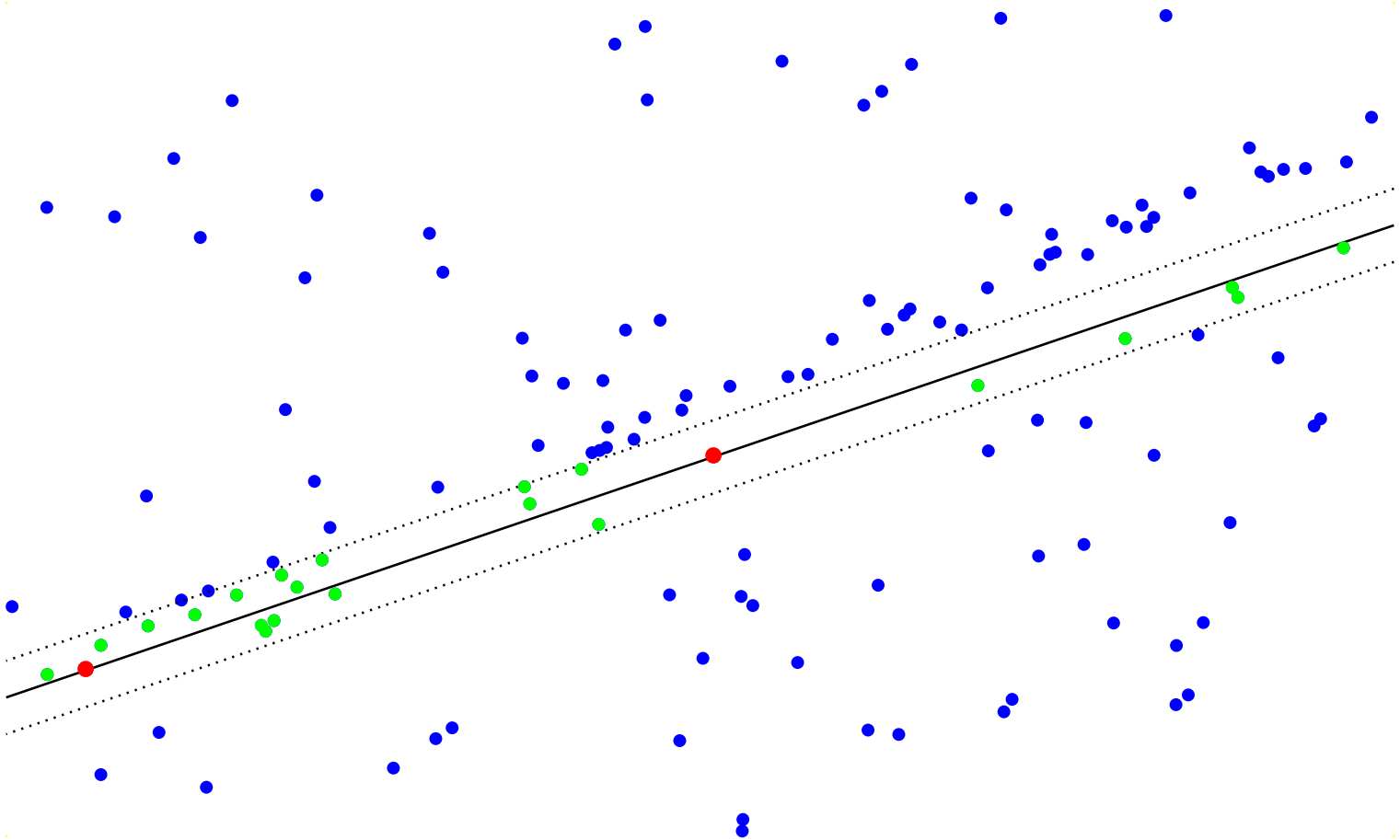
random sample consensus (RANSAC)



- data with outliers - pick two points at random - draw line through them - set margin on either side - count inlier points

Fischler and Bolles. CACM 1981. Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography.

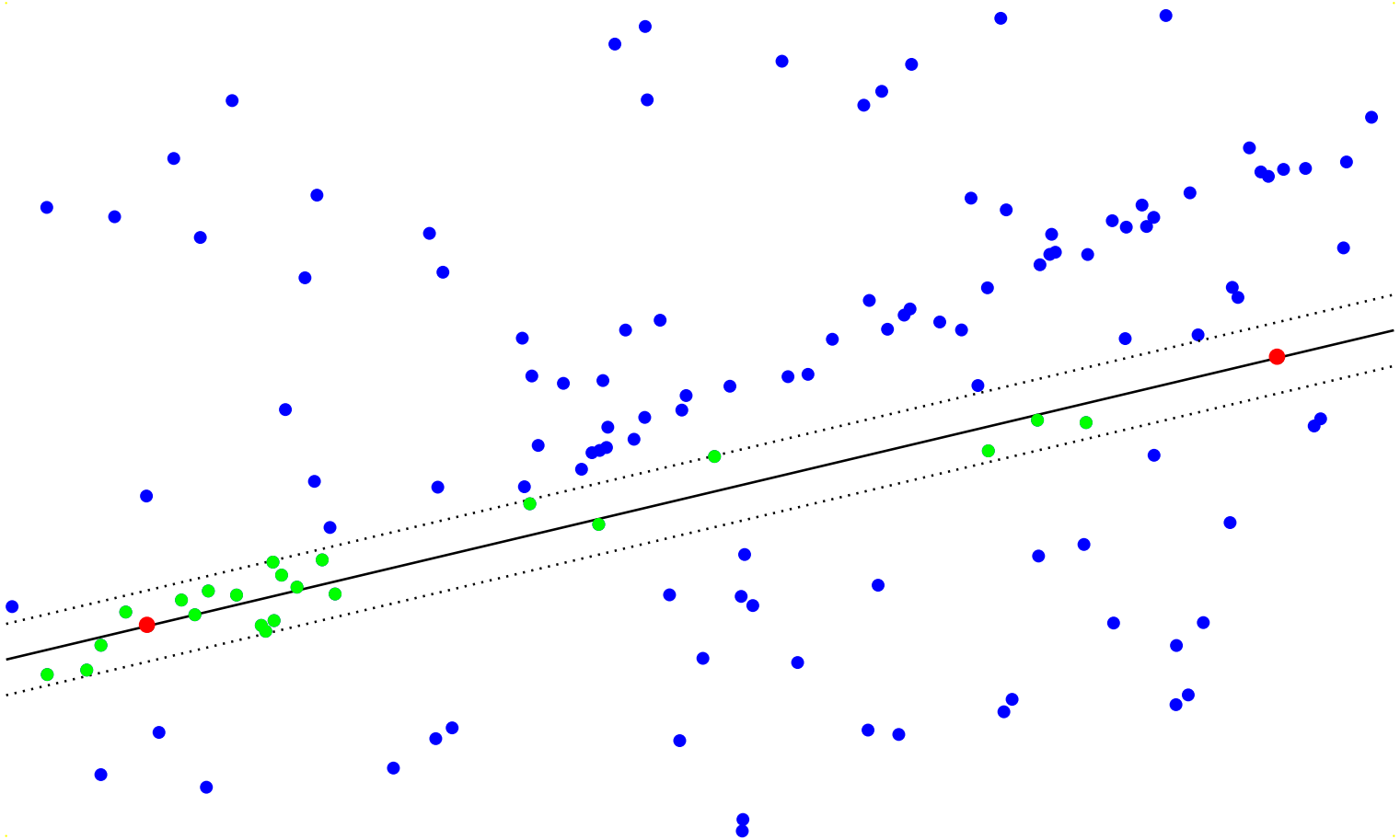
random sample consensus (RANSAC)



- repeat: pick two points at random, draw line through them, count inlier points at fixed distance to line, keep best hypothesis so far

Fischler and Bolles. CACM 1981. Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography.

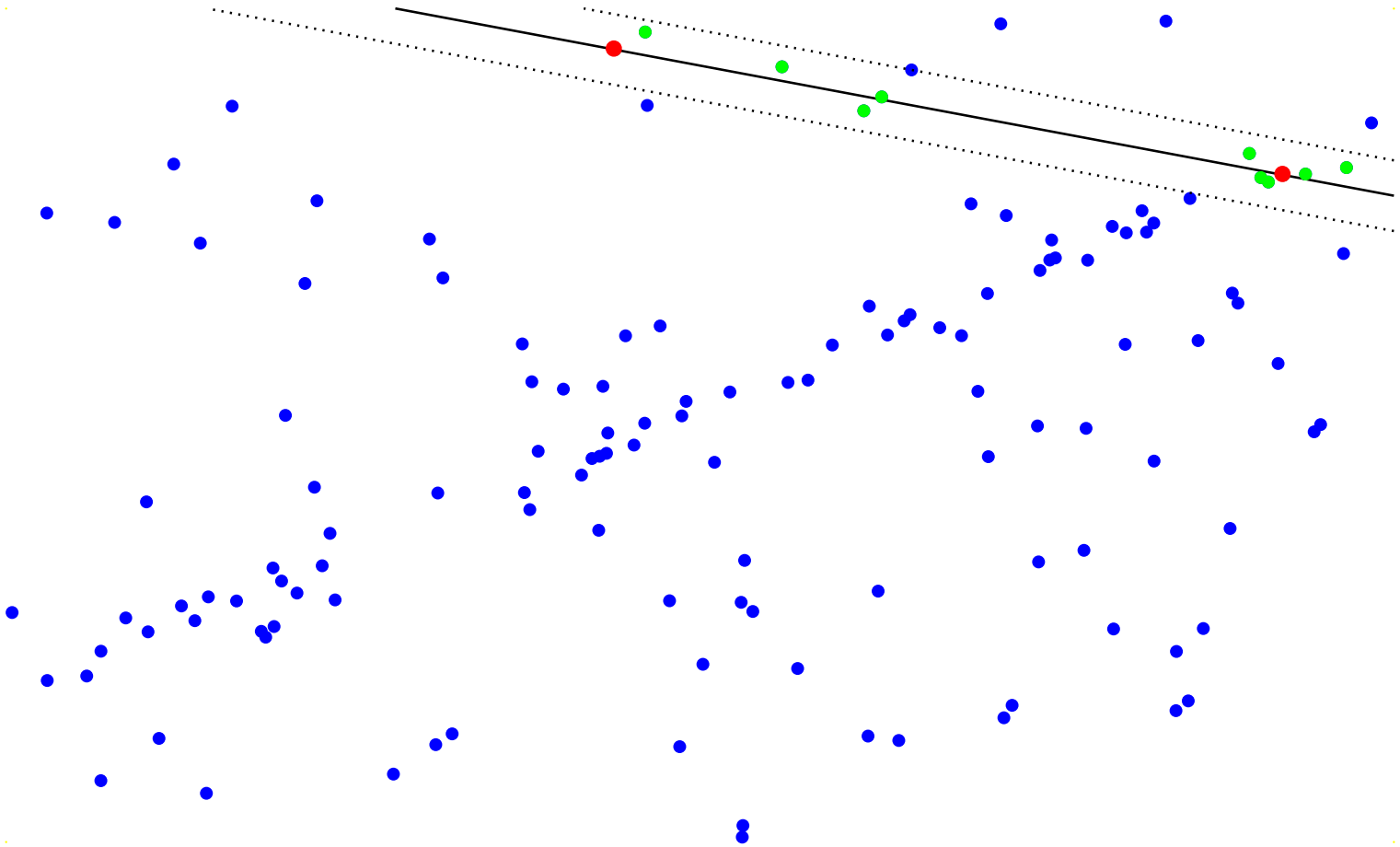
random sample consensus (RANSAC)



- repeat: pick two points at random, draw line through them, count inlier points at fixed distance to line, keep best hypothesis so far

Fischler and Bolles. CACM 1981. Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography.

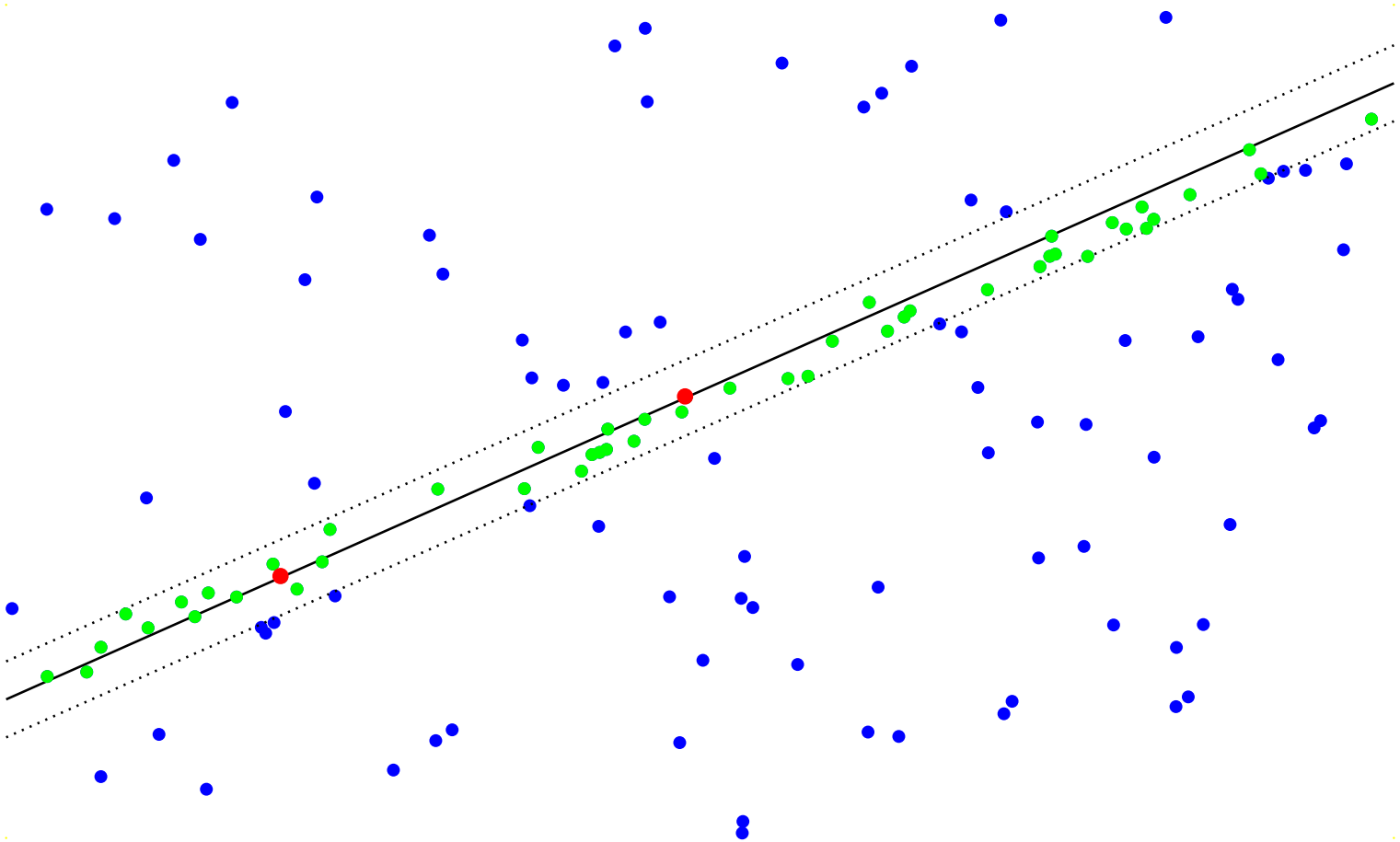
random sample consensus (RANSAC)



- repeat: pick two points at random, draw line through them, count inlier points at fixed distance to line, keep best hypothesis so far

Fischler and Bolles. CACM 1981. Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography.

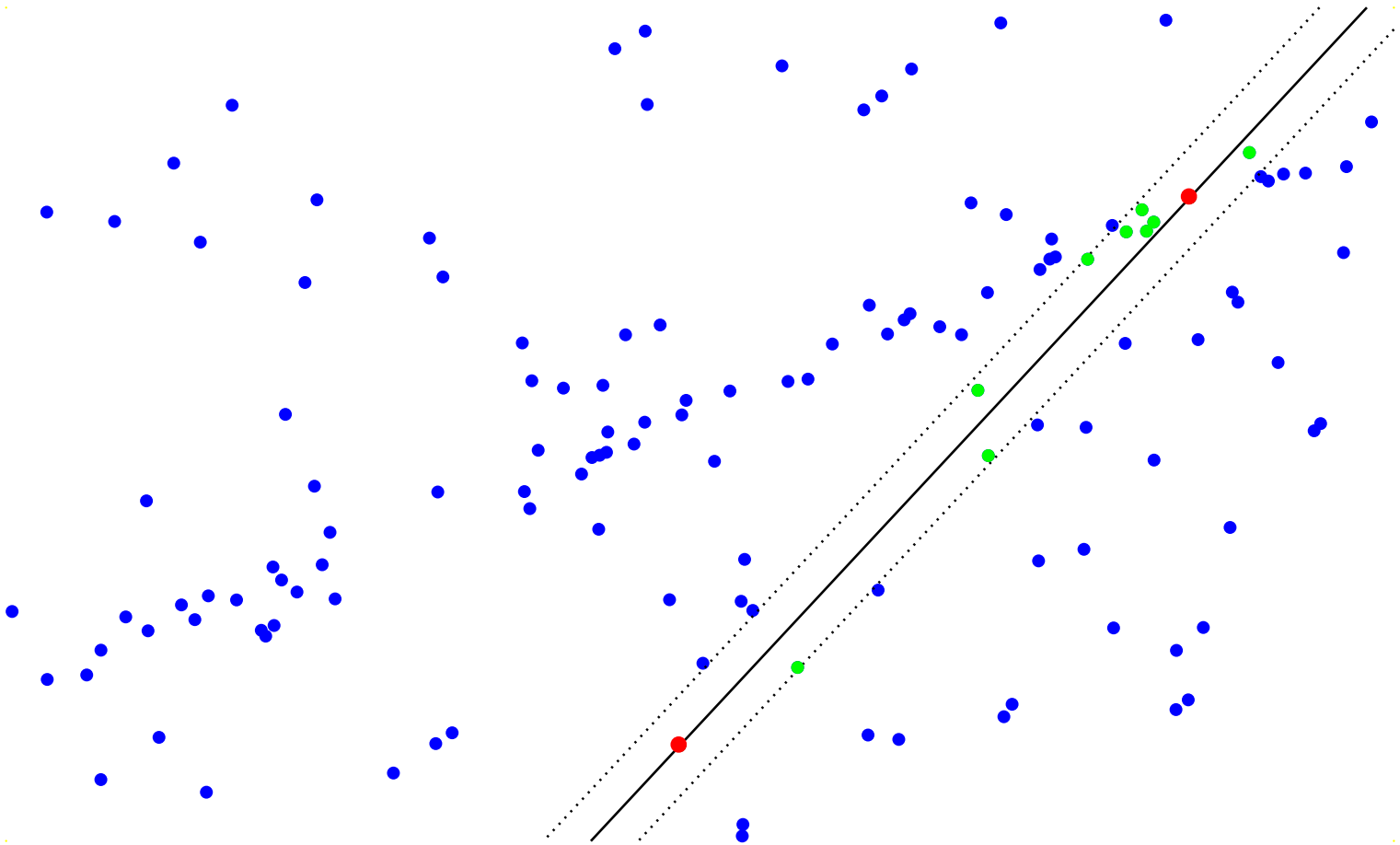
random sample consensus (RANSAC)



- repeat: pick two points at random, draw line through them, count inlier points at fixed distance to line, keep best hypothesis so far

Fischler and Bolles. CACM 1981. Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography.

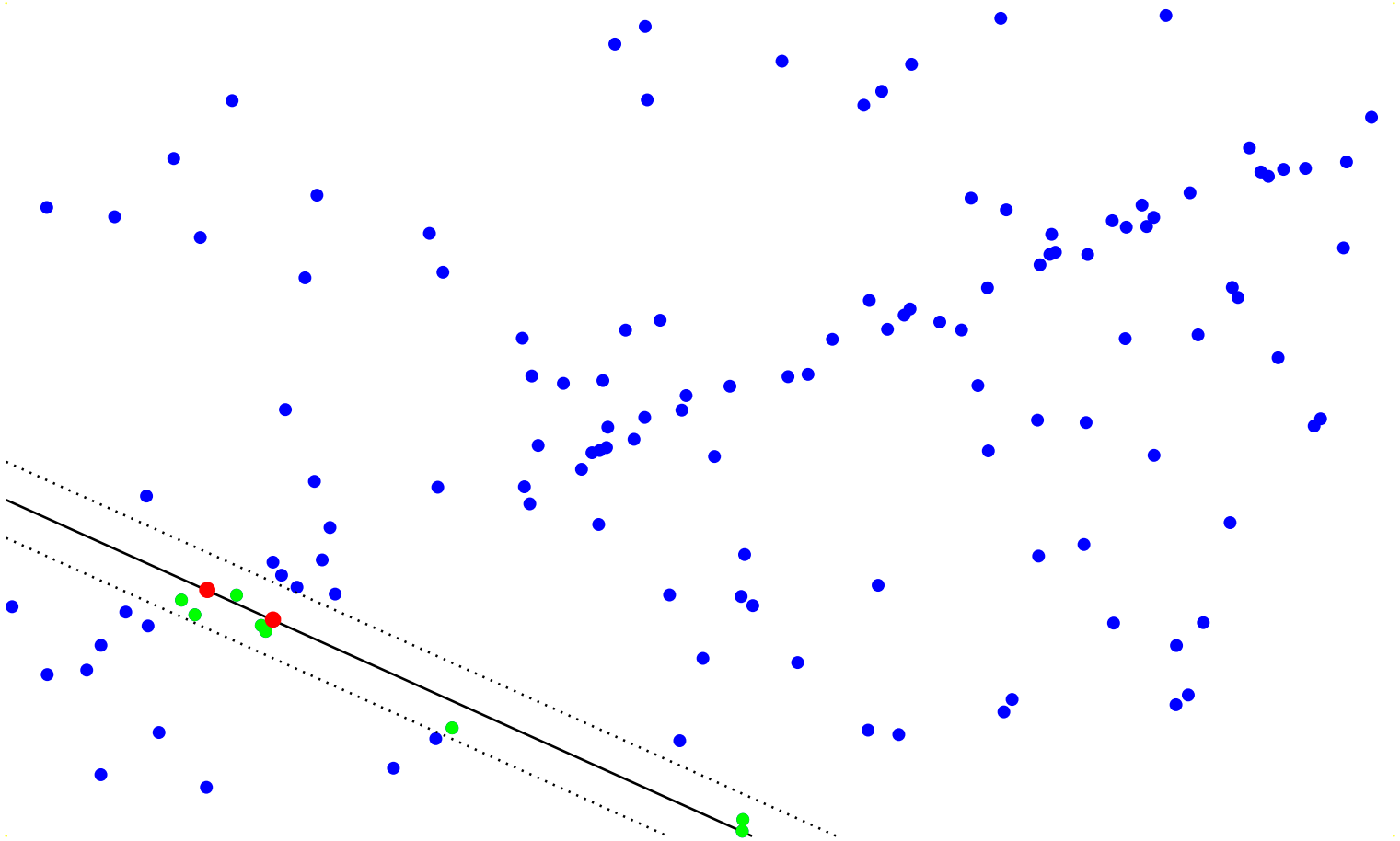
random sample consensus (RANSAC)



- repeat: pick two points at random, draw line through them, count inlier points at fixed distance to line, keep best hypothesis so far

Fischler and Bolles. CACM 1981. Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography.

random sample consensus (RANSAC)



- repeat: pick two points at random, draw line through them, count inlier points at fixed distance to line, keep best hypothesis so far

Fischler and Bolles. CACM 1981. Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography.

random sample consensus (RANSAC)

[Fischler and Bolles 1981]

- X: data (tentative correspondences)
- n: minimum number of samples to fit a model
- $s(x;\theta)$: score of sample x given model parameters θ
- repeat
  - hypothesis
    - draw n samples $H \subset X$ at random
    - fit model to H, compute parameters θ
  - verification
    - are data consistent with hypothesis? compute score $S = \sum_{x \in X} s(x;\theta)$
    - if $S^* > S$, store solution $\theta^* := \theta$, $S^* := S$

Fischler and Bolles. CACM 1981. Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography.

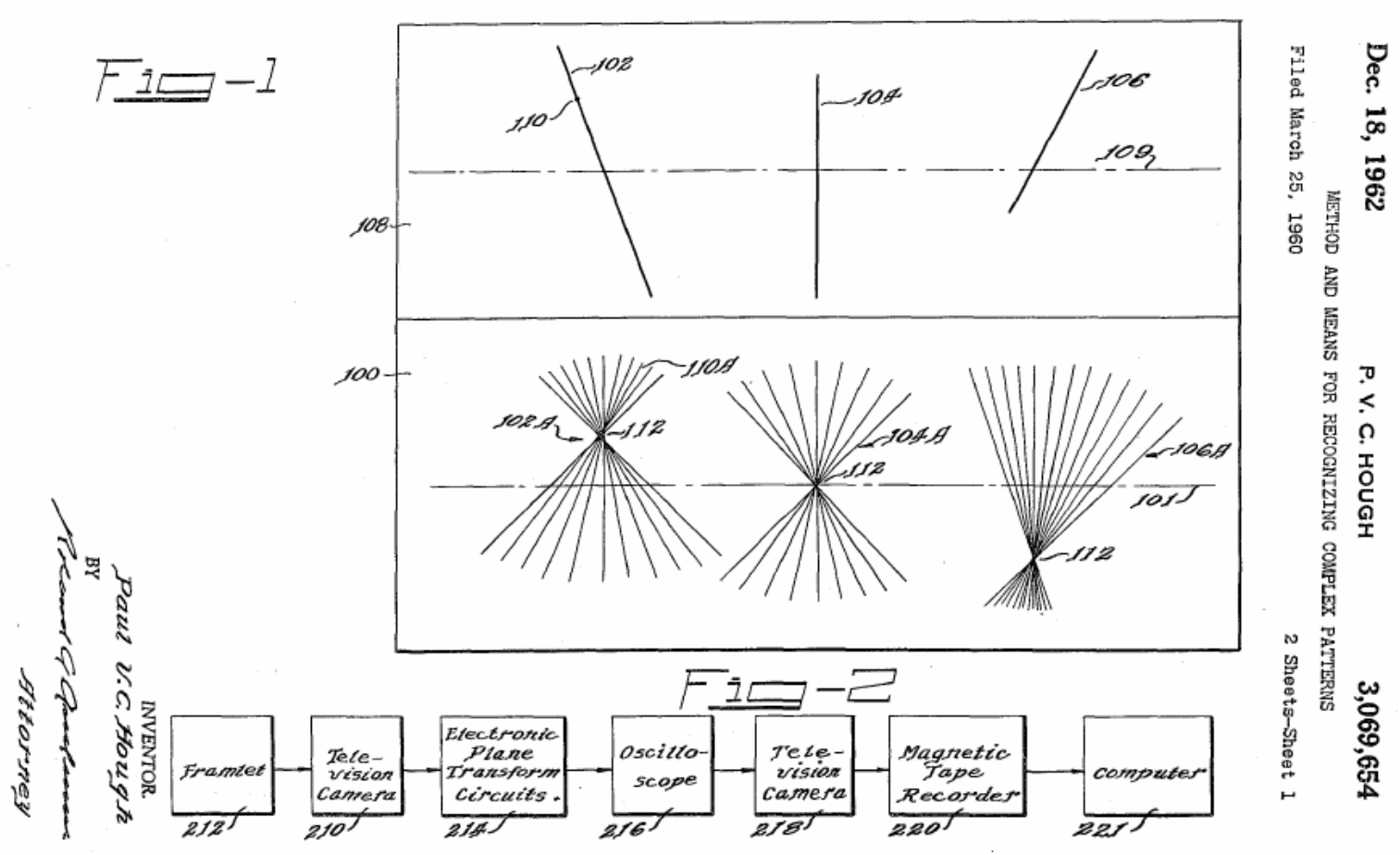
issues

- inlier ratio w unknown
- too expensive when minimum number of samples is large (*e.g.* $n > 6$) and inlier ratio is small *e.g.* $w < 10\%$): 10^6 iterations for 1% probability of failure

Fischler and Bolles. CACM 1981. Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography.

Hough transform

[Hough 1962]

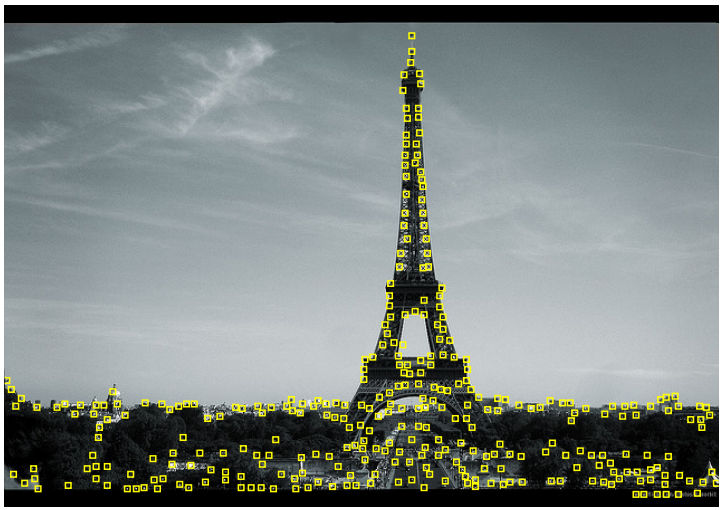


- detect lines by a voting process in parameter space
- slope-intercept parametrization unbounded for vertical lines

Hough. US Patent 1962. Method and Means for Recognizing Complex patterns.

Eiffel tower detection

model image points

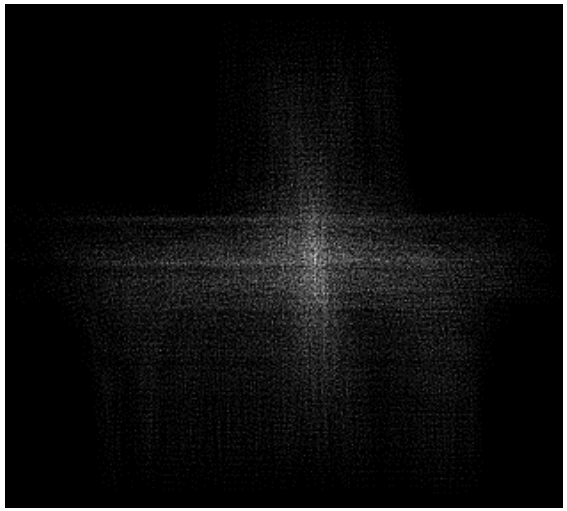


test image points

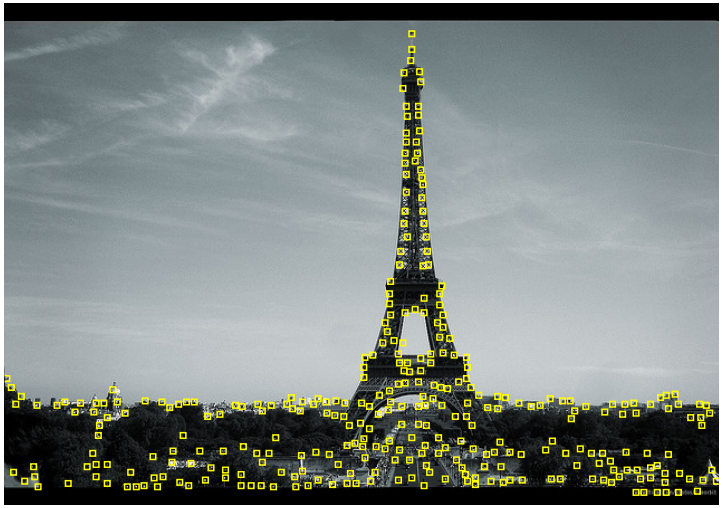
Ballard. PR 1981. Generalizing the Hough Transform to Detect Arbitrary shapes.

Eiffel tower detection

model image points



accumulator

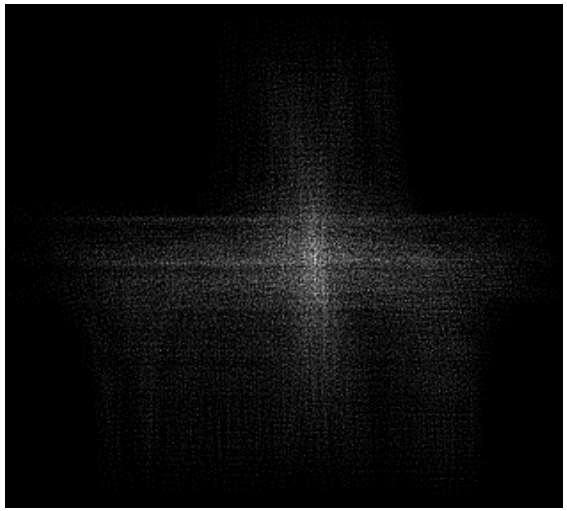


test image points

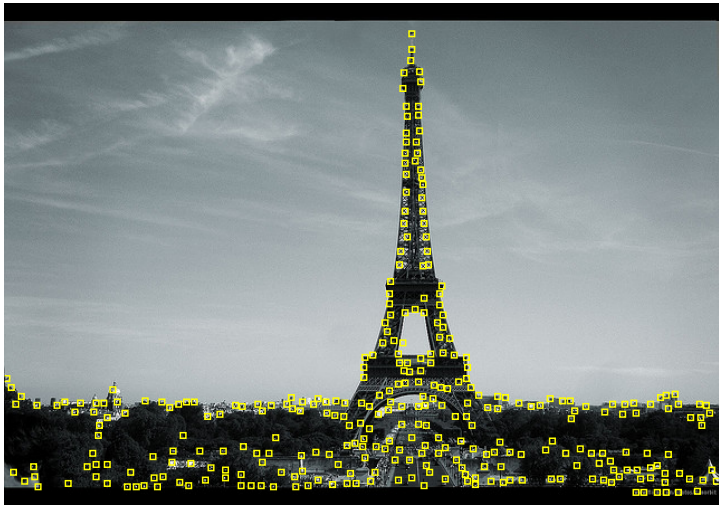
Ballard. PR 1981. Generalizing the Hough Transform to Detect Arbitrary shapes.

Eiffel tower detection

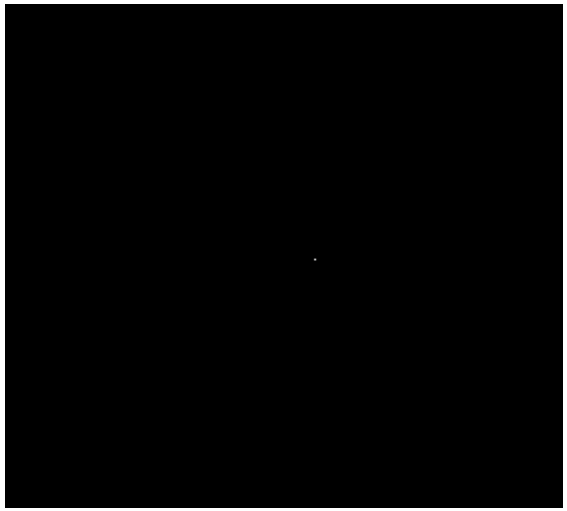
model image points



accumulator



test image points

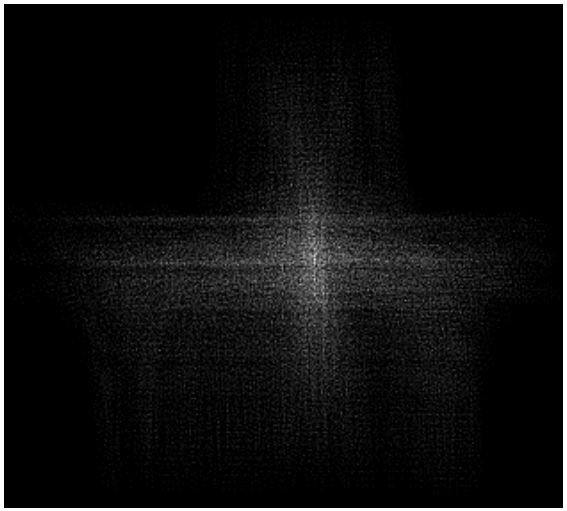


local maxima

Ballard. PR 1981. Generalizing the Hough Transform to Detect Arbitrary shapes.

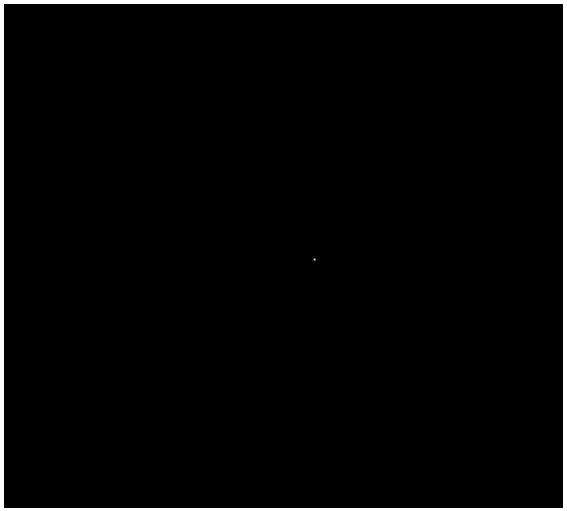
Eiffel tower detection

model image points



accumulator

detected location



local maxima

Ballard. PR 1981. Generalizing the Hough Transform to Detect Arbitrary shapes.

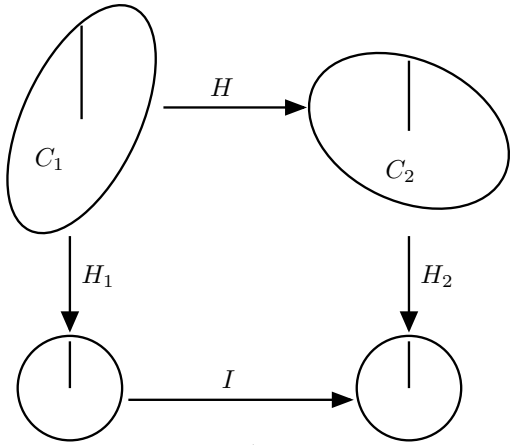
object recognition

Lowe. ICCV 1999. Object recognition from local scale-invariant features.

fast spatial matching

[Philbin et al. 2007]

Transformation	dof	Matrix
translation + isotropic scale	3	$\begin{bmatrix} a & 0 & t_x \\ 0 & a & t_y \end{bmatrix}$
translation + anisotropic scale	4	$\begin{bmatrix} a & 0 & t_x \\ 0 & b & t_y \end{bmatrix}$
translation + vertical shear	5	$\begin{bmatrix} a & 0 & t_x \\ b & c & t_y \end{bmatrix}$



- same idea, a single feature correspondence can yield a transformation that can be 3,4,5-dof
- but now use RANSAC where there is only one hypothesis per correspondence; all hypotheses can be enumerated and verified
- again, 6-dof fitting on inliers in the end
- so Hough can be seen as filtering of hypotheses by agreement

object retrieval

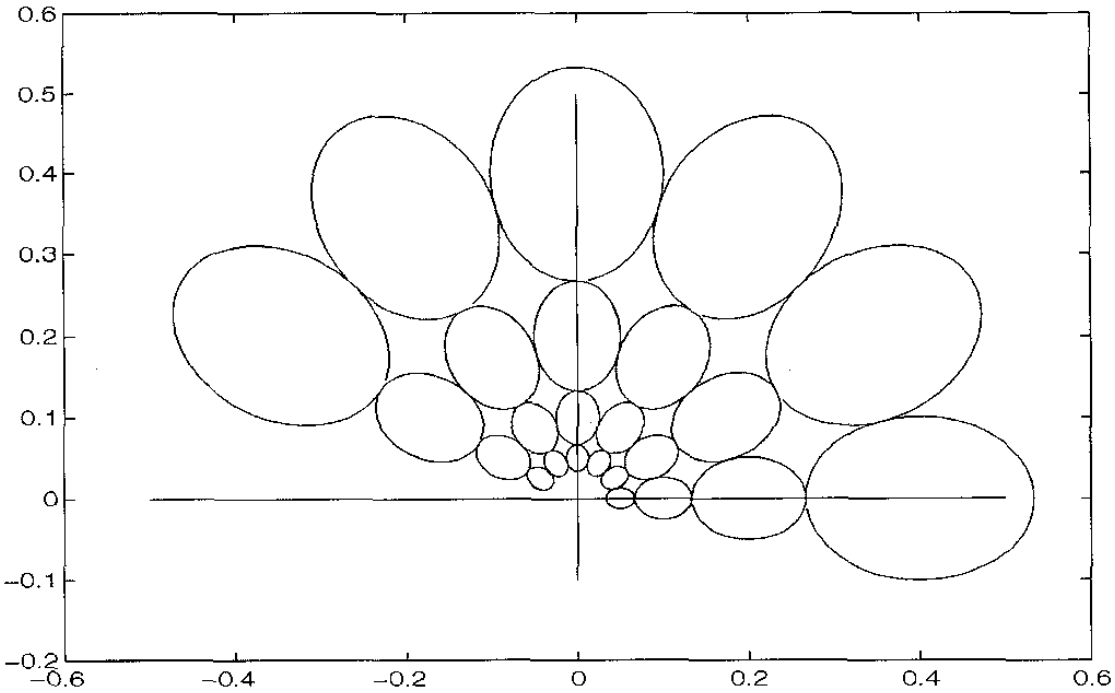
- image retrieval based on a bag-of-words representation
- fast spatial verification performed on top-ranking images

Philbin, Chum, Isard, Sivic and Zisserman. CVPR 2007. Object Retrieval With Large Vocabularies and Fast Spatial Matching.

visual descriptors

texture descriptors

[Manjunath and Ma 1996]



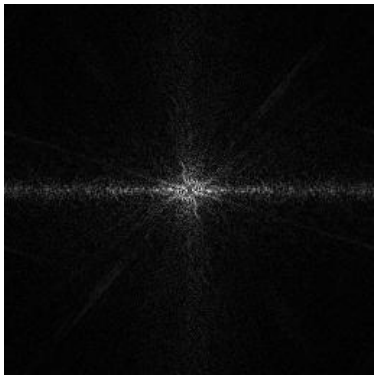
- same frequency sampling scheme
- filtering and global pooling in space domain
- popularized as part of MPEG-7 standard

Manjunath and Ma. PAMI 1996. Texture Features for Browsing and Retrieval of Image Data.

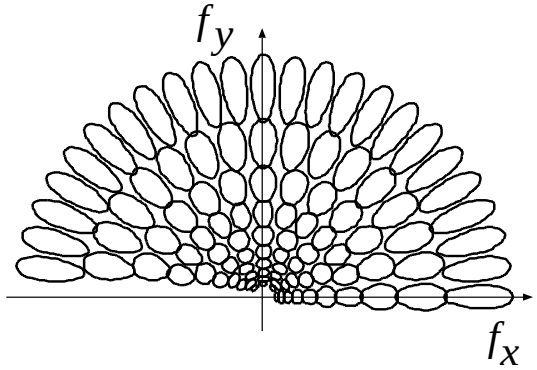
global descriptors

image

pre-processing



power spectrum

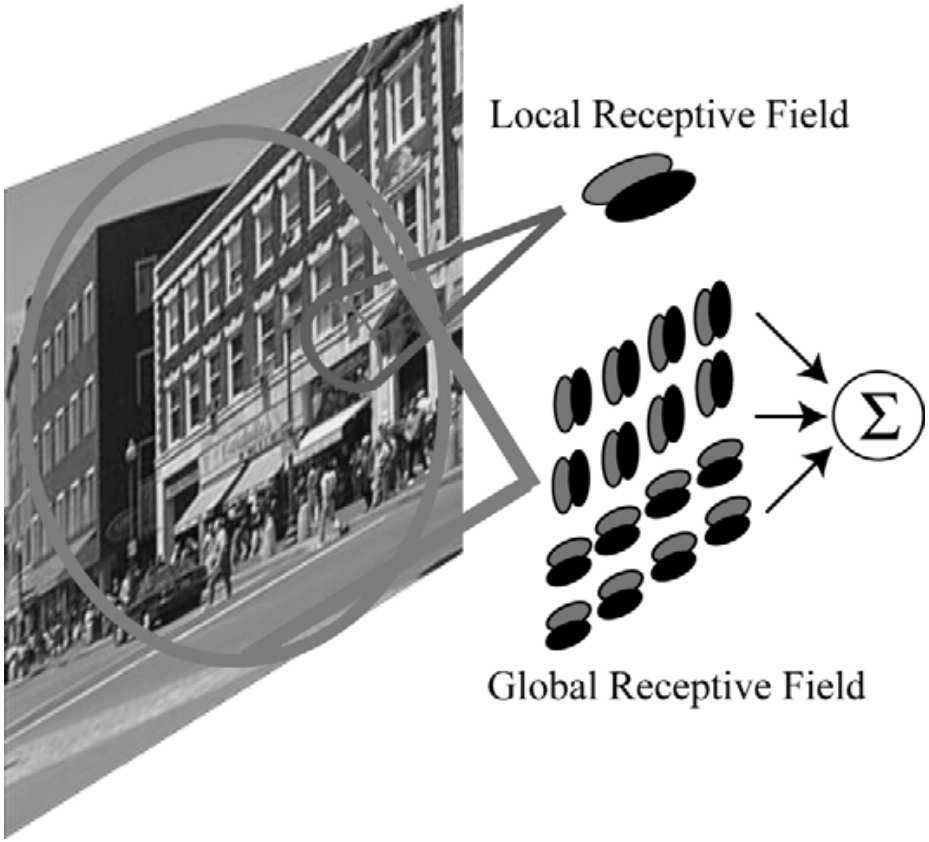


filter bank

- sampling scheme adapted to power spectrum statistics
- filtering and global pooling in frequency domain

Oliva, Torralba, Guerin-Dugue, Herault. ICCIR 1999. Global Semantic Classification of Scenes Using Power Spectrum Templates.

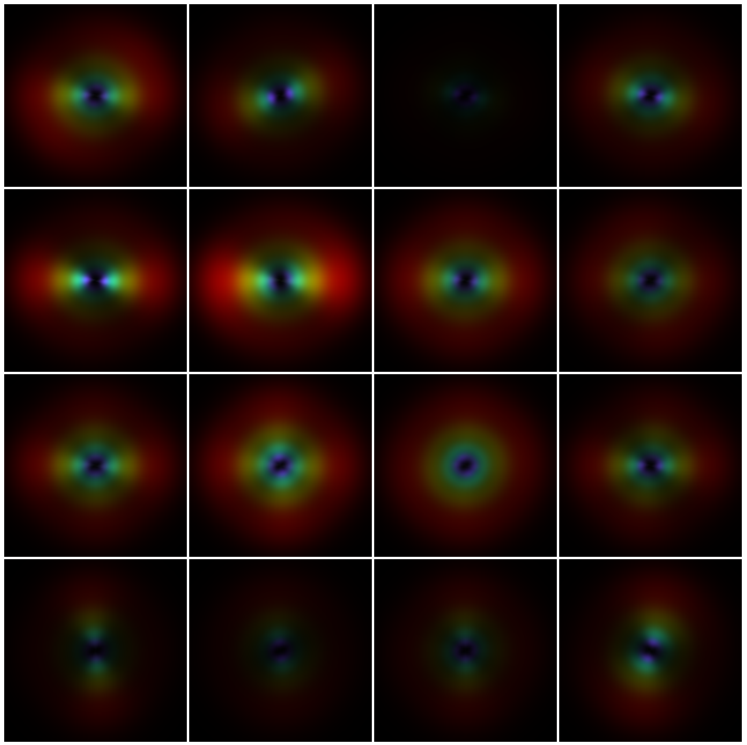
global vs. local receptive fields



- pool filter responses only locally
- next level in hierarchy can apply different spatial weights

Oliva and Torralba. VP 2006. Building the Gist of a Scene: the Role of Global Image Features in Recognition.

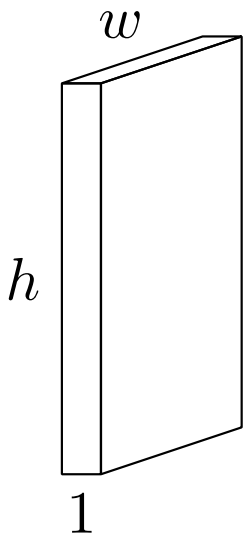
the gist descriptor



- apply filter bank to entire image in frequency domain
- partition image in 4×4 cells
- average pooling of filter responses per cell

Oliva and Torralba. VP 2006. Building the Gist of a Scene: the Role of Global Image Features in Recognition.

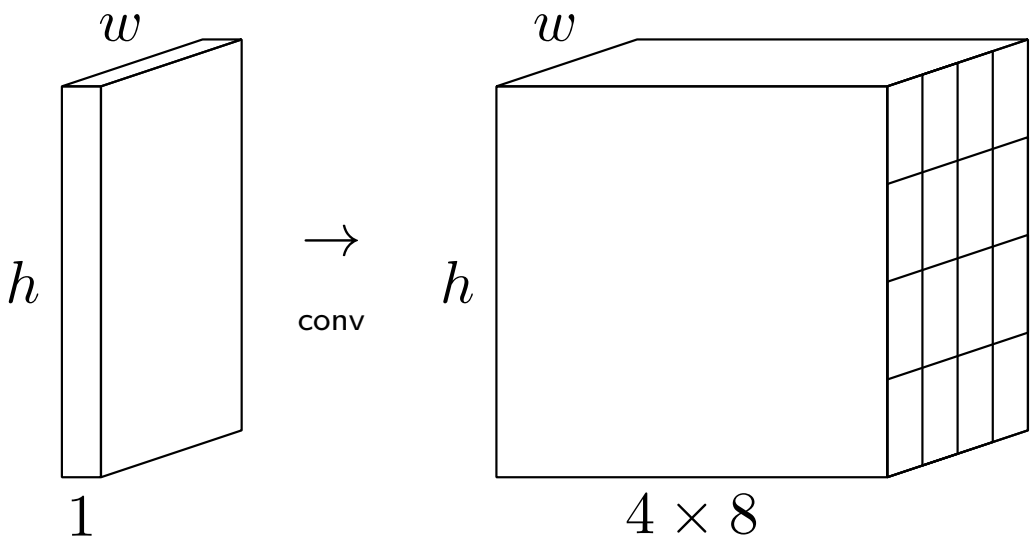
gist pipeline



- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- apply filters at 4 scales $\times$ 8 orientations
- average pooling on 4×4 cells $\rightarrow$ descriptor of length 512

Oliva and Torralba. VP 2006. Building the Gist of a Scene: the Role of Global Image Features in Recognition.

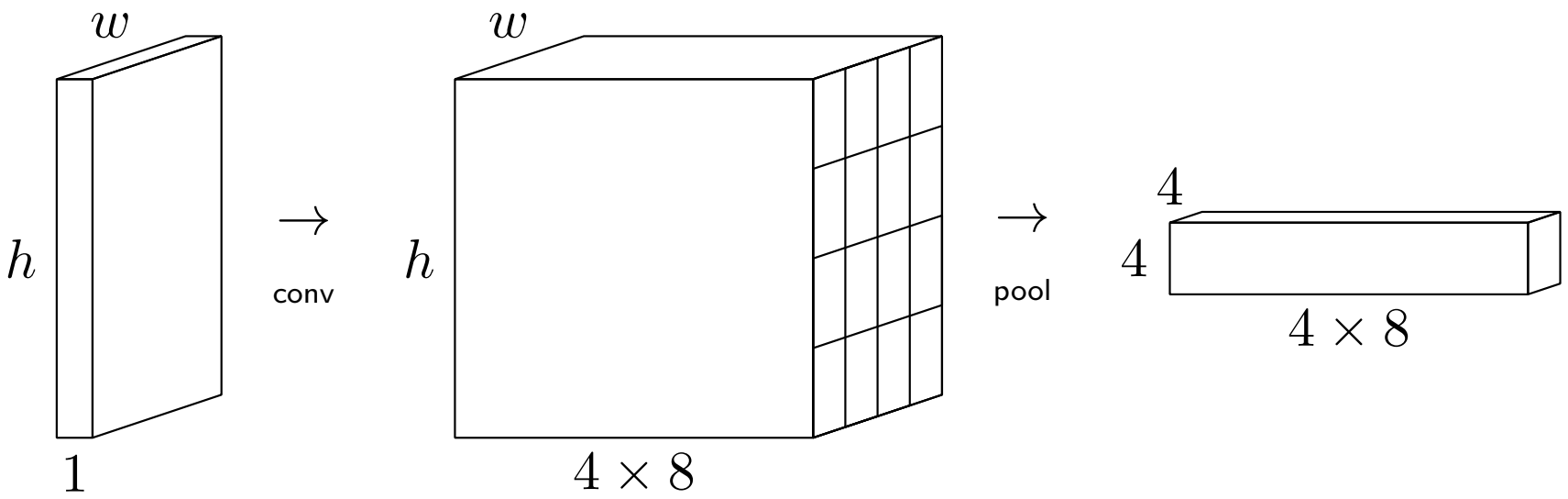
gist pipeline



- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- apply filters at 4 scales $\times$ 8 orientations
- average pooling on 4×4 cells $\rightarrow$ descriptor of length 512

Oliva and Torralba. VP 2006. Building the Gist of a Scene: the Role of Global Image Features in Recognition.

gist pipeline

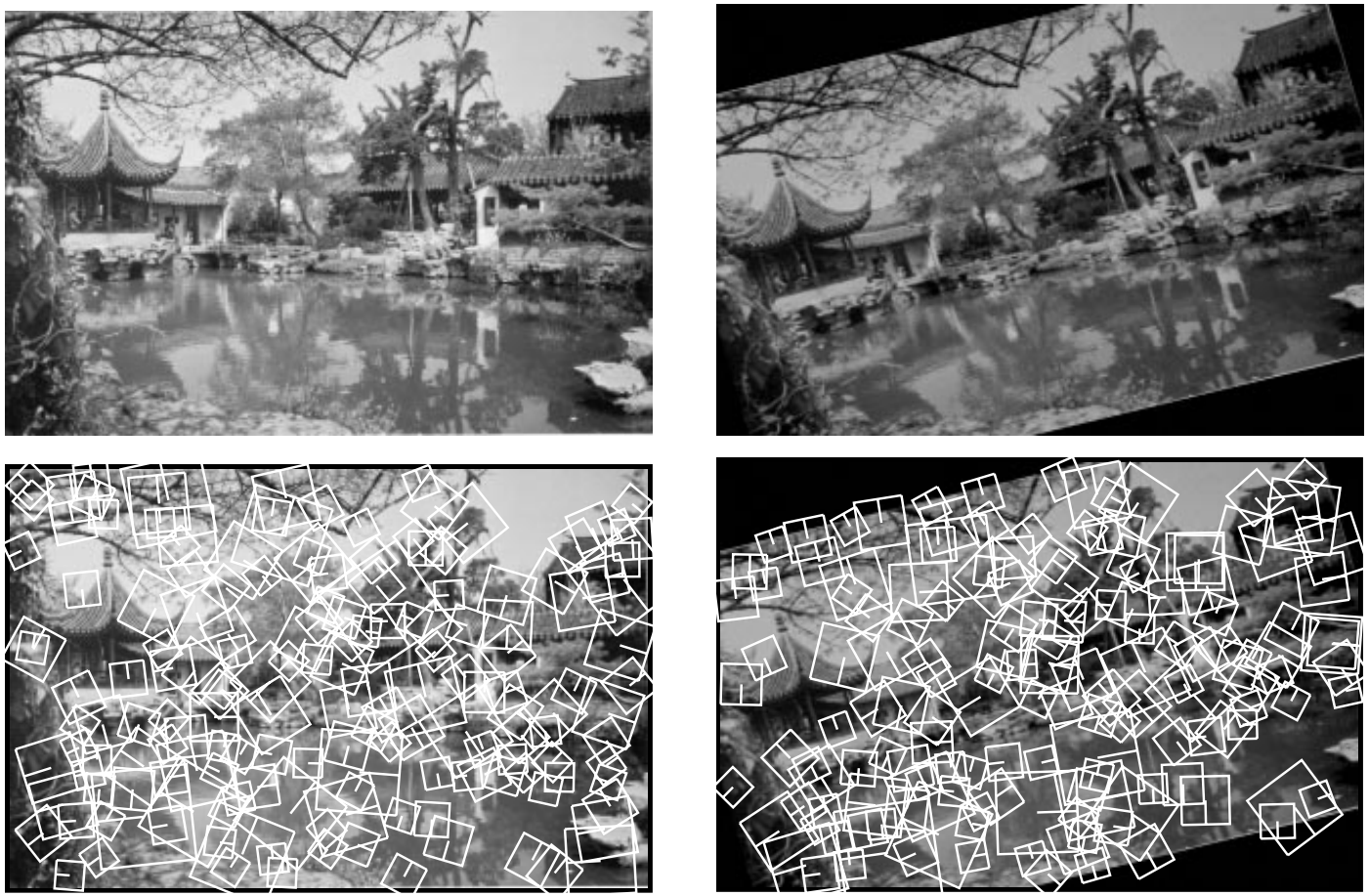


- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- apply filters at 4 scales $\times$ 8 orientations
- average pooling on 4×4 cells $\rightarrow$ descriptor of length 512

Oliva and Torralba. VP 2006. Building the Gist of a Scene: the Role of Global Image Features in Recognition.

scale-invariant feature transform

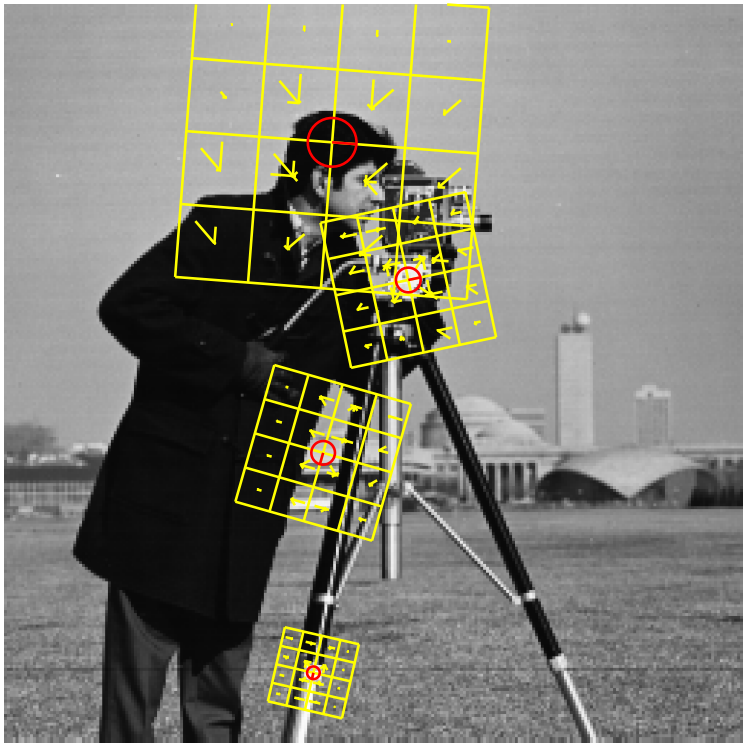
[Lowe 1999]

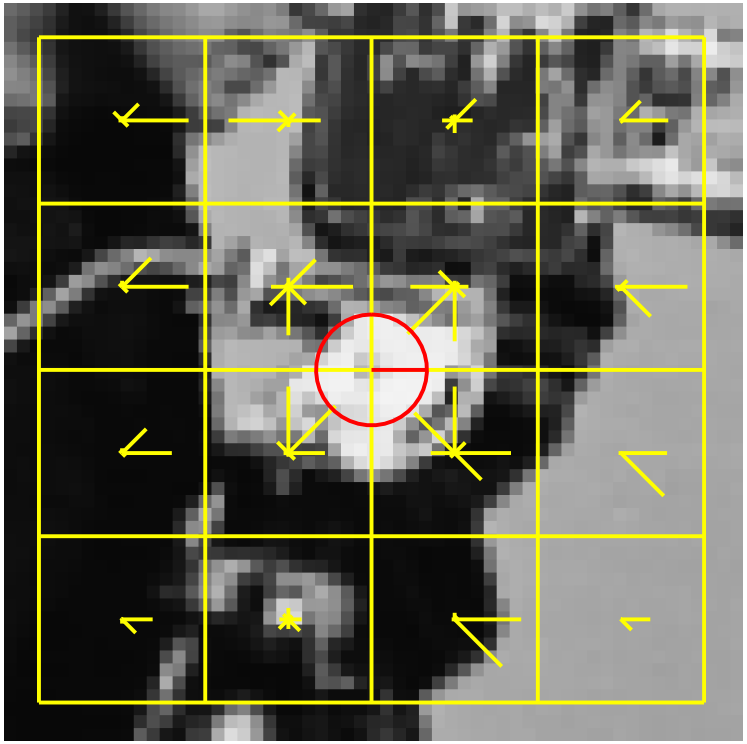


- detect a sparse set of "stable" features (rectangular patches), equivariant to translation, scale and rotation

Lowe. ICCV 1999. Object recognition from local scale-invariant features.

scale-invariant feature transform

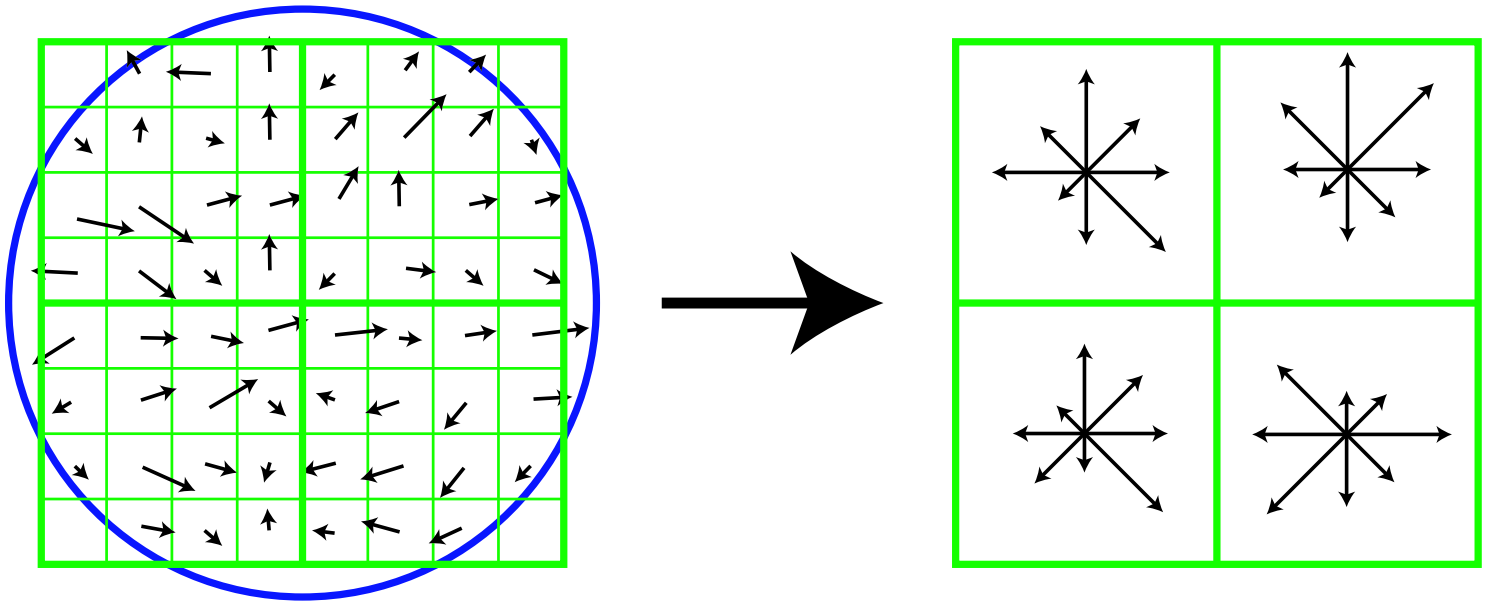




- for each patch
  - normalize with respect to scale and orientation
  - construct a histogram of gradient orientations

Lowe. ICCV 1999. Object recognition from local scale-invariant features.

the SIFT descriptor

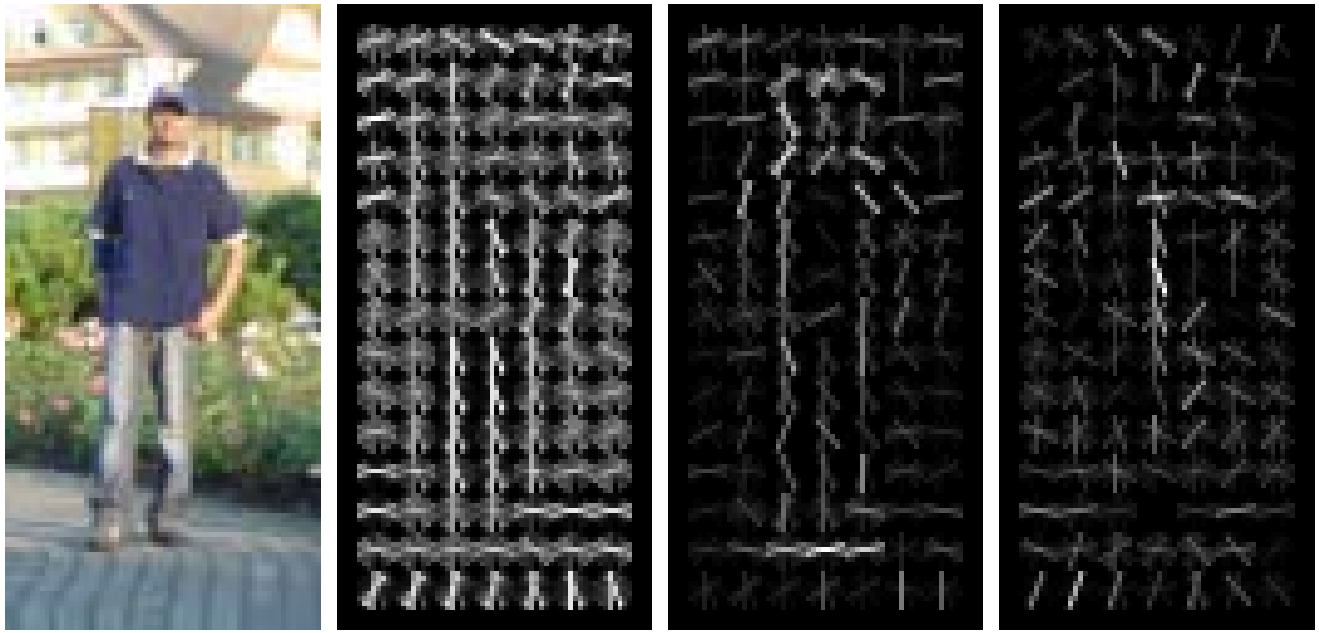


- votes in 8-bin orientation histograms weighted by magnitude and by Gaussian window on patch
- histograms pooled over 4×4 cells, trilinear interpolation
- 128-dimensional descriptor, normalized, clipped at 0.2, normalized

Lowe. ICCV 1999. Object recognition from local scale-invariant features.

histogram of oriented gradients

[Dalal and Triggs 2005]



- applied to person detection by sliding window and SVM
- classifier learns positive and negative weights on positions and orientations
- switch focus back to dense features for classification

Dalal and Triggs. CVPR 2005. Histograms of Oriented Gradients for Human Detection.

the HOG descriptor

- applied densely to adjacent cells of 8×8 pixels
- no scale or orientation normalization; just single-scale
- normalized by overlapping blocks of 3×3 cells—redundant

Dalal and Triggs. CVPR 2005. Histograms of Oriented Gradients for Human Detection.

so what is a histogram?

- consider a histogram h over integers $C = \{0, 1, 2, 3, 4\}$, computed from the following samples:

C	$=$	$\{$	0	1	2	3	4	$\}$		
3	$\rightarrow$	$($	0	0	0	1	0	$)$		
2	$\rightarrow$	$($	0	0	1	0	0	$)$		
0	$\rightarrow$	$($	1	0	0	0	0	$)$		
3	$\rightarrow$	$($	0	0	0	1	0	$)$		
2	$\rightarrow$	$($	0	0	1	0	0	$)$		
2	$\rightarrow$	$($	0	0	1	0	0	$)$	$+$	
h	$=$	$($	1	0	3	2	0	$)$	$/$	6

- each sample is encoded (*hard-assigned*) into a vector in $\mathbb{R}^5$; all such vectors are pooled (*averaged*) into one vector $h \in \mathbb{R}^5$
- encoding is always nonlinear and pooling is orderless
- C is a codebook or vocabulary

so what is a histogram?

- consider a histogram h over integers $C = \{0, 1, 2, 3, 4\}$, computed from the following samples:

C	$=$	$\{$	0	1	2	3	4	$\}$		
3	$\rightarrow$	$($	0	0	0	1	0	$)$		
2	$\rightarrow$	$($	0	0	1	0	0	$)$		
0	$\rightarrow$	$($	1	0	0	0	0	$)$		
3	$\rightarrow$	$($	0	0	0	1	0	$)$		
2	$\rightarrow$	$($	0	0	1	0	0	$)$		
2	$\rightarrow$	$($	0	0	1	0	0	$)$		
2	$\rightarrow$	$($	0	0	1	0	0	$)$	$+$	
h	$=$	$($	1	0	3	2	0	$)$	$/$	6

- each sample is encoded (*hard-assigned*) into a vector in $\mathbb{R}^5$; all such vectors are pooled (*averaged*) into one vector $h \in \mathbb{R}^5$
- encoding is always nonlinear and pooling is orderless
- C is a codebook or vocabulary

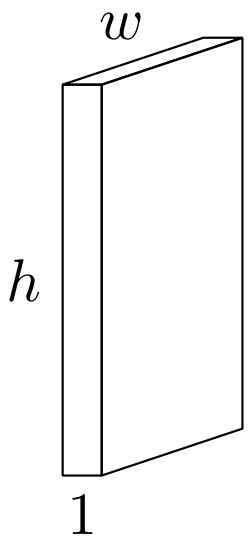
so what is a histogram?

- consider a histogram h over integers $C = \{0, 1, 2, 3, 4\}$, computed from the following samples:

C	$=$	$\{$	0	1	2	3	4	$\}$		
3	$\rightarrow$	$($	0	0	0	1	0	$)$		
2	$\rightarrow$	$($	0	0	1	0	0	$)$		
0	$\rightarrow$	$($	1	0	0	0	0	$)$		
3	$\rightarrow$	$($	0	0	0	1	0	$)$		
2	$\rightarrow$	$($	0	0	1	0	0	$)$		
2	$\rightarrow$	$($	0	0	1	0	0	$)$	$+$	
h	$=$	$($	1	0	3	2	0	$)$	$/$	6

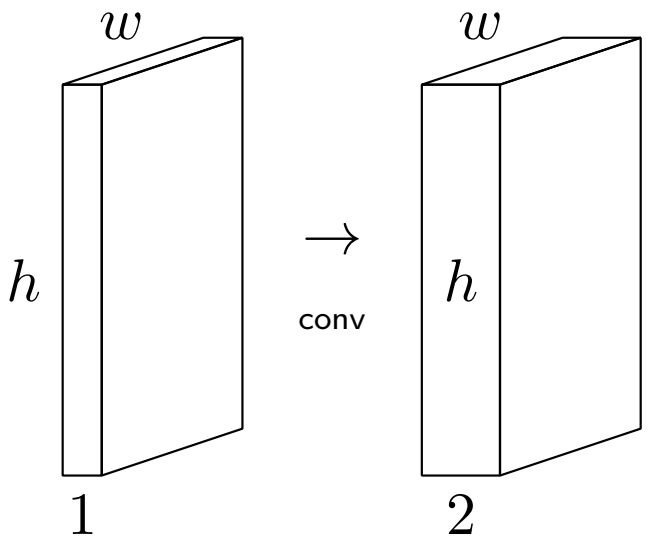
- each sample is encoded (*hard-assigned*) into a vector in $\mathbb{R}^5$; all such vectors are pooled (*averaged*) into one vector $h \in \mathbb{R}^5$
- encoding is always nonlinear and pooling is orderless
- C is a codebook or vocabulary

SIFT (HOG) pipeline



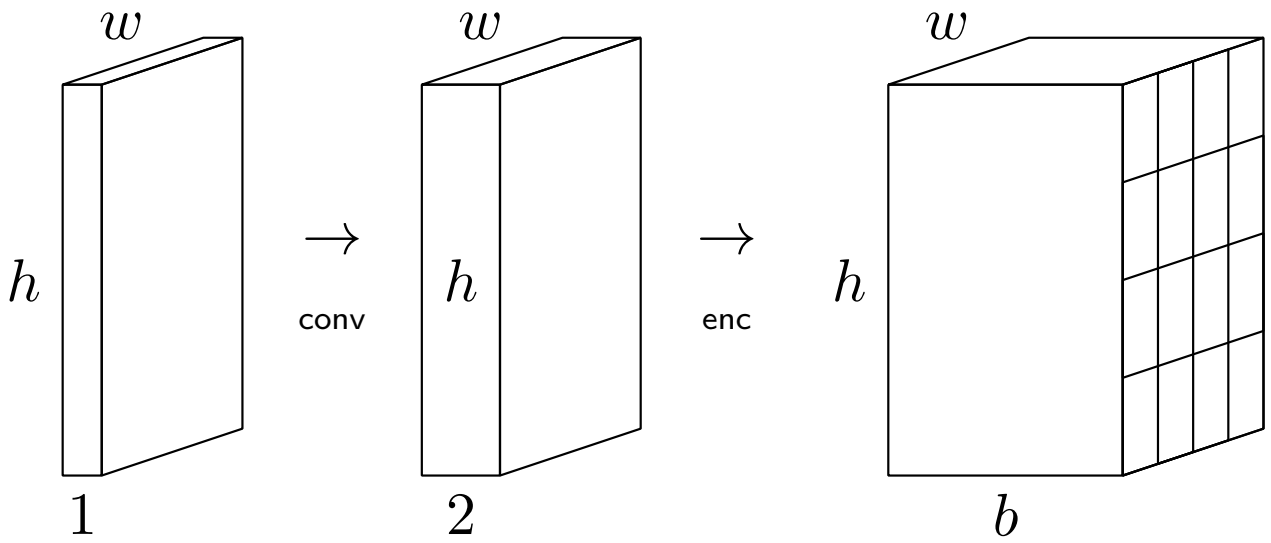
- 3-channel patch (image) RGB input $\rightarrow$ 1-channel gray-scale
- compute gradient magnitude & orientation
- encode into $b = 8$ (9) orientation bins
- average pooling on $c = 4 \times 4$ $(\lfloor w/8 \rfloor \times \lfloor h/8 \rfloor)$ cells
- descriptor of length $c \times b = 128$ (block-normalize $\rightarrow c \times (3 \times 3) \times b$)

SIFT (HOG) pipeline



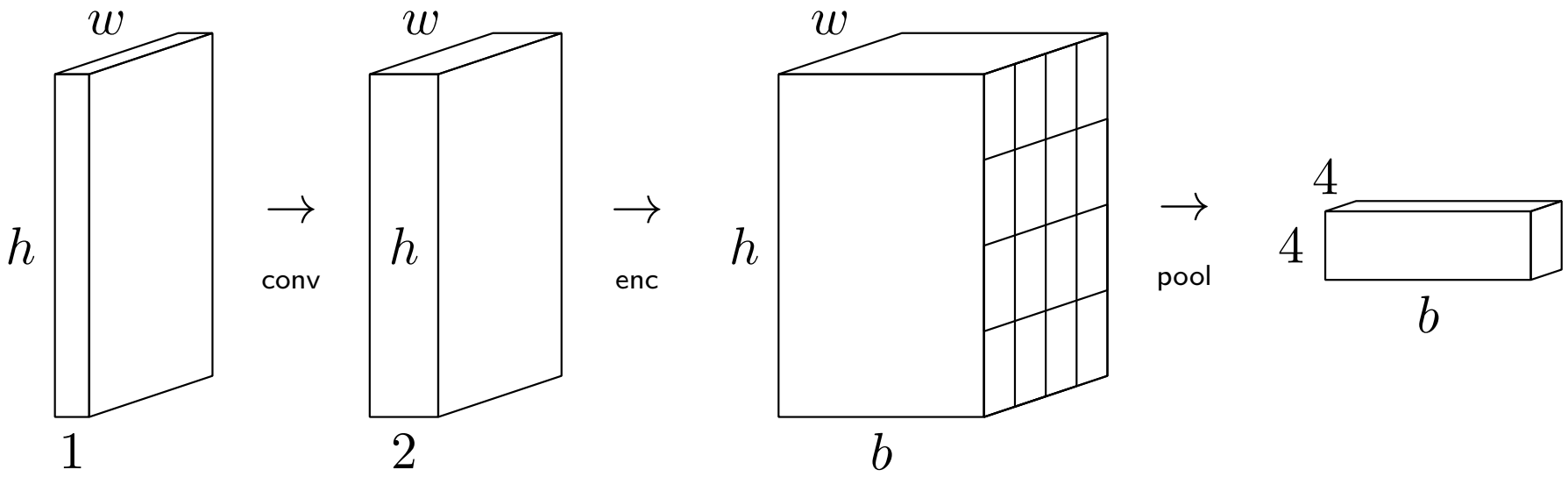
- 3-channel patch (image) RGB input $\rightarrow$ 1-channel gray-scale
- compute gradient magnitude & orientation
- encode into $b = 8$ (9) orientation bins
- average pooling on $c = 4 \times 4$ $(\lfloor w/8 \rfloor \times \lfloor h/8 \rfloor)$ cells
- descriptor of length $c \times b = 128$ (block-normalize $\rightarrow c \times (3 \times 3) \times b$)

SIFT (HOG) pipeline



- 3-channel patch (image) RGB input $\rightarrow$ 1-channel gray-scale
- compute gradient magnitude & orientation
- encode into $b = 8$ (9) orientation bins
- average pooling on $c = 4 \times 4$ ($\lfloor w/8 \rfloor \times \lfloor h/8 \rfloor$) cells
- descriptor of length $c \times b = 128$ (block-normalize $\rightarrow c \times (3 \times 3) \times b$)

SIFT (HOG) pipeline



- 3-channel patch (image) RGB input $\rightarrow$ 1-channel gray-scale
- compute gradient magnitude & orientation
- encode into $b = 8$ (9) orientation bins
- average pooling on $c = 4 \times 4$ ($\lfloor w/8 \rfloor \times \lfloor h/8 \rfloor$) cells
- descriptor of length $c \times b = 128$ (block-normalize $\rightarrow c \times (3 \times 3) \times b$)

bag of words (BoW)

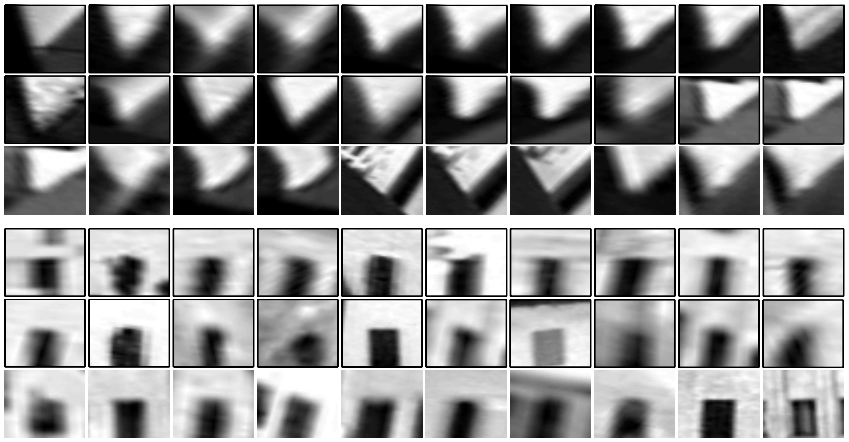
[Sivic and Zisserman 2003]

- two types of sparse features detected
- SIFT descriptors extracted from a dataset of video frames

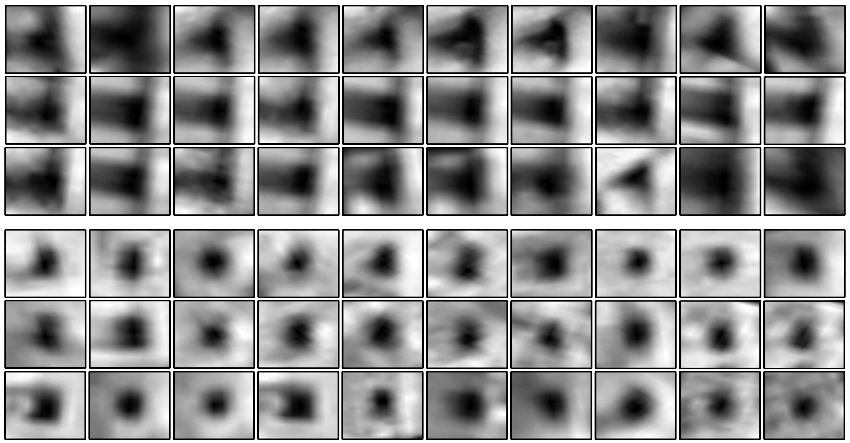
Sivic and Zisserman. ICCV 2003. Video Google: A Text Retrieval Approach to Object Matching in videos.

bag of words: retrieval

[Sivic and Zisserman 2003]



Harris affine
6k words



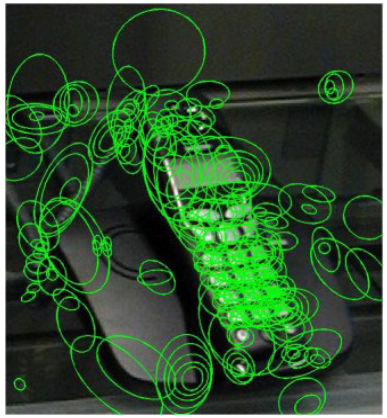
maximally stable
10k words

- "visual words" defined as clusters of SIFT descriptors learned from the dataset
- images described by visual word histograms
- matching is reduced to sparse dot product $\rightarrow$ fast retrieval

Sivic and Zisserman. ICCV 2003. Video Google: A Text Retrieval Approach to Object Matching in videos.

bag of words: classification

[Csurka et al. 2004]



features

visual words

phones, books, cars

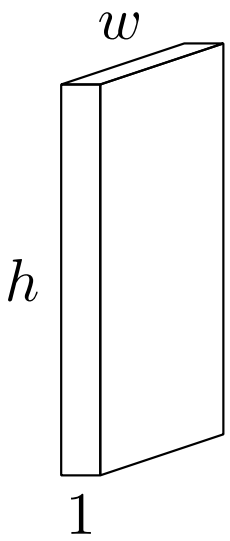
bikes, buildings, cars

buildings, cars, faces

- same representation, $k = 1000$ words, naive bayes or SVM classifier
- features soon to be replaced dense multiscale HOG or SIFT

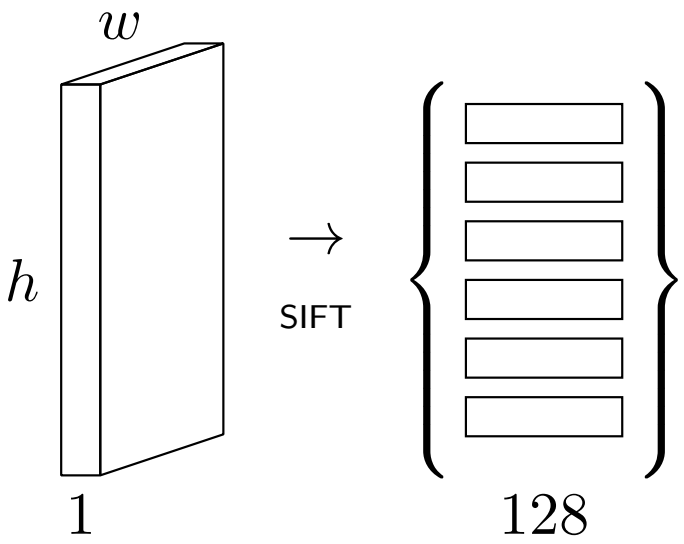
Csurka, Dance, Fan, Willamowski and Bray. SLCV 2004. Visual Categorization With Bags of Keypoints.

bag of words pipeline



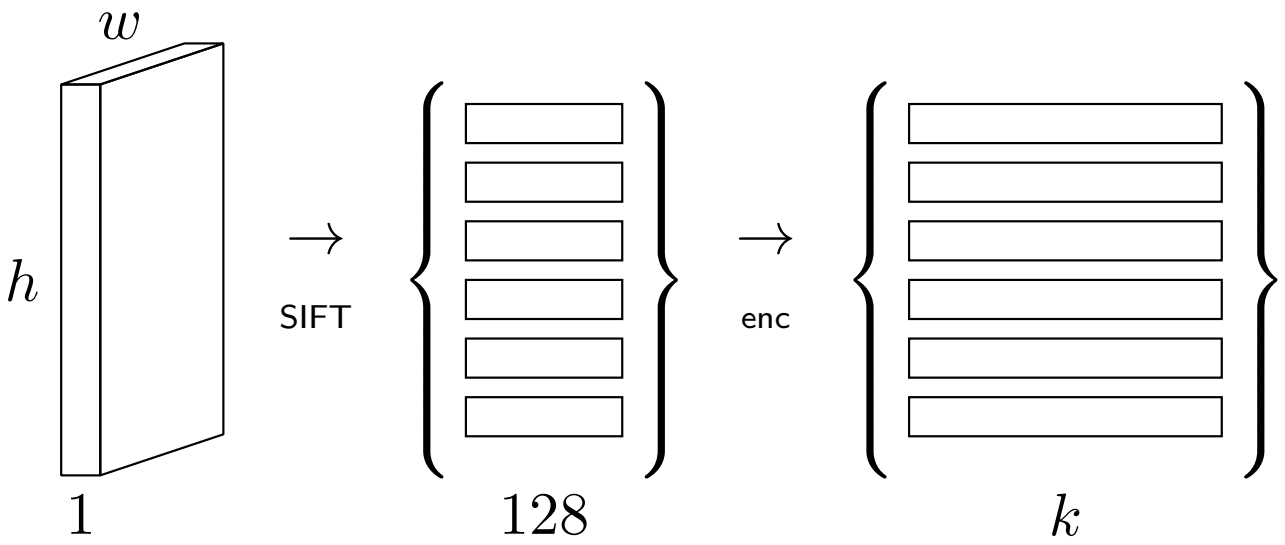
- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- set of ~ 1000 features $\times$ 128-dim SIFT descriptors
- element-wise encoding (hard assignment) on $k \sim 10^4$ visual words
- global sum pooling, ℓ^2 normalization

bag of words pipeline



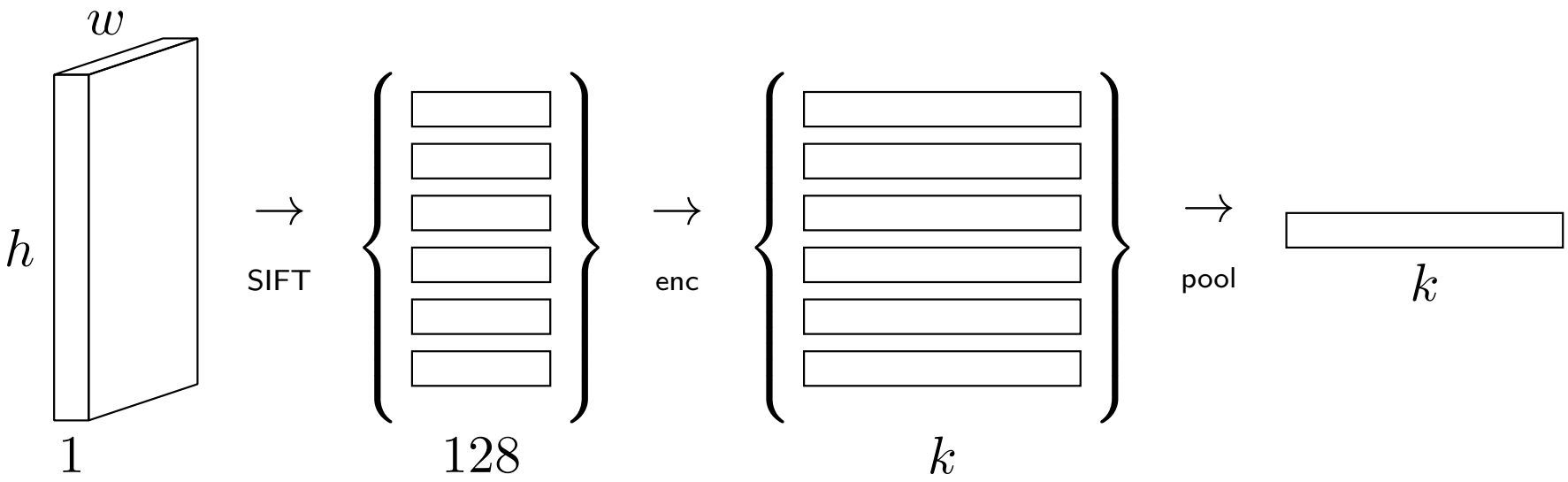
- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- set of ~ 1000 features $\times$ 128-dim SIFT descriptors
- element-wise encoding (hard assignment) on $k \sim 10^4$ visual words
- global sum pooling, ℓ^2 normalization

bag of words pipeline



- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- set of ~ 1000 features $\times$ 128-dim SIFT descriptors
- element-wise encoding (hard assignment) on $k \sim 10^4$ visual words
- global sum pooling, ℓ^2 normalization

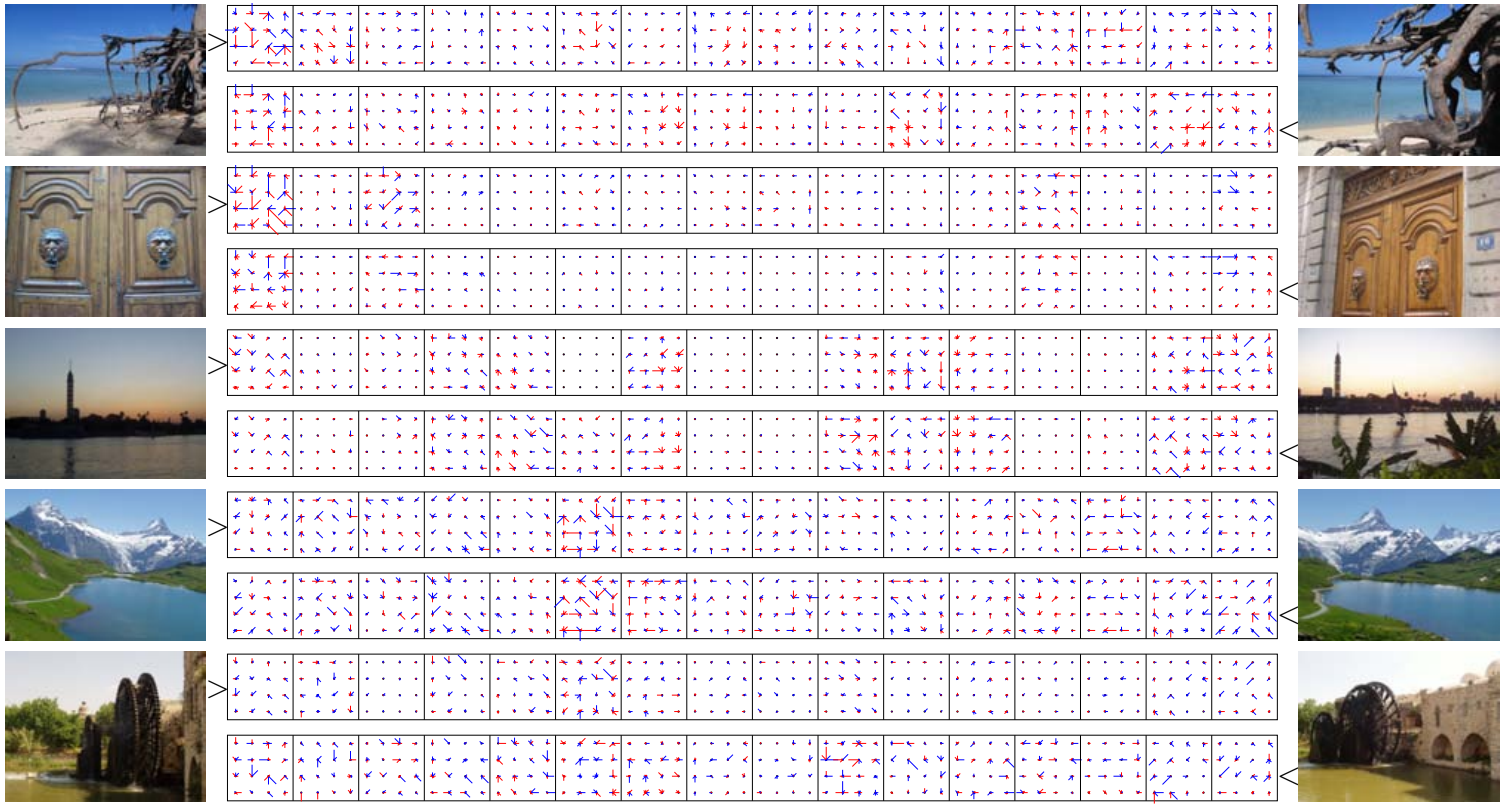
bag of words pipeline



- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- set of ~ 1000 features $\times$ 128-dim SIFT descriptors
- element-wise encoding (hard assignment) on $k \sim 10^4$ visual words
- global sum pooling, ℓ^2 normalization

vector of locally aggregated descriptors (VLAD)

[Jégou et al. 2010]



- encoding yields a vector per visual word, rather than a scalar frequency
- this vector is 128-dimensional like SIFT descriptors

Jégou, Douze, Schmid and Pérez. CVPR 2010. Aggregating Local Descriptors Into a Compact Image Representation.

VLAD definition

- input vectors: $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$
- vector quantizer: $q : \mathbb{R}^d \to C \subset \mathbb{R}^d$, $C = \{c_1, \ldots, c_k\}$

$$q(x) = \arg\min_{c \in C} \|x - c\|^2$$

- residual vector

$$r(x) = x - q(x)$$

- residual pooling per cell

$$V_c(X) = \sum_{\substack{x \in X \\ q(x)=c}} r(x) = \sum_{\substack{x \in X \\ q(x)=c}} x - q(x)$$

- VLAD vector (up to normalization)

$$\mathcal{V}(X) = [V_{c_1}(X), \ldots, V_{c_k}(X)]$$

Jégou, Douze, Schmid and Pérez. CVPR 2010. Aggregating Local Descriptors Into a Compact Image Representation.

VLAD definition

- input vectors: $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$
- vector quantizer: $q : \mathbb{R}^d \to C \subset \mathbb{R}^d$, $C = \{c_1, \ldots, c_k\}$

$$q(x) = \arg\min_{c \in C} \|x - c\|^2$$

- residual vector

$$r(x) = x - q(x)$$

- residual pooling per cell

$$V_c(X) = \sum_{\substack{x \in X \\ q(x)=c}} r(x) = \sum_{\substack{x \in X \\ q(x)=c}} x - q(x)$$

- VLAD vector (up to normalization)

$$\mathcal{V}(X) = [V_{c_1}(X), \ldots, V_{c_k}(X)]$$

Jégou, Douze, Schmid and Pérez. CVPR 2010. Aggregating Local Descriptors Into a Compact Image Representation.

VLAD definition

- input vectors: $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$
- vector quantizer: $q : \mathbb{R}^d \to C \subset \mathbb{R}^d$, $C = \{c_1, \ldots, c_k\}$

$$q(x) = \arg\min_{c \in C} \|x - c\|^2$$

- residual vector

$$r(x) = x - q(x)$$

- residual pooling per cell

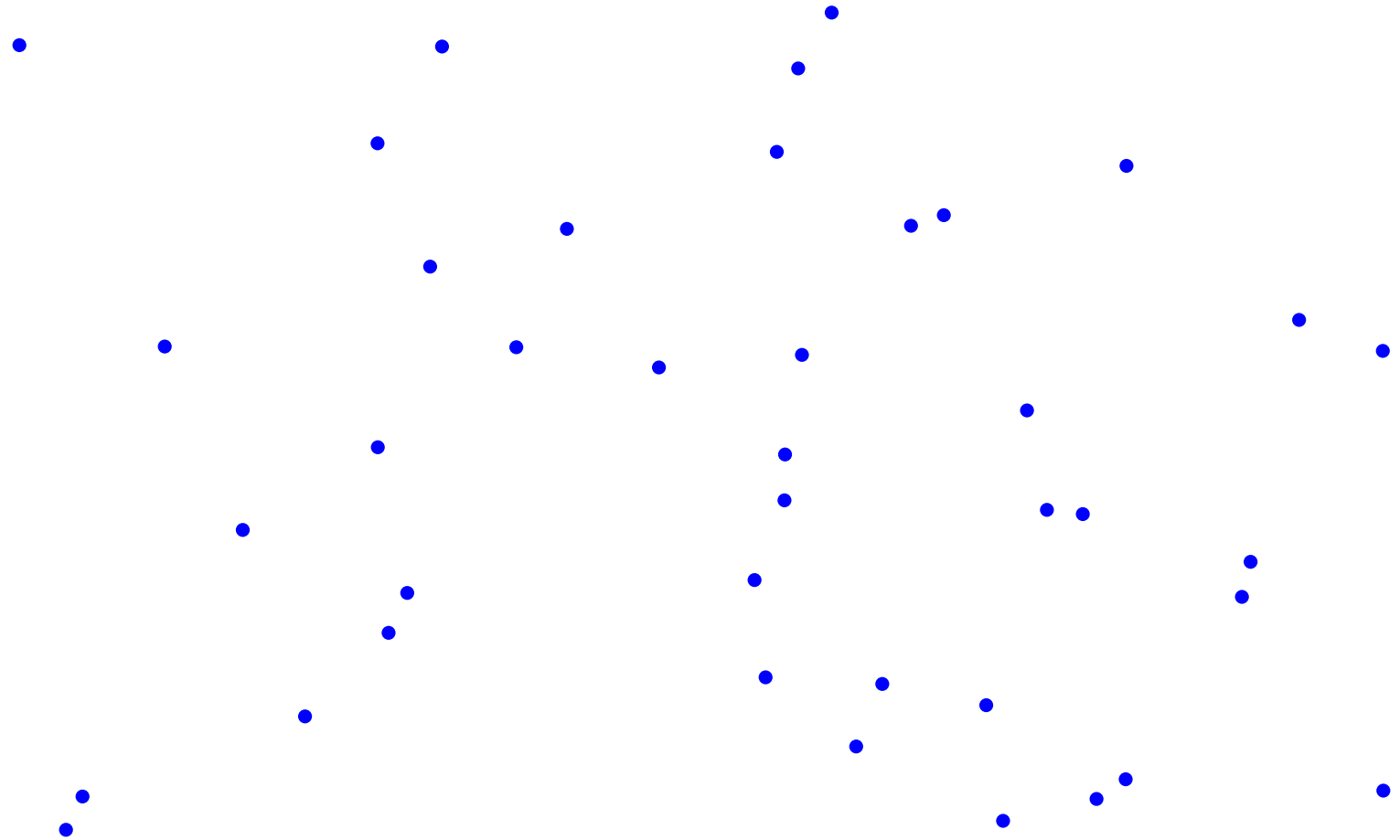
$$V_c(X) = \sum_{\substack{x \in X \\ q(x)=c}} r(x) = \sum_{\substack{x \in X \\ q(x)=c}} x - q(x)$$

- VLAD vector (up to normalization)

$$\mathcal{V}(X) = [V_{c_1}(X), \ldots, V_{c_k}(X)]$$

Jégou, Douze, Schmid and Pérez. CVPR 2010. Aggregating Local Descriptors Into a Compact Image Representation.

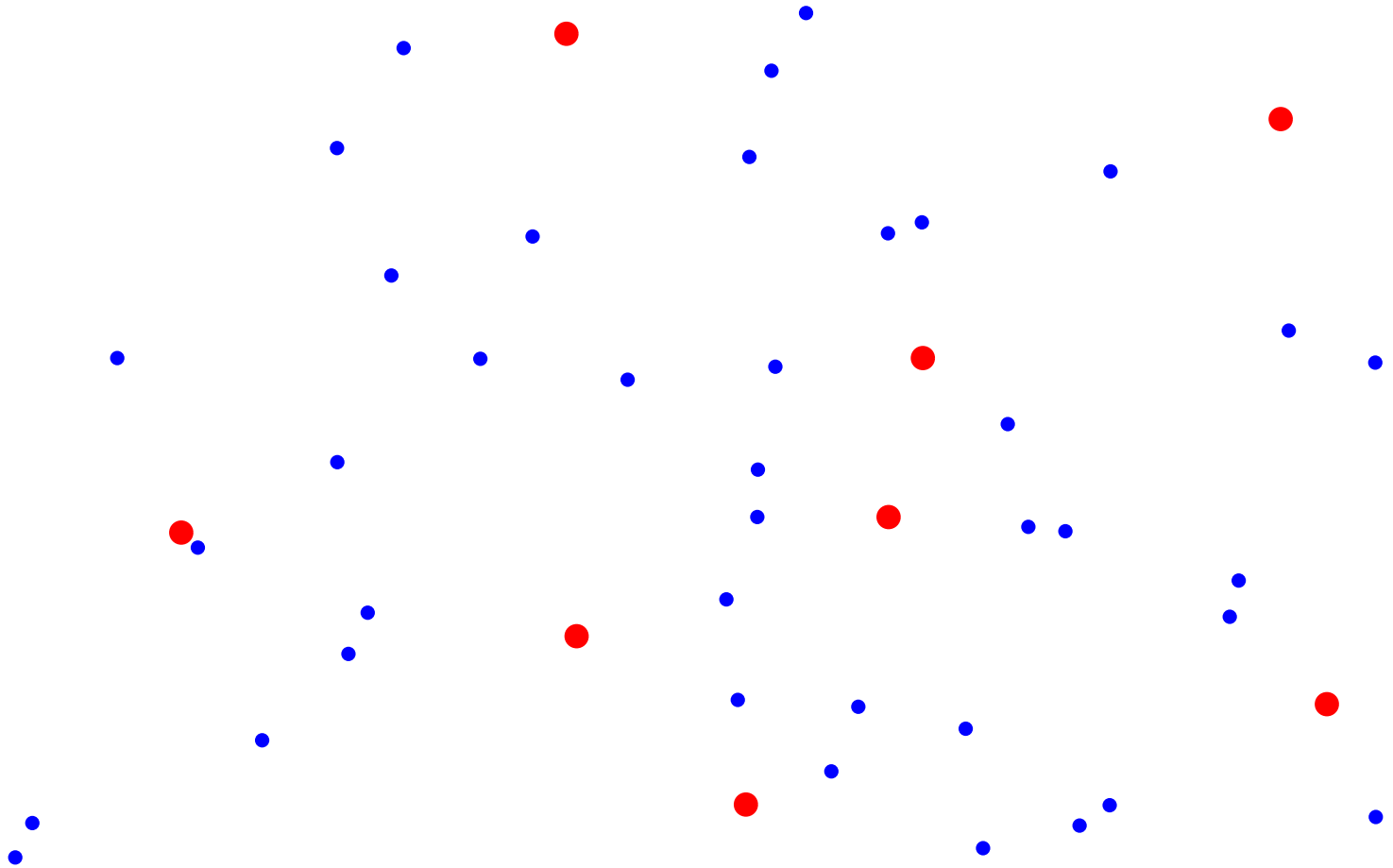
VLAD geometry



- input vectors – codebook – residuals – pooling

Jégou, Douze, Schmid and Pérez. CVPR 2010. Aggregating Local Descriptors Into a Compact Image Representation.

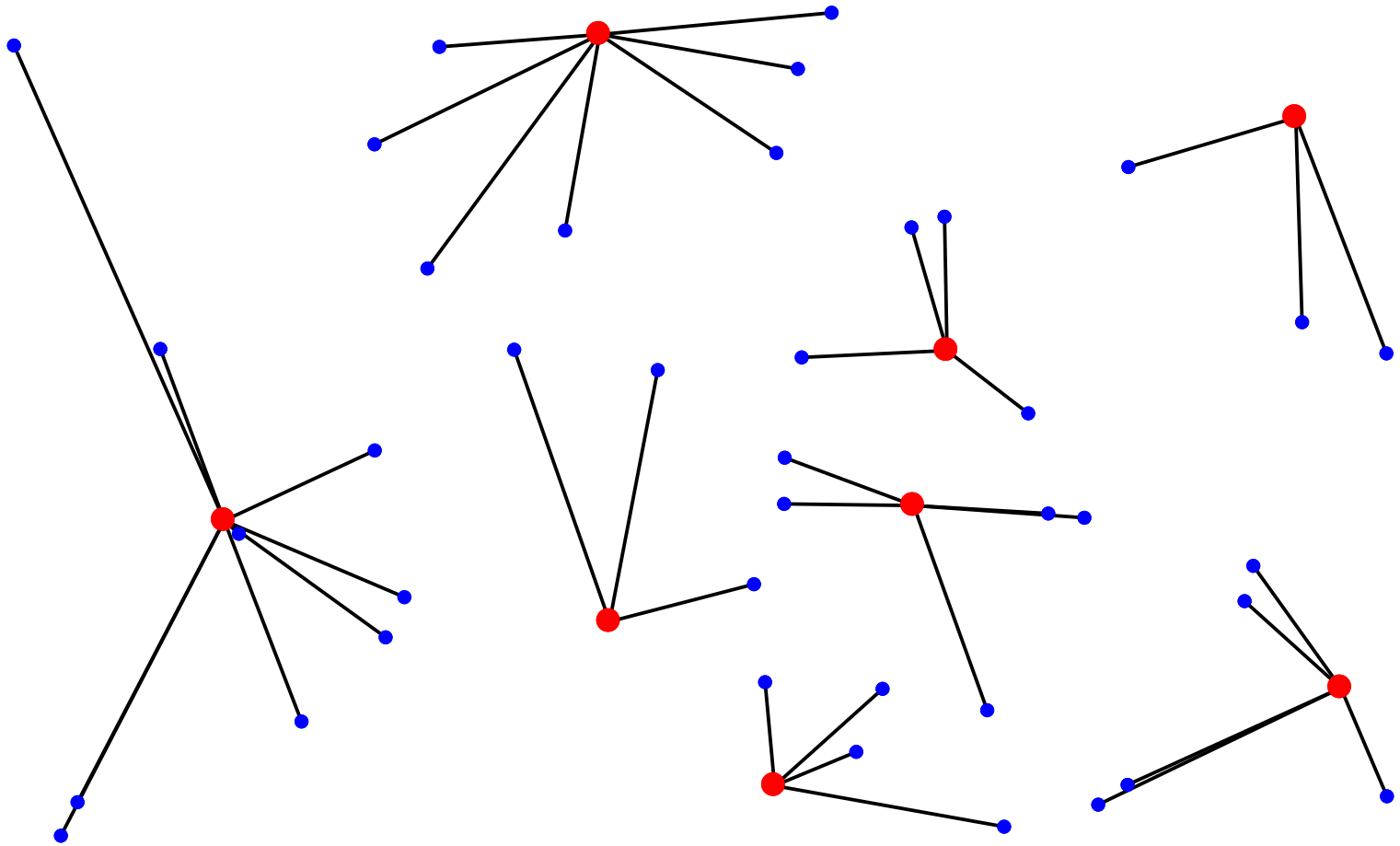
VLAD geometry



- input vectors – codebook – residuals – pooling

Jégou, Douze, Schmid and Pérez. CVPR 2010. Aggregating Local Descriptors Into a Compact Image Representation.

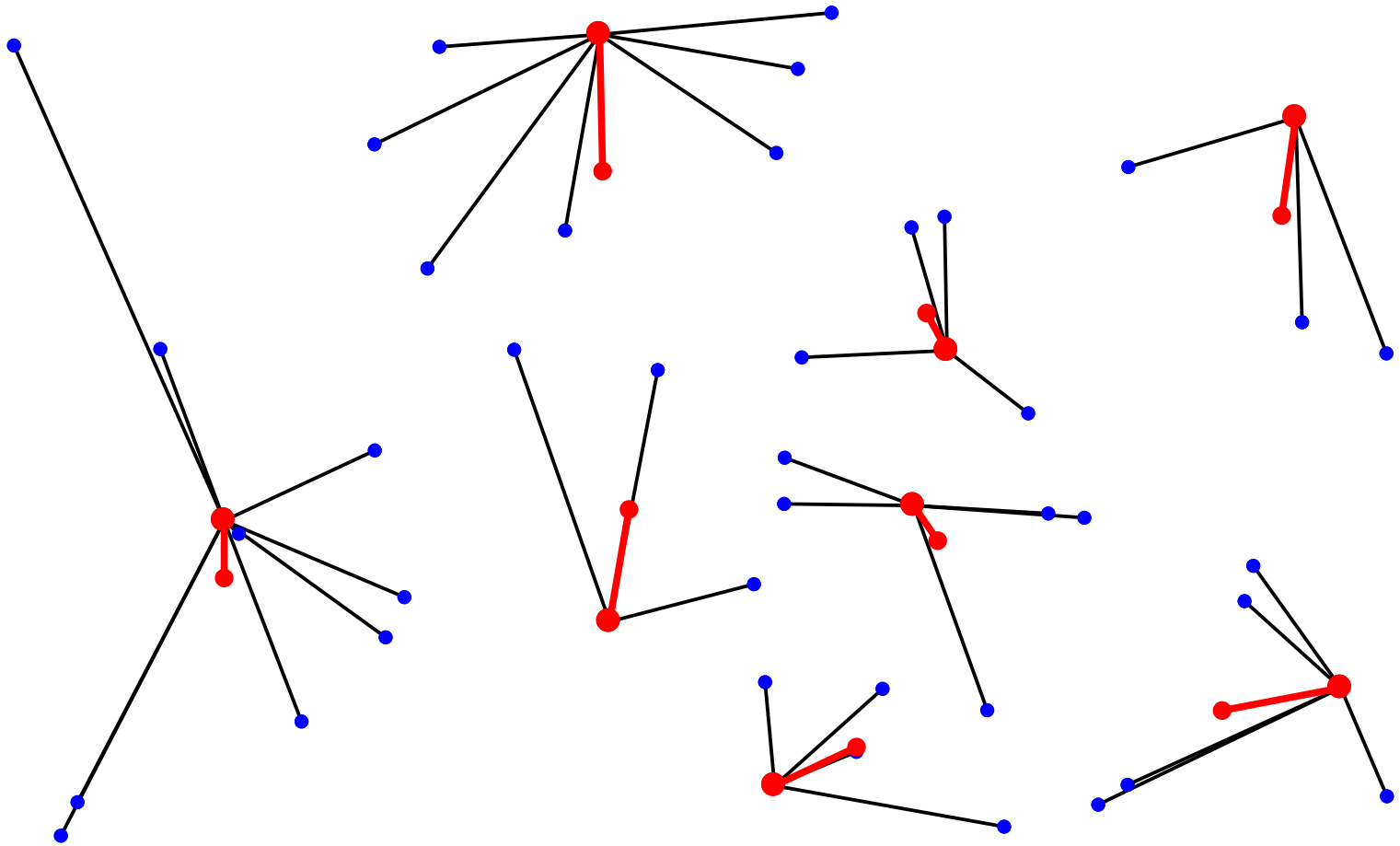
VLAD geometry



- input vectors – codebook – residuals – pooling

Jégou, Douze, Schmid and Pérez. CVPR 2010. Aggregating Local Descriptors Into a Compact Image Representation.

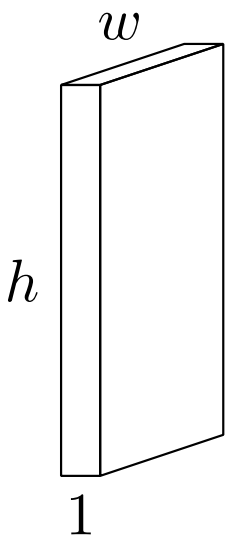
VLAD geometry



- input vectors – codebook – residuals – pooling

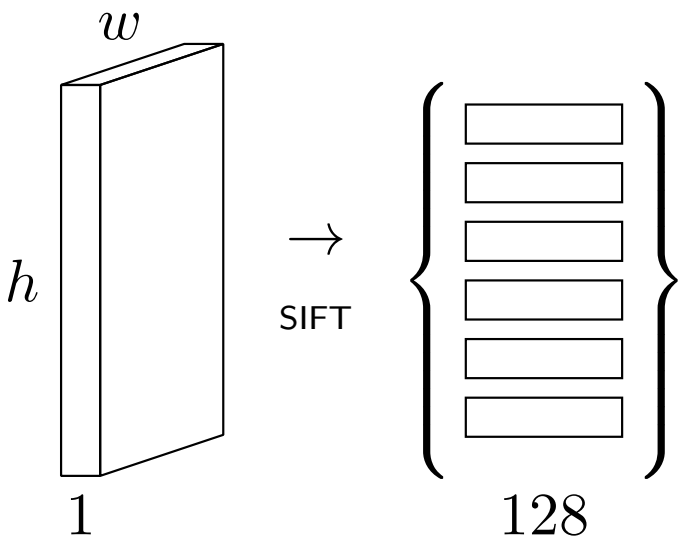
Jégou, Douze, Schmid and Pérez. CVPR 2010. Aggregating Local Descriptors Into a Compact Image Representation.

VLAD pipeline



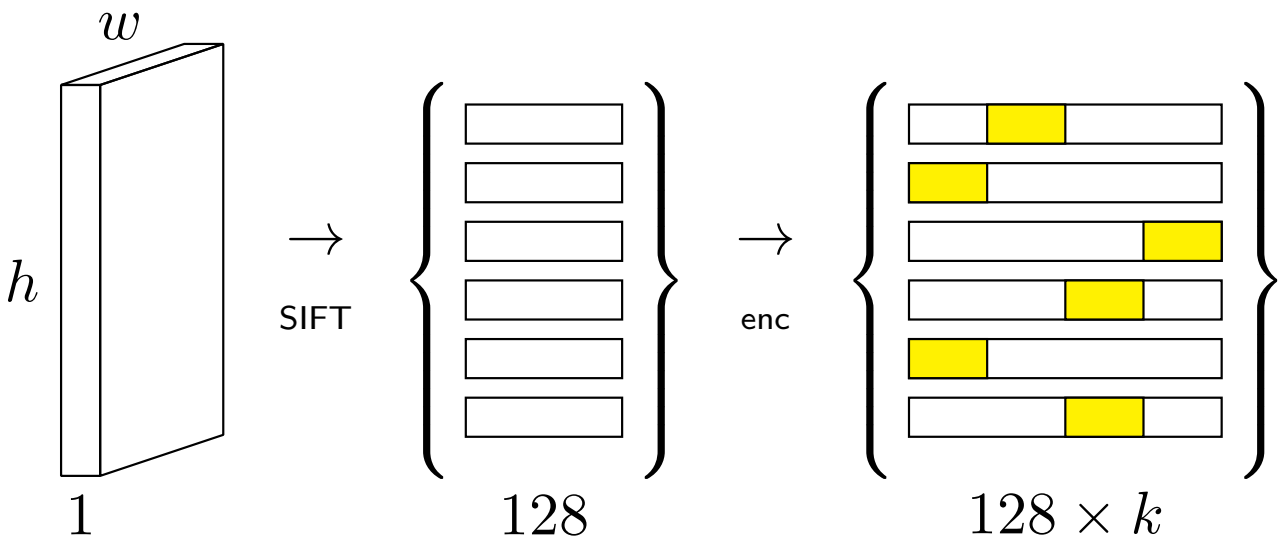
- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- set of ~ 1000 features $\times$ 128-dim SIFT descriptors
- element-wise encoding (hard assignment) on $k \sim 16$ visual words
- encoding now yields a residual vector rather than a scalar vote
- global sum pooling, ℓ^2 normalization

VLAD pipeline



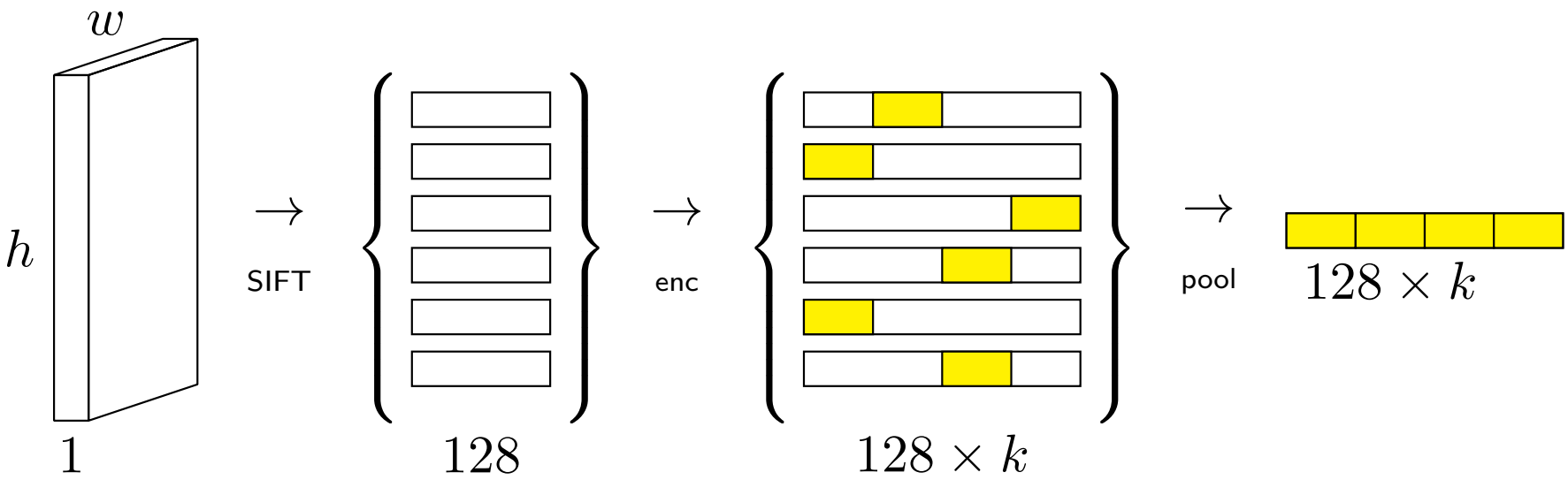
- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- set of ~ 1000 features $\times$ 128-dim SIFT descriptors
- element-wise encoding (hard assignment) on $k \sim 16$ visual words
- encoding now yields a residual vector rather than a scalar vote
- global sum pooling, ℓ^2 normalization

VLAD pipeline



- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- set of ~ 1000 features $\times$ 128-dim SIFT descriptors
- element-wise encoding (hard assignment) on $k \sim 16$ visual words
- encoding now yields a residual vector rather than a scalar vote
- global sum pooling, ℓ^2 normalization

VLAD pipeline



- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- set of ~ 1000 features $\times$ 128-dim SIFT descriptors
- element-wise encoding (hard assignment) on $k \sim 16$ visual words
- encoding now yields a residual vector rather than a scalar vote
- global sum pooling, ℓ^2 normalization

probabilistic interpretation

- if $p(X|C)$ is the likelihood of i.i.d observations X under a uniform isotropic Gaussian mixture model with component means C

$$p(X|C) \propto \prod_{x \in X} e^{-\frac{1}{2}\|x - q(x)\|^2}$$

- then the VLAD vector is proportional the gradient of $\ln p(X|C)$ with respect to the model parameters C

$$\mathcal{V}(X) \propto \nabla_C \ln p(X|C) = [\nabla_{c_1} \ln p(X|C), \ldots, \nabla_{c_k} \ln p(X|C)]$$

- if we were to optimize C to fit the data X, then $\hat{\mathcal{V}}(X)$ would be the direction in which to modify C

probabilistic interpretation

- if $p(X|C)$ is the likelihood of i.i.d observations X under a uniform isotropic Gaussian mixture model with component means C

$$p(X|C) \propto \prod_{x \in X} e^{-\frac{1}{2}\|x - q(x)\|^2}$$

- then the VLAD vector is proportional the gradient of $\ln p(X|C)$ with respect to the model parameters C

$$\mathcal{V}(X) \propto \nabla_C \ln p(X|C) = [\nabla_{c_1} \ln p(X|C), \ldots, \nabla_{c_k} \ln p(X|C)]$$

- if we were to optimize C to fit the data X, then $\hat{\mathcal{V}}(X)$ would be the direction in which to modify C

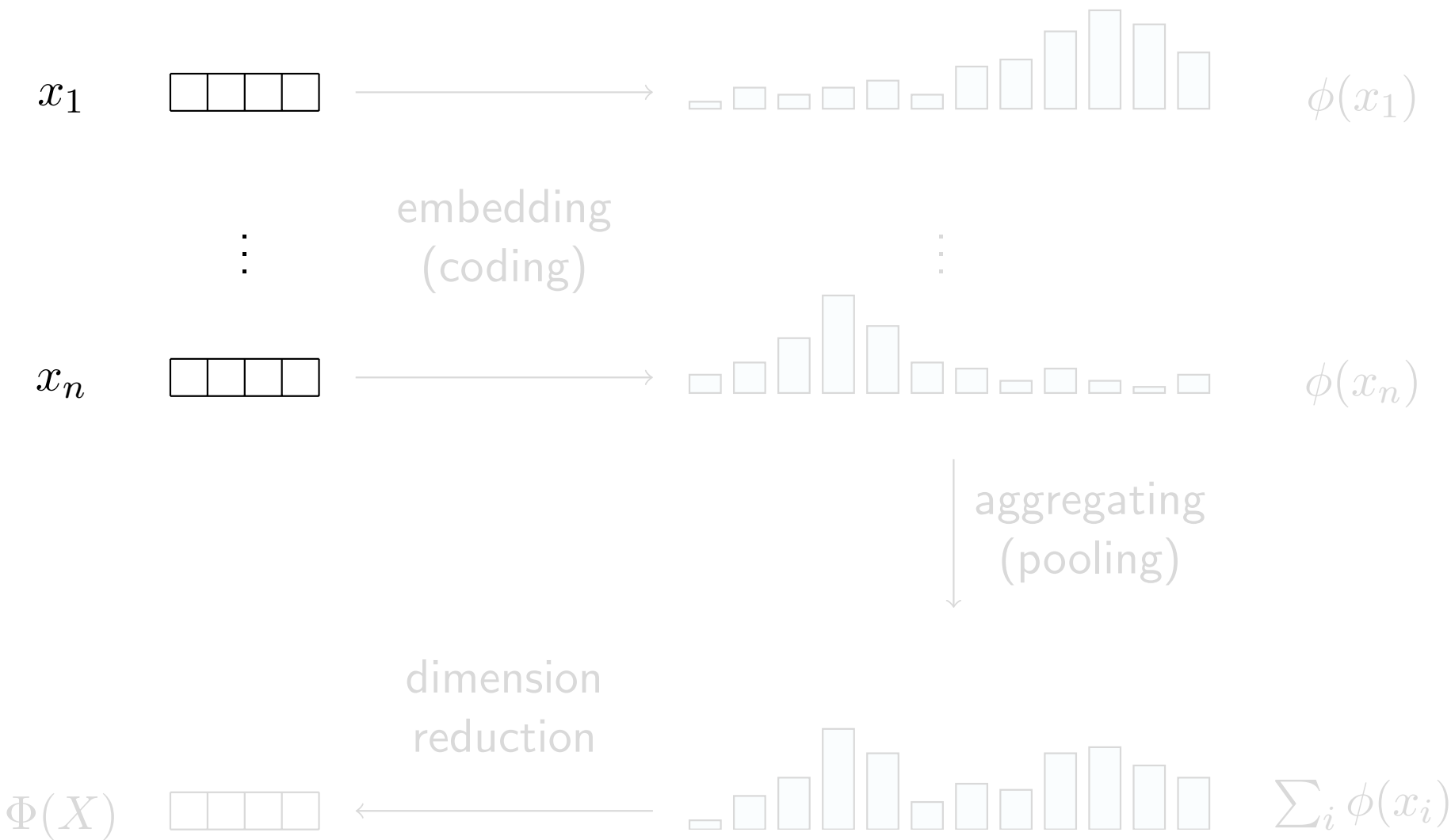
Fisher kernel

- the Fisher kernel generalizes to a non-uniform diagonal Gaussian mixture model

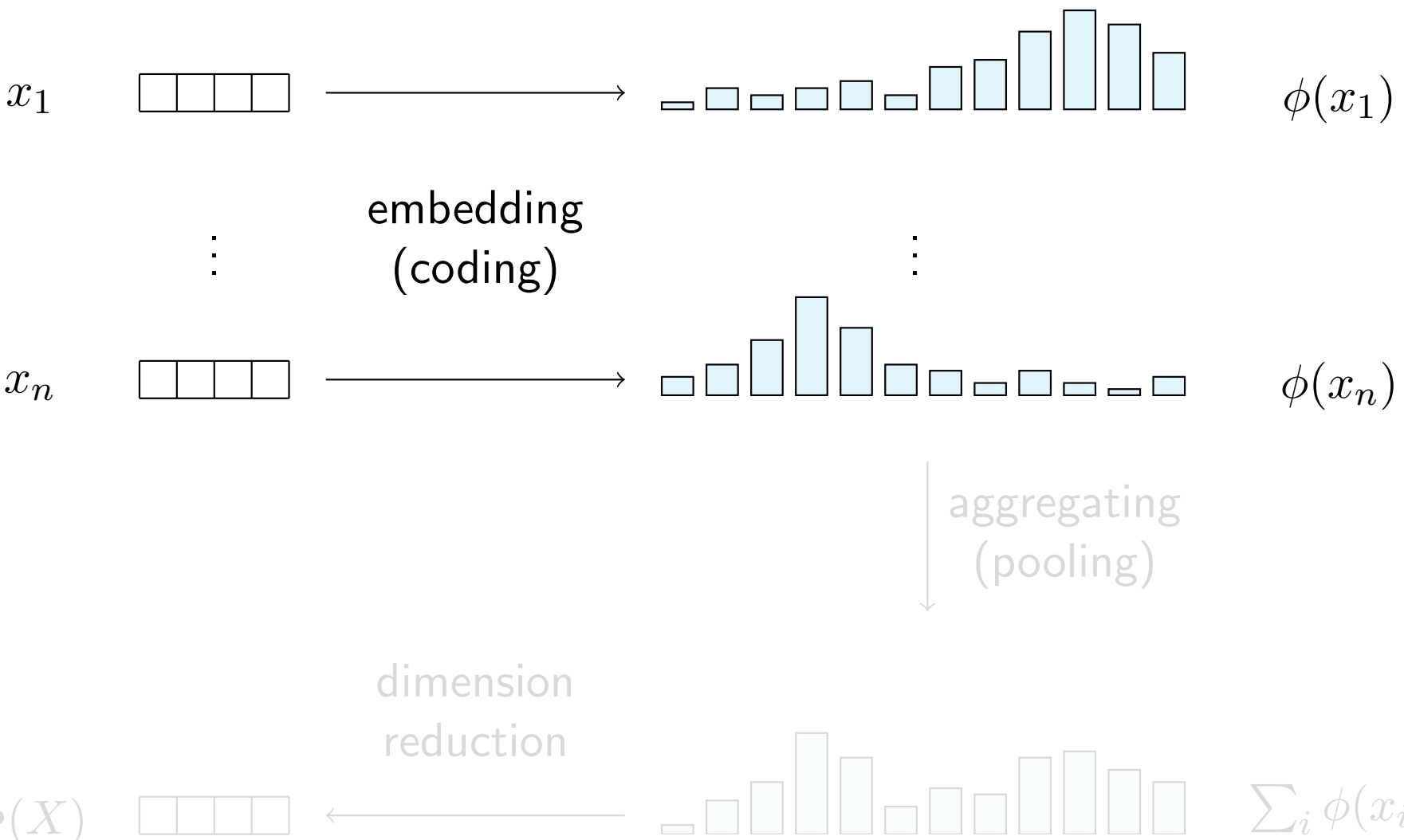
order statistics	parameter	model
0	mixing coefficient π	BoW
1	means μ	VLAD
2	standard deviations σ	Fisher

Perronnin and Dance CVPR 2006. Fisher Kernels on Visual Vocabularies for Image Categorization.

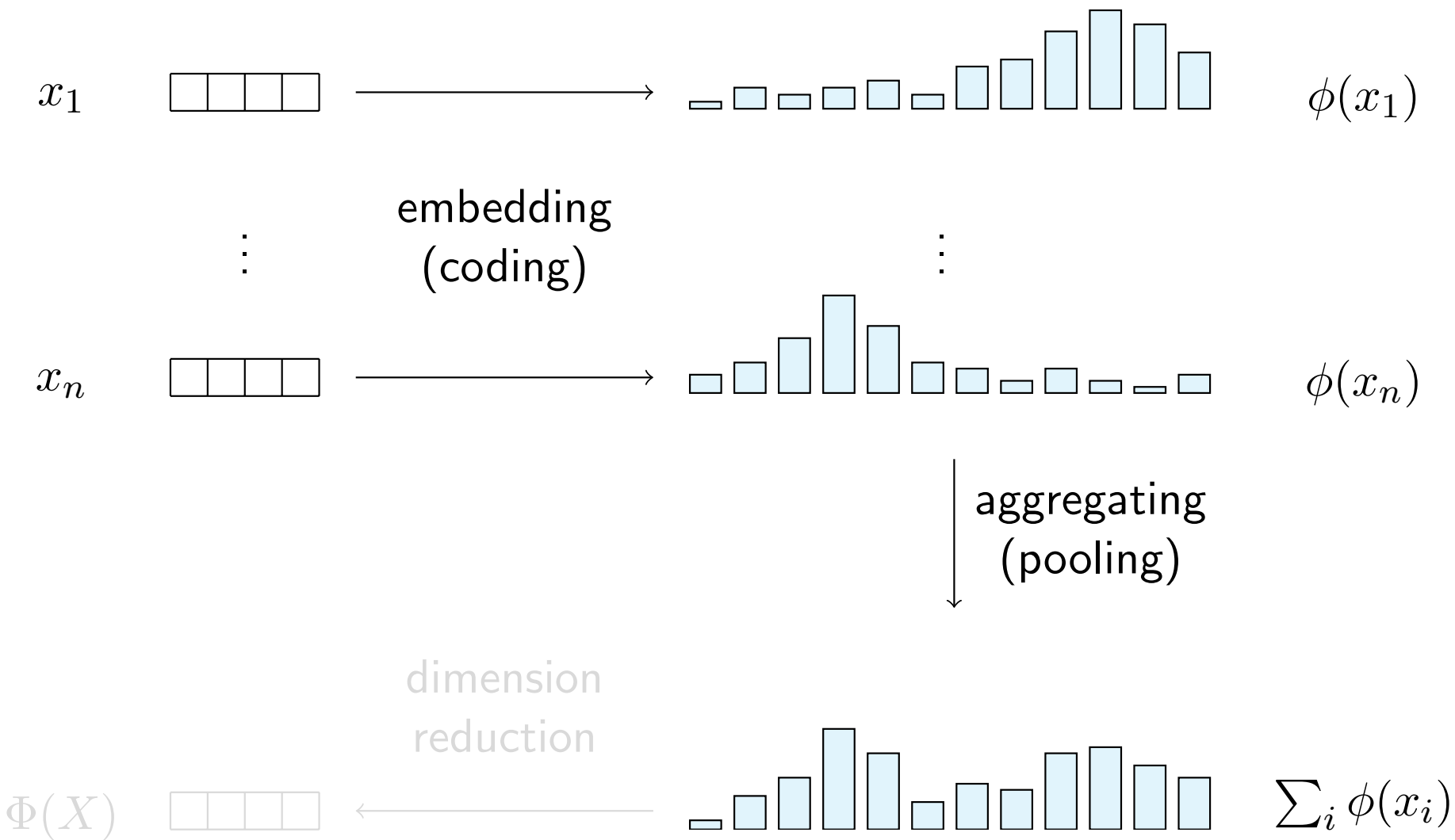
embeddings in general



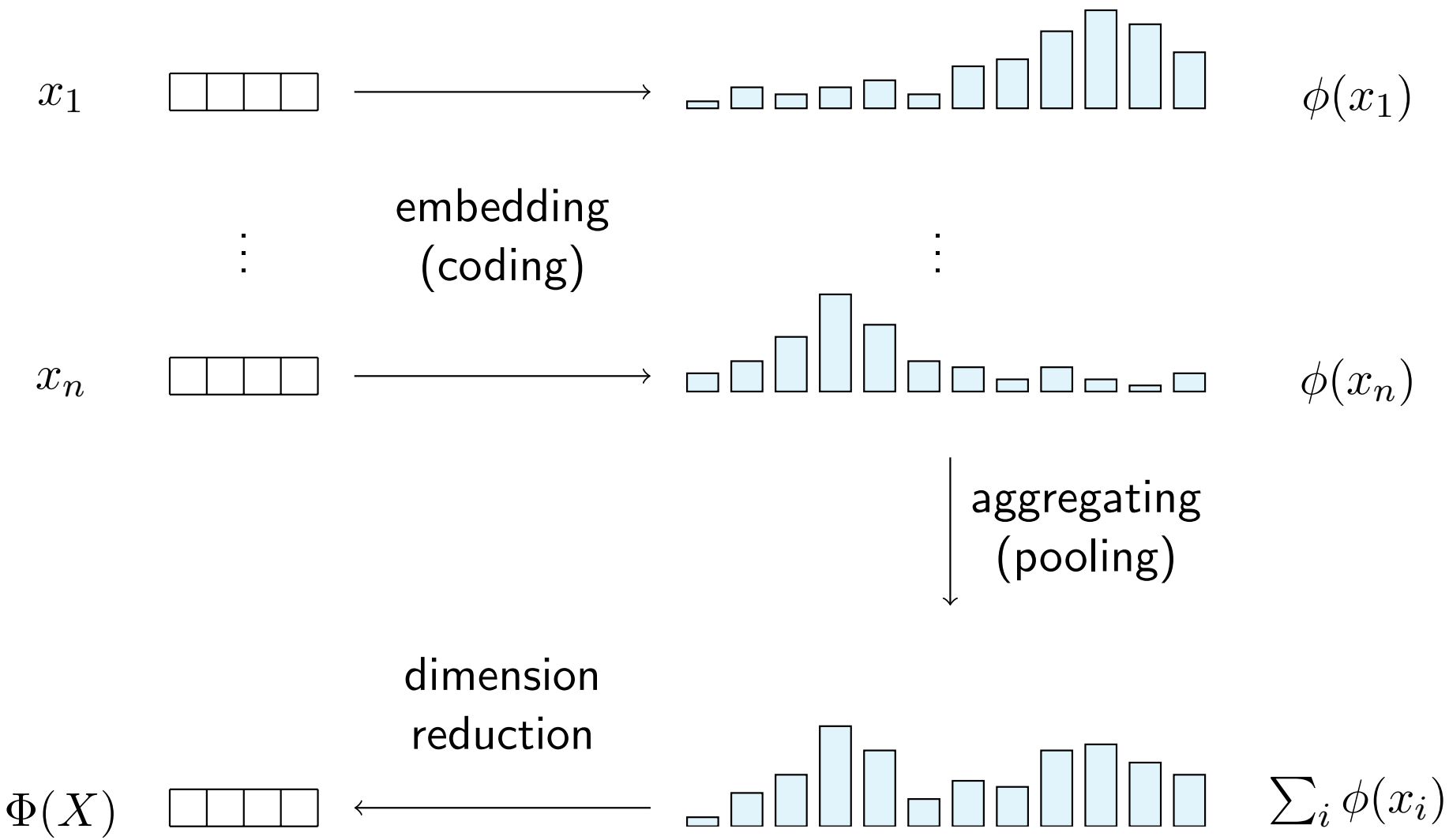
embeddings in general



embeddings in general

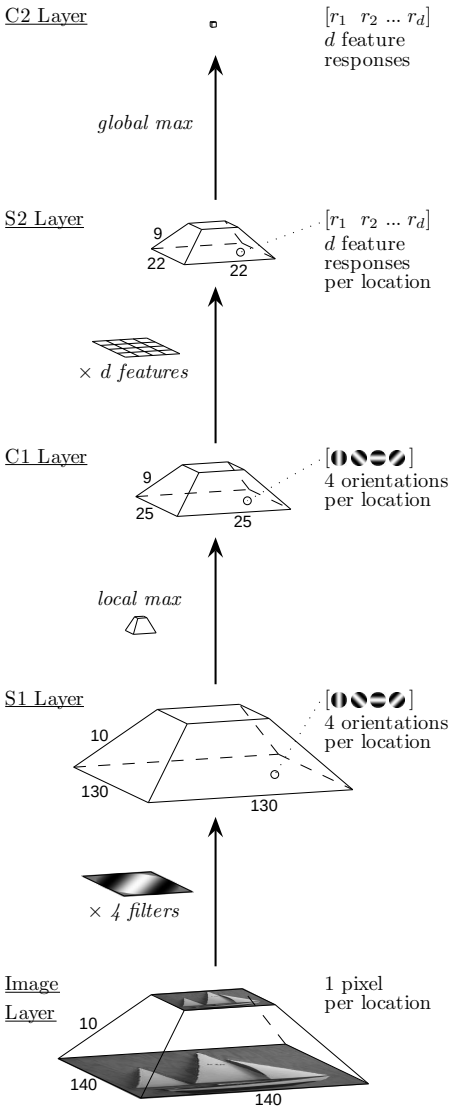


embeddings in general



improvements

[Mutch and Lowe 2006]



- image pyramid
- S1 inhibition: non-maxima suppression over orientations
- strided C1 max pooling (50% overlap)
- C1 sparsification: dominant orientations kept

Mutch and Lowe. CVPR 2006. Multiclass Object Recognition With Sparse, Localized features.

Conclusion

Derivatives, features detection, spatial matching
Visual descriptors: SIFT, HOG...
Embeddings: BOW, VLAD, Fisher.