ECE468/CS519: Digital Image Processing

Image Features

Prof. Sinisa Todorovic

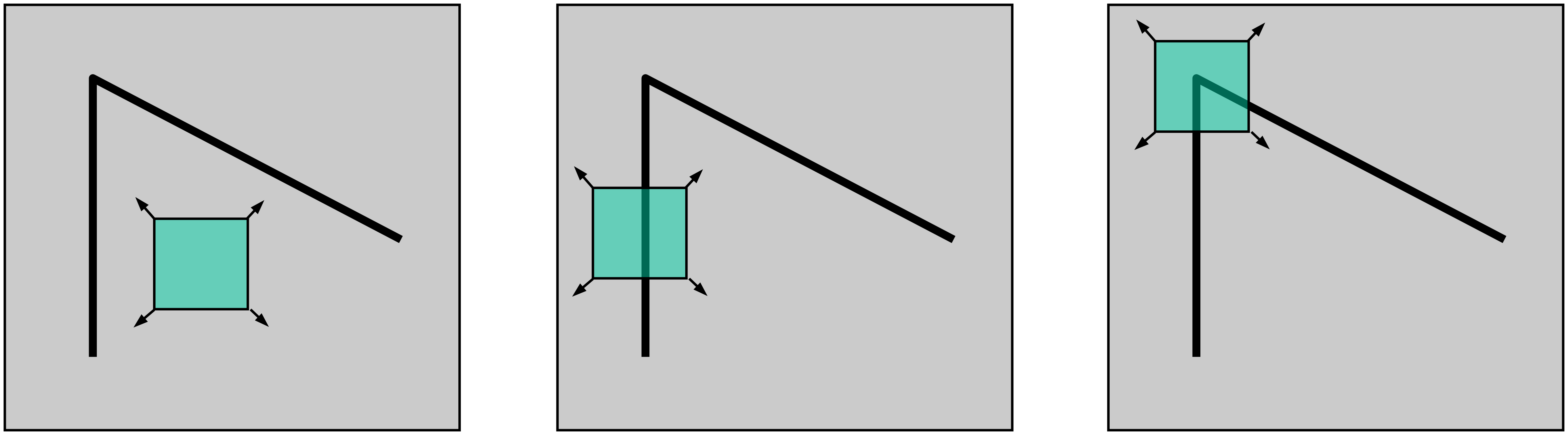
sinisa@eecs.oregonstate.edu

OSU Oregon State University

Outline

- Matlab
- Image features -- Interest points
- Point descriptors
- Homework 1

Harris Corner Detector



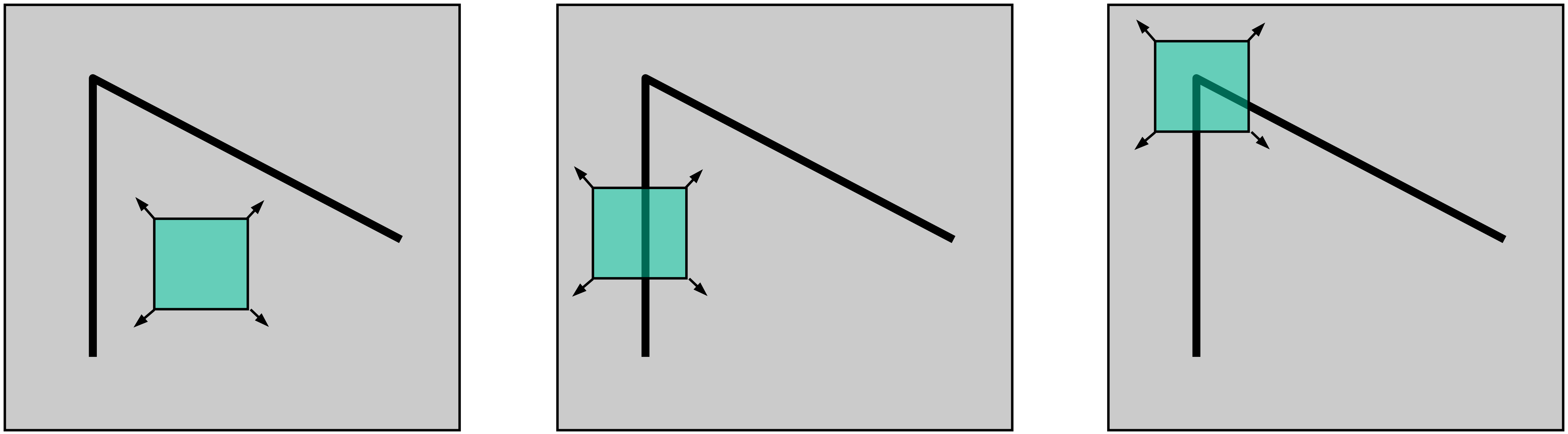
homogeneous region ⇓ no change in all directions

edge ⇓ no change along the edge

corner ⇓ change in all directions

Source: Frolova, Simakov, Weizmann Institute

Harris Corner Detector



$$E(x,y) = w(x,y) * [I(x+u, y+v) - I(x,y)]^2$$

2D convolution

Source: Frolova, Simakov, Weizmann Institute

Harris Corner Detector

Taylor series expansion

For small shifts

$$\begin{aligned} u &\rightarrow 0 \\ v &\rightarrow 0 \end{aligned} \quad \Rightarrow \quad I(x+u, y+v) \approx I(x,y) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v$$

$$I(x+u, y+v) \approx I(x,y) + [I_x \quad I_y] \begin{bmatrix} u \\ v \end{bmatrix}$$

image derivatives along x and y axes

Harris Corner Detector

$$E(x,y) = w(x,y) * \left(I(x,y) + [I_x \quad I_y] \begin{bmatrix} u \\ v \end{bmatrix} - I(x,y) \right)^2$$

$$= w(x,y) * \left([I_x \quad I_y] \begin{bmatrix} u \\ v \end{bmatrix} \right)^2$$

$$= w(x,y) * \left([I_x \quad I_y] \begin{bmatrix} u \\ v \end{bmatrix} \right)^{\mathrm{T}} \left([I_x \quad I_y] \begin{bmatrix} u \\ v \end{bmatrix} \right)$$

Harris Corner Detector

$$E(x,y) = w(x,y) * [u \quad v] \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \quad I_y] \begin{bmatrix} u \\ v \end{bmatrix}$$

$$= [u \quad v] \underbrace{\left(w(x,y) * \begin{bmatrix} I_x^2(x,y) & I_x(x,y) I_y(x,y) \\ I_x(x,y) I_y(x,y) & I_y^2(x,y) \end{bmatrix} \right)}_{M(x,y)} \begin{bmatrix} u \\ v \end{bmatrix}$$

Harris Corner Detector

$$E(x,y) = [u \quad v] M(x,y) \left[\begin{array}{c} u \\ v \end{array} \right]$$

$$M(x,y;\sigma) = w(x,y;\sigma) * \left[\begin{array}{cc} I_x^2(x,y) & I_x(x,y)I_y(x,y) \\ I_x(x,y)I_y(x,y) & I_y^2(x,y) \end{array} \right]$$

$$M(x,y;\sigma) = \left[\begin{array}{cc} w(x,y;\sigma) * I_x^2(x,y) & w(x,y;\sigma) * I_x(x,y)I_y(x,y) \\ w(x,y;\sigma) * I_x(x,y)I_y(x,y) & w(x,y;\sigma) * I_y^2(x,y) \end{array} \right]$$

Image Gradient

$$I_x(x,y) = I(x+1,y) - I(x,y)$$

$$= \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{D_x(x,y)} * I(x,y)$$

Image Gradient

$$I_y(x,y) = I(x,y+1) - I(x,y)$$

$$= \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{D_y(x,y)} * I(x,y)$$

Weighted Image Gradient

$$w(x, y; \sigma) * I_x(x, y) = w(x, y; \sigma) * D_x(x, y) * I(x, y)$$

$$= [w(x, y; \sigma) * D_x(x, y)] * I(x, y)$$

convolution is associative

Weighted Image Gradient

$$w(x, y; \sigma) * I_x(x, y) = w(x, y; \sigma) * D_x(x, y) * I(x, y)$$

$$= [w(x, y; \sigma) * D_x(x, y)] * I(x, y)$$

$$= [D_x(x, y) * w(x, y; \sigma)] * I(x, y)$$

convolution is commutative

Weighted Image Gradient

$$w(x, y; \sigma) * I_x(x, y) = w(x, y; \sigma) * D_x(x, y) * I(x, y)$$

$$= [w(x, y; \sigma) * D_x(x, y)] * I(x, y)$$

$$= [D_x(x, y) * w(x, y; \sigma)] * I(x, y)$$

$$= w_x(x, y; \sigma) * I(x, y)$$

derivative of the filter

Weighted Image Gradient

$$w(x, y; \sigma) * I_x(x, y) = w_x(x, y; \sigma) * I(x, y)$$

$$w(x, y; \sigma) * I_y(x, y) = w_y(x, y; \sigma) * I(x, y)$$

Image is discrete $\Rightarrow$ Gradient is approximate

We always find the gradient of the kernel !

Harris Corner Detector

$$E(x,y) = [u \quad v] M(x,y) \begin{bmatrix} u \\ v \end{bmatrix}$$

$$M(x,y) = \begin{bmatrix} (w_x * I)^2 & (w_x * I)(w_y * I) \\ (w_x * I)(w_y * I) & (w_y * I)^2 \end{bmatrix}_{(x,y)}$$

Harris Detector

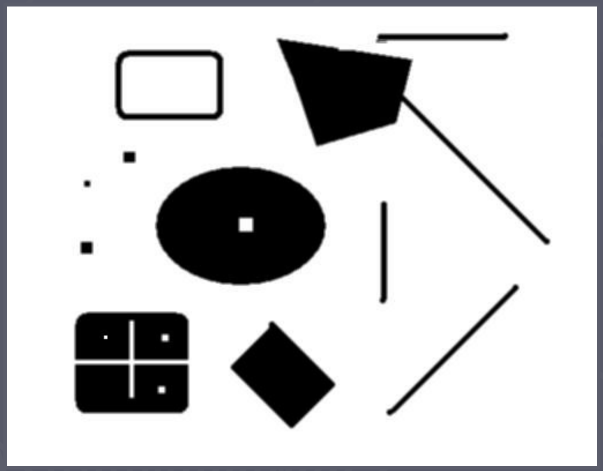
$$f(\sigma) = \frac{\lambda_1(\sigma)\lambda_2(\sigma)}{\lambda_1(\sigma) + \lambda_2(\sigma)} = \frac{\det(M(\sigma))}{\mathrm{trace}(M(\sigma))}$$

objective function

Example of Detecting Harris Corners

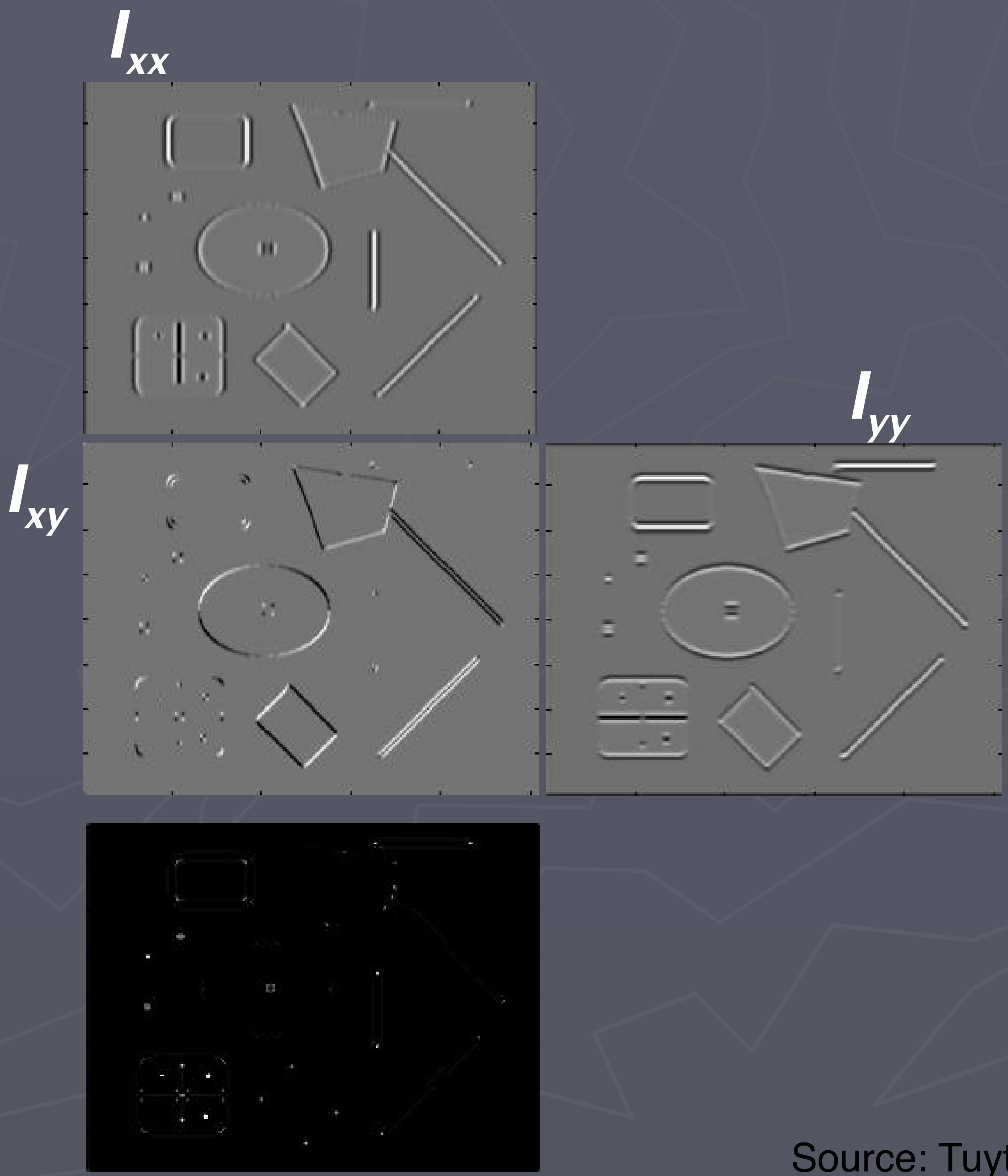
Hessian detector (Beaudet, 1978)

Hessian determinant



$$Hessian(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

$$\det(Hessian(I)) = I_{xx} I_{yy} - I_{xy}^2$$



Source: Tuytelaars

Hessian Detector

$$E(x,y) = [u \quad v] H(x,y) \begin{bmatrix} u \\ v \end{bmatrix}$$

$$H(x,y;\sigma) = w(x,y;\sigma) * \begin{bmatrix} I_{xx}(x,y) & I_{xy}(x,y) \\ I_{xy}(x,y) & I_{yy}(x,y) \end{bmatrix}$$

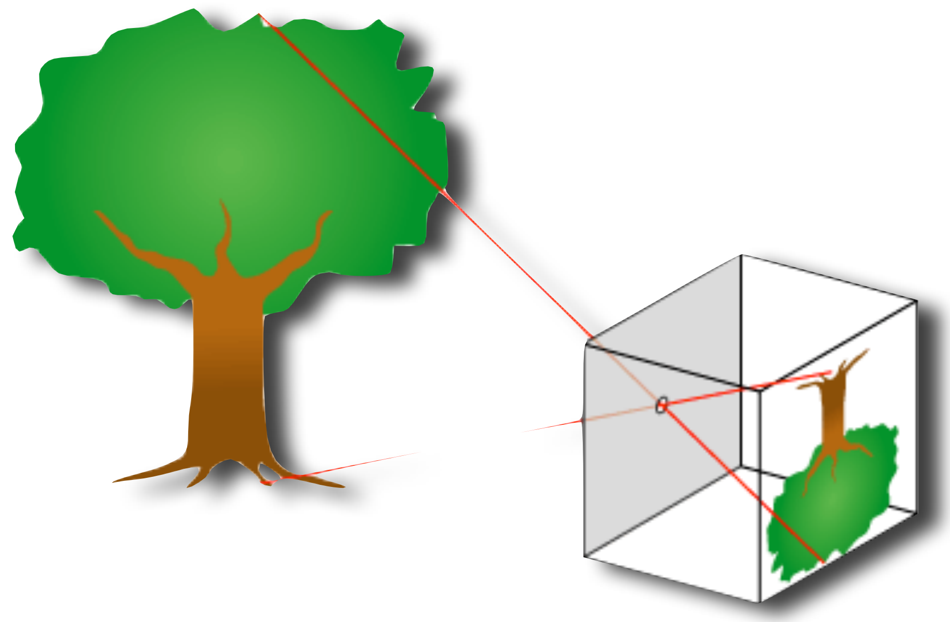
$$H(x,y;\sigma) = \begin{bmatrix} w_{xx}(x,y;\sigma) * I(x,y) & w_{xy}(x,y;\sigma) * I(x,y) \\ w_{xy}(x,y;\sigma) * I(x,y) & w_{yy}(x,y;\sigma) * I(x,y) \end{bmatrix}$$

Hessian Detector

$$f(\sigma) = \frac{\lambda_1(\sigma)\lambda_2(\sigma)}{\lambda_1(\sigma) + \lambda_2(\sigma)} = \frac{\det(H(\sigma))}{\mathrm{trace}(H(\sigma))}$$

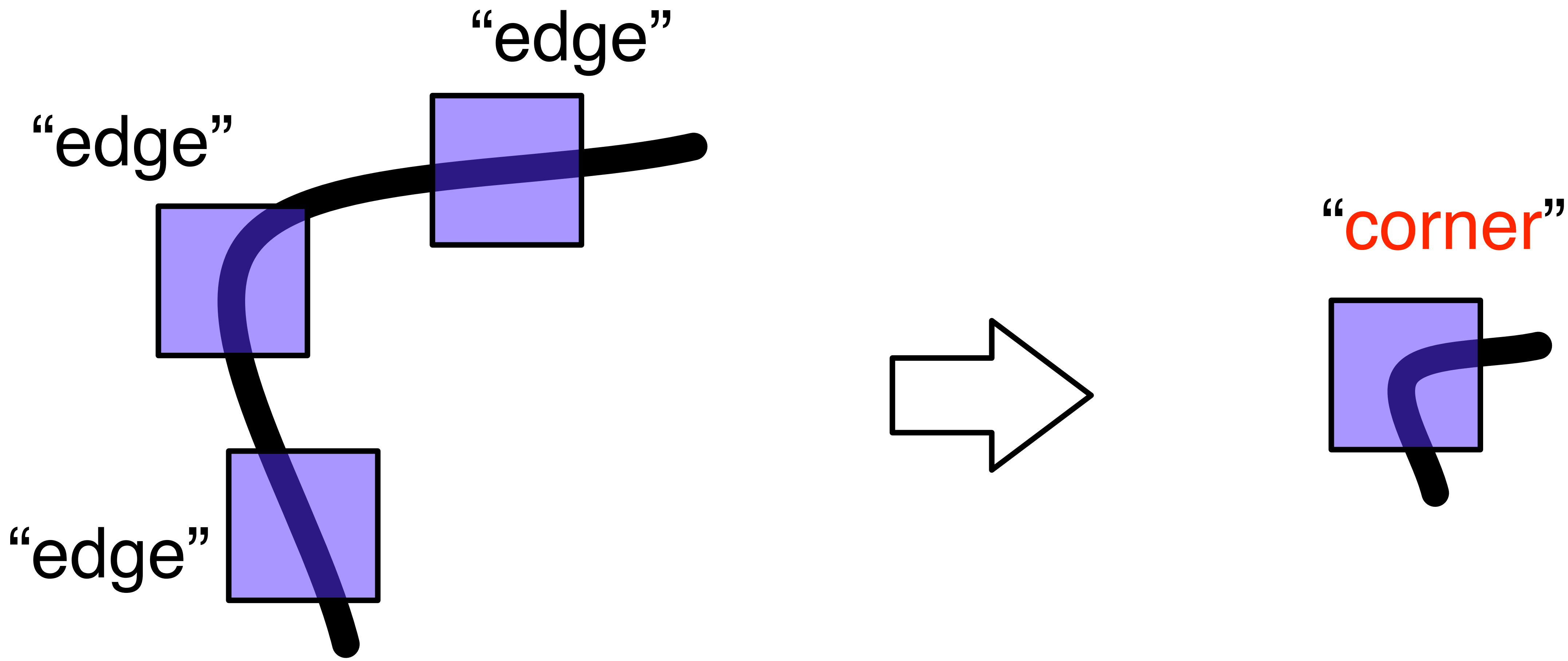
objective function

Properties of Harris Corners



- Invariance to variations of imaging parameters:
  - Illumination?
  - Camera distance, i.e., scale ?
  - Camera viewpoint, i.e., affine transformation?

Harris/Hessian Detector is NOT Scale Invariant

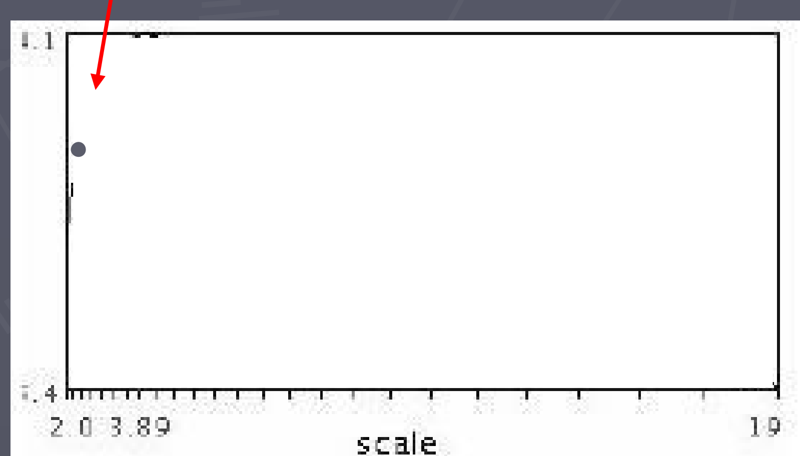


Source: L. Fei-Fei

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

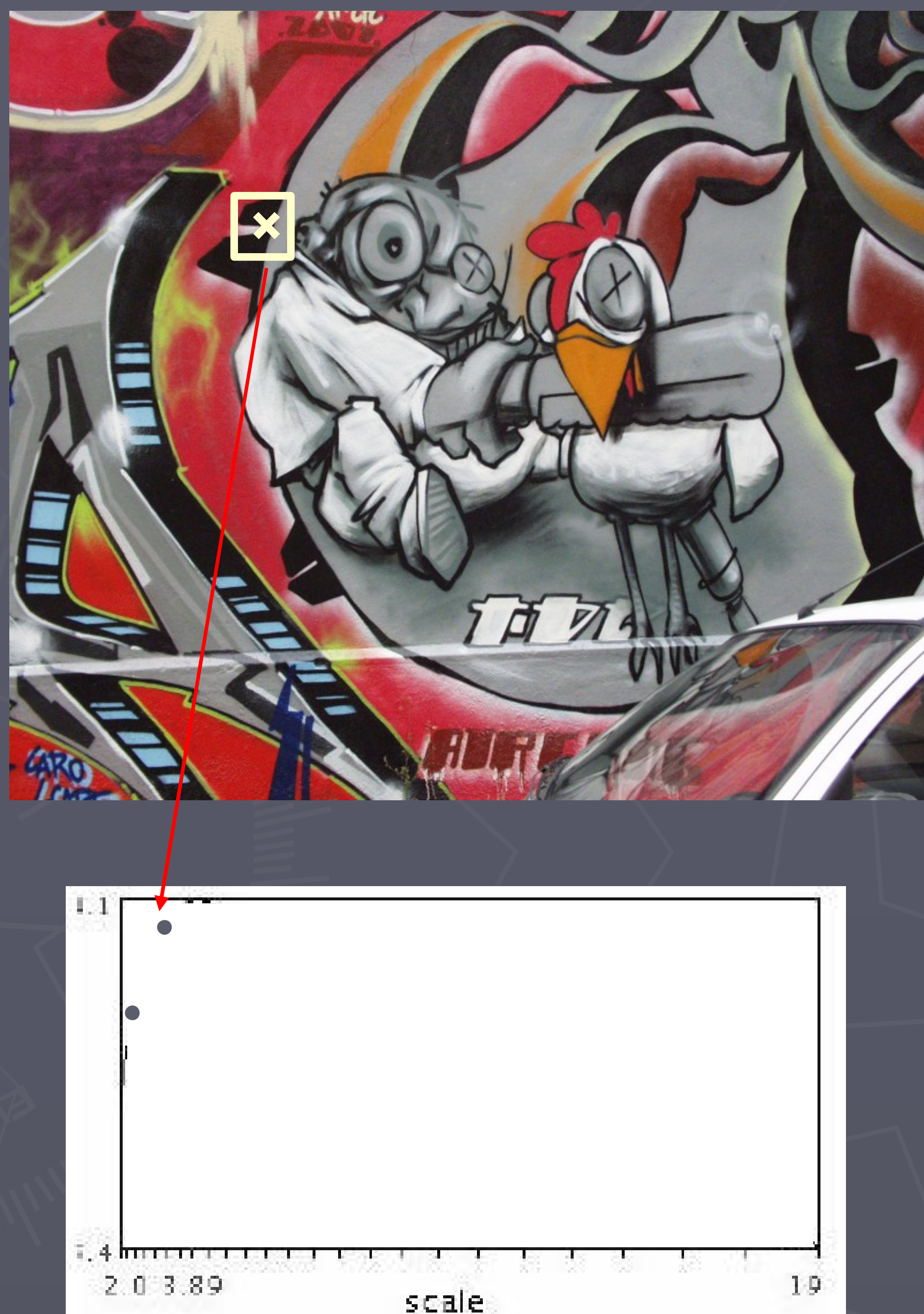


$f(I_{i_1 \ldots i_m}(x,\sigma))$

Source: Tuytelaars

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)



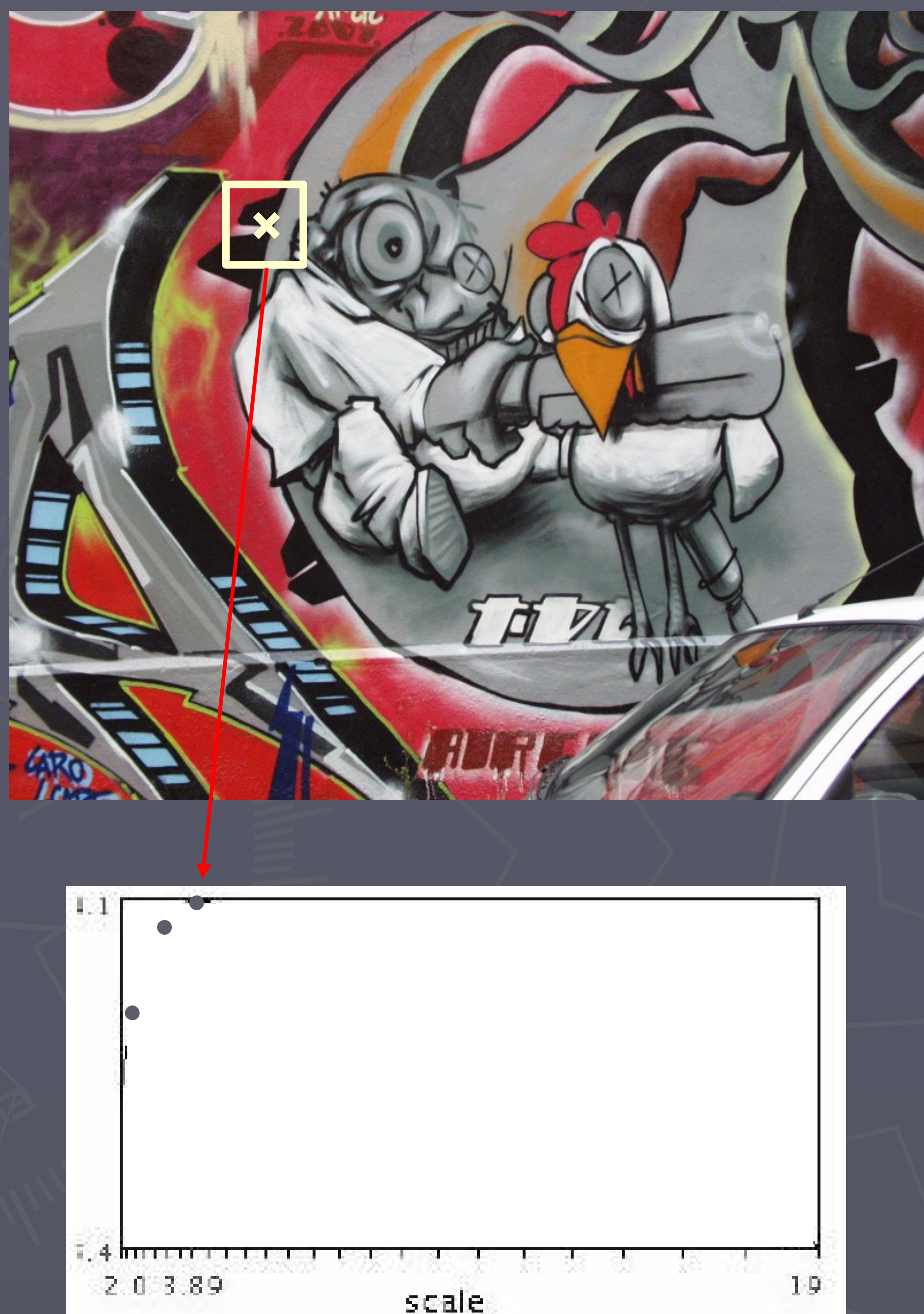
$f(I_{i_1 \ldots i_m}(x,\sigma))$

Source: Tuytelaars

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)



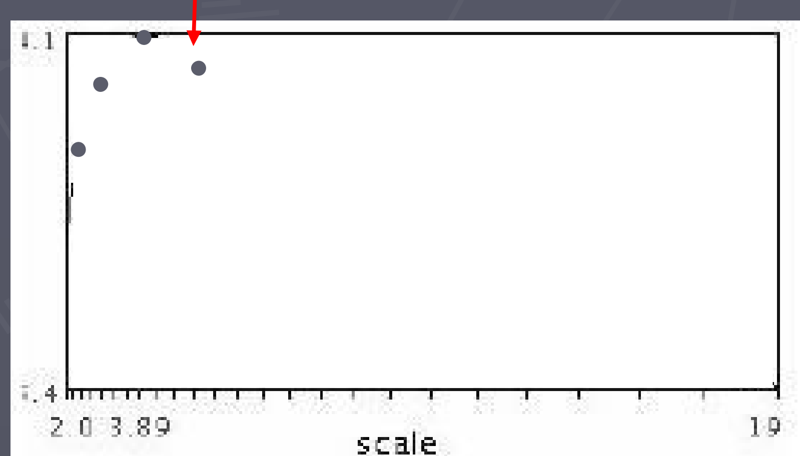
$f(I_{i_1 \ldots i_m}(x,\sigma))$

Source: Tuytelaars

Automatic scale selection

Function responses for increasing scale

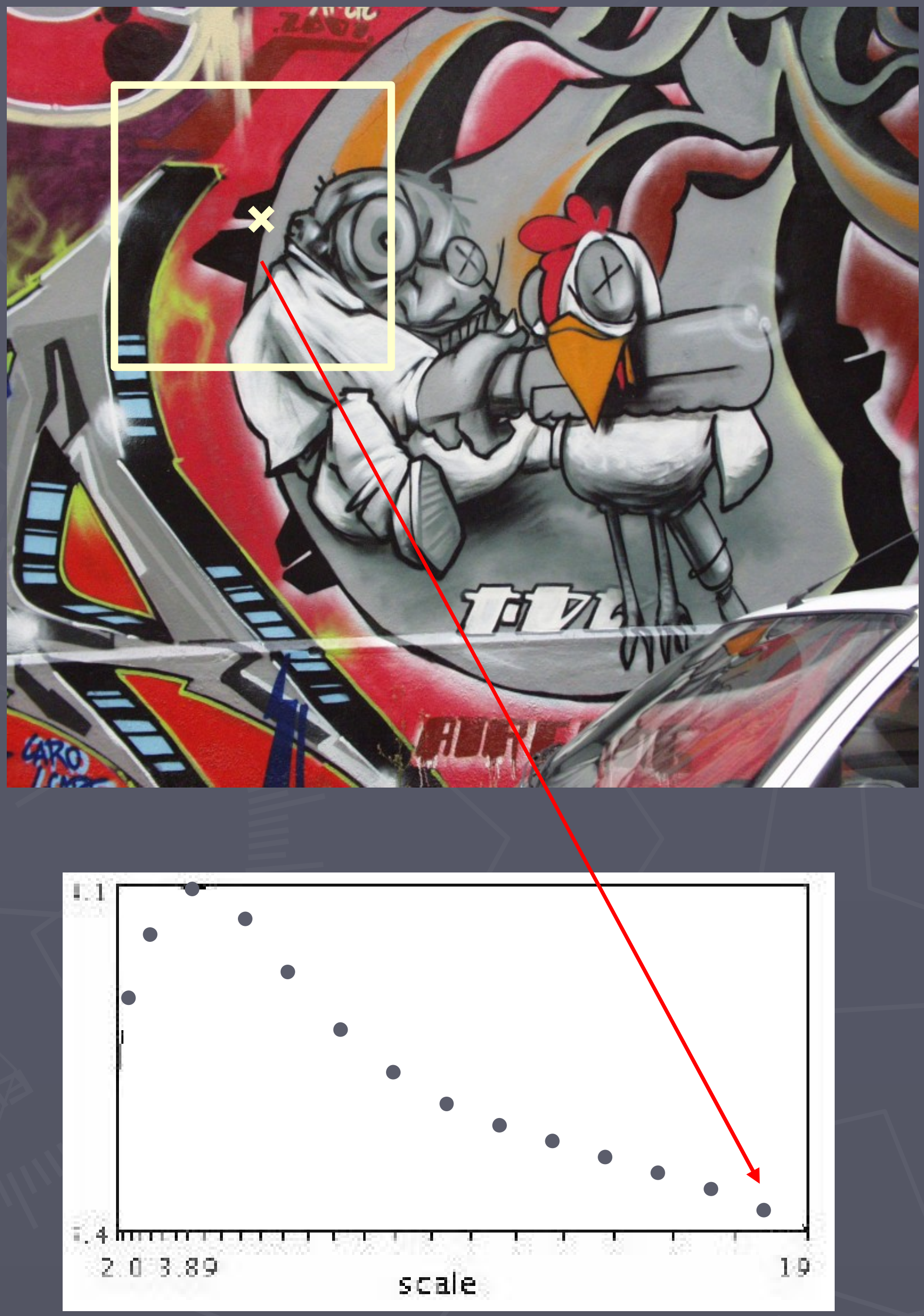
Scale trace (signature)



$f(I_{i_1 \ldots i_m}(x,\sigma))$

Source: Tuytelaars

Automatic scale selection



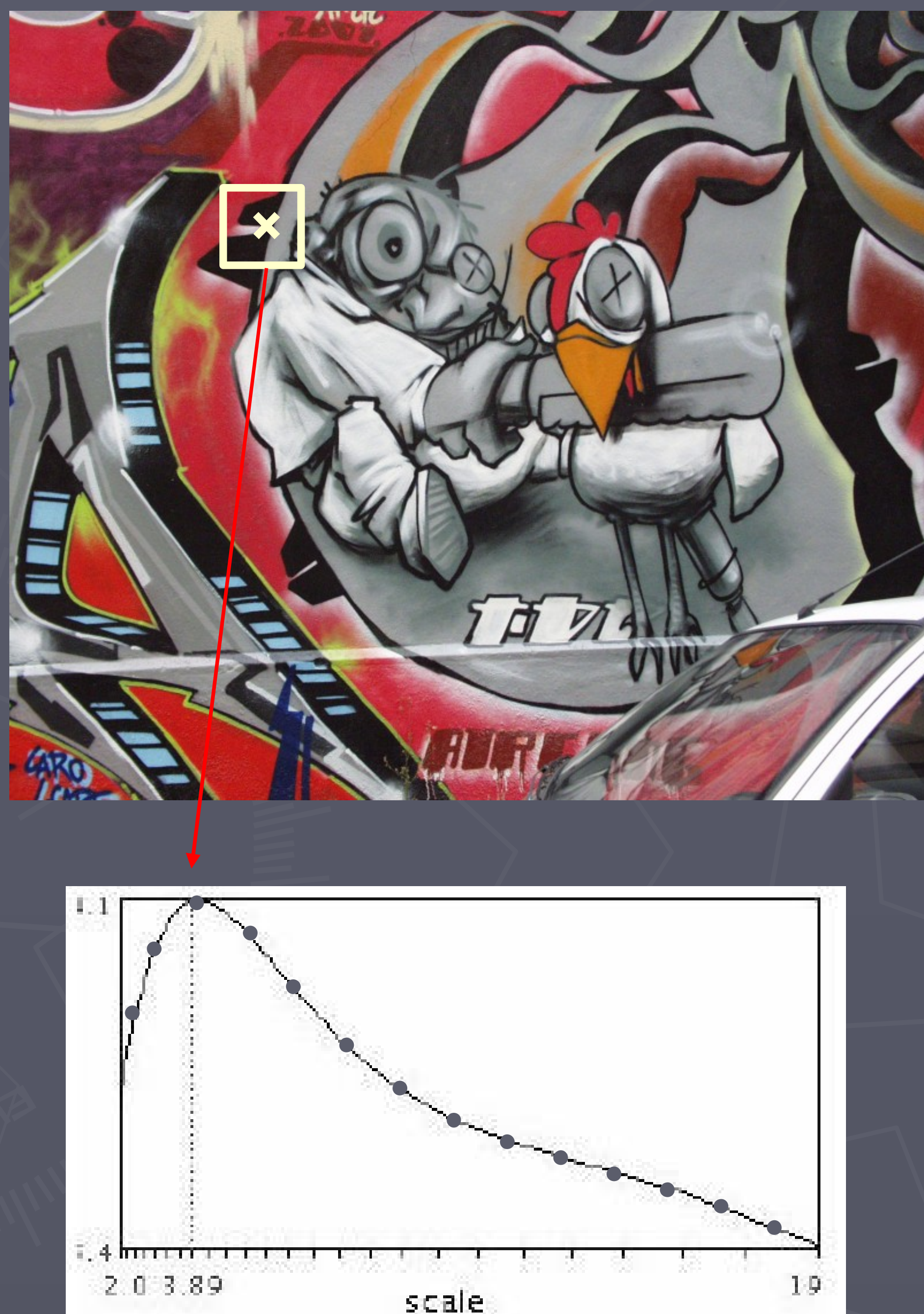
$f(I_{i_1 \ldots i_m}(x,\sigma))$

Source: Tuytelaars

Automatic scale selection

Function responses for increasing scale

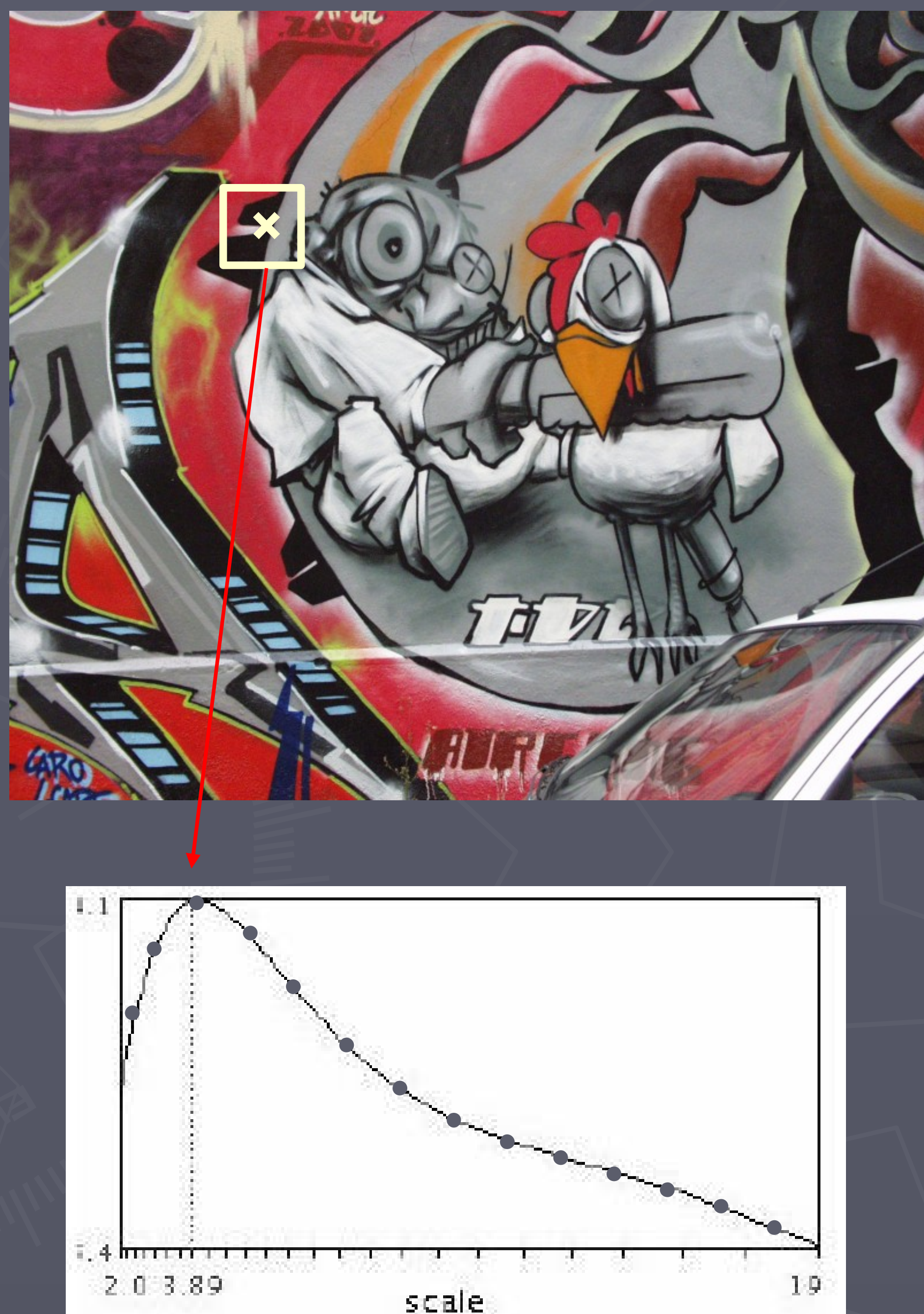
Scale trace (signature)



$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

Source: Tuytelaars

Automatic scale selection



$f(I_{i_1 \ldots i_m}(x,\sigma))$

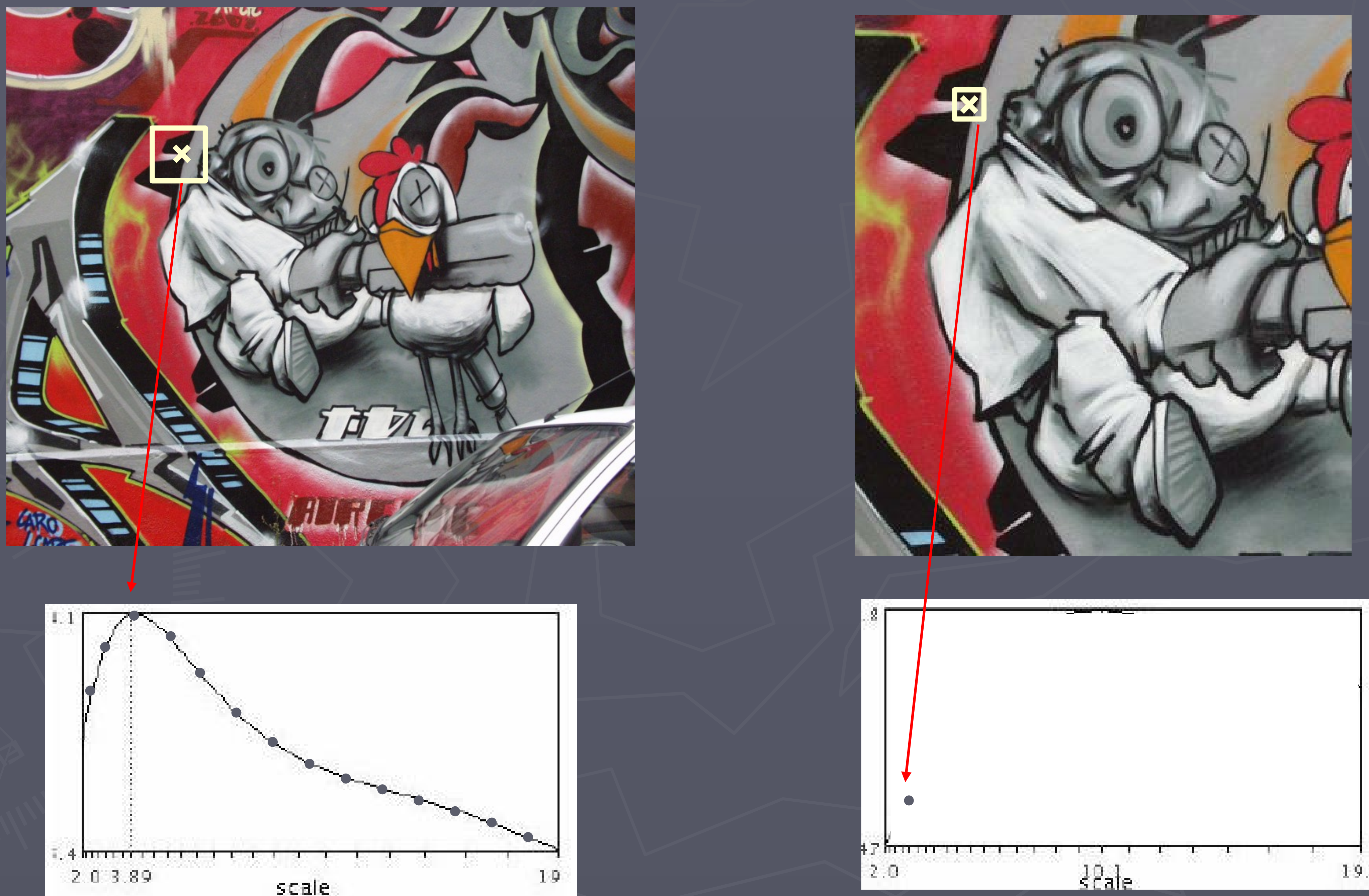
$f(I_{i_1 \ldots i_m}(x',\sigma))$

Source: Tuytelaars

Automatic scale selection

Function responses for increasing scale

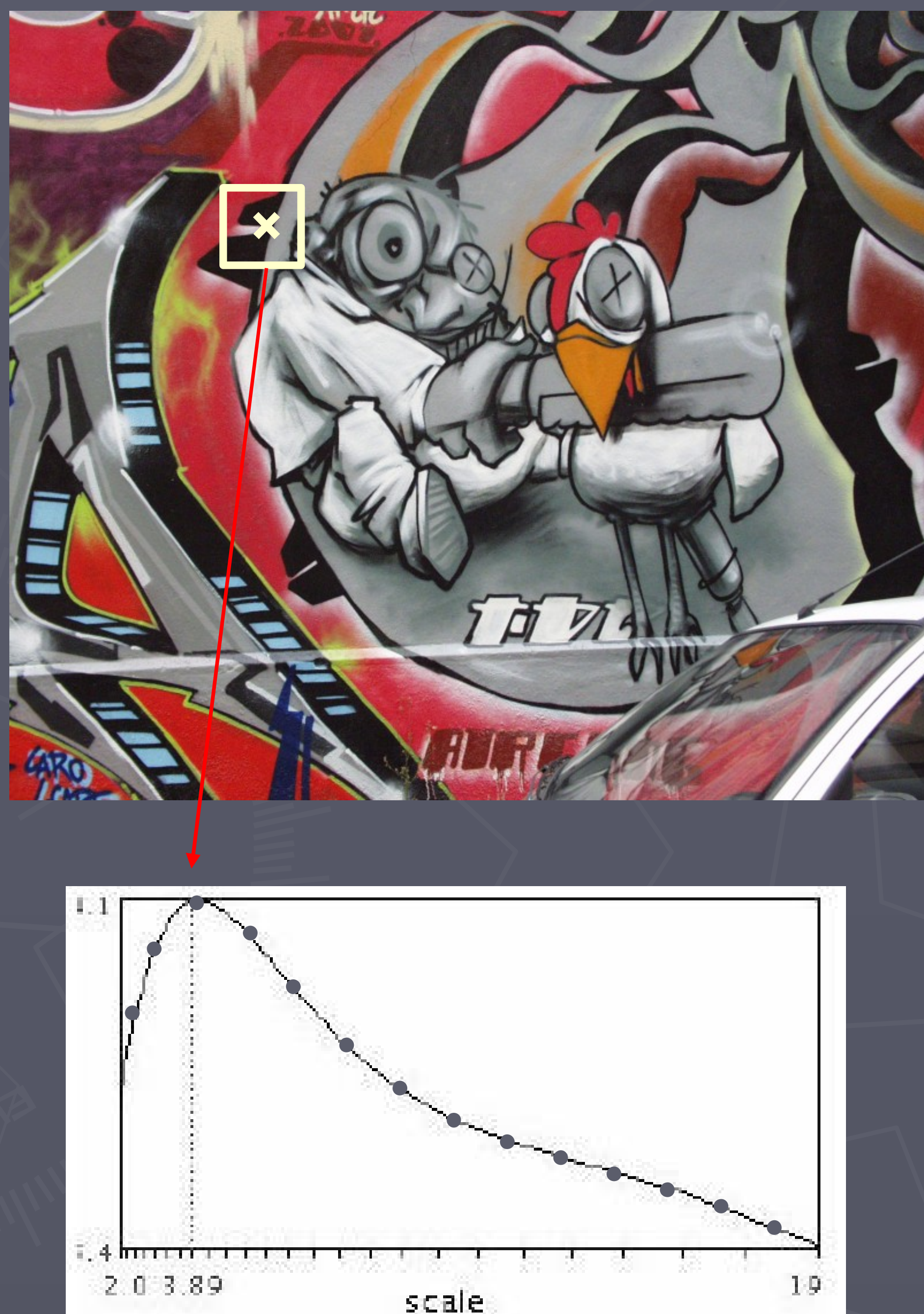
Scale trace (signature)



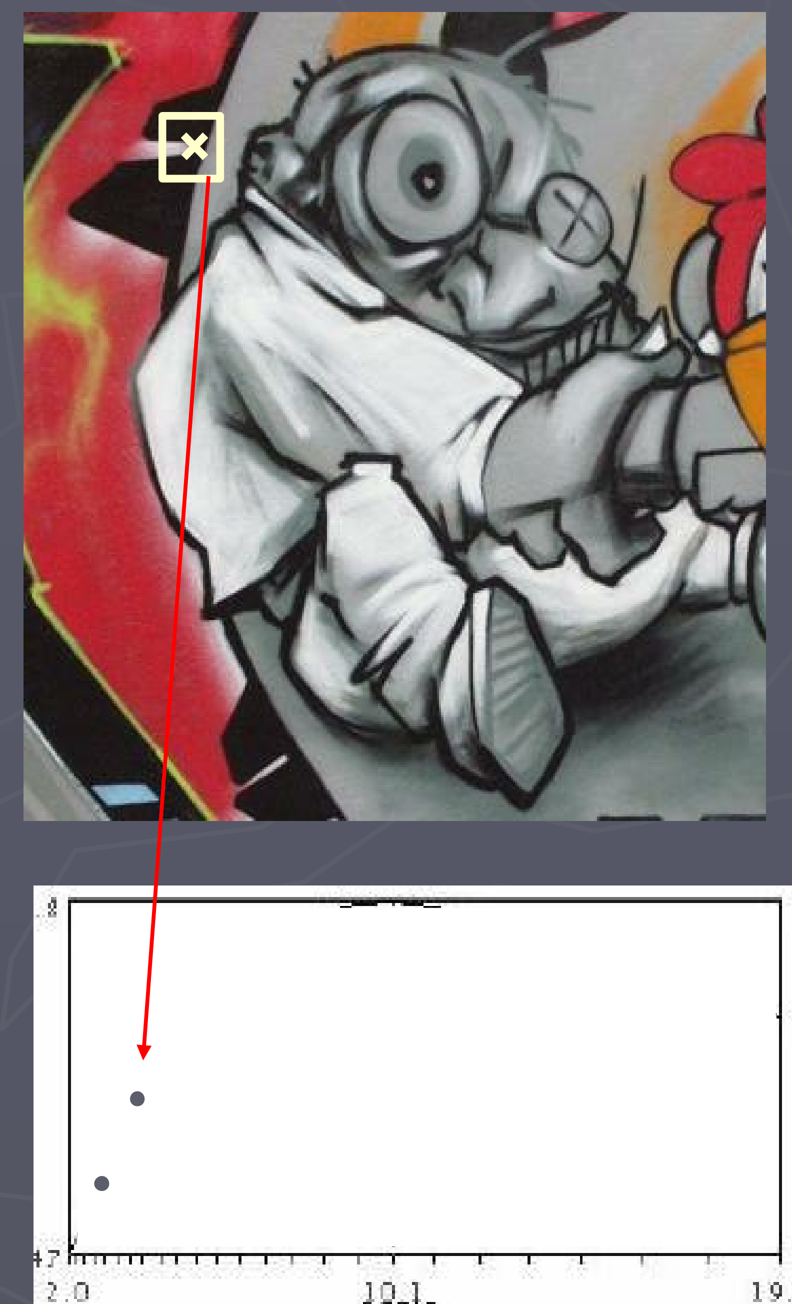
$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

Source: Tuytelaars

Automatic scale selection



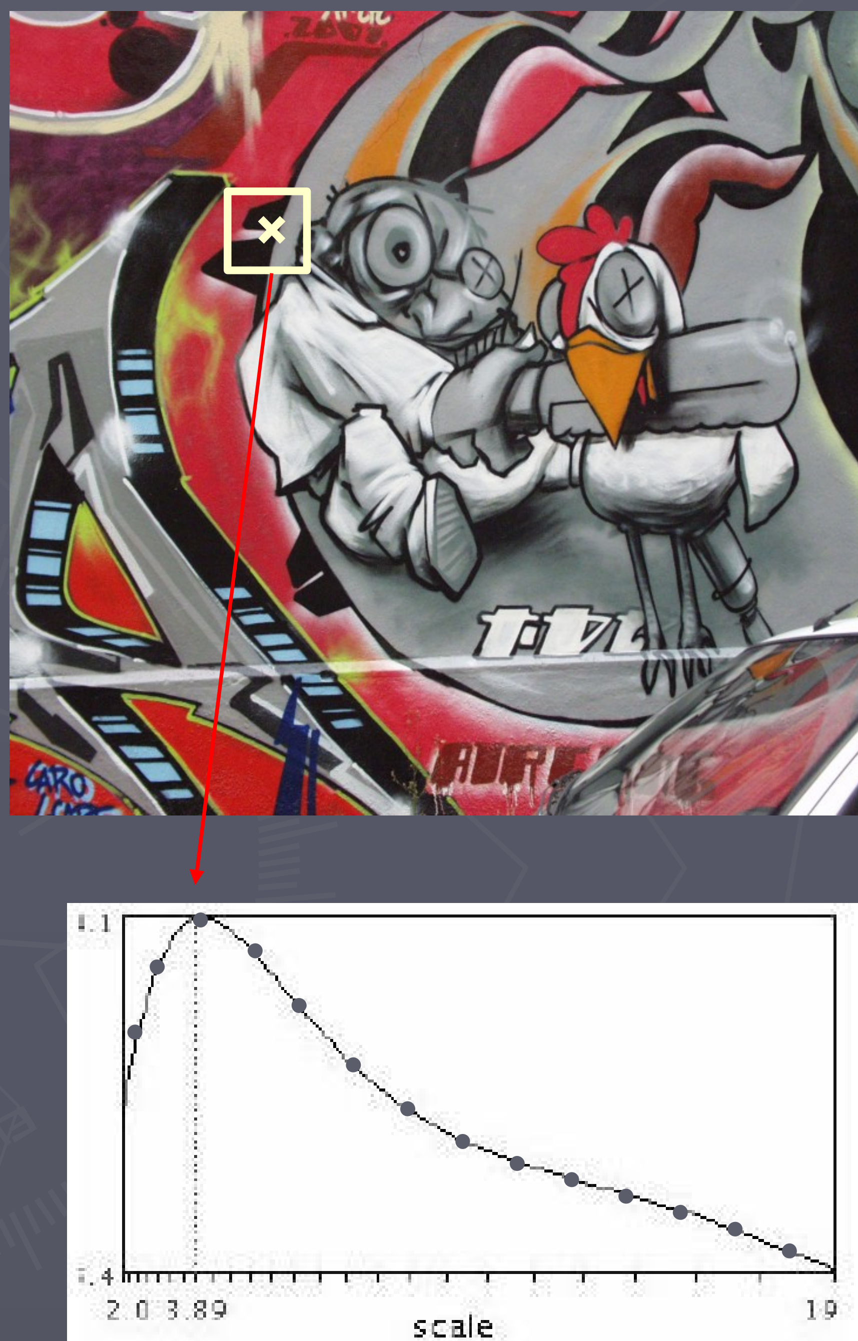
$f(I_{i_1 \ldots i_m}(x,\sigma))$



$f(I_{i_1 \ldots i_m}(x',\sigma))$

Source: Tuytelaars

Automatic scale selection

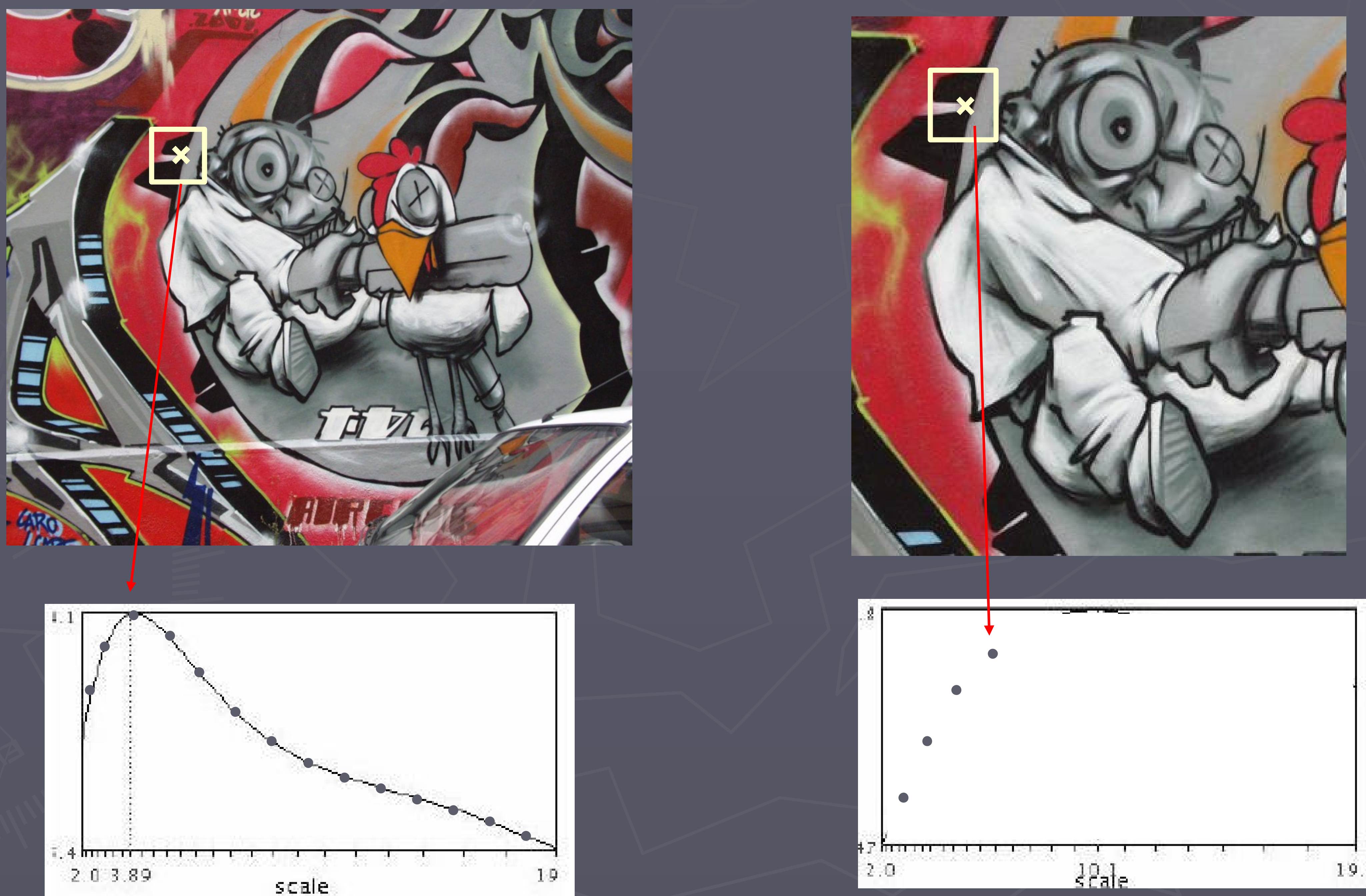


$f(I_{i_1 \ldots i_m}(x, \sigma))$

$f(I_{i_1 \ldots i_m}(x', \sigma))$

Source: Tuytelaars

Automatic scale selection



$f(I_{i_1 \ldots i_m}(x,\sigma))$

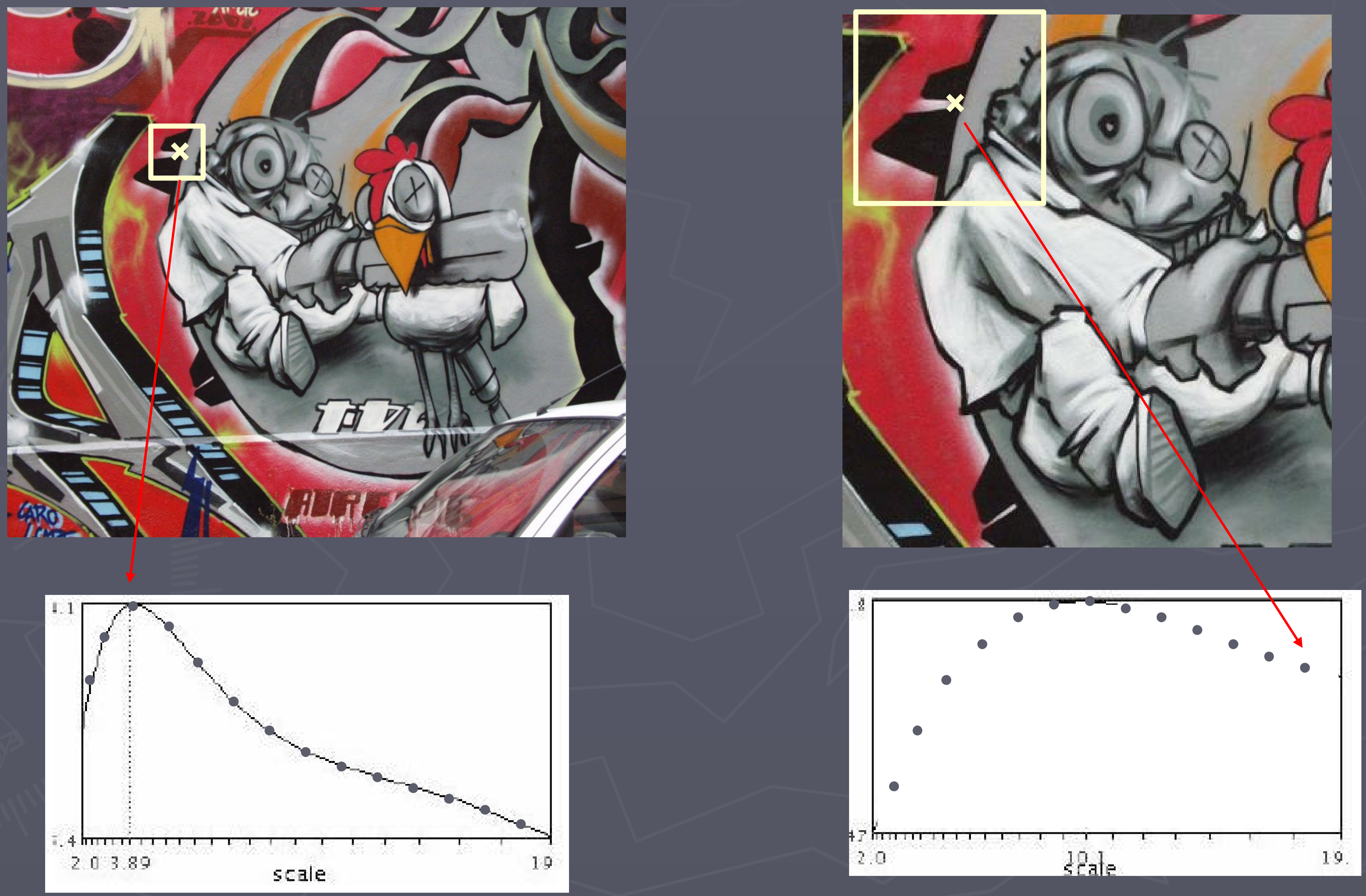
$f(I_{i_1 \ldots i_m}(x',\sigma))$

Source: Tuytelaars

Automatic scale selection

Function responses for increasing scale

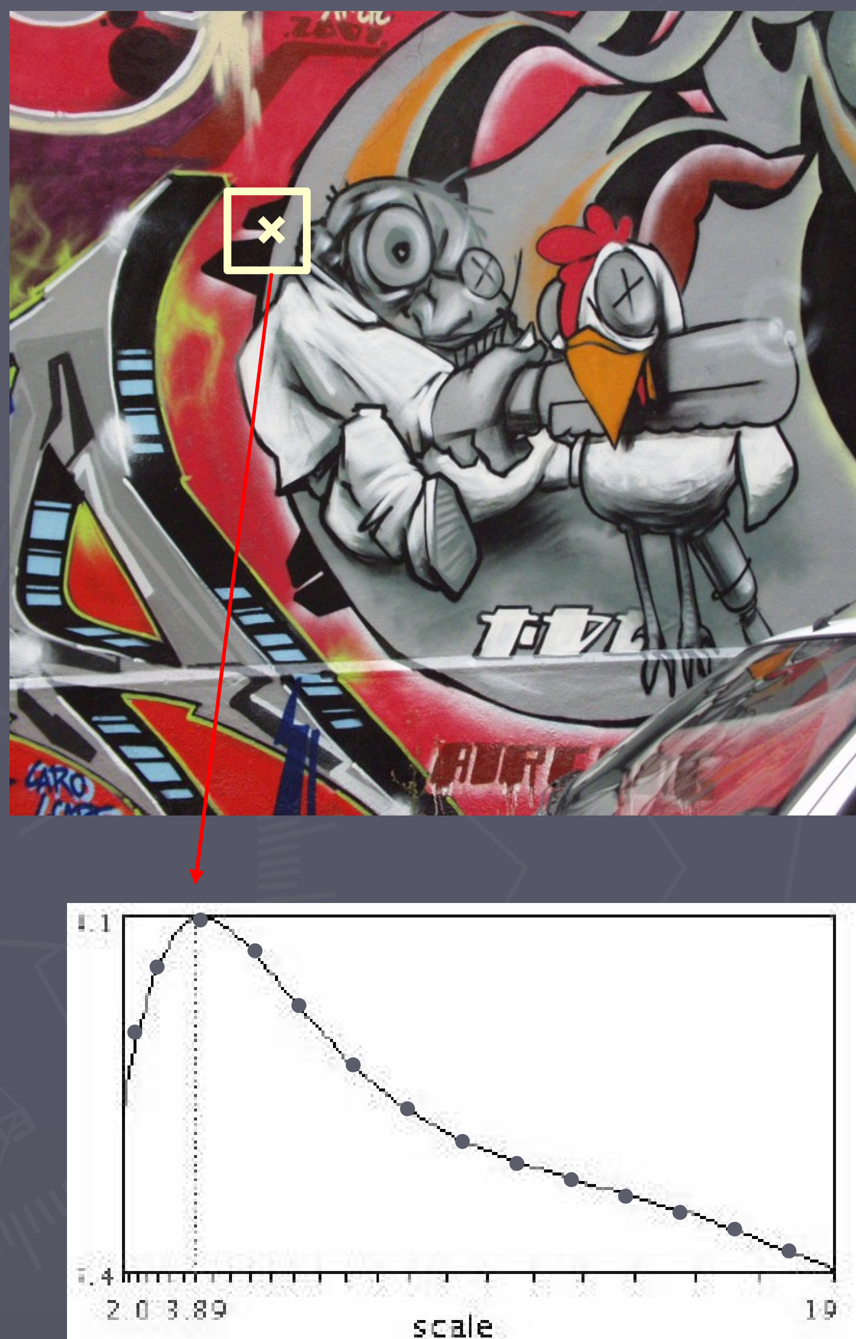
Scale trace (signature)



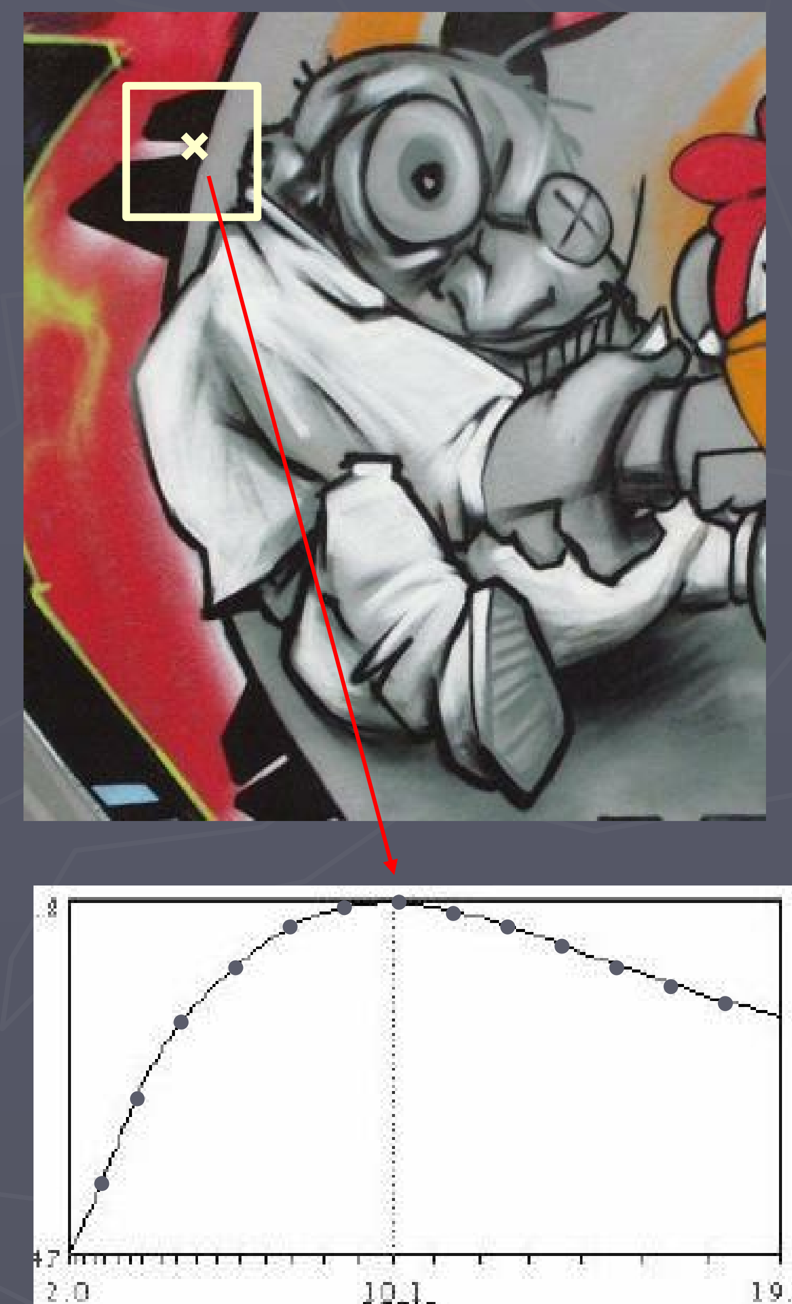
$f(I_{i_1 \ldots i_m}(x,\sigma))$ $\quad$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

Source: Tuytelaars

Automatic scale selection



$f(I_{i_1 \dots i_m}(x, \sigma))$

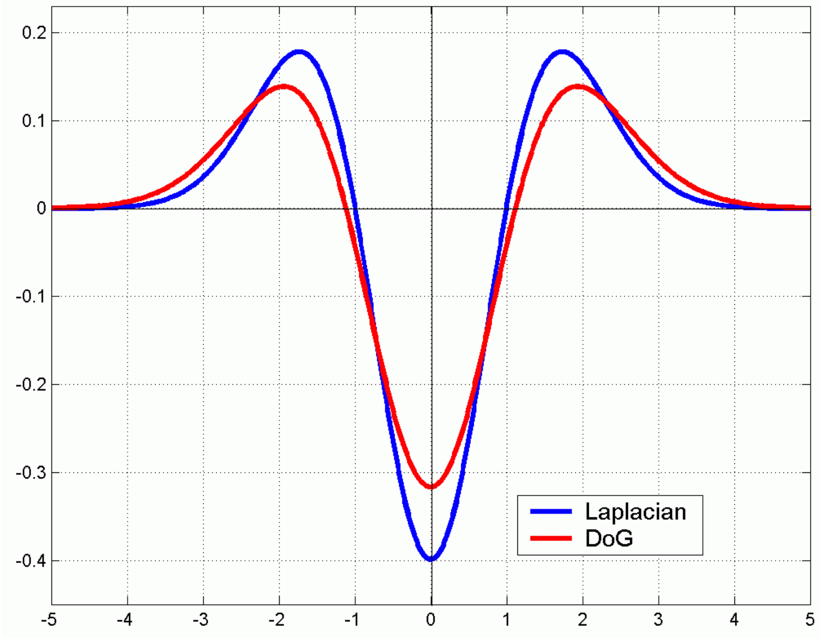


$f(I_{i_1 \dots i_m}(x', \sigma'))$

Source: Tuytelaars

Popular Kernels: Laplacian of Gaussians

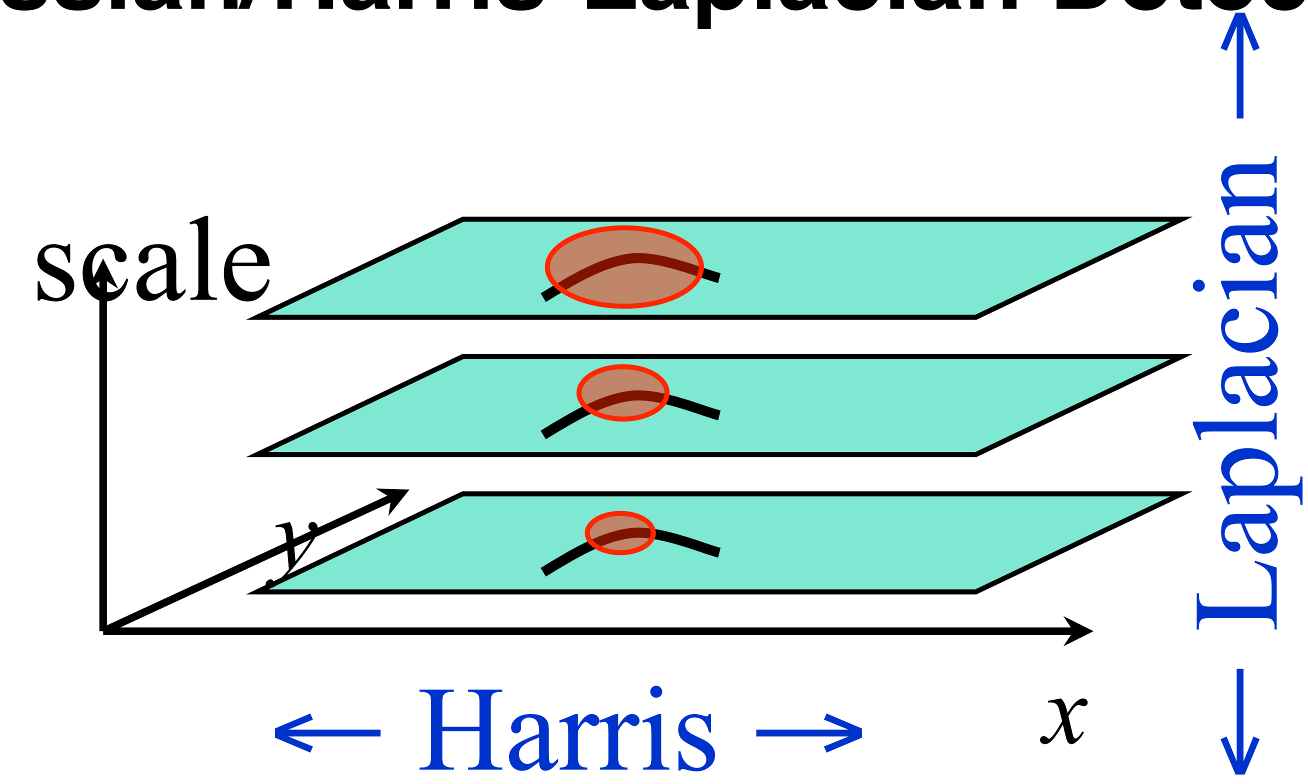
$$w(\sigma) * I$$



Gaussian $$G(x, y; \sigma) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Laplacian of Gaussians $$L(\sigma) = \sigma^2 (G_{xx}(x, y; \sigma) + G_{yy}(x, y; \sigma))$$

Hessian/Harris-Laplacian Detector



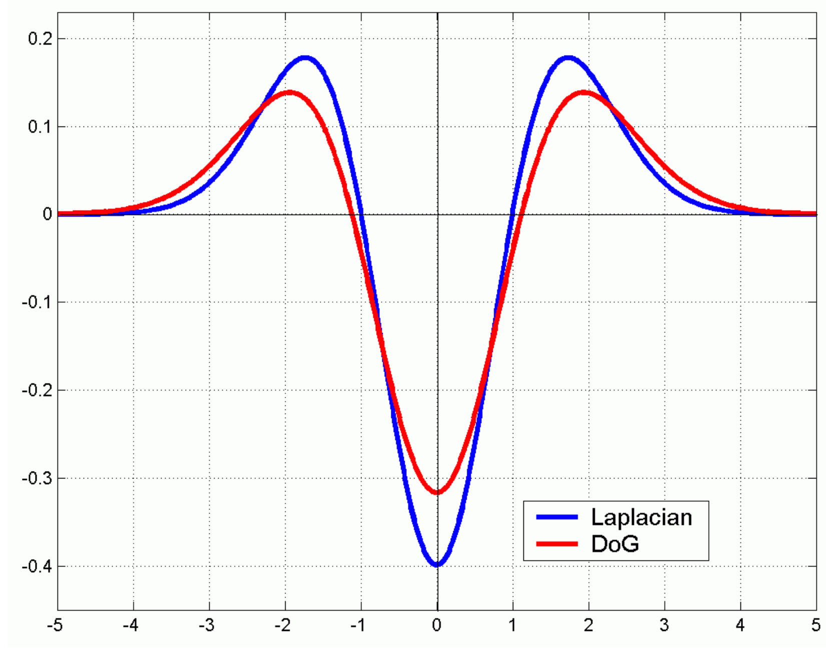
- Find a local maximum of Hessian/Harris function

$$E(x,y;\sigma) = \sigma^2 (G_{xx}(x,y;\sigma) + G_{yy}(x,y;\sigma)) * I(x,y)$$

- simultaneously in :
  - 2D space of the image
  - Scale

Popular Kernels: Difference of Gaussians

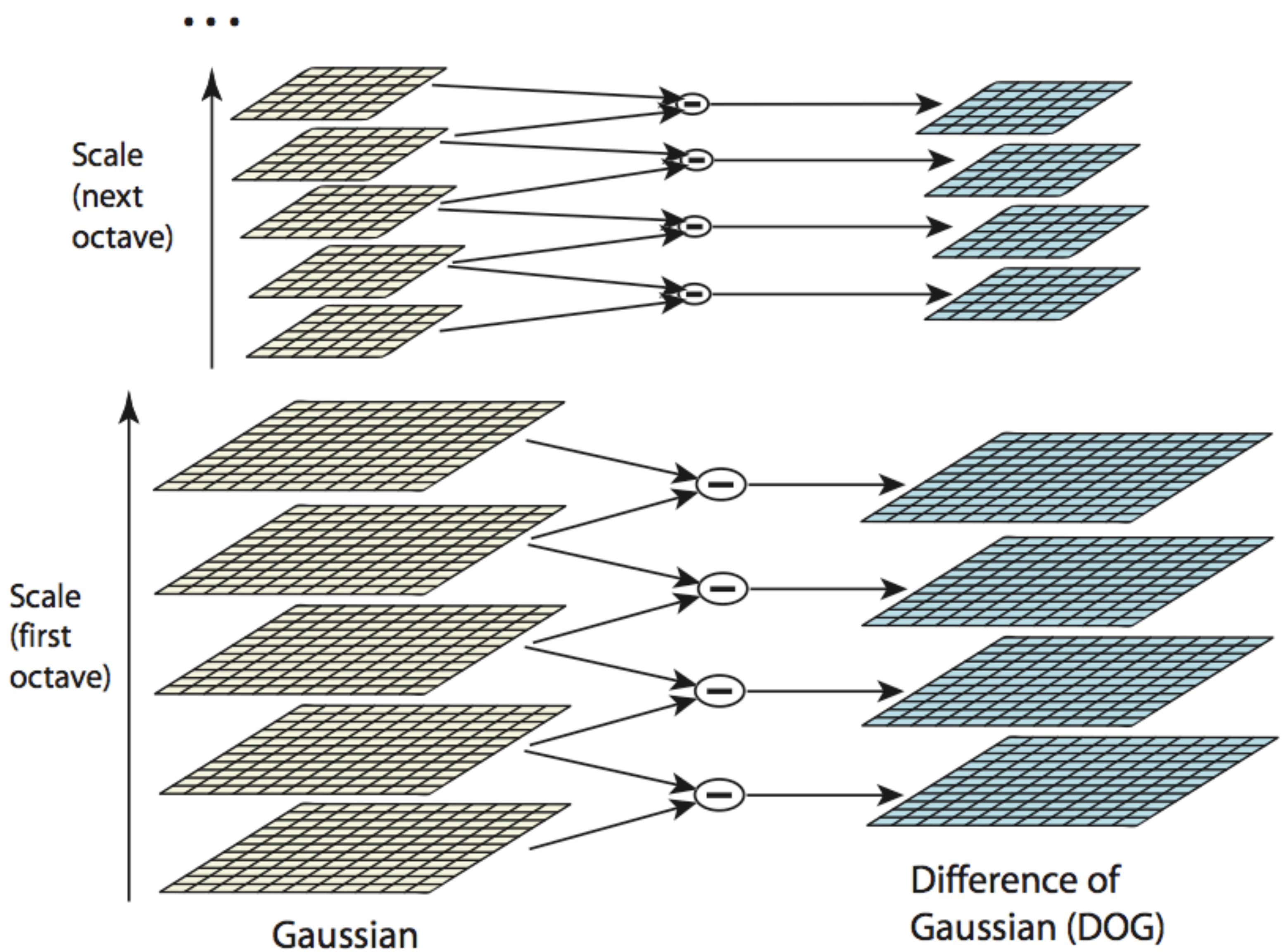
$$w(\sigma) * I$$



Gaussian $$G(x,y;\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

Difference of Gaussians $$DoG(\sigma) = G(x,y;k\sigma) - G(x,y;\sigma)$$

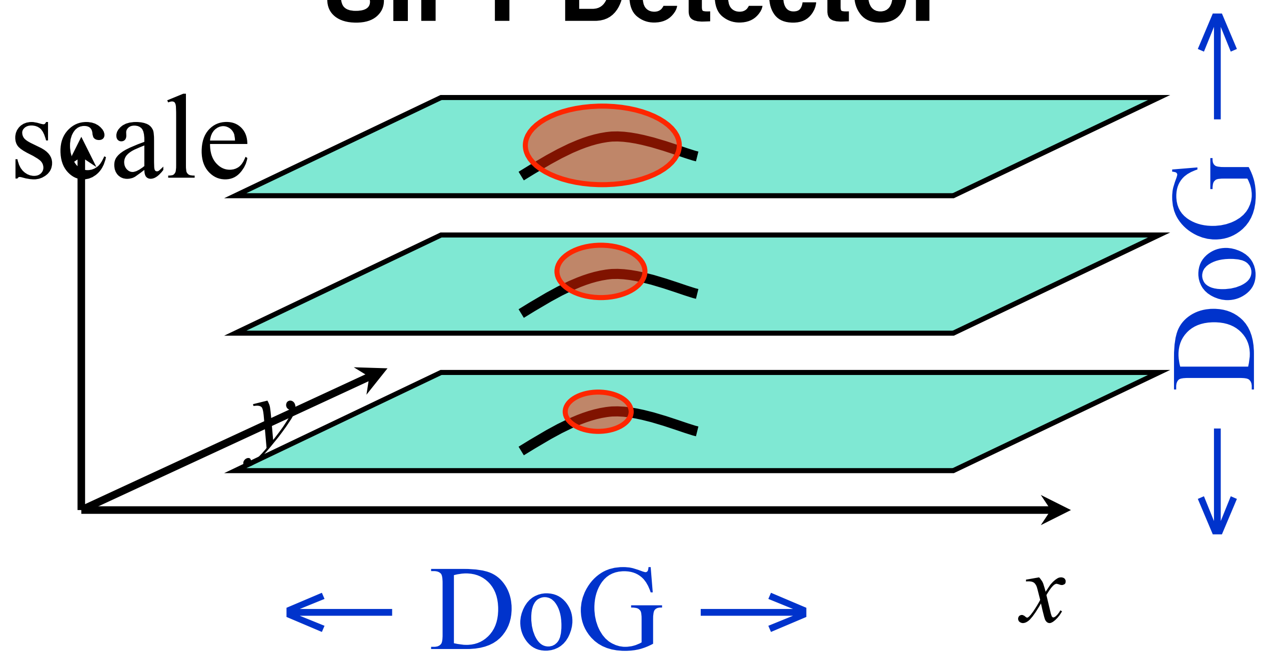
Selected Scales = Extrema of DoG/Laplacian



- Convolve the image with Gaussians whose sigma increases
- Then, subsample, and repeat the convolutions
- Finally, find extrema in the 3D DoG or Laplacian space

Source: D. Lowe

SIFT Detector

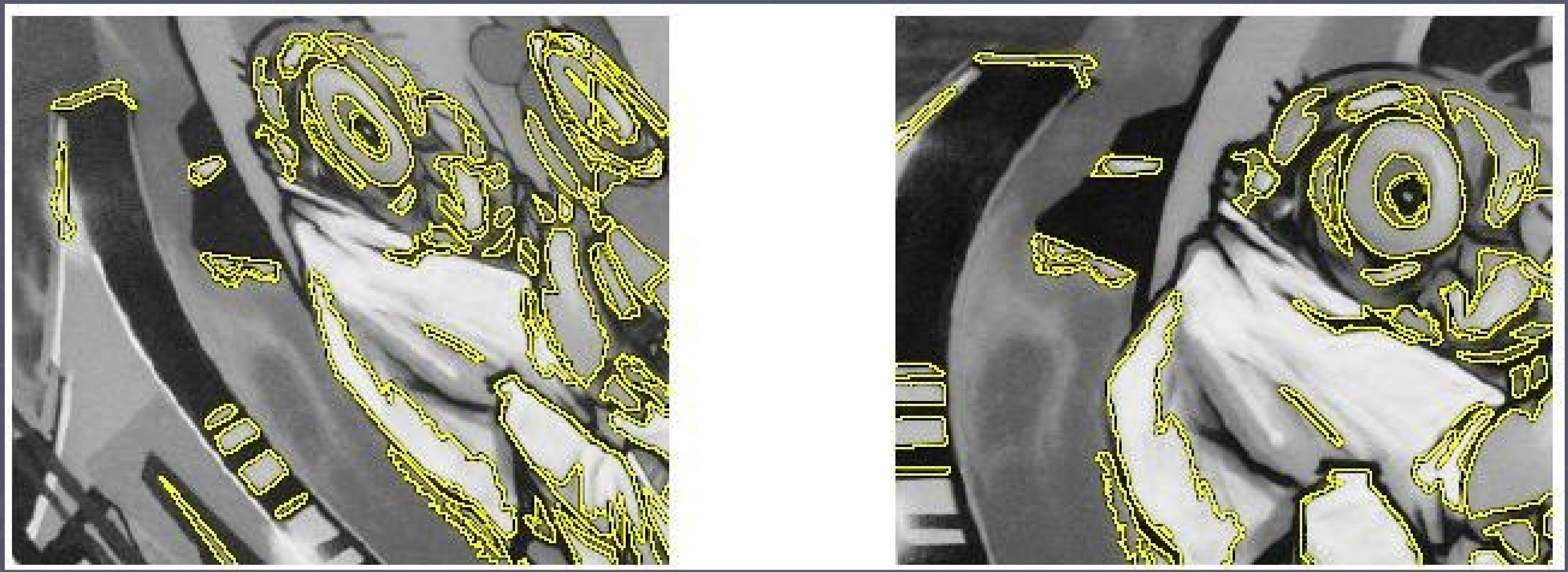


- Find a local maximum of

$$E(x, y; \sigma) = \mathrm{DOG}(x, y; \sigma) * I(x, y)$$

- simultaneously in:
  - 2D space of the image
  - Scale

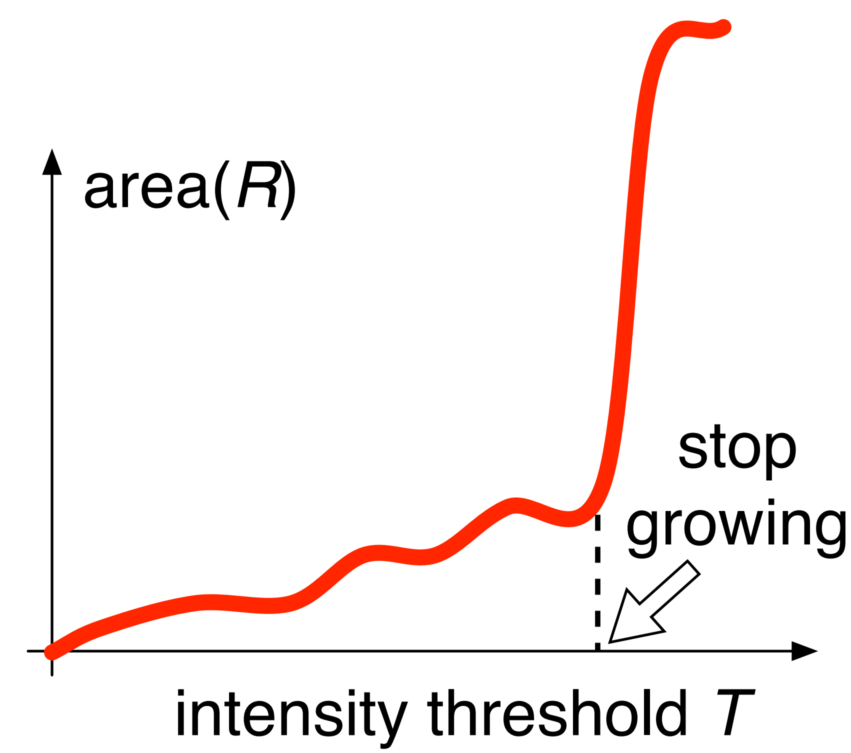
Maximally Stable Extremal Regions



Source: Tuytelaars

Maximally Stable Extremal Region Detector

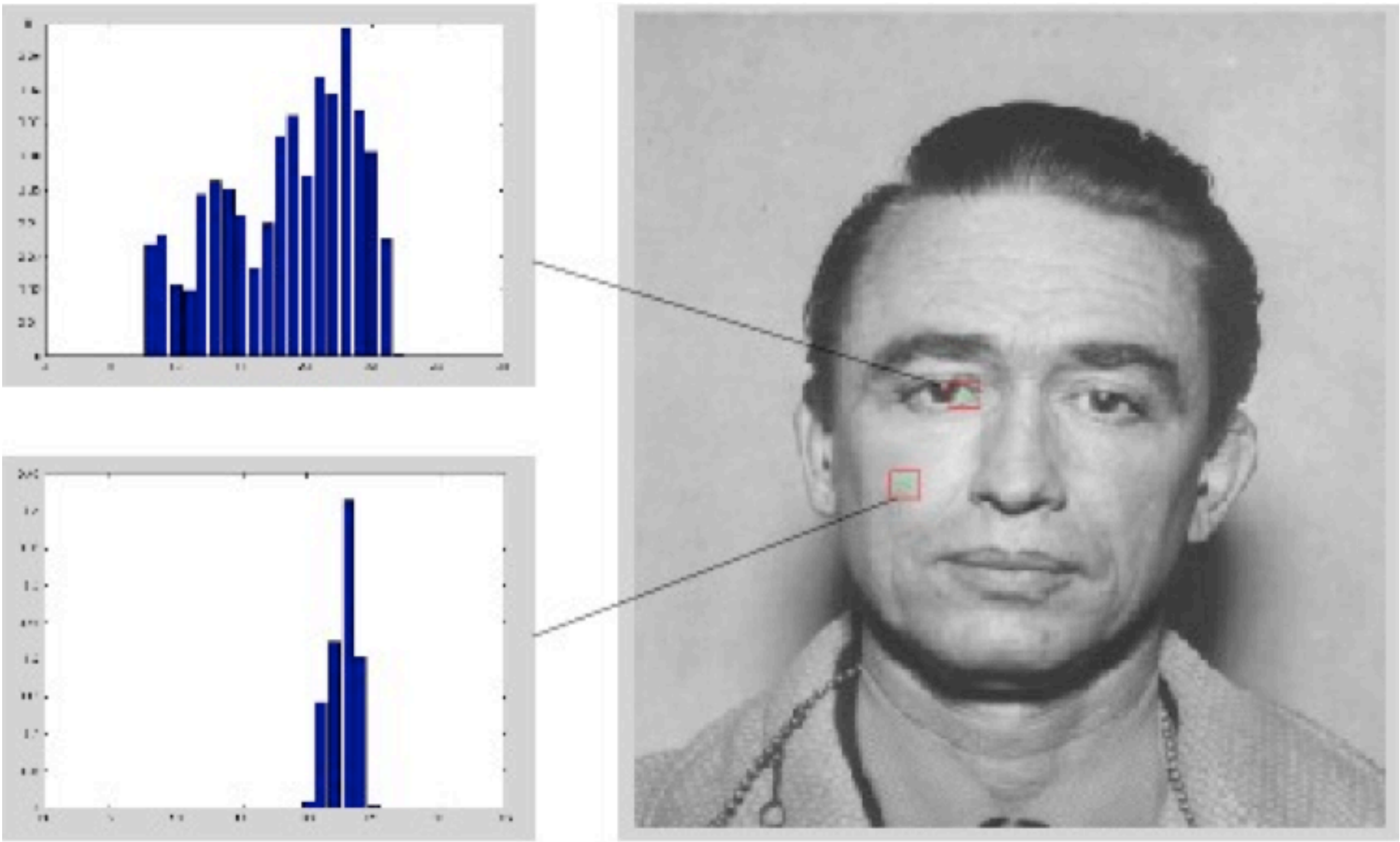
1. Sort pixels by intensities and place them in a stack *S*
2. Pop up the first pixel *p* from *S*
3. Start growing region *R* around *p* from adjacent pixels
    - 3.1. Increment the contrast threshold *T* from black to white
    - 3.2. For each *T* include in *R* all adjacent pixels to *p*
    - 3.3. Monitor a plot of area(*R*) vs. *T*
    - 3.4. If "large jump" in area(*R*), stop
4. Put *R* in the solution
5. Delete all pixels of *R* from *S*
6. If *S* is not empty, go to step 2

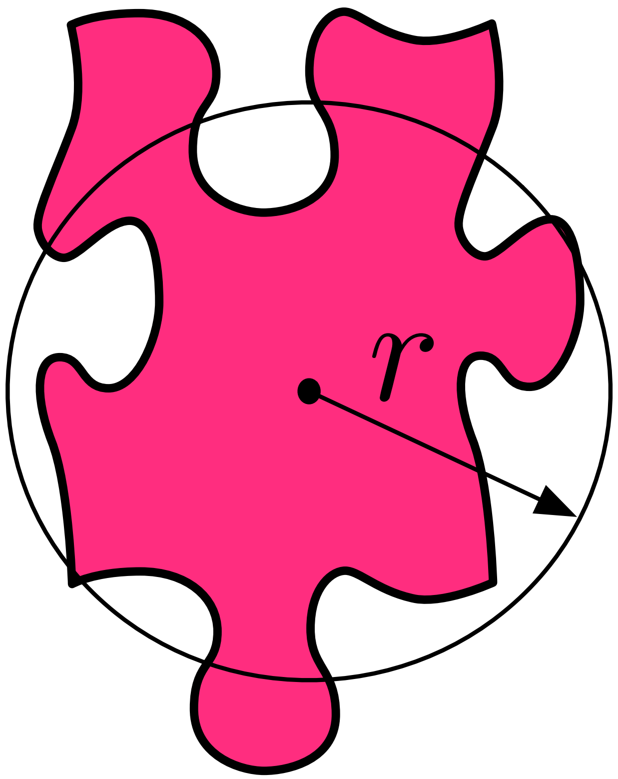


Kadir-Brady Saliency Detector

Saliency of a region = Function of entropy of pixels in that region

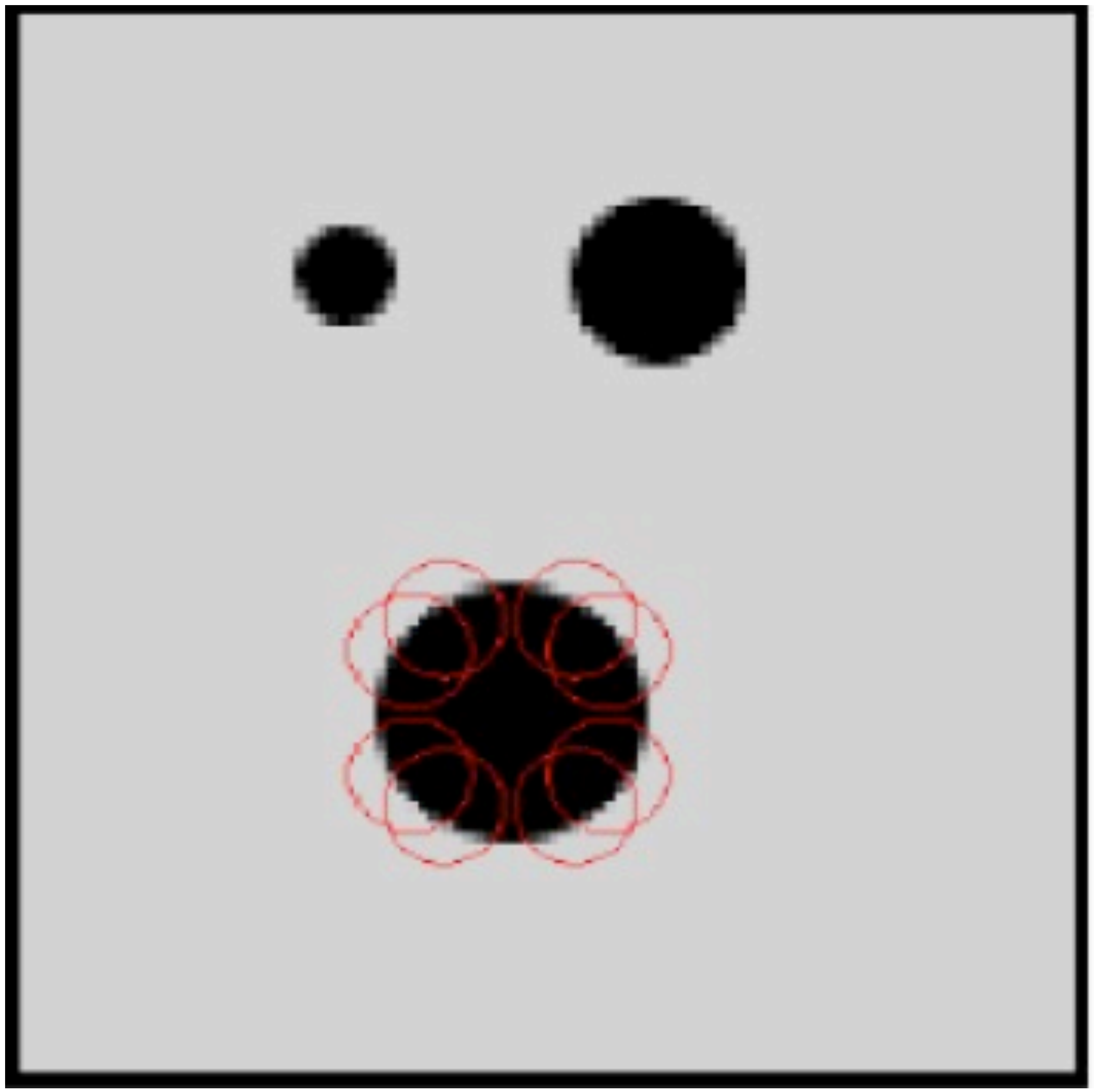
$$H_R(\sigma, \boldsymbol{x}) = -\int_{r \in R} p(r, \sigma, \boldsymbol{x}) \log p(r, \sigma, \boldsymbol{x}) dr$$





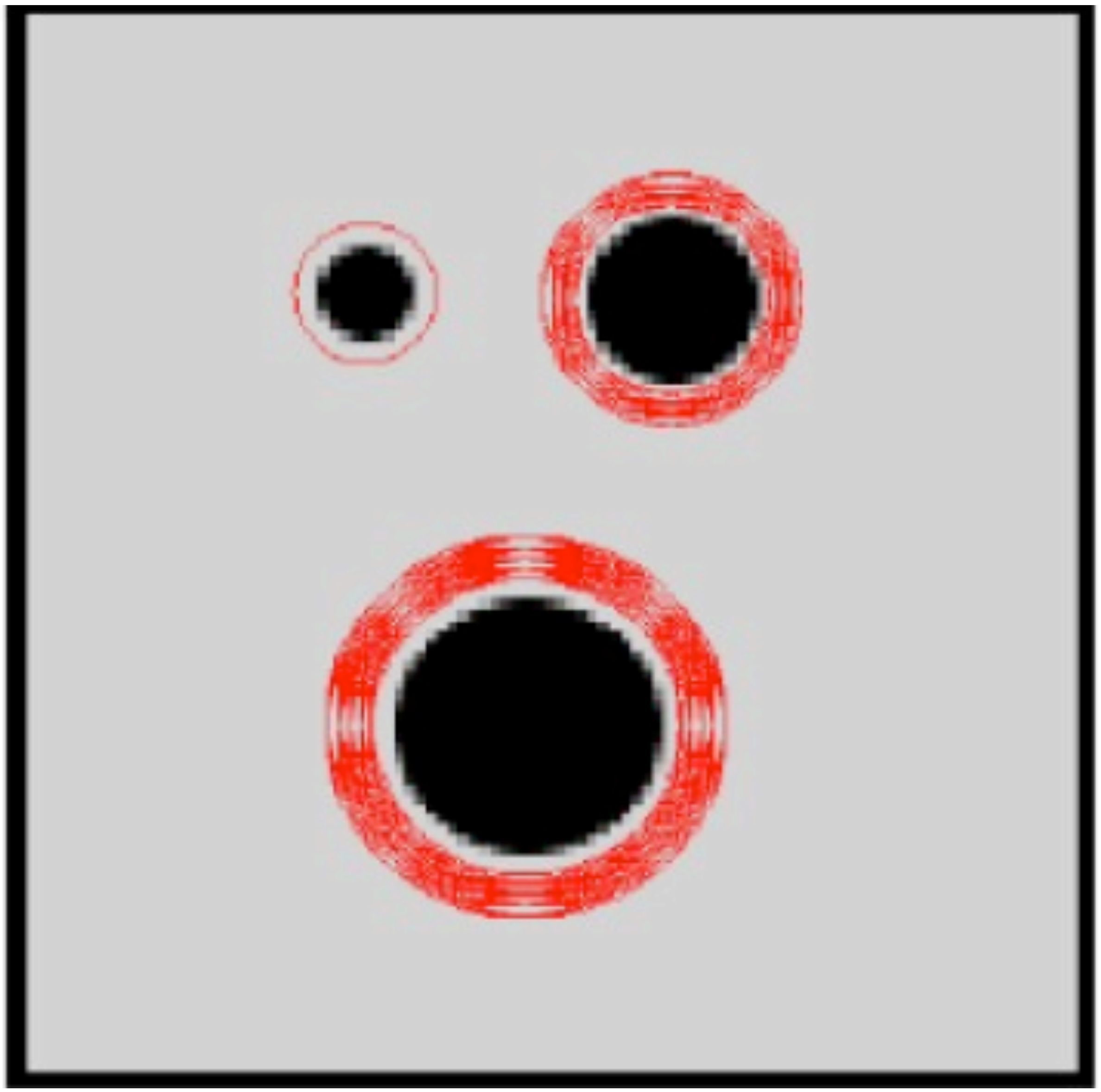
histogram ≈ pdf of pixels

Kadir-Brady Saliency Detector



detection using:

$$H_R(\sigma, \boldsymbol{x})$$



detection using:

$$Y_R(\sigma, \boldsymbol{x}) = H_R(\sigma, \boldsymbol{x}) \cdot W_R(\sigma, \boldsymbol{x})$$

Kadir-Brady Saliency Detector

- Salient region has a large
  - Entropy
  - Change of pdf across scales

Region saliency:

$$Y_R(\sigma, \boldsymbol{x}) = H_R(\sigma, \boldsymbol{x}) \cdot W_R(\sigma, \boldsymbol{x})$$

Change of pdf across scales:

$$W_R(\sigma, \boldsymbol{x}) = \sigma \left| \int_{r \in R} \frac{\partial p(r, \sigma, \boldsymbol{x})}{\partial \sigma} dr \right|$$

Next Class

- Other interest points
- Point descriptors