INFO - H - 501

Pattern recognition and image analysis

Detectors

Using Photographs to Enhance Videos of a Static Scene

Pravin Bhat, C. Lawrence Zitnick, Noah Snavely, Aseem Agarwala
Maneesh Agrawala, Michael Cohen, Brian Curless, Sing Bing Kang

what are the different techniques involved ?

what are the different techniques involved ?

- detection of specific points
- point pairing
- pose estimation
- 3D reconstruction
- structure from motion
- graph cut algorithm

Tracking

- object tracking (CCTV sequences, speed measurement)
- pose estimation
- co-registration
- stereo-vision
- 3D from motion
- object recognition
- robotic mapping and navigation (SLAM)
- image stitching
- 3D modeling
- gesture recognition,
- ...

Detectors

- Pattern matching
-

Pattern matching

- find a pattern *h(i,j)* in the *f(i,j)* image
- matching criteria

$$C_1(u,v) = \frac{1}{\max_{(i,j)\in V} |f(i+u,j+v) - h(i,j)|}$$

$$C_2(u,v) = \frac{1}{\sum_{(i,j)\in V} |f(i+u,j+v) - h(i,j)|}$$

$$C_3(u,v) = \frac{1}{\sum_{(i,j)\in V} [f(i+u,j+v) - h(i,j)]^2}$$

Pattern matching

Detectors

- Harris corner detection [Haris88]
- Fast [Rosten05]
- sift, surf, others
- DoG pyramid [HCVA] vol2 p 240
- Robust object detection [Viola01]
- Hough transform [IPAMV] p149

Recalls

- gradient

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$$

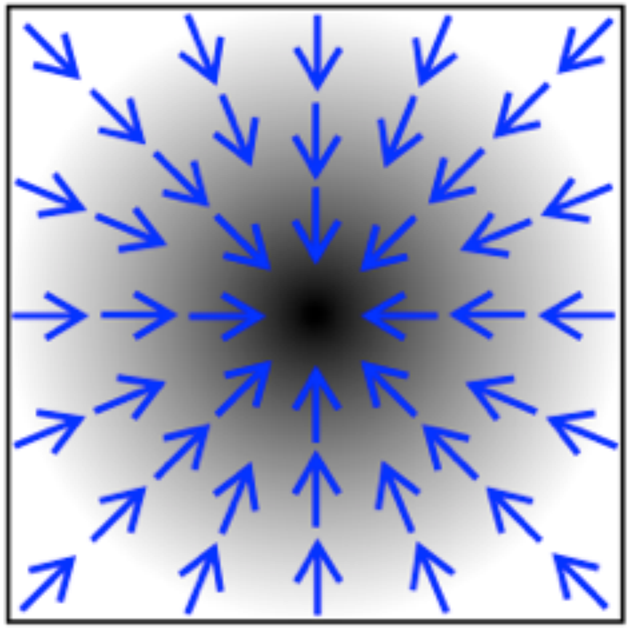
- gradient magnitude

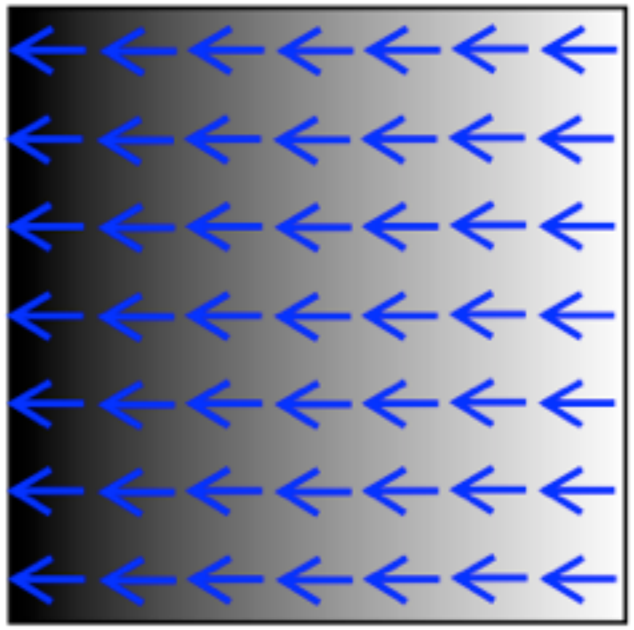
$$\|\nabla f\|^2 = \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2$$

- Laplacian

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- finite difference approx.





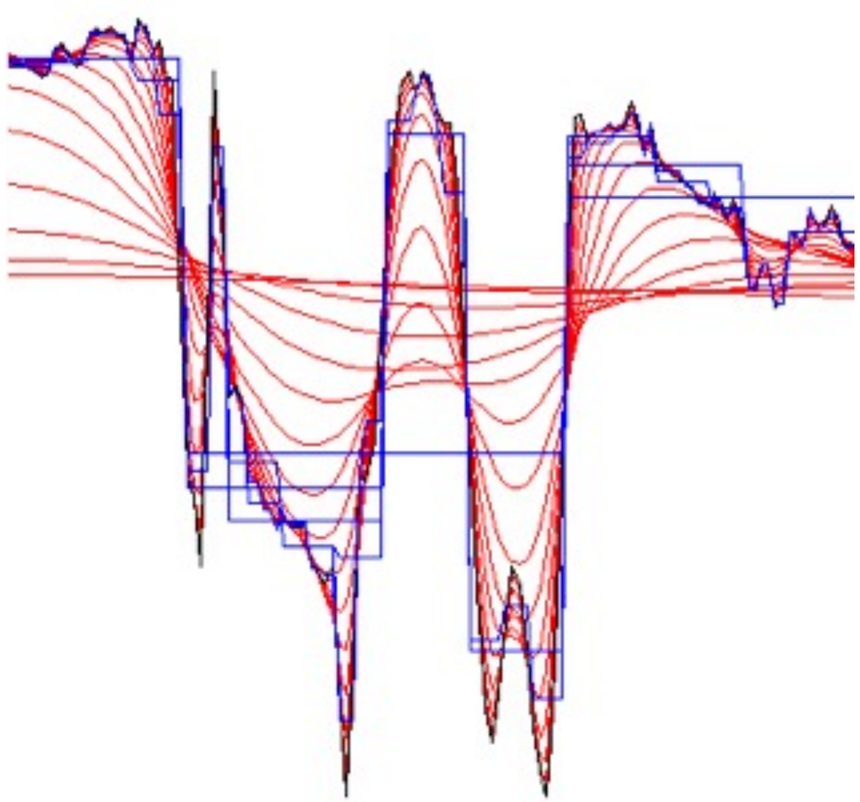
Recalls

- Smoothing filter
  - ex: gaussian filter

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

Border detection

- Marr-Hildreth
  - zero crossing detection $\nabla^2(G_\sigma * I)$
- Laplacian of Gaussian $(\nabla^2 G_\sigma) * I$
- Multi-scale $I(x, y, \sigma) = I(x, y) * G_\sigma(x, y)$



wikipedia

Border detection

- difference of gaussian

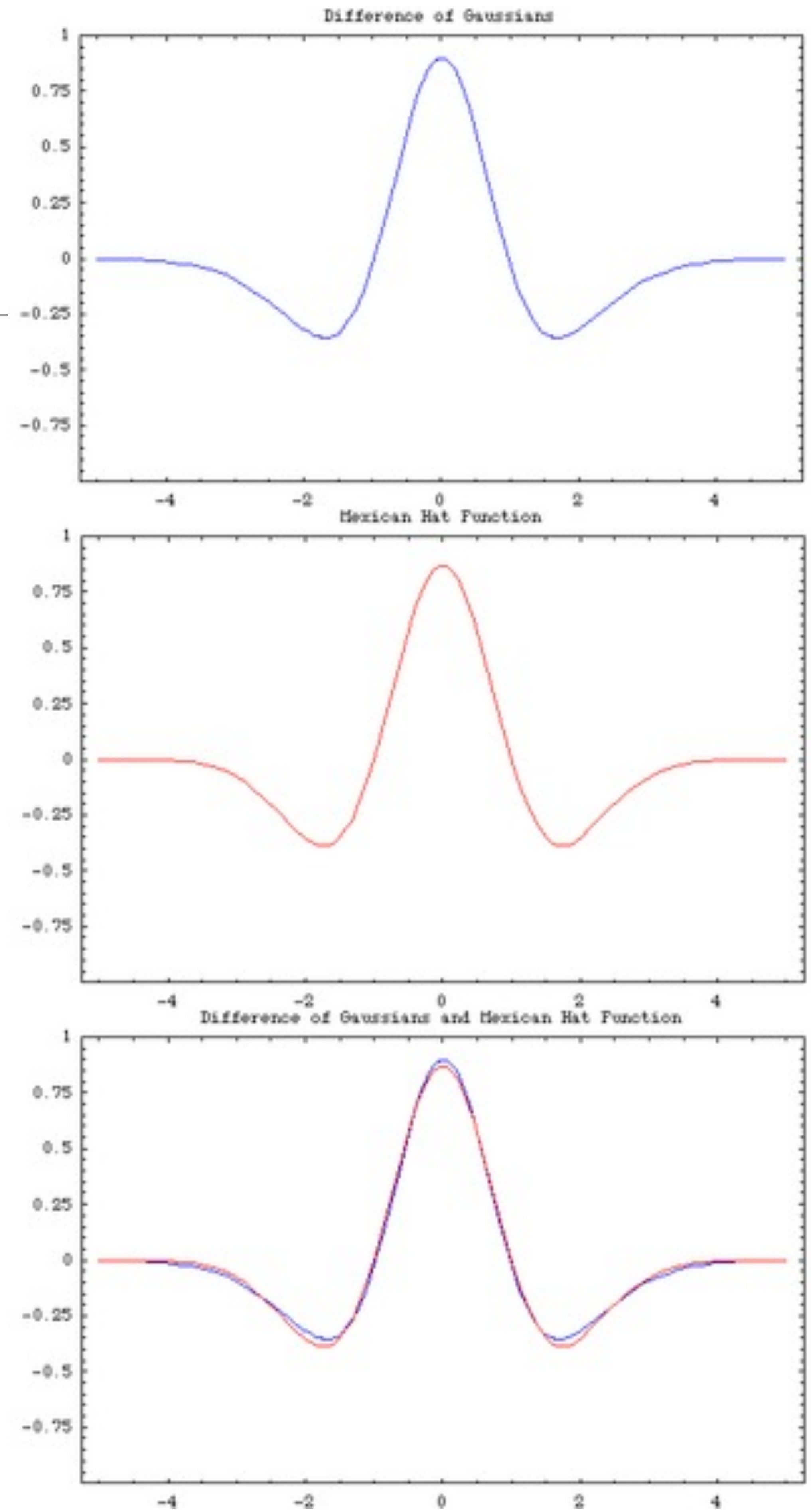
$$f(x;\mu,\sigma_1,\sigma_2) = \frac{1}{\sigma_1\sqrt{2\pi}}\,\exp\left(-\frac{(x-\mu)^2}{2\sigma_1^2}\right) - \frac{1}{\sigma_2\sqrt{2\pi}}\,\exp\left(-\frac{(x-\mu)^2}{2\sigma_2^2}\right)$$

- band pass filter

- approximate Laplacian of Gaussian
  (ratio 1.6)

Border detection

- approximate Laplacian of Gaussian (ratio 1.6)



wikipedia

Border detection

- Canny edge detection
  - low pass gaussian filtering
  - compute gradient and local border orientation

$$\begin{aligned}\mathbf{G} &= \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2} \\ \mathbf{\Theta} &= \arctan\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)\end{aligned}$$

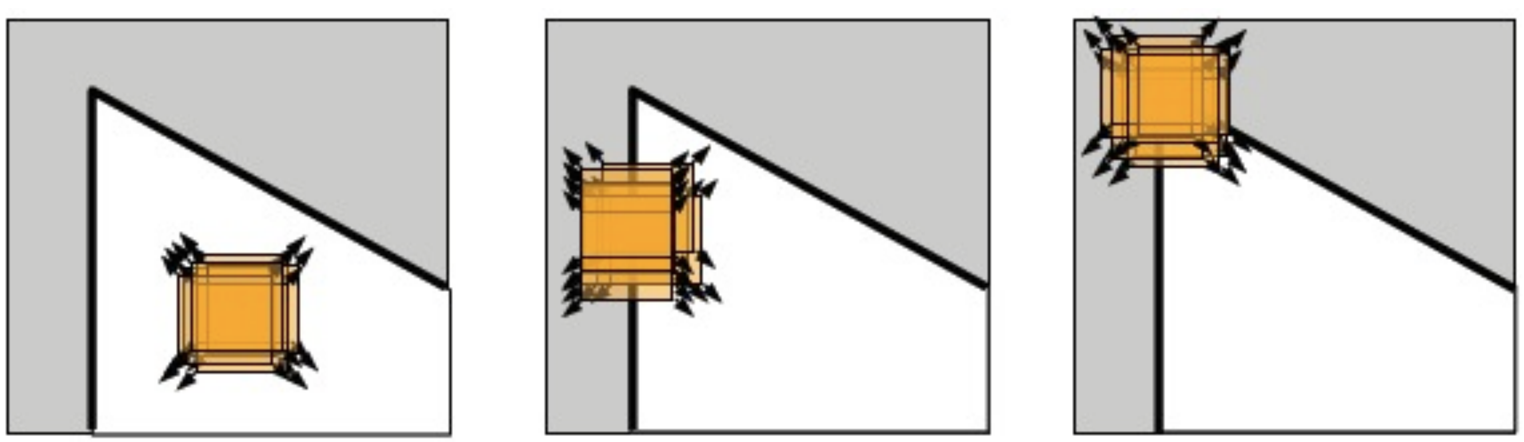
  - orientation are quantified such as [0°, 45°, 90°, 135°]
  - non-maximum suppression (gradient magnitude is maximum in its direction)
  - grouping maxima beginning by the highest

Corner detection

- aim: find stable points
  - scale change
  - orientation
  - projection, ...

Corner detection

- big difference with surrounding pixels



- weighted ''sum of squared differences'' (SSD)

$$S(x,y) = \sum_u \sum_v w(u,v)\ \left(I(u,v) - I(u+x,v+y)\right)^2$$

Harris corners

- image I is approximated by Taylor serie

$$I(u+x, v+y) \approx I(u,v) + I_x(u,v)x + I_y(u,v)y$$

- Ix and Iy are partial derivative in x and y

- S becomes

$$S(x,y) \approx \sum_u \sum_v w(u,v)\ \left(I_x(u,v)x + I_y(u,v)y\right)^2$$

Harris corners

- matrix notation

$$S(x,y) \approx \begin{pmatrix} x & y \end{pmatrix} A \begin{pmatrix} x \\ y \end{pmatrix}$$

$$A = \sum_u \sum_v w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix}$$

- eigen value of A : L1 and L2
  - L1 = 0 , L2 = 0 : no corner
  - L1 = 0 L2 >>0 : border
  - L1 >>0 et L2 >>0 : corner

$$M_c = \lambda_1 \lambda_2 - \kappa \left(\lambda_1 + \lambda_2\right)^2 = \det(A) - \kappa \,\operatorname{trace}^2(A)$$

Harris corners

- to avoid explicit eigen value calculation

$$M_c = \lambda_1\lambda_2 - \kappa\,(\lambda_1 + \lambda_2)^2 = \det(A) - \kappa\ \mathrm{trace}^2(A)$$

- local maxima = corners

Harris corners

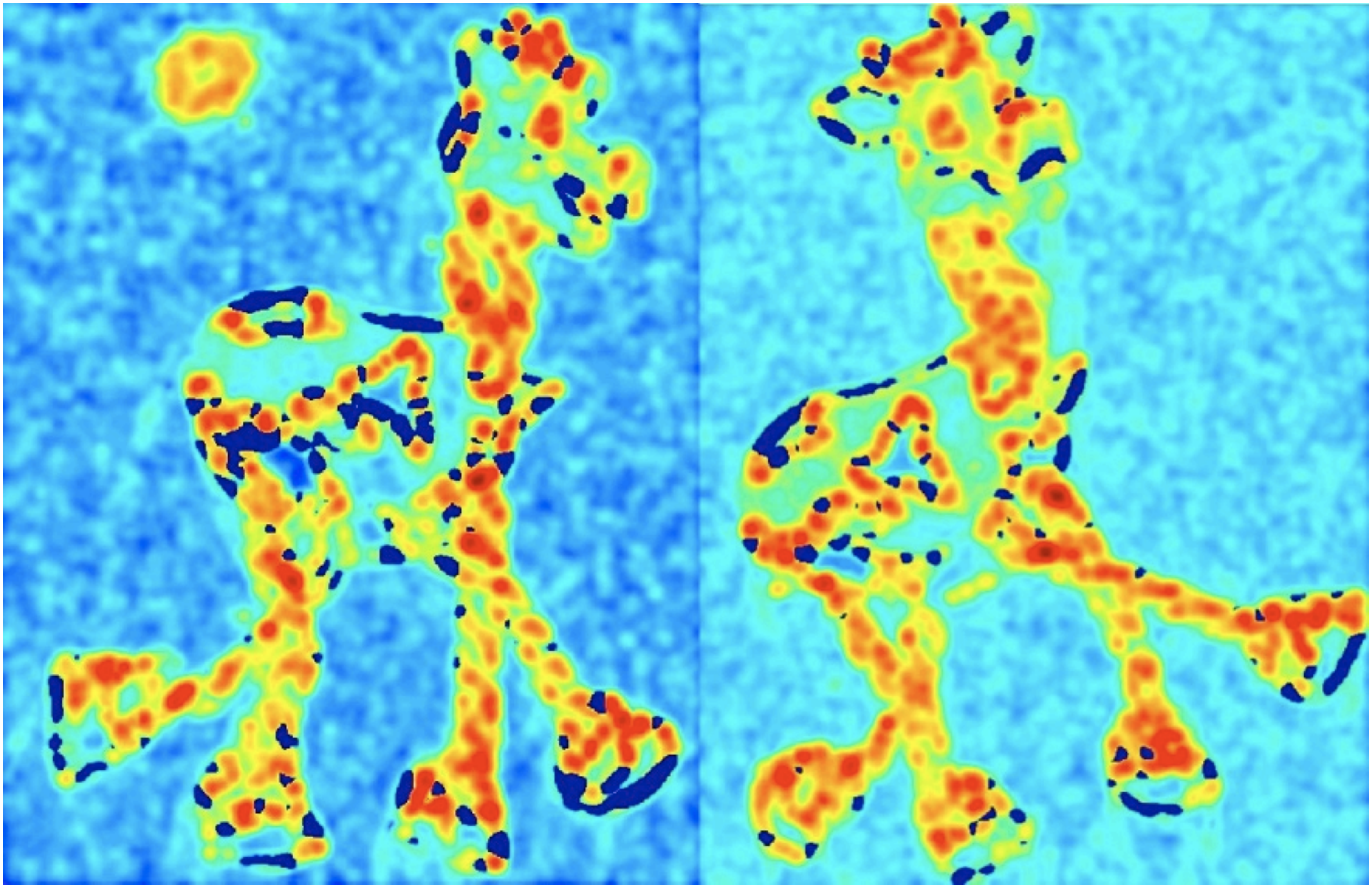
- properties :
  - rotation invariant
  - tolerant to level shift (derivative)
  - tolerant to scale (in level) maxima remains maxima

  - NO geometrical scale invariance :

Harris corners

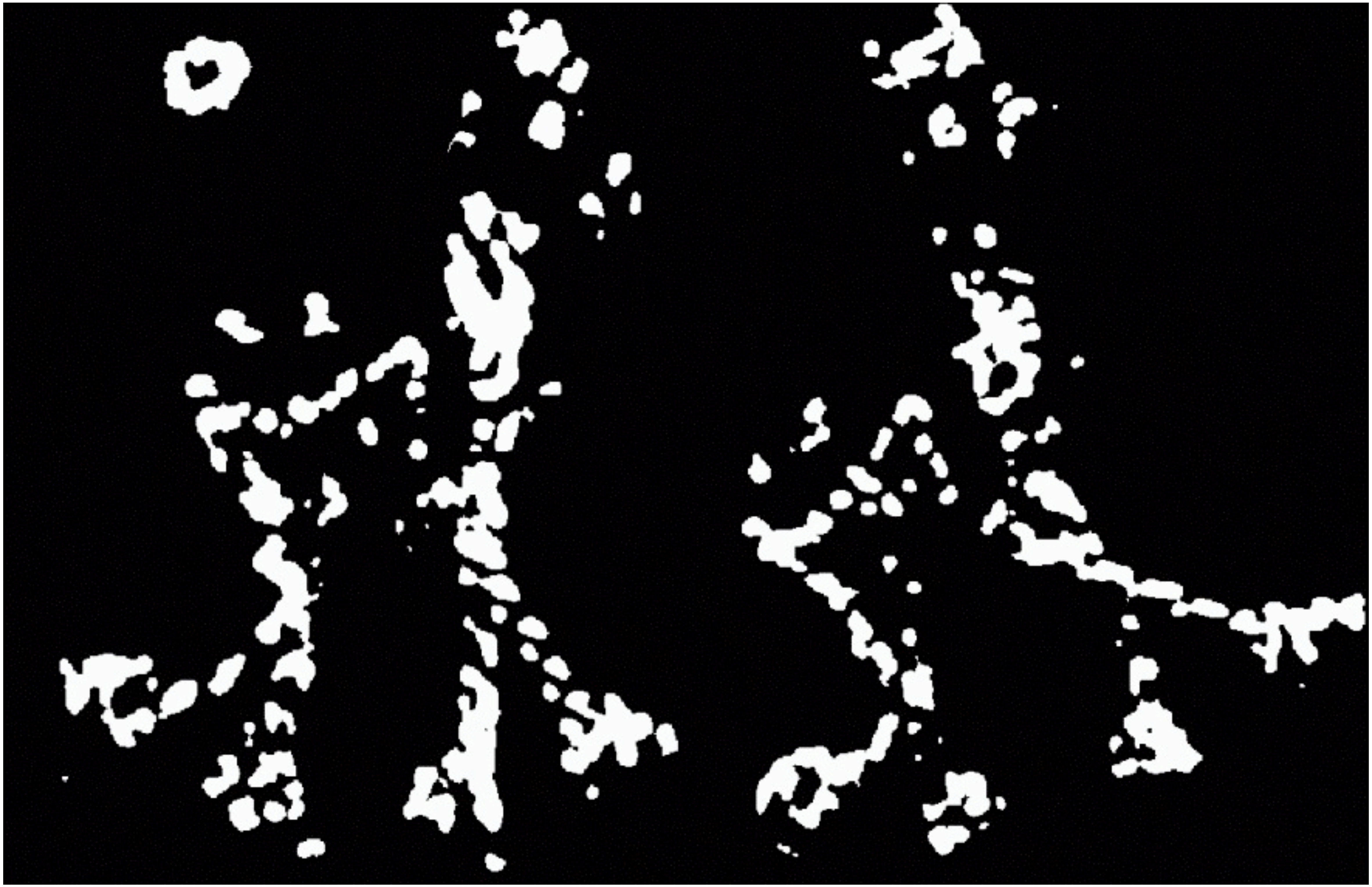
http://www.cs.cornell.edu/courses/cs664/2008sp/handouts/664%20corner%20detection.pdf

Harris corners



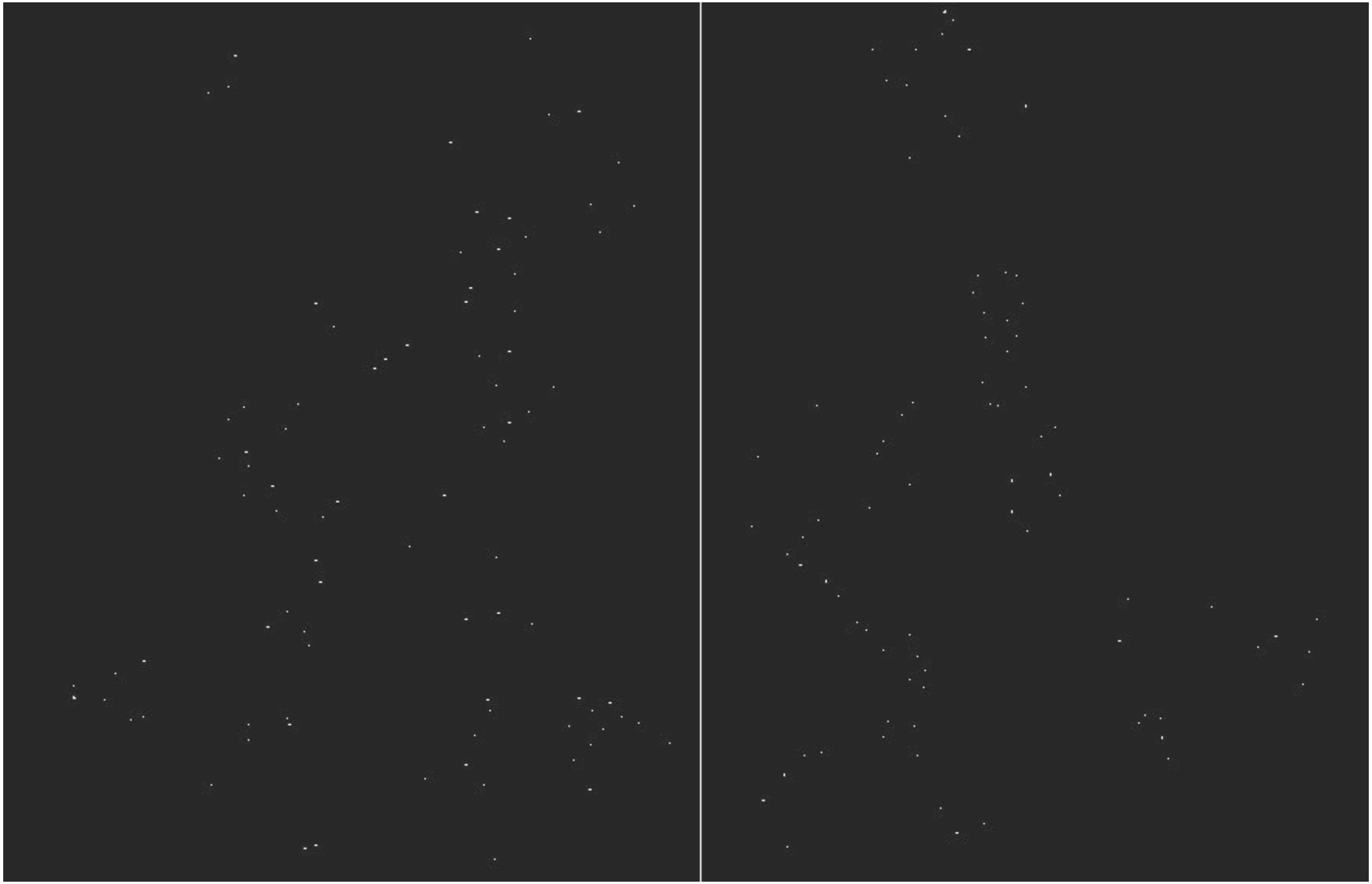
http://www.cs.cornell.edu/courses/cs664/2008sp/handouts/664%20corner%20detection.pdf

Harris corners



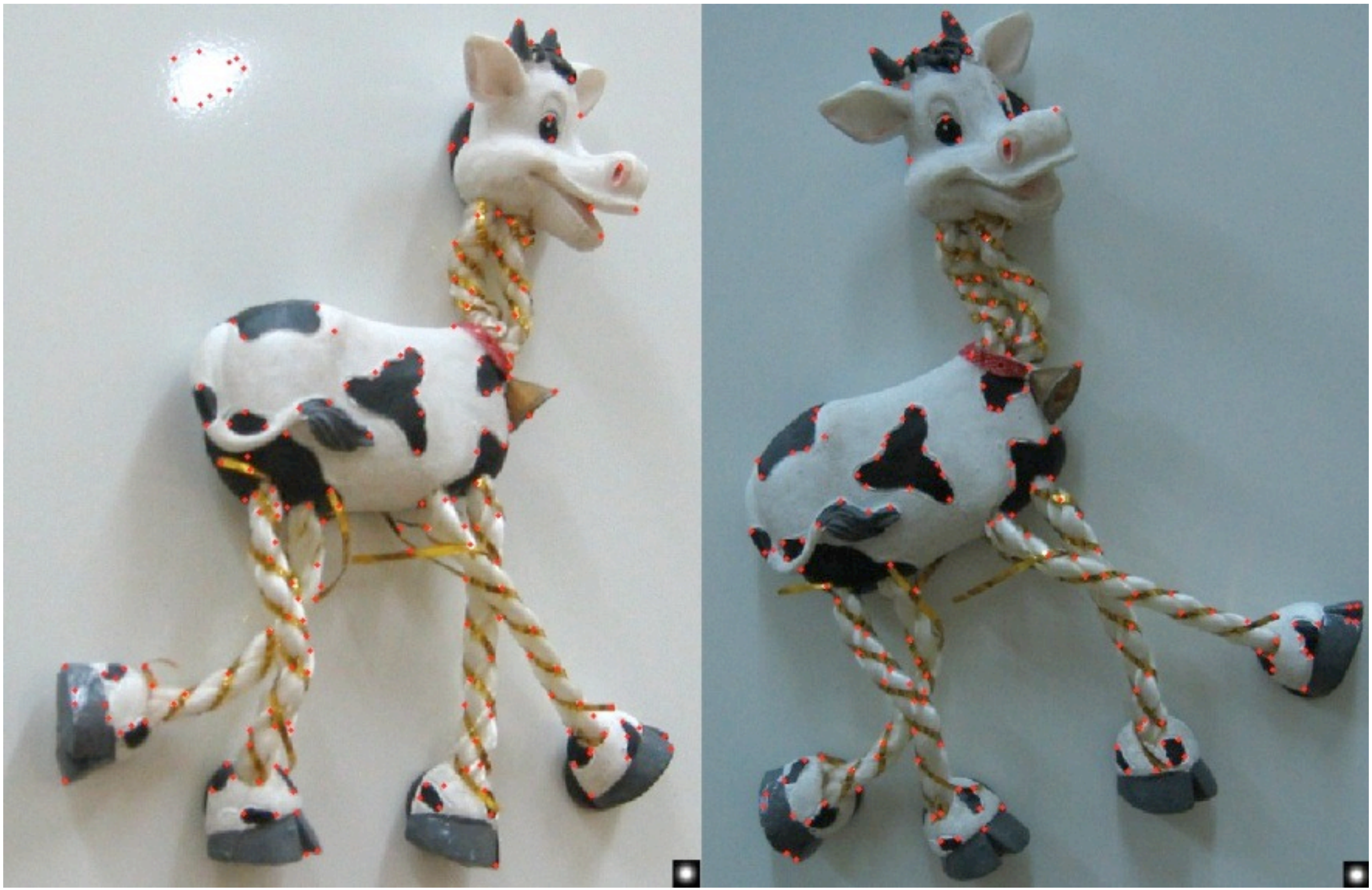
http://www.cs.cornell.edu/courses/cs664/2008sp/handouts/664%20corner%20detection.pdf

Harris corners



http://www.cs.cornell.edu/courses/cs664/2008sp/handouts/664%20corner%20detection.pdf

Harris corners



http://www.cs.cornell.edu/courses/cs664/2008sp/handouts/664%20corner%20detection.pdf

Kanade-Lucas-Tomasi (1994)

- Harris detector variant
- detected point are ranked by L1 (> threshold) decreasing
- close point are eliminate, process is iterated until no candidate point left

Scale invariance

- SIFT (Scale-invariant feature transform)
- SURF (Speeded Up Robust Features)

Scale invariance

- How to detect the scale ?
  - 2D gaussian
  - Scale-space Laplacian
  - Difference of Gaussian

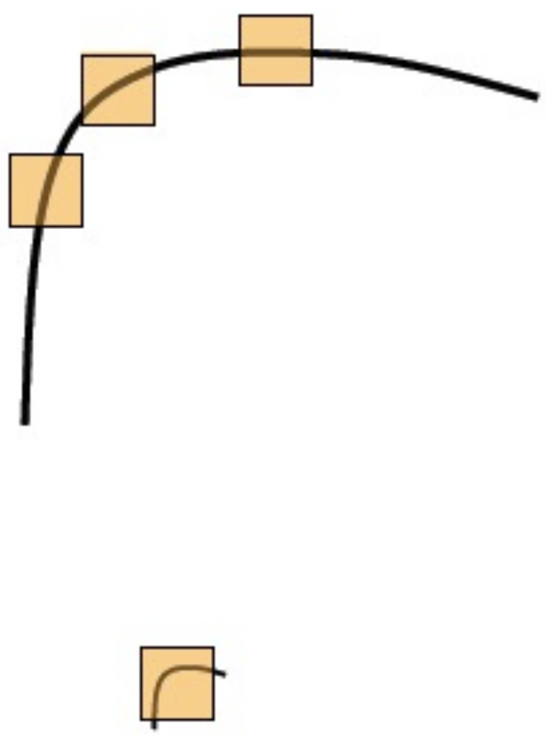
$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

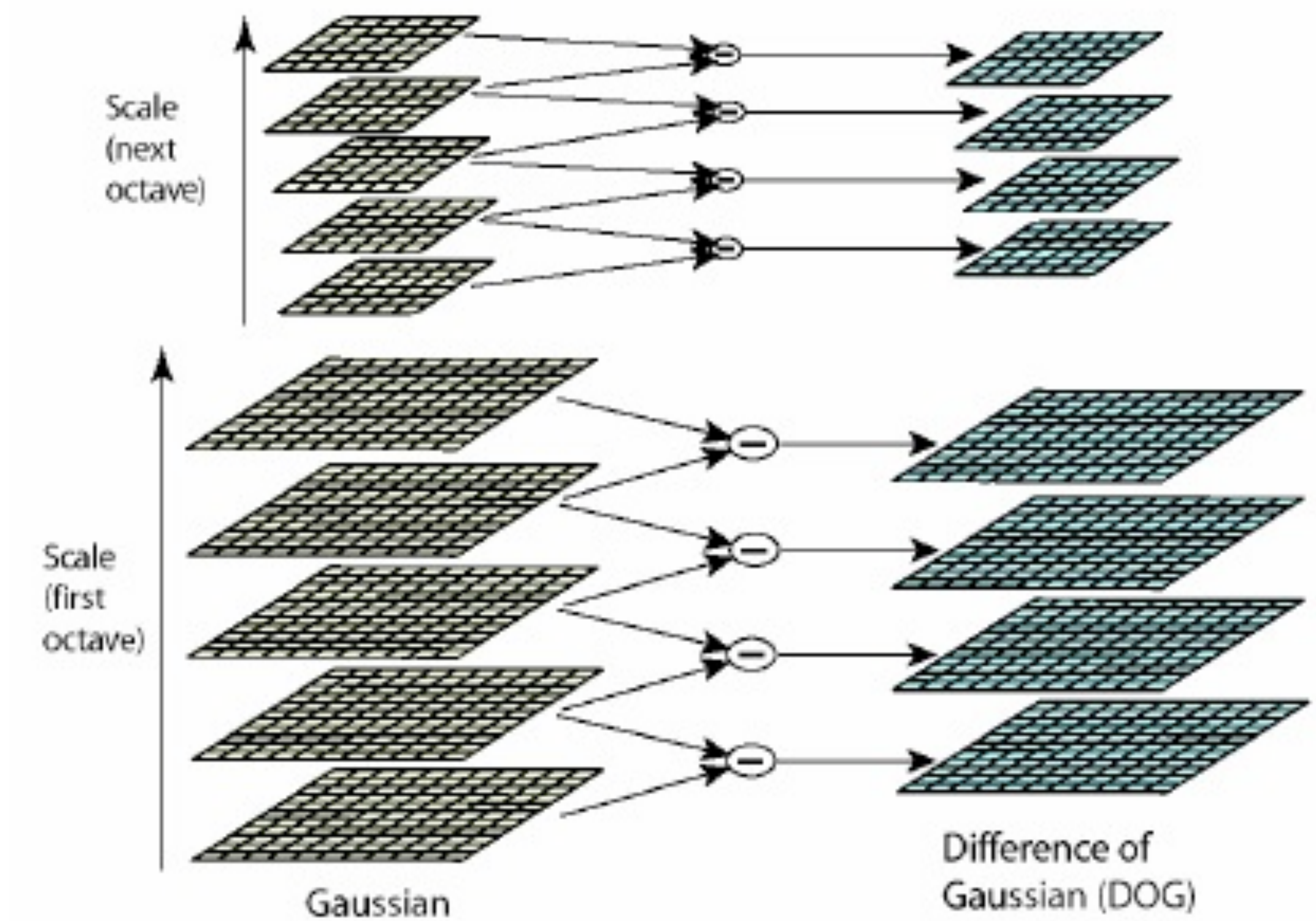
$$L = \sigma^2 (G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma))$$

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

Scale invariance

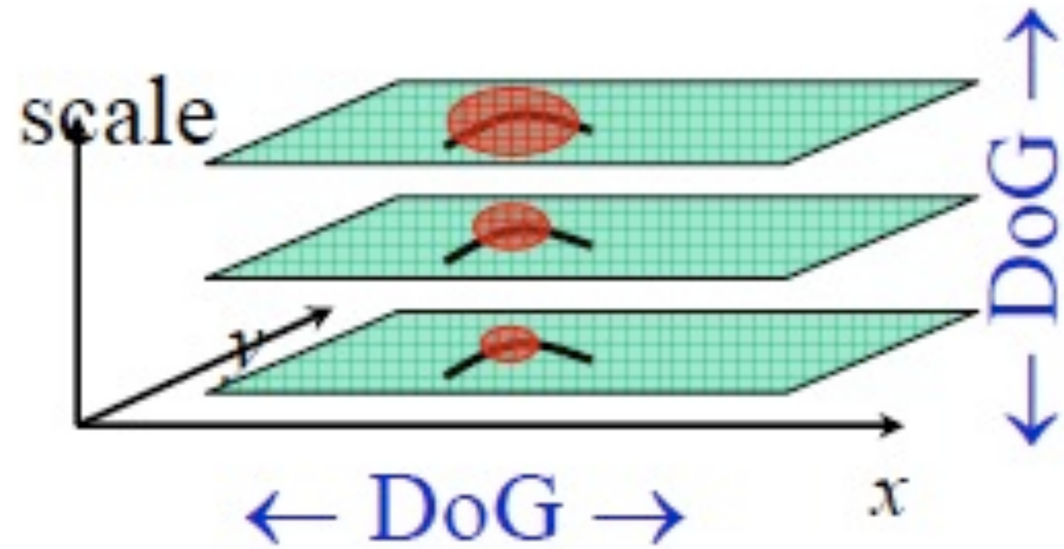
- Octave = sub-sampling
- intra octave : variable gaussian filtering sigma



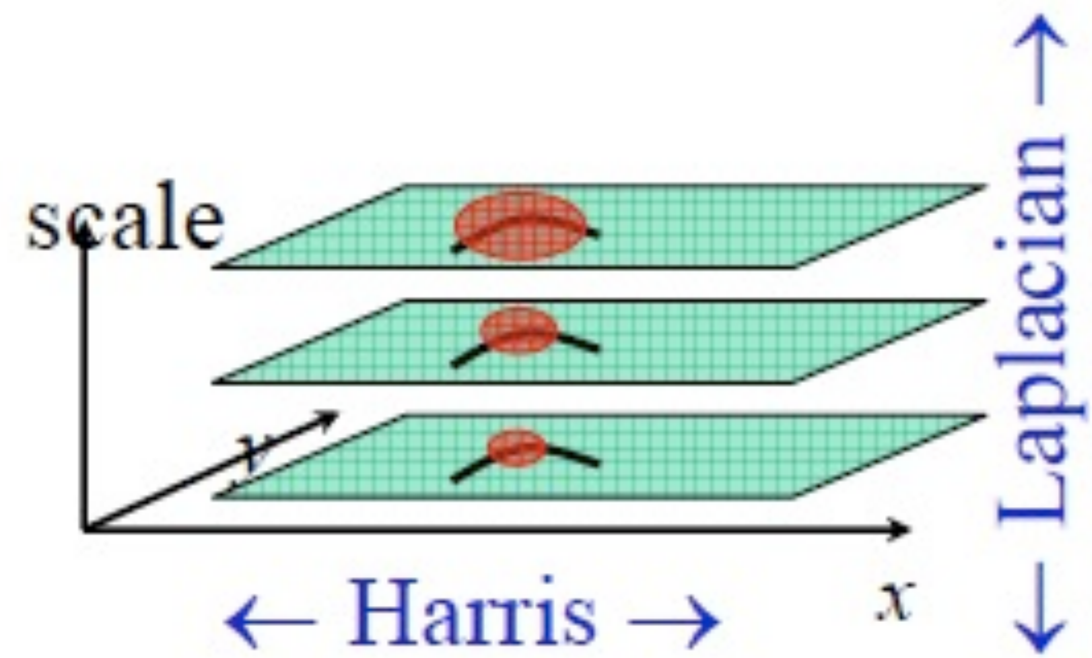


Scale invariance

- SIFT
  - local maximum of the DoG (spatial+scale)



- Harris Laplacian
  - maximum Harris (spatial dim.)
  - maximum DoG (scale dim.)



Sift

- Scale Invariant Feature Transform
- Lowe, David G. (1999). "Object recognition from local scale-invariant features". Proceedings of the International Conference on Computer Vision. 2. pp. 1150–1157.
- interest point detection
  - maxima and minima of the DoG in the scale space
  - scale space obtained by filtering and sub-sampling
  - lower contrast points are eliminated (local maximum with 26 neib.)

Sift

- **interpolation**
  - Taylor development close to the interest points

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \qquad \mathbf{x} = (x, y, \sigma)$$

  - H is computed as

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

  - too many points ...

Sift

- **elimination** of borders (2 eigen values highly different), if r is eigen value ratio

if

$$\mathrm{R} = \mathrm{Tr}\left(\mathbf{H}\right)^2 / \,\mathrm{Det}\left(\mathbf{H}\right)$$

greater than

$$\left(r_{\mathrm{th}} + 1\right)^2 / r_{\mathrm{th}}$$

---> border is rejected

Sift

- **local point orientation:**
  - gradient and magnitude are extracted for each point at there scale

$$I(x, y, \sigma) = I(x, y) * G_\sigma(x, y)$$

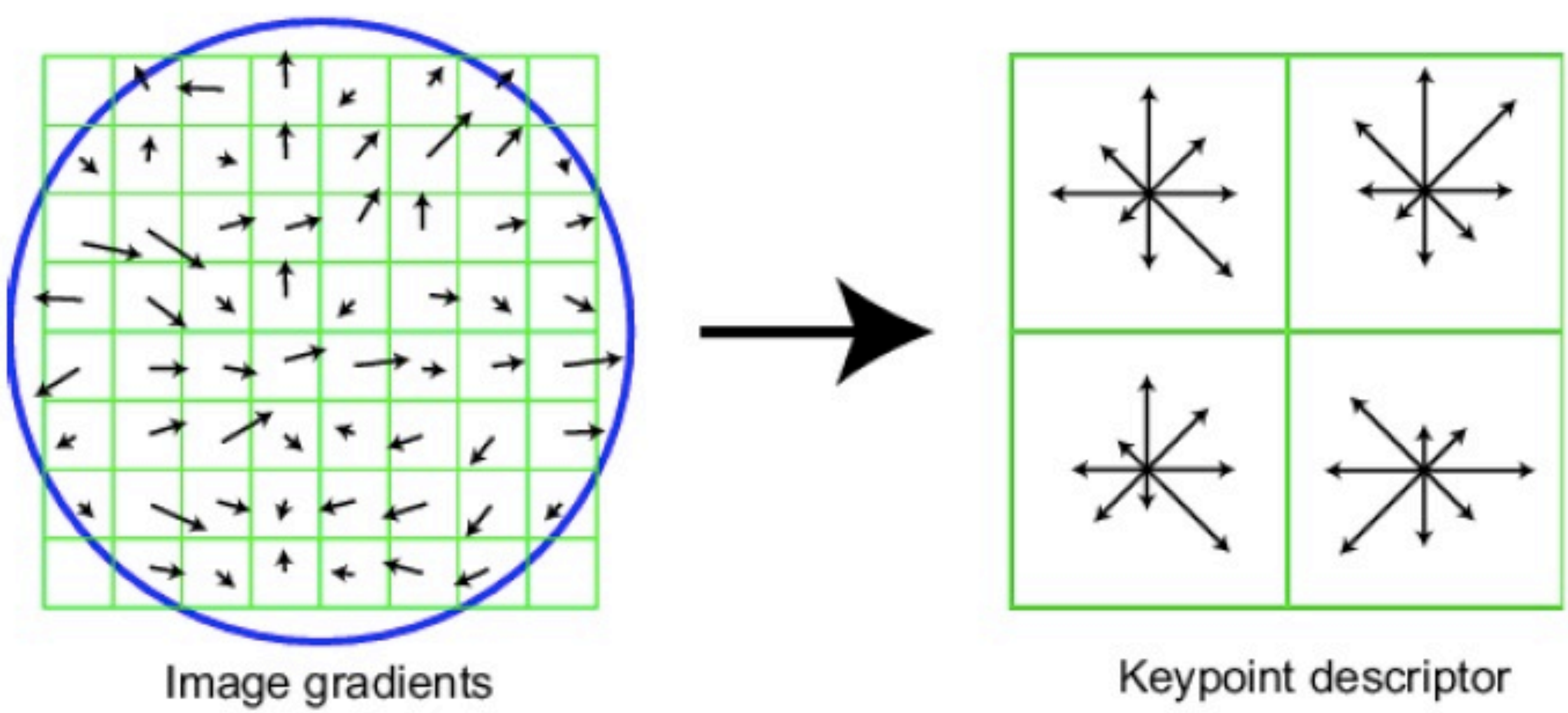
$$\begin{aligned} \mathbf{G} &= \sqrt{\mathbf{G}_x{}^2 + \mathbf{G}_y{}^2} \\ \mathbf{\Theta} &= \arctan\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right) \end{aligned}$$

  - direction histogram computed in the points neighborhood
  - main gradient direction is identified
  - invariance scale & rotation

Sift

- detected points are completed by descriptors based on a 16x16 region centered on the descritor

- a 4x4 orientation histogram matrix is build = 128 descriptors



Sift

- corresponding points by nearest neighbours

- applications
  - pose estimation
  - 3D
  - photo stitching
  - object tracking, ...

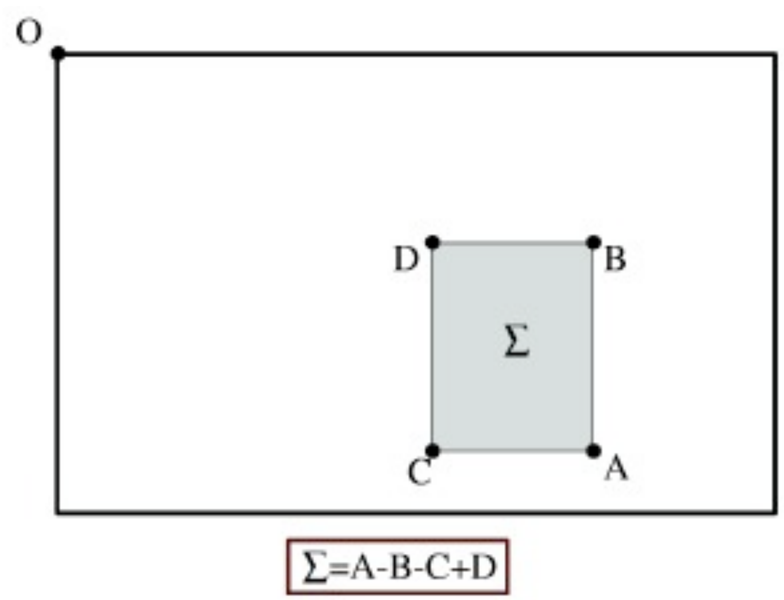
Surf

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

- speed up of the second derivative of the gaussian

- use of the integral image

$$I_{\Sigma}(x,y) = \sum_{x' \leq x, y' \leq y} I(x',y')$$

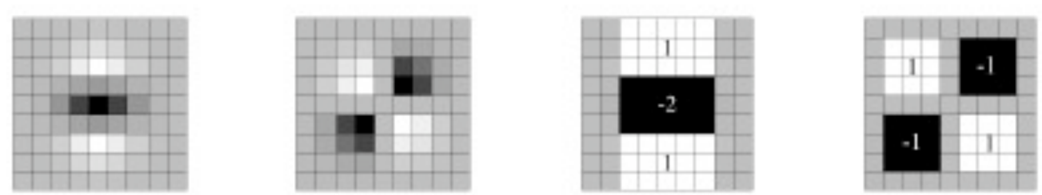


Surf

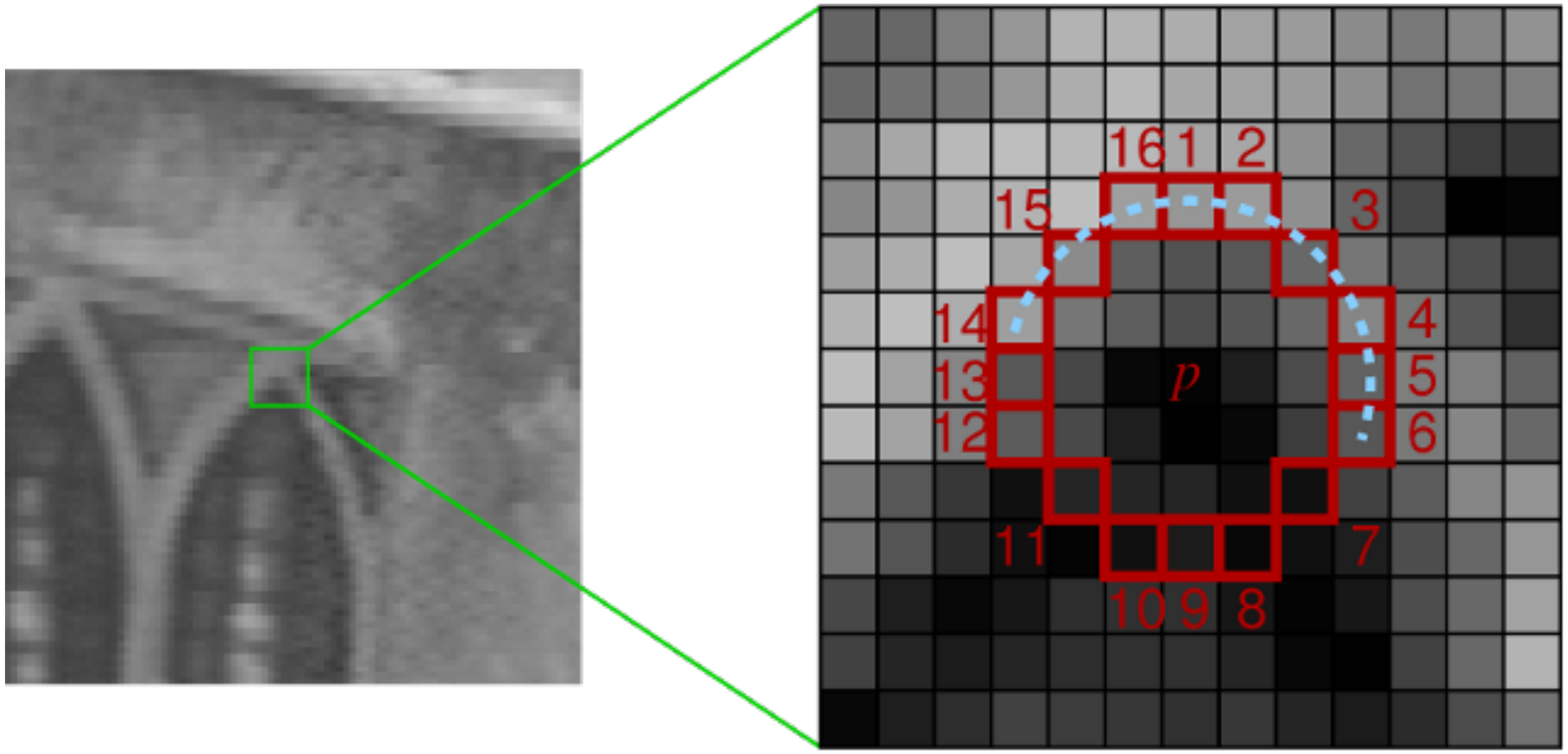
- Hessian matrix computation

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix}$$

$$L_{xx} = G_{xx}(x, y, \sigma)$$



Fast



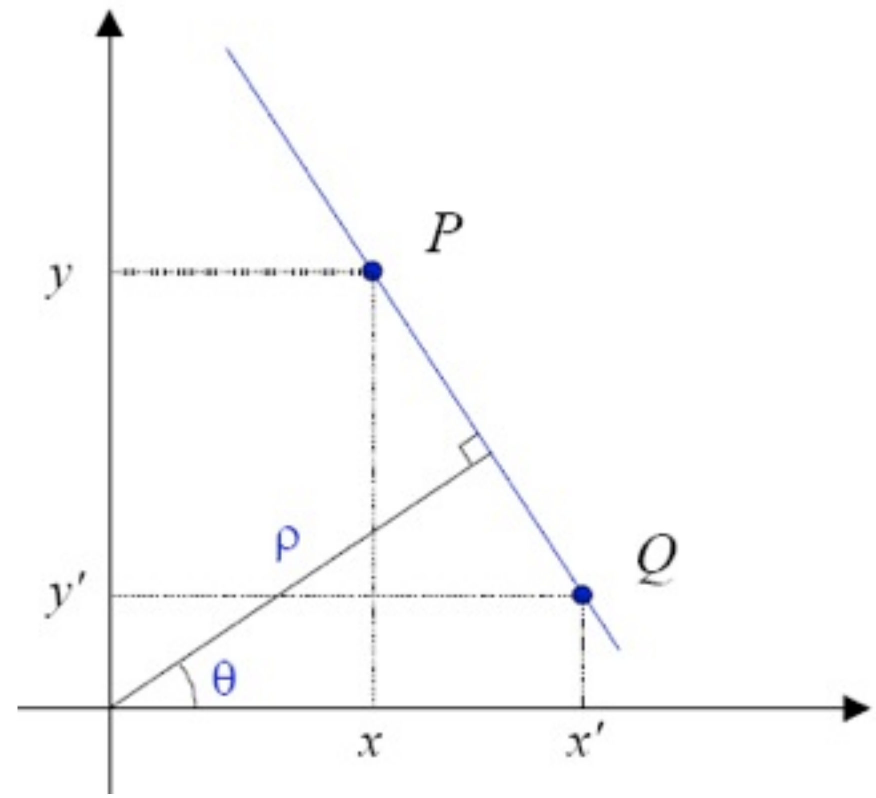
Rosten, E., and T. Drummond. 2005. "Fusing points and lines for high performance tracking."P. 1508–1515 in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1544896 (Accessed February 8, 2011).

Rosten, Edward, and Tom Drummond. 2006. "Machine learning for high-speed corner detection." *Computer Vision–ECCV 2006* 430–443. http://www.springerlink.com/index/y11g42n05q626127.pdf (Accessed February 8, 2011).

Hough transform

Hough transform

- parametric sets
  - line
  - circles, ...
- parameters space
  - discretized
  - counters



$$x\ cos(\theta) + y\ sin(\theta) = \rho$$

$$(x, y) \longleftrightarrow (\rho, \theta)$$

Hough transform

- **line detection**

- a point in the image space correspond to a cosinusoide

$$(x_0, y_0) \longleftrightarrow \rho = x_0 \, cos\theta + y_0 \, sin\theta$$

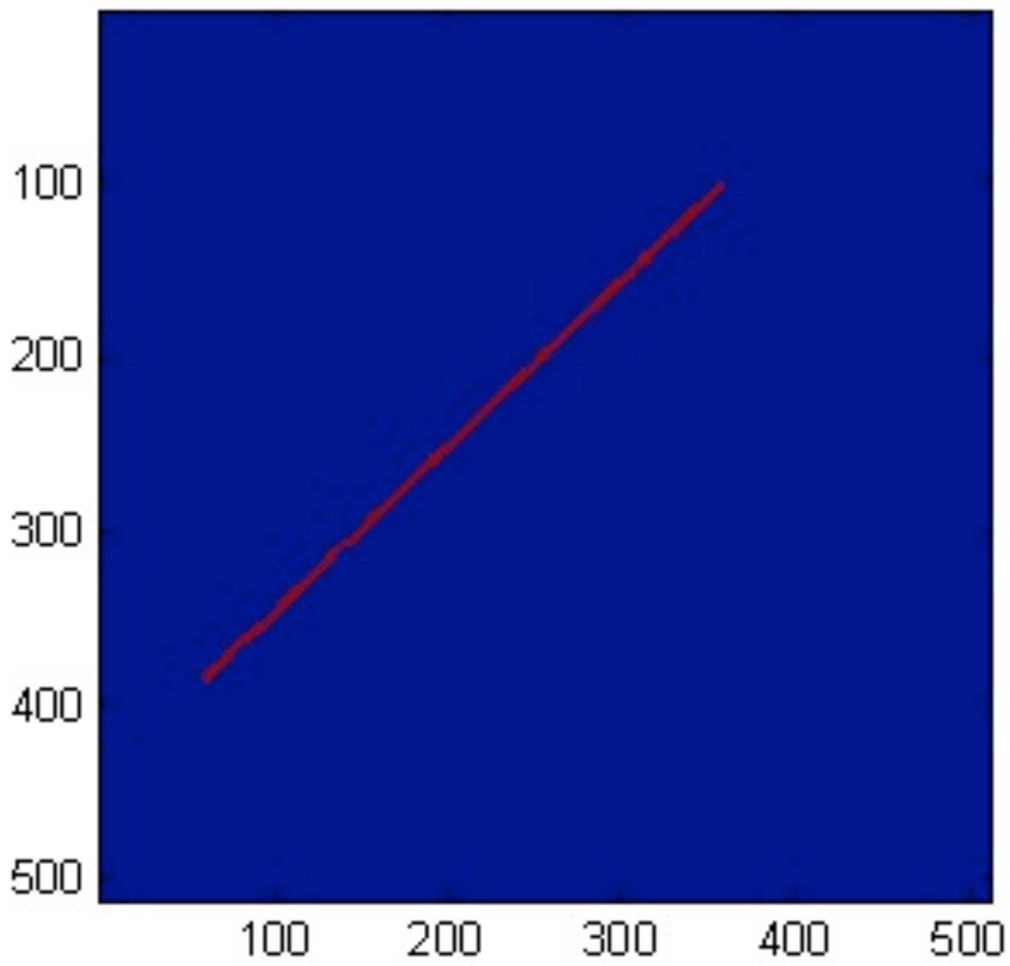
- a point in the parameters space correspond to one line in the image space

$$(\rho_0, \theta_0) \longleftrightarrow \rho_0 = x \, cos\theta_0 + y \, sin\theta_0$$

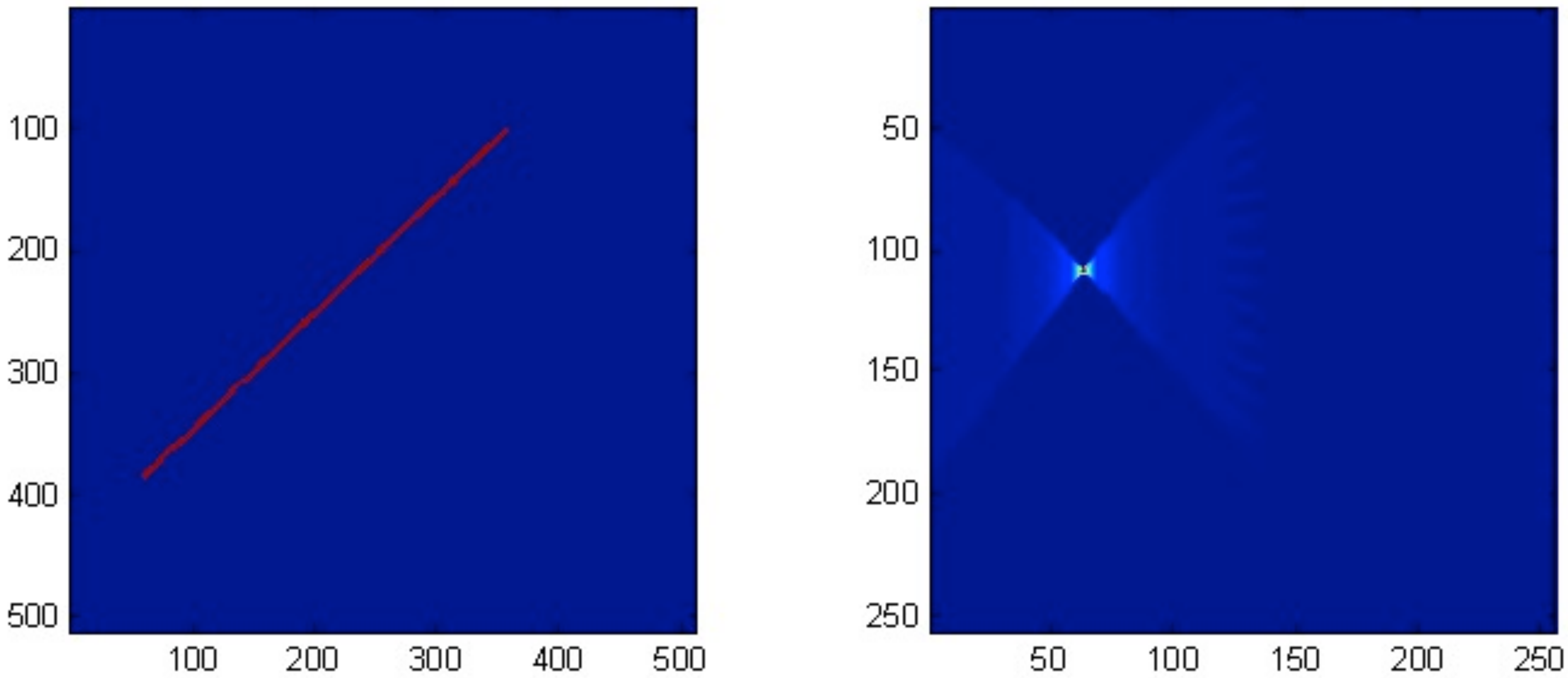
Hough transform

- parameters space discretized (rho,theta) = counters
- usually image is binarized
    - increment counter on the cosinusoide
    - higher counters = line segments !

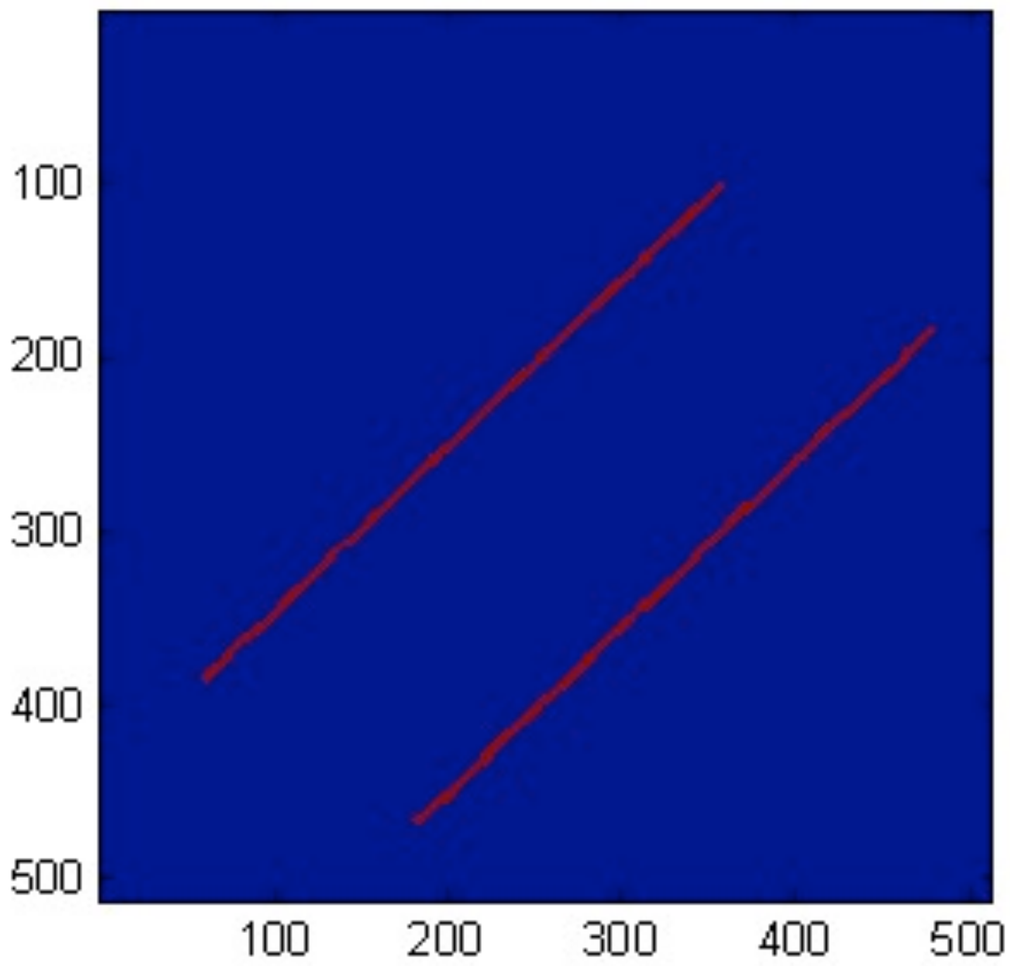
Hough transform



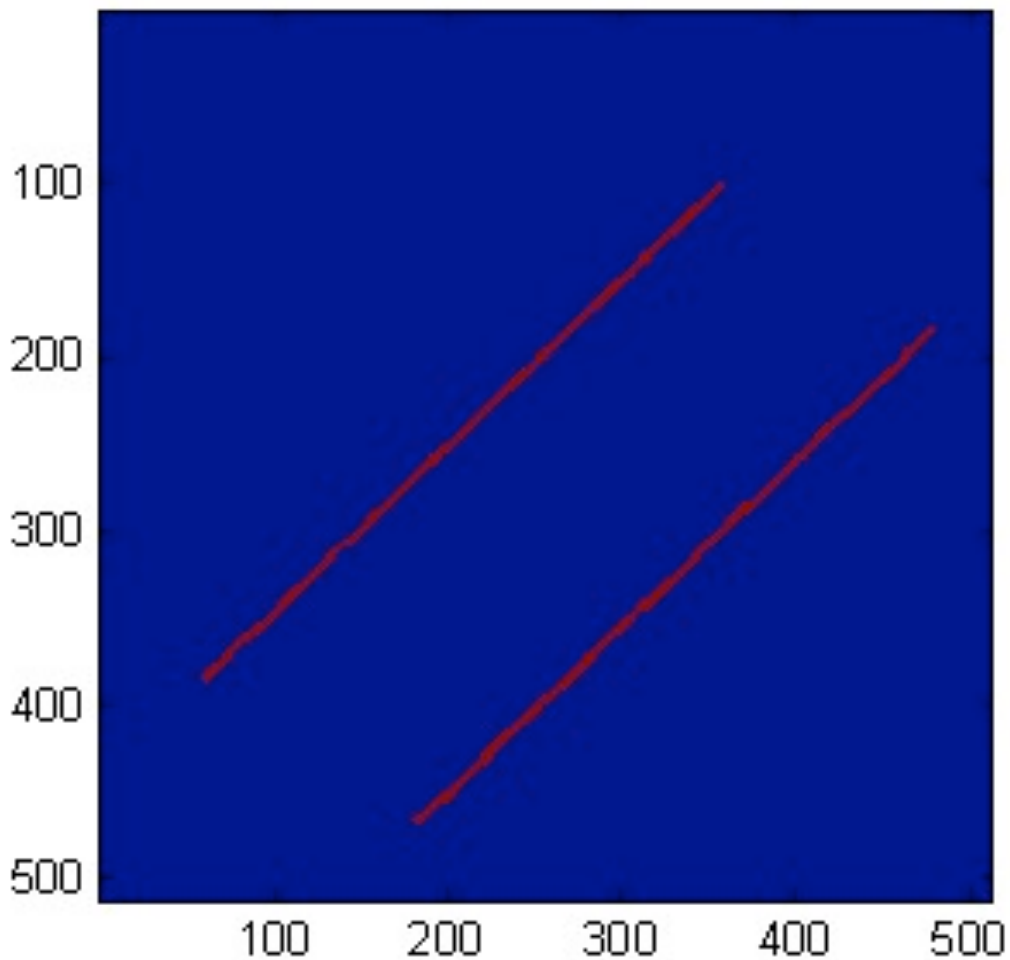
Hough transform

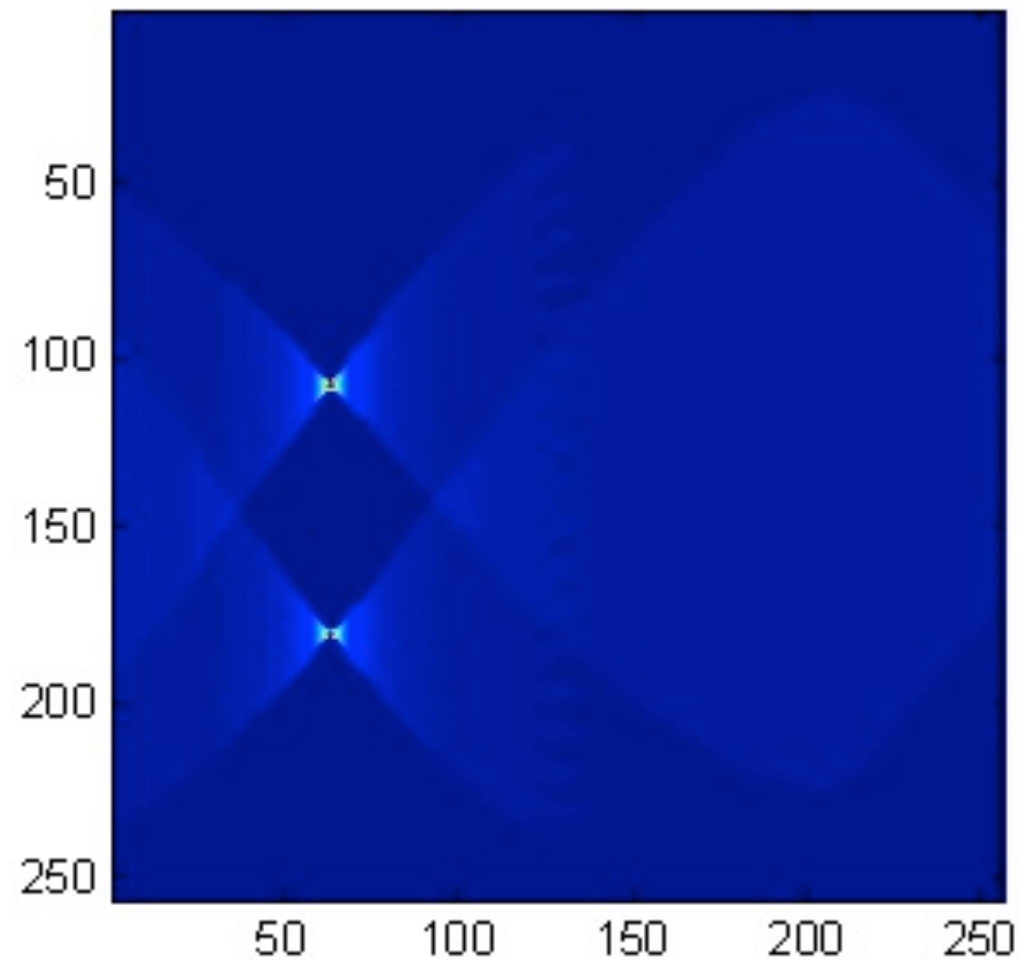


Hough transform

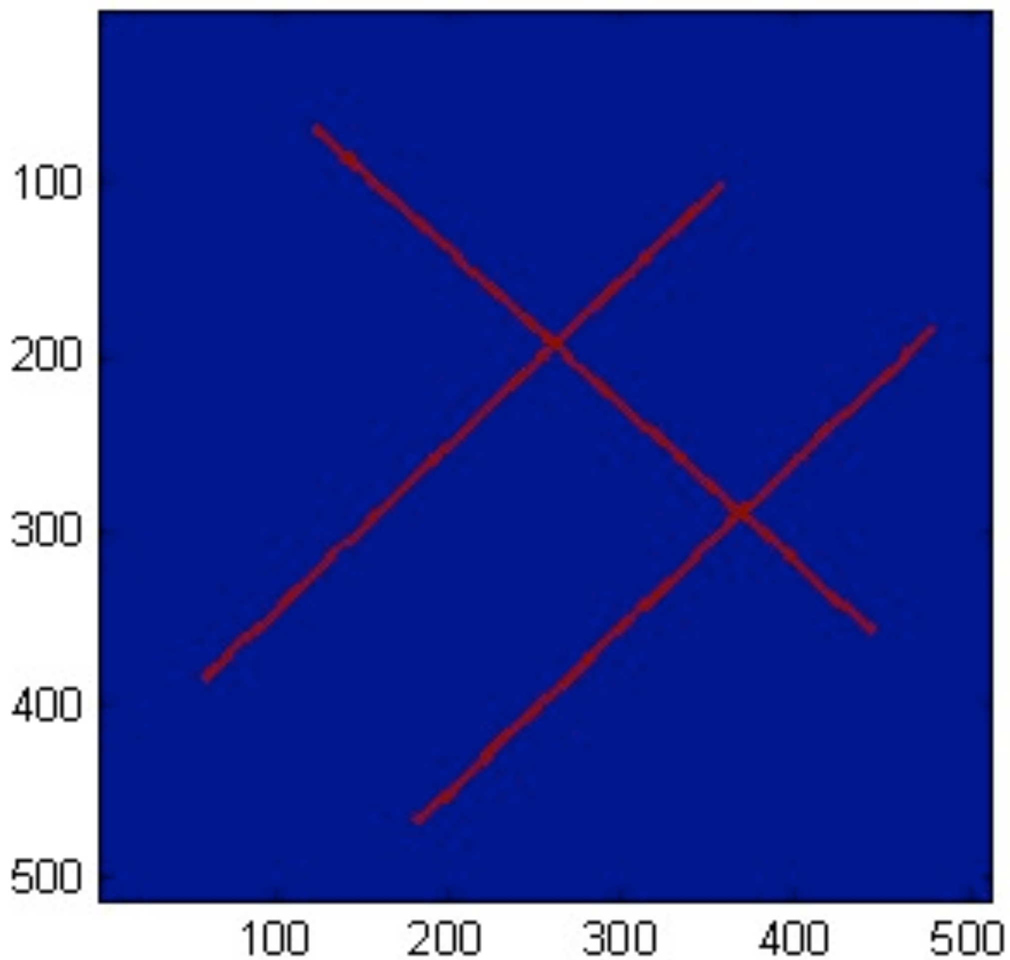


Hough transform

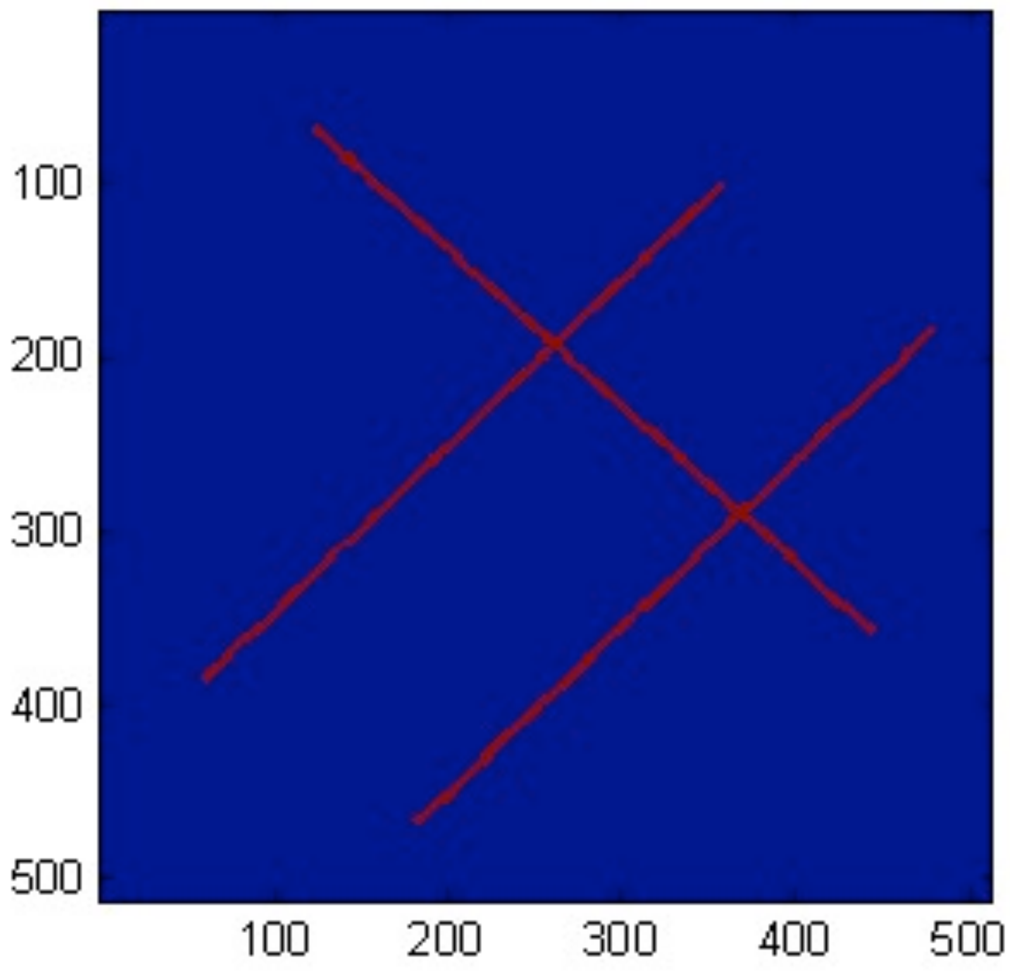


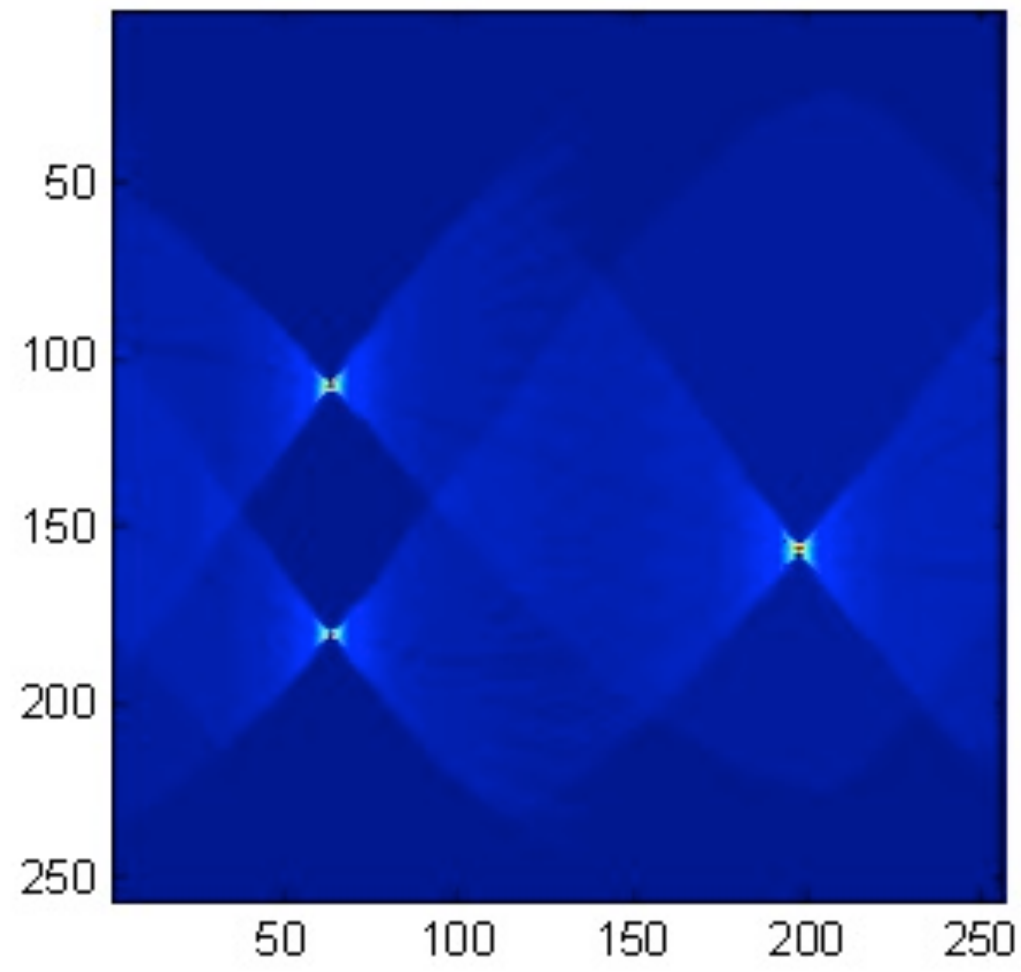


Hough transform

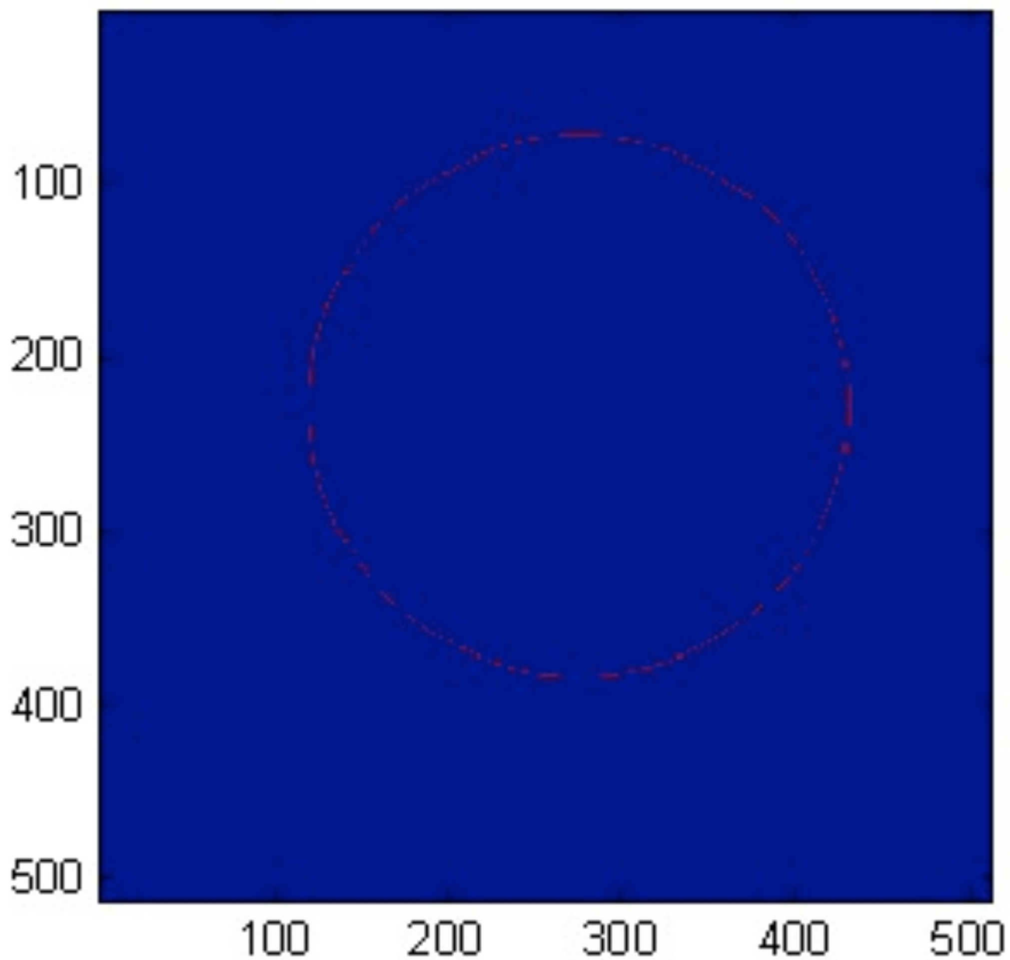


Hough transform

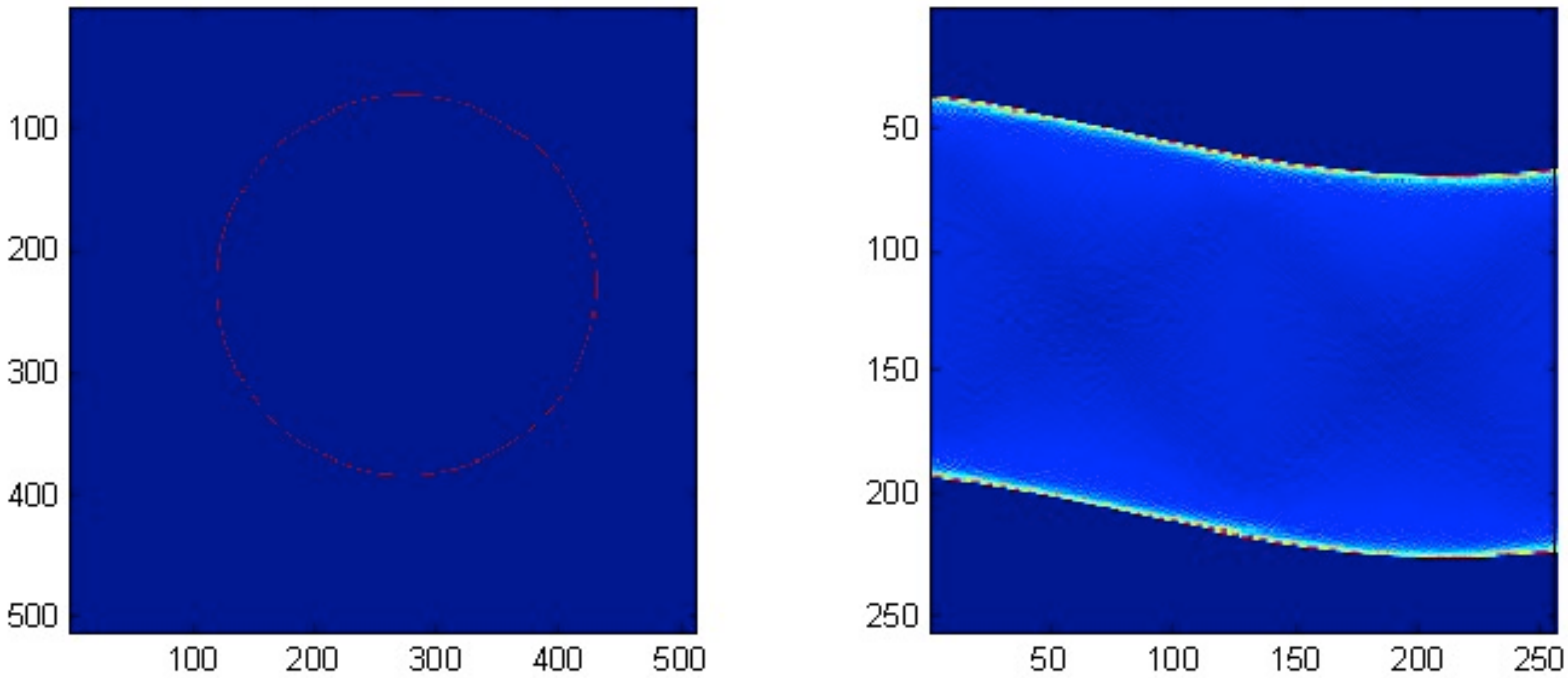




Hough transform



Hough transform

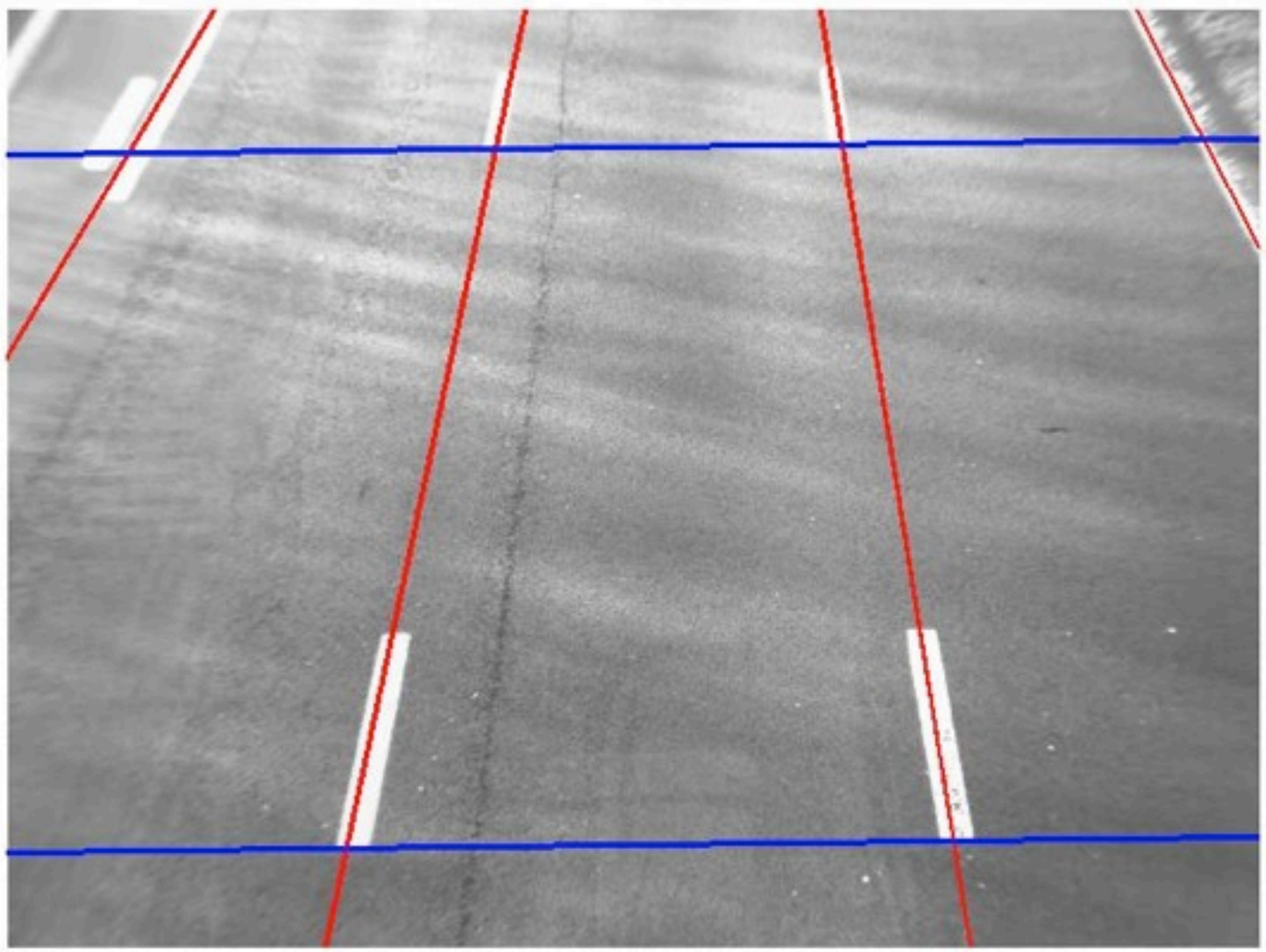


Hough transform

- computation optimization if gradient direction is known
- gradient is perp. to contours
- only counters close to perp. to gradient are incremented
- other transforms
  - circles $$(x - x_c)^2 + (y - y_c)^2 = R^2 \ , \ (x_c, y_x, R)$$
  - fixed radius circle
$$(x - x_c)^2 + (y - y_c)^2 = R_0^2 \ , \ (x_c, y_x)$$

Hough transform

- line detection



Juan-Carlos Tocino Dias - Macqe - 2008