Advanced Feature Extraction

李东晓

lidx@zju.edu.cn

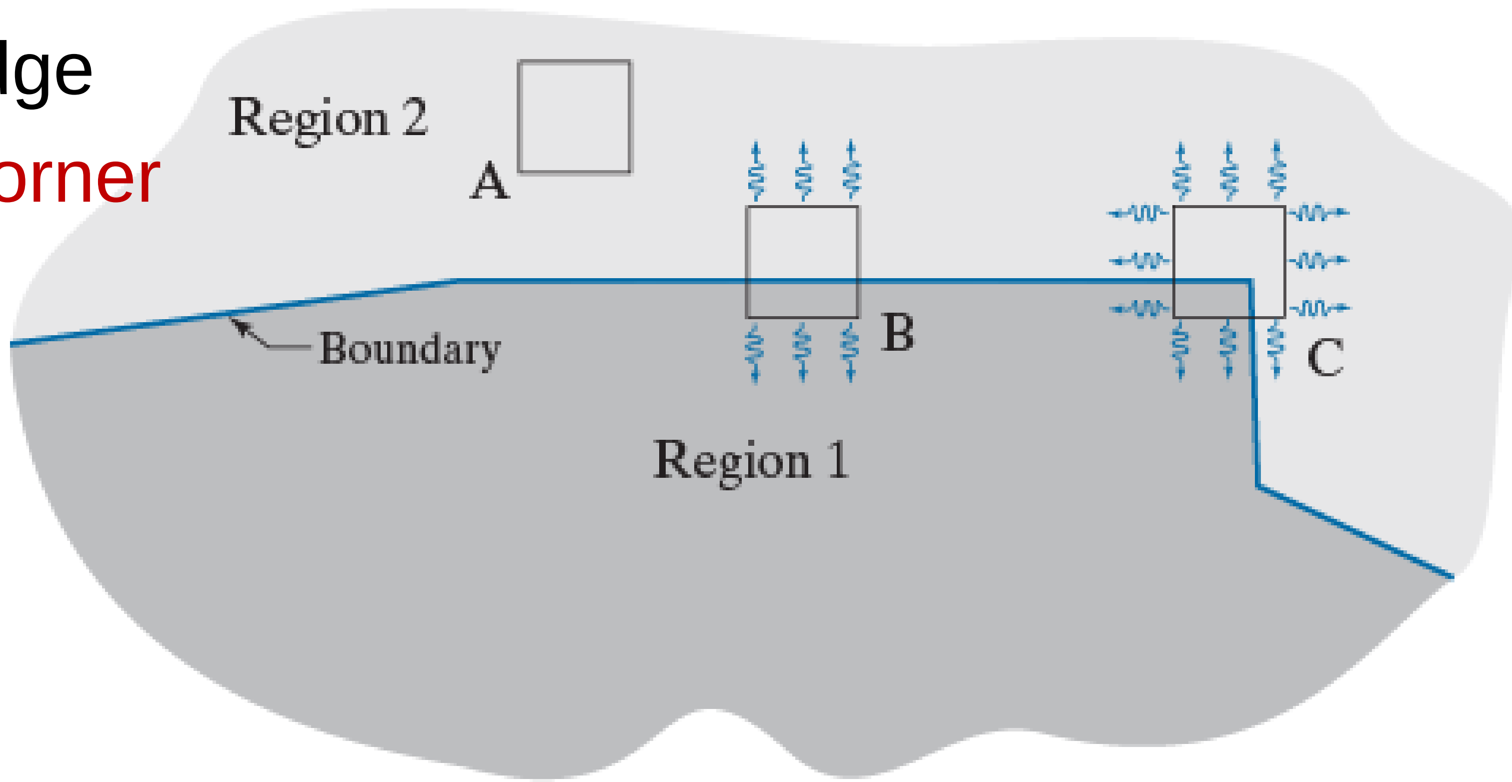
浙江大学信息与电子工程学院
College of Information Science & Electronic Engineering, Zhejiang University

Contents

- Advanced Feature Extraction
  - – 11.6 Whole-Image Features (Harris, MSERs)
  - – 11.7 Scale-Invariant Feature Transform (SIFT)

Harris-Stephens Corner Detector

- Corner: a rapid change of direction in a curve
- 3 types of subregions
  - A: flat
  - B: edge
  - C: Corner



Harris-Stephens Corner Detector

- Shifted patch approx. by linear terms of a Taylor expansion

$$f(s+x,t+y) \approx f(s,t) + xf_x(s,t) + yf_y(s,t)$$

- Weighted sum of squared differences between two patches

$$\begin{aligned} C(x,y) &= \sum_s \sum_t w(s,t)\left[f(s+x,t+y) - f(s,t)\right]^2 \\ &= \sum_s \sum_t w(s,t)\left[xf_x(s,t) + yf_y(s,t)\right]^2 = [x \;\; y]\,\mathbf{M}\begin{bmatrix} x \\ y \end{bmatrix} \end{aligned}$$

- Harris Matrix

$$\mathbf{M} = \sum_s \sum_t w(s,t)\,\mathbf{A}$$

$$\mathbf{A} = \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix}$$ symmetric

Harris-Stephens Corner Detector

- Weighted sum of squared differences between two patches

$$C(x,y)=\sum_s\sum_t w(s,t)\left[f(s+x,t+y)-f(s,t)\right]^2=\begin{bmatrix}x & y\end{bmatrix}\mathbf{M}\begin{bmatrix}x\\ y\end{bmatrix}$$

$$\mathbf{M}=\sum_s\sum_t w(s,t)\,\mathbf{A} \qquad \mathbf{A}=\begin{bmatrix}f_x^2 & f_xf_y\\ f_xf_y & f_y^2\end{bmatrix}$$ symmetric

- Weighting function
  - Box function: 1 inside the patch, 0 elsewhere
  - Exponential function: for data smoothing

$$w(s,t)=e^{-(s^2+t^2)/2\sigma^2}$$

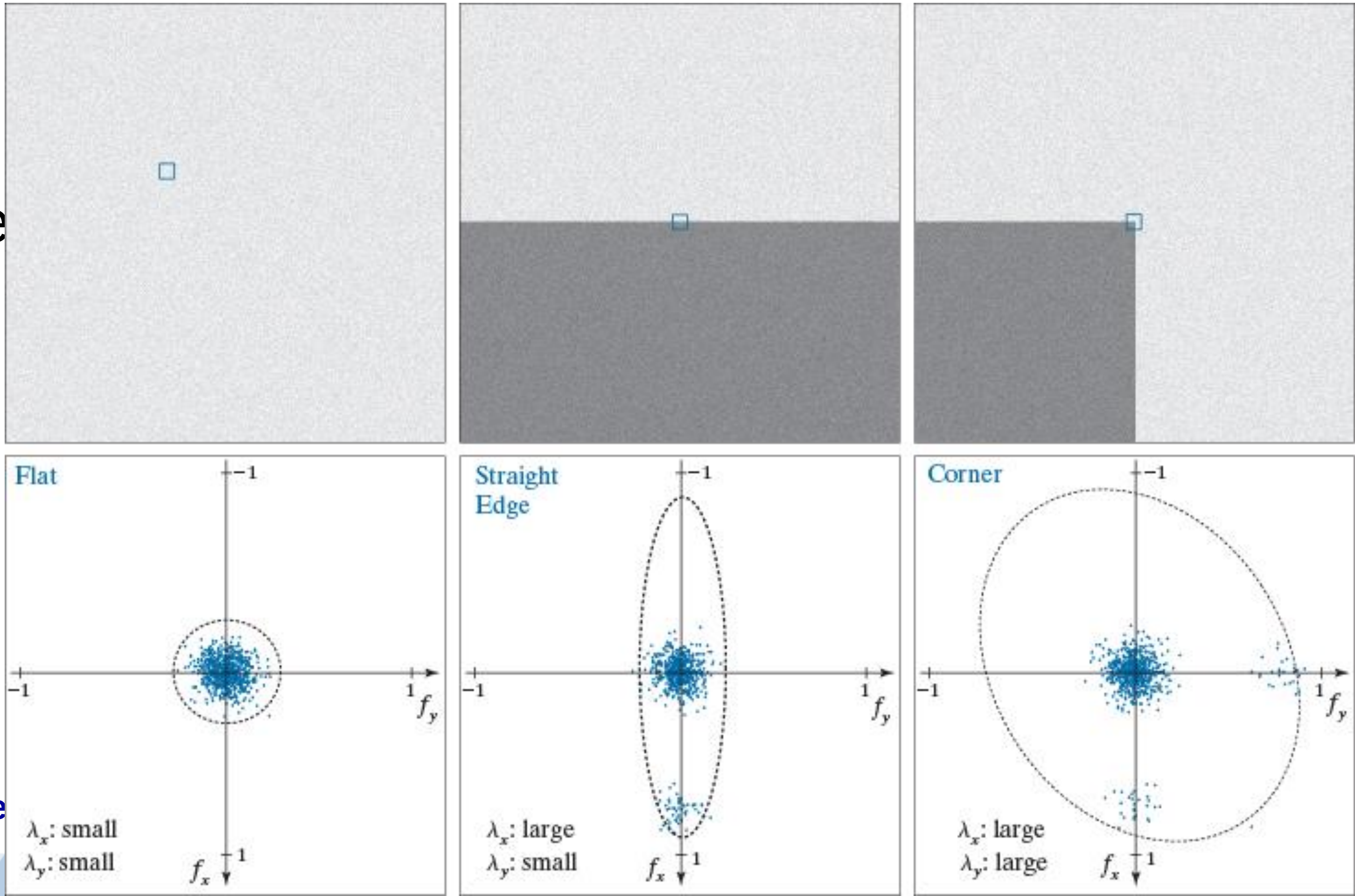
Harris-Stephens Corner Detector

$$w_x = w_y^T \quad w_y = [-1\ 0\ 1]$$

- How to differentiate between 3 cases?

Eigenvectors of a real, symmetric matrix point in the maximum data spread, (f_x, f_y)

Eigenvalues are proportional to the amount of data spread



Harris-Stephens Corner Detector

- Measure of corner response
  - Eigenvalues of $\mathbf{M} = \sum_s \sum_t w(s,t) \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix}$ computational expensive
  - HS detector

$$R = \lambda_x \lambda_y - k(\lambda_x + \lambda_y)^2$$

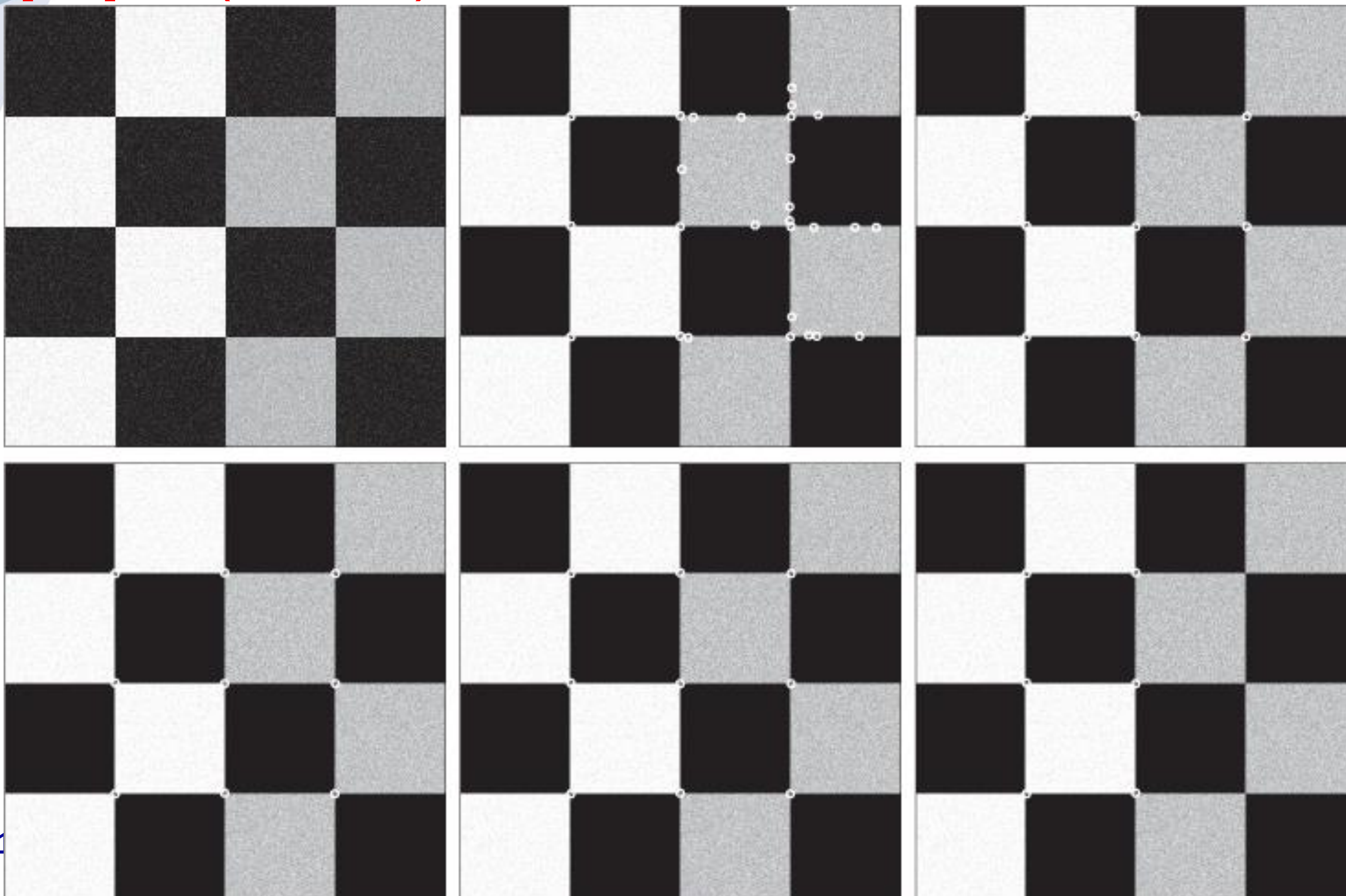
$$= \det(\mathbf{M}) - k\,\mathrm{trace}^2(\mathbf{M})$$

Sensitivity factor

- Corner: large positive R > T
- Edge : large negative R
- Flat: small absolute value R

Example 1

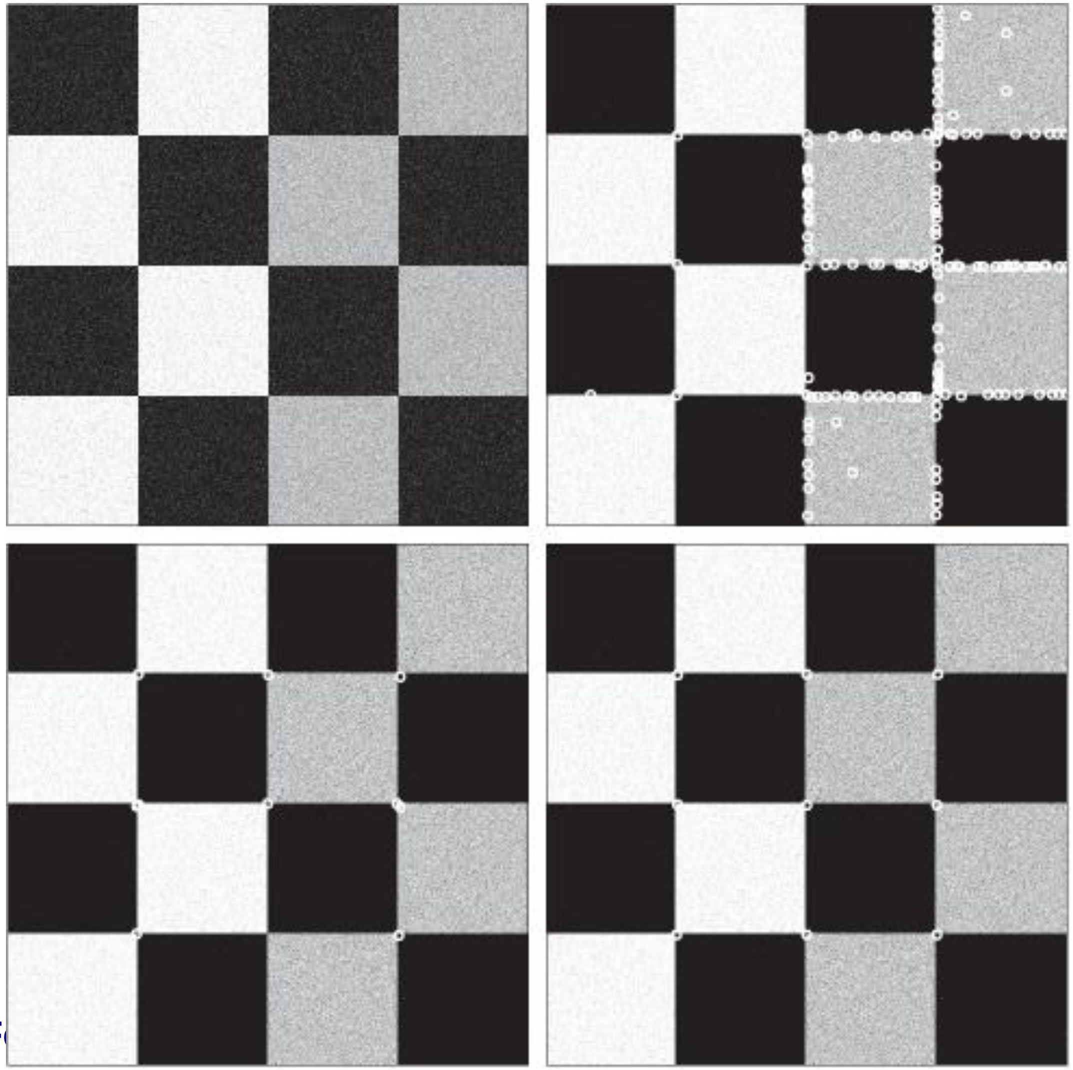
[0,1] + N(0, 0.006) k = 0.04, T = 0.01 k = 0. 1, T = 0.01



k = 0. 1, T = 0.1 k = 0. 04, T = 0.1 k = 0. 04, T = 0.3

Example 2

[0,1] + N(0, 0.01) k = 0.04, T = 0.01



k = 0. 249, T = 0.01 k = 0. 04, T = 0.15

Example 3

k = 0.04, T = 0.01 k = 0. 249, T = 0.01

k = 0. 17, T = 0.05 k = 0. 04, T = 0.05 k = 0. 04, T = 0.07

Example 4

k = 0. 04, T = 0.07

Maximally Stable Extremal Regions (MSERs)
最大稳定极值区域

- Extremal regions
- MSERs: Extremal regions that do not change size (number of pixels) appreciably over a range of threshold values

Detecting MSERs using Component Tree

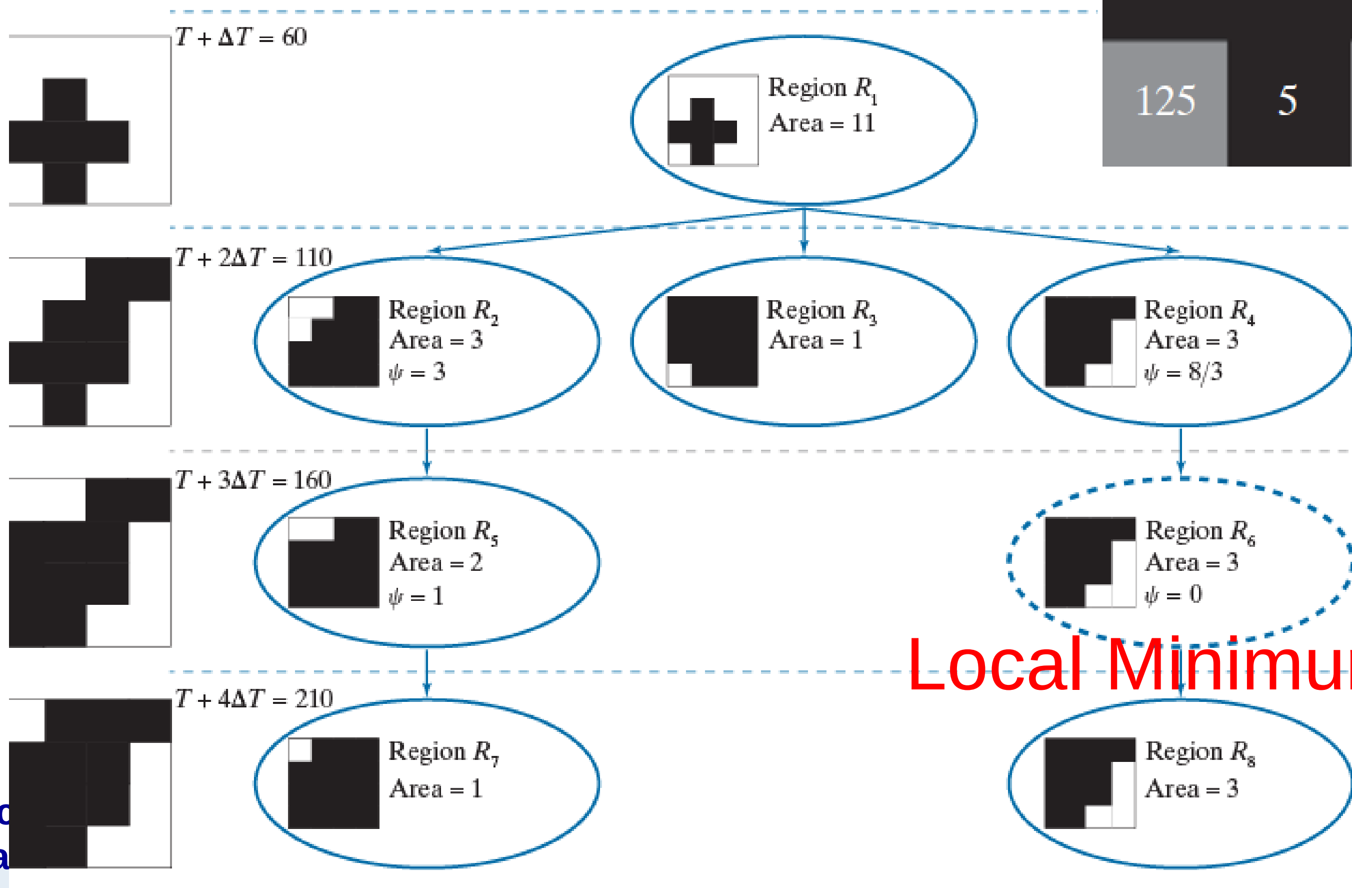
$$\forall p \in R \text{ and } \forall q \in \text{boundary}(R): I(p) > I(q)$$

225	175	90	90
125	5	90	225
5	5	5	225
125	5	225	225

Stability measure

$$\psi(R_j^{T+n\Delta T}) = \frac{\left|R_i^{T+(n-1)\Delta T}\right| - \left|R_k^{T+(n+1)\Delta T}\right|}{\left|R_j^{T+n\Delta T}\right|}$$

parent child

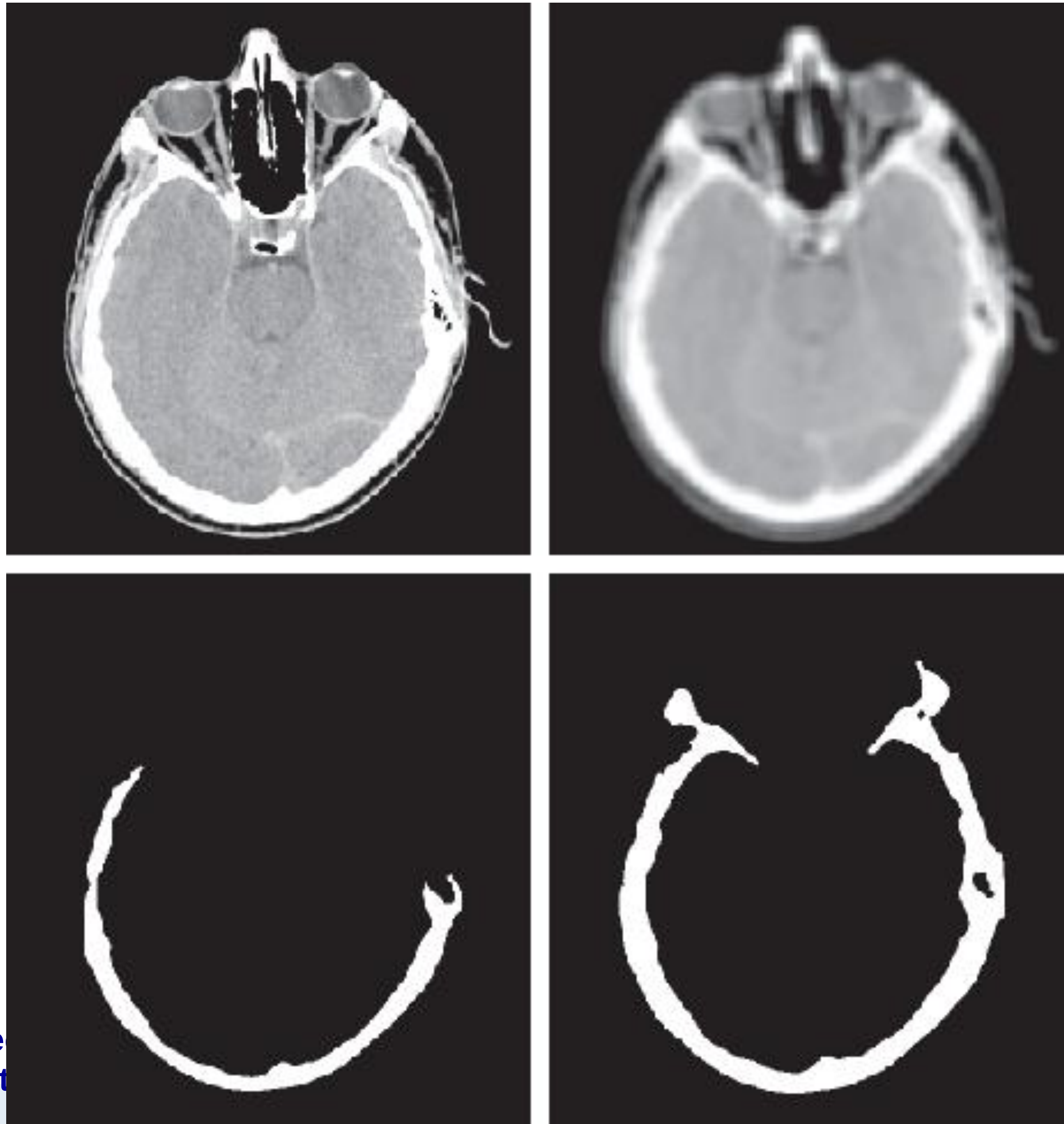


Local Minimum

Example 1

Brain CT

Smooth with 15x15 box filter



MSER

MSERs

Example 2

Building   Smooth with 5x5 box filter

MSERs using $T = 0, \Delta T = 10$

Example 3

Building rotated 5° Smooth with 5x5 box filter

MSERs

using $T = 0$, $\Delta T = 10$

Example 4

Building Half-sized Smooth with 3x3 box filter

MSERs

using $T = 0,\ \Delta T = 10$

Contents

- Advanced Feature Extraction
  - 11.6 Whole-Image Features (Harris, MSERs)

    Application : same scale, similar orientation, etc.
  - 11.7 Scale-Invariant Feature Transform (SIFT)

    Application : changes in scale, rotation, illumination, viewpoint, etc.

Scale-Invariant Feature Transform (SIFT)

Scale space

$$L(x,y,\sigma) = G(x,y,\sigma) \star f(x,y)$$

Generate a stack of smoothed images

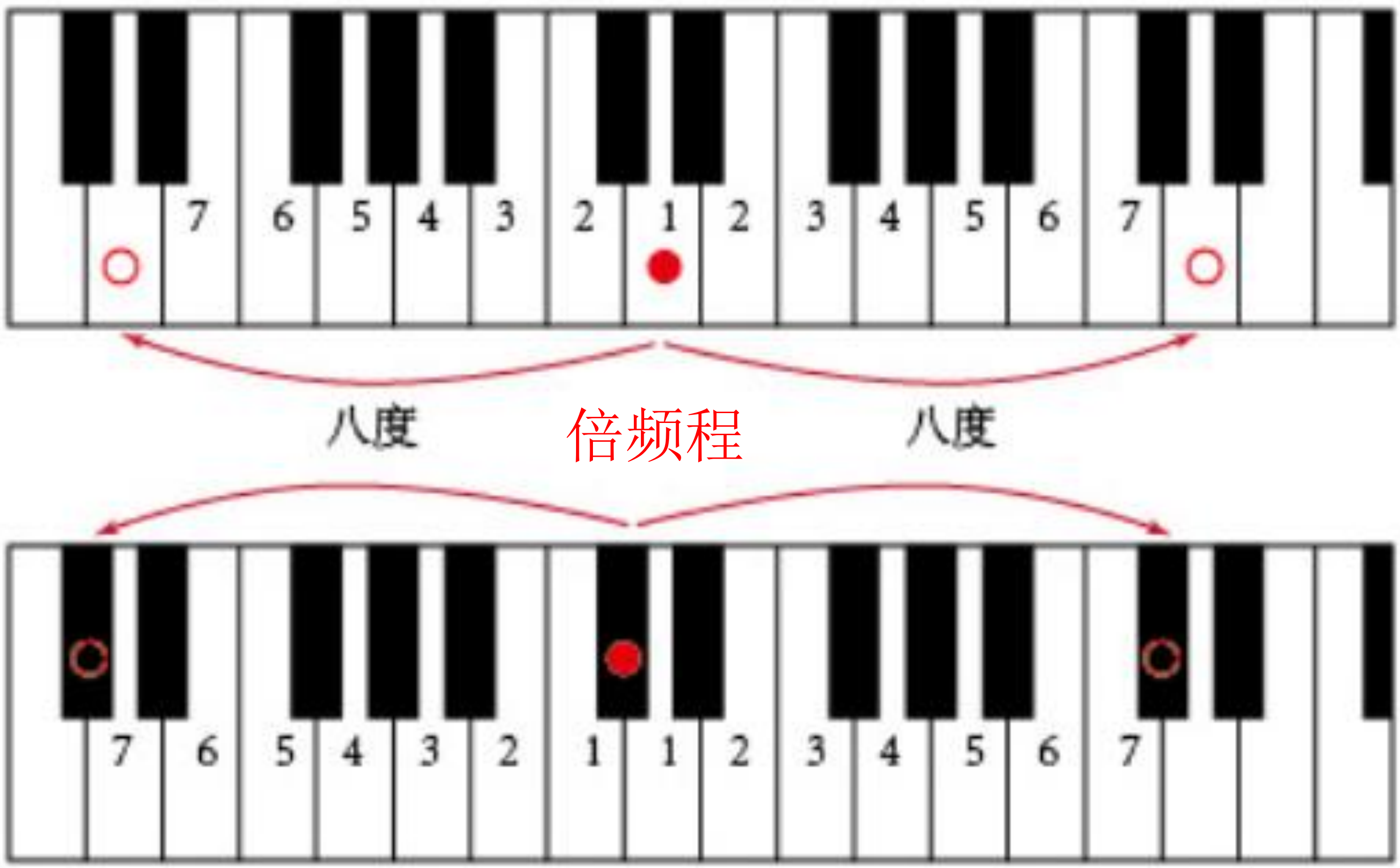
Gaussian kernel

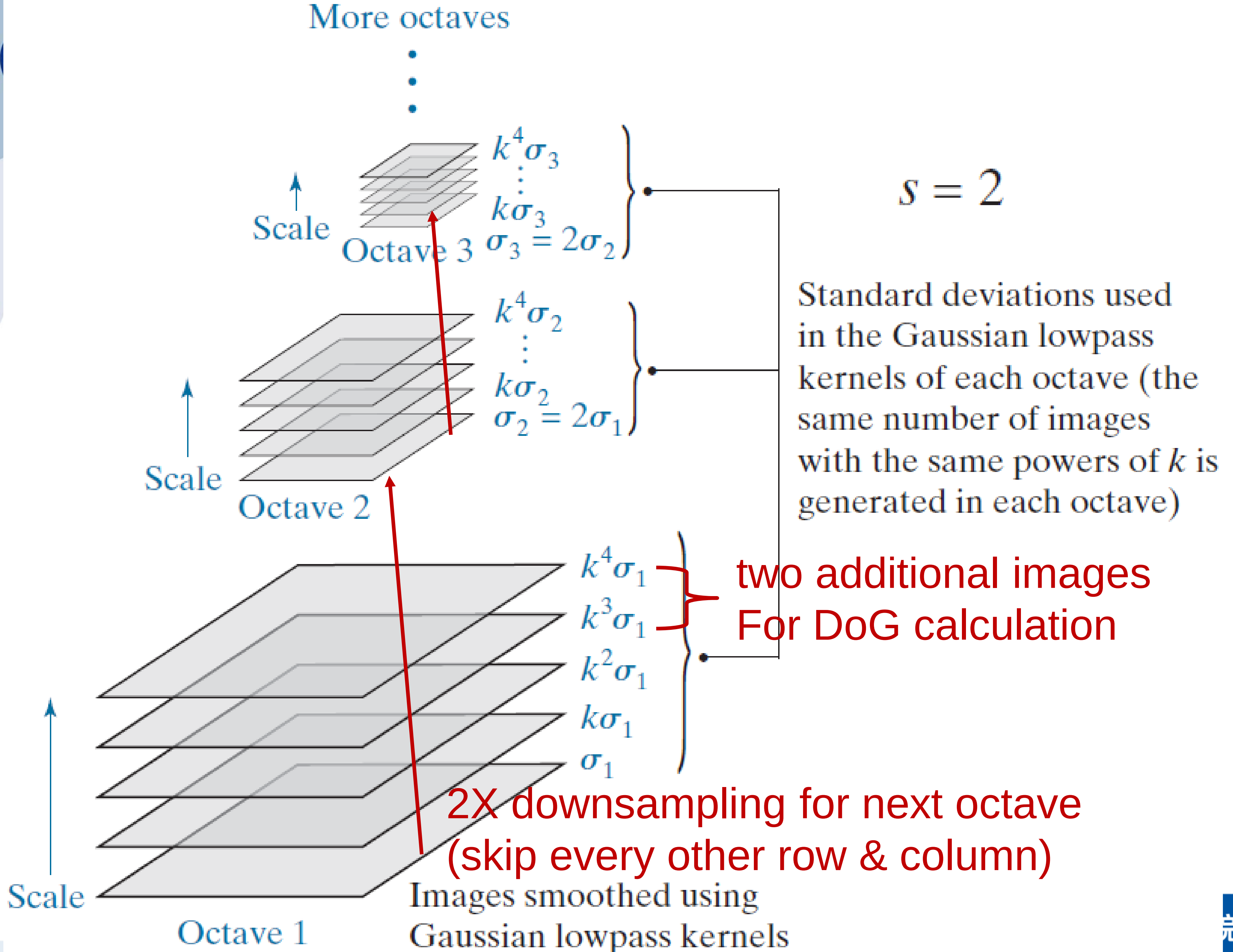
$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

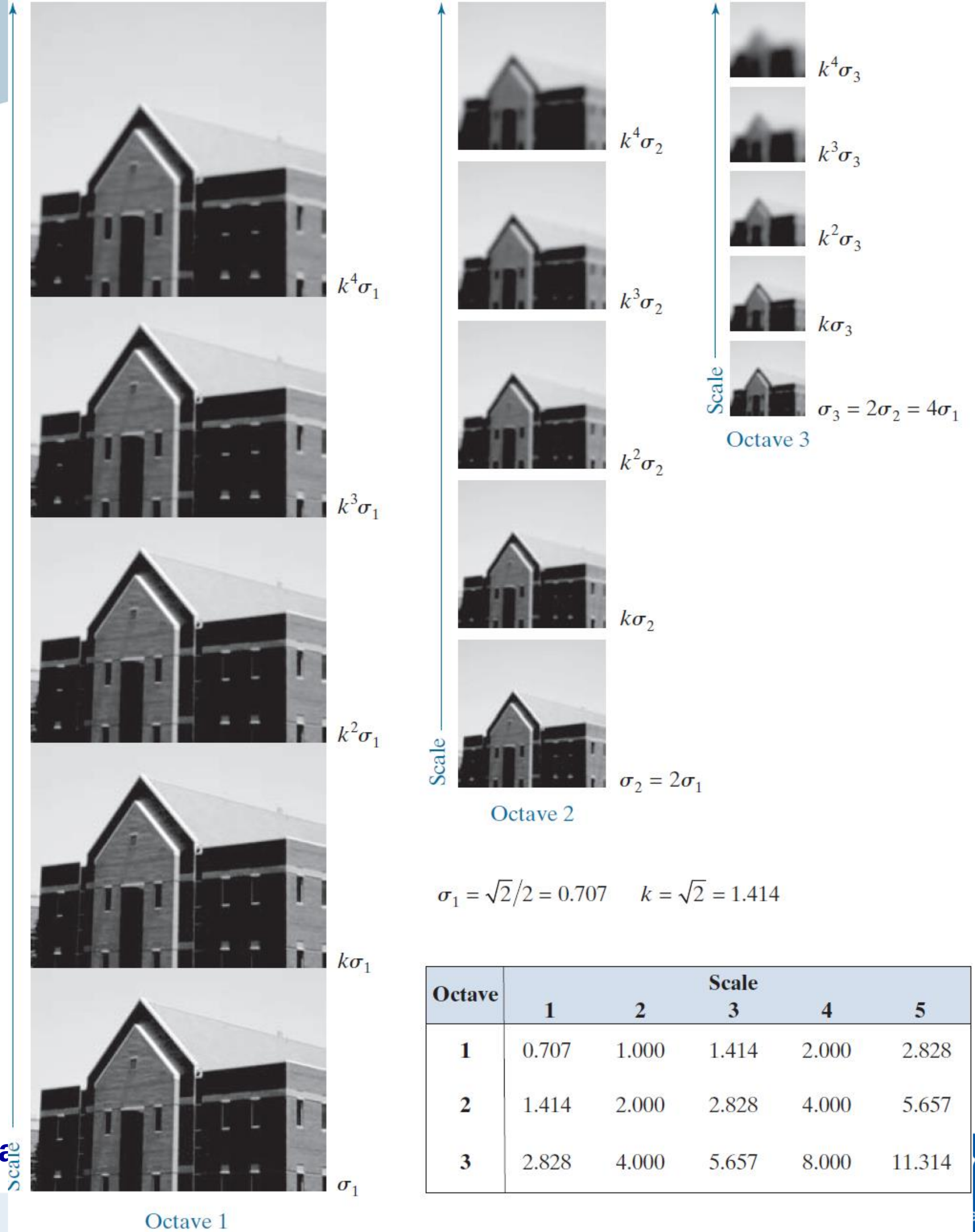
$$\sigma, k\sigma, k^2\sigma, k^3\sigma, \ldots$$

Octaves: $k^s\sigma = 2\sigma \Longrightarrow s = 2,\ k = \sqrt{2}$

Octave




More octaves
$k^4\sigma_3$
$k\sigma_3$
$\sigma_3 = 2\sigma_2$
Scale
Octave 3
$s = 2$
$k^4\sigma_2$
$k\sigma_2$
$\sigma_2 = 2\sigma_1$
Scale
Octave 2
Standard deviations used in the Gaussian lowpass kernels of each octave (the same number of images with the same powers of k is generated in each octave)
$k^4\sigma_1$
$k^3\sigma_1$
two additional images
For DoG calculation
$k^2\sigma_1$
$k\sigma_1$
σ_1
2X downsampling for next octave
(skip every other row & column)
Scale
Octave 1
Images smoothed using
Gaussian lowpass kernels



$\sigma_1 = \sqrt{2}/2 = 0.707 \qquad k = \sqrt{2} = 1.414$

Octave	Scale 1	2	3	4	5
1	0.707	1.000	1.414	2.000	2.828
2	1.414	2.000	2.828	4.000	5.657
3	2.828	4.000	5.657	8.000	11.314

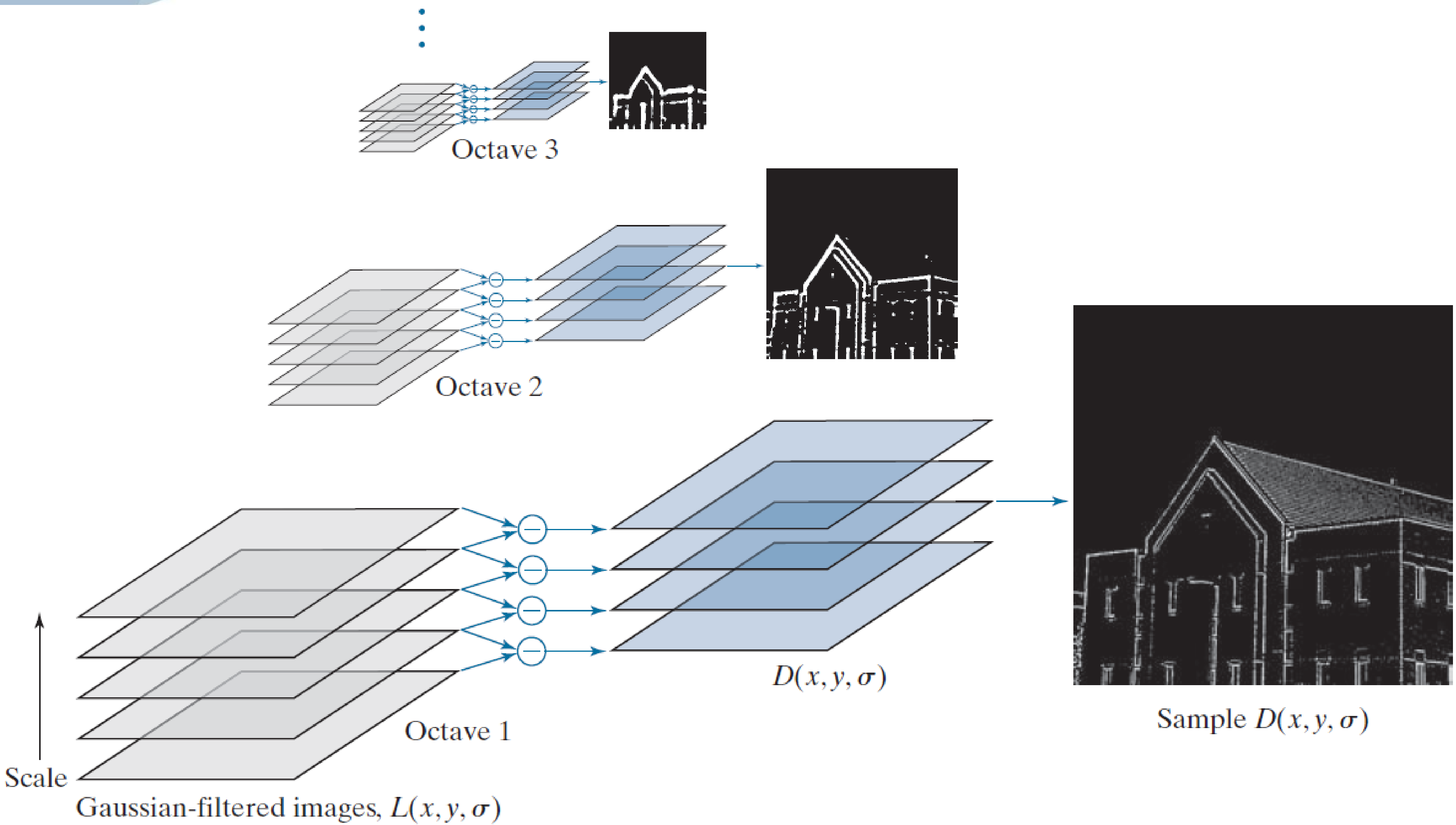
Find the Initial Keypoints

- Detect extrema in the difference of Gaussians of two adjacent scale-space images in an octave

$$
\begin{aligned}
D(x,y,\sigma) &= \left[G(x,y,k\sigma) - G(x,y,\sigma)\right] \star f(x,y) \\
&= L(x,y,k\sigma) - L(x,y,\sigma) \\
&\approx (k-1)\sigma^2\nabla^2 G
\end{aligned}
$$

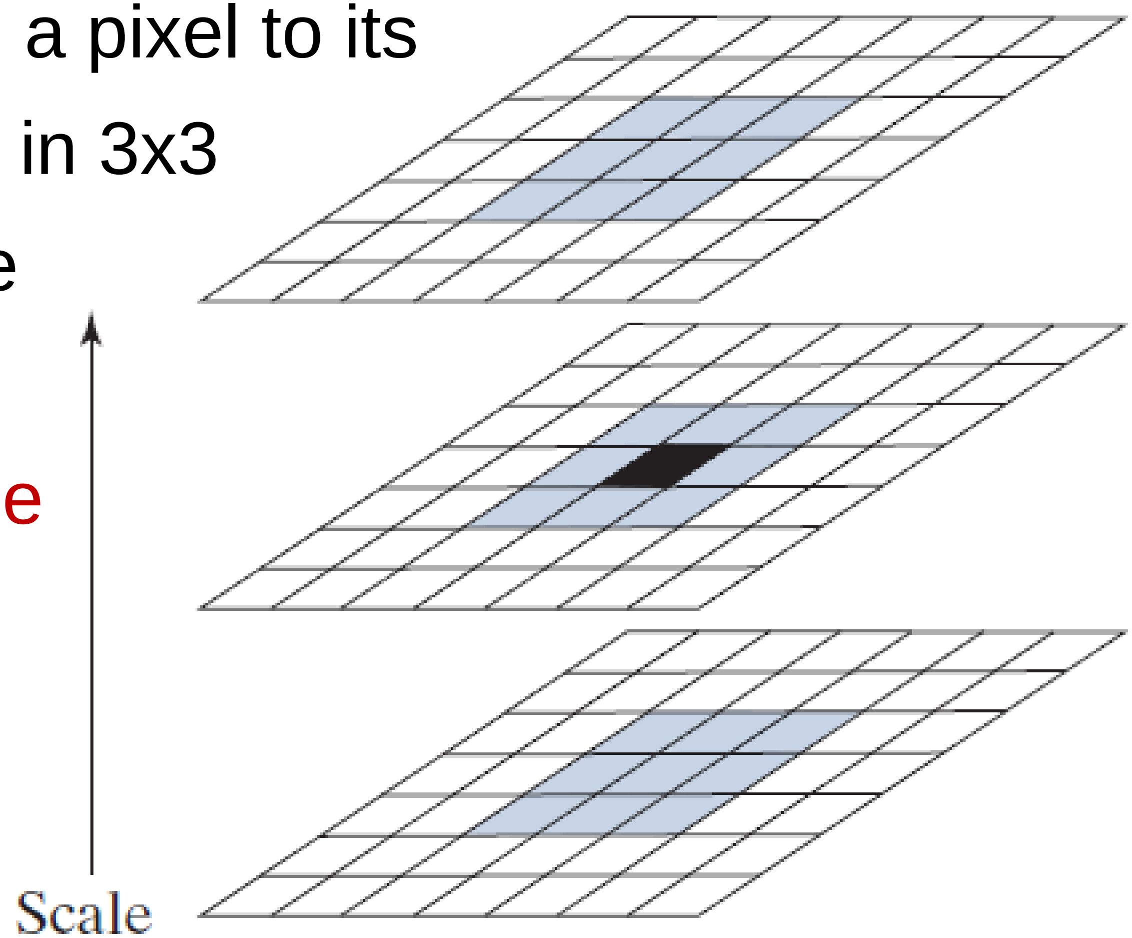
DoG: approximation to LoG

$s + 2$ difference functions



Detect local extrema (maxima or minima)

- Comparing a pixel to its 26 neighbors in 3x3 regions at the current and adjacent scale images



Corresponding sections of three contiguous $D(x, y, \sigma)$ images

Achieve subpixel accuracy

- Taylor series expansion of $D(x, y, \sigma)$

$$D(\mathbf{x}) = D + \left(\frac{\partial D}{\partial \mathbf{x}}\right)^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial D}{\partial \mathbf{x}}\right)\mathbf{x}$$

$$= D + (\nabla D)^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \mathbf{H}\,\mathbf{x}$$

Gradient operator

offset

$$\nabla D = \frac{\partial D}{\partial \mathbf{x}} = \begin{bmatrix} \partial D/\partial x \\ \partial D/\partial y \\ \partial D/\partial \sigma \end{bmatrix}$$

$$\mathbf{x} = (x, y, \sigma)^T$$

Hessian matrix $\mathbf{H} = \begin{bmatrix} \partial^2 D/\partial x^2 & \partial^2 D/\partial x \partial y & \partial^2 D/\partial x \partial \sigma \\ \partial^2 D/\partial y \partial x & \partial^2 D/\partial y^2 & \partial^2 D/\partial y \partial \sigma \\ \partial^2 D/\partial \sigma \partial x & \partial^2 D/\partial \sigma \partial y & \partial^2 D/\partial \sigma^2 \end{bmatrix}$

Achieve subpixel accuracy

- Location of the extremum

$$\hat{\mathbf{x}} = -\mathbf{H}^{-1}\left(\nabla D\right)$$

If the offset is greater than 0.5 in any of its three dimensions, move to the closer integer point and redo interpolation

- Extremum

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2}\left(\nabla D\right)^{T}\hat{\mathbf{x}}$$

Eliminating Edge Response

- Quantify difference between edges and corners

Eigenvalues of Hessian matrix are proportional to the local curvature of D

r < Threshold

$$\mathbf{H} = \begin{bmatrix} \partial^2 D/\partial x^2 & \partial^2 D/\partial x \partial y \\ \partial^2 D/\partial y \partial x & \partial^2 D/\partial y^2 \end{bmatrix} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix}$$

Largest Smallest Eigenvalue

$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta$$

$$\alpha = r\beta$$

$$\mathrm{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$$

$$\frac{[\mathrm{Tr}(\mathbf{H})]^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}$$

Eliminating Edge Response

$$\frac{[\mathrm{Tr}(\mathbf{H})]^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r}$$

Increases with $r \geq 1$, with minimum at $r = 1$

$r < 0$? Discard this point

- Keep "corner-like" point if

$$\frac{[\mathrm{Tr}(\mathbf{H})]^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r + 1)^2}{r}$$

e.g. $r = 10$

Example of SIFT keypoints

Keypoint orientation

- Gradient magnitude

$$M(x,y)=\left[\left(L(x+1,y)-L(x-1,y)\right)^2+\left(L(x,y+1)-L(x,y-1)\right)^2\right]^{\frac{1}{2}}$$

- Orientation angle

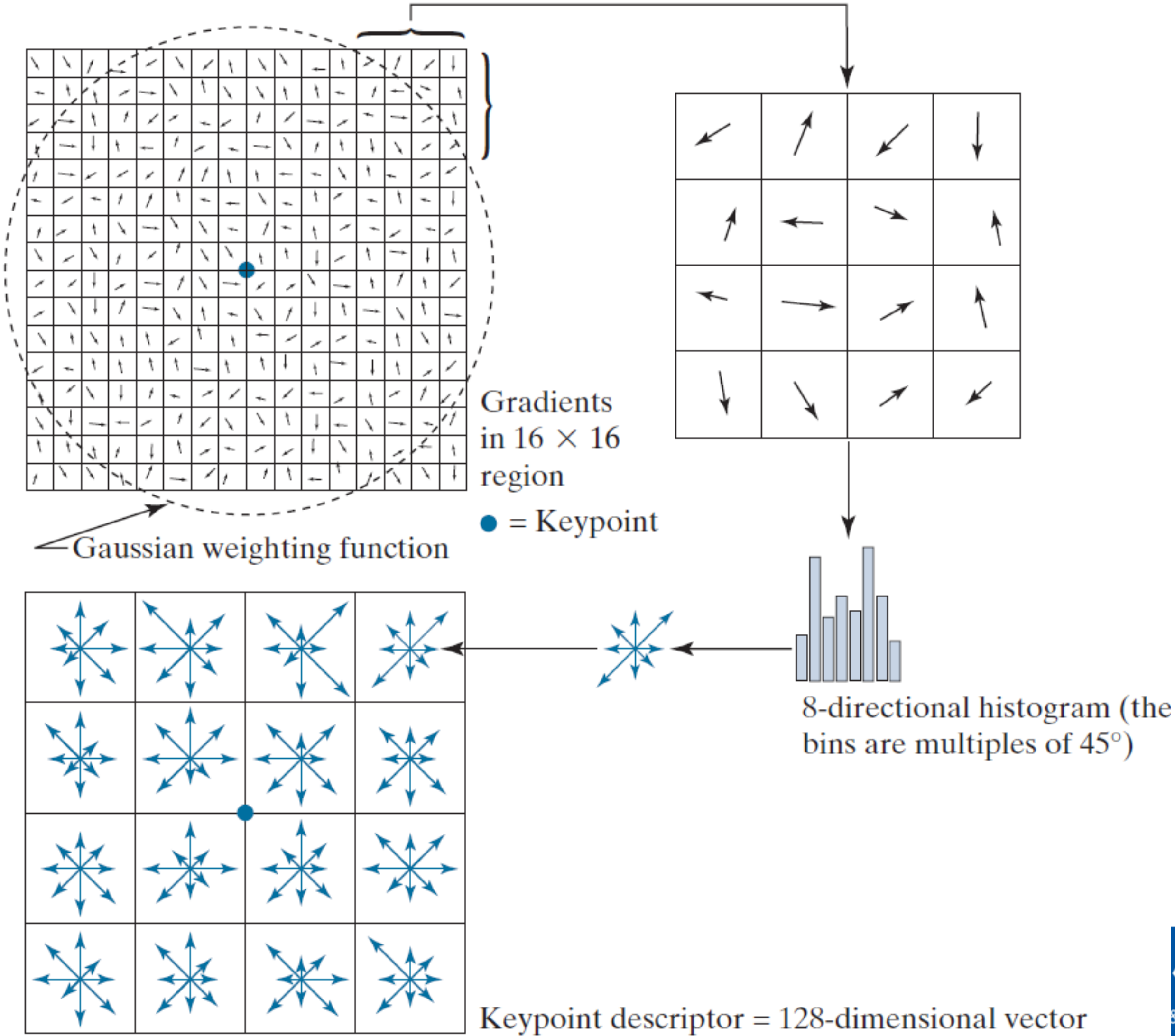
$$\theta(x,y)=\tan^{-1}\left[\left(L(x,y+1)-L(x,y-1)\right)/\left(L(x+1,y)-L(x-1,y)\right)\right]$$

- Histogram of orientations
  - Neighborhood of each keypoint
  - Weighted by its gradient magnitude
  - By a circular Gaussian function with 1.5 σ
  - 360 degrees → 36 bins
- Highest peak and ≥ 80% in the histogram
- Parabola fit to interpolate the peak position

ISEE

Keypoint descriptors

- Keypoint: location, scale, orientation
- Descriptor: local region around each keypoint



Summary of SIFT algorithm

$\sigma = 1.6,\ s = 2,$ three octaves

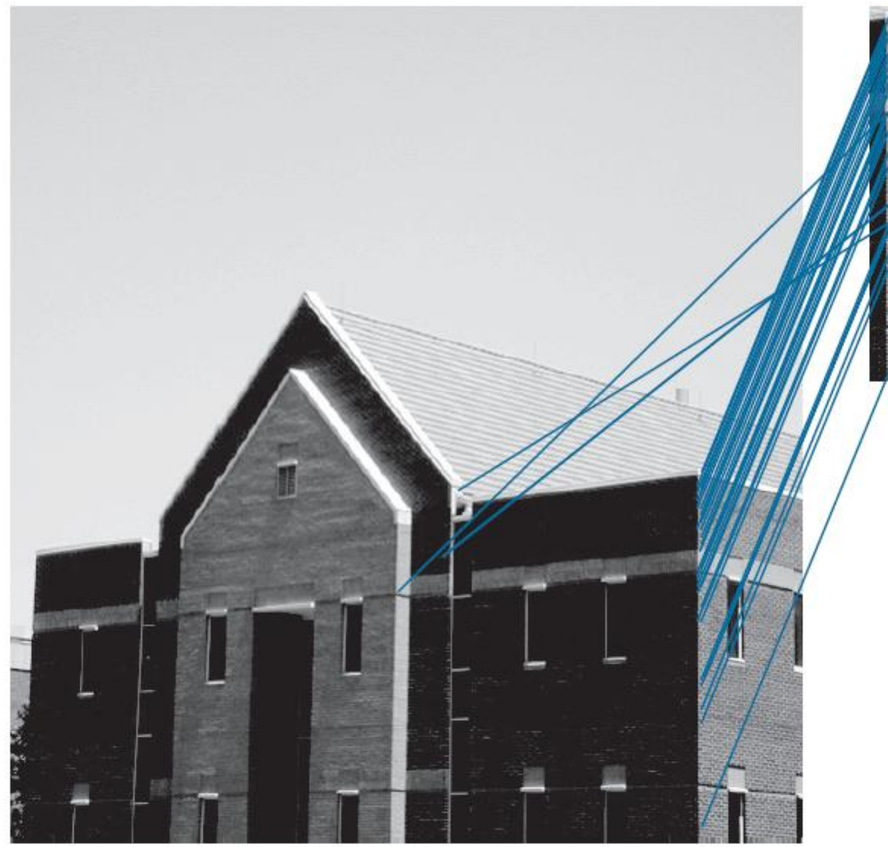
1. Construct the scale space
2. Obtain the initial keypoints
3. Improve the location of keypoints
4. Delete unsuitable keypoints
   – Low value of *D*
   – Edge
5. Compute keypoint orientations
6. Compute keypoint descriptors

   128-dimensional feature vector

Image matching using SIFT

54 keypoints

643 keypoints



Matched: 36 keypoints
3 are incorrect

Image matching using SIFT

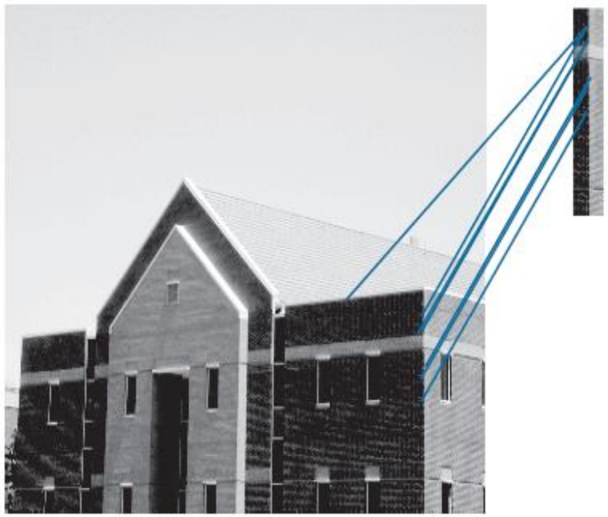
49 keypoints

Rotated by 5 degrees
547 keypoints

Matched: 26 keypoints
2 are incorrect

Image matching using SIFT

24 keypoints



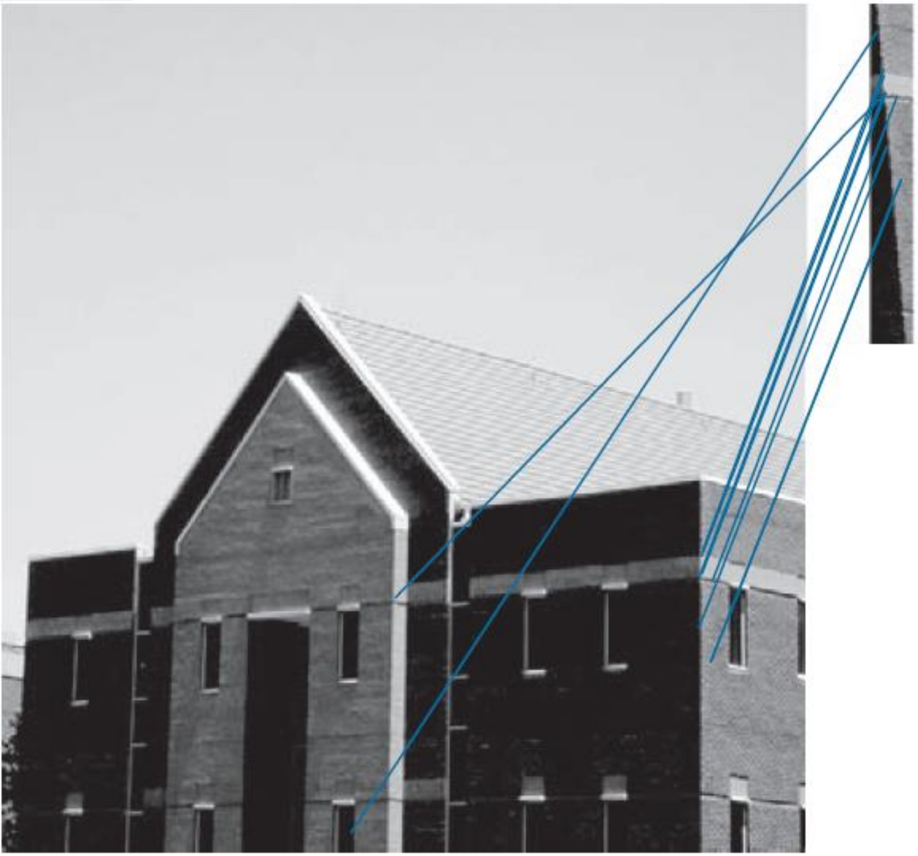
Half size in both directions
195 keypoints

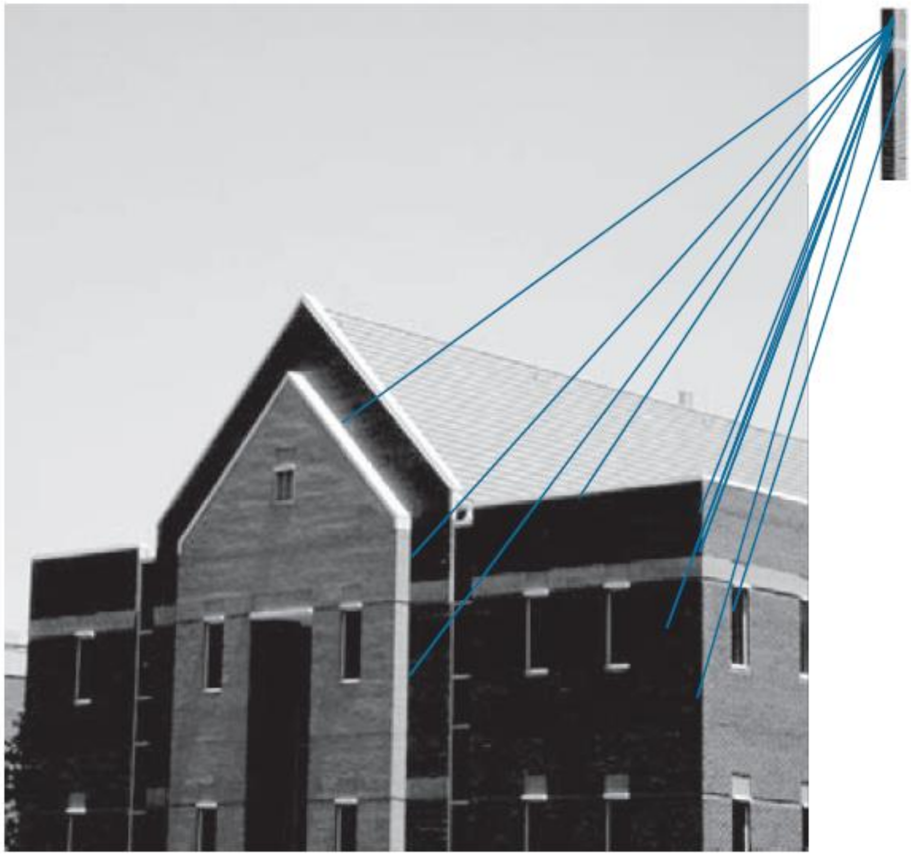
Matched: 7 keypoints
1 is incorrect

Image matching using SIFT

Rotated

Half-sized





Matched: 10 keypoints
2 are incorrect

Matched: 11 keypoints
4 are incorrect