2020 Future Defense Seminar

Feature Processing (Gradient-based)

SIFT, SURF

Sohee Lim

2020/10/29

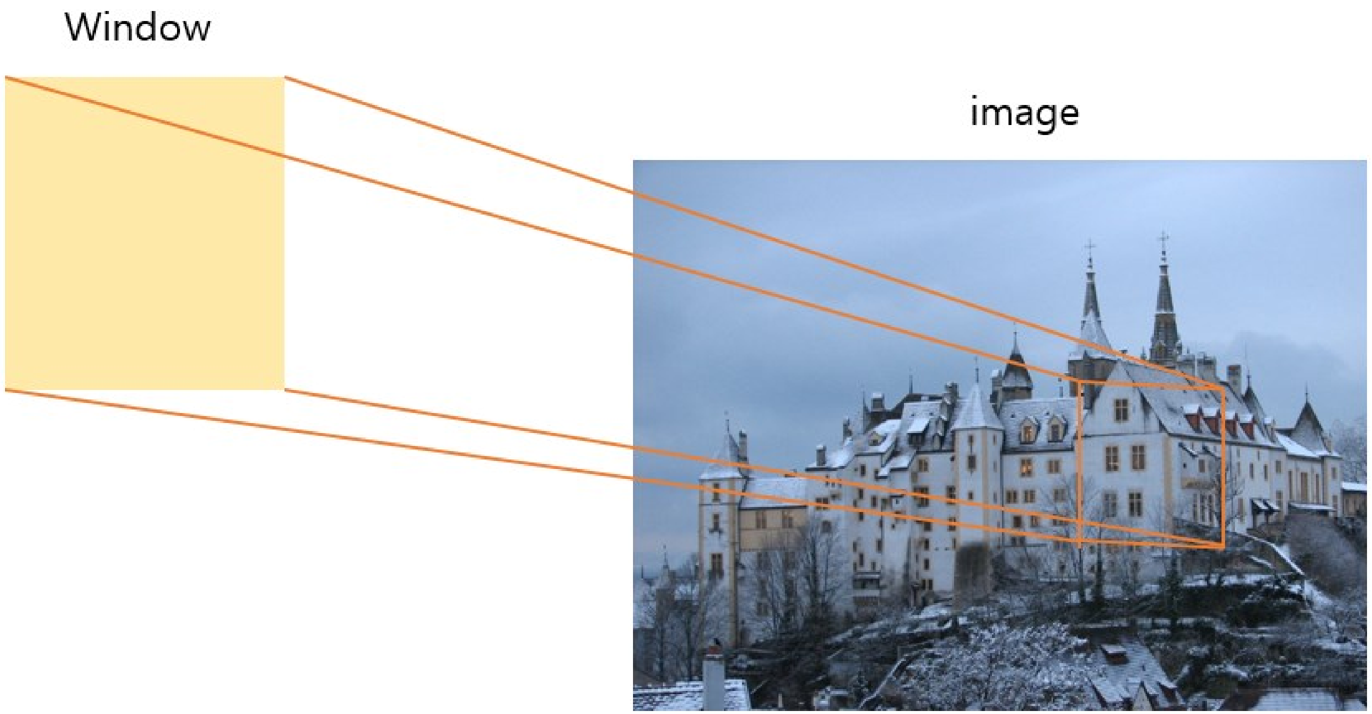
Keywords: Feature Extraction, Keypoints, Descriptors,

SIFT

Scale-Invariant Feature Transform

Background

- Scale



Background

- Scale

Large window

Small window

Same window

Seen from far

Seen from near

Background

- Scale

Large scale image

Small scale image

Different areas, same image size

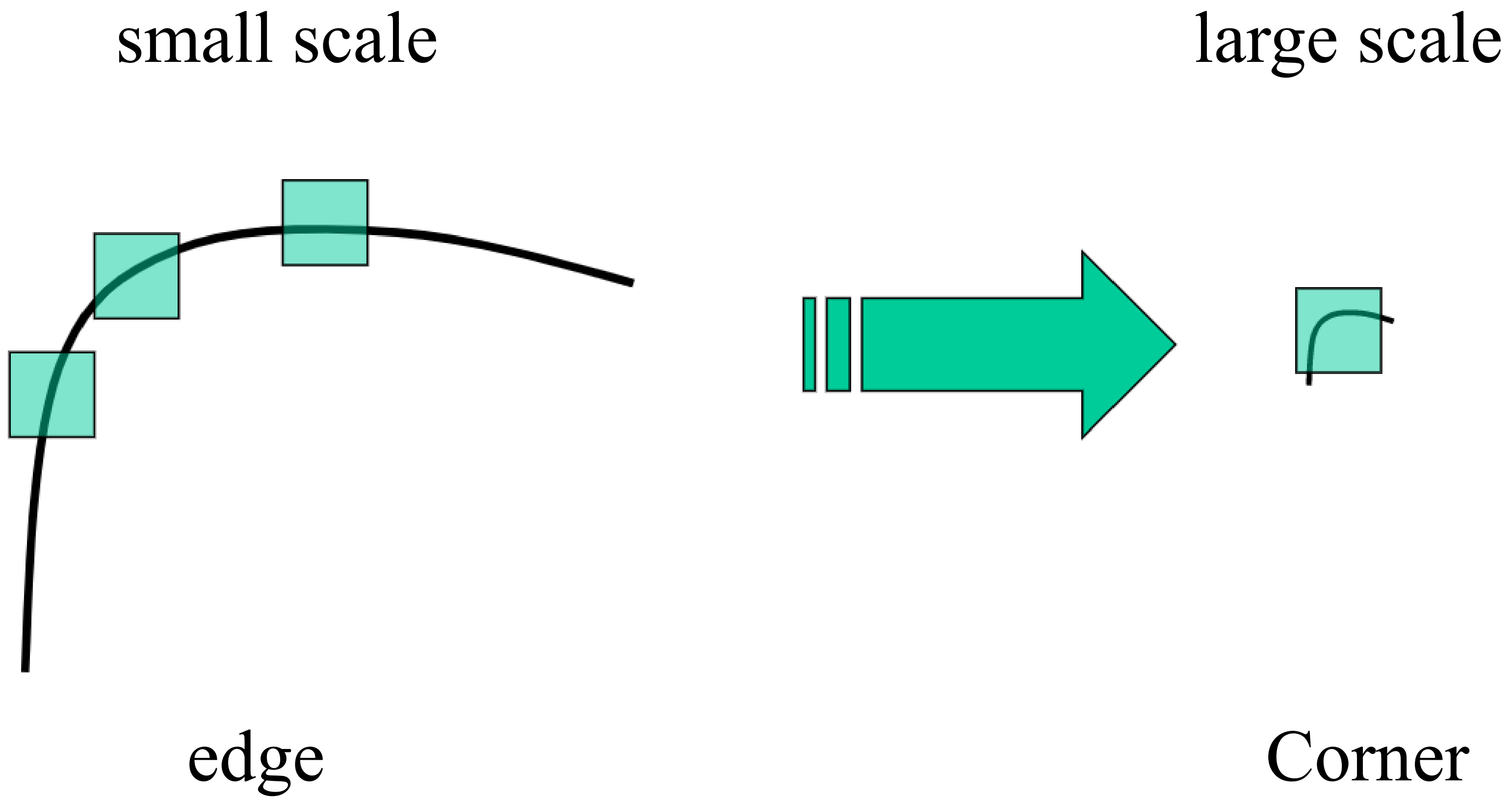
Background

- Scale

Same Area, Different Image Size

Background

- Harris corner detector is not scale-invariant



Background

- Scale Invariance

  - From Wikipedia

  In physics, mathematics and statistics, **scale invariance** is a **feature of objects or laws that do not change** if scales of length, energy, or other variables, are multiplied by a common factor, and thus represent a universality.

  - From stackExchange

  Scale invariance can be thought of as '**self-similarity**'. What this really means is that regardless of how much you zoom into or out of an object (be it a function, or a physical object, or the like) it looks exactly the same.

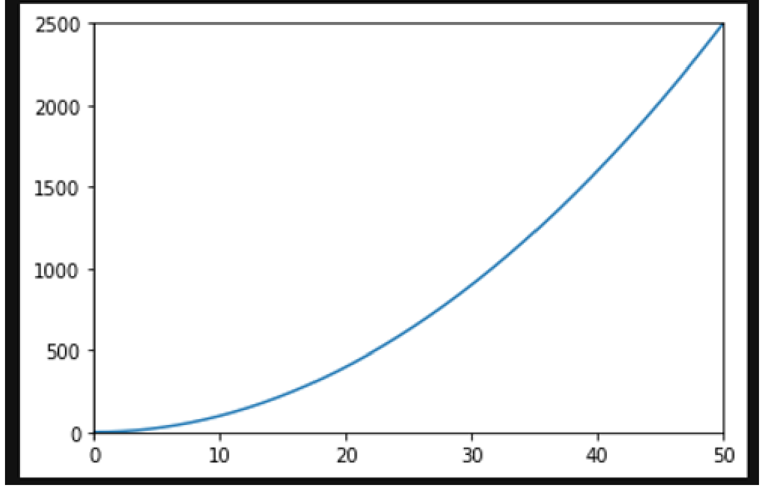
Background

- Scale Invariance

```
x = np.arange(0,100,0.1)
y = np.power(x, 2)

plt.xlim(0,50)
plt.ylim(0,2500)
plt.plot(x,y)
```

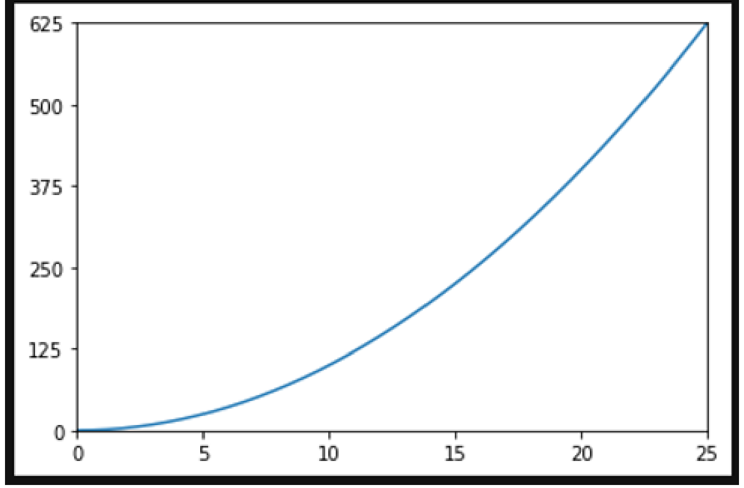
[<matplotlib.lines.Line2D at 0x235beaf1898>]




```
x = np.arange(0,50,0.1)
y = np.power(x, 2)

plt.xlim(0,50 * 1/2)
plt.ylim(0,2500 * 1/np.power(2,2))
plt.yticks(np.arange(0,626,step=125))
plt.plot(x,y)
```

[<matplotlib.lines.Line2D at 0x235c1c0dcc0>]

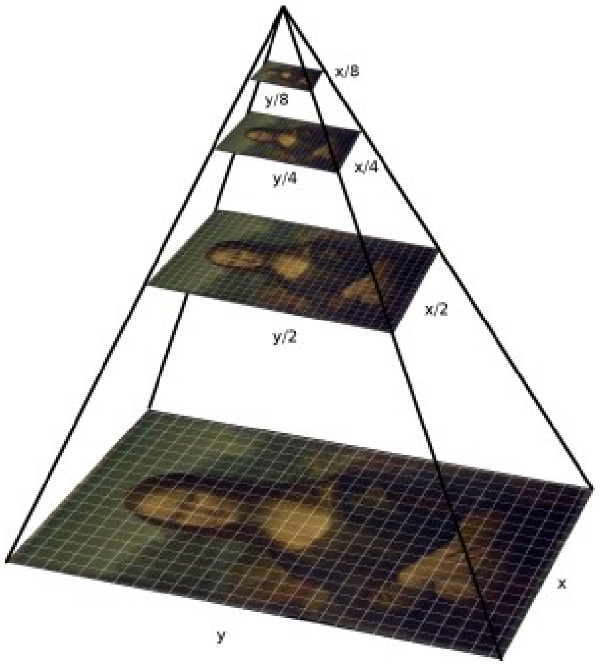


Scale-Invariant Feature Transform (SIFT)

- Scale-Space Extrema Detection
- Keypoint Localization
- Orientation Assignment
- Keypoint Descriptor

Scale-Space Extrema Detection

- Scale-space
  - Image Pyramid

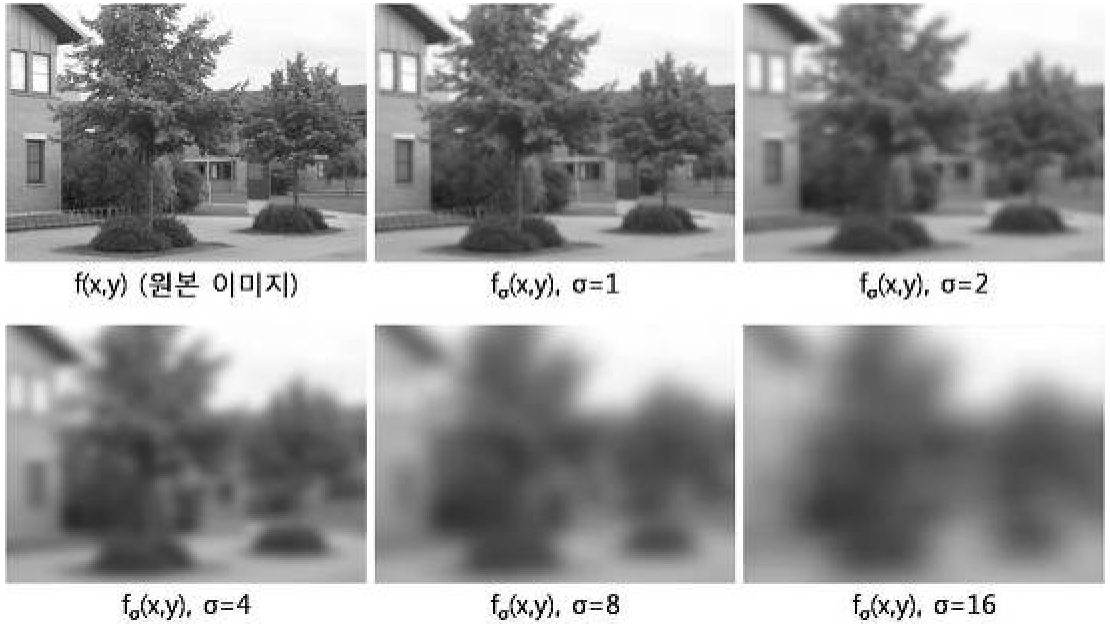


Scale-Space Extrema Detection

- Scale-space
  - Gaussian Filter

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

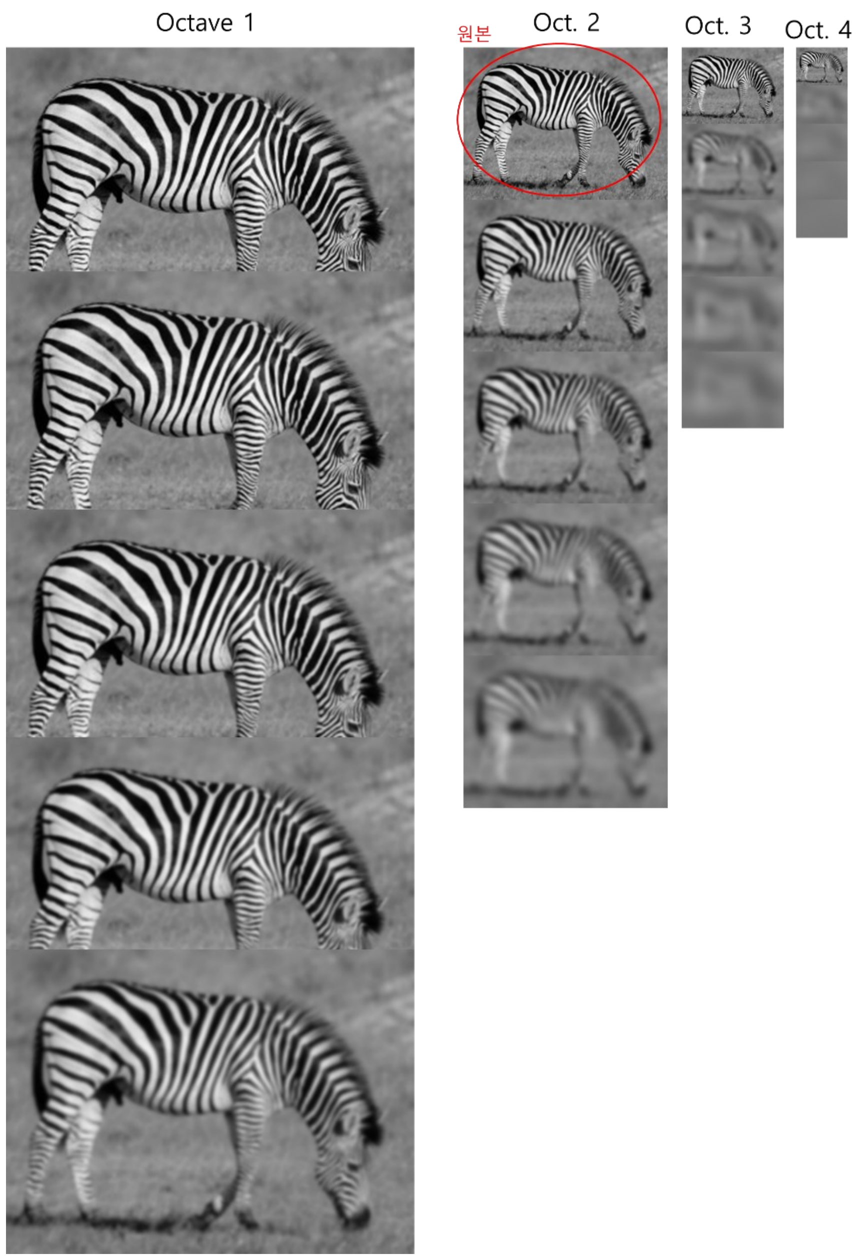
σ: scale parameter



$$f_\sigma(x, y) = g_\sigma(x, y) * f(x, y)$$

Scale-Space Extrema Detection

- Scale-space
  - Scale Step $k = \sqrt{2}$
    $\sigma_{i+1} = k\ \sigma_i$
  - Octave pyramid
  - $k = \sqrt{2}$
  - $g_{\sigma_s} * g_{\sigma_t} = g_\sigma$
  - $\sigma = \sqrt{\sigma_s^2 + \sigma_t^2}$



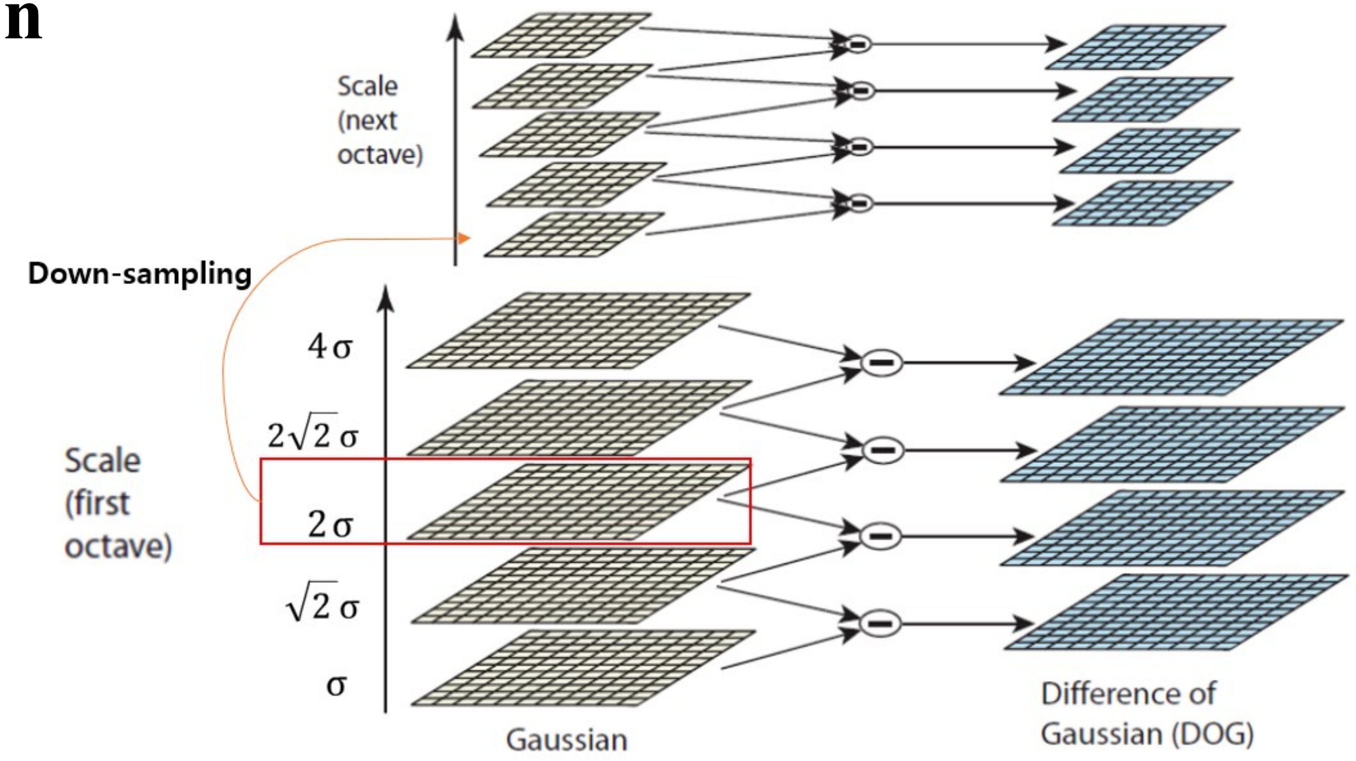
Scale-Space Extrema Detection

- Difference of Gaussian (DoG)

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

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

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}$$

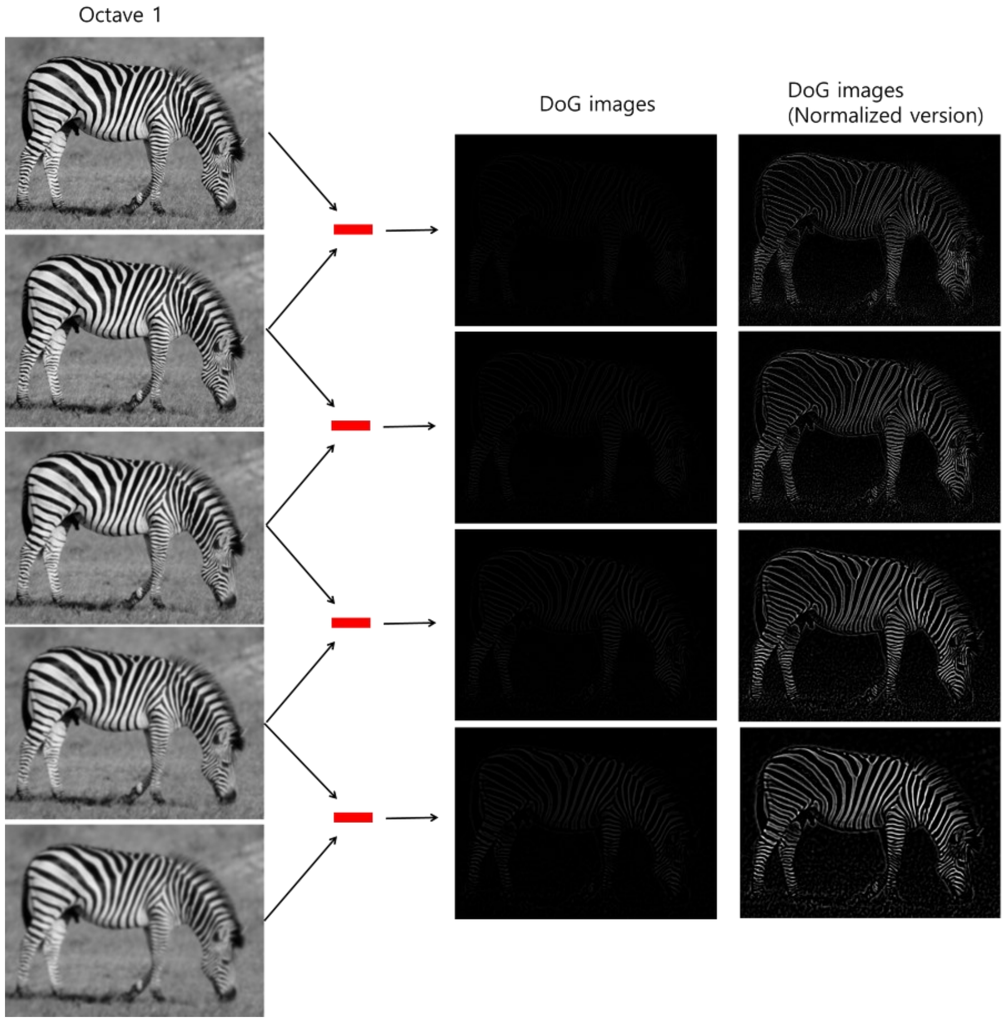


1Octave									
Gaussian	σ		$\sqrt{2}\sigma$		2σ		$2\sqrt{2}\sigma$		4σ
DOG		σ		$\sqrt{2}\sigma$		2σ		$2\sqrt{2}\sigma$	

2Octave									
Gaussian	2σ		$2\sqrt{2}\sigma$		4σ		$4\sqrt{2}\sigma$		8σ
DOG		2σ		$2\sqrt{2}\sigma$		4σ		$4\sqrt{2}\sigma$	

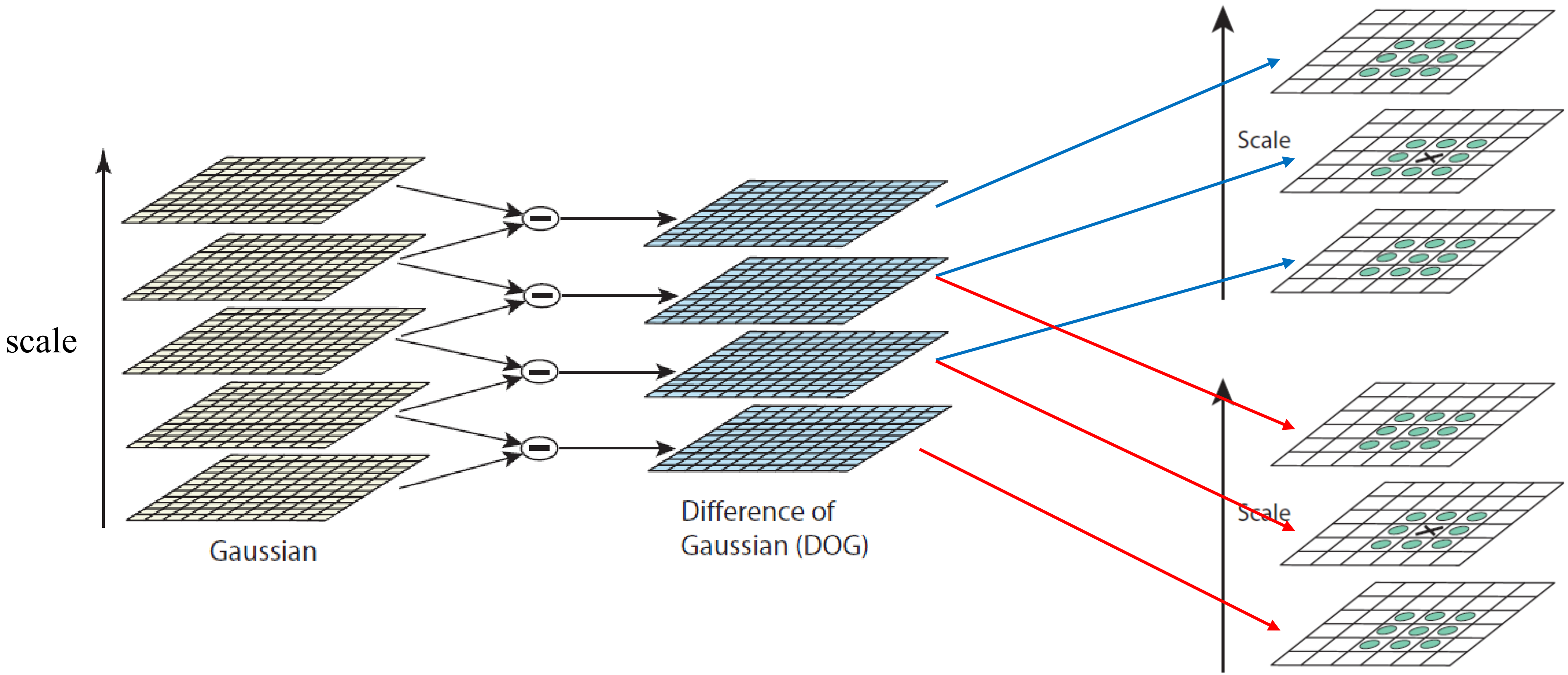
Scale-Space Extrema Detection

- Difference of Gaussian (DoG)



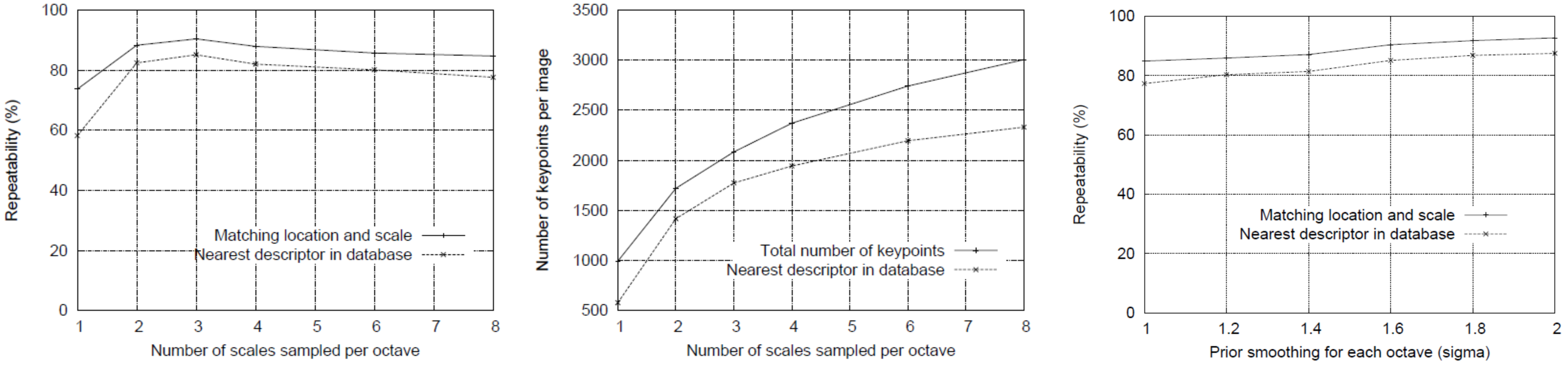
Scale-Space Extrema Detection

- Local extrema detection



Scale-Space Extrema Detection

- Optimal Parameters
  - number of octaves $= 4$
  - number of scale levels $= 5$
  - $\sigma = 1.6$
  - $k = \sqrt{2}$



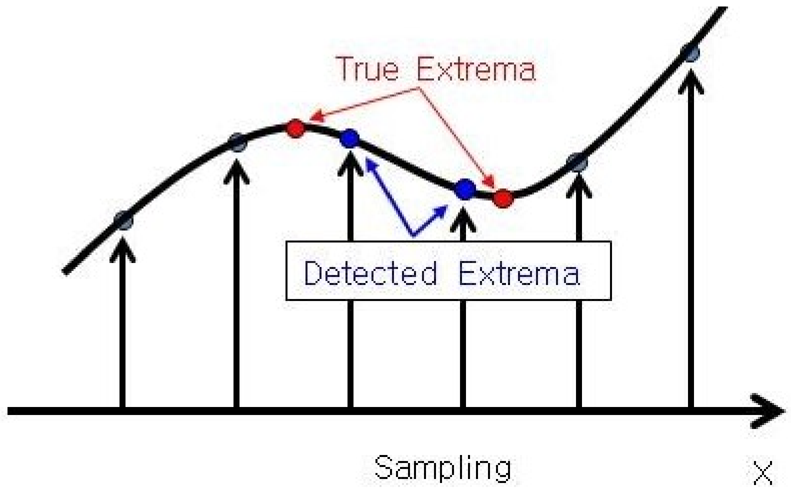
Keypoint Localization

- Taylor expansion (up to the quadratic terms)

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

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

▶ $$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}$$



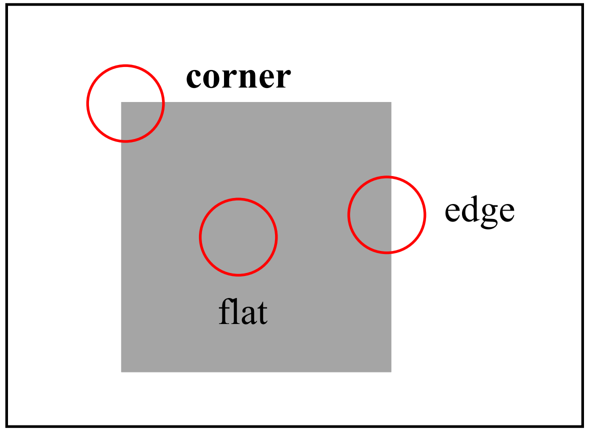
- Eliminating low constrast keypoints

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

if $|D(\hat{\mathbf{x}})| < 0.3 \Rightarrow$ discard

Keypoint Localization

- Eliminating edge response
  - A poorly defined peak in the difference-of-Gaussian function will have a large principal curvature across the edge but a small one in the perpendicular direction.
  - The eigenvalues of **H** are proportional to the principal curvatures of D.
  - Hessian matrix (**H**) – 2nd derivative



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

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

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

Let $\alpha = r\beta$. Then, $\dfrac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \dfrac{(\alpha+\beta)^2}{\alpha\beta} = \dfrac{(r\beta+\beta)^2}{r\beta^2} = \dfrac{(r+1)^2}{r}$

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r} \Rightarrow \text{accept} \qquad r = 10$$

Keypoint Localization

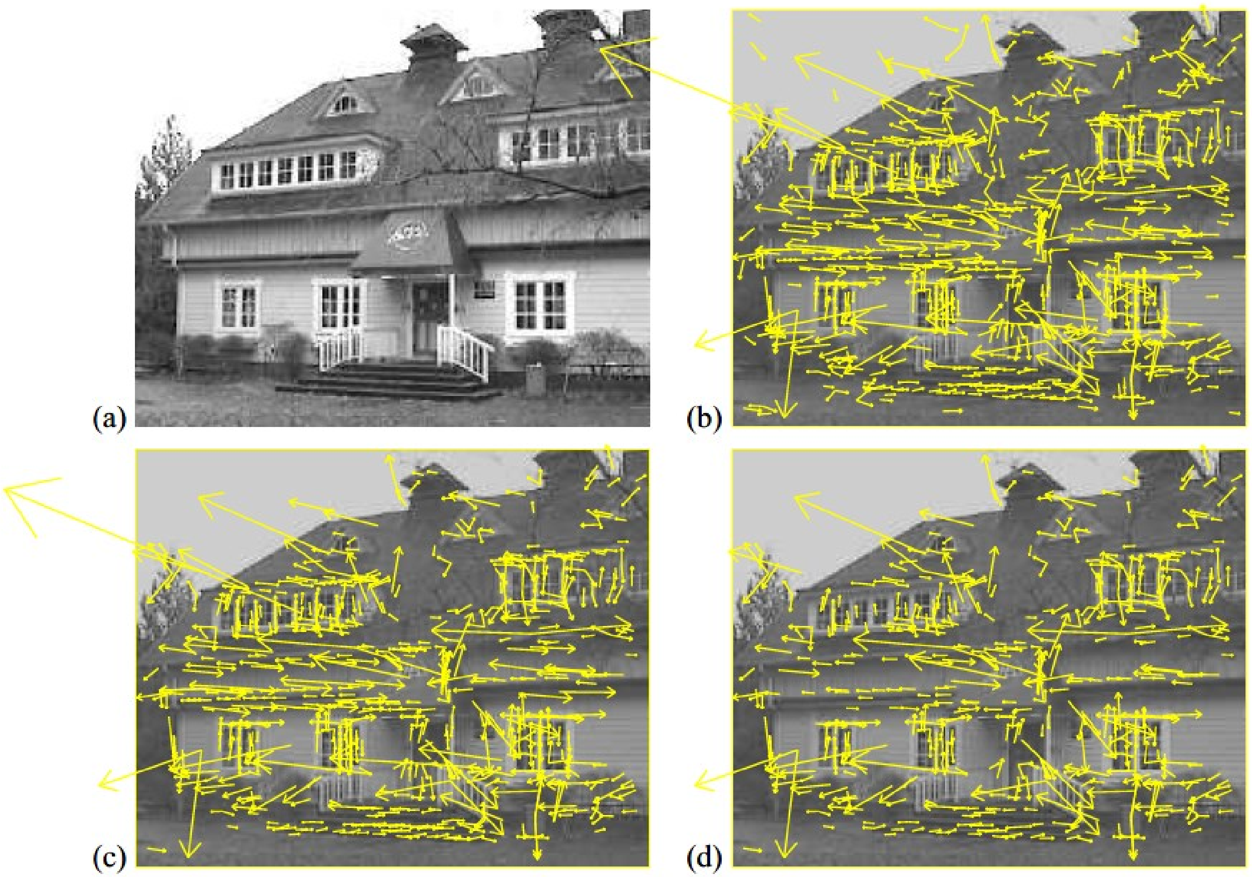


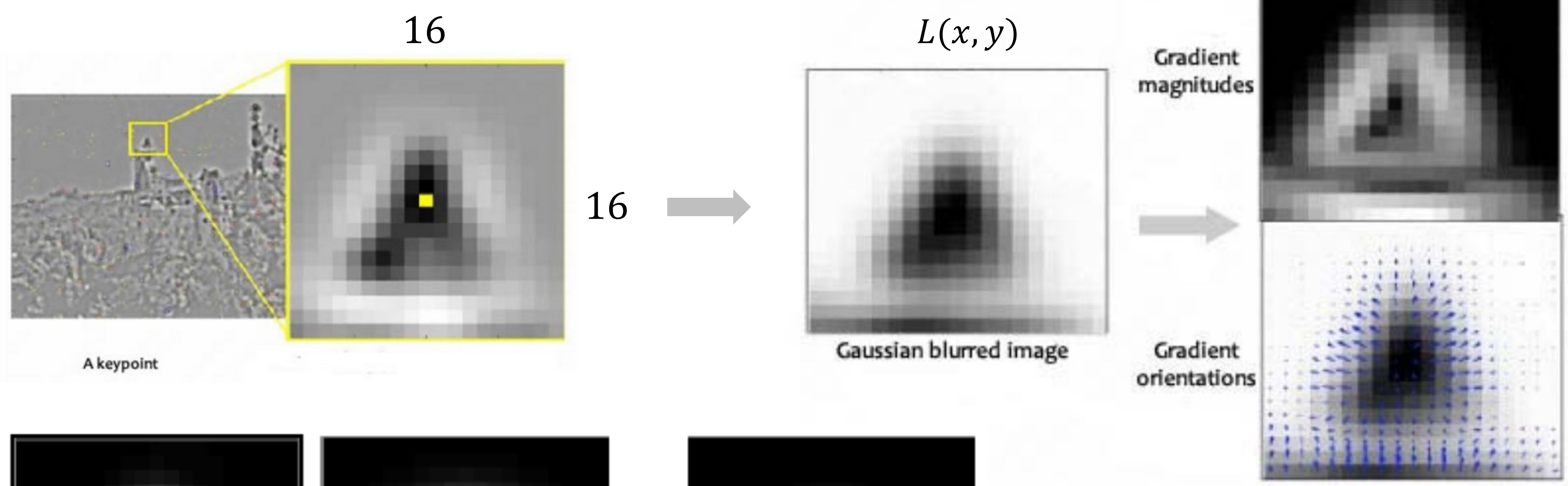
Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

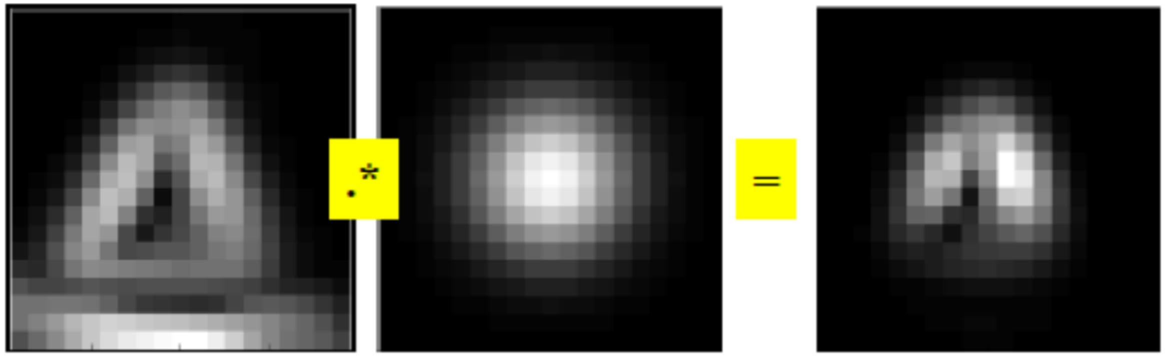
Orientation Assignment

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

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

- Rotation invariant





gradient magnitude

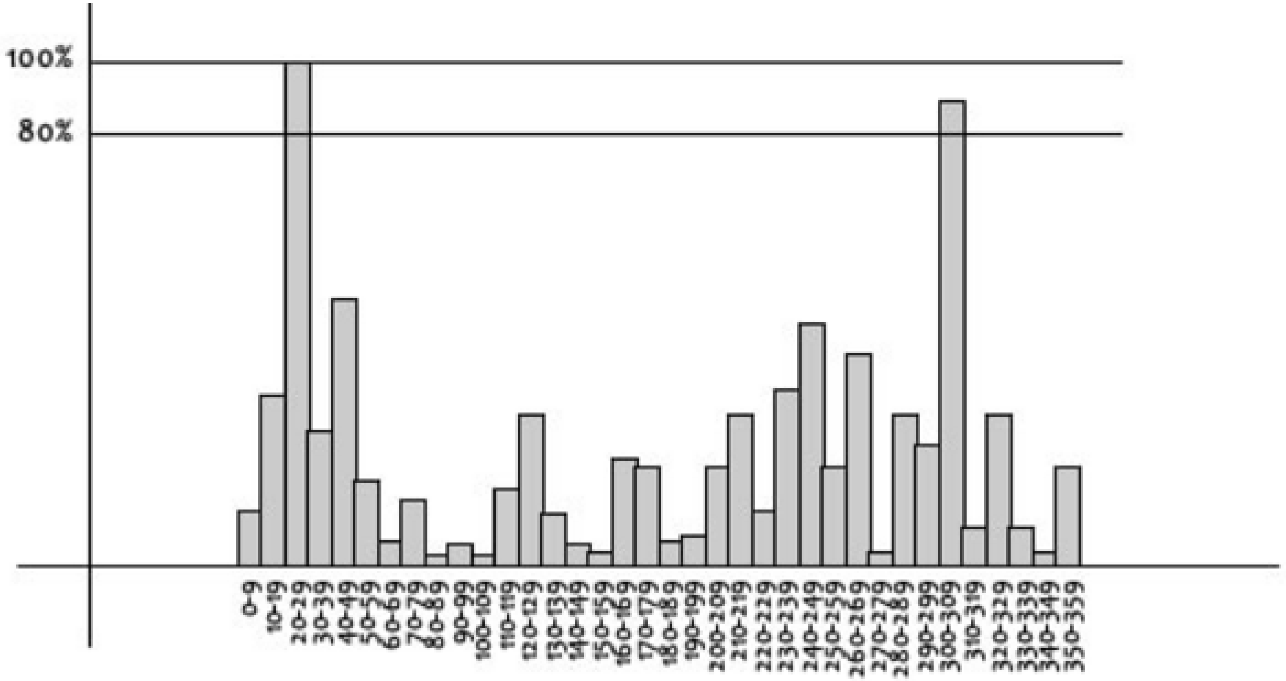
weighted by 2D gaussian kernel

$\sigma = 1.5 \times scale$

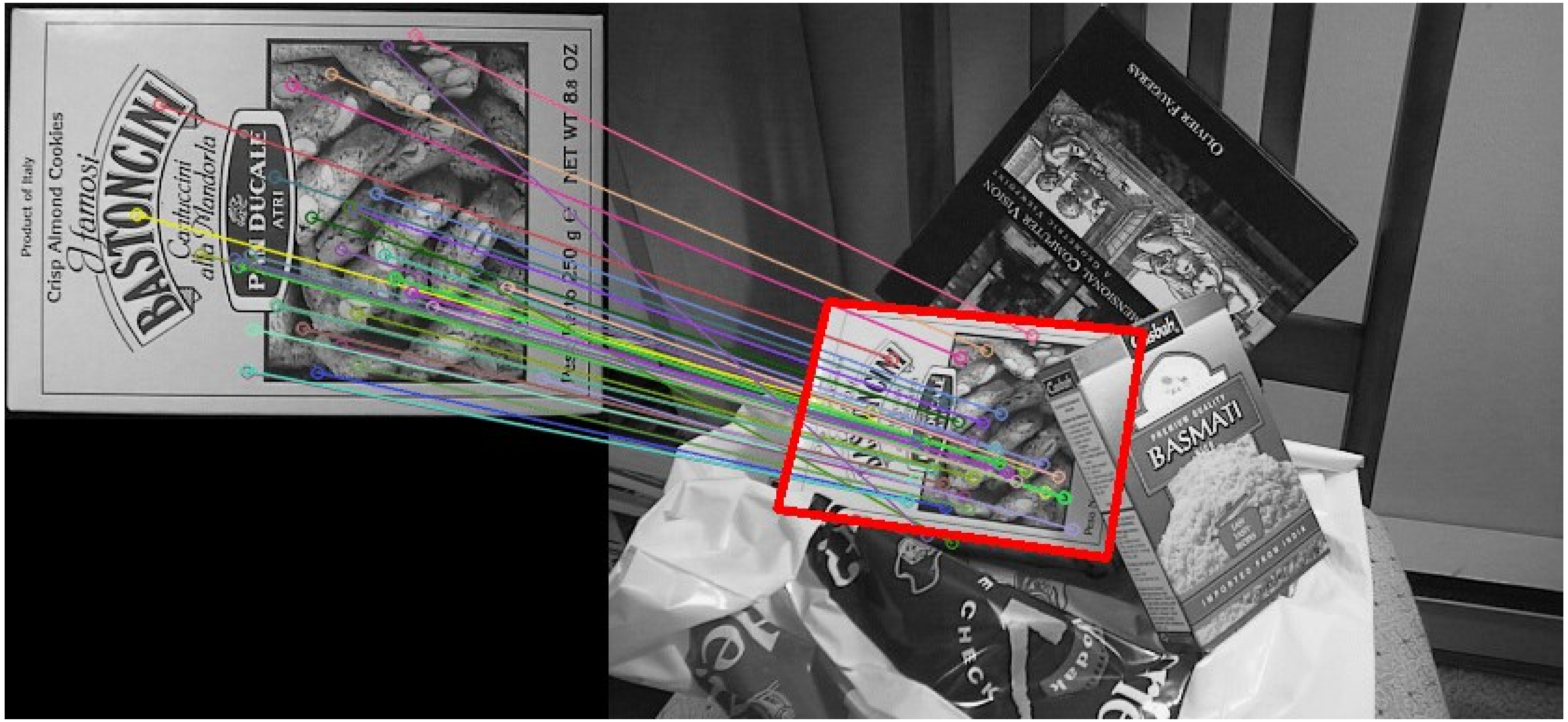
weighted gradient magnitude

Orientation Assignment

- Orientation histogram



Keypoint Descriptor



Keypoint Descriptor

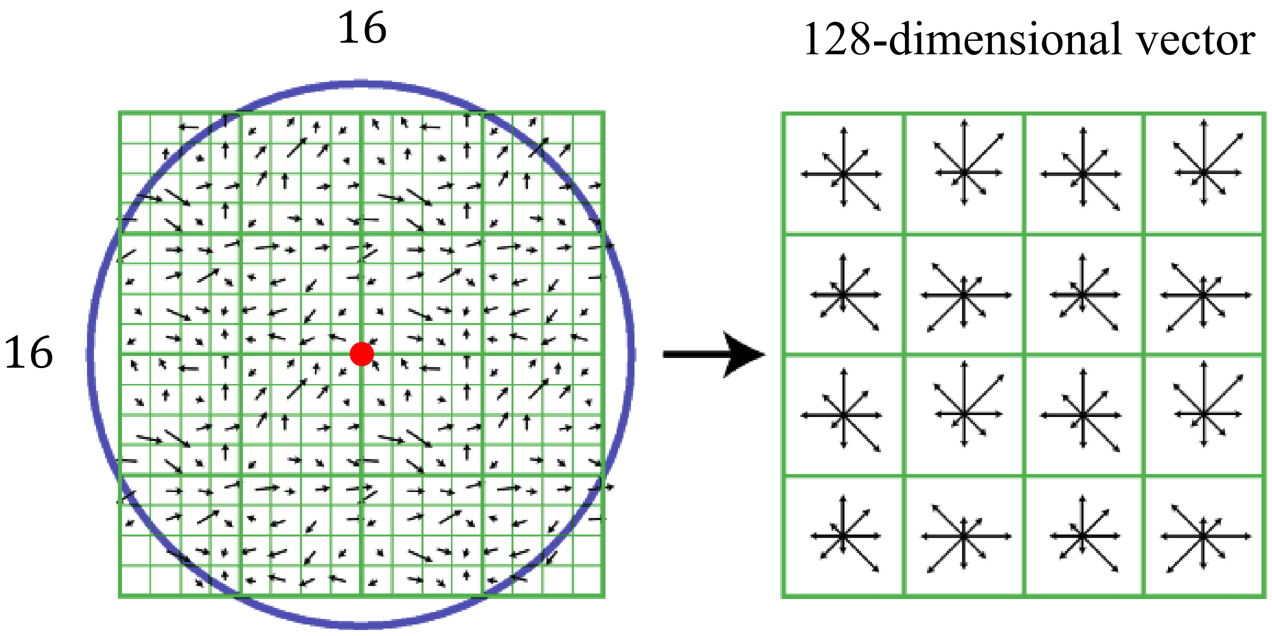


Image gradient

Keypoint descriptor

- 16×16 window
- Gaussian blurring ($\sigma = \frac{window\ size}{2}$)
- Compute gradient orientation, weighted magnitude
- Compute orientation histogram in 8 directions over 4×4 sample regions
- Rotation - rotation invariance
- Normalization - illuminance invariance

SURF

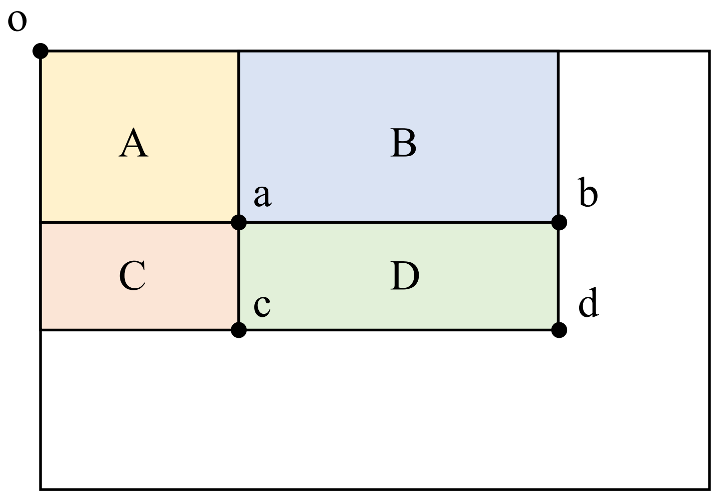
Speeded Up Robust Features

Interest Point Detection

- Integral images

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

$$I_{\Sigma}(\mathbf{x}) = \sum_{i=0}^{i \leqslant x} \sum_{j=0}^{j \leqslant y} I(i, j)$$



$$\begin{aligned} I(a) &= A \\ I(b) &= A + B \\ I(c) &= A + C \\ I(d) &= A + B + C + D \end{aligned}$$

D = I(d)+I(a)-I(b)-I(c)

IMAGE

0	1	1	1
1	2	2	3
1	2	1	1
1	3	1	0

INTEGRAL IMAGE

0	1	2	3
1	4	7	11
2	7	11	16
3	11	16	21

IMAGE

0	1	1	1
1	2	2	3
1	2	1	1
1	3	1	0

INTEGRAL IMAGE

0	1	2	3
1	4	7	11
2	7	11	16
3	11	16	21

$$21 + 4 - 11 - 11 = 3$$

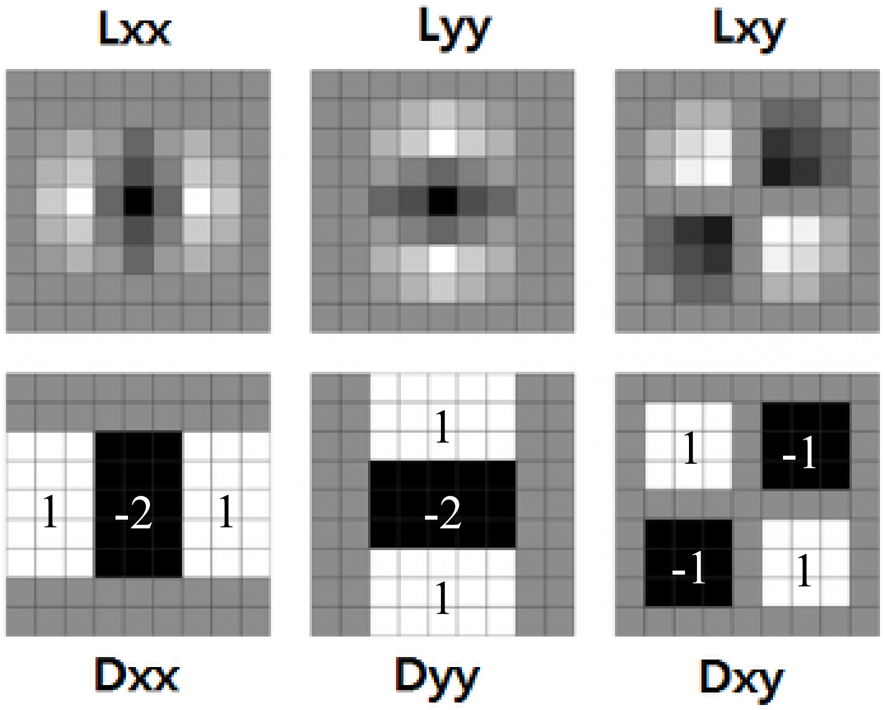
Interest Point Detection

- Hessian matrix-based interest points

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

- L_{xx}: Convolution of Gaussian 2nd derivative

Discretized 2nd Gaussian Derivatives



9×9 Box Filters

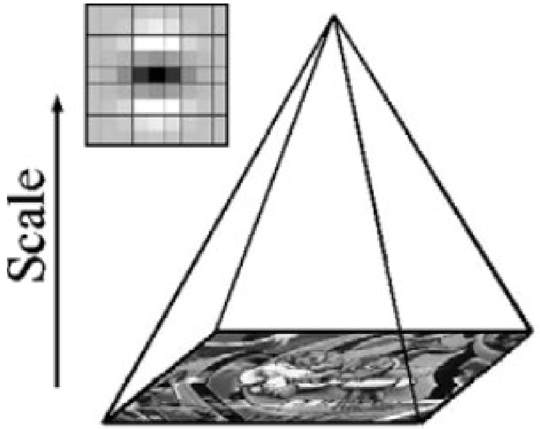
det(H)	sign of eigenvalues	extrema
$(-)$	$\neq$	$\times$
$(+)$	$=$	O

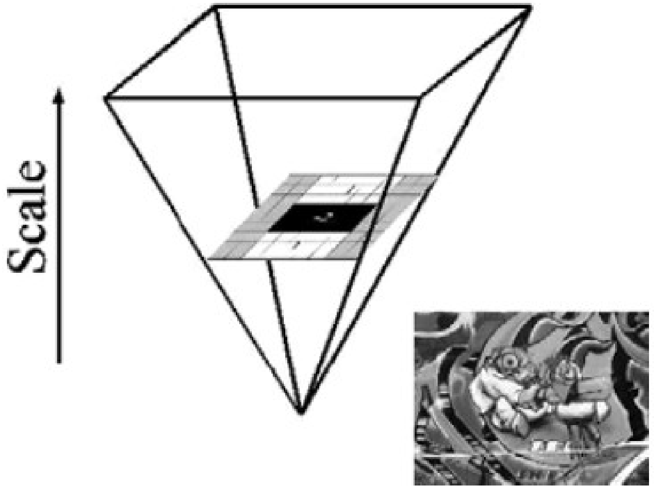
$$\det(\mathcal{H}_{\text{approx}})=D_{xx}D_{yy}-(wD_{xy})^2 \qquad \sigma=1.2$$

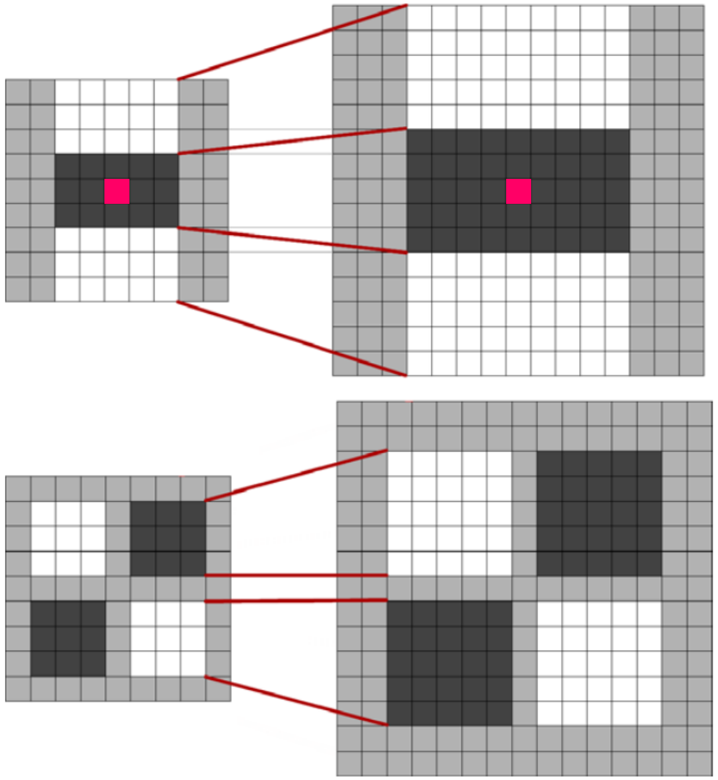
$$w=\frac{|L_{xy}(1.2)|_F\,|D_{yy}(9)|_F}{|L_{yy}(1.2)|_F\,|D_{xy}(9)|_F}=0.912...\simeq 0.9$$

Interest Point Detection

- Scale space representation







SIFT

- Fix filter size
- Down-sampling image size

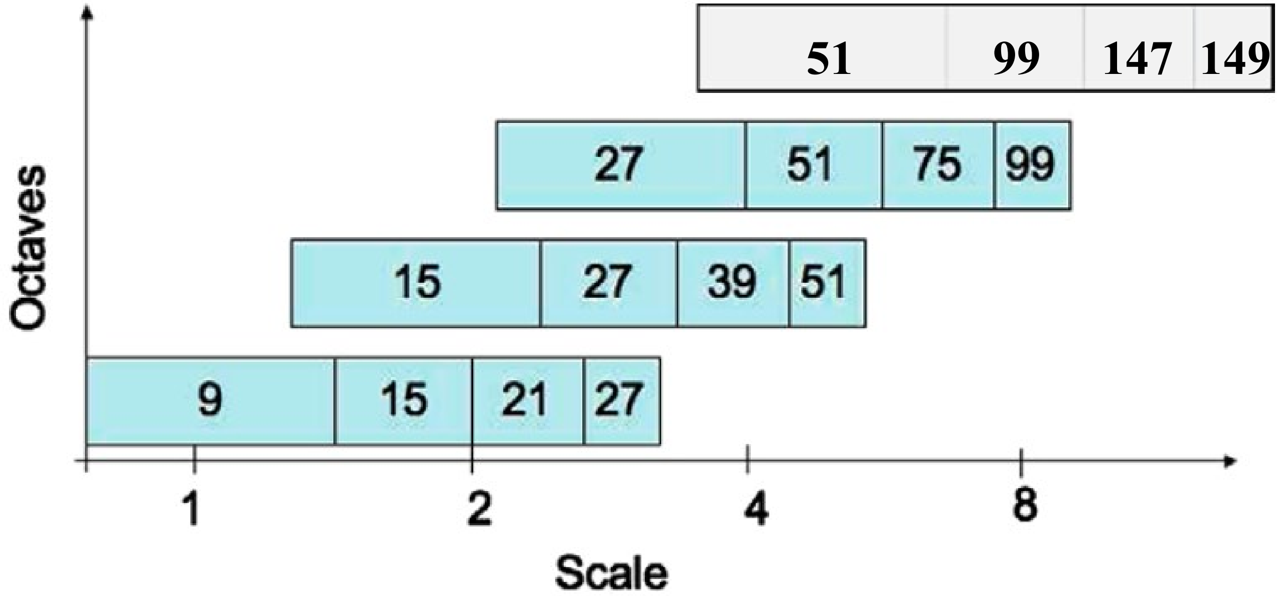
SURF

- Fix image scale
- Up-sampling filter size

(+) Computational efficiency
(+) No aliasing
(-) Limit scale-invariance

Interest Point Detection

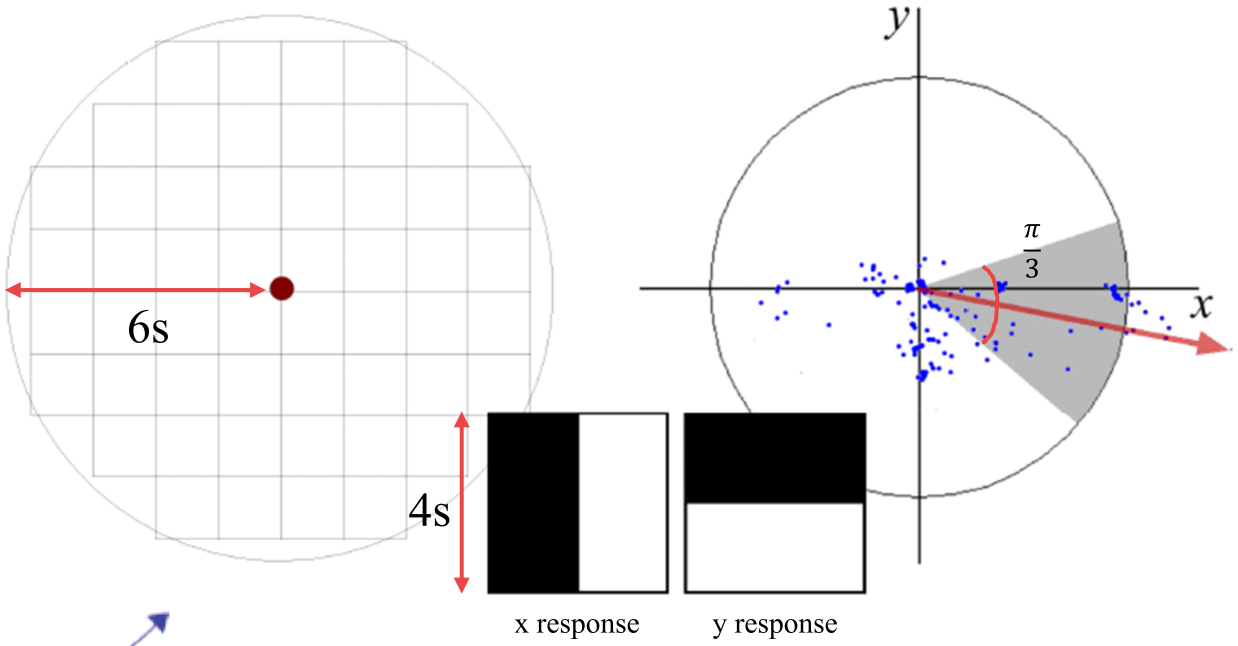
- Scale space representation



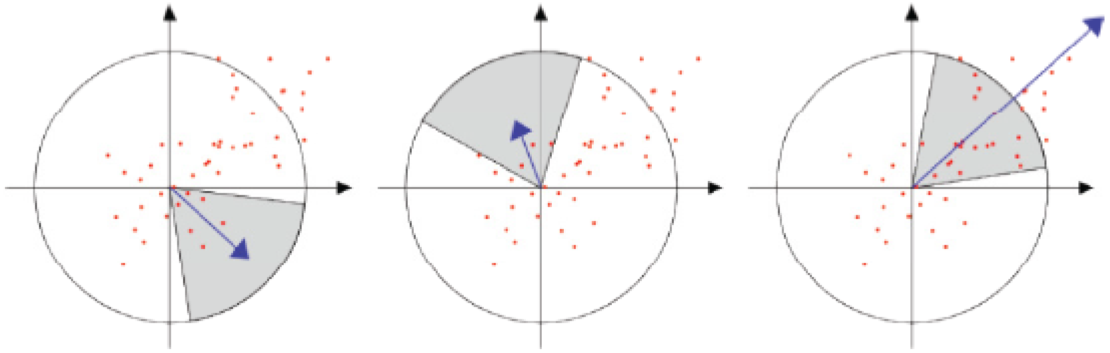
- Interest point localization
  - Non-maximum suppression

Interest Point Description and Matching

- Orientation assignment
  - Calculate the Haar wavelet responses in x and y direction within a circular neighbourhood of radius 6s around the interest point
  - The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window of size $\frac{\pi}{3}$



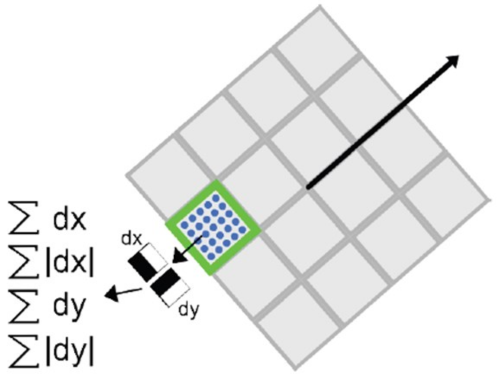
Haar wavelet filters

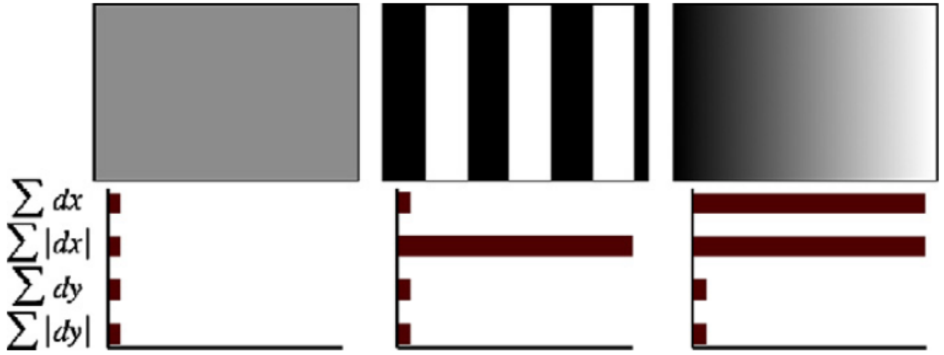


Interest Point Description and Matching

- Descriptor based on sum of Haar wavelet responses

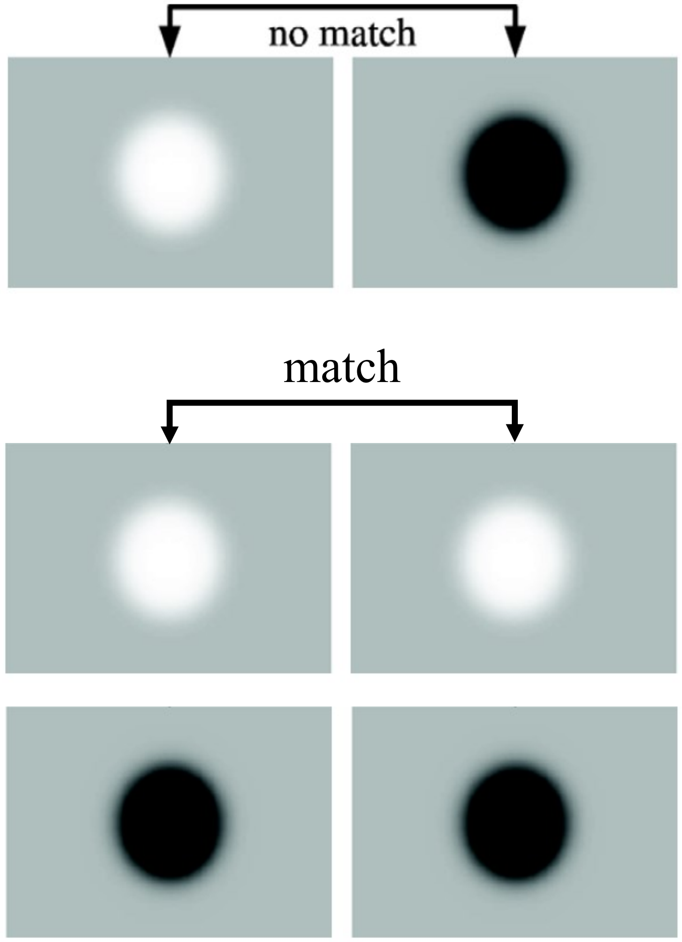
  - $20s \times 20s$ window
  - orient along dominant orientation selected
    rotation invariance
  - split 4×4 square sub-region, sampling 5×5
  - compute Haar wavelet responses
  - $\sum d_x, \sum|d_x|, \sum d_y, \sum|d_y|$
  - descriptor $4 \times 4 \times 4 = 64$-dimensional vector
  - normalization
    illuminance invariance





Interest Point Description and Matching

- Fast indexing for matching
  - Sign of Laplacian (i.e., tr(**H**))
    - ➢ bright blob
    - ➢ dark blob
  - No extra computational cost
  - Faster matching, without reducing the descriptor's performance



References

1. Lowe, David G. "Object recognition from local scale-invariant features." *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee, 1999.
2. Lowe, David G. "Distinctive image features from scale-invariant keypoints." *International journal of computer vision* 60.2 (2004): 91-110.
3. Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." *European conference on computer vision*. Springer, Berlin, Heidelberg, 2006.
4. Bay, Herbert, et al. "Speeded-up robust features (SURF)." *Computer vision and image understanding* 110.3 (2008): 346-359.

Derivative of taylor expansion

$$D(x) = D + \underbrace{\frac{\partial D}{\partial x}}_{\text{벡터}}^{T} x + \frac{1}{2} x^{T} \underbrace{\frac{\partial^2 D}{\partial x^2}}_{\text{행렬(Hessian)}} x$$

(D, x 모두 벡터)

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

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

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

$$\frac{\partial D(x)}{\partial x} = 2\frac{\partial D}{\partial x} + \frac{\partial^2 D}{\partial x^2} x + \frac{1}{2} x^{T}\left(\left(\frac{\partial^2 D}{\partial x^2}\right)^{T} + \frac{\partial^2 D}{\partial x^2}\right) = 0$$

을 만족시키는 x를 $\hat{x}$ 이라 하자.

$$\frac{\partial D(\hat{x})}{\partial x} = 2\frac{\partial D}{\partial x} + \frac{\partial^2 D}{\partial x^2}\hat{x} + \frac{1}{2}\hat{x}^{T}\left(\left(\frac{\partial^2 D}{\partial x^2}\right)^{T} + \frac{\partial^2 D}{\partial x^2}\right) = 0$$

이 때, $\frac{\partial^2 D}{\partial x^2}$은 Hessian Matrix 이다. Hessian은 "대칭행렬" 이므로, $A = A^T$를 만족한다. 그러므로 식은 다음과 같이 쓸 수 있다.

$$2\frac{\partial D}{\partial x} + \frac{\partial^2 D}{\partial x^2}\hat{x} + \frac{1}{2}\hat{x}^{T}\left(\frac{\partial^2 D}{\partial x^2} + \frac{\partial^2 D}{\partial x^2}\right) = 0$$

$$2\frac{\partial D}{\partial x} + \frac{\partial^2 D}{\partial x^2}\hat{x} + \hat{x}^{T}\left(\frac{\partial^2 D}{\partial x^2}\right) = 0$$

또, $\frac{\partial^2 D}{\partial x^2}$은 대칭행렬 이므로, $x^T A = A x$를 만족시킨다.

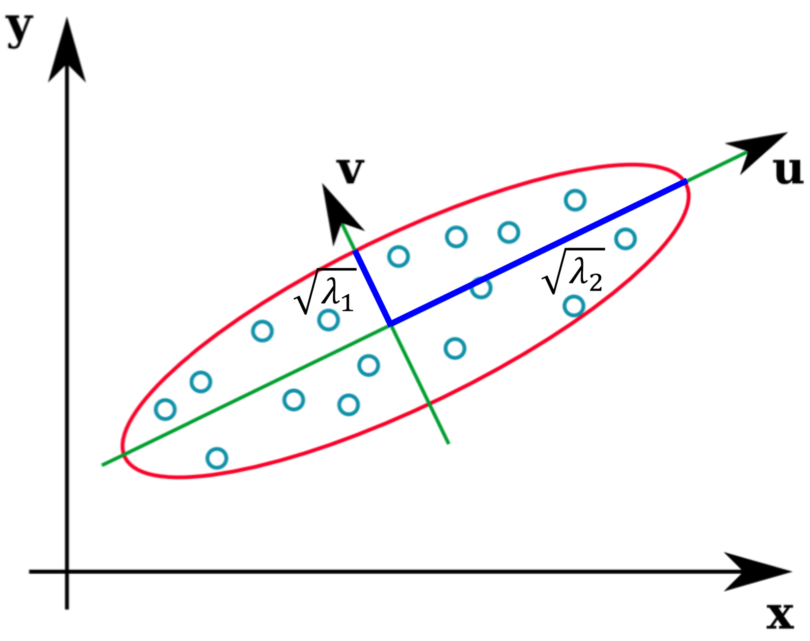
$$2\frac{\partial D}{\partial x} + 2\frac{\partial^2 D}{\partial x^2}\hat{x} = 0$$

$$\frac{\partial D}{\partial x} + \frac{\partial^2 D}{\partial x^2}\hat{x} = 0$$

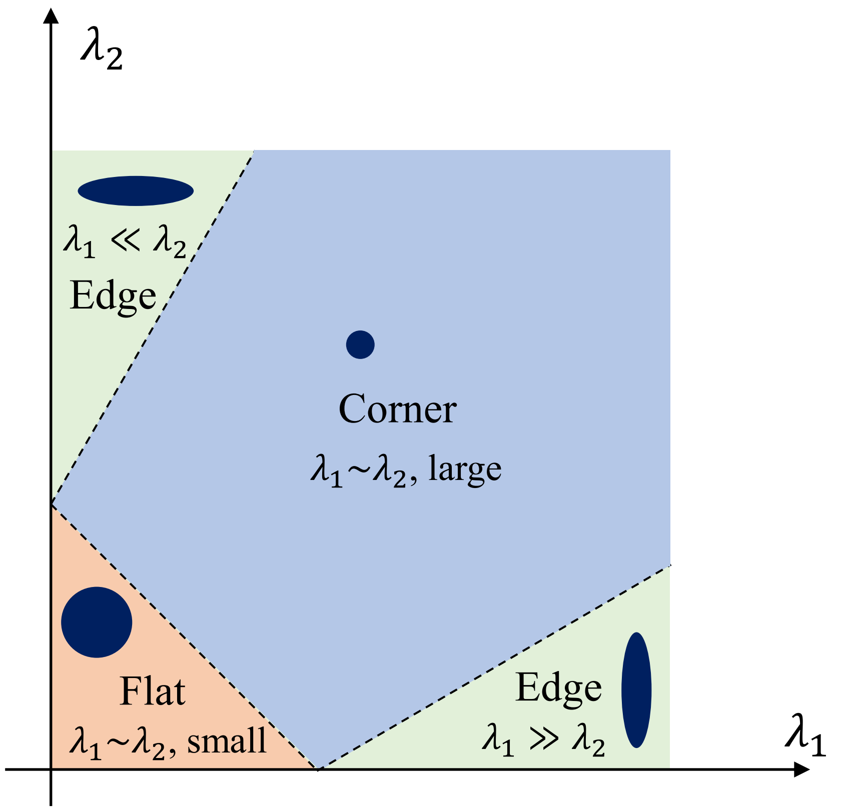
$$\therefore \hat{x} = -\left(\frac{\partial^2 D}{\partial x^2}\right)^{-1}\frac{\partial D}{\partial x}$$

Detection of Corner using the Eigenvalues of H

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} = \mathbf{R}^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \mathbf{R}$$

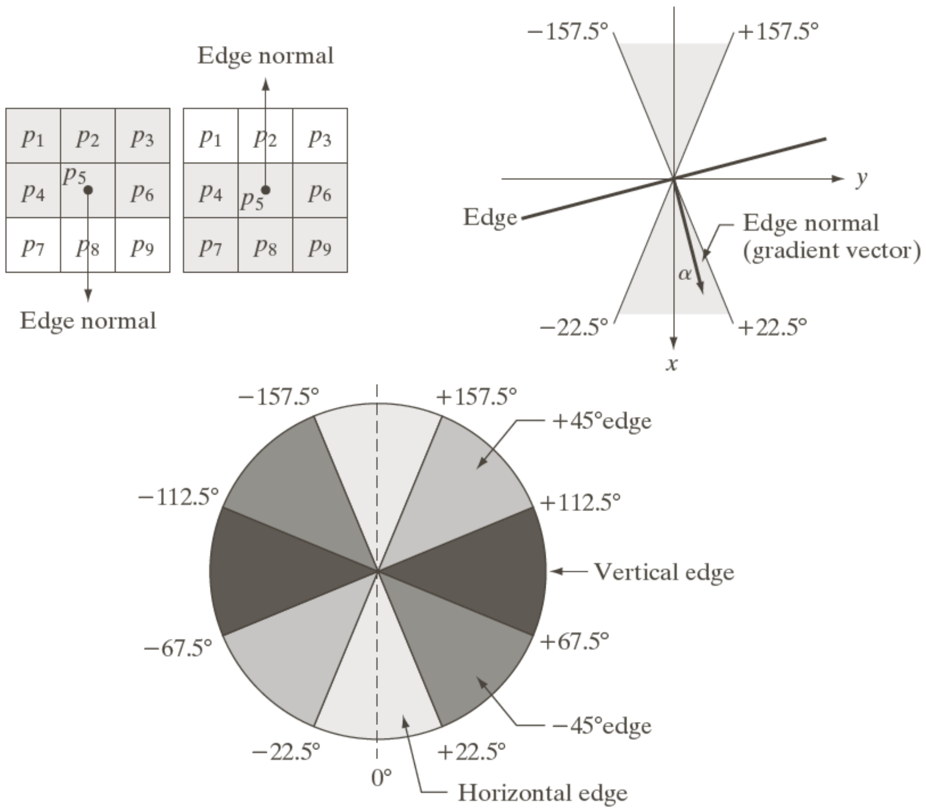


Eigenvalues and Eigenvectors of H



Corner Detector using eigen values

Non-Maximum Suppression



a b
c

FIGURE 10.24
(a) Two possible orientations of a horizontal edge (in gray) in a 3 × 3 neighborhood. (b) Range of values (in gray) of α, the direction angle of the *edge normal*, for a horizontal edge. (c) The angle ranges of the edge normals for the four types of edge directions in a 3 × 3 neighborhood. Each edge direction has two ranges, shown in corresponding shades of gray.

Non-Maximum Suppression

1. Find the direction d_k that is closest to $\alpha(x, y)$

2. if the value of $M(x, y)$ is less than at least one of its two neighbors long d_k, let $g_N(x, y) = 0\ (suppression)$; otherwise, let $g_N(x, y) = M(x, y)$

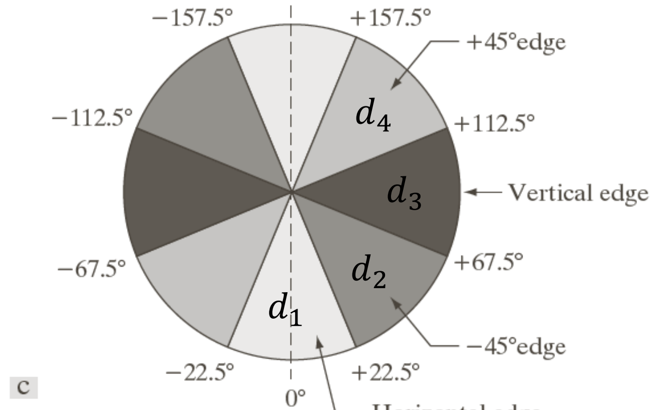
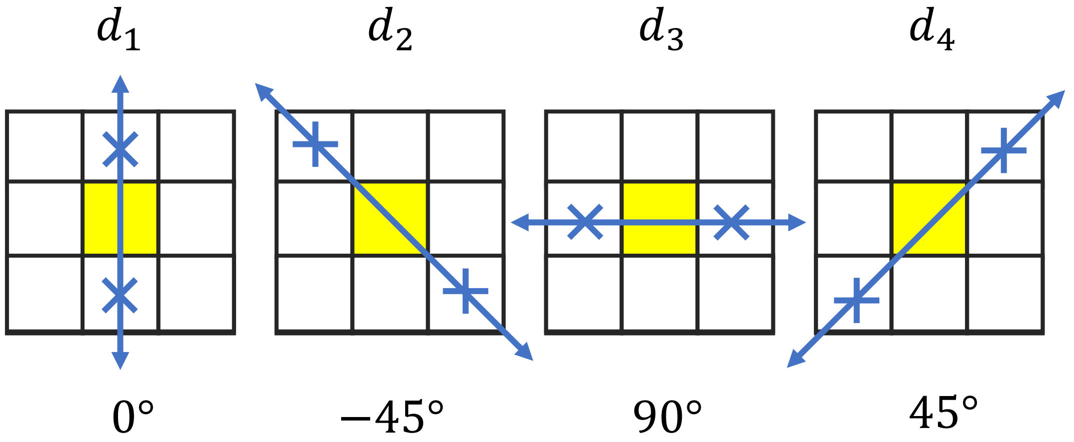
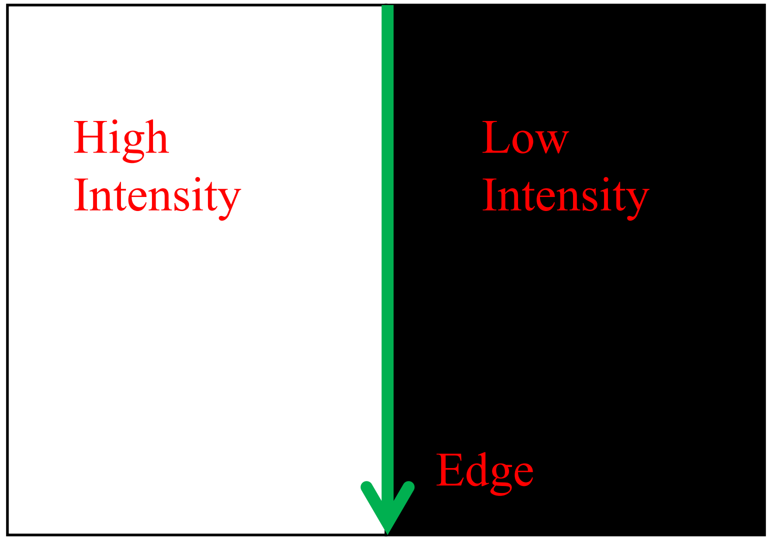


FIGURE 10.24

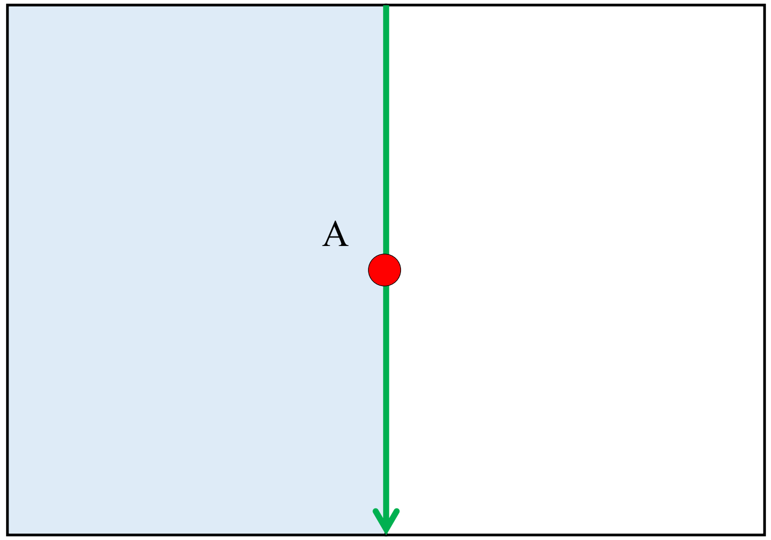


Non-Maximum Suppression

- https://youtu.be/Y_YPlGP4T44

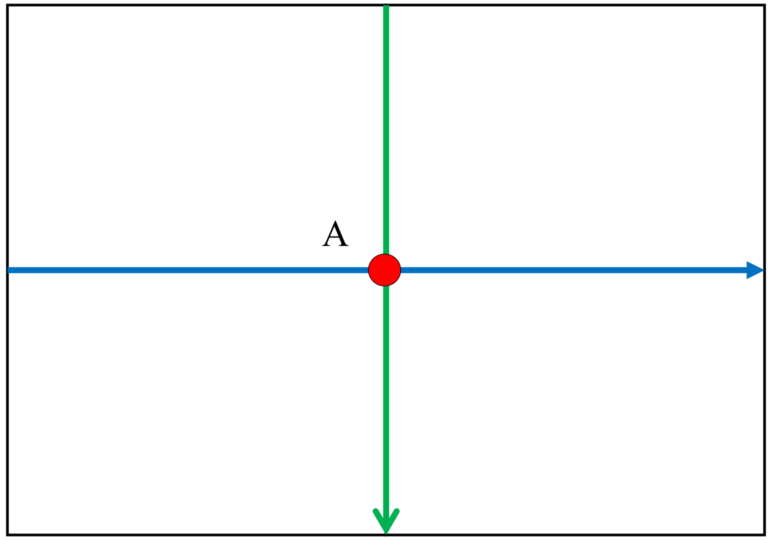


Non-Maximum Suppression



Edge

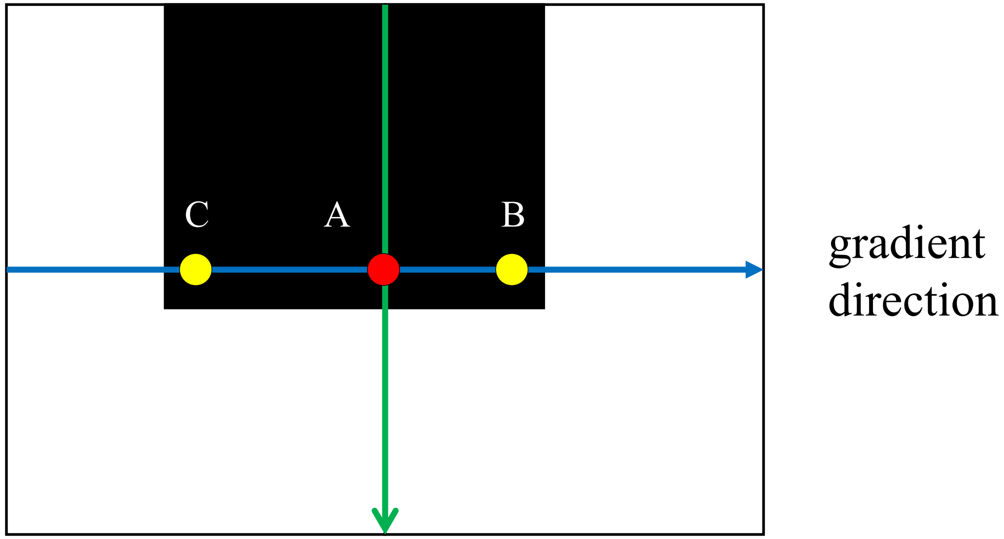
Non-Maximum Suppression



Non-Maximum Suppression

$$if\quad M(A) > M(B)$$
$$M(A) > M(\mathrm{C})$$

$$g_N(A) = M(A)$$
$$g_N(B) = 0$$
$$g_N(C) = 0$$



Non-Maximum Suppression