Feature and Descriptor

Byeongjoon Noh

Dept. of AI and Bigdata, SCH Univ.

powernoh@sch.ac.kr

Contents

Preview

Previous class…

- Edge detection
- Edge: Basic feature in image/video

Correspondence problem (대응점 찾기)

- 같은 장면을 다른 시점에서 찍은 두 영상에서 대응하는 점의 쌍을 찾는 문제
- 물체 인식, 추적 등 컴퓨터 비전의 중요한 문제 해결의 단조

- **검출/추출 → 기술(묘사) → 매칭**

1. Local feature

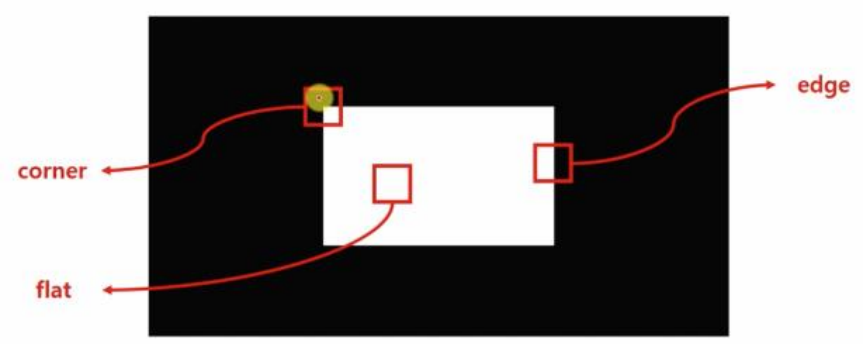
What is keypoint (feature) in image?

Edge

- Edge 강도와 방향 정보 뿐 ➔ matching에 참여하기에 턱없이 부족

대안

- 다른 곳과 두드러지게 달라 풍부한 정보 추출이 가능한 곳



- Corner (코너)
  - Corner: Edge 토막에서 곡률이 큰 지점
    - 1980년대… ➔ 2000년대 사라짐

- **Local feature (지역 특징)**

Local feature

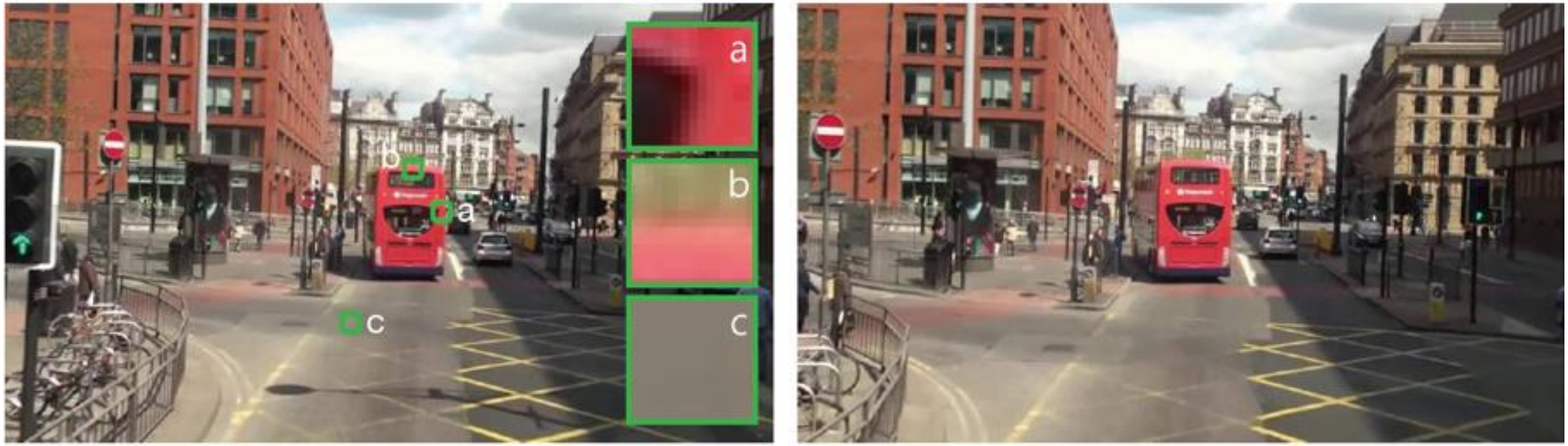
Idea

- 명암 영상에서 “직접” 검출
- 의식 전환 및 새로운 발상
  - Corner의 물리적 의미 ➔ 반복성(Matching 시에..) 강조
  - 실제 물체의 모퉁이에 정보가 위치 해야한다? ➔ No!
  - 좁은 지역을 보고 특징점 여부 결정

Local feature

간단한 인지 실험

- 왼쪽 세 특징 영상(a, b, c) 중 어느 것이 가장 오른쪽 영상에 matching하기 쉬운가?
- a > b> c



- ➔ 사람에게 쉬운 곳이 컴퓨터에게도 쉽지 않을까? ➔ 좋은 정도를 어떻게 “수량화” 할까?

Local feature

Local feature 표현 방법

- <위치, 스케일, 방향, 특징 벡터> = $((y, x), s, \theta, \mathbf{x})$
  - Non-italic character $\mathbf{x}$: vector
  - Detection stage ➔ $(y, x), s$ 정보 획득
  - Description stage ➔ $\theta, \mathbf{x}$ 정보 획득

Local feature가 만족해야 할 특성

- Repeatability, distinctiveness (분별력), locality (지역성)

  accuracy, cost-effectiveness
- ➔ 서로 trade-off이므로 활용 방안에 따라 적절하게 취사 선택 해야함

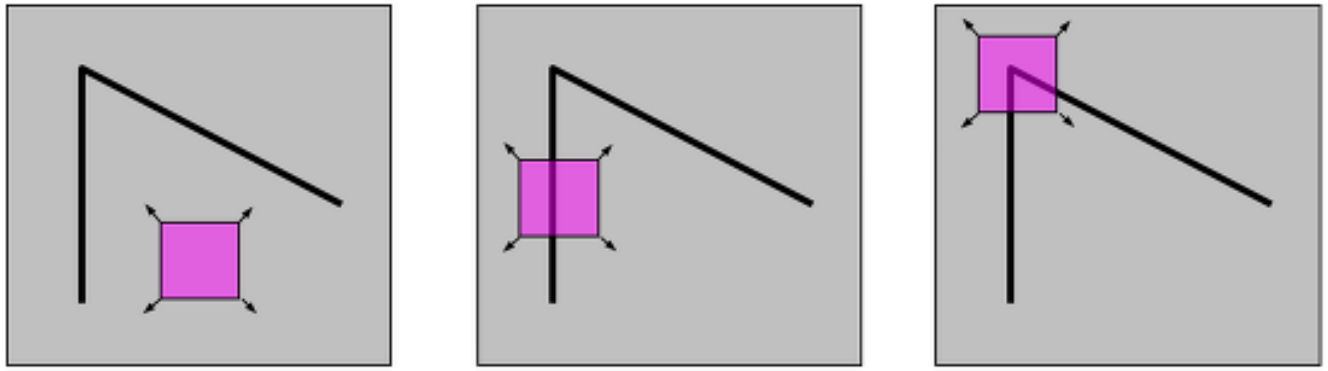
How to obtain this local feature?

Movement and rotation invariant local feature (위치) 검출

- Moravec (모라벡) algorithm
- Harris corner detection (해리스 코너검출)

Scale invariant local feature (위치) 검출

- SIFT
- SURF

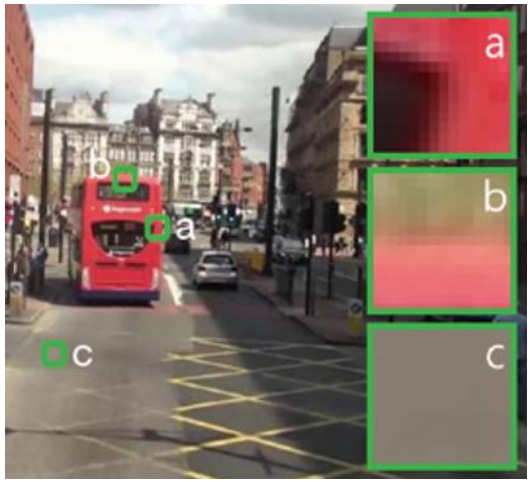


"flat" region: no change in all directions

"edge": no change along the edge direction

"corner": significant change in all directions

Moravec algorithm



Idea

- 인지 실험에 주목
  - a는 여러 방향으로 색상 변화가 있음
  - c는 어느 방향으로도 미세한 변화만 있어 찾기가 어려움

- 여러 방향에 대한 색상 변화를 측정(수량화)하는 방법 제안
  - 제곱차의 합으로 밝기 변화 측정

$$S(v,u) = \sum_{y}\sum_{x}\left(f(y+v,x+u) - f(y,x)\right)^2$$

Moravec algorithm

Example

- 다음 영상에서 a, b, c에 대해 제곱차 계산
  - b에 대한 S맵 (S_b) 계산
    - $S(0,1) = \sum_{3\le y\le 5}\sum_{2\le x\le 4}\big(f(y,x+1) - f(y,x)\big)^2 = 4$

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	c 0	0	0
2	0	0	0	1	0	0	0	0	0	0
3	0	0	0	1	1	0	0	0	0	0
4	0	0	0	b 1	1	1	0	0	0	0
5	0	0	0	1	1	1	1	0	0	0
6	0	0	0	1	1	1	1	a 1	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

a

v \ u	-1	0	1
-1	3	4	4
0	2	0	2
1	4	3	2

b

v \ u	-1	0	1
-1	3	1	6
0	3	0	4
1	3	0	3

c

v \ u	-1	0	1
-1	0	0	0
0	0	0	0
1	0	0	0

Moravec algorithm

Example

- 자세한 계산 과정

$$S(0,1) = \sum_{3\le y\le 5}\sum_{2\le x\le 4}\left(f(y,x+1)-f(y,x)\right)^2$$

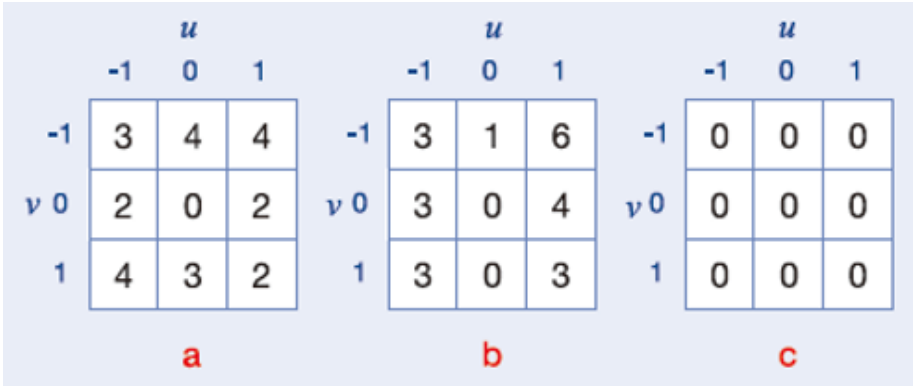
$$= \sum_{3\le y\le 5}\left(\left(f(y,3)-f(y,2)\right)^2+\left(f(y,4)-f(y,3)\right)^2+\left(f(y,5)-f(y,4)\right)^2\right)$$

$$= \left(\left(f(3,3)_{1}-f(3,2)_{0}\right)^2+\left(f(3,4)_{1}-f(3,3)_{1}\right)^2+\left(f(3,5)_{0}-f(3,4)_{1}\right)^2\right)$$

$$+\left(\left(f(4,3)_{1}-f(4,2)_{0}\right)^2+\left(f(4,4)_{1}-f(4,3)_{1}\right)^2+\left(f(4,5)_{1}-f(4,4)_{1}\right)^2\right)$$

$$+\left(\left(f(5,3)_{1}-f(5,2)_{0}\right)^2+\left(f(5,4)_{1}-f(5,3)_{1}\right)^2+\left(f(5,5)_{1}-f(5,4)_{1}\right)^2\right)=4$$

Moravec algorithm



의미 및 의도

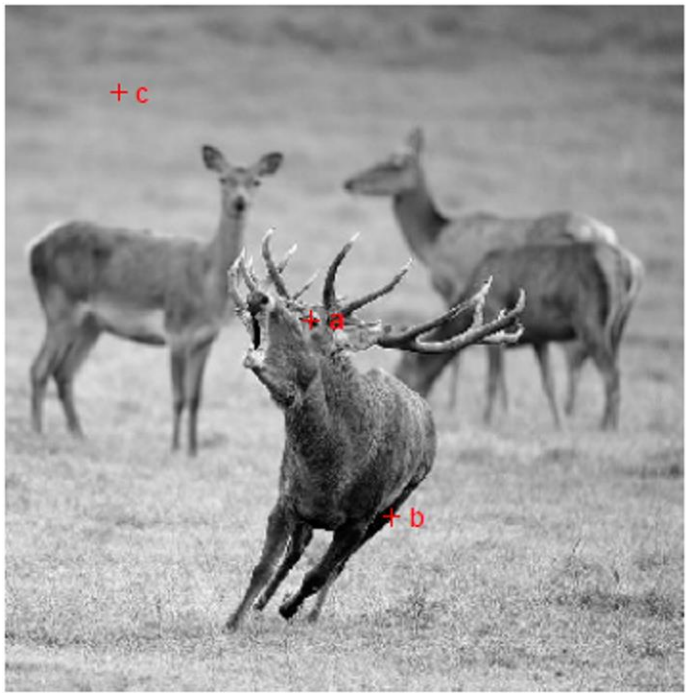
- S_a 해석
  - 정방향으로 값의 변화가 뚜렷 ➔ Local feature

- S_b 해석
  - 수평 방향에 대한 변화 ➔ S_b의 좌우로 큰 값
  - Edge일 가능성 높음

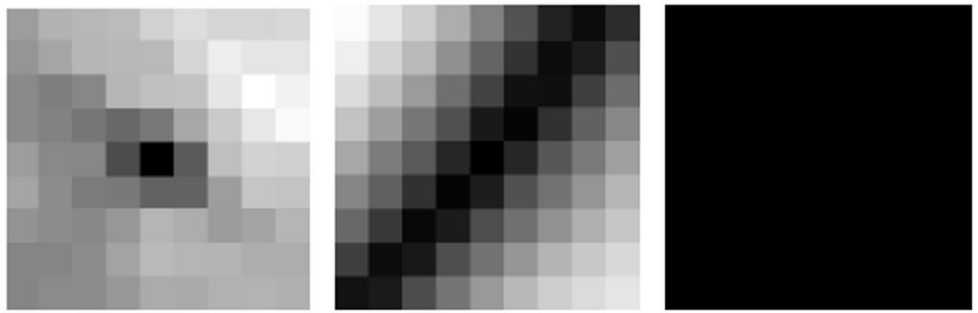
- S_c 해석
  - 값 변화 없음 ➔ 아무런 특징 X

Moravec algorithm

실 적용 예시



> 원래 영상



> a

> b

> c

Moravec algorithm

Moravec function

- 특징 가능성 값 C

$$C = \min(S(0,1), S(0,-1), S(1,0), S(-1,0))$$

- (이전 예제에서) $C(S_a) = 2$, $C(S_b) = 0$, $C(S_c) = 0$

Harris Corner detection

Idea

- Moravec: Local feature에 대한 개념 정의는 좋았으나 현실적용에 어려움
  - 현실 세계는 상하좌우만으로 해결되지 않음 ➔ 28도, 34도 등 정방향 회전이 아닌 경우
  - ➔ Harris: 가중치를 두면 어떨까?

- “가중치” 제곱 차의 합 (Weighted sum of squared difference, WSSD)를 이용한 잡음 대처
  - 가중치 = Gaussian mask
  - 현재까지 많이 활용함

$$S(v,u) = \sum_{y}\sum_{x} G(y,x)\big(f(y+v,x+u) - f(y,x)\big)^2$$

Harris Corner detection

Idea

- 모든 방향을 고려한 isotropic property (등방성)을 만족시켜야함

  ➔ 1차 미분, Taylor expansion 활용

- Taylor expansion
  - (y, x)로부터 작은 양 (v, u)만큼 이동한 점 $(v + y, u + x)$의 함수값 $f(v + y, u + x)$는 (y, x)에서 함수값 $f(y, x)$에 두 방향으로의 증가량인 $vd_y(y, x)$, $ud_x(y, x)$를 더하여 근사시킬 수 있음

$$f(v + y, u + x) \cong f(y, x) + vd_y(y, x) + ud_x(y, x)$$

$$d_y(y, x) = \frac{\partial f}{\partial y}, d_x(y, x) = \frac{\partial f}{\partial x}$$

Harris Corner detection

Idea

- 전개

$$S(v,u) \cong \sum_y \sum_x G(y,x)\left(v d_y(y,x) - u d_x(y,x)\right)^2 = \sum_y \sum_x G(y,x)\left(v^2 d_y^2 + 2vu d_y d_x + u^2 d_x^2\right)$$

$$= \sum_y \sum_x G(y,x)(v \quad u)\begin{pmatrix} d_y^2 & d_y d_x \\ d_y d_x & d_x^2 \end{pmatrix}\begin{pmatrix} v \\ u \end{pmatrix} = (v \quad u)\sum_y \sum_x G(y,x)\begin{pmatrix} d_y^2 & d_y d_x \\ d_y d_x & d_x^2 \end{pmatrix}\begin{pmatrix} v \\ u \end{pmatrix}$$

$$= (v \quad u)\begin{pmatrix} \sum_y \sum_x G(y,x) d_y^2 & \sum_y \sum_x G(y,x) d_y d_x \\ \sum_y \sum_x G(y,x) d_y d_x & \sum_y \sum_x G(y,x) d_x^2 \end{pmatrix}\begin{pmatrix} v \\ u \end{pmatrix} = \mathbf{uAu^T}$$

Harris Corner detection

2차 moment matrix

$$\mathbf{A} = \begin{pmatrix} G \odot d_y^2 & G \odot d_y d_x \\ G \odot d_y d_x & G \odot d_x^2 \end{pmatrix}$$

- $\mathbf{A}$: 2차 moment matrix 또는 auto-correlation
- $\mathbf{A}$ 는 화소 주위의 영상 구조를 표현 함
- $\mathbf{A}$ 의 분석 만으로 local feature 여부를 알 수 있음 (특징 가능성 계산)
  - ➔ (v, u)를 변화시키면서 맵을 생성할 필요가 없음
- (v, u): 실수 가능
  - 이전 예제에서는 $f(v + 0.2, u + 0.5)$에 대한 계산 불가능
  - 2차 moment matrix를 활용하면 가능함

Harris Corner detection

Example

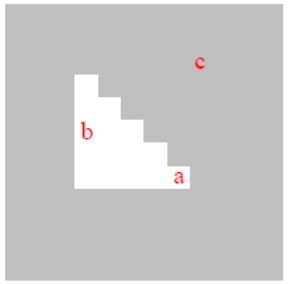
- 예시 영상에서 2차 모먼트 행렬 계산 과정

- 필요한 정보: $d_y, d_x, d_y^2, d_x^2, d_y d_x, G$
  - $\sigma = 1.0$ 일 때 Gaussian mask

$$G = \begin{bmatrix} .0751 & .1238 & .0751 \\ .1238 & .2042 & .1238 \\ .0751 & .1238 & .0751 \end{bmatrix}$$

- $d_y = [-1\ 0\ 1], d_x = [-1\ 0\ 1]^{\mathrm{T}}$

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	c 0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0
4	0	0	0	1	1	0	0	0	0	0	0	0
5	0	0	0	b 1	1	1	0	0	0	0	0	0
6	0	0	0	1	1	1	1	0	0	0	0	0
7	0	0	0	1	1	1	1	a 1	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0



(a) 원래 영상 f

$> d_y$ $> d_x$ $> d_y^2$ $> d_x^2$ $> d_y d_x$

(b) 도함수 영상(흰색은 1, 회색은 0, 검은색은 −1)

Harris Corner detection

Example

- 예시 영상에서 2차 모먼트 행렬 계산 과정

0	0	0	0	0	0	0	0	0	0	0	0
0	0	.075	.124	.075	0	0	0	0	0	0	0
0	0	.199	.403	.323	.075	0	0	0	0	0	0
0	0	.199	.527	.602	.323	.075	0	0	0	0	0
0	0	.075	.323	.602	.602	.323	.075	0	0	0	0
0	0	0	.075	.323	.602	.602	.323	.075	0	0	0
0	0	.075	.199	.349	.597	.726	.478	.124	0	0	0
0	0	.199	.527	.726	.801	.801	.522	.150	0	0	0
0	0	.199	.527	.726	.726	.651	.403	.124	0	0	0
0	0	.075	.199	.274	.274	.274	.199	.075	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

(c) $G \circledast d_y^2$

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	.075	.124	.150	.124	.075	0	0	0	0	0	0
0	.199	.403	.521	.478	.323	.075	0	0	0	0	0
0	.274	.651	.801	.726	.602	.323	.075	0	0	0	0
0	.274	.726	.801	.597	.602	.602	.323	.075	0	0	0
0	.274	.726	.726	.349	.323	.602	.602	.323	.075	0	0
0	.199	.527	.527	.199	.075	.323	.527	.403	.124	0	0
0	.075	.199	.199	.075	0	.075	.199	.199	.075	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

(d) $G \circledast d_x^2$

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	-.075	-.124	-.075	0	0	0	0	0	0
0	0	0	-.199	-.403	-.323	-.075	0	0	0	0	0
0	0	0	-.199	-.527	-.602	-.323	-.075	0	0	0	0
0	0	0	-.075	-.323	-.602	-.602	-.323	-.075	0	0	0
0	0	-.075	-.124	-.150	-.323	-.527	-.403	-.124	0	0	0
0	0	-.075	-.204	-.124	-.075	-.199	-.199	-.075	0	0	0
0	0	-.075	-.124	-.075	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

(e) $G \circledast d_y d_x$

Harris Corner detection

2차 moment matrix 분석

- 특징 가능성 계산
  - 고유값 (Eigen value) 활용 ➔ 특징 가능성을 수치화
  - $C = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$
    - λ_1, λ_2 : $\mathbf{A}$ 의 고윳값

  - Eigen value가 0에 근사 ➔ 변화가 거의 없음
  - 두 eigen value가 모두 크면 ➔ 여러 방향으로 변화가 있음 (특징점에 적합)

	a	b	c
2차 모멘트 행렬	$\mathbf{A} = \begin{pmatrix} 0.52 & -0.2 \\ -0.2 & 0.53 \end{pmatrix}$	$\mathbf{A} = \begin{pmatrix} 0.08 & -0.08 \\ -0.08 & 0.8 \end{pmatrix}$	$\mathbf{A} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$
고윳값	λ_1=0.72, λ_2=0.33	λ_1=0.81, λ_2=0.07	λ_1=0.0, λ_2=0.0
특징 가능성 값	C=0.1925	C=0.0237	C=0.0

Note.

Eigen vector / Eigen value

- Definition: 행렬 A가 주어질 때, 다음을 만족하는 λ를 고윳값, $\mathbf{x}$를 고유벡터

$$A\mathbf{x} = \lambda\mathbf{x} \rightarrow (A - \lambda)\mathbf{x} = 0$$

- 2x2 행렬 A 에 대한 예시

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$T = a + d, D = ad - bc$$

$$\lambda_1, \lambda_2 = \frac{T}{2} \pm \left(\frac{T^2}{4} - D\right)^{0.5}$$

$$\mathbf{x} = \begin{pmatrix} b \\ \lambda_1 - a \end{pmatrix}, \begin{pmatrix} b \\ \lambda_2 - a \end{pmatrix} \qquad \mathbf{x} = \begin{pmatrix} \lambda_1 - d \\ c \end{pmatrix}, \begin{pmatrix} \lambda_2 - d \\ c \end{pmatrix} \qquad \mathbf{x} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$c \neq 0 \qquad b \neq 0 \qquad b = 0, c = 0$$

Harris Corner detection

2차 moment matrix 분석

- Eigen value 계산 ➔ high computational power

- $A = \begin{pmatrix} p & r \\ r & q \end{pmatrix}$ ➔ $C = (pq - r^2) - k(p+q)^2$

k: 경험적 상수

  - Eigen value 계산을 피해 빠른 속도로 특징 가능성 검출

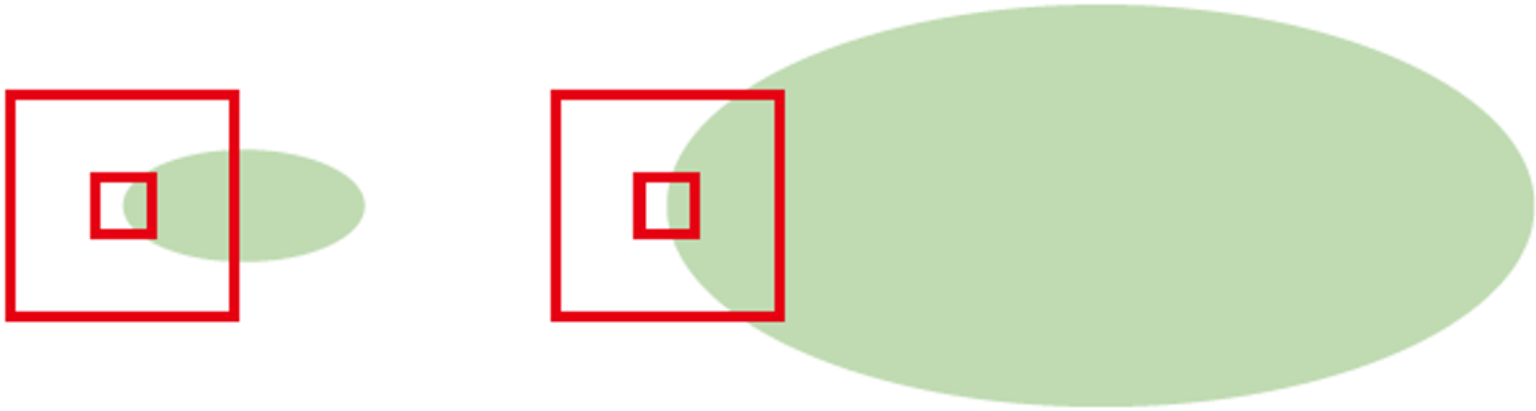
← C>0.02인 점을 검출

Harris Corner detection

Harris 특징점

- 물체의 실제 모퉁이 뿐 아니라 blob에서도 검출 가능
- “Corner” (모퉁이) ➔ Feature point, keypoint, interest point 등으로 부름

- 이동과 회전에 불변
- Scale에는 불변이 아님
  - 물체의 크기에 따라 mask의 크기를 적절하게 선택해주어야 함



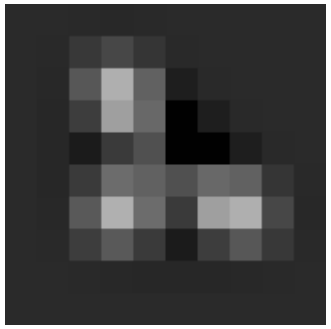
Harris Corner detection

Harris Corner detection 알고리즘

- 3_1_HarrisCorner.py

```
image = np.array([[0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
                  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
                  [0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
                  [0, 0, 0, 1, 1, 0, 0, 0, 0, 0],
                  [0, 0, 0, 1, 1, 1, 0, 0, 0, 0],
                  [0, 0, 0, 1, 1, 1, 1, 0, 0, 0],
                  [0, 0, 0, 1, 1, 1, 1, 1, 0, 0],
                  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
                  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
                  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]],
                  dtype=np.float32)
```

```
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 9. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 1. 1. 0. 0. 0. 0.]
 [0. 0. 0. 1. 1. 1. 1. 0. 0. 0.]
 [0. 0. 0. 9. 1. 1. 1. 9. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```



기타 특징점 검출 알고리즘

Moravec algorithm

Harris corner (feature) detection

Hessian matrix

$$C = \det(A) - k \times trace(A)^2 = (pq - r^2) - k(p+q)^2$$

LOG

$$C = \nabla^2 = trace(\mathbf{H}) = d_{yy}(\sigma) + d_{xx}(\sigma)$$

SUSAN (Small Uni-value Segment Assimilating Nucleus, 슈산)

$$C = \begin{cases} q - usan\_area(r_o), & usan\_area(r_o) \le t \\ 0, & else \end{cases}$$

Scale invariant local feature

사람은 scale 불변의 특징 사용

- 거리에 상관없이 같은 물체를 같은 물체라고 인식
- 거리에 따라 세세한 내용의 차이만 있음

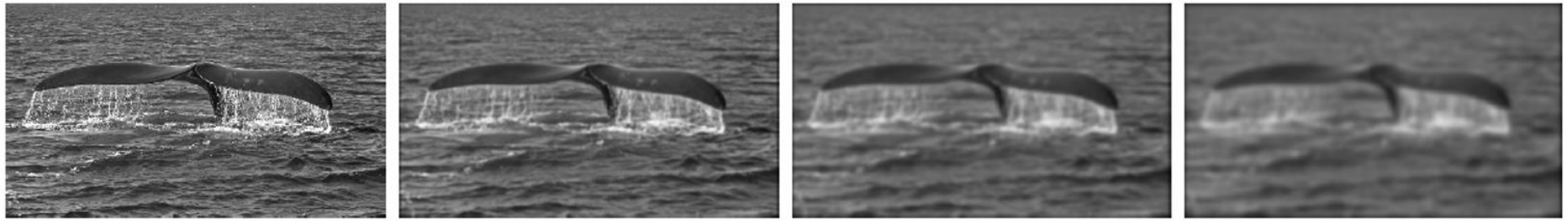
Scale space theory (스케일 공간 이론)

- 스케일 공간에서 특징점 검출 ➔ 스케일에 관계없이 특징점을 검출 할 수 있음
- 전략
  - 1. 입력 영상 f로 부터 **다중 스케일 영상** $\tilde{f}$를 구성
  - 2. $\tilde{f}$에 적절한 미분 연산을 적용하여 다중 스케일 미분영상 $\tilde{f}'$를 계산
  - 3. $\tilde{f}'$에서 극점을 찾아 특징점으로 취함

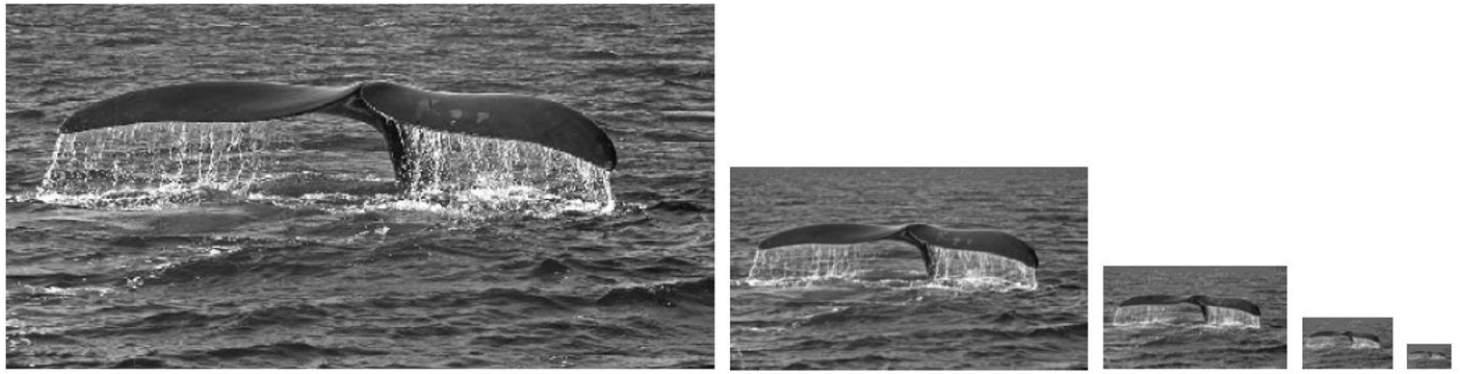
Scale space

다중 스케일 영상 구현 방법

- 입력 영상은 단 한 장 ➔ 여러 거리에서 보았을 때 나타나는 현상과 유사하게 구성
- Gaussian smoothing ➔ 스케일에 해당하는 σ가 연속 공간에 정의되어야 함



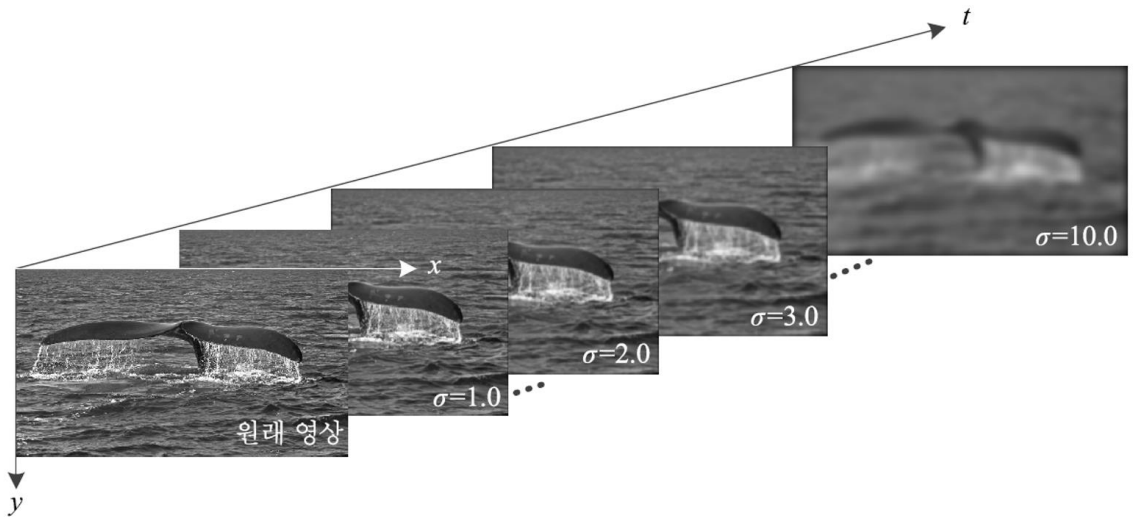
- Pyramid 방법
  - ½ 씩 줄어듦 ➔ 이산적이라는 단점 존재



Scale space

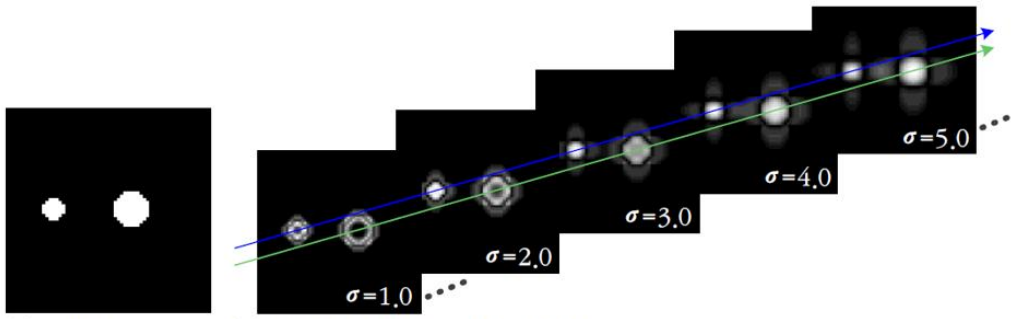
다중 스케일 영상 구현 방법

- Scale축을 추가한 3차원 공간
  - (y, x, t)로 표현

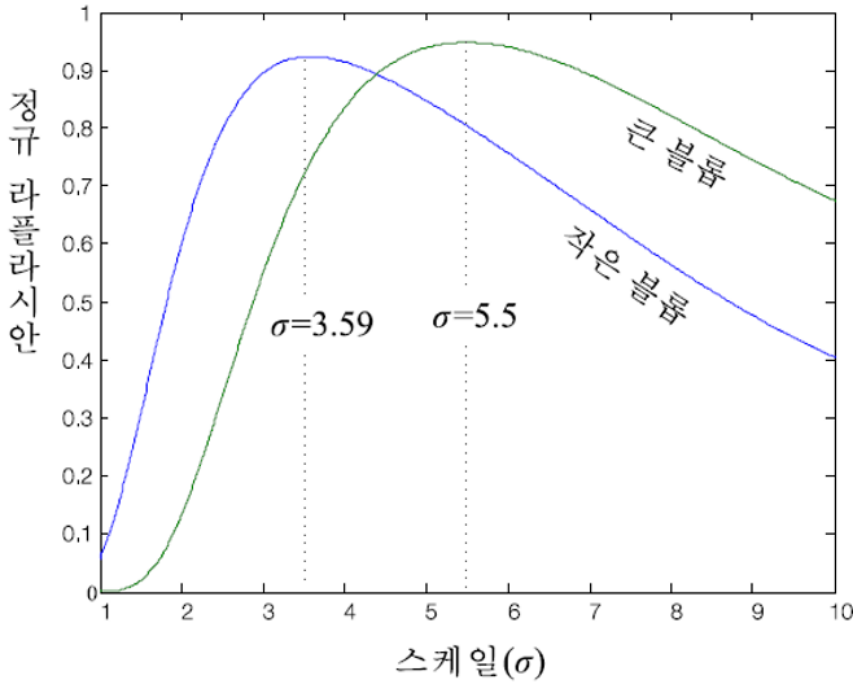


스케일 공간에서의 미분 ➔ 정규 Laplacian 활용

- Laplacian: $\nabla^2 f = d_{yy} + d_{xx}$
- 정규 Laplacian: $\nabla^2_{norm} f = \sigma^2 \left| d_{yy} + d_{xx} \right|$



(a) 원래 영상 (b) 스케일 공간에서 정규 라플라시안 영상



(c) 스케일 축에서 극점

SIFT 검출

SIFT의 등장

- Scale-Invariant Feature Transform
- 1999년 David Lowe 교수의 논문
- 2004년 IJCV에 확장된 논문 발표
- 성능이 매우 뛰어남 ➔ 현재에도 널리 사용, 다양한 변형

Distinctive image features from scale-invariant keypoints
DG Lowe - International journal of computer vision, 2004 - Springer
Abstract This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide ...
25413회 인용 관련 학술자료 전체 247개의 버전 Web of Science: 8507 인용 저장

← Google scholar

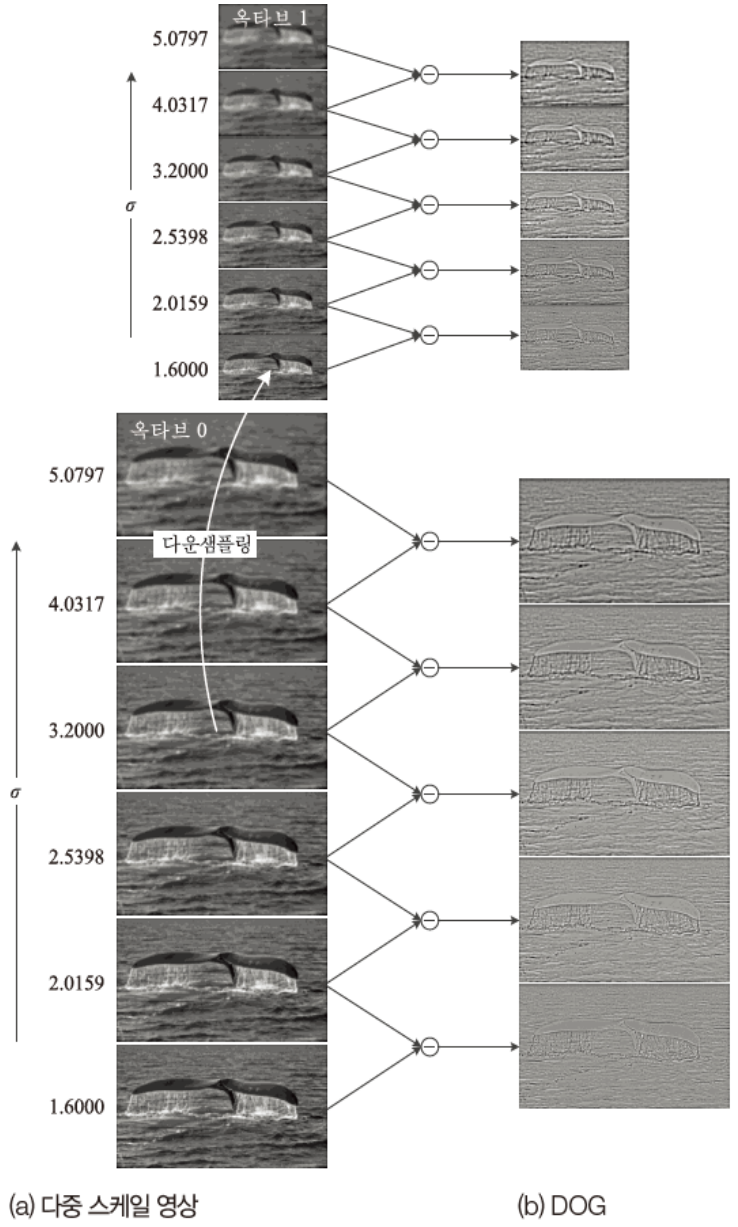
SIFT 검출

SIFT 검출 방법

- 1. 다중 스케일 영상 구축 (SIFT의 스케일 공간)
  - Gaussian smoothing과 Pyramid 공간을 결합
  - 각 층은 여섯 영상의 묶음(Octave)으로 구성
    - Octave의 영상은 σ_i로 smoothing
    - $\sigma_{i+1} = k\sigma_i$

- 2. 다중 스케일 영상에 미분 적용
  - $\nabla^2_{norm} f$ 계산
    - ➔ 계산 효율화를 위하여

      Difference of Gaussian (DoG) 사용
    - 한 영상에 대하여 이웃한 Gaussian에 대한 연산만 수행 (빠름)



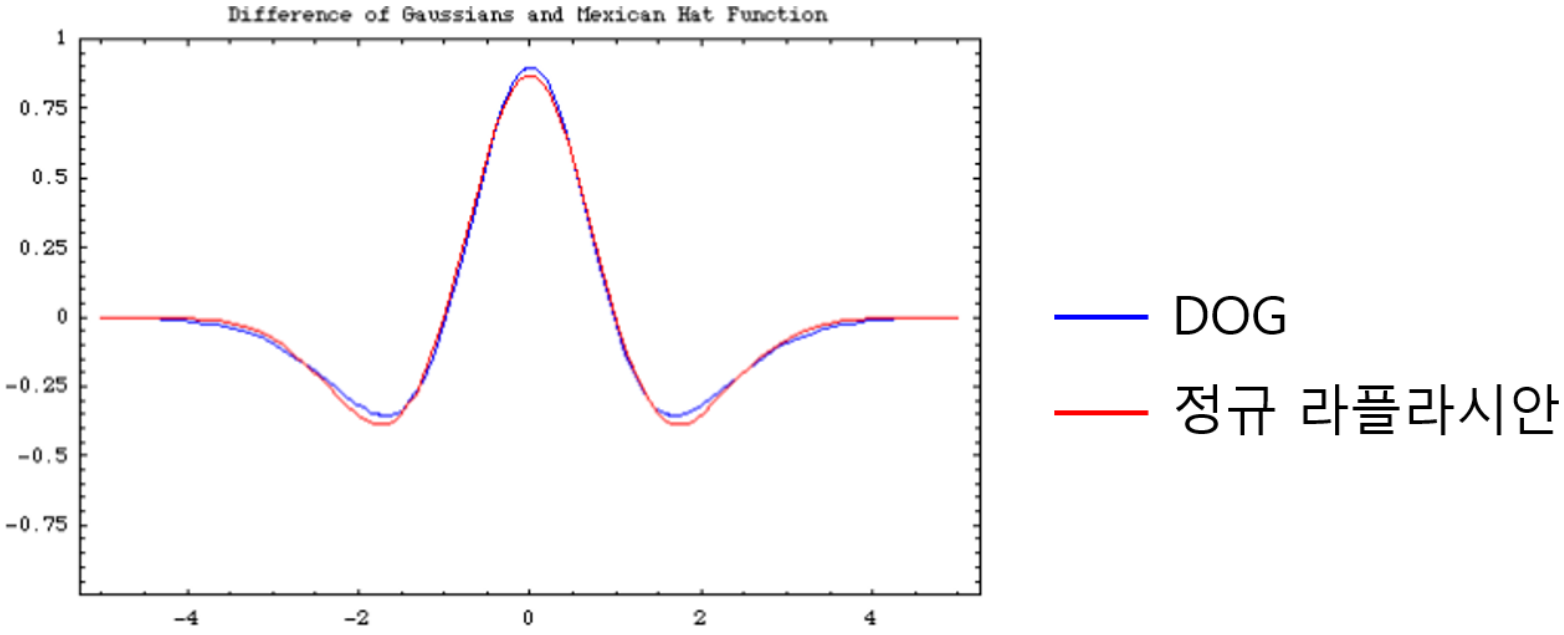
SIFT 검출

SIFT 검출 방법

- 2. 다중 스케일 영상에 미분 적용
  - DoG

$$DOG(\sigma_i) = G(\sigma_{i+1}) \odot f - G(\sigma_i) \odot f$$

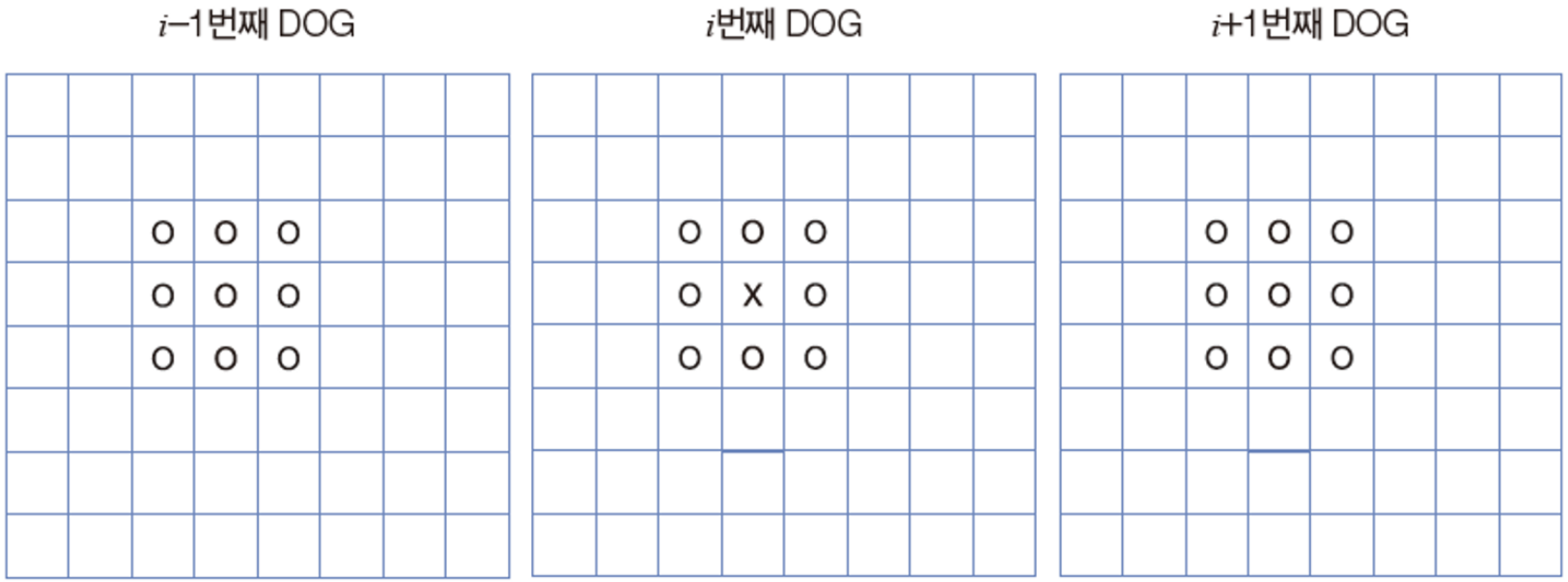
$$= G(k\sigma_i) \odot f - G(\sigma_i) \odot f = \big(G(k\sigma_i) - G(\sigma_i)\big) \odot f$$

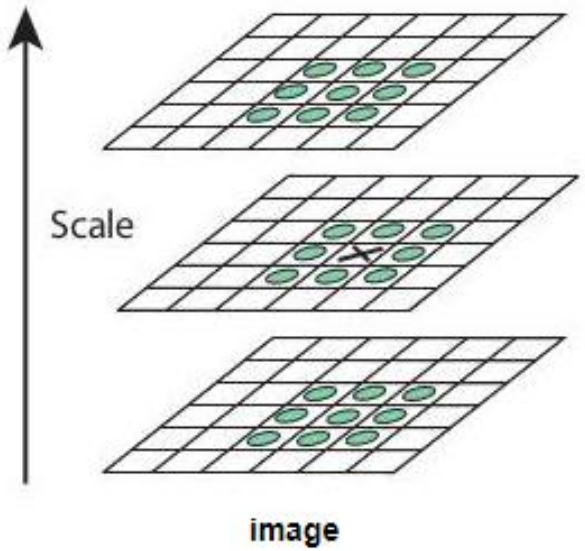


SIFT 검출

SIFT 검출 방법

- 3. 미분한 다중 스케일 영상에서 극점 검출
  - 극점 == 특징점
  - 3차원에서 NMS 적용 (주위 26개 이웃에서 max or min인 지점)



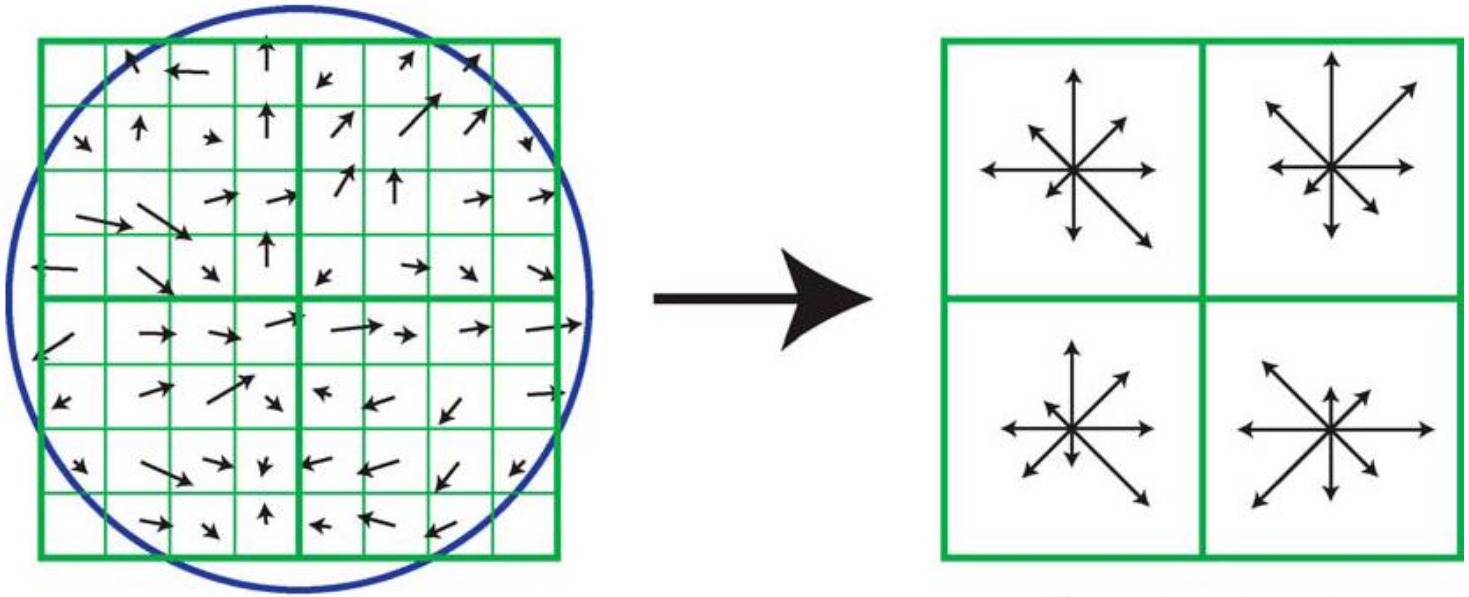


- 특징점 표현: (y, x, o, i); o: octave, i: scale level

SIFT Descriptor

Descriptor?

- 기술자
- 특징점 (특징 벡터)에 대한 정보를 기술자 라고 함
- 좋은 특징 검출 알고리즘 ➔ 좋은 기술자를 통해 매칭 등에 활용함
- Descriptor의 불변성: 영상의 회전, 크기, 광도 등의 변환 시에도 불변해야 함



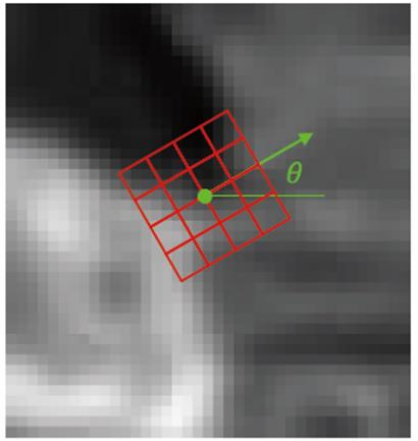
SIFT Descriptor

특징점 주위를 살펴 풍부한 정보를 가진 descriptor 추출

- 불변성 달성이 SIFT의 핵심

Descriptor 추출 알고리즘

- o와 i로 가장 가까운 Gaussian을 결정한 후 descriptor 추출 ➔ scale 불변성 달성
- 기준 방향을 지정한 후 기준 방향을 중심으로 특징 추출 ➔ 회전 불변성 달성
- Descriptor의 단위 벡터화 ➔ 조명 불변성 달성



(a) 지배적인 방향의 윈도우

(b) 16×16 부분 영역 샘플링과 기술자 추출

SIFT Descriptor

SIFT 예시

- 3_2_SIFT.py

```
sift = cv2.SIFT_create()

kp, des = sift.detectAndCompute(gray, None)

gray = cv2.drawKeypoints(gray, kp, None,
                         flags = cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
```

- 참고: sift.detect(), sift.comput() 따로 사용 가능

SURF 검출

SURF

- Speeded-Up Robust Feature
- 반복률 희생 없이 SIFT보다 빠른 알고리즘 추구
- Hessian matrix 활용

$$C = \det(\mathbf{H}) = d_{yy}(\sigma) + d_{xx}(\sigma) - d_{yx}(\sigma)^2$$

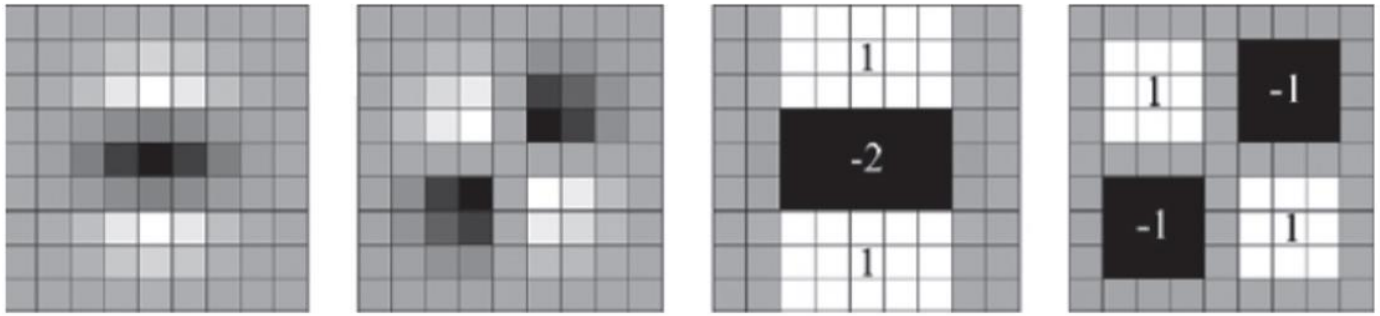
$$d_{yy}(\sigma) = \frac{\partial^2}{\partial y^2}(G(\sigma) \odot f) = \left(\frac{\partial^2}{\partial y^2} G(\sigma)\right) \odot f$$

- 빠른 행렬식 계산을 위해

SURF 검출

SURF

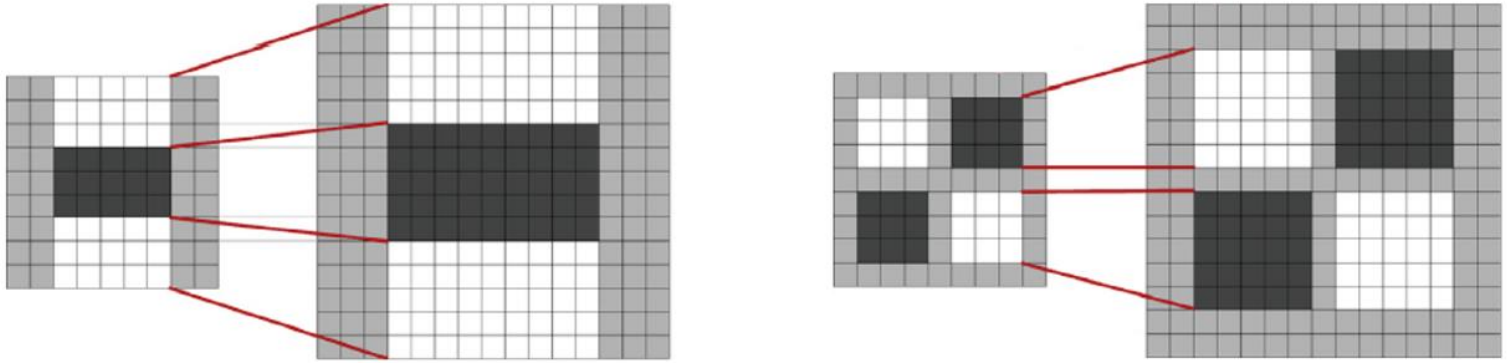
- 빠른 행렬식 계산을 위해 d_{yy}, d_{xx}, d_{yx}를 9x9 mask로 근사 계산
  - Mask 계산은 적분 영상 (Integral image) 이용
  - Gaussian 2차 미분 연산자 (회색 = 0)



SURF 검출

SURF

- SURF의 스케일 공간
  - 원본 영상은 그대로 둔 채 다중 스케일 mask를 적용 ➔ 다음 스케일의 연산자로 확장

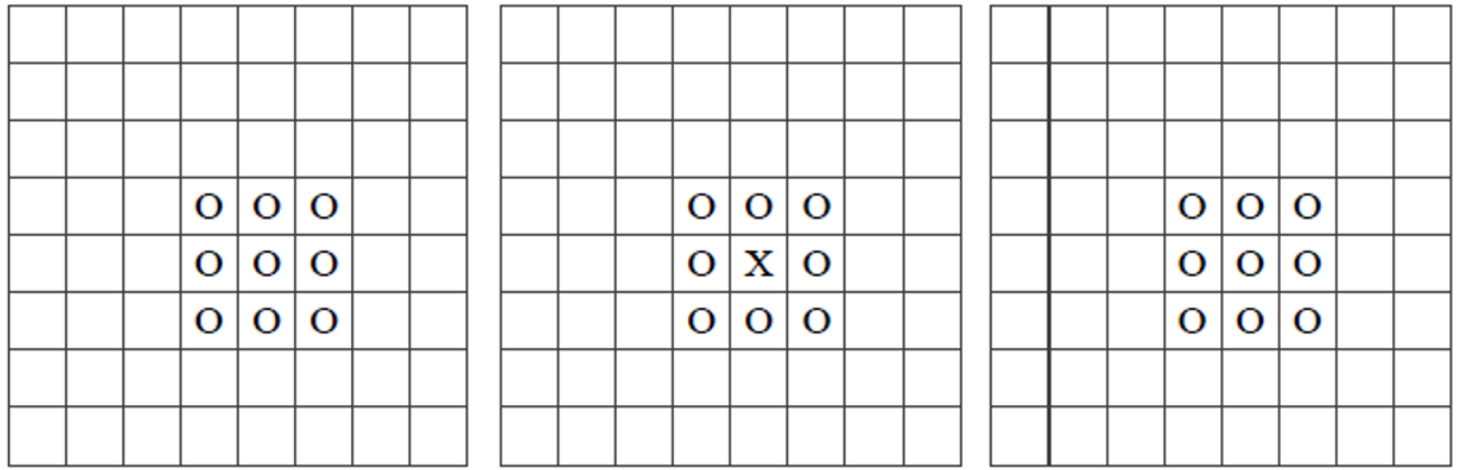


- Octave 구성
  - Octave level 1: 9x9, 15x15, 21x21, 27x27 mask 적용 (6씩 증가)
  - Octave level 2: 15x15, 27x27, 39x39, 51x51 (6➔12씩 증가)
  - Octrave level 3: 27x27, 51x51, 75x75, 99x99 (12➔24씩 증가)

SURF 검출

SURF

- 지역 극점 검출



- SIFT와의 차이점
  - SIFT: 단일 스케일 연산자를 다중 스케일 영상에 적용
  - SURF: 단일 스케일 영상에 다중 스케일 연산자를 적용

End of slide