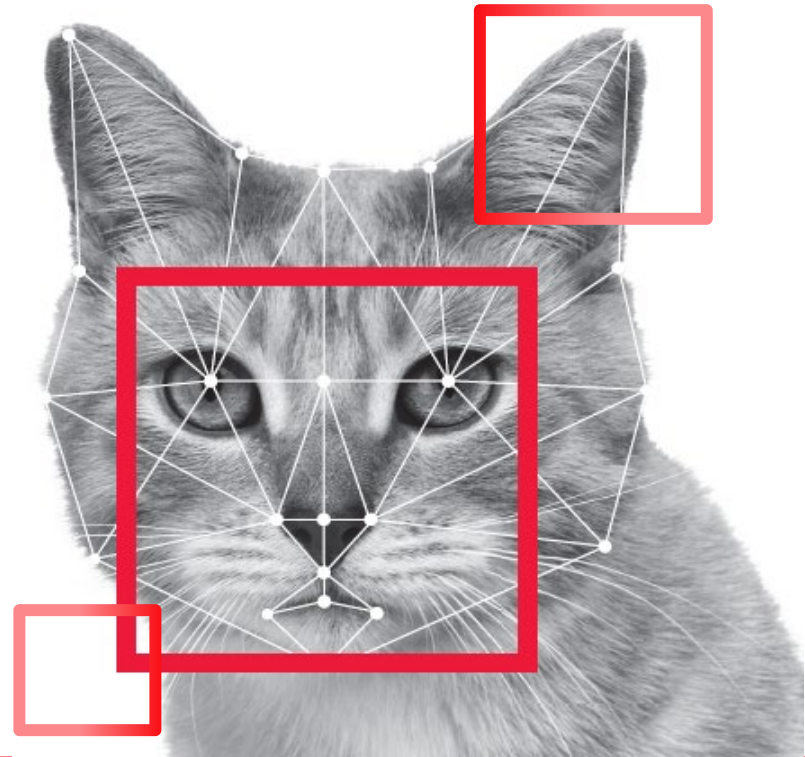


COMPUTER VISION 컴퓨터 비전

기본 개념부터 최신 모바일 응용 예까지



4장. 지역 특징 검출

PREVIEW

■ 대응점 찾기 (correspondence problem)

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

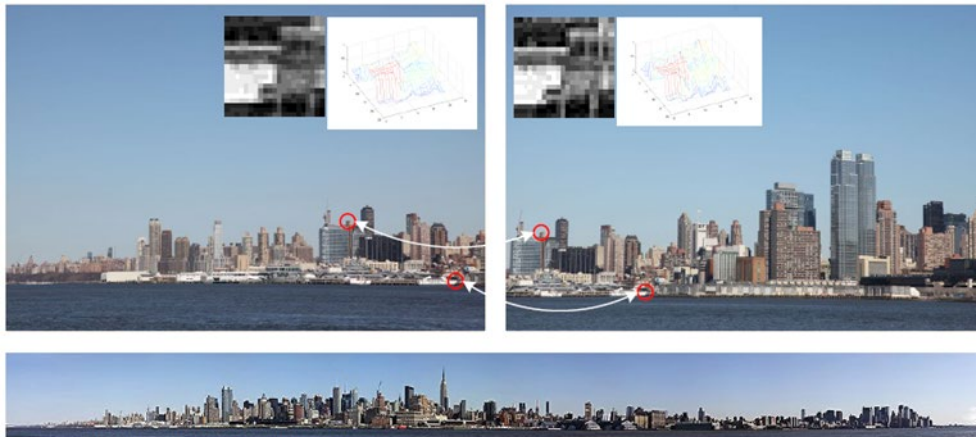
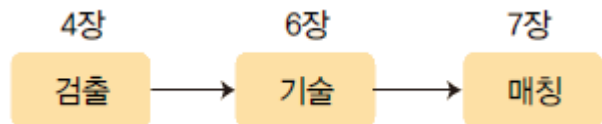


그림 4-1 대응점 찾기(확대 영상은 배의 풍무니 부근)

■ 세 단계로 해결



Detection Description Matching

각 절에서 다루는 내용

1. 지역 특징 검출의 기초
→ 지역 특징이 갖춰야 할 성질
2. 이동과 회전에 불변한 특징점 검출
3. 위치 찾기 알고리즘
→ 특징일 가능성이 높은 점들 중에서 어떤 것을 특징점으로 선택할 지
4. 스케일에 불변한 특징점 검출

4.1 지역 특징(local feature) 검출의 기초

4.1.1 특징 검출의 역사: 지역 특징의 대두

4.1.2 지역 특징의 성질

4.1.3 지역 특징 검출 원리

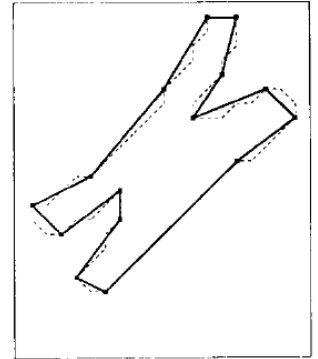
4.1.1 특징 검출의 역사: 지역 특징의 대두

■ 무엇을 특징점으로 쓸 것인가?

- 에지?
 - 에지 강도와 방향 정보만 가지므로, 매칭에 참여하기에 턱없이 부족

■ 다른 곳과 두드러지게 달라 풍부한 정보 추출 가능한 곳

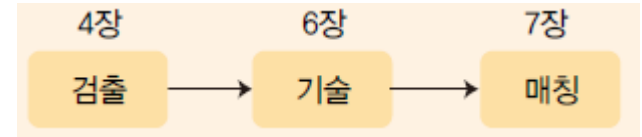
- 에지 토막에서 곡률이 큰 지점을 코너로 검출
 - 코너 검출, dominant point 검출 등의 주제로 80년대 왕성한 연구
 - 90년대 소강 국면, 2000년대 사라짐
 - 더 좋은 대안이 떠올랐기 때문
- 지역 특징이라는 새로운 물줄기
 - 명암 영상에서 직접 검출 (어떻게 생겼는 지는 한개도 안중요)
 - 의식 전환 : 코너의 물리적 의미에서 반복성(repeatability)
 - 반복성 : 한 영상에서 검출된 특징이 다른 영상에서 또 검출되는 것



4.1.2 지역 특징의 성질

■ 지역 특징

- $\langle \text{위치, 스케일, 방향, 특징 벡터} \rangle = ((y, x), s, \theta, \mathbf{x})$ 로 표현
 - 검출 단계 (4장): 위치와 스케일 알아냄
 - 기술 단계 (6장): 방향과 특징 벡터 알아냄



■ 지역 특징이 만족해야 할 특성

- 반복성(repeatability) : 같은 물체는 다른 영상에서도 동일한 속성을 가져야 함
- 분별력(distinctiveness) : 다른 곳과 충분히 구별되어야 함
- 지역성(locality) : 주변 영역의 정보만으로 검출
- 정확성(accuracy) : 여러 작업(ex. 스케일링)에도 같은 위치에서 검출
- 적당한 양 : 너무 적으면 부정확, 너무 많아도 오류 & 계산시간 문제
- 계산 효율 : 실시간성

■ 이들 특성은 길항(tradeoff) 관계

- 응용에 따라 적절한 특징을 선택해야 함

4.1.3 지역 특징 검출 원리

■ 원리

- 인지 실험
 - 대응점을 찾기가 쉬운(좋은) 점은? → 사람에게 쉬운 곳이 컴퓨터에게도 쉽다.
- 좋은 정도를 어떻게 수량화할까?
 - 여러 방향으로 **밝기 변화**가 나타나는 곳일수록 높은 점수

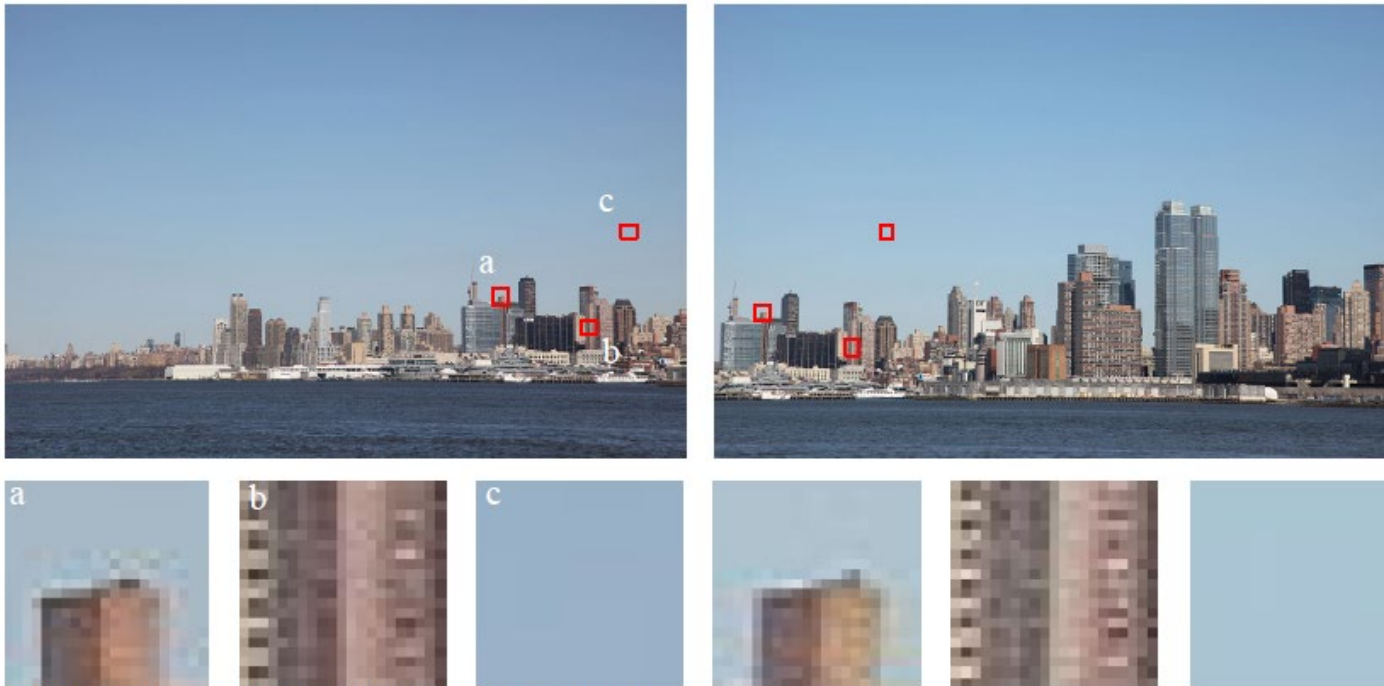


그림 4-2 a, b, c 중에 어느 곳이 지역 특징으로 유리할까?

$a > b > c$

4.2 이동과 회전에 불변한 특징점 검출

- 앞 절에서 특징이 '무엇'인지 공부하였다.
- 이제 '어떻게' 찾을 것인지 공부해보자.

4.2.1 모라벡 알고리즘

4.2.2 해리스 코너

4.2.3 2차 미분을 사용한 방법

4.2.4 슈산

4.2.1 모라벡 알고리즘 [Moravec 80]

■ 인지 실험에 주목한 모라벡

- 제곱차의 합(SSD, sum of squared difference)으로 밝기 변화 측정

$$S(v, u) = \sum_y \sum_x w(y, x) (f(y + v, x + u) - f(y, x))^2 \quad (4.1)$$

예제 4-1 제곱차 합 계산

[그림 4-3]은 삼각형을 가진 12×12 영상이다. 현재 조사하고 있는 점은 (5, 3)에 위치한 b이고, 마스크는 모든 값이 1인 3×3 크기의 박스형이라 하자. 이때 오른쪽으로 한 화소만큼 이동시킨 $S(0, 1)$ 을 계산해 보면 다음과 같이 4라는 값을 얻는다.

$$\begin{aligned} S(0, 1) &= \sum_y \sum_x w(y, x) (f(y, x + 1) - f(y, x))^2, \text{ 이때 } w(y, x) = \begin{cases} 1, & 4 \leq y \leq 6, 2 \leq x \leq 4 \\ 0, & \text{그 외} \end{cases} \\ &= \sum_{4 \leq y \leq 6} \sum_{2 \leq x \leq 4} (f(y, x + 1) - f(y, x))^2 = 4 \end{aligned}$$

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

같은 방식으로 나머지 v 와 u 값에 대해 $S(v, u)$ 를 계산해 보면, 왼쪽과 같은 $S(v, u)$ 맵을 완성할 수 있다. 점 b를 기준으로 생성한 맵으로, 연필을 들고 다른 점에 대해서도 계산해 보기 바란다. 손으로 직접 해 보는 것의 힘은 생각보다 강하다!

1	1	0
1	1	1
1	1	1

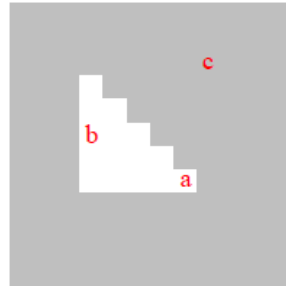
—

0	1	1
0	1	1
0	1	1

4.2.1 모라벡 알고리즘

	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
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	d	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	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) 합성 영상



▪ $S(v, u)$ 의 특성

c : 변화가 없으므로 전체적으로 낮은 값

b : 세로 방향의 에지, 가로 방향으로 큰 값

a : 코너에 해당하므로 모든 방향에서 큰 값

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

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

		u		
		-1	0	1
v	-1	0	0	0
	0	0	0	0
	1	0	0	0
		c		

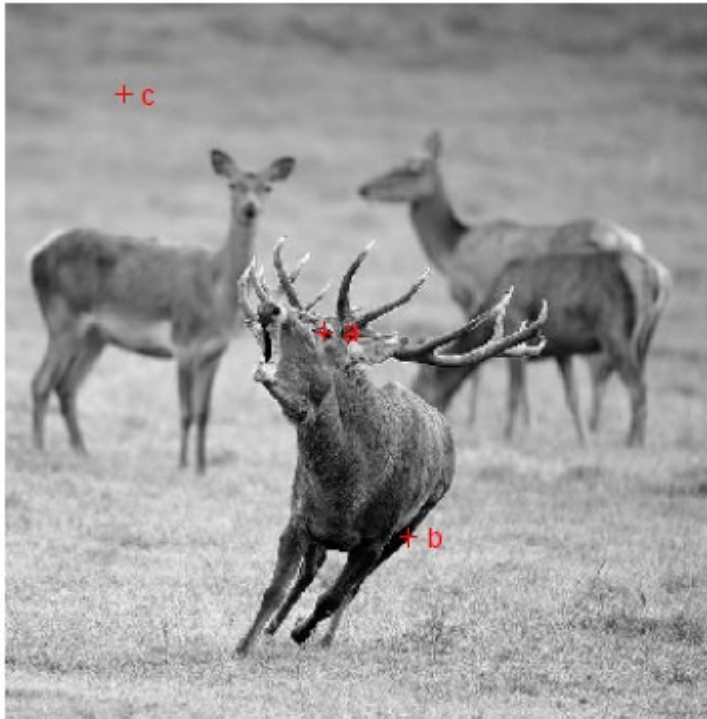
← 3*3 마스크로 측정

(b) 세 지점에서 $S(v, u)$ 맵

그림 4-3 $S(v, u)$ 맵

예제) d점에서 $S(v, u)$ 맵은?

4.2.1 모라벡 알고리즘



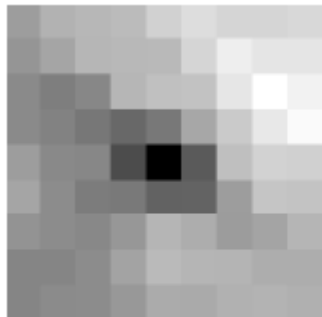
> 원래 영상

- $S(v, u)$ 의 특성

a : 눈, 모든 방향으로 큰 값

b : 에지에 수직인 방향으로 큰 값

c : 모두 낮은 값



> a



> b



> c

← 9*9 마스크로 측정

그림 4-4 실제 영상에서 S 맵(밝을수록 큰 값)

4.2.1 모라벡 알고리즘

■ $S(.)$ 맵을 관찰해 보면,

- a와 같은 코너에서는 모든 방향으로 변화가 심함
- b와 같은 에지에서는 에지 방향으로 변화 적지만, 에지에 수직 방향으로 변화 심함
- c와 같은 곳은 모든 방향으로 변화 적음
- a에 높은 값, c는 아주 낮은 값, b는 그 사이 값을 부여하는 함수를 만들면 됨

■ 모라벡의 함수 – 어떤 점이 코너라면 네 방향 모두 변화가 커야 한다

- 특징 가능성 값 C

$$C = \min(S(0, 1), S(0, -1), S(1, 0), S(-1, 0)) \quad (4.2)$$

- 한계
 - 한 화소만큼 이동하여 네 방향만 봄
 - 잡음에 대한 대처 방안 없음

4.2.2 해리스 코너 [Harris 88]

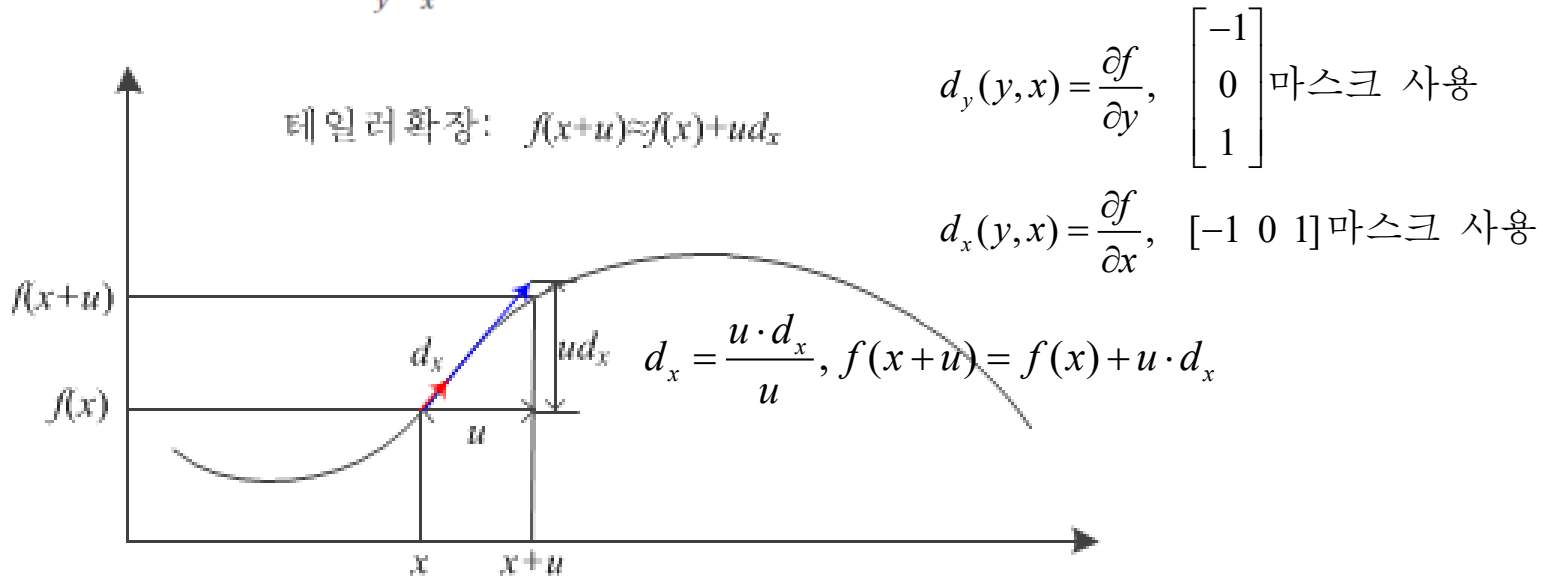
■ 해리스의 접근

- 가중치 제곱차(WSSD, weighted SSD)의 합을 이용한 잡음 대처

$$S(v, u) = \sum_y \sum_x G(y, x) (f(y + v, x + u) - f(y, x))^2 \quad (4.3)$$

- 테일러 확장 $f(y + v, x + u) \cong f(y, x) + vd_y(y, x) + ud_x(y, x)$ 을 대입하면,

$$S(v, u) \cong \sum_y \sum_x G(y, x) (vd_y(y, x) + ud_x(y, x))^2 \quad (4.5)$$



4.2.2 해리스 코너

■ 계속 유도하면,

$$\begin{aligned} S(v, u) &\cong \sum_y \sum_x G(y, x) (vd_y + ud_x)^2 \\ &= \sum_y \sum_x G(y, x) (v^2 d_y^2 + 2vud_y d_x + u^2 d_x^2) \\ &= \sum_y \sum_x G(y, x) (v \ u) \begin{pmatrix} d_y^2 & d_y d_x \\ d_y d_x & d_x^2 \end{pmatrix} \begin{pmatrix} v \\ u \end{pmatrix} \\ &= (v \ u) \sum_y \sum_x G(y, x) \begin{pmatrix} d_y^2 & d_y d_x \\ d_y d_x & d_x^2 \end{pmatrix} \begin{pmatrix} v \\ u \end{pmatrix} \end{aligned}$$

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

4.2.2 해리스 코너

- 2차 모멘트 행렬 A (second moment matrix, auto-correlation)

$$S(v, u) \cong (v \ u) \begin{pmatrix} G \circledast d_y^2 & G \circledast d_y d_x \\ G \circledast d_y d_x & G \circledast d_x^2 \end{pmatrix} \begin{pmatrix} v \\ u \end{pmatrix} = \mathbf{u} \mathbf{A} \mathbf{u}^T$$

$$\mathbf{A} = \begin{pmatrix} G \circledast d_y^2 & G \circledast d_y d_x \\ G \circledast d_y d_x & G \circledast d_x^2 \end{pmatrix} \quad (4.7)$$

요게 중요!!!
결국 x, y 방향의 변화량

$$d_y(y, x) = \frac{\partial f}{\partial y}, \quad \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \text{ 마스크 사용}$$

$$d_x(y, x) = \frac{\partial f}{\partial x}, \quad [-1 \ 0 \ 1] \text{ 마스크 사용}$$

- S 가 \mathbf{u} 와 \mathbf{A} 의 곱으로 표현
- \mathbf{A} 를 (v, u) 무관하게 계산
- \mathbf{A} 는 영상 구조를 나타냄 $\rightarrow \mathbf{A}$ 를 잘 분석하면 특징 여부를 판정

4.2.2 해리스 코너

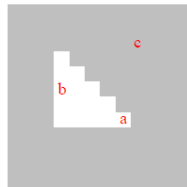
예제 4-2 2차 모멘트 행렬 A 계산

[예제 4-1]에서 사용한 [그림 4-3(a)]의 영상에서 행렬 **A**를 계산하는 과정을 살펴보자. [그림 4-5(a)]는 편의상 같은 영상을 다시 보여주는 것이고, [그림 4-5(b)]는 $d_y, d_x, d_y^2, d_x^2, d_y d_x$ 를 구한 영상이다. d_y 와 d_x 를 구하기 위해 각각 $[-1 \ 0 \ 1]^T$ 와 $[-1 \ 0 \ 1]$ 연산자를 사용하였다. [그림 4-5(c)~(e)]의 영상을 얻기 위해 다음과 같이 $\sigma=1.0$ 인 가우시안 마스크 **G**를 사용하였다.

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

이제 어떤 점의 행렬 **A**를 구할 수 있다. 예를 들어, 점 a의 행렬은 $\mathbf{A} = \begin{pmatrix} 0.522 & -0.199 \\ -0.199 & 0.527 \end{pmatrix}$ 이다.

	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	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	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	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)

4.2.2 해리스 코너

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) $G \otimes d_y^2$

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) $G \otimes d_x^2$

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) $G \otimes d_y d_x$

그림 4-5 2차 모멘트 행렬 A를 구하는 과정

4.2.2 해리스 코너

■ 2차 모멘트 행렬의 고유값(eigen value) 분석

- c와 같이 두 개의 고유값 모두 0이거나 0에 가까우면 → 변화가 거의 없는 곳
- b와 같이 고유값 하나는 크고 다른 하나는 작으면 → 한 방향으로만 변화가 있는 예지
- a와 같이 고유값 2개가 모두 크면 → 여러 방향으로 변화가 있는 지점, 특징점으로 적합!

표 4-1 세 점에서의 특징 가능성 값

	a	b	c
2차 모멘트 행렬	$\mathbf{A} = \begin{pmatrix} 0.522 & -0.199 \\ -0.199 & 0.527 \end{pmatrix}$	$\mathbf{A} = \begin{pmatrix} 0.075 & -0.075 \\ -0.075 & 0.801 \end{pmatrix}$	$\mathbf{A} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$
고유값	$\lambda_1=0.7235, \lambda_2=0.3255$	$\lambda_1=0.8087, \lambda_2=0.0673$	$\lambda_1=0.0, \lambda_2=0.0$
특징 가능성 값	$C=0.1925$	$C=0.0237$	$C=0.0$

4.2.2 해리스 코너

■ 특징 가능성 값 측정

$$C = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \quad k = 0.04 \quad (4.8)$$

- 고유값 계산을 피하고 다음과 같은 식으로 속도 향상

$$\mathbf{A} = \begin{pmatrix} p & r \\ r & q \end{pmatrix}$$

$$C = \det(\mathbf{A}) - k \times \text{trace}(\mathbf{A})^2 = (pq - r^2) - k(p + q)^2 \quad (4.9)$$

4.2.2 해리스 코너



그림 4-6 해리스 코너

← $C > 0.02$ 인 점을 검출, 너무 많아!!!

- 위치 찾기 문제(localization) 대두
 - 큰 C 값을 가진 큰 점들이 밀집되어 나타나므로 대표점 선택 필요
 - 4.3절 비최대억제
- 코너라는 용어가 적절한가?
 - 코너 → 특징점(feature point) 또는 관심점(interest point)

4.2.3 2차 미분을 사용한 방법

■ 헤시안 행렬(Hessian matrix)

- 가우시안을 포함한 헤시안 행렬
$$\mathbf{H} = \begin{pmatrix} d_{yy} & d_{yx} \\ d_{yx} & d_{xx} \end{pmatrix}$$

$$\mathbf{H} = \begin{pmatrix} d_{yy}(\sigma) & d_{yx}(\sigma) \\ d_{yx}(\sigma) & d_{xx}(\sigma) \end{pmatrix} \quad (4.11)$$

이때 $d_{st}(\sigma) = \frac{\partial}{\partial t} \left(\frac{\partial}{\partial s} (G(y, x, \sigma) \otimes f(y, x)) \right)$, s 와 t 는 y 또는 x

■ 2차 미분에서 특징 가능성 값 측정

- 헤시안의 행렬식(determinant)

$$C = \det(\mathbf{H}) = d_{yy}(\sigma) d_{xx}(\sigma) - d_{yx}(\sigma)^2 \quad (4.12)$$

- 가우시안 라플라시안(LOG)

$$C = \nabla^2 = \text{trace}(\mathbf{H}) = d_{yy}(\sigma) + d_{xx}(\sigma) \quad (4.13)$$

4.2.4 슈산(SUSAN, Smallest Univalue Segment Assimilation Nucleus)

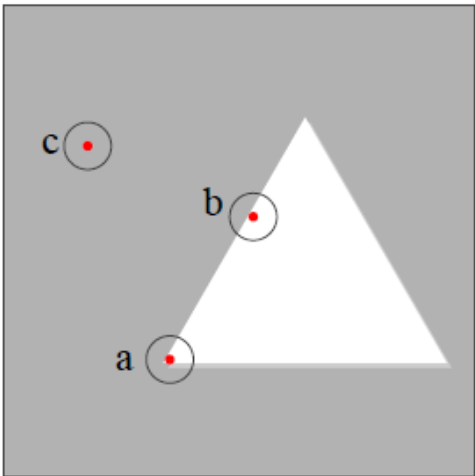


그림 4-7 슈산의 원리

		1	1	1		
	1	1	1	1	1	
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
	1	1	1	1	1	
		1	1	1		

그림 4-8 슈산이 사용하는 7×7 크기의 원형 마스크(넓이는 37)

원리

- 중심점과 인근 지역의 밝기 값이 얼마나 유사한지에 따라 특징 가능성 결정

$$usan_area(r_0) = \sum_r s(r, r_0)$$

$$\text{이때 } s(r, r_0) = \begin{cases} 1, & |f(r) - f(r_0)| \leq t_1 \\ 0, & \text{그렇지 않으면} \end{cases} \tag{4.14}$$

$$C = \begin{cases} q - usan_area(r_0), & usan_area(r_0) \leq t_2 \\ 0, & \text{그렇지 않으면} \end{cases} \tag{4.15}$$

q : 마스크 넓이의 0.75, t₂ : 마스크 넓이의 0.5

if) a에서 중심점과 다른 픽셀이 30개,

$$usan_area = 7$$

$$t_2 = 37 * 0.5 = 18.5$$

$$q = 37 * 0.75 = 27.75$$

$$C = q - usan_area = 20.75$$

4.3 위치 찾기 알고리즘

■ 지금까지 공부한 여러 가지 특징 가능성 측정 방법

- 모라벡 $C = \min(S(0, 1), S(0, -1), S(1, 0), S(-1, 0))$ (4.2)

- 해리스 $C = \det(\mathbf{A}) - k \times \text{trace}(\mathbf{A})^2 = (pq - r^2) - k(p + q)^2$ (4.9)

- 헤시안의 행렬식 $C = \det(\mathbf{H}) = d_{yy}(\sigma)d_{xx}(\sigma) - d_{yx}(\sigma)^2$ (4.12)

- LOG $C = \nabla^2 = \text{trace}(\mathbf{H}) = d_{yy}(\sigma) + d_{xx}(\sigma)$ (4.13)

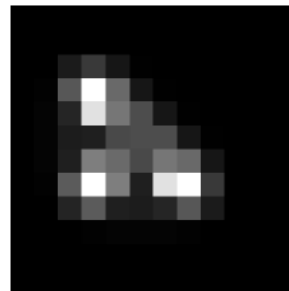
- 슈산 $C = \begin{cases} q - \text{usan_area}(r_0), & \text{usan_area}(r_0) \leq t_2 \\ 0, & \text{그렇지 않으면} \end{cases}$ (4.15)

4.3 위치 찾기 알고리즘

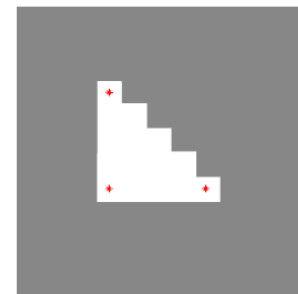
■ 해리스 적용 예

- 큰 값이 밀집되어 나타남 → 대표점 선택 필요

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	-.001	0	0	0	0	0	0	0	0
0	0	.021	.043	.017	-.001	0	0	0	0	0	0
0	-.002	.066	.191	.079	-.017	-.001	0	0	0	0	0
0	-.003	.028	.169	.089	-.058	-.017	-.001	0	0	0	0
0	-.003	-.021	.024	.055	-.058	-.058	-.017	-.001	0	0	0
0	-.003	.023	.095	.080	.055	.089	.079	.017	0	0	0
0	-.002	.068	.192	.095	.024	.169	.191	.043	-.001	0	0
0	0	.028	.068	.023	-.021	.028	.066	.021	0	0	0
0	0	0	-.002	-.003	-.003	-.003	-.002	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



(a) 식 (4.9)로 계산한 특징 가능성 맵(굵게 표시된 부분은 지역 최대점)



(b) 비최대 억제로 찾은 특징점

그림 4-9 특징 가능성 맵과 특징점 검출

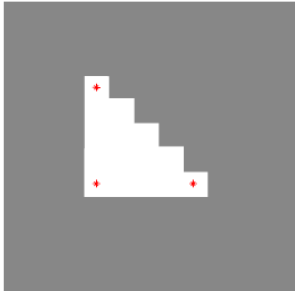
■ 비최대 억제 (non-maximum suppression)

- 이웃 화소보다 크지 않으면 억제됨 ⇒ 즉, 지역 최대만 특징점으로 검출됨

4.3 위치 찾기 알고리즘

■ 이동과 회전에 불변인가?

- 이동이나 회전 변환이 발생하여도 같은 지점에서 관심점이 검출되나?



(b) 비최대 억제로 찾은 특징점

그림 4-9 특징 가능성 맵과 특징점 검출

- 그림 4-9를 10° 회전한 영상
- 선형보간기법을 사용

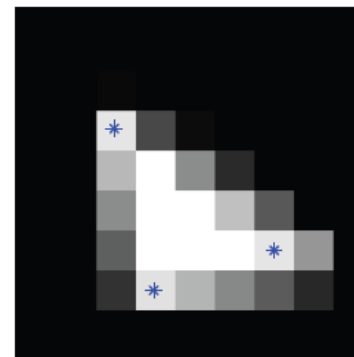
⇒ yes! 검출된다.



(a) 회전한 영상



(b) 가능성 맵



(c) 관심점 검출

그림 4-10 회전한 삼각형에서 관심점 검출

4.3 위치 찾기 알고리즘

■ 스케일에 불변인가?

- 스케일이 변해도 같은 지점에서 관심점이 검출되나?

⇒ 연산자 크기가 고정되어 있어 그렇지 않다.

스케일 변화에 대처하려면 연산자 크기를 조절하는 기능이 필수적임

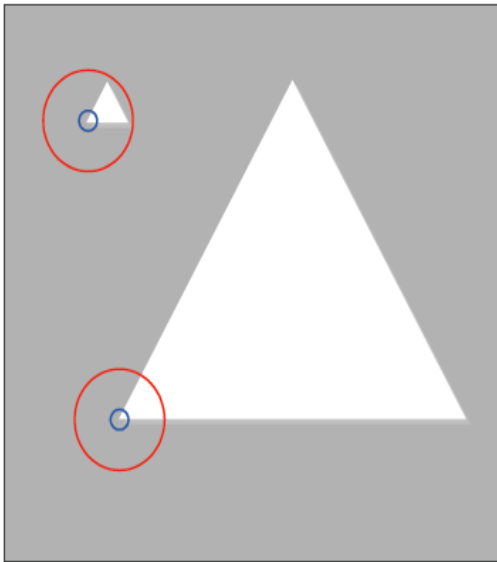


그림 4-11 스케일에 따른 적절한 연산자 크기

그럼, 연산자 크기를 조절해가면서 특징점을 조절하면 되나?

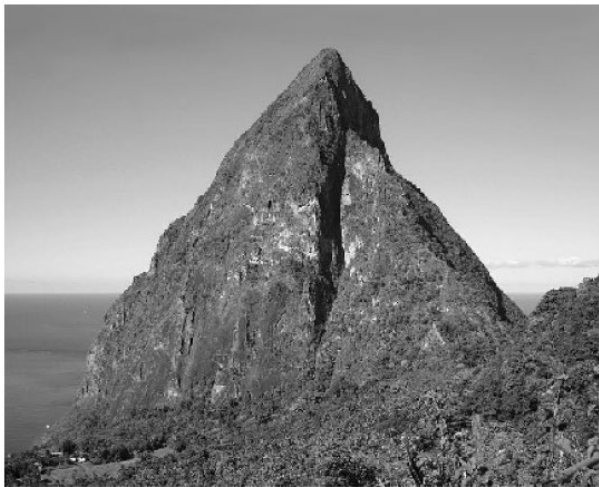
→ 비슷한 연산자(ex, 10, 11, 12, ...)에서 동시에 검출될 가능성이 큼

→ 스케일에 불변(scale-invariant)한 특징 검출

4.4 스케일에 불변한 특징점 검출

■ 거리에 따른 스케일 변화

- 예) 멀면 작고 윤곽만 어렴풋이 보이다가, 가까워지면 커지면서 세세한 부분 보임
- 사람은 강인하게 대처하는데, 컴퓨터 비전도 대처 가능한가?

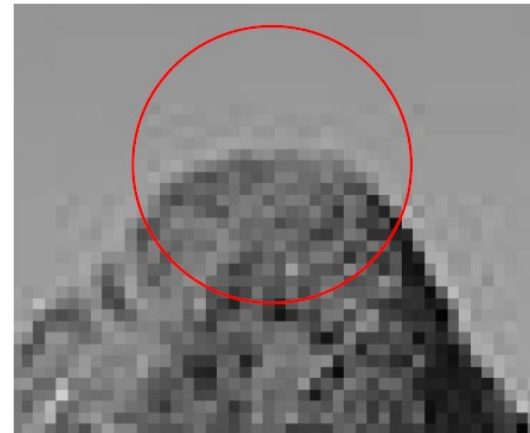


> 600×730



> 60×73

(a) 원래 영상과 1/10로 축소한 영상



> 원래 영상

(b) 산꼭대기를 확대한 부분 영상



> 축소 영상

그림 4-12 카메라와 물체 간의 거리에 따른 스케일 변화

- 해리스 알고리즘이 사용하는 크기 3인 미분 연산자는 원래의 꼭대기에서 코너가 아니라 수평 에지를 검출

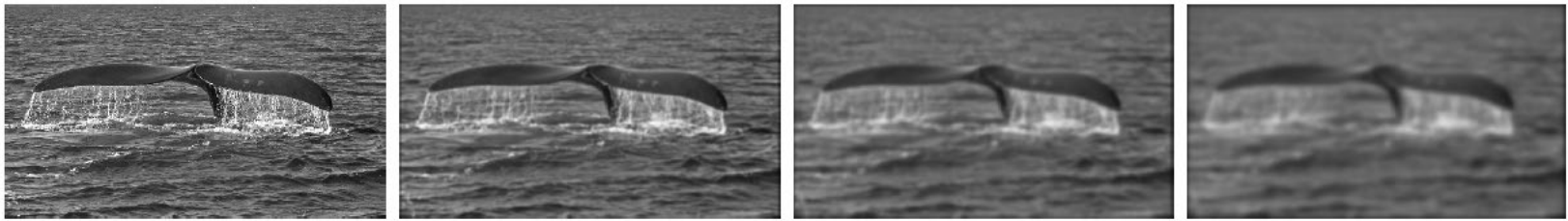
4.4.1 스케일 공간

■ 다중 스케일 접근 방법

- 스케일 축이 추가된 3차원 공간에서 **극점**(지역 최대 또는 최소점) 검출

■ 다중 스케일 영상(multi-scale image)을 구현하는 두 가지 방식

- 가우시안 스무딩 : 스케일에 해당하는 σ 가 연속 공간에 정의
- 피라미드: ½씩 줄어들므로 이산적인 단점



(a) 가우시안 스무딩



(b) 피라미드

그림 4-13 다중 스케일을 구현하는 두 가지 방식

4.4.1 스케일 공간

■ 가우시안 스무딩에 의한 스케일 공간(scale space)

- 스케일 축을 추가한 3차원 공간

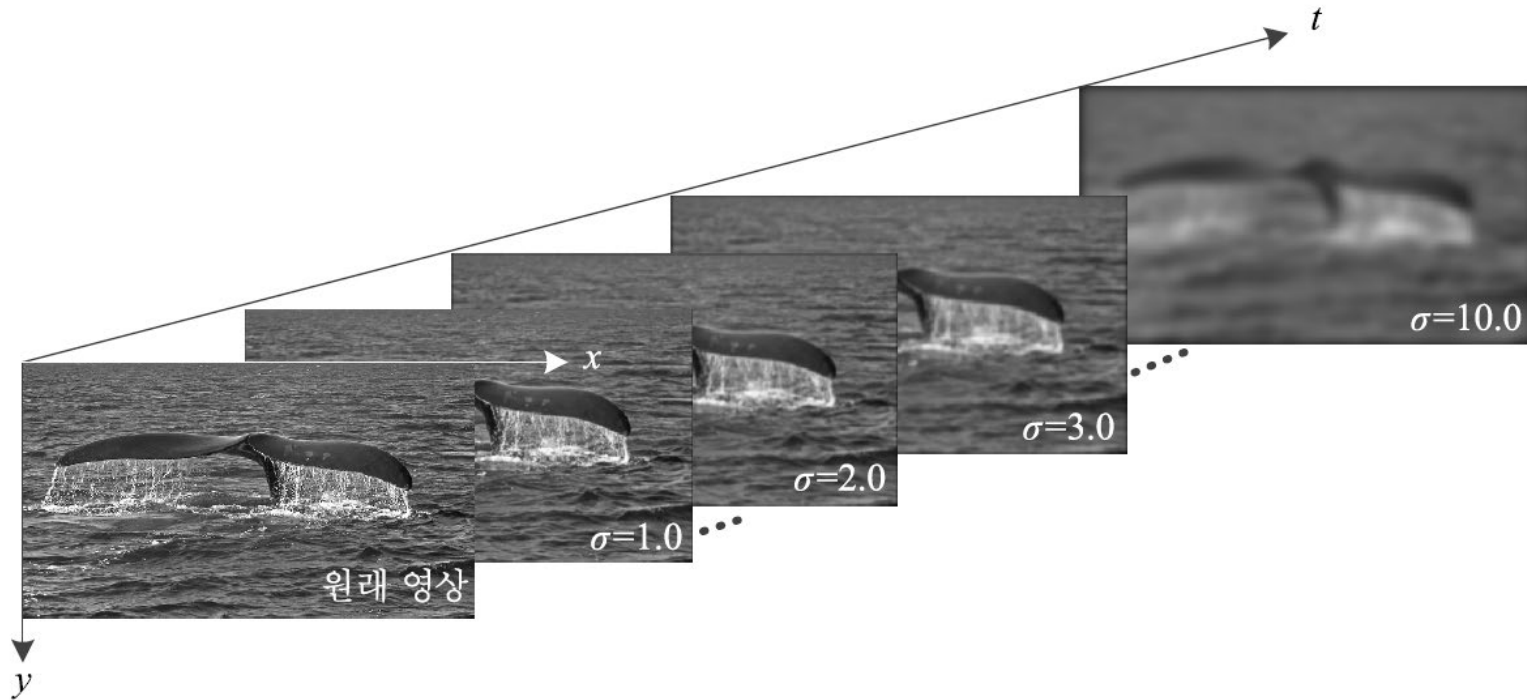


그림 4-14 (y, x, t) 로 표현되는 스케일 공간

- 스케일 매개 변수(scale parameter)

$$t = \sigma^2 \text{ (분산)}$$

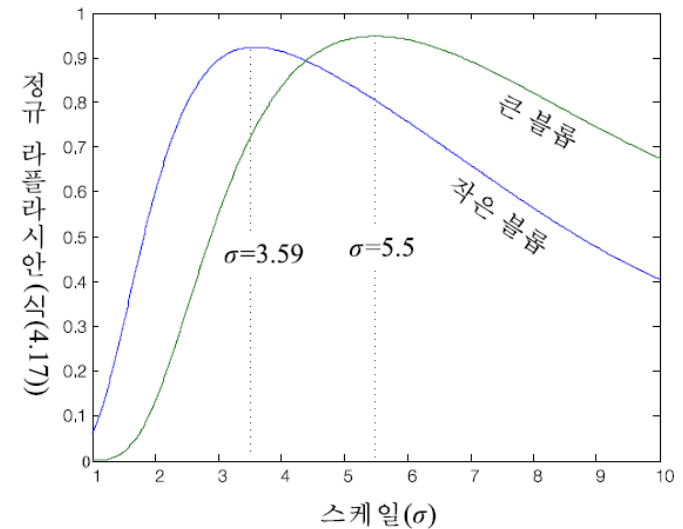
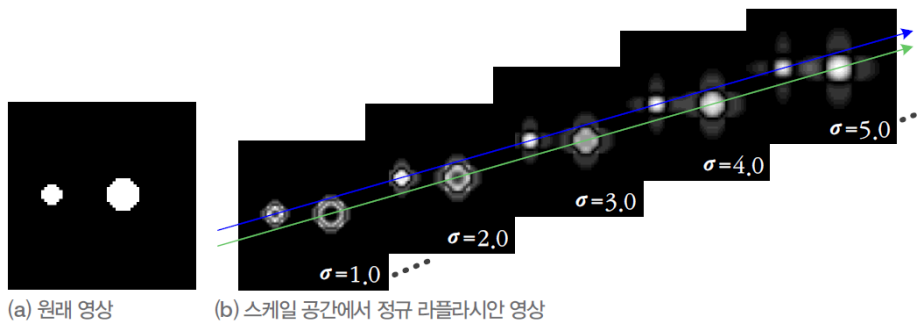
4.4.1 스케일 공간

■ t 축에서 지역 극점 탐색

- t 축을 따라 정규 라플라시안 측정해 보면, 극점 발생함

$$\nabla_{norm}^2 f = \sigma^2 |d_{yy}(\sigma) + d_{xx}(\sigma)| \quad (4.17)$$

- 실험에 따르면 t 축에서 정규 라플라시안이 가장 안정적으로 극점 생성
- 극점의 σ 값은 물체의 스케일에 해당



(c) 스케일 축에서 극점

그림 4-15 스케일 공간 (y, x, t) 에서 극점 검출

- 두 블롭(blob)의 비 : $11/7 = 1.57$
- σ 의 두 극점의 비 : $5.5/3.59 = 1.532$

4.4.2 해리스 라플라스 특징 검출

- (y, x, t) 의 3차원 공간에서 어떻게 극점을 찾을 것인가?
 - 그림 4.15의 실험은 (y, x) 를 알고 있는 상황이었음

- 해리스 라플라스의 전략

- 영상 공간 (y, x) 과 스케일 축 t 각각에서 잘 작동하는 식을 사용
 - 영상 공간에서는 해리스의 식 (4.9)를 사용
 - 스케일 축에서는 정규 라플라시안(식 (4.17)) 사용
- 해리스의 식을 다중 스케일로 확장

$$\mathbf{A}_{\text{scale_space}} = \sigma_D^2 G(\sigma_I) \circledast \begin{pmatrix} d_y^2(\sigma_D) & d_y(\sigma_D) d_x(\sigma_D) \\ d_y(\sigma_D) d_x(\sigma_D) & d_x^2(\sigma_D) \end{pmatrix} = \begin{pmatrix} p & r \\ r & q \end{pmatrix} \quad (4.18)$$

$$C = \det(\mathbf{A}_{\text{scale_space}}) - k \times \text{trace}(\mathbf{A}_{\text{scale_space}})^2 = (pq - r^2) - k(p + q)^2 \quad (4.19)$$

- 극점을 찾으면 (y, x, σ) 저장

4.4.3 SIFT (Scale-Invariant Feature Transform)검출

■ SIFT의 등장

- 1999년 David Lowe 교수의 논문 [Lowe 99]
- 2004년 IJCV에 확장된 논문 발표 [Lowe 2004]
- 반복성이 최고수준
- 계산 시간이 짧다
- 성능이 뛰어나 현재 가장 널리 사용되며, 다양한 변형이 개발되어 있음

[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

4.4.3 SIFT 검출

■ SIFT의 스케일 공간

- 피라미드+가우시안 구조
- 각 층은 여섯 영상의 묶음(옥타브, octave)으로 구성, 4x4가 될 때까지 반복
- 옥타브의 영상은 σ_i 로 스무딩
 - $\sigma_{i+1} = k\sigma_i$ ($\sigma_0 = 1.6, k = 2^{1/3}$)
- 옥타브 1의 1번째 영상은 옥타브 0의 4번째 영상, 즉 1번째 영상이 사용한 $\sigma_1 = 1.6$ 의 2배가 되는 영상을 반으로 축소하여 사용

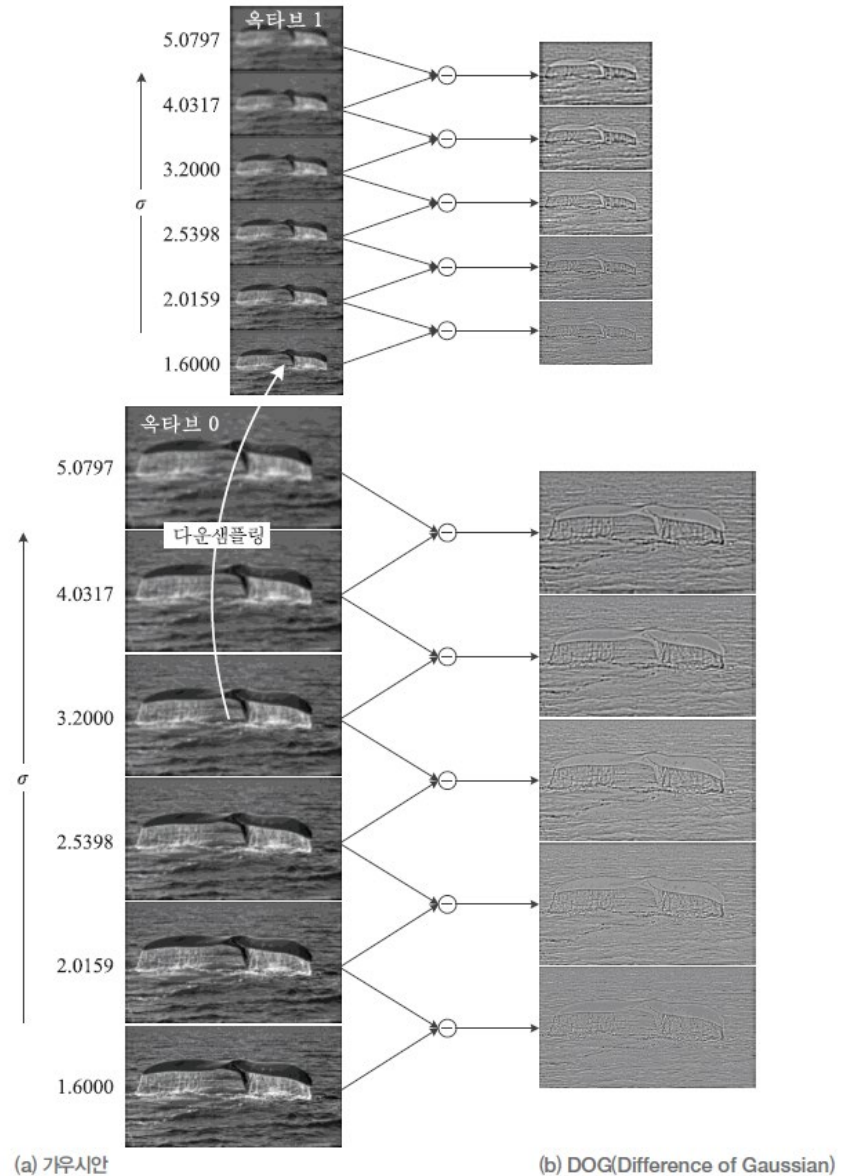


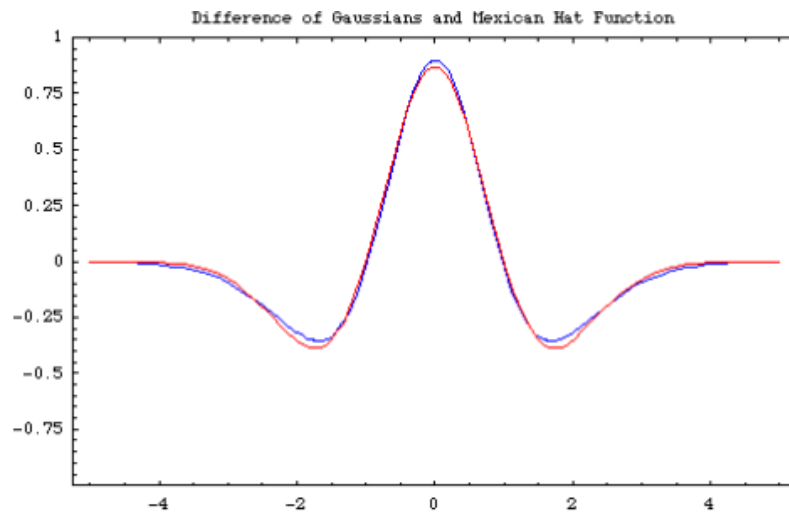
그림 4-16 SIFT가 사용하는 스케일 공간(DOG 피라미드 구조)

4.4.3 SIFT 검출

■ 정규 라플라시안 맵 구축

- [Mikolajczik 2002a]의 실험 결과에 따르면, 정규 라플라시안이 가장 안정적으로 극점 형성
- 정규 라플라시안과 유사한 DOG (Difference of Gaussian) 계산으로 대체
 - DOG는 단지 차영상을 계산하므로 매우 빠름

$$\begin{aligned} DOG(\sigma_i) &= G(\sigma_{i+1}) \otimes f - G(\sigma_i) \otimes f \\ &= G(k\sigma_i) \otimes f - G(\sigma_i) \otimes f = (G(k\sigma_i) - G(\sigma_i)) \otimes f \end{aligned} \quad (4.20)$$



— DOG

— 정규 라플라시안

4.4.3 SIFT 검출

■ 특징점 (키폰트, keypoint) 검출

- 한 옥타브에는 다섯 장의 DOG 영상
- 중간에 끼인 세 장의 DOG 맵에서 **극점** 검출
 - 주위 26개 이웃에 대해 최저 또는 최대인 점
- 검출된 극점을 키폰트라 부름

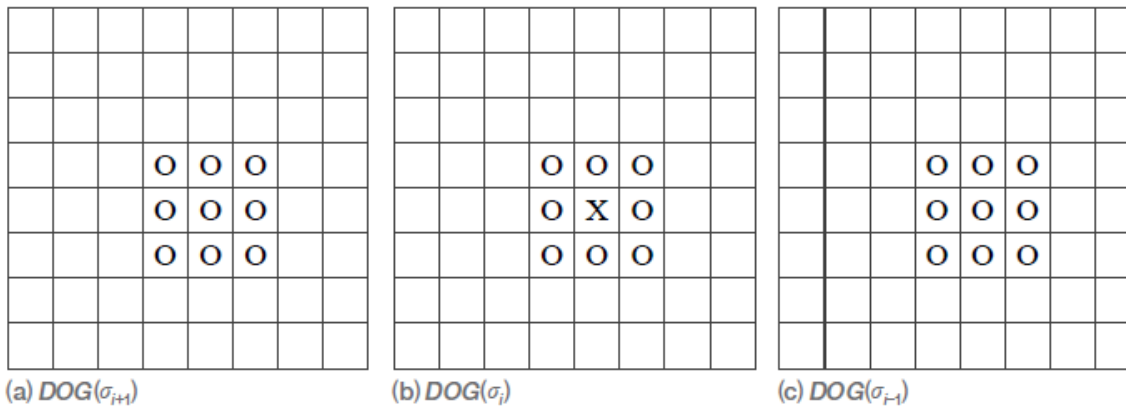


그림 4-17 DOG 맵에서 키폰트(지역 극점) 검출

4.4.3 SIFT 검출

■ 위치와 스케일 계산

- 키포인트는 $\langle y, x, o, i \rangle$ 정보를 가짐 (옥타브 o 의 i 번째 DOG 영상의 (y, x) 에서 검출)
- 미세 조정 (부분 화소 정밀도)을 거쳐 $\langle y', x', o, i' \rangle$ 로 변환됨
- 위치와 스케일 계산 식 적용 : 옥타브 0에 있는 영상의 위치 (y, x) 및 s 계산

$$\begin{aligned}(y, x) &= (y' \times 2^o, x' \times 2^o) \\ s &= 1.6 \times 2^{\frac{o+i'}{3}}\end{aligned}\tag{4.21}$$

■ 공개 소프트웨어

- David Lowe
- Rob Hess
- Andrea Vedaldi
- OpenCV

4.4.4 SURF(Speeded-Up Robust Features) 검출

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

■ SURF

- 반복률 희생 없이 SIFT보다 빠른 알고리즘 추구
- 헤시안의 행렬식 이용

$$C = \det(\mathbf{H}) = d_{yy}(\sigma) d_{xx}(\sigma) - d_{yx}(\sigma)^2 \quad (4.12)$$

$$d_{yy}(\sigma) = \frac{\partial^2}{\partial y^2} (G(\sigma) \circledast f) = \left(\frac{\partial^2}{\partial y^2} G(\sigma) \right) \circledast f \quad (4.22)$$

- 행렬식을 빠르게 계산하기 위해, d_{yy} , d_{xx} , d_{yx} 를 9*9 마스크로 근사 계산
 - 마스크 계산은 적분 영상 이용

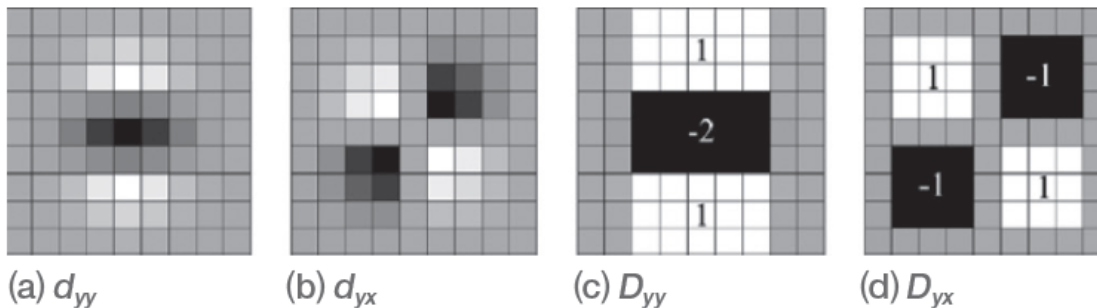


그림 4-18 가우시안을 두 번 미분한 연산자(화색은 0)

4.4.4 SURF 검출

■ SURF의 스케일 공간

- 원본 영상은 그대로 둔 채 다중 스케일 마스크를 적용

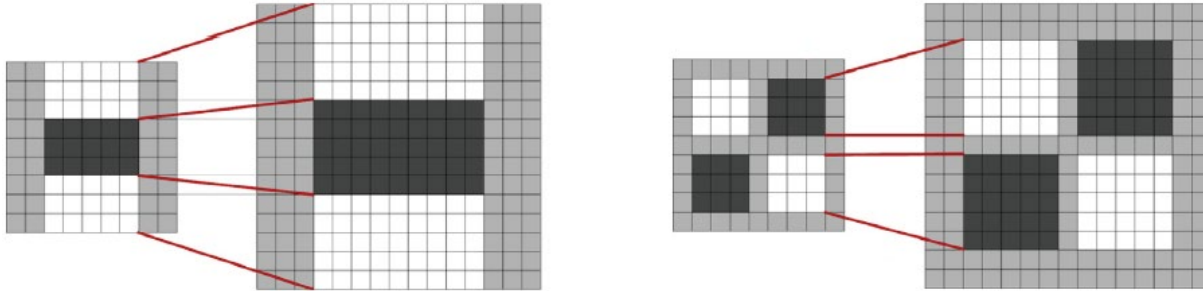


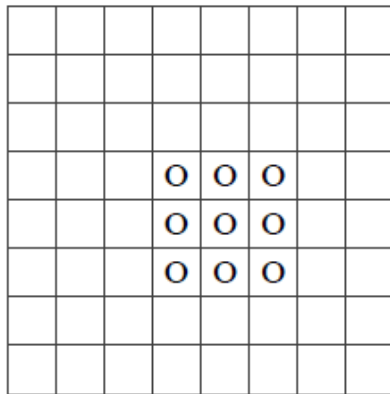
그림 4-19 다음 스케일의 연산자로 확장

- 옥타브 구성
 - 첫 번째 옥타브 9×9 , 15×15 , 21×21 , 27×27 마스크 적용 (6씩 증가)
 - 두 번째 옥타브 15×15 , 27×27 , 39×39 , 51×51
 - 첫 번째 옥타브의 두 번째 마스크 15×15 에서 시작하고 6의 두 배인 12씩 증가
 - 세 번째 옥타브 27×27 , 51×51 , 75×75 , 99×99
 - 두 번째 옥타브의 두 번째 마스크 27×27 에서 시작하고 12의 두 배인 24씩 증가

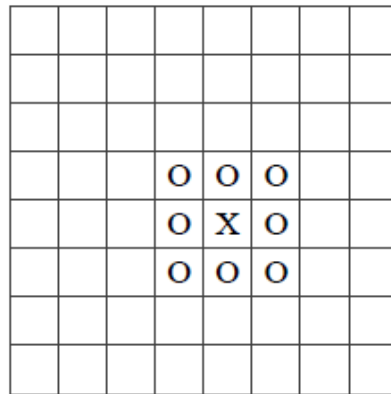
4.4.4 SURF 검출

■ 지역 극점 검출

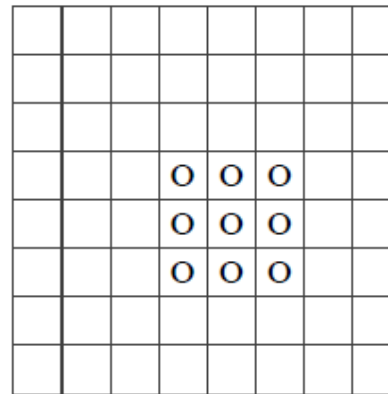
- 첫 번째 옥타브에서 중간에 끼인 15*15와 21*21에서 지역 극점 검출 (그림 4-17)
- 두 번째 옥타브에서 중간에 끼인 27*27과 39*39에서 지역 극점 검출
- ...



(a) $DOG(\sigma_{i+1})$



(b) $DOG(\sigma_i)$



(c) $DOG(\sigma_{i-1})$

그림 4-17 DOG 맵에서 키폰트(지역 극점) 검출

■ SURF의 속도 개선 보고 [Bay2008]

- 800*640 영상에서 SURF 70ms, SIFT 400ms, 해리스 라플라스 2100ms

4.4.5 지역 특징 검출 알고리즘의 특성 비교

■ 튜토리얼 논문

- [Tuytelaars2007] Tinne Tuytelaars and Krystian Mikolajczyk, "Local invariant feature detectors: a survey," Foundations and Trends of Computer Graphics and Vision, Vol.3, No.3, pp.177-280.

■ 성능 분석 논문

- [Schmid2000] Cordelia Schmid, et al., "Evaluation of interest point detectors," International Journal of Computer Vision, Vol.37, No.2, pp.151-172.
- [Mikolajczyk2005b] Krystian Mikolajczyk, et al., "A Comparison of Affine Region Detectors," International Journal of Computer Vision, Vol.65, No.1-2, pp.43-72.
- [Miksik2012] Ondrej Miksik and Krystian Mikolajczyk, "Evaluation of local detectors and descriptors for fast feature matching," International Conference on Pattern Recognition, pp.2681-2684.
- [Aanes2012] Henrik Aanes, et al., "Interesting interest points," International Journal of Computer Vision, Vol.97, pp.18-35.

■ 어떤 지역 특징을 선택해야 하나?

- 손수 성능 실험을 수행하고 판단
- [Tuytelaars2007, 7.1절]의 지침 참조