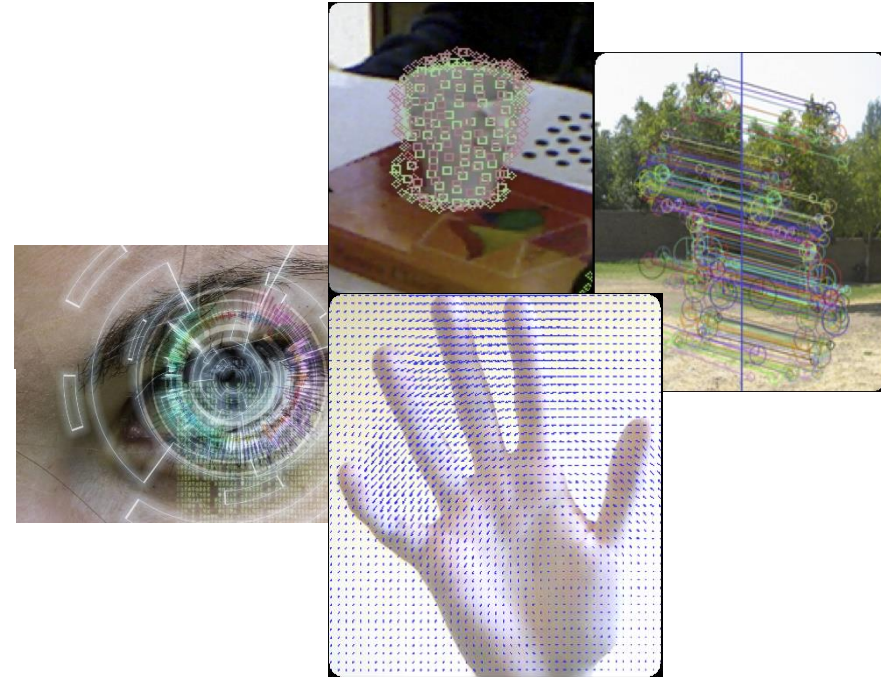


2023, Fall COMPUTER VISION

비전
프로그래밍



Dept. of IT Engineering, Sookmyung Women's University
Prof. Byung-Gyu Kim

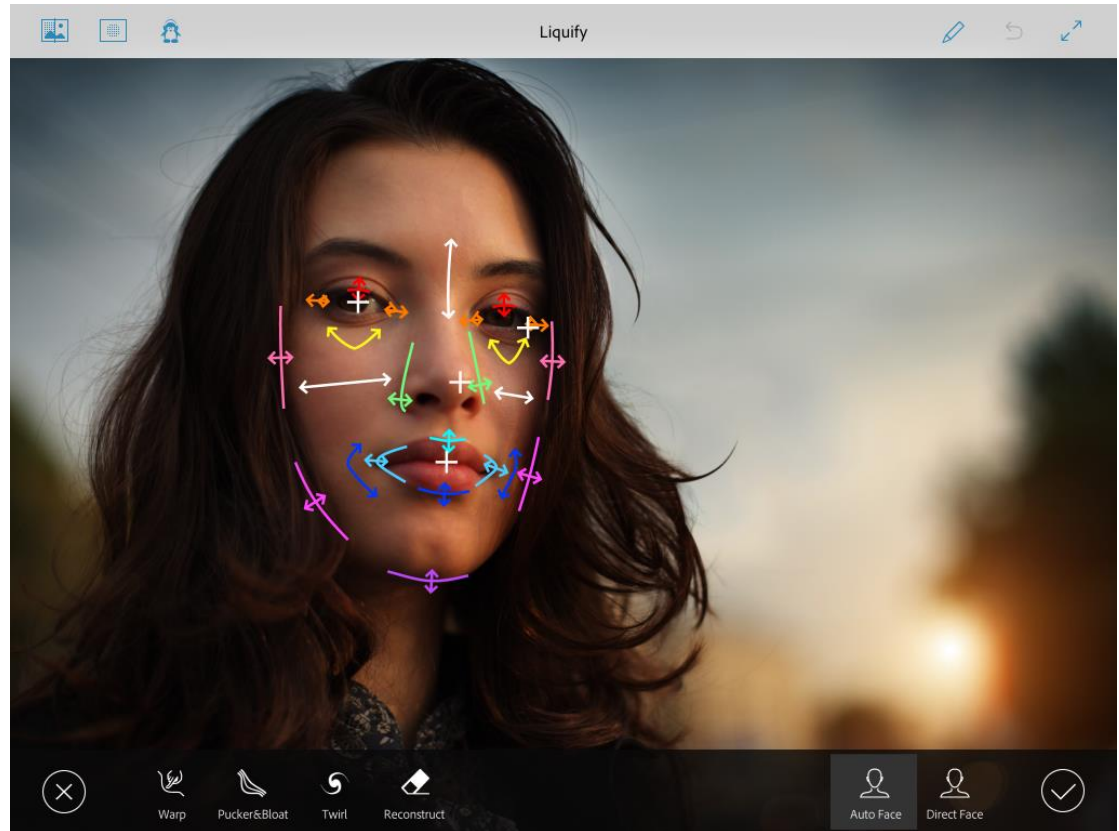
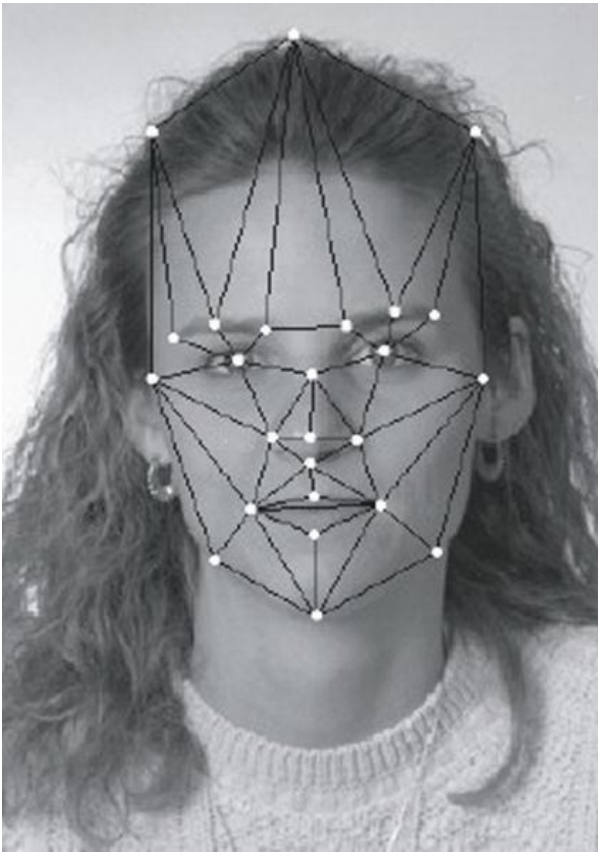
9장. Interest Points Detection & Matching (관
심점 검출 및 매칭) (Theory)

PREVIEW



1장의 wide view 이미지로?

PREVIEW



PREVIEW

■ Correspondence detection (대응점 찾기)

- 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)

- 해결을 위해 세 단계 구성(일반적)





학습 목표

- 관심점/특징점(Key Point)의 정의 및 특성
- 국부 특징 (local feature)
- 다양한 특징의 종류
 - Harris 코너
 - FAST
 - SIFT
 - SURF
 - 다중 해상도 기반 처리 등

The concept of interest points

■ Interest points (관심점)

- 영상 속의 객체가 가지는 **유의미한 화소 특징점**으로 정의
- **Keypoints** 또는 **feature points**로 알려져 있으며
- Can be used to solve many problems in object recognition, image registration, visual tracking, 3D reconstruction, and more.

■ Interest points (관심점) 활용 조건

- **sufficient number of such points** are detected in the images of interest
- these points are **distinguishing and stable features** that can be accurately localized

The concept of interest points: Local feature (지역 특징)

■ 무엇을 특징점으로 쓸 것인가?

■ 에지?

- 에지 강도와 방향 정보만 가지므로, 매칭에 참여하기에 턱없이 부족

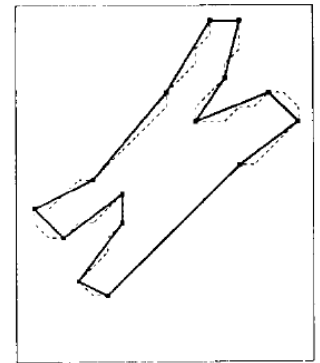
■ 다른 곳과 두드러지게 달라 풍부한 정보 추출 가능한 곳

■ 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 특징(Local feature)이라는 새로운 물줄기

- 명암 영상에서 직접 검출
- 의식 전환: 코너의 물리적 의미 → 반복성



The concept of interest points: Local feature (지역 특징)

■ 지역 특징(Local feature)

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

■ 지역 특징이 만족해야 할 특성

- 반복성
- 변별력
- 지역성
- 정확성
- 적당한 양
- 계산 효율

■ 대부분 특성은 trade-off 관계

- 응용에 따라 적절한 특징을 선택해야 함

The concept of interest points:

How to define local features? (지역 특징 검출 원리)

■ 원리

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

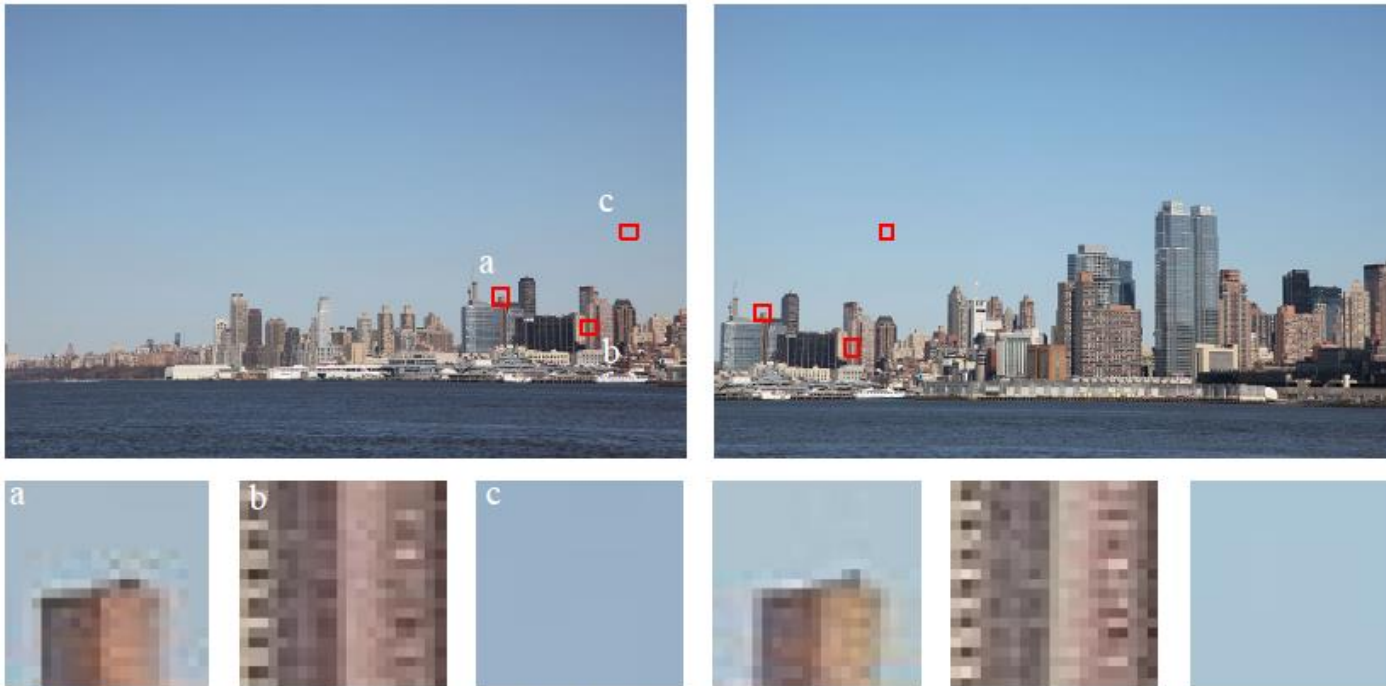


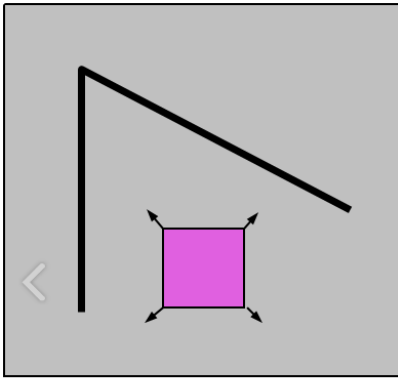
그림 4-2 a, b, c 중에 어느 곳이 지역 특징으로 유리할까?

The concept of interest points: Various kinds

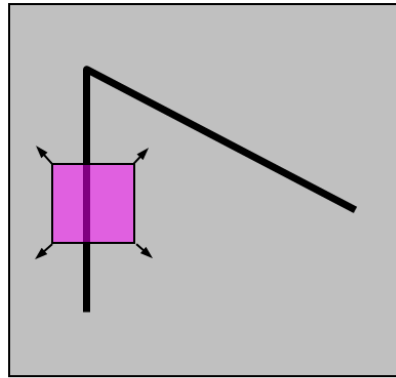
- Kinds of interest points (features)
 - Detecting **Harris corners**
 - Detecting **FAST features**
 - Detecting the **scale-invariant SURF features**
 - Describing **SURF features**

Harris corners (해리스 코너)

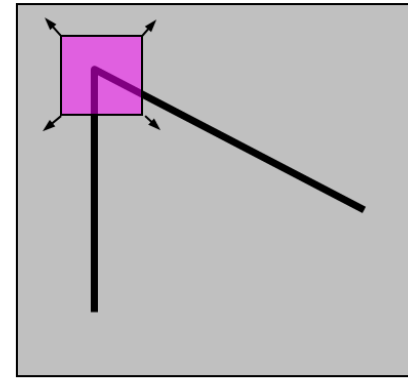
- Harris corners 검출기 개념 [Harris88]
 - 코너 점의 경우는 모든 방향으로 영상변화 크다.
 - 그 외 나머지 점에서는 영상 변화가 적거나 변화 방향이 적다.



“flat” region:
no change in all
directions



“edge”:
no change along the
edge direction



“corner”:
significant change in
all directions

Harris corners (해리스 코너)

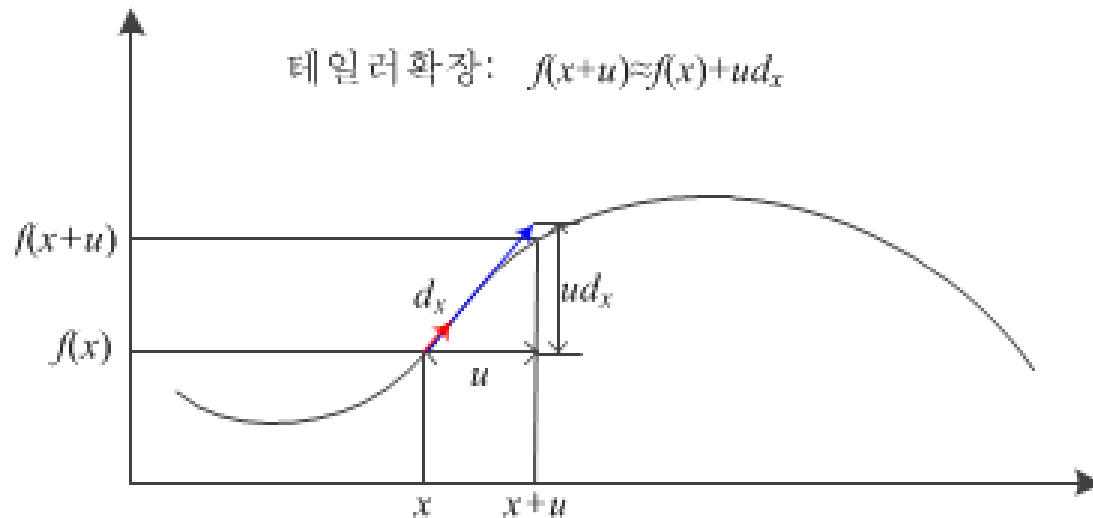
■ 해리스의 접근 [Harris88]

- 가중치 제곱차의 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,

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



Harris corners (해리스 코너)

- 계속 유도하면,

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

Harris corners (해리스 코너)

■ 2차 모멘트 행렬 A

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

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

- (v, u) 는 실수 가능
- \mathbf{A} 를 (v, u) 무관하게 계산할 수 있음 ($\because S$ 가 \mathbf{u} 와 \mathbf{A} 의 곱으로 인수 분해되어 있으므로)
- \mathbf{A} 는 영상 구조를 나타냄 \rightarrow \mathbf{A} 를 잘 분석하면 특징 여부를 판정할 수 있음

Harris corners (해리스 코너)

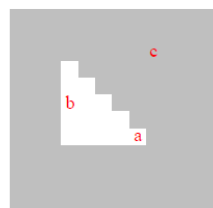
예제 4-2 2차 모멘트 행렬 A 계산

[예제 4-1]에서 사용한 [그림 4-3(a)]의 영상에서 행렬 \mathbf{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 \mathbf{A} 를 구할 수 있다. 예를 들어, 점 a 의 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)

Harris corners (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) $G \otimes d_x d_y$

그림 4-5 2차 모멘트 행렬 A를 구하는 과정

Harris corners (해리스 코너)

■ 2차 모멘트 행렬의 고유값(eigenvalue) 분석

- c와 같이 두 개의 고유값 모두 0이거나 0에 가까우면 → 변화가 거의 없는 곳 (flat)
- b와 같이 고유값 하나는 크고 다른 하나는 작으면 → 한 방향으로만 변화가 있는 에지 (edge)
- a와 같이 고유값 두 개가 모두 크면 → 2개 이상의 방향 또는 여러 방향으로 변화가 있는 지점(corner). **특징점으로 적합!**

표 4-1 세 점에서의 특징 가능성 값

| | a | b | c |
|-----------|--|--|--|
| 2차 모멘트 행렬 | $A = \begin{pmatrix} 0.522 & -0.199 \\ -0.199 & 0.527 \end{pmatrix}$ | $A = \begin{pmatrix} 0.075 & -0.075 \\ -0.075 & 0.801 \end{pmatrix}$ | $A = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ |
| 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 |

Harris corners (해리스 코너)

■ 특징 가능성 값 측정

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

→ 고유값(eigenvalue) 계산에 시간이 많이 소요됨

- 고유값 계산을 피해 속도 향상: **2차 모멘트 행렬을 직접 활용**

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

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

Harris corners (해리스 코너)



그림 4-6 해리스 코너

← $C > 0.02$ 인 점을 검출

- 위치 찾기 문제 대두
 - 큰 C 값을 가진 큰 점들이 밀집되어 나타나므로 대표점 선택 필요
- 코너라는 용어가 적절한가?
 - 코너 → 특징점 또는 관심점

Harris corners (해리스 코너): 2차 미분을 사용한 방법

■ Hessian matrix (헤시안 행렬)

$$\mathbf{H} = \begin{pmatrix} d_{yy} & d_{yx} \\ d_{yx} & d_{xx} \end{pmatrix}$$

- 가우시안을 포함한 헤시안 행렬

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

(4.11)

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

■ 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)

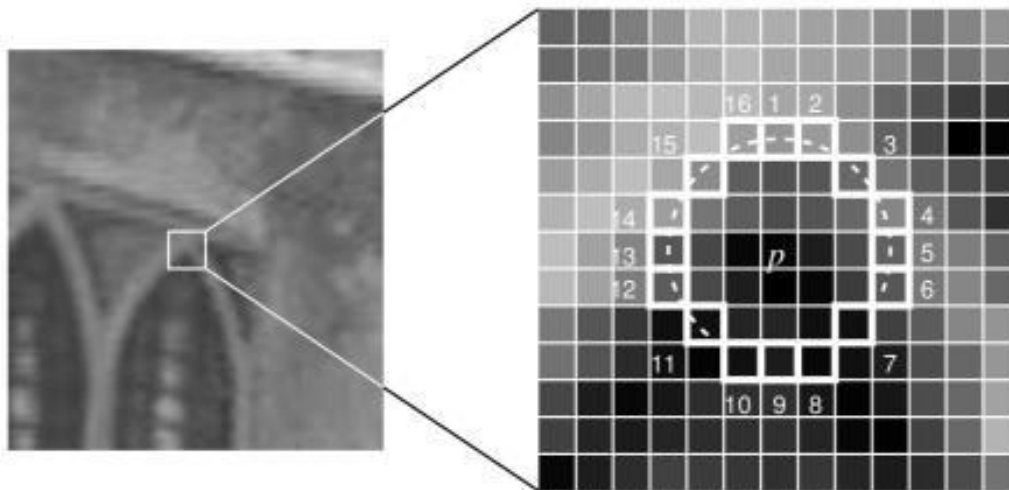
$$C = \nabla^2 = \text{trace}(\mathbf{H}) = d_{yy}(\sigma) + d_{xx}(\sigma) \quad (4.13)$$

Features from Accelerated Segment Test (FAST) 특징 추출

- 기본적으로 Corner point 검출

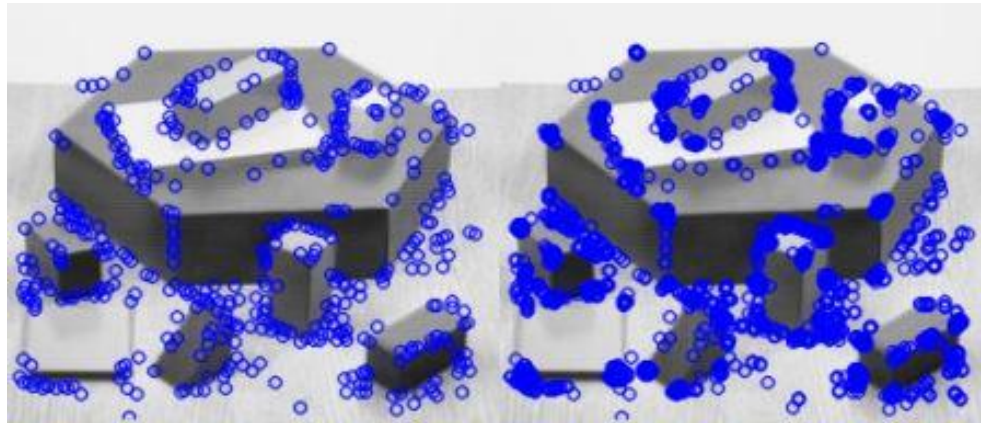
- Feature Detection using FAST

1. Select a pixel p in the image which is to be identified as an interest point or not. Let its intensity be I_p .
2. Select appropriate threshold value t .
3. Consider a circle of 16 pixels around the pixel under test. (See the image below)



Features from Accelerated Segment Test (FAST) 특징 추출

4. Now the pixel \mathbf{p} is a corner if there exists a set of \mathbf{n} contiguous pixels in the circle (of 16 pixels) which are all brighter than $I_p + \mathbf{t}$, or all darker than $I_p - \mathbf{t}$. (Shown as white dash lines in the above image). \mathbf{n} was chosen to be 12.
5. A **high-speed test** was proposed to exclude a large number of non-corners. This test examines only the four pixels at 1, 9, 5 and 13 (First 1 and 9 are tested if they are too brighter or darker. If so, then checks 5 and 13). If \mathbf{p} is a corner, then at least three of these must all be brighter than $I_p + \mathbf{t}$ or darker than $I_p - \mathbf{t}$. If neither of these is the case, then \mathbf{p} cannot be a corner. The full segment test criterion can then be applied to the passed candidates by examining all pixels in the circle. This detector in itself exhibits high performance, but there are several weaknesses:
 - It does not reject as many candidates for $n < 12$.
 - The choice of pixels is not optimal because its efficiency depends on ordering of the questions and distribution of corner appearances.
 - Results of high-speed tests are thrown away.
 - Multiple features are detected adjacent to one another.



<First image shows FAST with nonmaxSuppression and second one without nonmaxSuppression>

Feature points 위치 찾기 알고리즘: rotation invariant

■ 이동과 회전에 불변인가?

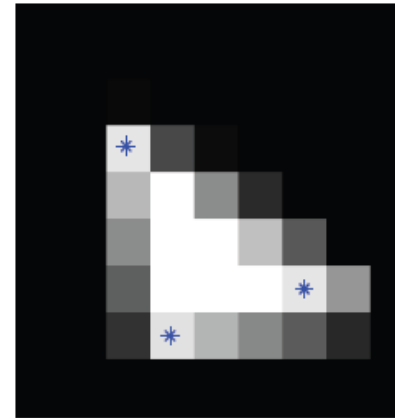
- 이동이나 회전 변환이 발생하여도 같은 지점에서 관심점이 검출되나?
⇒ yes! 검출된다.



(a) 회전한 영상



(b) 가능성 맵



(c) 관심점 검출

그림 4-10 회전한 삼각형에서 관심점 검출

Feature points 위치 찾기 알고리즘: scale invariant

■ 스케일에 불변인가?

- 스케일이 변해도 같은 지점에서 관심점이 검출되나?

⇒ 연산자 크기가 고정되어 있어 그렇지 않다.

스케일 변화에 대처하려면 연산자 크기를 조절하는 기능이 필수적임

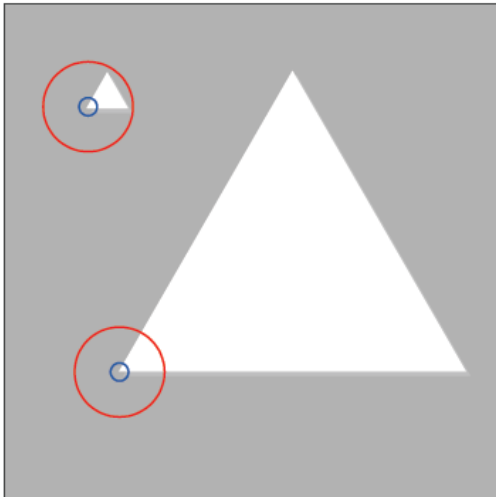


그림 4-11 스케일에 따른 적절한 연산자 크기

Feature points: Scale invariant (스케일 불변) 특징점 검출

■ 거리에 따른 스케일 변화

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

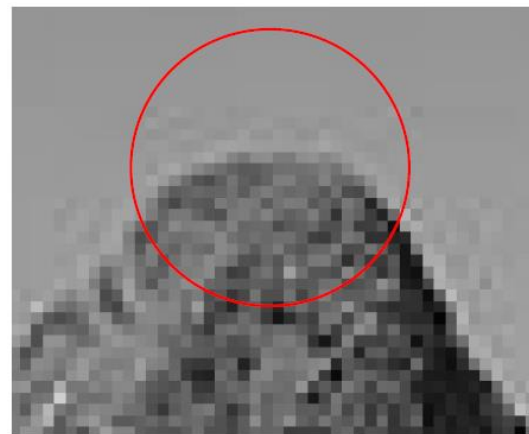


> 600×730



> 60×73

(a) 원래 영상과 1/10로 축소된 영상



> 원래 영상

(b) 산꼭대기를 확대한 부분 영상

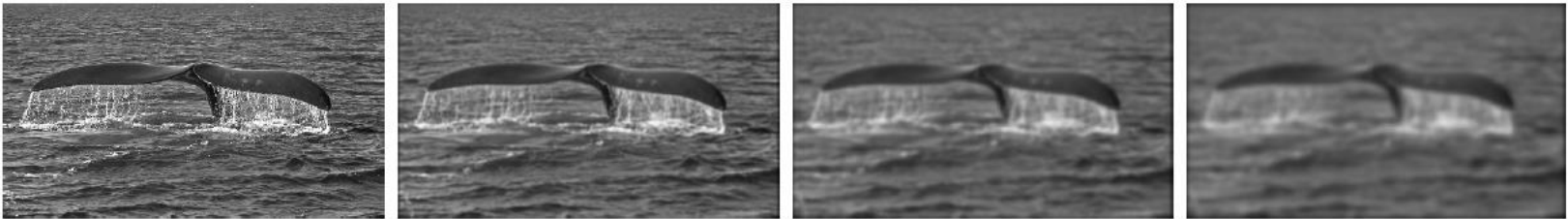


> 축소 영상

그림 4-12 카메라와 물체 간의 거리에 따른 스케일 변화

다중 스케일 공간(multi-scale space) 개념

- 다중 스케일 영상을 구현하는 두 가지 방식
 - 가우시안 스무딩: 스케일에 해당하는 σ 가 연속 공간에 정의
 - 피라미드: $1/2$ 씩 줄어듦으로 이산적인 단점



(a) 가우시안 스무딩



(b) 피라미드

그림 4-13 다중 스케일을 구현하는 두 가지 방식

다중 스케일 공간 (multi-scale space) 개념: 3차원 공간 데이터

- 가우시안 스무딩에 의한 스케일 공간
 - 스케일 축을 추가한 3차원 공간

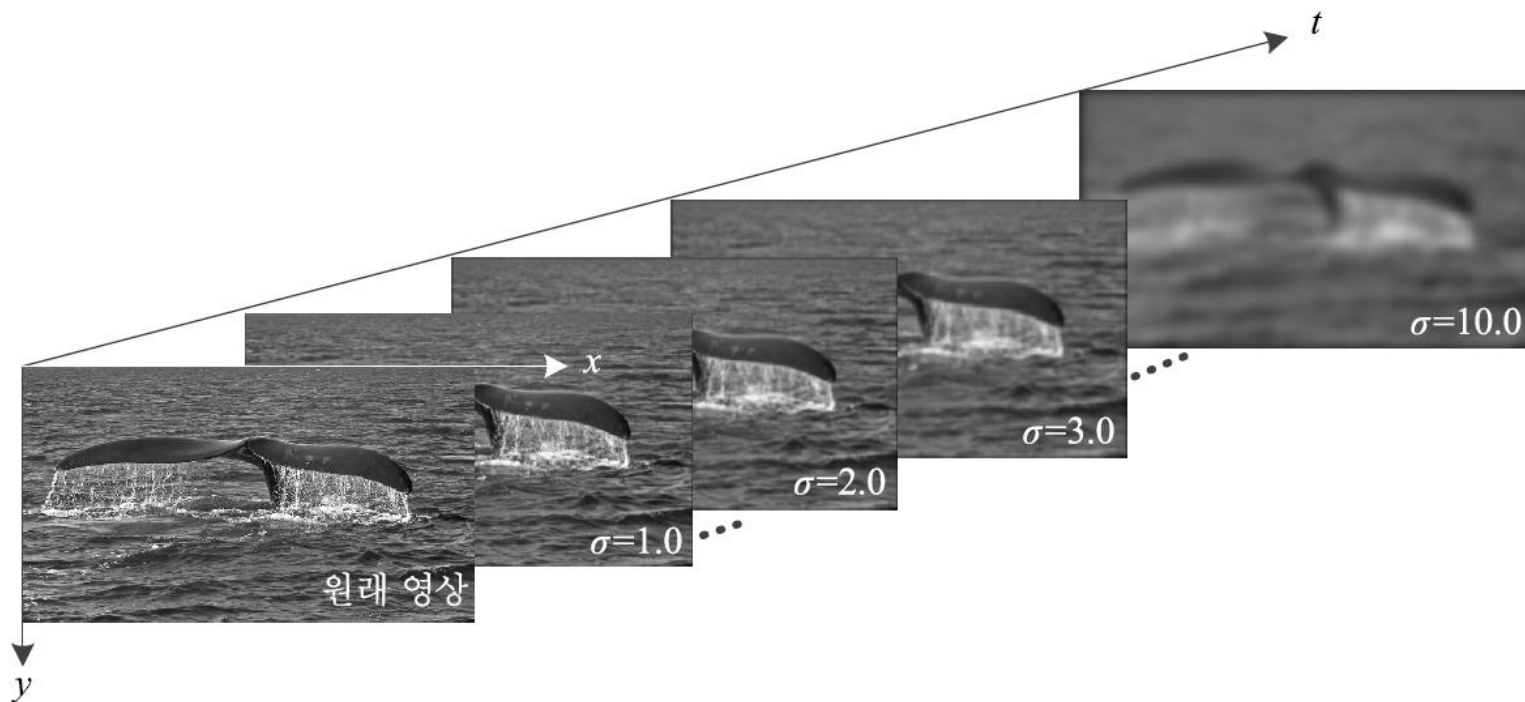


그림 4-14 (y, x, t)로 표현되는 스케일 공간

Scale-invariant feature transform (SIFT) 검색

■ SIFT의 등장

- 1999년 David Lowe 교수의 논문 [Lowe99]
- 2004년 IJCV에 확장된 논문 발표 [Lowe2004]
- 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 ...

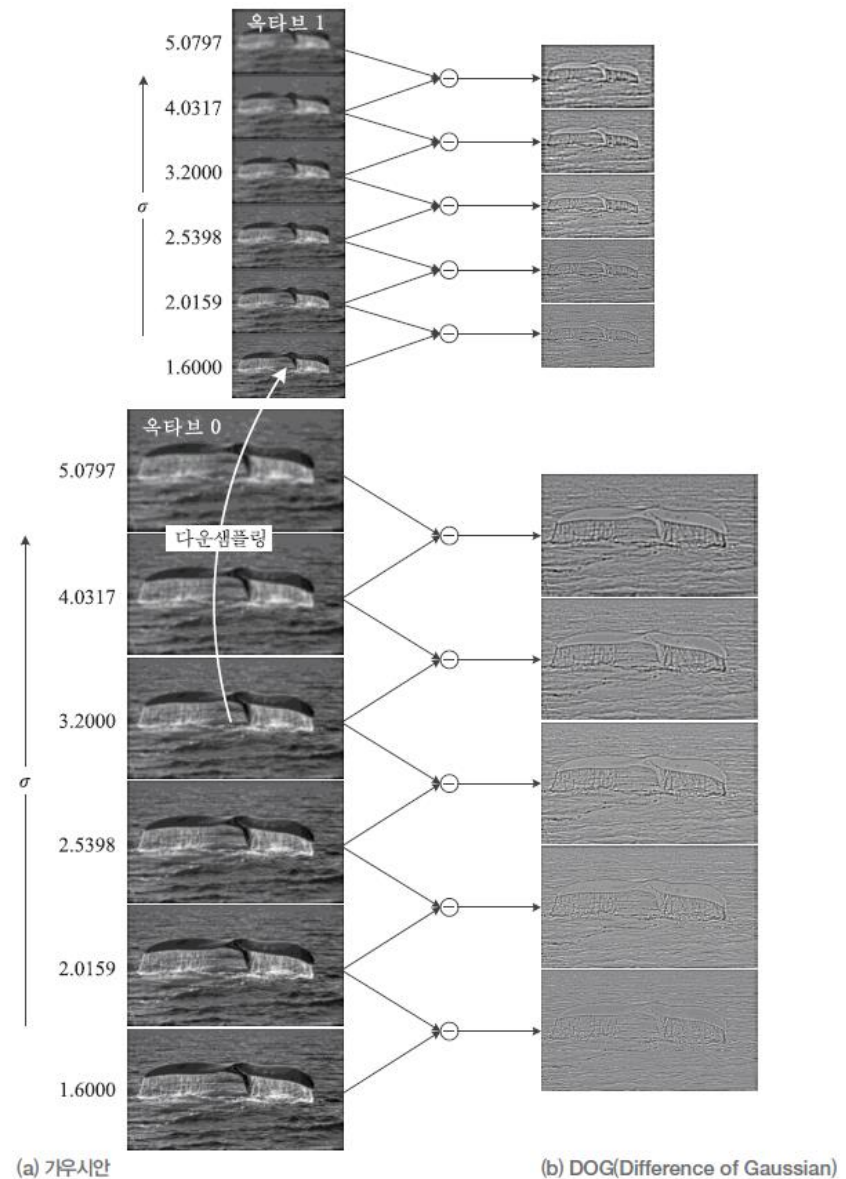
25413회 인용 관련 학술자료 전체 247개의 버전 Web of Science: 8507 인용 저장

← Google scholar

Scale-invariant feature transform (SIFT) 검출

■ SIFT의 스케일 공간

- 피라미드+가우시안 구조
- 각 층은 여섯 영상의 묶음(옥타브)으로 구성
- 옥타브의 영상은 σ_i 로 스무딩
 - $\sigma_{i+1} = k\sigma_i$ ($\sigma_0 = 1.6, k = 2^{1/3}$)



(a) 가우시안

(b) DOG(Difference of Gaussian)

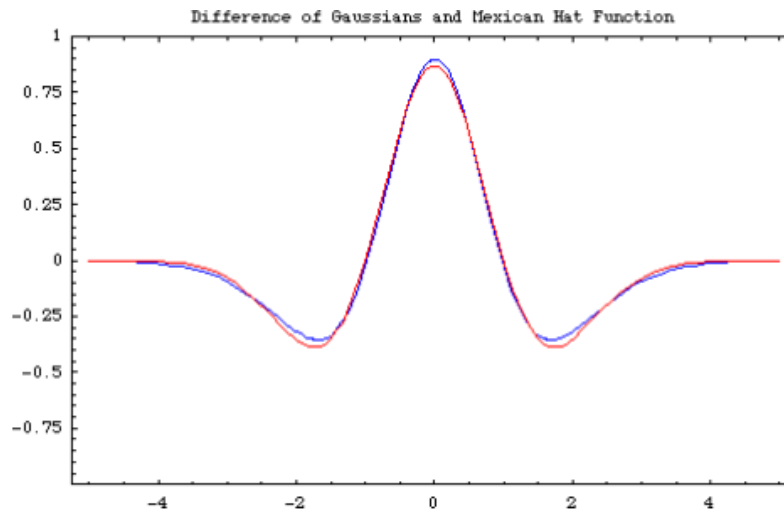
그림 4-16 SIFT가 사용하는 스케일 공간(DOG 피라미드 구조)

Scale-invariant feature transform (SIFT) 검출

■ 정규 라플라시안 맵 구축

- [Mikolajczyk2002a]의 실험 결과에 따르면, 정규 라플라시안이 가장 안정적으로 극점 형성
- 정규 라플라시안과 유사한 DOG 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

Scale-invariant feature transform (SIFT) 검출

■ 특징점 (keypoint) 검출

- 한 옥타브에는 다섯 장의 DOG 영상
- 중간에 끼인 세 장의 DOG 맵에서 **극점** 검출
 - 주위 26개($i+1, i, i-1$ scale에서) 이웃에 대해 최저 또는 최대인 점
- 검출된 극점을 **특징점(keypoint)**라 부름

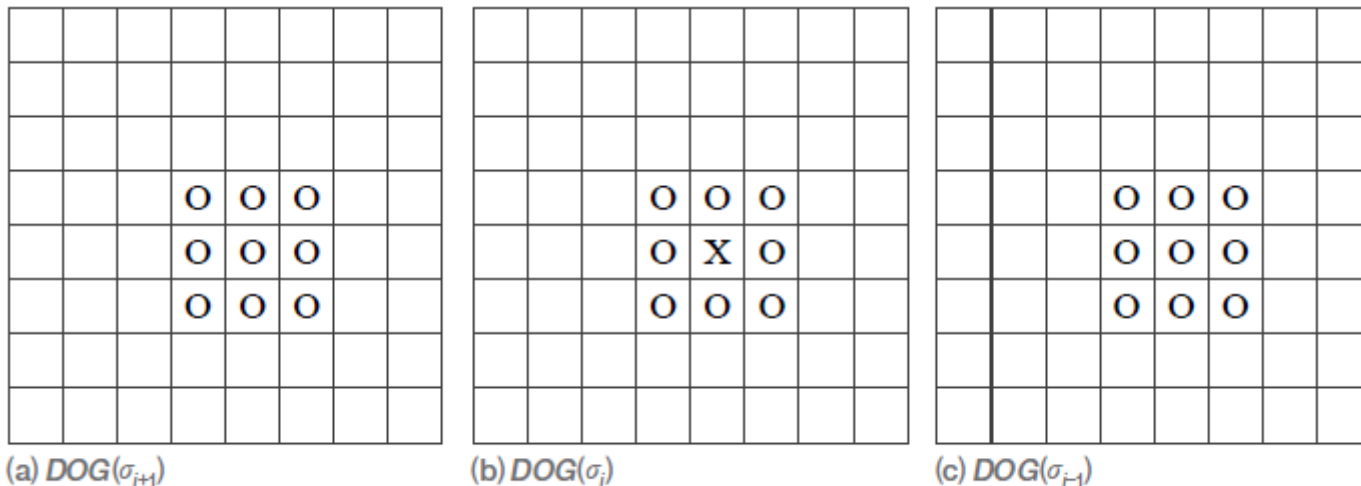


그림 4-17 DOG 맵에서 키포인트(지역 극점) 검출

Scale-invariant feature transform (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

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

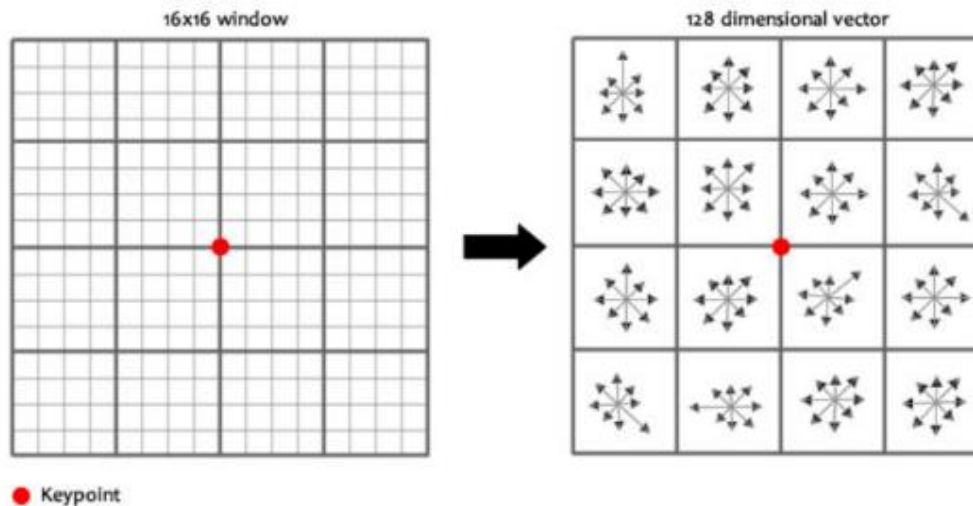
■ 공개 소프트웨어

- David Lowe
- Rob Hess
- Andrea Vedaldi
- OpenCV

Scale-invariant feature transform (SIFT) 특징 - Summary

■ SIFT descriptor (기술자)

- 코너 점 등 식별이 용이한 특징점들을 선택한 후에 각 특징점을 중심으로 한 로컬 패치(local patch)에 대해 아래 그림과 같은 특징 벡터를 추출한 것



- SIFT 특징벡터는 특징점 주변의 영상패치(16x16)를 4 x 4 블록으로 나누고 각 블록에 속한 픽셀들의 gradient 방향과 크기에 대한 히스토그램을 구한 후 이 히스토그램 bin 값들을 일렬로 쭉 연결한 128차원 벡터
- 기본적으로 특징점 주변의 로컬한 gradient 분포특성(밝기 변화의 방향 및 밝기 변화의 급격한 정도)을 표현하는 feature

Speeded up robust features (SURF) 검출

■ 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

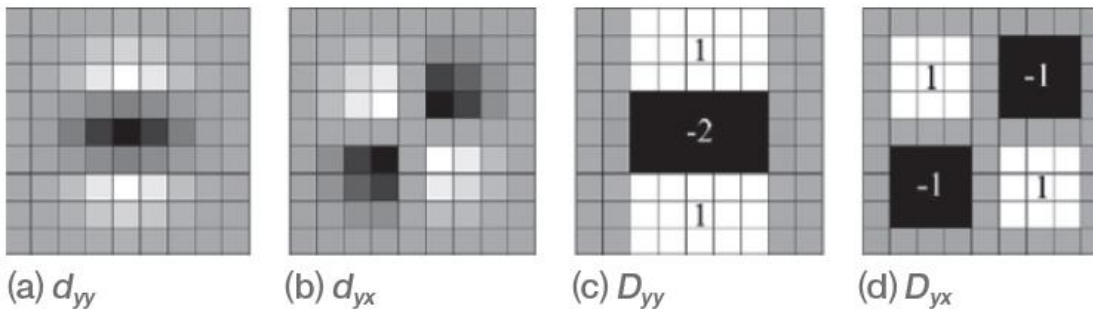


그림 4-18 가우시안을 두 번 미분한 연산자(회색은 0)

Speeded up robust features (SURF) 검출

■ SURF의 스케일 공간

- 원본 영상은 그대로 둔 채 다중 스케일 마스크를 적용

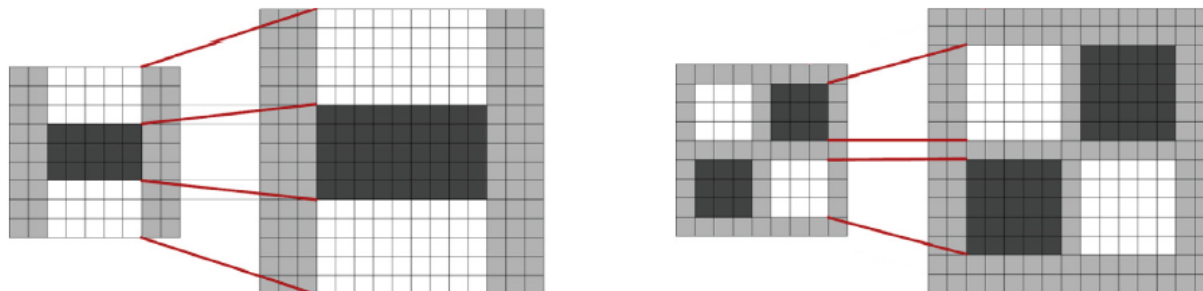


그림 4-19 다음 스케일의 연산자로 확장

■ 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*15에서 시작하고 6의 두 배인 12씩 증가
- 세 번째 옥타브 27*27, 51*51, 75*75, 99*99
 - 두 번째 옥타브의 두 번째 마스크 27*27에서 시작하고 12의 두 배인 24씩 증가

TIP SIFT는 단일 스케일 연산자를 다중 스케일 영상에 적용하는 반면
SURF는 단일 스케일 영상에 다중 스케일 연산자를 적용한다고 볼 수 있다.

Speeded up robust features (SURF) 검출

■ 지역 극점 검출

- 첫 번째 옥타브에서 중간에 끼인 15*15와 21*21에서 지역 극점 검출 (그림 4-17) ($9 \times 9 \rightarrow 15 \times 15 \rightarrow 21 \times 21 \rightarrow 27 \times 27$)
- 두 번째 옥타브에서 중간에 끼인 27*27과 39*39에서 지역 극점 검출 ($15 \times 15 \rightarrow 27 \times 27 \rightarrow 39 \times 39 \rightarrow 51 \times 51$)

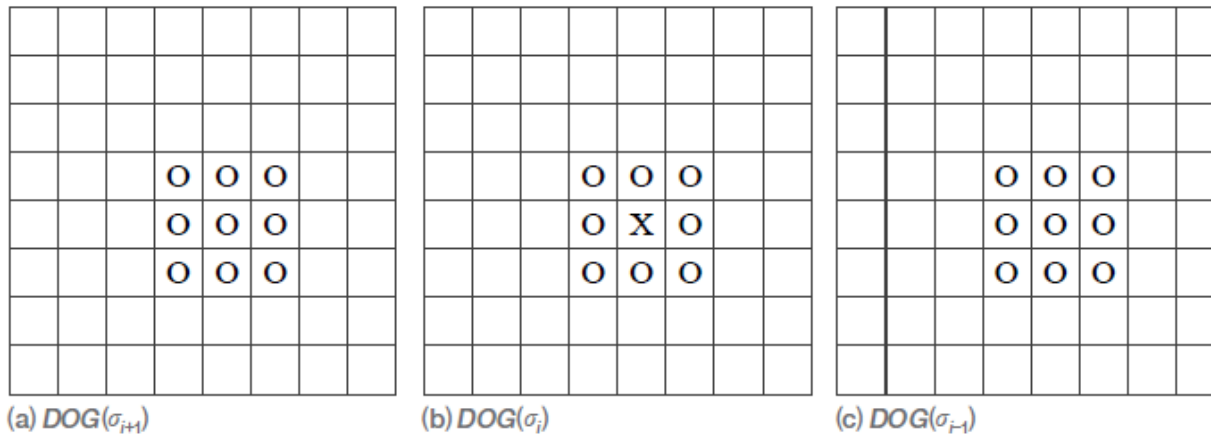


그림 4-17 DOG 맵에서 키포인트(지역 극점) 검출

■ SURF의 속도 개선 보고 [Bay2008]

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

기타: Local features (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

COMPUTER VISION 비전 프로그래밍

Thank you and question?

