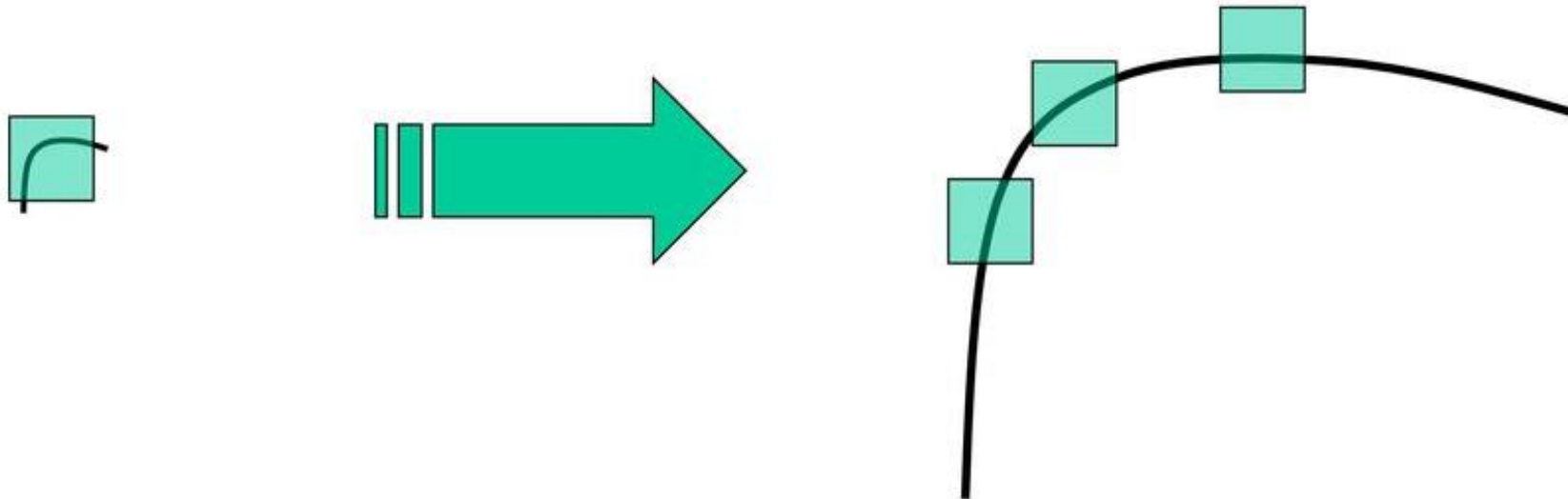


패턴인식 실습 #7

SIFT & SURF

해리스 코너 검출기 (Harris corner detector)의 속성

- 회전 불변(rotation invariant)인가? **Yes**
- 크기 불변(scale invariant)인가? **No**

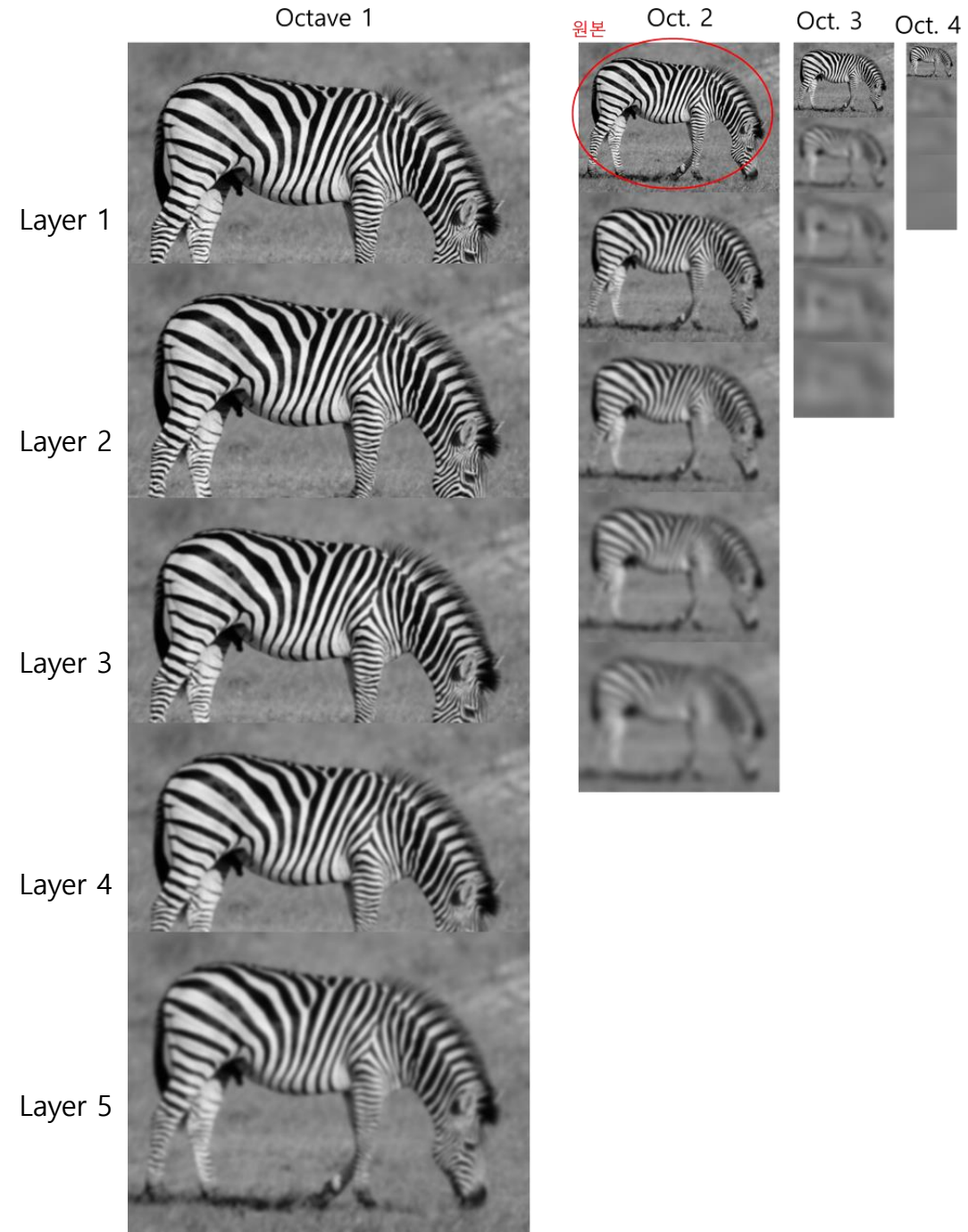


크기 불변 특징 검출 알고리즘

- 1) SIFT (Scale invariant feature transform)
 - 크기, 회전, 조명에 불변한 특징 검출
- 2) SURF (Speeded up robust feature)
 - SIFT에서 처리 속도를 개선한 검출기

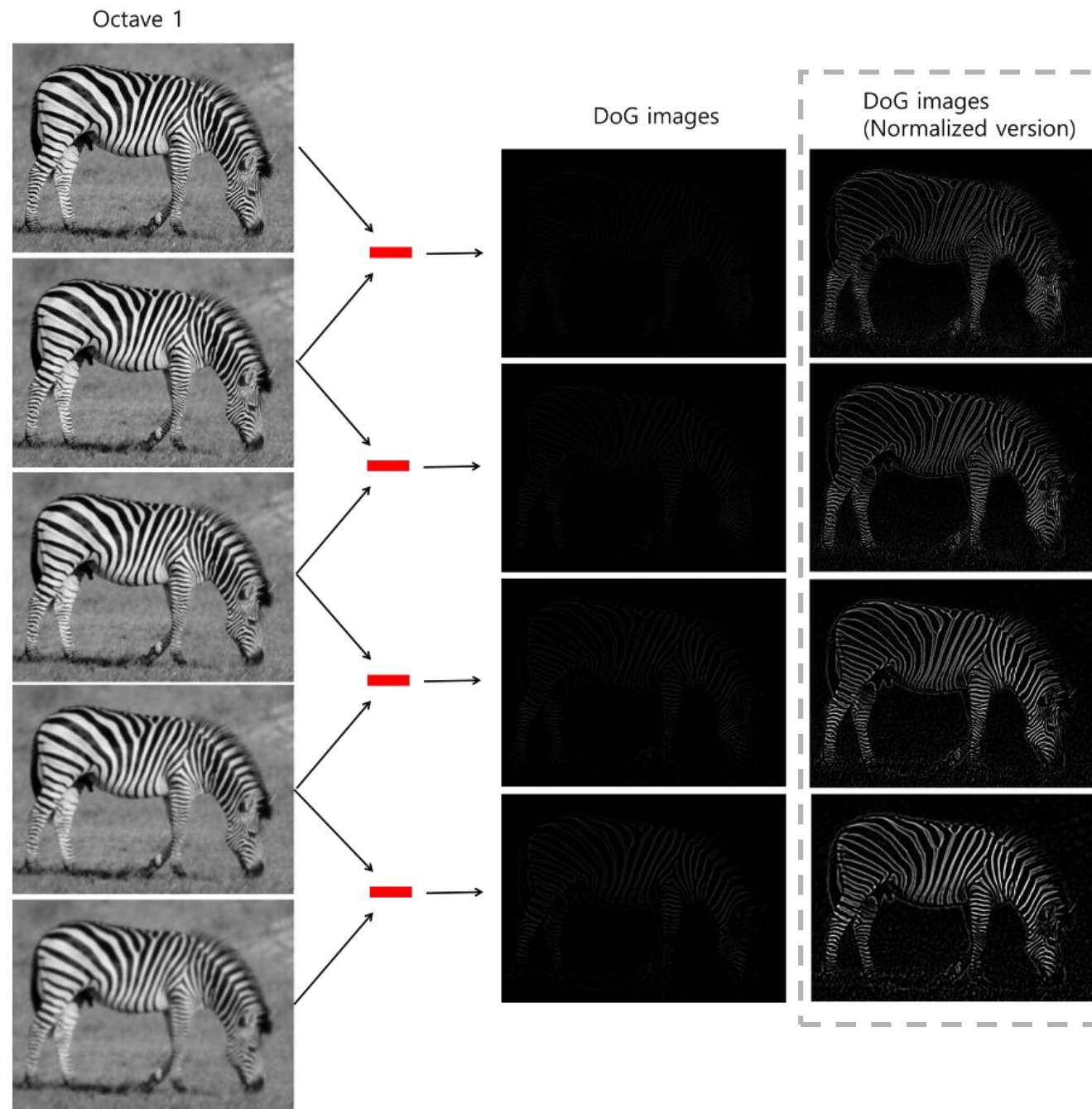
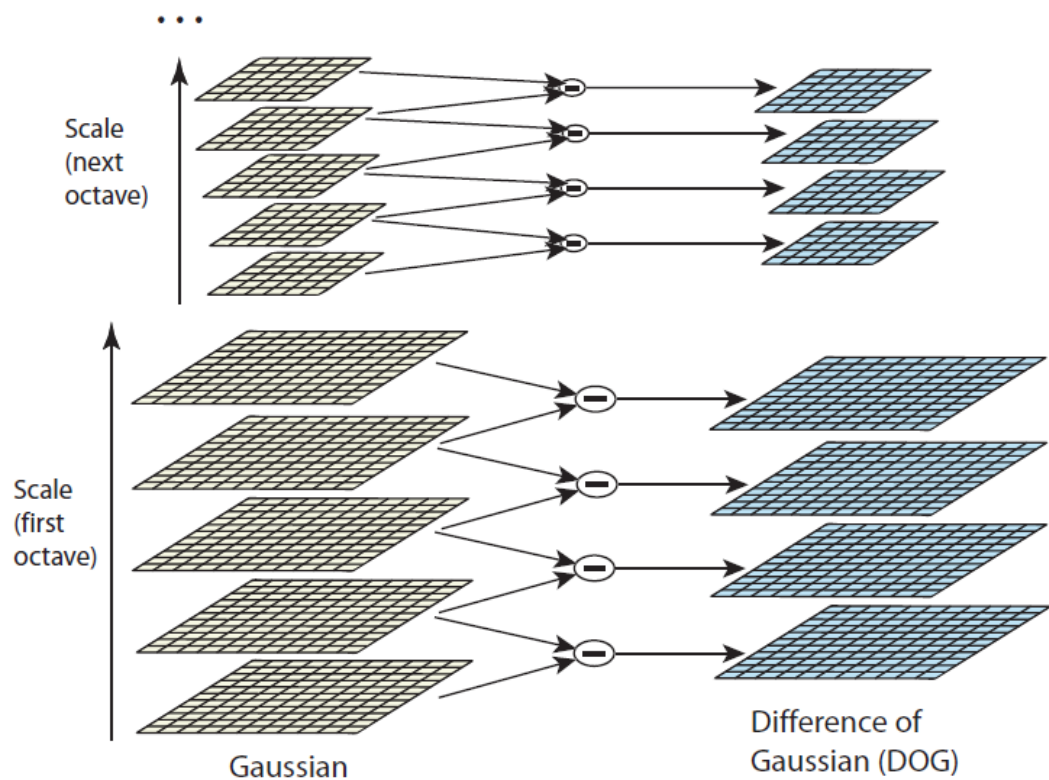
① 스케일 공간 (Scale space) 생성

- 크기 불변 속성 달성



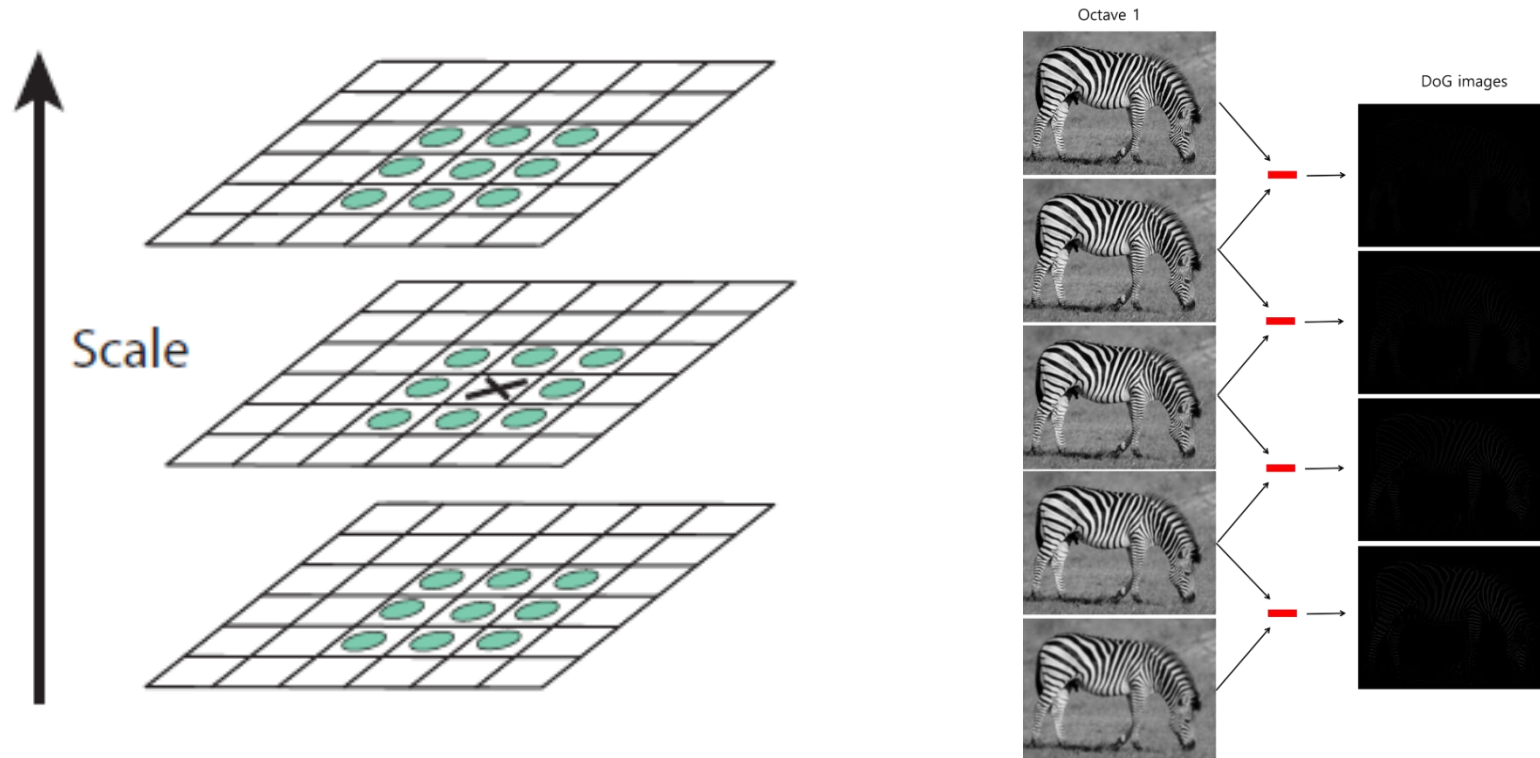
② 키포인트 (keypoint) 검출

- 계산 효율성을 위해
LoG 대신 DoG 사용



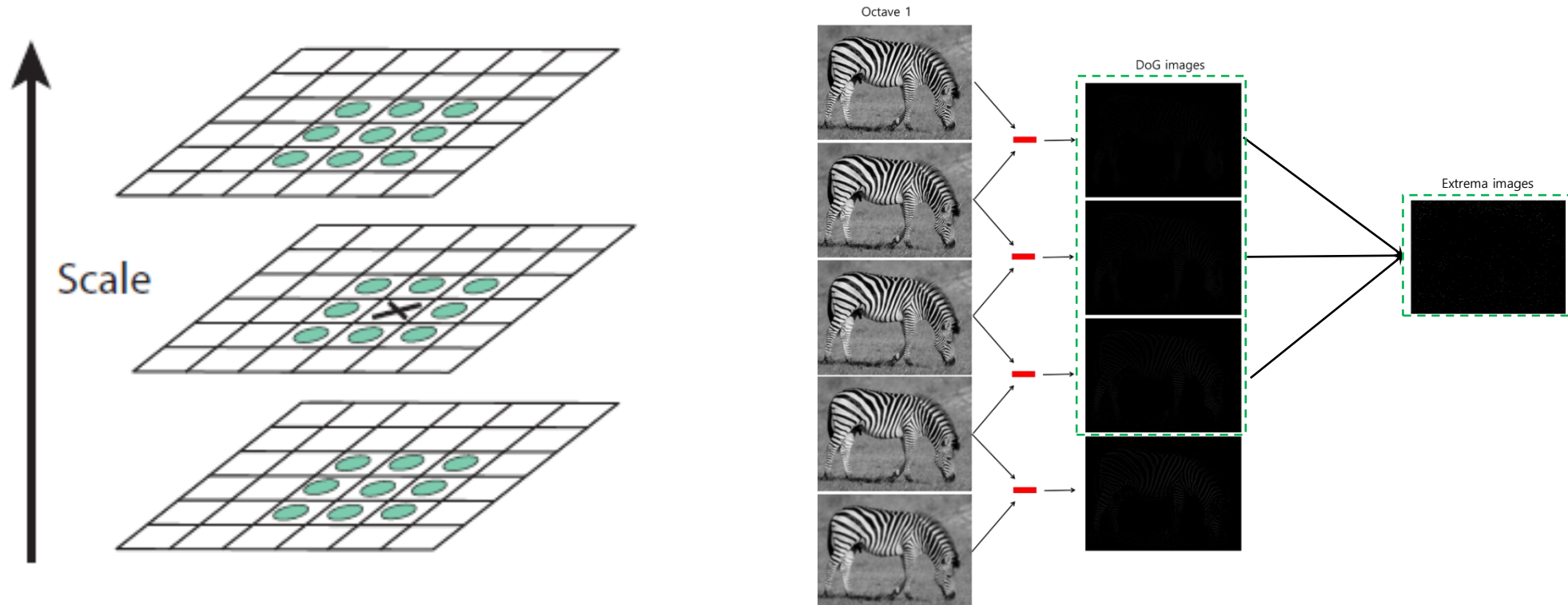
② 키포인트 (keypoint) 검출

- DoG 영상에서 가장 아래, 가장 위 layer를 제외한 세 영상에 대해 모든 픽셀을 순회하면서 극점(extrema) 탐색



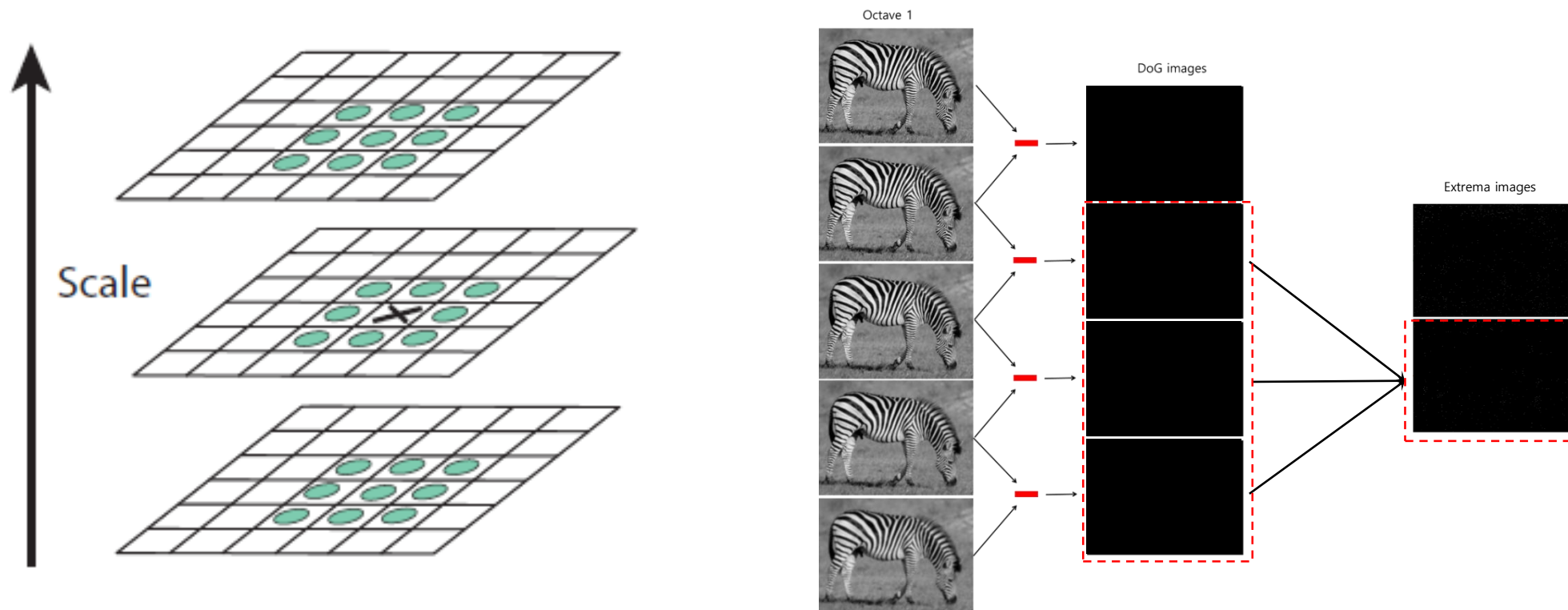
② 키포인트 (keypoint) 검출

- DoG 영상에서 가장 아래, 가장 위 layer를 제외한 세 영상에 대해 모든 픽셀을 순회하면서 극점(extrema) 탐색



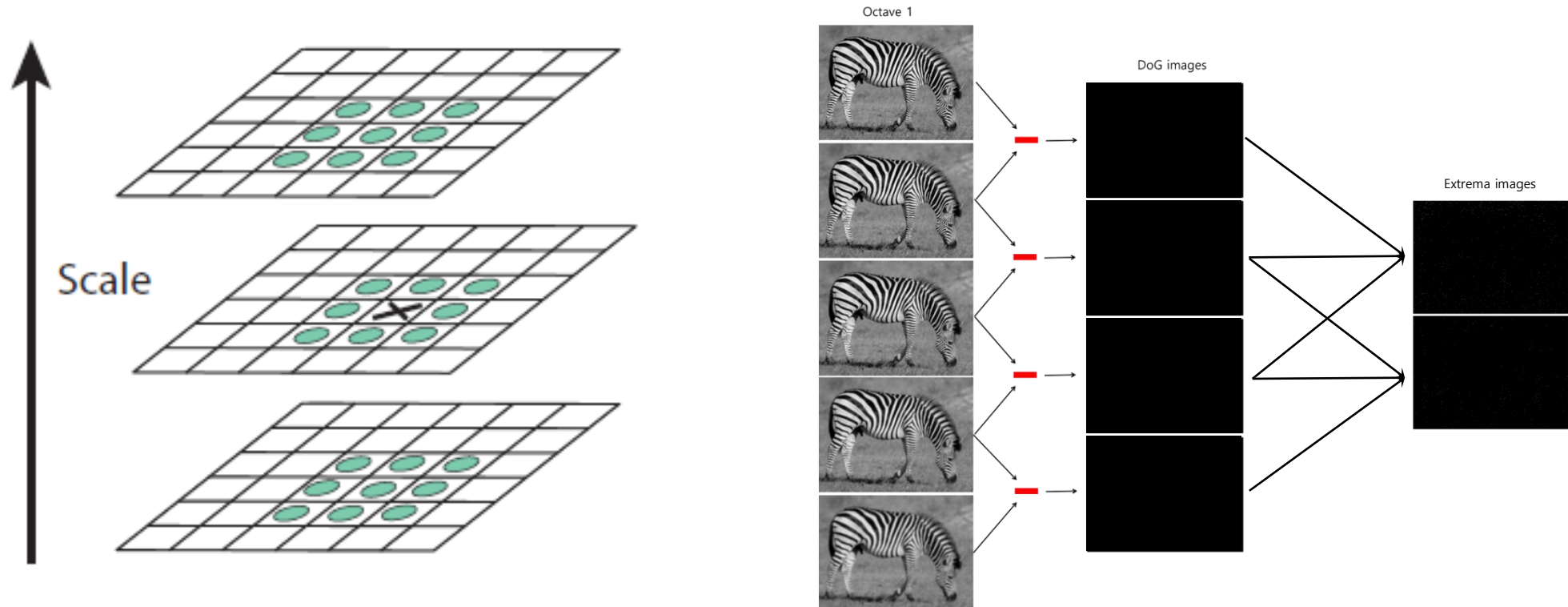
② 키포인트 (keypoint) 검출

- DoG 영상에서 가장 아래, 가장 위 layer를 제외한 세 영상에 대해 모든 픽셀을 순회하면서 극점(extrema) 탐색



② 키포인트 (keypoint) 검출

- DoG 영상에서 가장 아래, 가장 위 layer를 제외한 세 영상에 대해 모든 픽셀을 순회하면서 극점(extrema) 탐색



③ 나쁜 keypoint 제거

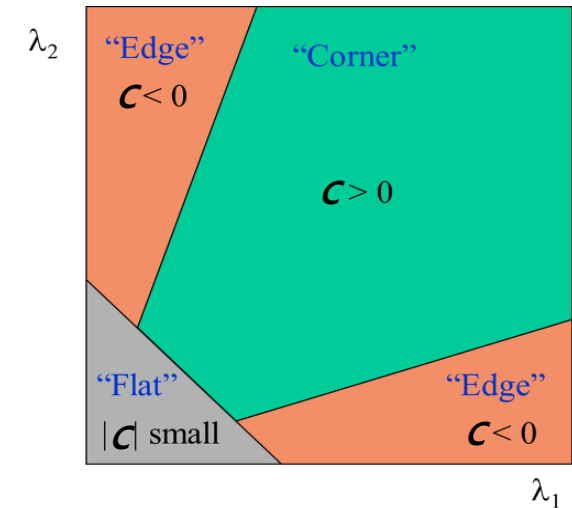
- 극값으로 찾은 keypoint들 중 활용가치가 떨어지는 것 제거
 - a. contrast가 낮거나 → 특정 임계값과 비교
 - b. edge에 해당 → Hessian 행렬의 특징 가능성 값을 기반으로 판단
 - remind)

■ 특징 가능성 값 측정

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

▪ 고유값 계산을 피해 속도 향상

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

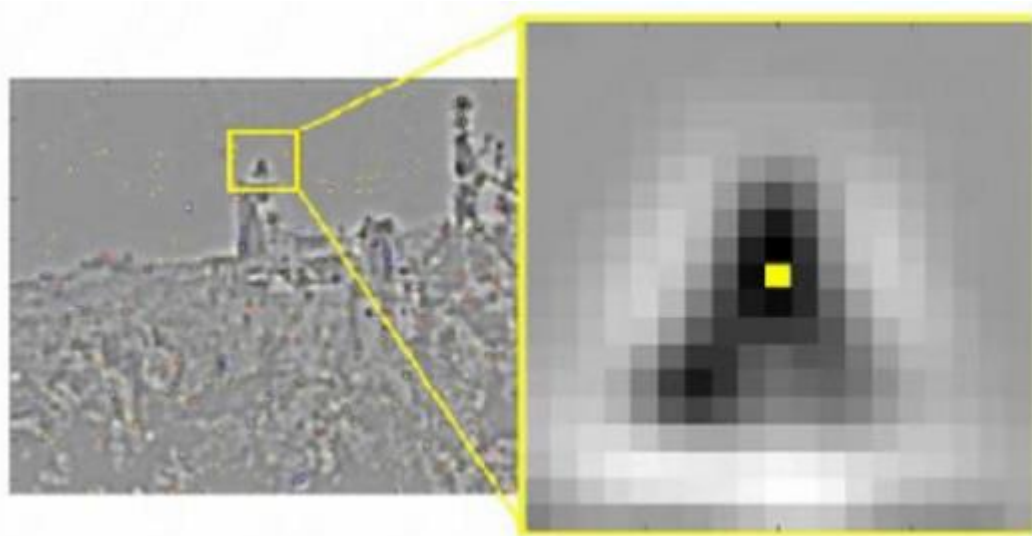


- 최종 keypoints 검출

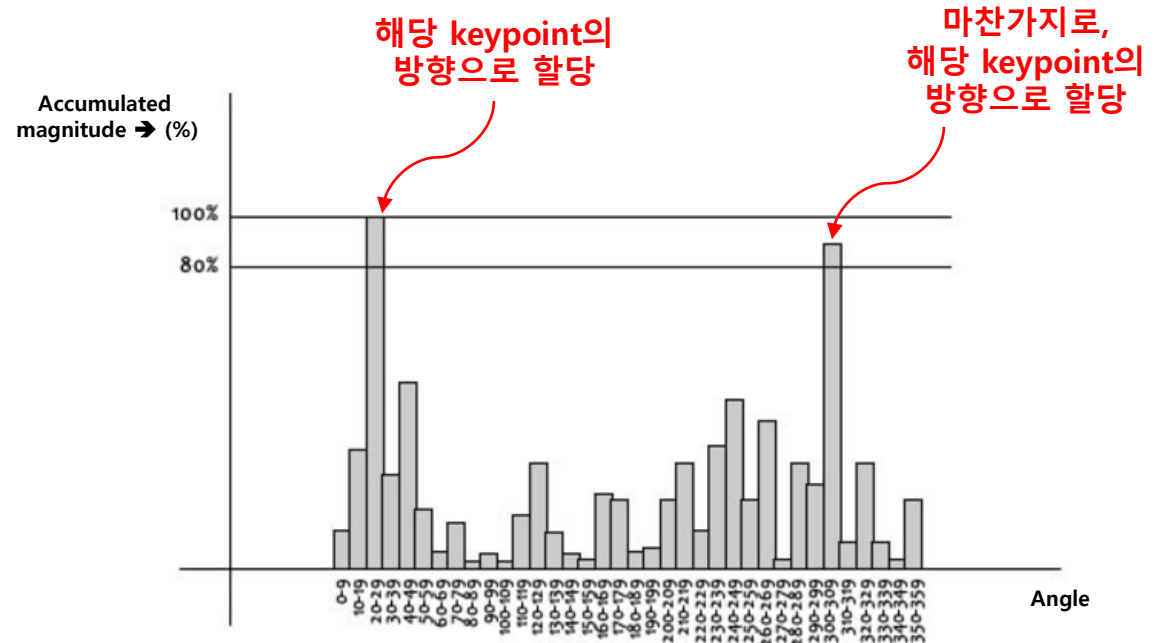
이후 '6장 - 특징 기술'에서 다룰 내용

④ keypoint에 방향(orientation) 할당

- 각 keypoint에 일정 크기의 윈도우를 씌운 후, 픽셀들의 gradient 방향 히스토그램 생성 → 지배적인 방향 할당
- 회전 불변 속성 달성

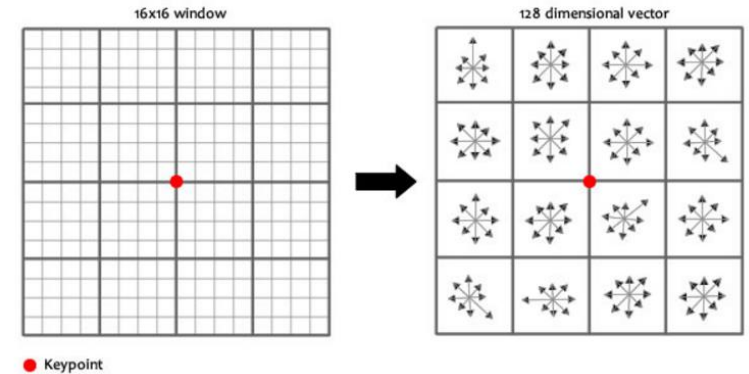


A keypoint



⑤ 특징 기술자(descriptor) 생성

- keypoint 주변 16×16 픽셀들의 gradient 히스토그램 계산
 - sub-block마다 8 bin의 방향 히스토그램 계산
 - $16 \times 8 = 128$ 길이의 방향 히스토그램 생성



• 회전 & 밝기 의존성 해결

- sub-block 방향 히스토그램에서 keypoint의 방향만큼 빼줌
➔ keypoint 방향에 상대적으로 변함 ➔ 일관된 방향 히스토그램 도출
- sub-block 내 밝기 정규화

- 128차원 특징 벡터(feature vector) 생성 ➔ 매칭 등에 활용

SURF (Speeded up robust feature)

- 성능 개선 요인
 - LoG(DoG) 대신 Hessian 행렬 사용
 - 행렬식 근사 계산
 - 적분 영상(integral image) 사용
 - 스케일 공간 생성 시 가우시안 스무딩 대신 근사 연산자 사용
- 800×640 영상에서 처리 속도 [Bay2008]
 - SURF (70ms) < SIFT (400ms) < 해리스 라플라스 (2100ms)

참고

- <https://bskyvision.com/21>

응용

- Object recognition
- Panorama stitching → 지도 앱의 로드뷰 기능
- 등등



과제 #5 – SIFT&SURF 속도 비교

- 임의의 이미지 한 장에 대하여 크기를 축소 및 확대시키면서 SIFT와 SURF의 keypoints 검출 속도를 각각 측정 후 출력하는 코드를 작성하세요. (단위: 초)
- .py 파일만 제출
- 예시)

```
>> (262, 336)
SIFT: 0.023991 sec
SURF: 0.198400 sec

>> (523, 672)
SIFT: 0.092752 sec
SURF: 0.053856 sec

>> (1046, 1344)
SIFT: 0.332113 sec
SURF: 0.209440 sec

>> (5230, 6720)
SIFT: 7.876117 sec
SURF: 3.709044 sec
```