가톨릭대학교
THE CATHOLIC UNIVERSITY OF KOREA

특징 검출, 디스크립터

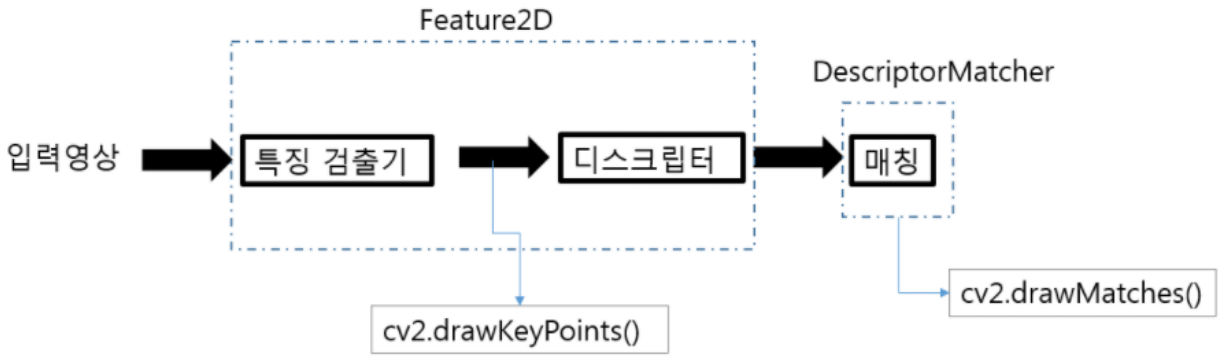
미디어기술콘텐츠학과
강호철

(Review) 특징점 검출

- 특징점(Keypoint 혹은 feature point)
  - 영상에서 특징이 될 만한 지점
  - 중요한 정보를 가지고 있다고 판단
  - 필요성
    - 객체 검출 및 인식
    - 추적
    - 영상 간 매칭

특징 검출기 및 디스크립터

- 특징 검출기
  - 영상에서 관심 있는 특징점(keypoint) 검출
    - 에지, 코너, 영역 등
    - KeyPoint 클래스 객체의 리스트로 반환
- 디스크립터
  - 검출된 특징점 주위의 밝기, 색상, 그래디언트 방향 등 매칭 정보 계산



특징 검출기 및 디스크립터

- KeyPoint Detector
  - 특징점을 검출하는 클래스
  - 코너 같은 부분을 주로 검출
    - FastFeatureDetector
    - MSER
    - SimpleBlobDetector
    - GFTTDetector

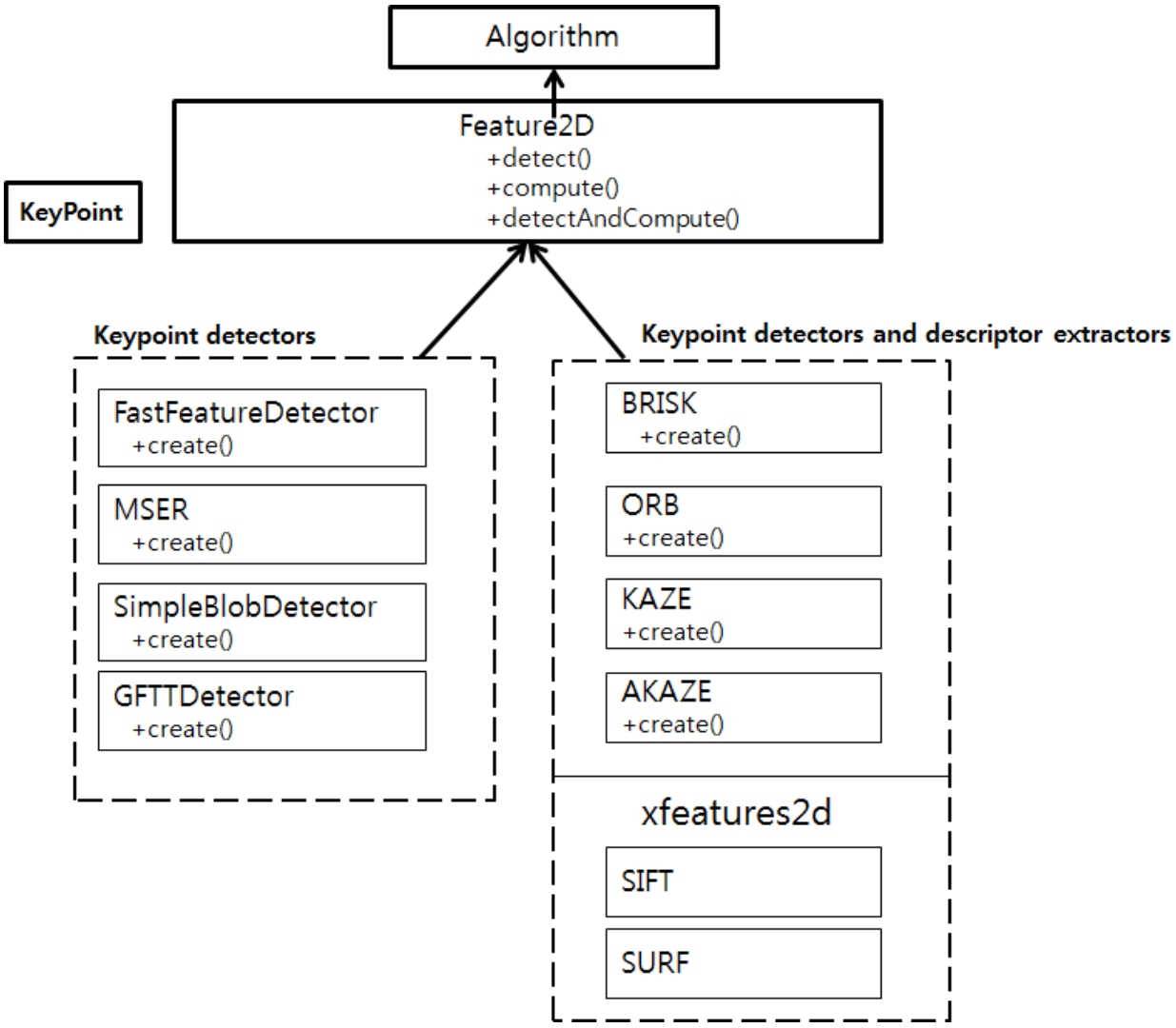
특징 검출기 및 디스크립터

- KeyPoint Detector + Descriptor
  - 특징점과 디스크립터를 같이 추출
  - 디스크립터는 특징점마다 존재함
  - 실제 유사도 판별을 위한 데이터로 활용
    - BRISK
    - ORB
    - KAZE, AKAZE
    - SIFT(xfeatures2d)
    - SURF(xfeatures2d)

특징 검출기 및 디스크립터

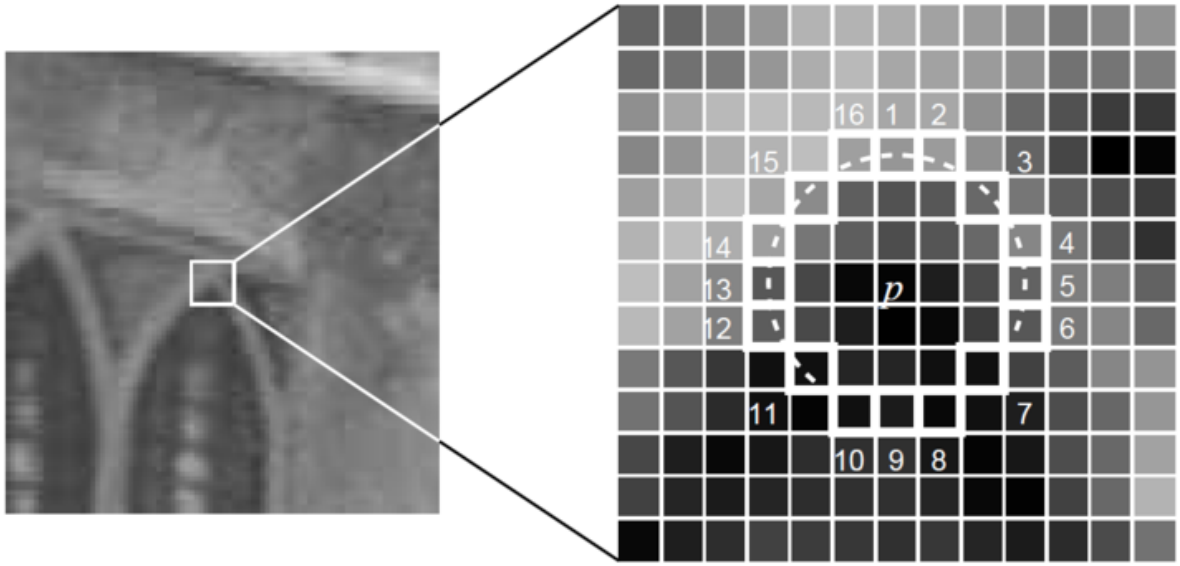
- Descriptor Matcher
  - 두 영상에서 추출된 디스크립터의 유사도 비교
  - 유사한 디스크립터들을 매칭하는 역할
  - 매칭은 KNN 알고리즘 기반
  - 계산 방법은 Brute-fore와 Flann 방법 제공
    - BFMatcher
    - FlannBasedMatcher

주요 특징점 검출기와 디스크립터 클래스



특징 검출기

- FastFeatureDetector
  - FAST 알고리즘 기반
    - 특징점 검출 및 디스크립터 생성



$$S_{p\to x} = \begin{cases} d, & I_{p\to x} \le I_p - t \quad \text{(darker)} \\ s, & I_p - t < I_{p\to x} < I_p + t \quad \text{(similar)} \\ b, & I_p + t \le I_{p\to x} \quad \text{(brighter)} \end{cases}$$

출처: https://wjddyd66.github.io/opencv/OpenCV(8)/#fastfeaturedetector-%ED%8A%B9%EC%A7%95-%EA%B2%80%EC%B6%9C

특징 검출기

- MSER (Maximally Stable Extremal Regions)
  - 그레이스케일 영상 혹은 컬러 영상에서 주변에 비해 밝거나 어두운 영역 중 임계값을 변화시키며 변화율이 작은 영역 검출
  - 주로 connected component로 연결되어 있는 BLOB(Binary Large OBjects) detection에 사용
  - delta로 임계값 지정
    - delta가 클수록 검출되는 영역은 감소
  - 사용 함수
    - cv2.MSER_create(…)
    - cv2.MSER_detectRegion(…)

특징 검출기

- SimpleBlobDetector
  - 원으로 blob 검출
  - 영상을 임계값 범위에서 이진영상 생성
  - 연결된 윤곽선 검출 및 중심점 계산
  - 중심점 사이의 최소 간격에 의해 인접한 중심점 그룹핑
  - 중심점 재 계산 및 그룹의 반지름을 특징 크기로 반환
  - 사용 함수
    - cv2.SimpleBlobDetector_Params()
    - cv2.SimpleBlobDetector_create(…)

특징 검출기

- GFTTDetector
  - goodFeaturesToTrack 함수를 내부적으로 사용하여 특징검출
  - Harris나 MinEigenVal을 이용하여 측정 된 코너점의 최대값과 설정 된 퀄리티 값을 곱하여 이 보다 작은 코너값은 모두 제거
  - 사용 함수
    - cv2.GFTTDetecor_create(…)

디스크립터

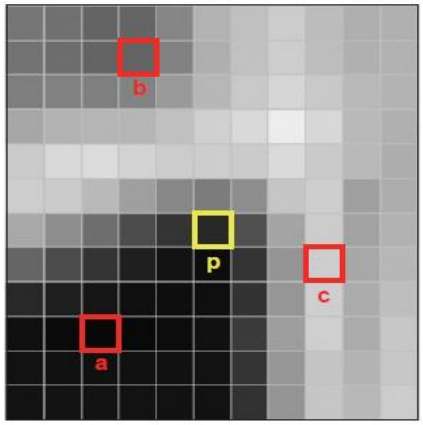
- ORB
  - Oriented BRIEF(Binary Robust Independent Elementary Features)
    - FAST + BRIEF + Harris corner detector
    - SIFT, SURF를 대체하기 위해 개발되었고 속도 빠름
    - 회전 불변성
  - 특징점 검출
    - 피라미드 FAST 혹은 Harris 응답 사용하여 특징 선택
    - 1차 모멘트 이용 방향 검색
  - 디스크립터 계산
    - BRIEF 사용
      - 특징점에 대한 디스크립터로 특징점 검출 방법은 제공하지 않음
      - FAST처럼 어두운 픽셀, 밝은 픽셀, 유사한 픽셀로 분류

디스크립터

- ORB
  - BRIEF
    - 특징점 주변의 픽셀 쌍을 미리 정하고 해당 픽셀 값 크기를 비교하여 0 혹은 1로 특징을 기술
    - 두 점 x와 y에서의 픽셀 값 크기 비교 테스트는 다음과 같이 정의

$$\tau(\mathbf{x},\mathbf{y}) = \begin{cases} 1 & I(\mathbf{x}) < I(\mathbf{y}) \text{일 때} \\ 0 & \text{그 외} \end{cases}$$

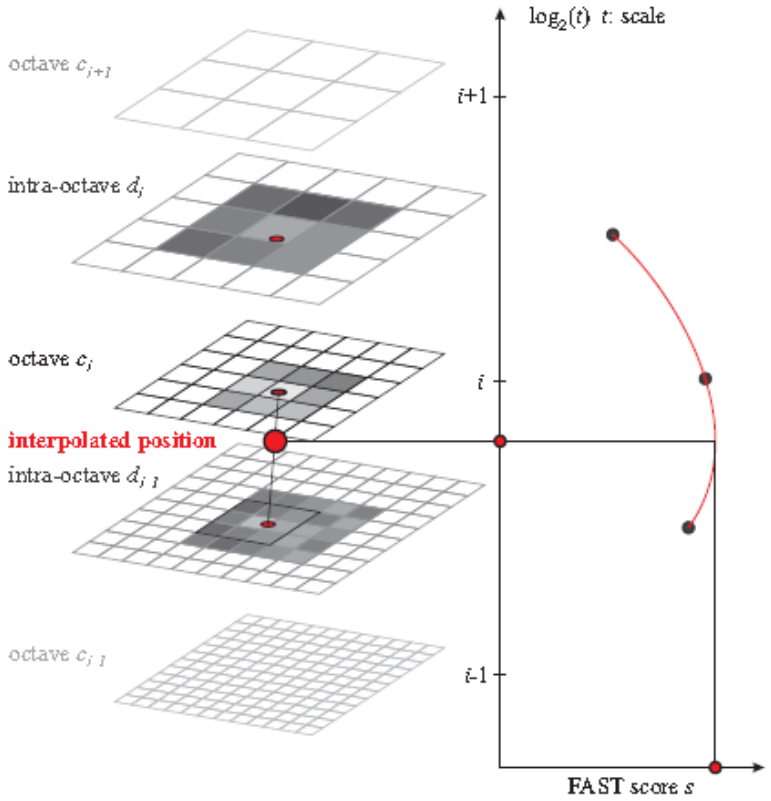
    - 특징점 p주변에 a, b, c 점 미리 정의하고 $\tau(\mathbf{a},\mathbf{b})$, $\tau(\mathbf{b},\mathbf{c})$, $\tau(\mathbf{c},\mathbf{a})$ 를 구하면? 110(2)



  - 사용 함수
    - cv2.ORB_create(…)

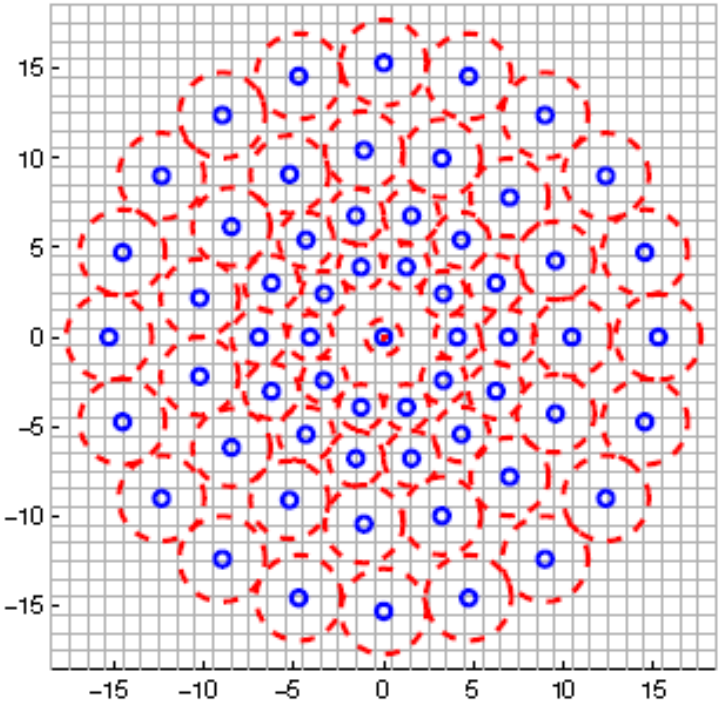
디스크립터

- BRISK
  - 특징점 검출
    - FAST 혹은 AGIST 피라미드 기반 특징점 검출



디스크립터

- BRISK
  - 디스크립터 계산
    - 특징점 근처에서 동심원 기반 샘플링 패턴을 이용하여 이진 디스크립터 계산



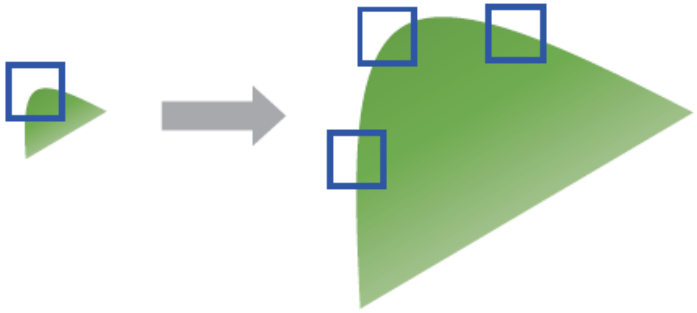
N = 60
Descriptor = 512 bit

디스크립터

- KAZE, AKAZE
  - 기존의 스케일 공간 특징점 검출 방법의 한계
    - 가우시안 피라미드로 인한 특징점 약화
    - 비선형 확신 필터링 사용 (nonlinear diffusion filtering)
  - 물체 경계 유지
  - 특징점 검출 정확도 높임
  - 속도는 SIFT와 비슷하고 SURF보다 느림
  - AKAZE는 FED(fast explicit diffusion)로 비선형 공간에서 피라미드 구축 시 속도 개선
  - 디스크립터: 64비트 정규화 벡터

디스크립터

- SIFT
  - Harris corner detection의 한계
    - 코너는 회전 불변의 특징점
    - 스케일 변환에 대응을 못함
      - 크기가 다른 두 객체 영상에서 코너 점을 이용하여 서로 같은 위치를 찾는 것은 한계가 있음



  - 크기 불변 특징 변환 등장
    - Scale Invariant Feature Transform
    - 2004년 캐나다 브리티시 컬럼비아 대학교 Lowe교수 발표

디스크립터

- SIFT
  - 가우시안 피라미드와 DoG 구성
    - 블러링 된 영상으로 스케일 스페이스 구성 – Octave
    - 영상 크기에 따라 여러 옥타브 구성

디스크립터

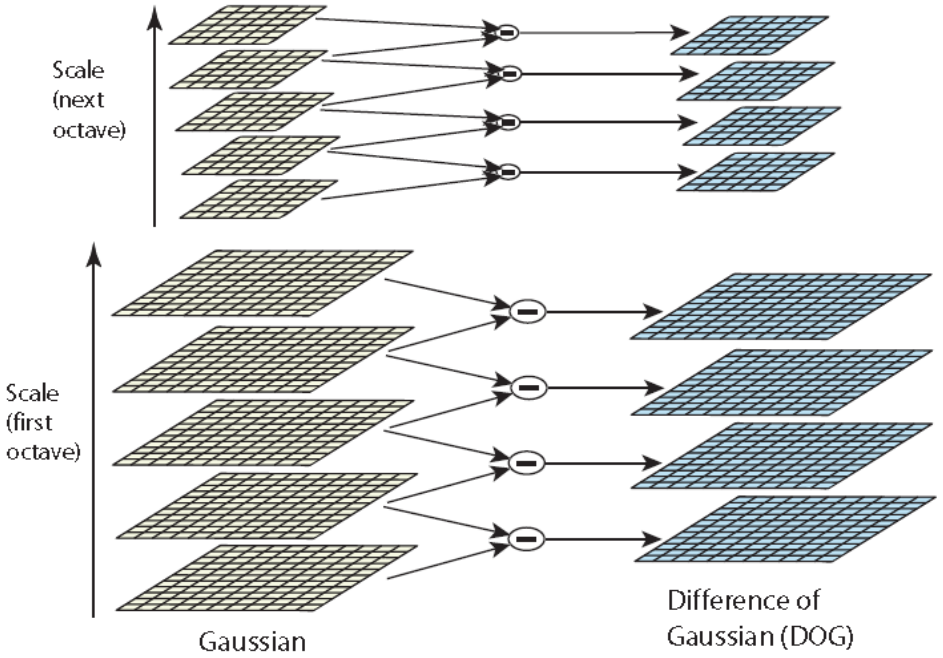
- SIFT
  - DoG (Difference of Gaussian)
    - 인접한 가우시안 블러링의 차영상 이용
    - SIFT는 DoG영상을 고려한 지역 극값 위치 특징점 사용
    - 엣지 성분이 강하거나 명암비가 낮은 지점은 특징점에서 제외

디스크립터

- SIFT
  - 특징점 검출

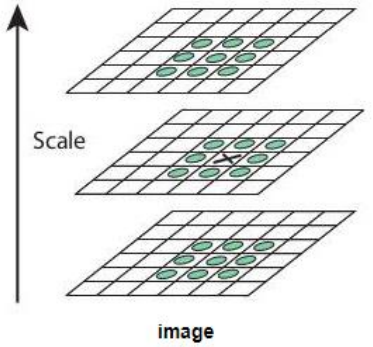
$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * f(x, y) = L(x, y, k\sigma) - L(x, y, \sigma)$$

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



디스크립터

- SIFT
  - 특징점 검출
    - DoG에 모든 점에 대한 극점 검사
      - 26개 이웃 대비 가장 크거나 작으면 keypoint
    - 통과한 점은 후보 특징점이 됨



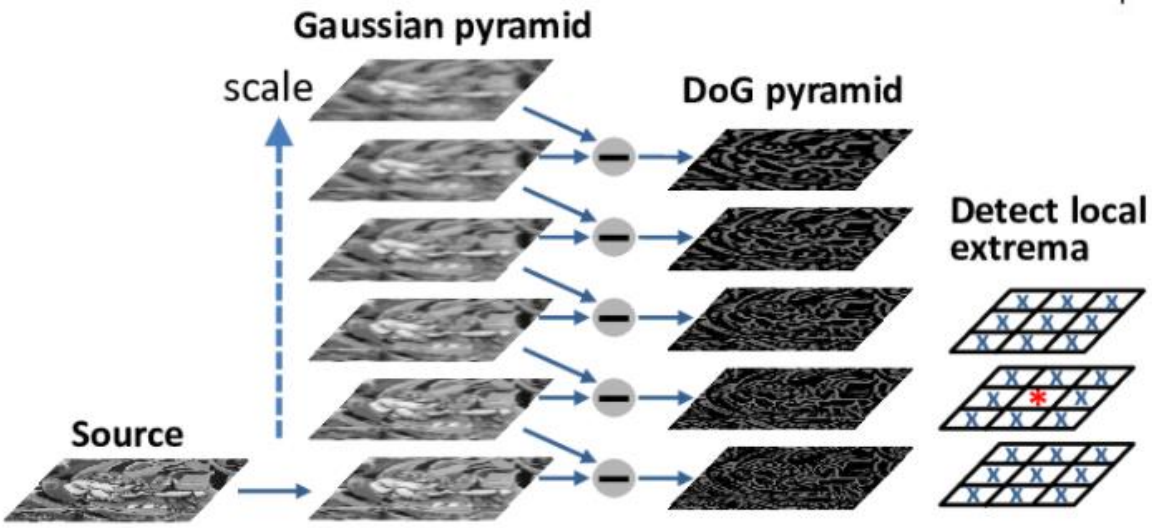
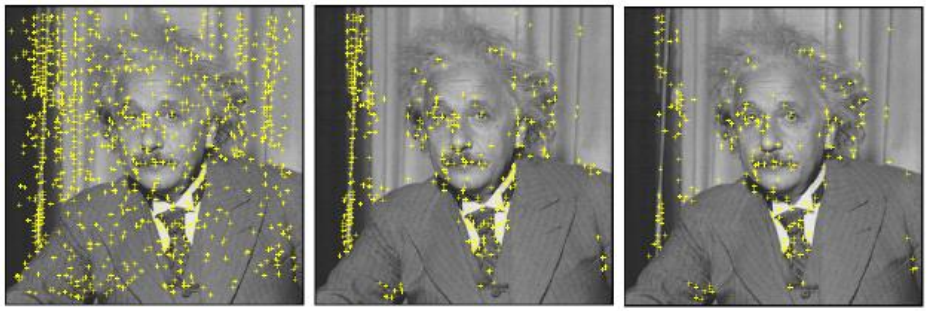


Diagram of SIFT keypoint detection algorithm showing one octave with 6 Gaussian image layers.

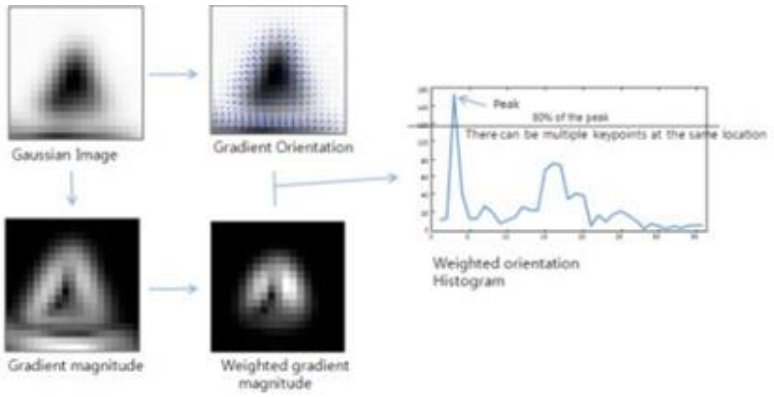
출처: https://www.researchgate.net/figure/Diagram-of-SIFT-keypoint-detection-algorithm-showing-one-octave-with-6-Gaussian-image_fig1_256546531

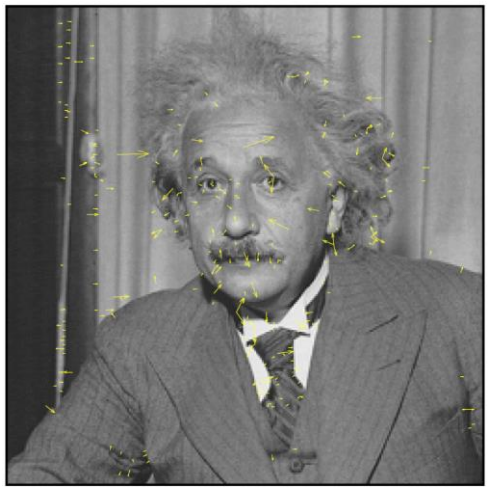
디스크립터

- SIFT
  - 특징점 검출
    - 극점 제거
      - 낮은 대비 극점 제거
      - 엣지에 놓인 극점 제거



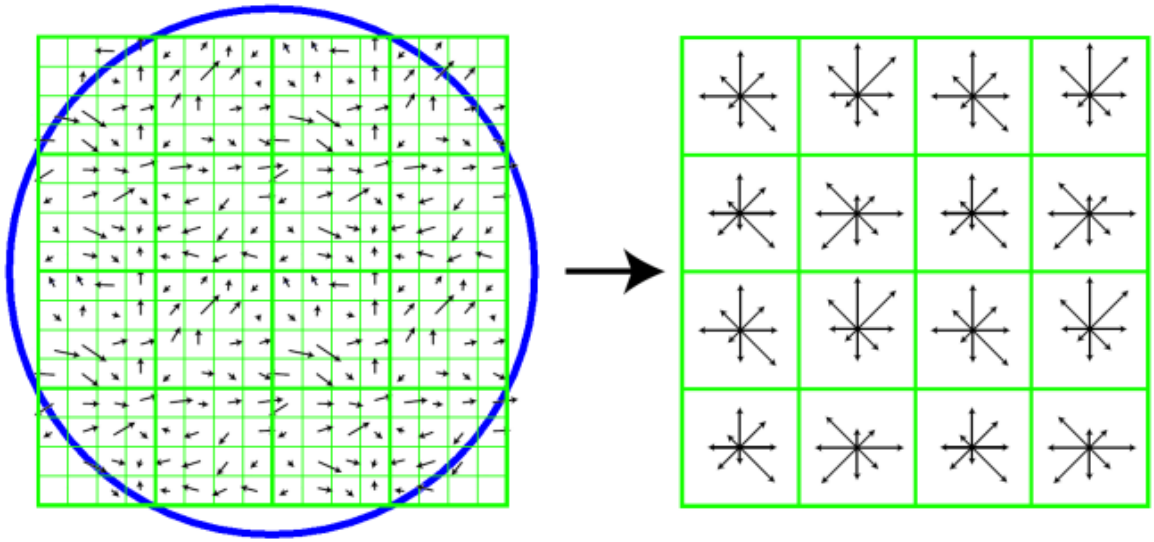
    - 방향 결정
      - 특징점에 대해 가장 강한 방향 결정
      - 특징점은 위치, 스케일, 방향을 고려하여 결정됨





디스크립터

- SIFT
  - 디스크립터 계산
    - 그래디언트 히스토그램
      - 특징점 주위 16 x 16윈도우 설정 후 4 x 4 윈도우로 재구성
      - 16개의 작은 윈도우 x 8개 bin 값 = 특징점 당 128개의 feature vector 생성



디스크립터

- SURF
  - Speed-Up Robust Feature
  - SIFT 알고리즘보다 속도 향상, 강인한 특징 추출
  - 박스 필터와 적분 영상 사용

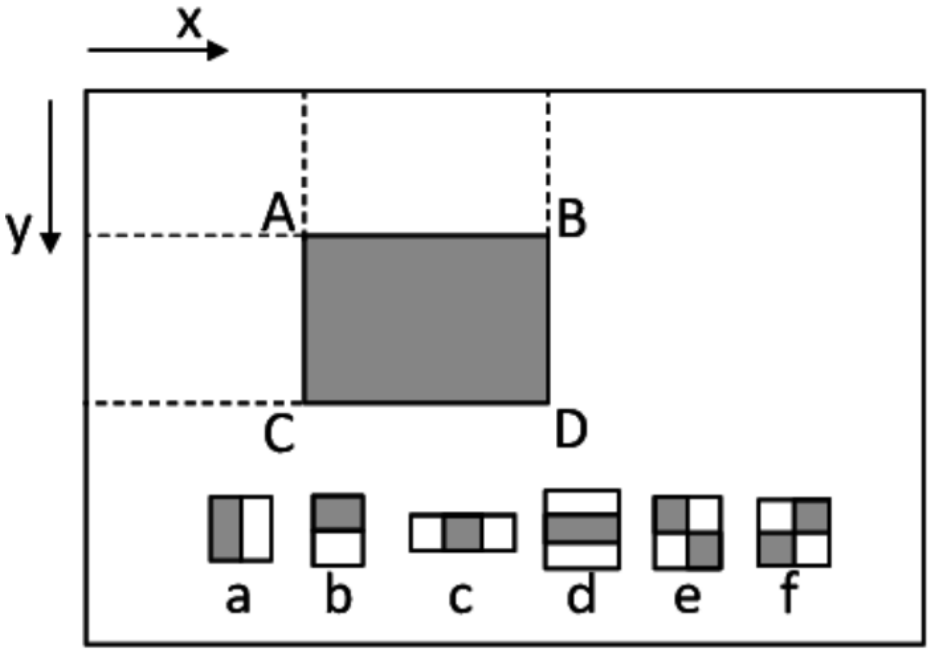


Illustration of the integral image and Haar-like rectangle features (a-f).

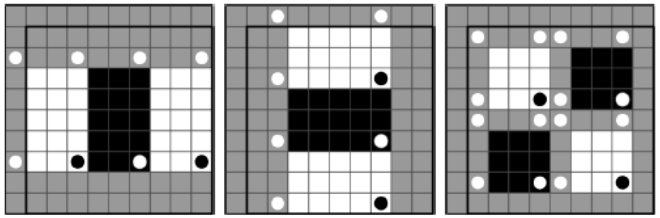
출처: https://www.researchgate.net/figure/Illustration-of-the-integral-image-and-Haar-like-rectangle-features-a-f_fig2_235616690

디스크립터

- SURF
  - 특징점 검출
    - Fast Hessian detector 사용

$$H(x,y,\sigma) = \begin{bmatrix} D_{xx}(x,y,\sigma) & D_{xy}(x,y,\sigma) \\ D_{xy}(x,y,\sigma) & D_{yy}(x,y,\sigma) \end{bmatrix}$$

      - Hessian 행렬 기반 특징점 검출 알고리즘
      - 행렬식이 최대 값인 위치에서 blob 검출 시 사용
      - 박스 필터와 적분 영상을 이용하여 Dxx, Dyy, Dxy 를 구함



    - SIFT는 영상을 scale하여 특징점 추출
    - SURF는 박스 필터 영역을 scale하여 특징점 추출

출처: https://m.blog.naver.com/PostView.nhn?blogId=laonple&logNo=220913997108&proxyReferer=https:%2F%2Fwww.google.com%2F

참고자료

- Python으로 배우는 OpenCV 프로그래밍
  - 김동근 지음
  - 가메 출판사, 2018
- OpenCV4 로 배우는 컴퓨터 비전과 머신러닝
  - 황선규 지음
  - 길벗, 2019