



Feature Points

Feb. 24, 2019

Sunho Kim

Contents



- Feature point
- SIFT
- ORB
- Correspondence matching

Feature point



- Basic concepts

- 정의: 영상 내에서 지역적 독특함을 갖는 점
 - 물체의 크기, 위치, 카메라의 시점, 조명 등이 변해도 식별할 수 있어야 한다.
- 활용 가능 분야: recognition, detection, tracking, classification ...
- 속성
 - 반복성: 동일 영역이 다른 영상에서도 발견될 수 있다.
 - 변별성: 다른 영역에서의 특징점은 다른 표현을 갖는다.
 - 효율성: 영상 내에서 특징점 갯수는 픽셀 수보다 작아야 한다.
 - 지역성: 특징점은 영상의 지역적 영역에만 관련된다.
- 구성
 - Keypoint: 특징점의 위치 정보
 - Descriptor: 주변 픽셀에 대한 정보를 설명하는 벡터



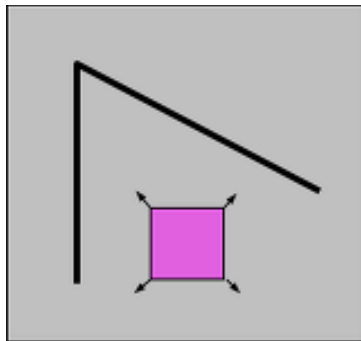
Feature point

- Harris corner detector

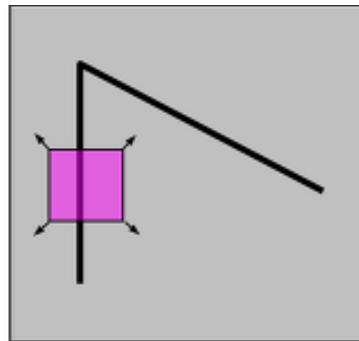
- 영상 내 작은 윈도우 내에서 모든 방향으로의 영상 변화가 큰 부분을 코너로 간주
- 픽셀의 영상변화량 값이 지역적으로 극대화되는 지점을 코너점으로 삼았다

- Shi-Tomasi

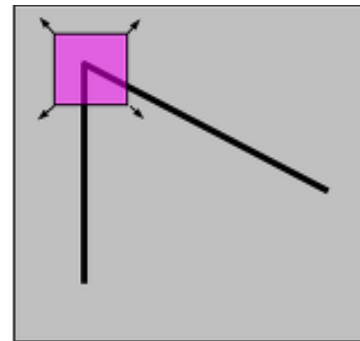
- 좋은 특징점은 추적 알고리즘에 최적화될 수 있어야 한다!
- 단순 평행이동과 affine 변화를 모두 고려한다.
- 두 고유값 중 최소가 되는 값만을 고려한다.



“flat” region:
no change in all
directions

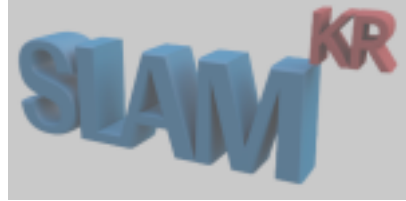


“edge”:
no change along the
edge direction



“corner”:
significant change in
all directions

Feature point



- Harris corner detector

$$I(x + \Delta x, y + \Delta y) \approx I(x, y) + I_x(x, y)\Delta x + I_y(x, y)\Delta y$$

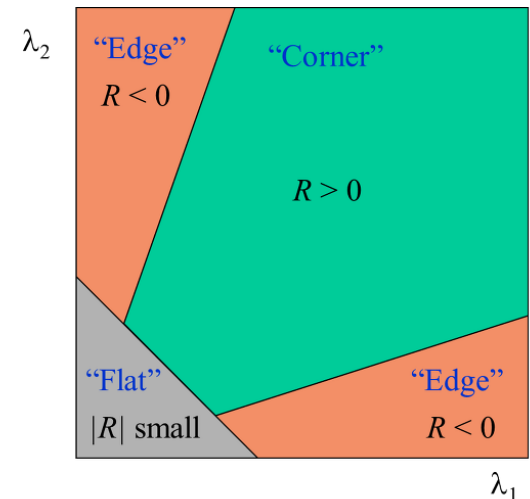
$$f(x, y) = \sum_{(x_k, y_k) \in W} (I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y))^2$$

$$f(x, y) \approx \sum_{(x, y) \in W} (I_x(x, y)\Delta x + I_y(x, y)\Delta y)^2,$$

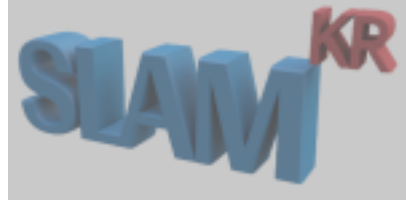
$$f(x, y) \approx \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} M \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix},$$

$$M = \sum_{(x, y) \in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \sum_{(x, y) \in W} I_x^2 & \sum_{(x, y) \in W} I_x I_y \\ \sum_{(x, y) \in W} I_x I_y & \sum_{(x, y) \in W} I_y^2 \end{bmatrix}$$

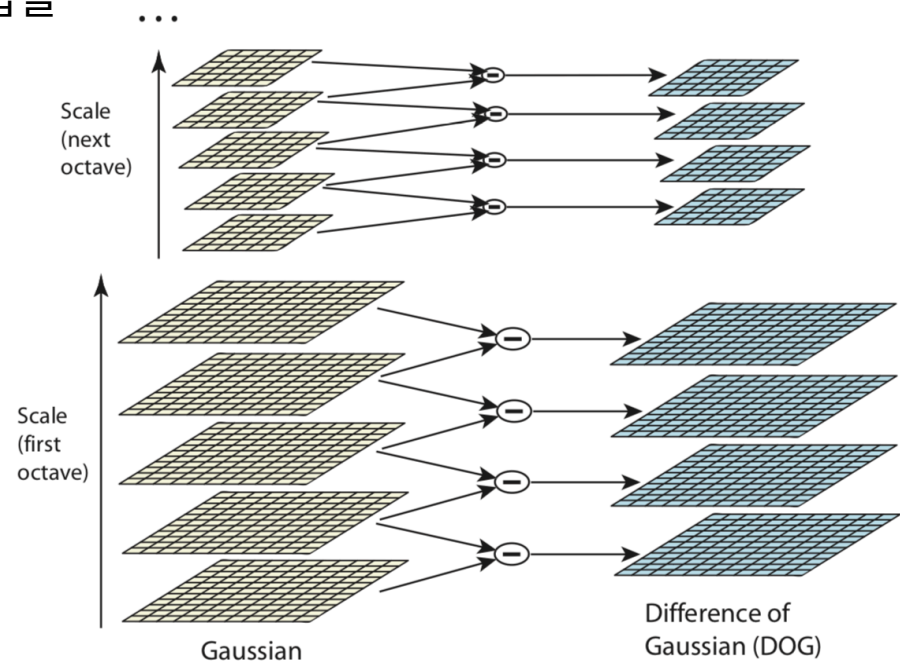
$$R = \det(M) - k(\text{trace}(M))^2 = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$



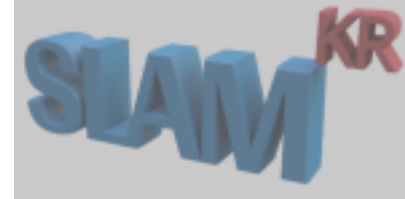
Scale Invariant Feature Transform(SIFT)



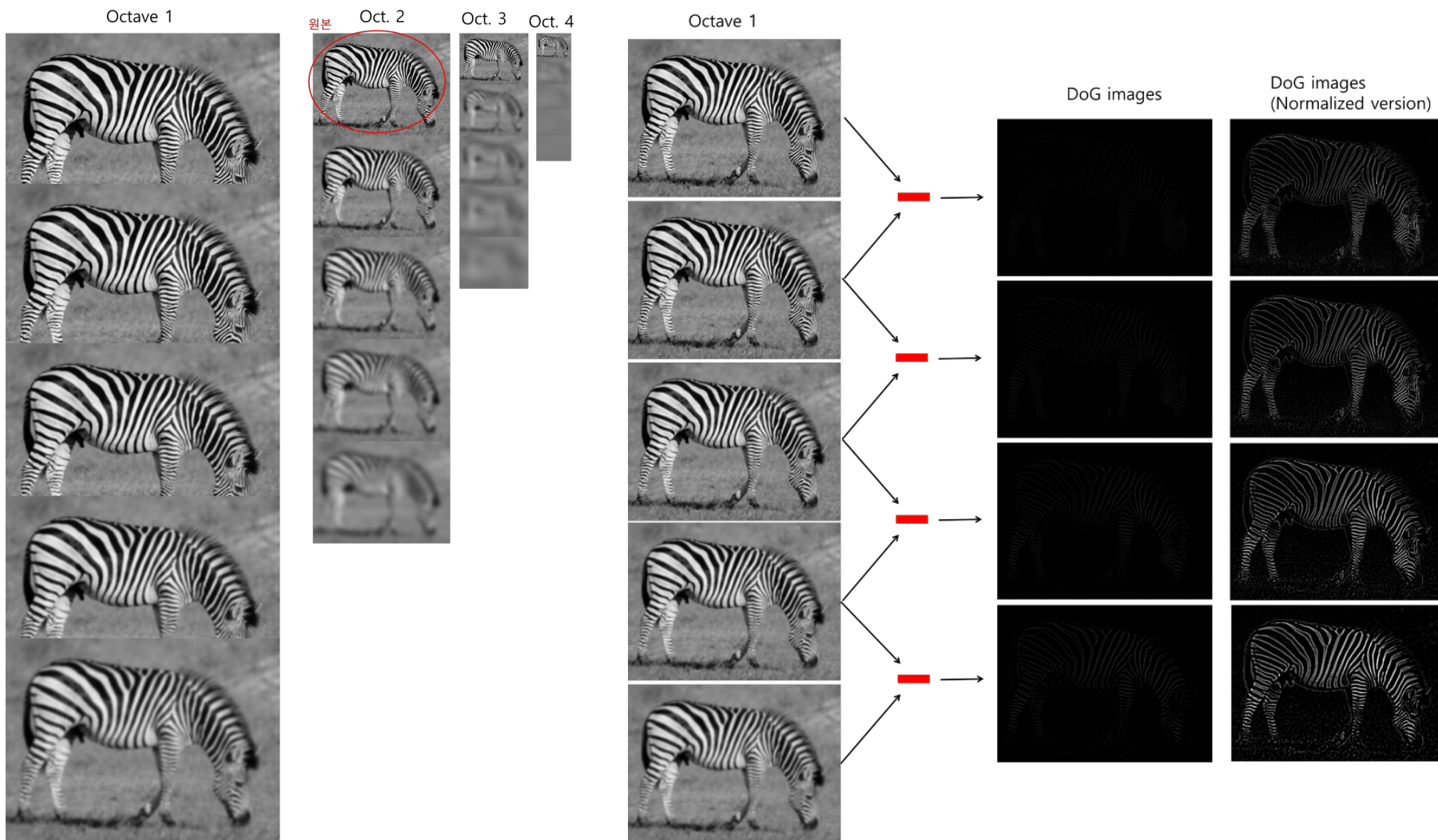
- 크기, 방향 및 조명 변화 등에 강인한 특징점이 필요하다!
 - 식별이 가능한 특징점을 중심으로 하는 로컬 패치에 대해 특징 벡터를 뽑은 결과
 - 지역적 밝기 변화의 방향과 급격한 정도를 표현한다.
 - 연산량이 많아 실시간 SLAM 구현에는 적합하지 못하다.
(물론 GPU 프로그래밍으로 보완 가능)
- Laplacian of Gaussian 방법의 많은 연산량을 보완하기 위해 Difference of Gaussian 방법을 사용했다.
- Laplacian: 이미지의 밝기 변화에 대한 2차 미분값



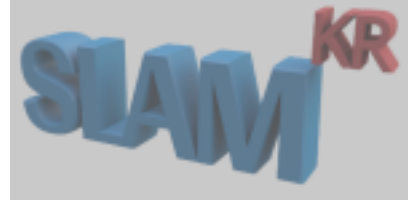
Scale Invariant Feature Transform(SIFT)



- Difference of Gaussian(DoG)

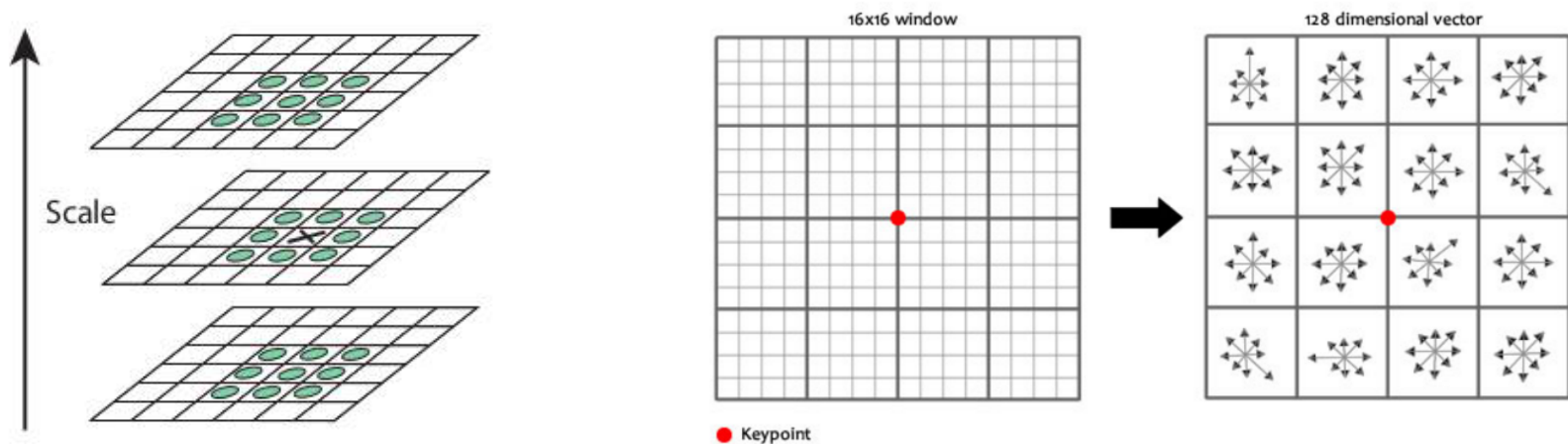


Scale Invariant Feature Transform(SIFT)



- Process

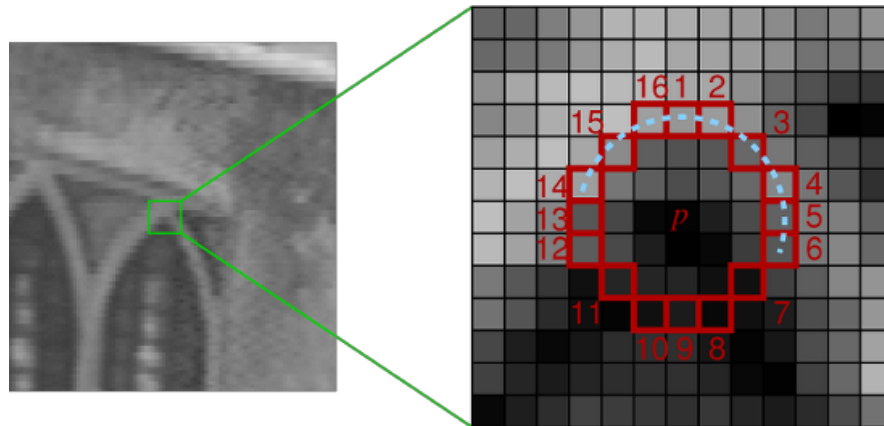
- Scale-space extrema detection: 모든 scale에 대해 difference of Gaussian을 기반으로 크기와 방향 변화에 강건한 잠재적 특징점 후보군들을 추출한다.
- Keypoint localization: 각 extrema에 대해 low-contrast 포인트들과 edge 영역에 있는 포인트들을 제거한다.
- Orientation assignment: 각 점들에 대해 local gradient를 기반으로 방향성 할당
- Keypoint descriptor: 16x16 크기의 블록 → 4x4 sub-block 단위로 묶어내기
→ 각 sub-block에서 방향성에 대한 8bin 히스토그램 생성
→ 최종적으로 128 dimensional vector 생성



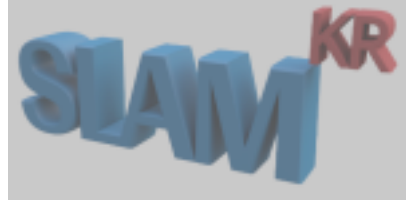
Oriented-FAST and Rotated-BRIEF(ORB)



- Features from Accelerated Segment Test(FAST)
 - Grayscale의 명백한 밝기 변화를 기반으로 빠르게 특징점을 검출하는 방법
 - Process
 - 중심점 p 에서의 밝기를 I_p 로 가정합니다.
 - 임계값 T 를 설정합니다 (예 : I_p 의 20 %).
 - 픽셀 p 를 중심으로 반지름이 3인 원의 원주에서 16개의 픽셀을 선택합니다.
 - 선택된 원의 연속적인 N 지점의 밝기가 $I_p + T$ 보다 크거나, $I_p - T$ 보다 작은 경우, 픽셀 p 는 특징점으로 간주 할 수 있습니다. ($N=9, 11, 12...$)
 - 위의 네 단계를 반복하여 각 픽셀에 대해 동일한 작업을 수행합니다.



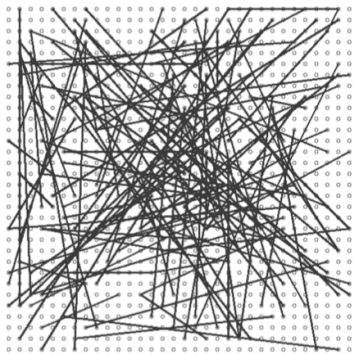
Oriented-FAST and Rotated-BRIEF(ORB)



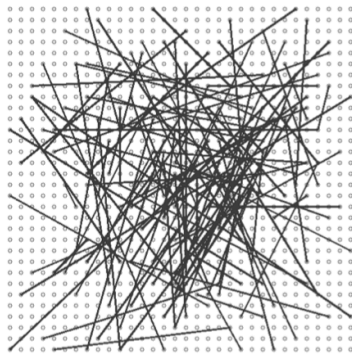
- Binary Robust Independent Elementary Features(BRIEF)
 - Descriptor vector가 이진수로 구성되었다.
 - 패치 내 두 픽셀간의 비교를 통해 벡터에 들어갈 이진수를 결정

$$\tau(p; x, y) := \begin{cases} 1 & \text{if } p(x) < p(y) \\ 0 & \text{otherwise} \end{cases}$$

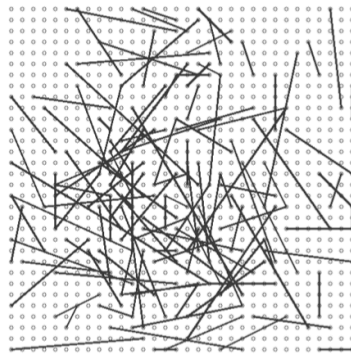
- 임의의 두 픽셀 쌍은 확률 분포를 기준으로 임의로 결정
- 128개의 쌍을 비교하여 128bit binary descriptor 생성



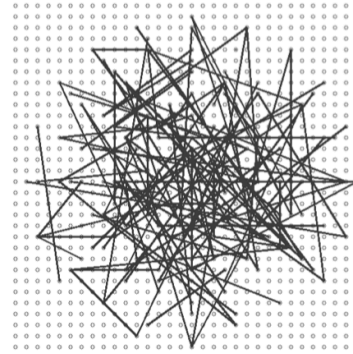
G I



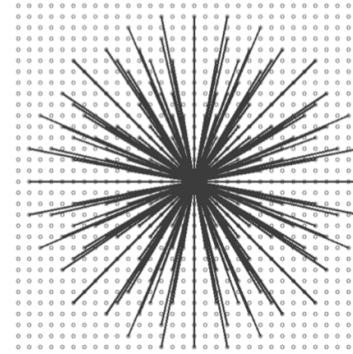
G II



G III



G IV



G V

Oriented-FAST and Rotated-BRIEF(ORB)



- Oriented-FAST

- FAST의 약점은 방향성과 스케일을 갖지 못한다는 것
- 이미지 피라미드를 통해 스케일 불변성 문제 해결
- 기하학적 중심 O 와 모멘트 기반 중심 C 를 연결하여 그에 대한 방향을 구한다.

$$\{\bar{x}, \bar{y}\} = \left\{ \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right\}$$

- Rotated-BRIEF

- 기존의 BRIEF는 회전 불변성을 갖지 않는다
- FAST 검출 과정에서 얻은 키포인트의 방향 정보 사용
- Steer BRIEF 사용: 방향 정보에 따라 회전을 시킨 후 BRIEF 수행

Correspondence matching

- Basic concepts
 - SLAM에서 data association 과정을 수행하기 위해 필요한 과정
 - 영상과 영상, 혹은 영상과 맵 간의 descriptor들을 올바르게 매칭시키는 과정
 - 반복되는 텍스처 등으로 인한 문제는 불일치 누적을 야기할 수 있기 때문에!
- Brute-force matcher
 - 서로 다른 두 영상 내 특징점들의 descriptor 거리가 가장 짧은 점을 매칭점으로 간주
 - 일반적으로 descriptor 거리는 유사도를 의미
 - 유클리드 거리, 해밍 거리...
 - 점의 갯수가 많아지면 계산량이 늘어 효율적이지 못하다.
- Fast Library for Approximate Nearest Neighbors(FLANN)
 - 고차원에서 빠르게 매칭을 수행할 수 있는 매칭 라이브러리
 - 실시간 매칭을 수행할 수 있어 SLAM에 적합하다

Correspondence matching





Q&A



Thank you