

Oriented FAST

2016010873 박정욱

목차

- FAST가 해결할 수 없는 문제
- Harris corner detector
 - Corner란? Edge란?
- oFAST: Oriented FAST
 - Image Pyramid
 - 세기 도심(Intensity Centroid)
- 정리
- Todo

FAST가 해결할 수 없는 문제

1. '모서리스러움(Corneriness)'의 척도 부재
2. 영상에 대한 스케일을 고려하지 않음
3. 방향 요소가 존재하지 않음

Harris corner detector

- Moravec(1980)의 아이디어에서 착안
 1. 영상 내에 작은 정사각형의 '**패치**'가 있다고 생각
 2. 패치를 8방향으로 움직였을 때, 화소 세기의 변화를 고려
- **해석적 확장**을 이용해, 임의의 방향에 대한 움직임을 고려
 - 특정 조건에서만 정의되는 함수의 조건을 완화시키는 방법
 - **테일러 전개**를 이용
- 원형 가우시안 블러를 패치에 적용하면 노이즈 억제 가능

Harris corner detector

- 패치 움직임에 따른 화소 세기의 변화 $E(x, y)$

$$E(x, y) = \sum_{u,v} \underbrace{w(u, v)}_{\text{패치}} \underbrace{[I(x + u, y + v) - I(u, v)]^2}_{\text{화소 세기 차이}}$$

- $E(u, v)$ 를 최대로 하기 위해 **화소 세기 차이** 항을 최대화!
 - 패치 함수 부분은 모든 x, y 에 대해 동일하기 때문

테일러 전개

$$E(x, y) = \sum_{u, v} w(u, v) [I(x + u, y + v) - I(u, v)]^2$$
$$\approx \sum_{u, v} \left(\frac{\partial I}{\partial u} x \right)^2 + \left(\frac{\partial I}{\partial v} y \right)^2 + 2 \frac{\partial I}{\partial u} \frac{\partial I}{\partial v} xy$$

- 다루기 어려운 함수 $f(x)$ 를 다항함수로 근사하는 방법
 - $x=a$ 에서 $f(x)$ 와 동일한 미분계수를 갖도록!
- 편미분계수는 각 축 방향에 대한 화소 세기 **그레이디언트(기울기)**
- 8방향만을 계산할 수 있던 함수를 임의의 방향으로 확장!

Harris corner detector

$$\sum_{u,v} \left(\frac{\partial I}{\partial u} x \right)^2 + \left(\frac{\partial I}{\partial v} y \right)^2 + 2 \frac{\partial I}{\partial u} \frac{\partial I}{\partial v} xy$$
$$= \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} \sum (\frac{\partial I}{\partial u})^2 & \sum \frac{\partial I}{\partial u} \frac{\partial I}{\partial v} \\ \sum \frac{\partial I}{\partial u} \frac{\partial I}{\partial v} & \sum (\frac{\partial I}{\partial v})^2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- 확장된 화소 세기 변화 함수 $E(x, y)$ 의 행렬 형태
- $E(x, y)$ 가 커지기 위해서는 중간값의 행렬값이 커져야 함

Harris corner detector

- 이 행렬을 **고윳값 분해**하여, Corner/Edge/Flat 여부를 판단
 - 고윳값 분해: 고윳값과 고유벡터를 찾는 작업
- 행렬, 즉 변환을 적용했을 때 방향이 바뀌지 않는 벡터
 - > 고유벡터
- 고유벡터를 원래 벡터의 크기로 만들어주는 숫자
 - > 고윳값

Corner란? Edge란?

- x축 방향과 y축 방향에 대한 그레이디언트로 행렬을 구성
 - > 고유벡터들 중 가장 큰 고유값을 가지는 벡터 방향
 - > 화소 세기 변화가 가장 큰 방향!
- Corner: 두 고윳값이 모두 큰 경우
- Edge: 두 고윳값 중 하나만 큰 경우
- Flat: 두 고윳값 모두 작은 경우

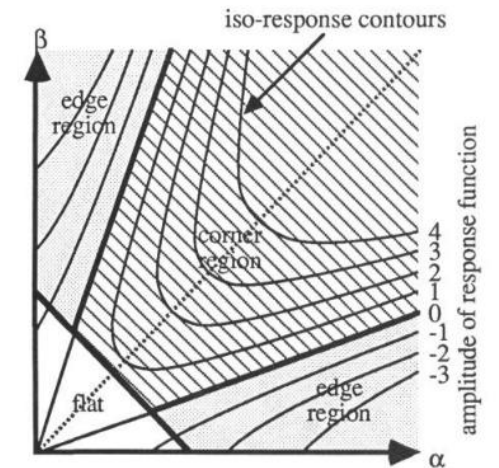


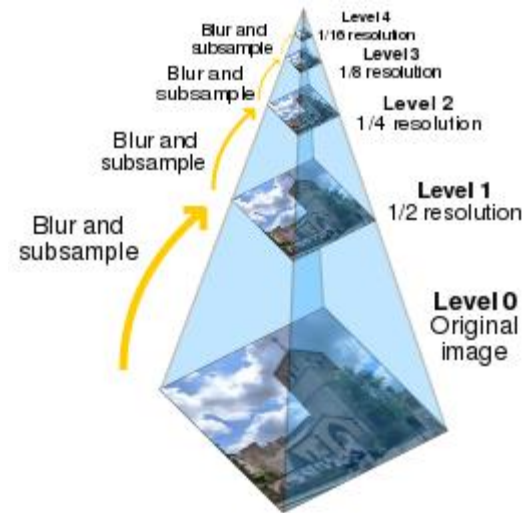
Figure 5. Auto-correlation principal curvature space- heavy lines give corner/edge/flat classification, fine lines are equi-response contours.

oFAST: Oriented FAST

1. Cornerness 척도를 이용하여 더 정확한 Corner 검출
 - 원하는 N개 이상이 나오도록 FAST Corner점 검출 후 Harris 적용
2. '**Image pyramid**'를 이용하여 스케일 고려
3. '**세기 중심(Intensity Centroid)**'를 이용하여 방향 요소 부여

Image pyramid

- 영상의 크기를 단계적으로 축소하며 일련의 영상 집합 생성
 - 블러링 - 다운샘플링을 반복
- 블러링 시 가우시안 필터 이용
- 다운샘플링 시 홀수번째 화소들만 취함



세기 도심(Intensity Centroid)

- 영상 **모멘트**(Image moment)를 사용
 - 영상 화소 세기에 대한 '가중평균' 개념
 - 이를 이용하여 간단한 이미지 속성들을 도출할 수 있음
- 모멘트란?
 - 물리량의 '분포상태'에 따라 결정되는 물리적 효과를 나타내는 방법
 - Harris detector에서 도출된 행렬도 일종의 모멘트(2차 모멘트 행렬)
 - 또는 자기상관행렬

$$\mu_n = r^n Q$$

(r은 거리, Q는 물리량, n은 차수)

세기 도심(Intensity Centroid)

- 일반적인 모멘트 식의 형태에 따른 영상 모멘트 정의

$$m_{pq} = x^p y^q I(x, y)$$

- x축 방향 요소를 p만큼, y축 방향 요소를 q만큼 반영한다는 뜻!

$$m_{00} = x^0 y^0 I(x, y) = I(x, y)$$

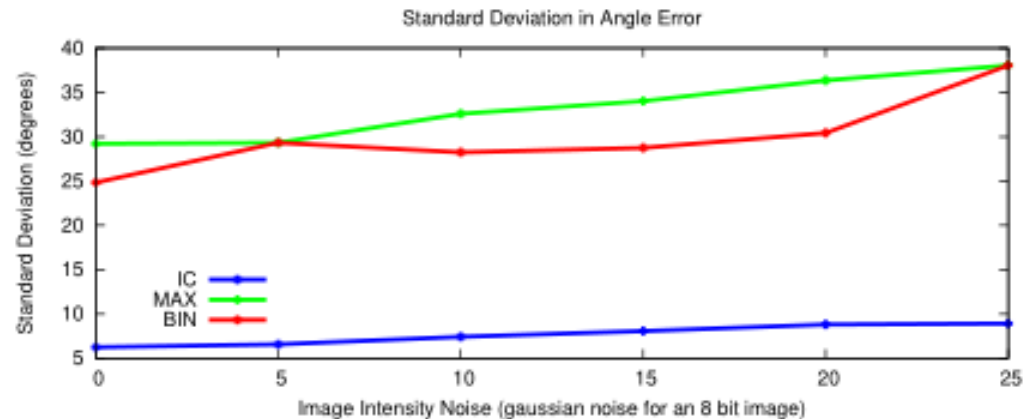
$$m_{10} = x^1 y^0 I(x, y) = xI(x, y)$$

$$m_{01} = x^0 y^1 I(x, y) = yI(x, y)$$

화소 세기를 반영한 도심 $C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$

세기 도심(Intensity Centroid)

- 이렇게 구한 세기 도심과 Corner의 중심점으로 벡터를 만듦
- 만들어진 벡터의 atan2 값이 곧 Corner의 방향 요소!
 - atan2 : 탄젠트의 역함수, 결과값으로 '각'을 반환



정리

- FAST가 해결하지 못했던 문제를 해결!
 1. Cornerness 척도 도입, 정확도 향상
 2. Image Pyramid 도입, 스케일 요소 고려
 3. 세기 도심 개념 도입, 방향 요소 추가
- 각 해결법의 근간이 되는 기반 지식
 - 테일러 전개, 고윳값 분해 – Corner를 수학적으로 정의
 - 블러링/다운샘플링 – 영상처리에서 스케일이 가지는 의미
 - 모멘트, 역탄젠트 – 영상 속성을 계산해내는 방법

Todo

- BRIEF
- Rotated BRIEF
- SLAM