**LOCAL IMAGE FEATURE**

**1. Local Image Feature의 정의**

* **Local Image Feature**는 \*\*"interesting part of an image"\*\*로 정의되며, 이미지 내에서 주목할 만한 부분을 찾아내는 특징

**2. Local Image Feature의 사용처**

* **Image Representation**: 이미지의 중요한 부분을 표현하여, 이미지 레벨에서의 디스크립터 생성과 물체 외형 모델링에 사용됩니다.
  + **Image-level Descriptor**: 이미지의 전체적인 특징을 요약하여 다른 이미지와의 비교를 가능하게 합니다.
  + **Object Appearance Modeling**: 포즈 변화에 민감하지 않고, 부분적으로 가려져도 강건하게 물체를 모델링할 수 있도록 도와줍니다.
  + **Matching between Multiple Views**: 스테레오 매칭, 3D 재구성, 이미지 스티칭 등 여러 시점에서의 이미지 간 매칭에 사용됩니다.

**3. 좋은 Local Image Feature의 조건**

* **Repeatability**: 다양한 조건에서 일관되게 검출될 수 있어야 합니다.
* **Saliency**: 특징점이 주변과 잘 구별되어 시각적으로 눈에 띄어야 합니다.
* **Locality**: 특징점이 이미지의 작은 영역에서 검출되어야 하며, 전체 이미지에 의존하지 않고 국소적인 정보로 특징을 구성해야 합니다.
* **Be Unambiguous for Matching**: 매칭 과정에서 모호하지 않아야 하며, 쉽게 구별할 수 있어야 합니다.
* **Well-defined Position**: 이미지 공간에서 명확한 위치를 가져야 합니다.
* **Local Area with Rich Information**: 풍부한 정보를 포함한 지역이어야 하며, 중요한 디테일을 표현할 수 있어야 합니다.
* **Stable Against Geometric and Photometric Deformations**: 기하학적 변형이나 광학적 변형에도 안정적인 특징을 유지해야 합니다.
* **Consistent and Reproducible**: 이미지의 일관된 부분을 반영하며 반복적으로 검출될 수 있어야 합니다.

**IMAGE PROCESSING**

이미지 프로세싱의 기본 : 필터링을 통해 이미지의 노이즈를 줄이고 이미지의 주요 특징 강조 (이미지에서 픽셀을 주변 픽셀의 가중 평균으로 대체하여 노이즈를 제거)

이미지를 함수로써 이해하기: 인풋으로 들어온 이미지의 각 픽셀을 수학적으로 f(x,y)로 해석할 수 있다.

예시 : (10, 20) 위치의 주변에는 3x3의 픽셀들이 있다고 가정할 수 있습니다. 예를 들어, 이 주변에는 다음과 같은 픽셀들이 있을 수 있다

* (9, 19), (10, 19), (11, 19)
* (9, 20), (10, 20), (11, 20)
* (9, 21), (10, 21), (11, 21)

이 주변 픽셀들의 값이 각각 140, 145, 150, 155, 150, 140, 135, 150, 160이라면, 이 픽셀들의 평균을 계산하여 (10, 20) 위치의 새로운 픽셀 값을 정하게 된다.

위의 픽셀 값들을 더해주고, 픽셀 개수로 나누면 된다.f(10,20)=(140+145+150+155+150+140+135+150+160)/9

계산 결과, 이 평균 값이 (10, 20) 위치의 새로운 픽셀 된다. 또한 이렇게

필터(커널)을 이미지에 적용하여 필터의 값과 이미지의 값들이 곱한 후 더하는 연산을 **컨볼루션**이라고 한다.

이런 식으로 이미지를 수학적 함수로 각 픽셀마다 표현 가능해지면 컴퓨터가 이미지 처리 작업 효율적으로 진행할 수 있다.

이미지 프로세싱의 연산 형태(가장자리 처리):

1. Full : 필터가 이미지의 가장자리까지 적용될 수 있도록 필터의 일부가 이미지 바깥까지 연산에 포함되도록 한다. 출력 이미지가 입력보다 커진다.
2. Same : 입력과 출력 크기를 동일하게 유지하기 위해 적절한 패딩 추가
3. Valid : 패딩 없이 이미지 내부에서만 필터를 적용하여 출력 크기를 줄임

이미지 프로세싱의 필터가 이미지를 넘어섰을 때 공백을 채우는 방법:

1. Symm(대칭): 이미지 대칭으로 확장 => 가장 티가 안 나고 좋다
2. Circular / wrap(순환): 이미지 반대쪽 가장자리 연결
3. Pad/fill : 0등 특정 값 채워 처리 => 출력 이미지 가장자리 쪽이 티날 수 있다.

**CONVOLUTION**

컨볼루션의 속성 1. Linear Property (선형성) : apply(I,f1​+f2​)=apply(I,f1​)+apply(I,f2​) 여기서 I는 이미지 f는 필터를 의미. 또한 아래의 선형대수학에서 통용되는 법칙들도 포함하여 외우자

* 교환법칙 : f∗g=g∗f, 필터와 이미지 순서 바뀌어도 결과 동일
* 결합법칙 : (f∗g)∗h=f∗(g∗h) 필터 여러 개일 때, 결합 순서는 중요하지 않음
* 분배법칙 : f∗(g+h)=f∗g+f∗h :
* 스칼라 배분 : kf\*g = f\*kg = k(f\*g)
* Identity (가운데 1이고 나머지 0인 필터) : 이미지와 결합 시 아무 변화 일어나지 않는다.

컨볼루션의 속성 2. Shift-Invariance(시프트 불변성) : 이미지 내에 필터가 어떤 위치로 이동하던지 같은 ‘작용’을 한다. 컨볼루션 연산은 특정 위치에 종속되지 않는다.

필터를 적용하는 다른 연산 방식 : 컨볼루션 외에 Cross-Correalation이라는 방법이 있다. Cross-Correalation은 컨볼루션과 유사하지만 필터를 뒤집지 않고 원래 방향 그래도 이미지에 적용한다. 즉 컨볼루션 연산으로 하는 필터는 적용 전 x축과 y축이 뒤진힌 (flip) 형태로 이미지에 적용 된다.

수학에서의 Continous 컨볼루션 연산: 컴퓨터로 이미지를 처리할 땐 이산적으로만 컨볼루션 연산이 가능하지만 수학적으로는 연속적인 컨볼루션이 가능합니다. Continuous convolution은 연속적 데이터(예: 아날로그 신호)에서 적용되며, 수식은 다음과 같다.

g(x, y) = ∫∫\_{−∞}^{∞} f(τ, μ) h(x − τ, y − μ) dτdμ

**FILTERS**

1. 리니어 필터 : 입력 픽셀 값과 가중치를 곱하여 합산하는 방식으로 작동되는 필터로 기본적인 필터. 배열에 따라 샤프닝 필터, 가우시안 필터등이 된다.
2. 박스 필터 : 네모난 모양의 기본 필터로 각 가중치가 모두 같은 값을 갖는 linear 필터의 일종
3. 샤프닝 필터 : 세부적인 디테일을 강조하는 방식으로 작동. 원본 이미지에서 smoothed 이미지를 뺀 후 디테일을 추출한 다음 가중치 α를 곱해 원본이미지에 더해준다. 가중치가 커질수록 샤프닝 적용이 극단적으로 엣지가 두드러지고 0이면 디테일 추가안된다.
4. 가우시안 필터 : 중심에 중점 두고 부드럽게 처리하여 엣지 보존하며 블러 등의 효과를 줘서 노이즈를 감수할 수 있다. 표준편차 σ가 커질수록 필터가 더 넓게 퍼지는 모양으로 이미지에 블러 효과 줄 때 더 확 블러링된다.
   1. 수식 : exp(-(x² + y²) / (2σ²)) 형태로 나타나며, σ는 필터의 스케일을 조절하는 매개변수
   2. 이미지에서 불필요한 고주파수를 제거하여 노이즈를 감소시키는데 주로 사용
   3. 필터의 크기가 커지거나 이미지의 크기가 커지면 연산량이 급격히 증가하기 때문에 이미지 처리 속도가 빠른 가우시안 필터는 요긴하다. 가우시안 필터가 이미지 처리 속도가 빠른 이유는가우시안 필터의 Separability(분리 가능성) 덕인데 이는 2차원 가우시안 필터를 1 차원(가로-세로)로 분리하여 적용할 수 있기 때문이다.
5. 파생(Derivative) 필터 : 이미지의 경계나 변화가 급격한 부분 강조, 이미지 픽셀간 차이 계산하여 경계선이나 에지 강조
6. 비선형 필터 : 출력이 입력의 단순한 가중합이 아닌, 복잡한 연산을 통해 결정되는 필터로 비선형적 분석으로 특정 효과 얻는다. 예사로 미디안 필터, 바이래털(양방향)필터, 어댑티브 필터 등이 있다.
7. 미디안 필터: salt and pepper같은 노이즈 있는 이미지에서 사용(가우시안 먹히지 않는 경우)
   1. 왜 salt and pepper에서 가우시안이 안먹히나요?
      1. 가우시안 필터는 성형필터라 평균값 기반 연산을 하는데 쏠트앤 페퍼 같은 극단적인 노이즈에 영향을 받아 결과물이 왜곡될 수 있다.
8. 카운팅 필터 : 특정 값이나 조건을 만족하는 픽셀의 개수를 세어서 그 값을 결과로 출력하는 방식. 즉 조건 기반의 필터링이라고 할 수 있다.
9. Missing data필터 : 이진 마스크(0,1로) 사용하여 데이터가 있는지 여부 확인하고 필터를 다르게 적용, 이때 이진마스크는 특정 임계값을 기준으로 만든다.
10. 바이래터럴(bilateral) 필터: 이미지 전체에 균일한 블러를 먹이는 가우시안 필터와 달리 중심 픽셀과 강도가 다른 픽셀은 낮은 가중치를 부여하는 ‘bias’적용하여 엣지 보존 스무딩에 적합, 엣지 보존하며 노이즈를 줄일 수 있다,

수식: g[i, j] = (1 / W\_sb) \* Σ Σ f[m, n] \* n\_σs [i - m, j - n] \* n\_σb (f[m, n] - f[i, j])

상황에 따른 필터 선택:

1. 노이즈 때문에 블러링해야 할 때 : 박스필터보다 가우시안 필터가 얼룩 없이 더 좋은 효과 낸다.
2. Salt and pepper noise – median 필터 사용
3. 데이터가 누락된 부분있을 때 - Missing data 필터
4. 엣지를 보존하며 노이즈를 줄여야 할 때 : bilateral 필터