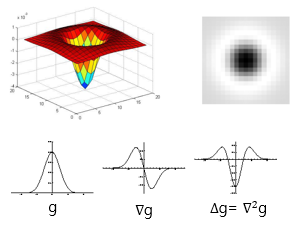
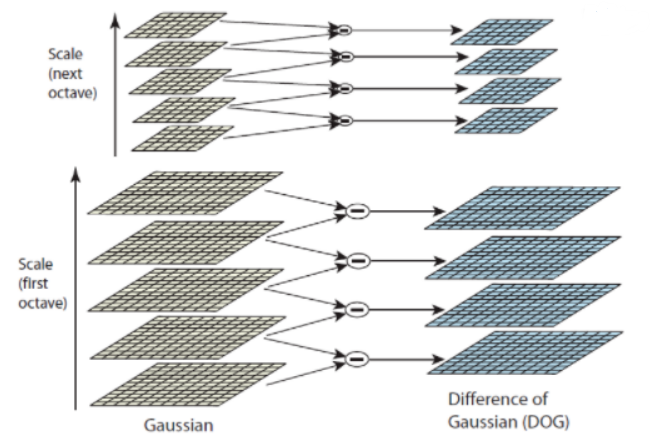
**DoG\_filter 리포트**

목차

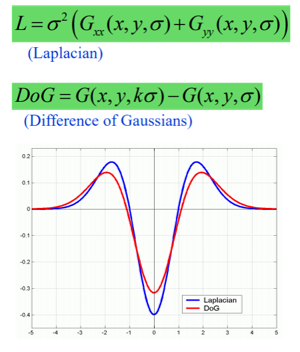
1. DoG detection이란?
2. DoG detection 사용 예시-SIFT알고리즘
3. 프로젝트 결과 분석
4. **DoG detection 이란?**

Difference of Gaussian detection을 뜻하고, 말 그대로 두 개의 서로 다른 가우시안 필터를 이용하여 엣지를 추출하는 것을 뜻한다. Difference of Gaussians는 라플라시안과 유사한 값을 갖고, 엣지, 코너, blob detection에 좋은 결과를 도출한다. 이러한 장점 때문에 이미지의 크기와 회전 등과 상관없이 특징을 추출하는 SIFT(Scale-Invariant Feature Transform)알고리즘에도 사용된다.

Difference of Gaussian detection을 설명하기에 앞서 Laplacian of difference 및 Laplacian에 대한 이해가 필요하다. 그림(가)는 corner, edge, blob등을 detection을 할 때 쓰는 Laplacian이다. 이를 이용해 그래디언트의 차이로 주변과 색이 다른 영역을 detection하여 blob을 찾는다. 그런데, 실제로는 LoG필터의 크기에 따라 detection 되는 blob이 매번 다르다는 문제점이 있고, 이에 대한 해결책으로 커널을 줄여가며 detection을 하는데, 이 역시도 계산 과정이 복잡하며, 구하기 어렵다는 문제점이 있다. 이에 커널은 그대로 두고, 이미지를 줄여가며 blob을 detection하는 방법을 채택했다. 추가로 그 과정 중에 줄여진 이미지로 인해 발생한 해상도에 따른 다양한 이미지 세트를 이미지 피라미드라고 한다.



그림(가) Laplacian 그림 (나) 이미지 피라미드

LoG를 하는 과정 중에 라플라시안을 두 번 미분하는 과정이 있는데, 오른쪽 그림과 같이 Laplacian 값과 가우시안 두 개를 뺀 값이 유사함을 확인할 수 있다. 이러한 점을 이용하여 Difference of Gaussians 를 라플라시안 대신 사용하여 엣지를 추출한다.

그림(나)의 왼쪽과 같이 여러 번 라플라시안을 거치는 경우보다 적은 횟수로 그림(나)의 오른쪽과 같이 Difference of Gaussians 를 거치므로 간단하고, 연산량이 줄어든다는 장점이 있다. 이러한 장점 때문에 SIFT알고리즘에서도 쓰인다.

**2. DoG detection 사용 예시-SIFT알고리즘**

SIFT (Scale-Invariant Feature Transform)은 이미지의 크기와 회전에 불변하는 특징을 추출하는 알고리즘이다. 서로 다른 두 이미지에서 SIFT 특징을 각각 추출한 다음에 서로 가장 비슷한 특징끼리 매칭해주면 두 이미지에서 대응되는 부분을 찾을 수 있다는 것이 기본 원리이다.

그 과정은 다음과 같이 크게 6가지로 나뉜다.

1. "Scale space" 만들기

2. Difference of Gaussian (DoG) 연산

3. keypoint들 찾기

4. 나쁜 Keypoint들 제거하기

5. keypoint들에 방향 할당해주기

6. 최종적으로 SIFT 특징들 산출하기

먼저 원본 이미지를 두 배로 크게 만든 다음에 점진적으로 블러시킨다. 그리고 원본 이미지를 점진적으로 블러시킨다. 그후 원본 이미지를 반으로 축소시킨 이미지에서 또한 점진적으로 블러된 이미지들을 만들어 낸다. 또, 반으로 축소시킨 다음 점진적으로 블러된 이미지들을 만든다. 여기서, 원본 이미지를 2배로 키우는 이유는 나중에 Difference of Gaussian (DoG) 이미지를 만들 때 같은 옥타브 내에서 인접한 2개의 블러 이미지를 활용해서 만들고, 또 그렇게 생성된 DoG 이미지들 중에서 인접한 세 개의 DoG 이미지를 활용해서 Keypoint들을 찾기 위해서이다.

다음, 앞서 언급한 DoG 연산을 진행하여, 극값들을 찾아 key point들을 찾고, 낮은 contrast를 갖은 키포인트와 엣지 위에 존재하는 key point를 제거하여 좋은 key point만을 남긴다. 이 key point들을 각각 방향을 할당해서 rotation invariance를 갖게 하고, 가장 두드러지는 방향을 찾는다. 구한 scale 과 rotation으로 4x4윈도우를 이용하여 SIFT 특징을 산출하고, 이미지 매칭을 해준다.

위 과정에서 설명한 바와 같이 keypoint를 찾을 때, DoG를 사용하고, LoG 대신 DoG를 채택하는 이유는 구현이 간단하고, 속도가 빠르기 때문이다.

**3. 프로젝트 결과 분석**

원본 이미지

벽, 실내, 꽃, 꽃다발이(가) 표시된 사진

자동 생성된 설명

Edge detection by DoG Gaussian1(3x3커널, 시그마=0.5) Gaussian2(3x3커널, 시그마=3)

텍스트, 벽이(가) 표시된 사진

자동 생성된 설명

결과값들은 위와 같다. 원본 이미지는 320\*480으로 설정하였고, 커널 사이즈와 시그마를 여러 번 수정하여 가장 잘 엣지가 나올 때를 연구해본 결과, 커널이 3x3일 때가 5x5일 때보다 훨씬 엣지가 뚜렷하게 잘 나왔고, 시그마 값이 커져도 blur에는 큰 차이가 없이 보였다. 현재의 결과값인 시그마가 0.5인 3x3 가우시안 커널로 블러 시킨 이미지와 시그마가 3인 3x3 가우시안 커널로 블러 시킨 이미지 두개를 차를 이용하여 엣지를 추출하는 것이 가장 뚜렷했다.