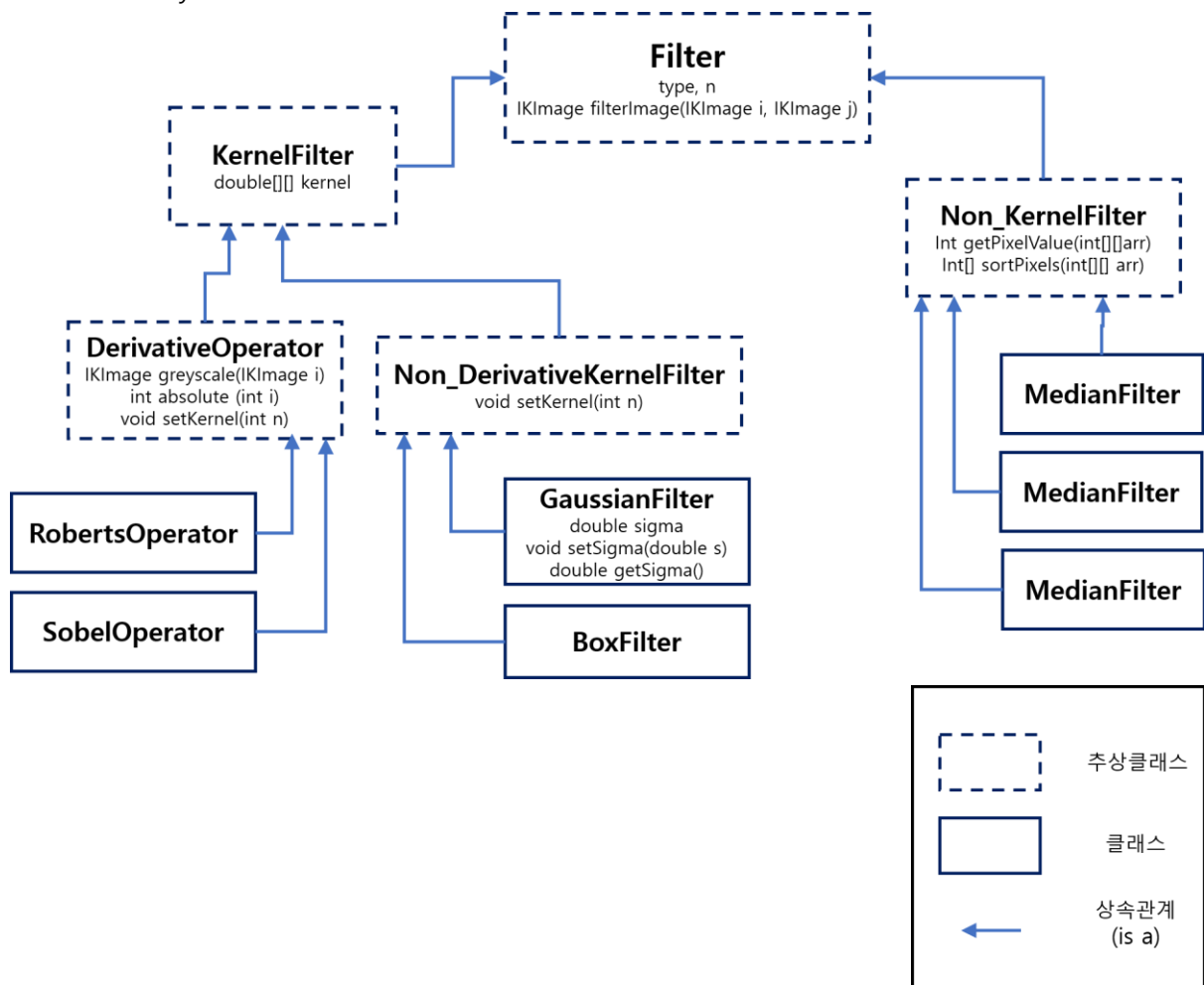


객체지향 프로그래밍 Homework 6

Class Hierarchy



1. 클래스 설계도에 대한 설명

모든 filter들의 최고 조상인 Filter클래스는 모든 필터들이 공통적으로 가지는 사이즈인 int n과 각 클래스가 어떤 타입인지를 표시하는 String type을 멤버변수로 갖는다. 또 개별 필터들이 공통적으로 가지는 과정인, 이미지에 필터를 씌우는 과정의 메서드를 추상메서드인 IKImage filterImage(IKImage i, IKImage j)를 가지도록 하여 자손 클래스에서 구현하도록 하였다. 실제 Filter 클래스가 인스턴스로 구현될 일은 없고 추상메서드를 가지므로 추상클래스로 만들었다.

KernelFilter클래스 역시 자손들이 공통적으로 가지게 될 멤버변수인 2차원 double배열 kernel을 가지도록 하고 아직 공통적으로 filterImage 메서드를 구현할 수 없으므로 추상클래스로 만들었다.

DerivativeOperator에서는 자손 클래스가 공통적으로 쓰게 될, 주어진 이미지를 greyscale화 시키는 greyscale메서드와 절댓값을 반환하는 absolute메서드를 구현해 놓았고 커널의 크기에 따라 2차원 커널배열을 구현하는 setKernel 추상메서드를 선언하였다. 여기서 n은 DerivativeOperator에 맞게 3에 고정되게끔 자손클래스 생성자에서 구현해 놓았다. 그리고 setKernel은 자손 생성자들에 의해 구현된다.

Non_DerivativeOperator클래스는 Gaussian, BoxFilter에 맞게 커널만 있으면 그 연산 과정이 동일하기 때문에 커널과 연산하여 필터를 씌우는 filterImage 메서드를 구현하여 놓았다. 하지만 커널을 구현하는 방식이 각자 다르기 때문에 setKernel메서드를 추상메서드로 선언하였고 GaussianFilter와 BoxFilter에 각각 따로 구현해 놓았다.

Filter클래스의 또 다른 자손인 Non_KernelFilter는 자손 클래스들이 공통적으로 필요로 하는 2차원배열로 받아서 1차원 오름차순 배열로 반환하는 sortPixels메서드를 구현하고, 이 결과에 따라 각 자손클래스에서 필요한 값을 반환하는 것을 공통으로 여겨 getPixelValue 추상메서드를 선언하였다. 각 자손클래스에서 이미지에 필터를 씌우는 연산 과정은 원 이미지에서 추출한 값만 다르고 연산과정은 같기 때문에 filterImage 메서드를 Non_KernelFilter에서 구현하였고 원 이미지에서 값을 추출하는 메서드인 getPixelValue는 각 자손 클래스에서 실제로 구현된다.

2. 각 필터 및 사이즈에 따른 이미지 결과 차이 분석

A. Box Filter



(좌측부터 size 3, 7, 11)

BoxFilter의 경우 원 이미지에 비해 흐릿하게 만들어 주는 효과를 만들어 주었다. 이는 커널의 크기가 커질수록 더 효과가 커짐을 볼 수 있다.

B. Gaussian Filter



(좌측부터 size 3, 7, 11)

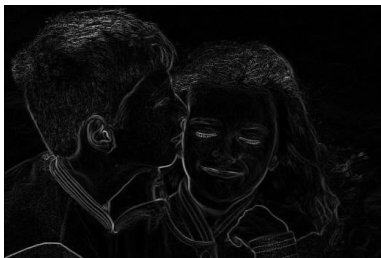
Gaussian Filter 역시 Box Filter와 마찬가지로 원 이미지에 비해 흐릿한 효과를 주며 사이즈가 커질 수록 그 효과 역시 더욱 커지는 것을 볼 수 있다. Box Filter와 비슷한 느낌이지만 BoxFilter가 겹쳐 보이는 듯한 흐릿함이라면 GaussianFilter의 경우 조금 더 부드럽게 흐릿해지는 것을 알 수 있다.

C. Sobel Operator



Sobel Operator의 경우 윤곽선만 흰 선으로 남아 처리되었다. 이는 픽셀의 색깔의 x와 y값의 미분 값, 즉 색깔의 변화정도를 취해 더한 커널에 의한 것으로, 윤곽선에서 색깔의 변화가 가장 심한 만큼 더 픽셀의 수가 높아 흰색으로 처리되고 비교적 비슷한 색을 유지하는 면의 경우 검은색으로 처리된 것을 알 수 있다.

D. Roberts Operator



Roberts Operator의 경우 Sobel Operator와 비슷한 원리로 윤곽선이 흰 선으로 처리되었으나, Sobel Operator와 달리 그 커널이 가지는 값의 크기가 작은 만큼 윤곽선의 뚜렷함도 덜 선명하게 처리되어 있음을 알 수 있다.

E. Median Filter



(좌측부터 size 3, 7, 11)

Median Filter의 경우 픽셀 주변의 중간 값을 취하는 필터인 만큼 이미지가 한층 흐릿해지면서 비

스한 색의 영역은 더 흐릿해지고 부드러워지는 효과가 있음을 볼 수 있다. 이미지가 마치 수채화 처럼 바뀌는 현상이 나타나고 이는 역시 size가 커질수록 더욱 효과가 커진다는 것도 알 수 있다.

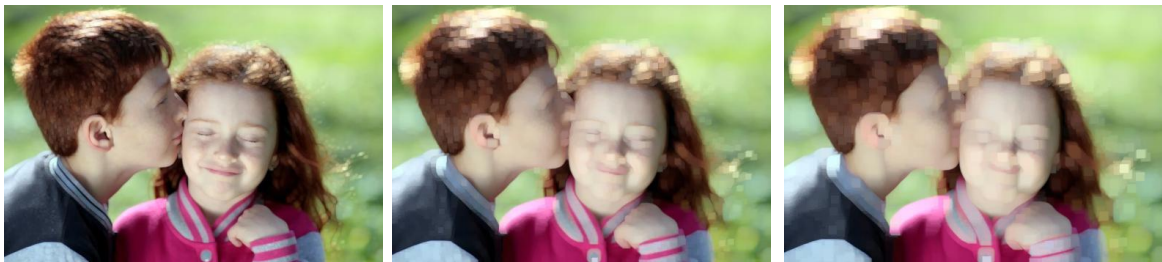
F. Min Filter



(좌측부터 size 3, 7, 11)

Min Filter의 경우 픽셀 주위로 최솟값의 색깔을 선택하다 보니 원 이미지보다 색깔이 어두운 부분이 더 확대되고 이미지 자체도 전체적으로 어두워지는 효과를 미쳤다. 이 역시 사이즈가 커질수록 그 효과는 더욱 컸음을 알 수 있다.

G. Max Filter



(좌측부터 size 3, 7, 11)

Max Filter의 경우 픽셀 주위의 최댓값의 색깔을 선택하는 만큼 원 이미지보다 밝은 색깔의 부분이 더 확대되고 이미지 자체도 전체적으로 밝아지는 효과를 가져왔다. 이 역시 사이즈가 커질수록 그 효과는 컸다.