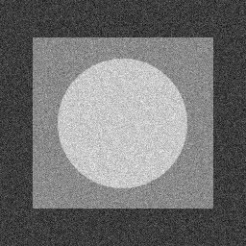
**< Smoothing Filter >**

**[ Original image 1 ]**

****

1. **Original + Gaussian Noise :**



|  |  |  |  |
| --- | --- | --- | --- |
| Mask Size | Gaussian Filter | Median Filter | Average Filter |
| 3\*3 |  |  |  |
| 5\*5 |  |  |  |
| 7\*7 |  |  |  |

수업시간에 배운대로 마스크 사이즈가 커질수록 블러가 많이 생겨 노이즈는 많이 제거되지만 윤곽선 정보는 흐릿해졌습니다. average 필터는 블러가 너무 심해져 경계를 알아보기가 어려웠습니다. 또한, 가우시안 필터와 미디안 필터와 약간의 색상 차이가 존재하는 것처럼 보입니다. average 필터는 평균값을 이용하기 때문에 색상이 섞여 기존에 없던 노란빛의 색상이 생겼습니다.

다만, 가우시안 노이즈는 가우시안 필터가 노이즈를 가장 효과적으로 제거할 것이라고 생각했는데, 제가 만든 필터에서는 미디안 필터에서 노이즈 제거 효과가 가장 뛰어났습니다.

1. **Original + Salt & Pepper Noise :**

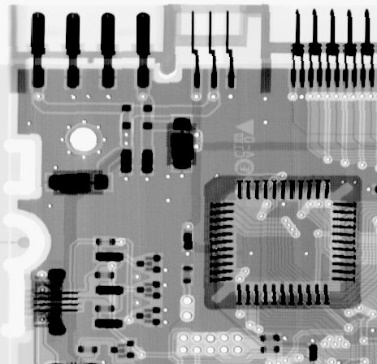


|  |  |  |  |
| --- | --- | --- | --- |
| Mask Size | Gaussian Filter | Median Filter | Average Filter |
| 3\*3 |  |  |  |
| 5\*5 |  |  |  |
| 7\*7 |  |  |  |

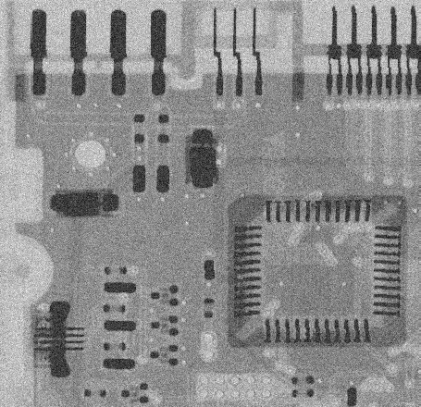
salt&pepper 노이즈에서는 미디언 필터가 확연히 좋은 성능을 보여주었습니다. 미디언 필터만 봤을 때, 마스크 크기에 따른 성능 차이는 크게 보이지 않았지만, 자세히 들여다보면 3\*3에서는 사라지지 않았던 작은 노이즈가 7\*7에서는 그 개수가 줄어들었음을 확인할 수 있습니다. 하지만 필터를 적용하면서 원본영상과 정확히 일치하는 영상을 구하지는 못했습니다. 원과 회색 사각형을 보면 삐뚤빼뚤한 테두리가 되었음을 확인할 수 있습니다.

가우시안 필터와 mean 필터를 비교해보면, 3\*3에서는 큰 차이가 느껴지지 않았지만, 마스크 사이즈가 커질수록 average 필터는 가우시안 필터보다 윤곽선 정보가 훨씬 많이 손실되었음을 확인할 수 있습니다. 노이즈만 봤을 때는 average 필터의 노이즈가 더 많이 제거된 것처럼 보이지만, 전체적으로 영상의 윤곽선 정보가 너무 많이 손실되어 원본영상을 알아보기가 더 어려워졌습니다. 가우시안 필터 또한 input 영상과 비교하면 노이즈는 많이 흐려졌지만 영상 자체의 화질 또한 낮아졌습니다.

**[ Original Image2 ]**



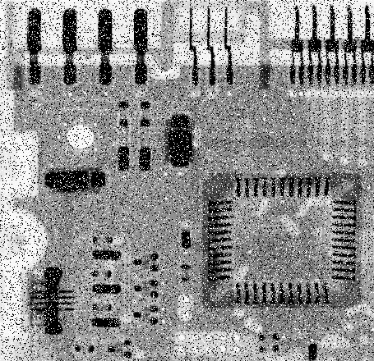
1. **Original + Gaussian noise :**



|  |  |  |  |
| --- | --- | --- | --- |
| Mask Size | Gaussian Filter | Median Filter | Average Filter |
| 3\*3 |  |  |  |
| 5\*5 |  |  |  |
| 7\*7 |  |  |  |

세 가지 필터를 비교했을 때, 이번에는 가우시안 필터를 적용했을 때 가장 선명하고 노이즈가 잘 제거된 영상을 얻을 수 있었습니다. 3\*3 윈도우에서는 큰 차이를 보이지 않았지만, 마스크 사이즈가 커질수록 가우시안 필터의 성능이 두드러졌습니다. 세 가지 크기의 마스크 전부, average filter가 가장 낮은 성능을 보였는데, 너무 뿌얘서 노이즈가 제거되었는지조차 확인하기 어려웠습니다. Gaussian filter, Median filter, Average filter 순으로 괜찮은 결과를 보였습니다.

1. **Original + Salt & Pepper Noise :**



|  |  |  |  |
| --- | --- | --- | --- |
| Mask Size | Gaussian Filter | Median Filter | Average Filter |
| 3\*3 |  |  |  |
| 5\*5 |  |  |  |
| 7\*7 |  |  |  |

이번에도 역시 salt&pepper noise 에서는 median 필터가 탁월하게 좋은 성능을 보이고 있습니다. 특히 5\*5 사이즈의 마스크에서는 거의 원본 영상과 다름 없는 결과가 나왔습니다. 7\*7 마스크는 3\*3, 5\*5 마스크보다 노이즈는 더 잘 제거했지만, 노이즈가 아닌 부분마저 노이즈로 인식해서 제거하여 영상이 전체적으로 흐릿해지고 디테일한 정보들이 손실되었습니다.

Gaussian filter와 Average filter를 비교해보면, average filter는 영상이 너무 흐릿해져서 원본 영상의 디테일한 정보는 거의 사라져 있었습니다. 특히, 원본 영상에서 흰 색으로 나타난 부분들은 거의 보이지 않았습니다. gaussian filter는 average 필터보다는 더 선명하지만 노이즈가 많이 남아있어 원본 영상의 정보를 알아보기 어렵습니다.

**< Sharpening Filter : Highboost Filtering >**

**[ Original Image 1 ]**



**Mask1 : [ [0, -1, 0] , [-1, 4, -1], [0, -1, 0] ]**

|  |  |
| --- | --- |
| A = 1.2 | A = 1.5 |
|  |  |

**Mask2 : [ [-1, -1, -1] , [-1, 8, -1], [-1, -1, -1] ]**

|  |  |
| --- | --- |
| A = 1.2 | A = 1.5 |
|  |  |

수업시간에 배운 두 가지 마스크를 이용하고 A(가중치)값을 조정하여 Highboost 필터를 적용하였습니다. mask1보다 mask2에서 윤곽선이 더 뚜렷하게 나타났으며, 디테일한 정보들도 잘 확인할 수 있었습니다. A값(=가중치)를 비교해보면, 두 마스크 모두 A = 1.2 일 때보다 A = 1.5 일 때 영상의 밝기가 높아졌습니다. 이는 원본 영상의 가중치를 높여서 발생한 결과입니다. 영상이 밝아지면서, 영상의 밝은 부분의 그림자가 사라져 디테일한 부분의 정보가 많이 손실되었습니다. 반면, 어두운 부분의 디테일한 정보가 미약하게나마 더 잘 관찰되는 것을 확인할 수 있었습니다.

**< Smoothing in Color Space >**

mask size = 5 \* 5로 설정

|  |  |  |  |
| --- | --- | --- | --- |
| Input Image | Median Filtering | Average Filtering | Gaussian Filtering |
| [ Lena\_noise ] |  |  |  |
| [ Gaussian Noise ] |  |  |  |
| [ Salt&pepper noise ] |  |  |  |

세 가지 필터 중 영상의 정보를 가장 잘 보존하는 (덜 흐려지는) 필터는 Gaussian filter였습니다. Lena\_noise뿐만 아니라 Gaussian noise 에서도 gaussian filter가 가장 뚜렷한 윤곽선 정보를 가지고 있습니다. gaussian noise에서는 median filter와 average filter는 노이즈 제거에 있어 가우시안 필터보다 조금 더 나은 성능을 보이고 있습니다. 하지만 성능이 좋은 만큼 blurring이 심해 gaussian filter에 비해 더 적은 윤곽선 정보를 가지고 있습니다.

실험하기 전, 저는 Lena\_noise에서 가장 흐려지게 만드는 필터는 average filter일 것이라고 예상했습니다. 하지만, 막상 프로그램을 실행해보니 median filter가 조금 더 흐린 것 같다는 생각이 들었습니다.

Salt&Pepper noise 에서는 세 가지 필터 모두 비슷한 성능을 나타내었는데, 노이즈 제거 효과에 있어서는 Gaussian filter가 가장 좋지 않은 효과를 보였습니다. average filter와 median filter는 모두 비슷한 노이즈 제거 효과를 보여주었습니다.

**< Edge Detection >**

**[ 1차 미분 연산자 ] : Roberts, Prewitt, Sobel**

|  |  |  |  |
| --- | --- | --- | --- |
| Origianl | Roberts | Prewitt | Sobel |
|  |  |  |  |
|  |  |  |  |

Edge 검출은 노이즈가 있는 영상과 없는 영상 두 가지에 대해 실습을 진행하였습니다. 두 가지 영상 모두, 실사용 빈도가 가장 높은 Sobel이 가장 좋은 성능을 보였고, Roberts Operator가 가장 좋지 않은 성능을 보였습니다.

Roberts 연산자는, Prewitt과 Sobel에 비해 엣지가 거의 강조되지 않고 있습니다. 연한 회색이 많이 있어서 Edge 자체가 잘 안뽑힌 것 같지만, Edge 자체는 Prewitt과 Sobel에 비해 많이 뽑혔습니다. 이는 수업시간에 배운 내용처럼 Roberts 연산자가 2\*2 mask여서 노이즈에 더 민감하기 때문이라고 유추할 수 있었습니다.

Prewitt과 Sobel은 Edge를 더 강하게 검출하였습니다. Prewitt과 Sobel을 비교해보면, Prewitt은 전반적으로 수직/수평 방향의 경계를 더 섬세하게 잡아냈습니다. 그에 반해 Sobel은 대각 방향의 경계를 잘 찾아내어 전체적으로 더 선명한 느낌을 주고 있습니다. 특히 Noise가 추가된 영상에서는 Prewitt보다 Sobel이 전체적인 디테일한 경계정보 뿐만 아니라 노이즈도까지 더 자세히 검출해내는 것을 확인할 수 있었습니다.

**[ LoG filter ]**

LoG1 : mask size = 3 \* 3 / σ = 1

LoG2 : mask size = 7 \* 7 / σ = 1.2

LoG3 : mask size = 13 \* 13 / σ = 1.4

|  |  |  |  |
| --- | --- | --- | --- |
| Input Image | LoG1 | LoG2 | LoG3 |
|  |  |  |  |

LoG 필터에서는 마스크 사이즈와 시그마 값을 조절하여 실험을 수행하였습니다. LoG3이 가장 큰 크기를 가지고, 시그마 값도 가장 높게 지정하였습니다. Input 이미지는 가우시안 필터를 LoG1에서 LoG3이 될수록 스케일이 커지고 디테일한 정보는 많이 사라져 변화가 큰 경계 및 윤곽 정보만을 나타내고 있습니다. 디테일한 정보가 많이 사라짐과 동시에 경계부분의 blur가 심해졌습니다.

**[ Canny Operator ]**

|  |  |
| --- | --- |
| Original image | Canny edge operator 적용 |
|  |  |

Thresholding을 수행하는 연산이라 여태까지 진행한 모든 엣지 검출 operator 중 가장 깔끔한 결과를 보여주고 있습니다. 하지만 그만큼 손실되는 정보가 많습니다. 회색 부분은 거의 검출되지 않고, 큰 변화를 보이는 경계들만 깔끔하게 보여주고 있습니다.

같은 영상이라도 노이즈의 종류, 마스크의 크기, 필터의 종류, (경우에따라) 가중치의 값, 시그마 값 등 여러 가 지 다양한 요인에 따라 결과는 극명히 달라졌습니다. 영상을 처리할 때, 영상의 종류에 따라 적절한 필터를 결정하고, 해당 필터의 특성과 영상의 특성에 따라 구체적인 값을 결정하는 것이 매우 중요한 일임을 영상처리를 직접 수행해보면서 더 체감할 수 있었습니다.