

## Noise 제거를 위한 filter 기술 조사 및 구현

### Bilateral Filter

Bilateral filter는 image의 edge를 보존하면서 smoothing하고 noise를 제거할 수 있는 알고리즘이다.

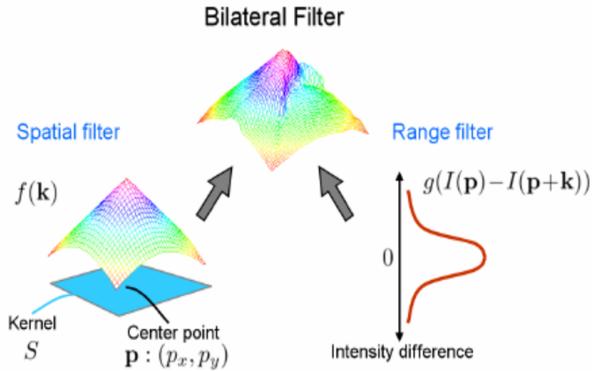


Figure 1: Bilateral filter

$$BF[I]_p = \frac{1}{W_p} \sum_W f(x+u, y+v) \times G((x+u)-x, (y+v)-y) \times G'(f(x+u)-f(x), f(y+v)-f(y)) \quad (1)$$

Bilateral filter의 수식은 Equation 1과 같다. Figure 1와 수식을 matching해보면  $G((x+u)-x, (y+v)-y)$ 는 normalized gaussian blur와 동일하며 Figure 1의 spatial filter와 동일하다.

$G'(f(x+u)-f(x), f(y+v)-f(y))$ 는 pixel값의 차이가 크면 gaussian은 0에 가까워지므로 edge를 보존하도록 하는 식이며 Figure 1의 range filter와 동일하다. 이를 간단하게 정리하면 gaussian blurring을 통해 filtering을 진행하고자 하지만 pixel간 intensity의 차이가 큰 edge를 보존하겠다는 의미로 수식을 해석할 수 있다. 즉, 일반적인(간단한) filter가 유일한 convolution kernel을 갖는것에 반해 pixel별로 kernel이 변경된다.  $\frac{1}{W_p}$ 는 가중치의 합을 1로 만들어주는 weight이다.

4	4	5	27	39	1/12	1/6	0
4	5	3	0	42	1/6	1/3	0
3	5	5	23	44	1/12	1/6	0
0	4	5	21	47			
6	7	5	26	46			

Table 1: Image의 pixel값(좌), Bilateral filter의 kernel 값(우)

Table 1은 bilateral filtering의 예시이다. 3x3의 kernel size를 갖는 bilateral filter라고 가정하자. Table 1의 좌측 table의 주황 부분에 대하여 bilateral filtering의 kernel을 구해보면 Table 1의 우측과 같이 나오게 된다. 값이 급격히 변하는 값에 대하여(주황 부분의 우측 값들) kernel size가 모두 0이 됨을 알 수 있다. 이는 Equation 1의 gaussian 미분값에 의해 생긴 결과이다. 이를 이용하면 일반 gaussian filter를 이용했을 때보다 작은 값을 출력으로 하게되는데 주황 부분이 sliding되어 우측으로 이동한다고 가정하면 이번엔 왼쪽 값이 오른쪽 값들에 비해 변화가 작으므로 가중치가 줄어들 것이다. 즉, 이번에는 gaussian filter를 이용했을 때보다 큰 값을 출력으로 하게될 것이다. 이러한 원리로 edge를 보존하여 blur된 image가 아님에도 noise를 효과적으로 제거할 수 있다.

## Median Filter

Median filtering은 한국말로 ”중간값” filtering이다. 이는 간단하게 중심 화소를 기준으로 일정 크기의 kernel size를 가질때, kernel 안에 있는 값들중 중간값을 고르는 filter이다. 보통 pulse가 심하게 차이나는 잡음인 salt and pepper 잡음이나 impulse 잡음에서 자주 사용된다. Pulse가 심하게 차이가 나면 그 값이 가장 큰 값이거나 가장 작은 값일 확률이 높기 때문에 중간값을 선택해주면 해결할 수 있다. Median filter는 앞의 bilateral filter나 gaussian filter처럼 convolutional 연산이 추가적으로 필요하지 않아 computation cost가 적은 장점도 가지고 있다. 하지만 RGB 컬러 영상에 대해서는 세가지 색의 조합에 의존도가 크기 때문에 gray scale에서 사용하는 것을 권장한다.

4	4	5	27	39
4	5	3	0	42
3	5	5	23	44
0	4	5	21	47
6	7	5	26	46

5	5	23
4	5	21
7	5	26

Table 2: Image의 pixel값(좌), 주황 영역의 pixel 값(우)

Table 2는 median filtering의 예시이다. Filter의 figure는 단지 중앙값을 선택하는 filter이므로 table을 설명하는 것으로 대체한다. 주황색 kernel 안의 값을 오름차순으로 나열해보면 4 5 5 5 7 21 23 26이다. 이때의 중앙값은 5이므로 해당 pixel은 5를 가지게된다.

## Average Filter

Average filtering은 중앙값 filtering과 비슷하게 평균값 filtering을 뜻한다. 말 그대로 일정 kernel size 안에 pixel값들의 평균을 값으로 사용한다는 것이다. 값들의 평균을 사용하기 때문에 전반적으로 blurring된 image가 출력으로 나오게된다. Median filter와 굉장히 유사하기 때문에 table을 위주로 예시를 들어 설명하겠다. Table 2와 동일한 kernel을 가진다고 가정하자. Table 2의 우측에 모든 값에 평균을 구하면 11.22 정도의 값이 나온다. 이를 반올림한 값인 11을 출력으로 사용한다. Median filtering은 kernel이 sliding함에 따라 중앙값이 급격하게 변할 가능성이 높지만 즉, edge를 어느정도 보존할 수 있지만 average filtering은 평균을 사용하는 것이기 때문에 median filtering에 비해 비교적 blurry한 image가 나온다.

## Result

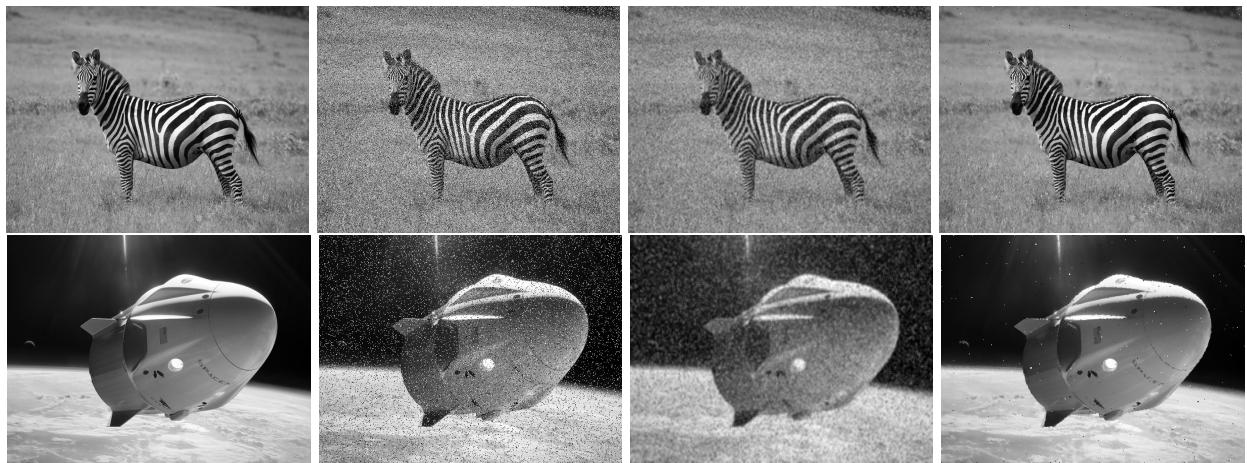


Figure 2: 좌측부터 순서대로 원본, noisy, average filtering, median filtering image

Bilateral filter는 color image에 gaussian noise를 입혀 진행하였고 median과 average filter는 gray image에 salt and pepper noise를 입혀 진행하였다. Figure 2은 gray image의 salt and pepper image에 대한 average, median filtering 결과이다. Average filtering 결과는 salt and pepper noise가 smoothing된 채로 blurry하게 나옴을 알 수 있다. 이에 비해 median filtering 결과는 덜 blurry하게 나왔지만 salt and pepper noise가 간혹 발견되는 것을 알 수 있다.

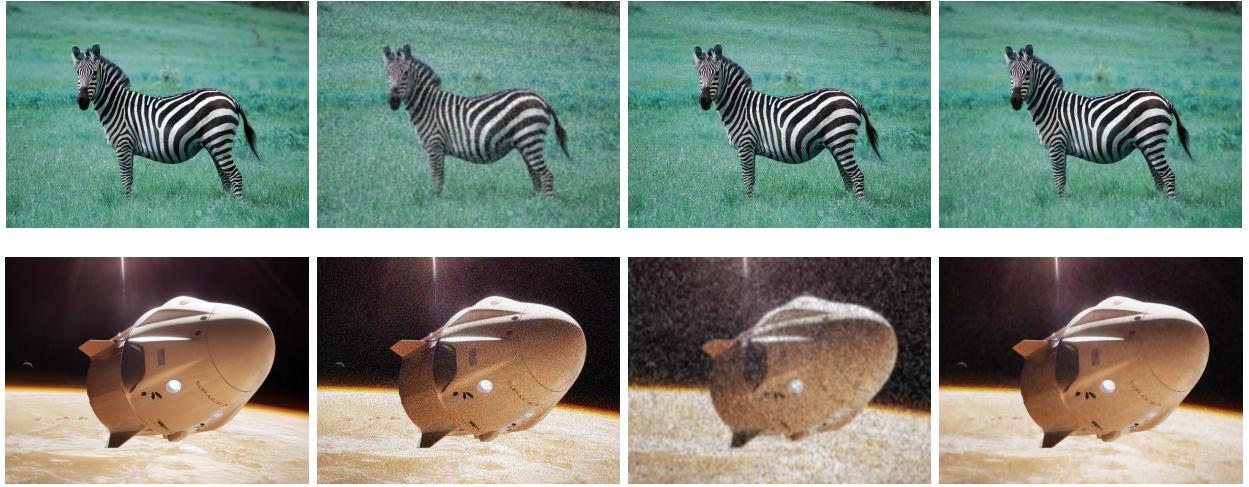


Figure 3: 좌측부터 순서대로 원본, noisy, gaussian filtering, bilateral filtering image

Figure 3은 color image의 gaussian noise에 대한 bilateral filtering 결과이다. Gaussian noise가 눈에 띄게 입혀지진 않았지만 확실히 noise가 있음을 확인할 수 있다. Gaussian filtering을 진행하면 average filtering과 유사하게 전체적으로 blurry한 image를 얻을 수 있다. 얼룩말 image에서는 noise가 조금 제거된 것이 보이지만 우주선 image에선 오히려 더 noise가 심해진 것을 확인할 수 있다. 하지만 bilateral filtering을 이용하면 edge를 유지시키며 noise를 제거하기 때문에 gaussian noise가 전반적으로 잘 복원됨을 알 수 있다.