

#1.

Pinhole 카메라는 렌즈 대신에 작은 구멍을 통해서 사진을 얻을 수 있다. pinhole 카메라를 사용하게 되면 formal circle이 생기게 되는데 이 모양이 pinhole에 의존적이다. 이 circle이 완벽한 원에 가까워야 사진의 대상이 블러하지 않으며. 구멍을 통해서 들어오는 빛의 양이 정기 때문에 각각 노출이 필요하며 꿈직이는 물체를 포착하는데 저작하지 않다. 하지만 이와는 다르게 lens를 사용하게 되면 모든 빛이 lens를 지나 굽절하게되어 image plane에 하나의 점으로 모이게 된다. 그래서 pinhole은 circle이 점으로 합하지 않아 blurry 해 질수 있다.

#2.

Field of view (FOV)는 sensor의 크기와 lens의 배율에 따라 달라지게 된다. FOV는 일종의 카메라의 시야를 나타내며 카메라가 보는 곳이 얼마나 커버리며 담기는지가 커버리의 시야가 된다. 이러한 FOV는 focal length 와 반비례 관계를 가지고 있다. Focal length가 커지면 FOV는 작아지기 때문에 좁은 곳을 자세하게 볼 수 있다. 그렇기 때문에 물체는 더욱 크게 나타나 보일 것이다. FOV에 영향을 주는 요인으로 focal length와 sensor size가 있고 앞서 focal length와는 반비례 하지만 sensor size와는 비례하는 관계를 가진다. 아무래도 sensor size가 커지면 카메라가 볼 수 있는 시야는 넓어지게 된다. Telescopic lens는 일반적으로 멀리 있는 피사체를 찍을 때 사용하며 다른 보통 lens 보다 FOV가 작다. Telescopic lens는 원거리용으로 설계가 되어 있기 때문에 FOV가 작고, 이 많은 focal length가 크다는 것이고, 이는 경통이 길어질수 밖에 없다. 그래서 telescopic lens가 다른 lens보다 더 bulkier 한 것이다.

#3.

Aperture, ISO, shutter speed를 종합하여 빛의 양을 조절할 수 있고, 이 3가지로 부터 exposure을 조절하는 방법이 다양하다. 이 3가지 요소를 적절히 조합하여 사진을 촬영할 수 있다. Aperture는 렌즈를 통해 빛이 들어오는 통로의 크기를 결정하고, Shutter speed는 카메라를 통해 빛이 들어오는 시간을 조절하며, ISO는 카메라 이미지 sensor의 감도를 변환함에 따라 통과하는 빛의 양을 조절해준다. Depth of field (DoF)는 한 사진의 초점이 맞은 것으로 인식이 되는 범위를 말한다. DoF는 카메라가 아닌 사진이 찍히는 물체 끝 범위이고, 만약 초점이 맞은 것으로 인식이 되는 부분(사진의 일부분)이고 나머지가 blurry하게 나타나면 이를 shallow depth of field라고 한다. 이 범위가 확장되어 large depth of field가 되면 나머지 부분도 초점이 맞은 것처럼 인식되어 모두 focus가 된다. 이때, aperture를 확장 열고 하나의 대상에 초점을 맞추기 되면 배경을 흐릿하게 할 수 있다. 그래서 aperture의 구멍이 커지게 되면 통과하는 빛의 양이 많아지고, 구멍이 작아지면 통과하는 빛의 양이 적어진다. 구멍이 작아지게 되면 같은 거리에 대해서 circle of confusion의 크기가 더 작아져 depth of field는 더욱 깊어지게 된다. Circle of confusion의 크기는 초점에서 벗어나면 빛이 날카로운 크기가 점점 커지는 데, image plane에서 아주 가까운 원을 형성하는 포사체의 각 부분들이 결국에는 흐리게 보여서 초점이 맞지 않은 것처럼 보이게 된다. 즉, aperture를 개방하여 shallow depth of field가 만들어진 사진을 찍게 되면, 우리는 우리가 원하는 특정 부분에 초점을 맞춰서 찍고 싶은 것하고, 반대로 aperture를 조이면 large depth of field가 만들어진 사진을 찍게 되면 전체적으로 focus가 확산된다.

Aperture를 크게하면 같은시간 대비 통과하는 빛의 양이 많지만, 그멍이 작으면 같은
양의 빛을 차단하기 위해서는 그만큼 시간을 길게 해야 한다. 하지만 방이나 어두운
장소에서는 빛이 적어 어려움이 있다. 그래서 이러한 시간을 조절하는 것을 shutter가
하고, shutter speed를 증가시켜 더 오랜시간 빛에 노출시킬 수 있다. shutter
speed를 통해 빛이 통과하는 시간을 짧게하면 빠르게 움직이는 피사체도
술간적으로 정지된 것처럼 짙을 수 있다. 하지만 이를 경우 빛이 통과하는 시간이 줄어
자칫 어두운 곳에서는 물체를 파악하기 어렵다. 반대로 shutter speed를 길게
하면 움직이는 물체에 대해서 빛을 오래 노출시키면 물체를 blurry하게 흐리거나
빛의 경로 등에 사건에 냉을 수 있다. ISO는 카메라 이미지 sensor의 감도를
변경해서 흡수하는 빛의 양을 조절한다. 이는 얼마나 빛에 대해 센서가 민감한지를
나타내는 지표다. ISO가 높아질수록 빛에 더욱 sensitive해서 조금의 빛만으로도
정직하게 사진을 찍을 수 있지만, 잘들하면 noise가 증가되는 현상을 발생시킨다.
반대로 ISO가 적을수록 덜 sensitive해서 좋은 exposure에 의해 더 많은 빛을
요구하게 되고 이를 때는 noise 문제는 덜 심하다.

#4.

#3번에서 이야기 했듯이 ISO를 증가시키는 것은 빛에 대해서 더 sensitive 해서 조금의 빛만으로도 적절하게 사진을 촬영할 수 있다. 이는 자칫 조명의 빛만을 사용하고 삶은데 잘 맞는 경우 빛에 더욱 민감해져서 noise를 증가시키는 현상을 발생시키게 된다. 보통 ISO가 2배가 되면 들어오는 빛의 양의 반 정도만 필요로 하기 된다. 그래서 ISO가 50보다 6400인 경우 빛에 대한 sensitivity 가 크게 증가할 것이고, 이는 쉽게 noise를 유발할 것이다.

#5.

Aperture \downarrow \rightarrow depth of field \uparrow \rightarrow blurry \downarrow

shutter speed \uparrow \rightarrow hand shake \uparrow \rightarrow blurry \rightarrow overexposure
(\therefore tripod x)

shutter speed \downarrow \rightarrow light \downarrow \rightarrow dark region 문제 \uparrow

ISO \uparrow \rightarrow light sensitivity \uparrow \rightarrow noise \uparrow

ISO \downarrow \rightarrow noise \downarrow . \rightarrow dark region 문제 \uparrow

빛이 짧은 환경에 대해서 상각대가 없고 noise와 blur를 최대한 줄여야 한다.
이로 인해서 aperture, shutter speed, ISO 값을 조절해야 하는데
한 번에 모든 것을 맞출기 어렵기 때문에 하나씩 조절해보려고 한다. 먼저,
빛이 짧은 환경이기 때문에 빛을 최대한 많이 받고 싶다. 그래서 aperture를
크게하고, shutter speed를 크게하고, ISO도 크게 해서 시작하려고 한다. 그러면
빛에 sensitive하고 같은 시간대비 통과하는 빛의 양도 많아져서 low-light
환경에서 좋은 수준이나, 상각대, noise, blur의 문제를 고려해 써야 한다.
shutter speed를 보면 exposure 시간이 길어지게 되는데. 문제는 상각대가
없이 손떨림 현상으로 blur가 발생할 수 있다. 그래서 이를 방지하고자
shutter speed를 줄여내어야 한다. 하지만 어두운 환경에 대해서

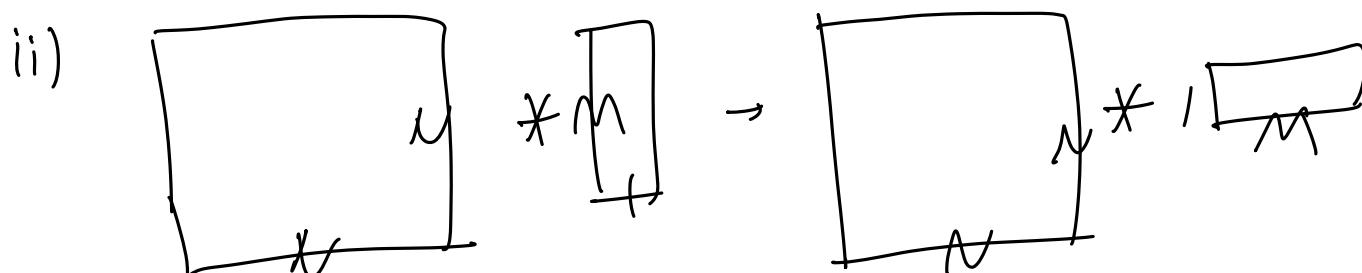
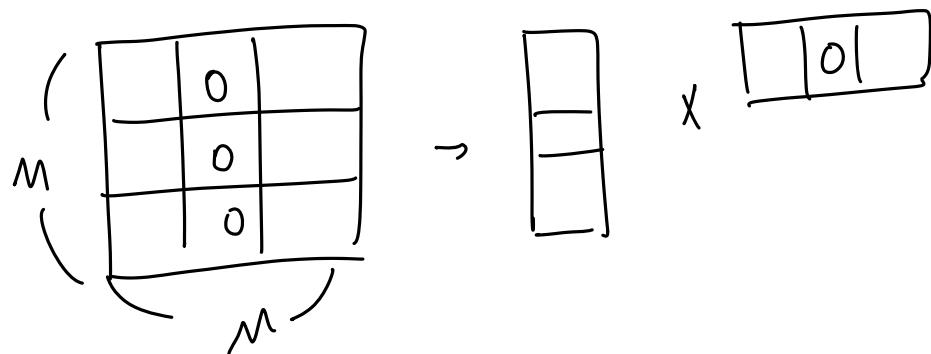
exposure 시간이 너무 짧으면 이미지에 흐를 조절해내기 어렵다. 그리고 ISO를 크게 시작하면 빛이 sensitive해서 좋지만 noise가 심하기에 ISO도 줄여내기의 noise를 해결하기 어렵다. Aperture의 크기는 빛을 많이 통과시키도록 크게 하였는데, 그러면 초점이 부분적으로 맞아보이는 shallow depth of field이 만들어지기 때문에 aperture의 크기를 줄여내면서 전자적으로 focus 상태를 맞춰가면 된다.

#6.

Sampling 과정에서 frequency가 원본보다 최대 frequency의 2배보다 높은 경우에 대해서 output이 오류로 드는 현상을 aliasing이라 한다. 이미지의 크기를 줄일 때 몇몇 픽셀에 대한 정보가 사라지게 되는데 픽셀이 사라져 남은 픽셀들이 원본의 세밀 모양을 만들어 이미지의 결과가 왜곡되어 보이게 된다. 카메라의 성능이나 해상도에 대해서 실험을 하거나, 고사양의 기임을 저사양의 화질로 낮출 때도 aliasing 현상이 발생하게 된다. aliasing 현상을 해결하기 위해서 가장 대표적인 방법은 antialiasing이다. 첫번째로는 Sampling 시 sample을 더 많이 얻는 것이다. 그러나 이 방법은 photodiode의 크기를 줄이고 카메라 시야도 바꾸는 등 실질적인 해결책이 되지 못한다. 두 번째는 신호 자체를 일 wiggly하게 만드는 것이다. High frequency를 제거하면 aliasing을 피할 수 있다. 하지만, 이 또한 정보를 잃어버리는 문제가 있지만, 정보를 잃어버리더라도 aliasing을 막는 것이 이미지 퀄리티에 있어서 더 좋다. High frequency는 low pass filter 등을 이용해 제거하면 된다.

#7.

Sobel filter는 edge 를 검출할 때 사용하는데 사용하는데 Sobel filter 중에서 horizontal Sobel filter를 사용한다고 생각해볼 것이다. 이미지의 크기는 $N \times N$ pixel 이고 filter size는 $m \times m$ 이다. Filtering operation에 대해서 2 가지 방법을 다음과 같이 비교하여 보자.



i) Input: Image, Kernel

Image-new[Image-h][Image-w]

for y in range(Image-h)

for x in range(Image-w)

Image-new[y][x] = 0

for k in range(kernel-h)

for l in range(kernel-w)

Image-new[y][x] += Image[y+k - $\frac{\text{kernel-h}}{2}$]

[x+l - $\frac{\text{kernel-l}}{2}$]

$\times \text{kernel}[k][l]$

$\rightarrow O(N^2 M^2)$

ii) Input : Image, Kernel

Image_new1 [Image-h] [Image-w]

for y in range (Image-h)

 for x in range (Image-w)

 Image_new1 [y] [x] = 0

 for k in range (kernel-h)

 Image_new1 [y] [x] += Image [y+k - $\frac{\text{kernel_h}}{2}$] [x]

 x kernel [k] [0]

Image_new2 [Image-h] [Image-w]

for y in range (Image-h)

 for x in range (Image-w)

 Image_new2 [y] [x] = 0

 for l in range (kernel-w)

 Image_new2 [y] [x] += Image_new1 [y] [x+l - $\frac{\text{kernel_w}}{2}$]

 x kernel [0] [l]

$\rightarrow O(N^2M + N^2M) = O(N^2M)$

\therefore (ii)처럼 나해서 convolution하면 (i)보다 빠르다.

#8.

Input : Image(I), Kernel(f)

$$I' = P(I)$$



Fourier Transform 하기 전에 이미지에
padding 방법 중 하나를 적용
(zero padding, reflective padding, ...)

$$f' = SP(f)$$

$$F = FFT$$

$$F^{-1} = \text{inverse FFT}$$



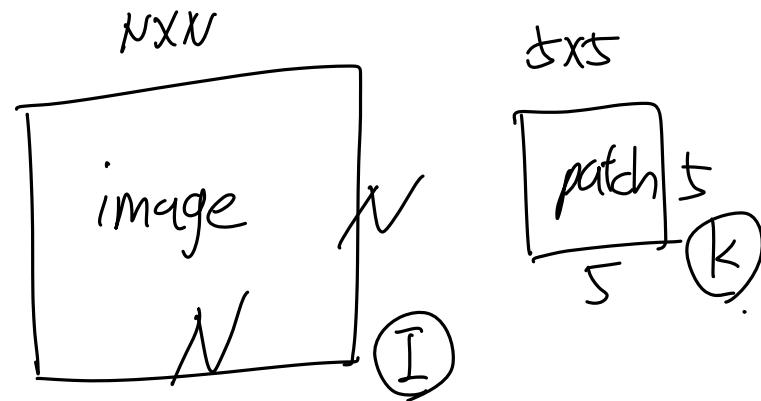
kernel을 image I' 과 같이 size가
같도록 zero padding 후 center로
shifting 해준다.

$$\text{result} = F^{-1}(F(I') \cdot F(f'))$$

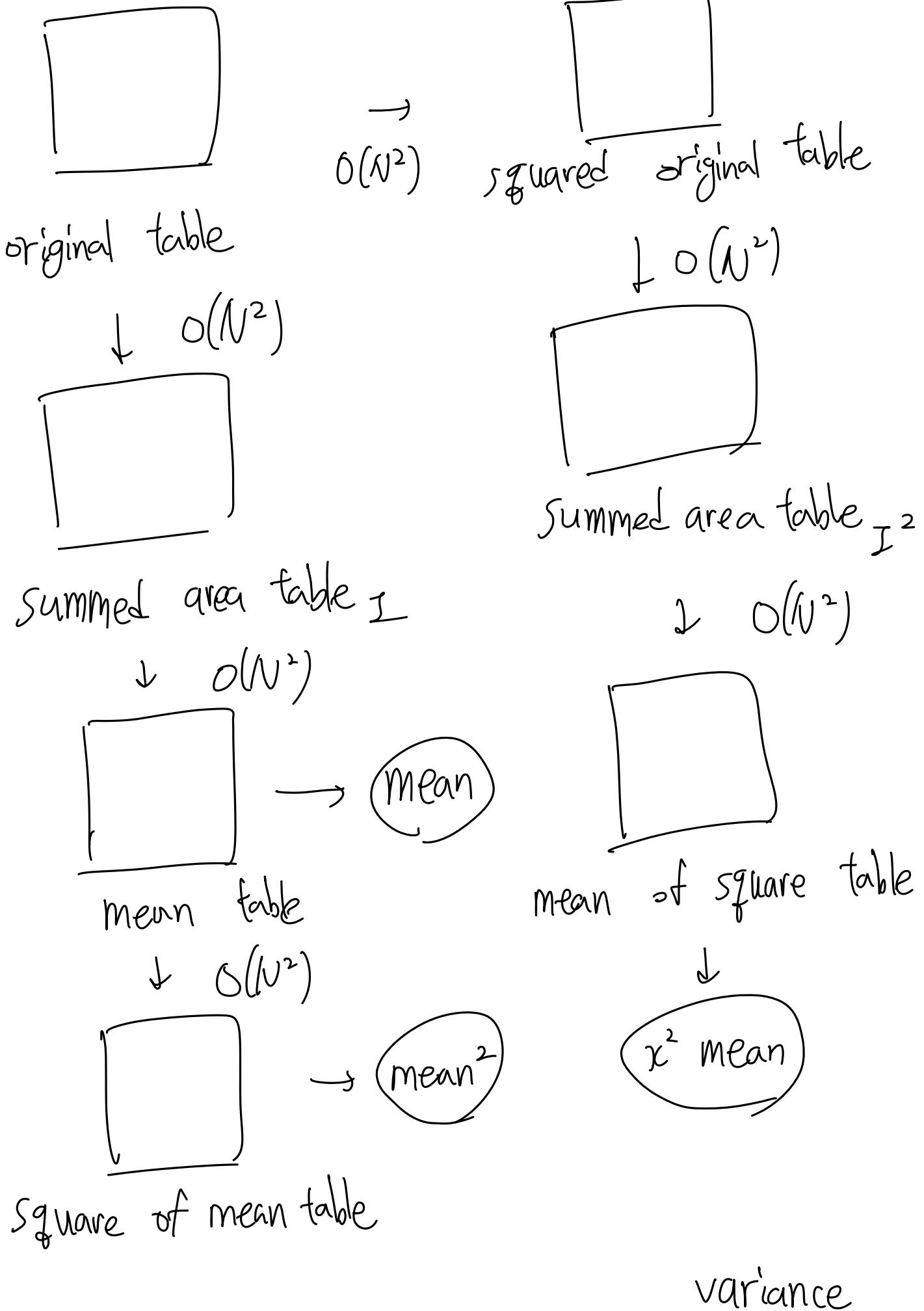


마지막 I' 와 f' 에 대해서
Fourier transform을 해주 후
frequency domain에서
 f' 을 제거하고 다시 inverse
Fourier transform을 해서
spatial domain으로 결과를
보내준다.

#9.



mean 과 variance를 계산하는지 위해서 먼저 01번에 대해서
horizontal 방향으로 sum을 계산하고 vertical 방향으로 다시
sum을 계산해 sumaed area table을 만들 것이다. 이를
5x5 patch를 통해서 mean을 계산하는 것은 summed area
table로부터 t와 -o연산을 통해서 해당 3가지의 summation을
계산할 수 있고 patch size로 나눠주면 mean 값을 계산할 수
있다. variance는 summed area table로부터 mean
과 pixel intensity의 제곱의 mean을 계산해 -o연산을
통해서 계산된다.



Summed area table $I = f_{\text{sum}}(I) = S_I$

mean table $= f_{\text{mean}}(S) = M_I$

square of mean table $= f_{\text{square}}(M_I) = M_I^2$

Squared original table $= f_{\text{square}}(I) = I^2$

Summed area table $I^2 = f_{\text{sum}}(I^2) = S_{I^2}$

mean of square table $= f_{\text{mean}}(S_{I^2}) = M_{I^2}$

variance $= M_{I^2} - M_I^2$

mean과 variance 모두 Summed area table을

만들어 진짜라면 계산 과정은 빨라지기도 실질적인

연산량은 적어 훨씬 $O(N^2)$ 에 가깝다.

Summed area table을 만들고 계산하고 나서 각각에
전부 $O(N^2)$ 이 때문이다.

#10.

Bilateral filtering은 다음과 같다.

$$I[m, n] = \frac{1}{w_{mn}} \sum_{k, l} g[k, l] r_{mn}[k, l] f[m+k, n+l]$$

g : spatial weighting function

r_{mn} : intensity range weighting function

w_{mn} : normalization factor

$$g[k, l] = \exp\left(-\frac{k^2 + l^2}{2\sigma_s^2}\right)$$

$$r_{mn}[k, l] = \exp\left(-\frac{|f[m+k, n+l] - f[m, n]|^2}{2\sigma_r^2}\right)$$

spatial weighting function은 target 픽셀에 가까울수록 큰 값을
공유하게 된다. Intensity range weighting function은 target 픽셀과
비슷한 intensity 값을 가지는 픽셀에 더 큰 weight를 부여한다. 그래서
spatial weighting function과 intensity range weighting
function 모두 gaussian function이며 spatial intensity function은
픽셀 distance를 기본으로, intensity range weighting function은

input 이미지의 intensity difference 를 기반으로 한다.

Bilateral filter는 2개의 gaussian function을 사용하기 때문에 parameter 2개인 σ_s 를 가진다. spatial weighting function의 σ_s 와 intensity range weighting function은 σ_r 이다.

σ_s 는 spatial weighting function의 크기를 조절하고, σ_r 은 intensity range weighting function의 크기를 조절한다. σ_s 의 값을 커두면 spatial weight의 범위를 넓힐수되어 작은 디테일들을 효과적으로 제거할 수 있다. σ_r 의 크기가 작다는 것은 서로 차이가 매우 작은 작은 intensity 간들을 이용해서 blend하기를 원하는 것이다. 반대로 σ_r 의 크기가 크다는 것은 서로 차이가 매우 큰 intensity인 경우에도 큰 weight를 부여해주는다는 이야기이다. $\sigma_r = 0$ 으로 한대는 모든 intensity 값에 constant weight를 부여해 더 이상 intensity range weighting function을 사용하지 않게 된다는 것이다. 이 2개의 parameter 들의 서로 다른 조합을 통해서 다른 결과를 만들 수 있게 된다. σ_r 의 값을 커두면 intensity 값의 차이가 큰 edge를 제거할 수 있다.