

# HOME EXAM

---

## FYS-2010-Digital image processing

March 12, 2019

Candidate number: 25

University of Tromsø

Contains 12 pages, including frontpage.

## Contents

<b>1</b>	<b>Part A</b>	<b>3</b>
1.1	1 . . . . .	3
1.2	2 . . . . .	3
1.3	3 . . . . .	3
<b>2</b>	<b>Part B</b>	<b>3</b>
2.1	1 . . . . .	3
2.2	2 . . . . .	4
2.3	3 . . . . .	4
2.4	4 . . . . .	5
2.5	5 . . . . .	6
<b>3</b>	<b>Part C</b>	<b>8</b>
3.1	1 . . . . .	8
3.2	2 . . . . .	10
3.3	3 . . . . .	11
<b>4</b>	<b>Appendix</b>	<b>12</b>
<b>5</b>	<b>References</b>	<b>12</b>

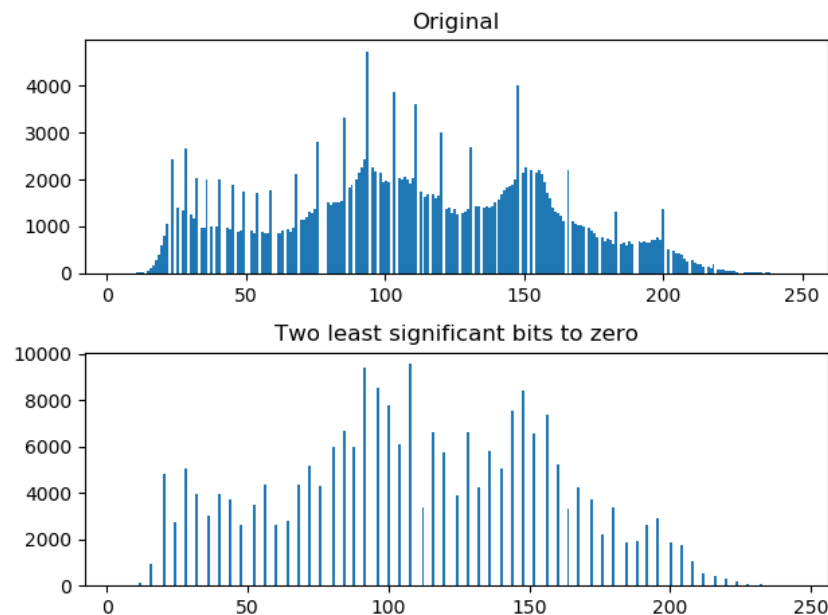
## 1 Part A

### 1.1 1

### 1.2 2

### 1.3 3

The bit plane which represents the most significant bits are the one with most details (*g7*). By setting the two least significant bits to zero shift the intensities to only be  $4 \cdot k$  intensity, where  $k$  is an integer. So here we would only have 4 shades between the intensities in the image. (the human eye cannot perceive the difference of abrupt change of such few intensities [p.626 Rafael C. Gonzales [2018]]). The histogram would look like figure(1)

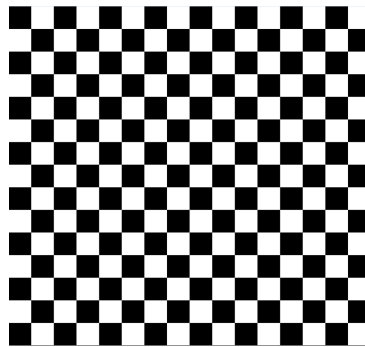


**Figure 1:** How the histogram changes when setting the two least significant bits(LSB) to zero.

## 2 Part B

### 2.1 1

Spatial aliasing is loss of information by not using sufficient sampling rate, depending on the change of frequencies in the intensities one want to have an image of. This follows from the Nyquist sampling which says we need to sample with  $f_s = 2f$  where  $f$  is the *real* frequency and  $f_s$  is the sampling frequency. If we use a checkerboard as an example,



**Figure 2:** An image of a checkerboard. Image taken from [Rafael C. Gonzales [2018]].

If we sample with less pixels (or increase the frequency rate (the rate of changing intensities) of the object we are imaging) we could end up with only getting every cycle of the black intensity such that we only get a black strip. We could also use a different sampling rate and end up getting an "Fake" image by reconstructing every second black intensity with one white in between such as shown in the sliced checkerboard in figure (3) below.



**Figure 3:** Aliased slice

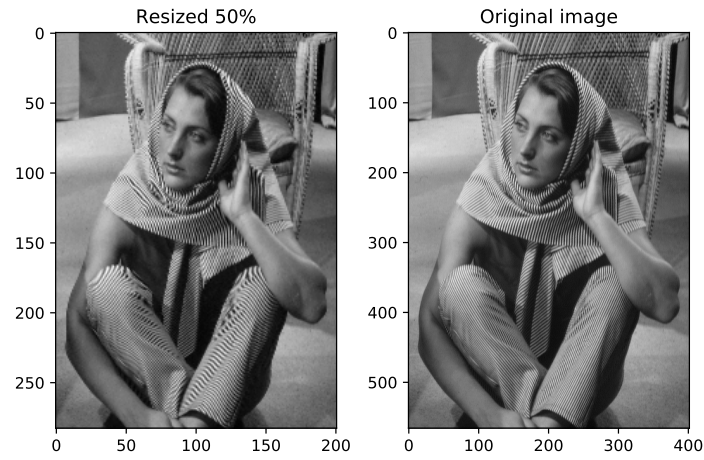
By sampling with less pixels we could end up having the "same" image, but with a longer period between the intensities. This also applies to resizing because when we resize down an image we need to remove pixels, which would destroy some of the information in the image. If we then resize it to the original shape we would apply interpolation and would most likely change the real intensity valued pixels to some other neighboring pixel value and therefore damage the quality of the image.

## 2.2 2

In figure 4.19 in the book [Rafael C. Gonzales [2018]] we can clearly see the aliasing effect in the area of the high frequency pixels. On the scarf, pants and the chair in the background the intensity values changes rapidly (high frequency). Therefore when resizing we would change the sampling rate below the Nyquist and lose information which we can't retrieve when resizing back to original shape.

## 2.3 3

We resize the image to 50% of its original size, the result can be shown in figure(4) below. The resizing is done by only taking out every second pixel in the original image into a new image.



**Figure 4:** Resized image to 50% of its original size, we can clearly observe the aliasing in the high frequency areas of the image.

In the figure we can see the aliasing effect as the high frequency areas won't be sampled at the right sampling rate to observe the rapid changing pattern.

## 2.4 4

We can blur the images prior to resizing by convolving an averaging filter with the image as follows,

$$g(x, y) = h(x, y) \star i(x, y) \quad (1)$$

where  $h$  is the average filter defined as in equation(2) and  $i(x, y)$  is the image,  $\star$  means convolve.

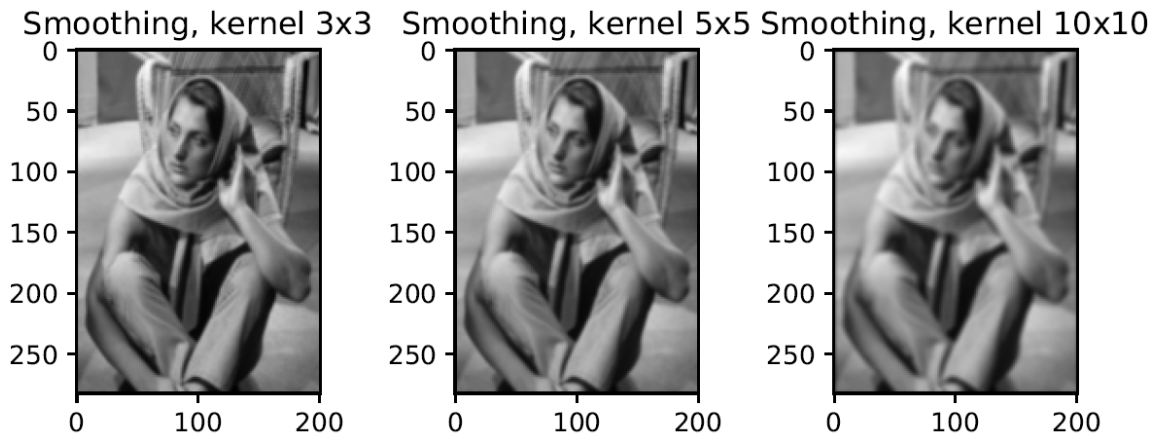
$$h = \frac{1}{MN} \begin{bmatrix} 1_{1,1} & 1_{1,2} & 1_{1,3} & \dots & 1_{N,1} \\ 1_{1,2} & 1_{2,2} & 1_{3,2} & \dots & 1_{N,2} \\ & \dots & \dots & \dots & \dots \\ 1_{1,M} & 1 & 1 & \dots & 1_{M,N} \end{bmatrix} \quad (2)$$

when convolving we are applying the filter in the spatial domain, since this is an averaging filter/-moving averaging(as it traverses the image), we essentially apply a lowpass filter. Lowpass filter will only allow the low frequency components of the image to go through. This means that the image will not have rapid changing intensities(more smudged out/smooth). This is clearly visible in figure(5). This will decrease the effect of aliasing. In fact this is how many of the anti-aliasing setting in video games work.[Wikipedia, the free encyclopedia [2019]] When applying a filter we want to zero pad our image, in the case of convolving in spatial domain we would have problems fitting the kernel on the edges of the image. In our case the the convolve method from scipy module in python does this.

```

1  # Convolve using scipy package
2  from scipy.signal import convolve2d
3  mask = conconvolve2d(image, laplacian, mode = 'same')
4

```



**Figure 5:** Smoothed the aliased image with different sizes of the averaging kernel

## 2.5 5

We can try to restore the image by using an sharpening filter we would emphasize the edges better by using filters such as Laplace. Laplace is a second order derivative so here we would emphasize pixels with high frequency and de-emphasize areas with low frequency (high pass filter). We use the Laplace kernel given in equation(3),

$$h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (3)$$

This stems from the fact that we use the Laplacian  $\nabla^2 f(x, y)$  which will generate the discrete version  $f[x, y+1] + f[x+1, y+1] + f[x+1, y] + f[x-1, y] + f[x-1, y-1] + f[x+1, y-1] - 4f[x, y]$  (needs to sum to zero). This is linear and therefore allowed to be used in convolution. By doing this convolution we get a mask  $g(x, y)$ ,

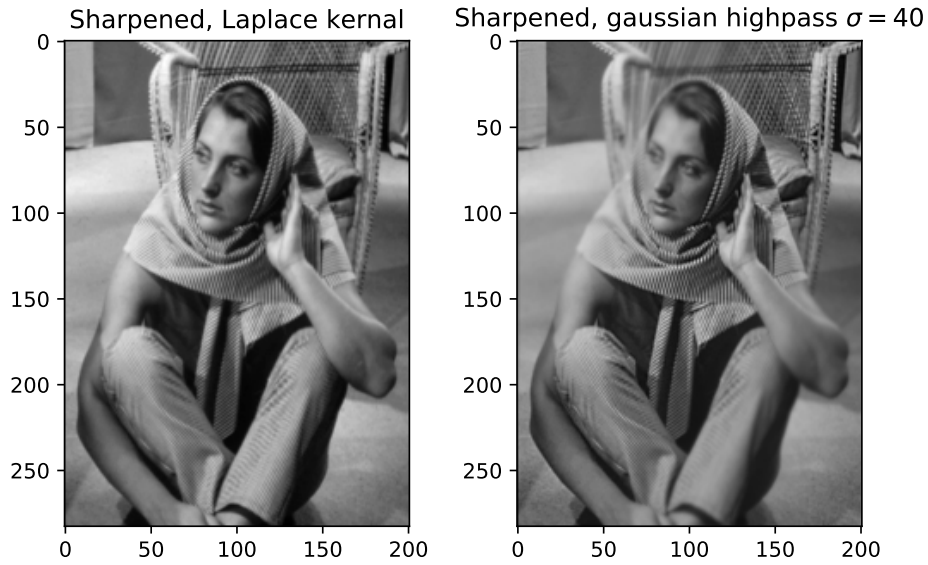
$$g(x, y) = h(x, y) \star i(x, y) \quad (4)$$

here  $h(x, y)$  is the Laplacian filter and  $i(x, y)$  is the image. To enhance the edges we add this mask to our image. By using the Laplace kernel in equation(3) we will sharpen in all direction. We could for instance use a star-shaped Laplace to neglect diagonal direction. (set diagonals in the matrix to zero.)

$$E(x, y) = i(x, y) + Cg(x, y) \quad (5)$$

Where  $E(x,y)$  is the enhanced image and  $C$  is an constant which is either -1 or 1 depending on which Laplacian kernel we use. In this case  $c = -1$ .

The result of the sharpening with two different methods are shown in figure(6).



**Figure 6:** Sharpened by using two different sharpening techniques.

In this figure we can see that it has more defined edges. The gaussian high pass sharpening method contains more blurring, but we could tune the parameter  $\sigma$ (cut off) to fit better. The image is much better then the resized version which contained aliasing.

The gaussian high pass filter uses the gaussian function (which is a lowpass filter),

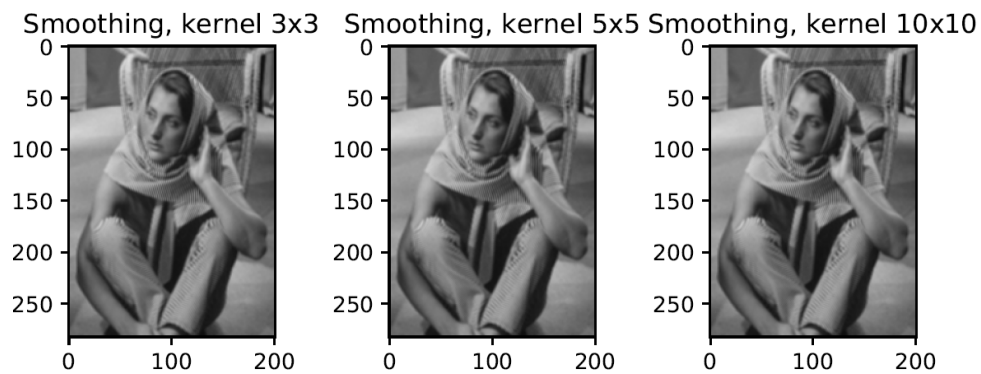
$$H_{lp}(u, v) = e^{D^2/2\sigma^2} \quad (6)$$

This function is in the frequency domain,  $D$  is the distance from the centered frequency and  $\sigma$  is the radius parameter. To get the high pass filter we need to subtract it from one.

$$H_{hp}(u, v) = (1 - H_{lp}) \quad (7)$$

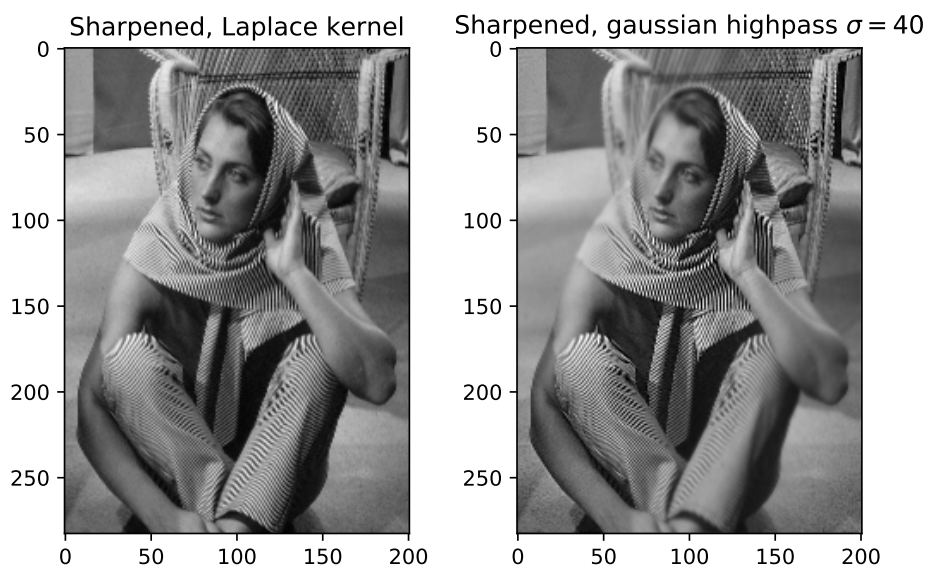
To apply this filter we multiply it by the centered frequency domain representation of the image  $I(u,v)$ .

In the previous case we blurred the image PRIOR to resizing, this made the image look good and without aliasing. We have these images in figure(7).



**Figure 7:** Smoothed images on images blurred after resizing.

If we blur the image after resizing and using Laplace sharpening we get the images in figure(8) below.



**Figure 8:** Sharpened by using two different sharpening techniques on image blurred after resizing.

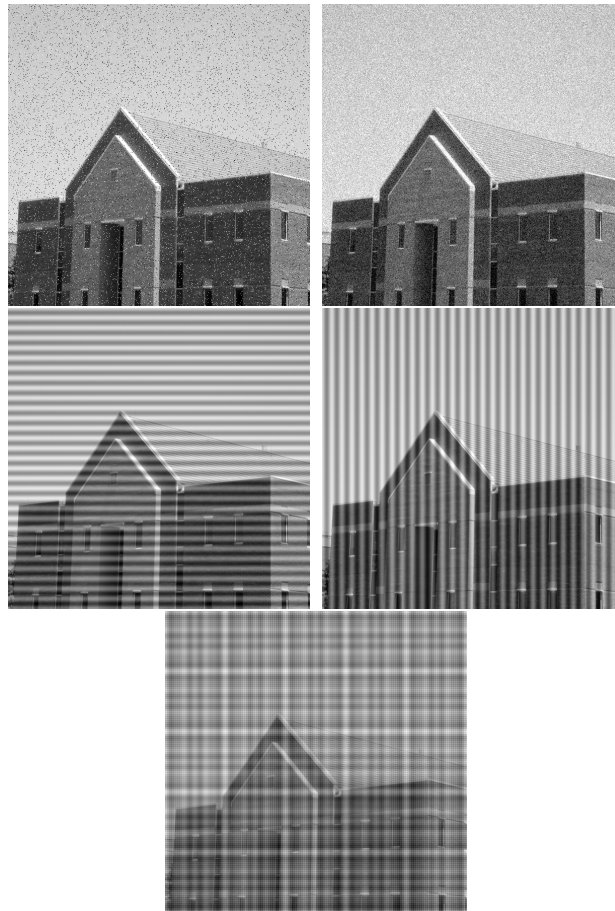
In this case we can see that the aliasing is back, so here we can see how sharpening works counter to blurring.

### 3 Part C

#### 3.1 1

By visually inspecting the images, in *F1.png* we can clearly see the distinctive *salt & pepper* noise with its black/white broken pixels. In *F2.png* it looks like gaussian noise, in *F3.png* we have vertical periodic

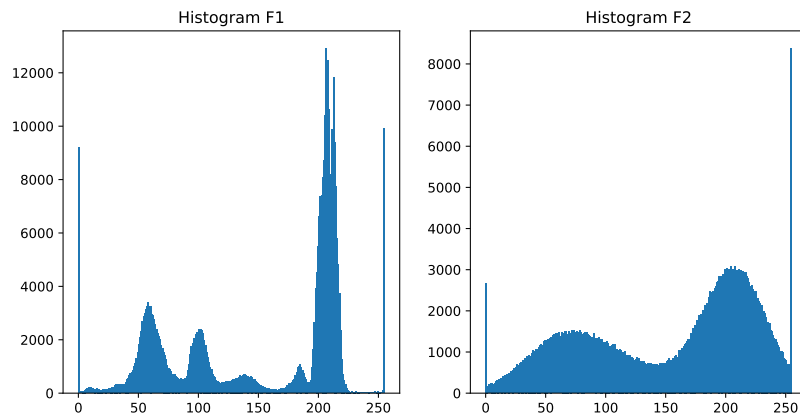




**Figure 9:** Image F1.png, F2.png, F3.png, F4.png, F5.png

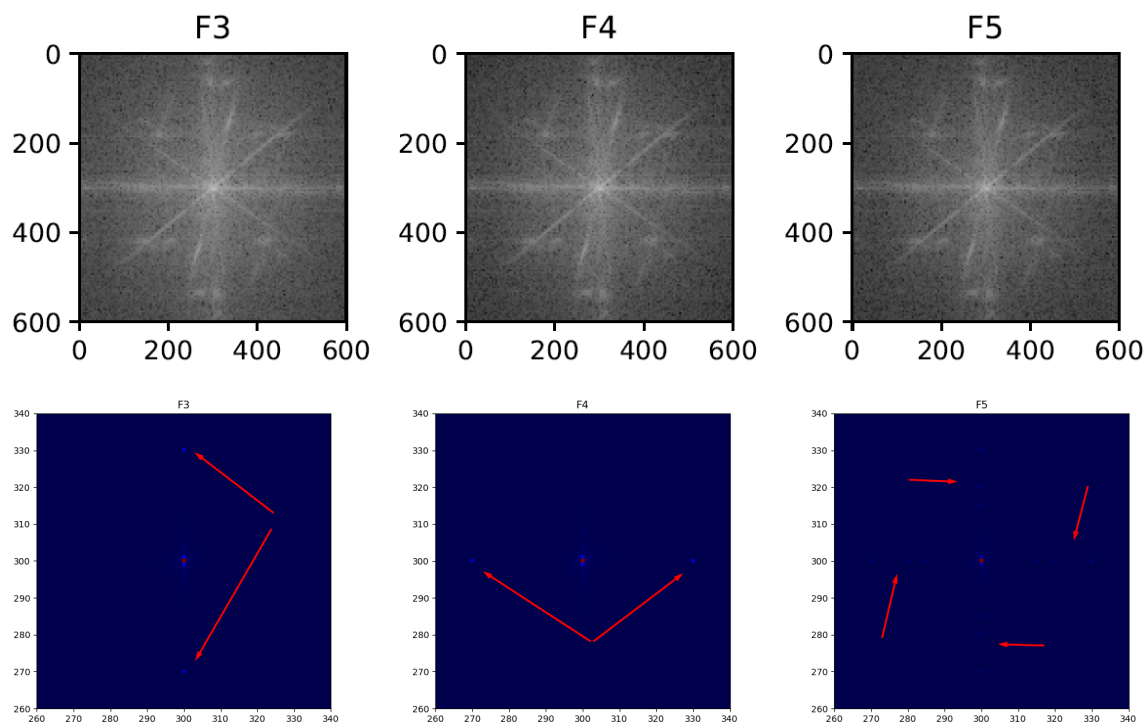
noise, in *F4.png* we have horizontal periodic noise. In the last image *F5.png* we have a superposition of horizontal and vertical periodic noise. The images of shown in figure(9).

We could find the histogram representation of the image to see the noise pattern, in figure(10) we can see that the gaussian noise form gaussian structures in the histogram. The salt and pepper noise has large spike at the white and black intensity(0 and 255), but the overall structure of the image stays intact, its much harder to know the structure of the image with the gaussian noise.



**Figure 10:** Histogram of the images

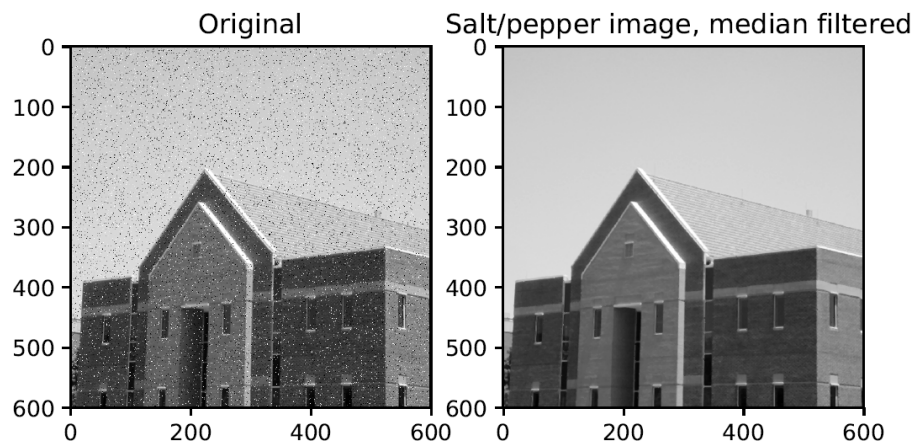
The periodic noises would be a lot easier to see in the frequency domain as sine waves in frequency domain is an conjugated impulses. In figure(11) below we can see the impulses.



**Figure 11:** First row show the frequency representation of the images in dB scale. In second row we can see the impulses more clearly.

### 3.2 2

For removing salt and pepper noise a median filter would be a good choice. With the median filter we rank the values from lowest to highest value and choose the 50% percentile intensity value in the kernel and set this value as the pixel of the kernel, we do this for all the pixels in the image.



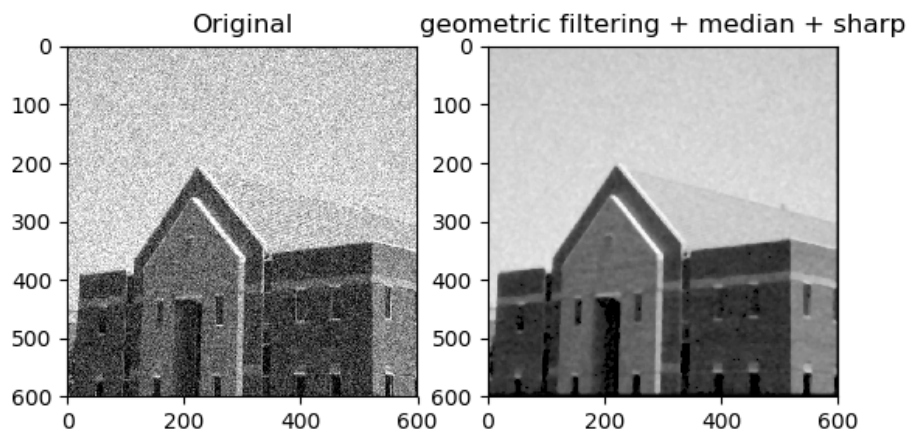
**Figure 12:** F1 image filtered using median filter with 3x3 kernel.

Here  $g(r,c)$  is the values from the image which are inside our convolving kernel.

For the gaussian noise image we use an geometric filter defined as,

$$\hat{f}(x,y) = \left[ \prod_{(r,c) \in S_{xy}} g(r,c) \right]^{1/MN} \quad (8)$$

The geometric filter will blur the image like the averaging filter, but to another degree. In the figure below we used kernel 3x3. After applying the geometric filter we still had some black spots in our image, like we had pepper noise, therefore we applied median filter with kernel 7x7. These filters made our very smooth, so we applied Laplacian filter to emphasize the edges. The result is shown in figure(13) below.



**Figure 13:** Filtered gaussian noise image, here we applied the geometric filter, median and sharpening

### 3.3 3

## **4 Appendix**

## **5 References**

Richard E. Woods Rafael C. Gonzales. *Digital Image Processing*. Pearson, fourth edition, 2018.

Wikipedia, the free encyclopedia. Spatial\_anti-aliasing, 2019. URL [https://en.wikipedia.org/wiki/Spatial\\_anti-aliasing](https://en.wikipedia.org/wiki/Spatial_anti-aliasing). [Online; accessed 12.03.2019].