FILTERING

David Esparza Alba
Ritsumeikan University

Filtering

- Sometimes images present noise that complicates working with them. Most algorithms are very sensitive to noise, and for that reason is necessary to eliminate or reduce noise.
- Filters are used to reduce the amount of noise in an image.
- There are different kinds of filters and in for some cases some filters work better than otters.

Filtering

- We are going to analyze and compare different kind of filters, such as:
  - Mean Filter
  - Median Filter
  - Gaussian Filter
  - Laplacian Filter
  - Sobel Filter
  - Prewitt Filter
  - Laplacian of Gaussian Filter

Noise

David Esparza Alba

Ritsumeikan University

Noise

- Noise can be defined as a variation of intensity or color in an image.
- Noise can occur for several reasons, luminosity, wrong functionality of the camera, environment conditions, etc.
- When working with images, we must assume that noise can be presented in the image. Then, the value of a pixel in position (x,y), is given by the following expression:

$$P_{x,y} = f(x,y) + e_{x,y}$$

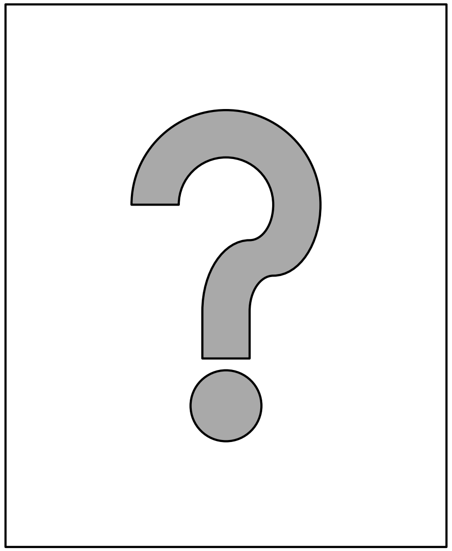
Noise

- Noise can be defined as a variation of intensity or color in an image.
- Noise can occur for several reasons, luminosity, wrong functionality of the camera, environment conditions, etc.
- When working with images, we must assume that noise can be presented in the image. Then, the value of a pixel in position (x,y), is given by the following expression:

$$g(x,y) = \underbrace{\boxed{f(x,y)}}_{\text{Real value of pixel (x,y)}} + \overbrace{\boxed{\epsilon(x,y)}}^{\text{Noise of pixel (x,y)}}$$

Noise

- What would be the real image we will be working with?



Original Image

45	67	120	12
45	89	128	255
10	15	12	43
27	7	35	26

Noise

2	-4	1	3
-2	0	-1	-1
0	-2	-5	2
1	0	0	-1

Noise

Real Image

45 + 2	67 + -4	120 + 1	12 + 3
45 + -2	89 + 0	128 + -1	255 + -1
10 + 0	15 + -2	12 + -5	43 + 2
27 + 1	7 + 0	35 + 0	26 + -1

Original Image

45	67	120	12
45	89	128	255
10	15	12	43
27	7	35	26

Noise

2	-4	1	3
-2	0	-1	-1
0	-2	-5	2
1	0	0	-1

Noise

Real Image

47	63	121	15
43	89	127	254
10	13	7	45
28	7	35	25

Original Image

45	67	120	12
45	89	128	255
10	15	12	43
27	7	35	26

Noise

2	-4	1	3
-2	0	-1	-1
0	-2	-5	2
1	0	0	-1

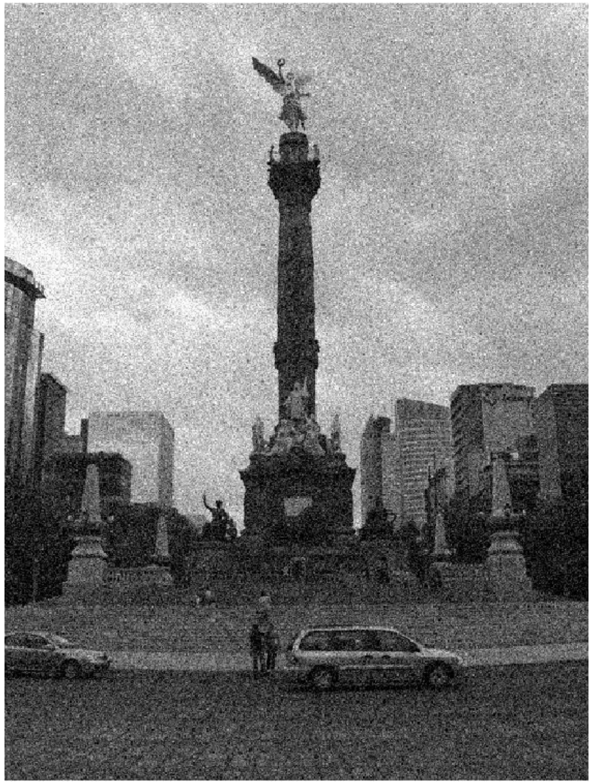
Noise

- There are several kinds of noise, such as:
  - Gaussian Noise
  - “Salt and Pepper” Noise
  - White Noise
  - Film Grain
  - etc.
- We are not to get deeper into this subject, instead we are going to focus in different kinds of methods to reduce the amount of noise.

Noise

Original Image

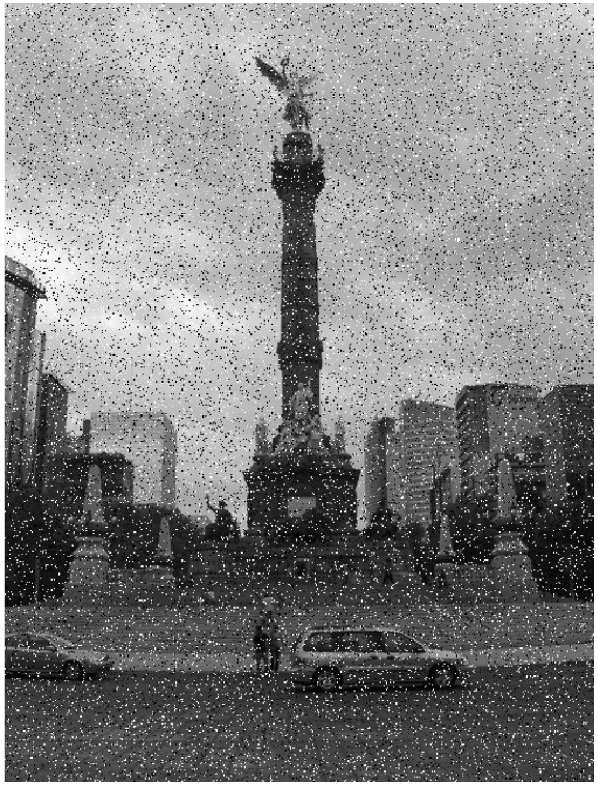
Gaussian Noise



Noise

Original Image

“Salt and Pepper” Noise



Filters

David Esparza Alba

Ritsumeikan University

Filters

- Filters help us to reduce the amount of noise and to detect some important features as edges.
- The value of the pixels change when we are using filters. These new values are obtained using the information of the neighbor pixels.
- Most of the filters use kernels to obtain the new values of the image.
- A kernel is a k x k matrix that is convolved with the image.

Convolution

- The convolution operator is defined with the following expression:

$$g = f \star h$$

- In our case, f represents the image and h the kernel, and the result is given by g. Each pixel of the new image is obtained by:

$$g(i,j) = \sum_{l} \sum_{k} f(i+k, j+l) h(k,l)$$

Convolution

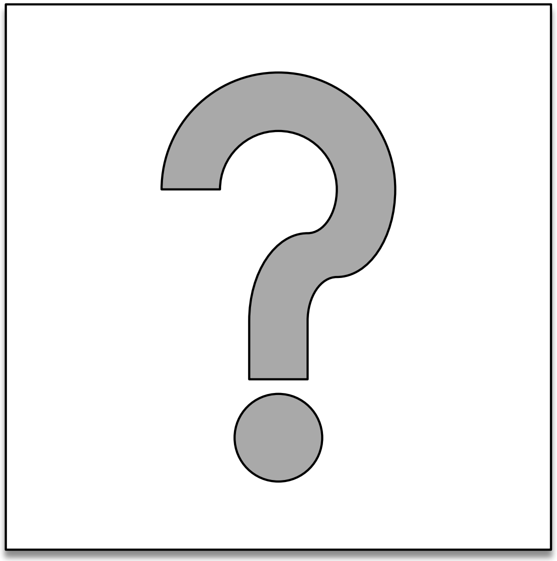
45	60	25	30	45	70	85	90
50	60	35	33	65	56	170	95
52	63	40	85	34	87	182	95
55	78	54	90	23	31	198	100
70	100	64	70	89	12	237	125
75	114	68	75	32	47	225	137
80	118	60	67	124	87	200	140
84	127	55	65	231	34	190	180

$\star$

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

$h(x,y)$

$=$



$f(x,y)$

$g(x,y)$

Convolution

45	60	25	30	45	70	85	90
50	60	35	33	65	56	170	95
52	63	40	85	34	87	182	95
55	78	54	90	23	31	198	100
70	100	64	70	89	12	237	125
75	114	68	75	32	47	225	137
80	118	60	67	124	87	200	140
84	127	55	65	231	34	190	180

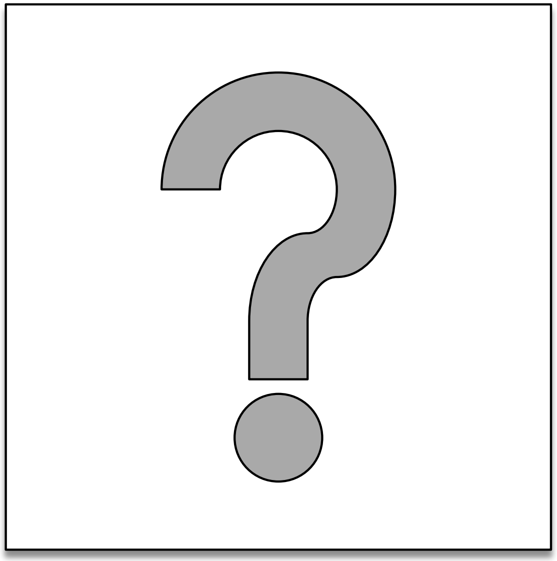
$$f(x, y)$$

★

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

$$h(x, y)$$

=



$$g(x, y)$$

60	35	33
63	40	85
78	54	90

window

Convolution

45	60	25	30	45	70	85	90
50	60	35	33	65	56	170	95
52	63	40	85	34	87	182	95
55	78	54	90	23	31	198	100
70	100	64	70	89	12	237	125
75	114	68	75	32	47	225	137
80	118	60	67	124	87	200	140
84	127	55	65	231	34	190	180

$\star$

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

$h(x, y)$

$=$

45	60	25	30	45	70	85	90
50	60	35	33	65	56	170	95
52	63	58	85	34	87	182	95
55	78	54	90	23	31	198	100
70	100	64	70	89	12	237	125
75	114	68	75	32	47	225	137
80	118	60	67	124	87	200	140
84	127	55	65	231	34	190	180

$f(x, y)$

$g(x, y)$

Noise Reduction Filters

David Esparza Alba

Ritsumeikan University

Mean Filter

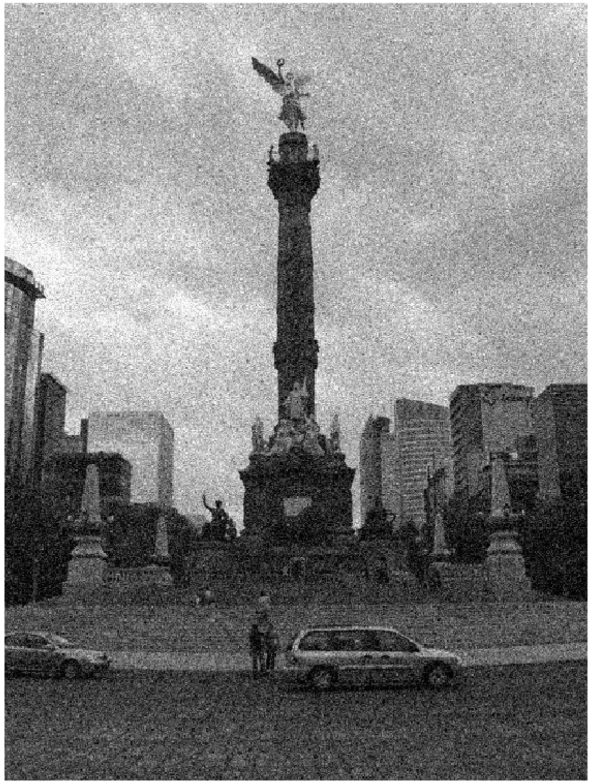
- Mean filter computes the mean of all elements in the window and assign that value to the new pixel.
- The 3 x 3 kernel of the mean filter looks like this:

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

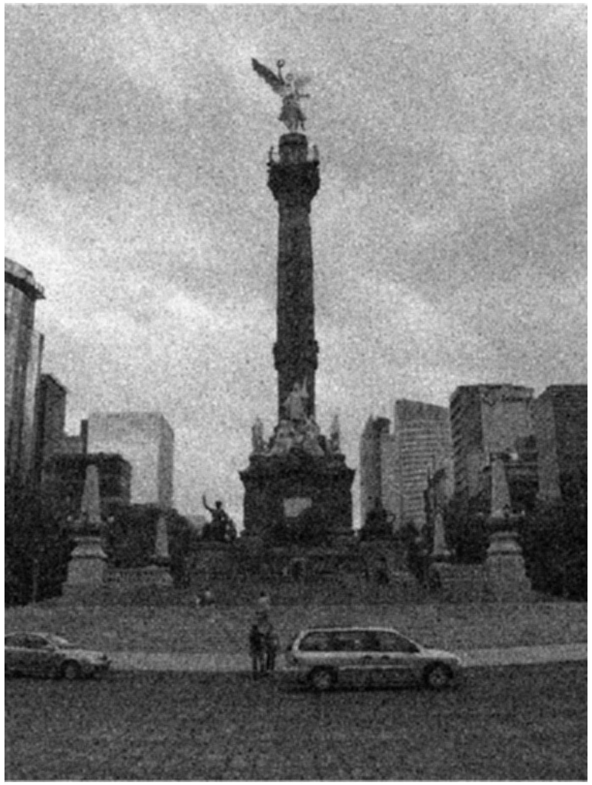
- The resulting image will be a smoother image, reducing the amount of noise.
- The size of the kernel can change, depending of the application. The value of each element is $1/k^2$.

Mean Filter

Image with Gaussian Noise



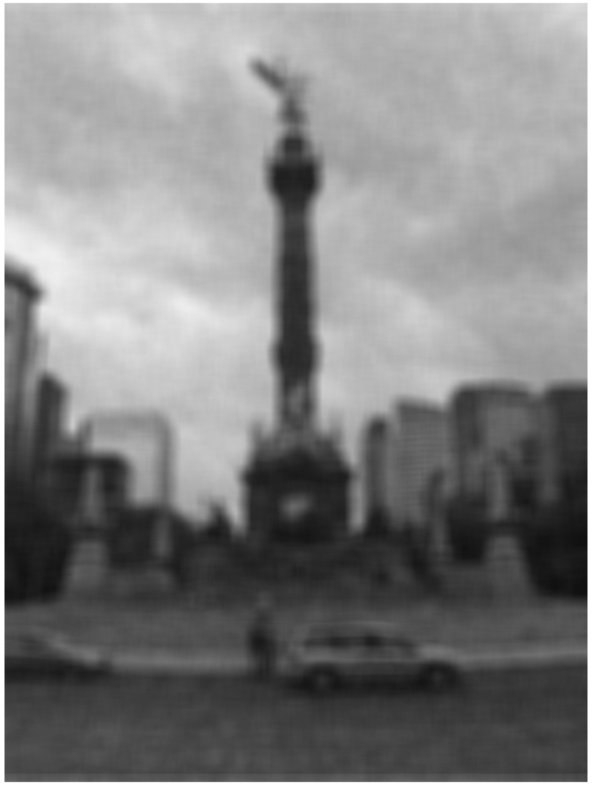
Smoothed Image Using a 3 x 3 Mean Filter



Mean Filter

Smoothed Image Using a 7 x 7 Mean Filter

Smoothed Image Using a 15 x 15 Mean Filter



Median Filter

- Median filter assign to the new pixel the value of the median of the values inside the window.
- There is no kernel for this filter.
- Be S the set of elements inside the window sorted according to their value.
- Assign to the new pixel the value of the middle element in S, according to the following expression:

$$g(x,y) = \begin{cases} S_{n/2} & \text{if } n \text{ is odd} \\ \frac{S_{n/2}+S_{n/2-1}}{2} & \text{if } n \text{ is even} \end{cases}$$

Median Filter

- What would be the value of the new pixel?

45	60	25	30	45	70	85	90
50	60	35	33	65	56	170	95
52	63	40	85	34	87	182	95
55	78	54	90	23	31	198	100
70	100	64	70	89	12	237	125
75	114	68	75	32	47	225	137
80	118	60	67	124	87	200	140
84	127	55	65	231	34	190	180

Median Filter

- What would be the value of the new pixel?

45	60	25	30	45	70	85	90
50	60	35	33	65	56	170	95
52	63	40	85	34	87	182	95
55	78	54	90	23	31	198	100
70	100	64	70	89	12	237	125
75	114	68	75	32	47	225	137
80	118	60	67	124	87	200	140
84	127	55	65	231	34	190	180

Sorted Elements:

33
35
40
54
60
63
78
85
90

Median Filter

- What would be the value of the new pixel?

45	60	25	30	45	70	85	90
50	60	35	33	65	56	170	95
52	63	40	85	34	87	182	95
55	78	54	90	23	31	198	100
70	100	64	70	89	12	237	125
75	114	68	75	32	47	225	137
80	118	60	67	124	87	200	140
84	127	55	65	231	34	190	180

Sorted Elements:

33
35
40
54
60
63
78
85
90

Median Filter

- What would be the value of the new pixel?

45	60	25	30	45	70	85	90
50	60	35	33	65	56	170	95
52	63	60	85	34	87	182	95
55	78	54	90	23	31	198	100
70	100	64	70	89	12	237	125
75	114	68	75	32	47	225	137
80	118	60	67	124	87	200	140
84	127	55	65	231	34	190	180

Sorted Elements:

33
35
40
54
60
63
78
85
90

Median Filter

Image with "Salt and Pepper" Noise

Smoothed Image Using a 3 x 3 Median Filter

Median Filter

Smoothed Image Using a 5 x 5 Median Filter

Smoothed Image Using a 9 x 9 Median Filter

Gaussian Filter

- The Gaussian filter convolves the image with a Gaussian function with mean μ and standard deviation σ.

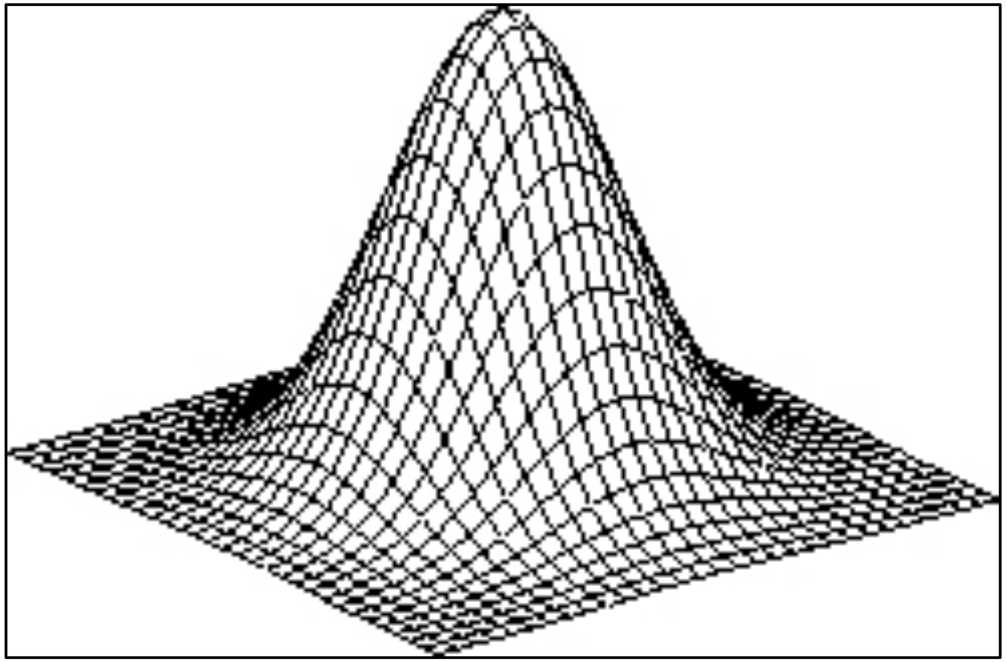
- This filter uses as a kernel a gaussian function, where the centered value has the largest value, which decreases symmetrically as distance from the center increases.

- If the center element of the kernel is located at (0,0), then the rest of the values of the kernel are obtained with:

$$h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Gaussian Filter

The values of the kernel comes from a Gaussian function.



Gaussian Filter

- Example of a 5 x 5 Gaussian Kernel with $\sigma = 1$

0.00	0.01	0.02	0.01	0.00
0.01	0.06	0.10	0.06	0.01
0.02	0.10	0.16	0.10	0.02
0.01	0.06	0.10	0.06	0.01
0.00	0.01	0.02	0.01	0.00

Gaussian Filter

- Example of a 9 x 9 Gaussian Kernel with $\sigma = 3$

0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.00	0.00
0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.00
0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
0.01	0.01	0.01	0.02	0.02	0.02	0.01	0.01	0.01
0.01	0.01	0.01	0.02	0.02	0.02	0.01	0.01	0.01
0.01	0.01	0.01	0.02	0.02	0.00	0.00	0.01	0.01
0.01	0.01	0.01	0.01	0.02	0.00	0.01	0.01	0.01
0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.02

Gaussian Filter

Image with Gaussian Noise	Smoothed Image Using a 5 x 5 Gaussian Filter with $\sigma = 1$
	

Gaussian Filter

Smoothed Image Using a 9 x 9 Gaussian Filter with $\sigma = 3$

Smoothed Image Using a 9 x 9 Gaussian Filter with $\sigma = 2$

Noise Reduction Filters

Image with Gaussian Noise

Mean Filer k = 9

Median Filter k = 9

Gaussian Filter K = 9, σ = 3

Noise Reduction Filters

Image with “Salt and Pepper” Noise

Mean Filer k = 9

Median Filter k = 9

Gaussian Filter K = 9, σ = 3

Edge Detection Filters

David Esparza Alba

Ritsumeikan University

Edge Detection

- Edge detection consists in finding those regions where the change of intensity is stronger.
- Is necessary to compute the first derivates in both directions X and Y in an image.
- The definition of derivate is given by the following expression:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- For our case, *h* will represent the distance between pixels.

Prewitt Filter

- Prewitt filter makes use of the first derivates horizontally, vertically and diagonally.
- The derivate in one direction is given by:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

- Prewitt filter uses the information of the nearest neighbors and set h = 1.

$$f'(x) = \frac{f(x+1) - f(x-1)}{2}$$

Prewitt Filter

- Then, the kernel for X direction is:

-1/2	0	1/2

- For simplicity multiply by 2 the entire row. This will only affect the magnitude of the derivate.

-1	0	1

- Adding the derivates in both diagonals we have that the kernel for X direction is :

$G_x =$

-1	0	1
-1	0	1
-1	0	1

Prewitt Filter

- Applying the same concept for the Y direction, we have the kernel for the derivate in Y is:

$$G_y = \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

- The new value of pixel (x,y) is given by the magnitude of the gradient.

$$g(x,y) = \sqrt{G_x^2 + G_y^2}$$

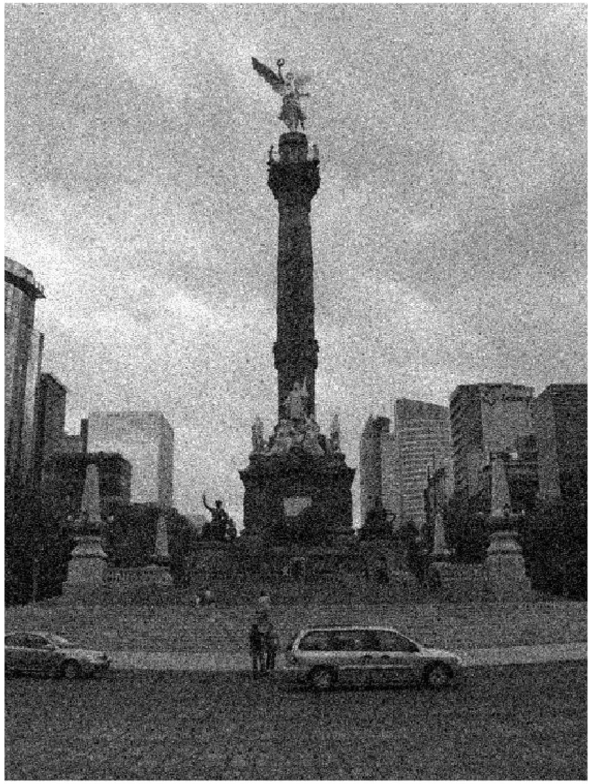
Prewitt Filter

Original Image

After Prewitt Filter

Prewitt Filter

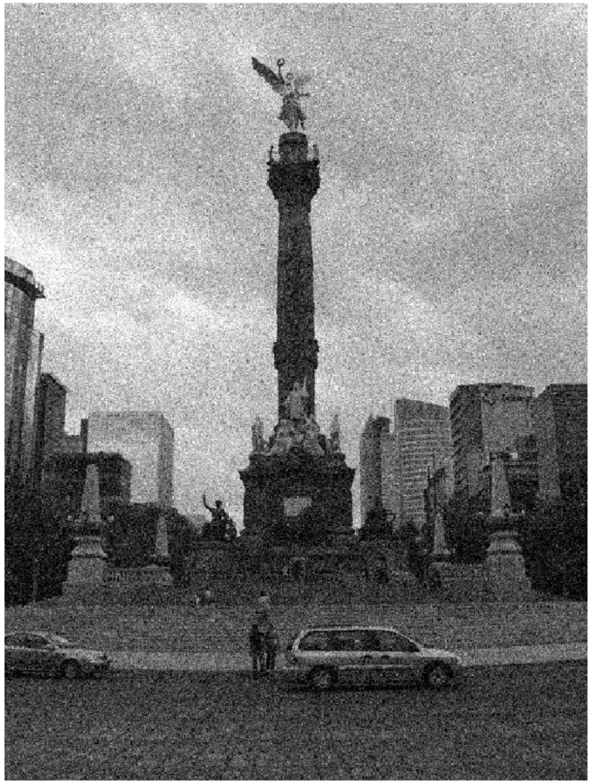
Image with Gaussian Noise

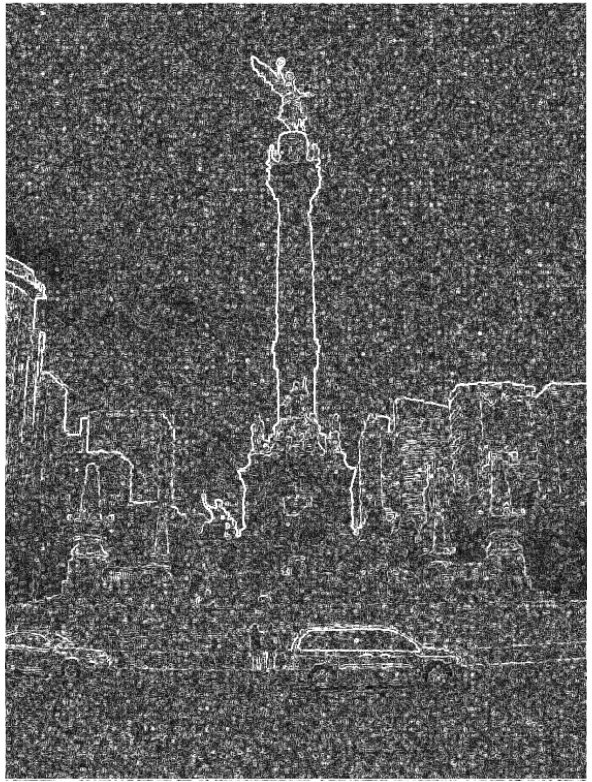


After Prewitt Filter

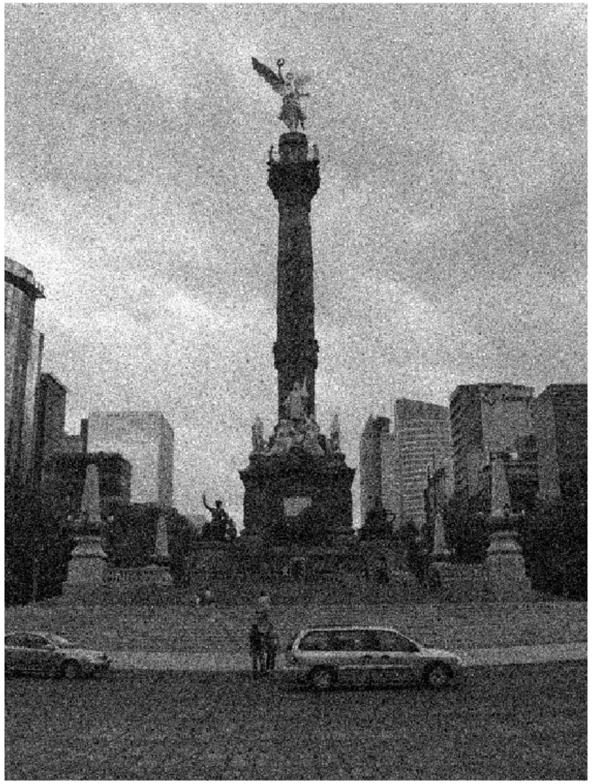
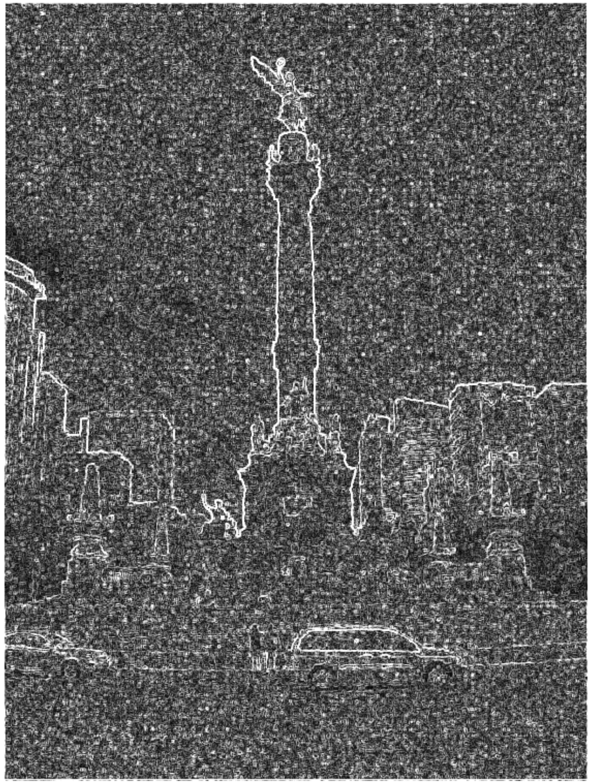
Prewitt Filter

Image with Gaussian Noise	After Prewitt Filter



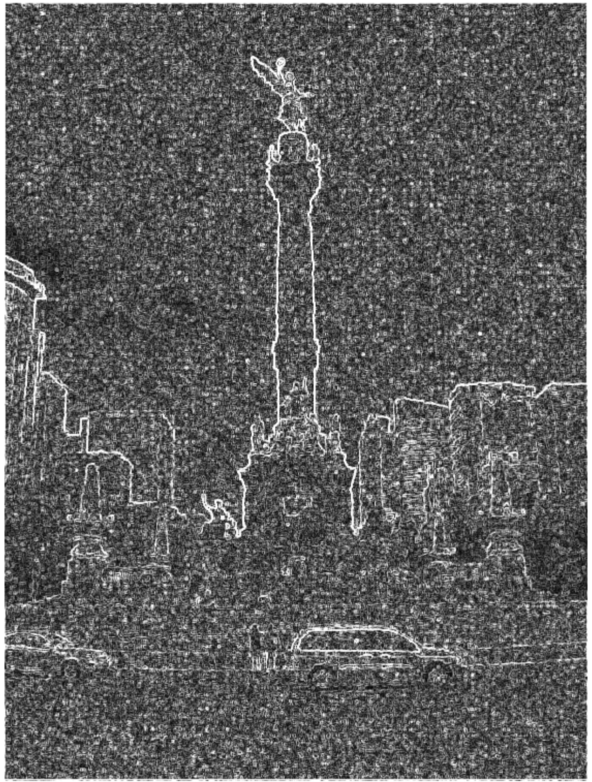


Prewitt Filter

Image with Gaussian Noise	After Prewitt Filter
	

First, we have to reduce the amount of noise with some of the filters we have already seen.

Prewitt Filter

After Gaussian Filter with K = 9, σ = 3	After Prewitt Filter
	

Then, apply Prewitt Filter to the smoothed image

Prewitt Filter

After Gaussian Filter with K = 9, σ = 3

After Prewitt Filter

Sobel Filter

- Sobel Filter is a weighted version of the Prewitt Filter. Making emphasis in the horizontal and vertical directions.
- The kernel of the derivates in X and Y directions are:

$G_x =$

-1	0	1
-2	0	2
-1	0	1

$G_y =$

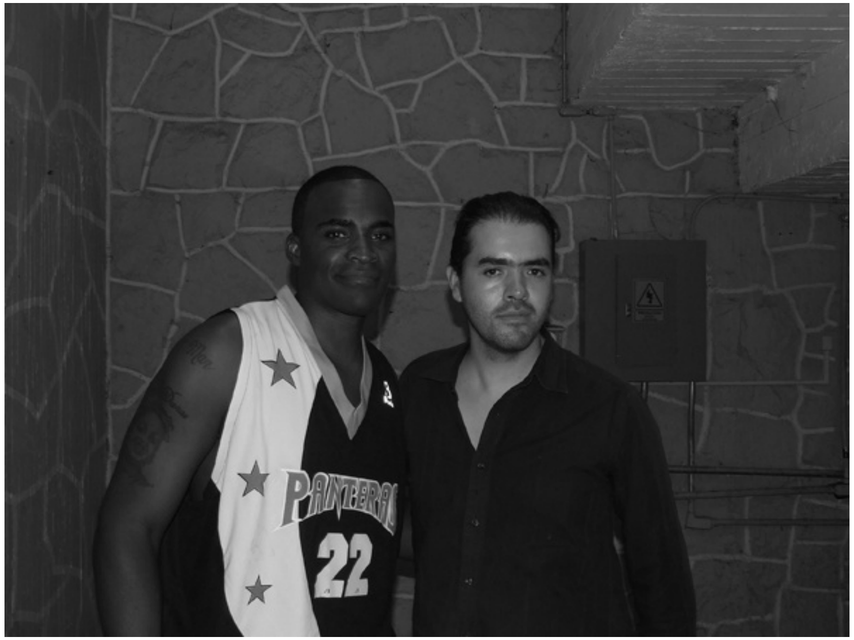
1	2	1
0	0	0
-1	-2	-1

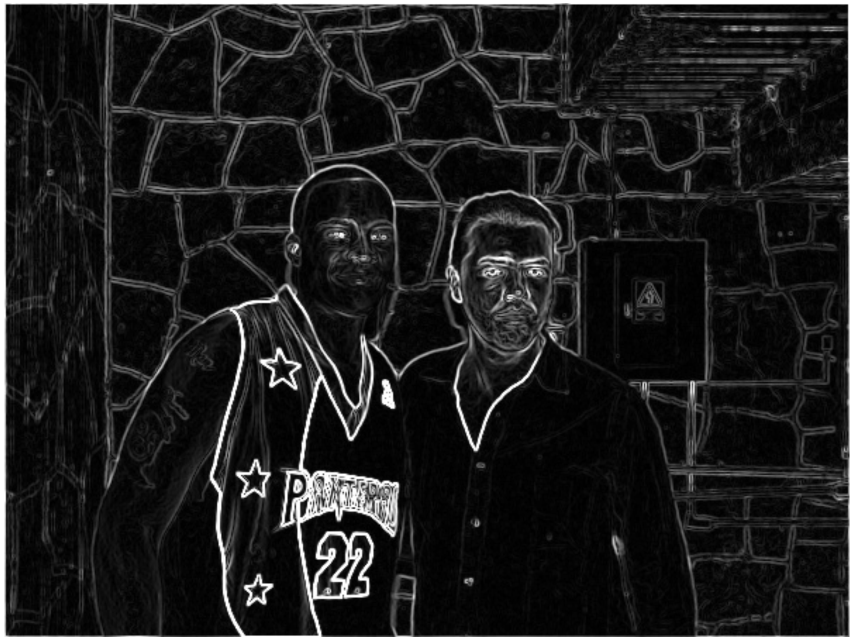
- The value of the pixel in position (x, y) is given by:

$$g(x, y) = \sqrt{G_x^2 + G_y^2}$$

Sobel Filter

Original Image	After Sobel Filter





Laplacian Filter

- The Laplacian uses second order derivates to obtain the new value of the pixel in each position.
- The first order derivate in one dimension is given by:

$$f'(x) = \frac{f(x+h) - f(x)}{h}$$

- Then the second order derivate would be:

$$f''(x) = \frac{f'(x+h) - f'(x)}{h}$$

Laplacian Filter

- The second order derivate can be expressed as:

$$f''(x) = \frac{\frac{f(x+h)-f(x)}{h} - \frac{f(x)-f(x-h)}{h}}{h}$$

- Simplifying this expression, we get:

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

Laplacian Filter

- If we set h = 1, then we have the following

$$f''(x) = f(x+1) - 2f(x) + f(x-1)$$

- The kernel in one direction would be:

1	-2	1

- The Laplacian is defined by the sum of the second partial derivates.

$$L(x,y) = \frac{\partial^2 g(x,y)}{\partial x^2} + \frac{\partial^2 g(x,y)}{\partial y^2}$$

Laplacian Filter

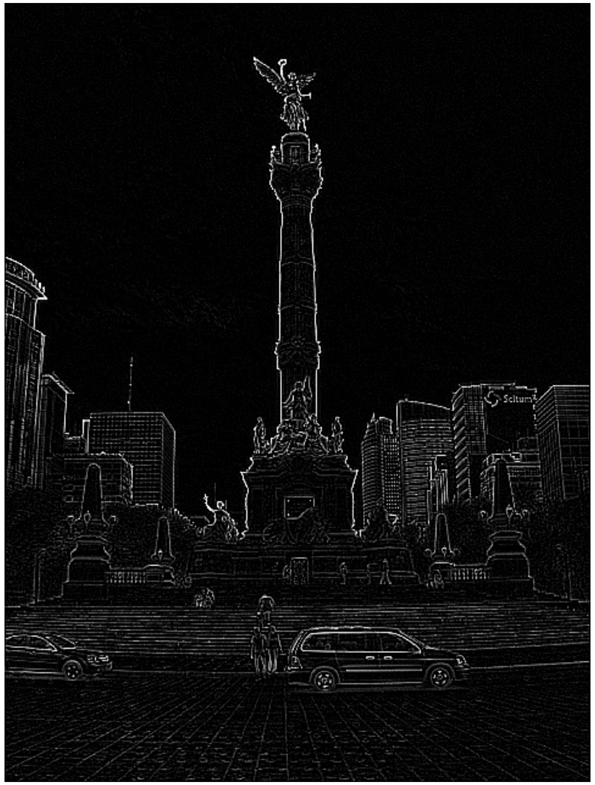
- Adding both kernels, it will result in a single kernel that looks as follows:

$$G_{xy} = \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

- If we add the second derivates of both diagonals into the kernel, we have:

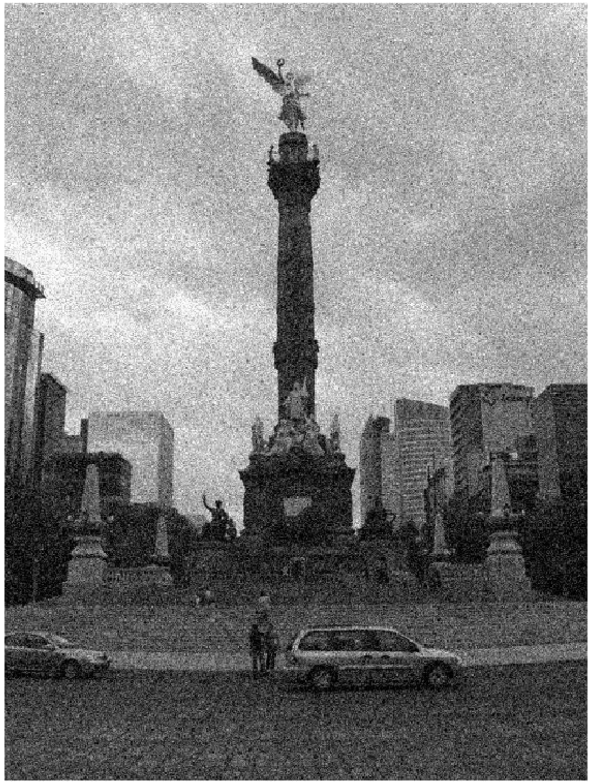
$$G_{xy} = \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & -8 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Laplacian Filter

Original Image	After Laplacian Filter with c = 1
	

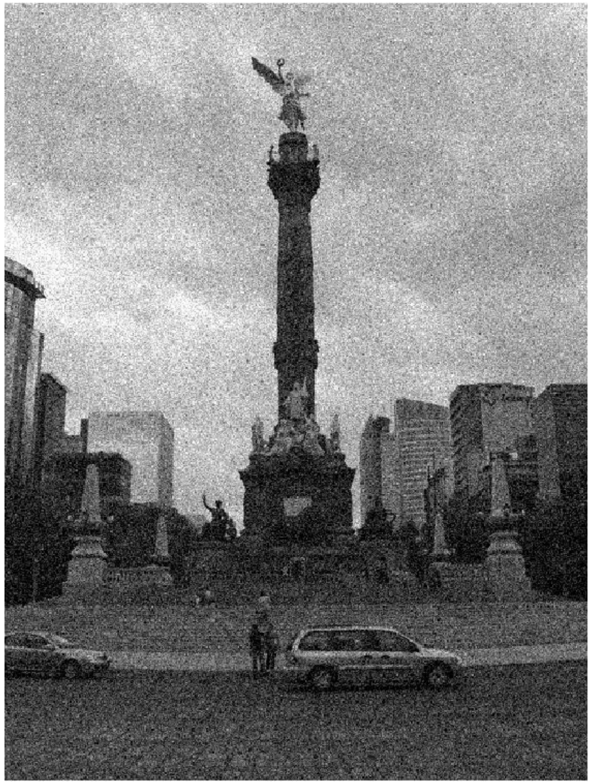
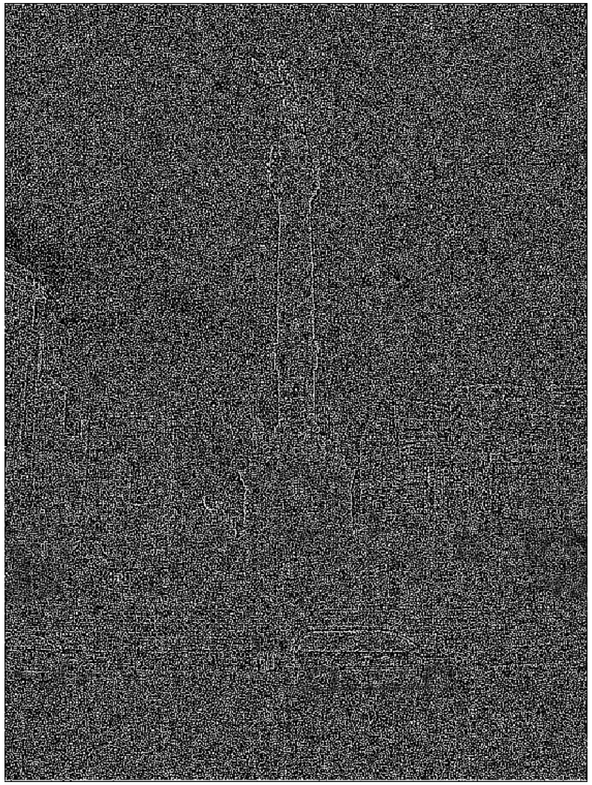
Laplacian Filter

Image with Gaussian Noise



After Laplacian Filter

Laplacian Filter

Image with Gaussian Noise	After Laplacian Filter
	

Laplacian Filter

Smoothed Image using Gaussian Filter with $k = 9$, $\sigma = 3$

After Laplacian Filter

Reducing the noise amount with a Gaussian filter and then applying the Laplacian filter we have the following result:

Laplacian of Gaussian (LoG)

- Applying a Gaussian filter to smooth the image, and then apply the Laplacian filter can be computationally expensive.
- We can convolve the Laplacian filter with the Gaussian filter and use this new kernel. This can save us some time.
- It's not enough? Then we can pre-calculate values of this new kernel and avoid the convolution of both filters.

Laplacian of Gaussian (LoG)

- The values of the Gaussian kernel are given by:

$$h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- The second order partial derivate in both directions X and Y are defined by:

$$\frac{\partial^2 h(x,y)}{\partial x^2} = \frac{1}{2\pi\sigma^6}(x^2 - \sigma^2) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\frac{\partial^2 h(x,y)}{\partial y^2} = \frac{1}{2\pi\sigma^6}(y^2 - \sigma^2) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Laplacian of Gaussian (LoG)

- The Laplacian is given by:

$$L(x,y) = \frac{\partial^2 h(x,y)}{\partial x^2} + \frac{\partial^2 h(x,y)}{\partial y^2}$$

- Then, the Laplacian of Gaussian is defined as:

$$cL(x,y) = \frac{c}{2\pi\sigma^6}(x^2 + y^2 - 2\sigma^2)e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Laplacian of Gaussian (LoG)

- LoG Filter kernel with $k = 9$, $\sigma = 1.7$ and $c = 1$

0.001	0.002	0.003	0.004	0.004	0.004	0.003	0.002	0.001
0.002	0.004	0.005	0.005	0.004	0.005	0.005	0.004	0.002
0.003	0.005	0.004	-0.002	-0.006	-0.002	0.004	0.005	0.003
0.004	0.005	-0.002	-0.018	-0.027	-0.018	-0.002	0.005	0.004
0.004	0.004	-0.006	-0.027	-0.038	-0.027	-0.006	0.004	0.004
0.004	0.005	-0.002	-0.018	-0.027	-0.018	-0.002	0.005	0.004
0.003	0.005	0.004	-0.002	-0.006	-0.002	0.004	0.005	0.003
0.002	0.004	0.005	0.005	0.004	0.005	0.005	0.004	0.002
0.001	0.002	0.003	0.004	0.004	0.004	0.003	0.002	0.001

Laplacian of Gaussian (LoG)

Image with Gaussian Noise

After LoG Filter with k = 9, σ = 2.0 and c = 30

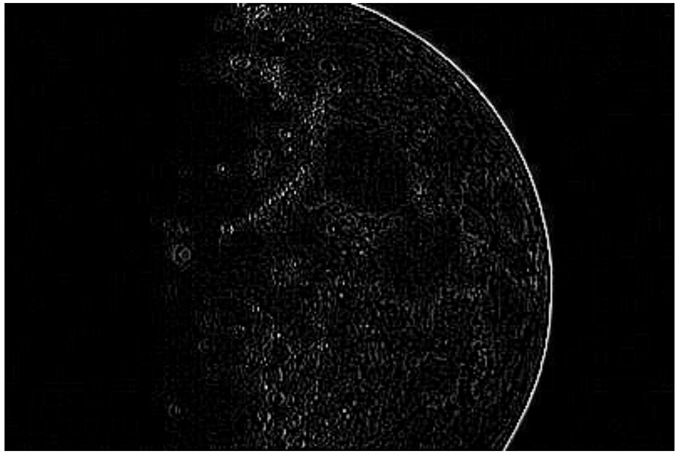
Sharpening

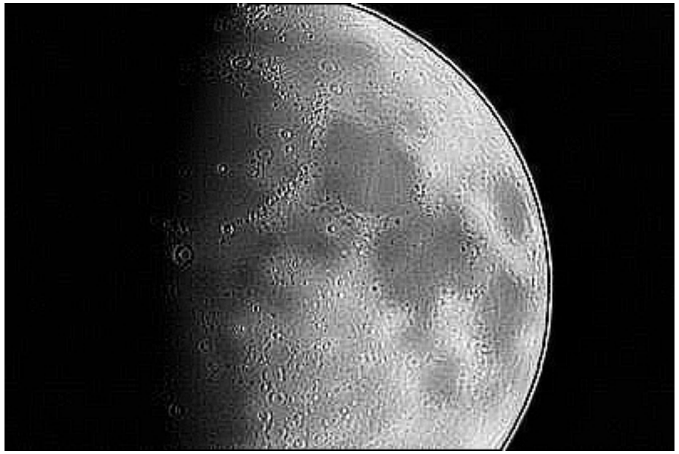
- The goal is to highlight fine details in the image.
- Laplacian Filter is useful to remark those important details.
- Let's apply the following process:

Sharpening

Original Image

After Laplacian Filter

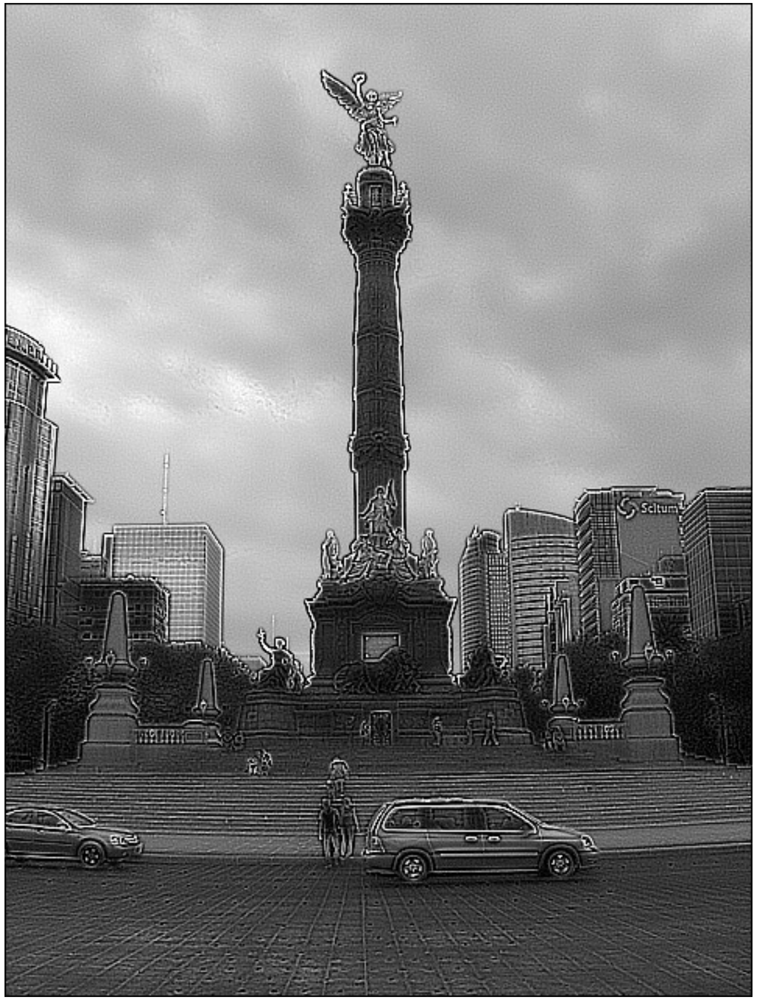




Sharpened Image

Sharpening

Original Image



Sharpened Image

Conclusions

David Esparza Alba

Ritsumeikan University

Conclusions

- After of applying any edge detection filter is important to smooth the image with another filter.
- Laplacian filter is very sensitive to intensity changes, then is very important to previously reduce the amount of noise in the image.
- Noise Reduction Filters works different depending in the kind of noise. We have to choose which filter is better for our application.
- Laplacian filter seems to work fine with the median filter.