UNIVERSIDAD POLITECNICA DE VALENCIA

ESCUELA POLITECNICA SUPERIOR DE GANDIA

Grado en Ing. Sist. de Telecom., Sonido e Imagen

UNIVERSIDAD POLITECNICA DE VALENCIA

ESCUELA POLITECNICA SUPERIOR DE GANDIA

"Investigation on Basic Methods for Digital Image Restoration"

FINAL BACHELOR THESIS

Author:
Olga Ballester Romaniv

Supervisor:
Ban-Sok Shin

BREMEN, 2015

I’m grateful to my better half Quico, who has held me out until the end; my best friends Guille, Jonatan, Vicente and Rafa, who have been there always and helped me constantly; and all the people that have believed in me.

Contents

Resumen

El siguiente documento, tras la presentación del trabajo y las motivaciones que llevan a escribirlo, va a tratar sobre el tratamiento digital de imagen y sus aplicaciones, focalizándose en la degradación y restauración de dichas imágenes. En primer lugar se explicará el filtrado espacial y frecuencial, para familiarizar al lector con dichos términos ya que serán de gran importancia en el desarrollo posterior. Una vez sentadas las bases se tratará el ruido, tanto con ruido Gaussiano como ruido impulsivo. A continuación las imágenes degradadas anteriormente serán restauradas mediante filtro paso bajo y filtro de mediana, respectivamente. Luego se habla del motion blur con ruido y sin ruido, usándose el filtro inverso y filtro de Wiener para su restauración. En el ultimo apartado se desarrollan sus conclusiones.

El següent document, després de la presentació del treball i les motivacions que porten a escriure-ho, tractarà sobre el tractament digital d'imatge i les seues aplicacions, focalitzant-se en la degradació i restauració de les dites imatges. En primer lloc s'explicarà el filtrat espacial i freqüencial, per a familiaritzar el lector amb els dits termes ja que seran de gran importància en el desenrotllament posterior. Una vegada assentades les bases es tractarà el soroll, tant el soroll Gaussià com el soroll impulsiu. A continuació les imatges degradades anteriorment seran restaurades per mitjà del filtre pas davall i filtre de mitjana, respectivament. Després es parla del motion blur amb soroll i sense soroll, usant-se el filtre invers i filtre de Wiener per a la seua restauració. En l'ultim apartat es desenrotllen les seues conclusions.

In the following text, after the presentation of the paper and the motivations that lead to write it, it is going to talk about digital image processing and its applications, focusing on the degradation and restoration of said images. In first place it explains the frequential and spatial filtering, to help the reader to get along with the terminology that will be very important it the following development. Once all the basics are learned, the noise will be treathed, both Gaussian and impulsive. Next, the previously degradated images will be restored using low pass filter and median filter, in that order. Then it talks about motion blur with and without noise, using inverse filter and Wiener filter for its restoration. In the last part the conclusion will be developed.

1. Motivation

The objective of this thesis is, as its title says, to investigate on basic methods for digital image restoration. There are different needs to enhance digital images, either its brightness, either its contrast. There exist two main models of digital image degradation that are going to be studied in this report: noise and motion.

Noise is a random fluctuation of image brightness and it can appear because of poor illumination while there is taking a picture, or because of a failure of the sensor of the camera that there is using, and also due to electronic circuit noise. The appearance that noise has is similar to grain and it dominates the image. In analog photos it was common to see this effect, but in digital images it is not, so there is a need to remove this type of degradation.

On the other hand, another principal reason of image distortion that has to be removed is motion blur effect. This type of degradation model appears due to rapid movement of the object that it wants to capture and because of a long time of exposure. The time of exposure is the time that the light lingers in the sensor of the camera that is using to take pictures. That means that as longer the light is being lingered in the sensor, longer is going to be opened the iris of the camera, so more movement is going to be captured. As more movement the camera captures, more distorted (motion blurred) is going to be the output image.

These two models of degradation can be restored by different ways, but in this thesis filters implemented with Matlab are going to be used.

1.1. Keywords

Digital image, digital processing, noise, degradation, restoration, mask, filtering, filter, pixels, coefficients, value, probability density function.

1.2. Notation

x, y	Spatial (plane) coordinates
u, v	Frequency domain coordinates
$f(x, y)$	Two-dimensional function in spatial domain
$\hat{f}(x, y)$	Estimated two-dimensional function in spatial domain
$F(u, v)$	Two-dimensional function in frequency domain
$\hat{F}(u, v)$	Estimated two-dimensional function in frequency domain
$\mathfrak{I}^{-1}$	Inverse Fourier Transform
z	Gray level of the image
σ^2	The variance of z
μ	The mean of the average value of z
$p(z)$	Probability density function, PDF
$\|H(u, v)\|$	The absolute value of $H(u, v)$
$H^*(u, v)$	The complex conjugate of $H(u, v)$
$g(x, y) * h(x, y)$	A convolution between $g(x, y)$ and $h(x, y)$ in spatial domain
$G(u, v)H(u, v)$	A scalar multiplication of $G(u, v)$ by $H(u, v)$ in frequency domain
$E\{f\}$	An expected value of f

1.3. Outline

1.3.1. Chapter 2

This section will explain some basics of digital image, where will be defined the main concepts of image processing and later will be seen linear filtering, a basic technique for knowing how degradation can be created onto a digital image in spatial domain. This chapter will also study masks and the importance of its length of pixels. Finally it will explain how is filtering in frequency domain works using Fourier transforms and which steps must be followed for it.

1.3.2. Chapter 3

In this part of report, noise corruption is going to be studied. Concretely there will be considered two basic noise models, called Gaussian noise and Impulse noise. There are going to be explained the reasons of appearance of these two types of noise in digital images and how they can be simulated in image processing. There are also going to be seen the differences between them and it will be studied the distribution that each model follows and their respective probability density function. Later on it will be investigated what kind of filter has to be applied for restore a digital image depending on the type of noise with which it has been degraded.

1.3.3. Chapter 4

Chapter 4 will be focused in motion blur. It will start studying the process of motion blur generation without noise in image processing and then with noise. Finally there are going to be seen the filters that can be applied for estimate the original image, depending always on with what kind of motion type it has been corrupted previously (either motion blur without noise either with it).

1.3.4. Chapter 5

This section will contain the conclusions of this project.

1.3.5. Chapter 6

Finally, after the conclusions seen in Chapter 5, the references used for develop this report are going to be provided.

2. Basics of Digital Image Processing

Before starting with techniques of image restoration, this part of thesis will show the basic concepts of image processing.

2.1. Digital Image Components

A digital image in digital image processing is a numerical representation of a two-dimensional function $f(x,y)$ where x and y are plane coordinates of the told image. In this project it is going to work with gray-scale images. Gray level represents different levels of intensity at each point of (x,y) and the value of these levels of intensity can vary from 0 (black color) to 255 (white color). There also exists an important element called pixel that has its own concrete location (x,y) and which value will represent the intensity level of its point location, so it will go from 0 to 255, as it has been already explained.

$f(x-1,y-1)$	$f(x,y-1)$	$f(x+1,y-1)$
$f(x-1,y)$	$f(x,y)$	$f(x+1,y)$
$f(x-1,y+1)$	$f(x,y+1)$	$f(x+1,y+1)$

Figure 1: 3x3 pixels digital image
Source: Own development

In Figure 1 it is shown an example of a digital image composed by 3x3 pixels, where every box represents a pixel and x and y represent the coordinates of each picture element. The value of f at each coordinate represents the gray level of told component situated there.

Thus, a gray scale digital image, also called $f(x,y)$, is composed of nxn pixels that will have its particular location at a point (x,y) and which value can go from 0 to 255 representing the intensity level of each point (x,y) of the image.

2.2. Spatial Domain Filtering

In this part it is going to be seen spatial filtering, a technique that will serve for distort and restore digital images in spatial domain. This process consists in applying the central coefficient of a mask of nxn coefficients onto every pixel of the image that is want to be filtered and the remaining components of the mask have to be applied onto the neighborhood of the pixel with that it is working. The objective of spatial filtering is to achieve a new value for every pixel of an image after applying the told mask so on this way the output result of the image is different to the original. This new value for every pixel it is obtained after adding the products between the value of every coefficient of the mask and the weight of every pixel that is situated under the area spanned by the mask.

In Figure 2 it is shown an example of this technique, where a 3x3 mask is being applied to a pixel $f(x,y)$ and thus it is also taking into account its neighbor pixels.

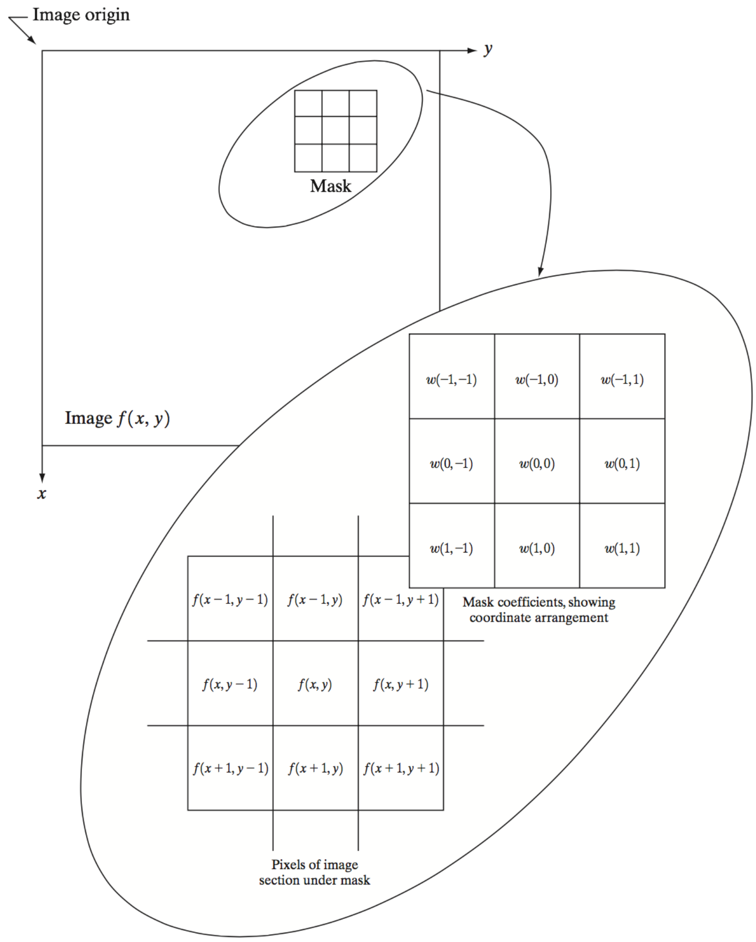


Figure 2: The mechanics of spatial filtering
Source: "*Digital Image Processing*", Rafael C. González, Richard E. Woods.

So in the given example, the result after applying the mask will be a new value of the pixel $f(x, y)$ and it will be calculated as it has been explained before and as it can discover in Eq. 1:

$$\begin{aligned} R = f(x-1, y-1)w(-1,-1) + f(x-1, y)w(-1,0) + f(x-1, y+1)w(-1,1) \\ + f(x, y-1)w(0,-1) + f(x,y)w(0,0) + \cdots \\ + f(x+1, y+1)w(1,1) \end{aligned} \tag{1}$$

As it can be observed, the central coefficient of the mask, $w(0,0)$, overlaps the pixel $f(x, y)$ of the image, which means that the mask is centered at that point when the filtering is occurring. The other fact that must be taken into account and it has been described before, is that the neighbor pixels of the one that it is working with are considered as well. This indication is important specially when the mask operates on the borders of the original image.

So, when it is working on an edge of a picture, there are going to be some missing neighbor pixels.

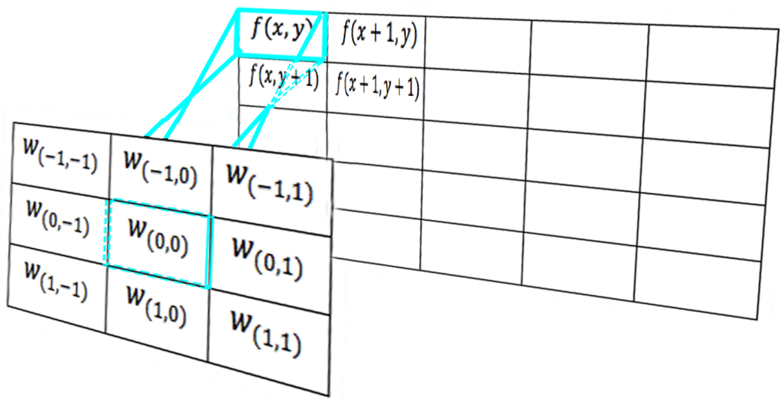


Figure 3: Filtering on a border of the image
Source: Own development

As it is shown at the Figure 3, when the central coefficient of the mask it is projected onto the pixel $f(x,y)$, only some coefficients of the mask ($w_{(0,0)}, w_{(0,1)}, w_{(1,0)}, w_{(1,1)}$) can operate on it and its corresponding neighbors. There is also can be detected that its shape is square, so the size will be always of n x n coefficients. The rest of the mask components have been left outside of the image plane. To avoid this problem the outline of the image must be padded with $(n-1)/2$ rows and columns full of zeros, as it can be seen in Figure 4.

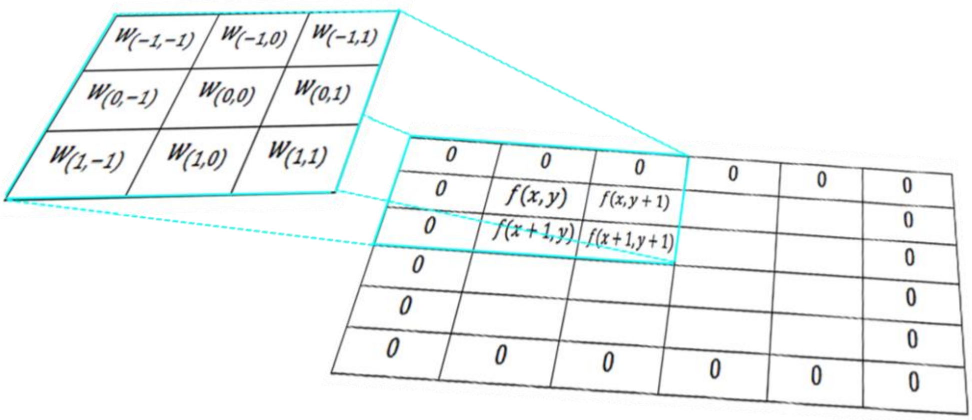


Figure 4: The borders of the image padded with the corresponding number of rows and columns of zeros to a mask of 3x3
Source: Own development

Generally, spatial filtering is expressed as the Eq. 2 says for images of M x N components (pixels) and masks of mxn coefficients:

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t) \qquad (2)$$

where $a = (m-1)/2$ and $b = (n-1)/2$. To ensure that the original image has been completely filtered, the previous equation has to be applied for values of $x = 0, 1, 2, \dots, M-1$ and for values of $y = 0, 1, 2, \dots, N-1$.

As it has been told before, in this thesis it is going to work only with gray-scale images and the general function that will define the spatial filtering at any point (x,y) of this type of images with an m x n mask, will be expressed as:

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_{mn} z_{mn} = \sum_{i=1}^{mn} w_i z_i \quad (3)$$

where w's are mask coefficients, z's are the values of the gray level of the image corresponding to those coefficients, and mn is the total number of coefficients of the mask. But it has seen before that the total number of coefficients of a mask will be the result of the product of n x n, because the shape of masks (also called filters) is going to be always squared.

2.3. Frequency Domain Filtering

2.3.1. Discrete Fourier Transform

In this section it is going to be studied filtering process in frequency domain. For this, first of all it is advisable to talk about Fourier transform and its inverse. In this project it is going to consider concretely the two-dimensional discrete Fourier transform. When it says *two-dimensional* DFT, it means that the Fourier transform will have two components, $F(u,v)$, as a result of transforming a function $f(x,y)$ of an image, that also has two components.

The discrete Fourier transform $F(u,v)$ of a function $f(x,y)$ is defined by the following equation:

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})} \quad (4)$$

where $j = \sqrt{-1}$, and M and N are respectively the height and width of the image. Conversely, given $F(u,v)$, it can obtain $f(x,y)$ by means of the inverse Fourier transform:

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v)\, e^{j2\pi(\frac{ux}{M}+\frac{vy}{N})} \quad (5)$$

For understand better the concept of Fourier transform, a little comparison between frequency domain filtering and spatial filtering is going to be seen. While in spatial filtering exists a direct interaction onto pixels of a picture for filtering, in frequency domain the Fourier transform must be used for interact with the frequency components of an image for filtering.

When the Fourier transform it is applied to a function, it is common to multiply the input function $f(x,y)$ by $(-1)^{x+y}$ so on this way the output of the Fourier transform function will be $F(u-\frac{M}{2}, v-\frac{N}{2})$. Thereby the origin coordinates of the DFT $F(0,0)$ is shifted on the center of the image, that is $(M/2, N/2)$. To make sure that the DFT is going to be in the center of the perimeter of the image, it is advisable to arrange that the M and the N values were even numbers. So, the value of $(u,v) = (0,0)$ is defined as:

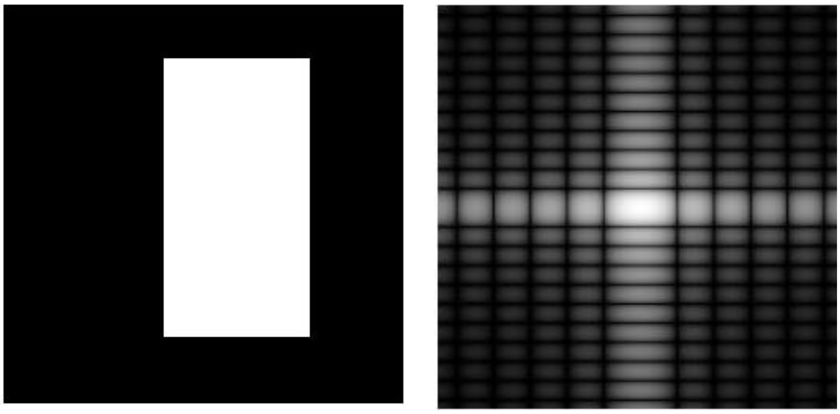
$$F(0,0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \quad (6)$$

which means that the value of the DFT at the origin is equivalent to the average of the image.

In Figure 5 can be seen an example of Fourier spectrum of a bitmap image that has been created with the help of Matlab. As it can observe in the picture (b), the separation of spectrum zeros of rectangles that follow the axis v is twice the separation of spectrum zeros of rectangles that follow the axis u. That is inversely corresponded to $\frac{20}{10}$ relation that there is between them. There has been used the Equation 7 for increase the grey-level detail.

$$s = c \log(1 + r) \qquad (7)$$

From this equation it should be known that s is the output image, whose grey-level has been enhanced, c is a constant value (set on 1) and r is the grey-level of the image, always ≥ 0.



$\downarrow x$ $\rightarrow y$

(a) Original image with a white rectangle of 20x10 size

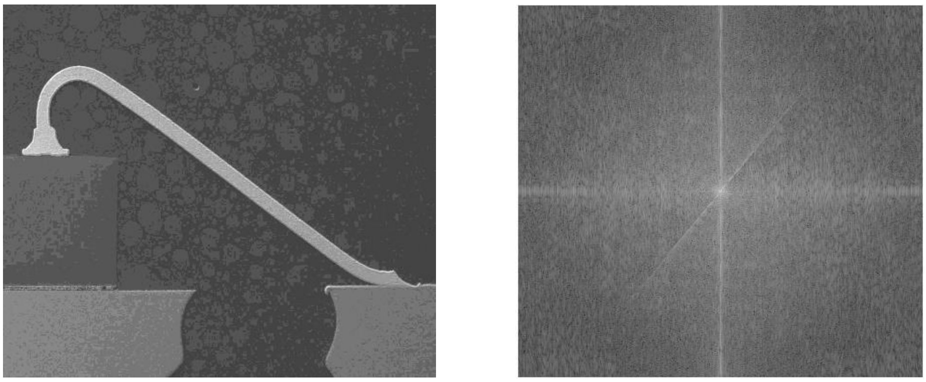
$\downarrow u$ $\rightarrow v$

(b) The Fourier spectrum of the image (a) after applying the log transformation

Figure 5
Source: Own development based on *Digital Image Processing*, Rafael C. González & Richard E. Woods

As seen in Figure 5, the Fourier spectrum represents the abrupt changes of an image. That is, when the frequencies of the image change noticeably (for example at the edges or when color changes), the spectrum of the Fourier transform will be represented with high frequencies and when the image does not have so evident intensity alteration (for example the background or the floor) it will be represented with low frequencies. In Figure 6 it is given another example that will help to explain and to follow the characteristics of Fourier spectrum.

In Figure 6 it can appreciate a scanning electron microscope image of a wire. There can be seen that there is an evident change of intensity between the wire and the background and it also noticed that the image has a little bit of noise. Thus, looking at the spectrum of the picture it can recognize where are the values of grey-level intensity changing. That is, the white wire is bent $\pm 45°$ and so are its edges. The Fourier transform spectrum exposes these edges along the $\pm 45°$ and, thus, the frequency alteration.



(a) SEM bond wire (b) Fourier spectrum of (a)

Figure 6

Source: Own development

2.3.2. Steps for filtering in Frequency Domain

Now that it has been talked about what is and for what serves the Fourier transform, there are going to be explained steps that must be followed for filtering in frequency domain:

1. Multiply the input image by $(-1)^{x+y}$ for center the Fourier transform it has been explained before.
2. Compute the DFT, $F(u, v)$, of the image from (1).
3. Multiply $F(u, v)$ by a filter function $H(u, v)$.
4. Compute the inverse DFT from (3).
5. Obtain the real part of the result from (4).
6. Multiply the result in (5) by $(-1)^{x+y}$.

Let's define a function $f(x, y)$ as the input image and the function $F(u, v)$ as its Fourier transform. As it mentioned in step (3), $F(u, v)$ must be multiplied by the filter equation, also called degradation function $H(u, v)$ so on this way the output result will be $G(u, v)$:

$$G(u, v) = H(u, v)F(u, v) \tag{8}$$

In this equation each component of H will multiply each component of F. Generally, the Fourier transform components are complex quantities, but the filters that in this project are going to work with have real components. The output filtered image $\hat{f}(x, y)$, is obtained by calculating the inverse Fourier transform of $G(u, v)$, as it is expressed in the Equation 9.

$$\hat{f}(x, y) = \Im^{-1}[G(u, v)] \tag{9}$$

For achieving the final filtered image the real part of $\hat{f}(x, y)$ has to be extracted and finally it must be multiplied by $(-1)^{x+y}$ to reverse the effect that it has originated at the beginning of filtering operation. As it has been commented before, the Fourier transform has complex components, but the input image and the filter do not, so when the inverse transform is implemented, its imaginary components should all be zero.

3. Noise Models

Noise models chapter will explain the two basic noise models that are Gaussian noise and Impulse noise. It is going to be seen how digital images can be corrupted by these types of noise and why, and also how they can be restored or how they can be approached. The differences between the two types, the advantages and disadvantages of its restoration process are also going to be explained.

3.1. Gaussian Noise

This type of noise it is used to simulate poor illumination while it is acquiring a picture or to represent gains differences of the sensor of the camera, high temperatures, random fluctuations in image signal, etc. The appearance of this effect is similar to grain over the entire digital image with different values of intensity on every pixel.

3.1.1. Noisy Image Generation

The Gaussian noise follows the next probability density function, which will be called PDF from now on:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2} \tag{10}$$

From this function, it’s important to know that z represents the gray level of the image, μ is the mean of the average value of z, and σ is its standard deviation. The standard normal distribution sets μ to 0 and σ to 1. The distribution that this type of noise follows can be studied in greater depth with the help of Figure 7.

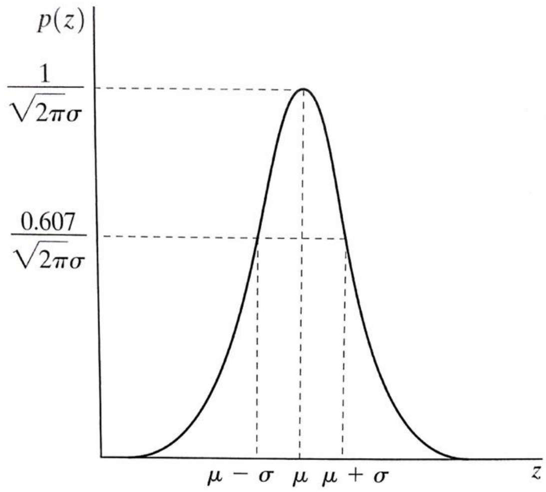


Figure 7: Gaussian PDF
Source: “*Digital Image Processing*”, Rafael C. González, Richard E. Woods.

From this figure it can be observed that if z follows the equation 10, almost 70% of values of z are in the interval of $[\mu-\sigma, \mu+\sigma]$, and almost 95% are situated in $[\mu-2\sigma, \mu+2\sigma]$. The distribution of Gaussian noise is continuous and it affects to every pixel of the digital image that it is want to be corrupted. The value of the corrupted pixel will depend on its previous value, that is, on its previous intensity level. As it has been mentioned before, Gaussian noise is random fluctuation in image signal or image brightness, so depending on how intense is a pixel, its corresponding “degraded value” will be intensified or not.

In practice, to generate a noisy image with Matlab, it is summed a noise term to the original image, like it’s expressed in the following equation:

$$y = x + z\sigma + \mu \quad (11)$$

where y is the output and degraded image, x is the original and undegraded image, z are pseudorandom values of the uniform distribution of the original image, σ^2 is the variance (which square root is the standard deviation) and finally μ is the mean of the average value of z. At the Table 1 some examples of output images varying the values of σ and μ can be seen. The original image that it is going to work with is showed in the Figure 8.

Figure 8: Original image
Source: "*Digital Image Processing*", Rafael C. González, Richard E. Woods.

What it can be observed in Table 1 is a practical example executed with Matlab of a digital image degraded by Gaussian noise. In the created program corresponded to this thesis, there are choices for changing both the mean value and the variance value.

As it can be seen in the said table, if the variance increases, the amount of noise is higher. The value of the mean, μ, is usually set by default on 0, but if it changes the only thing that it distorts is the lightness of the image. That is, if it increases, the brightness of the image is higher (almost white) and if it decreases, the brightness decreases (almost black).

Thus, after all explained previously, it can be assumed that Gaussian noise is an additive noise and when a digital image is corrupted by it, it must be known that each pixel of this image changes its value according to the Gaussian distribution.

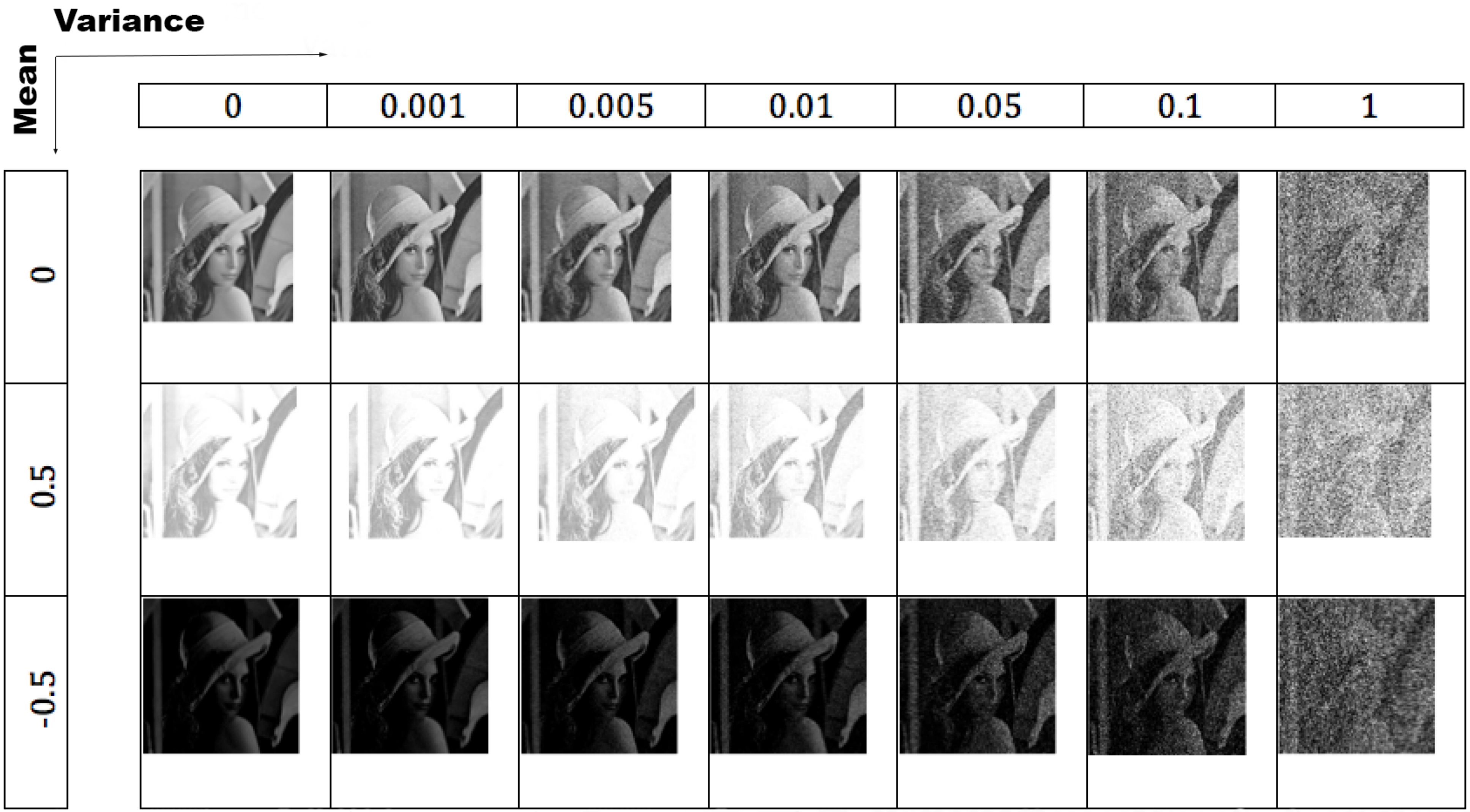


Table 1: Different results varying the variance and the mean
Source: Own development

3.1.2. Noisy Image Restoration

To restore images corrupted by Gaussian noise in digital image processing it's advisable to use low pass filter, but as it is going to be explained further on, this method doesn't restore completely the degraded image, it just reduces the damage caused by the noise generation process. It uses this type of filter because as noise has high frequencies in its spectrum, what low pass filter does is letting pass low frequencies and reducing the high ones. Thus, the noise effect in the image can be reduced. But there are some disadvantages that are going to be seen later, like for example if a big size of mask is used, the estimation of the original image is going to be blurred, so some information is going to be missed. Also, it is important to know that the edges of a digital image have high frequencies too like noise, and if low pass filter is used, these frequencies are going to be reduced like the noise effect.

To apply a low pass filter, a mask of 3x3 pixels can pass by each pixel of the original image (the degraded one) and do the average of the sum of the products resulted between each pixel of the mask and the pixels of the image which are situated under the area spanned by the filter. A bigger mask it can also be applied (always with an odd number of coefficients) but the little one it is easier to implement.

In the following example, the application of a low pass filter to achieve the restoration of a noisy image it is explained.

Mask $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Pixels of the image

158	140	150	123	161
159	141	168	128	154
160	160	155	134	158
158	159	142	140	162
157	141	139	141	150

Figure 9: Mask 3x3 and pixels of the image
Source: Own development

In Figure 9, a mask of $n = 3$ x $n = 3$ and an image of $M = 5$ x $N = 5$ pixels are shown. Now the mask it is going to be passed upon some random pixels of the image. There are going to be chosen only two pixels of the image (Figure 10) for make the example understandable, but when filtering, the mask has to pass onto all of them.

158	140	150	123	161
159	141	168	128	154
160	160	155	134	158
158	159	142	140	162
157	141	139	141	150

Figure 10: Pixels of the image that we're going to work with, which positions are $f(2,3)$ and $f(5,5)$.
Source: Own development

At this point, as it shown in Figure 11, the mask has to be situated on that way so its central coefficient coincides with the pixel that it is want to filter.

When the mask's central coefficient overlaps the point $f(x, y)$, the neighbor pixels of that point what coincide with the area spanned by the mask are overlapped too by the remaining coefficients of the filter, as it expressed in the Figure 12.

It can observe, that in Figure 12, the picture (b) has additional zeros, for the reason that has been explained at the Chapter 2. As one of the chosen pixels of the image it is situated at the border, to apply a mask of nxn coefficients and to filter the picture, the perimeter of the image has to be padded with $(n-1)/2$ rows and columns full of zeros.

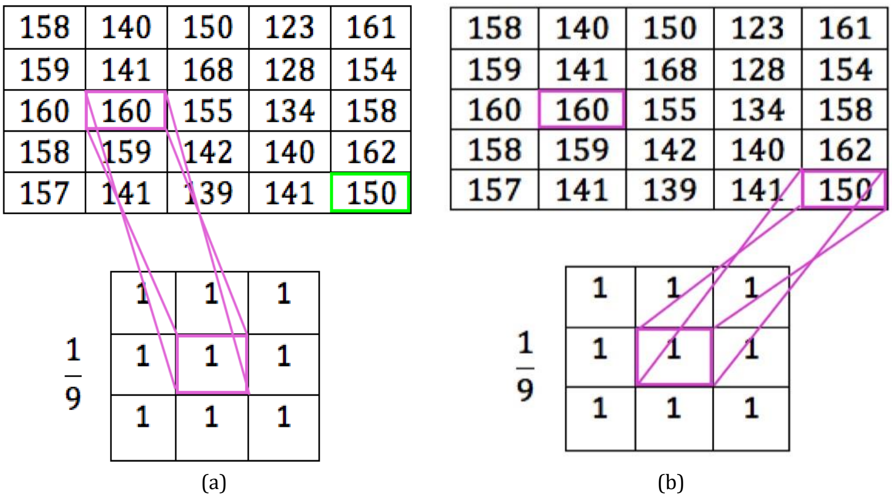


Figure 11: Focusing the central coefficient of the mask with the pixel of the image
Source: Own development

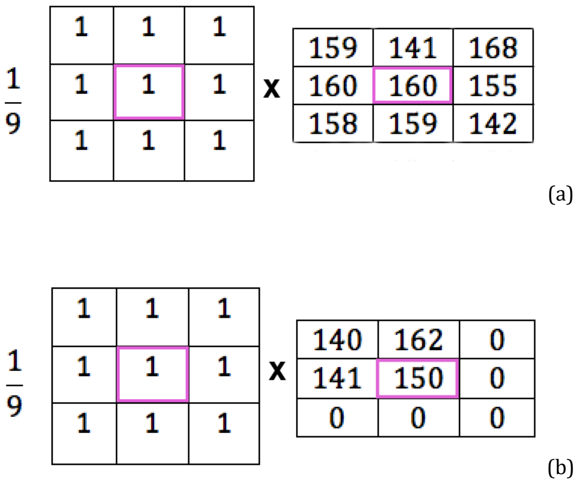


Figure 12: Products of the mask coefficients and the image pixel and its neighbors
Source: Own development

The result obtained after filtering these pixels with a low pass filter of 3x3 pixels is:

$$(a) = \frac{1}{9} \cdot (1 \cdot 159 + 1 \cdot 141 + 1 \cdot 168 + 1 \cdot 160 + \mathbf{1} \cdot \mathbf{160} + 1 \cdot 155 + 1 \cdot 158 + 1 \cdot 159 + 1 \cdot 142) = 155.77 \cong 156 \quad (12)$$

$$(b) = \frac{1}{9} \cdot (1 \cdot 140 + 1 \cdot 162 + 1 \cdot 0 + 1 \cdot 141 + \mathbf{1} \cdot \mathbf{150} + 1 \cdot 0 + 1 \cdot 0 + 1 \cdot 0 + 1 \cdot 0) = 65.88 \cong 66 \quad (13)$$

The equations 12 and 13 represent the new values of these two pixels after being filtered with a low pass mask.

In the Table 2 practical examples of images degraded with the Gaussian noise restoration can be seen. These corrupted images are the same that have been seen in the Table 1 with $\mu = 0$ and varying the σ value.

As it can be observed, the low pass filter only helps to restore a degraded image when the variance of Gaussian noise is low. That is because as the filter acts averaging the image pixels' values, as higher is the variance of noise, lower is going to be the probability of achieving the correct restoration, because the resulting average value after filtering each pixel is still going to be high. And as it's evident, if the density of noise is 100% (1 in this case, because it is working with decimal values) there is no way to restore the image.

Also it can be perceived, as it has been mentioned previously, that the result of the image after applying a low pass filter turns blurred. Until certain point it is possible to reduce the noise effect even if the image looses some information, but it is appreciated that the edges of the image are also reduced, as it has said before. Thus, it is demonstrated that the low pass filter is useful for restore images with a low value of noise variance and if the size of the mask that it is using is also not too big, because otherwise the noise effect would estimate the image and its boundaries could have not be distinguished.

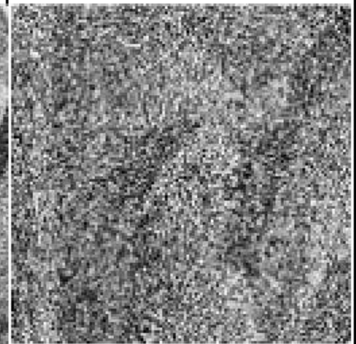
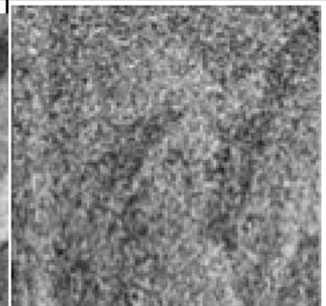
Variance	0.001	0.005	0.01	0.05	0.1	1
Degraded						
Restored						

Table 2: Gaussian noise restoration
Source: Own development

3.2. Impulse Noise

Impulse noise happens because of a failure of the sensor of the camera that there is using to take a picture. If one of the cells of the sensor is burned, impulse noise effect happens, turning the value of some pixel either into 0 either into 255. This type is also called *salt and pepper* noise, because, as it will be explained next, the values of the corrupted pixels turn either black either white, creating an effect of salt and pepper dispersed over the image. It is said that the pixels damaged by this type of noise are saturated, because it values change completely, unlike happens with Gaussian noise.

3.2.1. Noisy Image Generation

This type of noise only impacts on a part of the pixels of the image, unlike Gaussian noise that affects to every pixel of the image. The values that the damaged pixels take are very extreme, that is, either the value is very low either is very high. It produces an effect of brilliant or dark speckles situated randomly over the image.

Its PDF is given by:

$$p(z) = \begin{cases} P_a & for\ z = a \\ P_b & for\ z = b \\ 0 & otherwise \end{cases} \tag{14}$$

And the plot of this probability density function is illustrated in the Figure 13.

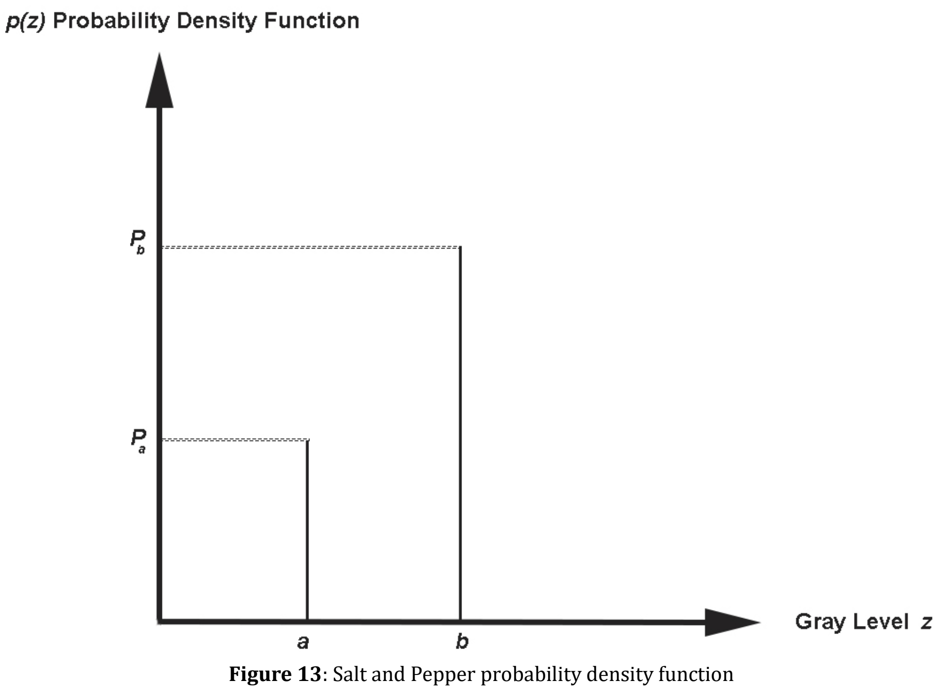


Figure 13: Salt and Pepper probability density function
Source: Own development based on "*Digital Image Processing*" by Rafael C. González, Richard E. Woods

From this figure it can be explained the distribution that this model of noise follows. For a given certain probability P_a there will be some pixels with value of z that will turn into value of a. And for another given probability P_b there will be other random pixels with values of z that will turn into b value. The values of a and b are 0 and 255 respectively, that is, black and white colors respectively. So the effect that this type creates is black and white dots situated randomly over the image. It is also appreciated that the probability of appearance of b value is higher than the probability of appearance of a value.

If both probabilities are zero, then the impulse noise is called *unipolar*. But when the value of both probabilities is different to zero, and its values are very close/similar, then the impulse noise is going to be called *salt-and-pepper*. The reason is because as both probabilities, P_a y P_b, are almost equal, the number of light and dark dots is going to be almost proportionate. Thus, the program that has been created for this thesis takes this aspect into account and makes that both probabilities were as equal as possible.

In practice, to generate images degraded with this noise using Matlab, first of all pseudo random values of the gray-level pixels of the image have to be found, which will be called z. With this, the program is taking care of locating random pixels that are going to be corrupted. Once these pixels are found, the quantity of light and dark dots is going to depend on the value of density of noise $p(z)$, that it is want to generate. When this value $p(z)$ it is known, Matlab has to find values of z smaller than $\frac{p(z)}{2}$, and these values will change to the value of a. And also, has to find pseudo random values of z higher or equal to $\frac{p(z)}{2}$ and at the same time lower than the whole density $p(z)$, that we desire and then change them to b.

Thus, what the program is trying to simulate is the appearance of both values with a similar probability to achieve a better effect of salt and pepper noise. And as it has been explained before the values of a and b are going to be equal to 0 and 255 representing the black and white color respectively.

3.2.2. Noisy Image Restoration

For restoring images degraded with this model of noise, a mask that will act as median filter must be passed over the image that is want to correct. It is used median filter and not low pass filter, because as it will be explained later, this filter does not reduces any frequencies and lets pass another ones. This filter just changes the corrupted pixels to its most approximate value, obtaining it from its neighborhood. As the neighborhood of the corrupted pixel is situated in the same area, there is a big probability of the neighbor pixels having the same or very similar intensity level. Thus, low pass filter does not have to be used because it would soften and increased the bright and dark dots rather than remove them.

The application of median filter consists in:

- Pass the mask over each pixel of the image.
- Order the values of the pixels of the image that are under the area spanned by the mask, from lowest value to the highest one.
- Change the value of the pixel that it is working onto to the value of the central pixel of those pixels that have been ordered before.

In the following example this process can be seen and it may help to understand the theory better.

First, a mask and pixels of an image have to be defined, like is figured in Figure 14.

Mask

1	1	1
1	1	1
1	1	1

Pixels of the image

145	167	159	98	180
178	132	98	99	200
67	45	23	43	90
134	167	210	167	98
156	189	56	101	144

Figure 14: Mask 3x3 and pixels of the image
Source: Own development

As it can observe, this time the mask is not divided by 9 even when its size and its coefficients are the same as when the low-pass filter was implemented in previous chapter. That is because when median filter is applied, there is no need to do the average of the values of the pixels that are covered by the mask; these pixels just need to be ordered from lower to high value.

In Figure 15, there are going to be chosen two random pixels, for making the example understandable, but as it has been said before, the mask has to pass through each pixel of the image that it want to restore.

145	167	159	98	180
178	132	98	99	200
67	45	23	43	90
134	167	210	167	98
156	189	56	101	144

Figure 15: pixels of the image that we are going to work with
Source: Own development

In Figure 16, a mask of 3x3 coefficients it is being applied to these two pixels that have been chosen before.

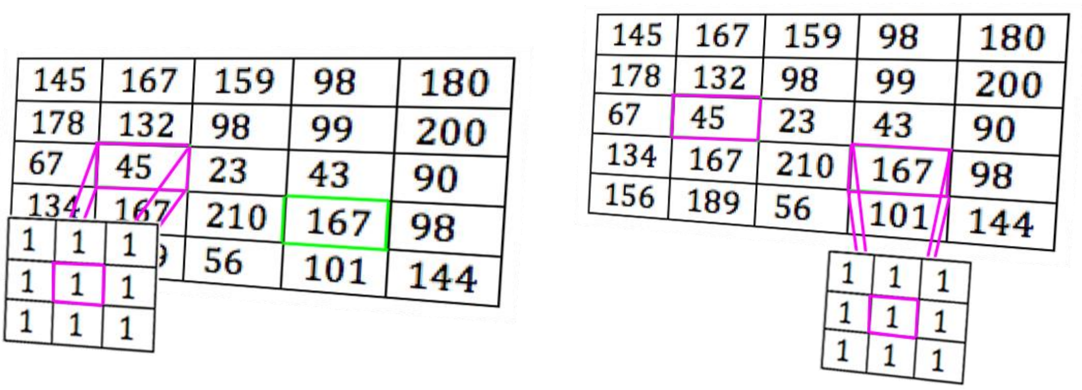


Figure 16: Focusing the central coefficient of the mask onto the pixels that we're going to work with
Source: Own development

In Figure 17, as the mask is passing by each pixel and its corresponding neighbors, it starts to order these ones according to its value, placing the lowest values at the left side and the highest at the right side. Thus, in the middle place of the array that it has created, there is a median value, that will be the new value of the pixel after being filtered.

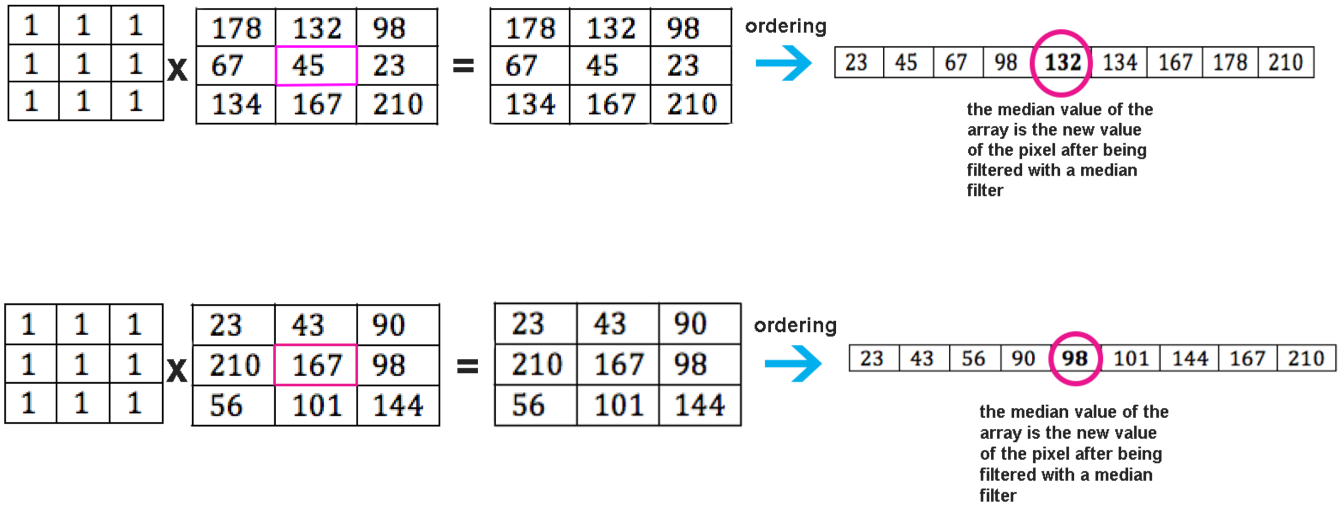


Figure 17: Choosing the median value
Source: Own development

As a result, the new values of both pixels that have been filtered are the highlighted ones in Figure 18.

145	167	159	98	180
178	132	98	99	200
67	**132**	23	43	90
134	167	219	**98**	98
156	189	56	101	144

Figure 18: The new values of pixels after being filtered with a median filter
Source: Own development

Now that it has been seen how the median filter works, in Table 3 there are some practical examples, implemented with Matlab, of generation and restoration of images degraded by impulse noise with different densities of noise.

As it can observe, as higher is the value of the noise density, more saturated of salt and pepper noise is going to be the image. Also, it can be detected that the number of dark dots and the number of bright dots are almost the same. That is because, as it has been mentioned before, for having salt and pepper effect the probability of pixels that have been turned into bright dots has to be almost the same (or very close to) as the probability of pixels that have been turned into dark dots.

Focusing on the restored images it can see that the median filter is very useful to reduce this kind of noise. It is obvious that the images where noise is dominant are not going to be completely restored, but even that, the output result after filtering is noticeable. The reason of why pictures with more noise density are less restored is because of the median filter effect. As it is known, this kind of filter acts reordering the values of each pixel of the neighborhood of the pixel that it is working on, taking the central value as the correct. So, if the density of noise is high, there are going to be more corrupted pixels. If there are more corrupted pixels, the probability of finding them in the neighborhood of the point that it is working with, is higher too. Thus, it could be happen that the median value coincides with a noisy pixel and on this way there would have not restore anything. Applying the same filter on the same corrupted image several times is also should be considered for making the restoration more evident. On this way the probability of degraded pixels would decrease.

It can be appreciated that the restored image does not turn blurred. As it has seen before, low pass filter blurs the image averaging the values of its pixels, but what the median filter does is just reordering them. On this way there is no missing information and the result is much better. There it has not been applied low pass filter for restore images with this type of noise, because as the values of corrupted pixels are too extreme, the result would have been a blurred image with blurred white and black dots. As these type of pixels are saturated unlike happens in Gaussian noise, that are brightness variation, if a blur filter it is applied there is not going to be a better result.

Finally, even the value of density has to be expressed in percentage, the value that it has been provided to Matlab is fractioned and that is why is given in decimal, as it did when it worked with Gaussian noise.

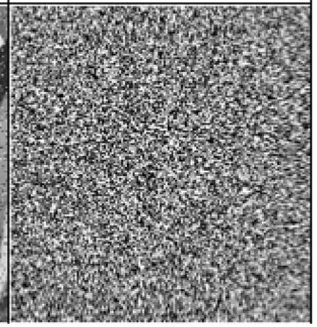
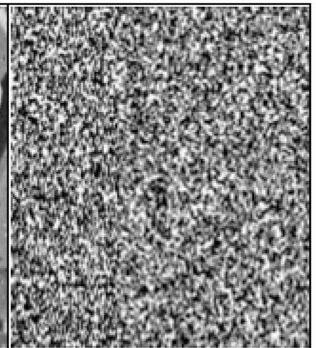
Density of noise $p(z)$	0.001	0.005	0.01	0.05	0.1	1
Degraded image						
Restored image						

Table 3: Result of degrading and restoring salt and pepper
Source: Own development

4. Motion Blur

As it has been mentioned before motion blur generates due to long time of exposure and to motion between the object and the sensor of the camera during the image acquisition. In this part of report it is going to study motion blur with and without noise and it will also investigate on basic methods of restoration of images with this type of distortion.

4.1. Common Model of Image Degradation

In spatial domain image degradation consists in convolving an input image $f(x,y)$ with a degradation function $h(x,y)$ and adding a noise term η for finally obtaining a degraded image $g(x,y)$ as a result. The first part of degradation that would be a convolution, $f(x,y) * h(x,y)$, in frequency domain would become into a product, $F(u,v)H(u,v)$, where F and H are the Fourier transforms of the original image function and of the degradation function respectively (as it has been seen previously, for working in frequency domain it must use Fourier transforms). The second part of the degrading process would be the sum of a noise term (it is still being a sum in both domains, but in frequency domain it would be its Fourier transform):

Spatial Domain $$g(x,y) = h(x,y) * f(x,y) + \eta(x,y) \quad (15)$$

Frequency Domain $$G(u,v) = H(u,v)F(u,v) + N(u,v) \quad (16)$$

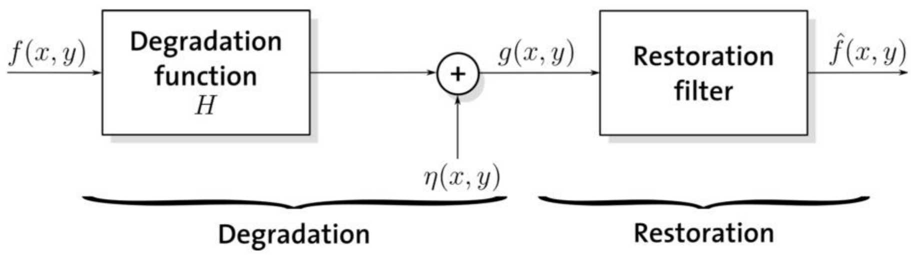


Figure 19: Common model of image degradation
Source: https://miac.unibas.ch/SIP/06-Restoration-media/figs/image_restoration_model.png

From Figure 19 is also can be observed a second part of a common model of image degradation and it is its restoration part and it consists in applying a restoration filter to the already degraded image function, obtaining an estimation of the original image called $\hat{f}$.

As it can perceive, working in frequency domain it is much easier than in spatial domain because there is avoiding the convolution process, and it is used a simple multiplication. But for working in this domain, as it has been explained in Chapter 2 Fourier transform of every function must be calculated. Thus, for generating motion blurs distortion it is going to work with frequency domain filtering and therefore with Fourier transforms of every function that it is going to use for this purpose.

4.2. Motion Blur Without Noise

4.2.1. Motion Blurred Image Generation

In this part it is going to study motion blur generation without noise and its mathematic expression follows the Equation 17:

$$G(u,v) = H(u,v)F(u,v) \tag{17}$$

where $G(u,v)$ is the Fourier transform of the output degraded image, $H(u,v)$ is the Fourier transform of the filter function (also called degradation function) that is multiplied by the Fourier transform of the original image $F(u,v)$. As it can observe this expression is different to Eq. 16 and that is because of noise term absence. Later on there will be studied how this process works with additive noise following the expression of Eq. 16.

The degradation function H has been implemented with Matlab function called $fspecial$. This function creates a 2-dimensional filter where must be specified the type of filter that is want to implement ('motion'). And also there must be provided the length of pixels that will decide how bigger is the motion blur effect and the angle that will determine the direction of the distortion.

Once the Fourier transform of the degradation function is obtained and is multiplied by the Fourier transform of the original image, there will be obtained the Fourier transform of the degraded image, as it is expressed in Equation 17. To see the result, the inverse Fourier transform of the degraded image obtained before must be calculated.

As it will be seen further on in some example, as bigger is the length of pixels that there is applying to the image, bigger is going to be the blur motion distortion. It will be seen also how the motion direction changes depending on the angle that there is provided. That is, if the value of the angle is 0°, the direction of the motion will be horizontal moving on to the left side; if the value is 90° then the direction of the motion will be vertical moving on upwards; if the angle value is 135° the direction will be horizontal motion moving on upwards and to the left side on the same time and so on.

In the following table (Table 4) there can be seen two of these distinct types of motion (horizontal and vertical) and how each of them changes according to the length of pixels and the angle that there has been applied, as it has been explained previously. The image that it is going to distort in said example is the one represented at Figure 20.

Figure 20: Car license plate
Source: Internet

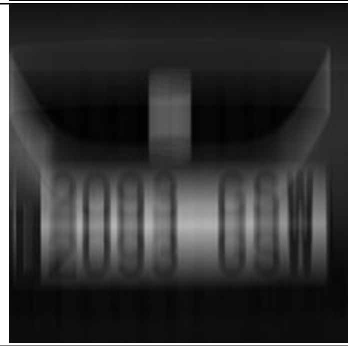
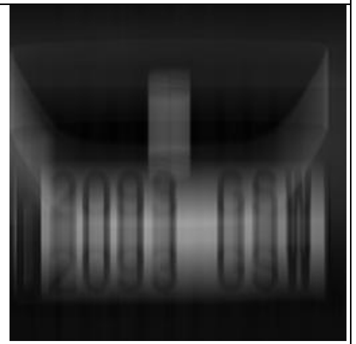
	9 pix	17 pix	31 pix	49 pix	61 pix
0˚ ≡ Horizontal motion					
90˚ ≡ Vertical motion					

Table 4: Horizontal, Vertical and Diagonal motion distortion with different sizes of mask
Source: Own development

4.2.2. Motion Blurred Image Restoration – Inverse Filter

The inverse filter method works as its own name says: inverting the filter that has been used for distort the image. So its mathematical process consists in:

1. Multiply the Fourier transform of the degraded image by the inverse of the Fourier transform of the filter. (Eq. 18).
2. Calculate the inverse Fourier transform of the output result form (1). (Eq. 19).

$$\hat{F}(u,v) = \frac{1}{H(u,v)} G(u,v) \tag{18}$$

$$\hat{f}(x,y) = \Im^{-1}(\hat{F}(u,v)) \tag{19}$$

This method of restoration is used to be applied on images that have been distorted with motion blur but without any kind of additional noise and it does not work very well in all cases. Sometimes, when $H(u,v)$ has values close to zero or zero, it results impossible to compute. So in this case, the solution would consist in limit the filter frequencies to values near the origin. From Eq.6 it is known that $H(0,0)$ is the average value of $h(x,y)$. That means that is the highest value of $H(u,v)$ in frequency domain. So if filter frequencies are approached to the origin, zeros and low values would be avoided and it would be possible to compute the inverse filter.

As it has been said before this filter is usually implemented in degraded images without noise, because otherwise, from Eq. 16 our restored image would have been mathematically expressed:

$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)} \tag{20}$$

And if there happens the case mentioned before (that $H(u,v)$ has zero values or values close to zero), then $\frac{N(u,v)}{H(u,v)}$ could dominate the estimation of $\hat{F}(u,v)$. And the Fourier transform of $N(u,v)$ would be also unknown because is a random function.

In Table 5 the results after implementing the inverse filter on the images from Table 4 can be seen. It is noticed that as the length of pixels increases, the probability of achieving a complete restoration is lower and there is more remaining distortion in the restored image. There is also can be discerned that this remaining distortion indicates the direction that was following its corresponding distorted image.

	9 pix	17 pix	31 pix	49 pix	61 pix
Horizontal motion					
Vertical motion					

Table 5: Inverse filter applied to corresponding images from Table 4, where no noise was applied
Source: Own development

4.3. Motion Blur With Noise

Motion-blurred images with noise can appear due to mix of a bad illumination and a rapid movement of the object that there is want to capture. There is also a possibility that the lens of the camera is dirty and there is motion happening between the object and the sensor. It can also occur that there is electronic noise in the circuitry of the camera and it influences during the image acquisition. However, noisy motion blurred images exist and there is a need to restore this type of distortion. In this part of thesis it is going to explain how this kind of corruption can be simulated and how images distorted with this type of degeneration can be restored or at least estimated.

4.3.1. Motion Blurred Image Generation

To add noise to an already distorted image with motion blur model, the value of variance of noise that is want to apply must be known and then it must be added to the image, as it expressed in the Eq. 16. In this project it is going to add Gaussian white noise, so its mathematical expression follows the Eq. 11. As it has been mentioned before, normally the noise term has random function so its Fourier transform seldom is known. To simulate this, in the Matlab program that has been developed for experiment in this project, the noise expression it is added in spatial domain after obtaining the inverse Fourier transform of the degraded image. The steps followed for its generation, were:

1. Multiply the Fourier transform of the degraded function H by the Fourier transform of the original image F to create the motion-blurred image, called G.
2. Calculate the inverse Fourier transform of G obtaining g.
3. Add a noise term μ to g, creating a noisy motion-blurred image.

In this section only the value of noise fluctuation or also called density is going to be changed and the value of mean will be set on 0. As it well known from previous chapters, as bigger is the variance of noise, bigger is going to be the distortion.

In Table 6 it can observe how changes the same image according to the variance that there is adding. There are not going to be added big values of variance because further on it is going to be harder to restore it. So in this example the values of this parameter are going to be 0.001 and 0.01. As it has been learned in previous sections of this report, these values are more than enough to create some noise and they allow its subsequent restoration without many problems.

There are also not going to be chosen all different length of pixels that there were seen in Tables 4 and 5. There will only be determined two of them (17 pix and 49 pix) that differ drastically to see how it affects to the distortion effect. And finally it is only going to work with the horizontal direction.

Thus, in Table 6 is seen that when a noise term is added, the already blurred image turns in a motion-blurred image with white dots, similar to dust, what creates the effect of noise. The amount of these white dots depends on the value of density, as it is known.

When the length of pixels increases along with the noise term, the distorted image turns too difficult to interpret specially, as it can perceive, when σ rises. Concretely when the number of pixels used to distort the image is 49 pix (the largest value) and when the noise term is also the largest of both that have been chosen.

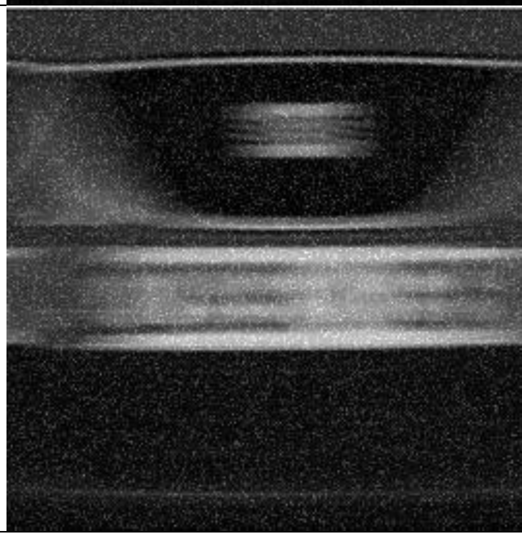
	17 pix	49 pix
$\sigma = 0.001$		
$\sigma = 0.01$		

Table 6: Blurred Motion with Gaussian White Noise
Source: Own development

4.3.2. Motion Blurred Image Restoration – Wiener Filter

In this section it is going to talk about one effective method of image restoration, called Wiener filter. This filter does not only serve for restoring images with motion degradation and without any additive noise; it also restores those images corrupted with noise. This method is also called *Minimum Mean Square Error Filtering*. It consists in considering images and noise as random processes and in finding an estimated recovered image $\hat{f}$ of the original image f, minimizing the mean square error between them:

$$e^2 = E\{(f-\hat{f})^2\} \tag{21}$$

As it can see in Eq. 21, the expected value of the argument is $E\{\cdot\}$. It supposes that the image and the noise are uncorrelated processes; that the mean value of one of them is zero; and also that the gray level of the estimated image is a linear function of the degraded one. This expression is given by the following function in the domain frequency:

$$\begin{aligned}\hat{F}(u,v) &= \left[\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)|H(u,v)|^2 + S_\eta(u,v)}\right]G(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \dfrac{S_\eta(u,v)}{S_f(u,v)}}\right]G(u,v) \\ &= \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + \dfrac{S_\eta(u,v)}{S_f(u,v)}}\right]G(u,v)\end{aligned} \tag{22}$$

From this function there are already known some components like $G(u,v)$ that is the Fourier transform of the degraded image; $H(u,v)$ is the transform of the degradation function; and $\hat{F}(u,v)$ is the Fourier transform of the estimated restored image. The new components are:

- $H^*(u,v)$ is the complex conjugate of $H(u,v)$
- $|H(u,v)|^2 = H^*(u,v)H(u,v)$
- $S_\eta(u,v) = |N(u,v)|^2$ is the power spectrum of the noise
- $S_f(u,v) = |F(u,v)|^2$ is the power spectrum of the undegraded image

Looking carefully it is observed that Wiener filter expression does not have the same problem with zero values of the degradation function $H(u,v)$ as it happens in the Inverse filter case, unless both of $H(u,v)$ and $S_\eta\ (u,v)$ are zeros for the same values of u and v. Also, if the value of noise component is zero, then the whole expression 22 corresponds to the Eq. 18, that is, the Inverse filter in frequency domain:

$$\begin{aligned}\hat{F}(u,v) &= \left[\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)|H(u,v)|^2 + S_\eta(u,v)}\right]G(u,v) \\ &= \left[\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)|H(u,v)|^2 + 0}\right]G(u,v) \\ &= \left[\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)H^*(u,v)H(u,v) + 0}\right]G(u,v) = \left[\frac{1}{H(u,v)}\right]G(u,v)\end{aligned} \tag{23}$$

If the noise that it is working with is white noise, then it considers its spectrum $|N(u,v)|^2$ as a constant for simplifying the operation. However, knowing the spectrum of the original image it is difficult, so when both of these two parameters are difficult to estimate, there is an approach of equation 22 where a constant K appears:

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right] G(u,v) \tag{24}$$

From this expression it is possible to intuit that $K = \frac{S_\eta}{S_f}$. When this parameter is equal to zero, the function becomes into the expression of the inverse filter, as it has been demonstrated in equation 23. When $K > 0$ it prevents the noise amplification when $|H(u,v)| = 0$. Thus, this factor provides a compromise between the inverse filter application and the noise amplification.

With the implementation of this function, it is available to vary the value of the constant K and see how the result of the output image changes. It can observe examples of this implementation in Table 7.

In the following table there can be seen different estimations of original image that change according to the value of K that there is providing. This term is intuitive and it can be changed until a good perception of the image is achieved.

As it can be detected and it has been commented before, if the variance of noise is high, then the restoration turns more difficult so the value of K has to be higher.

It is perceivable too that when the cost of K is too low or is very proximate to zero, the restoration is dominated by noise. This result is due to what has been commented previously: when this parameter is zero, the Wiener filter expression becomes in to the inverse filter expression, so on when this parameter is too close to zero, the result will be too similar to a result that would be obtained if inverse filter was applied to this type of image. However, this consequence (inverse filter applied to restore a digital image degraded with noise and motion blur) will be seen in Figure 21.

Also it can be seen that if the motion blur effect is big (high value of the length of pixels) and there is adding to it a high value of variance of noise, the output estimated image suffers a bad restoration. Even if the value of K is high there is no possible a complete or even normal estimation. To avoid this, the magnitude of noise or the value of pixels should be small.

There is also noticeable that in the estimated restoration of the images a remaining of noise exists, so the distortion does not disappear at all. This distortion staying is more evident in images that have been distorted with noise than in images that only have been distorted with motion-blur as it has seen with the inverse filter implementation.

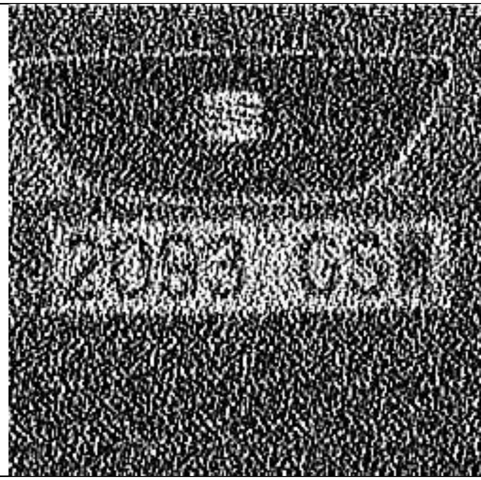
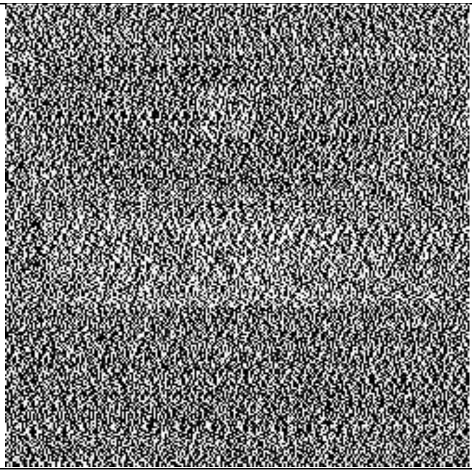
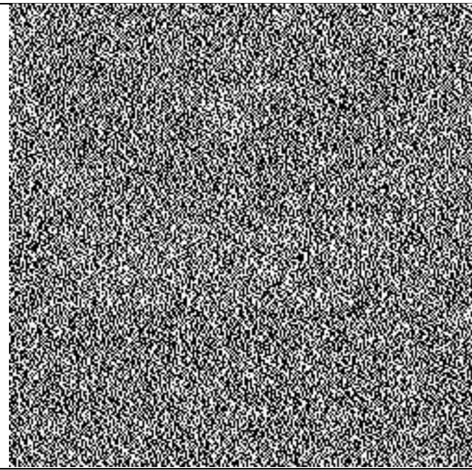
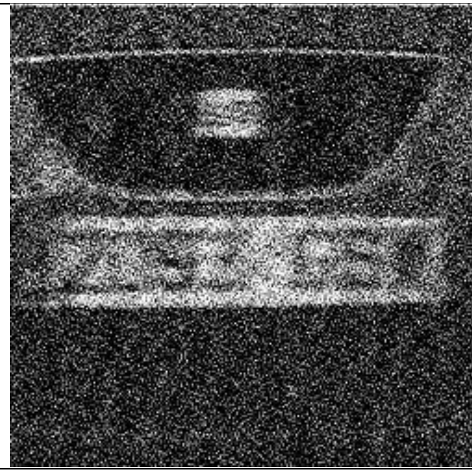
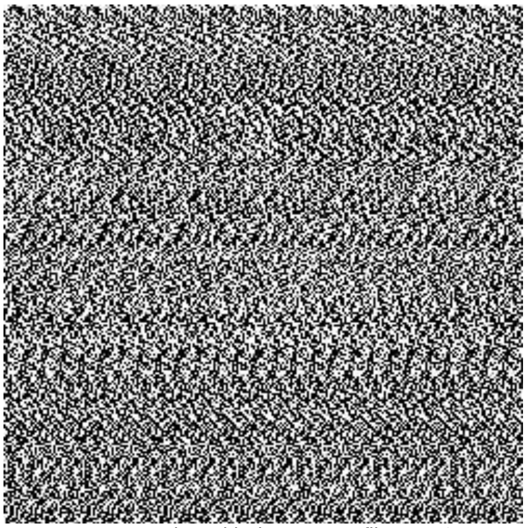
	17 pix		49 pix	
$\sigma = 0.001$				
	K=0.001	*K=1.83*	*K=0.001*	*K=1*
$\sigma = 0.01$				
	K=0.01	*K=55.04*	*K=0.01*	*K=17.78*

Table 7: Wiener filter implementation changing the value of K
Source: Own development

Observing Figure 21 it is perceived that if there is trying to restore a degraded image with noise using inverse filter, the result would be similar to images that appear in Table 7 with a very low value of parameter K, as it has been mentioned before. It has been seen before that if its value is zero, the expression of Wiener filter becomes into inverse filter. So if the value of K is very close to zero, the result after filtering the degraded image by motion and noise will be very similar to a result that would be obtained if it had filtered the image with the inverse filter.

Horizontal motion with 0.001 variance of additive noise



Restoration with the Inverse filter, K=0

Figure 21: Horizontal Motion blurred picture with 37 pixels of length and 0.001 variance of additive Gaussian White noise
Source: Own development

5. Conclusions

Now that all basic techniques of digital image degradation have been studied and explained, in this part of report it is going to redact a summary and see the conclusions.

This project was focused in studying basic methods of degradation of digital image and it subsequent restoration. The degradation methods were investigated to simulate real cases of digital image corruption and knowing how its nature works would help to estimate in a better way, how they could be restored.

First there have been explained some basic concepts of digital image processing including spatial and frequency domain filtering. These techniques were very important throughout this project. It has been used spatial domain filtering for restoring images that have been degraded with noise, concretely with Gaussian and Impulse noise. Frequency domain filtering was used for creating and restoring motion-blurred images.

It is concluded that images corrupted with Gaussian noise can be restored using low pass filters, but it must take into account that a big size of mask is not recommendable for not blurring the image. In this report it has used a mask of 3x3 pixels for restoring images with this type of noise and it resolves that if the variance of noise is too large, the restoration process does not achieve a good estimation of the original image. Thus, for obtaining an acceptable estimation of original image, the degraded one has to be degraded with a low variance of noise and be restored with a little mask (recommended with a 3x3 pixels mask).

On the other hand, there exist another common noise model called impulse noise. It concludes that for restoring this type of degradation the best way is using median filter. This filter does not blur the image unlike low-pass filter, and it can be applied to the degraded image several times if there still being corrupted pixels that have not been removed yet.

Finally, this report talks about motion blur distortion in digital images. It says that there exist two types of this distortion: with and without noise. Thus, there are studying its corresponding restoration methods.

First it has been seen motion-blurred images without noise. This type of images are created and restored in frequency domain, using Fourier transforms. It is concluded that the best and easiest way for restoring this type of distortion is using inverse filtering. It has to be taken into account that as bigger is the length of pixels used to distort the original image, bigger is going to be the remaining distortion in its estimation.

Finally, noisy motion-blurred images were studied. It concludes that the basic method of its restoration is Wiener filter. The effect of this filter depends on a parameter K that is intuitive and also on how distorted is the image. It is not the same having an insignificant variance of noise and slight motion effect, which would be easy to restore, than having a big variance of noise with also a big motion effect, which would be too hard to estimate. It is confirmed that as larger is the value of K, better is going to be the estimation of the original image as long as the values of variance of noise and the length of pixels are situated in an acceptable range, so the distorted image were moderately similar to the original one.

Ahora que todas las técnicas básicas de la degradación digital de imagen han sido estudiadas y explicadas, en esta parte del informe se va a redactar un sumario y ver las conclusiones.

Este proyecto ha estado focalizado en el estudio de los métodos básicos de degradación digital de imagen y su subsecuente restauración. Los métodos de degradaciones han sido investigados para simular casos reales de corrupción de imágenes y conocer cómo su naturaleza funciona puede ayudar a estimar, en una mejor manera, cómo pueden ser restauradas.

Primero se han explicado conceptos básicos de tratamiento digital de imagen incluyendo el filtrado espacial y en dominio frecuencial. Estas técnicas han sido muy importantes durante este proyecto. El filtro espacial ha sido usado para la restauración de imágenes que han sido degradadas con ruido, concretamente ruido Gaussiano y ruido impulsivo. El filtrado de dominio frecuencial ha sido usado para crear y restaurar imágenes degradadas con el filtro motion-blur.

Se ha concluido que las imágenes corrompidas con el ruido Gaussiano puede ser restauradas usando filtros de paso bajo, pero hay que tener cuenta que un gran tamaño de la máscara no es recomendable, pues puede dejar borrosa la imagen. En este documento se ha usado una mascara de 3x3 pixeles para restaurar imágenes con este tipo de ruido y se ha resuelto que si la varianza es demasiado alta, el proceso de restauración no logra una correcta estimación de la imagen original. Por tanto, para obtener una estimación aceptable de la imagen original, dicha imagen debe ser degradada con una varianza de ruido baja y, ser restaurado con una mascara pequeña (una mascara de 3x3 pixeles es recomendable).

Por otro lado, existe otro modelo de ruido común llamado ruido impulsivo. Lo que se concluye de esto es que para restaurar este tipo de degradación la mejor manera de hacerlo es el filtro mediana. Este filtro no deja borrosa la imagen como lo hace el filtro de paso bajo, y puede ser aplicado a la imagen degradada muchas veces si siguen existiendo pixeles corruptos que no han sido eliminados aún.

Para finalizar, este informe habla sobre la distorsión de imágenes digitales mediante el motion blur. Dice que existen dos tipos de esta distorsión, con o sin ruido. Aparte se han estudiado sus métodos de restauración correspondientes.

Primero se ha hablado de imágenes degradadas con motion blur sin ruido, este tipo de imágenes han sido creadas y restauradas con un filtro de dominio frecuencial, usando transformadas de Fourier. Esto ha concluido que la manera más sencilla y más eficaz de restaurar este tipo de distorsión es usar un filtro inverso. Se ha de tener en cuenta que cuanto mayor sea el numero de pixeles usados para distorsionar la imagen original, mayor será la distorsión residual en la estimación.

Finalmente las imágenes motion-blur con ruido han sido estudiadas. Se ha concluido que el método básico de restauración de estas imágenes es el filtro Wiener. El efecto de este filtro depende de un parámetro K que es intuitivo, y también de cómo ha sido distorsionada esta imagen. No es lo mismo tener una varianza insignificante del ruido y un leve efecto de movimiento, que puede ser fácil de restaurar, que tener una gran varianza del ruido y un gran efecto de movimiento, lo que puede ser muy difícil de estimar. Está confirmado que cuando mayor es el valor de K, mejor va a ser la estimación de la imagen original mientras que los valores de la variación del ruido y la longitud de los pixeles estén dentro de un rango aceptable, para que la imagen distorsionada sea moderadamente similar a la original.

Ara que totes les tècniques bàsiques de la degradació digital d'imatge han sigut estudiades i explicades en esta part de l'informe es va a redactar un sumari i veure les conclusions.

Este projecte ha estat focalitzat en l'estudi dels mètodes bàsics de degradació digital d'imatge i la seua subsegüent restauració. Els mètodes de degradacions han sigut investigats per a simular casos reals de corrupció d'imatges i, conèixer com la seua naturalesa funciona pot ajudar a estimar, en una millor manera, com poden ser restaurades.

Primer s'han explicat conceptes bàsics de tractament digital d'imatge incloent el filtrat espacial i en domini freqüencial. Estes tècniques han sigut molt importants durant este projecte. El filtre espacial ha sigut usat per a la restauració d'imatges que han sigut degradades amb soroll, concretament soroll Gaussià i soroll impulsiu. El filtre de domini freqüencial ha sigut usat per a crear i restaurar imatges degradades amb el filtre motion-blur.

S'ha conclòs que les imatges corrompudes amb el soroll Gaussià poden ser restaurades usant filtres de pas davall, però cal tindre compte que unes grans mides de la màscara no és recomanable per a deixar borrosa la imatge. En este document s'ha usat una màscara de 3x3 píxels per a restaurar imatges amb este tipus de soroll i hem resolt que si la variança és massa llarga el procés de restauració no aconsegueix una correcta estimació de la imatge original. Per tant, per a obtindre una estimació acceptable de la imatge original, la dita imatge ha de ser degradada amb una variança de soroll baixa i ser restaurat amb una màscara xicoteta (una màscara de 3x3 píxels és recomanable).

D'altra banda, hi ha un altre model de soroll comú cridat soroll impulsiu. El que es conclou d'açò és que per a restaurar este tipus de degradació la millor manera de fer-ho és el filtre mitjana. Aquest filtre no deixa borrosa la imatge com ho fa el filtre de pas davall, i pot ser aplicat a la imatge degradada moltes vegades si continuen existint píxels corruptes que no han sigut eliminats inclús.

Per a finalitzar, aquest informe parla sobre la distorsió d'imatges digitals per mitjà del motion blur. Diu que hi ha dos tipus d'esta distorsió, amb soroll o sense. A banda s'han estudiat els seus mètodes de restauració corresponents.

Primer s'ha parlat d'imatges degradades amb motion blur sense soroll, este tipus d'imatges han sigut creades i restaurades amb un filtre de domini freqüencial, usant transformades de Fourier. Açò ha conclòs que la manera més senzilla i més eficaç de restaurar este tipus de distorsió és usar un filtre invers. S'ha de tindre en compte que quant major siga el numere de píxels usats per a distorsionar la imatge original, major serà la distorsió residual en l'estimació.

Finalment les imatges motion-blur amb soroll han sigut estudiades. S'ha conclòs que el mètode bàsic de restauració d'estes imatges és el filtre Wiener. L'efecte d'este filtre depèn d'un paràmetre K que és intuïtiu, i també de com ha sigut distorsionada esta imatge. No és el mateix tindre una variança insignificant del soroll i un lleu efecte de moviment, que pot ser fàcil de restaurar, que tindre una gran variança del soroll i un gran efecte de moviment, la qual cosa pot ser molt difícil d'estimar. Està confirmat que quan major és el valor de K, millor serà l'estimació de la imatge original mentres que els valors de la variació del soroll i la longitud dels píxels estiguen dins d'un rang acceptable, perquè la imatge distorsionada siga moderadament semblant a l'original.

6. References

1. Rafael C. González and Richard E. Woods, (2002), *Digital Image Processing* (2nd edition), USR, New Jersey: Prentice Hall: pg. 1 – 2, 116 – 117, 149 – 152, 154 – 159, 221 – 226, 261 – 266.
2. José Ignacio Herranz Herruzo, (2015), *Image Restoration*, Spain, Gandía: Polytechnic School of Gandía, Polytechnic University of Valencia.
3. *Image Processing Laboratory*, Spain, Málaga: Technical High School of Computer Engineering, University of Málaga.
4. Daniel Seguí Simó, (2012), *Designing a man machine interface using the MATLAB environment for detection and removal of impulsive noise in pictures*, Spain, Gandía: Polytechnic School of Gandía, Polytechnic University of Valencia.
5. Javier Ramírez Pérez de Inestrosa, *Introduction to Matlab and Simulink*, Spain, Granada: Department of Signal Theory, Telematics and Communications, University of Granada.
6. J. S. Park, *Matlab GUI (Graphical User Interface) Tutorial for Beginners, Chapter 6*, University of Incheon.
7. http://www.imageprocessingplace.com/
8. http://de.mathworks.com/