

Instituto Tecnológico de Culiacán

**Asignatura:**

Procesamiento Digital de Imágenes

**Profesor:**

Dr. Modesto Guadalupe Medina Melendrez

**Equipo 2.**

Joel Mendoza García

Moisés Ezequiel Domínguez Salcedo

Adrián Torres Sánchez

**Práctica 2:**

Manipulación global de imágenes

**Fecha de entrega:**

Martes 27 de abril del 2021

Contenido

[Manipulación Global de imágenes 3](#_Toc70362899)

[Competencias 3](#_Toc70362900)

[Introducción 3](#_Toc70362901)

[Marco Teórico 3](#_Toc70362902)

[Materiales y Equipos 3](#_Toc70362903)

[Programas 4](#_Toc70362904)

[Procedimiento 4](#_Toc70362905)

[FILTRADO: 4](#_Toc70362906)

[DETECCIÓN DE BORDES: 4](#_Toc70362907)

[PERFILADO: 5](#_Toc70362908)

[Ejercicios: 6](#_Toc70362909)

[Filtrado 6](#_Toc70362910)

[Detección de bordes 13](#_Toc70362911)

[Perfilado 20](#_Toc70362912)

[Conclusiones 21](#_Toc70362913)

[Referencias 22](#_Toc70362914)

# Manipulación Global de imágenes

## Competencias

El estudiante deberá desarrollar las siguientes competencias específicas:

• Comprender la importancia y ubicación del procesamiento de imágenes como una etapa de preprocesamiento requerida para resolver diversos problemas de aplicación práctica.

• Utilizar una herramienta de programación para implementar y utilizar algoritmos que permitan realizar el procesamiento de imágenes.

• Aplicar técnicas o métodos básicos del dominio espacial (operadores aritméticos y convolución) para mejorar el contenido de imágenes.

• Emplear algoritmos y sistemas para realizar filtrado de imágenes en el dominio espacial.

• Desarrollar proyectos de procesamiento de imágenes para aplicaciones reales.

El estudiante fortalecerá las siguientes competencias genéricas que son acordes al perfil de egreso:

• Conocimiento sobre el área de estudio y la profesión.

• Capacidad de aplicar los conocimientos en la práctica.

• Capacidad de investigación.

• Habilidades en el uso de las tecnologías de la información y de la comunicación.

• Habilidades para buscar, procesar y analizar información procedente de fuentes diversas.

• Capacidad de abstracción, análisis y síntesis.

• Capacidad de trabajo en equipo.

• Capacidad para organizar y planificar el tiempo.

• Capacidad de comunicación oral y escrita.

• Capacidad para identificar, plantear y resolver problemas.

• Capacidad creativa.

• Capacidad para formular y gestionar proyectos.

• Capacidad para tomar decisiones.

## Introducción

Jalese Compa Adrian

## Marco Teórico

Jalese compa Adrian

## Materiales y Equipos

* PC
* Camara

## Programas

* Matlab Versión 2011b en adelante con el toolbox digital image processing.

## Procedimiento

Se realizará la manipulación de imágenes mediante convoluciones espaciales. El procedimiento para seguir se lista a continuación:

### FILTRADO:

#### FILTRADO DE RUIDO GAUSSIANO

1. Genere una imagen con ruido gaussiano, dicho ruido deberá tener una media de 0 y varianza de 0.02.
2. Filtrar la imagen generada en el paso anterior mediante filtros de suavizado (media y gaussiano) y de orden (al menos 2 de los vistos en clase).
3. Analizar subjetivamente los resultados obtenidos y concluir cual es el filtro con mejor desempeño.

#### FILTRADO DE RUIDO SAL Y PIMIENTA

1. Genere una imagen con ruido sal y pimienta con densidad de 0.15.
2. Filtrar la imagen generada en el paso anterior mediante filtros de suavizado (al menos 2) y de orden (al menos 2).
3. Analizar subjetivamente los resultados obtenidos y concluir cual es el filtro con mejor desempeño.

#### FILTRADO DE RUIDO UNIFORME PERIÓDICO

1. Busque y seleccione una imagen con ruido frecuencial en uno de sus ejes.
2. Filtrar la imagen seleccionada en el paso anterior mediante un filtro (máscara) diseñado por usted mismo.
3. Analice subjetivamente los resultados obtenidos en el paso 8 y documente sus observaciones.

### DETECCIÓN DE BORDES:

#### GRADIENTES

1. Seleccione 1 de las imágenes que va a procesar en su proyecto final.
2. Genere máscaras de derivada de prewitt (vertical y horizontal) y de sobel (vertical y horizontal).
3. Aplique las máscaras generadas en el paso 11 sobre la imagen seleccionada en el paso 10.
4. Calcule la magnitud del gradiente de cada par de imágenes obtenidas en el paso 12, y umbral ice para detectar bordes.
5. Analice subjetivamente las imágenes obtenidas en el paso 13 y concluya cuál de ellas representa mejor los bordes de las imágenes analizadas.

#### SUAVIZADO MÁS GRADIENTE

1. Aplique un filtro de suavizado con una ventana de la media del tamaño suficiente para eliminar ruidos presentes en la imagen seleccionada en el paso 10.
2. Repita los pasos del 11 al 13 procesando la imagen resultante del paso 15.
3. Analice subjetivamente las imágenes obtenidas en el paso 16 en comparación con las obtenidas en el paso 13, mencione las diferencias y concluya cuales son mejores para detectar bordes.

#### LAPLACIANO

1. Aplique un filtro laplaciano a la imagen seleccionada en el paso 10 y umbralice el resultado para obtener una imagen de bordes.
2. Analice subjetivamente las imágenes obtenidas en el paso 19 en comparación con las obtenidas en el paso 14, mencione las diferencias y concluya cuales son mejores para detectar bordes.

#### SUAVIZADO MÁS LAPLACIANO

1. Aplique un filtro laplaciano a la imagen suavizada obtenida en el paso 15.
2. Compare subjetivamente las imágenes obtenidas en el paso 20 con las obtenidas en el paso 18, mencione las diferencias y concluya cuales son mejores para detectar bordes.

### PERFILADO:

1. Seleccione una máscara laplaciana y conviértala en una máscara de perfilado con el factor de ponderación que mejores resultados subjetivos arroje al procesar la imagen seleccionada en el paso 10.
2. Analice las imágenes perfiladas y comente sobre los resultados obtenidos.

## Ejercicios:

### Filtrado

Durante esta etapa estamos trabajando con imágenes en el modelo de color de escala de grises, para evitar tener que aplicar las etapas de filtrado en cada capa de un modelo con más de una capa (ej. RGB).

#### Código:

clc; clear all; close all;

%---------PUNTO 1------------------------------------------

imagen=imread('imagendeprueba.bmp');

imagenruidoGaus = double(imnoise(imagen,'gaussian',0,0.02))/255;

%---------PUNTO 2 Ruido Gaussiano---------------------------

%Suavizado por Media

mascaraMedia=fspecial('average',[3,3]);

imagen1(:,:,1)=conv2(imagenruidoGaus(:,:,1),mascaraMedia,'same'); %Same quita bordes negros

imagen1(:,:,2)=conv2(imagenruidoGaus(:,:,2),mascaraMedia,'same'); %Same quita bordes negros

imagen1(:,:,3)=conv2(imagenruidoGaus(:,:,3),mascaraMedia,'same'); %Same quita bordes negros

%Suavizado por Mascara Gaussiana

mascaraGussiana=fspecial('gaussian',[13,13],2);

imagen2(:,:,1)=conv2(imagenruidoGaus(:,:,1),mascaraGussiana,'same'); %Same quita bordes negros

imagen2(:,:,2)=conv2(imagenruidoGaus(:,:,2),mascaraGussiana,'same'); %Same quita bordes negros

imagen2(:,:,3)=conv2(imagenruidoGaus(:,:,3),mascaraGussiana,'same'); %Same quita bordes negros

%Filtro de mediana

ventana=[1 1 1;1 1 1;1 1 1];

imagen3(:,:,1)=ordfilt2(imagenruidoGaus(:,:,1),5,ventana);

imagen3(:,:,2)=ordfilt2(imagenruidoGaus(:,:,2),5,ventana);

imagen3(:,:,3)=ordfilt2(imagenruidoGaus(:,:,3),5,ventana);

%Filtro de minimo

imagen4(:,:,1)=ordfilt2(imagenruidoGaus(:,:,1),1,ventana);

imagen4(:,:,2)=ordfilt2(imagenruidoGaus(:,:,2),1,ventana);

imagen4(:,:,3)=ordfilt2(imagenruidoGaus(:,:,3),1,ventana);

% impresion por pantalla

figure(1), subplot(3,2,1),imshow(imagenruidoGaus),title('Imagen Contaminada: Ruido Gausiano')

subplot(3,2,2),imshow(imagen1),title('Imagen Filtrada: mascara media')

subplot(3,2,3),imshow(imagen2),title('Imagen Filtrada: mascara Gaussiana')

subplot(3,2,4),imshow(imagen3),title('Imagen Filtrada: Filtro de Mediana')

subplot(3,2,5),imshow(imagen4),title('Imagen Filtrada: Filtro de minimos')

subplot(3,2,6),imshow(imagen),title('Imagen Original')

%---------PUNTO 4 Sal y Pimienta------------------------------

imagenTR=imread('imagendeprueba.bmp');

imagenTRGray=rgb2gray(imagenTR);

imagenruidoSalyPim = double(imnoise(imagenTRGray,'salt & pepper',0.15))/255;

%---------PUNTO 5 Sal y Pimienta------------------------------

%Suavizado por Media

mascaraMedia2=fspecial('average',[5,5]);

imagenSalyPimMedia=conv2(imagenruidoSalyPim,mascaraMedia2,'same'); %Same quita bordes negros

%Suavizado por Mascara Gaussiana

mascaraGussiana2=fspecial('gaussian',[13, 13],2);

imagenSalyPimGaus=conv2(imagenruidoSalyPim,mascaraGussiana2,'same');

%Filtro de mediana

imagenSalyPimMediana=ordfilt2(imagenruidoSalyPim,13,ones(5));

%Filtro de minimo

imagenSalyPimMaximo=ordfilt2(imagenruidoSalyPim,1,ones(5));

figure(2), subplot(3,2,1),imshow(imagenruidoSalyPim),title('Imagen Contaminada: Sal y Pimienta')

subplot(3,2,2),imshow(imagenSalyPimMedia),title('Imagen Filtrada: mascara media')

subplot(3,2,3),imshow(imagenSalyPimGaus),title('Imagen Filtrada: mascara Gaussiana')

subplot(3,2,4),imshow(imagenSalyPimMediana),title('Imagen Filtrada: Filtro de Mediana')

subplot(3,2,5),imshow(imagenSalyPimMaximo),title('Imagen Filtrada: Filtro de minimos')

subplot(3,2,6),imshow(imagenTRGray),title('Imagen Original')

% % % % % Ruido periodico

imagenRP=imread('filter\_fftdenoise\_before.jpg');

imagenRPn=double(imagenRP)/255;

figure(101),imshow(imagenRPn);,title('Imagen Original');

ventana=[

0 0 0 0 1;

0 0 0 1 0;

0 0 1 0 0;

0 1 0 0 0;

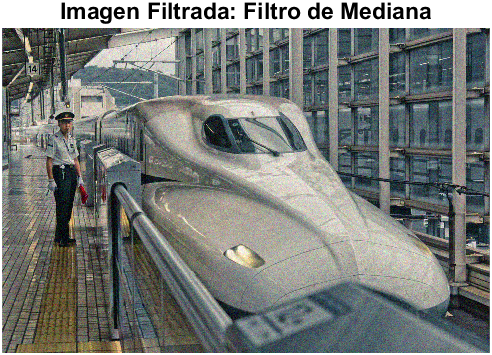
1 0 0 0 0

]/5

imagenRPmedia=conv2(imagenRPn,ventana,'same');

figure(102),imshow(imagenRPmedia);title('ImagenFiltrada: Mascara de la media');

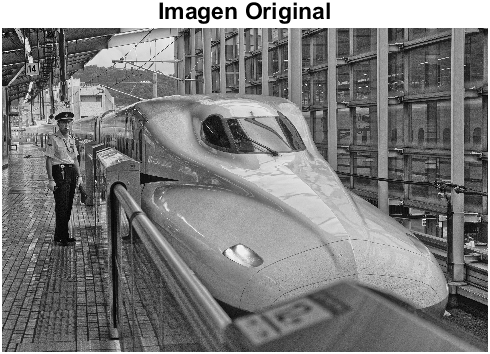
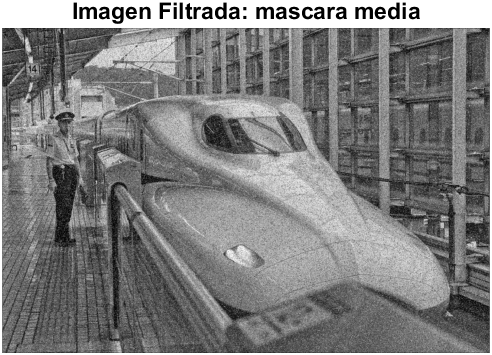
#### FILTRADO DE RUIDO GAUSSIANO



Análisis Subjetivo: Se decide durante el análisis los sig.: el filtro mas apto para eliminar ruido gaussiano es el filtro de mascara de gauss. Sin embargo, todos los demás filtros realizan un buen trabajo, madamas el de mínimos oscurece bastante la imagen.

1. Resultados: Función “Conv2” Utilizados para filtros de la media.
   1. Mascara de la media. Para este caso podemos mencionar que los efectos de suavizado en la imagen, gran parte del ruido gaussiano se ha atenuado, sin embargo, aún se puede apreciar una imagen un tanto contaminada.
   2. Mascara Gaussiana. En este caso la imagen, se aprecia bastante limpia podemos ver como la misma es bastante fiel a la original.
2. Resultados: función “Ordfilt” utilizado en filtros de orden.
   1. Filtro de la mediana, Este filtro en el caso de el ruido gaussiano
   2. Filtro de mínimos, este oscurece la imagen (defecto del filtro) y es capaz de eliminar los valores altos (255), donde se encuentre contaminación.

#### FILTRADO DE RUIDO SAL Y PIMIENTA



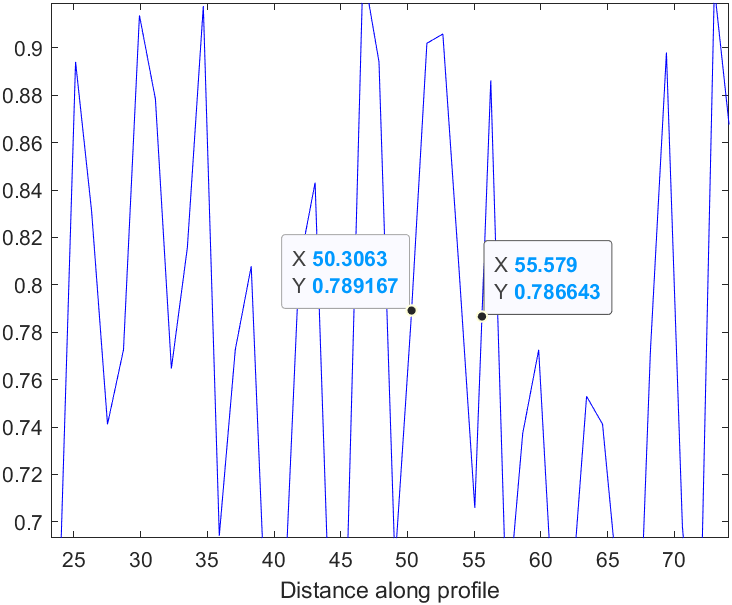
Análisis Subjetivo: Durante el análisis de estos filtros, destacamos que la mejor para forma para reducir ruido sal y pimienta es utilizar el filtro de orden en este caso el que mejor funciona es el de la mediana.

1. Resultados: Función “Conv2” Utilizados para filtros de la media.
   1. Mascara de la media. Para este caso la imagen es bastante similar al proceso de filtrado con mascaras de gauss. Sin embargo, estas aun no alcanzan a corregir totalmente el ruido en la imagen.
   2. Mascara Gaussiana. En este caso la imagen, se aprecia bastante limpia, si bien se reduce el nivel de ruido aun se miran los efectos provocados por una mascara gaussiana en la imagen. Como si el ruido se hubiese hecho un poco menos perceptible. En varias zonas.
2. Resultados: función “Ordfilt” utilizado en filtros de orden.
   1. Filtro de la mediana, En este caso se pueden apreciar resultados a partir de filtrado, bastante buenos nótese como la imagen original es bastante fiel a la imagen filtrada.
   2. Filtro de mínimos, Este filtro elimino los rastros del ruido sal, sin embargo, oscureció tanto la imagen que podríamos decir que se arruino la misma a nivel visual no se alcanzan a percibir casi detalles en la imagen.

#### FILTRADO DE RUIDO UNIFORME PERIOODICO



Análisis subjetivo: Durante este ejercicio se estuvieron revisado varias de las posibilidades (que tipo de mascara de la media usar por ej.), sin embargo, al observar la imagen vemos que tenemos un comportamiento periódico, en una de las diagonales, para el análisis de la variación de pixeles, estuvimos usando la función “improfile”, determinamos que el valor mas adecuado seria aprox. 5 en base a los resultados del análisis de los resultados de las variaciones. Se destaca que el hecho de filtrar en el régimen espacial ruido periódico, se vuelve un poco más difícil porque hay que analizar que tipo de ruido periódico tenemos.



### Detección de bordes

#### Codigo:

clc; clear all;close all;

%---------PUNTO 10----------------------------------------------------

imagen=imread('placas4.jpeg');

subimagen=imagen(302:590,:,:);

placasGray=double(rgb2gray(subimagen))/255;

figure(1),imshow(placasGray),title('Imagen Original')

%---------PUNTO 11----------------------------------------------------

%Mascaras de derivadas

prewittHorizontal=fspecial('prewitt');

prewittVertical=prewittHorizontal';

sobelHorizontal=[-1 -2 -1;0 0 0;1 2 1];

sobelVertical=[-1 0 1;-2 0 2;-1 0 1];

%---------PUNTO 12 Aplicacione de las mascaras------------------------

PreHorizontal=imfilter(placasGray,prewittHorizontal,'conv');

figure(2),imshow(PreHorizontal),title('Prewitt Horizontal')

PreVertical=imfilter(placasGray,prewittVertical,'conv');

figure(3),imshow(PreVertical),title('Prewitt Vertical')

SobHorizontal=imfilter(placasGray,sobelHorizontal,'conv');

figure(4),imshow(SobHorizontal),title('Sobel Horizontal')

SobVertical=imfilter(placasGray,sobelVertical,'conv');

figure(5),imshow(SobVertical),title('Sobel Vertical')

%---------PUNTO 13 Magnitud del gradiente-----------------------------

MagnitudGradientePrewitt=sqrt(PreHorizontal.^2+PreVertical.^2);

figure(6),imshow(MagnitudGradientePrewitt,[]),title('Magnitud Gradiente Prewittl')

MagnitudGradienteSobel=sqrt(SobHorizontal.^2+SobVertical.^2);

figure(7),imshow(MagnitudGradienteSobel,[]),title('Magnitud Gradiente Sobel')

%---------PUNTO 15 SUAVIZADO MÁS GRADIENTE---------------------------

mascaraMedia=fspecial('average',[5,5]);

imagenSuavizada=conv2(placasGray,mascaraMedia,'same'); %Same quita bordes negros

figure(8),imshow(imagenSuavizada),title('Imagen Suavizada')

%---------PUNTO 16---------------------------------------------------

PreSuavHorizontal=imfilter(imagenSuavizada,prewittHorizontal,'conv');

figure(9),imshow(PreSuavHorizontal),title('Suavizada con Prewitt Horizontal')

PreSuavVertical=imfilter(imagenSuavizada,prewittVertical,'conv');

figure(10),imshow(PreSuavVertical),title('Suavizada con Prewitt Vertical')

SobSuavHorizontal=imfilter(imagenSuavizada,sobelHorizontal,'conv');

figure(11),imshow(SobSuavHorizontal),title('Suavizada con Sobel Horizontal')

SobSuavVertical=imfilter(imagenSuavizada,sobelVertical,'conv');

figure(12),imshow(SobSuavVertical),title('Suavizada con Sobel Vertical')

MagnitudGradienteSuavizadoPrewitt=sqrt(PreSuavHorizontal.^2+PreSuavVertical.^2);

figure(13),imshow(MagnitudGradienteSuavizadoPrewitt,[]),title('Magnitud Gradiente Prewittl Suavizado')

MagnitudGradienteSuavizadoSobel=sqrt(SobSuavHorizontal.^2+SobSuavVertical.^2);

figure(14),imshow(MagnitudGradienteSuavizadoSobel,[]),title('Magnitud Gradiente Sobel Suavizado')

%---------PUNTO 18 Laplaciano---------------------------------------------------

mascaraLaplaciana=fspecial('laplacian');

imagenLaplaciana=imfilter(placasGray,mascaraLaplaciana,'conv');

figure(15),imshow(imagenLaplaciana),title('Imagen con filtro laplaciano')

Ib=abs(imagenLaplaciana)>0.15; %Umbralizado de la imagen

figure(16),imshow(Ib),title('Imagen Umbralizada')

%---------PUNTO 18 Laplaciano---------------------------------------------------

mascaraLaplaciana=fspecial('laplacian');

imagenLaplaciana=imfilter(placasGray,mascaraLaplaciana,'conv');

figure(15),imshow(imagenLaplaciana),title('Imagen con filtro laplaciano')

Ib=abs(imagenLaplaciana)>0.15; %Umbralizado de la imagen

figure(16),imshow(Ib),title('Imagen Umbralizada')

%---------PUNTO 20 Suavizado + Laplaciano-----------------------------------------

imagenSUAVmasLAPL=imfilter(imagenSuavizada,mascaraLaplaciana,'conv');

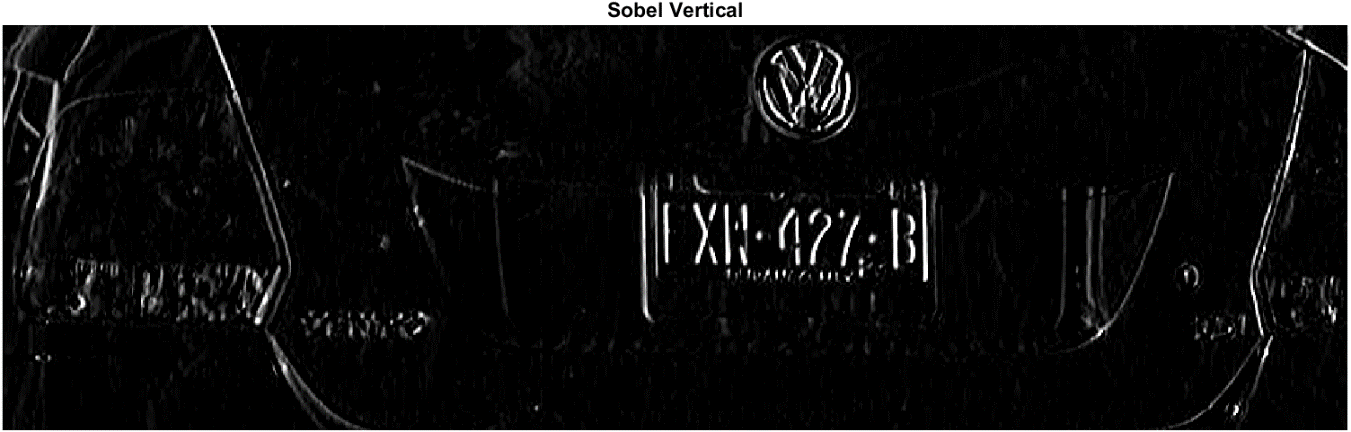
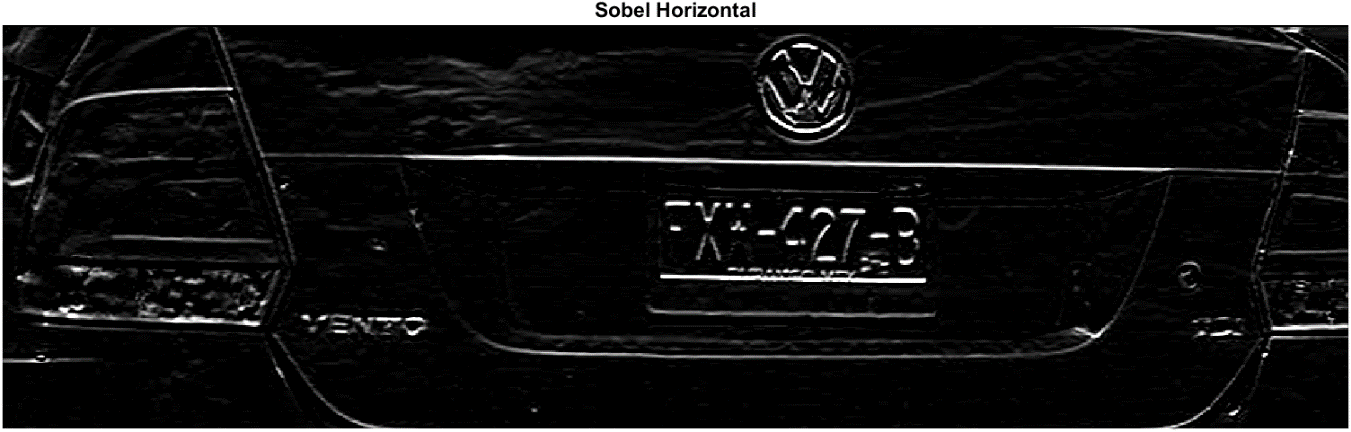
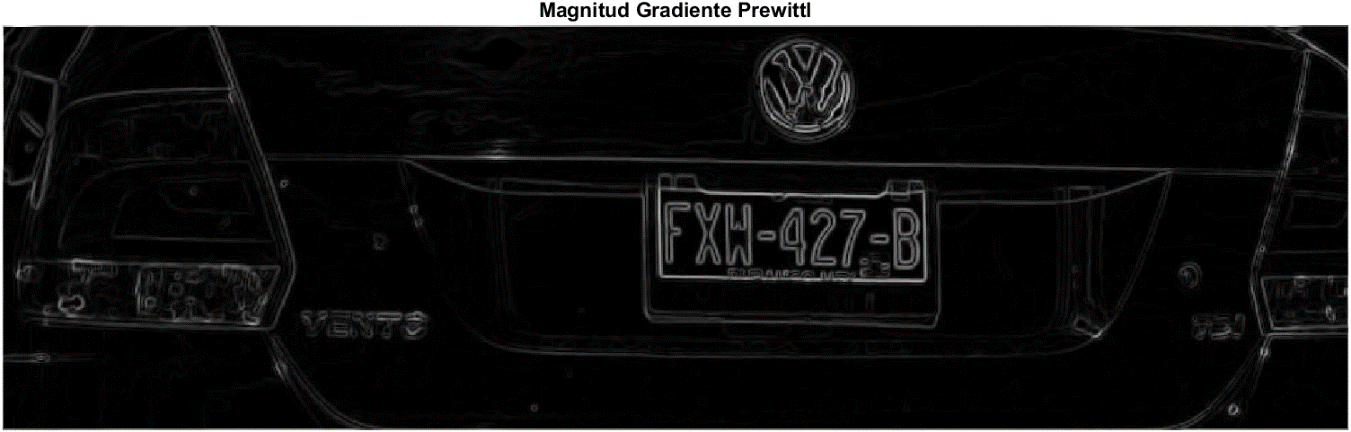
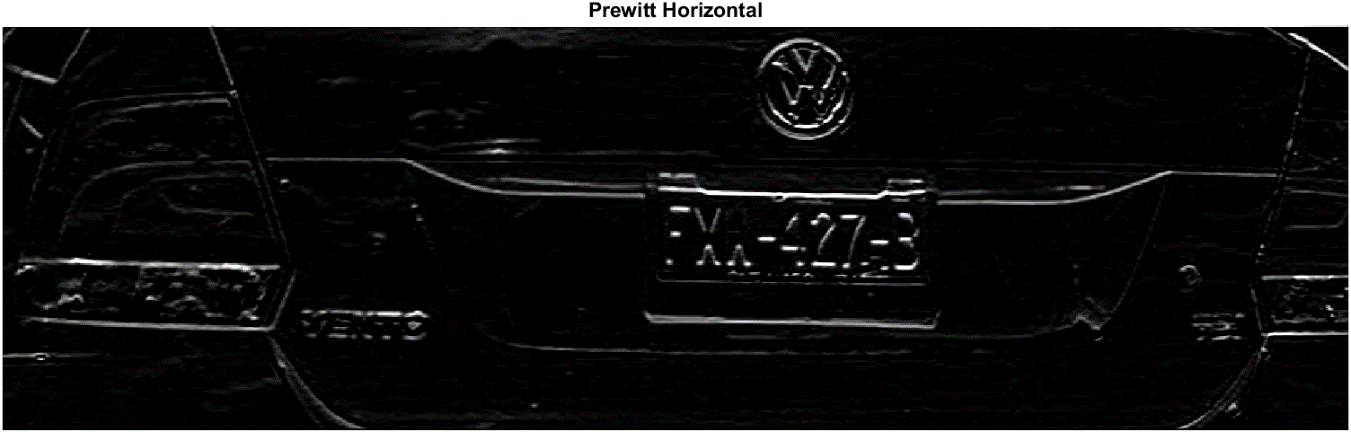
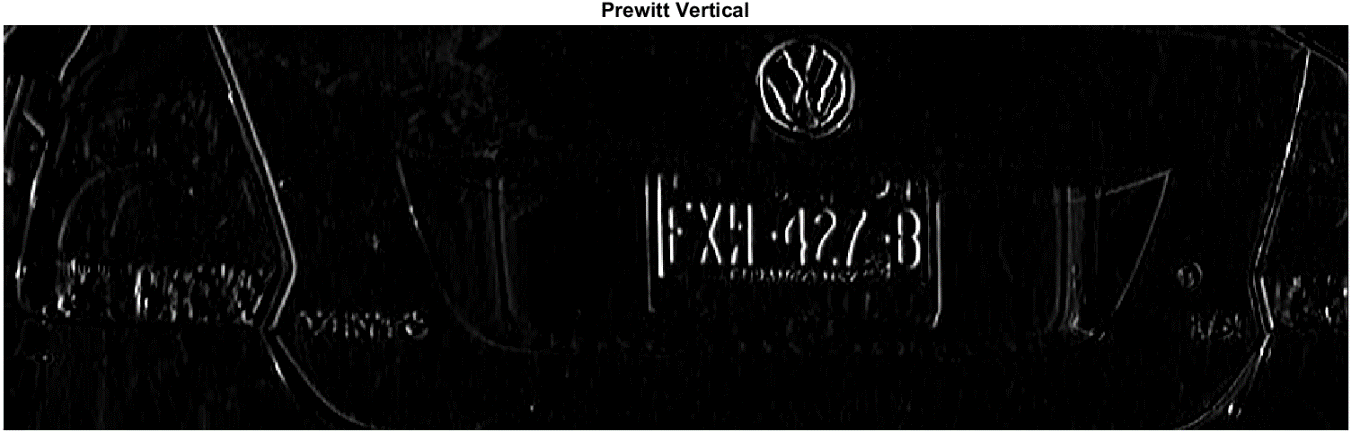
figure(17),imshow(imagenSUAVmasLAPL),title('Imagen Suavizada + Laplaciano')

Ib2=abs(imagenSUAVmasLAPL)>0.05; %Umbralizado de la imagen

figure(18),imshow(Ib2),title('Imagen Umbralizada 2')

#### GRADIENTES

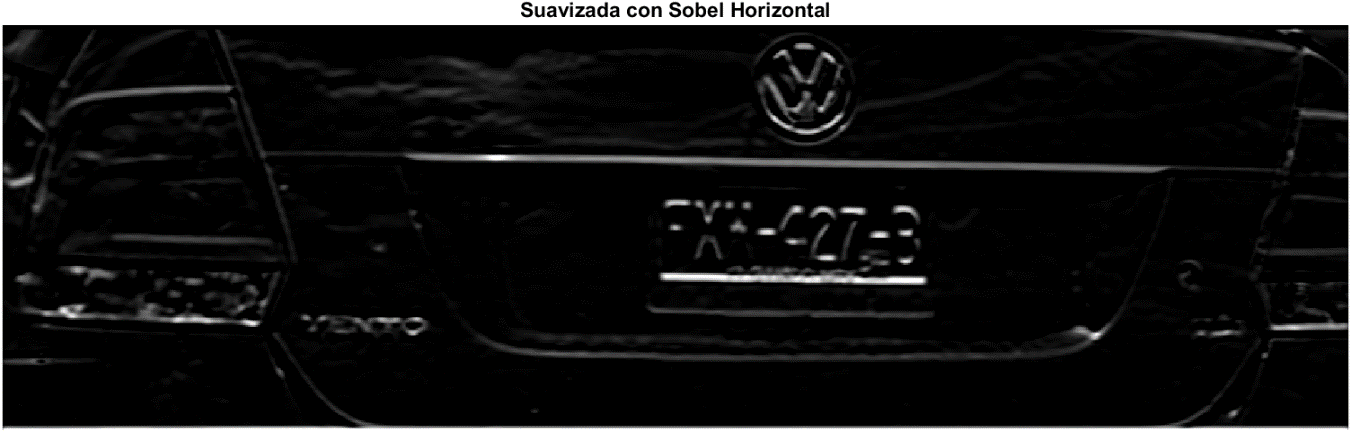
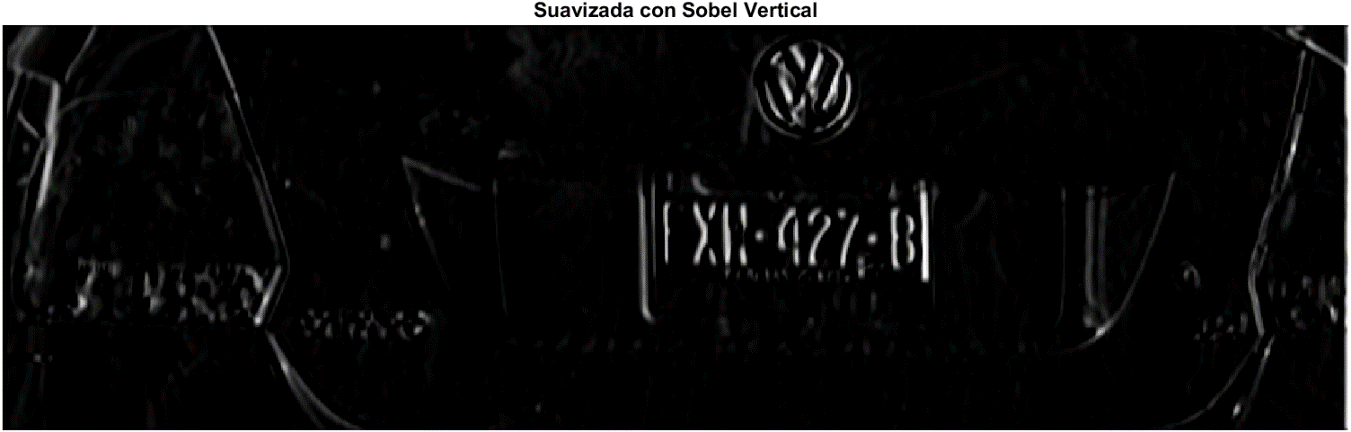
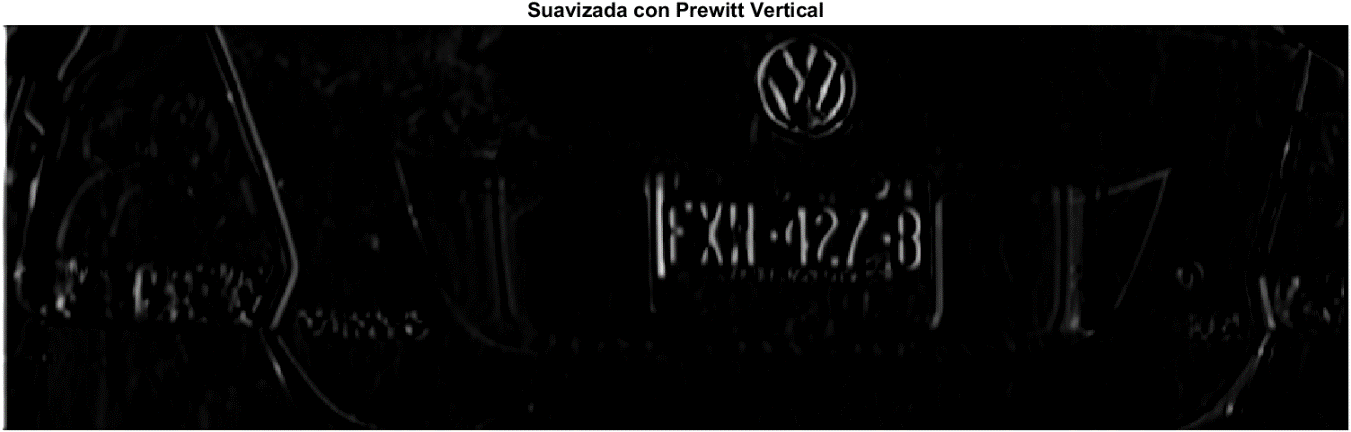
Sobel vs Prewittt. En este caso realmente se vuelve imperceptible las diferencias en las imágenes. Sin embargo, a nivel de detección creo que tiene más capacidad el filtro sobel, tiende a detectar menos falsos positivos como lo podemos apreciar en las imágenes sig. Prewitt en nuestro caso arroja líneas en el filtro horizontal que en sobel no existen mismas que no nos sirven de nada ya que son provocadas por las sombras.



#### SUAVIZADO MAS GRADIENTES

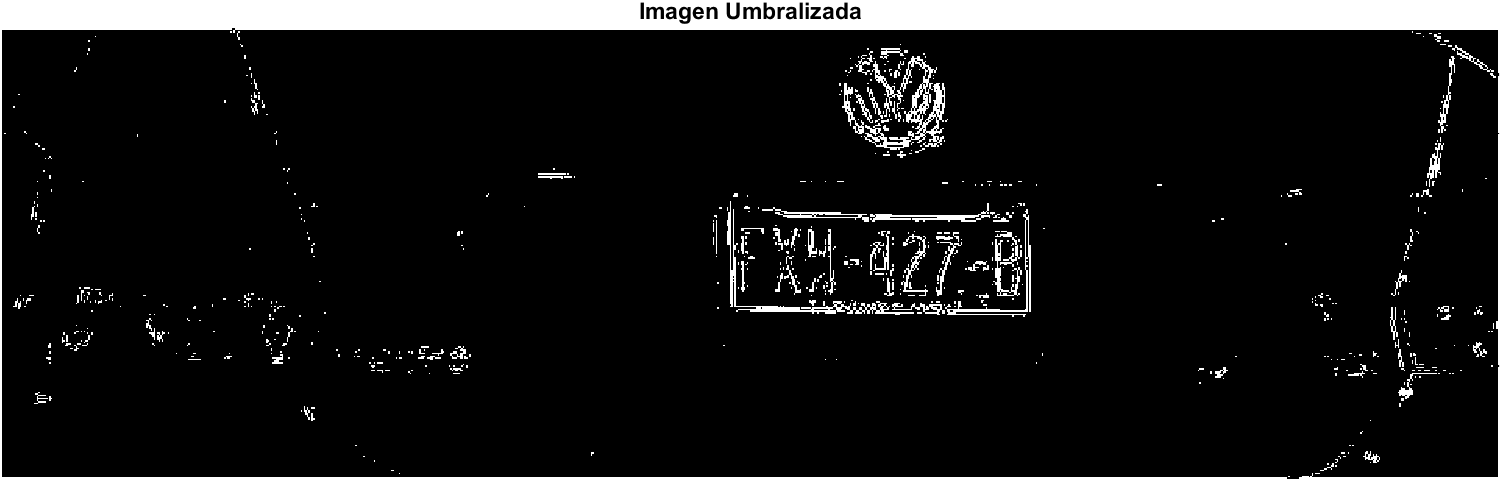
Suavizado vs. No suavizado: A diferencia de las máscaras de derivadas de sobel en prewitt, se puede apreciar como las imágenes son un tanto inferiores en cuestión de calidad al filtro de sobel, las mimas contiene mas ruido, puedo decir que la máscara de sobel suaviza un poco mas la imagen aún.

A diferencia de las imágenes anteriores donde no se aplican procesos de suavizado antes de detectar cambios o aplicar las máscaras de derivada, supone que las mismas en este caso se miran un poco más borrosas en algunas zonas provocando que no se detecten cambios por ejemplo en las imágenes las calaveras eran rojas al igual que el carro, al suavizar la imagen se provoco que se difuminaran un poco mas esos colores y en algún punto empataran los cambios, como si en esas zonas no existieran cambios.



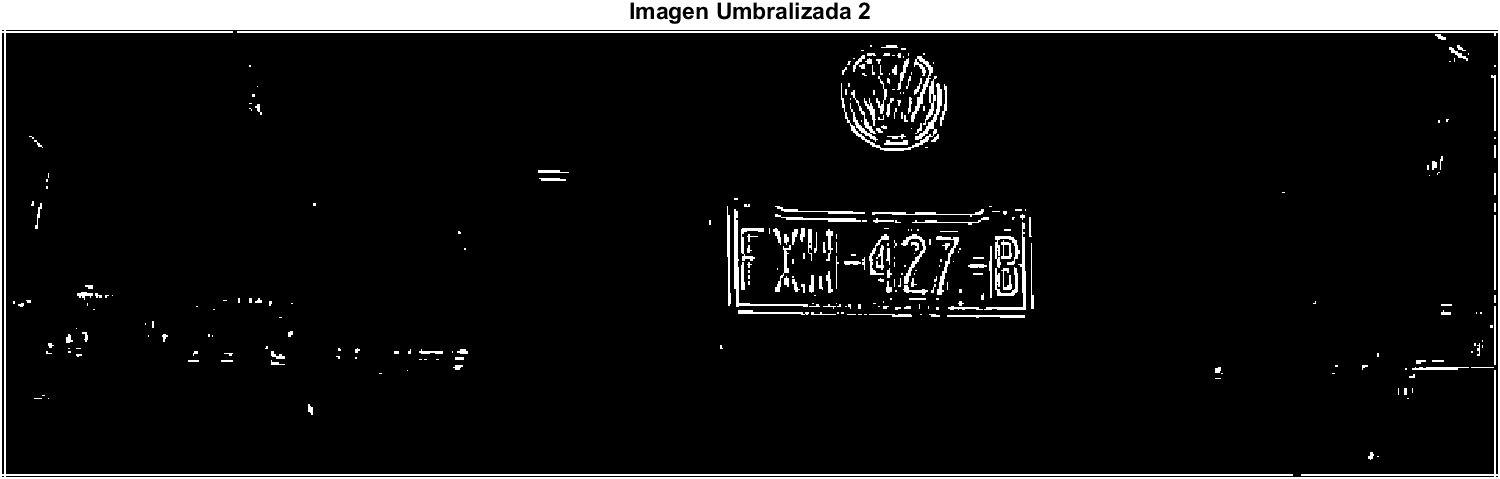
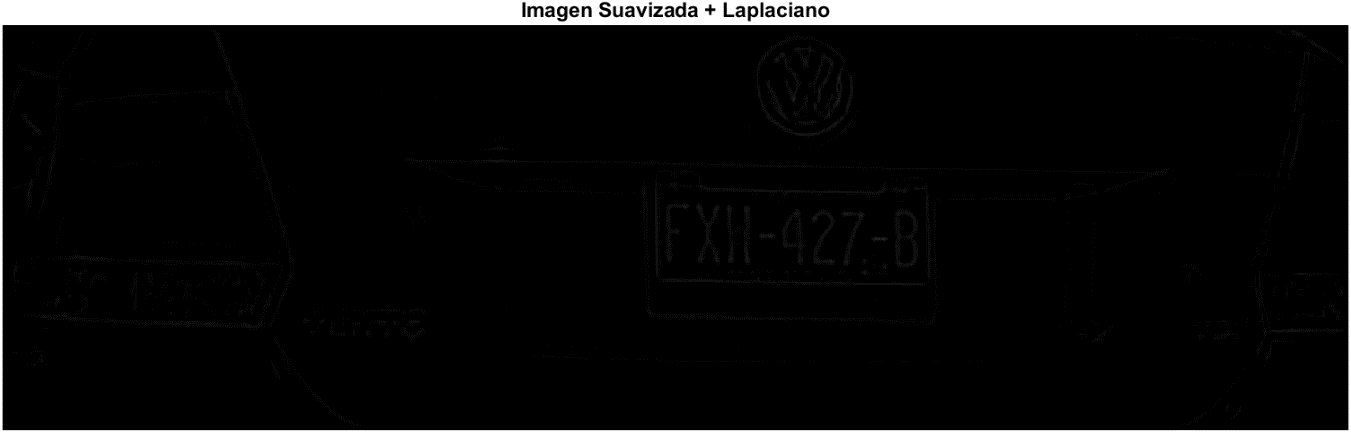
#### LAPLACIANO

Laplaciano Vs Gradiente. Durante este proceso detectamos cambios sobre cambios al ser una máscara de la segunda derivada (mascara Laplaciana). Entonces obtenemos imágenes con una menor cantidad de cambios. En comparación a las que teníamos con el cálculo de la magnitud del gradiente.



#### SUAVIZADO MAS LAPLACIANO

Suavizado más Laplaciano Vs Laplaciana. En estas imágenes podemos ver que los bordes mas grandes que estaban menos definidos en los otros casos se han desaparecido, dado que la máscara laplaciana al aplicarle a la imagen un preprocesado(en este casos suavizado), disminuimos aun mas el contenido de bordes en la imagen.



### Perfilado



#### Código:

%---------PUNTO 22 Perfilado-----------------------------------------

mascaraLaplaciana=fspecial('laplacian');

M=[0,0,0;0,1,0;0,0,0]-2\*mascaraLaplaciana

imagenPerfilada=imfilter(placasGray,M,'same');

figure(19),imshow(imagenPerfilada),title('Perfilado')

## Conclusiones

Durante el análisis de filtrado de imágenes y detección de cambios, podemos decir que contamos con los conocimientos necesarios para realizar o desarrollar algunos algoritmos con los que podamos detectar por ejemplo bordes, limpiar imágenes finalmente si bien es cierto el análisis realizado durante esta práctica solo involucra el proceso de binarizacion mediante detección de cambios (sobel, Prewitt), con estos resultados ya podríamos empezar a trabajar sobre un proyecto más grande. La idea es implementar estos criterios en nuestro trabajo final de detección de placas.

## Referencias

* Apuntes sobre Procesamiento Digital de Imágenes.
* Manuales de Image Processing Toolbox de MATLAB.
* R.C. Gonzalez, R.E. Woods, Digital Image Processing, Ed. Addison Wesley, 1992.