

Asignatura: Procesamiento de imágenes

Profesor: D. Sc. Gerardo García Gil

Alumno: Jonathan Guillermo Díaz Magallanes.

Registro: 19310153. **Ciclo:** 2022-B

Ingeniería en Desarrollo de Software

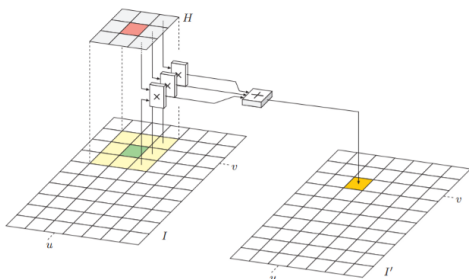
Centro de Enseñanza Técnica Industrial (CETI)

Introducción

A lo largo de ésta práctica se documentará una práctica que consiste en la aplicación de los filtros de borrado simple, box, Gaussiano y el filtro para la detección de bordes Laplaciano con las máscaras proporcionadas a una imagen en escala de grises usando MATLAB como el lenguaje de cálculo.

Filtro lineal

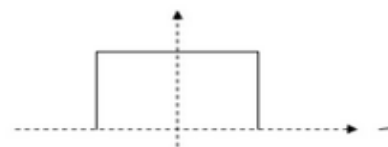
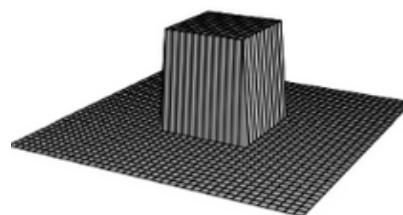
El filtro lineal es un filtro que opera el valor de píxel en la región de soporte de manera lineal (es decir, como suma ponderada). La región de soporte se especifica mediante la 'matriz de filtro' y se representa como $H(i,j)$. El tamaño de H se llama 'región de filtro' y la matriz de filtro tiene su propio sistema de coordenadas, i es índice de columna y j es índice de fila. El centro de la misma es la ubicación de origen y se llama el 'punto caliente'.



Filtro Box

El filtrado de cajas es básicamente un tipo de filtrado de imágenes de promedio de píxeles circundantes.

En realidad, es un filtro de convolución que es una operación matemática comúnmente utilizada para el filtrado de imágenes. El poder del filtro box es que uno puede escribir un filtro de imagen general que puede enfocar, emboscar, detectar bordes, suavizar, desenfocar movimiento, etcétera.

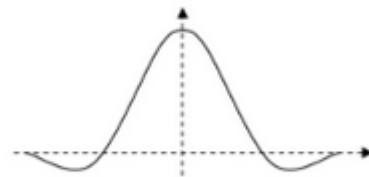
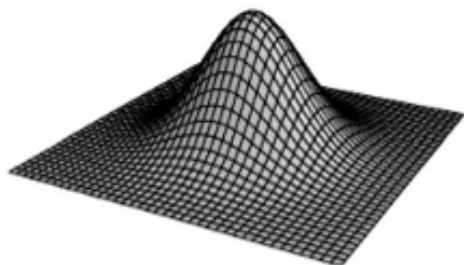


0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

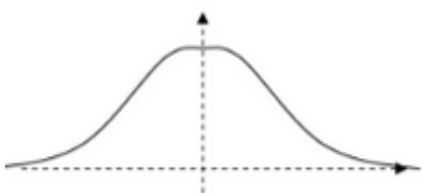
Filtro Gaussiano

Los filtros gaussianos juegan un papel importante en el filtrado de diferentes tipos de superficies. La simplicidad del algoritmo, la facilidad de implementación y la robustez de los resultados hacen de este tipo de filtración la primera opción para la filtración en muchas aplicaciones. El filtro gaussiano lineal es muy popular en la caracterización de

superficies, ha sido ampliamente utilizado entre los investigadores y se ha convertido en un estándar de filtración industrial. Los filtros gaussianos se pueden aplicar a la superficie de entrada haciendo girar la superficie medida con una función de ponderación gaussiana.



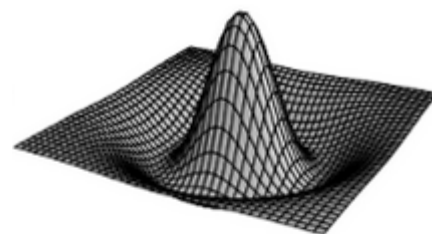
0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0



0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

Filtro de sombrero mexicano o laplaciano

El filtro laplaciano es un filtro derivado de segundo orden utilizado en la detección de bordes, en el procesamiento digital de imágenes. En los filtros derivados de 1er orden, detectamos el borde junto con las direcciones horizontal y vertical por separado y luego combinamos ambas. Pero usando el filtro laplaciano detectamos los bordes en toda la imagen a la vez.



Desarrollo

Lo primero que se realizó fué guardar a trabajar dentro de una variable y convertirla a escala de grises con las funciones `imread` para la lectura y `rgb2gray` para la conversión, las cuales son proporcionadas por las librerías de MATLAB.



Después de ello, obtendremos las dimensiones de la imagen y dentro de una nueva variable guardaremos la misma imagen pero convertida a valores `double` para poder manipular los valores de la matriz

matemáticamente.

```
% Convertimos la imagen A a escala de grises
I = rgb2gray(A);
[Filas, Columnas, P] = size(I);
% Pasamos a double la imagen
D = double(I);
Dr = D;
--
```

Una vez terminado los preparativos, empezamos multiplicando los valores de la matriz de la imagen por la máscara de cada uno de los filtros por medio de un barrido de dos for anidados.

Filtro simple

Para el filtro simple la máscara utilizada es:

1	1	1
1	1	1
1	1	1

Siendo ésta una matriz de 3x3 por lo que a la sumatoria del peso de los pixeles se va a dividir entre 9, proporcionandonos el siguiente resultado:



Filtro box

Para el filtro box se usó la misma máscara que en el filtro simple con la diferencia que se añadió el peso a los píxeles de alrededor de la vecindad-8:

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Debido al cambio de la cantidad de píxeles los cuales están siendo afectados también cambia, pero al ser de peso nulo el valor del borde exterior se mantendrá la misma división de peso, siendo ésta 9. Obteniendo la siguiente imagen:



Filtro Gaussiano

El tamaño de la matriz para el filtro gaussiano es la misma que en el filtro box, siendo ésta de 25 píxeles, sin embargo el peso que van a tener cada uno de los píxeles cambia al siguiente:

0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

Debido al cambio de peso de los píxeles el efecto borroso de la imagen aumentará y será más notorio a comparación de los dos filtros anteriores. Por la misma razón del cambio del peso, se deberá aumentar el factor de divisor de 9 a 57, para que al momento de hacer el efecto de no se vea todo blanco y se haga la transición entre los píxeles correctamente. Cambiando los valores se obtiene lo siguiente:

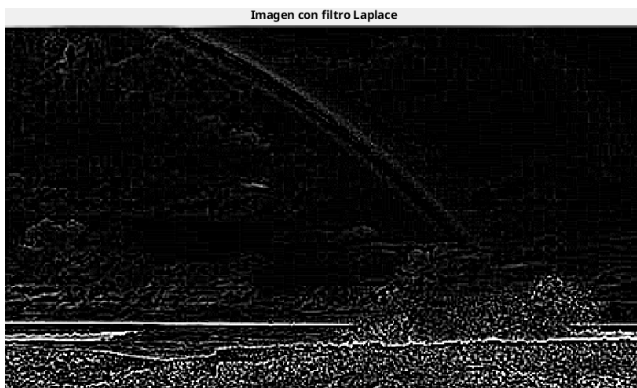


Filtro laplaciano

Para el filtro detector de bordes o laplaciano también se usó una máscara del mismo tamaño (25 píxeles), sin embargo el peso de los píxeles cambió nuevamente, siendo éstos ahora:

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Otra diferencia con el factor de divisor es que ésta vez no dividiremos los valores por ninguno, éso debido a que la imagen adopte valores extremos para que se logre apreciar los bordes detectados por el algoritmo. Al momento de aplicar éste filtro obtenemos la siguiente imagen:



Código

% borrado de variables y termianle

```
clear all; clc;
```

```
% Leemos la imagen
```

```
A = imread("arco.jpg");
```

```
% Convertimos la imagen A a escala de grises
```

```
I = rgb2gray(A);
```

```
[Filas, Columnas, P] = size(I);
```

```
% Pasamos a double la imagen
```

```
D = double(I);
```

```
Dr = D;
```

```
%
```

```
% Empezamos el filtro borroso
```

```
for r=2:Filas-1
```

```
    for c=2:Columnas-1
```

```
        Dr(r,c)=(1/9)*(D(r,c)+D(r+1,c)+D(r+1,c-1)+D(r,c-1)+D(r-1,c-1)+D(r-1,c)+D(r-1,c+1)+D(r,c+1)+D(r+1,c+1));
```

```
    end
```

```
end
```

```
Dr = uint8(Dr);
```

```
%
```

```
% Empezamos el filtro box lineal
```

```
Fr = D;
```

```
for r=3:Filas-2
```

```
    for c=3:Columnas-2
```

```
        Fr(r,c)=(1/9)*((D(r,c))+D(r+1,c))+D(r+1,c-1))+D(r,c-1
```

```

))+(D(r-1,c-1))+(D(r-1,c))+(D(r-1,c+1))+(D(r,c+1))+(D(r+
1,c+1))+(D(r+2,c)*0)+(D(r+2,c-1)*0)+(D(r+2,c-2)*0)+(
D(r+1,c-2)*0)+(D(r,c-2)*0)+(D(r-1,c-2)*0)+(D(r-2,c-2)*
0)+(D(r-2,c-1)*0)+(D(r-2,c)*0)+(D(r-2,c+1)*0)+(D(r-2,c
+2)*0)+(D(r-1,c+2)*0)+(D(r,c+2)*0)+(D(r+1,c+2)*0)+(
D(r+2,c+2)*0)+(D(r+2,c+1)*0));

```

```

end

```

```

end

```

```

Fr = uint8(Fr);

```

```

%
-----
-----

```

```

% Empezamos con el filtro Gauss

```

```

Gr = D;

```

```

for r=3:Filas-2

```

```

    for c=3:Columnas-2

```

```

Gr(r,c)=(1/57)*((D(r,c)*9)+(D(r+1,c)*5)+(D(r+1,c-1)*3)
+(D(r,c-1)*5)+(D(r-1,c-1)*3)+(D(r-1,c)*5)+(D(r-1,c+1)*
3)+(D(r,c+1)*5)+(D(r+1,c+1)*3)+(D(r+2,c)*2)+(D(r+2,c
-1)*1)+(D(r+2,c-2)*0)+(D(r+1,c-2)*1)+(D(r,c-2)*2)+(D(
r-1,c-2)*1)+(D(r-2,c-2)*0)+(D(r-2,c-1)*1)+(D(r-2,c)*2)
+(D(r-2,c+1)*1)+(D(r-2,c+2)*0)+(D(r-1,c+2)*1)+(D(r,c+
2)*2)+(D(r+1,c+2)*1)+(D(r+2,c+2)*0)+(D(r+2,c+1)*1))
;

```

```

end

```

```

end

```

```

Gr = uint8(Gr);

```

```

%
-----
-----

```

```

% Empezamos con el filtro Laplace

```

```

Lr = D;

```

```

for r=3:Filas-2

```

```

    for c=3:Columnas-2

```

```

Lr(r,c)=(1/1)*((D(r,c)*16)+(D(r+1,c)*-2)+(D(r+1,c-1)*-
1)+(D(r,c-1)*-2)+(D(r-1,c-1)*-1)+(D(r-1,c)*-2)+(D(r-1,c
+1)*-1)+(D(r,c+1)*-2)+(D(r+1,c+1)*-1)+(D(r+2,c)*-1)+(
D(r+2,c-1)*0)+(D(r+2,c-2)*0)+(D(r+1,c-2)*0)+(D(r,c-2)
*-1)+(D(r-1,c-2)*0)+(D(r-2,c-2)*0)+(D(r-2,c-1)*0)+(D(r
-2,c)*-1)+(D(r-2,c+1)*0)+(D(r-2,c+2)*0)+(D(r-1,c+2)*0
)+(D(r,c+2)*-1)+(D(r+1,c+2)*0)+(D(r+2,c+2)*0)+(D(r+2
,c+1)*0));

```

```

end

```

```

end

```

```

Lr = uint8(Lr);

```

```

% -----AREA DE
IMPRESION-----

```

```

% Impresion de la imagen original

```

```

% subplot(2,2,1)

```

```

figure;

```

```

imshow(I)

```

```

title('Imagen original')

```

```

% Impresion de la imagen con filtro borroso

```

```

% subplot(2,2,2)

```

```

figure;

```

```

imshow(Dr)

```

```

title('Imagen con filtro borroso')

```

```

% Impresion de la imagen con filtro Gauss

```

```

% subplot(2,2,3)

```

```

figure;

```

```

imshow(Fr)

```

```

title('Imagen con filtro Box lineal')

```

```
% Impresion de la imagen con filtro Gauss
```

```
% subplot(2,2,4)
```

```
figure;
```

```
imshow(Gr)
```

```
title('Imagen con filtro Gauss')
```

```
% Impresion de la imagen con filtro Laplace
```

```
% subplot(2,2,4)
```

```
figure;
```

```
imshow(Lr)
```

```
title('Imagen con filtro Laplace')
```

septiembre de 2022, de

<https://idoc.pub/documents/filtros-laplaci>

ano-8jlkmyg6zzn5

- *Tech-Algorithm.com ~ Box Filtering*. (s. f.). Recuperado 19 de septiembre de 2022, de <https://www.tech-algorithm.com/articles/boxfiltering/>

Conclusión

A lo largo del desarrollo de la práctica no se presentaron problemas mayores al momento de hacer la multiplicación de la matriz de la imagen por la máscara del filtro, sin embargo resultó un poco tedioso tener que escribir la línea completa, en específico a partir de las matrices de 5x5. Este proceso se realizó más sencillo gracias a que el documento fue editado por el editor de textos NVIM en lugar del que ofrece la IDE de MATLAB.

Referencias

- Jiang, J. & Scott, P. J. (2020). *Gaussian filter*. ScienceDirect. Recuperado 18 de septiembre de 2022, de <https://www.sciencedirect.com/topics/engineering/gaussian-filter>
- Juárez Sandoval, O. U. (2019). *Filtros Laplaciano*. Recuperado 19 de