

Filtrado Lineal

Tomas Lopez Pérez

May 2024

1 Introduction

El procesamiento de imágenes digitales se utiliza para mejorar la calidad de las imágenes y extraer información valiosa de ellas. Esto puede implicar técnicas como la mejora de la imagen, la segmentación de la imagen y la transformación de la imagen. Estas técnicas se utilizan en una variedad de campos [HST71], incluyendo el reconocimiento de patrones y la codificación eficiente de imágenes. Un aspecto importante del procesamiento de imágenes digitales es el manejo de imágenes RAW. Las imágenes RAW son datos sin procesar directamente del sensor de la cámara. Estas imágenes ofrecen una mayor flexibilidad en el procesamiento posterior porque contienen más detalles y menos artefactos de compresión que las imágenes procesadas.

2 Fundamento teórico

2.1 Procesamiento de imágenes

El Procesamiento de imágenes es un campo de investigación muy extenso que involucra diversas áreas del conocimiento, ya que involucra diversos procesos, tales como: la adquisición, transmisión, representación y procesamiento. En términos generales el procesamiento de imágenes se utiliza para modificar una imagen con el fin de mejorar la apariencia visual para un observador y para resaltar convenientemente el contenido de la imagen de cara a la percepción por parte de máquinas, es decir, se hace una manipulación de imágenes con objeto de producir nuevas imágenes que son mejores, en algún sentido. El procesamiento de imágenes comprende distintas técnicas que se utilizan para mejorar, restaurar e inclusive comprimir imágenes ya existentes; la implementación y desarrollo de dichas técnicas agregadas a sistemas ya existentes nos permiten editar imágenes de forma más sencilla y precisa [Alo18]. La Figura 1 muestra el flujo de las etapas del procesamiento digital de imágenes propuestas por Gonzalez en donde se establece como la etapa inicial la adquisición de la imagen digital, lo que se puede hacer mediante algún sistema de sensores que permitan digitalizar la imagen u obtener las imágenes de algún repositorio o colección. El conjunto de imágenes adquiridas conforma inicialmente la base de conocimiento, ya que para algunas

aplicaciones basta con tener la información pura, sin embargo en la mayoría de las aplicaciones del procesamiento de imágenes, es necesario enriquecer la base del conocimiento con al menos las etapas de procesado y segmentación, lo cual requiere aplicar una serie de operadores que transformen las imágenes para resaltar características relevantes, en el caso del procesado, o separar los diferentes elementos que componen la imagen, en el caso de la segmentación, de modo que los resultados de estas dos etapas permiten tener una base del conocimiento más amplia y robusta. Respecto a la etapa de representación y descripción, las imágenes procesadas y/o segmentadas son usadas para extraer características más específicas sobre la imagen, tales como cantidad, dimensión, forma, posición, etc. de los objetos de interés en las imágenes, esta información también puede formar parte de la base del conocimiento, de modo que en la última etapa, reconocimiento e interpretación, se puedan obtener respuestas a partir de las imágenes y la información generada por estas.

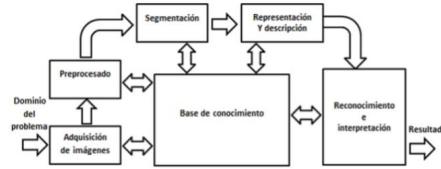


Figure 1: Etapas fundamentales del procesamiento digital de imágenes.

2.2 Filtrado Lineal

El filtrado es una técnica para modificar o mejorar una imagen. Por ejemplo, se puede filtrar una imagen para realzar ciertas características o eliminar otras. Algunas de las operaciones de procesamiento de imágenes que se implementan con filtros son el suavizado, el realce de la nitidez y la mejora de los bordes. El filtrado es una operación de entorno, en la que el valor de un píxel particular de la imagen de salida se determina aplicando un algoritmo a los valores de los píxeles del entorno del píxel de entrada correspondiente. El entorno de un píxel es un conjunto de píxeles definido por sus ubicaciones en relación con dicho píxel. El filtrado lineal consiste en el filtrado en el que el valor de un píxel de salida es una combinación lineal de los valores de los píxeles del entorno del píxel de entrada.

2.3 Imagen digital

Una imagen digital puede ser definida como una función bidimensional $f(x, y)$ de la luz y la intensidad, donde el f representa la intensidad de luz capturada en una coordenada espacial (x, y) , para la representación de $f(x, y)$ [GW18]. Una imagen digital es comúnmente representada mediante una matriz de $M \times N$ valores discretos y finitos, cada elemento de la matriz recibe el nombre de píxel,

que corresponden al valor de intensidad da cada punto (x, y) de la matriz que representa la imagen. Por lo general, los píxeles están organizados en una matriz rectangular ordenada. El tamaño de una imagen está determinado por las dimensiones de esta matriz de píxeles. El ancho de imagen es el número de columnas M y la altura es el número de filas N en la matriz de píxeles [She13]. La notación matricial de una imagen se muestra en 1.

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \dots & f(1, N - 1) \\ \vdots & \vdots & & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & \dots & f(M - 1, N - 1) \end{bmatrix} \quad (1)$$

El utilizar una matriz para almacenar una imagen plantea muchas ventajas al momento de procesarlas, ya que las operaciones sobre la imagen y la manipulación de los píxeles se lleva acabo haciendo operaciones sobre matrices.

En las imágenes digitales los valores de intensidad son cuantificados en códigos binarios de enteros positivos, la cuantificación consiste en la conversión de la amplitud de una señal analógica a un valor finito y discreto, que representa un nivel de gris o intensidad [Sán02], proceso a través del cual se le asigna un valor discreto a la muestra de una señal continua. En la Figura 2 tomada de [GW18] se esboza la creación de una imagen digital.

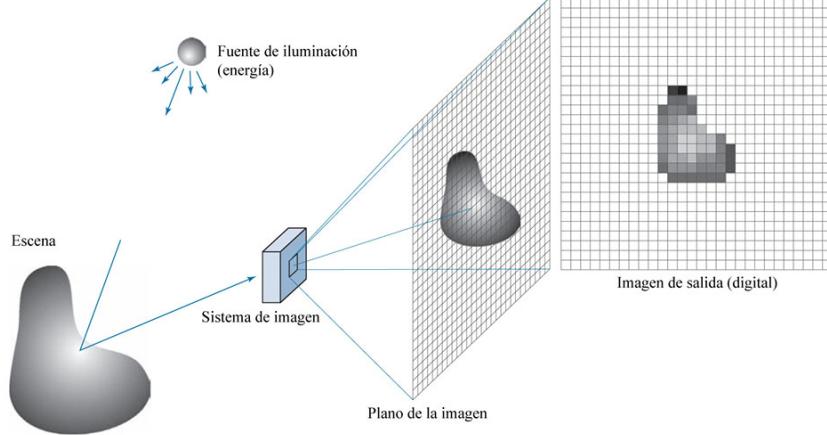


Figure 2: Ejemplo de adquisición de una imagen digital.

2.4 Convolución

La convolución es una operación de filtrado digital de señales, que combina dos señales correspondientes a la imagen original $f(t)$ y una función filtro $g(t)$. La convolución es una operación que representa la magnitud en que se superpone

una función $g(t)$ a medida que se desplaza sobre la función $f(t)$, y se define mediante una tercera función $f_c(t)$ de modo que para una señal en términos de una variable se tiene que

$$f_c(t) = f(t) * g(t) = \int_{-\infty}^{\infty} f(t)g(t-x)dx \quad (2)$$

Lo anterior, trasladado al procesamiento de imágenes, donde tenemos que la imagen de entrada es finita, discreta y representada por la función de intensidades en término de dos variables $f(i, j)$ y la función de filtrado $w(m, n)$ dada por la matriz:

$$w = \begin{bmatrix} w_{0,0} & w_{0,1} & \dots & w_{0,n-1} \\ w_{1,0} & w_{1,1} & \dots & w_{1,n-1} \\ \vdots & \vdots & & \vdots \\ w_{m-1,0} & w_{m-1,1} & \dots & w_{m-1,n-1} \end{bmatrix} \quad (3)$$

tenemos que la convolución discreta se define como:

$$f_c(i, j) = w(m, n) * f(i, j) = \sum_{m=-k}^k \sum_{n=-p}^p w(m, n)f(i-m, j-n) \quad (4)$$

donde w es la matriz deslizante de dimensiones $(2k + 1) \times (2p + 1)$, también llamada kernel o máscara de convolución, y las variables i y j corresponde al renglón y columna de la imagen original [PAA97].

la Figura 3 tomada de [Ran17] ejemplifica la operación de convolución aplicada a una ventana de una imagen y una máscara w de 3×3 .

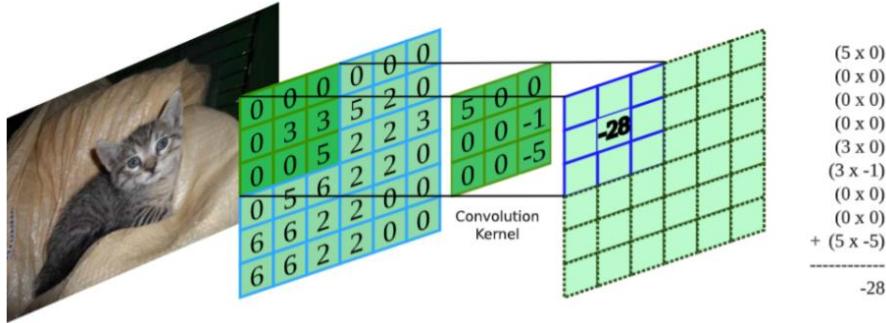


Figure 3: Ejemplo de la operación de convolución.

3 Metodología

La metodología propuesta en este trabajo constituye una serie de pasos para aplicar el filtrado lineal a las imágenes, se realizó la implementación del filtrado

gaussiano , laplca , low pass filter, sharpening filter.

3.1 Implementar Convolucion

Este es una parte importante, ya que se necesita la convolucion para realizar la aplicacion de kernel sobre la imagen original figura 4. Realizando el recorrido de la imagen y en cada pixel multiplicarla por cada valor del kernel , continuando con la suma de todas los productos , el resultado obtenido de la suma se se asigana en la nueva imagen y asi recorremos toda la imagen

```
function imgConvolucionada = convolucion2D(img,nneighbors,kernel)
    imgResize = padarray(img, [nneighbors, nneighbors], 0);
    imgConvolucionada = zeros(size(img));
    [row, col ] = size(imgResize);
    for r=nneighbors+1:row-nneighbors
        for c=nneighbors+1:col-nneighbors
            tempConvo = double(imgResize(r-nneighbors:r+nneighbors,c-nneighbors:c+nneighbors)) .* kernel;
            imgConvolucionada(r-nneighbors,c-nneighbors) = sum(tempConvo(:));
        end
    end
end
```

Figure 4: Codigo Convolucion

3.2 Valores de Kernel

Para Gauss se utiliza el kernel 5 Para el kernel de Laplace se utiliza 6

0.0037	0.0147	0.0256	0.0147	0.0037
0.0147	0.0586	0.0916	0.0586	0.0147
0.0256	0.0952	0.1502	0.0952	0.0256
0.0147	0.0586	0.0952	0.0586	0.0147
0.0037	0.0147	0.0256	0.0147	0.0037

Figure 5: Gauss

4 Implementacion de Low Pass Filter

Para el filtro Low pass Filter se realiza la sumatoria de cos de ambos ejes x y y, para despues realizar el producto punto. El cual lo podemos visualiza de la siguiente forma 8

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

Figure 6: Laplace

```
function kernel = LowPassFilter()
    [x , y] = meshgrid([-2 : 0.9 : 2]);
    fx = (1 + 2 * cos(x) + 2 * cos(2*x));
    fy = 1 + 2 * cos(y) + 2 * cos(2*y);
    f = fx .* fy;
    kernel = f/sum(sum(f));
end
```

Figure 7: Low pass filter

5 Implementacion de Sharpening Filter

Se realiza la operacion para cada uno de los ejes $fx = 2 - (1/2 * \cos(x))$; $fy = 1/2 * \cos(y)$ y se realiza la diferencia de estas para que nos de la visualizacion del kernel figura 12

6 Resultados

En la figura 14 podemos ver como el filtro gauss vuelve mi imagen original ligeramente borrosa y el filtro Laplace encuentra los bordes de la imagen, en este caso se aplicaron los kernels 10 y 13.

En la figura 15 aplicamos los filtros Low pass y Sharpening, en donde dependiendo de kernel se vera el cambio agresivo como podemos observar en la figura 16 en donde se aplica un kernel 9 en donde le da menos fuerza a los valores de alrededor.

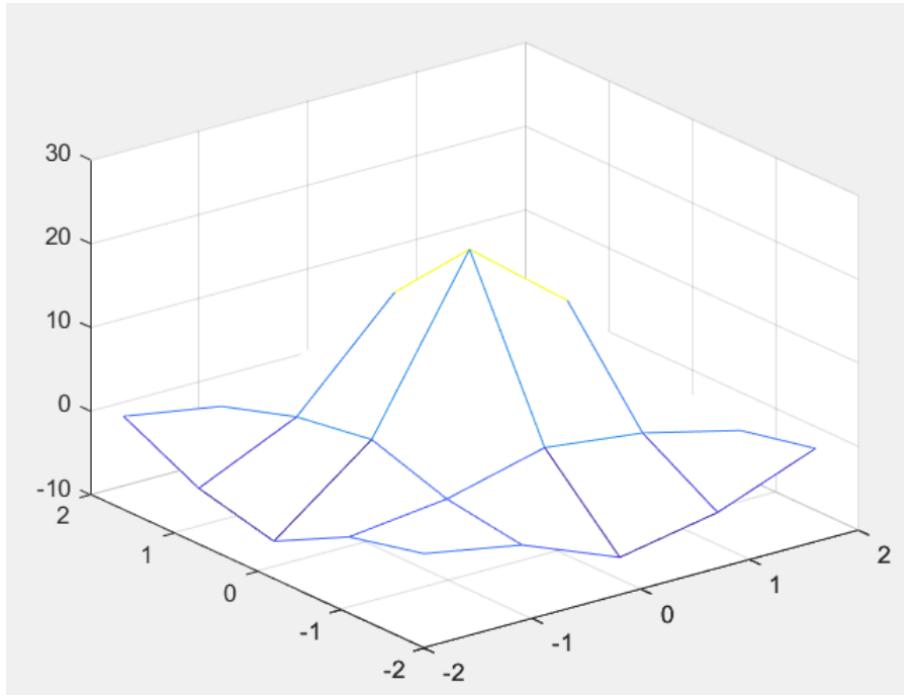


Figure 8: Low pass filter kernel

7 Conclusion

La aplicación de una variedad de filtros como los filtros lineales, Gaussianos, Laplacianos, así como los filtros de low pass y de realce (sharpening), resalta la versatilidad y la importancia de la manipulación de imágenes en el procesamiento digital. Estas técnicas permiten mejorar la calidad visual de las imágenes, destacar características específicas, eliminar ruido, suavizar imperfecciones y realzar detalles relevantes. Desde la reducción de la distorsión hasta la detección precisa de bordes, la combinación estratégica de estos filtros ofrece un amplio abanico de posibilidades para adaptarse a una variedad de necesidades en el procesamiento de imágenes.

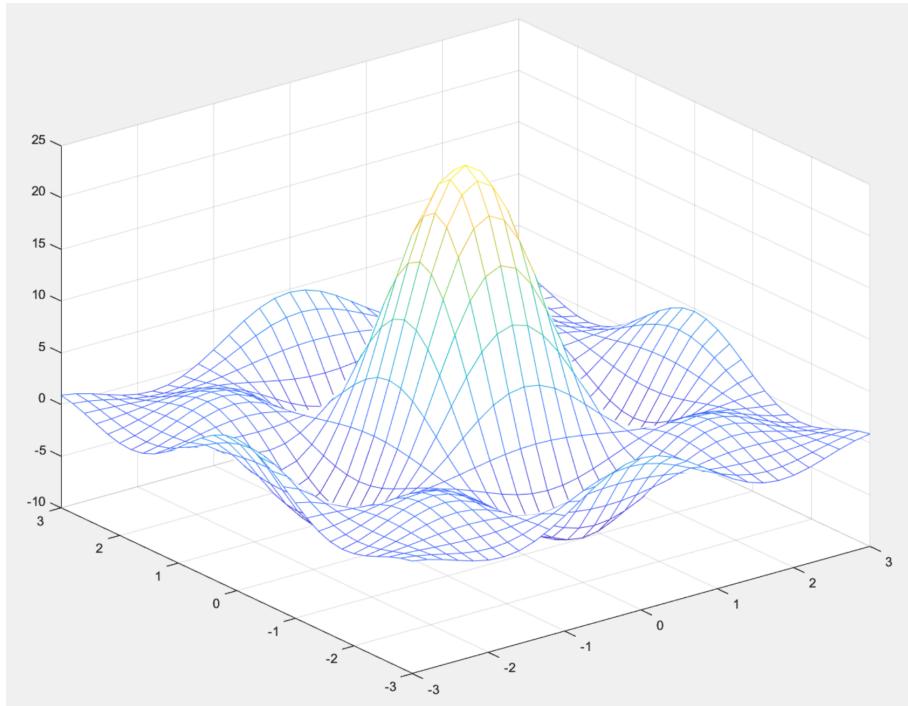


Figure 9: Low pass filter kernel version 2

0.0337	-0.0216	-0.1420	-0.0849	0.0312
-0.0216	0.0138	0.0910	0.0544	-0.0200
-0.1420	0.0910	0.5985	0.3576	-0.1315
-0.0849	0.0544	0.3576	0.2137	-0.0786
0.0312	-0.0200	-0.1315	-0.0786	0.0289

Figure 10: Low pass filter kernel

```

function kernel = sharpeningFilter()
    [x , y] = meshgrid([-5 : 0.8 : 5]);
    fx = 2 - (1/2 * cos(x));
    fy = 1/2 * cos(y);
    f = fx - fy;
    kernel = f / sum(sum(f));
end

```

Figure 11: Sharpening

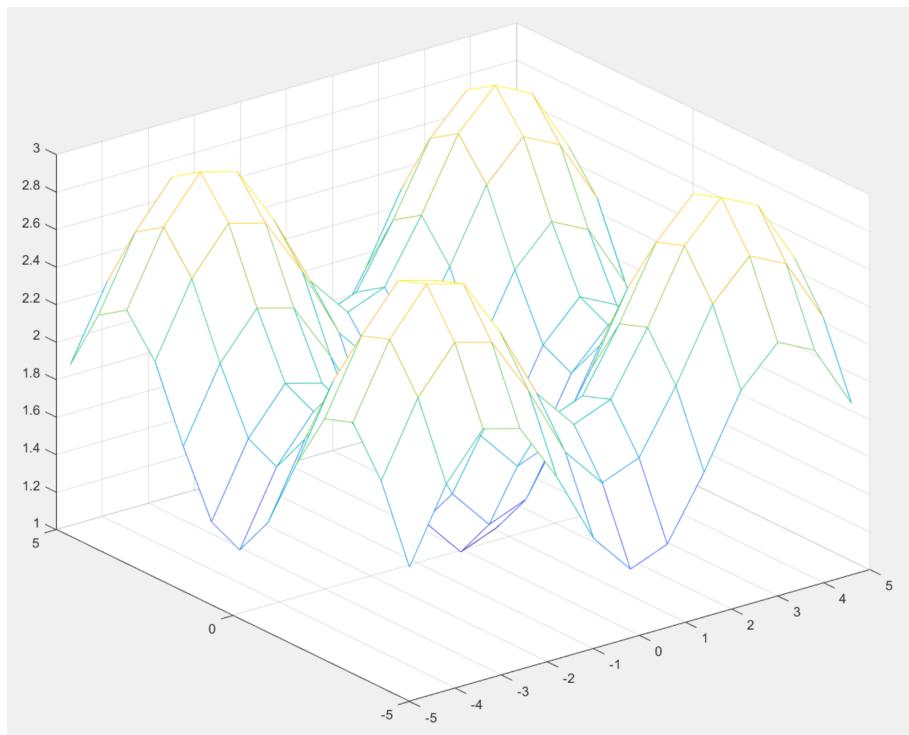


Figure 12: filter Shapering

0.0047	0.0057	0.0064	0.0062	0.0054	0.0043	0.0037	0.0039	0.0048	0.0059	0.0064	0.0061	0.0052
0.0057	0.0068	0.0074	0.0073	0.0064	0.0054	0.0048	0.0050	0.0059	0.0069	0.0075	0.0072	0.0063
0.0064	0.0074	0.0081	0.0079	0.0071	0.0060	0.0054	0.0056	0.0065	0.0076	0.0081	0.0078	0.0069
0.0062	0.0073	0.0079	0.0078	0.0069	0.0059	0.0053	0.0055	0.0064	0.0074	0.0080	0.0077	0.0068
0.0054	0.0064	0.0071	0.0069	0.0061	0.0050	0.0044	0.0046	0.0055	0.0066	0.0071	0.0068	0.0059
0.0043	0.0054	0.0060	0.0059	0.0050	0.0040	0.0034	0.0036	0.0045	0.0055	0.0061	0.0058	0.0049
0.0037	0.0048	0.0054	0.0053	0.0044	0.0034	0.0028	0.0030	0.0039	0.0049	0.0055	0.0052	0.0043
0.0039	0.0050	0.0056	0.0055	0.0046	0.0036	0.0030	0.0032	0.0041	0.0051	0.0057	0.0054	0.0045
0.0048	0.0059	0.0065	0.0064	0.0055	0.0045	0.0039	0.0041	0.0050	0.0060	0.0066	0.0063	0.0054
0.0059	0.0069	0.0076	0.0074	0.0066	0.0055	0.0049	0.0051	0.0060	0.0071	0.0076	0.0073	0.0064
0.0064	0.0075	0.0081	0.0080	0.0071	0.0061	0.0055	0.0057	0.0066	0.0076	0.0081	0.0079	0.0070
0.0061	0.0072	0.0078	0.0077	0.0068	0.0058	0.0052	0.0054	0.0063	0.0073	0.0079	0.0076	0.0067
0.0052	0.0063	0.0069	0.0068	0.0059	0.0049	0.0043	0.0045	0.0054	0.0064	0.0070	0.0067	0.0058

Figure 13: Shapering

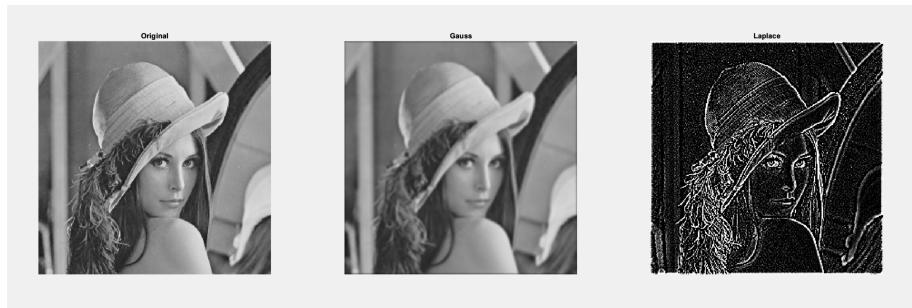


Figure 14: Resultados de filtro gauss y laplace



Figure 15: Resultados de filtro Low pass y Shapering



Figure 16: Resultados de filtro Low pass con una matriz mas grande