Casa abierta al tiempo
UNIVERSIDAD AUTÓNOMA METROPOLITANA
Unidad Cuajimalpa

PROYECTO FINAL
(Tercera revisión Comienza capitulo 5: pag 53)

# Transformaciones lineales y sus aplicaciones en los filtros de imágenes.

LICENCIATURA:
Ingeniería en Computación

UEA:
PROCESAMIENTO DE IMAGENES

AUTOR:
Luis Angel Sánchez Hernández

---

DOCENTE:
Dr. Saúl Zapotecas Martínez

27 de febrero de 2020

# Capítulo 1

# Introducción.

### Introducción.

Este proyecto estará enfocado solamente en hacer que lector entienda la teoría detrás de los filtros de imágenes, sin embargo, el conocimiento adquirido aquí podría tener muchas otras aplicaciones.

Para comprender del todo este proyecto es necesario tener ciertos conceptos de álgebra lineal, así que te introduciremos en ellos.

El procesado digital de imágenes PDI puede definirse como un conjunto de procedimientos que se realizan sobre una imagen, para su almacenamiento, transmisión o tratamiento. El PDI es una rama de la ingeniería de software nacida a partir de la unión y avance de las matemáticas con los computadores digitales. A raíz de esto, se empieza a usar el PDI en otras áreas interdisciplinares como la informática, ciencias básicas, ingeniería, biología, radiología, patología, geología, meteorología, oncología, entre otros. [Jácome et al., 2017]

El álgebra lineal es una rama de las matemáticas modernas que juega un papel central debido a que se encarga del estudio de conceptos tales como vectores, **matrices**, sistemas de ecuaciones lineales , espacios vectoriales y **transformaciones lineales**. En álgebra lineal, los conceptos son tan importantes como los cálculos, por lo que se convierte en un curso adecuado para introducir el pensamiento abstracto, debido a que una gran parte de su campo tiene una interpretación geométrica, que puede ayudar precisamente a visualizar esos conceptos.

Una matriz es un arreglo bidimencional de números, y en su mayor generalidad de elementos de un anillo. Se usan generalmente para describir:

- Sistemas de ecuaciones lineales.
- Sistemas de ecuaciones diferenciales.
- **Aplicaciones lineales.**
- Representar coeficientes de ecuaciones.

Nosotros vamos a tratar a las imágenes como matrices, como haremos esto se explicará más a detalle en el siguiente capítulo.

# Capítulo 2

# Transformaciones Lineales y filtros.

## 2.1. El álgebra lineal y el procesamiento digital de imágenes.

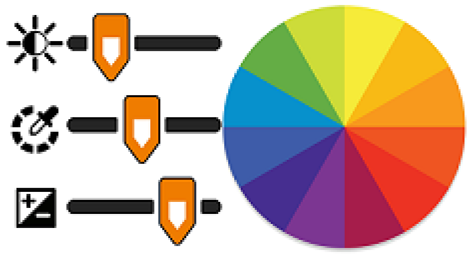
El procesamiento digital de imágenes ha adquirido, en años recientes, un papel importante en las tecnologías de la información y el cómputo. Actualmente, es la base de una creciente variedad de aplicaciones que incluyen diagnosis médica, percepción remota, exploración espacial, visión por computadora, etc. Como resultado directo de la reducción en el precio de las computadoras, el procesamiento digital de imágenes actualmente se puede efectuar (aunque con ciertas limitantes) en una computadora personal. [Torres, 1996]

Los conceptos de espacio vectorial y álgebra lineal resultan ser muy naturales en el procesamiento de señales. Y más aun en el procesamiento de imágenes digitales, ya que una imagen digital es una matriz. Las operaciones matriciales juegan un papel importante en el procesamiento de imágenes digitales. Una noción fundamental en el desarrollo de este estudio tiene que ver con que una imagen puede visualizarse de diferentes formas, y la manipulación de su representación es una de las más poderosas herramientas disponibles. Las manipulaciones útiles pueden ser lineales (ejemplo, las transformaciones) o no lineales (ejemplo, la cuantificación). [Melo, 2009]

Como ya se dijo anteriormente, una imagen puede ser representada como una matriz de píxeles, donde cada píxel se expresa como un vector tridimensional, compuesto por la cantidad de rojo, verde y azul del color (RGB). Hay dos tipos principales de tratamientos de una imagen:

- Cuando se cambia el color de cada píxel, utilizando una función que recibe como entrada el píxel original, o en casos más complejos, una submatriz de píxeles (por lo general, submatrices alrededor del pixel en la matriz, en función de un factor extra).
- Cuando los pixeles cambian su posición dentro de la imagen, o más precisamente, cuando cada píxel de la matriz se construye basado en otro pixel de la matriz, pero sin alterar su color.

Los procedimientos del procesamiento de imágenes del primer tipo son generalmente llamados filtros (Capitulo 3). En el segundo tipo se pueden mencionar: Rotación, volteo, escalado, sesgo y traslación.



## 2.2. ¿Qué son las transformaciones lineales?

Sean V y W dos espacios vectoriales sobre un campo F. Una transformación lineal de V en W es una función T de V en W, tal que T($cu$ + $v$) = $k$T($u$) + T($v$) para todos los vectores $u$ y $v$ de V y todos los escalares $k$ de un campo $F$. [Hoffman et al., 1973] [Roa-Fuentes and Oktaç, 2010]

En otras palabras, una transformación lineal es una función. Por ser función, tiene su dominio y su codominio, con la particularidad de que éstos son espacios vectoriales. Tenemos dos espacios vectoriales $V$ y $W$, y una función que va de $V$ a $W$. O sea una regla de asignación que transforma vectores de $V$ en vectores de $W$. Pero no toda función que transforme vectores de $V$ en vectores de $W$ es una transformación lineal. Debe cumplir ciertas condiciones:

$$T : V \to W \text{ es una transformación lineal si y sólo si:}$$

- $T(u + v) = f(u) + f(v) \forall u, v \in V$
- $T(k \cdot v) = k \cdot f(v) \forall v \in V, \forall k \in R$

## 2.3. Propiedades de las transformaciones lineales.

**Propiedad 1**

La imagen del vector nulo del dominio $0_v$ es el vector nulo del codominio $0_w$:

$$T(0_v) = 0_w$$

**Demostración:**

$$T(0_v) = T(0 \cdot v) = 0 \cdot T(v) = 0 \cdot w = 0_w$$

Donde hemos expresado a $0_v$ como el producto del escalar 0 por cualquier vector del espacio vectorial $V$, hemos usado la segunda condición que debe cumplir una transformación lineal, y finalmente hemos vuelto a usar la propiedad de espacios vectoriales sobre el producto del escalar 0 por cualquier vector.

**Propiedad 2**

La imagen del vector–$v$ es igual al opuesto de la imagen de $v$:

$$T(–v) = –T(v)$$

**Demostración:**

$$T(–v) = T(–1 \cdot v) = –1 \cdot T(v) = –T(v)$$

La justificación de los pasos dados en la demostración es similar a la anterior.

**Propiedad 3**

Consideremos $r$ vectores del espacio vectorial $V$:

$$v_1, v_2, \ldots, v_r \in V$$

Tomemos una combinación lineal en el dominio:

$$\alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 + ... + \alpha_r v_r$$

Donde $\alpha_i \in R$

Si aplicamos la transformación lineal F de V a W, teniendo en cuenta las propiedades enunciadas en la definición, resulta:

$$f(\alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 + ... + \alpha_r v_r) = \alpha_1 f(v_1) + \alpha_2 f(v_2) + \ldots \alpha_r f(v_r)$$

Es decir que una transformación lineal «transporta» combinaciones lineales de $V$ a $W$, conservando los escalares de la combinación lineal.

## 2.4. Filtros

Desde el punto de vista del álgebra lineal, los filtros se aplican a cada píxel de la matriz usando la función de filtro. Como se explicó anteriormente, la entrada de esta función puede ser sólo un pixel como el ajuste de brillo, o una submatriz de píxeles como el desenfoque, donde el orden de la submatriz dependerá del radio que se utilice. Consideremos la matriz M, como la matriz asociada a una imagen a color:

$$M = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Aquí, $x_{ij}$ es el píxel en la posición $(i, j)$, que se representa como el vector:

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

En el caso más sencillo (el filtro necesita solamente un píxel como entrada), la función puede ser un **transformación lineal**, que transforma un vector tridimensional (píxel) en otro vector tridimensional, o no.
Cuando se trata de una **transformación lineal**, la transformación se puede representar como una matriz T de orden 3x3, donde:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = T \cdot \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

# Capítulo 3

# Filtros Orientados al pixel.

## 3.1. Filtros Escenciales.

### 3.1.1. Escala de Grises.

Para hacer esto de forma más interactiva vamos a hacerlo real, tenemos una imagen de 455x512 pixeles, la misma que usaremos para aplicar todos los filtros. Por lo tanto, tenemos una matriz $M_{n*m}$ donde $n = 455$ y $m = 512$, cada elemento de la matriz es un vector de profundidad 3, el cual representa el pixel, es decir:

$$M_{n\cdot m} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} x_{(11\ ,\ \dots\ ,\ nm)} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

La transformación lineal $T$ que usaremos para aplicar el filtro es la siguiente:

$$T = (R + G + B)/3$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} (r+g+b)/3 \\ (r+g+b)/3 \\ (r+g+b)/3 \end{bmatrix}$$

Obteniendo lo siguiente:

(a) Original (b) Escala de grises

### 3.1.2. Negativo.

Ya que usaremos la misma imagen para hacer todos los filtros partiremos de que tenemos la matriz identidad

$$M_{n \cdot m} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} x_{(11\ ,\ \ldots\ ,\ nm)} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

a la cual vamos a aplicar la siguiente transformación lineal:

$$T = \lambda - x$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $\lambda$ es igual a 255.0, que se refiere a la intensidad de color que puede tener el pixel y $x$ es el pixel.

Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} \lambda - x_r \\ \lambda - x_g \\ \lambda - x_b \end{bmatrix}$$

Obteniendo lo siguiente:

(a) Original

(b) Negativo

## 3.2. Filtros de Aclarado.

### 3.2.1. Filtro logarítmico.

Partiremos de que tenemos la matriz identidad

$$M_{n\cdot m} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} x_{(11\ ,\ \ldots\ ,\ nm)} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

a la cual vamos a aplicar la siguiente transformación lineal:

$$T = Aln(\alpha x + 1),$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $\alpha > 1$ es dado por el usuario, $x$ es el pixel, $A = \lambda ln(\alpha\lambda + 1)$ y $\lambda$ vale 255.0

Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} Aln(\alpha x_r + 1) \\ Aln(\alpha x_g + 1) \\ Aln(\alpha x_b + 1) \end{bmatrix}$$

Obteniendo lo siguiente:

(a) Original (b) Filtro Logaritmico

### 3.2.2. Filtro seno.

Partiremos de que tenemos la matriz identidad

$$M_{n \cdot m} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} x_{(11\ ,\ \ldots\ ,\ nm)} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

a la cual vamos a aplicar la siguiente transformación lineal:

$$T = \lambda \sin(\tfrac{\pi x}{2\lambda})$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $x$ es el pixel, $\lambda$ vale 255.0 y $\pi$ es pi que equivale a 3,1416

Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} \lambda \sin(\frac{\pi x_r}{2\lambda}) \\ \lambda \sin(\frac{\pi x_g}{2\lambda}) \\ \lambda \sin(\frac{\pi x_g}{2\lambda}) \end{bmatrix}$$

Obteniendo lo siguiente:

(a) Original

(b) Filtro seno

### 3.2.3. Filtro exponencial.

Partiremos de que tenemos la matriz identidad

$$M_{n \cdot m} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} x_{(11\ ,\ \dots\ ,\ nm)} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

a la cual vamos a aplicar la siguiente transformación lineal:

$$T = A(1 - e^{-1(\frac{x}{\lambda})})$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

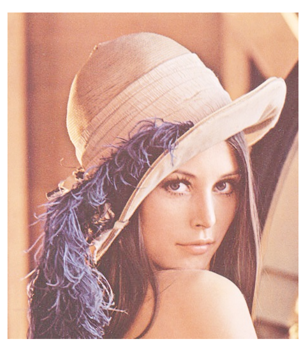
Donde: $x$ es el pixel, $A = \frac{\lambda}{1-e^{-\alpha}}$, $\lambda$ vale 255.0, $e$ equivale a 2.71828 y $\alpha > 1$ dado por el usuario.

Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} A(1 - e^{-1(\frac{x_r}{\lambda})}) \\ A(1 - e^{-1(\frac{x_g}{\lambda})}) \\ A(1 - e^{-1(\frac{x_b}{\lambda})}) \end{bmatrix}$$

Obteniendo lo siguiente:

(a) Original



(b) Filtro exponencial

## 3.3. Filtros de Oscurecimiento.

### 3.3.1. Filtro coseno.

Partiremos de que tenemos la matriz identidad

$$M_{n \cdot m} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} x_{(11\ ,\ ...\ ,\ nm)} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

a la cual vamos a aplicar la siguiente transformación lineal:

$$T = \lambda(1 - \cos(\tfrac{\pi x}{2\lambda}))$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $x$ es el pixel, $\lambda$ vale 255.0 y $\pi$ es pi que equivale a 3,1416

Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} \lambda(1 - \cos(\frac{\pi x_r}{2\lambda})) \\ \lambda(1 - \cos(\frac{\pi x_g}{2\lambda})) \\ \lambda(1 - \cos(\frac{\pi x_b}{2\lambda})) \end{bmatrix}$$

Obteniendo lo siguiente:

(a) Original

(b) Filtro coseno

### 3.3.2. Filtro exponencial creciente.

Partiremos de que tenemos la matriz identidad

$$M_{n \cdot m} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} x_{(11\ ,\ \dots\ ,\ nm)} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

a la cual vamos a aplicar la siguiente transformación lineal:

$$T = A(e^{\alpha(\frac{x}{\lambda})} - 1)$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

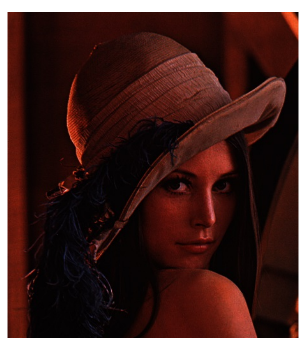
Donde: $x$ es el pixel, $A = \frac{\lambda}{e^{\alpha}-1}$, $\lambda$ vale 255.0, $e$ equivale a 2.71828 y $\alpha > 0$ dado por el usuario.

Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} A(e^{\alpha(\frac{x_r}{\lambda})} - 1) \\ A(e^{\alpha(\frac{x_g}{\lambda})} - 1) \\ A(e^{\alpha(\frac{x_b}{\lambda})} - 1) \end{bmatrix}$$

Obteniendo lo siguiente:

(a) Original



(b) Filtro exponencial creciente

## 3.4. Filtros de Aclarado/Oscurecimiento.

### 3.4.1. Filtro de ajuste lineal.

Partiremos de que tenemos la matriz identidad

$$M_{n\cdot m} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} x_{(11\ ,\ \ldots\ ,\ nm)} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

a la cual vamos a aplicar la siguiente transformación lineal:

$$T = \begin{cases} min(\lambda, x+\alpha) & aclaramiento \\ max(0, x-\alpha) & oscurecimiento \end{cases}$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $x$ es el pixel, $\lambda$ vale 255.0.
Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente dependiendo la decisión del usuario:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} min(\lambda, x_r+\alpha) \\ min(\lambda, x_g+\alpha) \\ min(\lambda, x_b+\alpha) \end{bmatrix} \text{ ó } \begin{bmatrix} max(\lambda, x_r-\alpha) \\ max(\lambda, x_g-\alpha) \\ max(\lambda, x_b-\alpha) \end{bmatrix}$$

Obteniendo lo siguiente para el aclarado:

(a) Original

(b) Ajuste lineal aclarado

y lo siguiente para el oscurecimiento:

(a) Original (b) Ajuste lineal oscurecimiento

### 3.4.2. Filtro Corrección gamma.

Partiremos de que tenemos la matriz identidad

$$M_{n \cdot m} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} x_{(11\ ,\ \dots\ ,\ nm)} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

a la cual vamos a aplicar la siguiente transformación lineal:

$$T = \lambda(\tfrac{x}{\lambda})^{\gamma}$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $x$ es el pixel, $\lambda$ vale 255.0 y $\gamma$ es dado por el usuario, para valores $0 < \gamma < 1$ aclarará y valores $\gamma > 1$ oscurecerá

Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} min(\lambda, x_r + \alpha) \\ min(\lambda, x_g + \alpha) \\ min(\lambda, x_b + \alpha) \end{bmatrix}$$

Obteniendo lo siguiente para el aclarado:

(a) Original

(b) Corrección Gamma Aclarado

y lo siguiente para el oscurecimiento:

(a) Original

(b) Corrección Gamma Oscurecimiento

## 3.5. Filtros de Alto Contraste.

### 3.5.1. Filtro sigmoide senoidal.

Partiremos de que tenemos la matriz identidad

$$M_{n \cdot m} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} x_{(11\ ,\ \ldots\ ,\ nm)} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

a la cual vamos a aplicar la siguiente transformación lineal:

$$T = x - \alpha \sin(\tfrac{2\pi x}{\lambda})$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $x$ es el pixel, $\lambda$ vale 255.0, $\pi$ equivale a 3,1416 y $\alpha$ dado por el usuario. Esto significa que de forma interna, a cada uno de los pixeles $(0 < \alpha < 40)$ se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} x - \alpha \sin(\frac{2\pi x_r}{\lambda}) \\ x - \alpha \sin(\frac{2\pi x_g}{\lambda}) \\ x - \alpha \sin(\frac{2\pi x_b}{\lambda}) \end{bmatrix}$$

Obteniendo lo siguiente:

(a) Original (b) Filtro sigmoide senoidal

### 3.5.2. Filtro sigmoide tangente hiperbólica.

Partiremos de que tenemos la matriz identidad

$$M_{n\cdot m} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} x_{(11\ ,\ \dots\ ,\ nm)} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

a la cual vamos a aplicar la siguiente transformación lineal:

$$T = \frac{\lambda}{2}[1 + \tanh[\alpha(x - \frac{\lambda}{2})]]$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $x$ es el pixel, $\lambda$ vale 255.0 y $\alpha$ dado por el usuario. $\alpha > 0$
Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} \frac{\lambda}{2}[1 + \tanh[\alpha(x_r - \frac{\lambda}{2})]] \\ \frac{\lambda}{2}[1 + \tanh[\alpha(x_g - \frac{\lambda}{2})]] \\ \frac{\lambda}{2}[1 + \tanh[\alpha(x_b - \frac{\lambda}{2})]] \end{bmatrix}$$

Obteniendo lo siguiente:

(a) Original (b) Filtro sigmoide tangente hiperbolica

## 3.6. Filtros de Bajo Contraste.

### 3.6.1. Filtro sigmoide senoidal para bajo contraste.

Partiremos de que tenemos la matriz identidad

$$M_{n\cdot m} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} x_{(11\ ,\ \ldots\ ,\ nm)} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

a la cual vamos a aplicar la siguiente transformación lineal:

$$T = x + \alpha \sin(\tfrac{2\pi x}{\lambda})$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $x$ es el pixel, $\lambda$ vale 255.0, $\pi$ equivale a 3,1416 y $\alpha$ dado por el usuario. Esto significa que de forma interna, a cada uno de los pixeles $(0 < \alpha < 40)$ se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} x + \alpha \sin(\frac{2\pi x_r}{\lambda}) \\ x + \alpha \sin(\frac{2\pi x_g}{\lambda}) \\ x + \alpha \sin(\frac{2\pi x_b}{\lambda}) \end{bmatrix}$$

Obteniendo lo siguiente:

(a) Original

(b) Filtro sigmoide senoidal para bajo contraste

### 3.6.2. Filtro sigmoide polinomial cúbico.

Partiremos de que tenemos la matriz identidad

$$M_{n \cdot m} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} x_{(11\ ,\ \dots\ ,\ nm)} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

a la cual vamos a aplicar la siguiente transformación lineal:

$$T = \frac{\lambda}{2} + \frac{4}{\lambda^2}(x - \frac{\lambda}{2})^3$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $x$ es el pixel, $\lambda$ vale 255.0

Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} \frac{\lambda}{2} + \frac{4}{\lambda^2}(x_r - \frac{\lambda}{2})^3 \\ \frac{\lambda}{2} + \frac{4}{\lambda^2}(x_g - \frac{\lambda}{2})^3 \\ \frac{\lambda}{2} + \frac{4}{\lambda^2}(x_b - \frac{\lambda}{2})^3 \end{bmatrix}$$

Obteniendo lo siguiente:

(a) Original

(b) Filtro sigmoide polinomial cubico

## 3.7. Filtros en Escala de Grises.

### 3.7.1. Reducción de grises.

Partiremos con la matriz en escala de grises (primer filtro), en este caso esta es la identidad. A la cual vamos a aplicar la siguiente transformación lineal:

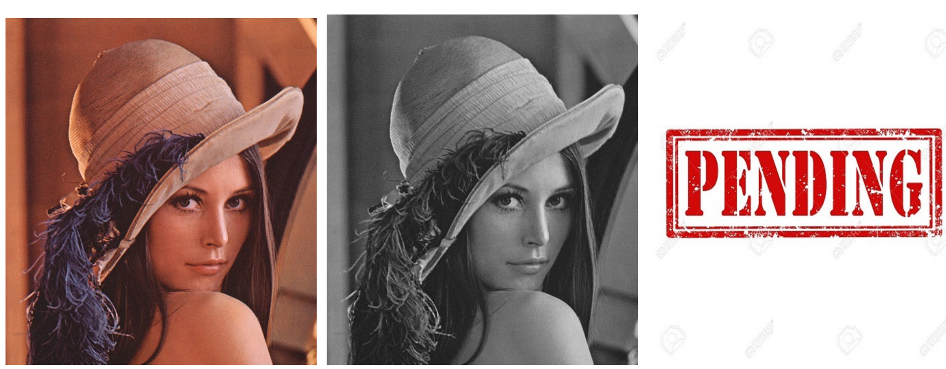
$$T = \begin{cases} 0 & \text{si } x \leq \alpha_1 \\ q_1 & \text{si } \alpha_1 < x \leq \alpha_2 \\ q_2 & \text{si } \alpha_2 < x \leq \alpha_3 \\ \vdots & \vdots \\ q_n & \text{si } \alpha_{n-1} < x \leq \lambda \end{cases}$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $x$ es el pixel, $\lambda$ vale 255.0 y $\alpha$ es dado por el usuario
Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} Pendiente \\ Pendiente \\ Pendiente \end{bmatrix}$$

Obteniendo lo siguiente:




(a) Original (b) Escala de grises (c) Reducción de grises

### 3.7.2. Filtro de binarización por umbral.

Partiremos con la matriz en escala de grises (primer filtro), en este caso esta es la identidad. A la cual vamos a aplicar la siguiente transformación lineal:

$$T = \begin{cases} 0 & \text{si } x \leq \alpha \\ \lambda & \text{en otro caso} \end{cases}$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $x$ es el pixel, $\lambda$ vale 255.0 y $\alpha$ es dado por el usuario

Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ ó } \begin{bmatrix} \lambda \\ \lambda \\ \lambda \end{bmatrix}$$

Obteniendo lo siguiente:

(a) Original

(b) Escala de grises

(c) Filtro binarización por umbral

### 3.7.3. Filtro de binarización por intensidad de una imagen.

Partiremos con la matriz en escala de grises (primer filtro), en este caso esta es la identidad. A la cual vamos a aplicar la siguiente transformación lineal:

$$T = \begin{cases} 0 & \text{si } x \leq I \\ \lambda & \text{en otro caso} \end{cases}$$

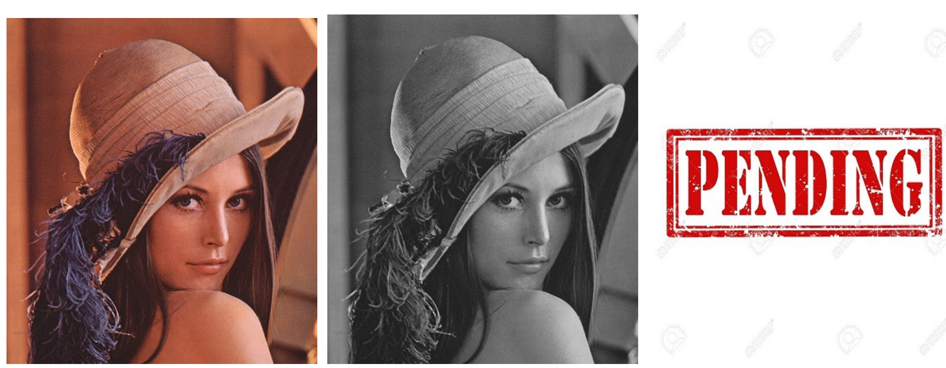
$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $x$ es el pixel, $\lambda$ vale 255.0 e $I = \frac{1}{m*n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} x_{ij}$ $m$ y $n$ siendo el alto y ancho de la imágen respectivamente

Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ ó } \begin{bmatrix} \lambda \\ \lambda \\ \lambda \end{bmatrix}$$

Obteniendo lo siguiente:




(a) Original (b) Escala de grises (c) Filtro binarización por intensidad de la imagen

### 3.7.4. Filtro de intervalo de umbral binario.

Partiremos con la matriz en escala de grises (primer filtro), en este caso esta es la identidad. A la cual vamos a aplicar la siguiente transformación lineal:

$$T = \begin{cases} \lambda & \text{si } x \leq \alpha \text{ ó } x \geq \beta \\ 0 & \text{en otro caso} \end{cases}$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $x$ es el pixel, $\lambda$ vale 255.0 con $\alpha$ y $\beta$ dados por el usuario $(0 < \alpha < \beta < \lambda)$.

Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} \lambda \\ \lambda \\ \lambda \end{bmatrix} \text{ ó } \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Obteniendo lo siguiente:

(a) Original

(b) Escala de grises

(c) Filtro de intervalo de umbral binario

### 3.7.5. Filtro de intervalo de umbral binario invertido.

Partiremos con la matriz en escala de grises (primer filtro), en este caso esta es la identidad. A la cual vamos a aplicar la siguiente transformación lineal:

$$T = \begin{cases} 0 & \text{si } x \leq \alpha \text{ ó } x \geq \beta \\ \lambda & \text{en otro caso} \end{cases}$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $x$ es el pixel, $\lambda$ vale 255.0 con $\alpha$ y $\beta$ dados por el usuario ($0 < \alpha < \beta < \lambda$).

Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ ó } \begin{bmatrix} \lambda \\ \lambda \\ \lambda \end{bmatrix}$$

Obteniendo lo siguiente:

(a) Original (b) Escala de grises (c) Filtro de intervalo de umbral binario invertido

### 3.7.6. Filtro de umbral de la escala de grises.

Partiremos con la matriz en escala de grises (primer filtro), en este caso esta es la identidad. A la cual vamos a aplicar la siguiente transformación lineal:

$$T = \begin{cases} \lambda & \text{si } x \leq \alpha \text{ ó } x \geq \beta \\ x & \text{en otro caso} \end{cases}$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $x$ es el pixel, $\lambda$ vale 255.0 con $\alpha$ y $\beta$ dados por el usuario ($0 < \alpha < \beta < \lambda$).

Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} \lambda \\ \lambda \\ \lambda \end{bmatrix} \text{ ó } \begin{bmatrix} x_r \\ x_g \\ x_b \end{bmatrix}$$

Obteniendo lo siguiente:

(a) Original (b) Escala de grises (c) Filtro de umbral de la escala de grises

### 3.7.7. Filtro de umbral de la escala de grises invertido.

Partiremos con la matriz en escala de grises (primer filtro), en este caso esta es la identidad. A la cual vamos a aplicar la siguiente transformación lineal:

$$T = \begin{cases} 0 & \text{si } x \leq \alpha \text{ ó } x \geq \beta \\ x & \text{en otro caso} \end{cases}$$

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Donde: $x$ es el pixel, $\lambda$ vale 255.0 con $\alpha$ y $\beta$ dados por el usuario ($0 < \alpha < \beta < \lambda$).

Esto significa que de forma interna, a cada uno de los pixeles se les asignará lo siguiente:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ ó } \begin{bmatrix} x_r \\ x_g \\ x_b \end{bmatrix}$$

Obteniendo lo siguiente:

(a) Original (b) Escala de grises (c) Filtro de umbral de la escala de grises invertido

# Capítulo 4

# Filtros Orientados a la región/vecindad.

### 4.0.1. Teoría.

Las operaciones orientadas a la región transforman a la imagen modificando un pixel a la vez y toman en cuenta para dicha transformación los pixeles vecinos. Y como es natural la transformación se puede aplicar a toda la imagen o a una región de ella.
Diremos que los píxeles vecinos de primer orden son aquellos contiguos a él, en una retícula cartesiana regular un pixel, digamos aquel ubicado en la coordenada (i, j) tiene 8 primeros vecinos, si denotamos por x[i, j] al pixel. [Ortiz, 2013]

$$\begin{bmatrix} x[i-1][j-1] & x[i][j-1] & x[i+1][j-1] \\ x[i-1][j] & \mathbf{x\ [i][j]} & x[i+1][j] \\ x[i-1][j+1] & x[i][j+1] & x[i+1][j+1] \end{bmatrix}$$

Para poder aplicar este tipo de filtros es necesario mencionar los siguientes conceptos:
**Producto Hadamar**
Sean $A$ y $B$ matrices en $R^{m*n}$. El producto Hadamar de $A$ con $B$ (denotado $A*B$) está definido por:

$$\begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{m1} & h_{m2} & \cdots & h_{mn} \end{bmatrix}$$

[Lafuente, 2009]

**Convolución**
Sea $A$ y $B$ dos matrices rn $R_{m*n}$ con elementos $a_{ij}$ y $b_{ij}$ respectivamente.
La operación convolución $A*B$ está definida por:

$$A * B = \sum_i^m \sum_j^n h_{ij}$$

Donde $h_{ij}$ es el elemento $i, j$ de la matriz Hadamar de $A * B$ (i.e, $H = A * B$)
En otras palabras:

$$A * B = \sum_{i=1}^m \sum_{j=1}^n a_{ij} * b_{ij} = z$$

[Elizondo and Maestre, 2002]

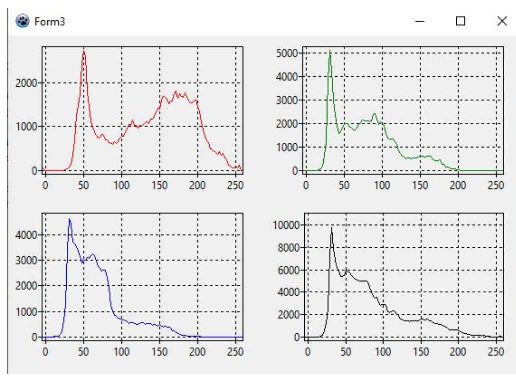
### 4.0.2. Histogramas.

El histograma de una imágen cuantifica la intensidad de cada pixel por cada canal en una imágen.

$$H_r = H_r[i] = \sum_{i=1}^{\lambda} p(x,y) \epsilon I / p(x,y) = i$$
$$H_g = H_g[i] = \sum_{i=1}^{\lambda} p(x,y) \epsilon I / p(x,y) = i$$
$$H_b = H_b[i] = \sum_{i=1}^{\lambda} p(x,y) \epsilon I / p(x,y) = i$$

Por ejemplo, para la siguiente imágen obtendremos los histogramas correspondientes en cada canal (R, G, B) y al mismo tiempo escala de grises.

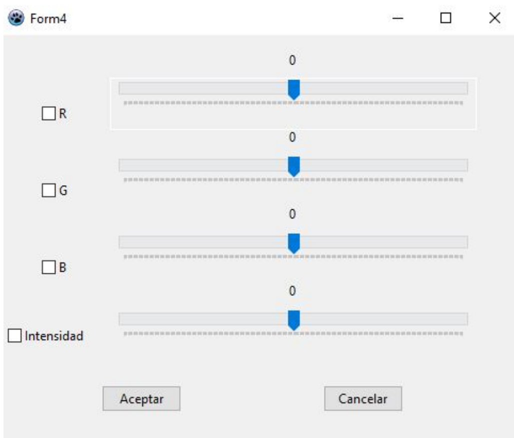
(a) Lena




(b) Histogramas

### 4.0.3. Ecualizador.

Ecualización es la sintonización de intención en cada uno de los canales de la imágen.




(a) Ecualizador

De tal manera que:

$$p_r(x,y) = p_r(x,y) + \alpha, \text{ satura ó disminuye el canal R}$$
$$p_g(x,y) = p_g(x,y) + \alpha, \text{ satura ó disminuye el canal G}$$
$$p_b(x,y) = p_b(x,y) + \alpha, \text{ satura ó disminuye el canal B}$$

Saturar o disminuir el canal dependerá del valor de $\alpha$, ya que $-127 < \alpha < 127$ es decir, si $\alpha$ es negativo disminuye la intensidad del canal, pero si $\alpha$ es positivo satura el canal.

### 4.0.4. Filtros Pasa Baja (Low Pass) para suavizado.

Los filtros pasa baja (Low Pass) asemejan el nivel de intensidad del pixel con el entorno. Además de reducir el ruido aumenta la homogenidad en la intensidad. Sin embarho se pierden los detalles (bordes se difuminan).

#### Media Aritmética.

El modelo más simple corresponde a la media aritmética., este considera la media de los pixeles en un entorno de 3x3 centrado en un pixel (x , y). Para aplicar esté filtro tenemos una transformación lineal dada por:

$$T = \frac{1}{k^2} * \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

Ésta transformación se lleva a cabo mediante convolución, es decir, $T$ es la máscara que aplicará el filtro a regiones $K * K$ (sub matrices) de la matriz original (imágen).

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1k} \\ x'_{21} & x'_{22} & \cdots & x'_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{k1} & x'_{k2} & \cdots & x'_{kk} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k1} & x_{k2} & \cdots & x_{kk} \end{pmatrix}$$

Obteniendo lo siguiente:

(a) Original

(b) 3x3

(c) 5x5

(d) 7x7

### Media Ponderada.

Para aplicar esté filtro tenemos una transformación lineal dada por:

$$T = \frac{1}{(k^2-1)+p} * \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

$P = 1$ (Media aritmética)
$P = 2$ (LP1, Low Pass1)
$P = 4$ (LP2, Low Pass2)
$P = 12$ (LP3, Low Pass3)

Ésta transformación se lleva a cabo mediante convolución, es decir, $T$ es la máscara que aplicará el filtro a regiones $K*K$ (sub matrices) de la matriz original (imágen).

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1k} \\ x'_{21} & x'_{22} & \cdots & x'_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{k1} & x'_{k2} & \cdots & x'_{kk} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k1} & x_{k2} & \cdots & x_{kk} \end{pmatrix}$$

Obteniendo lo siguiente:

(a) Original (b) 3x3 P=12 (c) 5x5 P=12 (d) 7x7 P=12

### Media Gaussiana.

**Gaussiana Fuerte.**

Para aplicar esté filtro tenemos una transformación lineal dada por:

$$T = \frac{1}{2\pi s^2} e^{\left(-\frac{x^2+y^2}{2s^2}\right)} * \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

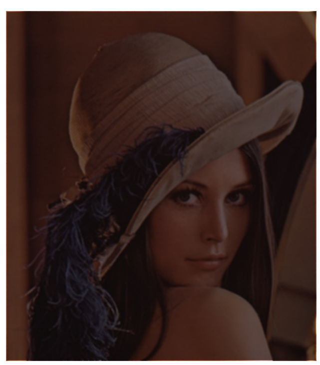
Donde: $s^2 = 2{,}5464$ y $(x, y)$ se refieren al pixel donde nos encontramos en ese momento.

Ésta transformación se lleva a cabo mediante convolución, es decir, $T$ es la máscara que aplicará el filtro a regiones $K*K$ (sub matrices) de la matriz original (imágen).

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1k} \\ x'_{21} & x'_{22} & \cdots & x'_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{k1} & x'_{k2} & \cdots & x'_{kk} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k1} & x_{k2} & \cdots & x_{kk} \end{pmatrix}$$

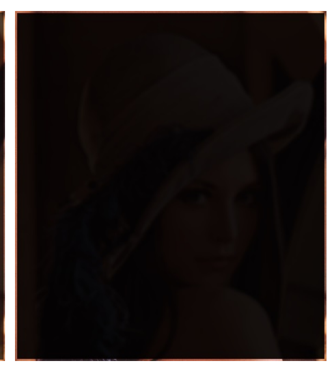
Obteniendo lo siguiente:

(a) Original



(b) 3x3

(c) 5x5



(d) 7x7

**Gaussiana Débil.**

Para aplicar esté filtro tenemos una transformación lineal dada por:

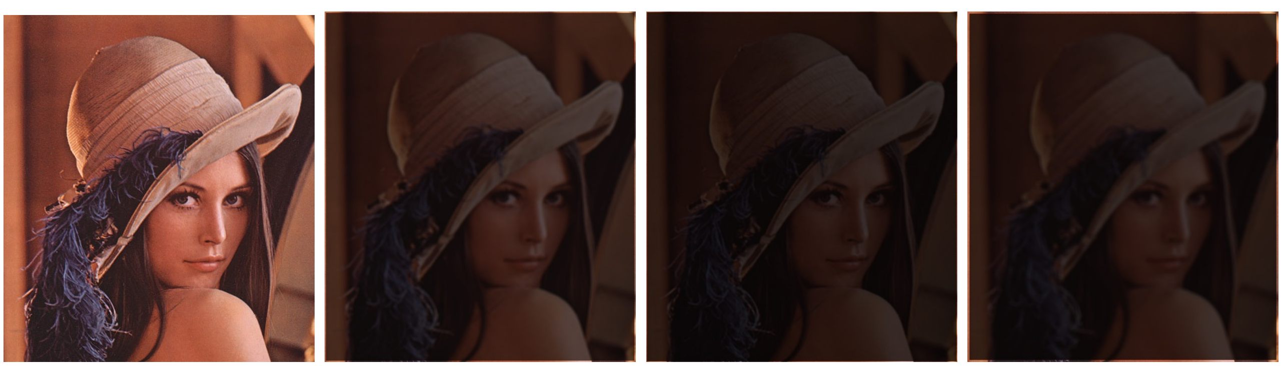
$$T = \frac{1}{2\pi s^2} e^{\left(-\frac{x^2+y^2}{2s^2}\right)} * \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

Donde: $s^2 = 5{,}0929$ y $(x, y)$ se refieren al pixel donde nos encontramos en ese momento.

Ésta transformación se lleva a cabo mediante convolución, es decir, $T$ es la máscara que aplicará el filtro a regiones $K*K$ (sub matrices) de la matriz original (imágen).

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1k} \\ x'_{21} & x'_{22} & \cdots & x'_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{k1} & x'_{k2} & \cdots & x'_{kk} \end{pmatrix} = T \cdot \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k1} & x_{k2} & \cdots & x_{kk} \end{pmatrix}$$

Obteniendo lo siguiente:



(a) Original (b) 3x3 (c) 5x5 (d) 7x7

### Media Geométrica.

Una manera de calcular el promedio, consiste en utulizar una norma cuadrática. La media geométrica para un conjunto de datos: Para aplicar esté filtro tenemos una transformación lineal dada por:

$$X = X_1, X_2, X_3, ..., X_n$$
$$< X >_G = \sqrt[n]{X_1 * X_2 * X_3 * ... * X_n}$$

Ésta transformación se lleva a cabo mediante la norma cuadrática ($< X >_G$), es decir, se calculará la norma cuadrática a regiones $K*K$ (sub matrices) de la matriz original (imágen).

$$Imagen[\tfrac{k}{2}][\tfrac{k}{2}] = < X >_G \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k1} & x_{k2} & \cdots & x_{kk} \end{pmatrix}$$

Siendo $\frac{k}{2}$ división entera.

Obteniendo lo siguiente:

(a) Original

(b) 3x3

(c) 5x5

(d) 7x7

### Promedio Direccional.

Para aplicar esté filtro tenemos una transformación lineal dada por:

$P_1 = (I[i-1][j-1] + I[i][j] + I[i+1][j+1])/3{,}0$
$P_2 = (I[i][j-1] + I[i][j] + I[i][j+1])/3{,}0$
$P_3 = (I[i-1][j+1] + I[i][j] + I[i+1][j+1])/3{,}0$
$P_4 = (I[i-1][j] + I[i][j] + I[i+1][j])/3{,}0$

$$PD = max(P1, P2, P3, P4)$$

Ésta transformación se lleva a cabo mediante el cálculo de los promedios en cruz y diagonales, es decir, $PD$ es la máscara que aplicará el filtro a regiones $K * K$ (sub matrices) de la matriz original (imágen).

$$Imagen[\tfrac{k}{2}][\tfrac{k}{2}] = PD \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k1} & x_{k2} & \cdots & x_{kk} \end{pmatrix}$$

Siendo $\frac{k}{2}$ división entera.

Obteniendo lo siguiente:


(a) Original


(b) 3x3


(c) 5x5


(d) 7x7

### 4.0.5. Filtros Pasa Alta (High Pass) para detección de bordes.

La detección de contornos es un paso intermedio en el reconocimiento de patrones en imágenes digitales. En una imagen, los contornos corresponden a los limites de los objetos presentes en la imagen. Para hallar los contornos se buscan los lugares en la imagen en los que la intensidad del píxel cambia rápidamente, generalmente usando alguno de los siguientes criterios:

- Lugares donde la primera derivada (gradiente) de la intensidad es de magnitud mayor que la de un umbral predefinido.
- Lugares donde la segunda derivada (laplaciano) de la intensidad tiene un cruce por cero.

[Elizondo and Maestre, 2002]

Los filtros paso alto (High Pass) enfatizan diferencias en el nivel de intensidad, aumentan el ruido y estimula los límites o bordes, e.g Filtro Gaussiano ó Laplaciano. La detección de bordes enfatiza los bordes que rodean a un objeto en una imágen. El gradiente (en términos de imágenes digitales), calcula las variaciones del nivel de intensidad en pixeles consecutivas.

$$\frac{df}{dx} = \frac{f(x+1,y)-f(x,y)}{dx}$$
$$\frac{df}{dy} = \frac{f(x,y+1)-f(x,y)}{dy}$$

Por lo tanto, el gradiente está dado por:

$$\nabla f \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix}$$

**Magnitud P-ésima del vector gradiente.**

$$||x||_p = \sqrt[p]{x_1^p + x_2^p + ... + x_n^p}$$
$$Mag(x,y) = ||\nabla f \begin{bmatrix} G_x \\ G_y \end{bmatrix} || = \sqrt[p]{x_1^p + x_2^p + ... + x_n^p}$$

Usualmente $p = 2$

$$Mag(x,y) = |G_x| + |G_y| \text{ (Aproximación)}$$

**Direccion del vector gradiente.**
Denotado por $\alpha(x,y)$ se define por:

$$\alpha(x,y) = \arctan(\frac{G_y}{G_x})$$

### Filtro de Roberts.

Éste tipo de filtros necesitan más de una máscara para poder llevar a cabo un filtro completo, como el nombre lo dice, vamos a hacer detección de bordes, y debido a que son filtros que también están orientados a la región/ vecindad, vamos a hacer las operaciones de convolución. La transformación lineal de este filtro está dada por las siguientes máscaras:

$$M_{rx} = \begin{bmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ Horizontal.} \qquad M_{ry} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ Vertical.}$$

$$M_{r45} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ 45 Grados.} \qquad M_{r-45} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ -45 Grados.}$$

Mientras el filtro completo está dado por el promedio de estas matrices aplicadas a la imagen:

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1k} \\ x'_{21} & x'_{22} & \cdots & x'_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{k1} & x'_{k2} & \cdots & x'_{kk} \end{pmatrix} = (M_{rx} + M_{ry} + M_{r45} + M_{r-45})/4$$

Reiterando que $M_{rx}$, $M_{ry}$, $M_{r45}$ y $M_{r-45}$ fueron aplicadas previamente en las regiones K*K de la imágen.

Obteniendo lo siguiente:

(a) Original (b) Horizontal (c) Vertical

(a) 45 grados (b) -45 grados (c) Completo

### Filtro de Prewitt.

La transformación lineal de este filtro está dada por las siguientes máscaras:

$$M_{rx} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ Horizontal.} \qquad M_{ry} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \text{ Vertical.}$$

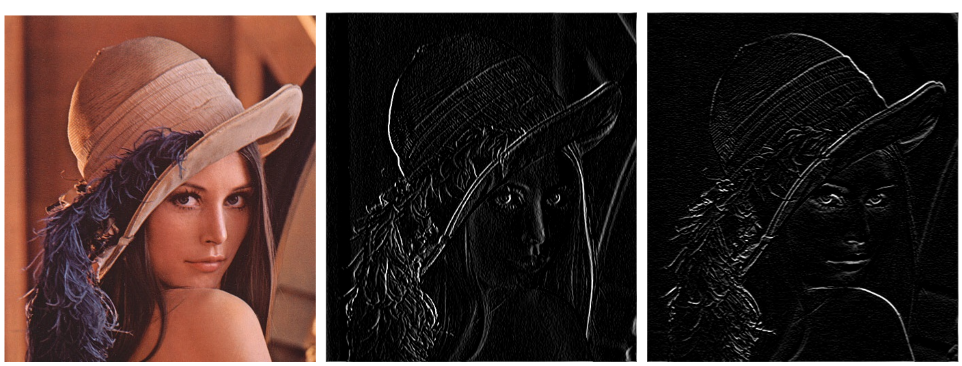
$$M_{r45} = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \text{ 45 Grados.} \qquad M_{r-45} = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \text{ -45 Grados.}$$

Mientras el filtro completo está dado por el promedio de estas matrices aplicadas a la imagen:

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1k} \\ x'_{21} & x'_{22} & \cdots & x'_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{k1} & x'_{k2} & \cdots & x'_{kk} \end{pmatrix} = (M_{rx} + M_{ry} + M_{r45} + M_{r-45})/4$$

Reiterando que $M_{rx}$, $M_{ry}$, $M_{r45}$ y $M_{r-45}$ fueron aplicadas previamente en las regiones K*K de la imágen.

Obteniendo lo siguiente:



(a) Original (b) Horizontal (c) Vertical

(a) 45 grados (b) -45 grados (c) Completo

## Filtro de Sobel.

La transformación lineal de este filtro está dada por las siguientes máscaras:

$$M_{rx} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ Horizontal.} \qquad M_{ry} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{ Vertical.}$$

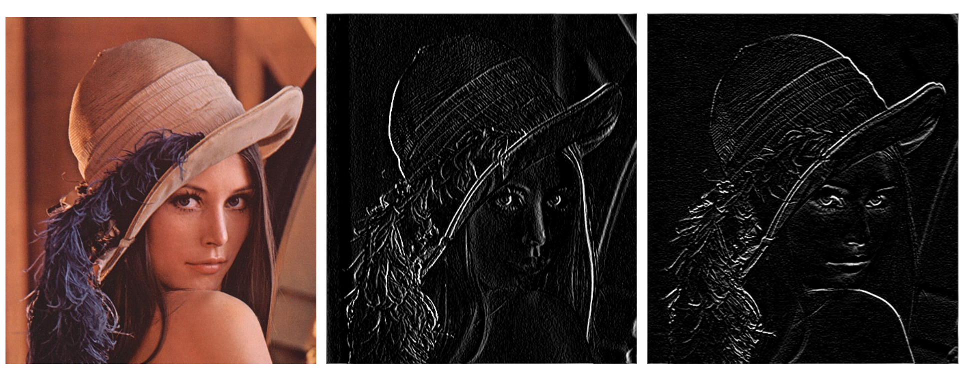
$$M_{r45} = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \text{ 45 Grados.} \qquad M_{r-45} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \text{ -45 Grados.}$$

Mientras el filtro completo está dado por el promedio de estas matrices aplicadas a la imagen:

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1k} \\ x'_{21} & x'_{22} & \cdots & x'_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{k1} & x'_{k2} & \cdots & x'_{kk} \end{pmatrix} = (M_{rx} + M_{ry} + M_{r45} + M_{r-45})/4$$

Reiterando que $M_{rx}$, $M_{ry}$, $M_{r45}$ y $M_{r-45}$ fueron aplicadas previamente en las regiones K*K de la imágen.

Obteniendo lo siguiente:



(a) Original (b) Horizontal (c) Vertical

(a) 45 grados (b) -45 grados (c) Completo

**Filtro de Kirsch.**

La transformación lineal de este filtro está dada por las siguientes máscaras:

$$M_0 = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \text{ 0 grados.} \qquad M_{45} = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \text{ 45 grados.}$$

$$M_{90} = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \text{ 90 Grados.} \qquad M_{135} = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \text{ 135 Grados.}$$

$$M_{180} = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \text{ 180 Grados.} \qquad M_{225} = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \text{ 225 Grados.}$$

$$M_{270} = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \text{ 270 Grados.} \qquad M_{315} = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \text{ 315 Grados.}$$

Máscaras obtenidas de [Aramayo, 2017]

Mientras el filtro completo está dado por el promedio de estas matrices aplicadas a la imagen:

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1k} \\ x'_{21} & x'_{22} & \cdots & x'_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{k1} & x'_{k2} & \cdots & x'_{kk} \end{pmatrix} = (M_0 + M_{45} + M_{135} + M_{180} + M_{225} + M_{270} + M_{315})/8$$

Reiterando que $M_0$, $M_{45}$, $M_{90}$, $M_{135}$, $M_{180}$, $M_{225}$, $M_{270}$ y $M_{315}$ fueron aplicadas previamente en las regiones K*K de la imágen.

Obteniendo lo siguiente:


(a) Completo


(b) 0 grados


(c) 45 grados


(a) 90 grados


(b) 135 grados


(c) 180 grados


(a) 225 grados


(b) 270grados


(c) 315 grados

**Filtro de Frei-Chen.**

La transformación lineal de este filtro está dada por las siguientes máscaras:

$$\frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} \text{ Horizontal.} \qquad \frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} \text{ Vertical.}$$

$$\frac{1}{2\sqrt{2}}\begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix} \text{ 45 Grados.} \qquad \frac{1}{2\sqrt{2}}\begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix} \text{ -45 Grados.}$$

Máscaras obtenidas de [Romero Manchado, 2013]

Mientras el filtro completo está dado por el promedio de estas matrices aplicadas a la imagen:

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1k} \\ x'_{21} & x'_{22} & \cdots & x'_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{k1} & x'_{k2} & \cdots & x'_{kk} \end{pmatrix} = (M_{rx} + M_{ry} + M_{r45} + M_{r-45})/4$$

Reiterando que $M_{rx}$, $M_{ry}$, $M_{r45}$ y $M_{r-45}$ fueron aplicadas previamente en las regiones K*K de la imágen.

Obteniendo lo siguiente:


(a) Original


(b) Horizontal


(c) Vertical


(a) 45 grados


(b) -45 grados


(c) Completo

### Laplaciana.

La transformación lineal de este filtro está dada por las siguientes máscaras:

$$M_{L+} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad M_{LX} = \begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix} \text{ Diagonales.}$$

Horizontal/Vertical.

Mientras el filtro completo está dado por el promedio de estas matrices aplicadas a la imagen:

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1k} \\ x'_{21} & x'_{22} & \cdots & x'_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{k1} & x'_{k2} & \cdots & x'_{kk} \end{pmatrix} = (M_{L+} + M_{LX})/2$$

Reiterando que $M_{L+}$ y $M_{LX}$ fueron aplicadas previamente en las regiones K*K de la imágen.

Obteniendo lo siguiente:

(a) Original (b) Horizontal/Vertical (c) Diagonales (d) Completo

### Laplaciana de la Gaussiana.

La transformación lineal de este filtro está dada por las siguientes máscaras:

$$M_{LG+} = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \qquad M_{LGX} = \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} \text{ Diagonales.}$$

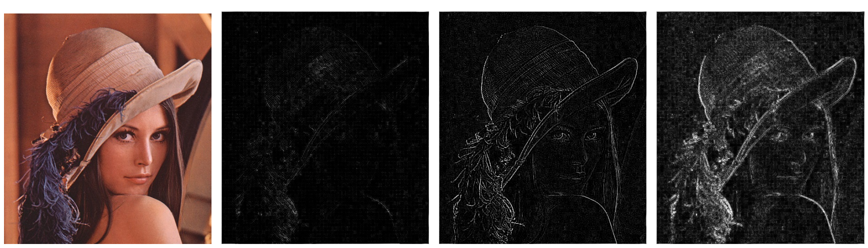
Horizontal/Vertical.

Mientras el filtro completo está dado por el promedio de estas matrices aplicadas a la imagen:

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1k} \\ x'_{21} & x'_{22} & \cdots & x'_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{k1} & x'_{k2} & \cdots & x'_{kk} \end{pmatrix} = (M_{LG+} + M_{LGX})/2$$

Reiterando que $M_{LG+}$ y $M_{LGX}$ fueron aplicadas previamente en las regiones K*K de la imágen.

Obteniendo lo siguiente:



(a) Original (b) Horizontal/Vertical (c) Diagonales (d) Completo

# Capítulo 5

# Operaciones geométricas en imágenes digitales.

### 5.0.1. Transformaciones geométricas.

Las transformaciones Geométricas modifican la relación espacial entre píxeles. En términos del procesamiento de imágenes digitales una transformación geométrica consiste de dos operaciones básicas:

- Una transformación espacial que define la reubicación de los píxeles en el plano imagen.
- Interpolación de los niveles de grises, los cuales tienen que ver con la asignación de los valores de intensidad de los píxeles en la imagen transformada.

En términos Matemáticos las transformaciones afines son las más usadas en imágenes digitales 2D por su representación y manejo matricial. Una Transformación afín es aquella (transformación) en la que las coordenadas $(x', y')$ del punto imagen son expresadas linealmente en términos de las del punto original $(x, y)$. Es decir, la transformación viene dada por las ecuaciones:

$$x' = ax + by + m$$
$$y' = cx + dy + n$$

[Melo, 2009]

### 5.0.2. Escalado Lazy.

Este método de escalado, también conocido como .$^{\text{el}}$ método flojo", es el más sencillo, aunque no el mejor, pues la imagen perderá mucha información en el caso de Escalado entre 2, mientras que en el Escalado por 2, se rellenan los espacios vacíos con la misma información de la imagen, lo cual funciona, pero no obtendremos los mejores resultados.

### Escalado por 2.

Como se explicó en lo anterior, este método consistirá en aumentar el tamaño de la imagen x2, tomando todos los pixeles originales, pero al ser transformada a una imagen de mayor tamaño, hay que rellenar los espacios vacíos, de tal forma que, los pixeles nuevos serán el pixel izquierdo más cercano, por ejemplo, una imagen de 3x3 al ser escalada al doble de su tamaño (6x6) con este método quedaría algo asi:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 2 & 2 & 3 & 3 \\ 1 & 1 & 2 & 2 & 3 & 3 \\ 4 & 4 & 5 & 5 & 6 & 6 \\ 4 & 4 & 5 & 5 & 6 & 6 \\ 7 & 7 & 8 & 8 & 9 & 9 \\ 7 & 7 & 8 & 8 & 9 & 9 \end{bmatrix}$$

Se puede notar que el pixel de la izquierda basicamente se triplica en los pixeles vecinos, de tal manera que al aplicarlo en una imagen obtenemos lo siguiente:

(a) Original (b) EscaladoX2

### Escalado entre 2.

Este método consistirá en disminuir el tamaño de la imagen por la mitad, tomando todos los pixeles originales, pero al ser transformada a una imagen de mayor tamaño, hay que rellenar los espacios vacíos, de tal forma que, los pixeles nuevos serán el pixel izquierdo más cercano, por ejemplo, una imagen de 6x6 al ser escalada a la mitad de su tamaño (3x3) con este método quedaría algo asi:

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 \\ 19 & 20 & 21 & 22 & 23 & 24 \\ 25 & 26 & 27 & 28 & 29 & 30 \\ 30 & 31 & 32 & 33 & 34 & 35 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 3 & 5 \\ 13 & 15 & 17 \\ 25 & 27 & 29 \end{bmatrix}$$

Se puede notar que en este escalado vamos a perder información, algo practicamente imposible de evitar, pero se cumple con el objetivo, de tal manera que obtenemos lo siguiente:

(a) Original

(b) EscaladoE2

#### Escalado libre.

### 5.0.3. Escalado con interpolación lineal.

#### Escalado por 2.

#### Escalado entre 2.

### 5.0.4. Escalado con interpolación bi-lineal.

#### Escalado por 2.

#### Escalado entre 2.

### 5.0.5. Reflexión.

#### Horizontal.

La reflexión horizontal es una funcionalidad bastante sencilla e intuitiva en realidad, supongamos que tenemos una imagen de tamaño 5x3, el resultado de la imagen es simplemente cambiar la posicion de los pixeles, es decir, el primero se va al final y el último se va al principio, hablando de los pixeles recorridos fila por fila.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \\ 13 & 14 & 15 \end{bmatrix} \rightarrow \begin{bmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \\ 9 & 8 & 7 \\ 12 & 11 & 10 \\ 15 & 14 & 13 \end{bmatrix}$$

Aplicando este método a una imagen digital obtendremos lo siguiente:

(a) Original

(b) Reflexion Horizontal

### Vertical.

La reflexión vertical es extremadamente parecida a la anterior, con la diferencia que el cambio de pixeles lo hacemos en las columnas y no en las filas, supongamos que tenemos una imagen de tamaño 5x3, el resultado de la imagen es simplemente cambiar la posicion de los pixeles, es decir, el primero se va al final y el último se va al principio, hablando de los pixeles recorridos columna por columna.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \\ 13 & 14 & 15 \end{bmatrix} \rightarrow \begin{bmatrix} 13 & 14 & 15 \\ 10 & 11 & 12 \\ 7 & 8 & 9 \\ 4 & 5 & 6 \\ 1 & 2 & 3 \end{bmatrix}$$

Aplicando este método a una imagen digital obtendremos lo siguiente:

(a) Original

(b) Reflexion Vertical

## 5.0.6. Rotación.

### Rotación de 90 grados.

### Rotación de 180 grados.

### Rotación libre.

# Bibliografía


[Aramayo, 2017] Aramayo, F. R. (2017). Técnicas de procesamiento de imágenes, para la detección o diagnóstico de enfermedades en imágenes del sector agrícola. *Difusiones*, 13(13):75–97.

[Elizondo and Maestre, 2002] Elizondo, J. E. and Maestre, L. P. (2002). Fundamentos de procesamiento de imágenes. *Documentación Universidad Autónoma de Baja California, Unidad Tijuana.*

[Hoffman et al., 1973] Hoffman, K., Kunze, R., and Finsterbusch, H. (1973). *Álgebra lineal.* EDICIONES DEL CASTILLO.

[Jácome et al., 2017] Jácome, M., Torres, C., and Araujo, C. (2017). Enseñanza del procesamiento digital de imágenes a través de objetos virtuales de aprendizaje en entornos e-learning. *Revista colombiana de tecnologías de avanzada (RCTA)*, 2(28).

[Lafuente, 2009] Lafuente, R. (2009). Normas inducidas del multiplicador de hadamard.

[Melo, 2009] Melo, S. B. (2009). Transformaciones geométricas sobre imágenes digitales. *Facultad de Ciencias-Carrera de Matemáticas. Universidad Distrital Francisco José de Caldas.*

[Ortiz, 2013] Ortiz, M. M. (2013). Procesamiento digital de imágenes. *Benemérita universidad Autónoma de Puebla http://www. cs. buap. mx/˜ mmartin/pdi.*

[Roa-Fuentes and Oktaç, 2010] Roa-Fuentes, S. and Oktaç, A. (2010). Construcción de una descomposición genética: Análisis teórico del concepto transformación lineal. *Revista latinoamericana de investigación en matemática educativa*, 13(1):89–112.

[Romero Manchado, 2013] Romero Manchado, A. (2013). Determinación automática de puntos de fuga en imágenes monoscópicas: aplicación al patrimonio histórico.

[Torres, 1996] Torres, A. D. (1996). Procesamiento digital de imágenes. *Perfiles Educativos*, (72).