

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO



PROCESAMIENTO DE IMÁGENES EN 2D, CASO DE ESTUDIO:
DOMINIO ESPACIAL

TESIS

QUE PARA OBTENER EL TÍTULO DE: LICENCIADO EN
INFORMÁTICA

PRESENTA

JOSÉ LUIS GONZÁLEZ PÉREZ

DIRIGIDA POR

M. C. VICENTE RODRÍGUEZ HERNÁNDEZ

CENTRO UNIVERSITARIO
QUERÉTARO, QRO. – MÉXICO

2001

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMATICA

No. Adq. F06705
Clasif. 621.367
Cutter 9693p

DEDICATORIA:

A Dios,
por iluminar mi camino.

A mis padres y hermanas,
con todo mi amor y agradecimiento por el apoyo incondicional que siempre me han
brindado.

A Gloria Angélica Ruiz Aboytes,
Gracias por tu amor, comprensión, y confianza que has depositado en mí.

A mis amigos:
Daniel, Juanita, Juan Luis y Mary

Al L.I. Ramón López Velarde Fonseca,
por sus enseñanzas y por ser un excelente amigo y ser humano.

A mi director de tesis:
M.C. Vicente Rodríguez Hernández,
por su compromiso e incansable labor por fomentar la investigación para generar el
conocimiento.

CONTENIDO

INTRODUCCIÓN	1
PARTE I.- ANÁLISIS DEL PROBLEMA	
<hr/>	
Capítulo 1.- Conceptos Generales.	2
1.1.- Antecedentes del Procesamiento Digital de Imágenes.	3
1.2.- Descripción del problema.	4
1.3.- Justificación de la tesis.	4
1.4.- ¿A quién está orientada la tesis?	4
PARTE II.- TRATAMIENTO DE IMÁGENES	
<hr/>	
Capítulo 2.- Metodología empleada.	5
2.1.- Imagen.	6
2.2.- Fundamentos de la imagen en 2D.	6
2.3.- Fundamentos de la digitalización de imágenes.	7
2.3.1.- Imagen digital.	7
2.3.2.- Adquisición, muestreo y cuantización.	7
2.3.3.- Almacenamiento y formatos de archivos de imágenes.	10
2.3.4.- Despliegue	10
2.4.- Procesamiento Digital de Imágenes.	11
2.5.- Relaciones entre píxeles.	11
2.5.1.- Adyacencia.	11
2.5.2.- Píxeles vecinos.	12
2.5.3.- Conectividad.	12
2.6.- Fundamentos del modelo de color RGB.	13
Capítulo 3.- Procesamiento Puntual.	14
3.1.- Balance de los tonos de gris.	15
3.1.1.- Transformación logarítmica.	15
3.1.2.- Transformación exponencial.	16
3.2.- Histograma de una imagen.	16
3.2.1.- Métodos basados en el histograma.	18
3.3.- Igualación del histograma.	19
3.4.- Negativo de una imagen.	20
3.5.- Binarización.	22
3.6.- Operaciones algebraicas entre dos imágenes.	23
3.6.1.- Adición de imágenes.	23
3.6.2.- Sustracción de imágenes.	24
3.6.3.- Multiplicación y división de imágenes.	25
Capítulo 4.- Procesamiento Espacial.	26
4.1.- Procesamiento basado en una región de la imagen.	27
4.1.1.- Convolución.	28

4.2.- Reducción de ruido.	29
4.2.1.- Filtro del promedio.	29
4.2.2.- Filtro de mediana.	30
4.2.3.- Filtro de pasa – bajas.	32
4.3.- Detección de contornos.	33
4.3.1.- Filtros de pasa – altas.	33
4.3.2.- Resaltamiento de bordes.	34
4.3.3.- Resaltamiento de líneas.	36
4.3.4.- Gradientes.	38
4.3.4.1.- Operadores de Roberts.	39
4.3.4.2.- Operadores de Prewitt.	40
4.3.4.3.- Operadores de Sobel.	40
4.3.5.- Laplaciano.	42
Capítulo 5.- Metodología del sistema de Procesamiento Digital de Imágenes.	45
5.1.- Descripción y justificación de la plataforma utilizada.	46
5.2.- Características del FastGraph for Windows.	46
5.3.- Tipos de procesamiento empleadas en el software.	47
Capítulo 6.- Resultados de aplicar el sistema a determinadas imágenes.	48
LÍNEAS DE INVESTIGACIÓN FUTURAS.	59
CONCLUSIONES.	61
APÉNDICE A: Estructura del software de Procesamiento Digital de Imágenes.	63
A.1.- Estructuras de datos empleadas en los diferentes métodos de procesamiento.	64
A.1.1.- Método de procesamiento puntual.	68
A.1.2.- Método de procesamiento espacial.	68
A.2.- ¿Cómo se estructuró el software de Procesamiento Digital de Imágenes?	70
BIBLIOGRAFÍA.	76
GLOSARIO.	78

INTRODUCCIÓN

Con el surgimiento de las computadoras, se dio origen a un campo llamado Procesamiento Digital de Imágenes (PDI), que tiene como objetivo mejorar imágenes representadas en forma digital, para ser procesadas por medio de las computadoras.

El presente trabajo de tesis, está organizado en dos partes. La primera parte se llama "Análisis del Problema", la cual abarca el capítulo 1. La segunda parte denominada "Tratamiento de imágenes", emplea la metodología del PDI, e incluye los siguientes 5 capítulos.

El capítulo 1, inicia describiendo los antecedentes del PDI; posteriormente se explican los aspectos generales sobre el desarrollo de esta tesis, donde se incluye la justificación de haber realizado este trabajo, también la descripción y delimitación del problema que se pretende resolver; por último se especifica a quién está orientada.

El capítulo 2, trata sobre la metodología del PDI. Primeramente se define lo que es una imagen, luego se explican los fundamentos de las imágenes en 2D, inmediatamente después se estudia la representación de imágenes digitales, y sobre los conceptos que definen al PDI, incluyendo las técnicas que en esta tesis se abordaran; así como las etapas que abarca la metodología. Posteriormente se analizan las relaciones que existen entre los píxeles de una imagen. Finalmente se revisan los fundamentos del modelo de color RGB, que es el que se utiliza más frecuentemente para gráficos por computadora.

En el capítulo 3, se estudia lo referente al método de procesamiento puntual. En esta parte, se explicará la forma en que se puede lograr el resaltamiento a una imagen, por medio de las transformaciones exponencial, logarítmica y los procesos de igualación de histogramas, las cuales producen efectos en el brillo y contraste en las imágenes; también se incluye las operaciones de negativo y binarización. El capítulo termina abordando las operaciones algebraicas entre imágenes digitales, como lo es la adición, sustracción, multiplicación y división.

En el capítulo 4, se analiza el método de procesamiento espacial, el cual abarca temas sobre la reducción del ruido en las imágenes, a través de la implantación de filtros de mediana, promedio y filtros "pasa - bajas". Por otra parte, se revisan también filtros que permiten resaltar bordes, como lo son los filtros de "pasa - altas", resaltamiento de bordes, resaltamiento de líneas, gradientes y laplacianos.

En el capítulo 5, se describe la metodología empleada para el desarrollo del sistema de PDI.

En el capítulo 6, se muestran los resultados de usar el software para procesar algunas imágenes digitales en 2D.

Por otra parte, se presentan también las líneas de investigación futuras de esta tesis, así como las conclusiones finales.

Finalmente el Apéndice A, muestra la estructura del software de PDI.

PARTE I
ANÁLISIS DEL PROBLEMA
CAPÍTULO 1
CONCEPTOS GENERALES

1.1.- Antecedentes del Procesamiento Digital de Imágenes

El PDI surgió como una tecnología asociada a las Ciencias de la Computación, siendo una proyección de la *Visión Artificial* dentro del campo de la Inteligencia Artificial.

La primera vez que se utilizaron técnicas de imágenes digitales fue en los años veinte en la transmisión de imágenes de fotografías periodísticas a través de cable submarino entre Londres y Nueva York.

Sin embargo, es hasta los años 50-60 en que empiezan a desarrollarse las técnicas de procesamiento de imágenes de forma sistemática, debido a que es en este momento en que aparecen las primeras computadoras digitales de gran potencia y además surge la necesidad de disponer de técnicas para la transmisión y procesamiento de imágenes desde satélite. En 1964 en el Laboratorio de Propulsión Espacial (en Pasadena, California) se empezaron a utilizar técnicas computacionales para mejorar las imágenes recibidas de una sonda espacial, tal es el caso de las imágenes de la Luna transmitidas por el Ranger 7, las cuales fueron procesadas por computadora para corregir diversos tipos de distorsión.

Los conceptos de *técnicas para el procesamiento de imágenes digitales*, significan el conjunto de todas aquellas técnicas asociadas a la captura, codificación y representación de las imágenes¹.

El interés por los métodos de tratamiento digital de imágenes se derivó de dos áreas principales de aplicación: la mejora de la información pictórica para la interpretación humana y el procesamiento de los datos de la escena para la percepción autónoma por una computadora².

Durante la década de los ochenta, las técnicas de procesamiento de imágenes se desarrollan de forma vertiginosa, debido a la gran cantidad de aplicaciones que comienzan a surgir y al mejoramiento de las arquitecturas de computadoras.

Actualmente existen aplicaciones que incluyen sofisticadas técnicas de tratamiento de imágenes, algunas de las aplicaciones son las siguientes:

- Diagnóstico médico (análisis de radiografías, tomografías, muestras de sangre).
- Observaciones aéreas (análisis de distribuciones urbanísticas, estudio y control de cultivos agrarios, predicción atmosférica, análisis de objetos de imágenes adquiridas en misiones espaciales).
- Procesos industriales (control de calidad de productos fabricados e inspección de los procesos de fabricación).
- Reconocimiento de caracteres escritos (OCR).
- Robótica (supervisión y control de robots, guiado de vehículos móviles).
- Tratamiento automático de huellas digitales, rostros, etc.

¹ PÉREZ, N. B. (1995). *Fundamentos del tratamiento de imágenes. Visión por computador*. <http://www.uco.es/informa/local/cordoba/ccordoba/ccordoba.html>

² GONZÁLEZ, R. C. (1992). *Digital image processing*. Addison-Wesley Publishing Company.

1.2.- Descripción del problema

La delimitación del problema comprende la realización del análisis, diseño e implantación de un sistema computacional para procesar imágenes digitales en 2D. Este software de PDI, incluye los algoritmos del procesamiento en el dominio espacial.

Por otra parte, esta tesis pretende apoyar el trabajo de alumnos de la facultad de informática, reforzando el estudio de la materia llamada Gráficas Interactivas por Computadora para licenciatura y la de Visión Computacional en maestría. Este trabajo también cubrirá las necesidades de todas aquellas personas que se encuentren investigando sobre el PDI, y para aquellas que inicien líneas de investigación futuras.

1.3.- Justificación de la tesis

Actualmente se encuentran por medios diversos programas que se encargan de realizar procesamiento de imágenes; sin embargo, estos no proporcionan el código fuente el cual sería de gran utilidad para realizar mejoras o adecuaciones para el procesamiento de imágenes específicas, otros más son de carácter comercial y sus costos para adquirirlos no son factibles para la facultad. Por lo anterior, el desarrollo de esta tesis es producto de la investigación sobre el PDI y, obtiene como resultado la elaboración de un software de carácter general que sirve para mejorar imágenes digitales en 2D para su análisis posterior. El documento incluye la manera de estructurar un software de esta naturaleza; así como un CD-ROM que contiene los códigos fuentes correspondientes.

1.4.- ¿A quién está orientada la tesis

Esta tesis esta orientada principalmente a:

- Alumnos de la Facultad de Informática.
- Futuros tesisas que desarrollen líneas de investigación derivadas del PDI.

PARTE II
TRATAMIENTO DE IMÁGENES

CAPÍTULO 2
METODOLOGÍA EMPLEADA

La metodología empleada para la construcción del software de PDI en 2D, es la propia del *Procesamiento Digital de Imágenes*, la cual se describe en este capítulo.

2.1.- Imagen

La formación de la imagen sucede en el momento en que un sensor (ojo, cámara) registra la radiación (luz) que interactúa con ciertos objetos físicos³.

De forma general, el concepto de imagen se puede asociar al de un “mapa espacial” producido por algún tipo de sensor que actúa sobre una determinada información. Ejemplos de este tipo de “mapas” son nuestras percepciones de una determinada escena, originadas de nuestros sistemas sensoriales (vista, oído, tacto, gusto y olfato). Si ahora nos enfocamos en el modelo visual humano, el concepto de ver se puede asociar al de percibir una señal luminosa con una intensidad mínima, en un rango de frecuencia espectral dado; sin embargo, actualmente existe la posibilidad de obtener imágenes a partir de sensores que trabajan en condiciones distintas de iluminación a las que es sensible el ojo humano, p.e. el infrarrojo, rayos X, etc.⁴.

2.2.- Fundamentos de la imagen en 2D

El término *imagen* se refiere a una función bidimensional de la luz y la intensidad, indicada por $f(x, y)$, donde el valor o amplitud de f en las coordenadas espaciales (x, y) da la intensidad (iluminación) de la imagen en este punto. Debido a que la luz es una forma de energía, $f(x, y)$ debe de ser estrictamente mayor que cero y finita, como se indica en la ecuación (2.2-1).

$$0 < f(x, y) < \infty \quad (2.2-1)$$

Las imágenes que se perciben en las actividades visuales cotidianas provienen normalmente de la luz reflejada por los objetos. La naturaleza básica de $f(x, y)$ puede estar caracterizada por dos componentes: 1) la cantidad de luz incidente procedente de la fuente sobre la escena contemplada y 2) la cantidad de luz reflejada por los objetos de la escena. De forma apropiada, reciben el nombre de *componentes de iluminación y reflectancia*, y se indican por $i(x, y)$ y $r(x, y)$ respectivamente. Las funciones $i(x, y)$ y $r(x, y)$ se combinan, como producto, para dar $f(x, y)$, como se muestra en la ecuación (2.2-2).

$$f(x, y) = i(x, y)r(x, y) \quad (2.2-2)$$

donde

$$0 < i(x, y) < \infty \quad (2.2-3)$$

³ Idea tomada del archivo notas.ps. Disponible en:
<http://campus.mor.itesm.mx/~esucar/Vision/indice.html>

⁴ PÉREZ, N. B. (1995). *Fundamentos del tratamiento de imágenes. Visión por computador*.
<http://www.uco.es/informa/local/cordoba/ccordoba/ccordoba.html>

y

$$0 < r(x, y) < 1 \quad (2.2-4)$$

La ecuación (2.2-4) indica que la reflectancia está acotada entre 0 (absorción total) y 1 (reflexión total). La naturaleza de $i(x, y)$ está determinada por la fuente de luz y $r(x, y)$ está determinada por las características de los objetos de la escena.

A la intensidad de una imagen monocromática f en las coordenadas (x, y) se la denominará *nivel de gris* (l) de la imagen en este punto. De las ecuaciones anteriores, resulta que l está en el rango L_{min} y L_{max} .

El único requisito sobre L_{min} es que sea positiva y sobre L_{max} que sea finita. Al intervalo $[L_{min}, L_{max}]$ se le denomina la *escala de grises*⁵.

De aquí en adelante se considerarán solamente imágenes en escala de grises, debido a que el PDI se ha desarrollado en torno al uso de formas y el color no es una forma, sino un atributo de las formas.

2.3.- Fundamentos de la digitalización de imágenes

2.3.1.- Imagen digital

Una imagen digital, es una imagen $f(x,y)$ que ha sido discretizada en valor de intensidad (niveles de gris) y en coordenadas espaciales, es decir, que se ha realizado un muestreo de la función continua. En otras palabras, de una escena que por definición es continua, donde los puntos son infinitamente pequeños, el proceso de la digitalización se da de la siguiente manera: a la escena se le sobreimpone una rejilla cuadrada que la subdivide en regiones iguales. Sobre cada cuadro la integra, la promedia y proporciona una media de ella a distancia. El siguiente paso es la construcción de una tabla de asignación de niveles de gris, es decir el rango dinámico. Finalmente se despliega la imagen digital. Por lo anterior una imagen digital puede ser representada por un arreglo bidimensional, donde x, y denotan los índices del arreglo; estos índices de renglón y columna identifican la posición de un punto en la imagen, y el valor del correspondiente elemento del arreglo, el nivel de gris cuantizado en ese punto. Los elementos de este arreglo bidimensional son llamados elementos de imagen, o más comúnmente pixels (este término deviene del inglés "picture-element", elemento de imagen).

2.3.2.- Adquisición, muestreo y cuantización

Para obtener una imagen digital se necesitan dos elementos. El primero es un dispositivo físico sensible a una determinada banda del espectro de energía electromagnética (como las bandas de rayos X, ultravioleta, visible o infrarrojo) y que produzca una señal eléctrica de salida proporcional al nivel de energía detectado. El segundo, denominado *digitalizador*, es un dispositivo para convertir la señal de salida del sistema sensible a forma digital⁶.

⁵ GONZÁLEZ, R. C. (1992). *Digital image processing*. Addison-Wesley Publishing Company.

⁶ GONZÁLEZ, R. C. (1992). *Digital image processing*. Addison-Wesley Publishing Company.

Las dos formas más usuales para adquirir imágenes digitales es utilizando un "scanner" (escáner) o una cámara digital.

De acuerdo a lo explicado hasta este momento, una función $f(x, y)$ debe ser digitalizada tanto espacialmente como en su amplitud, para que pueda ser utilizable en el procesamiento por medio de computadora. La digitalización de las coordenadas espaciales (x, y) se denomina *muestreo de la imagen* y la digitalización de la amplitud se conoce bajo el nombre de *cuantificación del nivel de gris*.

El muestreo consiste en representar una imagen continua $f(x, y)$ en forma de matriz de $N \times M$, la cual se obtiene de forma aproximada por una serie de muestras igualmente espaciadas, como se indica en la ecuación (2.3-1), donde cada elemento de la matriz es una cantidad discreta:

$$f(x, y) \approx \begin{pmatrix} f(0, 0) & f(0, 1) & \dots & f(0, M-1) \\ f(1, 0) & f(1, 1) & \dots & f(1, M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1, 0) & f(N-1, 1) & \dots & f(N-1, M-1) \end{pmatrix} \quad (2.3-1)$$

El término de la derecha de la ecuación representa una *imagen digital*. En la siguiente figura se observan los efectos producidos por el muestreo.

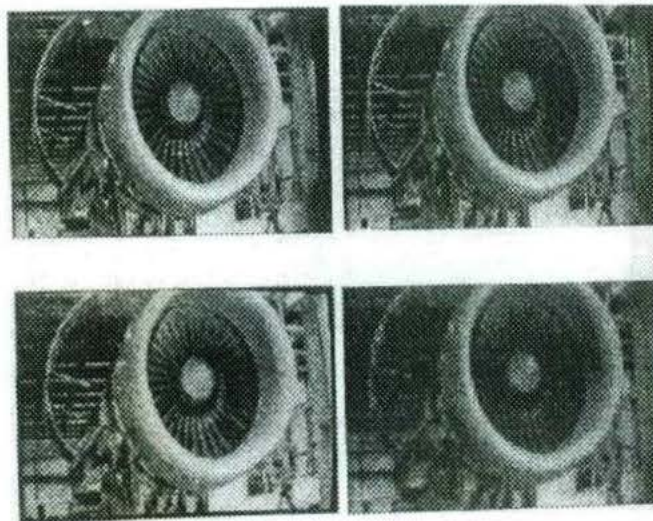


Figura 2.1 Se muestran cuatro imágenes digitales de la misma escena, con una resolución espacial progresivamente reducida: 512x512, 256x256, 128x128 y 64x64 píxeles, respectivamente⁷.

⁷ Imagen tomada de: ALLENDE, D. G. (1993). *Reconocimiento de formas y visión artificial*. Addison-Wesley Iberoamericana.

Después de la discretización del espacio (muestreo), se tiene la cuantización, que consiste en la asignación de promedios de intensidad de la escena a cada pixel de imagen a un número finito de niveles de gris determinado. El efecto de la cuantización se presenta debido a la imposibilidad de tener un rango infinito de valores de medida para la intensidad de brillo de los pixeles. La tecnología actual, generalmente permite tener 256 niveles de gris para codificar este valor lumínico⁸. La siguiente figura muestra cuatro imágenes en las que se puede apreciar los efectos de la cuantización.

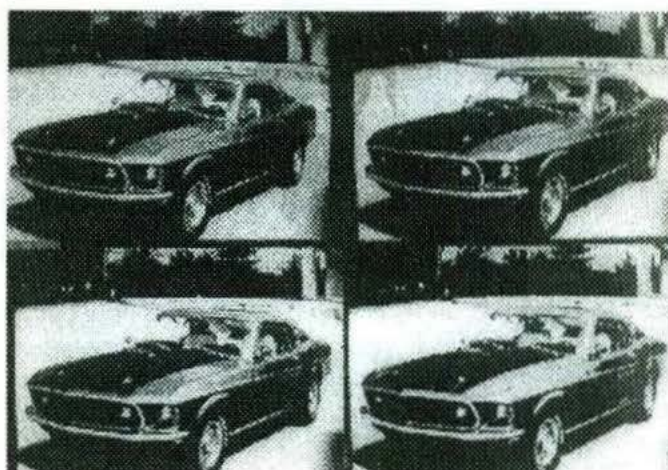


Figura 2.2 Se muestran 4 imágenes digitales de la misma escena, que se han codificado en 256, 16, 8 y 4 niveles de gris, respectivamente⁹.

Para el proceso de digitalización se necesita que se tomen decisiones sobre los valores para N , M y el número de niveles de gris permitidos para cada pixel. En el procesamiento digital de imágenes es una práctica común que estas cantidades sean potencias enteras de 2, como se observa en las siguientes ecuaciones:

$$N = 2^n, \quad M = 2^k \quad (2.3-2)$$

y

$$G = 2^m \quad (2.3-3)$$

donde G indica el número de niveles de gris¹⁰.

Es importante en este punto hacer énfasis sobre la dificultad para determinar lo que es una buena imagen, puesto que la calidad de la misma es muy subjetiva; además de que también depende mucho de las necesidades de una aplicación determinada.

⁸ PÉREZ, N. B. (1995). *Fundamentos del tratamiento de imágenes. Visión por computador*. <http://www.uco.es/informa/local/cordoba/ccordoba/ccordoba.html>

⁹ Imagen tomada de: ALLENDE, D. G. (1993). *Reconocimiento de formas y visión artificial*. Addison-Wesley Iberoamericana.

¹⁰ GONZÁLEZ, R. C. (1992). *Digital image processing*. Addison-Wesley Publishing Company.

2.3.3. Almacenamiento y formatos de archivos de imágenes

Comúnmente cada pixel o elemento de imagen se almacena en un byte tomando alguno de los 256 valores correspondientes.

Empleando las ecuaciones (2.3-2) y (2.3-3) obtenemos el número b , de bits necesarios para almacenar una imagen digitalizada, como se muestra en la siguiente ecuación:

$$b = N \times M \times m \quad (2.3-4)$$

Si $M = N$, entonces la ecuación (2.3-4) se transforma en la ecuación (2.3-5).

$$b = N^2 m \quad (2.3-5)$$

Para una imagen con resolución espacial $256 \times 256 \times 256$ (niveles de gris) la memoria utilizada sería 65536 bytes = 65 Kb; p.e., una imagen de 8 bits y 1.024×1.024 pixeles necesita un millón de bytes de memoria.

Existen dos clases para representar datos gráficos: vectoriales y mapas de bits.

Datos vectoriales. - Constituyen una forma de representar líneas, polígonos o curvas, etc.

Mapas de bits. - Están formados por un conjunto de valores numéricos que representan los colores de los pixeles.

El PDI se basa en archivos de mapas de bits, de ahí que nuestro interés sea solamente esta clase. Existen varios formatos de archivos de mapas de bits, sin embargo todos ellos tienen básicamente la misma estructura. El contenido de este tipo de archivos está formado primeramente por una cabecera, la cual puede estar en binario o ASCII y contiene información sobre el mapa de bits que se encuentra en el archivo. Dicha información describe p.e., el tamaño de la imagen en filas y columnas, el identificador que define el formato de almacenamiento (bmp, gif, jpg, ...), etc. También se encuentra dentro del archivo los datos como mapa de bits, que definen a la imagen y esta puede estar almacenada por filas, por columnas o por planos de bits. Por último es posible que se almacene una paleta de colores, que generalmente se ubica después de la cabecera.

2.3.4.- Despliegue

Una de las pantallas de video que se utiliza en el PDI es la de tubos de rayos catódicos (CRT), que utiliza un haz de electrones para barrer y excitar a una pantalla de fósforo produciendo puntos de luz de intensidad proporcional al nivel de gris correspondiente a cada punto.

Las tecnologías de estado sólido (pantallas planas) desarrolladas actualmente, están reduciendo el dominio del CRT. Las pantallas planas de cristal líquido y de plasma

requieren un solo barrido para desplegar una imagen. Sin embargo, al modificarse un solo pixel es necesario barrer de nuevo la pantalla.

2.4.- Procesamiento Digital de Imágenes

El PDI, está formado por un conjunto de técnicas representadas en forma de algoritmos computacionales (es decir implementados en software), que operan sobre una imagen digital, la cual puede ser en blanco y negro (binaria), o en niveles de gris y a su vez puede ser almacenada en un archivo con formato de imagen apropiado p.e. bmp, gif, jpg, etc. Estas técnicas permiten realzar, restaurar, modificar y/o analizar la información contenida en una imagen con la finalidad de producir nuevas imágenes que resalten algunos de los elementos que conforman la escena, de forma que se facilite el análisis posterior realizado ya sea por un usuario (humano) o un sistema de visión artificial. El PDI nació y se desarrolló como una disciplina de aplicación, que se encarga de extraer información del mundo real, de manera automática a partir de una imagen observada. Los procesos pueden variar de acuerdo con los objetivos y las aplicaciones que se persigan.

Las técnicas de procesamiento que en esta tesis se abordarán, se clasifican en dos categorías: procesamiento puntual y procesamiento espacial.

Procesamiento puntual.- Se basa en puntos de la imagen, y agrupa aquellas operaciones de procesamiento donde el valor del pixel en la imagen de salida sólo depende de su valor en la imagen de entrada.

Procesamiento espacial.- Incluye técnicas basadas en una región de la imagen, donde un pixel de la imagen de salida depende tanto del valor del pixel en la imagen de entrada como en la vecindad del mismo.

2.5.- Relaciones entre pixeles

2.5.1.- Adyacencia

Dos pixeles son adyacentes si y solo si, ellos tienen en común una de sus fronteras, o al menos una de sus esquinas. La figura 2.3 muestra pixeles adyacentes.



Figura 2.3 Píxeles adyacentes: (a) adyacentes por frontera, (b) adyacente por esquina.

2.5.2.-Píxeles vecinos

Dos píxeles son vecinos, si estos cumplen con la definición de adyacencia. Si los píxeles comparten una de sus fronteras se dice entonces que son "Vecinos Directos", si por el contrario se tocan solamente en una de sus esquinas, se llaman "Vecinos Indirectos".

Un píxel p de coordenadas (x, y) tiene cuatro vecinos directos y cuatro indirectos. Las coordenadas de los vecinos directos vienen dadas por:

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

Este conjunto de píxeles, denominado los *4-vecinos* de p , se representa por $N_4(p)$. Cada píxel está a una unidad de distancia de (x, y) , y algunos de los vecinos de p caen fuera de la imagen digital si (x, y) está en el borde de la imagen.

Los cuatro vecinos indirectos de p tienen las coordenadas:

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

y se representan por $N_D(p)$. Estos puntos, junto a los 4-vecinos, se denominan los *8-vecinos* de p , y se representan por $N_8(p)$. Algunos puntos de $N_D(p)$ y $N_8(p)$ caen fuera de la imagen si (x, y) está en el borde de la misma.

En la figura 2.4, se observan vecindades de 4 y de 8. La primera esta formada, por píxeles que son vecinos directos, mientras que la vecindad de 8 esta formada tanto por vecinos directos como por indirectos.



Figura 2.4 Vecindades: (a) vecindad de 4, (b) vecindad de 8

2.5.3.-Conectividad

La conectividad entre píxeles es un concepto importante usado para identificar las fronteras de objetos y las regiones componentes de una imagen. Para determinar si dos píxeles están conectados se debe establecer si son adyacentes en algún sentido.

2.6.- Fundamentos del modelo de color RGB

El modelo RGB es uno de los que se usan para gráficos por computadora ya que los CRT usan fósforo rojo, verde y azul para crear el color deseado; además, se basa en los 3 sensores humanos, considerando que todos los colores son una combinación de 3 colores básicos R (rojo), G (verde), B (azul). En este modelo, cada color aparece en sus componentes primarias espectrales rojo, verde y azul. El modelo se basa en un sistema de coordenadas cartesiano. El subespacio de interés es el cubo que se muestra en la siguiente figura.

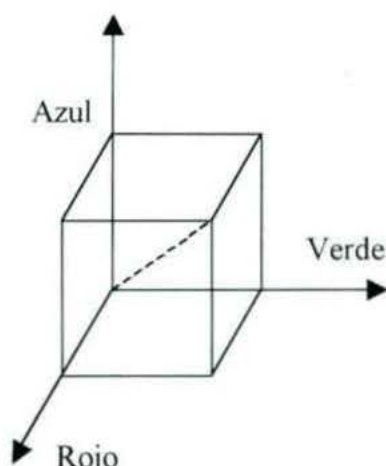


Figura 2.5 Cubo de color RGB. Los puntos en la diagonal principal tienen niveles de gris desde el negro en el origen (0,0,0) al blanco en el punto (255, 255, 255).

Las imágenes en el modelo de color RGB están formadas por tres planos de imágenes independientes, cada una de los colores primarios. Cuando son introducidas en un monitor RGB, las tres imágenes se combinan en la pantalla de fósforo para producir una imagen de color compuesta¹¹. En este sentido, para fines de este trabajo, es posible transformar una imagen a color a su correspondiente escala de grises. El proceso es que para cada pixel de la imagen se obtiene el valor de los tres componentes (rojo, verde y azul), luego se suman estos valores y se dividen entre tres. Finalmente se asigna este valor (nivel de gris) al pixel en cuestión.

¹¹ MOLINA, R. S. y CORTIJO, F. J. B. (1997).
http://www-etsi2.ugr.es/depar/ccia/robotica/indice_apuntes.html

CAPÍTULO 3

PROCESAMIENTO PUNTUAL

El procesamiento puntual aplicado a una imagen de entrada, utiliza operaciones que transforman dicha imagen en una imagen de salida, de forma que cada pixel de la imagen de salida sólo depende del correspondiente pixel en la misma posición de la imagen de entrada. Esto se puede expresar como: $h(x,y) = f(g(x,y))$

De acuerdo a la expresión anterior, la función f especifica el mapeo del nivel de gris de la entrada al nivel de gris de la salida.

El procesamiento puntual utiliza la técnica de mejora de imagen basándose en la intensidad de los pixeles. De esta forma, los procesos punto a punto se encuentran entre las técnicas de mejoramiento en las imágenes más simples.

3.1.- Balance de los tonos de gris

Las transformaciones sobre el balance de tono de gris se utilizan, para resaltar detalles en la imagen que ocurren en las regiones oscuras y luminosas de la misma. Entre estas transformaciones se encuentran las transformaciones logarítmicas y exponenciales.

3.1.1.- Transformación logarítmica

Esta transformación se utiliza principalmente para comprimir el rango dinámico de los valores del nivel de gris de aquellas imágenes que tienen rangos dinámicos muy grandes de niveles de gris; superiores a los 256 que la tecnología actual generalmente puede representar. La forma de expresar esta transformación es la siguiente:

$$f(x, y) = k/\alpha \ln[1 + (e^\alpha - 1) f(x, y)] \quad (3.1-1)$$

donde k es una constante, $f(x, y)$ es el nivel de gris de la imagen en el punto (x, y) y α es un parámetro de escalación dado por el usuario.

El procedimiento general de esta operación se encuentra definida de la siguiente forma (ver Apéndice A):

procedure TFormMain.Contraste (alfa: real);

Los valores que puede tomar alfa están entre $0 < \alpha < 1$. Si el valor de α está cercano a 1, la imagen se aclara poco (ver figura 3.1 (b)), pero conforme se tome valores más pequeños sin llegar a cero, la imagen se aclara mucho más, como se observa en la figura 3.1 (c).

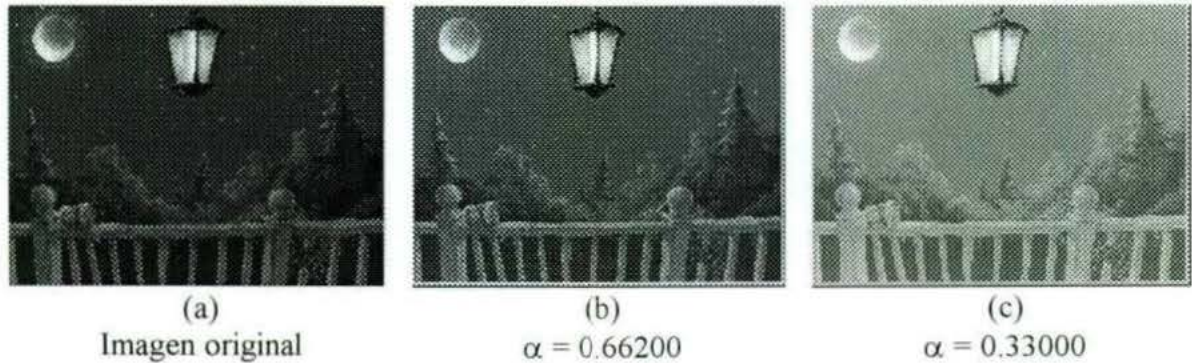


Figura 3.1 Transformación logarítmica con diferentes parámetros de α .

3.1.2.- Transformación exponencial

Esta transformación es inversa al efecto de la transformación logarítmica, es decir, sirve para oscurecer regiones claras, como se aprecia en la figura 3.2 y se puede expresar de la siguiente forma:

$$f(x, y) = k/\alpha [(1 + \alpha)^{f(x, y)} - 1] \quad (3.1-2)$$

donde k es una constante, $f(x, y)$ es el nivel de gris de la imagen en el punto (x, y) y α es un parámetro de escalación dado por el usuario, como se observa a continuación (ver Apéndice A):

procedure TFormMain.Contraste (alfa: real);

El rango de valores que puede tomar alfa es: $\alpha > 1$. En la figura 3.2 (b) y (c) se observa el resultado de aplicar la fórmula (3.1-2) a la imagen de la figura 3.2 (a). Conforme α sea cada vez mayor que uno, la imagen se oscurece más.

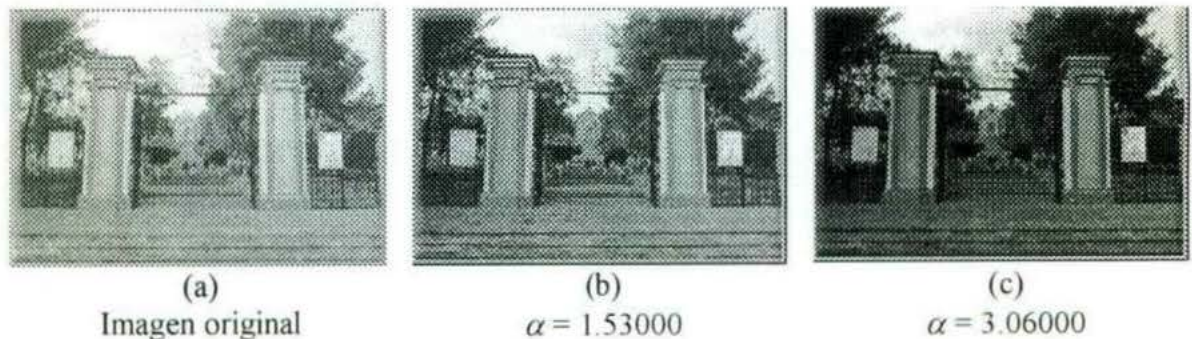


Figura 3.2 Transformación exponencial con diferentes parámetros de α .

3.2.- Histograma de una imagen

El histograma de una imagen digital con niveles de gris en el rango $[0, L - 1]$, donde L es el número total de niveles de gris, es una función discreta de la forma $p(r_k) = n_k/n$, donde r_k es

el k -ésimo nivel de gris, n_k es el número de píxeles de la imagen con ese nivel de gris, n es el número total de píxeles de la imagen y $k = 0, 1, 2, \dots, L - 1$.

De forma general, un histograma de una imagen es la distribución de cada nivel de intensidad dentro de la imagen, es decir nos da un estimado de la probabilidad de ocurrencia de cada nivel de gris, o en otras palabras es el número de píxeles por nivel de intensidad. De esta forma, se puede decir que $p(r_k)$ da una idea del valor de la probabilidad de que aparezca el nivel de gris r_k . La representación gráfica de esta función para todos los valores de k proporciona una descripción global de la apariencia de una imagen, proporcionando información acerca del brillo, contraste y rango dinámico de una imagen. El gráfico es bidimensional, donde la abscisa o eje X es el valor de los niveles de gris y la ordenada o eje Y es el número de píxeles que tienen un nivel de gris; p.e., si el número de niveles de gris de una imagen es $L = 256$, por lo que $k = [0, 1, \dots, 255]$, y su tamaño es 256×256 , es decir $n = 65536$. La construcción del histograma consiste en contar de la imagen original cuántos píxeles tienen un valor de gris de 0, cuántos un valor de gris de 1, y así sucesivamente; si se encontró por ejemplo, que la imagen digital tiene $n_0 = 9020$ píxeles con valor de gris 0, entonces la probabilidad de que en la imagen existan píxeles con valor de $r_k = 0$ es: $p(0) = 9020/65536 = 0.1376343$, esto se repite para todos los valores de k . En la siguiente figura 3.3 se puede apreciar el ejemplo de un histograma típico.

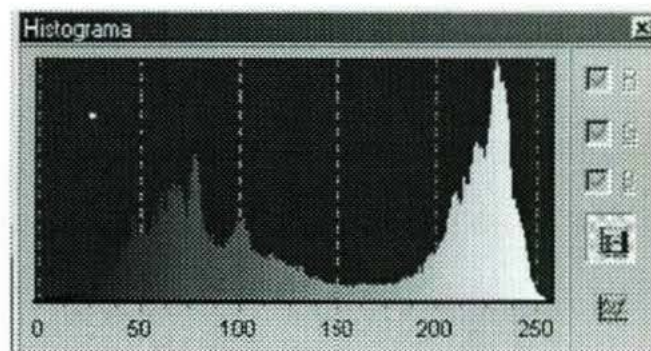


Figura 3.3 Ejemplo de la gráfica de un histograma típico.

Las características de un histograma son:

1. El histograma muestra la información en cuanto a la frecuencia de ocurrencia de píxeles que tienen varios niveles de gris, pero no tiene información de la localización o distribución de los píxeles en la imagen.
2. Al cambiar la orientación de los objetos, el histograma no presenta cambios. En esencia el histograma es una medida integrada por los valores de escalas de gris sobre la forma completa.
3. Un histograma es único para una imagen específica.
4. Un histograma proporciona información acerca de rangos disponibles de gris que presenta una imagen en su forma y en su contorno.
5. La forma del histograma indica las posibilidades de mejora del contraste de la imagen (ver figura 3.4).

3.2.1.- Métodos basados en el histograma

Los métodos de realce basados en el histograma, utilizan la información contenida en el histograma de la imagen digital, para efectuar la operación de transformación, de manera que cada nivel de gris de entrada le corresponda un nivel de gris de salida modificado por un operador de transformación, con la finalidad de que la imagen mejore en su visualización.

Los efectos de la transformación eventualmente redundan en una mejoría del contraste, en todo caso pueden conducir a un empleo óptimo del rango dinámico de la imagen.

Cuando el rango de valores del histograma es el máximo posible y cuando la masa de puntos esta repartida entre los posibles valores [0,255] de forma que el histograma se asemeje a la forma de una campana entonces el sistema visual humano consigue una máxima respuesta en su apreciación de la imagen¹².

La forma gráfica del histograma describe la densidad de niveles de gris en la imagen. Por ejemplo, la figura 3.4(a) muestra que el histograma se concentra en los niveles bajos de gris, entonces la imagen es relativamente oscura. Sucede justo lo contrario en el caso de la figura 3.4(b), debido a que el histograma presenta una concentración de píxeles con un nivel alto de gris. El histograma mostrado en la figura 3.4(c) tiene un perfil estrecho, lo que significa que el rango dinámico es pequeño y que por tanto la imagen tiene bajo contraste. Como todos los niveles de gris caen en la zona central de la escala de grises, la imagen aparecería como gris turbio. Finalmente, la figura 3.4(d) muestra un histograma con una dispersión considerable, que corresponde a una imagen de alto contraste.

¹² PÉREZ, N. B. (1995). *Fundamentos del tratamiento de imágenes. Visión por computador*.
<http://www.uco.es/informa/local/cordoba/ccordoba/ccordoba.html>

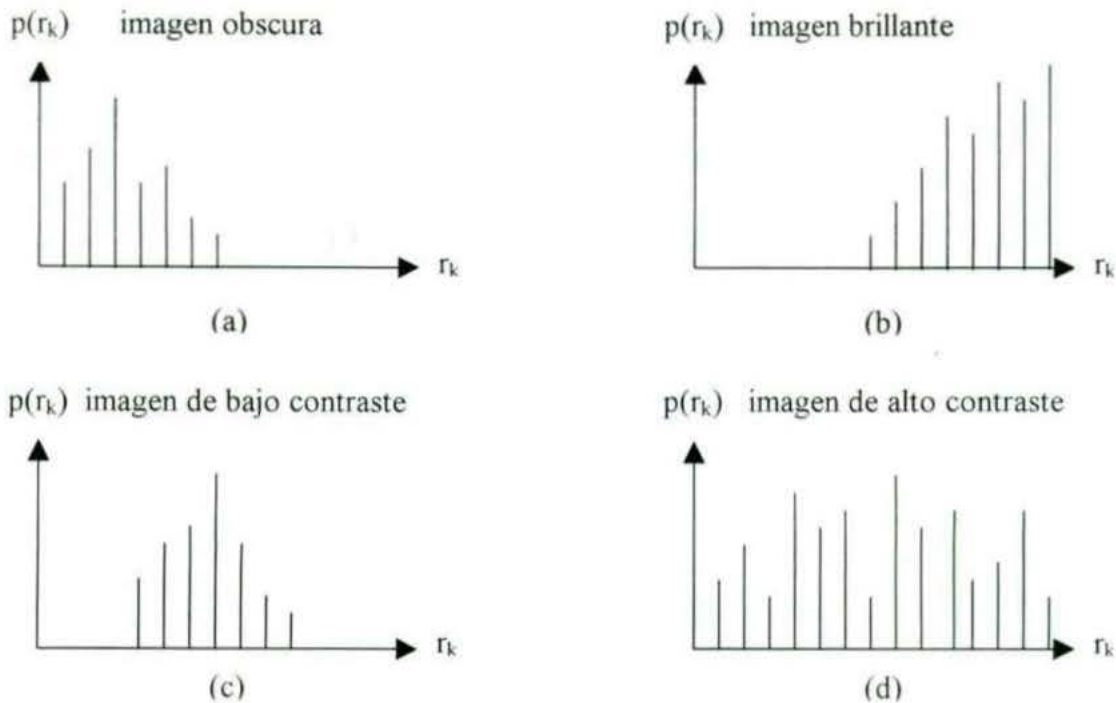


Figura 3.4 Histogramas correspondientes a cuatro tipos básicos de imágenes.

Por otra parte, una forma más de resaltar una imagen es aplicar una transformación que modifique su histograma en una forma predeterminada y deseada. Para ello existe una transformación conocida como igualación del histograma, la cual se explicará en la siguiente sección.

3.3.- Igualación del histograma

La igualación de histogramas es una técnica de transformación no lineal que pretende generar una imagen cuyo histograma es uniforme (i.e., una imagen donde cada nivel de gris ocurre en un número igual de veces: de aquí el término *igualación*). El efecto es incrementar el rango dinámico de los niveles de gris, en donde los niveles de gris i de una imagen de entrada con histograma $h(i)$ son modificados mediante la transformación $g = T(i)$, con el objeto de producir una imagen de salida que tenga un histograma $h(g)$. Generalmente la transformación T es una función monótona creciente. De esta forma se considera la apariencia visual de la imagen, resaltando los detalles tanto de la región oscura como de la clara de la imagen.

La igualación del histograma, es un procedimiento que redistribuye los niveles de gris del histograma de una imagen, utilizando de la mejor manera posible, el rango disponible de niveles de gris en la imagen de salida. En la figura 3.5, se observa un ejemplo de igualación del histograma. El procedimiento de llamada es el siguiente:

procedure TformMain.IgualarHistograma (L: byte);

donde L es el número de niveles de gris que se desea que se distribuya a la imagen, para más detalles de su funcionamiento ver el Apéndice A.

Un caso particular de igualación del histograma que es utilizado, corresponde a considerar que el histograma de la imagen de salida es constante. Un algoritmo sencillo para implantar la igualación aproximada en este caso es el siguiente¹³:

a) Se calcula el valor promedio de la ocurrencia de los niveles de gris presentes en el histograma de la imagen de entrada como:

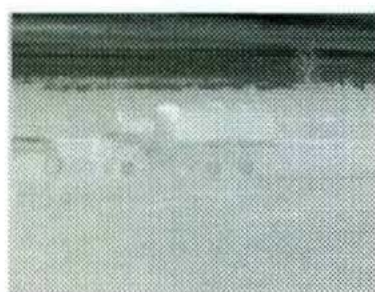
$$\bar{h} = \frac{1}{L} \sum_{i=0}^{L-1} h(i) \quad (3.3-1)$$

b) Comenzando a partir del nivel más bajo de gris presente en la imagen original, se va sumando el número de ocurrencias correspondiente a cada nivel de gris, hasta que el total esté lo más próximo posible al promedio calculado en a).

c) Se obtiene el promedio correspondiente a los niveles de gris, cuyos números de pixeles fueron sumados en b).

d) Los pixeles de la imagen cuyos valores correspondan a los considerados para realizar la sumatoria en b), son sustituidos por el pixel promedio obtenido en c).

e) Comenzando a partir del nivel de gris subsiguiente a los considerados en b) para definir la banda anterior, se repite el proceso de sumatoria así como los pasos siguientes hasta cubrir todo el rango de niveles de gris.



(a)
Imagen original



(b)
Imagen ecualizada

Figura 3.5 Resultado de aplicar a la imagen (a) la igualación del histograma (b).

3.4.- Negativo de una imagen

El negativo de una imagen consiste en calcular su imagen invertida, ya que aspectos que resultan difíciles de percibir pueden estar más claros en la imagen invertida. La idea es invertir el orden de blanco y negro, de forma que la intensidad de la imagen de salida disminuya conforme la intensidad de la imagen de entrada aumente, es decir, se pretende

¹³ MEDINA, R.
<http://www.ing.ula.ve/~rmedina/tecnicas/>

que las zonas claras se hagan oscuras y las oscuras claras. En la figura 3.6, se muestra la transformación que tiene tal efecto. La negación de la imagen, puede resultar de utilidad cuando se quiere apreciar los detalles en las porciones brillantes de una imagen, pues el ojo humano, es más capaz de discernir los detalles en áreas oscuras de una imagen que en las áreas más brillantes¹⁴.

La expresión que calcula la inversa de una imagen de niveles de grises es la siguiente:

$$g(x,y) = L - f(x,y) \quad (3.4-1)$$

donde $L = 255$.

Este tipo de transformación es utilizado ampliamente en aplicaciones medicas, como lo son imágenes de radiografías¹⁵.

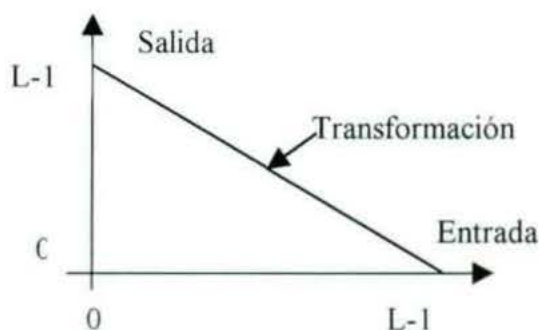


Figura 3.6 Transformación utilizada para obtener el negativo de una imagen.

En la figura 3.7 se muestra un ejemplo de la aplicación de la operación de negativo definida en la expresión de la ecuación (3.1-4). El resultado de aplicar esta transformación a la imagen oscura 3.7 (a), se observa claramente en 3.7 (b), donde el caballo y la mujer se aprecian mejor en la inversa de la imagen original.

¹⁴ MEDINA, R.

<http://www.ing.ula.ve/~rmedina/tecnicas/>

¹⁵ PÉREZ, N. B. (1995). *Fundamentos del tratamiento de imágenes. Visión por computador*.

<http://www.uco.es/informa/local/cordoba/ccordoba/ccordoba.html>



(a)
Imagen original



(b)
Negativo de la imagen

Figura 3.7 Resultado de la aplicación del negativo a una imagen.

3.5.- Binarización

La binarización es una técnica que permite convertir imágenes con niveles de gris, a través de una transformación no-lineal, en una imagen binaria (blanco y negro). De acuerdo a tal técnica, los valores de pixel en la imagen de entrada que son menores a un cierto umbral especificado previamente, son convertidos a negro, cuyo valor es 0, mientras que los pixeles con valores mayores al umbral, son convertidos a blanco, cuyo valor es 255. En la figura 3.8 (a) se muestra la transformación que permite realizar la binarización. En algunas ocasiones se desea realizar una binarización tal que a una banda especificada por dos umbrales, se les asigne el color blanco, mientras que los pixeles de la imagen de entrada cuyos valores están fuera de la banda especificada, se les asigne el color negro. Esta transformación se muestra en la figura 3.8 (b). En las figuras 3.9 y 3.10 se muestran los ejemplos de aplicar binarización y binarización de una banda respectivamente.

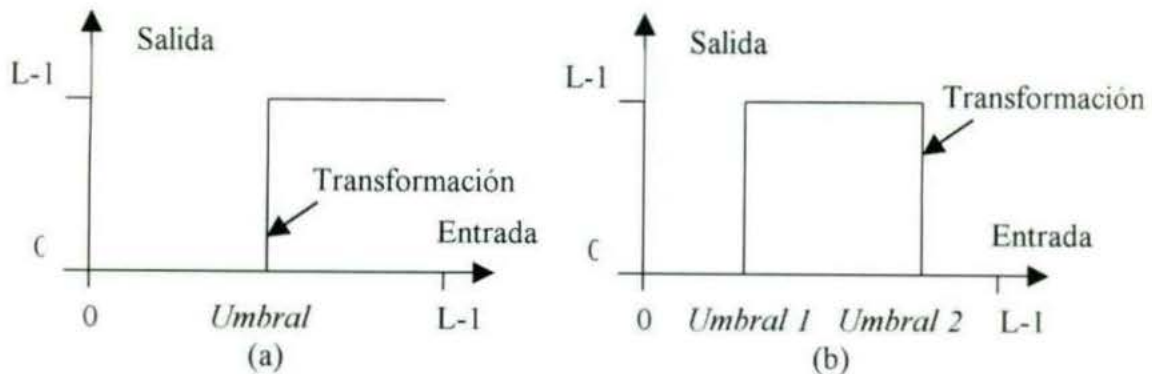


Figura 3.8 Transformaciones utilizadas para binarizar una imagen. a) transformación para la binarización; b) binarización de una banda.



Figura 3.9 Aplicación de un umbral de 200 (imagen (b)) a la imagen en (a).

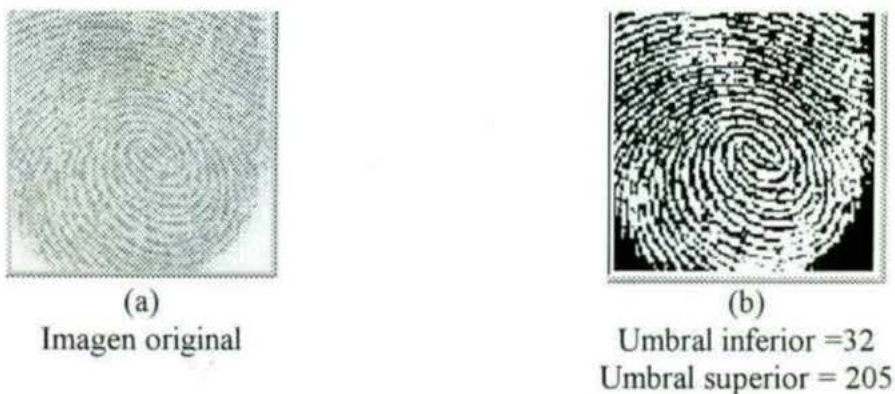


Figura 3.10 Resultado de la aplicación de un umbral por rangos a la imagen (a), con umbral inferior de 32 y un superior de 205.

3.6.- Operaciones algebraicas entre dos imágenes

Sean $f(x, y)$ y $g(x, y)$ las imágenes de entrada y $h(x, y)$ la imagen de salida. Las operaciones aritméticas entre las dos imágenes de entrada se realizan con el valor del pixel contenido en la posición (x, y) de la primera imagen con el valor del pixel en la segunda imagen en la posición (x, y) . El resultado de la función de transformación es un nuevo valor para el pixel en la misma posición en la imagen de salida.

3.6.1.- Adición de imágenes

Consiste en sumar una primer imagen a una segunda imagen, pixel por pixel y el resultado quedará en una tercera imagen. La suma entre dos imágenes $f(x, y)$ y $g(x, y)$ se expresa de la siguiente forma:

$$h(x, y) = f(x, y) + g(x, y) \quad (3.6-1)$$

Esta operación puede ser utilizada para reducir los efectos de ruido en los datos mediante el promediado de imágenes.

Un ejemplo de esta operación se muestra en la figura 3.11, en la cual se puede observar que al sumar dos imágenes idénticas se obtiene una imagen con el doble de brillo. Un aspecto importante que se debe considerar al aplicar esta operación es que la suma puede originar que un pixel tenga un valor superior de 255 y hasta 510, lo cual no debe ser posible pues el máximo valor es 255.

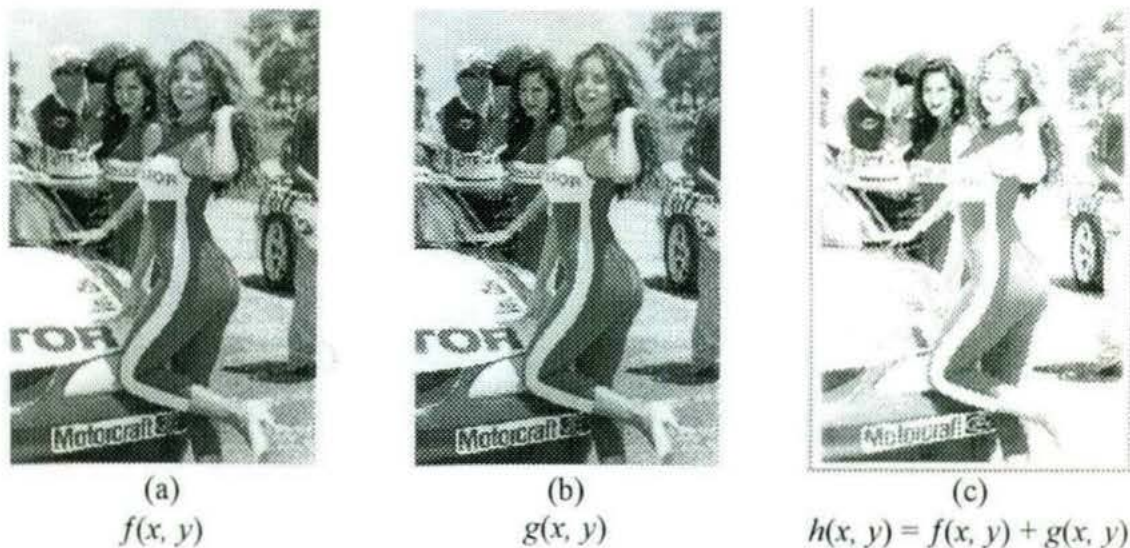


Figura 3.11 Suma de dos imágenes iguales, produce una imagen idéntica con dos veces el brillo.

3.6.2.- Sustracción de imágenes

La sustracción entre dos imágenes $f(x, y)$ y $g(x, y)$, se expresa de la siguiente forma:

$$h(x, y) = f(x, y) - g(x, y) \quad (3.6-2)$$

Una aplicación en donde se emplea esta operación es, cuando se quiere encontrar los contornos de una imagen u objeto. Se hace la sustracción de una imagen original dilatada contra la imagen original, resaltando las partes que son diferentes, es decir, su contorno. Un ejemplo de este tipo de aplicación se muestra en la figura 3.12.

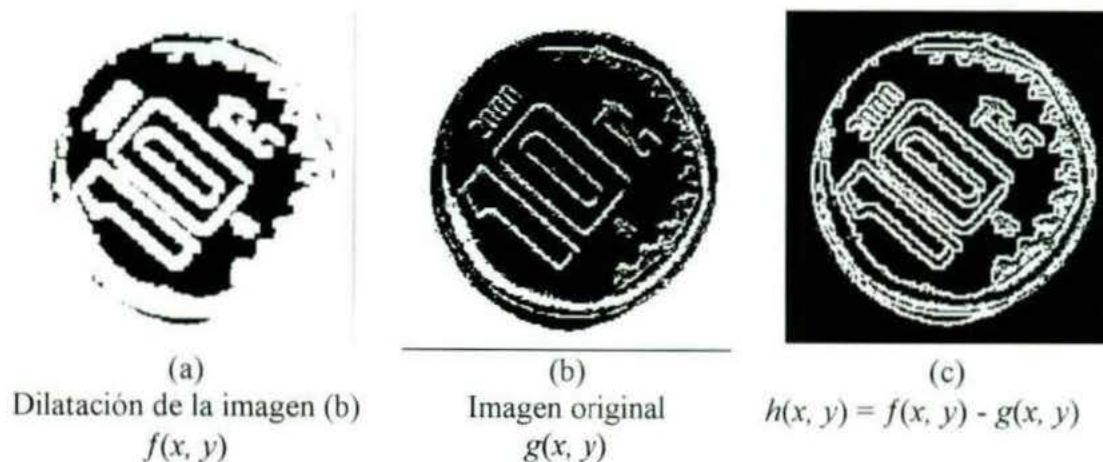


Figura 3.12 Resultado de la aplicación de la sustracción de imágenes.

3.6.3.- Multiplicación y división de imágenes

Uno de los principales empleos de la operación de multiplicación de imágenes consiste en corregir las variaciones de los niveles de gris que se deben a una iluminación no uniforme o a la respuesta desigual del sensor empleado para captar la imagen¹⁶.

La operación de multiplicación se expresa de la siguiente forma:

$$h(x, y) = f(x, y) g(x, y) \quad (3.6-3)$$

La operación de división de imágenes no es muy utilizado en el procesamiento de imágenes, sin embargo, aquí es considerado como una opción mas que puede ser aplicado a imágenes.

$$h(x, y) = f(x, y) / g(x, y) \quad (3.6-4)$$

¹⁶ GONZÁLEZ, R. C. (1992). *Digital image processing*. Addison-Wesley Publishing Company.

CAPÍTULO 4

PROCESAMIENTO ESPACIAL

El término *dominio espacial* se refiere al conjunto de píxeles que componen una imagen y las técnicas en el procesamiento espacial son procedimientos que operan directamente sobre los píxeles. La representación de las operaciones de procesamiento de una imagen digital en el dominio espacial es la siguiente:

$$g(x, y) = T[f(x, y)] \quad (4.1)$$

donde $f(x, y)$ es la imagen de entrada, $g(x, y)$ es la imagen procesada y T es un operador que actúa sobre f , definido en algún entorno de (x, y) .

Lo que se pretende, es obtener g en un punto (x, y) a partir de los valores de f en un entorno predefinido de (x, y) .

4.1.- Procesamiento basado en una región de la imagen

Las técnicas de procesamiento basadas en una región de la imagen, pretenden aplicar operadores de transformación a cada píxel de la imagen de entrada, realizando un barrido píxel por píxel, según las filas y columnas (desplazándose de izquierda a derecha y de arriba hacia abajo), considerando el grupo de píxeles definidos alrededor del píxel (x, y) a modificar. Para ello se utiliza un arreglo bidimensional, donde el número de filas y de columnas es un valor impar (generalmente de 3×3). A este arreglo se le conoce como máscara, ventana o filtro y es centrada en el píxel (x, y) , por tanto el nuevo valor del píxel en la misma posición (x, y) en la imagen de salida, dependerá del operador que se aplique al conjunto de píxeles definidos por la ventana alrededor del píxel (x, y) . En la figura 4.1 se muestra una subárea de la imagen, definida por una ventana de 3×3 , centrada en (x, y) .

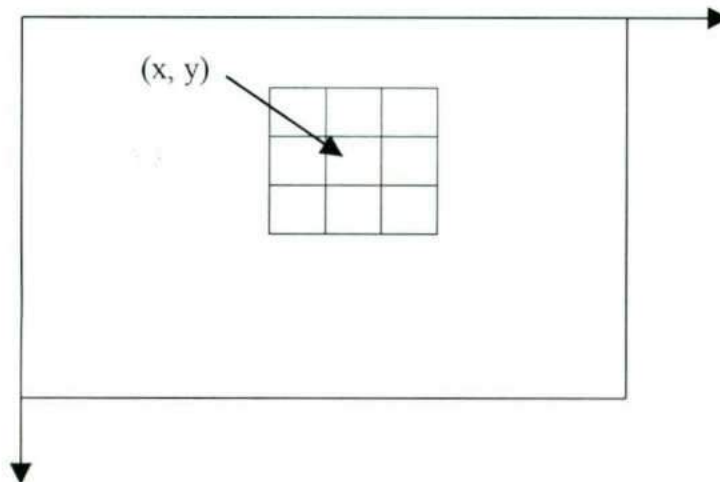


Figura 4.1 Una ventana de 3×3 alrededor de un punto (x, y) de una imagen.

Las técnicas de procesamiento basadas en una región de la imagen, usan filtros que atenúan o realzan ciertos componentes de la imagen.

Los métodos del procesamiento espacial pueden clasificarse en lineales y no lineales. Los filtros espaciales lineales, utilizan máscaras con coeficientes definidos; mientras que en los filtros espaciales no lineales sus operaciones están basadas directamente en los valores de los píxeles, sin usar coeficientes como en los filtros lineales.(explicar adecuadamente)

La técnica de convolución se utiliza dentro de la aplicación de los filtros lineales (en esta clasificación se encuentran los filtros de pasa-bajas, pasa-altas, resaltamiento de bordes, resaltamiento de líneas, gradientes, laplacianos, etc.); mientras que un filtro no lineal es p.e. la mediana. En la implantación de los métodos anteriores se sigue el siguiente procedimiento¹⁷:

- a) Se realiza una sola pasada sobre la imagen de entrada realizando un barrido pixel por pixel, según las filas y columnas.
- b) Cada pixel de la imagen de entrada es procesado, considerando una vecindad del mismo utilizando un algoritmo apropiado.
- c) El nuevo valor del pixel, obtenido de acuerdo a lo especificado en b), es ubicado en la imagen de salida, ocupando la misma posición que ocupaba en la imagen de entrada.

4.1.1.- Convolución

El proceso de convolución es el siguiente, para un pixel dado dentro de la imagen de entrada $f(x, y)$, cada pixel de la vecindad es multiplicado por el coeficiente del pixel correspondiente en la máscara de convolución, así mismo cada uno de estos productos es sumado, de manera que el nuevo valor del pixel en la imagen de salida $g(x, y)$ estará dado por la suma de todos estos productos, como se muestra en la figura 4.2. El procesamiento de toda la imagen se realiza desplazando la máscara y repitiendo para cada punto el mismo procedimiento (definir correstamente).

¹⁷ MEDINA, R.
<http://www.ing.ula.ve/~rmedina/tecnicas/>

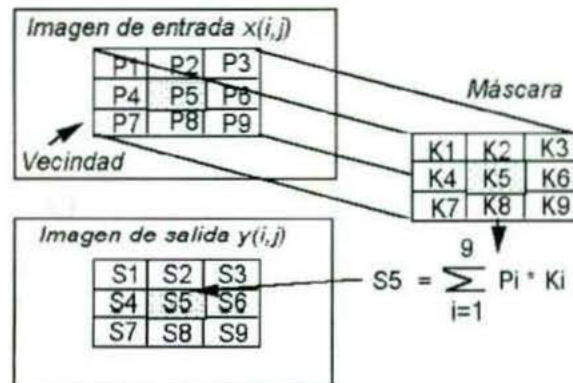


Figura 4.2 Ilustración del proceso de convolución con una máscara. Cada pixel en la imagen de salida es el resultado de la suma de los productos entre los pixeles de la máscara y los pixeles incluidos en la vecindad correspondiente en la imagen de entrada¹⁸.

El algoritmo de convolución tiene detalles en su implantación que deben ser resueltos, como por ejemplo, el procesamiento de los pixeles localizados en los bordes de la imagen, debido a que en tal caso, la máscara no esta superpuesta sobre 9 pixeles de la imagen de entrada.

Lo anterior puede tener las siguientes soluciones¹⁹:

- 1) Los pixeles de los bordes pueden ser ignorados durante el proceso de convolución y a la salida se deja el mismo valor que tenían en la imagen de entrada.
- 2) Los pixeles ubicados en los bordes se replican de modo que se tengan los pixeles necesarios para realizar el producto
- 3) Los pixeles de la primera columna pueden considerarse como vecinos de los de la última columna y los de la primera fila vecinos de la última fila.
- 4) Idem que 3) pero se promedian.

4.2.- Reducción de ruido

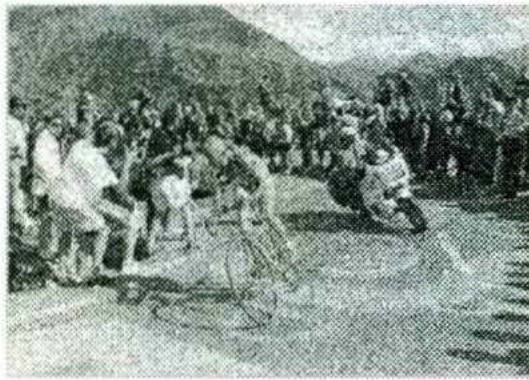
4.2.1.- Filtro del promedio

Consiste en reemplazar el nivel de gris de cada pixel de la imagen por el promedio de los niveles de gris en una vecindad de este pixel.

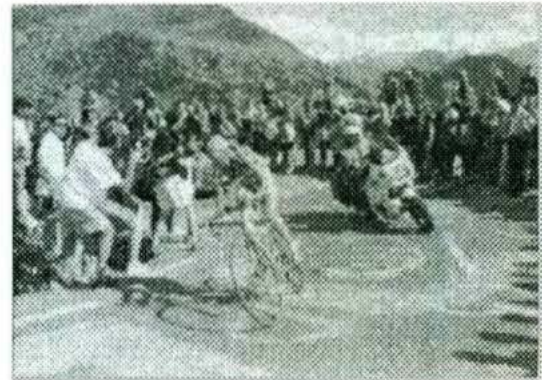
¹⁸ Figura tomada de: MEDINA, R.
<http://www.ing.ula.ve/~rmedina/tecnicas/>

¹⁹ MEDINA, R.
<http://www.ing.ula.ve/~rmedina/tecnicas/>

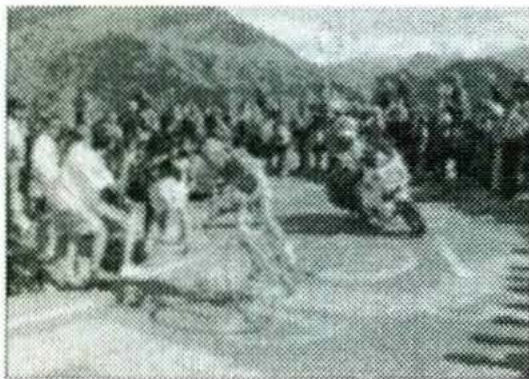
En la figura 4.3 se muestra la aplicación de este filtro utilizando ventanas de 3×3 , 5×5 y 7×7 , en donde se observa que a medida que se aumenta el tamaño de la ventana, se tiene mayor grado de borrosidad en la imagen reduciendo de esta manera el ruido que existe en la imagen. Un efecto negativo al aplicar este filtro es que se difuminan los contornos y otros detalles de forma.



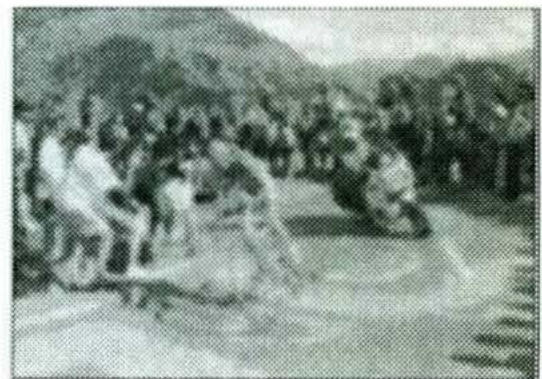
(a)
Imagen original



(b)
Promedio con ventana de 3×3



(c)
Promedio con ventana de 5×5



(d)
Promedio con ventana de 7×7

Figura 4.3 Resultado de aplicar el filtro del promedio a la imagen en (a): (b) aplicación del promedio utilizando una ventana de 3×3 ; (c) aplicación del promedio utilizando una ventana de 5×5 ; (d) aplicación del promedio utilizando una ventana de 7×7 .

4.2.2.- Filtro de mediana

El filtraje de mediana, es un procedimiento no-lineal que es utilizado para reducir el ruido, evitando en lo posible emborronamientos en las fronteras, ya que se desea preservar la agudeza de los bordes. En la figura 4.5 se ilustra con un ejemplo la técnica de mediana

Para aplicar el filtro de mediana a un pixel de la imagen, utilizando como ejemplo una ventana de 3×3 , la cual esté centrada en dicho pixel (x, y) , se requiere obtener los 9 valores de cada pixel delimitados por la ventana y clasificarlos en orden ascendente. Después se toma el valor central, cuya obtención es relativamente sencilla considerando que el número de elementos que contiene dicha máscara es impar, y se asigna al pixel

original (x, y) . En la figura 4.4 se muestra un ejemplo del proceso de la aplicación del filtro de mediana, en la cual se puede observar que el nuevo valor es 29 y sustituye al 60 que es relativamente mayor que los otros valores en la ventana y puede ser considerado como ruido.

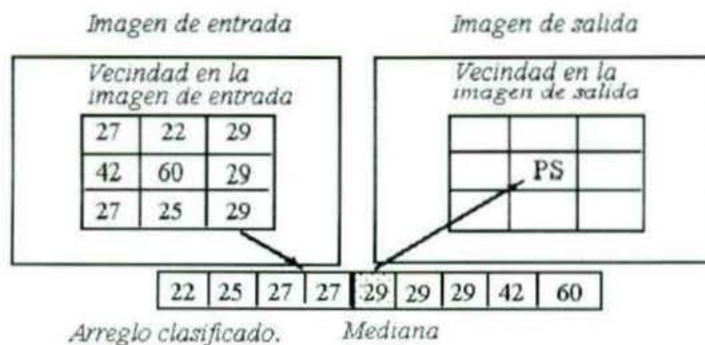


Figura 4.4 Ejemplo del proceso que se sigue para la aplicación del filtro de mediana²⁰.

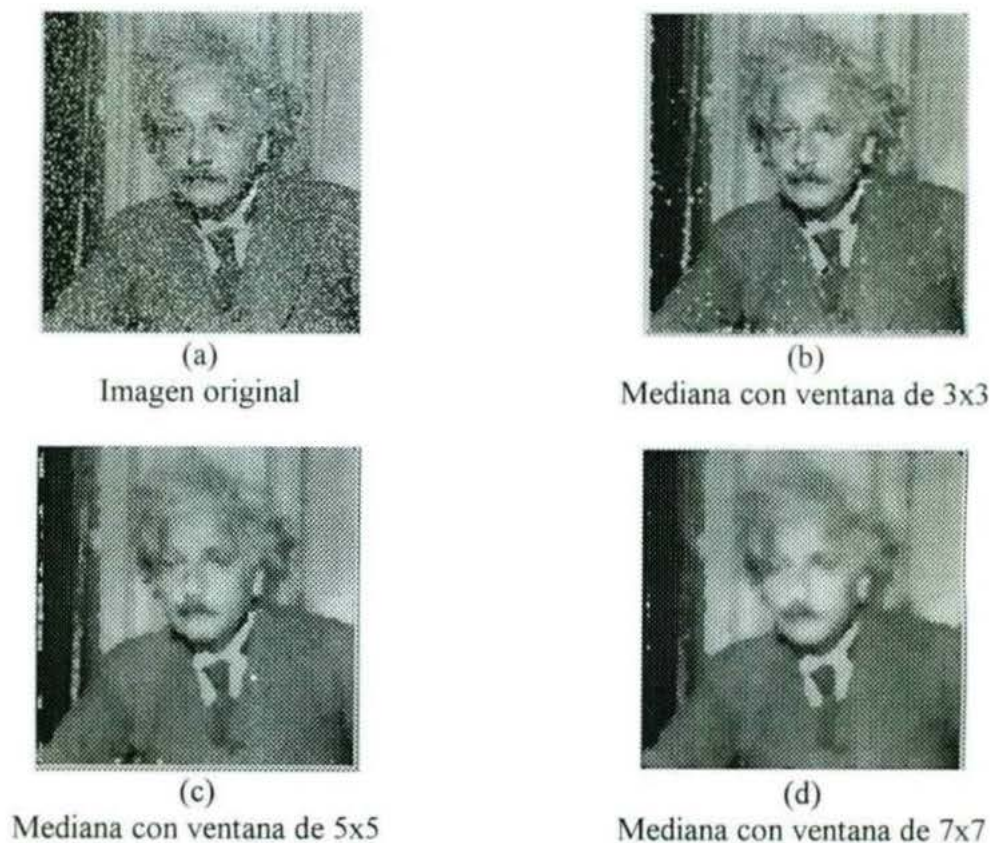


Figura 4.5 Resultado de la aplicación del filtro de la mediana con una vecindad de 3x3, 5x5 y 7x7, respectivamente.

²⁰ Figura tomada de: MEDINA, R.
<http://www.ing.ula.ve/~rmedina/tecnicas/>

4.2.3.- Filtros de pasa – bajas

Este tipo de filtros tienen como objetivo, atenuar los contenidos de alta frecuencia e inalterar los de baja. Su función principal es reducir el ruido presente en la imagen. En la figura 4.7 se muestra un ejemplo de aplicar un filtro de pasa-bajas.

Las características más importantes de los filtros de pasa bajas son:

1. La suma de los valores de cada máscara debe ser igual a la unidad.
2. Los coeficientes deben ser positivos.

A continuación se muestra la figura 4.6, en donde se tienen tres ventanas que representan filtros de pasa-bajas.

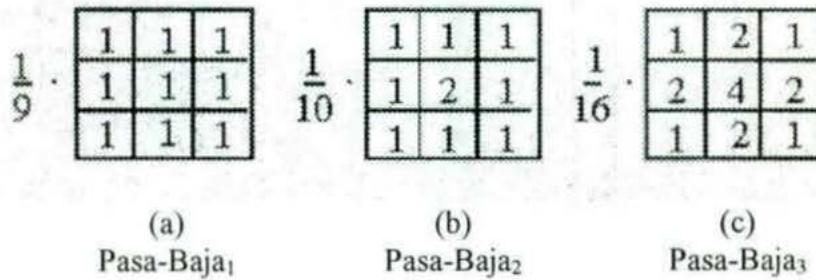


Figura 4.6 Tres ventanas que representan filtros de pasa-bajas.

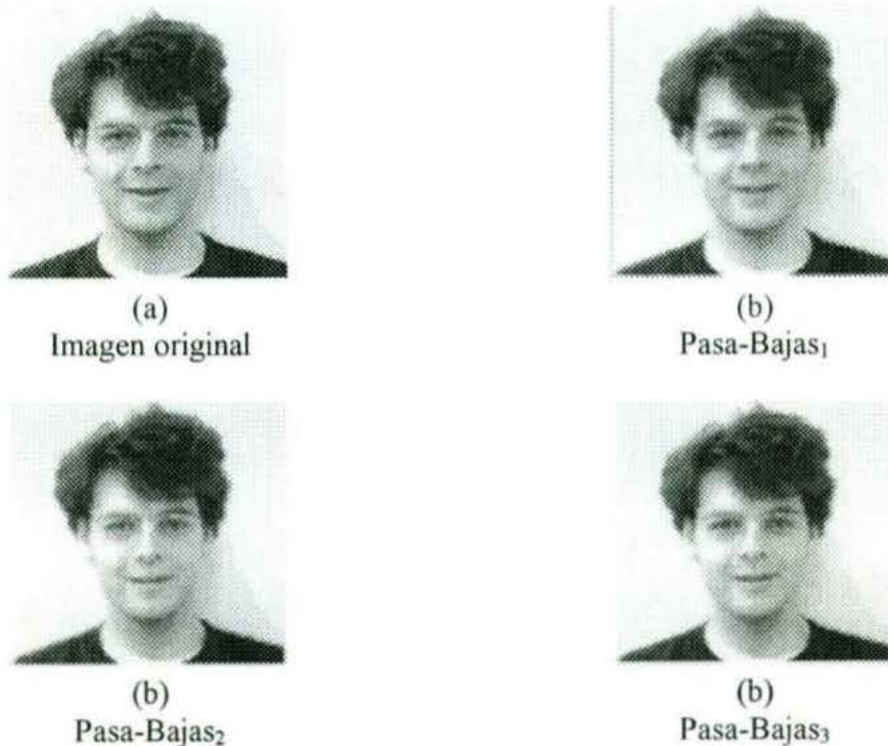


Figura 4.7 Se muestra una imagen y el resultado de aplicar los filtros de Pasa-Bajas definidos en la figura 4.6.

Un efecto negativo de este tipo de filtros es que puede generar borrosidad.

4.3.- Detección de contornos

Un contorno en una imagen, representa un cambio de la intensidad de los niveles de gris presentes en ella. El paso de nivel oscuro a uno brillante o el caso contrario, determinan un contorno.

Un borde de una imagen, es un límite o contorno, es decir, la discontinuidad en la intensidad de la imagen o su derivada. Las discontinuidades pueden ser de dos tipos: (1) de *salto* donde la intensidad cambia abruptamente de un valor en un lado de la discontinuidad a otro valor diferente en el lado opuesto (2) discontinuidades de *línea*, donde las intensidades cambian abruptamente volviendo en poco espacio a su valor inicial. En la figura 4.8 se muestran perfiles de bordes²¹.

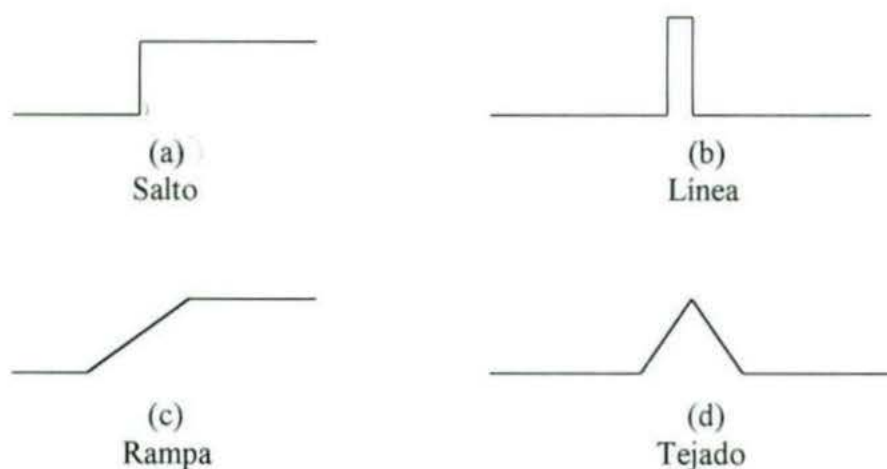


Figura 4.8. Perfiles de bordes. Modelos idealizados (a) y (b); modelos observados (c) y (d).

4.3.1.- Filtros de pasa - altas

Los filtros de pasa-altas, tienden a acentuar los contenidos de alta frecuencia de una imagen, utilizando niveles de gris más claros, lo contrario sucede con las zonas con componentes de baja frecuencia. Así pues, este tipo de filtro se utiliza para reforzar los bordes presentes en una imagen. En la figura 4.9 se presentan tres máscaras que representan filtros de pasa-altas.

Una de las características de las máscaras de filtros pasa-altas es que tienen valores positivos en el centro de la función y valores negativos a su alrededor. Este último, garantiza que en zonas donde la imagen tenga variaciones pequeñas o sea plana el resultado del filtrado será cero o un valor pequeño. Por tanto, este tipo de filtro hace que desaparezcan las bajas frecuencias definidas por dicho tipo de zonas, y por consiguiente

²¹ MOLINA, R. S. y CORTIJO, F. J. B. (1997).
http://www-etsi2.ugr.es/depar/ccia/robotica/indice_apuntes.html

disminuye el contraste en la imagen. En la figura 4.10 se muestra el resultado de aplicar las tres máscaras de pasa-altas de la figura 4.9 a la imagen de 4.10 (a).

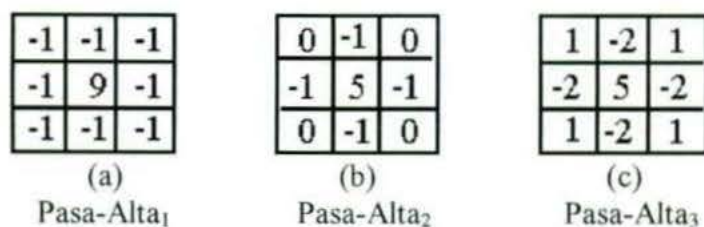


Figura 4.9 Tres ventanas que representan filtros de pasa-altas.

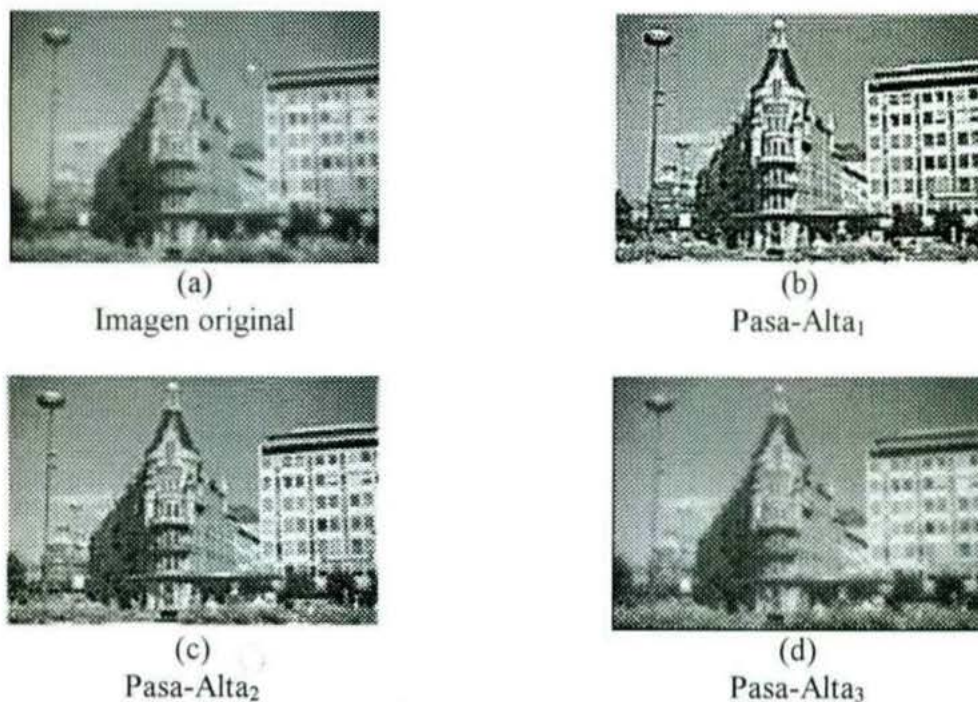


Figura 4.10 Las imágenes de los incisos: (b), (c) y (d), son el resultado de aplicar los tres filtros de pasa altas a la imagen del inciso (a).

Uno de los efectos negativos de estos filtros es que pueden acentuar el ruido de la imagen.

4.3.2.- Resaltamiento de bordes

La tendencia de estas técnicas es reforzar los bordes de los componentes de la escena, al mismo tiempo que atenúan los valores de nivel de gris de las regiones casi constantes en la imagen.

En la siguiente figura 4.11, se tienen ocho direcciones aplicadas en el resaltamiento de bordes.

-1	1	-1
-1	-2	-1
1	1	-1

Norte

1	-1	-1
1	-2	-1
1	1	1

Noreste

-1	-1	-1
1	-2	1
1	1	1

Este

-1	-1	1
-1	-2	1
1	1	1

Sureste

-1	1	1
-1	-2	1
-1	1	1

Sur

1	1	1
-1	-2	1
-1	-1	1

Suroeste

1	1	1
1	-2	1
-1	-1	-1

Oeste

1	1	1
1	-2	-1
1	-1	-1

Noroeste

Figura 4.11 Máscaras para el resaltamiento de bordes en sus ocho direcciones.

A continuación se presenta un ejemplo de la aplicación de cada una de las ocho máscaras presentadas en la figura 4.11; así como una omnidireccional.



(a)
Imagen original



(b)
Norte



(c)
Noreste



(d)
Este



(e)
Sureste



(f)
Sur



(g)
Suroeste



(h)
Oeste



(i)
Noroeste



(j)
Omnidireccional

Figura 4.12 Resultado de aplicar las ocho máscaras para el resaltamiento de bordes y una omnidireccional a la imagen del inciso (a).

4.3.3.- Resaltamiento de líneas

Los detectores de líneas presentan la forma de la línea que se desea encontrar. Consisten en una línea de valores altos rodeada de valores bajos. A continuación se muestran cuatro detectores de líneas:

-1	-1	-1
2	2	2
-1	-1	-1

Vertical

-1	2	-1
-1	2	-1
-1	2	-1

Horizontal

-1	-1	2
-1	2	-1
2	-1	-1

Diagonal Izq-Der.

2	-1	-1
-1	2	-1
-1	-1	2

Diagonal Der-Izq.

Figura 4.13 Cuatro máscaras utilizadas para detectar líneas.

Estos detectores de líneas funcionan adecuadamente cuando la línea a buscar tiene un ancho de un píxel. Si las líneas presentan anchos mayores, los detectores deben ajustarse a dichas medidas. En general, un detector de líneas sólo localizará aquellas que coincidan tanto en dirección como en anchura²². En la figura 4.14 se muestra el resultado de aplicar los cuatro detectores a la imagen 4.14 (a).

²² MOSQUERA, A. G. *Temas sobre procesado digital de imágenes*.
<http://www.paisvirtual.com/informatica/navegadores/toniomos/pdi/temas-pdi.htm>.



(a)
Imagen original



(b)
Vertical



(c)
Horizontal



(d)
Diagonal de Izquierda a derecha



(e)
Diagonal de derecha a izquierda

Figura 4.14 Resultado de aplicar los cuatro detectores de líneas a la imagen (a).

4.3.4.- Gradientes^{23,24,25,26.}

La detección general de bordes (zonas de transición en alguna propiedad) se basa en diferenciar la imagen, esto es, encontrar la derivada respecto a los ejes x y y . El método más común de cálculo de la derivada en un punto de la imagen es el cálculo de su gradiente. La mayoría de los detectores de borde están basados, de alguna forma, en la medida de la intensidad del gradiente en un punto de la imagen.

Se define el gradiente en cada punto (x, y) de la imagen $f(x, y)$ como un vector cuyas componentes son la derivada en el sentido del eje x y la derivada en el sentido del eje y . Su notación matemática es:

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix} \quad (4.3-1)$$

Al igual que con cualquier vector, se puede calcular su magnitud (la magnitud del gradiente da una idea de lo abrupto que es el cambio de valores) y su orientación (la orientación del gradiente indica la dirección en la cual se produce el cambio más abrupto). El gradiente de un vector indica la dirección de la máxima variación de $f(x, y)$ en (x, y) . Una importante cantidad de bordes es el módulo de este vector, al que generalmente se le hace referencia, para simplificar, como el gradiente, con la notación ∇f , donde:

$$|\nabla f| = [(\partial f/\partial x)^2 + (\partial f/\partial y)^2]^{1/2} \quad (4.3-2)$$

Esta cantidad es igual a la máxima variación de $f(x, y)$ por unidad de distancia en la dirección de ∇f .

Se puede aproximar el gradiente tomando la diferencia de valores contiguos en la imagen. Para una sección de 2×2 como se muestra en la figura 4.15, la representación sería:

$$\nabla f = [(I_{1,2} - I_{1,1})^2 + (I_{2,1} - I_{1,1})^2]^{1/2} \quad (4.3-3)$$

²³ MOSQUERA, A. G. *Temas sobre procesamiento digital de imágenes*.
<http://www.paisvirtual.com/informatica/navegadores/toniomos/pdi/temas-pdi.htm>

²⁴ Notas.ps. Disponible en:
<http://campus.mor.itesm.mx/~esucar/Vision/indice.html>

²⁵ MEDINA, R.
<http://www.ing.ula.ve/~rmedina/tecnicas/>

²⁶ SUCAR, E. S. (2000). *Procesamiento de imágenes y visión computacional*.
<http://www.ruv.itesm.mx/cursos/mcc/Cs-00-139/homedoc.htm>



Figura 4.15 Sección de 2x2 de una imagen.

En lugar de emplear cuadrados y raíces cuadradas, es posible obtener resultados similares empleando valores absolutos:

$$\nabla f = |I_{1,2} - I_{1,1}| + |I_{2,1} - I_{1,1}| \quad (4.3-4)$$

Otra opción es considerar las diferencias cruzadas:

$$\nabla f = [(I_{1,1} - I_{2,2})^2 + (I_{1,2} - I_{2,1})^2]^{1/2} \quad (4.3-5)$$

o con valores absolutos:

$$\nabla f = |I_{1,1} - I_{2,2}| + |I_{1,2} - I_{2,1}| \quad (4.3-6)$$

Podemos también considerar una sección de 3 x 3, como se muestra en la figura 4.16 y aproximar el gradiente de la siguiente forma:

$$\nabla f = |(I_{1,3} + I_{2,3} + I_{3,3}) - (I_{1,1} + I_{2,1} + I_{3,1})| + |(I_{3,1} + I_{3,2} + I_{3,3}) - (I_{1,1} + I_{1,2} + I_{1,3})| \quad (4.3-7)$$

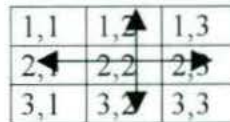


Figura 4.16 Sección de 3x3 de una imagen.

Las aproximaciones anteriores se pueden implementar como filtros espaciales (máscaras). Operadores como el de Roberts, Prewitt y Sobel, se implementan con dos máscaras: una para dx y otra para dy .

4.3.4.1.- Operadores de Roberts

Los dos operadores para las diferencias cruzadas de 2 x 2 son:



En la figura 4.17 (h), (i) y (j) se muestra el ejemplo de aplicar operadores de Roberts a una imagen.

4.3.4.2.- Operadores de Prewitt

Los operadores de Prewitt están basados en la estimación del módulo del gradiente y corresponden a las diferencias en secciones de 3x3. Los dos operadores, en la dirección del eje x y en la dirección del eje y son:

-1	-1	-1
0	0	0
1	1	1

Vertical

-1	0	1
-1	0	1
-1	0	1

Horizontal

La mayor parte de los bordes existentes en una imagen no presentan transiciones bruscas. Por el contrario, los bordes suelen ser zonas de transiciones suaves entre dos valores de intensidad. En este caso usualmente se obtiene, conforme se atraviesa el borde, una subida, un pico y un descenso en el valor del gradiente. Extraer el borde ideal supone la búsqueda de la curva óptima de la magnitud del gradiente. En la figura 4.17 (e), (f) y (g) se muestra el ejemplo de aplicar operadores de Prewitt a una imagen.

Los detectores de bordes Roberts y Prewitt tienden a amplificar el ruido en la imagen, ya que se usan muy pocos píxeles para aproximar el gradiente.

4.3.4.3.- Operadores de Sobel

Estos operadores pueden implementarse por las siguientes máscaras de convolución:

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

Vertical

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

Horizontal

La ventaja adicional que presentan estas máscaras sobre las anteriores es que además de estimar el valor del módulo del gradiente, producen un suavizamiento en la imagen que es beneficioso, dado el comportamiento ruidoso que presentan las estimaciones basadas en derivadas. En la figura 4.17 (b), (c) y (d) se muestra el ejemplo de aplicar operadores de Sobel a una imagen.

Este tipo de filtros dan como salida una nueva imagen cuyos valores son, en cada píxel, el módulo del gradiente en dicho punto de la imagen de entrada.



(a)
Imagen original



(b)
Sobel Horizontal



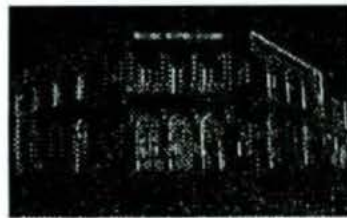
(c)
Sobel Vertical



(d)
Sobel Omnidireccional



(e)
Prewitt Horizontal



(f)
Prewitt Vertical



(g)
Prewitt Omnidireccional



(h)
Robert Horizontal



(i)
Robert Vertical



(j)
Robert Omnidireccional

Figura 4.17 Resultado de aplicar Operadores de Roberts, Prewitt y Sobel a una imagen.

4.3.5.- Laplaciano^{27,28,29,30.}

Los detectores de fronteras anteriores calculan el gradiente y si su valor está por encima de un umbral, se supone la existencia de un punto de borde. El resultado es demasiados puntos de borde. Una mejor aproximación sería encontrar sólo los puntos que tienen máximos locales en el gradiente y considerar estos los puntos de borde. Esto significa que en los puntos de borde existe un pico en la primera derivada y equivalentemente un cruce por cero en la segunda. Así pues los puntos de borde se detectan encontrando los cruces por cero de la segunda derivada en las intensidades de la imagen.

Existen otros operadores que intentan encontrar directamente esos picos de la magnitud del gradiente.

Considerar en primer lugar el caso uni-dimensional: Hallar el borde ideal es equivalente a hallar el punto en el cual la derivada es un máximo o un mínimo. Por otra parte, para localizar los máximos y mínimos de una función uni-dimensional se necesita, simplemente, localizar las posiciones en las cuales la primera derivada es cero. Por tanto, para localizar los bordes óptimos (máximos de la magnitud del gradiente) será necesario localizar los puntos en los cuales la segunda derivada es nula.

Cuando se aplican diferenciales a imágenes, los puntos con valores nulos de segunda derivada raramente coinciden exactamente en un pixel de la imagen (por problemas de muestreo). En cualquier caso es sencillo determinar su posición considerando los cruces por cero (parejas de pixeles vecinos en los cuales uno tiene valor positivo y el otro negativo).

Una propiedad muy interesante del análisis de cruces por cero de la segunda derivada es que proporcionan siempre caminos cerrados (excepto cuando dicho camino se extiende fuera de los límites de la imagen). Sin embargo, aparecen dos problemas típicos del método de cruces por cero: Pueden producirse bordes gruesos y los bordes son muy sensibles al ruido.

Para dos dimensiones hay una medida simple, similar a la magnitud del gradiente que mide la segunda derivada y recibe el nombre de Laplaciano. El Laplaciano de una imagen $f(x, y)$ en el punto (\bar{x}, \bar{y}) es el vector bi-dimensional

²⁷ MOSQUERA, A. G. *Temas sobre procesamiento digital de imágenes.*
<http://www.paisvirtual.com/informatica/navegadores/toniomos/pdi/temas-pdi.htm>.

²⁸ MOLINA, R. S. y CORTIJO, F. J. B. (1997).
http://www-etsi2.ugr.es/depar/ccia/robotica/indice_apuntes.html

²⁹ MEDINA, R.
<http://www.ing.ula.ve/~rmedina/tecnicas/>

³⁰ SUCAR, E. S. (2000). *Procesamiento de imágenes y visión computacional.*
<http://www.ruv.itesm.mx/cursos/mcc/Cs-00-139/homedoc.htm>

$$\nabla^2 f = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} \\ \frac{\partial^2 f}{\partial y^2} \end{pmatrix} \quad (4.3-8)$$

Una propiedad muy interesante del Laplaciano es su invarianza frente a rotaciones. En virtud de esta propiedad se pueden elegir cualquier pareja de direcciones con la única restricción de que sean ortogonales.

Se pueden localizar bordes en la imagen sin más que localizar los puntos de cruce por cero en el Laplaciano de la imagen a analizar.

Para imágenes discretas, se puede aproximar el Laplaciano en base a diferencias de la primera derivada.

$$\frac{\partial f(x, y)}{\partial x^2} = [f(x, y) - f(x - 1, y)] - [f(x + 1, y) - f(x, y)] \quad (4.3-9)$$

$$\frac{\partial f(x, y)}{\partial y^2} = [f(x, y) - f(x, y - 1)] - [f(x, y + 1) - f(x, y)] \quad (4.3-10)$$

A partir de las cuales se obtiene:

$$\nabla^2 f(x, y) = 4f(x, y) - f(x - 1, y) - f(x, y - 1) - f(x + 1, y) - f(x, y + 1) \quad (4.3-11)$$

La ecuación anterior es equivalente a la convolución de la siguiente máscara:

$$\nabla^2 \approx \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Algunos otros operadores son:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

El requisito básico para la definición de una ventana de un operador Laplaciano digital es que el coeficiente asociado con el pixel central sea positivo y los coeficientes asociados con los pixeles exteriores sean negativos. Como el operador Laplaciano es una derivada, la suma de los coeficientes debe ser cero. Por lo tanto, la respuesta es cero siempre que el punto en cuestión y sus vecinos tengan el mismo valor. En la figura 4.18 se muestra un ejemplo de aplicar el laplaciano a una imagen.

El análisis de cruces por cero de este operador puede utilizarse para la localización de bordes. No hay que olvidar que este operador es muy sensible al ruido. Como norma

general cuanto más alta es la derivada utilizada más sensible al ruido es el operador que la implementa.

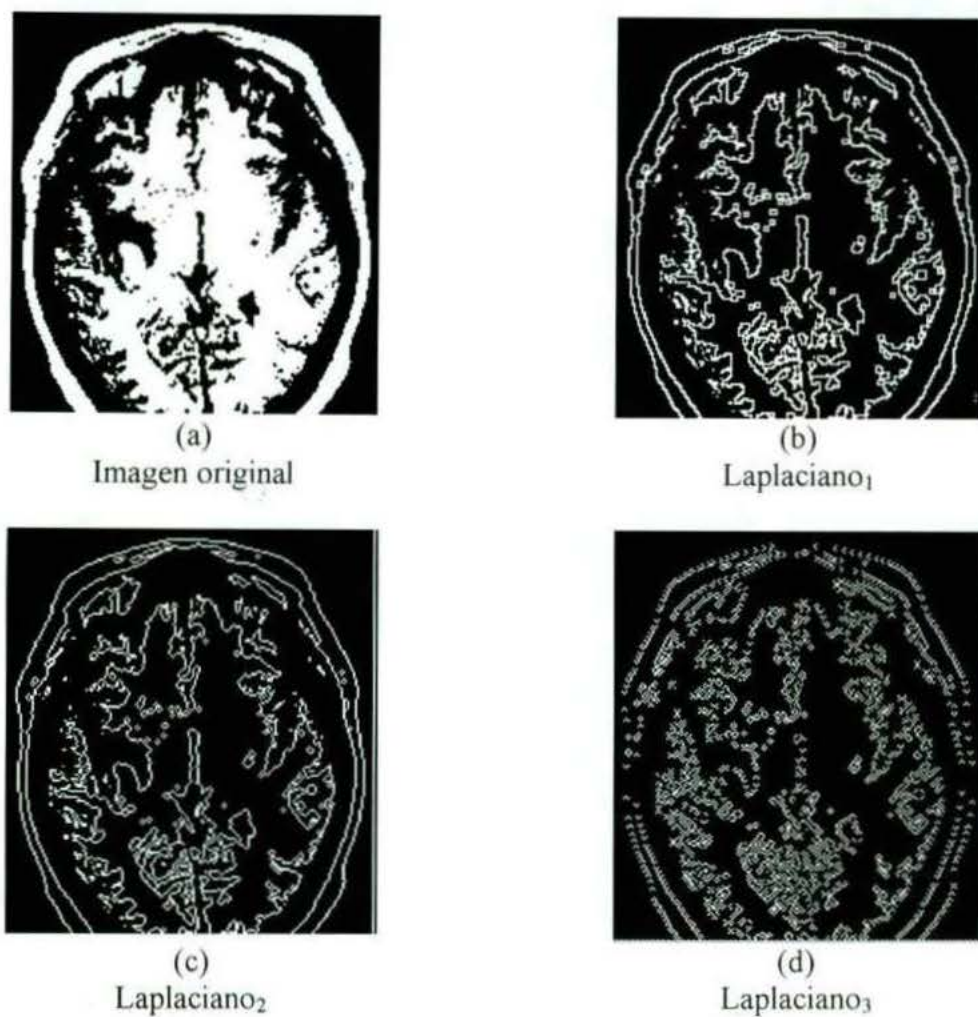


Figura 4.18 Resultado de aplicar tres máscaras del operador laplaciano a una imagen del cerebro.

**CAPÍTULO 5
METODOLOGÍA DEL
SOFTWARE DE
PROCESAMIENTO DIGITAL
DE IMÁGENES**

5.1 Descripción y justificación de la plataforma utilizada

La plataforma sobre la que se desarrolló el software fue el ambiente de desarrollo visual Delphi. El entorno de Delphi permite crear rápidamente herramientas que pueden compartir las ventajas de Windows (entorno visual, multitarea, multimedia, etc.), el poder del lenguaje Pascal, con su poder matemático y de estructuras de datos, las utilerías desarrolladas por Borland en sus compiladores, la cual le permite darle a Pascal entre otras cosas, la Orientación a Objetos. Por otra parte, un beneficio adicional de Delphi es que permite reutilizar código que haya sido desarrollado anteriormente bajo plataforma DOS.

A continuación se explica el por qué no se usaron otras plataformas:

DOS: La desventaja principal que hizo descartar a DOS como plataforma de desarrollo fue (como siempre), el viejo problema del límite de 640 Kb de memoria, con sus bloques de datos de 64 Kb que puede manejar.

Visual C++: Dentro de los entornos visuales bajo Windows, Visual C++, al heredar al lenguaje C++ cuenta con un gran poder, pero desafortunadamente, Visual C++ está todavía lejos de dar las facilidades de programación que ofrecen Visual BASIC y Delphi.

Visual BASIC: A diferencia de Visual C++, es un ambiente de desarrollo más sencillo de usar, pero desafortunadamente, el manejo de las estructuras de datos no es tan natural como en otras plataformas. Además, que como Delphi y Visual C++, no tienen inmerso a un lenguaje nativo del paradigma de Objetos, pero, a diferencia de ellos, Visual BASIC no puede crear nuevos objetos, sino que solamente puede derivar "nuevos" objetos a partir de los ya existentes. Otro punto en contra es que Visual BASIC no puede crear programas ejecutables independientes, sino que dependen de infinidad de utilerías externas que necesitan estar presentes en la máquina en que se vaya a ejecutar la aplicación.

Limitaciones de Delphi: La principal limitación que tiene Delphi para el desarrollo de este Software radica en su componente Image, el cual permite abrir y desplegar de manera sencilla archivos BMP, WMF e ICO, pero que solamente permite desplegar 4 tonos de gris cuando se transforma la paleta de la imagen a escalas de gris, necesitando en este proyecto 256. Para salvar este obstáculo se requirió la adquisición de utilerías externas, como es el FastGraph for Windows.

5.2. Características del FastGraph for Windows

El FastGraph for Windows permite una gran velocidad en el manejo de imágenes, pues, para realizar un desplegado no utiliza el GDI de Windows. El FastGraph for Windows contiene librerías para compiladores de 16 y 32 bits, como son: Borland C++, Symantec C++, Visual C++, Watcom C/C++, Delphi, y MSF PowerStation. Otra de las ventajas es que permite un fácil manejo de paletas de 256 colores, por lo que el manejar 256 tonos de gris es algo natural³¹.

³¹ TED, G. (1995). Software: *Fastgraph. Programmer's graphics library for Windows*.

5.3 Tipos de procesamiento empleadas en el software

Se emplearon dos métodos de procesamiento de imágenes, los cuales se describen a continuación:

- **Procesamiento Puntual:** Se altera el valor de los niveles de gris de los píxeles de la imagen basándose ya sea, en sus valores originales, o bien en la colocación que estos tienen dentro de la propia imagen.
- **Procesamiento Espacial:** Este método de procesamiento modifica los valores de los niveles de gris de los píxeles de la imagen, dependiendo de los valores que tienen sus vecinos.

Así mismo, en la Tabla 5.1 se muestra las diferentes opciones con las que cuenta este software.

Tabla 5.1 Tipos de procesamiento empleadas por el software de PDI.

Procesamiento	Opciones
Puntual	<ul style="list-style-type: none"> • Resaltamiento de la imagen <ul style="list-style-type: none"> • Igualación de Histograma • Logarítmica • Exponencial • Negativo • Operación entre dos imágenes (Adición, Sustracción, Multiplicación y División) • Umbrales <ul style="list-style-type: none"> • Umbral • Umbral de Rangos
Espacial	<ul style="list-style-type: none"> • Reducción de Ruido <ul style="list-style-type: none"> • Promedio. • Mediana. • Pasa-Bajas (contiene tres tipos diferentes) • Detección de Bordes <ul style="list-style-type: none"> • Pasa-Altas (contiene tres tipos diferentes) • Resaltamiento de Bordes (N, NE, E,..., NO, Omnidireccional) • Laplaciano (contiene tres tipos diferentes) • Resaltamiento de Línea (Horizontal, Vertical, Diagonal Izq.-Der. y Der.-Izq.) • Gradientes <ul style="list-style-type: none"> • Sobel • Prewitt • Roberts

CAPÍTULO 6
RESULTADOS DEL EMPLEO
DEL SISTEMA DE
PROCESAMIENTO DIGITAL
DE IMÁGENES

LOGARÍTMICA



(a)
Imagen original



(b)
 $\alpha = 0.66000$



(c)
 $\alpha = 0.50000$



(d)
 $\alpha = 0.33000$

Figura 6.1 Transformación logarítmica con diferentes parámetros de α .



(a)
Imagen original



(b)
 $\alpha = 0.66000$



(c)
 $\alpha = 0.50000$



(d)
 $\alpha = 0.33000$

Figura 6.2 Transformación logarítmica con diferentes parámetros de α .



(a)
Imagen original



(b)
 $\alpha = 0.66000$



(c)
 $\alpha = 0.50000$



(d)
 $\alpha = 0.33000$

Figura 6.3 Transformación logarítmica con diferentes parámetros de α .

EXPONENCIAL



(a)
Imagen original



(b)
 $\alpha = 1.53000$



(c)
 $\alpha = 2.55000$



(d)
 $\alpha = 3.06000$

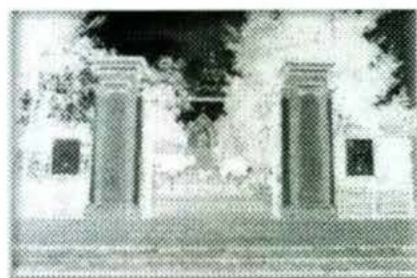
Figura 6.4 Transformación exponencial con diferentes parámetros de α .

NEGATIVO



(a)

Imagen original



(b)

Negativo de la imagen

Figura 6.5 Resultado de la aplicación del negativo a una imagen.



(a)

Imagen original

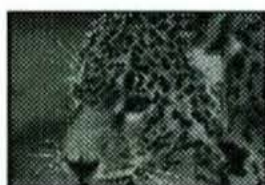


(b)

Negativo de la imagen

Figura 6.6 Resultado de la aplicación del negativo a una imagen.

BINARIZACIÓN



(a)

Imagen original



(b)

Umbral de 85



(c)

Umbral inferior de 45
Umbral superior de 205

Figura 6.7 Resultado de la aplicación de binarización a imagen en (a). Umbral de 85 en (b), Umbral inferior de 45 y superior de 205 en (c).



Figura 6.8 Resultado de la aplicación de binarización a la imagen en (a). Umbral de 128 en (b), Umbral inferior de 60 y superior de 180 en (c).



Figura 6.9 Resultado de la aplicación de binarización a la imagen en (a). Umbral de 128 en (b), Umbral inferior de 65 y superior de 185 en (c).

ADICIÓN DE IMÁGENES

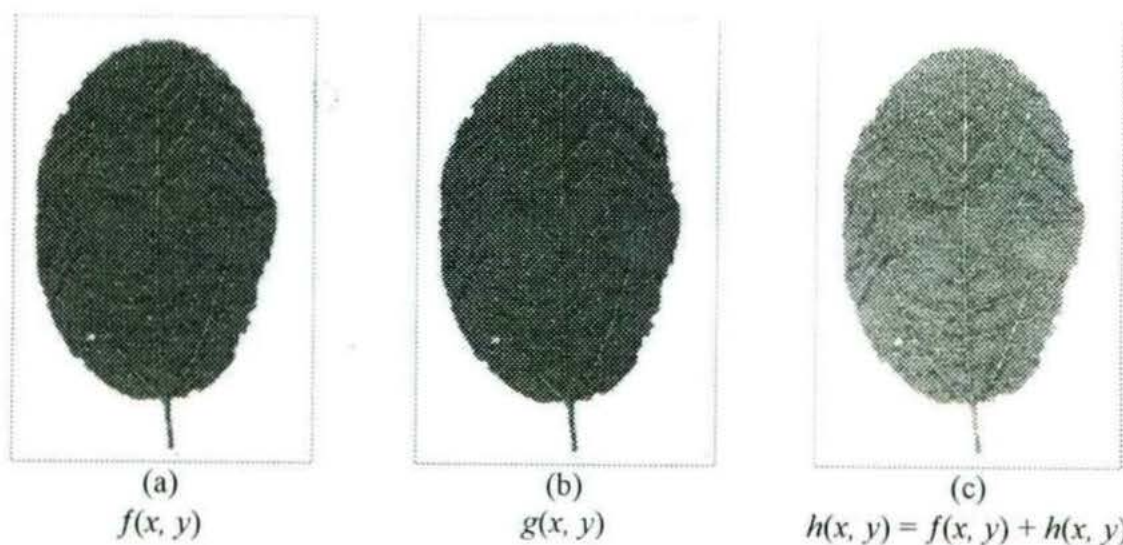


Figura 6.10 Suma de dos imágenes iguales, produce una imagen idéntica con dos veces el brillo.

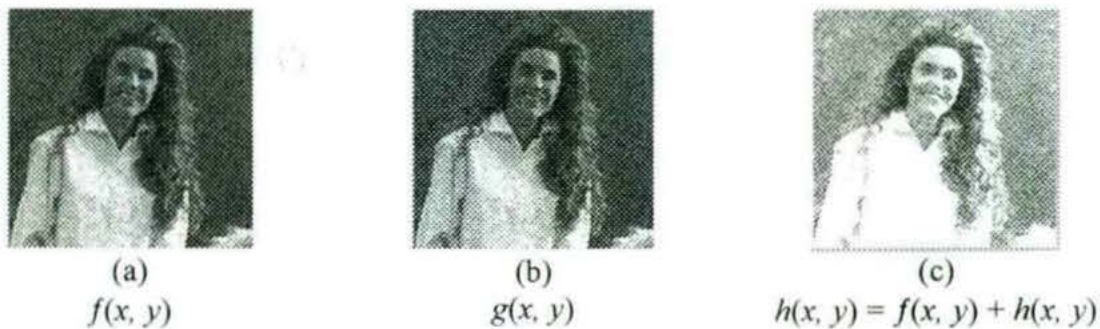


Figura 6.11 Suma de dos imágenes iguales, produce una imagen idéntica con dos veces el brillo.

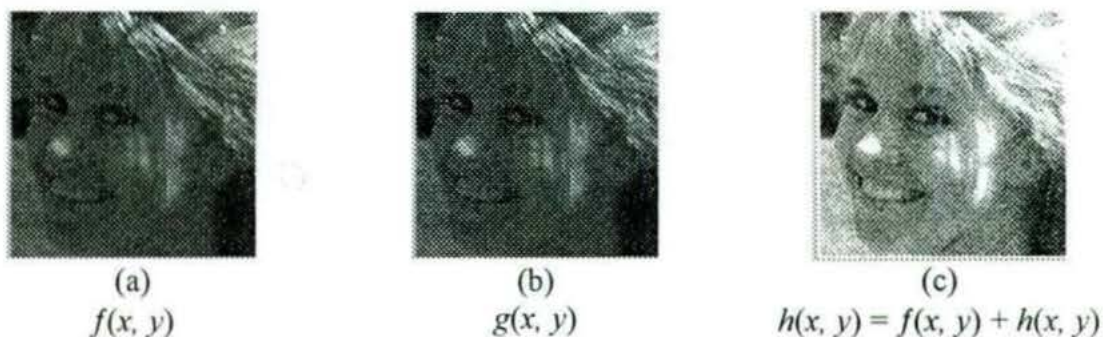


Figura 6.12 Suma de dos imágenes iguales, produce una imagen idéntica con dos veces el brillo.

SUSTRACCIÓN DE IMÁGENES



Figura 6.13 Resultado de aplicar la sustracción de imágenes.

PROMEDIO



(a)
Imagen original



(b)
Promedio con ventana de 3x3



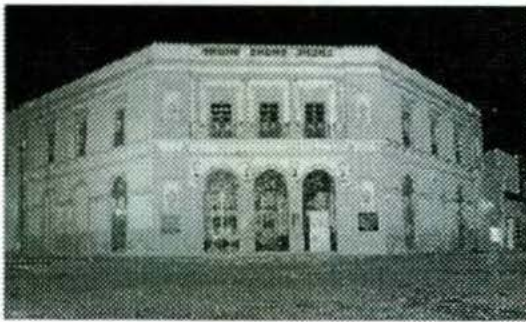
(c)
Promedio con ventana de 5x5



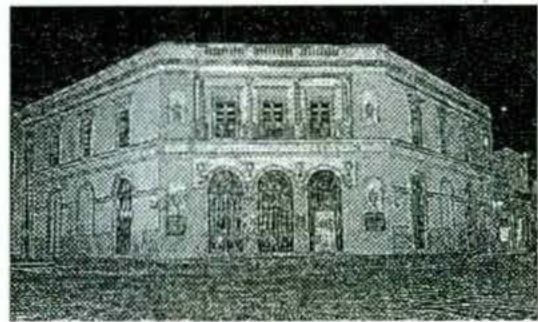
(d)
Promedio con ventana de 7x7

Figura 6.14 Resultado de aplicar el filtro del promedio a la imagen en (a). Promedio utilizando una ventana de 3x3 en (b), promedio utilizando una ventana de 5x5 en (c), promedio utilizando una ventana de 7x7 en (d).

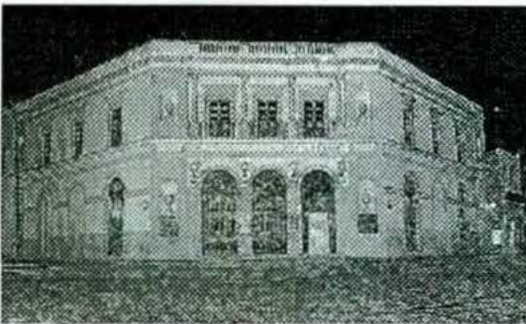
PASA-ALTAS



(a)
Imagen original



(b)
Pasa-Altas₁



(c)
Pasa-Altas₂



(d)
Pasa-Altas₃

Figura 6.15 Las imágenes de los incisos: (b), (c) y (d), son el resultado de aplicar los tres filtros de pasa altas a la imagen del inciso (a).

RESALTAMIENTO DE BORDES



(a)
Imagen original



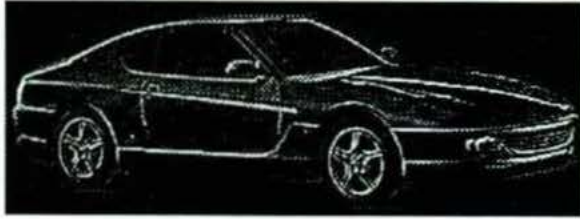
(b)
Norte



(c)
Noreste



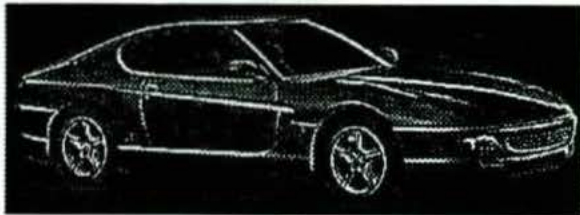
(d)
Este



(e)
Sureste



(f)
Sur



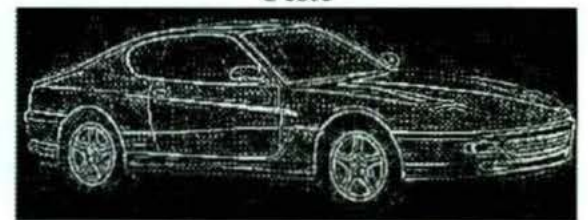
(g)
Suroeste



(h)
Oeste



(i)
Noroeste



(j)
Omnidireccional

GRADIENTE



(a)

Imagen original



(b)

Sobel Horizontal



(c)

Sobel Vertical



(d)

Sobel Omnidireccional



(e)

Prewitt Horizontal



(f)

Prewitt Vertical



(g)

Prewitt Omnidireccional



(h)

Roberts Horizontal



(i)

Roberts Vertical



(j)

Roberts Omnidireccional

LAPLACIANO



(a)
Imagen original



(b)
Laplaciano₁



(c)
Laplaciano₂

LÍNEAS DE INVESTIGACIÓN FUTURAS

El Procesamiento Digital de Imágenes, tiene en la actualidad muchos campos de aplicación, de ahí que este trabajo es realizado con el objetivo de que sea genérico, es decir, se aplique el procesamiento a diversos tipos de imágenes y marque la base para iniciar nuevas líneas de investigación sobre el Procesamiento Digital de Imágenes, pero analizando tipos de imágenes específicas.

Las líneas de investigación alternas son:

Procesamiento de imágenes para el reconocimiento de OVNI's.- En la actualidad sigue habiendo gran controversia cuando se habla sobre temas relacionados con OVNI's. El procesamiento de imágenes de este tipo no tiene como fin último afirmar sobre la existencia de OVNI's, sino solamente poder determinar si es auténtica la imagen o ha sido obtenida a través de un fotomontaje.

Procesamiento digital de imágenes en 3D.- Debido a que nuestra vida gira entorno a un mundo tridimensional, es importante implementar un software de procesamiento de imágenes de esta índole, para poder identificar elementos o características que sólo bajo estas circunstancias sea posible obtener mejores resultados.

Autenticación de firmas.- Este ha sido uno de los grandes problemas que se tiene para combatir el fraude. Con el tratamiento de imágenes pueden detectarse algunas diferencias en el documento y posteriormente aplicar técnicas de reconocimiento de patrones para determinar si una firma es legítima o no.

CONCLUSIONES

El desarrollo de esta tesis, implicó que investigara y analizara la metodología del Procesamiento Digital de Imágenes, para poder desarrollar el software de PDI, a través de la programación de los filtros espaciales.

Cabe mencionar que aunque el tema de tesis hace mención del procesamiento en el dominio espacial, se consideró conveniente incluir previamente el estudio e implantación de algoritmos correspondientes al procesamiento puntual, por su gran relación y como una forma de comprender mejor el funcionamiento de estas técnicas.

Es importante aclarar, que el software desarrollado es de carácter general, lo que implica que la información que se desee extraer dependerá del tipo de imagen que se desee procesar y del conocimiento a priori que se tenga de la misma.

Durante el desarrollo de esta tesis, existieron algunos obstáculos y limitantes que se tuvieron que superar, algunos de los cuales fue que no se contó con el equipo de cómputo más apropiado para procesar imágenes; la bibliografía sobre este tema con que cuenta la biblioteca de informática es escasa.

Por otra parte, considero que los objetivos se han cumplido, debido a que se ha desarrollado un software de PDI, disponible para ser utilizado por toda persona que lo necesite en el tratamiento de algunas imágenes.

**APÉNDICE A
ESTRUCTURA DEL
SOFTWARE DE
PROCESAMIENTO DIGITAL
DE IMÁGENES**

A.1.- Estructuras de datos empleadas en los diferentes métodos de procesamiento.

Para todos los métodos, se utiliza una estructura conocida como buffer virtual que es parte de las librerías del Fastgraph for Windows. El buffer virtual se maneja como una matriz de dimensiones $m \times n$, y cuyo contenido de cada posición es el valor del color correspondiente a ese pixel, teniendo un rango de 0 a 255. Este valor del color es el que se utiliza para manejar la escala de 256 tonos de gris.

Es importante mencionar, que las imágenes contenidas en un buffer virtual no se despliegan hasta el momento en que se indique explícitamente. Además, el Fastgraph proporciona métodos que permiten acceder y modificar los valores contenidos dentro del buffer virtual.

Dentro de los dos métodos de procesamiento considerados en esta tesis, se utilizan dos buffer virtuales correspondientes a la imagen original y la imagen que ha recibido el proceso, para poder efectuar en caso de ser necesaria, una operación de deshacer.

La forma que utiliza el Fastgraph para inicializar los buffers virtuales es, primero declarando variables de tipo entero que han de guardar el número de buffer virtual específico (Fastgraph soporta hasta 32 buffers virtuales activos al mismo tiempo). La declaración es la siguiente:

```
hvb1, hvb2: integer;  
hist1, hist2: integer;
```

En el siguiente método se observa la creación de los buffers virtuales:

```
procedure TFormMain.FormCreate(Sender: TObject);  
begin  
  dc := GetDC(FormMain.Handle);  
  fg_setdc(dc);  
  hpal := fg_defpal;  
  fg_realize(hpal);  
  fg_vbinit;  
  hvb1 := fg_vballoc(n, n);  
  hvb2 := fg_vballoc(n, n);  
  hist1 := fg_vballoc(n, 141);  
  hist2 := fg_vballoc(n, 141);  
  fg_vbopen(hist1);  
  fg_vbcolors;  
  fg_setcolor(255);  
  fg_fillpage;  
  fg_vbopen(hist2);  
  fg_vbcolors;  
  fg_fillpage;  
  fg_vbopen(hvb2);  
  fg_vbcolors;  
  fg_vbopen(hvb1);
```

```

fg_vbcolors;
fg_setcolor (0);
Application.OnActivate := AppOnActivate;
ImgA.Init;
ImgB.Init;
OldA.Init;
OldB.Init;
FileName := '';
end;

```

El valor de la constante n a la que se hace referencia es de 256. La inicialización del entorno para los buffers virtuales se hace mediante la instrucción:

```
fg_vbinit;
```

A continuación se crean en memoria los buffers virtuales:

```

hvb1 := fg_vballoc(n, n);
hvb2 := fg_vballoc(n, n);
hist1 := fg_vballoc (n, 141);
hist2 := fg_vballoc (n, 141);

```

Los primeros dos corresponden a la imagen original y a la imagen procesada, mientras que los siguientes dos corresponden a sus respectivos histogramas. Posteriormente se abre como buffer actual cada uno y se le asigna la paleta actual, llenándose todo el buffer con un valor de 255 (correspondiente al blanco).

```

fg_vbopen(hist1);
fg_vbcolors;
fg_setcolor (255);
fg_fillpage;
fg_vbopen(hist2);
fg_vbcolors;
fg_fillpage;
fg_vbopen(hvb2);
fg_vbcolors;
fg_vbopen(hvb1);
fg_vbcolors;

```

Para liberar de la memoria, los buffers virtuales son destruidos y se cierra el entorno de los buffers virtuales, como se muestra en el siguiente código:

```

procedure TFormMain.FormDestroy(Sender: TObject);
begin
    fg_vbclose;
    fg_vbfree(hvb1);
    fg_vbfree(hvb2);
    fg_vbfree(hist1);

```

```

fg_vbfree(hist2);
fg_vbfin;
DeleteObject(hpal);
ReleaseDC(FormMain.Handle,dc);
ImgA.Done;
ImgB.Done;
OldA.Done;
OldB.Done;
end;

```

Además se explican algunos de los métodos que son necesarios dentro de todo el código:

fg_vbopen (hvb1): Activa un buffer como buffer actual, de acuerdo al valor de hvb1. Si existía un buffer abierto, lo cierra y pone el nuevo como actual.

fg_vbclose: Desactiva el buffer actual.

fg_vbpaste (x1, x2, y1, y2, posx, posy): Toma el área establecida por x1, y1, x2 y y2, dentro del buffer actual y la despliega dentro del área de la ventana cliente a partir de las coordenadas posx, posy.

fg_getpixel (x, y): Regresa el valor del pixel de la posición (x, y) del buffer actual.

fg_setcolor (color): Establece el color actual para ser utilizado por las demás rutinas.

fg_point (x, y): Asigna el color actual dentro del buffer actual en la posición (x, y).

Las imágenes que se utilizan dentro del software están en formato BMP. Para tomar estas imágenes de los archivos y pasarlas a los buffers virtuales, se utilizó el siguiente método:

```

procedure TFormMain.AbrirArchivo (op: Byte);
var
  FileExt: string[4];
  Aceptar: Boolean;
  bmp_h: array [0..53] of byte;
  pcx_h: array [0..127] of byte;
  i: Byte;
  j, r2, g2, b2: Integer;
  r, g, x, b: Real;
  cvb, longvb: integer;
  FN: string;
begin
  if OpenFileDialog.ShowModal <> mrOk then exit;
  h:= 0;
  w:= 0;
  FN := OpenFileDialog.FileName;
  FileExt := UpperCase(ExtractFileExt(FN));
  if (FileExt = '.BMP') or (FileExt = '.PCX') then
  begin
    if op = 1 then

```



```

    cvb := hvb1
else
    cvb := hvb2;
fg_vbopen (cvb);
fg_setcolor (255);
fg_fillpage;
if FileExt = '.BMP' then
begin
    fg_bmphead (FN + Chr(0), bmp_h);
    Move (bmp_h [18], w, 4);
    Move (bmp_h [22], h, 4);
end
else
if FileExt = '.PCX' then
begin
    fg_pcxhead (FN + Chr(0), pcx_h);
    Move (pcx_h [ 8], w, 2);
    Move (pcx_h [10], h, 2);
end;
longvb := fg_vballoc (w, h);
fg_vbopen (longvb);
fg_vbcolors;
if FileExt = '.BMP' then
    fg_showbmp (FN + Chr(0), $00)
else
    if FileExt = '.PCX' then
        fg_showpcx (FN + Chr(0), $00);
Escala (cvb, longvb);
fg_vbopen (cvb);
fg_vbfree (longvb);
fg_vbopen (cvb);
if op = 1 then
begin
    VerImg1 := True;
    VerImg2 := False;
    FileName := FN;
end
else
    VerImg2 := True;
IDeshacer.Enabled := False;
CalculaHistograma (op);
end;

for i :=0 to 255 do
begin
    fg_getrgb(i,r2,g2,b2);
    r:= r2;g:= g2;b:= b2;
    x:=( r+ g+ b)/3;
    fg_setrgb(i, Round(x), Round(x), Round(x));

```

```

end;
Refresh;
end;

```

Para desplegar las imágenes y sus histogramas desde los buffers virtuales hasta la ventana cliente, se utilizó el manejador de evento OnPaint:

```

procedure TFormMain.FormPaint(Sender: TObject);
begin
  ShowImage;
  ShowOldImage;
end;

```

El método anterior se utiliza cuando se actualiza la información de la pantalla de manera implícita, como en el caso de hacer cambios entre aplicaciones, o de manera explícita, por medio del método **Refresh**.

Por su parte, los métodos ShowImage y ShowOldImage realizan el despliegado de los buffers virtuales en caso de estar activas estas imágenes:

```

procedure TFormMain.ShowImage;
begin
  if not VerImg1 then exit;
  fg_vbopen (hvb1);
  fg_vbpaste (0, n-1, 0, n-1, 25, n + 24);
  fg_vbopen (hist1);
  fg_vbpaste (0, 255, 0, 140, 24, 446);
end;

```

```

procedure TFormMain.ShowOldImage;
begin
  if not VerImg2 then exit;
  fg_vbopen (hvb2);
  fg_vbpaste (0, n-1, 0, n-1, 320, n + 24);
  fg_vbopen (hist2);
  fg_vbpaste (0, 255, 0, 140, 320, 446);
end;

```

A.1.1.- Método de procesamiento puntual.

El procesamiento puntual únicamente utiliza como estructura a los buffer virtuales.

A.1.2.- Método de procesamiento espacial.

Además de la estructura de los buffers virtuales, el procesamiento espacial hace uso de otras dos estructuras de datos, y una extra para el proceso de la mediana específicamente. La declaración es la siguiente:

```

type
  Ivec7x7 = Array [-3..3, -3..3] of integer;
  Rvec7x7 = Array [-3..3, -3..3] of real;
  ALista  = Array [1..121] of Byte;

```

Las matrices Ivec7x7 y Rvec7x7 son utilizadas para tomar la información de las ventanas dentro de la imagen, siendo la primera de tipo entero y la segunda de tipo real. Ambas son de 7x7 (de -3 a 3), permitiéndose, de acuerdo a los algoritmos utilizados, que la ventana correspondiente sea de 7x7, 5x5 y 3x3.

En el siguiente método se muestra como se lee una ventana de acuerdo a una posición dada y a un tamaño de ventana y se regresa dentro del vector vv de tipo IVec7x7:

```

procedure TFormMain.LeeVentana (x, y: integer; var vv: Ivec7x7;
                                size: byte);
var
  i, j, valor: integer;
  SizeDiv2: Integer;
begin
  SizeDiv2 := size div 2;
  FillChar (vv, SizeOf (vv), 0);
  for i := x - SizeDiv2 to x + SizeDiv2 do
    for j := y - SizeDiv2 to y + SizeDiv2 do
      if (i >= 0) and (j >= 0) and (i <= (n - 1)) and (j <= (n -
1)) then
        vv [i - x, j - y] := fg_getpixel (i, j);
end;

```

En este método, se aplica a una ventana tomada de la imagen original un filtro especificado en el vector vfilt y, se regresa la acumulación de la multiplicación elemento a elemento. El valor regresado se asignará a la posición a partir de donde se haya tomado la ventana:

```

function TFormMain.Multipluca (var vfilt: Rvec7x7;
                                var vv: Ivec7x7; size: byte):
integer;
var
  res: real;
  i, j: integer;
  SizeDiv2: integer;
begin
  res := 0.0;
  SizeDiv2 := size div 2;
  for i := -SizeDiv2 to SizeDiv2 do
    for j := -SizeDiv2 to SizeDiv2 do
      res := res + (vfilt [j, i] * vv [i, j]);
  Multipluca := Ajusta (res);
end;

```

El siguiente método es muy importante, pues dada una ventana con un filtro y un tamaño de ventana se calcula cualquier filtro:

```
procedure TFormMain.CalculaVentana (var vfilt: Rvec7x7; size:
byte);
var
  x, y: integer;
  vv: Ivec7x7;
begin
  for x := 0 to n-1 do
    for y := 0 to n-1 do
      begin
        fg_vbopen (hvb1);
        LeeVentana (x, y, vv, size);
        fg_vbopen (hvb2);
        fg_setcolor (Multiplica (vfilt, vv, size));
        fg_point (x, y);
      end;
    end;
end;
```

A continuación se muestra un ejemplo de una llamada al método anterior para calcular una detección de bordes con dirección Norte, definiendo la matriz:

```
procedure TFormMain.Norte1Click(Sender: TObject);
begin
  FillChar (vfilt, SizeOf (vfilt), 0);
  vfilt[-1, -1] := 1; vfilt[-1, 0] := 1; vfilt[-1, 1] := 1;
  vfilt[ 0, -1] := 1; vfilt[ 0, 0] := -2; vfilt[ 0, 1] := 1;
  vfilt[ 1, -1] := -1; vfilt[ 1, 0] := -1; vfilt [ 1,1] := -1;
  Operaciones (opVentana);
end;
```

A.2.- ¿Cómo se estructuró el software de Procesamiento Digital de Imágenes?

El software cuenta con rutinas que le permiten realizar la operación **deshacer**, para lo cual se requirió contar con buffers virtuales y varias instancias del Vector2D, así como un método que controla de manera general la realización del deshacer, como se observa en la figura A.1., en la que se puede apreciar la interrelación de los diferentes métodos de la forma principal.

Primeramente se declararon una serie de constantes para cada uno de los procesos. Estas constantes pueden corresponder a procesos individuales o a procesos generales, como en algunos casos de los procesos espaciales con una ventana de 3x3. Las constantes definidas son las siguientes:

```
const
  opNegativo          = 0;
```

```

opVentana           = 1;
opPromedio          = 2;
opMediana           = 3;
opSuma              = 4;
opResta             = 5;
opMultiplica        = 6;
opDivide            = 7;
opUmbral            = 8;
opUmbralRangos      = 9;
opUserDef           = 10;
opExponencial       = 11;
opLogaritmica       = 12;
opIgualarHisto      = 13;
opVerPaleta         = 14;

```

Clase TFormMain

Después se crearon los manejadores de eventos para cuando una opción del menú fuera seleccionada. p.e.:

```

procedure TFormMain.INegativosClick(Sender: TObject);
begin
  Operaciones (opNegativo);
end;

```

Posteriormente se crea el método que resuelve cada uno de los procesos individuales o generales:

```

procedure TFormMain.Negativo (fuente, destino, limite: byte);
var
  x, y: Word;
begin
  for x := 0 to n-1 do
  begin
    for y := 0 to n-1 do
    begin
      fg_vbopen (fuente);
      if fg_getpixel (x, y) <= limite then
        fg_setcolor (255 - fg_getpixel (x, y))
      else
        fg_setcolor (fg_getpixel (x, y));
      fg_vbopen (destino);
      fg_point (x, y);
    end;
    PBar.Progress := Round ((x / (n - 1)) * 100);
  end;
end;

```

La integración se hará mediante el método Operaciones, el cual llama a todos los diferentes procesos y se encarga de las operaciones para mantener el deshacer:

```
procedure TFormMain.Operaciones (op: Word);
var
  e: integer;
  v: word;
  r: real;
begin
  PBar.Progress := 0;
  case op of
    opExponencial, opLogaritmica: begin
      if op = opExponencial then
        begin
          FBrillo.Caption := 'Exponencial';
          FBrillo.Accion.Caption := 'Oscurecimiento';
          FBrillo.dividendo := 0.51;
          FBrillo.Valor.Min := 2;
          FBrillo.Valor.Max := 8;
          FBrillo.Valor.Position := 5; .
        end
      else
        begin
          FBrillo.Caption := 'Logarítmica';
          FBrillo.Accion.Caption := 'Aclaramiento';
          FBrillo.dividendo := 0.001;
          FBrillo.Valor.Min := 1;
          FBrillo.Valor.Max := 999;
          FBrillo.Valor.Position := 500;
        end;
      if FBrillo.ShowModal = mrOk then
        begin
          PBar.Visible := True;
          Contraste (FBrillo.Valor.Position * FBrillo.dividendo);
        end;
      end;
    opVerPaleta: VerPaleta;
    opUserDef: begin
      end;
    opNegativo: begin
      FDiag255.Caption := 'Límite del Negativo';
      FDiag255.Leyenda.Caption :=
        'Rango de escala de gris entre 0-255: ';
      FDiag255.Escala.Value := 255;
      if FDiag255.ShowModal = mrOk then
        begin
          PBar.Visible := True;
          Negativo (hvb1, hvb2, FDiag255.Escala.Value);
        end;
      end;
  end;
```

```

end;
opVentana: begin
  PBar.Visible := True;
  CalculaVentana (vfilt, v3x3);
end;
opPromedio: begin
  if FWinSize.ShowModal = mrOk then
  begin
    PBar.Visible := True;
    Promedio (FWinSize.WinSize.Value);
  end;
end;
opMediana: begin
  if FWinSize.ShowModal = mrOk then
  begin
    PBar.Visible := True;
    Mediana (FWinSize.WinSize.Value);
  end;
end;
opUmbral: begin
  FDiag255.Caption := 'Límite Superior del Umbral';
  FDiag255.Leyenda.Caption :=
    'Rango de escala de gris entre 0-255: ';
  FDiag255.Escala.Value := 128;
  if FDiag255.ShowModal = mrOk then
  begin
    PBar.Visible := True;
    UmbralSuperior (FDiag255.Escala.Value);
  end;
end;
opUmbralRangos: begin
  if FDiagRango.ShowModal = mrOk then
  begin
    PBar.Visible := True;
    UmbralRangos (FDiagRango.Desde.Value,
                  DiagRango.Hasta.Value);
  end;
end;
opIgualarHisto: begin
  FDiag255.Caption := 'Igualación de Histograma';
  FDiag255.Leyenda.Caption := 'Niveles de Gris 0-255: ';
  FDiag255.Escala.Value := 255;
  if FDiag255.ShowModal = mrOk then
  begin
    PBar.Visible := True;
    IgualarHistograma (FDiag255.Escala.Value);
  end;
end;
opSuma,

```

```

opResta,
opMultiplica,
opDivide: begin
  if not VerImg2 or (VerImg2 and
    (MessageDlg ('Deseas utilizar las dos imágenes en
      pantalla', mtConfirmation, [mbYes, mbNo], 0)
      = mrNo)) then

    begin
      AbrirArchivo (2);
      Refresh;
      MessageDlg ('Pulsa Ok para comenzar', mtInformation,
        [mbOk], 0);
    end;
  PBar.Visible := True;
  case op of
    opSuma: Suma;
    opResta: Resta;
    opMultiplica: Multiplicacion;
    opDivide: Divide;
  end;
end;
end;
if (op <> opSuma) and (op <> opResta) and (op <> opMultiplica)
and
  (op <> opDivide) then
  begin
    VerImg1 := True;
    VerImg2 := True;
    Cambia (hvb1, hvb2);
    Cambia (hist1, hist2);
    ImgBin2 := ImgBin1;
    CalculaHistograma (1);
    Refresh;
    IDeshacer.Enabled := True;
  end
  else
  begin
    VerImg1 := True;
    VerImg2 := False;
    CalculaHistograma (1);
    Refresh;
    IDeshacer.Enabled := False;
  end;
  PBar.Visible := False;
end;

```


Por último, se presenta el código del método Deshacer:

```
procedure TFormMain.IDeshacerClick(Sender: TObject);
begin
  VerImg2 := False;
  ImgBin1 := ImgBin2;
  Cambia (hvb1, hvb2);
  Cambia (hist1, hist2);
  Refresh;
  IDeshacer.Enabled := False;
end;
```

BIBLIOGRAFÍA

Libros:

- [1] ALLENDE, D. G. (1993). *Reconocimiento de formas y visión artificial*. 1ª. ed. Madrid, España: Addison-Wesley Iberoamericana.
- [2] BAXES, A. G. (1994) *Digital image processing*. John Wiley & Sons.
- [3] GONZÁLEZ, R. C. (1992). *Digital image processing*. Addison-Wesley Publishing Company.
- [4] HERNÁNDEZ, R., FERNÁNDEZ, C., y BAPTISTA, P. (1991). *Metodología de la investigación*. 2ª. ed. McGraw-Hill Interamericana Editores.
- [5] TED, G. (1995). *Software: Fastgraph. Programmer's graphics library for Windows*.
- [6] TEIXEIRA, S. y PACHECO, X. (2000). *Guía de desarrollo Delphi 5*. Vol. 1 y 2. Prentice Hall.

Páginas WEB:

- [7] BISHEIMER, G. y SONNAILLON, M. (1999). *Procesamiento digital de señales con aplicación a imágenes*.
7 archivos .PDF zippeados. Disponible en:
<http://www.geocities.com/SiliconValley/Code/3101/PDSAI.html>
- [8] BRAVO, A. J. V. (1999). *Procesamiento digital de imágenes*.
<http://www.ing.ula.ve/~abravo/document/tutorial/imagenes/indice.html>
- [9] CERQUIDES, J. R. B. *Tratamiento Digital de Imágenes*.
puntuales.ps y espaciales.ps. Diponible en:
<http://viento.us.es/jramon/docencia/tdi/tdi.htm>
- [10] FERNÁNDEZ, R. (1998). *Procesamiento digital de imágenes, una herramienta informática para la identificación de NN*. Año 2, No. 54. Univ. Nac. De Mar del Plata.
<http://www.de.fcen.uba.ar/prensa/educyt/1998/ed54b.htm>
- [11] *Introducción al procesado de imagen*.
<http://wgpi.tsc.uvigo.es/libro/procesim.htm>
- [12] MEDINA, R.
<http://www.ing.ula.ve/~rmedina/tecnicas/>
- [13] MOLINA, R. S. y CORTIJO, F. J. B. (1997).
http://www-etsi2.ugr.es/depar/ccia/robotica/indice_apuntes.html
- [14] MOSQUERA, A. G. *Temas sobre procesado digital de imágenes*.
<http://www.paisvirtual.com/informatica/navegadores/toniomos/pdi/temas-pdi.htm>
- [15] Notas.ps. Disponible en:
<http://campus.mor.itesm.mx/~esucar/Vision/indice.html>
- [16] PASCUAL, D. M. (1996). *Convoluciones*.
<http://www.geocities.com/CollegePark/Union/3114/graficos.htm>
- [17] PÉREZ, N. B. (1995). *Fundamentos del tratamiento de imágenes. Visión por computador*.
<http://www.uco.es/informa/local/cordoba/ccordoba/ccordoba.html>
- [18] SUCAR, E. S. (2000). *Procesamiento de imágenes y visión computacional*.

GLOSARIO

Borde	Es un cambio abrupto del nivel de gris con respecto a sus vecinos.
Brillantez	Es el valor asociado con un punto de la imagen, que representa la cantidad de luz que emana o refleja el objeto en ese punto.
Conectividad	Relación geométrica que un pixel establece con sus vecinos inmediatos.
Contorno	Frontera de una región perteneciente a un objeto dentro de una imagen, la cual se define por cambios abruptos en niveles de gris.
Convolución	Es un proceso matemático que combina dos funciones para producir una tercera función.
Digitalización	Es el proceso de convertir una escena de una imagen a una forma digital.
Dilatación	La dilatación es la transformación morfológica que combina dos vectores utilizando la suma.
Dominio espacial	Se refiere a transformaciones sobre las coordenadas espaciales de la imagen.
Detección de bordes	Es una técnica, en el cual un pixel borde es identificado en base a sus vecinos.
Escala de gris	Es el conjunto de todos los posibles niveles de gris en una imagen digital.
Filtro	En el dominio espacial, un filtro digital es una función sobre los pixeles de la vecindad de cada pixel de la imagen.
Gradiente	Vector que apunta en la dirección de máximo cambio en la vecindad de un pixel. Se usa como filtro para resaltar los contornos de una región.
Histograma	Representa una estimación discreta de la función de densidad de probabilidad de los niveles de gris de una imagen digital.
Igualación de histograma	Consiste en obtener una distribución uniforme de la función de densidad de probabilidad de los niveles de gris de una imagen.
Imagen binaria	Es una imagen que tiene solamente dos niveles de gris (usualmente el negro y el blanco).
Imagen digital	Una imagen digital es un arreglo bidimensional de enteros cuyos índices de filas y columnas denotan la posición de un punto, y el valor del punto corresponde a un nivel de gris.
Mediana	Este filtro no lineal selecciona la mediana del conjunto de pixeles de una vecindad. Elimina el ruido aleatorio en forma líneas y puntos de la imagen.
Mejoramiento de la imagen	Es un conjunto de procesos que intentan mejorar la apariencia visual de la imagen.
Nivel de gris	Es el valor asociado con un pixel en una imagen digital, representando la brillantez de la escena original en un punto que es representado por el pixel.
Operación algebraica	Es la operación de suma, resta, multiplicación o división que se realiza pixel por pixel entre dos imágenes.
Pixel	Acrónimo formado con las palabras inglesas 'Picture' 'element'. Elemento pictórico de la imagen.
Promedio	Este filtro lineal selecciona el promedio del conjunto de pixeles de una vecindad.

Ruido	Componentes no pertenecientes a una imagen en el que estorba el reconocimiento e interpretación de los datos.
Umbral	Valor de un nivel de gris utilizado como referencia para separar dos regiones.
Vecinos	Es un conjunto de pixeles localizados alrededor de un pixel dado.
Visión artificial	Conjunto de técnicas y modelos que permiten el procesamiento, análisis y explicación de cualquier tipo de información espacial obtenida a través de imágenes digitales.