



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
CUAUTITLAN**



**"SISTEMA DE PROCESAMIENTO DIGITAL DE IMAGENES
UTILIZANDO TECNOLOGIA ORIENTADA A OBJETOS"**

T E S I S
QUE PARA OBTENER EL TITULO DE:
INGENIERO MECANICO ELECTRICISTA
P R E S E N T A N:

ALBERTO CORTES ZARAGOZA
LEONARDO MOISES GUZMAN FLORES

ASESOR: FIS. J. JESUS CRUZ GUZMAN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

14
23



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
CUAUTITLAN**



**“SISTEMA DE PROCESAMIENTO DIGITAL DE IMAGENES
UTILIZANDO TECNOLOGIA ORIENTADA A OBJETOS”**

**T E S I S
QUE PARA OBTENER EL TITULO DE:
INGENIERO MECANICO ELECTRICISTA
P R E S E N T A N:**

**ALBERTO CORTES ZARAGOZA
LEONARDO MOISES GUZMAN FLORES**

ASESOR: FIS. J. JESUS CRUZ GUZMAN



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN
UNIDAD DE LA ADMINISTRACIÓN ESCOLAR
DEPARTAMENTO DE EXÁMENES PROFESIONALES

U. N. A. M.
FACULTAD DE ESTUDIOS
SUPERIORES CUAUTITLÁN

ASUNTO: VOTOS APROBATORIOS



DR. JAIME KELLER TORRES
DIRECTOR DE LA FES-CUAUTITLÁN
P R E S E N T E .

AT'N: Ing. Rafael Rodríguez Gómez
Jefe del Departamento de Exámenes
Profesionales de la F.E.S. - C.

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la TESIS TITULADA:

"Sistema de Procesamiento Digital de Imágenes utilizando Tecnología

Orientada a Objetos"

que presenta el pasante: Leonardo Moisés Guzmán Flores
con número de cuenta: 8323785-6 para obtener el TÍTULO de:
Ingeniero Mecánico Electricista

Considerando que dicha tesis reúne los requisitos necesarios para ser discutida en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

A T E N T A M E N T E .

"POR MI RAZA HABLARA EL ESPIRITU"

Cuatitlán Izcalli, Edo. de Méx., a 07 de Ago de 1995

PRESIDENTE Fis. J. Cruz Guzmán
VOCAL Ing. Jorge Buendía Gómez
SECRETARIO Ing. Roselío Ramos Carranza
PRIMER SUPLENTE Ing. Martha Urreta Vargas
SEGUNDO SUPLENTE Ing. Juan González Vega

[Firma] 7-08-95
[Firma] 21/08/95
7-08-95
7-08-95



UNIVERSIDAD NACIONAL
AVENIDA DE
MEXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN
UNIDAD DE LA ADMINISTRACION ESCOLAR
DEPARTAMENTO DE EXAMENES PROFESIONALES

U. N. A. M.
FACULTAD DE ESTUDIOS
SUPERIORES CUAUTITLAN

ASUNTO: VOTOS APROBATORIOS



DEPARTAMENTO DE

DR. JAIME KELLER TORRES
DIRECTOR DE LA FES-CUAUTITLAN
P R E S E N T E .

AT'N: Ing. Rafael Rodríguez Caballero
Jefe del Departamento de Exámenes
Profesionales de la F.E.S. - C.

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la TESIS TITULADA:

Sistema de Procesamiento Digital de Imágenes utilizando
Tecnología Orientada a Objetos.

que presenta el pasante Alberto Cortés Zaragoza.

con número de cuenta: 8333063-4 para obtener el TITULO de:
Ingeniero Mecánico Electricista.

Considerando que dicha tesis reúne los requisitos necesarios para ser discutida en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

A T E N T A M E N T E .

"POR MI RAZA HABLARA EL ESPIRITU"

Cuatitlán Izcalli, Edo. de Méx., a 21 de Agosto de 199 5

PRESIDENTE	Ing. J. Jesús Cruz Guzmán	<u>[Firma]</u> 7-09-95
VOCAL	Ing. Jorge Buendía Gómez.	<u>[Firma]</u> 21/08/95
SECRETARIO	Ing. Rogelio Ramos Carranza.	<u>[Firma]</u> 7-08-95
PRIMER SUPLENTE	Ing. Martha Urrutia Vargas.	<u>[Firma]</u> 12-08-95
SEGUNDO SUPLENTE	Ing. Juan González Vega.	<u>[Firma]</u> 12-08-95

AGRADECIMIENTOS.

Queremos hacer patente nuestro agradecimiento al profesor Jesús Cruz, por su gran apoyo, paciencia y motivación que tuvo para con nosotros en el desarrollo de esta tesis.

Igualmente agradecemos al Ing. Jorge Buendía por sus valiosos comentarios para la edición de este trabajo.

Agradecemos a toda la gente que de una o de otra manera, ya sea con su apoyo en forma directa o con sus ánimos, nos ayudaron a terminar lo que parecía un trabajo interminable: Moisés, Miguel, Francisco, Gloria, Gabriel, Sra. Alicia, Mary, Profesores y Amigos.

Un agradecimiento muy especialmente al programa UNAM-CRAY, cuyo apoyo a la Cátedra de Procesamiento Digital de Imágenes permitió llevar a cabo el presente trabajo.

A mis padres Moisés y Carmelita, por su apoyo y motivación que me brindaron siempre.

A mis hermanos, Bety, Gerardo y Alejandro, por que forman una parte muy importante de mi vida.

A Meli, por su paciencia, comprensión y apoyo incondicional que me ha brindado siempre.

A mi tía Esperanza y a Male por su ayuda y animo que siempre me proporcionaron.

A mi amigo Alberto, por que si el no hubiera sido posible este trabajo.

También mi reconocimiento al gran interés y apoyo de todas aquellas personas a quien aprecio: Moises H., Miguel A., Ivan D., Lourdes D., Diego D., Cesar D., Manuel D., Mauricio, Jaime, Profesores y Amigos.

Leonardo Moisés Guzmán F.

AGRADECIMIENTOS.

Queremos hacer patente nuestro agradecimiento al profesor Jesús Cruz, por su gran apoyo, paciencia y motivación que tuvo para con nosotros en el desarrollo de esta tesis.

Igualmente agradecemos al Ing. Jorge Buendía por sus valiosos comentarios para la edición de este trabajo.

Rogelio Ramos C., Martha Urrutia V. y Juan González V. por todo lo que aportaron a nuestra formación profesional.

Agradecemos a toda la gente que de una o de otra manera, ya sea con su apoyo en forma directa o con sus ánimos, nos ayudaron a terminar lo que parecía un trabajo interminable: Moisés, Miguel, Francisco, Gloria, Gabriel, Sra. Alicia, Mary, Profesores y Amigos.

Un agradecimiento muy especialmente al programa **UNAM-CRAY**, cuyo apoyo a la Cátedra de Procesamiento Digital de Imágenes permitió llevar a cabo el presente trabajo.

Este trabajo está dedicado...

A mis padres: Carlos y María del Refugio.

A mis adoradas: Martha y Aglaé quienes tanto me han ayudado.

A mis hermanos: Carlos, Domingo, Alejandra, Victoria, Leticia, Marcos y Alma.

A todos mis amigos que sabrán perdonar el que no los nombre aquí.

**A todos ellos agradezco por todo el apoyo, comprensión y amor que me han
brindado en esta vida.**

Alberto Cortés Z.

A mis padres Moisés y Carmelita, por su apoyo y motivación que me brindaron siempre y a quienes les dedico este trabajo.

A mis hermanos, Bety, Gerardo y Alejandro, por que forman una parte muy importante de mi vida.

A Meli, por su paciencia, comprensión y apoyo incondicional que me ha brindado siempre.

A mi tía Esperanza y a Male por su ayuda y animo que siempre me proporcionaron.

A mi amigo Alberto, por que si el no hubiera sido posible este trabajo.

También mi reconocimiento al gran interés y apoyo de todas aquellas personas a quien aprecio: Moises H., Miguel A., Ivan D., Lourdes D., Diego D., Cesar D., Manuel D., Mauricio, Jaime, Profesores y Amigos.

Leonardo Moisés Guzmán F.

PROLOGO.

PROLOGO

Desde la antigüedad el hombre ha hecho uso de la información contenida en las imágenes que miraba a su alrededor, tales como el observar animales; fenómenos meteorológicos; e incluso en la actitud que podía percibir de otros hombres, tal como la ira, o el miedo; todas estas imágenes eran adquiridas por el medio natural, es decir el ojo humano.

El avance que ha tenido en los últimos años el desarrollo de la tecnología y más recientemente la llegada de la computadora, ha permitido al ser humano llegar a un grado muy adelantado en el campo del procesamiento de imágenes.

En nuestros días el procesamiento de imágenes es de gran importancia en diferentes campos de la investigación, tales como el área biológica, física, médica, geográfica, robótica, por solo mencionar algunas de ellas y se espera que en un futuro se use aún más ampliamente en cualquier área.

El procesamiento de imágenes requiere de la computadora a causa de la gran cantidad de información que maneja (cálculo, despliegue, etc.), por lo que su desarrollo se encuentra directamente ligado a esta. Por lo que los métodos de programación son de gran importancia, y hoy en día existe un nuevo paradigma llamado "orientado a objetos".

Nuestra capacidad de creación de software de computación no está a la par con la evolución del hardware, se necesita una revolución industrial en el software.

Es probable que esta revolución provenga de las técnicas Orientadas a Objetos, combinadas con herramientas CASE, generadores de código, programación visual y desarrollo basado en depósitos. La meta es maximizar la reutilización de código, así como construir y almacenar objetos complejos.

Las técnicas orientadas a objetos modifican el punto de vista de los analistas de sistemas de información, acerca del mundo. En vez de pensar en los procesos y su descomposición, piensan en objetos y su comportamiento. Los objetos pueden ser complejos desde el punto de vista interno, como una máquina electrónica, pero los analistas no tienen que comprender toda esa complejidad (a menos que la diseñen). Les basta conocer los comportamientos de los objetos y su forma de uso.

PROLOGO.

En esta tesis nos proponemos hacer uso de estas nuevas técnicas de desarrollo de software para implementar una herramienta de procesamiento de imágenes, bajo una interfase con el usuario de fácil utilización.

El desarrollo del presente trabajo está dividido en cinco capítulos:

El primer capítulo expone los fundamentos tanto del Procesamiento de Imágenes, como de la programación Orientada a Objetos.

El segundo capítulo trata acerca de las técnicas del Procesamiento Digital de Imágenes, que se divide en dos importantes áreas: el dominio del espacio y el dominio de la frecuencia.

En el tercer capítulo se explica el Análisis y el Diseño utilizado en la Programación Orientada a Objetos.

El cuarto capítulo se desarrolla la aplicación, utilizando todos los conceptos de los capítulos anteriores.

En el quinto capítulo se presenta un ejemplo de un tumor cerebral en un recién nacido, resuelto con el sistema que desarrollamos.

INDICE

PROLOGO	1
CAPITULO 1	
INTRODUCCION	2
1.1 Fundamentos del Procesamiento Digital de Imágenes	2
1.1.1 Imagen digital	2
1.1.2 Clasificación de imágenes	4
1.1.3 Elementos del procesamiento digital	5
1.1.4 Características de las imágenes	8
1.1.4.1 Escala de niveles de gris	8
1.1.4.2 Luminancia	8
1.1.4.3 Color	10
1.1.5 Imágenes multiespectrales	13
1.1.6 Muestreo	15
1.2 Fundamentos de tecnología orientada a objetos	22
1.2.1 Fundamentos sobre objetos	22
1.2.2 ¿Que es la programación orientada a objetos?	23
1.2.3. ¿Que es un objeto?	24
1.2.4. ¿Que es un tipo de objeto?	26
1.2.5. Métodos	26
1.2.6. Encapsulamiento	27
1.2.7. Mensajes	29
1.2.8. Clase	29
1.2.9. Herencia	30

CAPÍTULO 2

TECNICAS DE PROCESAMIENTO DIGITAL IMAGENES	33
2.1 Procesamiento de imágenes en el dominio espacial	33
2.1.1 Histograma	34
2.1.1.1 Fundamentos	34
2.1.1.2 Ecuación del histograma	37
2.1.2 Convolución	47
2.1.2.1 Convolución en el dominio espacial	47
2.1.2.2 Filtros usando convolución	51
2.1.2.3 Filtros de alisamiento	54
2.1.2.4 Filtros para afinamiento de imágenes	55
2.1.2.5 Detectores direccionales de bordes simples	57
2.2 Procesamiento de imágenes en el dominio de la frecuencia	58
2.2.1 Transformación de imágenes.	58
2.2.1.1 Transformación unitaria de dos dimensiones	58
2.2.2 Transformada de Fourier	59
2.2.2.1 Transformada rápida de Fourier (fft)	60
2.2.3 Procesamiento de imágenes por transformada	63
2.2.3.1 Transformada Haar	63
2.2.3.2 Transformada Hadamard	65
2.2.3.3 Transformada Karhunen-loeve	67
2.2.3.4 Transformada Slant.	69
2.2.3.5 Transformada Coseno	72
2.2.3.5.1 Transformada coseno de simetría semejante	73
2.2.3.5.2 Transformada coseno de simetría impar	74
2.2.3.6 Transformada Seno	75

CAPITULO 3

ANALISIS ORIENTADO A OBJETOS 78

3.1. Análisis orientado a objetos 78

3.1.1 Conceptos sobre los objetos 78

3.1.2 Identificación de objetos 81

3.1.3 Especificación de atributos 84

3.1.4 Definición de las operaciones 84

3.1.5 Comunicación entre objetos 86

3.1.6. Resumen de análisis 88

3.2. Diseño orientado a objetos 89

3.2.1. Conceptos del diseño orientado a objetos 90

3.2.2. Descripciones de los objetos 91

3.2.3. Una notación para el D.O.O 94

3.2.4. Representación de relaciones entre clases y entre objetos 94

3.2.5. Modularización del diseño 96

3.2.6. Diseño detallado para la implementación 97

3.2.7. Resumen de diseño 99

CAPITULO 4.

DESARROLLO DE LA APLICACIÓN 101

4.1. Análisis de requerimientos 101

4.1.1. Descripción del sistema 101

4.1.2. Documento de análisis requeridos 103

4.2. Análisis del dominio 103

4.2.1. Definición de clases 104

4.2.2. Diagrama de clases 104

INDICE

4.2.3. Especificación de clases	105
4.2.4. Diagramas de escenarios de objetos	108
4.2.5. Funciones del sistema	109
4.2.6. Diagramas de escenarios de objetos	109
4.3. Diseño del sistema	117
4.3.1. Definición de arquitectura inicial	117
4.3.2. Planear versiones ejecutables	118
4.3.3. Plan para las versiones ejecutables	118

CAPITULO 5

APLICACION PRACTICA	121
----------------------------	------------

CONCLUSIONES	131
---------------------	------------

BIBLIOGRAFIA	132
---------------------	------------

INTRODUCCION

CAPITULO

1

INTRODUCCION

1.1 FUNDAMENTOS DEL PROCESAMIENTO DIGITAL DE IMAGENES

1.1.1 IMAGEN DIGITAL

Una imagen digital es una representación, semejanza o imitación de un objeto o cosa; una descripción de una gráfica o animación, una presentación de algo para representar otro algo.

Una imagen digital se considera como un arreglo matricial de números. Un ejemplo es mostrado en la figura 1.1.1, la escena es un cuadro negro en un fondo blanco, en la misma figura se presenta una posible representación digital. Cada punto dentro de la representación digital corresponde a una área de un objetivo en el espacio; un valor digital es asignado a cada punto dentro de la imagen digital relacionando la intensidad del área del objeto en el espacio.

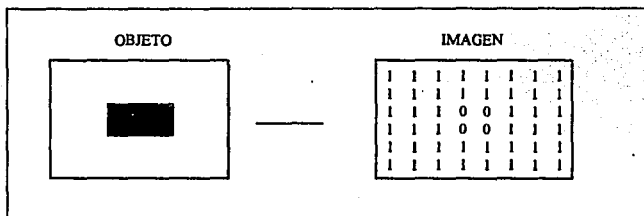


Figura 1.1.1 Modelo de una imagen digital

El componente individual de los elementos dentro de la imagen digital son referidos "picture elements". Dos abreviaciones comunes para este termino son pixel y pel. El valor digital de cada pixel representa la intensidad digital. Usando esta notación, la intensidad

INTRODUCCION

digital del pixel de la línea 3, columna 4, en la imagen mostrada en la fig. 1.1.1 es cero. En una imagen en blanco y negro, una intensidad de cero representa el negro y una intensidad de 1 representa el blanco.

La figura 1.1.2 contiene una representación digital diferente de la misma escena mostrada en la figura 1.1.1; la misma frecuencia de muestreo (que será analizada posteriormente) ha sido usada para generar esta versión digital, pero el número de bits usados para representar la intensidad digital ha sido incrementada de 1 a 8, por lo tanto, los valores de la intensidad digital pueden ser de un rango de 0 a 255, con el cero se representa el negro y con el 255 se representa el blanco.

Los efectos que ocurren en una representación digital cuando se muestra una escena continua pueden ser vistos en la figura 1.1.2. Primero, los valores de intensidad digital representan el cuadro negro. Los cuatro píxeles representan la uniformidad del cuadro negro y tienen valores de intensidad digital de 3, 2, 2 y 4.

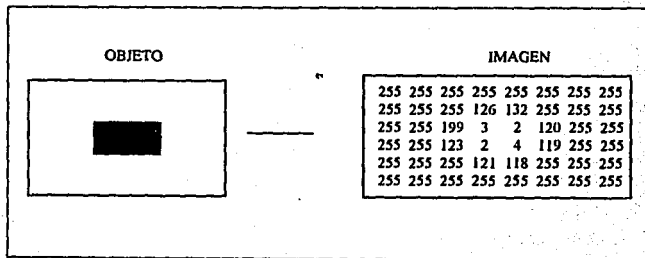


Figura 1.1.2 Representación digital de la figura 1.1.1, intensidad digital representada como 8 bits por píxel.

INTRODUCCION

La mayoría de los sistemas están provistos de representaciones de muestras digitales de una escena, la cual presenta un grado de ruido no deseado dentro de la imagen muestreada. Si no se presenta ruido durante el proceso de muestreo, la intensidad de los valores digitales de los 4 pixeles sería la misma.

Un segundo efecto que puede ser observado es la falta de una transición aguda en la imagen digital muestreada a los bordes del cuadro negro. En la figura 1.1.1 el cuadro es claramente definido en los límites, pero en la 1.1.2 se presenta lo que realmente ocurre cuando el límite es muestreado para crear una representación digital discreta en la escena.

1.1.2 CLASIFICACION DE IMAGENES

Las imágenes se presentan en varias formas: visibles, no visibles, abstractas, reales, adecuadas para el análisis por computadora y no adecuadas.

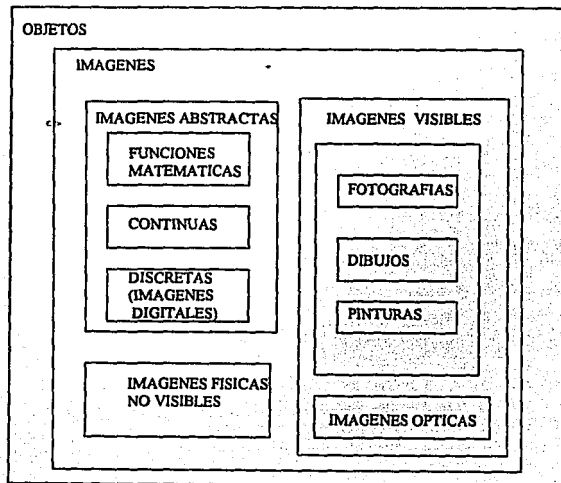


Figura 1.1.3 Tipos de imágenes

INTRODUCCION

Las imágenes se clasifican tomando en cuenta la forma o método con el cual son generadas. Si consideramos como un conjunto a todos los objetos, las imágenes forman un subconjunto (figura 1.1.3).

Dentro del subconjunto de las imágenes, se encuentran varios subconjuntos, uno de ellos es el de las imágenes visibles (percibidas por el ojo humano), en este se encuentran otros subconjuntos: fotografías, dibujos, pinturas; también está el subconjunto de las imágenes ópticas (generadas por lentes y hologramas).

Las imágenes físicas no visibles son distribuciones de propiedades físicas medibles, por ejemplo, temperatura, presión, altitud, densidad de población. El otro conjunto contiene las imágenes abstractas, entre las que se encuentran las funciones matemáticas, las funciones continuas y discretas o imágenes digitales (solo las imágenes digitales pueden ser procesadas por la computadora).

1.1.3 ELEMENTOS DEL PROCESAMIENTO DIGITAL.

Los elementos básicos de un sistema de procesamiento digital de imagen se mostrados en la figura 1.1.4

A) Procesador de imagen.

El procesador de la imagen es el corazón de un sistema de procesamiento digital de imágenes. Un procesador de imagen consiste en módulos hardware que ejecutan cuatro funciones básicas: adquisición de la imagen, almacenamiento, procesamiento de bajo nivel y despliegue.

INTRODUCCION

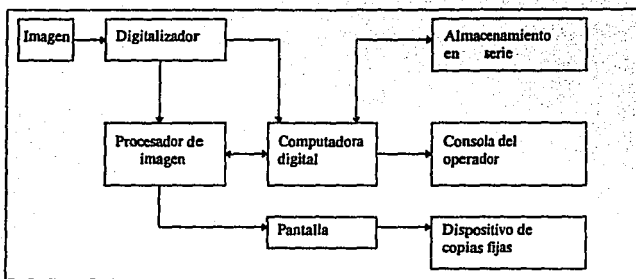


Figura 1.1.4 Elementos de un sistema de procesamiento digital de imagen.

El módulo de adquisición de imagen, también llamado framegrabber, generalmente tiene como entrada una señal de video de TV y convierte esta señal a su forma digital. El módulo de almacenamiento, también llamado memoria de refrescamiento, es una memoria capaz de almacenar una imagen digital entera. El módulo de procesamiento ejecuta funciones de bajo nivel como operaciones aritméticas y lógicas, este módulo es también llamado Unidad Aritmética Lógica (ALU). La función del módulo de despliegue es de leer una imagen de memoria, convertir la información digital en una señal de video analógica y direccionar la señal de salida a un monitor u otro dispositivo.

B) Digitalizador

Un digitalizador convierte una imagen a una representación numérica propia para una entrada en la computadora digital. Entre los dispositivos más comúnmente usados están los microdensitómetros, lentes luminosos de los scanners, analizadores de imagen y cámaras vidicon. Los dos primeros requieren de la imagen digitalizada, en forma de la transparencia o

INTRODUCCION

fotografía; los restantes también las aceptan pero tienen la ventaja adicional de digitalizar imágenes naturales que tengan suficiente intensidad de luz para excitar el detector.

C) Computadora digital.

Los sistemas de computación usados para procesar una imagen van desde microprocesadores hasta grandes sistemas de computación capaces de ejecutar funciones computacionalmente intensivas dentro de grandes arreglos de imagen. Los parámetros que influyen en la estructura de una computadora dependen de la aplicación y los datos que se requieran para llevarla a cabo. Una imagen digital requiere de gran cantidad de memoria durante el procesamiento.

Una imagen digital que consiste en 512×512 píxeles, cada una de los cuales es cuantizada dentro de ocho bits y requiere de 0.25 megabytes de almacenamiento. Entre los dispositivos de almacenamiento usados para este trabajo son discos magnéticos, por ejemplo un disco de 700 Mb de capacidad almacenaría 2800 imágenes del tamaño antes mencionado.

Los monitores monocromático y en color son los principales dispositivos de despliegue usados en los sistemas de procesamiento de imágenes actuales. Los monitores son direccionados por las salidas del módulo de despliegue de imagen dentro del procesador de imágenes. Estas señales pueden ser alimento dentro del dispositivo de grabación de imagen cuya función produce una copia fija (pantalla fotográfica, fotografía y transparencias) de la esencia de la imagen dentro de la pantalla del monitor. Otros medios de despliegue incluyen el tubo de rayos catódicos (TRC) y dispositivos de impresión.

INTRODUCCION

1.1.4 CARACTERISTICAS DE LAS IMAGENES

1.1.4.1 ESCALA DE NIVELES DE GRIS

Las imágenes digitalizadas tienen únicamente dos niveles de escala de gris: blanco y negro. Se pueden tener con 8 bits, 256 niveles que van desde el blanco hasta el negro, pasando por una escala de gris con una alta resolución. La resolución de una imagen depende en gran parte de los niveles de la escala de gris.

El ojo humano puede diferenciar aproximadamente 40 diferentes matices de gris, en la práctica el número empleado rebasa la capacidad del ojo humano.

1.1.4.2 LUMINANCIA

Desde que la luz es una forma de energía, $f(x, y)$ debe ser finita y no ser cero, esto es,

$$0 < f(x, y) < \infty \quad (1.1)$$

Las imágenes que percibimos en nuestras actividades diarias consisten de luz reflejada de los objetos. La naturaleza básica de $f(x, y)$ puede ser considerada como su esencia caracterizada por dos componentes. Un componente es la suma del origen de la luz, incidida sobre la escena vista, mientras que el otro es la suma de la luz reflejada por el objeto en la escena. Estos componentes son llamados iluminación y reflejo de los componentes, y

INTRODUCCION

son denotados por $i(x, y)$ y $r(x, y)$, respectivamente. Las funciones $i(x, y)$ y $r(x, y)$ combinadas como un producto forman $f(x, y)$:

$$f(x, y) = i(x, y) \cdot r(x, y) \quad (1.2)$$

donde

$$0 < i(x, y) < \infty \quad (1.3)$$

y

$$0 < r(x, y) < 1 \quad (1.4)$$

La ecuación 1.4 indica el factor de reflejancia, el cual es limitado por 0 (total absorción) y por 1 (total reflejancia). La naturaleza de $i(x, y)$ es determinada por el origen de la luz, mientras que $r(x, y)$ es determinada por las características de el objeto en la escena.

La intensidad de una imagen monocromático f en coordenadas (x, y) es llamado nivel de gris de la imagen en ese punto desde la ecuación 1.2 hasta 1.4, es evidente que f es falso en el rango

$$L_{\min} \leq f \leq L_{\max} \quad (1.5)$$

En teoría, se requiere que $L_{\min} = (i_{\min}) \cdot (r_{\min})$ y $L_{\max} = (i_{\max}) \cdot (r_{\max})$. Usando por encima los valores de iluminación y reflejancia como una norma, uno puede esperar los valores de L_{\min} es de aproximadamente 0.005 y para L_{\max} es de aproximadamente 100 dentro de aplicaciones del procesamiento de imágenes.

El intervalo $[L_{\min}, L_{\max}]$ es llamado escala de gris. En la práctica es común variar estos intervalos numéricos a intervalos $[0, 1]$ donde $L = 0$ es considerado como negro y $L = 1$ es

INTRODUCCION

considerado como blanco en esta escala. Todos los valores intermedios son considerados matices de gris variando continuamente desde el negro al blanco.

1.1.4.3 Color

La luz es una forma de energía electromagnética, esto es que, la imagen esta dada por la distribución de ondas largas. No toda la radiación electromagnética es visible para el ojo humano. De hecho, la porción entera visible de la radiación es solo dentro del límite de la banda de onda larga de 380 a 780 nm. (Figura 1.1.5). Esta radiación cuando incide sobre el ojo produce una variedad de sensaciones, por ejemplo, la sensación de brillo producida por el conjunto de varias ondas largas. Sin embargo, la luz de varias ondas largas también producen otra importante información visual llamada color.

Las ondas largas de la luz producen un estímulo visual. Como estas ondas varían, los colores percibidos (o matiz) de los cambios de luz varían, desde el rojo (corresponde a la onda larga mas larga) pasando por el naranja, amarillo, verde y del azul al violeta. Las

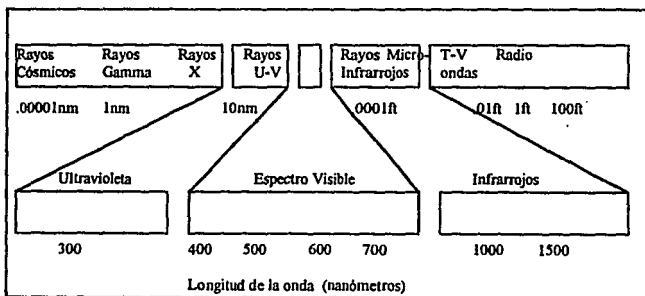


Figura 1.1.5 Espectro de energía electromagnética, mostrando la porción del espectro visible.

INTRODUCCION

mezclas de las ondas largas producen la percepción de matices, algunos de los cuales (por ejemplo, el púrpura) no pueden ser obtenidos desde una sola onda larga. La luz blanca (o gris) también puede ser obtenida por una mezcla de colores, de hecho, cualquier matiz puede ser obtenido por una mezcla de tres colores primarios (rojo, verde y azul) en la proporción adecuada (figura 1.1.6).

El color es un importante aspecto de energía radiante visible por el cual un observador puede distinguir entre diferentes composiciones espectrales. Por eso, un estímulo de color, es especificado por energía radiante visible de una intensidad dada y una composición espectral.

Los estímulos de color son generalmente mas agradables para el ojo humano que el estímulo "blanco y negro". El colorido de imágenes blanco y negro por transformaciones de intensidad dentro de los colores es llamado pseudo-color. Esta adición de color regularmente se realiza en proporción a la escala de gris, esto es, a cada valor le corresponderá una determinada escala de gris.

La especificación del color consiste de tres partes: (a) Semejanza del color, (b) diferencias del color, (c) Apariencia del color (o percepción del color).

El estudio del color es importante en diseño y desarrollo en sistemas de visión del color. El uso del color en el despliegue de imágenes no solo es más agradable, sino que facilita la recepción de más información visual. El ojo humano puede distinguir sólo cerca de 32 sombras de gris discreto; por el contrario es menos limitado en la discriminación del color y es capaz de distinguir por arriba de varios cientos de sombras de color discreto. Por esta razón, el color es una importante herramienta en el realce de la imagen.

INTRODUCCION

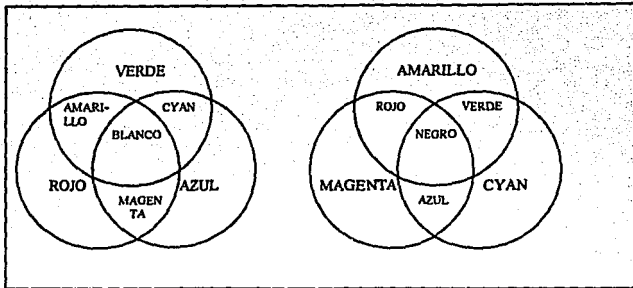


Figura 1.1.6. Los tres colores primarios y los matices obtenidos al ser mezclados.

El color varias veces es usado en la interpretación de imágenes que podrían desplegarse en blanco y negro. Cuando esto es así, el color es adherido como un mecanismo de ayuda al ser humano en la interpretación del análisis de datos. Una imagen de 8 bits contiene intensidades desde cero hasta 255 que pueden ser transformadas dentro de una imagen en color. El color producido en una imagen monocromático dentro de la presentación de colores es llamada falso color de imagen. Esto no tiene relación entre el color producido en el falso color de imagen y el verdadero color del objeto que fue scaneado.

El falso color en las imágenes es producido por intensas transformaciones. La intensidad de los rangos de entrada del digitalizador de imagen es transformada dentro de tres rangos de intensidad separados a la salida: uno es para el rojo, otro para el verde y otro para el azul, estos componentes de un color salen de la composición de la imagen, es decir, de una entrada particular de imagen, se producen tres salidas de imagen.

INTRODUCCION

Los atributos de la percepción son el brillo, el matiz y la saturación. Los brillos presentan la luminancia percibida. El matiz de un color se refiere a la "rojura", "verdor" y así sucesivamente. Para el origen de luces monocromáticas, la diferencia en matices son manifestados por la diferencia de ondas largas. La saturación es ese aspecto, es la percepción de más y más luz sumada a una luz monocromático. Estas definiciones son algo imprecisas porque el matiz, la saturación y el brillo cambian cuando la onda larga, la intensidad, el matiz o la suma de la luz blanca en un color es cambiado.

1.1.5 IMAGENES MULTIESPECTRALES

Las técnicas de procesamiento digital de imágenes no son inherentemente limitadas para las imágenes "blanco y negro" o "color sencillo". La figura 1.1.7 muestra un concepto de un sensor que adquiere una imagen digital de la misma escena en diferentes partes del espectro electromagnético. La misma entrada de la imagen es pasada a través de tres filtros espectrales, dentro del cual se separa la imagen en tres componentes espectrales. Cada componente es entonces digitalizado, de modo que uno de los tres componentes de la imagen en color de la escena que representa, es producido. En este ejemplo, los tres filtros corresponden al rojo, verde y azul en respuesta a las regiones de la película de color convencional, y las tres imágenes digitales podían ser representaciones de los componentes espectrales rojo, verde y azul de la escena original.

Una variedad de técnicas son usadas para el proceso, manipulación y despliegue de imágenes multiespectrales. El concepto de imagen multiespectral es bastante general, las

INTRODUCCION

técnicas de procesamiento multiespectral de imagen pueden involucrar la manipulación de la imagen conteniendo más de tres componentes espectrales.

La Figura 1.1.8 ilustra el concepto más general de una imagen multiespectral, en el cual un gran número de componentes de imagen son representadas. Sobre el lado izquierdo de la figura 1.1.8, una imagen multiespectral muestra 11 componentes espectrales separados. Varios de los componentes de la imagen multiespectral no son derivaciones originales de los sensores de imagen. Sobre el lado derecho de la figura 1.1.8 incluyen tres componentes de imagen (rojo, verde y azul) más otros componentes de imagen que son representados por valores de población, elevación y calidad del aire. En este caso, el formato de la imagen digital ha sido usada para la representación del dato que no es adquirido originalmente por un sensor de imagen.

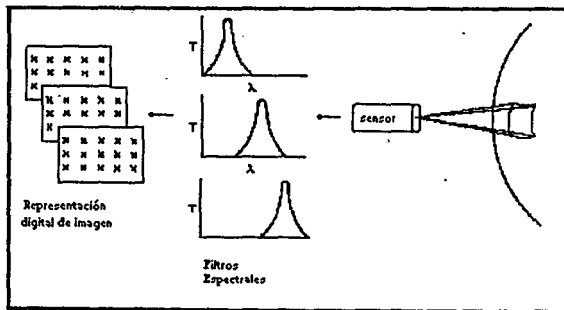


Figura 1.1.7 Concepto de una imagen multiespectral.

El procesamiento digital de imagen provee el establecimiento de técnicas para combinación, unión, análisis y despliegue múltiple de desiguales tipos de datos.

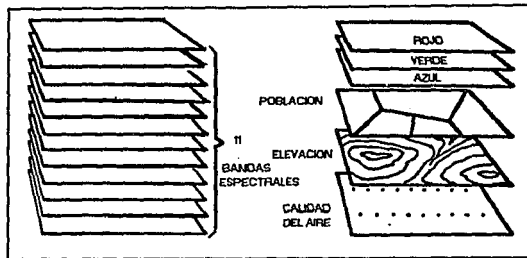


Figura. 1.1.8 concepto generalizado de imagen multispectral.

1.1.6. MUESTREO

En el sistema de procesamiento digital de imágenes, una parte importante es el muestreo de imágenes el cual consiste en tomar puntos representativos de la imagen, formar un arreglo de números para posteriormente procesarlos, de aquí se obtiene otro arreglo de números los cuales son usados para la reconstrucción de la imagen.

El proceso de muestreo es fundamental no solo para preparar las señales para ser transmitidas, sino que también es importante para el análisis de datos y en el procesamiento por computadora.

Una señal muestreada $f^*(t)$ contiene toda la información de una señal $f(t)$ y de $f_s(t)$. Considerando una señal $f(t)$ limitada en banda B Hertz, con una transformada de Fourier como la que se muestra en la figura 1.1.9, puede demostrarse que información no se destruye en el proceso de muestreo. Utilizando una señal de conmutación o de retención $S(t)$ (figura 1.1.10), la cual tiene un ancho de pulso τ que para fines prácticos es muy pequeño.

INTRODUCCION

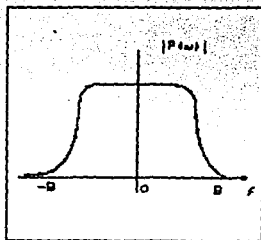


Figura 1.9 Señal de banda limitada

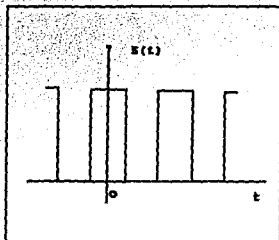


Figura 1.10 Función de conmutación periódica

El proceso de muestreo consiste en la multiplicación de las dos señales en el tiempo, es decir:

$$f^*(t) = f(t) S(t) \quad (1.6)$$

lo que nos da como resultados la señal mostrada en la figura 1.1.11. Si se toman muestras muy espaciadas, ocurre el fenómeno de traspase de los espectros, por lo que se deduce que la frecuencia de muestra es igual a¹:

$$f_c(t) \geq 2 AB \quad (1.7)$$

¹ Para más detalle véase, Capítulo 3, Mischa Schwartz, Transmisión de información y ruido, Mc Graw-Hill.

INTRODUCCION

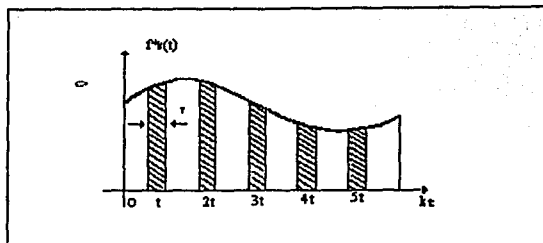


Figura 1.1.11 Proceso de muestreo.

En el análisis espectral se observa que los componentes armónicos se alejan o se aproximan unas de otras dependiendo de las frecuencias de muestreo f_c (figura 1.1.12).

Con lo anterior se puede comprobar que el proceso de muestreo no destruye la información de $f(t)$ ².

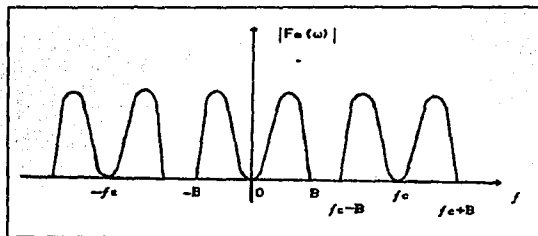


Figura 1.1.12 Espectro de amplitud

En el caso de muestreo de imágenes el procesamiento es similar. El término imagen se refiere a una función de dos dimensiones luz-intensidad, denotada por $f(x, y)$, en donde el

²Para más detalles del proceso de muestreo vease, Capítulo 3, Mischa Schwartz, Transmisión de información Modulación y Ruido, Mc Graw-Hill.

INTRODUCCION

valor o amplitud de f en coordenadas espaciales (x, y) da la intensidad (brillo) de la imagen en un punto.

Para una forma conveniente en el procesamiento por computadora, una función de imagen $f(x, y)$ debe ser digitalizada en espaciamiento y en amplitud. La digitalización de las coordenadas espaciales (x, y) serán referidas como muestreo de imagen, mientras que la amplitud de digitalización será llamada cuantización de nivel de gris.

Suponemos a $f(x, y)$ como una imagen continua, dentro de un arreglo de muestras de la forma $N \times N$ como se observa en la ecuación 1.10 donde cada elemento del arreglo es una cuantización discreta:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,N-1) \end{bmatrix} \quad (1.8)$$

El lado derecho de la ecuación representa lo que comúnmente es llamado imagen digital, donde cada elemento del arreglo es referido como pixel.

Los procesos de digitalización requieren de tomar una decisión para determinar un valor de N y también sobre el número de niveles de gris discretos permitidos para cada pixel. Esto es en la práctica común del procesamiento digital de imágenes para permitir que estas cantidades sean entero de potencia de base dos es decir,

$$N = 2^n \quad (1.9)$$

y

$$G = 2^m \quad (1.10)$$

INTRODUCCION

donde G denota el numero del nivel de gris. Esos niveles discretos son igualmente espaciados entre 0 y l dentro de la escala. Usando la ecuación 1.9 y 1.10 el numero, δ , de bits requeridos para almacenar una imagen esta dado por

$$\delta = N \times N \times m \quad (1.11)$$

Por ejemplo, una imagen de 128×128 con 64 niveles de gris requiere de 98,304 bits de almacenamiento. La Tabla 1.1 resume los valores de δ para rangos típicos de $N \times m$. La Tabla 1.2 da los números correspondientes de 8 bits. Generalmente, esto no es práctico desde el punto de vista de la programación para llenar un byte completamente, si ésta emplea el traslapamiento de un pixel desde un byte a al siguiente. Así en la tabla 1.1 representa el número de bytes necesarios para cada valor de N y m donde no es permitido el traslapamiento. Por ejemplo, si $m = 5$, se asume que solo un pixel es almacenado en un byte, se pierden tres bits que no fueron usados del byte.

$N \backslash m$	1	2	3	4	5	6
32	1,024	2,048	3,072	4,096	5,120	6,144
64	4,096	8,192	12,288	16,384	20,480	24,576
128	16,384	32,768	49,152	65,536	81,920	98,304
256	65,536	131,072	196,608	262,144	327,680	393,216
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864

Tabla 1.1 Numero de bit de almacenamiento para varios valores de N y m

INTRODUCCION

Puesto que la ecuación 1.8 es una aproximación a una imagen continua, una pregunta razonable a responder de este punto es ¿cuántas muestras y niveles de gris son necesarios para una buena aproximación?. La resolución (ejemplo, el grado de detalle deseable) de una imagen es dependiente de N y m , entre más se incrementa el parámetro, el arreglo digitalizador será más próximo a la imagen original. Sin embargo, la ecuación 1.11 claramente señala el hecho infortunado que, almacenar una imagen hace consecuentemente que los requerimientos de procesamiento de imágenes se incrementan rápidamente como una función de N y m .

En vista del comentario anterior, se debe considerar el efecto de las variaciones que se tiene en N y m dentro de la calidad de la imagen. Como es de sospecharse, una "buena" imagen, es difícil de definir porque la calidad requiere de variaciones acordes a la aplicación.

La figura 1.1.12 muestra el efecto de reducir el tamaño de la rejilla de muestreo en la imagen. La figura 1.1.12a es de 512×512 , 256 niveles, en la imagen se muestra un astronauta durante el primer alunizaje en la luna. De la figura 1.1.12b hasta la 1.1.12f se muestra la misma imagen, pero con $N = 256, 128, 64, 32$ y 16 . En todos los casos el máximo número de niveles de gris permitidos es de 256.

$N \backslash m$	1	2	3	4	5	6
32	128	256	512	512	1,024	1,024
64	512	1,024	2,048	2,048	4,096	4,096
128	2,048	4,096	8,192	8,192	16,348	16,348
256	8,192	16,384	32,768	32,768	65,536	65,536
512	32,768	65,536	131,072	131,072	262,144	262,144

Tabla 1.2 Numero de 8 bytes de almacenamiento para varios valores de N y m .

INTRODUCCION

El área usada para cada imagen será la misma (512 x 512 puntos desplegados), los píxeles en baja resolución de imagen se duplica en orden para ocupar el campo entero de despliegue. Este produce un efecto de chequeo, el cual es particularmente importante en la baja resolución de imagen. Esto denota en la imagen de 256 x 256 que es mas cerrada que la de la figura 1.1.12a pero la calidad de la imagen se deteriora rápidamente por los otros valores de N.

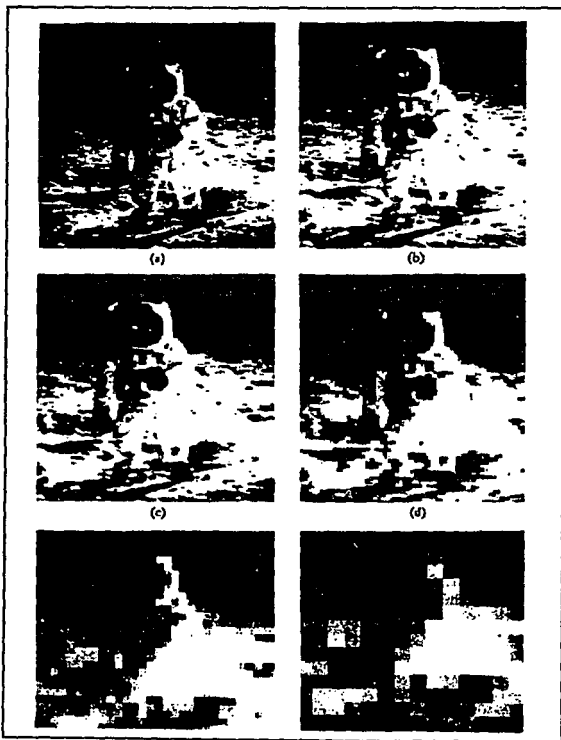


Figura 1.1.12 Efectos de la reducción del tamaño de la rejilla de muestreo.

INTRODUCCION

1.2 FUNDAMENTOS DE TECNOLOGIA ORIENTADA A OBJETOS

Para el desarrollo del software de procesamiento de imágenes (WPDI), es conveniente buscar una técnica que nos permita el poder utilizar lo que ya se ha hecho, y que también en caso necesario, en un futuro nos permita modificar e incrementar lo más fácilmente posible el programa. El paradigma actual en la programación se llama Programación Orientada a Objetos (P.O.O.), para ello es necesario realizar un Análisis y un Diseño Orientado a Objetos (A.O.O. y D.O.O. respectivamente).

1.2.1 FUNDAMENTOS SOBRE OBJETOS

Debido a la creciente complejidad en el desarrollo de software, se ha hecho necesario el buscar cada vez mejores técnicas que nos permitan poder sacar el mayor provecho; por una parte de lo que ya se ha desarrollado, es decir reutilizar todo lo más posible, y por otra parte, poder crear aplicaciones nuevas lo más rápidamente posibles, de mayor eficiencia, que sean fáciles de mantener y modificar.

Debido a estas necesidades surgió el paradigma de la orientación a objetos, que trata el software como objetos reales, es decir como unidades de diferentes tipos por medio de las cuales podemos ir armando diferentes objetos de mayor complejidad, y así sucesivamente hasta lograr grandes niveles de complejidad.

Algunos de los conceptos fundamentales para comprender la tecnología orientada a objetos son los siguientes:

INTRODUCCION

Objetos

Métodos

Encapsulamiento

Mensajes

Clases

Herencia

Aunque estos conceptos son la base del software orientado a objetos, son similares a los que forman la base de todos los seres vivos.

1.2.2 ¿QUE ES LA PROGRAMACION ORIENTADA A OBJETOS?

La Programación Orientada a Objetos es un método de implementación en el cual los programas son organizados como colecciones de objetos, cada uno de los cuales representa una instancia de alguna clase, y cuyas clases son todas miembros de una jerarquía de clases unidas vía relaciones de herencia.

Hay tres partes importantes en esta definición.

- 1) La Programación Orientada a Objetos usa objetos, no algoritmos, como su bloque fundamental lógico de construcción.

INTRODUCCION

- 2) Cada "objeto" es una instancia de alguna clase.
- 3) Las clases están relacionadas a otra clase vía relación de herencia.

Un programa puede parecer ser orientado a objetos, pero si cualquiera de estos elementos falta, entonces no se trata de un programa orientado a objetos, le llamamos programación con tipos de datos abstractos.

1.2.3. ¿QUE ES UN OBJETO?

Las personas nos formamos conceptos desde temprana edad. Cada concepto es una idea particular o una comprensión de nuestro mundo. Los conceptos adquiridos nos permiten sentir y razonar acerca de las cosas en el mundo. A estas cosas a las que se aplican nuestros conceptos se llaman objetos. Un objeto podría ser real o abstracto como por ejemplo un automóvil, una organización, etc.

En el análisis y diseño orientado a objetos nos interesa el comportamiento del objeto. Si construimos software, los módulos de software O.O. se basan en los tipos de objetos. El software que implanta el objeto contiene estructuras de datos y operaciones que expresan dicho comportamiento, ver figura 1.2.1.

Las operaciones se codifican como métodos. La representación en software orientado a objetos del objeto es entonces una colección de tipos de datos y métodos, es decir, un objeto es cualquier cosa real o abstracta, acerca del cual almacenamos datos y los métodos que controlan dichos datos.

INTRODUCCION

1.2.4. ¿QUE ES UN TIPO DE OBJETO?

Los conceptos que poseemos se aplican a tipos determinados de objetos. Por ejemplo, empleado se aplica a los objetos que son personas empleadas por alguna organización. Algunas instancias de empleado podrían ser Juan Pérez, María Sánchez, etc. En el análisis orientado a objetos, estos conceptos se llaman tipos de objetos; las instancias se llaman objetos.

Un tipo de objeto es una categoría de objeto.

Un objeto es una instancia de un tipo de objeto.

En el mundo O.O. la estructura de datos y los métodos de cada tipo de objeto se manejan juntos. No se puede tener acceso o control de la estructura de datos excepto mediante los métodos que forman parte del tipo de objeto.

1.2.5. METODOS

Los métodos especifican la forma en que se controlan los datos de un objetos. Los métodos en un tipo de objeto solo hace referencia a las estructuras de datos de ese tipo de objeto. No deben tener acceso directo a las estructuras de datos de otros objetos. Para utilizar la estructura de datos de otro objeto, se debe enviar un mensaje a éste. El tipo de objeto empaca juntos los tipos de datos y los métodos.

INTRODUCCION

Un objeto es entonces una cosa cuyas propiedades están representadas por tipos de datos y su comportamiento por métodos.

Un método asociado con el tipo de objeto *factura* podría ser aquel que calcule el total de una factura. Otro podría transmitir la factura a un cliente, otro podría verificar de manera periódica si la factura ha sido pagada, y en caso contrario, añadir cierta tasa de interés.

1.2.6. ENCAPSULAMIENTO

El empaque conjunto de datos y métodos se llama encapsulado o encapsulamiento. El objeto esconde sus datos de los demás objetos y permite el acceso a los datos mediante sus propios métodos. Esto recibe el nombre de ocultamiento de la información. El encapsulamiento evita la corrupción de los datos de un objeto. Si todos los programas pudieran tener acceso a los datos en cualquier forma que quisieran los usuarios, los datos se podrían corromper o usar de mala manera. El encapsulamiento protege los datos del uso arbitrario y no pretendido.

El encapsulamiento oculta los detalles de su implantación interna a los usuarios de un objeto. Los usuarios se dan cuenta de las operaciones que pueden solicitar del objeto pero desconocen los detalles de como se lleva a cabo la operación. Todos los detalles específicos de los datos del objeto y la codificación de sus operaciones están fuera del alcance del usuario.

INTRODUCCION

Encapsulado es el resultado (o acto) de ocultar los detalles de implantación de un objeto respecto de su usuario (Figura 1.2.3).

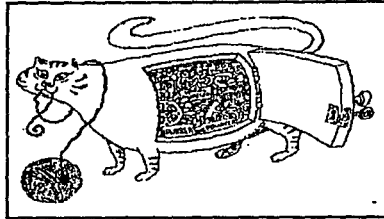


Fig. 1.2.3 El encapsulamiento oculta los detalles de la implementación de un objeto.

El encapsulamiento es importante porque separa el comportamiento del objeto de su implantación. Esto permite la modificación de la implantación de un objeto sin que se tengan que modificar las aplicaciones que la utilizan.

Es más fácil modificar los programas que utilizan el encapsulado porque se modifica al mismo tiempo un tipo de objeto. Si esto ocurre sólo se afectan los métodos y las estructuras de datos asociados con dicho tipo de objeto; lo usual es que solo se vean afectados algunos de los métodos y las estructuras de datos. El comportamiento del tipo objeto se puede modificar y probar de manera independiente a los demás tipos de objetos.

INTRODUCCION

1.2.7. MENSAJES

Para que un objeto haga algo, le enviamos una solicitud, ésta hace que se produzca una operación. La operación ejecuta el método apropiado y, de manera opcional, produce una respuesta. El mensaje que constituye la solicitud contiene el nombre del objeto, el nombre de una operación y, a veces, un grupo de parámetros.

La programación orientada a objetos es una forma de diseño modular en la que, con frecuencia, el mundo se piensa en términos de objetos, operaciones, métodos y mensajes que se transfieren entre tales objetos.

Un solicitud invoca una operación específica, con uno o más objetos como parámetros. Los objetos pueden ser muy complejos, puesto que pueden contener muchos subobjetos. La persona que utilice el objeto no tiene que conocer su complejidad interna, sino la forma de comunicarse con él y la forma en que responde.

1.2.8. CLASE

El término clase se refiere a la implantación en software de un tipo de objeto.

El tipo de objeto es una noción de concepto. Especifica una familia de objetos sin estipular la forma en que se implanten. Los tipos de objetos se especifican durante el análisis O.O. sin embargo, al implantar los tipos de objetos, se utilizan otros términos.

En los lenguajes orientados a Objetos los tipos de objetos se implantan como clases.

INTRODUCCION

Una clase es una implantación de un tipo de objeto, especifica una estructura de datos y los métodos operativos permisibles que se aplican a cada uno de sus objetos.

Cada clase define un conjunto de operaciones permisibles que permiten el acceso y modificación de los datos del objeto. Una clase Empleado podría incluir operaciones como contratar, promover y cambiar número de extensión para una clase.

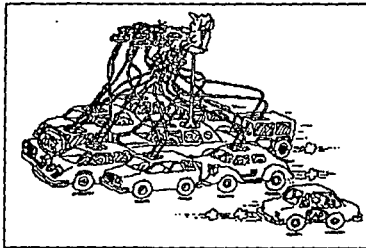


Fig. 1.2.4 Una clase representa a un grupo de objetos que muestran una estructura y un comportamiento comunes.

1.2.9. HERENCIA

Como en los seres vivos los hijos heredan las características y comportamiento de los padres, así también en la tecnología orientada a objetos se tiene el concepto de herencia, en donde las clases "hijas" heredan el comportamiento y atributos de su clase "padre", ha este hecho le llamamos herencia simple, pero también puede darse el caso que una clase "hija"

INTRODUCCION

herede características de dos o más clases, a lo cual nos referimos como herencia múltiple.

Cabe hacer notar que no todos los lenguajes orientados a objetos soportan la herencia múltiple (Figura 1.2.5).

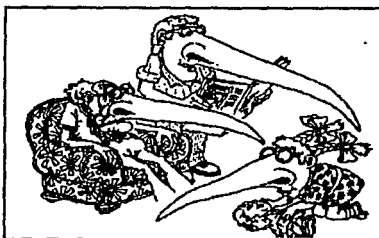


Fig. 1.2.5 Una subclase puede heredar la estructura y comportamiento de su superclase.

CAPITULO

2

CAPITULO 2

TECNICAS DE PROCESAMIENTO DIGITAL DE IMAGENES

2.1. PROCESAMIENTO DE IMAGENES EN EL DOMINIO ESPACIAL

El sistema WPDI (Procesamiento Digital de Imágenes para Windows) el cual es el objetivo final de este trabajo de tesis, esta basado en técnicas de procesamiento digital de imágenes en el dominio del espacio, que son ampliamente usadas y que nos ayudan a resolver una gran cantidad de problemas dentro de esta área.

El objetivo principal de las técnicas que utilizamos para la creación del sistema WPDI, es el de procesar una imagen, y que el resultado de la información obtenida de la imagen restaurada nos sea de mayor utilidad para una aplicación específica, que la que tenemos acerca de la imagen original. La palabra "específica" es importante porque establece el comienzo de que técnicas discutiremos a continuación. Por ejemplo, un método bastante usado para el mejoramiento de imágenes de Rayos-X, no puede ser necesariamente la mejor aproximación en el mejoramiento de una imagen de Marte transmitida por una base espacial.

El procesamiento de imágenes se puede dividir en dos categorías: dominio espacial y dominio de la frecuencia. En esta parte del capítulo vamos a tratar al dominio espacial, el cual se refiere a una imagen plana en sí misma. Las aproximaciones en esta categoría están basadas en la manipulación directa de los píxeles de la imagen.

2.1.1 Histograma

Un histograma que contiene niveles de gris, proporciona una descripción global de la apariencia de una imagen. El método usado para el desarrollo de este trabajo es el de realzamiento por modificación del histograma de una imagen dada de una manera específica. El tipo y grado de realzamiento obtenido depende de la naturaleza del histograma establecido.

2.1.1.1 Fundamentos

En esta parte se establecen los fundamentos por los cuales vamos a usar el histograma como herramienta de sistema WPD1 y como se formula. La variable r representa los niveles de gris de los pixeles de la imagen. Por simplicidad será asumido, en esta discusión que, los valores de los pixeles van ha ser normalizados en el rango de

$$0 \leq r \leq 1 \quad (2.1.1)$$

con $r = 0$ representado por negro y $r = 1$ por blanco en la escala de grises .

Para cualquier r en el intervalo de $[0, 1]$, enfocaremos nuestra atención a la transformación que tiene la forma

$$s = T(r) \quad (2.1.2)$$

la cual produce el nivel s para todos los valores de r en la imagen original. Esta es asumida como la transformación de la función dada por $s=T(r)$ satisfaciendo las condiciones:

(a) $T(r)$ es un valor singular y monótonicamente incrementado en el intervalo

$$0 \leq r \leq 1$$

(b) $0 \leq T(r) \leq 1$ para $0 \leq r \leq 1$

la condición (a) preserva el orden desde negro hasta blanco en la escala de grises mientras la condición (b) garantiza un mapeo consistente con el rango permitido del valor del pixel. Una transformación de la función que satisface esta condición es ilustrada en la figura 2.1.1.

La transformación inversa desde la última s hasta r será determinada por

$$r = T^{-1}(s) \quad \text{con} \quad 0 \leq s \leq 1 \quad (2.1.3)$$

donde se asume que $T^{-1}(s)$ también satisface las condiciones (a) y (b) con respecto a la variable s .

Los niveles de gris en una imagen son aleatorios cualitativamente en el intervalo $[0, 1]$, asumiendo por el momento a estas como variables continuas. El nivel de gris original y transformado puede ser caracterizado por su función de densidad de probabilidad $Pr(r)$ y $P_s(s)$ respectivamente. Un gran trato puede ser decir, acerca de las características generales de la imagen desde la función de densidad de estos niveles de gris por ejemplo, una imagen cuyos niveles de gris tiene una función parecida a la mostrada en la fig. 2.1.2(a) tendría

justamente características oscuras ya que la mayoría de estos niveles, están concentrados en la región oscura de la escala de gris. Una imagen cuyos niveles de gris tienen una función de densidad parecida a la mostrada en la fig. 2.1.2 (b) en el otro lado tendrían predominancia los tonos suaves, entonces la mayoría de estos pixeles son gris tenue.

Siguiendo esta elemental teoría de probabilidad decir que, $Pr(r)$ y $T(r)$ son conocidos y $T^{-1}(s)$ satisface la condición (a), entonces la probabilidad de la función de densidad

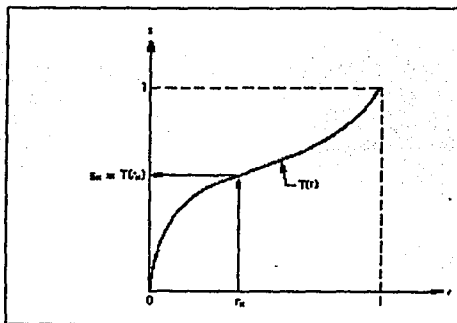


Figura 2.1.1 Función de transformación de un nivel de gris

de la transformación de niveles de gris esta dada por la relación:

$$p_s(s) = \left[p_r(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)} \quad (2.1.4)$$

la siguiente técnica de realzamiento está basada en la modificación de la apariencia de una imagen controlando los niveles de gris por la función de densidad de probabilidad via la función de transformación $T(r)$.

2.1.1.2 ECUALIZACION DEL HISTOGRAMA

Como ya vimos y considerando la función de transformación como

$$s = T(r) = \int_0^r p_r(w)dw \quad 0 \leq r \leq 1 \quad (2.1.5)$$

donde w es la variable cambiante de integración. La mayor parte del lado derecho de la ecuación 2.1.5 es reconocida como la función de distribución acumulativa (CDF) de r . Las dos condiciones dadas a conocer en la parte anterior son satisfechas por la función de transformación entonces la CDF incrementa monótonamente desde 0 hasta 1 como una función de r . Las dos condiciones dadas a conocer en la parte anterior son satisfechas por

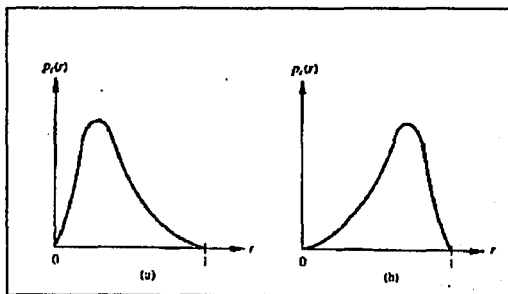


Figura 2.1.2 Funciones de densidad de probabilidad de niveles de gris.

(a) Una imagen "oscura" y (b) una imagen "clara".

la función de transformación entonces, la CDF incrementa monótonamente desde 0 hasta 1 como una función de r .

De la Eq. 2.1.5, la derivada de s con respecto a r esta dada por

$$\frac{ds}{dr} = P_r(r) \quad (2.1.6)$$

sustituyendo dr/ds dentro de la ecuación 2.1.4 y reduciendo

$$\begin{aligned} P_s(s) &= \left[P_r(r) \frac{1}{P_r(r)} \right]_{r=T^{-1}(s)} \\ &= [1]_{r=T^{-1}(s)} \\ &= 1 \quad 0 \leq s \leq 1 \end{aligned} \quad (2.1.7)$$

la cual es una densidad uniforme en el intervalo de definición de la variable transformada s . Se nota que este resultado es independiente de la función de transformación inversa. Esto es importante porque no siempre es fácil obtener t de $T^{-1}(s)$ analíticamente.

El desarrollo anterior esta usando la función de transformación igual a la función acumulativa de los productos de r en la imagen cuyos niveles de gris tienen una densidad uniforme. En términos de realzamiento, esto implica, un aumento en el rango dinámico de los pixeles, los cuales, como se verá abajo, puede tener un considerable efecto en la apariencia de una imagen.

Ejemplo: El anterior procedimiento con una discusión de variables discretas permite considerar una simple ilustración de el uso de las ecuaciones (2.1.4) y (2.1.5). Considere que los niveles r tienen una función de probabilidad de densidad mostrada en la Fig. 2.1.4(a) en este caso $Pr(r)$ esta dado por:

$$p_r(r) = \begin{cases} -2r+2 & 0 \leq r \leq 1 \\ 0 & \text{otro} \end{cases}$$

sustituyendo esta expresión en la ec. (2.1.5), y reduciendo la función de transformación tenemos que

$$\begin{aligned} s &= T(r) = \int_0^1 (-2r+2)dw \\ &= \int_0^1 -2w dw + \int_0^1 2dw \\ &= -w^2 + 2w \\ &= -r^2 + 2r \end{aligned}$$

aunque solo necesitamos $T(r)$ para la equalización del histograma este será instructivo para demostrar que la densidad de resultado $Ps(s)$ es factor uniforme desde que este se requiere hasta que $T^{-1}(s)$ se obtiene. En la practica este caso no es requerido porque la Ec. (2.1.7) es independiente de la función inversa de transformación. Resolviendo para r en términos de s construimos

$$r = T^{-1}(s) = 1 \pm \sqrt{1-s}$$

entonces la única solución de r en el intervalo $[0, 1]$ es

$$r = T^{-1}(s) = 1 - \sqrt{1-s}$$

válida.

La función de probabilidad de densidad de s se obtiene por la sustitución del resultado obtenido de la ecuación (2.1.4)

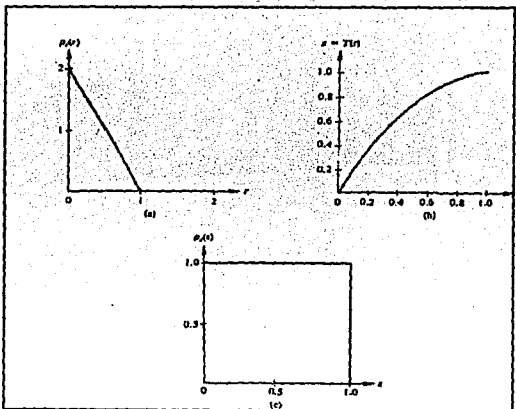


Figura 2.1.3 Ilustración del método transformación de densidad uniforme.
 (a) Función de densidad de probabilidad. (b) Función de transformación. (c) Resultado de densidad uniforme.

$$\begin{aligned}
 P_r(s) &= \left[P_r(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)} \\
 &= \left[(-2r+2) \frac{dr}{ds} \right]_{r=1-\sqrt{1-s}} \\
 &= \left[(2\sqrt{1-s}) \frac{d}{ds} (1-\sqrt{1-s}) \right] \\
 &= 1 \quad 0 \leq s \leq 1
 \end{aligned}$$

la cual es una densidad uniforme en el nivel deseado. La función de transformación $T(r)$ es demostrada en la Fig. (2.1.4).

Para ser útil en procesamiento digital de imágenes, los conceptos mencionados anteriormente deben ser formulado en forma discreta. Para valores de gris que asumen valores discretos trataremos con las probabilidades dadas por la relación:

$$P_r(r_k) = \frac{n_k}{n} \quad \left. \begin{array}{l} 0 \leq r_k \leq 1 \\ k = 0, 1, 2, \dots, L-1 \end{array} \right\} \quad (2.1.8)$$

donde L es el número de niveles, $P_r(r_k)$ es la probabilidad de la k ésimo nivel de gris, n_k es el número de veces que este nivel aparece en la imagen y n es el número total de píxeles en la imagen, por lo tanto, un gráfico de $P_r(r_k)$ vs r_k es normalmente llamado histograma. La técnica usada para obtener un histograma uniforme es, como ya sabemos, ecualización del histograma.

La forma discreta de la Ec. (2.1.5) esta dada por la relación

$$s_k = T(r_k) \sum_{j=0}^k \frac{n_j}{n}$$

$$= \sum_j p_r(r_j) \Big|_{k=0,1,\dots,L-1} \quad \begin{matrix} 0 \leq r_k \leq 1 \\ 0,1,\dots,L-1 \end{matrix} \quad (2.1.9)$$

Tabla 2.1

r_k	n_k	$p_r(r_k) = n_k / n$
$r_0 = 0$	790	0.19
$r_1 = 1/7$	1023	0.25
$r_2 = 2/7$	850	0.21
$r_3 = 3/7$	656	0.16
$r_4 = 4/7$	329	0.08
$r_5 = 5/7$	245	0.06
$r_6 = 6/7$	122	0.03
$r_7 = 1$	81	0.02

La transformada inversa es representada por

$$r_k = T^{-1}(s_k) \quad 0 \leq s_k \leq 1$$

donde ambos $T(r_k)$ y $T^{-1}(s_k)$ satisfacen las condiciones a y b que mencionamos anteriormente; esto significa que la función de transformación $T(r_k)$ puede ser computarizada desde la imagen en cuestión usando la ecuación (2.1.9).

Ejemplo: Supongamos una imagen de 64 x 64 con 8 niveles, tienen una distribución de niveles de gris mostrada en la tabla 2.1.

El histograma de estos niveles de gris es demostrado en la Fig. 2.1.5(a).

La función de transformación es obtenida usando la ecuación (2.1.9). Por instancia

$$\begin{aligned}
 S_0 &= T(r_0) = \sum_{j=0}^L p_r(r_j) \\
 &= pr(r_0) \\
 &= 0.19
 \end{aligned}$$

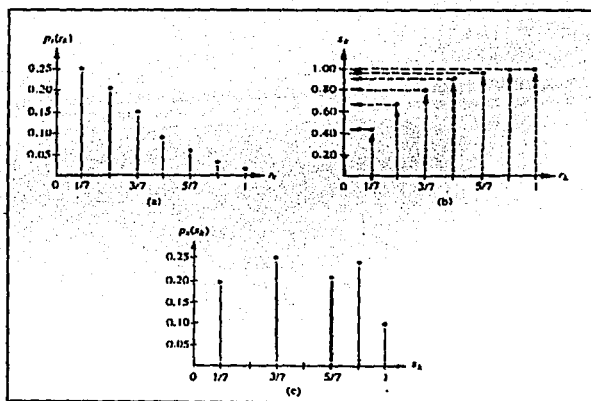


Figura 2.1.4 Ilustración del método de ecuación del histograma (a) Histograma original (b) Función de transformación (c) Ecuación del histograma.

así sucesivamente:

$$S_1 = T(r_1) = \sum_{j=0}^L p_r(r_j)$$

$$= pr(r_0) + pr(r_1)$$

$$= .44$$

y

$$s_2 = 0.65 \quad s_5 = 0.95$$

$$s_3 = 0.81 \quad s_6 = 0.98$$

$$s_4 = 0.89 \quad s_7 = 1.00$$

La función de transformación tiene la forma de escalera mostrada en la fig. 2.1.5 (b).

Entonces 8 niveles igualmente espaciados son alojados en este caso, cada uno de los valores transformados deben ser asignado al nivel valido más cercano. De este modo tenemos

$$s_0 = 1/7 \quad s_4 = 6/7$$

$$s_1 = 3/7 \quad s_5 = 1$$

$$s_2 = 5/7 \quad s_6 = 1$$

$$s_3 = 6/7 \quad s_7 = 1$$

se nota que solo hay 5 distancias de eualización del histograma de niveles de gris.

Redefiniendo la notación para tomar en cuenta esto, producimos los siguientes niveles.

$$s_0 = 1/7 \quad s_3 = 6/7$$

$$s_1 = 3/7 \quad s_4 = 1$$

$$s_2 = 5/7$$

entonces $r_0 = 0$ fue mapeado para $s_0 = 1/7$, hay 790 pixeles transformados con este nuevo valor. También hay 1023 pixeles con valores $s_1 = 3/7$ y 850 pixeles con valor $s_2 = 5/7$. Sin embargo desde los niveles r_3 y r_4 donde mapeamos para $s_3 = 6/7$ hay ahora $656 + 329 = 985$ pixeles con este nuevo valor. Similarmente, hay $245 + 122 + 81 = 448$ pixeles con el valor $s_4=1$. Dividiendo estos números por $n = 4096$ obtenemos el histograma mostrado en la fig. (2.1.5(c)). Entonces un histograma es una aproximación a la función de densidad de probabilidad, rara vez se obtienen resultados perfectamente planos cuando trabajamos con niveles discretos.

Ejemplo: Como una ilustración de ecualización del histograma, consideremos la imagen mostrada en la fig. (2.1.5(a)) la cual es una foto de un dólar, debido a una densa sombra es apenas visible. El angosto rango de valores ocupados por los pixeles de esta imagen es evidente en el histograma mostrado en la fig. (2.1.6(b)). La ecualización del histograma se muestra en la fig. (2.1.5(c)) y el procesamiento de la imagen es mostrada en la fig. (2.1.5(d)). Mientras la ecualización del histograma, como supusimos, no era perfectamente plana por todas partes, de todo el rango de niveles de gris, hay considerable mejoramiento sobre la imagen original que, fue realizado por el efecto dimensional de la técnica de ecualización del histograma. aunque el resultado final no es una fotografía ideal, se debe tener en cuenta la calidad tan pobre de la original.

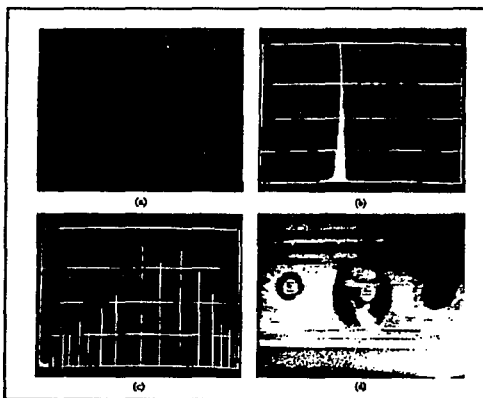


Figura 2.1.5 Ilustración de la equalización del histograma. (a) Imagen original. (b) Histograma original. (c) Ecuación del histograma (d) Imagen realzada.

2.1.2 CONVOLUCION

Convolución es una operación que se relaciona con el procesamiento digital de imágenes y que nosotros utilizamos para la creación del sistema WPDI para la parte de filtros ya que esta operación es usada para filtrar imágenes, para reducir el ruido o para aumentar la visibilidad de una información específica de la imagen.

Convolución puede ser directamente computarizada a imágenes. Alternativamente pueden ser computarizadas después de ser transformadas dentro de un dominio diferente. En esta parte nos concentraremos en el dominio directo o espacial para cuestiones del sistema WPDI. En esta tesis no veremos la implementación en el dominio de Fourier pero lo trataremos teóricamente en una sección posterior. Además de implementar en la computadora, veremos la importancia de esta técnica y las aplicaciones de ella. Algunas de las subalternas teorías de sistemas lineales es también presentado, pero no es necesario entender los resultados que podemos apreciar de la técnica.

2.1.2.1 CONVOLUCION EN EL DOMINIO ESPACIAL

Convolución espacial es una operación de bordes (vecindad) que usa una función de promedio local. En una operación de vecindad computarizamos un resultado a cada pixel de salida utilizando el pixel de entrada correspondiente y los pixeles circunvecinos. Para una convolución el cálculo, es una suma de productos. La Fig. (2.1.6) muestra una máscara de convolución de tres pixeles cuadrados. Los elementos individuales de la máscara son

llamados "PESOS". Los datos en la máscara determinan los efectos de la convolución para definir el filtro a ser empleado. Los valores son determinados desde la "función punto de diseminación" en un filtro en particular. Esta función es la definición del filtro. Expondremos en este trabajo ejemplos de funciones de punto de diseminación particulares, máscaras y sus efectos; por el momento nos concentraremos en el cálculo de la misma.

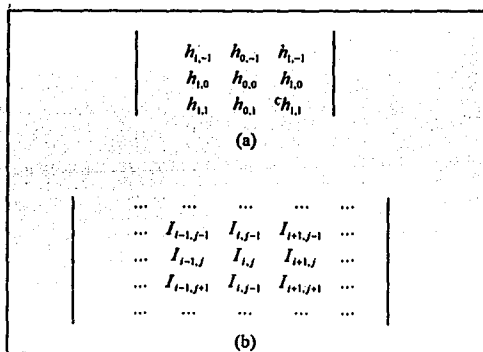


Figura 2.1.6 Convolución espacial (a) Una máscara de convolución 3 x 3 (b) Una vecindad de 3 x 3 dentro de una imagen.

El índice usado en la máscara tiene origen en el pixel central. La imagen es vinculada al índice de la esquina superior izquierda en el camino usual. Usando la terminología de la fig. 2.1.6 podemos escribir la ecuación describiendo el algoritmo como sigue:

$$R_{ij} = \sum_{k=-1}^{+1} \sum_{l=-1}^{+1} I_{i-k,j-l} h_{-k,-l} \quad (2.1.10)$$

Cada pixel en turno es tomado desde la frontera de la imagen y multiplicado por uno de los elementos de la máscara. La suma individual de estos productos constituye el resultado del pixel. Cada pixel participa en exactamente una multiplicación, como lo hace cada elemento de la máscara.

Veremos a continuación porque este algoritmo es útil en procesamiento de imágenes. Antes de hacerlo, existe un patrón en el proceso computacional el cual vale la pena explorar. No es particularmente visible en la ecuación 2.1.10, pero podemos verlo si expandimos la sumatoria explícitamente.

$$R_{ij} = \begin{matrix} I_{i-1,j-1}h_{1,1} + I_{i,j-1}h_{0,1} + I_{i+1,j-1}h_{-1,1} \\ I_{i-1,j}h_{1,0} + I_{i,j}h_{0,0} + I_{i+1,j}h_{-1,0} \\ I_{i-1,j+1}h_{1,-1} + I_{i,j+1}h_{0,-1} + I_{i+1,j+1}h_{-1,-1} \end{matrix} \quad (2.1.11)$$

La ecuación (2.1.11) está Arreglada para que los pixeles contengan términos individuales en la vecindad de la imagen, con una secuencia de la esquina superior izquierda a la esquina inferior derecha. Los elementos que corresponden a la máscara en cada termino están secuenciados precisamente en la dirección opuesta. Una consecuencia interesante de esto, es que si rotamos la máscara 180°, la secuencia de términos en la imagen y la máscara es exactamente la misma. Es como si la máscara rotada estuviera siendo puesta sobre la imagen y las multiplicaciones estuvieran siendo realizadas entre un elemento de la máscara y el pixel de la imagen que se esconde bajo esta. En realidad, algunas implementaciones automáticamente arreglan los elementos de la máscara de tal manera que puedan usarse

máscaras específicas. Muchas máscaras de convolución son simétricas. Consecuentemente no hay cambio cuando la máscara es reordenada. En todo caso, para máscaras asimétricas es importante saber como el sistema interpreta los datos de imagen y que arreglo, si existe, es llevado a cabo.

El ejemplo de la figura 2.1.6 muestra una máscara de tres por tres. Las máscaras no tienen que ser cuadradas pero, por lo regular lo son. Son normalmente números impares de pixeles en el ancho y largo, para que la originalidad de la máscara esté en el centro geométrico. Como con otras operaciones en la vecindad, no es posible computarizar valores de convolución cerca de las fronteras de la imagen, los cálculos requeridos para los valores de los pixeles que están en los límites externos de la entrada de datos de la imagen. Como una alternativa, es común colocar una línea de pixeles alrededor de la frontera de la imagen a cero. El ancho de la línea desde la esquina superior y el fondo es de $(y_m/2)$ y el largo de la línea es $(x_m/2)$. Aquí x_m es el ancho de la máscara y y_m es el largo. La máscara de tres por tres pixeles del ejemplo, dejará una línea grande de pixeles que no pueden ser procesados en la computadora alrededor de la imagen completa. Puede ser importante saber que hace un sistema de procesamiento de imagen en particular en estas circunstancias. Probablemente el archivo resultante del proceso de la imagen es mas pequeño que el archivo original por el numero de imágenes y columnas que la computadora no puede procesar por estar en la frontera de la imagen. Alternativamente el archivo de la imagen puede quedar del mismo tamaño y solo tener cero en la línea alrededor del borde de la imagen. Esta segunda opción es la más popular.

Esto significa que los resultados pueden ser directamente comparados con la original y pueden participar en futuras operaciones computacionales mientras que conservan linealidad. En todo caso, los ceros en la línea del borde afectan los resultados de las imágenes estáticas para algunos grados y pueden tener efectos inversos en otra función de procesamiento de imágenes.

Algunos sistemas pueden ofrecer caminos para completar el proceso computacional correctamente para los bordes en las orillas de la imagen, para usar datos desde otra parte como si fueran adicionados columnas y renglones de pixeles fuera de los limites de la orilla de la imagen. Una técnica semejante es, usando renglones desde la parte superior para suministrar los datos requeridos en las orillas y viceversa. Similarmente las columnas extras en la izquierda son copiadas desde esta, hasta la derecha y viceversa. No hay justificación física para estas técnicas, estas simplemente proporcionan caminos para inventar datos que no están actualmente presentes mientras que evitan las dificultades potenciales que tenemos con las líneas a ceros.

2.1.2.2 FILTROS USANDO CONVOLUCION

Convolución provee un camino para filtrar imágenes. La naturaleza exacta del filtro es determinada por los valores de la máscara de convolución. Esta máscara es, en realidad, creada a partir de **La Función de Punto de Diseminación** del filtro. La Función de punto de diseminación es la imagen que resulta de aplicar el filtro a una fuente de luz en punto ideal.

Usando diferentes máscaras podemos proveer de un gran rango de filtros, incluyendo en estos los que ayudan a reducir el ruido, a suavizar la imagen o asegurar el rasgo de una cierta clase de punto culminante antes de ver algunos ejemplos específicos, necesitamos entender con mas detalle que es, lo que entendemos por filtro, cuando este termino es aplicado a imágenes. Para hacer esto comenzaremos por considerar la situación comúnmente mas encontrada de filtrado de señales, como es el sonido.

La noción de frecuencia es bien entendida cuando la aplicamos a sonido. El sonido esta cargado de vibraciones en el aire y la frecuencia es una cuenta de el número de vibraciones que ocurren en un segundo. La frecuencia esta directamente relacionada, con la forma en la cual, nosotros percibimos el sonido. La frecuencia alta de una nota musical por ejemplo. El equipo para reproducir música es a menudo probado con filtros para que el sonido pueda ser ajustado para que le vaya bien con las preferencias de oyente. Usualmente hay un "BASS" (contrabajo) controlando los efectos de las notas de bajas frecuencias y un "TREBLE" (clave de sol) para controlar las altas frecuencias. Estos pueden ser manipulados de tal manera que el bass incrementa la proporción de componentes de bajas frecuencias en el conjunto de datos de salida, de la misma manera el treble incrementa el contenido de altas frecuencias.

Visto que la frecuencia del sonido en una medida de cambio ocurrente en el tiempo, frecuencia, puede ser aplicado a imágenes, es una medida de cambios ocurrentes en el espacio.

Consecuentemente esto es conocido como frecuencia espacial. Frecuencia espacial es una medida de como rápidamente ocurren variaciones en la brillantez y el

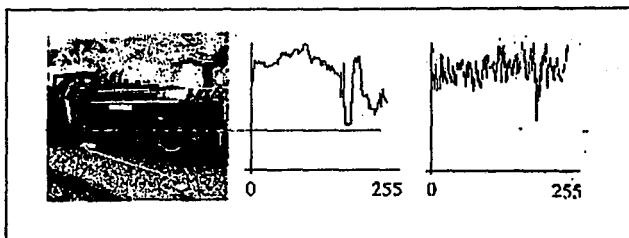


Figura 2.1.7. Altas y bajas frecuencias. (a) Imagen original marcada con dos secciones cruzadas. La parte superior es justamente la región liza, mientras que la parte inferior es la cima que exhibe considerable textura. (b) Sección cruzada a partir de la sección liza. (c) Sección cruzada a partir de la región que contiene frecuencias espaciales altas.

color dentro de la imagen cuando la recorremos. Las áreas de una imagen que tienen una gran cantidad de detalles finos visibles o texturas, tienen un alto contenido de frecuencias espaciales. La brillantez y el color varían rápidamente cuando atravesamos el área de la imagen. Si trazamos una sección cruzada a través del área, el trazo muestra rápidamente variabilidad en los niveles de brillantez y color. Un área que contiene cambios planos de brillantez o color, tiene pocas componentes de altas frecuencias espaciales y esta sección de referencia cruzada observa los alizamientos correspondientes. La figura 2.1.7. muestra la sección cruzada a través de la parte liza y con textura de la imagen.

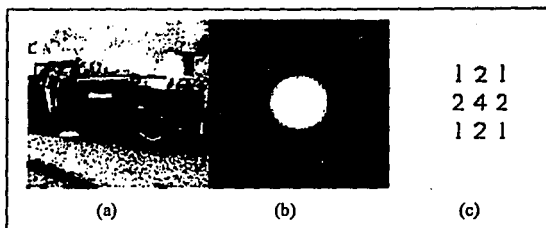


Figura 2.1.8. Alisamiento Gaussiano. (a) Imagen alizada. (b) Función de punto de diseminación. (c) Máscara de convolución.

2.1.2.3 FILTROS DE ALIZAMIENTO

Los filtros de alisamiento reducen el ruido en una imagen reduciendo la proporción de componentes de altas frecuencias. Por lo que son conocidos también como filtros "pasabajas". Estos filtros seleccionan las bajas frecuencias dejándolas pasar sin alteraciones, pero atenuando las altas frecuencias. En estos filtros los valores de cada pixel vecino escogido, fue usado sin cambiarlo, dan también, más peso a algunos pixeles vecinos que a otros. La figura 2.1.8 muestra los efectos de la aplicación de un filtro de alisamiento a la imagen del tren. Hay un notable aspecto borroso particular que se presentada en la imagen cuando usamos este filtro. La figura 2.1.8(b) muestra la función de punto de diseminación de el filtro y la figura 2.1.8(c) muestra la máscara de convolución usada para el alisamiento de la imagen.

2.1.2.4 FILTROS PARA AFINAMIENTO DE IMAGENES

Estos filtros ayudan a reducir el aspecto borroso de una imagen aumentando la proporción de componentes de altas frecuencias. Por lo que estos son también conocidos como **filtros pasaaltas**. Estos permiten el paso de altas frecuencias sin cambiarlas pero atenuando las bajas frecuencias. Actualmente, el término es todavía aplicado incluso cuando el filtro estimula los componentes de altas frecuencias de la imagen. Los filtros pasaaltas de basan en operaciones como derivadas. La amplitud de la derivada de una señal crece, mientras, que la frecuencia de la señal aumenta incluso, si la misma amplitud permanece constante. Desafortunadamente a causa de que la derivada registra solamente cambios en la imagen, las áreas en que no hay cambios en la entrada de datos de la imagen, se vuelven cero en el resultado. Esta es la mejor forma para determinar la discontinuidad de brillantez como veremos enseguida. Para afinamiento de imágenes, sin embargo, la predisposición respecto a las altas frecuencias es demasiado cara cuando usamos solo derivadas. Para reparar el balance, los filtros de afinamiento pueden ser construidos combinando un filtro derivativo con una imagen original.

La figura 2.1.9 muestra un ejemplo de filtros pasa altas aplicado a una imagen. La figura 2.1.9(a) muestra la imagen filtrada. Parece mas fresca, los objetos mejor definidos en los bordes y con mejor detalle. La digitalización de ruido, sin embargo, inherente en la imagen original ha sido modificado haciéndolo mas obvio. Este ruido puede verse como líneas horizontales a lo largo de la imagen. Esto es prácticamente visible en la misma locomotora. La figura 2.1.9(b) muestra la máscara del filtro usado para afinar la imagen.

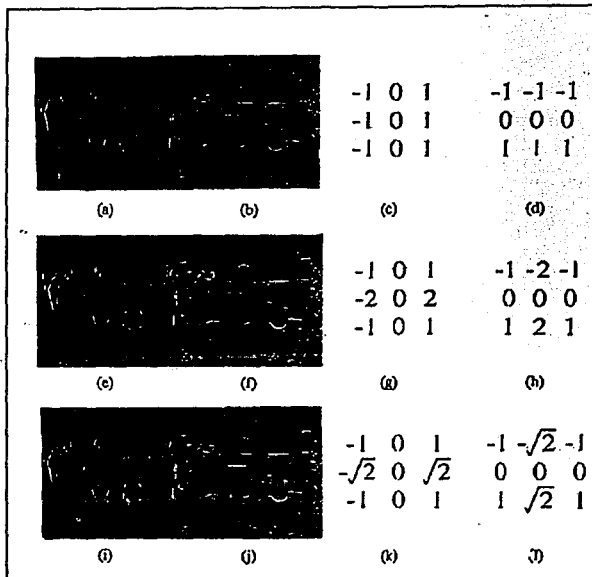


Figura 2.1.9 Detectores de bordes direccionales. (a) Filtro Prewitt componente x, (b) Filtro Prewitt componente y, (c) Máscara de filtro Prewitt en dirección x, (d) Máscara de filtro Prewitt en dirección y, (e) Filtro Sobel componente x, (f) Filtro Sobel componente y, (g) Máscara de filtro Sobel en dirección x, (h) Máscara de filtro Sobel en dirección y, (i) Filtro Isotrópico componente x, (j) Filtro Isotrópico componente y (k) Máscara de filtro Isotrópico en dirección x (l) Máscara de filtro Isotrópico en dirección y.

La Sumatoria inversa Laplaciana aplicada a una imagen original es la que se muestra en la figura 2.1.9(c) donde podemos ver los efectos que produce este filtro y la máscara es mostrada en la figura 2.1.9(d). La única diferencia entre la Laplaciana inversa y el filtro pasaaltas es el peso central, el 8 por el 9.

2.1.2.5 DETECTORES DIRECCIONALES DE BORDES SIMPLES

Algunos de los filtros que hemos visto no contienen información direccional, en contraste con el gradiente que es inherentemente direccional.

Entonces veremos filtros de detección de bordes que están usando gradiente, consecuentemente están ellos mismos direccionados. El direccionamiento se expresa por tener un par de máscaras de convolución por cada filtro. Una retroalimentación de componentes de los gradientes a lo largo de la dirección x y otro tanto de estos componentes en la dirección y . Estos requieren dos operaciones de convolución separados.

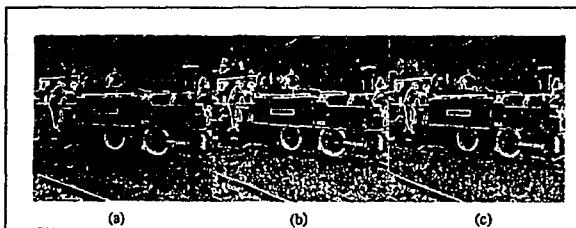


Figura 2.1.10 Magnitudes de los Gradientes desde los detectores de bordes direccionales.
(a) Filtro Prewitt, (b) Filtro Sobel, (c) Filtro Isotrópico.

La figura 2.1.9. muestra 3 detectores de bordes que se usaron en WPDI todos basados en gradientes. Cada filtro es mostrado, aplicado a la imagen del tren. Los componentes de x y y de un gradiente son ilustrados juntos con la máscara del filtro usado. Nótese como los bordes verticales tienden a ser revelados por la máscara en dirección x y los horizontales por la máscara en dirección y . El la figura 2.1.10 muestra la magnitud del gradiente para cada uno de los filtros.

2.2 PROCESAMIENTO DE IMAGENES EN EL DOMINIO DE LA FRECUENCIA

2.2.1 TRANSFORMACION DE IMAGENES.

La teoría de la transformación ha jugado un papel importante en el procesamiento digital de imágenes por varios años, y continua siendo un tópico interesante tanto en teoría como en aplicaciones en este campo. La transformación de imágenes se ha desarrollado importantemente debido a los adelantos tanto en software con la tecnología de circuitos integrados de estado sólido, como en hardware con los algoritmos de convolución rápida.

Los sistemas digitales van adquiriendo mayor flexibilidad en el procesamiento de algoritmos, procesos iterativos, procesos de evaluación de requerimientos y en la toma de decisiones.

La transformada de dos dimensiones es usada para el mejoramiento, restauración, codificación y descripción de imágenes.

2.2.1.1 TRANSFORMACION UNITARIA DE DOS DIMENSIONES.

La transformación unitaria de dos dimensiones ha encontrado tres aplicaciones principales en el procesamiento de imágenes. Primeramente se ha utilizado para extraer las características principales de las imágenes.

Por ejemplo, en la transformada de Fourier el valor promedio es proporcional al promedio del brillo de la imagen y en términos de alta frecuencia da una indicación de la

amplitud y una orientación del contorno dentro de una imagen. Otra aplicación es la codificación de la imagen por la transformada en la cual una reducción de ancho de banda es realizada por descartación o una cuantización en conjunto de baja magnitud de coeficientes transformados. La tercera aplicación es la reducción dimensional en computación.

2.2.2 TRANSFORMADA DE FOURIER.

La transformada discreta de Fourier de dos dimensiones de un arreglo de imagen se define por una serie de la siguiente forma:

$$F(u, v) = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} F(j, k) \exp \left[\frac{-2\pi i}{N} (uj + vk) \right] \quad (2.2.1)$$

donde $i = \sqrt{-1}$. La transformada inversa esta dada por:

$$F(j, k) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) \exp \left[\frac{2\pi i}{N} (uj + vk) \right] \quad (2.2.2)$$

Los índices (u, v) son llamados, frecuencias espaciales de la transformación, en analogía con la transformada continua de Fourier. La transformada de dos dimensiones puede ser calculada como una secuencia de renglones y columnas de una transformada unidimensional. Las funciones base de la transformada son de exponenciales complejos que pueden ser descompuestos en componentes seno y coseno. Esto es,

$$A(j, k; u, v) = \exp\left[\frac{-2\pi i}{N}(vj + uk)\right] = \cos\left[\frac{2\pi i}{N}(uj + vk)\right] - i \operatorname{sen}\left[\frac{2\pi i}{N}(uj + vk)\right] \quad (2.2.3)$$

$$B(j, k; u, v) = \exp\left[\frac{-2\pi i}{N}(vj + uk)\right] = \cos\left[\frac{2\pi i}{N}(uj + vk)\right] + i \operatorname{sen}\left[\frac{2\pi i}{N}(uj + vk)\right] \quad (2.2.4)$$

La transformada de Fourier de dos dimensiones de una imagen es esencialmente una representación de la serie de Fourier de un área de dos dimensiones.

Si el arreglo de la imagen representa una área de luminancia, $F(j, k)$ sería una función positiva real. Sin embargo, la transformada de Fourier es, en forma general, compleja. La figura 2.2.1 contiene fotografías de una imagen y varias versiones de la transformada de Fourier.

2.2.2.1. TRANSFORMADA RAPIDA DE FOURIER (FFT)

Considerando un arreglo bidimensional de valores de imagen $B_{m,n}$ obtenidas por muestreo, donde $m = 0, 1, 2, \dots, M-1$ y $n = 0, 1, 2, 3, \dots, N-1$. Cada $B_{m,n}$ es proporcional al brillo de la imagen muestreada, su posición está dada por m y n .

La transformada discreta de Fourier de la función discreta $B_{m,n}$ está dada por:

$$F_{k,i} = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} B_{m,n} \exp\left[-j2\pi\left[\frac{km}{M} + \frac{in}{N}\right]\right] \quad (2.2.5)$$

donde $k = 0, 1, \dots, M-1$ e $i = 0, 1, \dots, N-1$.

La inversa es:

$$\beta_{m,n} = \frac{1}{\sqrt{MN}} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F_{k,l} \exp \left[j2\pi \left[\frac{km}{M} + \frac{ln}{N} \right] \right] \quad (2.2.6)$$

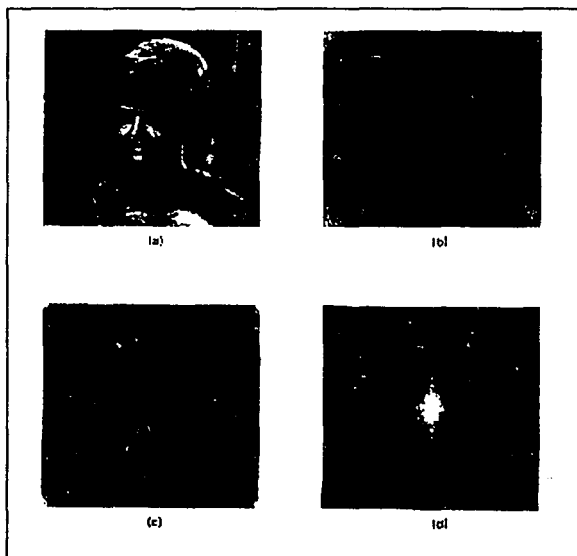


Figura 2.2.1 Transformada de Fourier de una imagen. (a) Original, (b) Magnitud logarítmica no ordenada, (c) Magnitud comprimida no ordenada, (d) Magnitud comprimida ordenada.

A fin de representar una matriz de pixeles, con una subposición de tamaño Lmn , de exponenciales complejos de la transformada discreta de Fourier, se escogen L exponenciales con lo que se alarga la magnitud de los coeficientes, minimizándose el mínimo error cuadrático.

El método usado es dividir la imagen dentro de sub-bloques rectangulares. Cada sub-bloque se expande en una transformada discreta de Fourier y los coeficientes son subejecutados para un umbral adaptivo.

Una cantidad L es determinada para cada sub-bloque tal que sea proporcional a la parte entera de la desviación estándar o de los sub-bloques de los pixeles;

$$\sigma = \left[\frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=1}^{N-1} (\beta_{m,n} - \alpha)^2 \right]^{1/2} \quad (2.2.7)$$

Las frecuencias y amplitudes complejas de los L coeficientes de Fourier de mayor valor son transmitidos; la cuantización lineal adaptiva de los coeficientes de Fourier son usados como precedente para su transmisión.

Las fases de los coeficientes complejos de Fourier son cuantizados y se determinan por σ . También se usan 2, 3, 4 o 5 bits para cuantizar todas las fases y amplitudes muestreadas dentro de cada sub-bloque

Son varias las cosas para considerar el tamaño del sub-bloque entre ellas el valor de σ que se incrementa con el área del sub-bloque, porque la correlación entre los puntos de imagen generalmente decrece conforme la distancia entre los pixeles aumenta. En un sub-bloque de mayor tamaño es muy probable que las áreas de pequeña variación en el brillo

TECNICAS DE PROCESAMIENTO DIGITAL DE IMAGENES

sean incluidas con áreas de gran variación. En áreas de poca variación es probable que den pérdidas, por la presencia del mismo sub-bloque de las áreas de gran variación.

2.2.3 PROCESAMIENTO DE IMAGENES POR TRANSFORMADA.

Entre las transformadas utilizadas para el procesamiento digital de imágenes se encuentra principalmente la transformada de Fourier y sus derivaciones, que tienen un amplio rango de aplicación en problemas de procesamiento de imágenes; de las cuales ya se habló en el punto anterior. Con la utilización de otras formas de transformación se han obtenido mejoras al ser aplicadas en el procesamiento de imágenes.

Con la transformada Haar, Slant, Hadamard, Karhunen-Loeve, Coseno Seno se han obtenido adelantos en su aplicación en casos específicos, porque se han creado algoritmos de rápidos resultados, reducido el número de cálculos y consecuentemente los métodos de implementación.

2.2.3.1 TRANSFORMADA HAAR.

Haar propuso una transformada simple compuesta de unos, menos unos y ceros. La matriz Haar de 4×4 está dada por:

$$H_4 = \frac{1}{\sqrt{4}} \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & \sqrt{2} \end{vmatrix} \quad (2.2.8)$$

La matriz de 8 x 8 queda como sigue:

$$H_8 = \frac{1}{\sqrt{8}} \begin{vmatrix} 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \sqrt{2} & \sqrt{2} & \sqrt{2} & \sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & \sqrt{2} & \sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{vmatrix} \quad (2.2.9)$$

Esta transformada, ha sido implementada para procesos de muestreo en el cual los renglones de la matriz transformada muestreada la secuencia de datos de entrada con una fina resolución incrementándose en potencia de 2. Sin embargo, la calidad de la imagen resultante es mala respecto a otras transformadas por lo que no es muy usada. La figura 2.2.2 muestra la Transformada Haar de una imagen.

2.2.3.2 TRANSFORMADA HADAMARD.

La transformada Hadamard se basa en la matriz Hadamard la cual es un arreglo de mas y menos unos, cuyos renglones y columnas son ortogonales; la matriz Hadamard satisface la relación.

$$HH^T = I$$

La matriz Hadamard de 2 x 2 está dada por:

$$H_2 = \frac{1}{\sqrt{2}} \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix} \quad (2.2.9)$$

Esta matriz no se restringe a 2 y es extensible a cualquier valor de $N = 2^n$ donde n es un entero, en este caso si H_n es una matriz de tamaño N la matriz es:

$$H_{2n} = \frac{1}{\sqrt{2}} \begin{vmatrix} H_n & H_n \\ H_n & -H_n \end{vmatrix} \quad (2.2.10)$$

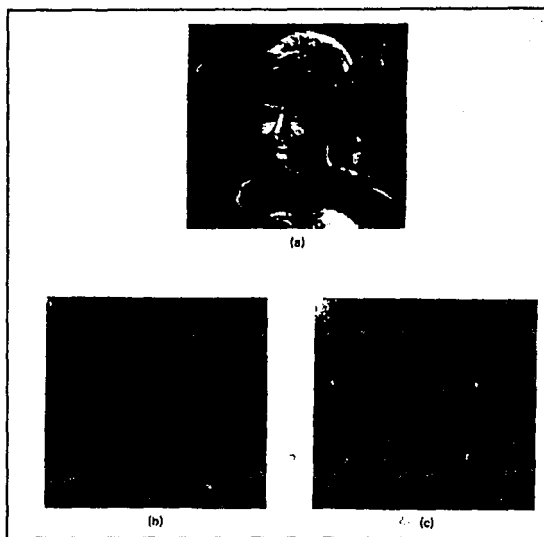


Figura 2.2.2 Transformada de una imagen, (a) Original, (b) magnitud logarítmica, (c) magnitud comprimida.

Esta es una matriz de orden $2n$; H. Harmut en su artículo "transmission of information by orthogonal functions" sugirió que es posible construir una matriz Hadamard de orden $N = 2n$ en la cual el número de signos por renglón se incrementará de 0 hasta $N-1$; los renglones pueden ser considerados muestras de ondas rectangulares con subperíodo de $1/N$ unidades; estas son llamadas funciones Walsh. Es por esto que ésta transformada es también llamada Walsh o Walsh-Hadamard.

Para matrices simétricas de orden $N = 2n$ una transformada Hadamard de dos dimensiones se puede escribir en series de la siguiente forma:

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} F(x, y) (-1)^{p(x, y, u, v)} \quad (2.2.11)$$

donde

$$p(x, y, u, v) = \sum_{i=0}^{N-1} (x_i y_i + u_i v_i) \quad (2.2.12)$$

Los términos u_i , x_i , v_i y y_i son representaciones binarias de u , x , v y y , respectivamente. La figura 2.2.3 presenta la transformada Hadamard de una imagen.

2.2.3.3 TRANSFORMADA KARHUNEN-LOEVE.

La técnica para la transformación de señales continuas, fué originalmente desarrollada por H. Karhunen y M. Loeve; a H. Hotelling se le acreditan los procesos de conversión de transformación de las señales discretas. Sin embargo, a Karhunen-Loeve se les reconoce ambas transformaciones.

La transformada Karhunen-Loeve es de forma general:

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) A(x, y; u, v) \quad (2.2.13)$$

para la cual $A(x, y; u, v)$ satisface la ecuación:

$$\lambda(u, v)A(x, y; u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} K_f(x, y; x', y')A(x, y; u, v) \quad (2.2.15)$$

donde $K_f(x, y; x', y')$ puede separarse en renglones y columnas resultando:

$$K_f(x, y; x', y') = K_c(x, x')K_m(y, y') \quad (2.2.16)$$

de (2.2.16) resulta:

$$A(x, y; u, v) = A_c(x, u)A_k(y, v) \quad (2.2.17)$$

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)A_c(x, u)A_k(y, v) \quad (2.2.17)$$

que es la transformada Karhunen-Loeve en forma de serie.

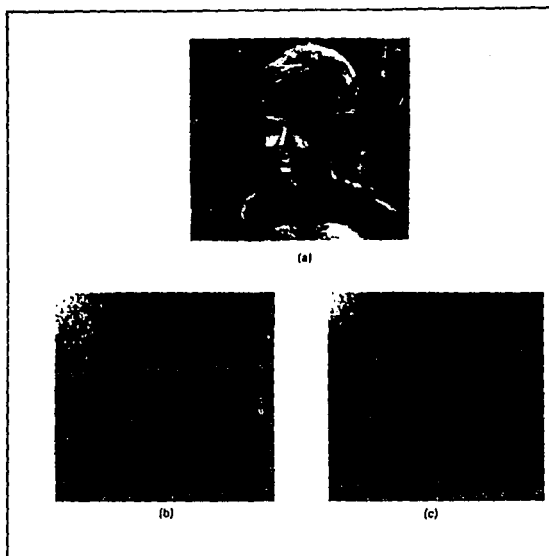


Figura 2.2.3. Transformada Hadamard de una imagen, (a)Original, (b)Magnitud logarítmica, (c) Magnitud comprimida

2.2.3.4 TRANSFORMADA SLANT.

La transformada Slant fue diseñada específicamente para codificación de imagen; utiliza un algoritmo computacional rápido, lo que provoca que se obtenga una mayor reproducción de calidad de imagen. Por sus características de funcionamiento es un poco menos óptimo que la transformada Karhunen-Loeve, pero más compleja que la transformada

rápida de Fourier. Para un vector de longitud $N = 2$ la transformada Slant es idéntica a la transformada Hadamard de orden 2, es decir:

$$S_2 = \frac{1}{\sqrt{2}} \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix} \quad (2.2.18)$$

La transformada Slant de orden 4 es obtenida por la operación:

$$S_4 = \frac{1}{2^{1/2}} \begin{vmatrix} 1 & 0 & 1 & 0 \\ a_4 & b_4 & -a_4 & -b_4 \\ 0 & 1 & 0 & 1 \\ -b_4 & a_4 & b_4 & a_4 \end{vmatrix} \begin{vmatrix} S_2 & 0 \\ 0 & S_2 \end{vmatrix} \quad (2.2.19)$$

o bien,

$$S_4 = \frac{1}{\sqrt{2}} \begin{vmatrix} 1 & 1 & 1 & 1 \\ a_4 + b_4 & a_4 - b_4 & -a_4 + b_4 & -a_4 - b_4 \\ 1 & 1 & 1 & 1 \\ a_4 - b_4 & -a_4 - b_4 & a_4 + b_4 & -a_4 + b_4 \end{vmatrix} \quad (2.2.20)$$

donde a_4 y b_4 son constantes de escala real cuyos valores se determinan de acuerdo con las condiciones que hacen a S_4 ortogonal, se ha encontrado que este valor es $a_4 = 2b_4$ y la condición de ortogonalidad dice que $S_4 S_4^t = I$ lleva a encontrar $b_4 = 1/\sqrt{5}$.

TECNICAS DE PROCESAMIENTO DIGITAL DE IMAGENES

La matriz general para N términos está dada por:

$$S_N = \frac{1}{\sqrt{2}} \left(\begin{array}{ccc|ccc|c} 1 & 0 & & 1 & 0 & & \\ a_N & b_N & 0 & -a_N & b_N & 0 & \\ \hline 0 & I_{(n/2)-2} & & 0 & I_{(n-2)-2} & & \\ \hline 0 & 1 & & 0 & -1 & & \\ -b_N & a_N & 0 & b_N & a_N & 0 & \\ \hline 0 & I_{(N/2)-2} & & 0 & -I_{(n-2)-2} & & \end{array} \right) \begin{array}{c} S_{N/2} & 0 \\ 0 & S_{N/2} \end{array} \quad (2.2.21)$$

donde $I_{(n/2)-2}$ es la matriz identidad de dimensión $(N/2)-2$. Las constantes a_N y b_N pueden ser encontradas por las siguientes ecuaciones:

$$\begin{aligned} a_N &= 1 \\ b_N &= [1 + 4(a_{N/2})^2]^{-1/2} \\ a_{N/2} &= 2b_N a_{N/4} \\ a_{2N} &= [(3N^2) / (4N^2 - 1)]^{1/2} \\ b_{2N} &= [(N^2 - 1) / (4N^2 - 1)]^{1/2} \end{aligned}$$

La transformada Slant requiere de un total de $N \log_2 N + (N/2)-2$ adiciones y sustracciones más $2N-4$ multiplicaciones para un vector con una dimensión de datos N . Fotografías de la transformada Slant de una imagen se muestran en la figura 2.2.4.

2.2.3.5 TRANSFORMADA COSENO.

Es conocido que en las series de Fourier para cualquier función real continua y simétrica contienen únicamente coeficientes reales que corresponden a los términos coseno de dichas series.

Esto puede extenderse a la transformada de Fourier bajo su propia interpretación. Existen dos caminos en los cuales una imagen puede hacerse simétrica, como se muestra

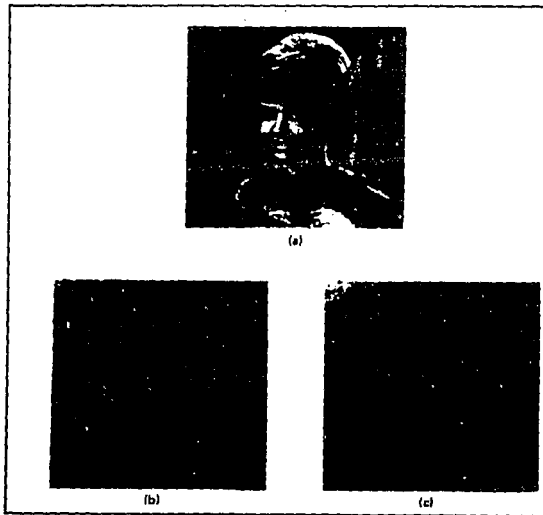


Figura 2.2.4 Transformada Slant de una imagen, (a) Original, (b) Magnitud logarítmica, (c) Magnitud comprimida.

en la figura 2.2.5. Por la primer técnica las imágenes son dobladas hacia un contorno y en el segundo método las imágenes son dobladas y superpuestas por un pixel.

De este modo, para una imagen de $N \times N$ pixeles, el primer método es llamado Transformada Coseno de Simetría Semejante, resultando un arreglo de $2N \times 2N$ pixeles;

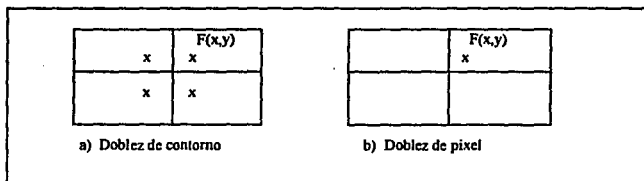


Figura 2.2.5 Transformada coseno de simetría semejante.

el segundo método llamado de Transformada Coseno de Simetría Impar produce un arreglo de $2N - 1 \times 2N - 1$ pixeles.

2.2.3.5.1 TRANSFORMADA COSENO DE SIMETRÍA SEMEJANTE.

Si se tiene un arreglo de imagen simétrica será formada por reflexión del arreglo de imagen hacia los contornos de acuerdo a la relación:

$$F = (x,y) = \begin{cases} F(x,y) & x \geq 0, y \geq 0 \\ F(-1-x,y) & x < 0, y > 0 \\ F(x,-1-y) & x \geq 0, y < 0 \\ F(-1-j,-1-k) & x < 0, y < 0 \end{cases} \quad (2.2.22)$$

Para esta construcción $F_s(x,y)$ es simétrica hacia el punto $x = -1/2$, $y = -1/2$ y tomando la transformada de Fourier hacia el punto de simetría resulta:

$$\mathcal{F}_s(u,v) = -\frac{1}{2N} \sum_{x=0}^{N-1} \sum_{y=-N}^{N-1} F_s(x,y) \exp\left\{-\frac{2\pi}{2N} \left[u \left[x + \frac{1}{2} \right] + v \left[y + \frac{1}{2} \right] \right]\right\} \quad (2.2.23)$$

para $u, v = -N, \dots, -1, 0, 1, \dots, N-1$; debido a que $F_s(x,y)$ es real y simétrica, la ecuación (5.31) se reduce a:

$$\mathcal{F}_s(u,v) = \frac{2}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} F(x,y) \cos\left[-\frac{\pi}{N} u \left[x + \frac{2}{2} \right]\right] \cos\left[-\frac{\pi}{N} v \left[y + \frac{2}{2} \right]\right] \quad (2.2.24)$$

De manera alternativa, los términos de las ecuaciones (2.2.23) y (2.2.24) pueden ser obtenidas por una transformada de Fourier de $F(x,y)$ sobre $2N$ puntos obteniéndose:

$$\mathcal{F}_s(u,v) = \frac{2}{N} R_s \left\{ \exp\left[\frac{i u \pi}{2N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} F(x,y) \exp\left[\frac{-2i \pi}{2N} (ux + vy) \right] \right] \right\} \quad (2.2.25)$$

2.2.3.5.2 TRANSFORMADA COSENO DE SIMETRÍA IMPAR.

Para la transformada coseno de simetría impar, el arreglo de imagen queda definido de la siguiente forma:

$$F_s(x, y) = \begin{cases} F(x, y) & x, y \geq 0 \\ F(-x, y) & x < 0, y \geq 0 \\ F(x, -y) & x \geq 0, y < 0 \\ F(-x, -y) & x, y < 0 \end{cases} \quad (2.2.26)$$

Tomando en cuenta la transformada de Fourier se genera el arreglo:

$$\mathcal{F}_s(u, v) = -\frac{1}{2N-1} \sum_{x=-N+1}^{N-1} \sum_{y=-N+1}^{N-1} F_s(x, y) \exp\left[\frac{-2i\pi}{2N-1}(xu, yv)\right] \quad (2.2.27)$$

para $u, v = N+1, \dots, -1, 0, 1, \dots, N-1$.

2.2.3.6. TRANSFORMADA SENO.

La transformada seno, fue introducida como un algoritmo rápido para sustituir a la transformada Karhunen-Loeve.

Esta transformada es definida como:

$$\mathcal{F}_s(u, v) = -\frac{1}{N+1} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} F_s(x, y) \operatorname{sen}\left[\frac{(x+1)(u+1)}{N+1} \pi\right] \operatorname{sen}\left[\frac{(y+1)(v+1)}{N+1} \pi\right] \quad (2.2.28)$$

su inversa es de forma similar. La transformada seno puede también ser calculada con un algoritmo de la transformada de Fourier. La figura 2.2.6 muestra la fotografía de la transformada seno de una imagen.

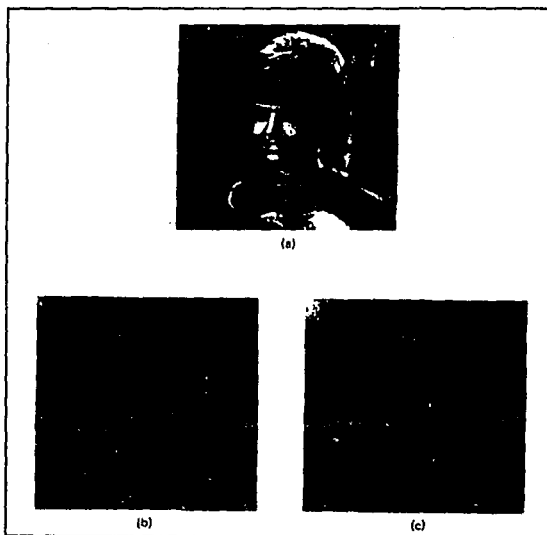


Figura 2.2.6 Transformada seno de una imagen, (a) Original, (b) magnitud logaritmica, (c) Magnitud comprimida.

CAPITULO

3

3.1. ANALISIS ORIENTADO A OBJETOS

3.1.1 CONCEPTOS SOBRE LOS OBJETOS.

Para poder entender el punto de vista de “Orientado a Objetos” es necesario antes que nada entender lo que es un objeto, para ello consideremos un ejemplo de un objeto del mundo real, tomemos para ello el caso de una silla.

Silla es un miembro (también se utiliza el término “instancia”) de una clase de objetos mucho mayor que denominamos *Mueble*. Se puede asociar un conjunto de atributos genéricos a cada objeto de la clase *Mueble*. Por ejemplo, todo mueble tiene precio, dimensiones, peso, situación y color, entre muchos atributos posibles. Estos atributos se aplican ya sea que estemos hablando de una mesa, de una silla, un sofá o de cualquier mueble. Dado que *Silla* es un miembro de la clase *Mueble*, hereda todos los atributos definidos para la clase.

En la figura 3.1.1 se ilustra esquemáticamente este concepto.

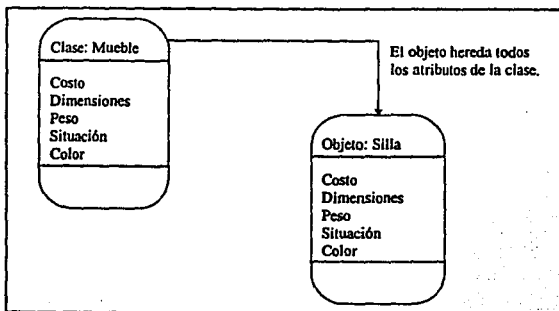


Figura 3.1.1 Herencia de Clase a Objeto

Una vez que se ha definido la clase, se pueden reutilizar los atributos creando nuevas instancias de la clase. Por ejemplo, supongamos que vamos a definir un nuevo objeto denominado *Mella* (un cruce entre mesa y silla) que sea miembro de la clase *Mueble*. *Mella* hereda todos los atributos de *Mueble*.

Hasta ahora se ha intentado definir de una forma amena una clase describiendo sus atributos, pero sin embargo, aún nos falta algo. Cada objeto de la clase *Mueble* puede ser manipulado de muchas formas. Puede ser vendido y comprado, modificado físicamente (por ejemplo, se le puede pintar de otro color o quitársele una pata), o ser movido de un sitio a otro. Cada una de esas operaciones (otros términos que se emplean son servicios o métodos) modifican uno o más atributos del objetos. Por ejemplo, si el atributo *situación* es realmente un elemento de datos compuestos que está definido como:

situación = edificio + piso + habitación.

Entonces, una operación denominada *mover* modificará uno o más de esos elementos de datos (edificio, piso o habitación) que componen el atributo *situación*. Para poder hacerlo, *mover* ha de ser "consciente" de la existencia de esos elementos de datos. Se puede utilizar la operación *mover* para una *Silla* o una *Mesa*, mientras que sean instancias de la clase *mueble*. Todas las operaciones válidas (por ejemplo *vender*, *comprar*, *pesar*) para la clase *Mueble* están "conectadas" a la definición del objeto como se muestra en la figura 3.1.2.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

Ya que se han introducido algunos conceptos básicos, resultará más adecuada una definición más formal de la "orientación a objetos". Coad y Yourdon¹ definen el término de la siguiente forma:

orientación a objetos = objetos + clasificación + herencia + comunicación.

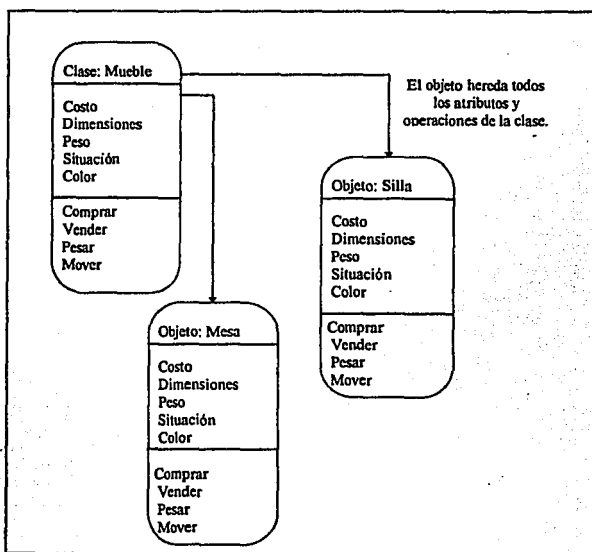


Figura 3.1.2 Herencia de operaciones de clase a objeto.

¹ ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS.
James Martin y James J. Odell

3.1.2. IDENTIFICACION DE OBJETOS.

Si miramos alrededor en una habitación, veremos un conjunto de objetos físicos que pueden ser identificados fácilmente, así como clasificados y definidos (en términos de atributos y operaciones). Pero cuando observamos el “espacio” del problema de una aplicación de software, resulta más complicado encontrar los objetos.

Podemos empezar a identificar los objetos examinando la descripción del problema o llevando a cabo un “análisis gramatical” de la narrativa del procesamiento del sistema a construir. Determinamos los objetos subrayando cada nombre o cláusula nominal y añadiéndolo en una tabla, se deben anotar los sinónimos. Si un objeto es necesario para implementar una solución, entonces es parte del espacio de la solución; en caso contrario, cuando el objeto solo es necesario para describir la solución, entonces es parte del espacio del problema.

Los objetos se manifiestan de una de las siguientes formas las cuales se representan en la figura 3.1.3.

- **Entidades externas** (p.ej.: otros sistemas, dispositivos, gente) que producen o consumen información a ser utilizada en el sistema.
- **Cosas** (p.ej.: informes, visualizaciones, cartas, señales) que son parte del dominio de información del problema.

ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

- **Ocurrencias o Sucesos** (p.ej.: una transferencia de una propiedad o la terminación de una serie de movimientos de un robot) que ocurren en el contexto de operación del sistema.
- **Papeles** que juegan las personas que interactúan con el sistema (p.ej.: gestor, investigador, ingeniero, vendedor).
- **Unidades Organizativas** (p.ej.: división, grupo, equipo) que son relevantes para la aplicación.
- **Lugares** (p.ej.: sala de facturación o muelle de descarga) que establecen el contexto del problema del funcionamiento general del sistema.
- **Estructuras** (p.ej.: sensores, vehículos de cuatro ruedas o computadoras) que definen clases de objetos o, en casos extremos, clases de objetos relacionadas.

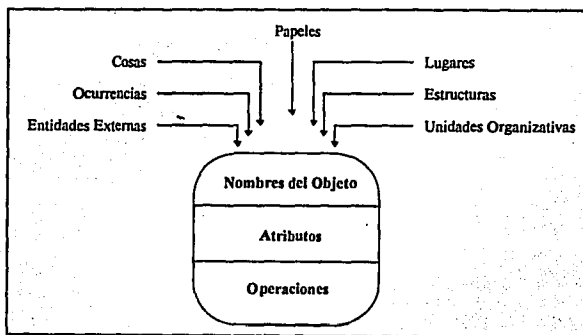


Fig. 3.1.3 Posibles Manifestaciones de los Objetos.

ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

Coad y Yourdon sugieren seis características selectivas que debe usar el analista para considerar la inclusión o no, de cada objeto potencial en el modelo de análisis:

- **Información Retenida.** El objeto potencial será útil durante el análisis solo si la información sobre el mismo ha de ser recordada para que el sistema pueda funcionar.
- **Servicios Necesarios.** El objeto potencial debe tener un conjunto de operaciones identificables que puedan cambiar de alguna forma el valor de sus atributos.
- **Múltiples atributos.** Durante el análisis de requisitos, se debe centrar la atención en la información "principal"; un objeto con un sólo atributo puede resultar útil durante el diseño, pero probablemente se represente mejor como atributo de otro objeto durante la fase de análisis.
- **Atributos Comunes.** Se pueden definir conjuntos de atributos para los objetos potenciales y aplicar así esos atributos a todas las ocurrencias del objeto.
- **Operaciones Comunes.** Se pueden definir conjuntos de operaciones para los objetos potenciales y aplicar así esas operaciones a todas las ocurrencias del objeto.
- **Requisitos Esenciales.** En el modelo de requisitos casi siempre se definirán como objetos las entidades externas que aparecen en el espacio del problema y que producen o consumen información esencial para el funcionamiento de cualquier solución que se desarrolle para el sistema.

ANALISIS Y DISEÑO ORIENTADO A OBJETOS

Para que un objeto sea considerado como legítimo para ser incluido en el modelo de requisitos, el objeto potencial debe satisfacer todas (o casi todas) las características anteriores. La decisión de incluir o no un objeto potencial en el modelo de análisis puede resultar algo subjetiva.

3.1.3. ESPECIFICACION DE ATRIBUTOS.

Los atributos describen un objeto que haya sido seleccionado para ser incluido en el modelo de análisis. Esencialmente, son los atributos lo que definen un objeto -son los que aclaran el significado del objeto en el contexto del espacio del problema-.

Para desarrollar un conjunto significativo de atributos para un objeto, el analista debe de estudiar de nuevo la narrativa de procesamiento (o descripción de alcance) para el problema concreto y seleccionar aquello de "pertenencia" de forma razonable al objeto. Además, se debe responder a la siguiente pregunta para cada objeto: ¿Qué elementos de datos (elementales y/o compuestos) definen completamente ese objeto en el contexto del problema actual?

3.1.4. DEFINICION DE LAS OPERACIONES

Una operación cambia un objeto de alguna forma. Más concretamente, cambia valores de uno o más atributos que están contenidos en el objeto, por tanto, una operación debe tener "conocimiento" de la naturaleza de los atributos del objeto y deben ser

ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

implementadas de forma que se les permita manipular las estructuras de datos que hayan sido derivadas de los atributos.

Aunque existen muchos tipos distintos de operaciones generalmente se pueden dividir en tres grandes categorías:

- 1) Operaciones que manipulan los datos de alguna forma (p.ej.: adiciones, eliminaciones, cambios de formato, selecciones).
- 2) Operaciones que realizan algún cálculo.
- 3) Operaciones que monitorizan un objeto frente a la ocurrencia de algún suceso de control.

Para realizar un primer intento de obtener un conjunto de operaciones para los objetos del modelo de análisis, el analista ha de volver a estudiar la narrativa de procesamiento del problema y seleccionar aquellas operaciones que “correspondan” razonablemente a cada objeto. Para hacerlo, se lleva a cabo de nuevo un análisis gramatical y se aíslan los verbos. Algunos de los verbos serán operaciones legítimas y se podrán conectar fácilmente con un objeto específico, además, con el análisis gramatical podemos comprender mejor las operaciones considerando la comunicación que ocurre entre los objetos.

3.1.5. COMUNICACION ENTRE OBJETOS.

La definición de los objetos del contexto del modelo de análisis puede ser suficiente para establecer una base para el diseño, pero se debe añadir algo más (durante el análisis o durante el diseño) para que se pueda construir el sistema se debe establecer un mecanismo para la comunicación entre los objetos. Este mecanismo se denomina mensaje.

En la figura 3.1.4 se muestra esquemáticamente el uso de mensajes, donde cuatro objetos **A**, **B**, **C** y **D**, se comunican entre sí mediante el paso de mensajes. Por ejemplo, si el objeto **B** requiere un procesamiento asociado con la operación *op10* del objeto **D**, entonces le envía a **D** un mensaje que puede tener la siguiente forma:

mensaje: (*destino, operación, argumentos*).

donde:

- *destino* define el objeto (en este caso, **D**) que recibe el mensaje.
- *operación* refiere la operación que va a recibir el mensaje (*op10*).
- *argumentos* proporciona la información requerida para que se pueda llevar a cabo la operación.

Como parte de la ejecución de la operación *op10*, el objeto **D** puede enviar un mensaje al objeto **C** de la forma:

mensaje: (*C, op08, <datos>*)

ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

C busca *op08*, la realiza y devuelve el control a D. La operación *op10* termina y se devuelve el control a B.

Cox² describe el intercambio entre objetos de la siguiente forma:

A un objeto se le pide que realice una de sus operaciones enviándole un mensaje que le indique qué es lo que tiene que hacer. El objeto receptor responde al mensaje escogiendo primero la operación que implementa el nombre del mensaje, ejecutando esa operación y devolviendo el control al emisor.

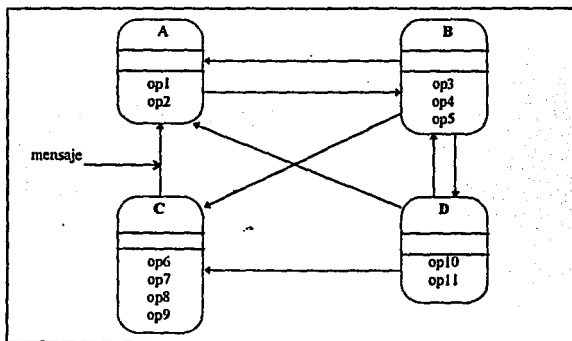


Fig. 3.1.4 Paso de Mensajes.

La mensajería es importante para la implementación de un sistema orientado a objetos, pero no es necesario considerarla detalladamente durante el análisis de requisitos.

² OBJECT-ORIENTED PROGRAMMING
Cox B. J.

ANALISIS Y DISEÑO ORIENTADO A OBJETOS

De hecho, en esta etapa nuestro único cometido es usar el concepto como ayuda para la determinación de operaciones candidatas para un objeto concreto.

La definición de las operaciones es el último paso para tener la especificación de un objeto completa.

3.1.6. RESUMEN DEL ANALISIS

Los métodos orientado a objetos para el análisis de requisitos del software permiten al analista obtener el modelo de un problema representando clases, objetos, atributos y operaciones como componentes principales de modelización. Los puntos de vista orientados a objetos combinan la clasificación de objetos, la herencia de atributos y los mensajes de comunicación dentro del contexto de la notación de modelización.

- Los objetos modelizan casi cualquier aspecto identificable del ámbito del problema: entidades externas, cosas, sucesos, papeles, unidades organizativas, lugares y estructuras pueden ser representadas como objetos. Como punto importante los objetos encapsulan datos y procesos. Las operaciones de procesamiento son parte del objeto y son iniciadas pasando un mensaje al objeto. Una definición de una clase forma la base para la reusabilidad en los niveles de modelización, diseño e implementación. Se pueden instanciar nuevos diseños de una clase.

El objetivo principal del A.O.O. es identificar las clases de las que se instanciarán los objetos.

3.2. DISEÑO ORIENTADO A OBJETOS

El diseño orientado a objetos (D.O.O.) al igual que otras metodologías de diseño orientadas a la información, crea una representación del campo del problema del mundo real y la hace corresponder con el ámbito de la solución, que es el software. A diferencia de otros métodos, el D.O.O. produce un diseño que interconecta objetos de datos (elementos de datos) y operaciones de procesamiento en una forma que modulariza la información y el procesamiento, en lugar de dejar aparte el procesamiento.

La naturaleza única del diseño orientado a objetos queda reflejada en su capacidad de construir sobre tres pilares conceptuales importantes del diseño de software: abstracción, ocultamiento de información y modularidad. Todos los métodos de diseño intentan desarrollar software con esas tres características fundamentales, pero sólo el D.O.O. proporciona un mecanismo que permite al diseñador conseguir las tres sin complejidad.

Wiener y Sincover³ resumen la metodología D.O.O. de la siguiente forma:

Ya no es necesario que el diseñador del sistema haga corresponder el ámbito del problema con estructuras de datos y de control predefinidas que se encuentren en el lenguaje de implementación. En cambio, el diseñador puede crear sus propios tipos abstractos de datos y abstracciones funcionales y hacer corresponder el campo del mundo real con esas abstracciones creadas por el propio programador. Esta correspondencia será la mayoría de las veces mucho más natural, ya que el rango de tipos abstractos de datos que puede

³ SOFTWARE ENGINEERING WITH MODULA-2 AND ADA
Wiener, R. and R. Sincover

ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

inventar el diseñador es virtualmente ilimitado. Más aún, el diseñador de software se desliga de los detalles de representación de los objetos de datos que se usan en el sistema. Así, se pueden cambiar muchas veces esos detalles de representación, sin que ello afecte al sistema de software global.

El A.O.O., el D.O.O. y la P.O.O. comprenden un conjunto de actividades de ingeniería del software para la construcción del sistema orientado a objetos.

Es prematuro decir que el D.O.O. es un método maduro, sin embargo a medida que sigue creciendo la popularidad del enfoque orientado a objetos, se impone la utilización de métodos de diseño para crear sistemas orientados a objetos.

3.2.1. CONCEPTOS DEL DISEÑO ORIENTADO A OBJETOS.

Al igual que otros métodos de diseño, el D.O.O. introduce un nuevo conjunto de términos, notaciones y procedimientos para la derivación del diseño del software. Ahora introduciremos algunos conceptos relevantes para el diseño.

Muchos objetos del mundo físico tienen características razonablemente similares y realizan operaciones razonablemente similares. Si observamos la planta de fabricación de una planta de equipos pesados, veremos máquinas fresadoras, taladradoras y perforadoras. Aunque cada uno de estos objetos es diferente, todos pertenecen a una clase superior denominada "máquinas herramientas". Todos los objetos de la clase máquinas herramienta tienen atributos comunes (p.e.: corte, arranque, parada). Por tanto, clasificando un "torno"

ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

como miembro de la clase máquinas herramienta, ya sabemos algo acerca de los atributos y sus operaciones, incluso sin saber exactamente cuál es su función.

Las realizaciones en software de objetos del mundo real se clasifican de forma muy parecida. Todos los objetos son miembros de una clase mayor y heredan la estructura de datos privada y las operaciones que han sido definidas para esa clase. Dicho de otra forma, una clase es un conjunto de objetos que tienen las mismas características. Así, un objeto es una instancia de una clase mayor.

El uso de clases, subclases y herencia es de crucial importancia en la ingeniería de software moderna. La reutilización de componentes de software se lleva a cabo creando objetos (instancias) que se forman sobre los atributos y las operaciones existentes heredadas de una clase o de una subclase. Sólo es necesario especificar cómo son las diferencias entre el nuevo objeto y la clase, en lugar de definir todas las características del nuevo objeto.

3.2.2. DESCRIPCIONES DE LOS OBJETOS

La descripción de diseño de un objeto (una instancia de una clase o de una subclase) puede tener dos formas distintas:

- 1) Una descripción del protocolo que establece la interfaz del objeto, definiendo cada mensaje que puede recibir el objeto y las operaciones que realiza el objeto cuando recibe el mensaje.

ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

- 2) Una descripción de la implementación, que muestra los detalles de cada operación implicada en la recepción de un mensaje por el objeto. Los detalles de implementación incluyen la información sobre la parte privada del objeto, esto es, los detalles internos de la estructura de datos, y los detalles procedimentales que describen las operaciones.

La descripción del protocolo no es otra cosa que un conjunto de mensajes con sus correspondientes comentarios asociados. Para grandes sistemas con muchos mensajes, a menudo se pueden crear categorías de mensajes.

Una descripción de la implementación de un objeto proporciona los detalles internos ("ocultos") requeridos para su implementación, aunque no necesariamente para su invocación. Es decir, el diseñador del objeto proporciona una descripción de su implementación y, consecuentemente, crea los detalles internos del objeto. Sin embargo, otro diseñador o programador que utilice el objeto u otras instancias del objeto, sólo requiere la descripción del protocolo, no la descripción de la implementación.

Una descripción de la implementación está compuesta por la siguiente información:

- 1) Una especificación del nombre del objeto y de la referencia a una clase.
- 2) Una especificación de la estructura de datos privada, con la indicación de los elementos de datos y sus tipos.

ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

- 3) Una descripción procedimental de cada operación o, alternativamente, referencias a esas descripciones procedimentales. La descripción de la implementación debe tener suficiente información para que se puedan manejar de forma adecuada todos los mensajes descritos en la descripción del protocolo.

Un usuario de un "servicio" proporcionado por un objeto debe familiarizarse con el protocolo de invocación de ese servicio, es decir, con la forma de especificar qué es lo que desea. Al suministrador del servicio (el propio objeto), lo que le concierne es cómo proporcionar el servicio al usuario, es decir, los detalles de implementación. Este concepto, denominado encapsulación, se resume de la siguiente forma:

Un objeto proporciona encapsulación, en cuanto que pone en servicio una estructura de datos y un grupo de procedimientos para accederla, de tal forma que los usuarios de esa facilidad puedan accederla a través de unas interfaces cuidadosamente documentadas, controladas y estandarizadas. Esas estructuras de datos encapsuladas, denominadas objetos, cuentan como datos activos a los que se les puede pedir que hagan cosas, mandándoles mensajes.

3.2.3. UNA NOTACION PARA EL D.O.O.

Booch⁴ propone una notación⁵ que combina cuatro diagramas distintos para la creación del D.O.O. Un Diagrama de Clases refleja las clases y sus relaciones. Un Diagrama de Objetos representa objetos específicos (instancias de una clase) y los mensajes que pasan por ellos. Como parte del diseño físico, se asignan las clases y los objetos a componentes de software específico. El Diagrama de Módulos, a veces denominado Diagrama de Booch, sirve para ilustrar esos componentes de programa. Dado que muchos sistemas orientados a objetos contienen varios programas que se pueden ejecutar en varios procesadores distribuidos, Booch también sugiere una notación, denominada Diagrama de Procesos⁶, que permite al diseñador reflejar cómo se asignan los procesos a procesadores concretos en un gran sistema.

3.2.4. REPRESENTACION DE RELACIONES ENTRE CLASES Y ENTRE OBJETOS.

El Diagrama de Clases ilustrado en la figura 3.2.1 indica la "arquitectura" de clases de un sistema. La herencia se representa con un símbolo de flecha y las conexiones de doble trazo indican que una clase utiliza la información contenida en otra clase. Los símbolos de los extremos de la notación *usa* (doble trazo) indican la cardinalidad. Por ejemplo, en la

⁴OBJECT-ORIENTED ANALYSIS AND DESIGN
With Applications. 2a Ed.
Grady Booch

⁵ Será la notación utilizada para desarrollar nuestro sistema.

⁶ No se usará en nuestra aplicación, ya que no se trata de una aplicación que funcione en forma distribuida.

ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

figura 3.2.1, para cada instancia de la clase A habrá una o más instancias (indicadas por el símbolo +) de la clase D. Además del propio diagrama de clases, se puede definir cada clase con una plantilla que incluya su nombre, sus atributos y sus operaciones, así como la información sobre la herencia, los parámetros y el comportamiento.

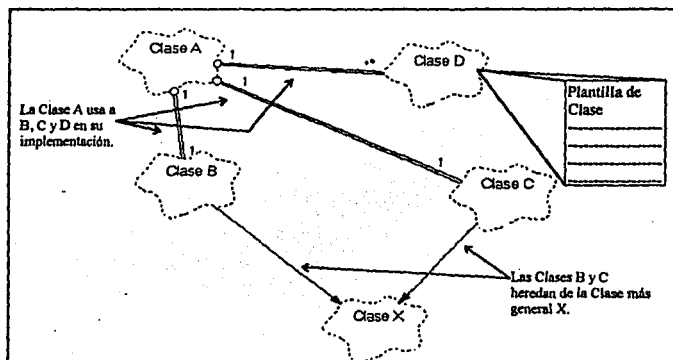


Fig. 3.2.1. Notación básica de un Diagrama de Clases.

Hay que tener en cuenta que la plantilla para la clase no es realmente una parte del diagrama, sino que se crea para complementarlo.

Las clases se definen como parte del diseño de un sistema y existen independientemente del comportamiento en ejecución del sistema. Sin embargo, los objetos se crean y se destruyen dinámicamente a medida que se ejecuta el sistema.

ANALISIS Y DISEÑO ORIENTADO A OBJETOS

El Diagrama de Objetos en la figura 3.2.2 refleja cada instancia de una clase, las operaciones que se le aplican y los mensajes que se le pasan. Las líneas que conectan los objetos indica el tipo de datos que contiene el mensaje. La flecha paralela a la línea indica el tipo de sincronización entre los objetos (p.e.: síncrono, asíncrono, retardado). Para cada objeto se crea una plantilla de objeto que define el objeto y los mensajes que se le pasan.

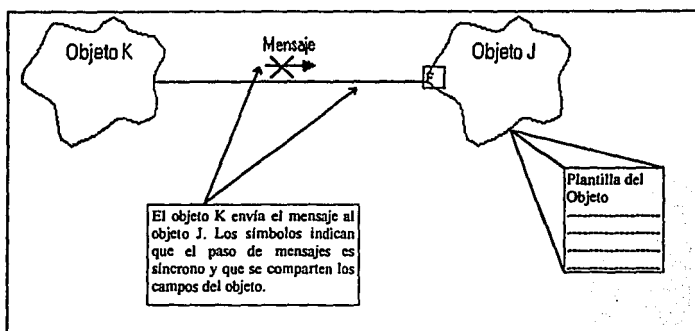


Fig. 3.2.2 Notación básica de un Diagrama de Objetos.

3.2.5. MODULARIZACION DEL DISEÑO

Para definir el concepto de módulo en el D.O.O. tenemos que usar una notación de diseño que asigna clases y objetos a los componentes físicos de programa (módulos) y, por tanto, representa una visión de implementación de la arquitectura del programa. El diagrama de módulos representa un componente de programa (objeto) como una caja dividida en una

ANALISIS Y DISEÑO ORIENTADO A OBJETOS

parte de especificación (visible al exterior) y una parte privada (también denominada parte del cuerpo) que está oculta al exterior. En esta etapa, todavía no se especifican los detalles de implementación de la parte privada o cuerpo del paquete. Los objetos de datos se representan con un óvalo alargado, mientras que las operaciones que actúan sobre ellos se representan mediante rectángulos.

La figura 3.2.3 ilustra el uso de los diagramas de módulos para representar las dependencias entre los componentes de programa. Las flechas de conexión implican dependencia; es decir, el paquete o componente en el origen de la flecha depende del paquete o componente de programa en la punta de la flecha. En la figura 3.2.3, el componente de programa de mayor nivel, A, depende de los objetos y las operaciones contenidas en B, C, y D, necesarios para satisfacer su función.

3.2.6. DISEÑO DETALLADO PARA LA IMPLEMENTACION

El paso de diseño detallado en el D.O.O. es similar en muchos aspectos al diseño detallado de las otras metodologías de diseño. Se describen las interfaces con detalle; se refinan y se especifican las estructuras de datos; se diseñan los algoritmos para cada unidad del programa usando los conceptos fundamentales de diseño, tales como el refinamiento sucesivo y la programación estructurada. La diferencia clave del D.O.O. es que se puede aplicar recursivamente en cualquier momento el proceso descrito en las secciones anteriores. De hecho, la definición recursiva de la estrategia de solución es esencial para conseguir un nivel de diseño y de abstracción de datos del que se pueda derivar el detalle de implementación.

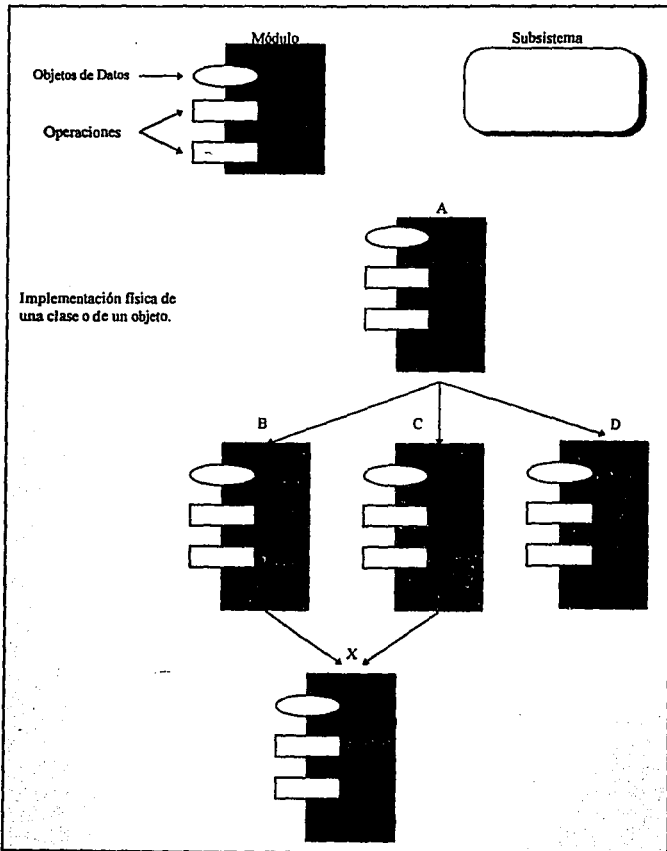


Fig. 3.2.3 Notación básica de un Diagrama de módulo.

3.2.7. RESUMEN DEL DISEÑO

El D.O.O. crea un modelo del mundo real que puede ser realizado en software. Los objetos proporcionan un mecanismo para representar el ámbito de información, mientras que las operaciones describen el procesamiento asociado con el ámbito de información. Los mensajes proporcionan el medio por el que se invocan las operaciones. La característica distintiva de A.O.O./D.O.O. es que los objetos "saben" que operaciones se pueden aplicar sobre ellos. Este conocimiento se consigue combinando abstracciones de datos y de procedimientos en un solo componente de programa (denominado objeto o paquete).

A.O.O./D.O.O. ha evolucionado como resultado de una nueva clase de lenguajes de programación orientados a objetos, tales como Smalltalk, C++, Objective-C, Eiffel, entre otros. Consecuentemente, las representaciones del diseño orientado a objetos son más adecuadas que otras que dependen del lenguaje de programación.

La metodología A.O.O./D.O.O. consiste en tres pasos que requieren que el diseñador establezca el problema, defina una estrategia informal de resolución y formalice la estrategia, identificando objetos y operaciones, especificando interfaces y proporcionando detalles de implementación para las abstracciones de datos y de procedimientos. El papel del D.O.O. es tomar las clases y los objetos básicos definidos en el A.O.O. y refinarlos con detalles adicionales de diseño. Los diseños se representan mediante algunas de las notaciones gráficas existentes y el lenguaje de diseño de programas.

El D.O.O. nos proporciona un método para romper las "barreras" entre los datos y el procesamiento. Haciéndolo, podemos mejorar la calidad del software.

CAPITULO

4

DESARROLLO DE LA APLICACION

4. DESARROLLO DE LA APLICACION

FASES DEL PROCESO DE DESARROLLO.

- - **Análisis de Requerimientos.**
Proporciona la descripción general del sistema.
- - **Análisis del Dominio.**
Proporciona la estructura lógica del sistema.
- - **Diseño del sistema.**
Mapea la estructura lógica en la estructura física.
Concluye en versiones ejecutables.

4.1. ANALISIS DE REQUERIMIENTOS.

Objetivo Determinar el funcionamiento del sistema.

Productos *Objetivos del Sistema.*- Definición general de las metas del sistema.

Funciones del Sistema.- Casos básicos de uso del sistema.

4.1.1. DESCRIPCION DEL SISTEMA.

Debe ser un sistema que lea el contenido de un archivo de imágenes (en formato BMP) y despliegue la imagen en la pantalla; además muestre un menú para poder efectuar cualquiera de las operaciones listadas abajo, modificando dicha imagen y posteriormente mostrar la nueva imagen resultante como una nueva ventana, que pueda guardar la nueva

DESARROLLO DE LA APLICACION

imagen o volverla a modificar. También debe poder crear el histograma de una imagen y realizar las operaciones correspondientes sobre el histograma.

Debe tener la capacidad de imprimir tanto la imagen como el histograma y de almacenar cualquiera de las imágenes modificadas.

La interfaz con el usuario debe de ser de fácil manejo.

Operaciones a realizar sobre la Imagen.

Filtros en el Espacio:

SobelX.

SobelY

PrewittX.

PrewittY.

Laplaciano.

Promedio.

Mediana.

Segmentación en el Espacio:

OTSU.

Umbral.

Operaciones a realizar sobre el Histograma.

Ecuilización.

Gaussiana.

Línea.

Elongación

DESARROLLO DE LA APLICACION

4.1.2. DOCUMENTO DE ANALISIS REQUERIDOS.*

El sistema de procesamiento de imágenes debe ofrecer las siguientes funcionalidades.

- Abrir y desplegar un archivo de imagen.
- Guardar una imagen.
- Hacer operaciones (en el dominio del espacio) a la imagen.
- Desplegar histograma de la imagen..
- Imprimir Histograma.
- Modificar el Histograma.
- Imprimir una Imagen.

4.2. ANALISIS DEL DOMINIO

Objetivo	Definir en forma precisa las abstracciones de clases y objetos que modelan el dominio del problema
Productos	Diagramas de Clases. Especificaciones de Clases. Diagramas de Objetos para escenarios.

4.2.1. DEFINICION DE CLASES¹

Imagen
DatosImagen
Control
Entrada
Salida
Salida
Histo

4.2.2. DIAGRAMA DE CLASES

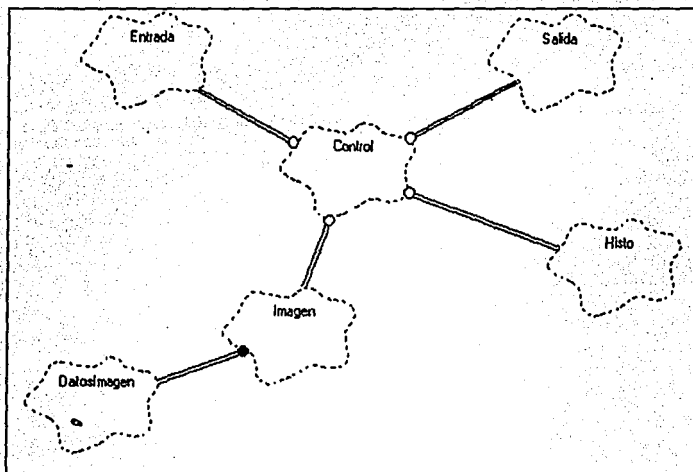


Figura 4.1. Diagrama de Clases de la Aplicación.

¹ Por razones de espacio no se muestra todo el desarrollo que se realizó, únicamente las clase finales.

4.2.3. ESPECIFICACION DE CLASES

Clase Control

La Clase Control es la intermediaria entre la interfaz con el usuario (que está conformada por las clases Entrada y Salida) y el dominio de nuestra aplicación.

Funciones

Carga_Imagen()	Filtra_Laplaciano()
Guarda_Imagen()	Filtra_Promedio()
Muestra_Histo()	Filtra_Mediana()
Imprime_Imagen()	Segmenta_Otsu()
Imprime_Histo()	Segmenta_Umbral()
Filtra_Sobel_X()	Ecualiza_Histo()
Filtra_Sobel_Y()	Gaussiana_Histo()
Filtra_Prewitt_X()	Linea_Histo()
Filtra_Prewitt_Y()	Elonga_Histo()

Clase Imagen

La Clase Imagen es del Dominio de nuestra aplicación, esta se encarga de manejar el arreglo de la imagen, con un atributo que son los datos de la imagen. Efectúa las operaciones sobre las imágenes y entrega los datos a la Clase control para que ordene a la Clase Salida que los despliegue.

DESARROLLO DE LA APLICACION

Atributos

Datos_Imagen

Matriz_Imagen

Funciones

Crea_Imagen()

Laplaciano()

Entrega_Imagen()

Promedio()

Sobel_X()

Mediana()

Sobel_Y()

Otsu()

Prewitt_X()

Umbral()

Prewitt_Y()

Clase DatosImagen

La Clase DatosImagen contiene los datos de la imagen como puede ser las dimensiones, la paleta utilizada, etc. Se decidió hacerlo como una clase aparte para que en caso de usar otros formatos gráficos sea muy sencillo su incorporación al sistema.

Atributos

Ancho

Alto

Paleta

Funciones

Entrega_Tamaño()

Entrega_Paleta()

DESARROLLO DE LA APLICACION

Clase Entrada

La Clase Entrada se encarga de validar las opciones que solicite el usuario. No está a nivel de detalle ya que no corresponde propiamente al análisis del dominio, es propio de las herramientas utilizadas para la implementación del sistema

Funciones

Valida_Opcion()

Valida_Archivo()

Clase Salida

La clase Salida se encarga de mostrar los resultados al usuario, ya sea en la pantalla, la impresora o guardándolos en un sistema de almacenamiento. Al igual que la Clase Entrada corresponde a la interfaz con el usuario.

Funciones

Despliega_Imagen()

Despliega_Histo()

Imprime_Imagen()

Imprime_Histo()

Graba_Imagen()

Clase Histo

La Clase Histo es del Dominio de nuestra aplicación, esta se encarga de crear y manejar el histograma de la imagen, con los atributos que son los niveles de gris de la imagen y sus valores

DESARROLLO DE LA APLICACION

correspondientes. Efectúa las operaciones sobre el histograma y entrega los datos a la Clase control para que ordene a la Clase Salida que los despliegue.

Atributos

Niveles_Gris

Valor_Nivel

Funciones

Crea_Histo()

Gaussiana()

Entrega_histo()

Linea()

Ecualiza()

Elonga()

4.2.4. DIAGRAMAS DE ESCENARIOS DE OBJETOS.

Los escenarios de casos de usuario modelan el comportamiento dinámico del sistema. Para representarlos, la notación nos proporciona los diagramas de objetos. Por cada escenario se crea un diagrama de objetos el cual representa todos los objetos que colaboran en el escenario. El diagrama puede acompañarse con una descripción enlistando en el orden correspondiente los mensajes con algún cierto comentario.

DESARROLLO DE LA APLICACION

4.2.5. FUNCIONES DEL SISTEMA

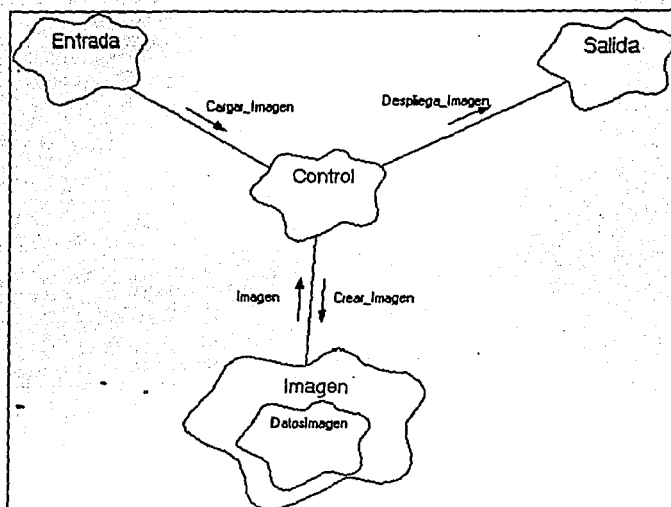
Escenarios

- Abrir y desplegar un archivo de imagen.
- Guardar una imagen.
- Hacer operaciones (en el dominio del espacio) a la imagen.
- Desplegar histograma de la imagen..
- Imprimir Histograma.
- Modificar el Histograma.
- Imprimir una Imagen.

4.2.6. DIAGRAMAS DE ESCENARIOS DE OBJETOS

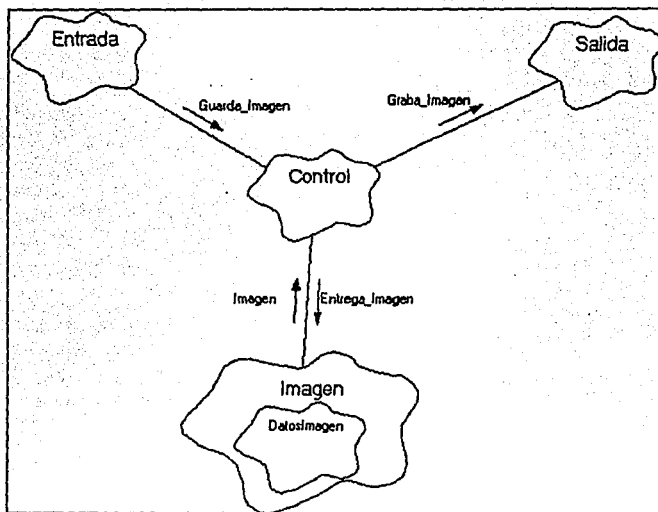
A continuación se presentan los diagramas de escenarios de objetos, solo mostrando los escenarios principales.

Abrir y Desplegar una Imagen.



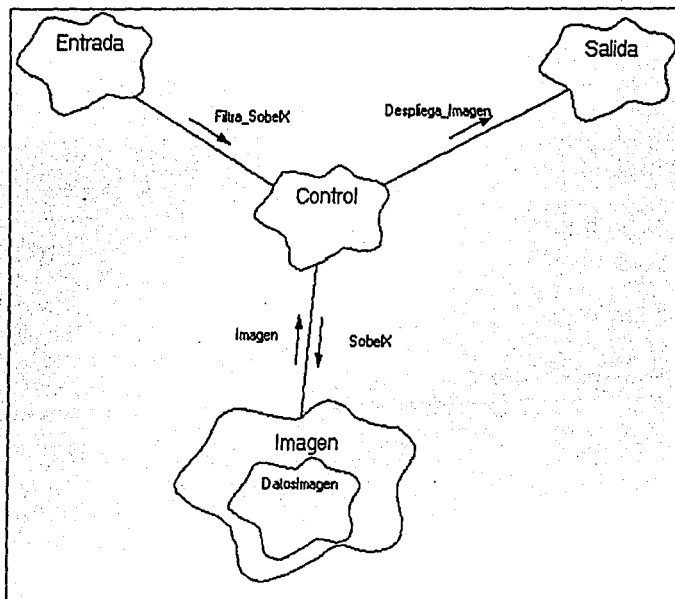
En este caso de uso el objeto entrada manda la selección que realizó el usuario al objeto control, este crea un objeto imagen con los datos correspondientes, y finalmente le indica al objeto salida que despliegue la imagen.

Guardar una Imagen.



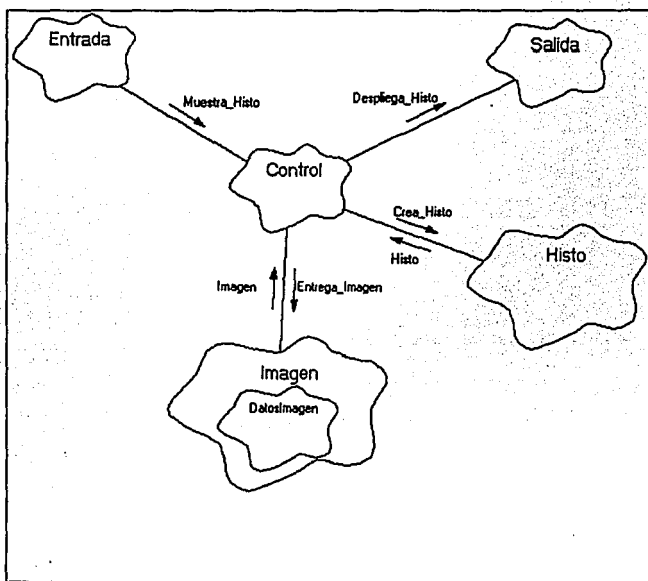
Este caso de uso nos muestra al objeto entrada mandando un mensaje que indica que se desea guardar la imagen, el objeto control pide al objeto imagen los datos de la imagen e indica al objeto salida que los guarde.

Filtrar una Imagen en el Dominio del Espacio.



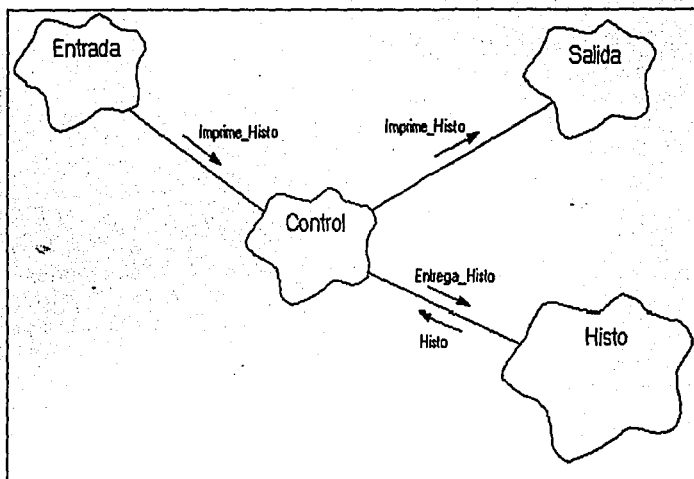
En este caso de uso el usuario indica que desea filtrar la imagen con el filtro SobelX, el objeto entrada valida la opción y le manda el mensaje correspondiente al objeto control, el cual le indica a la imagen la operación a realizar, esta la efectúa y regresa el resultado para que se le indique al objeto salida que lo muestre.

Desplegar el Histograma de una Imagen.



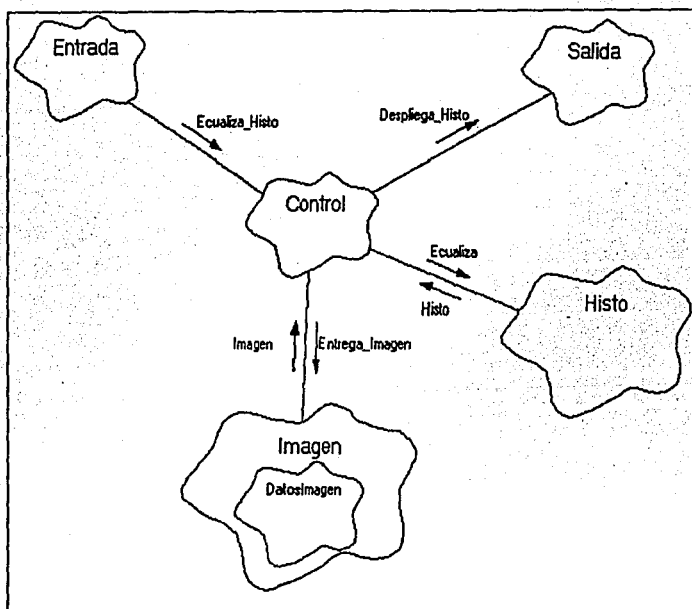
Con este caso de uso apreciamos que el objeto entrada manda un mensaje al objeto control para que muestre el histograma de una imagen, el objeto control pide los datos de la imagen y ordena a histograma que cree el correspondiente histograma, este le regresa los datos a control para que los pase a salida y los despliegue.

Imprimir el Histograma.



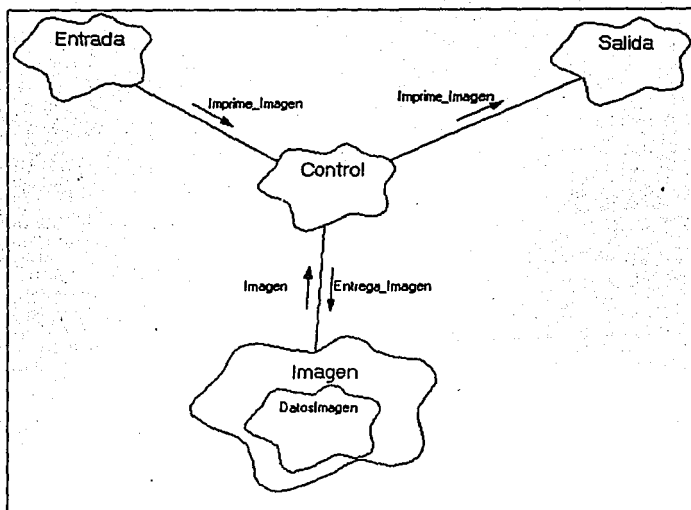
Aquí el usuario indica que desea imprimir el Histograma, por lo cual el objeto entrada pasa el mensaje al objeto control, el cual pide al histograma que le de los valores del histograma activo y posteriormente le indica a salida que los imprima.

Modificar el Histograma.



En este caso de uso el usuario indica que desea modificar el histograma con la opción de ecualización, el objeto entrada valida la opción y le manda el mensaje correspondiente al objeto control, el cual pide a la imagen los valores necesarios para indicarle a histograma la operación a realizar, esta la efectúa y regresa el resultado para que se le indique al objeto salida que lo muestre.

Imprimir una Imagen.



Aquí el usuario indica que desea imprimir la Imagen, por lo cual el objeto entrada pasa el mensaje al objeto control, el cual pide a la Imagen que le de los datos de la imagen activa y posteriormente le indica a salida que lo imprima.

4.3. DISEÑO DEL SISTEMA

El objetivo del diseño es extender y convertir el modelo del sistema, producto del análisis del dominio, en una implementación.

Consiste en aplicar una serie de decisiones para poder llegar a una implementación eficiente y además reutilizar lo más que se pueda de las bibliotecas que ya se tienen.

No existe por el momento una receta que nos diga como hacer el buen diseño, por lo pronto nos restringimos a presentar ciertos principios del buen diseño y una descripción muy general de algunas actividades.

Estos son los pasos que se siguen en la fase del diseño:

- Definir una arquitectura inicial.
- Planear versiones ejecutables.
- Desarrollar versiones ejecutables.²

4.3.1. Definición de arquitectura inicial

La arquitectura para el sistema PDI será:

Para funcionar en una PC con sistema operativo Windows 3.1 o superior.

En modo de ventanas.

Sin persistencia.

Sin comunicaciones.

²Por la extensión de este punto, solo se presentan los planes para esta parte.

DESARROLLO DE LA APLICACION

4.3.2. PLANEAR VERSIONES EJECUTABLES

El sistema hay que planearlo como una serie de pequeñas versiones, donde cada una de las versiones implementa una función o tarea bien definida. La versión ejecutable es un mini-sistema el cual por lo general es una combinación de:

- Nuevo código de esta versión.
- Código de versiones anteriores verificado.
- Utilerías artificiales, artefactos desarrollados rápidamente para simular las partes que quedan por desarrollar.
- Versiones simplificadas de clases nuevas, usando por ejemplo tipos primitivos en vez de tipos que faltan por definir.

4.3.3. PLAN PARA LAS VERSIONES EJECUTABLES.³

El plan para desarrollar las versiones ejecutables es el siguiente:

- Nombre de la versión. **WinPDI alfa 1.01**
 - Objetivo. **Probar que se carga y despliega una imagen.**
 - Clases que se implementarán. **Entrada, Salida, Control, Imagen, DatosImagen.**
 - Clases previamente implementadas que se usarán. **Ninguna.**
 - Casos de uso que se implementarán. **Cargar Imagen.**
 - Entradas y simulaciones de entradas. **Cargar la imagen.**
 - Salidas y simulaciones de salidas. **Desplegar la Imagen.**
-

³ Solo se muestran algunos ejemplos.

DESARROLLO DE LA APLICACION

- Nombre de la versión. **WinPDI alfa 1.02**
- Objetivo. **Probar que se modifica una Imagen.**
- Clases que se implementarán. **Entrada, Salida, Control, Imagen, DatosImagen.**
- Clases previamente implementadas que se usarán. **Ninguna.**
- Casos de uso que se implementarán. **Modificar Imagen.**
- Entradas y simulaciones de entradas. **Modificar la imagen.**
- Salidas y simulaciones de salidas. **Desplegar la Imagen resultante.**
-

- Nombre de la versión. **WinPDI alfa 1.02**
- Objetivo. **Probar que se modifica un Histograma.**
- Clases que se implementarán. **Entrada, Salida, Control, Imagen, DatosImagen.**
- Clases previamente implementadas que se usarán. **Ninguna.**
- Casos de uso que se implementarán. **Modificar Histograma.**
- Entradas y simulaciones de entradas. **Modificar el Histograma.**
- Salidas y simulaciones de salidas. **Desplegar el histograma resultante.**
-

La version final del programa se presenta en su versión ejecutable anexo a la presente tesis (disquete de instalación).

CAPITULO

5

5. APLICACION PRACTICA

El problema que a continuación vamos a tratar es un problema típico en la actualidad, en el campo de la medicina. Cuando un paciente sufre de un tumor, es necesario determinar si se trata de un tumor benigno o maligno, para poder así decidir si es necesario extirparlo o no. En caso de ser necesario el erradicar el tumor se necesita la información más fiable sobre la zona que está ocupando el mal, ya que de no tener esta información no se podría hacer nada que no implicara un riesgo aún mayor.

Desde este punto comienza a utilizarse el procesamiento de imágenes en esta área, desde el momento en que se hace necesario tomar la tomografía y el realizar otros procesos para poder obtener la información deseada.

Posteriormente se hacen varios cortes a la tomografía (como si fuesen rebanadas) a diferentes niveles del cuerpo hasta llegar al punto en donde se encuentra el problema. Una vez obtenido dicho corte entonces comienza otro proceso para obtener la mayor información posible de la imagen en cuestión.

Y es aquí donde comienza nuestro ejemplo.

A continuación presentamos una imagen de un corte encefalográfico de un recién nacido, que tiene un tumor cerebral; para poder ser operado es necesario determinar exactamente el área afectada por dicho mal.

La imagen fue tomada desde la parte superior de la cabeza, por lo que el rostro estará situado en la parte superior de la imagen.

En la figura 5.1 se muestra la imagen original (junto con su histograma) de la cual se desprende todo el proceso a realizar en el presente capítulo. El nombre de la imagen original es IMA_30.BMP, y el nombre que daremos a las sucesivas imágenes creadas será el de X1.BMP, X2.BMP y así sucesivamente.

APLICACION PRACTICA

Aprovechamos esta primera imagen para indicar las partes de las que consta nuestra interfaz con el usuario, y ver la sencillez con la que cuenta el sistema.

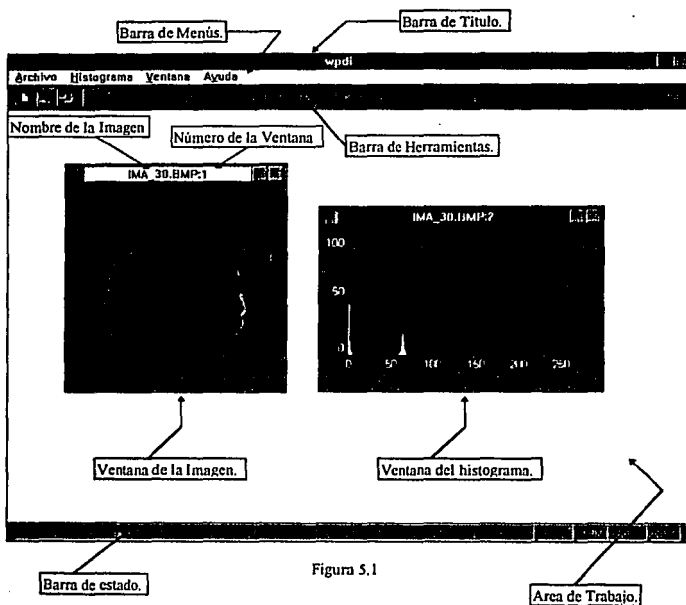


Figura 5.1

En la figura anterior se muestra la imagen original (IMA_30.BMP) en donde únicamente podemos apreciar (sabiendo de que se trata la imagen), el cráneo del bebe (la parte más clara alrededor de la cabeza), los oídos y la nariz. Con ello nos damos cuenta de la posición del cuerpo y el punto desde el que fue tomada la imagen. Desde esta imagen casi es imperceptible el tumor que es el problema a resolver.

APLICACION PRACTICA

Como no existen reglas sobre que procedimiento seguir en el procesamiento de imágenes (ello depende tanto de la imagen en cuestión como de la información a buscar en esta), optamos por realizar diferentes procesos a la imagen original, para que, a partir de éstas, seleccionar las que nos convengan y continuar con el proceso.

A continuación se muestran algunas imágenes obtenidas directamente de la original, por medio de diferentes procedimientos.

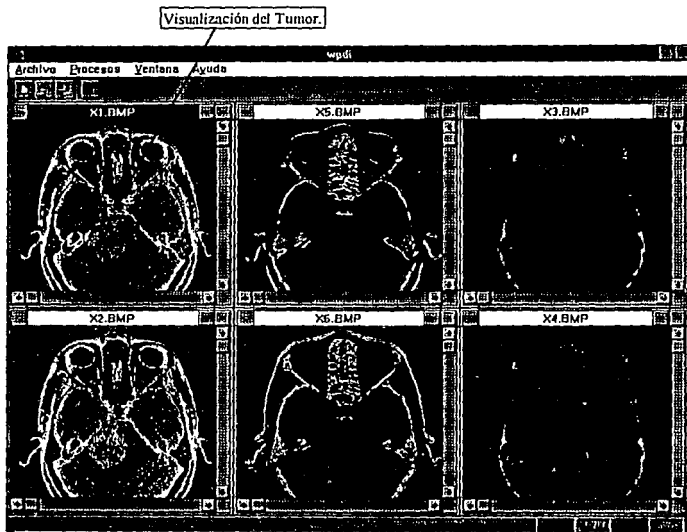


Figura 5.2

Podemos apreciar en la figura 5.2 que nos conviene tomar la imagen X1.BMP (correspondiente a la equalización del histograma), ya que nos permite apreciar más claramente las diferentes regiones de la imagen y ya podemos percibir un poco más sobre donde se encuentra el tumor en cuestión y la zona que abarca este.

Por otra parte la imagen X4.BMP (a la que se le ha pasado un filtro para eliminar el ruido) será de importancia, más adelante, en nuestro proceso.

APLICACION PRACTICA

A continuación mostramos la imagen original junto con la primera imagen con la que vamos a desarrollar el trabajo, esta corresponde a una eualización del histograma de la imagen original.

En esta nueva imagen ya podemos apreciar muchos detalles, incluso el soporte del cuello y la sabana que cubría al paciente.

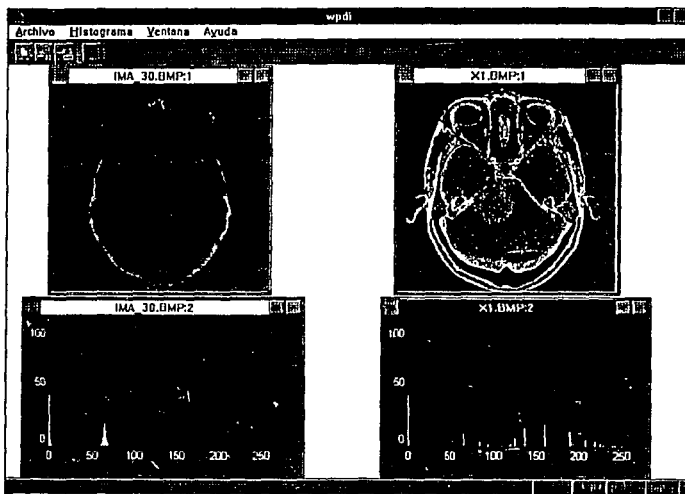


Figura 5.3

La imagen original comparándola con la imagen que muestra la eualización del histograma y sus respectivos histogramas. Podemos ver que en el histograma de la imagen original se concentran en una pequeña región los niveles de gris contenidos, lo que provoca una mala claridad de la imagen. Mientras que en el otro histograma se hace una distribución de los niveles de gris, dando por resultado una mayor definición general de la imagen.

APLICACION PRACTICA

En la figura 5.4 la primera de las imágenes muestra la imagen filtrada para eliminar el ruido, la siguiente tiene un proceso de filtrado por la técnica de Sobel_x y finalmente el umbral que nos marca el borde de las zonas en la imagen.

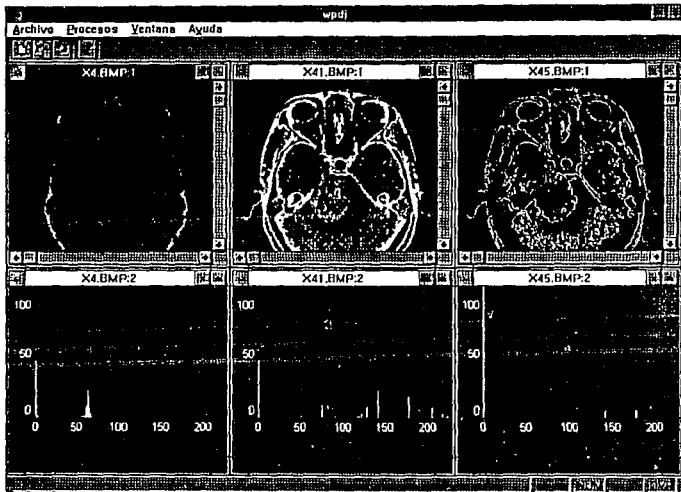


Figura 5.4

En los histogramas correspondientes podemos apreciar las diferencias, en el primero se aprecian dos zonas donde se concentra la gran mayoría de los niveles de gris; mientras que en el segundo existe una distribución más uniforme en todo el rango de niveles de gris; a diferencia de los anteriores el tercero únicamente está formado por dos niveles de gris.

APLICACION PRACTICA

A causa de la gran cantidad de procesos por el que pasa una imagen para poder obtener la información requerida, a continuación solo presentamos algunos de los más importantes.

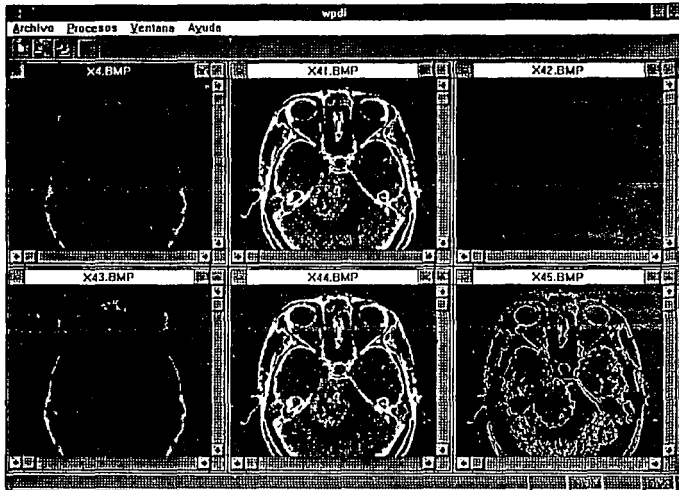


Figura 5.5

Las imágenes X4.BMP, X41.BMP y X45.BMP se explicaron anteriormente. La imagen X42.BMP presenta una segmentación, mientras que la imagen X43.BMP y X44.BMP presentan diferentes combinaciones de filtros

APLICACION PRACTICA

En la figura 5.6, mostramos las últimas imágenes, en las cuales consideramos que la zona del tumor ya se encuentra perfectamente definida y de las cuales ya es posible tomar decisiones sobre dicho tumor.

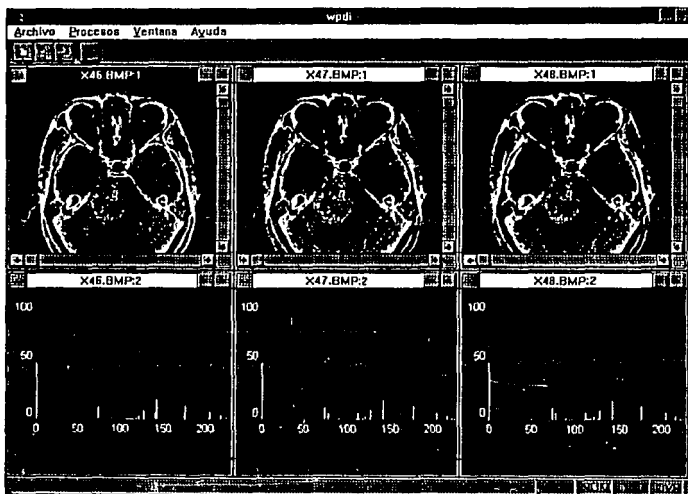


Figura 5.6

Se muestran sus correspondientes histogramas para apreciar las diferencias entre ellos, las cuales son mínimas. Podemos apreciar que los histogramas son muy parecidos, teniendo en común su distribución en todo el rango de niveles de gris.

APLICACION PRACTICA

En la figura 5.7 se presentan las dos imágenes más importantes, en las que se encierra todo el proceso realizado, estas son: la imagen original y la imagen final.

Podemos apreciar las grandes diferencias entre estas imágenes, a pesar de que una surge de la otra.

Se aprecia claramente que en la imagen original un médico no podría obtener la información necesaria para visualizar el tumor y la zona afectada, mientras que en la última dicho médico podría fácilmente conocer la magnitud del mal.

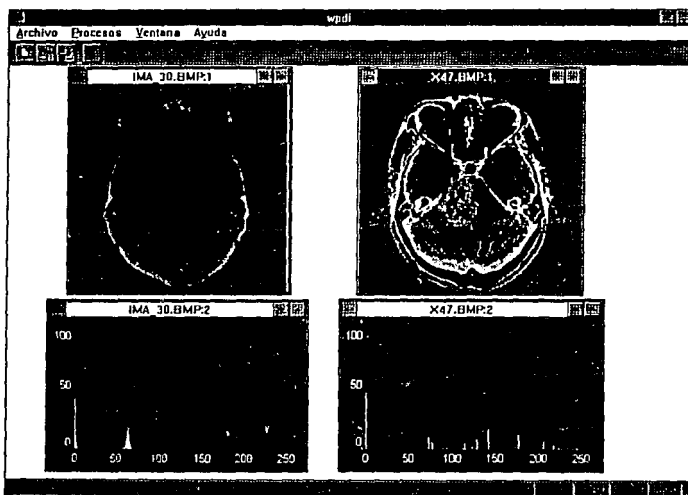


Figura 5.7

También podemos ver la diferencia enorme entre sus histogramas. En el primero, los niveles de gris se ven muy amontonados, mientras que el segundo hay una distribución más conveniente.

APLICACION PRACTICA

La figura 5.8 presenta una síntesis de todo el procedimiento llevado a cabo para este ejemplo.

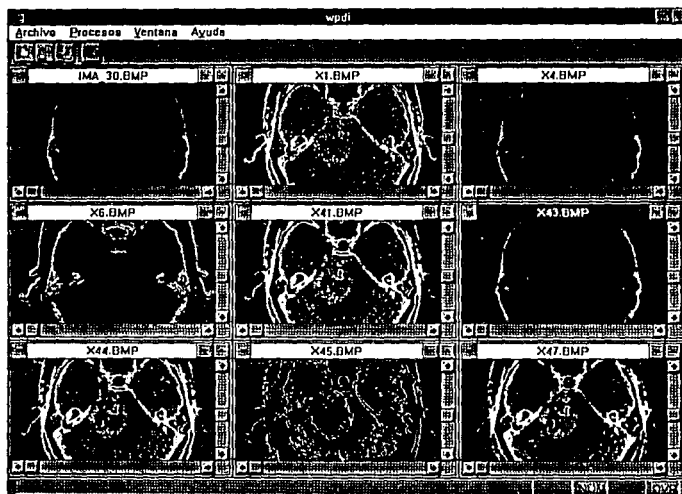
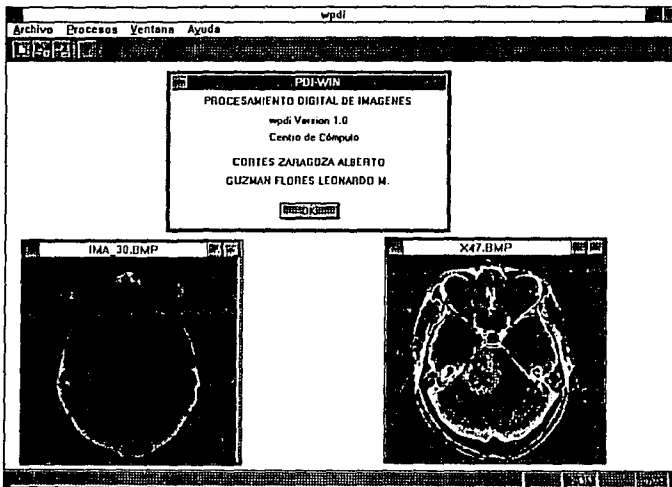


Figura 5.8

Para poder apreciar las diferencias, las imágenes se centraron en la parte afectada por el tumor.

APLICACION PRACTICA

Para finalizar la tesis, presentamos la ventana "acerca de" de la aplicación, sobre la imagen original y la imagen final del ejemplo.



CONCLUSIONES

Durante el desarrollo del sistema siempre tuvimos como objetivo el crear un programa que sirviese como apoyo didáctico a la enseñanza del procesamiento de imágenes, por lo que su uso debería ser lo más sencillo posible.

El resultado final de esta tesis ha sido un programa dentro del ambiente Windows con todas las virtudes de esta interfaz con el usuario, que es de muy fácil manejo. No se necesita ser un experto en el procesamiento de imágenes o un programador experimentado para interactuar con él, usarlo y obtener resultados.

Hacemos notar que el sistema de procesamiento digital de imágenes que hemos implementado, hace uso exclusivamente de las técnicas concernientes al dominio del espacio, las cuales son muy usadas y se puede llegar a obtener resultados como los mostrados en nuestro ejemplo, en el que por medio de filtros y manipulación directa del nivel de gris de la imagen podemos extraer su información más relevante.

Puesto que en México el procesamiento digital de imágenes no se encuentra tan difundido como en otras partes del mundo, consideramos que es de vital importancia el hacer desarrollo e investigación en esta área, ya que en caso contrario continuaremos dependiendo enteramente en este campo de otros países.

BIBLIOGRAFIA

BIBLIOGRAFIA

BIBLIOGRAFIA

DIGITAL IMAGE PROCESSING.

Gonzalez R Addison-Wesley,
Massachusetts, 1977

DIGITAL IMAGE PROCESSING

Castlerman R., Prentice Hall
U.S.A. 1979

DIGITAL IMAGE PROCESSING

Green, William b. Van Nostrand Reinhold
U.S.A. 1989

FUNDAMENTAL OF DIGITAL IMAGE PROCESSING

Jain, Anil k., Prentice Hall
U.S.A., 1989

DIGITAL IMAGE PROCESSING

Pratt, William K. Kiley Interscience
U.S.A., 1978

NUMERICAL RECIPES IN C. THE ART OF SCIENTIFIC COMPUTING.

Press W. H. , B. P. Flannery, S.A. Teukolsky and W.T. Vetterling
Cambridge University Press, New York, 1988.

FORMER USING BORLAND C ++ 4

Perry Paul, QUE
U.S.A. 1994

BITMAPPED GRAPHICS PROGRAMMING IN C ++

Luse Marv Addison-Wesley
U.S.A. 1993

INGENIERIA DEL SOFTWARE

Un enfoque práctico. 3a Ed.
Roger S. Pressman
Ed. Mc Graw Hill

OBJECT-ORIENTED

SOFTWARE ENGINEERING WITH C++

Darrel Ince
Ed. Mc Graw Hill.

ANALISIS Y DISEÑO ORIENTADO A OBJETOS.

James Martin y James J. Odell.
De. Prentice Hall

BIBLIOGRAFIA

BORLAND C++ 4 DEVELOPER'S GUIDE

Nabajyoti Barkakati.
De. SAMS

OBJECT-ORIENTED ANALYSIS AND DESIGN

With Applications. 2a Ed.
Grady Booch
Ed. The Benjamin/Cummings Publishing Company, Inc.

OBJECT-ORIENTED ANALISYS

Coad, P and E. Yourdon
Ed. Prentice Hill

OBJECT-ORIENTED PROGRAMMING IN TURBO C++

Robert Lafore
De. Waite Group Press.

PROGRAMACION AVANZADA DE GRAFICOS EN C PARA WINDOWS

Lee Adams
Ed. Mc Graw Hill

THE WAITE GROUP'S TURBO C++ BIBLE

Nabajyoti Barkakati.
De. SAMS

APLIQUE TURBO C++ PARA WINDOWS

Herbert Schildt.
Ed. Osborne / Mc Graw Hill.

OBJECT-ORIENTED PROGRAMMING

Cox B. J.
Ed. Addison-Wesley

SOFTWARE ENGINEERING WITH MODULA-2 AND ADA

Wiener, R. and R. Sincover
Ed. Wiley

TRANSMISION DE INFORMACION Y RUIDO

Mischa Schuwarztz
Ed. Mc Graw Hill