

UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE INGENIERÍA

CLASIFICADOR DE BAYES EN UN
SISTEMA DE RECONOCIMIENTO DE
PATRONES PARA RECONOCER OBJETOS
POR SU COLOR

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

P R E S E N T A N :

ORTÍZ SUÁREZ NOEMÍ
LOZADA MARTÍNEZ EVERARDO

DIRECTORA DE TESIS
DRA. ANA MARÍA VÁZQUEZ VARGAS

MÉXICO, D. F. 2008





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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 esta 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.

Índice

PRIMERA PARTE	4
Generalidades Sobre un Sistema Reconocedor de Patrones	4
INTRODUCCIÓN	5
CAPÍTULO 1	7
Sistema Reconocedor de Patrones	7
1. El reconocimiento de patrones	8
1.1 Reconocimiento del habla (RAH)	8
1.1.1 Diseño de un sistema de RAH	9
1.1.2 Decodificador acústico-fonético	9
1.1.3 Modelo del lenguaje	9
1.1.4 Reconocimiento de una gramática restringida	10
1.1.5 Características de los sistemas existentes	10
1.1.6 Clasificación	11
1.1.7 Usos y aplicaciones	11
1.2 Sistema de Reconocimiento Automático de Formas (SRAF)	13
1.2.1 Funcionamiento básico de un SRAF	13
1.2.2 Base de datos	14
CAPÍTULO 2	15
Métodos de clasificación	15
2. Clasificación de Patrones	16
2.1 No estadísticos	16
2.1.1 Método de las K-medias	16
2.2 Clasificadores Estadísticos	20
2.2.1 Clasificador de Bayes	21
2.2.2 Clasificador de Bayes para patrones normales	22
CAPÍTULO 3	26
Técnicas del Procesamiento Digital de Imágenes	26
3. Proceso de Clasificación.	27
3.1 Fundamentos.	27
3.1.1 Imágenes ópticas.	27
3.1.2 Imágenes digitales.	28
3.1.3 Nivel de gris.	28
3.2 Fundamentos de color.	28
3.2.1 Identificación y caracterización de la luz.	28
3.2.2 Generación.	29
3.2.3 Caracterización.	30
3.2.4 Colorimetría.	30
3.3 Modelos de Color.	30
3.3.1 Generalidades.	30
3.3.2 Modelos.	31
3.3.3 Formatos gráficos.	33
3.4 Histograma.	34
3.4.1 Definición.	34
3.4.2 Binarización.	35
3.5 Operaciones morfológicas.	35
3.5.1 Dilatación.	36

3.5.2	Erosión	37
3.5.3	Apertura	37
3.5.4	Cerramiento	38
3.6	Filtrado	39
3.6.1	Fundamentos	39
3.6.2	Convolución	40
3.6.3	Tipos	40
3.7	Segmentación	43
3.7.1	Objetivos	43
3.7.2	Métodos	43
3.8	Detección de discontinuidades	44
3.8.1	Detección de puntos aislados	44
3.8.2	Detección de líneas y de su orientación	45
3.8.3	Detección de bordes	45
3.9	Extracción de características	48
3.9.1	Descomposición del color	48
3.9.2	Captura analógica	48
3.9.3	Captura digital	50
SEGUNDA PARTE		51
Implementación del Sistema		51
CAPÍTULO 4		53
Obtención del Clasificador		53
4.	Problema que se quiere resolver	54
4.1.	Adquisición de los datos: de los datos físicos a las cantidades numéricas	54
4.2.	Características de la cámara	54
4.3	Método utilizado	58
4.3.1	Color	58
4.3.2	Forma	68
4.4	Datos escogidos, número de clases y variables discriminatorias	72
4.5	Clasificador K-Medias	73
4.6	Clasificador de Bayes	75
CAPÍTULO 5		80
Conclusiones		80
Apéndices		82
APENDICE I	Datos obtenidos	83
APENDICE II		86
a)	Sobre el programa de desarrollo: Matlab	86
b)	Códigos en Matlab	90
01)	Código para la binarización de las imágenes	90
02)	Código para la obtención del ancho y alto del contorno	91
03)	Código para la obtención de las tres clases con el algoritmo de K-Medias, usando las variables de color y forma	95
04)	Código para la obtención de las fronteras de decisión de Bayes con una variable (cociente del color)	97
c)	Ejemplo de ejecución del Código	99
Bibliografía		101

PRIMERA PARTE



*Generalidades
Sobre un Sistema
Reconocedor de Patrones*

INTRODUCCIÓN

Actualmente las imágenes, tanto en la vida cotidiana como en procesos industriales han tomado gran relevancia y debido al avance tecnológico ahora pueden ser procesadas de manera digital lo que hace más sencillo obtener información de ellas.

Debido a las múltiples aplicaciones de las imágenes dentro del campo del Reconocimiento de Patrones y de la evolución de sus técnicas decidimos desarrollar un sistema en el que utilizamos esta herramienta para clasificar colores y de esta manera nos planteamos como objetivo principal crear un programa que fuera capaz de clasificar frutas tomando en cuenta algunas características físicas.

En consecuencia, se analizaron imágenes de frutas, donde las técnicas utilizadas fueron: la descomposición del color, en sus componentes primarios RGB (rojo, verde, azul), histograma, binarización, operaciones morfológicas, es decir, operaciones que realizan transformaciones en las formas de los objetos de la imagen.

Todas estas técnicas permiten seleccionar los rasgos discriminantes de una imagen, lo que permite construir un vector de características que la representa. Dichas características tienen que cumplir con las siguientes propiedades:

- Capacidad discriminante; deben separar de la forma más clara posible las clases existentes
- Fiabilidad: Los objetos de una misma clase deben tener la menor dispersión

El problema que se quiere resolver es: clasificar frutas discriminando por colores.

A partir de un universo de estudio formado por un conjunto finito de clases (frutas), se trata de encontrar una función discriminante que a través de un proceso automatizado (SRP), es decir, un programa, determine de qué fruta se trata.

En este trabajo utilizamos como variable discriminante: el color, para el cual encontramos un clasificador.

Estructura de la Tesis.

Consiste en dos partes, la primera definimos el problema y, en la segunda, presentamos nuestra solución. La distribución por capítulos es como sigue:

- Capítulo 1, muestra las generalidades sobre los sistemas reconocedores de patrones (SRP) y entre sus aplicaciones, hablamos un poco sobre reconocimiento automatizado del habla (RAH) y sobre sistemas reconocedores de formas (SRAF).
- Capítulo 2, presentamos los principales tipos de clasificadores: estadísticos (en particular el clasificador de Bayes) y no estadísticos (métodos heurísticos de clasificación, K-medias).
- Capítulo 3, describimos algunas técnicas del procesamiento digital de imágenes aplicables al reconocimiento de patrones.
- Capítulo 4, detallamos cómo obtuvimos los datos y cómo fue obtenido nuestro clasificador.
- Capítulo 5, presentamos nuestras conclusiones.

CAPÍTULO 1

Sistema Reconocedor de Patrones

1. El reconocimiento de patrones

El reconocimiento de patrones es la ciencia que se ocupa de los procesos sobre ingeniería, computación y matemáticas relacionados con objetos físicos y/o abstractos, con el propósito de extraer información que permita establecer propiedades de o entre conjuntos de dichos objetos ^[1]. Alternativamente, puede definirse como el proceso mediante el cual se asignan clases a señales que son obtenidas a partir de objetos utilizando un sistema de sensores ^[2].

El objetivo es clasificar patrones con base en un conocimiento a priori o información estadística extraída de los patrones. Los patrones a clasificar suelen ser grupos de medidas u observaciones, definiendo puntos en un espacio multidimensional apropiado.

Un sistema de reconocimiento de patrones completo consiste en construir un sensor que recoja las observaciones a clasificar, un sistema de extracción de características que transforma la información observada en valores numéricos o simbólicos, y un sistema de clasificación o descripción que, basado en las características extraídas, clasifica la medición.

Entre las aplicaciones del reconocimiento de patrones se pueden mencionar el reconocimiento de voz, la clasificación de documentos, el reconocimiento de escritura, reconocimiento de caras humanas, colores, frutas, etcétera. Casi siempre utilizan técnicas de análisis de imágenes, un subconjunto del reconocimiento de patrones por ejemplo la toma imágenes digitales como entradas del sistema de clasificación.

El reconocimiento de patrones de voz es más complejo cuando se usan plantillas para generar variantes. Por ejemplo, en castellano, las frases a menudo siguen el patrón "sujeto-predicado", pero se requiere cierto conocimiento de la lengua para detectar el patrón en una o dos dimensiones. El reconocimiento de patrones se estudia en muchos campos, incluyendo psicología, etología¹, informática o procesado digital de la señal.

A continuación se proporcionan dos ejemplos del reconocimiento de patrones, el primero aplicado en el Reconocimiento Automático del Habla (RAH), y el segundo, un Sistema de Reconocimiento Automático de Formas (SRAF) o imágenes.

1.1 Reconocimiento del habla (RAH)

El Reconocimiento Automático del Habla (RAH) o de voz es una parte de la Inteligencia Artificial que tiene como objetivo permitir la comunicación hablada entre seres humanos y computadoras electrónicas. El problema que se plantea en un sistema de RAH es el de hacer cooperar un conjunto de informaciones que proceden de diversas fuentes de conocimiento (acústica, fonética, fonológica, léxica, sintáctica, semántica y pragmática), en presencia de

¹ Estudio científico del carácter y modo de comportamiento del hombre o de los animales.

ambigüedades, incertidumbres y errores inevitables para llegar a obtener una interpretación aceptable del mensaje acústico recibido.

1.1.1 Diseño de un sistema de RAH

Un aspecto crucial en el diseño de un sistema de RAH ^[3] es la elección del tipo de aprendizaje que se utilice para construir las diversas fuentes de conocimiento. Básicamente, existen dos tipos:

- Aprendizaje Deductivo

Las técnicas de Aprendizaje Deductivo se basan en la transferencia de los conocimientos que un experto humano posee a un sistema informático. Un ejemplo paradigmático de las metodologías que utilizan tales técnicas lo constituyen los Sistemas Basados en el Conocimiento y, en particular, los Sistemas Expertos.

- Aprendizaje Inductivo

Las técnicas de Aprendizaje Inductivo se basan en que el sistema pueda, automáticamente, conseguir los conocimientos necesarios a partir de ejemplos reales sobre la tarea que se desea modelar. En este segundo tipo, los ejemplos los constituyen las redes neuronales artificiales que son configuradas automáticamente a partir de muestras de aprendizaje.

En la práctica, no existen metodologías que estén basadas únicamente en el Aprendizaje Inductivo, de hecho, se asume un compromiso deductivo-inductivo en el que los aspectos generales se suministran deductivamente y la caracterización de la variabilidad inductivamente.

1.1.2 Decodificador acústico-fonético

Las fuentes de información acústica, fonética, fonológica y posiblemente léxica, con los correspondientes procedimientos interpretativos, dan lugar a un módulo conocido como decodificador acústico-fonético (o en ocasiones a un decodificador léxico). La entrada al decodificador acústico-fonético es la señal vocal convenientemente representada; para ello, es necesario que ésta se someta a un preproceso de parametrización.

En esta etapa previa es necesario asumir algún modelo físico, contándose con modelos auditivos y modelos articulatorios.

1.1.3 Modelo del lenguaje

Las fuentes de conocimiento sintáctico, semántico y pragmático dan lugar al modelo del lenguaje del sistema. Cuando la representación de la Sintaxis y de la Semántica tiende a integrarse, se desarrollan sistemas de RAH de gramática restringida para tareas concretas.

1.1.4 Reconocimiento de una gramática restringida

El reconocimiento de la gramática restringida trabaja reduciendo las típicas frases reconocidas a un tamaño más pequeño que la gramática formal. Este tipo de reconocimiento trabaja mejor cuando el hablante proporciona respuestas breves a cuestiones o preguntas específicas: las preguntas de “sí” o “no”, al elegir una opción del menú, un artículo de una lista determinada, etcétera. La gramática especifica las palabras y frases más típicas que una persona diría como respuesta rápida y después asocia esas palabras o frases a un concepto semántico.

Por ejemplo, un “sí” puede entenderse cuando se oye un “sip”, “vale”, “yes” u “okey”, y un “no” con un “nop”, “nada” o “en absoluto”.

Si el hablante dice algo que gramaticalmente no tiene sentido, el reconocimiento fallará. Normalmente, si el reconocimiento falla, la aplicación incitará al usuario a repetir lo que ha dicho y el reconocimiento se intentará de nuevo. Si el sistema está correctamente diseñado y es repetidamente incapaz de entender al usuario (debido a que no se ha entendido bien la pregunta, un acento cerrado, interferencias o demasiado ruido alrededor), se retirará y desviará la llamada a otro operador.

La investigación ^[3] muestra que las llamadas a las que se las pide replantear la pregunta o cuestión una y otra vez, en poco tiempo se frustran.

Los modelos del lenguaje más complejos necesitan para sus correctos funcionamientos grandes incorporaciones de voz y de texto escrito para el aprendizaje y la evaluación de los correspondientes sistemas. Gracias a ellos, se pueden abordar gramáticas más complejas y acercarse al Procesamiento de lenguajes naturales.

1.1.5 Características de los sistemas existentes

Los sistemas comerciales han estado disponibles desde el año 1990. A pesar del aparente éxito de estas tecnologías, muy pocas personas utilizan el sistema del reconocimiento del habla en sus computadoras. Parece ser que muchos de los usuarios utilizan el ratón y el teclado para guardar o redactar documentos, porque les resulta más cómodo y rápido a pesar del hecho de que todos podemos hablar a más velocidad de la que tecleamos. Sin embargo, mediante el uso de ambos, el teclado y el reconocimiento del habla, nuestro trabajo será mucho más efectivo.

Este sistema donde está siendo más utilizado es en aplicaciones telefónicas: agencias de viajes, atención al cliente, información etcétera. La calidad de estos sistemas de reconocimiento del habla ha ido aumentando y su eficacia cada vez es mayor.

Los programas comerciales más famosos son:

- Via Voice de IBM

- Dragon Naturally Speaking
- Philips FreeSpeech

En software libre hay:

- CVoiceControl Se graba la orden como entrenamiento.
- PerlBox Sin entrenamiento, pero en inglés.

1.1.6 Clasificación

Los sistemas de reconocimiento de voz pueden clasificarse según los siguientes criterios:

- Entrenabilidad: determina si el sistema necesita un entrenamiento previo antes de empezar a usarse.
- Dependencia del hablante: determina si el sistema debe entrenarse para cada usuario o es independiente del hablante.
- Continuidad: determina si el sistema puede reconocer habla continua o el usuario debe hacer pausas entre palabra y palabra.
- Robustez: determina si el sistema está diseñado para usarse con señales poco ruidosas o, por el contrario, puede funcionar aceptablemente en condiciones ruidosas, ya sea ruido de fondo, ruido procedente del canal o la presencia de voces de otras personas.
- Tamaño del dominio: determina si el sistema está diseñado para reconocer lenguaje de un dominio reducido (unos cientos de palabras por ejemplo reservas de vuelos o peticiones de información meteorológica) o extenso (miles de palabras).

1.1.7 Usos y aplicaciones

Aunque en teoría cualquier tarea en la que se interactúe con un ordenador puede utilizar el reconocimiento de voz, actualmente las siguientes aplicaciones son las más comunes:

- Dictado automático: El dictado automático es, en la actualidad, el uso más común de las tecnologías de reconocimiento de voz. En algunos casos, como en el dictado de recetas médicas y diagnósticos o el dictado de textos legales, se usan corpus² especiales para incrementar la precisión del sistema.
- Control por comandos: Los sistemas de reconocimiento de habla diseñados para dar órdenes a un computador (por ejemplo "Abrir Firefox", "cerrar ventana") se llaman Control por comandos. Estos sistemas reconocen un vocabulario muy reducido, lo que incrementa su rendimiento.
- Telefonía: Algunos sistemas permiten a los usuarios ejecutar comandos mediante el habla, en lugar de pulsar tonos. En muchos casos se pide al usuario que diga un número para navegar un menú.

² Un *Corpus lingüístico* es un conjunto, normalmente muy amplio, de ejemplos reales de uso de una lengua. Estos ejemplos pueden ser textos (típicamente), o muestras orales (normalmente transcritas).

- **Sistemas portátiles:** Los sistemas portátiles de pequeño tamaño, como los relojes o los teléfonos móviles, tienen unas restricciones muy concretas de tamaño y forma, así que el habla es una solución natural para introducir datos en estos dispositivos.
- **Sistemas diseñados para discapacitados:** Los sistemas de reconocimiento de voz pueden ser útiles para personas con discapacidades que les impidan teclear con fluidez, así como para personas con problemas auditivos, que pueden usarlos para obtener texto escrito a partir de habla. Esto permitiría, por ejemplo, que los aquejados de sordera pudieran recibir llamadas telefónicas.

1.2 Sistema de Reconocimiento Automático de Formas (SRAF)

Los sistemas de reconocimiento automático de formas (SRAF) ^[4], permiten identificar precisamente los objetos o cualidades de una imagen bidimensional para permitir calcular sus propiedades básicas como área, perímetro y forma.

Existen diversas maneras de implementar un SRAF, sin embargo, en el que se presenta a continuación se analiza únicamente una de ellas.

1.2.1 Funcionamiento básico de un SRAF

Un SRAF^[4] opera con un universo de trabajo previamente definido. Este universo de trabajo (UT) está formado por todos los posibles objetos individuales que se van a reconocer de manera automática. Estos objetos predefinidos se denominan patrones de clase.

La primera etapa del reconocimiento de formas es el proceso de aislamiento de los objetos, que consiste en extraer de la imagen los objetos individuales. La etapa siguiente se basa en la extracción de las características discriminantes de cada objeto, estas son las características que permiten diferenciar un objeto de otro. Por último, se compara las características discriminantes de los objetos del UT con las de los objetos encontrados y de esta forma se finaliza el proceso de reconocimiento.

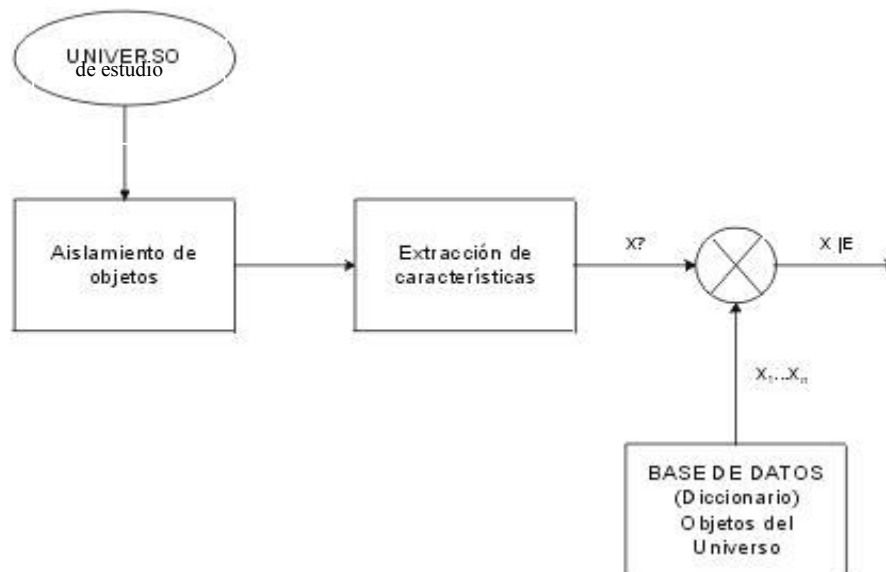


Fig. 2.1 Sistema de Reconocimiento Automático de Formas

Las técnicas utilizadas para construir el programa y su fundamentación teórica son las siguientes:

- El universo de estudio. Consiste en el conjunto de objetos o cualidades que se estudia para la clasificación.

Las imágenes que admite como entrada el SRAF. Estas imágenes deben estar dadas en formato Device Independent Bitmap (.bmp).

- Aislamiento de objetos

Esta etapa consiste en la de detección de contorno y otra de vectorización. La etapa de detección de contorno convierte la imagen entrante en una imagen de dos colores, blanco y negro, que contiene sólo los contornos de los objetos en la imagen. Esta etapa es muy compleja debido al ruido que poseen las imágenes digitalizadas. La etapa de vectorización aísla los contornos detectados en la etapa anterior y los transforma en una representación vectorial basada en curvas Bezier³.

- Extracción de características

Se basa en determinar cuales son las características de una forma representada mediante curvas Bezier que la diferencian de otras formas distintas.

Las propiedades que deben cumplir la elección y la prueba de características son:

Las características deben ser discriminantes: valores numéricos diferentes para objetos de clases distintas.

Las características deben ser fiables: es decir, con cambios numéricos pequeños para objetos de una misma clase. Si no se cumple será necesario un enfoque estadístico en el diseño de las funciones discriminantes.

Las características no deben tener una correlación fuerte entre ellas, o sea, decir lo mismo con la máxima economía de términos (Variables).

Las características deben calcularse en un tiempo aceptable. Este es un requisito que puede llegar a ser determinante en ciertas aplicaciones de tiempo real.

Las características deben obtenerse con sensores económicos.

1.2.2 Base de datos

Los datos de las clases pertenecientes al universo de trabajo se pueden almacenar en una base de datos. En esta BD se guarda el objeto en su representación mediante curvas Bezier. Y su representación mediante características discriminantes. La primera representación se utiliza para la edición de las figuras mientras que la segunda es necesaria para la identificación de los objetos del universo analizado.

³ Se denomina curvas de Bézier a un sistema que se desarrolló hacia los años 1960, para el trazado de dibujos técnicos, en el diseño aeronáutico y de automóviles.

CAPÍTULO 2

Métodos de clasificación

2. Clasificación de Patrones

La clasificación de patrones es el proceso mediante el cual se asignan clases a señales que se han obtenido de objetos utilizando un sistema de sensores, estos objetos forman parte del mundo físico que forma el entorno del sistema de medición. Los objetos están agrupados de acuerdo a un criterio que esta relacionado a los objetos físicos que originaron los objetos. Se asume que hay un número diferente de procesos físicos, cada proceso define una clase. La tarea del sistema de medición es recuperar la clase de cada uno de los objetos. Para eso, el sistema debe contar con un subsistema de sensores, con el cual las propiedades físicas de los objetos son captadas y transformadas en una señal. La señal es utilizada para asignar una clase a un objeto.

La señal obtenida a través del sistema de sensores puede tener varias formas. Se puede tener un grupo de señales dependientes del tiempo, y a su vez, éstas pueden ser discretas o continuas. Incluso podemos pensar en un grupo de señales dependientes del tiempo, ordenadas en un vector que cumpla con lo anterior, es decir, el vector de características.

El subsistema de sensores es diseñado de tal manera que el vector de características transmita la información necesaria para clasificar todos los objetos correctamente. Si este es el caso, los vectores de características de todos los objetos se comportan de acuerdo a un patrón regular. Idealmente, las propiedades físicas son escogidas de tal manera que todos los objetos agrupados dentro de una clase, en el espacio de medición, no se superpongan o traslapen dentro de los grupos formados por otras clases.

1.1 No estadísticos.

También denominados clasificadores por funciones de distancias, son métodos heurísticos que se basan en la distancia que hay entre los datos o agrupación, para poder clasificar, el siguiente es uno de ellos.

1.1.1 Método de las K-medias.

Este algoritmo está basado en seleccionar la distancia mínima de un punto por clasificar a los centros de masa de la clase ^[5], K es el número de clases que es conocida.

$$d(\vec{x}, \vec{z}_j(k)) = \sqrt{\sum_{x \in S_j(k)} (x - z_j)^2} \quad \forall \quad Z_j \quad j=1, \dots, K$$

Para \vec{x} en alguna clase.

En donde Z_j es la media de la clase j y \vec{x} es un vector de observaciones.

$d(\vec{x}, \vec{z}_j(k))$ denota la distancia euclidiana entre los puntos \vec{x} y $\vec{z}_j(k)$, k es la iteración.

$\vec{z}_j(k)$ centro de masa de la clase j en la iteración k.

$S_j(k)$ contiene los puntos asociados a la clase j en la iteración k .

El siguiente diagrama de flujo muestra el proceso que se sigue al aplicar éste algoritmo:

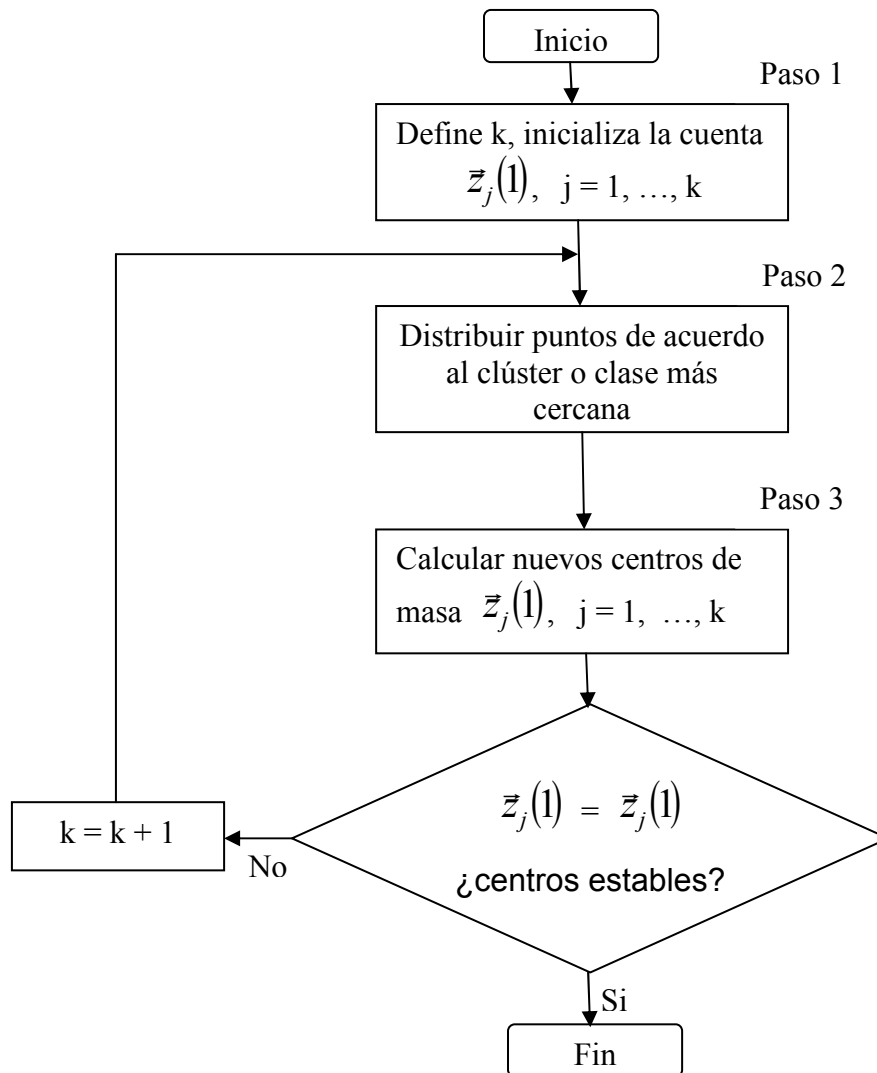


Fig. 2.1 Diagrama de flujo del algoritmo de k -medias.

El procedimiento de agrupamiento de éste algoritmo puede ser ilustrado por las medias del ejemplo que se muestra a continuación, que para mayor claridad contiene una muestra de $n=20$ observaciones con un patrón de dos dimensiones.

$\bar{x}_1(0,0)$	$\bar{x}_2(1,0)$	$\bar{x}_3(0,1)$	$\bar{x}_4(1,1)$	$\bar{x}_5(2,1)$
$\bar{x}_6(1,2)$	$\bar{x}_7(2,2)$	$\bar{x}_8(3,6)$	$\bar{x}_9(6,6)$	$\bar{x}_{10}(7,6)$
$\bar{x}_{11}(8,6)$	$\bar{x}_{12}(6,7)$	$\bar{x}_{13}(7,7)$	$\bar{x}_{14}(8,7)$	$\bar{x}_{15}(9,7)$
$\bar{x}_{16}(7,8)$	$\bar{x}_{17}(8,8)$	$\bar{x}_{18}(9,8)$	$\bar{x}_{19}(8,8)$	$\bar{x}_{20}(9,8)$

Tabla 2.1 Conjunto de 20 observaciones o puntos.

El objetivo es calcular la distancia mínima de \vec{x} a cada una de las clases y clasificar en donde la distancia sea menor.

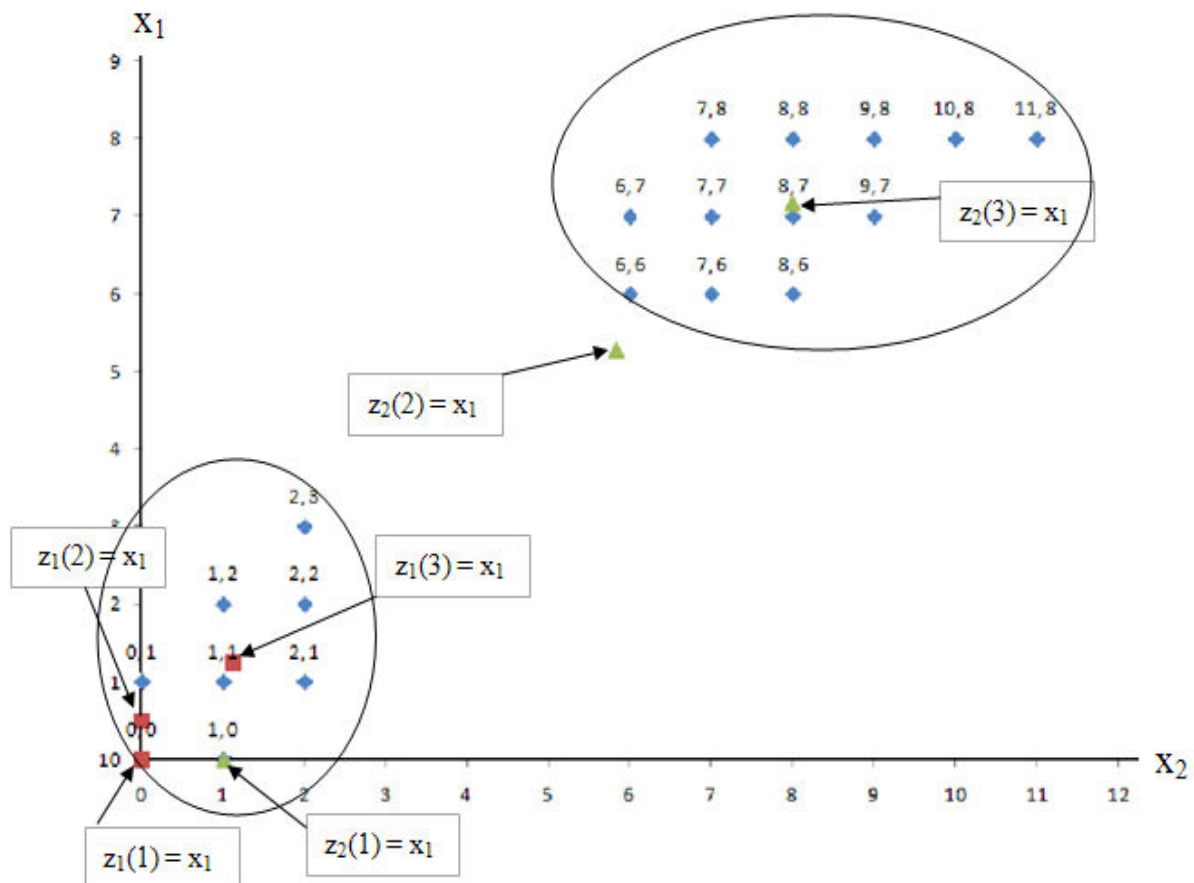


Fig. 2.2 Clasificación de los $n=20$ puntos en dos clases.

Ejemplo:

1. Se seleccionan dos muestras, de manera arbitraria, como centros de cada clase.

$$\begin{aligned} z_1(1) = x_1 &= (0,0)^T \\ z_2(1) = x_2 &= (1,0)^T \end{aligned}$$

El número entre paréntesis, de z , indica el orden de la iteración.

2. Distribuir las muestras x entre los dominios de clase de acuerdo a la siguiente regla.

$$x_1 \in S_1(1) \quad \text{desde } |x_1 - z_1(1)| < |x_1 - z_i(1)| \\ \forall i, i = 1, 2, \dots, K$$

$$x_4 \in S_2(1) \quad \text{desde } |x_4 - z_2(1)| < |x_4 - z_i(1)| \\ \forall i, i = 1, 2, \dots, K$$

Donde $K = 2$ para este caso, por lo tanto,

$$S_1(1) = [x_1, x_3] \\ S_2(1) = [x_2, x_4, x_5, \dots, x_{20}]$$

3. Actualizar los centros de clase con la ecuación siguiente.

$$z_j(k+1) = \frac{1}{N_1} \sum_{x \in S_j(k)} x \quad j = 1, 2, \dots, K$$

o

$$z_1(2) = \frac{1}{N_1} \sum_{x \in S_1(1)} x = \frac{1}{2}(x_1 + x_3) = \begin{pmatrix} 0.0 \\ 0.5 \end{pmatrix}$$

y

$$z_2(2) = \frac{1}{N_2} \sum_{x \in S_2(1)} x = \frac{1}{18}(x_2 + x_4 + \dots + x_{20}) = \begin{pmatrix} 5.863 \\ 5.28 \end{pmatrix}$$

Hay que hacer notar que el ajuste en el centro de las clases, los cuales son las medias de todas las muestras patrón, en su respectivo dominio de clase, minimizará la suma de los cuadrados de la distancia de todos los puntos en $S_j(k)$ al nuevo centro de clase. En este caso, $j=2$, $k=2$. N_1 y N_2 son respectivamente los números de muestras en $S_1(1)$ y $S_2(1)$.

4. Como $z_j(2) \neq z_j(1)$, con $j = 1, 2$, el algoritmo no converge, regresamos al paso 2 y repetimos el proceso. De otro modo, el proceso se termina.

5. Con los nuevos centros de clase, obtenemos

$$|x_l - z_1(2)| < |x_l - z_2(2)| \quad \text{para } l = 1, 2, \dots, 8$$

y

$$|x_l - z_2(2)| < |x_l - z_1(2)| \quad \text{para } l = 9, 10, \dots, 20$$

El dominio de la clase $S_1(2)$ y $S_2(2)$ son, respectivamente

$$S_1(2) = [x_1, x_2, \dots, x_8] \\ S_2(2) = [x_9, x_{10}, \dots, x_{20}]$$

6. Actualizar los centros de clase.

$$z_1(3) = \frac{1}{N_1} \sum_{x \in S_1(2)} x = \frac{1}{8} (x_1 + x_2 + \dots + x_8) = \begin{pmatrix} 1.13 \\ 1.25 \end{pmatrix}$$

y

$$z_2(2) = \frac{1}{N_2} \sum_{x \in S_2(2)} x = \frac{1}{12} (x_9 + x_{10} + \dots + x_{20}) = \begin{pmatrix} 8.00 \\ 7.17 \end{pmatrix}$$

7. Regresar al paso 2, ya que

$$z_j(3) \neq z_j(2) \quad j = 1, 2$$

8. Se produce el mismo resultado de la iteración previa:

$$S_1(3) = S_1(2) \quad \text{y} \quad S_2(3) = S_2(2)$$

9. Ya que $z_j(4) = z_j(3)$, con $j = 1, 2$, el algoritmo converge. Los centros de clase obtenidos son:

$$z_1 = \begin{pmatrix} 1.13 \\ 1.25 \end{pmatrix} \quad z_2 = \begin{pmatrix} 8.00 \\ 7.17 \end{pmatrix}$$

Por el proceso del algoritmo anterior, es fácil observar que los centros de clase son actualizados iterativamente y teóricamente si los datos están bien clasificados cada grupo converge a su media muestral. Este es el por qué a éste método se le conoce también como el algoritmo K-Medias. Otra de sus características, es que su ejecución se ve influenciada por el número de clases que se escogen en el inicio y también por el orden en que las muestras patrón son introducidas al sistema influyen en la rapidez del algoritmo. También es influenciado por las propiedades geométricas de los datos analizados.

1.2 Clasificadores Estadísticos.

Dada una partición en M clases conocidas $\omega_1, \omega_2, \dots, \omega_M$ se intenta llegar a un esquema de clasificación tomando en cuenta las propiedades estadísticas de las clases patrón. Podemos estudiar la función de densidad de probabilidad de las clases y aprovechar la inferencia estadística a fin de minimizar el riesgo que corremos de clasificar mal.

Se miden los eventos con consideraciones estadísticas sobre las clases ω_i de aleatoriedad bajo el supuesto de que se distribuyen como una ley de distribución Normal con media y varianza estimadas a partir de la muestra, el clasificador de Bayes es uno de ellos.

1.2.1 Clasificador de Bayes

El trabajo del clasificador es encontrar la decisión óptima que minimiza el riesgo promedio o costo de clasificar [6].

Todas estas estrategias suponen M clases $\omega_1, \omega_2, \dots, \omega_M$ como variables aleatorias y que estas clases tienen una función de distribución de probabilidad que puede ser investigada.

Según lo antes mencionado de Bayes tenemos que dada una partición A_1, A_2, \dots, A_M , como lo muestra la figura, es posible calcular la probabilidad del evento B utilizando la fórmula de Bayes:

$$P[A_i | B] = \frac{P[B | A_i]P[A_i]}{P[B]}$$

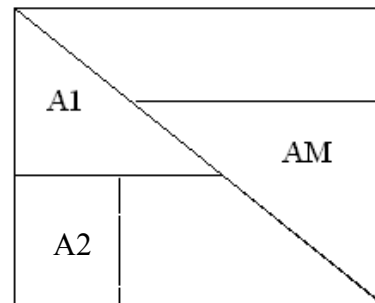


Fig 2.3 Particiones

Por el teorema de probabilidad total tenemos que

$$P[B] = \sum_{i=1}^n P[B|A_i] P(A_i)$$

$$P[A_i | B] = \frac{P[B | A_i] P[A_i]}{\sum_{i=1}^n P[B | A_i] P[A_i]}$$

La regla de Bayes minimiza la pérdida de escoger equivocadamente una clase w_i que no le corresponde.

Y que la regla de decisión para la clase w_i queda dada por:

$$d_i(\mathbf{x}) = p(\mathbf{x} | w_i) p(w_i)$$

Suponemos que $p(w_i)$ puede estimarse mediante la muestra, y que $p(\mathbf{x} | w_i)$ es la función de verosimilitud de la clase w_i , es decir, es la probabilidad condicional de \mathbf{x} dada w_i que se puede calcular a priori.

Otro hecho es que $p(w_i | \mathbf{x})$ es desconocida y para encontrar el umbral de rechazo entre las clases, mencionaremos el siguiente teorema:

1.2.2 Clasificador de Bayes para patrones normales.

Cuando se conoce o es razonable suponer que la función de probabilidad $p(\mathbf{x} | \omega_i)$ es normal multivariada, se cae en la solución de una clase de problemas en donde es fácil determinar las fronteras de decisión. Para el caso de una variable, la función de densidad de probabilidad es:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}\right]$$

Ec. 2.1

La cual queda completamente determinada si se especifican los valores de los parámetros μ y σ^2 . Donde μ representa a la media y σ^2 a la varianza.

Estos dos parámetros están definidos por

$$\mu = E[X] = \int_{-\infty}^{\infty} xp(x)dx$$

y

$$\sigma^2 = E[(X - \mu)^2] = \int_{-\infty}^{\infty} (x - \mu)^2 p(x)dx$$

Una manera de especificar que X se distribuye como una Normal es:

$$p(x) \rightarrow N(\mu, \sigma^2)$$

Para el caso de M clases o M variables tendremos a función de distribución normal multivariada, esto es:

$$p(x | \omega_i) = \frac{1}{(2\pi)^{M/2} |\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2} (x - \mu_i) \Sigma_i^{-1} (x - \mu_i)\right] \quad i=1, \dots, M$$

Ec 2.2

En donde

$$\mu_i = E_i(x) \quad y \quad \Sigma_i = E_i[(x - \mu_i)(x - \mu_i)']$$

μ_i es el vector de medias de las clases y

Σ_i es la matriz de varianza-covarianza de las clases

$|\Sigma_i|$ indica el determinante de la matriz de varianza-covarianza Σ_i .

Σ_i es una matriz simétrica y semidefinida positiva en donde los elementos de la diagonal principal son uno.

$\sigma_{ii} = \sigma^2$ de la clase i y los elementos restantes son:

σ_{ij} covarianza de la clase i con la clase j

Cuando x_i y x_j son independientes $\sigma_{ij} = 0$.

La función de densidad normal multivariada queda completamente determinada

mediante el cálculo de $\frac{n+1}{2}$ parámetros que corresponden a las n medias de las M clases y el valor de los elementos independientes de la matriz de varianza-covarianza de la ecuación 2.2

La frontera de decisión de Bayes está dada por la implementación de las funciones de decisión:

$$d_i(\mathbf{x}) = p(\mathbf{x} | \omega_i) p(\omega_i) \quad i=1, \dots, M$$

ya que los valores que nos interesa calcular se encuentran en el exponente de la función de densidad normal, es más fácil trabajar que tomar el logaritmo natural se tendrá:

$$d_i(\mathbf{x}) = \ln p(\omega_i) - \frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_i| - \frac{1}{2} \left[(\mathbf{x} - \mu_i)' \Sigma_i^{-1} (\mathbf{x} - \mu_i) \right] \quad i=1, \dots, M$$

Ec.2.3

ya que $\frac{n}{2} \ln 2\pi$ no depende de i se puede eliminar sin alterar la frontera de decisión y $d_i(\mathbf{x})$ se convierte en:

$$d_i(\mathbf{x}) = \ln p(\omega_i) - \frac{1}{2} \left[(\mathbf{x} - \mu_i)' \Sigma^{-1} (\mathbf{x} - \mu_i) \right] \quad i=1, \dots, M$$

Ecuación (2.2) y (2.3) representan las fronteras de decisión de Bayes

si todas las σ^2 son iguales para todas las clases entonces

$$\Sigma_i = \Sigma \quad \forall i=1, M$$

$$d_i(\mathbf{x}) = \ln p(\omega_i) + \mathbf{x}' \Sigma^{-1} \mu_i - \frac{1}{2} \mu_i' \Sigma^{-1} \mu_i \quad i=1, \dots, M$$

que podemos identificar a las ecuaciones de decisión lineales.

$$d_i(\mathbf{x}) - d_j(\mathbf{x}) = \ln p(\omega_i) - \ln p(\omega_j) + \mathbf{x}' \Sigma^{-1} (\mu_i - \mu_j) - \frac{1}{2} \mu_i' \Sigma^{-1} \mu_i + \frac{1}{2} \mu_j' \Sigma^{-1} \mu_j = 0$$

La cual representa un conjunto de funciones de decisión lineales.

Si además son independientes las clases entonces, por ejemplo, tomando el caso en que la Σ (matriz de varianza – covarianza es igual a la matriz identidad).

Caso 1, varianzas iguales para todas las clases y varianzas no correlacionadas.

$$\Sigma = \mathbf{I} \quad \text{la} \quad p(w_i) = \frac{1}{M} \quad i=1, \dots, M$$

Consideremos M clases y el problema de reconocimiento de una dimensión

$$p(x|w_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)$$

Ejemplo.

Suponemos M = 2 clases que siguen una distribución normal con:

$$w_1 \rightarrow N(0,2)$$

$$w_2 \rightarrow N(2,2)$$

Conocemos $\mu_1 = 0$, $\mu_2 = 2$ y $\sigma_1 = \sigma_2 = 2$

Para la frontera de decisión:

$$p(x|w_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right) \quad \text{y} \quad p(w_i) = \frac{1}{M}$$

$$d_i(x) = p(x|w_i)p(w_i) = \frac{1}{M} \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)$$

Tomando logaritmo natural

$$d_i(x) = -\ln \sqrt{2\pi} - \ln \sigma_i - \ln M - \frac{(x-\mu_i)^2}{2\sigma_i^2}$$

Quitando los términos independientes de i:

$$d_i(x) = -\ln \sigma_i - \frac{(x-\mu_i)^2}{2\sigma_i^2}$$

si

$$\mu_1 = 0 \quad \text{y} \quad \sigma_1 = 2$$

$$\mu_2 = 2 \quad \text{y} \quad \sigma_2 = 2$$

Sustituyendo en

$$d_i(x) = p(x|w_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)$$

Si

$$\mu_1 = 0 \quad y \quad \sigma_1 = 2 \quad y \quad \sigma_1^2 = 4$$

Sustituyendo

$$d_1(x) = \frac{1}{2\sqrt{2\pi}} \exp\left(-\frac{(x-0)^2}{2(4)}\right) = \frac{1}{2\sqrt{2\pi}} \exp\left(-\frac{x^2}{8}\right)$$

para la clase ω_2

$$\mu_2 = 2 \quad y \quad \sigma_2 = 2 \quad y \quad \sigma_2^2 = 4$$

$$d_2(x) = \frac{1}{2\sqrt{2\pi}} \exp\left(-\frac{(x-2)^2}{8}\right)$$

Para obtener el umbral hacemos $d_1 = d_2$ y resolviendo para x :

$$d(x) = d_1(x) - d_2(x)$$

$$\Rightarrow \exp\left[-\frac{x^2}{8}\right] = \exp\left[-\frac{(x-2)^2}{8}\right]$$

$$-\frac{x^2}{8} = -\frac{(x-2)^2}{8} \Rightarrow x = 1$$

Regla de decisión:

si patrón $x > 1$ entonces asignar x a la clase ω_1

si patrón $x < 1$ entonces asignar x a la clase ω_2

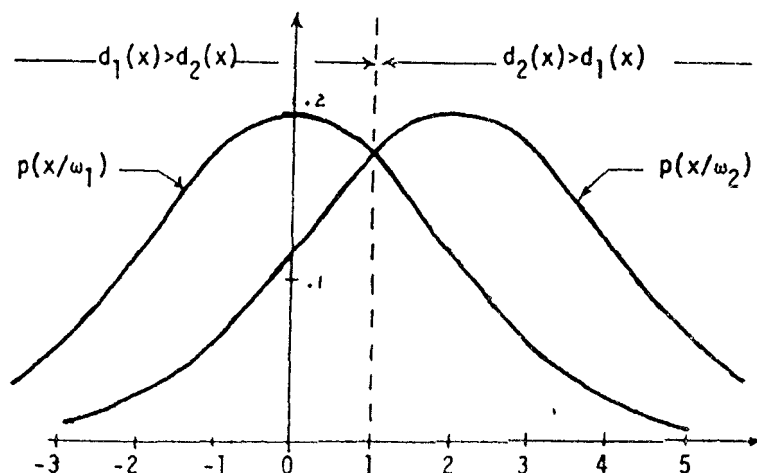


Fig 2.4 Frontera de decisión de Bayes se satisface en $x=1$

CAPÍTULO 3

Técnicas del Procesamiento Digital de Imágenes



3. Proceso de Clasificación.

Dada una imagen digitalizada, el proceso de reconocimiento de patrones se compone de tres etapas principales [7]. La primera, es la llamada segmentación de la imagen o aislamiento del objeto, en la cual, el objeto dentro de la imagen es aislado del resto de la escena.

La segunda etapa es llamada extracción de características. En esta es en dónde los objetos son medidos. Una medición es el valor de alguna propiedad cuantificable de un objeto. Una característica es una variable que se piensa va a discriminar.

La tercera etapa es aquella en donde se clasifican los objetos de acuerdo a los valores obtenidos en la etapa anterior, mediante un método de clasificación, para identificar el patrón, es decir, identificar el objeto que representa. Las técnicas utilizadas son, entre otras:

- Distancias
- Correspondencias de cadena
- Estadísticas
- Redes Neuronales

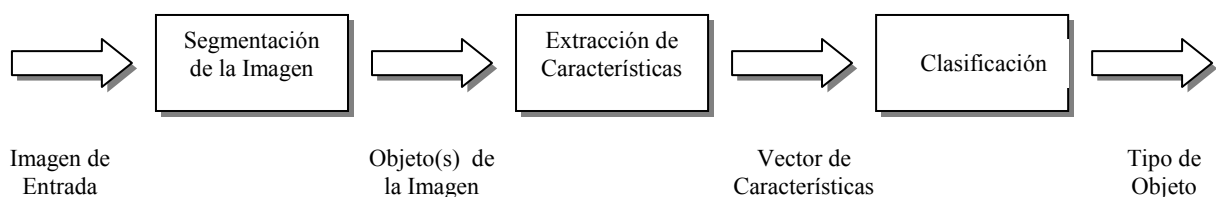


Fig. 3.1 Diagrama del proceso de clasificación.

1.1 Fundamentos.

2.1.1 Imágenes ópticas.

Una imagen es una función bidimensional de la luz y la intensidad. Se representa como $f(x,y)$, y para cada punto de la imagen (x,y) , la función devuelve el valor de la intensidad de la imagen en dicho punto [8].

Una imagen es una forma de energía y, por tanto, ha de ser finita y estrictamente mayor que cero, tomando valores reales enteros que verifican $0 < f(x,y) < \infty$.

La imagen consta de dos componentes o formas de energía:

- $i(x,y)$ es la *iluminación*, la luz que incide sobre un objeto. $0 < i(x,y) < \infty$
- $r(x,y)$ es la *reflectividad*, la cantidad de luz que refleja.

Un objeto puede absorber toda la luz (1) o nada (0). $0 < r(x,y) < 1$

Una imagen viene definida por:

$$f(x,y) = i(x,y) * r(x,y)$$

2.1.2 Imágenes digitales.

Proceden del muestreo espacial y en intensidad de la imagen óptica. Están formadas por una matriz de elementos (píxeles). El píxel, "*picture element*", es el valor de color o intensidad asociado a cada elemento de la matriz; tiene coordenadas espaciales. Puede tomar valores dentro del rango [0, 255].

2.1.3 Nivel de gris.

En las imágenes monocromáticas cada uno de los valores que puede tomar un píxel coinciden, al ser una imagen de un solo color. El valor de luminancia L estará comprendido entre unos valores mínimo y máximo.

$L_{\min} \leq L \leq L_{\max}$ donde L es el nivel de gris.

Escala de grises es un conjunto de los niveles de gris que puede tomar cada píxel (L : Level):

$[L_{\min}, L_{\max}]$ son 256 valores: [0, 255]

1.2 Fundamentos de color.

2.2.1 Identificación y caracterización de la luz.

No se conoce cómo el cerebro percibe los colores, pero se puede describir la naturaleza física del color —de la luz—, mediante su longitud de onda (λ). Un color se diferencia de otro según como sea la longitud de onda predominante [9].

- Longitud de onda **equilibrada** —> blanco. Refleja todas las λ .
- Longitud de onda **no equilibrada** —> coloreada. Refleja el color de una λ determinada.

Una fuente de luz se caracteriza mediante tres parámetros:

- Radiancia: cantidad de energía (luz) que emite una fuente de luz (radiante). Se mide en vatios (w).
- Luminancia: cantidad de luz que percibe el observador. Se mide en lúmenes (lm).
- Brillo: es un descriptor subjetivo de la intensidad de luz.

2.2.2 Generación.

- a) Colores primarios de luz: debido a la estructura del ojo humano, cualquier color puede especificarse en función de tres colores: rojo (red), verde (green) y azul (blue): $f(R, G, B)$

En 1931 la CIE (Comisión Internacional de la Iluminación) determinó que los valores fijos para las longitudes de onda, λ , eran:

$$R = 700 \text{ nm}$$

$$G = 546,1 \text{ nm}$$

$$B = 435,8 \text{ nm.}$$

Se les denomina colores primarios de luz y son de naturaleza aditiva, los colores se obtienen al combinarlos. La presencia de los tres colores primarios origina el blanco, de valor en escala de grises 255. Por el contrario, la ausencia de los tres colores provoca la oscuridad, negro = 0.

Los colores secundarios de luz se obtienen al combinar dos a dos los colores primarios de luz. Son el cian, el magenta y el amarillo.

primario + primario = secundario

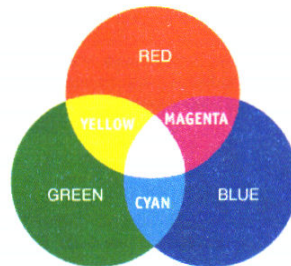


Fig. 3.2 Colores Primarios de Luz.

- b) Colores primarios de pigmentos: son aquellos que absorben o sustraen completamente un color primario de luz y transmiten o reflejan los otros dos. Son de naturaleza sustractiva, es decir, un color se obtiene substrayendo los primarios de luz.

Los colores primarios de pigmentos son los secundarios de luz.

Al unir los tres colores primarios de pigmentos se obtiene el negro. La ausencia de estos tres produce el blanco.

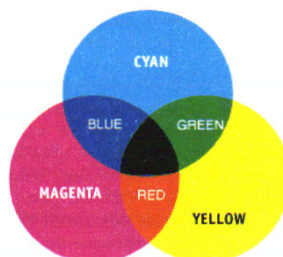


Fig. 3.3 Colores Primarios de Pigmentos.

- b) Aplicaciones: los colores primarios de luz son utilizados en televisión. Los primarios de pigmentos, secundarios de luz, impresora o los utilizados en la paleta de un pintor.

2.2.3 Caracterización.

a) Características: para describir un color se utilizan los siguientes parámetros:

- Tono (Hue): es el matiz, la tonalidad del color. Viene determinado por la longitud de onda, λ del color que se ve.
- Saturación (Saturation): indica la pureza relativa del color. En qué grado, medido en tanto por ciento, es puro un color. Si es 100% saturado, sólo existe ese color. Para otro valor, disminuye la cantidad de color puro.
- Brillo (Luminance): da una noción de la intensidad del color.

b) Especificaciones:

b. 1) Para especificar el color, si va dirigido a un ser humano, se utiliza la cromaticidad (engloba el tono y la saturación, es decir, las componentes cromáticas de color) y el brillo.

b.2) Si la especificación va dirigida a un componente (un ordenador, etcétera), se utilizan los coeficientes tricromáticos, que son valores triestímulo que indican la cantidad de rojo, verde y azul que tiene un color. Estos valores se obtienen de tablas o curvas que especifican qué cantidad de rojo, verde y azul tiene cada color y se normalizan entre 0 y 1:

$$x = x / (x + y + z) \quad y = y / (x + y + z) \quad z = z / (x + y + z)$$

2.2.4 Colorimetría.

Es la ciencia que se ocupa de la medida cuantitativa del color. Pretende medir de forma tangible los distintos colores o especificar algún procedimiento para que los dispositivos que utilicen colores se entiendan entre sí. Para ello, utiliza los modelos de color y los modos de color.

1.3 Modelos de Color.

2.3.1 Generalidades.

a) Objetivos: Obtener y disponer de un método de representación del color normalizado.

b) Definición: Un modelo de color consiste en la especificación de un sistema de coordenadas tridimensionales y de un subespacio dentro de ese sistema, donde cada color está representado por un único punto.

2.3.2 Modelos.

- **RGB**

Se representa mediante un sistema cartesiano cuyo subespacio es el cubo unidad (lado = 1). Cada color es un punto determinado por un vector que tiene como origen el de coordenadas y acaba en dicho punto. El origen de coordenadas representa el negro. Cada eje está asociado a un color (rojo, verde y azul). En cuestión de luminosidad, cuanto más cercano al origen, más oscuro será el color.

El sistema depende de la fuente de luz para crear el color.

El espectro visible se representa combinando los tres componentes básicos de luz coloreada en proporciones e intensidades diferentes (colores primarios): Rojo, Verde y Azul.

Al superponerse crean los colores secundarios: Cian, Magenta y Amarillo

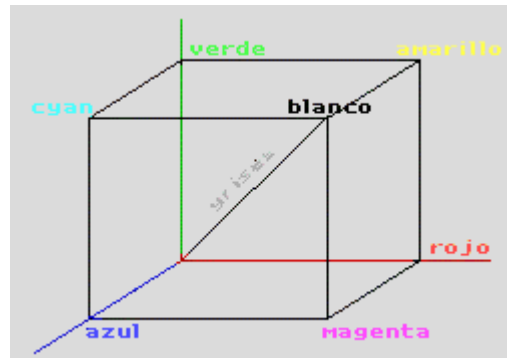


Fig. 3.4 Modelo RGB.

El sistema RGB será útil siempre que se quieran visualizar imágenes mediante adición de colores, como es el caso de los monitores y sistemas de televisión o las imágenes multiespectrales.

En la televisión, cada punto que se visualiza, llamado píxel, está formado por una triada de elementos que tienen tres componentes: R, G, B.

Otra utilidad es la visualización de imágenes en falso color, como las imágenes procedentes de un satélite. Por ejemplo, la imagen enviada por el satélite Landsat (7 bandas): R + G + 2 infrarrojos, los infrarrojos sustituyen al B, son no visibles, por lo que se dice que es una imagen en falso color.

- **HSL**

Significa tono, saturación y brillo o luminancia. El sistema sobre el que se representa es un doble tetraedro, llamado **sólido de color**. Separa por un lado la luminancia (de 0 —negro- a 1 —blanco) de la cromaticidad (saturación y tono), la cual se mide por los planos horizontales que cortan al sólido.

Para medir el tono y la saturación se utiliza un triángulo equilátero, denominado **triángulo de color**. La distancia del centro del triángulo, vértice que no representa los colores primarios de luz ni de pigmento, al color es importante.



Fig. 3.5 Modelo HSL.

Se toma un vector que une dicho vértice con el color, el ángulo de éste es el tono (hue) y la longitud o módulo del vector, la saturación. La saturación máxima es el 100%; cuanto más alejado del centro esté el color, más puro será.

Hue (Tono):

1. Es la longitud de onda de la luz reflejada o transmitida.
2. Se mide como una posición en la rueda de colores estándar, y se expresa en grados entre 0° y 360° (va del rojo al verde y luego al azul para volver al rojo).
3. Describe la sombra o el tinte del color

Saturation (Saturación):

1. También llamada cromatismo, es la fuerza o pureza del color.
2. Representa la cantidad de gris que existe en proporción al tono, y se mide como un porcentaje comprendido entre el 0% (gris) y el 100% (saturación completa).
3. En la rueda de colores estándar, aumenta al acercarse al borde y decrece al aproximarse al centro.

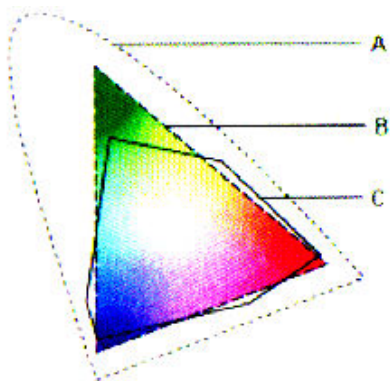
Luminance (Luminancia):

1. brillo, es la luminosidad u oscuridad relativa del color.
2. Normalmente se mide como un porcentaje comprendido entre el 0% (negro) y el 100% (blanco).

La utilidad de este modelo está basada en que desacopla la componente cromática de la componente de luminancia, que es la forma que tiene el ser humano de percibir bien los colores.

- **Gamas.**

Son la variedad de colores que pueden mostrarse o imprimirse en un sistema de color.



Los colores **fuera de gama** son aquellos que pueden visualizarse en un monitor pero no imprimirse.

A: L^*a^*b , es la gama más amplia (incluye a RGB y CMYK).

B: RGB: se usan en televisiones y monitores. No tiene el cian o amarillo puros.

C: CMYK, es la gama menor (son las tintas de impresión).

Fig. 3.6 Modelo HSL.

2.3.3 Formatos gráficos.

Al almacenar una imagen, su información: número de colores, compresión de la imagen, etcétera, se realiza con un tipo de formato ^[8].

Los tipos que hay son:

BMP: *BitMaP format*, imagen en mapa de bits. Es el más sencillo de todos. Ocupa mucho espacio. *.BMP

CCITT FAX: *International Telegraph and Telephone Consultative Committee Facsimile Transmission data formats*.

GIF: *Graphics Interchange Format*. Es el mejor criterio para transmitir datos a través de la línea telefónica.

GRASP: *Graphical System for Presentation*. Se utiliza para pseudolenguajes de programación que pueden ser usados para crear y animar demostraciones graficas.

JPEG: *Join Photographic Experts Group*. Creado como formato estándar internacional para imágenes, con ilimitadas aplicaciones.

MAC PAINT: *Macintosh Paintformat*. Para aplicaciones de Macintosh.

RLE: *Run Length Encoded*. Es un formato de mapa de bits comprimido estándar. La compresión es especialmente efectiva en aquellos mapas de bits que contienen regiones grandes de colores sólidos.

SunRaster: *Sun Raster Data Format*. Es un formato simple, principalmente para aplicaciones en Sun Workstations. Los cuatro primeros bytes van en hexadecimal. *.rast, *.ras, *.im, *.iml, *.im8, *.im24, *.im32.

TGA: para aplicaciones con productos Truevision y para datos gráficos.
*.TGA, TPIC

TIFF: *Tag image File Format*. Utilizado para aplicaciones de video, imágenes medicas, de satélites, etcétera * TIFF

TIGERILINE Census Files: *Topologically Integrated Geographic Encoding and Referencing/Line Censos Files*.

WMF: *Window MetaFile*. Utilizado en programas que funcionen bajo el entorno de Windows 3.0 o superior.

X Bitmap: *X Windows System Bitmap*. Formato para mapa bits (1 bit por píxel) que puede ser utilizado en C, incluso compilado con alguna modificación.

1.4 Histograma.

2.4.1 Definición.

De forma cuantitativa, el histograma de una imagen consiste en el número de píxeles que tiene la imagen para cada uno de los niveles de gris [8]. Se representa mediante un eje de coordenadas cartesianas. En las abscisas se representan los niveles de gris (de 0 a 255) y en ordenadas, el número de píxeles.

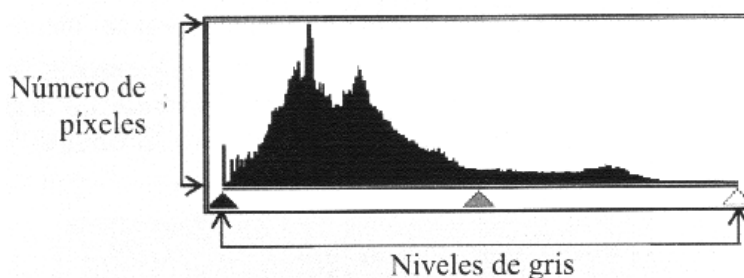


Fig. 3.7 Histograma.

Cualitativo.

De forma cualitativa, el histograma da una idea de la distribución de la intensidad en la imagen. Sirve para mejorar el brillo (aclarando u oscureciendo), el contraste, etcétera, así como para la segmentación, ya que puede ser útil para indicar agrupamientos de píxeles.

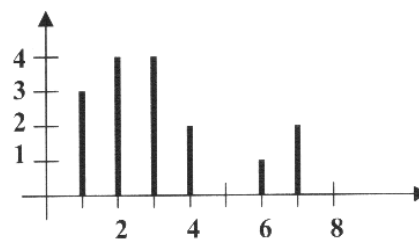


Fig. 3.8 Histograma Cualitativo.

Cuantitativo.

Cuantitativamente, sirve como descriptor de la imagen, pero presenta un inconveniente, cuando dos imágenes distintas tengan el mismo histograma.

3	2	7	1
1	4	2	3
1	4	2	3
6	7	2	3

Fig. 3.9 Histograma Cuantitativo.

Por tipo de imagen.

- a) Niveles de gris: pueden tomar valores desde 0 a 255.
- b) Blanco y negro (B/N): los píxeles toman el valor 0 ó 1.
- c) Imágenes en color: hay tres histogramas, uno para cada plano, correspondiendo cada uno con los colores básicos: rojo, verde y azul, que pueden tomar valores desde 0 a 255 y que se representan de forma superpuesta.

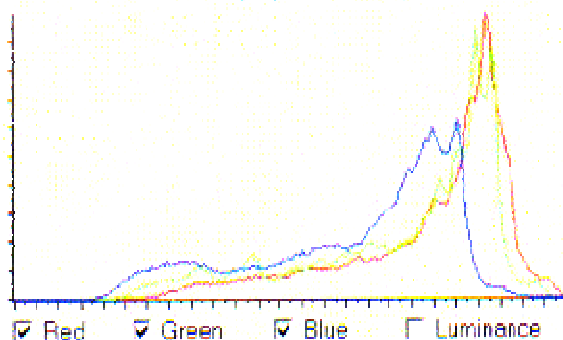


Fig. 3.10 Histograma en color.

Canales

Además de considerar el blanco y el negro como los valores máximo y mínimo, se pueden utilizar otros, ya que el histograma permite trabajar con canales.

Trabajando con canales RGB, si se usa el histograma para manipular el canal R (rojo), por ejemplo, el valor mínimo será el negro, pero el valor máximo será el rojo puro en vez del blanco. Los valores intermedios son una gradación de color del negro al rojo.

Se seguirá trabajando con 256 valores, ya que, en realidad cuando se trabaja con un canal es como trabajar con una imagen en blanco y negro, donde el blanco es el valor máximo de dicho canal (rojo en el R, verde en el G, y azul en el B) y el negro sigue siendo el negro.

Todo lo anterior se ha referido a imágenes de 8 bits por canal, pero pueden ser de más. En el caso particular de imágenes en formato CINEON (10 bits en escala logarítmica), primero hay que realizar una linealización de la imagen antes de poder aplicarle un histograma.

2.4.2 Binarización.

Consiste en generar una imagen en dos tonos (generalmente blanco y negro) a partir de otra con múltiples niveles de gris. Es un caso particular de la ampliación de contraste.

Entre sus aplicaciones están las siguientes:

- Resalte de las partes más brillantes de la imagen (suele corresponder a los objetos, frente a un fondo oscuro).
- En imágenes térmicas, segmentación de los cuerpos calientes de los fríos.
- Compresión de la imagen cuando la información de textura no es necesaria.
- Preparación de la imagen para una detección de bordes, formas, etcétera
- Eliminación del ruido introducido por un subsistema de captura cuando se digitaliza una imagen binaria.

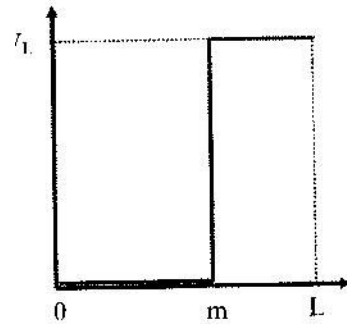


Fig. 3.11 Niveles binarios.

Como refleja el mapa de transición de luminancias, el valor umbral m determina qué valor pasan a tener los píxeles (blanco o negro).

Generalmente el fondo es negro y el objeto, en primer plano, blanco.

1.5 Operaciones morfológicas.

Estas operaciones realizan transformaciones en las formas de los objetos de la imagen.

2.5.1 Dilatación.

La dilatación no es una operación de memoria cero. Es una de las dos operaciones básicas (la otra es la erosión). Se necesita conocer información de la imagen original.

Se aplica típicamente a imágenes binarias aunque hay versiones que trabajan con imágenes en escala de grises. El efecto básico es agrandar gradualmente los límites de una región de píxeles de primer plano (agranda la imagen, haciendo más grandes sus contornos).

Para ello, toma dos estructuras de datos como entrada: la primera es la imagen que va a ser dilatada (con fondo de valor 0 y objetos de valor 1) y la segunda es un conjunto (generalmente pequeño) de coordenadas de puntos que se conocen como **kernel**. Se va pasando el kernel por todos los píxeles del fondo (de valor 0). Cuando alguno de los píxeles del kernel —cuyo píxel central está sobre el del fondo— coincide con un píxel de frente, se pone este píxel a 1 (primer plano o frente).



Fig. 3.12 Dilatación.

El kernel de operación que suele utilizar es una máscara de 3x3, 5x5 o cualquier otro número impar (7x7, etcétera):

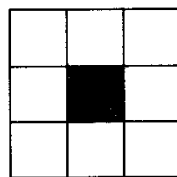


Fig. 3.13 Kernel 3x3, de la dilatación.

Dada una máscara centrada en el píxel central (2,2). Para realizar la dilatación se pasa por los píxeles del fondo. Al superponerla por todos los elementos del fondo (por todos los 0), si está sobre algún píxel que valga 1, se pone el central también a 1.

Esta operación se utiliza para:

- detección de bordes, tomando la imagen dilatada y luego restándosela a la imagen original. Así quedaría el contorno.
- ampliación de una región para disminuir el ruido.

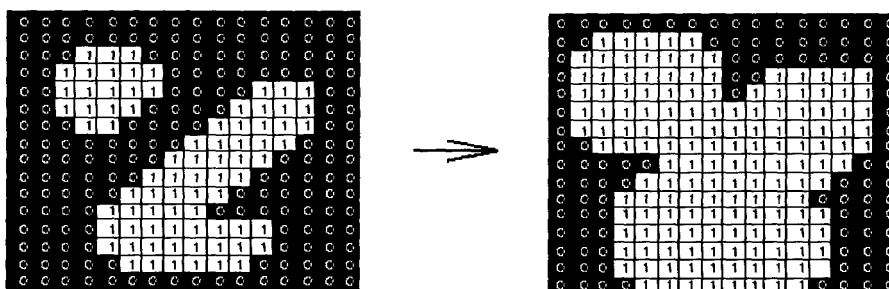


Fig. 3.14 Ejemplo de la dilatación de una imagen.

2.5.2 Erosión.

Es otra operación morfológica básica, junto con la dilatación. Se aplica típicamente a imágenes binarias aunque hay versiones que trabajan con imágenes en escala de grises. Realiza una erosión de los límites de una región del frente.

El procedimiento es el siguiente: se superpone un kernel por todos los elementos del primer plano (valor 1) y si alguno se superpone a uno que valía 0, el elemento central pasará a valer 0.

Entre sus aplicaciones están:

- Separar objetos que se tocan en una imagen binaria de forma que puedan contarse usando un algoritmo de etiquetado.
- Se puede usar también para detectar bordes tomando una imagen erosionada y restándosela luego a la imagen original.
- Como base para otras operaciones morfológicas.

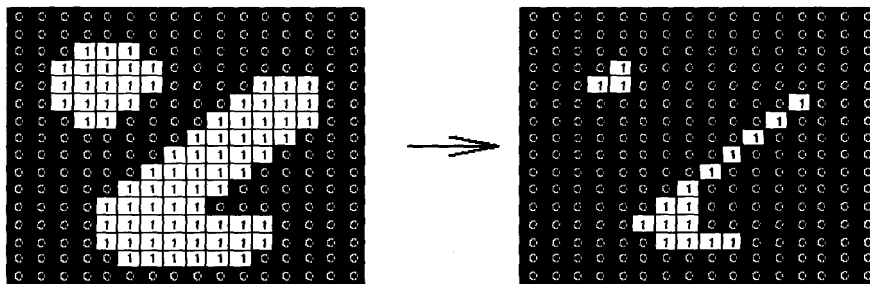


Fig. 3.16 Ejemplo de la erosión de una imagen.

2.5.3 Apertura.

De forma parecida a la erosión, pretende eliminar algunas zonas de los píxeles del frente, siendo en general menos destructiva que la erosión. Se utiliza para eliminar ruido de las imágenes. Es una combinación de las dos operaciones morfológicas básicas:

$$\text{APERTURA} = \text{EROSIÓN} + \text{DILATACIÓN}$$

Se puede realizar una erosión y posteriormente una dilatación usando el mismo kernel para ambas operaciones, pero otra forma de llevarla a cabo es utilizando un kernel con una forma determinada (cuadrado, triángulo, etcétera). Se superpone el kernel por los píxeles del primer plano de la imagen y se mantienen aquellas regiones que tengan una forma similar a este kernel, o que puedan contener completamente al kernel, mientras se eliminan las otras regiones de los píxeles del frente.

La apertura es el dual del cerramiento; abrir los



Fig. 3.15 Erosión.



Fig. 3.17 Apertura.

píxeles del frente con un determinado kernel es lo mismo que cerrar los píxeles del fondo con el mismo kernel.

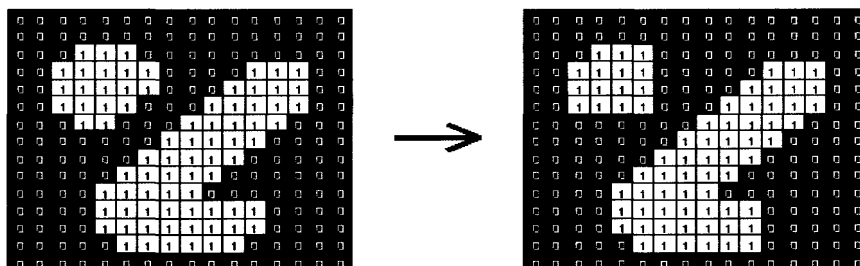


Fig. 3.18 Ejemplo de la apertura de una imagen.

2.5.4 Cerramiento.

Es parecido a la dilatación ya que ambas tienden a agrandar los bordes de regiones del frente, pero más constructiva que esta operación. El cerramiento dilata suavemente, cerrando un poco los huecos. Es también una combinación de las operaciones morfológicas básicas:



Fig. 3.19 Cerramiento.

$$\text{CERRAMIENTO} = \text{DILATACIÓN} + \text{EROSIÓN}$$

Se utiliza un kernel y se superpone por los píxeles del fondo (excepto en los bordes), preservando el fondo de regiones que tienen una forma similar a él o que pueden contener completamente al kernel, mientras que se eliminan todas las otras regiones de píxeles del fondo donde no entre (poniendo su valor a 1).

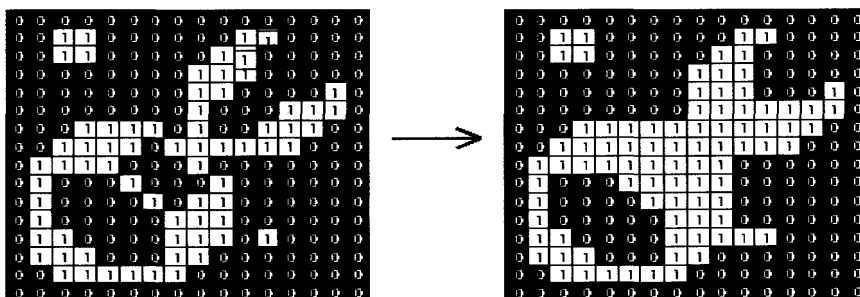


Fig. 3.20 Ejemplo del cerramiento de una imagen.

Es una operación de apertura pero en inverso. Se realiza una dilatación seguida de una erosión usando el mismo kernel. Es la operación dual de la apertura. Realizar el cerramiento de los píxeles del frente con un kernel particular equivale a abrir el fondo con el mismo kernel.

Ejemplos:

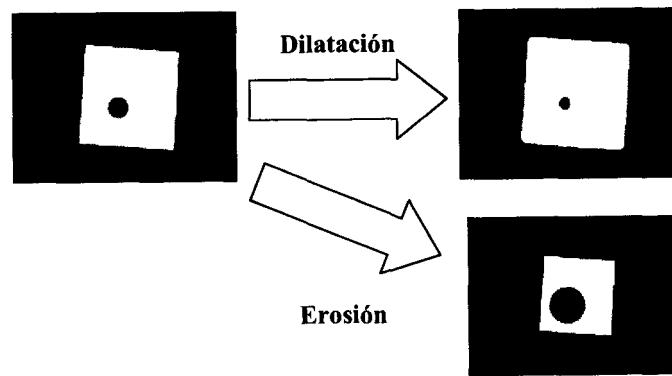


Fig. 3.21 Dilatación vs. Erosión.

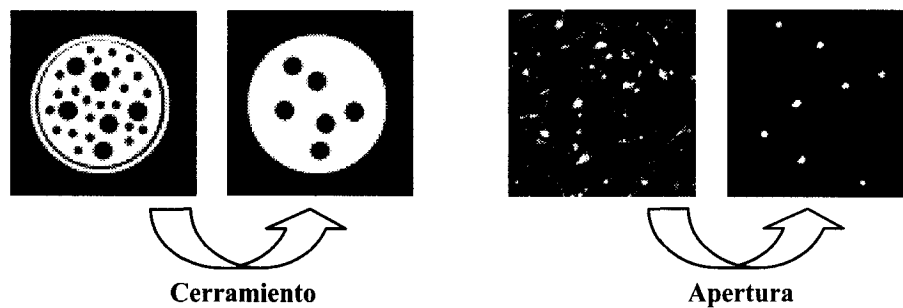


Fig. 3.22 Cerramiento vs. Apertura.

1.6 Filtrado.

El filtrado permite realzar o atenuar, suavizando los detalles, alguna característica de la imagen, así como eliminar el ruido o las partes que no interesen.

2.6.1 Fundamentos:

es necesario aplicar convolución: Imagen * Filtro

$$g(x, y) = f(x, y) * h(x, y)$$

El operador de convolución para realizar el filtrado en el dominio del espacio, es un kernel de convolución, el cual estará formado por una matriz. Se aplica a todos los píxeles de la imagen. Se caracteriza porque es invariante a la posición, es decir, el resultado de la operación no depende de la posición en la que está un píxel, sino de los valores del kernel y de la imagen.

2.6.2 Convolución:

Se centra el kernel en el píxel a tratar y si se utiliza como kernel una matriz 3x3, el resultado sería:

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 = \sum (w_i z_i) / p \quad l = 1 \dots 9 \quad p = \sum w_j$$

Se divide por p para evitar que quede en blanco.

El resultado del nuevo valor se queda en el píxel en el que se centró el kernel.

2.6.3 Tipos.

a) Filtro paso Bajo: Permite el paso de las bajas frecuencias de una imagen mientras que atenúa o rechaza los componentes de alta frecuencia. La frecuencia de corte viene determinada por el tamaño del kernel del filtro y los coeficientes del kernel.

Las zonas con pocos cambios conservan un valor similar al antiguo. En cambio, las zonas con muchos cambios pasan a tener un valor promedio del valor anterior con los píxeles del entorno.

Estos filtros se utilizan para eliminar ruido en una imagen (las altas frecuencias, al ser pocos píxeles, implica cambios bruscos). Restando de una imagen esa misma tras aplicarle un filtro de paso bajo se obtiene un realce de los bordes (filtro de paso alto).

Los kernels se caracterizan porque tienen todos los coeficientes positivos.

Ejemplos:

1 1 1	1 1 1	1 1 1	1 1 1	1 2 1
1 1 1	1 2 1	1 4 1	1 12 1	2 4 2
1 1 1	1 1 1	1 1 1	1 1 1	1 2 1
9	10	12	20	16
Average	LP1	LP2	LP3	“Gaussian”

Fig. 3.23 Ejemplos de filtros paso bajos.

b) Filtro paso Alto: Amplifica los detalles de alta frecuencia de una imagen mientras mantiene la integridad de los detalles de baja frecuencia. Resalta los bordes, las zonas de cambios.

El valor del centro de los kernels es grande, los coeficientes negativos pequeños se utilizan para reducir bastante el efecto del gran factor de ponderación central.

Se utilizan para acentuar o identificar mejor los objetos de una imagen. Hay que destacar que aunque se realzan los bordes, también se aumenta el ruido. Las partes con altas frecuencias se vuelven más brillantes y las de bajas frecuencias se oscurecen.

Ejemplos:

-1 -1 -1	0 -1 0	1 -2 1	0 -1 0
-1 9 -1	-1 5 -1	-2 5 -2	-1 20 -1
-1 -1 -1	0 -1 0	1 -2 1	0 -1 0
1	1	1	16
Mean removal	HP1	HP2	HP3

Fig. 3.24 Ejemplos de filtros paso alto.

c) Realce por Laplaciano: Se distingue de otros operadores y filtros por ser omnidireccional (realza los bordes en todas las direcciones). Calcula la derivada segunda de la función respecto a x e y . La primera derivada indica cómo va modificándose la intensidad (la pendiente indica mayor o menor velocidad de cambio). La segunda derivada muestra con qué rapidez se producen esos cambios.

Produce una definición de la forma de los bordes mayor que otras técnicas, teniendo el realce que produce tanto pendientes de intensidad positivas como negativas. Tiene como inconveniente que aumenta el ruido de la imagen.



-1 0 -1	-1 -1 -1	-1 -2 -1
0 4 0	-1 8 -1	-2 4 -2
-1 0 -1	-1 -1 -1	-1 -2 -1
1	1	1
LAPL1	LAPL2	LAPL3
Diagonal	Horizontal	Vertical
0 -1 0	0 -1 0	0 0 0
-1 4 -1	0 2 0	-1 2 -1
0 -1 0	0 -1 0	0 0 0
1	1	1

Fig. 3.25 Realce por Laplaciano y diferentes filtros de este tipo.

d) Detección de bordes y filtros de contorno: se utilizan filtros como el Sobel, que utiliza la derivada primera para detectar los bordes. Es no lineal (se producen grandes cambios ante pequeñas variaciones) dando mejores resultados con menor computación.

Trabaja basándose en el gradiente, que representa la manera en la que la intensidad cambia de píxel a píxel en una imagen. Es lo mismo que tomar la derivada de la imagen. El gradiente resalta especialmente aquellos lugares en los que en una imagen se producen los mayores cambios de intensidad píxel.

A estos filtros se les llama también filtros de contorno y se usan para ayudar a clasificar formas de objetos dentro de una imagen (como leucocitos en

imágenes médicas) así como para desarrollar mapas de contorno para estudios geográficos.

El filtro Prewitt es una aproximación de la primera derivada y el Sobel además tiene un suavizado promediado. Ambos son direccionales.

Sobel			Prewitt								
1	2	1	1	0	-1	-1	-1	-1	1	0	-1
0	0	0	2	0	-2	0	0	0	1	0	-1
-1	-2	-1	1	0	-1	1	1	1	1	0	-1
1			1			1			1		
Horizontal			Vertical			Horizontal			Vertical		

Fig. 3.26 Ejemplo de filtros Sobel y Prewitt.

e) Detección de bordes Canny: también se basa en la aplicación de la primera derivada, que es usada por que toma el valor de cero en todas las regiones donde no varía la intensidad y tiene un valor constante en toda la transición de intensidad^[10]. Por tanto un cambio de intensidad se manifiesta como un cambio brusco en la primera derivada, característica que permite para detectar un borde, y en la que se basa el algoritmo de Canny.

El algoritmo de Canny consiste en tres grandes pasos:

- Obtención del gradiente: en este paso se calcula la magnitud y orientación del vector gradiente en cada píxel.
- Supresión no máxima: en este paso se logra el adelgazamiento del ancho de los bordes, obtenidos con el gradiente, hasta lograr bordes de un píxel de ancho.
- Histéresis de umbral: en este paso se aplica una función de histéresis basada en dos umbrales; con este proceso se pretende reducir la posibilidad de aparición de contornos falsos.

Para la obtención del gradiente, lo primero que se realiza es la aplicación de un filtro gaussiano a la imagen original con el objetivo de suavizar la imagen y tratar de eliminar el posible ruido existente. Sin embargo, se debe de tener cuidado de no realizar un suavizado excesivo, pues se podrían perder detalles de la imagen y provocar un pésimo resultado final.

Una vez que se suaviza la imagen, para cada píxel se obtiene la magnitud y módulo (orientación) del gradiente, obteniendo así dos imágenes.

Las dos imágenes generadas sirven de entrada para generar una imagen con los bordes adelgazado, donde se consideran cuatro direcciones identificadas por las orientaciones de 0°, 45°, 90° y 135° con respecto al eje horizontal. Para cada píxel se encuentra la dirección que mejor se aproxime a la dirección del ángulo de gradiente.

La imagen obtenida en el paso anterior suele contener máximos locales creados por el ruido. Una solución para eliminar dicho ruido es la histéresis del umbral. Este proceso consiste en tomar la imagen obtenida del paso anterior, tomar la orientación de los puntos de borde de la imagen y tomar dos umbrales, el primero más pequeño que el segundo. Para cada punto de la imagen se debe localizar el siguiente punto de borde no explorado que sea mayor al segundo umbral. A partir de dicho punto seguir las cadenas de máximos locales conectados en ambas direcciones perpendiculares a la normal del borde siempre que sean mayores al primer umbral. Así se marcan todos los puntos explorados y se almacena la lista de todos los puntos en el contorno conectado. Es así como en este paso se logra eliminar las uniones en forma de Y de los segmentos que confluyen en un punto

1.7 Segmentación.

Consiste en subdividir una imagen en sus partes u objetos constituyentes.

2.7.1 Objetivos.

- Nivel de Subdivisión: Depende del problema a resolver. Deberá detenerse cuando los objetos de interés de una aplicación hayan sido aislados. Por ejemplo, en las aplicaciones autónomas de adquisición de blancos aire- tierra: para identificar vehículos de carretera, habrá que segmentar la carretera, y luego segmentar del contenido de la carretera los objetos cuya longitud o tamaño coincida con los de un vehículo.
- Segmentación autónoma: Es una de las tareas más difíciles del procesamiento de imágenes. Determina el eventual éxito o fracaso del análisis. En aplicaciones de inspección industrial, se suele tomar alguna medida de control sobre el entorno (objetos de referencia, etcétera).

2.7.2 Métodos.

Se utilizan algoritmos de segmentación, que se basan en:

a) Discontinuidades del nivel de gris: Consiste en dividir una imagen basándose en los cambios bruscos de nivel de gris. Es aplicable tanto a imágenes estáticas como dinámicas.

Las principales técnicas son:

- Detección de puntos aislados.
- Detección de líneas de una imagen.
- Detección de bordes de una imagen.

b) Similitud del nivel de gris: Se obtiene la segmentación basándose en:

- las propiedades de distribución de los píxeles, como la intensidad o el color,

- en encontrar directamente las regiones. utilizando para ello varios métodos. Las principales técnicas son:
 - Umbralización.
 - Crecimiento de regiones.
 - División y fusión de regiones.

1.8 Detección de discontinuidades.

La forma más común de de ver las discontinuidades es pasar una máscara o kernel a través de la imagen a analizar ^[9]. Para una máscara como la siguiente:

w ₁	w ₂	w ₃
w ₄	w ₅	w ₆
w ₇	w ₈	w ₉

Fig. 3.27 Máscara general de 3x3.

El procedimiento implica la suma de los productos de los coeficientes por los niveles de gris contenidos en la región encerrada en la máscara, Es decir, la respuesta en cualquier punto de la imagen es:

$$R = w_1 z_1 + \dots + w_9 z_9 = \sum_{i=1}^9 w_i z_i$$

Donde z_i es el nivel de gris asociado con el coeficiente de la máscara w_i.

2.8.1 Detección de puntos aislados:

Se realiza midiendo las diferencias ponderadas entre el punto central y sus vecinos. Se parte de la idea de que el nivel de gris de un punto aislado será bastante diferente del de sus vecinos.

Para la detección de los puntos se utiliza un filtro de paso alto, que incrementa las diferencias, mediante el siguiente kernel:

-1	-1	-1
-1	8	-1
-1	-1	-1

Fig. 3.28 Kernel para detectar puntos aislados.

Se sabe que se ha detectado un punto en la posición en la que está centrada la máscara si |R| > T siendo T un umbral no negativo, y R está dada por la ecuación anterior.

Aquí sólo interesan las diferencias suficientemente grandes ($>T$) para que se consideren puntos aislados (píxel único en un contorno uniforme).

Para determinar el valor umbral, si la imagen es conocida, ya está determinado. En otro caso, se hace un análisis de la imagen para ver cuáles son los valores típicos y cuáles se salen del rango.

2.8.2 Detección de líneas y de su orientación:

Dados los siguientes cuatro kernels de línea:

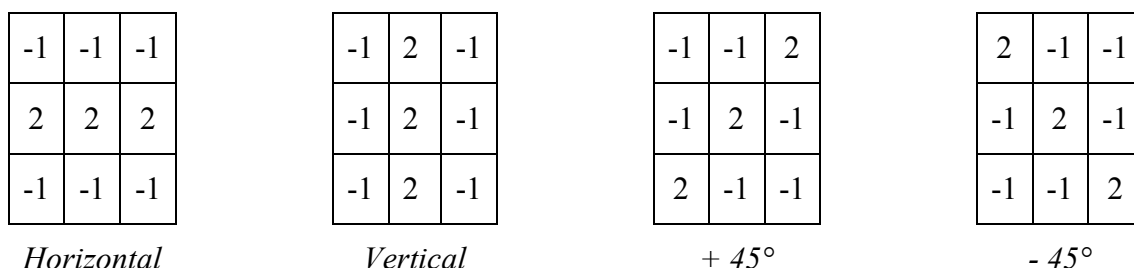


Fig. 3.28 Kernels de línea.

Y sean R_1 , R_2 , R_3 y R_4 las respuestas de dichos kernels al pasar sobre un punto concreto de la misma imagen. Si para este punto $|R_i| > |R_j|$, para todo $j \neq i$, este punto tendrá la mayor probabilidad de estar asociado con una línea en la dirección de la máscara i .

2.8.3 Detección de bordes

Es el método más común para detectar discontinuidades significativas en el nivel de gris (los puntos aislados y las líneas delgadas no aparecen frecuentemente).

Un borde es la frontera entre dos regiones con propiedades de nivel de gris relativamente distintas. Se parte del supuesto de que las regiones de estudio son lo suficientemente homogéneas para que la transición entre dos de ellas se pueda determinar sobre la base de las discontinuidades de nivel de gris solamente. En otro caso se utiliza umbralización o segmentación orientada a regiones. La mayor parte de las técnicas de detección de bordes es el cálculo de un operador local de derivación, como se puede observar en la figura siguiente, donde las transiciones entre los niveles blanco y gris se ven acentuadas al aplicar la primera y segunda derivadas.

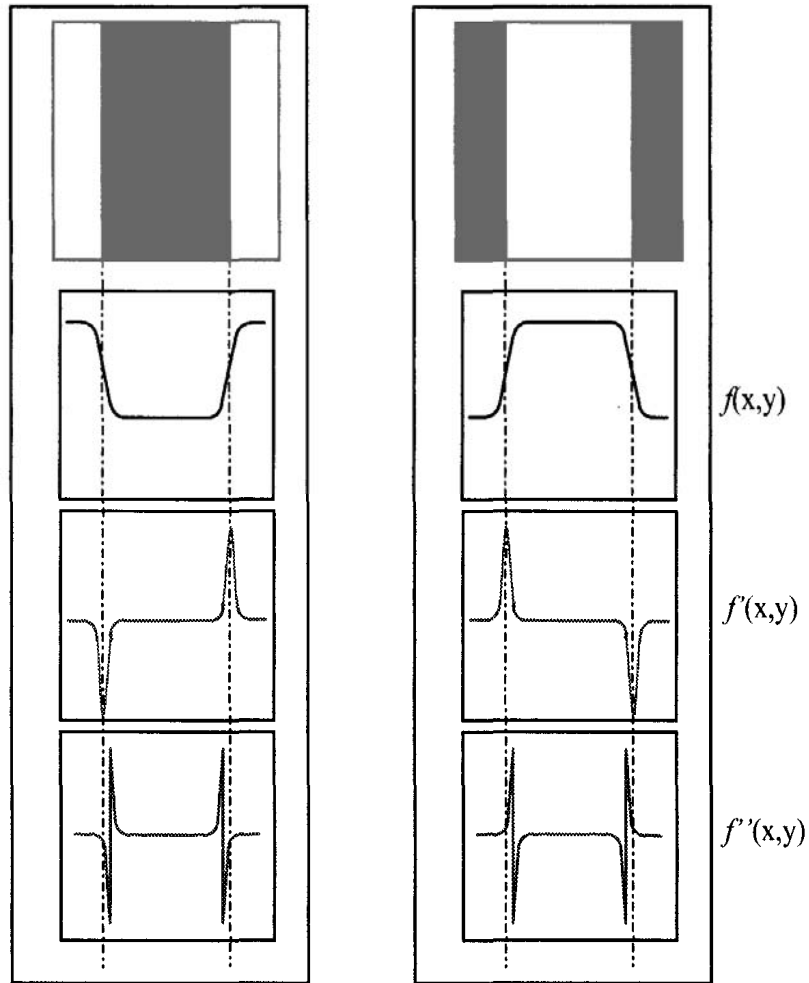


Fig. 3.29 Comportamiento del Laplaciano.

Los **operadores Gradiente** detectan diferencias entre regiones adyacentes y pendientes de las curvas (derivadas). El gradiente de una imagen $f(x,y)$ en la posición (x,y) es el vector:

$$\nabla f = \begin{pmatrix} G_x \\ G_y \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

donde la dirección (fase) es $\alpha(x,y) = \tan^{-1}(G_y/G_x)$.

El gradiente de un vector indica la dirección de la máxima variación de f en (x,y) .

$$\|\nabla f\| = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2}$$

donde *mag* representa *magnitud*.

El módulo es igual a la máxima variación de f en (x,y) por unidad de distancia en la dirección del gradiente. Es común aproximar la magnitud del gradiente por sus valores absolutos:

$$\|\nabla f\| \approx |G_x| + |G_y|$$

Una aproximación de esta ecuación para una región de 3×3 , en el punto z_5 , es

$$\|\nabla f\| \approx |(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)| + |(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)|$$

que se implementa con el kernel conocido como operadores de *Prewitt*.

Otra aproximación del módulo del gradiente la dan los operadores de *Sobel*.

$$\|\nabla f\| \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

Los operadores de Sobel proporcionan tanto una diferenciación como un suavizado (las derivadas realzan el ruido). Por ser una derivada, la suma de sus coeficientes es cero. Por tanto, la respuesta es cero siempre que el punto en cuestión y sus vecinos tengan el mismo valor.

También se utiliza el operador de *Roberts* de gradiente, que realiza la siguiente aproximación:

$$\|\nabla f\| \approx [(z_5 - z_9)^2 + (z_6 - z_8)^2]^{1/2} = |(z_5 - z_9)| + |(z_6 - z_8)|$$

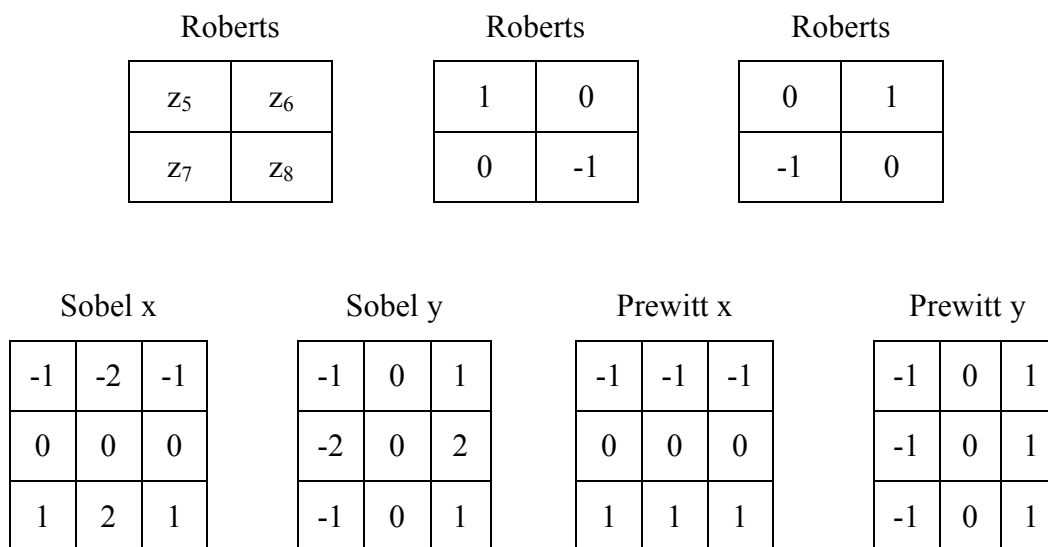


Fig. 3.30 Operadores Roberts, Sobel y Prewitt.

La suma de todos los coeficientes de una máscara es igual a cero, indicando una respuesta de cero en las áreas constantes, como sería de esperar tratándose de un operador diferencial.

El Laplaciano de una función bidimensional $f(x, y)$ es una derivada de segundo orden, definida por:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Puede implementarse digitalmente de varias maneras, siendo la más frecuente (región 3 x 3):

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$$

Laplaciano

0	-1	0
-1	4	-4
0	-1	0

El **requisito básico** para la definición del laplaciano digital es que:

- el coeficiente asociado con el píxel central sea positivo
- y los coeficientes asociados con los píxeles exteriores sean negativos.

Aunque responde a las transiciones de intensidad, no se suele utilizar para la detección de bordes porque

- produce bordes dobles,
- es muy sensible al ruido,
- no detecta direcciones de bordes.

Se suele utilizar para encontrar la *ubicación* de bordes utilizando sus propiedades de paso por cero (mediante la segunda derivada).

1.9 Extracción de características

2.9.1 Descomposición del color.

El empleo del color en el procesamiento de imágenes está motivado por dos factores ^[8]; primero, el color representa un potente descriptor que a menudo simplifica la identificación de un objeto y su extracción de una escena; segundo, nuestro ojo es capaz de discernir miles de matices e intensidades de color, en comparación con sólo dos docenas de niveles de gris.

a) Seres humanos: nuestro ojo es capaz de discernir miles de matices e intensidades de color, en comparación con sólo dos docenas de niveles de gris.

b) Sistema automático de procesamiento de imágenes: el color representa un potente descriptor que a menudo simplifica la identificación de un objeto y su extracción de una escena. Para que sea útil, además del color se necesita conocer su coordenada espacial. Las imágenes se pueden capturar por un dispositivo analógico o digital. En ambos casos, se siguen una serie de etapas.

2.9.2 Captura analógica.

a) Dispositivo de captura: obtiene la imagen analógica mediante cámaras:

- Cámara de fotos: se forma la imagen sobre una película de plástico con sales de plata fotosensible. Por el diafragma u obturador se regula la cantidad de luz y el tiempo que incide en la película. A partir del negativo que crea se pasa a positivo mediante técnicas de revelado,

consiguiendo la imagen en soporte de papel o en diapositivas. Es una imagen analógica.

- Cámara de vídeo: graba en una cinta magnética la impresión de lo que captura la cámara. Es una imagen analógica.

b) Digitalización: se necesita un convertidor analógico-digital (DAC) ya que las imágenes sobre las que se trabaja son digitales, aunque el modo de obtenerlas y transmitir las sea analógico. Como dispositivos digitalizadores se puede utilizar:

- Escáner: convierte la imagen analógica, que está en papel, en digital, con el formato que se le indique. Funciona iluminando el documento o la ilustración y, la luz reflejada a través de una serie de espejos y lentes, la dirige hacia un elemento fotosensible, normalmente un circuito electrónico (CCD) compuesto de fotositos¹ sensibles a la luz; éstos traducen los niveles de brillo a señales electrónicas procesadas posteriormente para formar la imagen digital. Hay otros sistemas usados por algunos escáneres, como pueden ser CIS (Sensor de Imagen de Contacto) que permite dispositivos más manejables a costa del sacrificio de la calidad, y PMT (Tubo FotoMultiplicador) que lee imágenes a muy alta calidad.
- Tarjeta digitalizadora (Frame Grabber): a partir de la imagen analógica que le llega por la entrada de vídeo, realiza un muestreo (25 fotogramas por segundo,
- por ejemplo) y la convierte a digital.

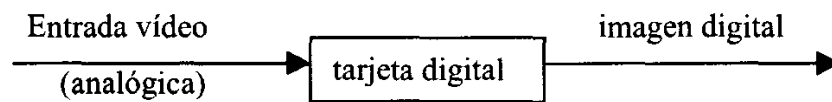


Fig. 3.31 Conversión analógico-digital.

c) Tratamiento: mediante el software del ordenador.

d) Visualización: utilizando dispositivos como el monitor, LCD's o impresoras. El monitor es lo más habitual y Consiste en un tubo de rayos catódicos que emite un haz de electrones sobre una pantalla de fósforo y que tiene una serie de pantallas reflectantes por las que pasa una tensión que va dirigiendo el haz de electrones; es una tecnología analógica. En cuanto a los LCD's (dispositivos de cristal líquido), consiguen una alta resolución. El cristal líquido es una estructura cristalina que tiene unas moléculas que cambian su orientación según se las someta a distintos campos magnéticos. Tiene una fuente de luz y dos mallas que contienen el líquido.

e) Almacenamiento: se pueden utilizar dispositivos magnéticos (disco duro, disquetes, etcétera), ópticos (cd, dvd).

¹ Los dispositivos CCD (Charge Coupled Device) se basan en multitud de acumuladores de luz que pueden ir dispuestos en línea (fax o escaners) o en cuadro. Cada uno de estos acumuladores de luz es un detector puntual de la imagen llamado fotosito, lo que nos generara un píxel en el monitor. El tamaño de estos fotositos es unas pocas micras (de 4 a 25)^[11].

2.9.3 Captura digital.

De manera similar al caso anterior, se puede tener en cuenta la siguiente serie de pasos:

a) Dispositivos de captura:

- Cámara de fotos digital.
- Escáner 3D: tiene un palpador y realiza una lectura de un objeto en tres dimensiones. Se utiliza en la fabricación de automóviles y en problemas de modelado.

b) Digitalización: este paso no existe, ya que el dispositivo de captura obtiene una imagen digital.

c) Tratamiento: mediante el software del ordenador.

d) Visualización: por pantalla, por LCD o por impresora, al igual que en el caso analógico.

e) Almacenamiento: en dispositivos magnéticos como el disco duro, ópticos, .

2.9.4 CCD (Charge Coupled Devices).

Los CCD actúan como fotosensores dispuestos en forma de matriz, formando una superficie de forma cuadrada o rectangular ^[12]. La cantidad de carga acumulada en cada fotosensor depende de la intensidad de la luz reflejada, la cual depende, en el caso del escáner, por ejemplo, de la sombra de cada píxel del documento. La carga del sensor genera un voltaje que recoge un conversor analógico-digital para convertirlo a un valor digital.

Los CCD se utilizan por que:

- Son dispositivos extremadamente lineales y por lo tanto efectúan una conversión directamente proporcional a la cantidad de carga acumulada.
- La relación señal/ruido es muy buena, por lo que la señal producida es limpia.

Una imagen en color está formada por la mezcla de los tres colores básicos: rojo, verde y azul. Los CCD se construyen para dar respuesta a esta característica. En este caso, en lugar de una fuente de luz hay tres fuentes de luz para determinar las intensidades de los píxeles para el rojo, el verde y el azul.

SEGUNDA PARTE



Implementación Del Sistema

Tomamos el caso de clasificar tres frutas y nuestra solución consistió en “medir” con la ayuda de una computadora las variables forma y color de cada una de ellas. Las cuales fueron:

- Manzanas rojas
- Manzanas amarillas
- Naranjas

Los resultados obtenidos mostraron que la variable “forma” no discriminó, y que era posible clasificar con una sola variable: el “color”, ya que las frutas son de diferentes colores: rojo, amarillo y anaranjado.

CAPÍTULO 4

Obtención del Clasificador



2. Problema que se quiere resolver.

A partir de una población de frutas, se quiere hacer una clasificación para poder separarlas automáticamente de acuerdo a sus características físicas; haciendo uso de los métodos que proporciona el reconocimiento de patrones y el procesamiento digital de imágenes, estas características son transformadas en señales digitales para ser procesadas mediante un ordenador para facilitar el proceso antes mencionado.

2.1. Adquisición de los datos: de los datos físicos a las cantidades numéricas.

El primer paso consistió en tomar fotografías de las frutas, en éste sentido la primera dificultad que encontramos fue la determinar que sería mejor, si tomar las fotos con una cámara fotográfica normal y después digitalizar las fotos, o tomarlas con una cámara digital y así obtener las imágenes directamente; nos decidimos por la segunda opción, ya que era más directa, pero para poder hacerlo tuvimos que tomar en cuenta las siguientes características:

a) Tipo de cámara.

Se utilizó una cámara digital marca Canon, modelo A560 de 5 megapíxeles.

b) Distancia del lente a la fruta.

La distancia de la fruta al lente de la cámara fue de 12 [cm].

c) Iluminación.

Fue obtenida por la luz natural o ambiente.

2.2. Características de la cámara

Modelo de la cámara: Cámara Canon Power Shot A530 ^[13].



Zoom: óptico 4x, digital 4x y combinado 16x.

Píxeles efectivos: 5.0 Megapíxeles.

Sensor de imagen: CCD de 1/2.5 pulgada.
Alcance de enfoque normal: 45 cm a infinito.
Visor LCD: Visor a color de 1.8 pulg.
AF: 9 puntos de AiAF o 1 punto AF (al centro).
Sensibilidad: Auto; ISO 80/100/200/400/800
Control balance a blanco: Auto, luz de día, nublado, tungsteno, fluorescente, fluorescente H y Personalizado.
Velocidad de obturador: 15-1/2000 seg.
Apertura: f/2.6-f/8.0 (W),
Modos de disparo: Auto, P, Tv, Av, M, Retrato, Paisaje, Escena Nocturna, Escena Especial (Follaje, Nieve, Playa, Fuegos Artificiales, Interiores, Niños y Mascotas, Nocturna, Acentuar Color, Intercambiar Color) Stitch Assist, Video (estándar, compacto, acentuar color, intercambiar color).
Velocidad de disparo continuo: Aprox. 2.1 disparos por segundo.
Tarjeta de Memoria: SD o Multimedia.
Formato de grabación de imagen: Imagen fija: JPEG, Video con audio: AVI.
Máxima Resolución: 2592 x 1944 píxeles.
My Colors: Colores Vivos, Neutrales, Sepia, Blanco y Negro, Diapositiva, Tono de piel más clara, Tono de piel más oscura y Color personalizado (Contraste, nitidez, saturación, rojo, verde, azul y tono de piel).
Impresión directa: Conexión directa a las impresoras fotográficas PIXMA, Selphy y PictBridge compatibles.
Interfases: USB y salida de Audio/Video .
Fuente de poder: 2 Baterías tamaño AA (alcalinas, recargables NiMH).
Dimensiones: 90.4 x 64.0 x 43.2 mm.
Peso (sin batería o tarjeta SD): Aprox. 170 grs.

Sensor de Imagen: CCD de 1/2.5 pulgadas (número total de píxeles 5.3 millones aproximadamente)

El sensor de imagen de la cámara es el dispositivo que permite capturar la imagen, está formado por fotodiodos que reciben el impacto de la luz y la transforman en píxeles (Punto de color, el conjunto de los cuales conforma una imagen. La resolución de imagen de una cámara digital se determina en cantidad de píxeles: 640 x 840 ó 1280 x 960, por ejemplo.

Cuantos más píxeles haya, mayor será la resolución de la imagen y, consecuentemente, también tendrá una mejor calidad), es decir, en información digital. A cada fotodiodo (Elemento sensible del sensor de imagen que capta el brillo de la luz para cada uno de los píxeles) le corresponde un píxel. El sensor de imagen es un mecanismo analógico, pero la información que captura la digitaliza un convertidor A/D (El convertidor A/D convierte la señal analógica captada por el sensor de imagen en señal digital, esto es, en píxeles) Hay diversos tipos de sensores de imagen, en función del fabricante. Sony, por ejemplo, ha utilizado en sus últimos modelos el CCD SuperHad, y Fujifilm, el SuperCCD.

Abertura: Término que se refiere a la abertura del diafragma, situado en el interior del objetivo de la cámara. El diafragma controla la cantidad de luz que alcanza al sensor de imagen, según sea su abertura mayor o menor. La

abertura puede establecerla automáticamente la cámara o puede controlarla el fotógrafo mediante los modos manual o el de prioridad a la apertura.

Una vez definidas las características de la cámara a utilizar, se tiene que especificar la resolución que más convenga, para nuestro caso tomamos fotografías con una resolución de 3 megapíxeles.

Para tomar las fotografías, las frutas se colocaron a una distancia de 12 cm. del lente de la cámara digital, utilizando un fondo blanco para evitar que la imagen resultante tuviera muchas sombras y, a la vez, resaltar más el objeto de interés de la imagen. Para tomar las fotografías no utilizamos el flash, ya que éste nos produciría más reflejos y sombras.

Para poder tomar la fotografía utilizamos la opción de enfoque automático con el cual cuenta la cámara, almacenando las imágenes en formato JPEG (*Joint Photographic Experts Group*) es el formato que se utiliza comúnmente para almacenar fotografías también se utiliza en documentos HTML para Internet, guarda toda la información referente al color en RGB.

También utiliza un sistema de compresión, que de forma eficiente reduce el tamaño de los archivos mediante la identificación y el descarte de los datos redundantes que no son esenciales para mostrar la imagen.

Las características de la computadora para poder procesar imágenes son mínimas, por ejemplo, deben de reunir pocos requisitos algunos de los más importantes:

- 1.- Una pantalla y una tarjeta gráfica de calidad.
- 2.- Memoria Ram, un mínimo de 128 MB.
- 3.- Un disco duro con capacidad de almacenamiento superior a los 30 GB o más.

Aquí se muestran las primeras imágenes obtenidas de las frutas, y en las cuáles basamos el tema a desarrollar en este proyecto.





Fig. 4.1 Frutas iniciales: Naranja, Manzana Amarilla, Pera y Manzana Roja.

Nota: las fotografías anteriores fueron obtenidas con otra cámara de mayor resolución y no con la que se describe en el apartado 4.1, pero sólo fue utilizada para tomar las imágenes que nos permitieron definir las características a estudiar, y para la obtención del clasificador si fue utilizada la que se menciona en esta tesis.

A partir de las imágenes anteriores definimos cuales son los objetos a analizar, en este caso, sólo elegimos tres de ellas:

Naranjas
Manzanas Amarillas
Manzanas Rojas

Y a cada una de estas frutas le corresponde una clase o grupo en el que debe de ser clasificado:

Clase 1: Naranjas
Clase 2: Manzanas Amarillas
Clase 3: Manzanas Rojas

Posteriormente, para poder hacer la separación en cada grupo, se deben de encontrar cuales son sus diferencias, y la vez, evitar sus semejanzas, vimos entonces que el color es una diferencia importante, pero principalmente entre las manzanas rojas y las otras dos frutas, por lo que debimos encontrar alguna característica que nos permitiera diferenciar entre las naranjas y las manzanas amarillas pues tienen un color más parecido, encontramos entonces que las naranjas difieren en forma a la de las manzanas. Tomando esto en cuenta decidimos optar por utilizar estas dos variables: la forma y el color.

Estas características serán utilizadas para construir el clasificador, el cual estuvo conformado inicialmente, por las variables de color y de forma, más adelante se explica la manera en que se obtuvieron sus respectivos valores.

4.3 Método utilizado.

Para poder hacer la diferenciación entre las distintas frutas, las características utilizadas son las siguientes ^[14]:

- Color
- Forma

1.3.1 Color

El color se obtuvo a través del programa Adobe Photoshop, primero seleccionamos el área de interés de la imagen (la fruta), y el programa descompone dicha sección en los tres colores primarios RGB que nos indican el valor de su histograma, de los cuales se utilizaron el Rojo y el Verde para poder discriminar de una manera adecuada ya que bibliográficamente se recomienda utilizar estos dos colores^[14] (el color azul, o B, no permite discriminar, ya que tiene valores muy inconstantes como se puede apreciar en las tablas del apéndice I), y mediante ellos obtener un coeficiente que llamamos C, el cual consiste en obtener el valor de la razón de G/R y que nos proporciona la cantidad de color.

$$C = \frac{G}{R}$$

Esto se realizó con cada una de las frutas obteniéndose así la razón de color.

Como sabemos la mayor parte de las cámaras tienen posibilidad de conectarse en forma directa a la computadora mediante un cable USB, este fue utilizado como interfaz entre la cámara y al computadora para realizar la transferencia.

Una vez transferidas las fotografías a la computadora se procedió a trabajar en la primera etapa, la cual consiste en determinar el valor de las componentes RGB (rojo, verde y azul) de las frutas para después poder trabajar y clasificarlas con estas características. Para esto utilizamos el programa Photoshop, el cual nos permite obtener un valor de los niveles de los que esta compuesta la fotografía en RGB.

Para determinar el valor de los componentes RGB con Photoshop, seleccionamos con el ratón un segmento de la imagen, los valores en dicho segmento corresponden a los puntos máximos del histograma.

Como se muestra en la figura siguiente, dentro del rectángulo rojo se pueden apreciar los valores de las componentes RGB, del pixel antes seleccionado:

Rojo = 171

Verde = 81

Azul = 70

dichos valores representan el nivel de color que contiene la imagen; el nivel que puede tomar cada color se distribuye en el rango de 0 a 255.

Los valores anteriores corresponden a la manzana roja mostrada en la figura siguiente.

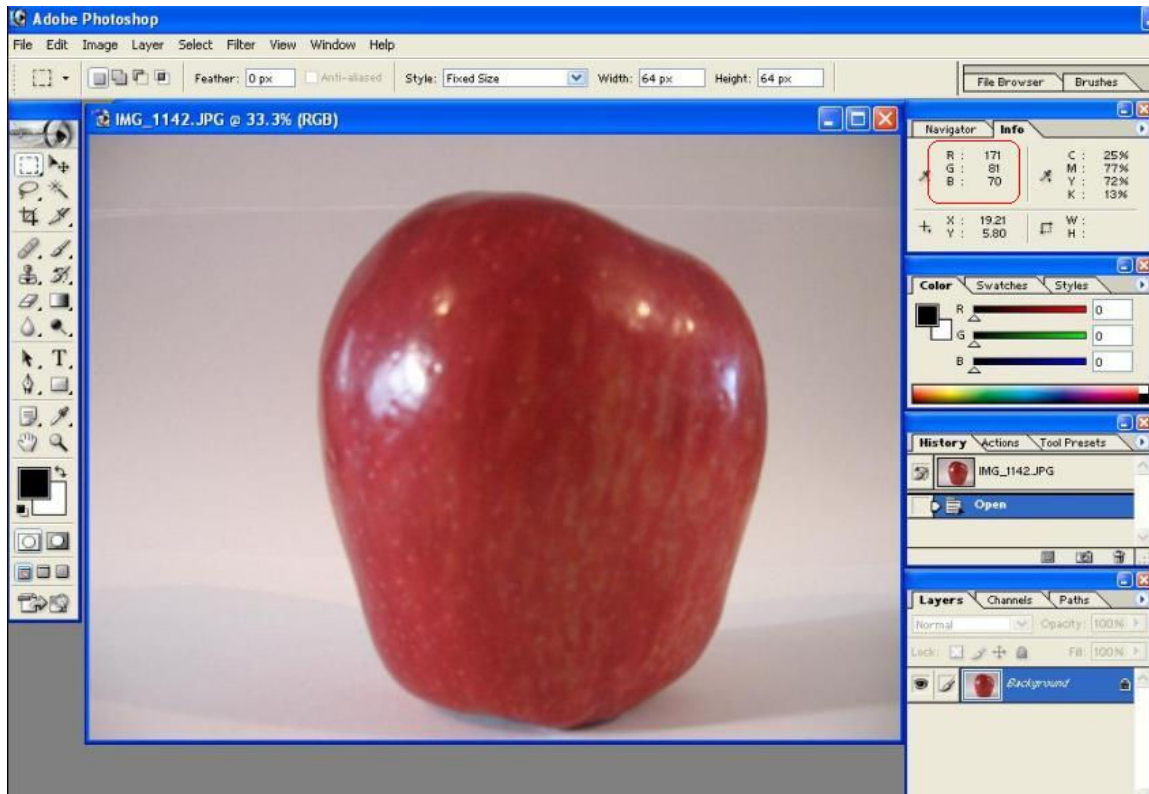


Fig. 4.2 Pantalla Photoshop: Manzana Roja

Esto se realizo para todas las fotografías de las frutas.

Para la imagen de la manzana amarilla, los valores son:

Rojo = 168

Verde = 150

Azul = 99

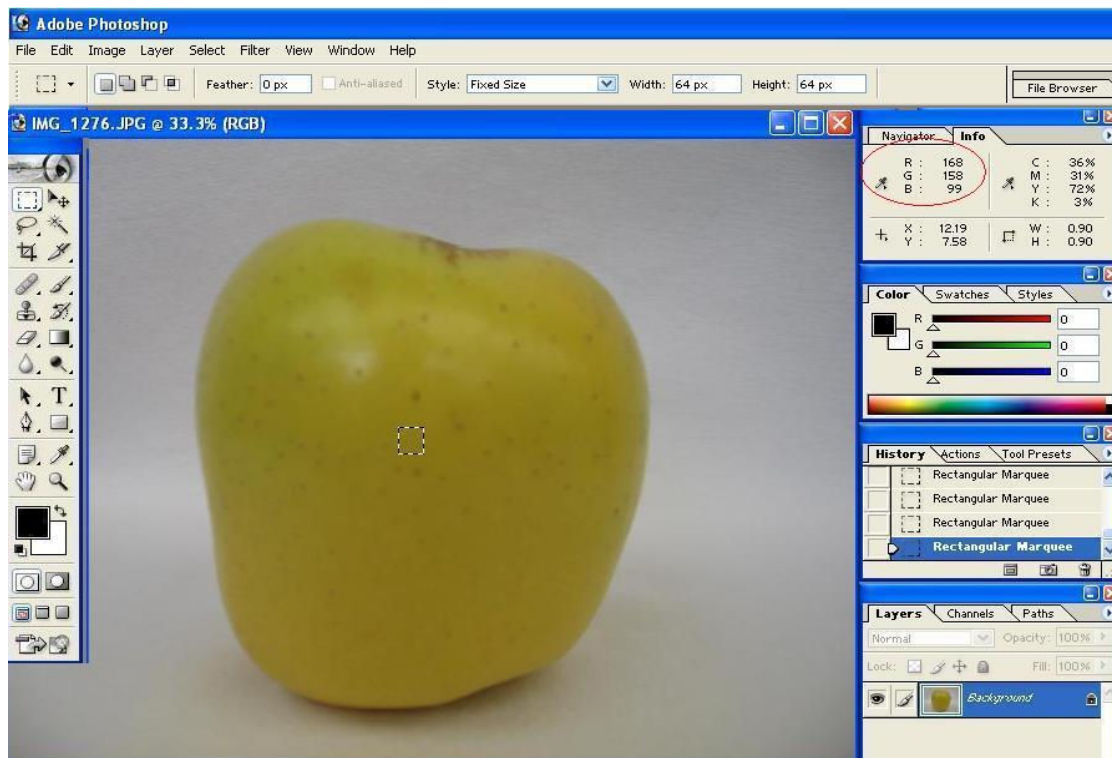


Fig. 4.3 Pantalla Photoshop: Manzana Amarilla

Y en el caso de la imagen de la naranja.

Rojo = 217

Verde = 170

Azul = 98

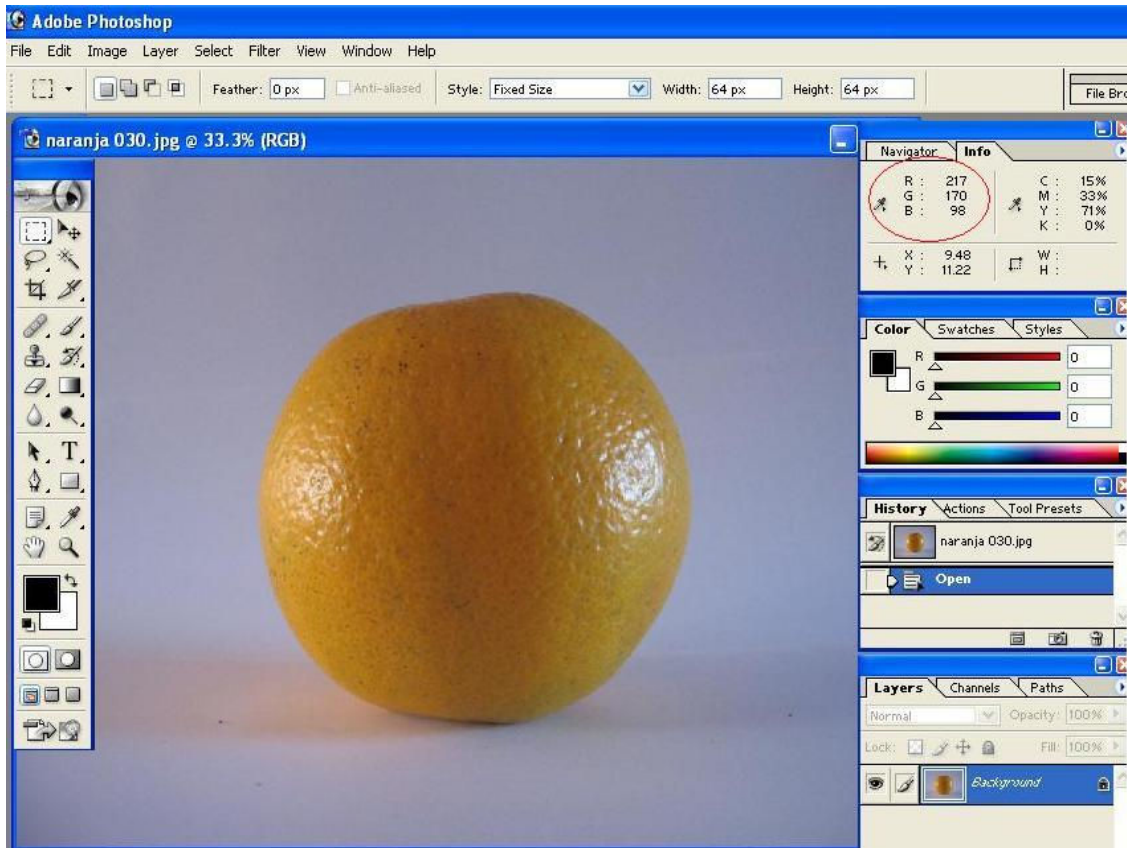


Fig. 4.4 Pantalla Photoshop: Naranja

De la discriminación de los tres colores mediante Photoshop se obtuvieron los resultados que se muestran en las siguientes tablas:

Datos obtenidos con Adobe Photoshop.

Manzanas Amarillas			
	R	G	B
1	175	155	32
2	170	155	51
3	170	155	60
4	169	155	58
5	170	155	54
6	167	155	55
7	168	158	61
8	171	158	54
9	167	161	65
10	168	160	98
11	171	149	50
12	170	146	60
13	171	148	54
14	172	158	59
15	171	158	54
16	169	158	66
17	169	153	58
18	168	157	67
19	170	161	58
20	169	156	54
21	172	162	77
22	168	156	54
23	169	151	49
24	168	153	68
25	169	149	52
26	170	154	77
27	170	153	49
28	168	155	51
29	169	148	59
30	172	150	49
\bar{X}	169.6666666667	154.7333333333	58.4333333333
σ^2	2.9885057471	17.2367816092	132.8057471264

R = Red (rojo)
 G = Green (verde)
 B = Blue (azul)

Tabla 4.1 Manzanas Amarillas

Donde \bar{X} es el promedio de la muestra y, σ^2 es la varianza

Manzanas Rojas			
	R	G	B
1	176	96	109
2	168	97	101
3	180	104	91
4	183	91	76
5	172	94	81
6	182	98	121
7	181	93	105
8	171	93	73
9	170	91	97
10	165	91	114
11	182	90	111
12	173	92	107
13	171	81	70
14	172	95	75
15	170	89	85
16	170	81	103
17	173	90	118
18	174	93	74
19	171	91	104
20	170	86	84
21	169	70	73
22	169	90	109
23	178	90	115
24	171	94	100
25	171	79	100
26	172	95	113
27	172	95	103
28	170	89	108
29	170	99	113
30	170	90	91
\bar{X}	172.8666666667	90.9000000000	97.4666666667
σ^2	21.2919540230	42.9206896552	242.6022988506

R = Red (rojo)
G = Green (verde)
B = Blue (azul)

Tabla 4.2 Manzanas Rojas

Naranjas			
	R	G	B
1	228	172	123
2	230	182	116
3	240	197	154
4	226	188	152
5	219	162	93
6	229	163	77
7	214	151	80
8	212	161	96
9	219	172	116
10	216	165	110
11	212	153	85
12	225	179	129
13	236	189	145
14	242	183	139
15	224	170	132
16	207	144	91
17	217	176	132
18	231	197	162
19	206	160	98
20	234	171	120
21	197	162	124
22	200	161	106
23	235	188	136
24	226	170	111
25	221	157	86
26	226	168	104
27	209	154	74
28	202	158	121
29	209	148	65
30	231	189	147
\bar{X}	220.7666666667	169.6666666667	114.1333333333
σ^2	145.9091954023	210.7126436782	673.9126436782

R = Red (rojo)
 G = Green (verde)
 B = Blue (azul)

Tabla 4.3 Naranjas

Nota:

De los datos anteriores (las varianzas), podemos ver que los valores correspondientes al color azul tienen una dispersión mayor, por lo que no son útiles para discriminar. En consecuencia sólo utilizamos las componentes del rojo y del verde.

Eliminamos el color azul y, sólo usamos el rojo y el verde, para nuestros cálculos del cociente, según la recomendación del libro “Pattern Recognition” del autor Marques de Sá ^[14].

A continuación se muestran las tablas con los cocientes calculados:

	Manzanas Amarillas C		
	R	G	G/R
1	175	155	0.8857142857
2	170	155	0.9117647059
3	170	155	0.9117647059
4	169	155	0.9171597633
5	170	155	0.9117647059
6	167	155	0.9281437126
7	168	158	0.9404761905
8	171	158	0.9239766082
9	167	161	0.9640718563
10	168	160	0.9523809524
11	171	149	0.8713450292
12	170	146	0.8588235294
13	171	148	0.8654970760
14	172	158	0.9186046512
15	171	158	0.9239766082
16	169	158	0.9349112426
17	169	153	0.9053254438
18	168	157	0.9345238095
19	170	161	0.9470588235
20	169	156	0.9230769231
21	172	162	0.9418604651
22	168	156	0.9285714286
23	169	151	0.8934911243
24	168	153	0.9107142857
25	169	149	0.8816568047
26	170	154	0.9058823529
27	170	153	0.9000000000
28	168	155	0.9226190476
29	169	148	0.8757396450
30	172	150	0.8720930233
\bar{X}	169.6666666667	154.7333333333	0.91210009
σ^2	2.9885057471	17.2367816092	0.0007298626

Tabla 4.4 C₁ Manzanas Amarillas

Manzanas Rojas C			
	R	G	G/R
1	176	96	0.5454545455
2	168	97	0.5773809524
3	180	104	0.5777777778
4	183	91	0.4972677596
5	172	94	0.5465116279
6	182	98	0.5384615385
7	181	93	0.5138121547
8	171	93	0.5438596491
9	170	91	0.5352941176
10	165	91	0.5515151515
11	182	90	0.4945054945
12	173	92	0.5317919075
13	171	81	0.4736842105
14	172	95	0.5523255814
15	170	89	0.5235294118
16	170	81	0.4764705882
17	173	90	0.5202312139
18	174	93	0.5344827586
19	171	91	0.5321637427
20	170	86	0.5058823529
21	169	70	0.4142011834
22	169	90	0.5325443787
23	178	90	0.5056179775
24	171	94	0.5497076023
25	171	79	0.4619883041
26	172	95	0.5523255814
27	172	95	0.5523255814
28	170	89	0.5235294118
29	170	99	0.5823529412
30	170	90	0.5294117647
\bar{X}	172.8666666667	90.9000000000	0.525880241
σ^2	21.2919540230	42.9206896552	0.0013091156

Tabla 4.5 C₁ Manzanas Rojas

Naranjas C			
	R	G	G/R
1	228	172	0.7543859649
2	230	182	0.7913043478
3	240	197	0.8208333333
4	226	188	0.8318584071
5	219	162	0.7397260274
6	229	163	0.7117903930
7	214	151	0.7056074766
8	212	161	0.7594339623
9	219	172	0.7853881279
10	216	165	0.7638888889
11	212	153	0.7216981132
12	225	179	0.7955555556
13	236	189	0.8008474576
14	242	183	0.7561983471
15	224	170	0.7589285714
16	207	144	0.6956521739
17	217	176	0.8110599078
18	231	197	0.8528138528
19	206	160	0.7766990291
20	234	171	0.7307692308
21	197	162	0.8223350254
22	200	161	0.8050000000
23	235	188	0.8000000000
24	226	170	0.7522123894
25	221	157	0.7104072398
26	226	168	0.7433628319
27	209	154	0.7368421053
28	202	158	0.7821782178
29	209	148	0.7081339713
30	231	189	0.8181818182
\bar{X}	220.7666666667	169.6666666667	0.76908766
σ^2	145.9091954023	210.7126436782	0.0018131541

Tabla 4.6 C₁ Naranjas

1.3.2 Forma

La otra variable que se menciona es la razón de alturas RA, la cual se obtuvo a través de un programa en Matlab, el cual consiste en obtener el contorno de la fruta, seleccionar su altura H y seleccionar una parte de esta altura, que denominamos con la letra X, y que consiste en la sección de la altura que va desde la parte más alta de ésta, hasta la más ancha (véase figura 4.5), ya que es un elemento que nos permite clasificar, dado que el ancho se encuentra a diferentes niveles en las frutas de distinta clase.

$$RA = X/H$$

Dicho programa determina, sobre el eje x, cual es el primer y cual es el último punto de la imagen, para obtener el nivel de altura en el que se encuentra el punto más ancho y así establecer el valor de X como lo muestra la figura 1:

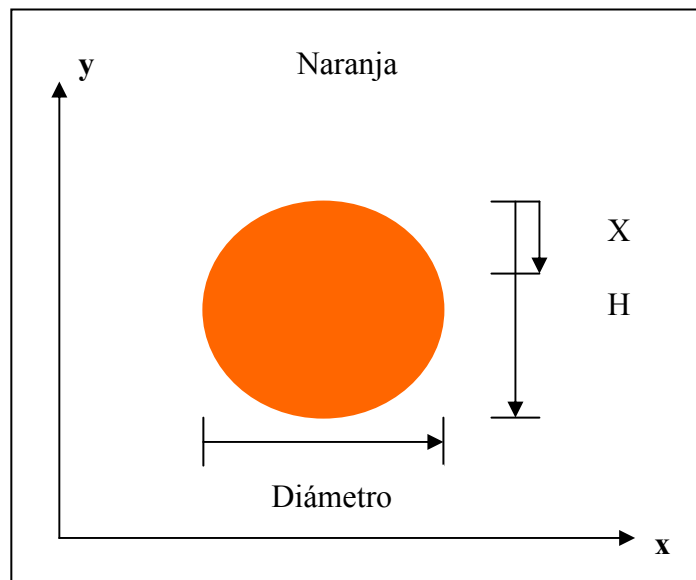


Fig. 4.5 Datos del contorno.

Para determinar las medidas de las frutas tenemos lo siguiente:

Manzanas Amarillas

	H	X	X/H
1	7.1	3.9	0.5492957746
2	6.9	3.6	0.5217391304
3	7.1	3.8	0.5352112676
4	6.8	3.6	0.5294117647
5	7	3.8	0.5428571429
6	7.1	3.8	0.5352112676
7	6.8	3.5	0.5147058824
8	6.8	3.6	0.5294117647
9	6.9	3.8	0.5507246377
10	6.8	3.8	0.5588235294
11	7	3.9	0.5571428571
12	6.7	3.9	0.5820895522
13	7.2	3.9	0.5416666667
14	7.2	4	0.5555555556
15	7.3	4.1	0.5616438356
16	7.3	4.1	0.5616438356
17	7.2	4	0.5555555556
18	7.1	3.9	0.5492957746
19	7.2	4.2	0.5833333333
20	7.2	4	0.5555555556
21	6.9	3.9	0.5652173913
22	7.1	3.9	0.5492957746
23	6.5	3.6	0.5538461538
24	7	3.9	0.5571428571
25	7	3.6	0.5142857143
26	6.3	3.5	0.5555555556
27	7.3	4	0.5479452055
28	6.8	3.6	0.5294117647
29	7	3.8	0.5428571429
30	6.9	3.8	0.5507246377
\bar{X}	6.9833333333	3.8266666667	0.5479052294

H = altura total.
X = valor de la altura al punto más ancho.
X/H = proporción de alturas.

Tabla 4.7 X/H Manzanas Amarillas

Manzanas Rojas

	H	X	X/H	
1	7.4	3.4	0.4594594595	H = altura total. X = valor de la altura al punto más ancho. X/H = proporción de alturas.
2	8.3	4	0.4819277108	
3	7.2	3.5	0.4861111111	
4	7.1	3.1	0.4366197183	
5	7.6	3.5	0.4605263158	
6	7.3	3.4	0.4657534247	
7	7.7	3.4	0.4415584416	
8	7.5	3.4	0.4533333333	
9	7.6	3.1	0.4078947368	
10	7.1	3.1	0.4366197183	
11	7	3.4	0.4857142857	
12	7.3	3.2	0.4383561644	
13	7.1	3	0.4225352113	
14	7	3.5	0.5000000000	
15	7.1	3.2	0.4507042254	
16	7.6	3.5	0.4605263158	
17	7.9	3.5	0.4430379747	
18	7.7	3.7	0.4805194805	
19	8.3	3.6	0.4337349398	
20	8.1	3.2	0.3950617284	
21	7.9	3.5	0.4430379747	
22	7.4	3.4	0.4594594595	
23	7.2	3.3	0.4583333333	
24	7.7	3.5	0.4545454545	
25	7.4	3.4	0.4594594595	
26	7.5	3.4	0.4533333333	
27	7.8	3.6	0.4615384615	
28	7.6	3.3	0.4342105263	
29	7.3	3.1	0.4246575342	
30	7.7	3.5	0.4545454545	
\bar{X}	7.5133333333	3.3900000000	0.4514371763	

Tabla 4.8 X/H Manzanas Rojas

Naranjas

	H	X	X/H	
1	7.5	3.9	0.5200000000	H = altura total. X = valor de la altura al punto más ancho. X/H = proporción de alturas.
2	7.6	3.6	0.4736842105	
3	7.5	3.3	0.4400000000	
4	8	3.7	0.4625000000	
5	7.8	4.3	0.5512820513	
6	8	3.8	0.4750000000	
7	7.8	3.8	0.4871794872	
8	7.7	4.1	0.5324675325	
9	7.9	3.8	0.4810126582	
10	7.8	3.9	0.5000000000	
11	7.9	3.9	0.4936708861	
12	7.7	4	0.5194805195	
13	7.9	3.9	0.4936708861	
14	8.1	4	0.4938271605	
15	8.1	4.1	0.5061728395	
16	8.3	4.1	0.4939759036	
17	8.4	4	0.4761904762	
18	8.4	3.9	0.4642857143	
19	7.9	4.2	0.5316455696	
20	8	4	0.5000000000	
21	7.8	4.1	0.5256410256	
22	7.6	4	0.5263157895	
23	7.7	4.1	0.5324675325	
24	7.5	3.9	0.5200000000	
25	8	4	0.5000000000	
26	7.9	3.9	0.4936708861	
27	7.8	4.1	0.5256410256	
28	7.5	3.6	0.4800000000	
29	7.6	3.4	0.4473684211	
30	7.7	3.7	0.4805194805	
\bar{X}	7.846666667	3.903333333	0.4975890019	

Tabla 4.9 X/H Naranja

Nota:

La variable X/H no muestra una variación importante entre las tres clases.

4.4 Datos escogidos, número de clases y variables discriminatorias.

Una vez obtenidos los datos del color y de las características físicas se graficaron estas proporciones dando como resultado lo siguiente:

Gráfica de los datos.

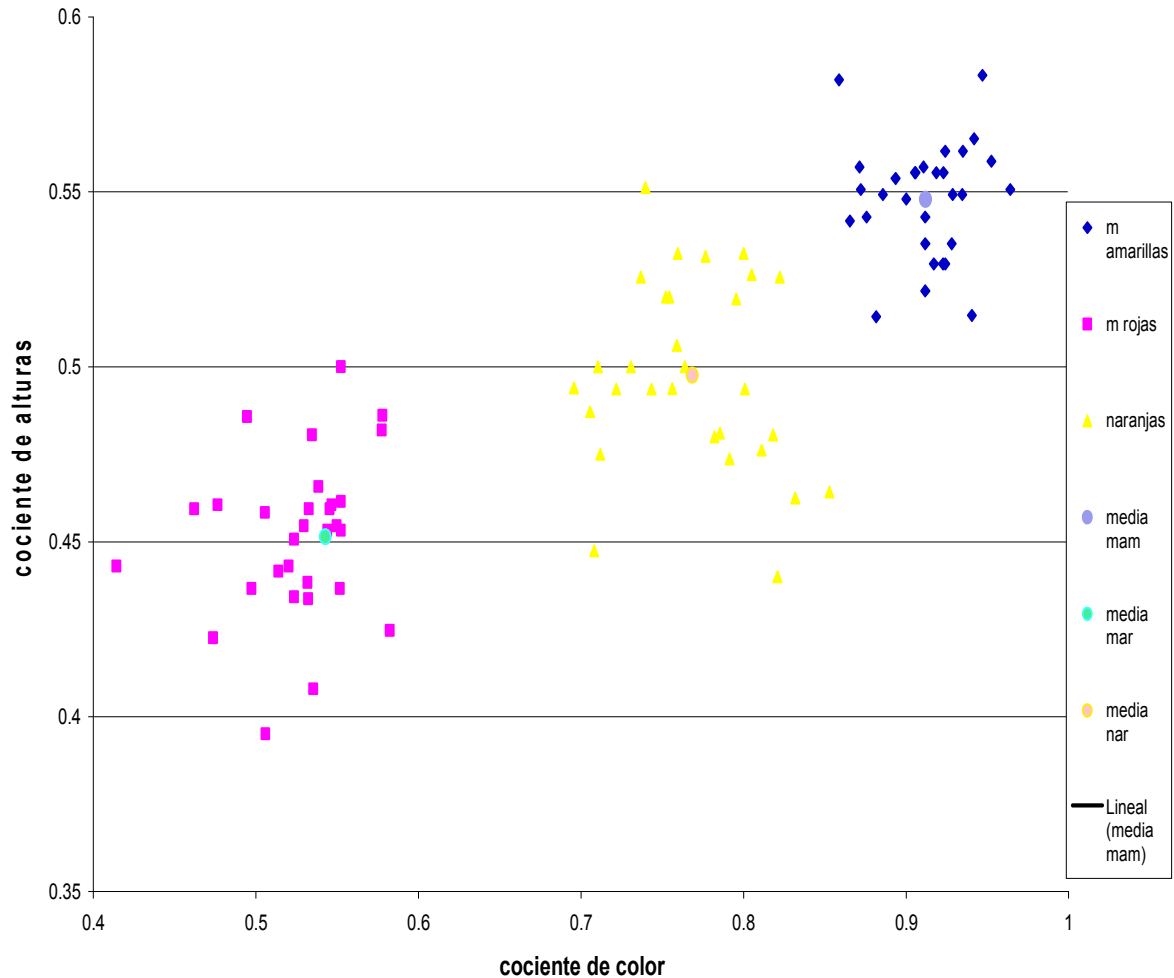


Fig. 4.6 Gráfica de los Datos

El número de clases que se analizaron son tres: manzanas amarillas, manzanas rojas y naranjas, y de éstas se tomaron 30 muestras por cada clase. En la gráfica anterior, de izquierda a derecha se encuentran agrupadas las manzanas rojas, las naranjas y, por último, las manzanas amarillas.

4.5 Clasificador K-Medias.

La primer técnica que utilizamos para ver el agrupamiento de nuestras tres clases y poder hacer la clasificación mediante las características del color y la forma fueron las K-Medias, como ya se explico en el capítulo 2 en que consiste esta técnica mostraremos los resultados que se obtuvieron.

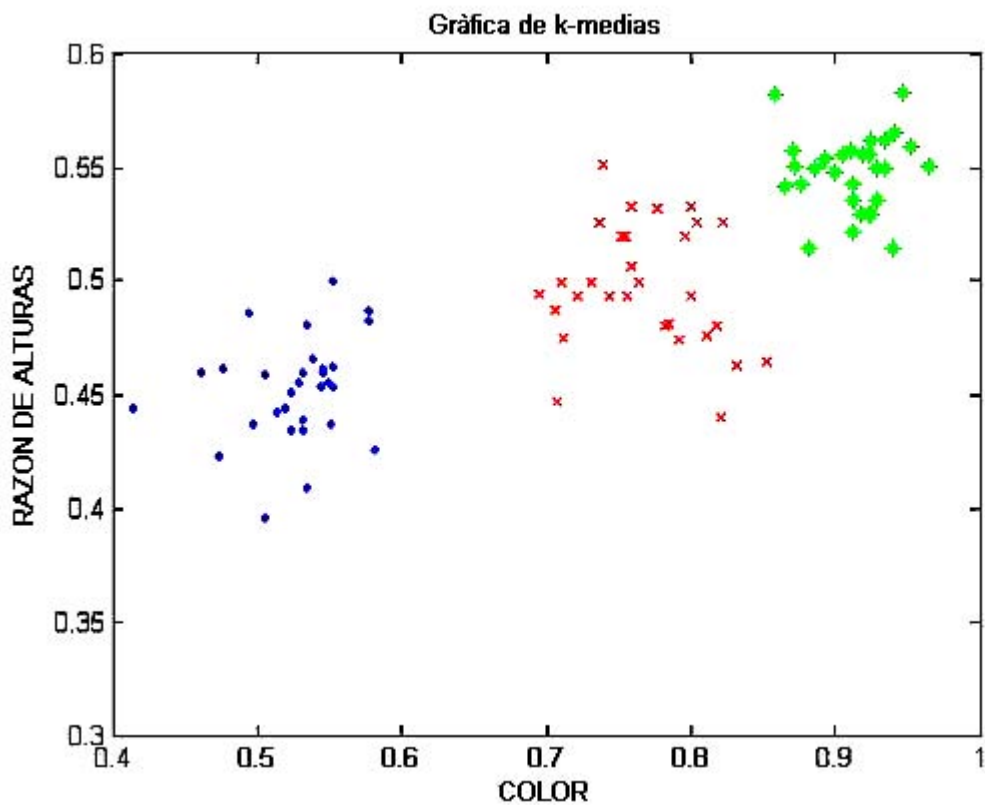


Fig. 4.7 Gráfica de los K-medias

A continuación se muestra una tabla con los datos agrupados en tres clases, con el algoritmo de las K-Medias.

La C representa a la variable de color y la H la razón de alturas.

clase 1		clase 2		clase 3	
C	H	C	H	C	H
0.54545	0.45946	0.75439	0.52	0.88571	0.5493
0.57738	0.48193	0.7913	0.47368	0.91176	0.52174
0.57778	0.48611	0.82083	0.44	0.91176	0.53521
0.49727	0.43662	0.83186	0.4625	0.91716	0.52941
0.54651	0.46053	0.73973	0.55128	0.91176	0.54286
0.53846	0.46575	0.71179	0.475	0.92814	0.53521
0.51381	0.44156	0.70561	0.48718	0.94048	0.51471
0.54386	0.45333	0.75943	0.53247	0.92398	0.52941
0.53529	0.40789	0.78539	0.48101	0.96407	0.55072
0.55152	0.43662	0.76389	0.5	0.95238	0.55882
0.49451	0.48571	0.7217	0.49367	0.87135	0.55714
0.53179	0.43836	0.79556	0.51948	0.85882	0.58209
0.47368	0.42254	0.80085	0.49367	0.8655	0.54167
0.55233	0.5	0.7562	0.49383	0.9186	0.55556
0.52353	0.4507	0.75893	0.50617	0.92398	0.56164
0.47647	0.46053	0.69565	0.49398	0.93491	0.56164
0.52023	0.44304	0.81106	0.47619	0.90533	0.55556
0.53448	0.48052	0.85281	0.46429	0.93452	0.5493
0.53216	0.43373	0.7767	0.53165	0.94706	0.58333
0.50588	0.39506	0.73077	0.5	0.92308	0.55556
0.4142	0.44304	0.82234	0.52564	0.94186	0.56522
0.53254	0.45946	0.805	0.52632	0.92857	0.5493
0.50562	0.45833	0.8	0.53247	0.89349	0.55385
0.54971	0.45455	0.75221	0.52	0.91071	0.55714
0.46199	0.45946	0.71041	0.5	0.88166	0.51429
0.55233	0.45333	0.74336	0.49367	0.90588	0.55556
0.55233	0.46154	0.73684	0.52564	0.9	0.54795
0.52353	0.43421	0.78218	0.48	0.92262	0.52941
0.58235	0.42466	0.70813	0.44737	0.87574	0.54286
0.52941	0.45455	0.81818	0.48052	0.87209	0.55072

Tabla 4.10 Datos clasificados con K-Medias

Los datos que son los centros de masa de cada una de las tres clases son:

	Centros de masa	
clase 1	0.52588	0.45144
clase 2	0.7681	0.49759
clase 3	0.9121	0.54791

Tabla 4.11 Centros de masa de cada clase

El código desarrollado en Matlab, se encuentra en el Apéndice II.

4.6 Clasificador de Bayes.

Haciendo un análisis más profundo de la gráfica, vimos que:

- Las clases se encuentran alineadas.
- Con respecto al eje de la Razón de Alturas se sobreponen.
- Con respecto al eje de la variable de Color, están separadas.

De lo anterior podemos concluir que la variable de la Razón de Alturas no permitió discriminar, pues está altamente correlacionada, por lo cual no es necesaria para realizar el clasificador, entonces, para calcular el clasificador de Bayes sólo utilizaremos la variable del Color, pues el rango de valores que toma es independiente para cada una de las clases, como se puede apreciar en la siguiente tabla.

Manzanas Amarillas	Mínimo	0.8588
	Máximo	0.9641

Naranjas	Mínimo	0.8588
	Máximo	0.9641

Manzanas Rojas	Mínimo	0.4142
	Máximo	0.5824

Tabla 4.12 Valores máximos y mínimos del cociente de color para las tres clases.

Considerando lo anterior, obtuvimos el clasificador de Bayes con una sola variable, por lo que nuestros datos quedan como sigue:

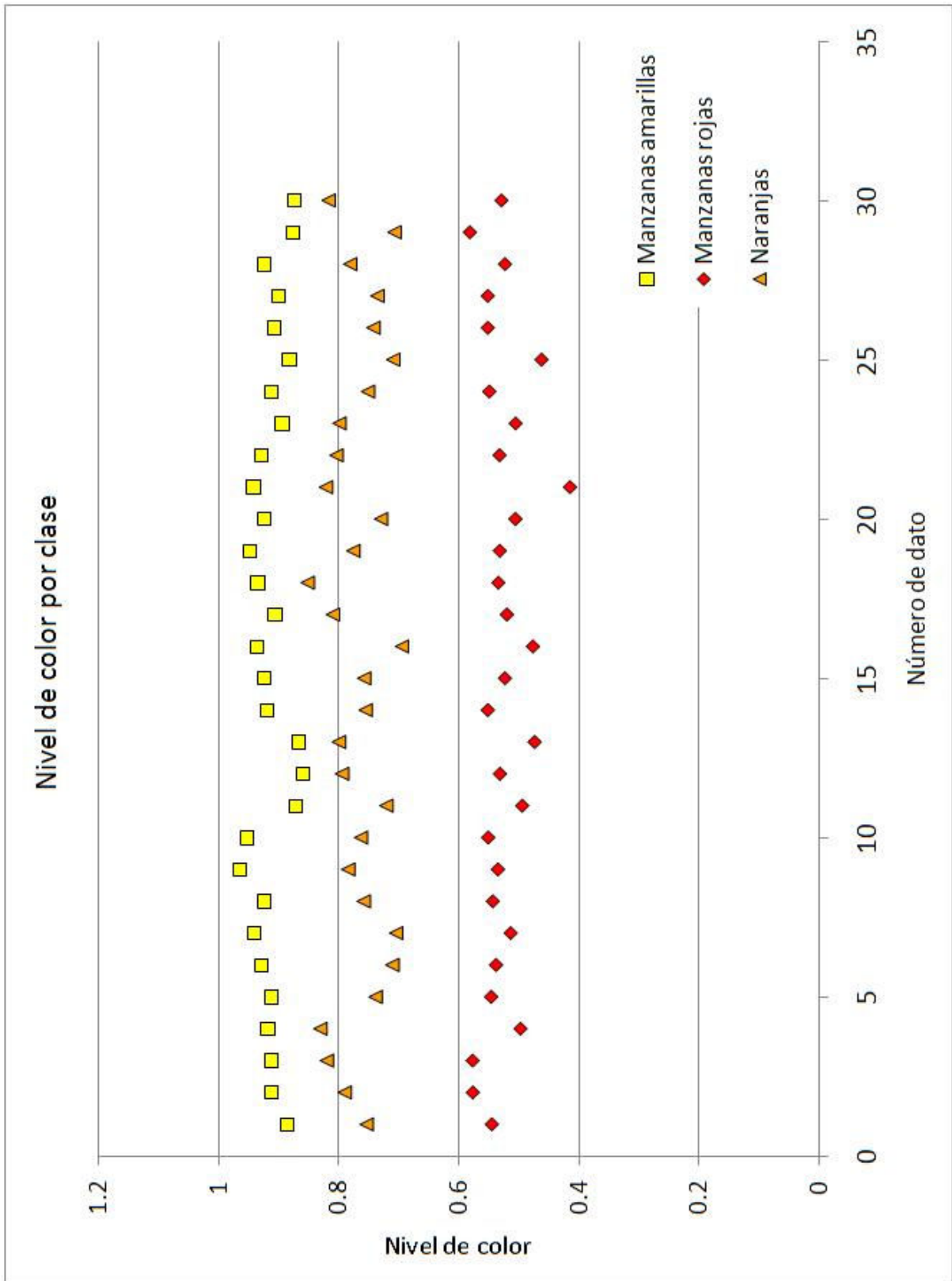


Fig. 4.9 Datos usando solamente la variable de color

Para poder aplicar Bayes, suponemos que los datos tienen una distribución normal o gaussiana.

La información estadística de los datos se muestra en la siguiente tabla:

Manzanas Amarillas	<i>media</i>	$\hat{\mu}$	0.9121
	<i>varianza</i>	$\hat{\sigma}^2$	0.0007
	<i>desviación</i>	$\hat{\sigma}$	0.027
Naranjas	<i>media</i>	$\hat{\mu}$	0.7681
	<i>varianza</i>	$\hat{\sigma}^2$	0.0018
	<i>desviación</i>	$\hat{\sigma}$	0.0426
Manzanas Rojas	<i>media</i>	$\hat{\mu}$	0.5259
	<i>varianza</i>	$\hat{\sigma}^2$	0.0013
	<i>desviación</i>	$\hat{\sigma}$	0.0362

Tabla 4.13 Valores de los parámetros estadísticos de las tres clases.

Y, a partir de la función de distribución gaussiana G:

$$G = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

Obtenemos la distribución gaussiana para cada una de las clases.

- Manzanas Amarillas

$$G_{mam} = \frac{1}{(0.027)\sqrt{2\pi}} \exp\left[-\frac{(x-0.9121)^2}{2(0.0007)}\right]$$

- Naranjas

$$G_{nar} = \frac{1}{(0.0426)\sqrt{2\pi}} \exp\left[-\frac{(x-0.7681)^2}{2(0.0018)}\right]$$

- Manzanas Rojas

$$G_{mro} = \frac{1}{(0.0362)\sqrt{2\pi}} \exp\left[-\frac{(x-0.5259)^2}{2(0.0013)}\right]$$

Las cuales se pueden observar en la siguiente gráfica:

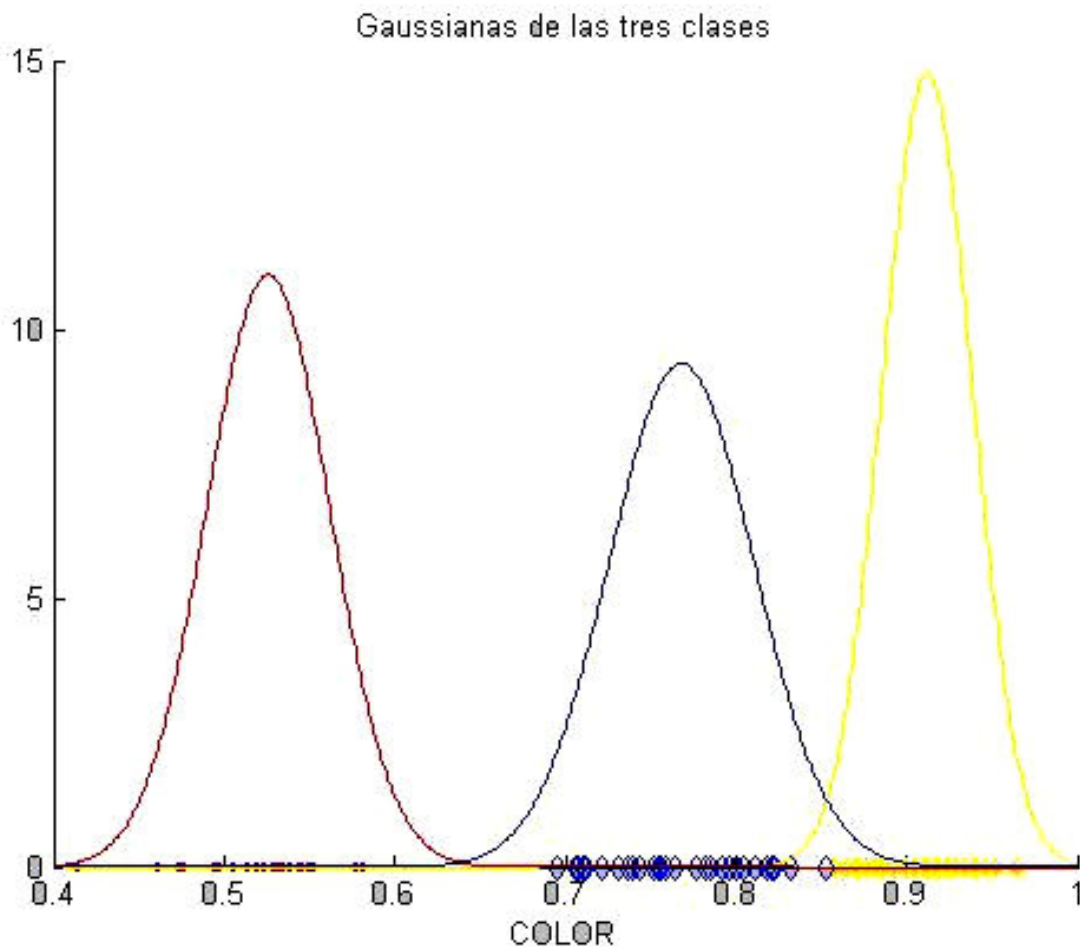


Fig. 4.10 Gaussianas de las tres clases con una sola variable

En rojo se encuentran representadas las manzanas rojas; en azul, las naranjas; y en amarillo, las manzanas amarillas.

Para poder separar las clases, lo que tuvimos que hacer es encontrar los valores, en el eje del color, en el que las funciones se intersecan, si de acuerdo a la gráfica anterior, nuestras clases están ordenadas como siguen:

- Clase 1: Manzanas Rojas
- Clase 2: Naranjas
- Clase 3: Manzanas Amarillas

Las fronteras de decisión están dadas por los valores:

- $\lambda_{12} = 0.6382$ entre la clase 1 y 2.
- $\lambda_{23} = 0.8533$ entre la clase 2 y 3.

Al trazar líneas perpendiculares sobre estos puntos, podemos observar la separación de las clases como sigue:

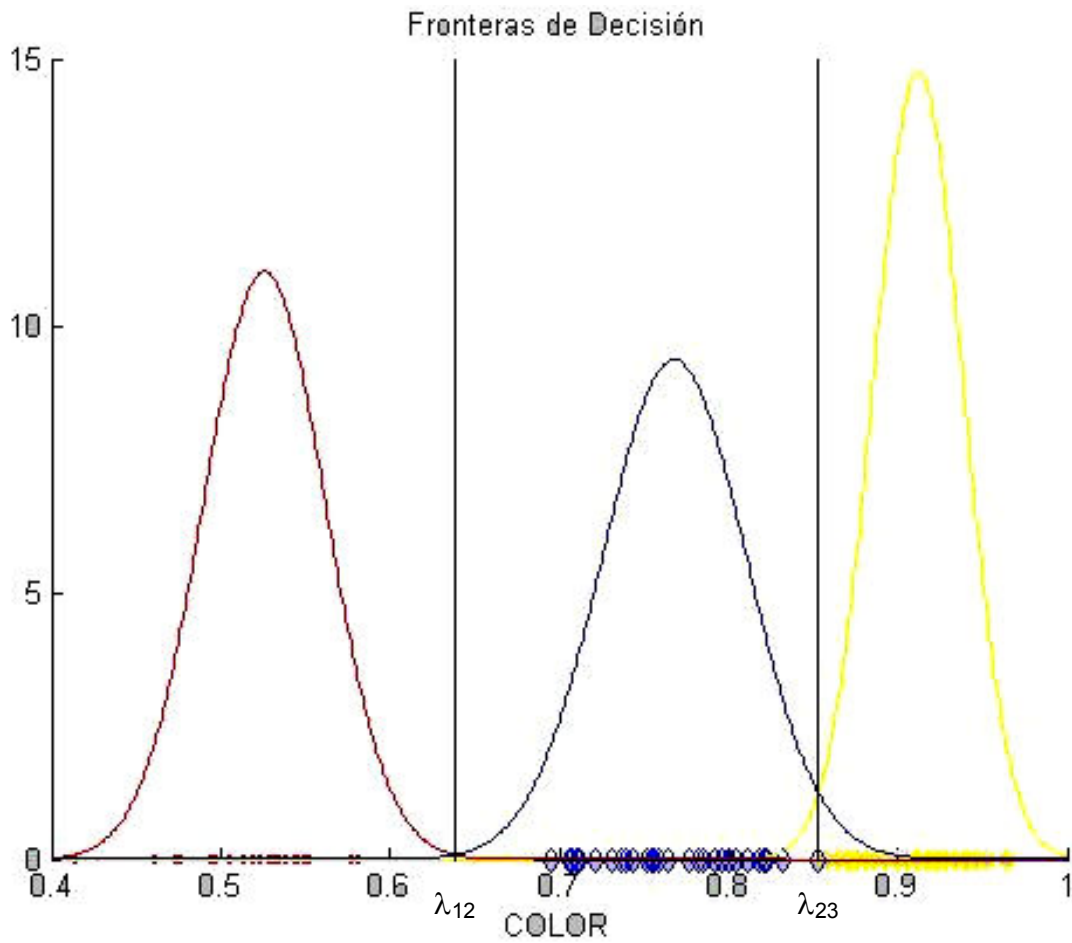


Fig. 4.11 Fronteras de las tres clases

Por lo que las reglas de decisión para cada clase quedaron definidas por los intervalos siguientes:

- Clase 1 (Manzanas Rojas): (0.4 a 0.6379)
- Clase 2 (Naranjas): (0.638 a 0.8547)
- Clase 3 (Manzanas Amarillas): (0.8547 a 1.0)

Si $0.4 < \text{color} < 0.6379$ asignar a la clase 1.
 Si $0.638 < \text{color} < 0.8547$ asignar a la clase 2.
 Si $0.8547 < \text{color} < 1.0$ asignar a la clase 3.

CAPÍTULO 5

Conclusiones



En esta tesis se logró aplicar algunas de las técnicas y métodos que se aprendieron durante el transcurso de la carrera, centrándonos en un área que actualmente está en auge, el reconocimiento de patrones a través de imágenes digitales, pues como sabemos, gracias a la nueva tecnología digital es más fácil adquirir y manejar la información, que en nuestro caso son imágenes digitales (fotos de frutas).

El objetivo principal de nuestra tesis fue hacer un programa que nos facilitara la clasificación de frutas, inicialmente determinamos dos de sus características: el color y su forma; entonces aplicamos el método de las k-medias, utilizando esas dos variables, lo que permitió agrupar los datos obtenidos de tal manera que fueran separados en las tres diferentes clases predeterminadas.

Pero haciendo un análisis de los datos nos dimos cuenta que sólo era necesario trabajar con el color, ya que la variable de la forma presentó valores muy parecidos, es decir, nos enfrentamos al problema de que todos tienen forma similar (ver tablas 4.7, 4.8 y 4.9) y como consecuencia los datos para esta variable no permitieron tener una forma para distinguir una fruta de otra, lo cual nos llevó a trabajar únicamente con el color.

Una vez detectado el inconveniente y la decisión de sólo utilizar una variable discriminante, aplicamos el teorema de Bayes, al considerar que los datos se comportaron según la distribución normal, de esta manera fue como obtuvimos las fronteras de decisión para poder clasificar las tres frutas.

Con base en este trabajo de tesis, podemos decir que el color es una característica muy útil para diferenciar objetos y que sí logramos nuestro objetivo principal, que fue obtener un programa que permitiera clasificar objetos, en nuestro caso: frutas.

Su aplicación en la industria es una de tantas opciones factibles de utilización de este tipo de programas, pues el poder separar objetos con una imagen, mediante un programa y hardware específicos, optimizará el proceso de separación.

La prospectiva de aplicación de este trabajo, se necesitan mejorar las condiciones en que se obtienen las imágenes como la calidad de la iluminación y los dispositivos para procesar las imágenes, así también una resolución adecuada de la cámara, redundaría en mejores resultados.

Al desarrollar todo este trabajo y lograr hacer la clasificación de las frutas mediante fotos gracias a las técnicas antes mencionadas, quedamos muy satisfechos, pues logramos aportar algo práctico dentro de este campo de estudio y a la investigación en la Universidad.

Apéndices



APENDICE I Datos obtenidos.

Manzanas Amarillas

	R	G	B	G/R	H	X	X/H
1	175	155	32	0.88571429	7.1	3.9	0.54929577
2	170	155	51	0.91176471	6.9	3.6	0.52173913
3	170	155	60	0.91176471	7.1	3.8	0.53521127
4	169	155	58	0.91715976	6.8	3.6	0.52941176
5	170	155	54	0.91176471	7	3.8	0.54285714
6	167	155	55	0.92814371	7.1	3.8	0.53521127
7	168	158	61	0.94047619	6.8	3.5	0.51470588
8	171	158	54	0.92397661	6.8	3.6	0.52941176
9	167	161	65	0.96407186	6.9	3.8	0.55072464
10	168	160	98	0.95238095	6.8	3.8	0.55882353
11	171	149	50	0.87134503	7	3.9	0.55714286
12	170	146	60	0.85882353	6.7	3.9	0.58208955
13	171	148	54	0.86549708	7.2	3.9	0.54166667
14	172	158	59	0.91860465	7.2	4	0.55555556
15	171	158	54	0.92397661	7.3	4.1	0.56164384
16	169	158	66	0.93491124	7.3	4.1	0.56164384
17	169	153	58	0.90532544	7.2	4	0.55555556
18	168	157	67	0.93452381	7.1	3.9	0.54929577
19	170	161	58	0.94705882	7.2	4.2	0.58333333
20	169	156	54	0.92307692	7.2	4	0.55555556
21	172	162	77	0.94186047	6.9	3.9	0.56521739
22	168	156	54	0.92857143	7.1	3.9	0.54929577
23	169	151	49	0.89349112	6.5	3.6	0.55384615
24	168	153	68	0.91071429	7	3.9	0.55714286
25	169	149	52	0.8816568	7	3.6	0.51428571
26	170	154	77	0.90588235	6.3	3.5	0.55555556
27	170	153	49	0.9	7.3	4	0.54794521
28	168	155	51	0.92261905	6.8	3.6	0.52941176
29	169	148	59	0.87573964	7	3.8	0.54285714
30	172	150	49	0.87209302	6.9	3.8	0.55072464
	169.666667	154.733333	58.433333	0.91209963	6.98333333	3.82666667	0.54790523

Los valores de la columna **G/R** corresponden a los valores de la variable C, y los valores de la columna **X/H** corresponden a los valores de la variable RA, para las manzanas amarillas.

Manzanas Rojas

	R	G	B	G/R	H	X	X/H
1	176	96	109	0.54545455	7.4	3.4	0.45945946
2	168	97	101	0.57738095	8.3	4	0.48192771
3	180	104	91	0.57777778	7.2	3.5	0.48611111
4	183	91	76	0.49726776	7.1	3.1	0.43661972
5	172	94	81	0.54651163	7.6	3.5	0.46052632
6	182	98	121	0.53846154	7.3	3.4	0.46575342
7	181	93	105	0.51381215	7.7	3.4	0.44155844
8	171	93	73	0.54385965	7.5	3.4	0.45333333
9	170	91	97	0.53529412	7.6	3.1	0.40789474
10	165	91	114	0.55151515	7.1	3.1	0.43661972
11	182	90	111	0.49450549	7	3.4	0.48571429
12	173	92	107	0.53179191	7.3	3.2	0.43835616
13	171	81	70	0.47368421	7.1	3	0.42253521
14	172	95	75	0.55232558	7	3.5	0.5
15	170	89	85	0.52352941	7.1	3.2	0.45070423
16	170	81	103	0.47647059	7.6	3.5	0.46052632
17	173	90	118	0.52023121	7.9	3.5	0.44303797
18	174	93	74	0.53448276	7.7	3.7	0.48051948
19	171	91	104	0.53216374	8.3	3.6	0.43373494
20	170	86	84	0.50588235	8.1	3.2	0.39506173
21	169	70	73	0.41420118	7.9	3.5	0.44303797
22	169	90	109	0.53254438	7.4	3.4	0.45945946
23	178	90	115	0.50561798	7.2	3.3	0.45833333
24	171	94	100	0.5497076	7.7	3.5	0.45454545
25	171	79	100	0.4619883	7.4	3.4	0.45945946
26	172	95	113	0.55232558	7.5	3.4	0.45333333
27	172	95	103	0.55232558	7.8	3.6	0.46153846
28	170	89	108	0.52352941	7.6	3.3	0.43421053
29	170	99	113	0.58235294	7.3	3.1	0.42465753
30	170	90	91	0.52941176	7.7	3.5	0.45454545
	172.866667	90.9	97.466667	0.54273353	7.51333333	3.39	0.45143718

Los valores de la columna **G/R** corresponden a los valores de la variable C, y los valores de la columna **X/H** corresponden a los valores de la variable RA, para las manzanas rojas.

Naranjas

	R	G	B	G/R	H	X	X/H
1	228	172	123	0.75438596	7.5	3.9	0.52
2	230	182	116	0.79130435	7.6	3.6	0.47368421
3	240	197	154	0.82083333	7.5	3.3	0.44
4	226	188	152	0.83185841	8	3.7	0.4625
5	219	162	93	0.73972603	7.8	4.3	0.55128205
6	229	163	77	0.71179039	8	3.8	0.475
7	214	151	80	0.70560748	7.8	3.8	0.48717949
8	212	161	96	0.75943396	7.7	4.1	0.53246753
9	219	172	116	0.78538813	7.9	3.8	0.48101266
10	216	165	110	0.76388889	7.8	3.9	0.5
11	212	153	85	0.72169811	7.9	3.9	0.49367089
12	225	179	129	0.79555556	7.7	4	0.51948052
13	236	189	145	0.80084746	7.9	3.9	0.49367089
14	242	183	139	0.75619835	8.1	4	0.49382716
15	224	170	132	0.75892857	8.1	4.1	0.50617284
16	207	144	91	0.69565217	8.3	4.1	0.4939759
17	217	176	132	0.81105991	8.4	4	0.47619048
18	231	197	162	0.85281385	8.4	3.9	0.46428571
19	206	160	98	0.77669903	7.9	4.2	0.53164557
20	234	171	120	0.73076923	8	4	0.5
21	197	162	124	0.82233503	7.8	4.1	0.52564103
22	200	161	106	0.805	7.6	4	0.52631579
23	235	188	136	0.8	7.7	4.1	0.53246753
24	226	170	111	0.75221239	7.5	3.9	0.52
25	221	157	86	0.71040724	8	4	0.5
26	226	168	104	0.74336283	7.9	3.9	0.49367089
27	209	154	74	0.73684211	7.8	4.1	0.52564103
28	202	158	121	0.78217822	7.5	3.6	0.48
29	209	148	65	0.70813397	7.6	3.4	0.44736842
30	231	189	147	0.81818182	7.7	3.7	0.48051948
	220.517241	169.586207	113.827586	0.76903831	7.846666667	3.903333333	0.497589

Los valores de la columna **G/R** corresponden a los valores de la variable C, y los valores de la columna **X/H** corresponden a los valores de la variable RA, para las naranjas.

APENDICE II

a) Sobre el programa de desarrollo: Matlab.

Matlab es la abreviatura de Matrix Laboratory (laboratorio de matrices) ^[15]. Es un programa de análisis numérico creado por The MathWorks en 1984. Está disponible para las plataformas Unix, Windows y Mac OS X.

La primera versión surgió con la idea de emplear unos paquetes de subrutinas escritas en Fortran en los cursos de álgebra lineal y análisis numérico, sin necesidad de escribir programas en Fortran.

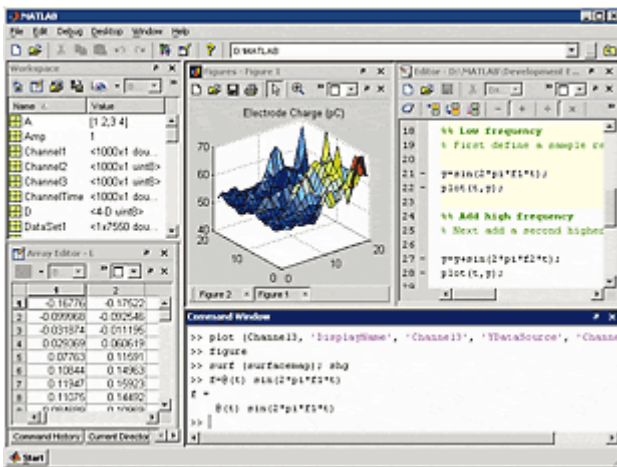


Fig. A.1 Ejemplo de una aplicación en Matlab

Usa un lenguaje de programación creado en 1970 para proporcionar un sencillo acceso al software de matrices LINPACK y EISPACK sin tener que usar Fortran. También tiene su propio compilador.

Matlab es un lenguaje de computación y de ambientes interactivos de alto nivel para desarrollo de algoritmos, análisis numérico, visualización y análisis de datos ^[16]. Permite la solución

de problemas técnicos de cómputo más rápido que con lenguajes tradicionales como C, C++ o FORTRAN.

Matlab puede ser utilizado en una variada gama de aplicaciones incluyendo el procesamiento de señales e imágenes, comunicaciones, para el diseño, prueba y medición de control, modelado y análisis financiero y cálculos biológicos. Además cuenta con "Toolboxes" o colecciones de funciones de propósitos especiales, las cuales amplían el ambiente de Matlab para resolver clases particulares de problemas en las áreas antes mencionadas ^[17]. También cuenta con otras herramientas como Simulink, que permite la simulación de sistemas y contiene una extensa librería, que está compuesta de algoritmos computacionales que van desde los comandos sencillos como: "sum, sine, cosine", hasta funciones de aritmética compleja, y funciones sofisticadas como la inversa de una matriz, los valores característicos, las funciones de Bessel, las transformadas rápidas de Fourier entre muchas otras.

Es un software muy usado en universidades, centros de investigación y por ingenieros. En los últimos años ha incluido muchas más capacidades, como la

de programar directamente procesadores digitales de señales, crear código VHDL y otras.

Matlab es un programa de cálculo numérico, orientado a matrices y vectores, por tanto, desde el principio hay que pensar que todo lo que se pretenda hacer con el, será mucho más rápido y efectivo si se piensa en términos de matrices y vectores [8].

MODOS

a) **Explicación:** dada una imagen digital, indican qué cantidad de colores puede presentar como máximo una imagen, la riqueza de color que tiene. Para cada píxel tiene un número de colores, así se puede tener una imagen en:

Mapa de bits: es la menor gama; sólo hay dos colores para representar un píxel, blanco o negro. La profundidad de color será de 1 bit.

Escala de grises: en este caso se tienen 256 niveles (matices) de gris, con una profundidad de color de 8 bit.

Color indexado o paleta de color: menor tamaño y calidad, el mínimo valor será 0 y el máximo 256. Tiene la misma profundidad de color que el anterior.

Color real: es multicanal; son tres imágenes superpuestas (R, G, B) cada una de 8 bits. Así se tienen dos posibilidades: 16 bit que determinan 64000 colores o 24 bits que determinan 16 millones de colores.

Nota: la profundidad de color es el número de bits necesarios para representar el color.

b) **Representación en Matlab:** MATLAB almacena la mayoría de las imágenes como arrays bidimensionales (matrices), donde cada elemento de la matriz corresponde a un píxel de la imagen. Una imagen compuesta de 200 por 300 píxeles podría almacenarse en MATLAB como una matriz de 200 por 300 elementos.

Trabajar en MATLAB con imágenes es similar a trabajar con cualquier otro tipo de matriz de datos. Puede seleccionarse un píxel determinado en una imagen mediante la notación matricial:

`I(4, 10)`. Devuelve el píxel que se encuentra en la fila 4, columna 10 de la imagen 1.

Tipos de datos.

Por defecto, MATLAB almacena la mayoría de los datos en arrays de clase `double` (números en coma flotante con doble precisión -64 bits-). Todas las funciones de MATLAB pueden trabajar con estos vectores.

Para el procesamiento de imágenes, esta representación no es siempre adecuada, ya que el número de píxeles en una imagen puede ser muy grande y este tipo de datos supone ocupar gran cantidad de memoria. Para reducir los tamaños de almacenamiento, MATLAB soporta almacenar las imágenes en matrices de datos de clase `uint8` (enteros sin signo de 8 bits).

Esta diferencia en los tipos de datos hace que:

- Los valores no enteros se puedan almacenar únicamente en matrices de tipo *double*.

- No todas las funciones pueden operar con vectores de tipos *uint8*.

b1) Imágenes indexadas (Paleta de color).

Está formada por dos matrices, una matriz de imagen y un mapa de color.

- El mapa de color es un conjunto ordenado de valores que representa los colores que hay en la imagen.

- La matriz de la imagen contiene, para cada píxel, un valor que es un índice en el mapa de color.

El mapa de color.

- Es una matriz de m-por-3 elementos de clase *double*.
- Cada fila de la matriz del mapa de color especifica los valores rojo, verde y azul, (RGB) para un color simple.
- Los valores R, G y B son escalares reales que pueden tomar valores desde el 0 (negro) al 1.0 (blanco correspondiente a la intensidad total).

Relación entre la matriz de imagen y el mapa de color.

- Según sean los valores de la matriz de imagen, *double* o *uint8*, pasará lo siguiente:
 - Matriz de imagen *double*. El valor 1 apunta a la primera fila en el mapa de color, el valor 2 a la segunda y así sucesivamente.
 - Matriz de imagen *uint8*. Hay un offset, apuntando el valor 0 a la primera fila del mapa de color, el valor 1 a la segunda, y así sucesivamente.

b2) Imágenes en escala de grises (Intensity images).

Matlab almacena una imagen en escala de grises en una única matriz, correspondiendo cada elemento de ella a un píxel de la imagen.

La matriz puede ser de clase:

- *double*. En este caso contendrá valores dentro del rango [0, 1]
- *uint8*. Pudiendo tomar los datos valores dentro del rango [0, 255]

Los elementos de la matriz representan diferentes niveles de grises, correspondiendo el valor 0 con el negro, y el valor 1, o el 255 con el blanco.

b3) Imágenes binarias.

En las imágenes binarias cada píxel puede tomar únicamente dos valores discretos, *on* y *off*.

Se almacena como una matriz bidimensional de 0's (*off*) y 1 's (*on*).

Pueden almacenarse en matrices de clase *uint8* o *double*. Es preferible almacenarla en matrices de clase *uint8* porque ocupan menos memoria.

En la versión 3.1 de la Toolbox de Procesamiento de Imágenes, cualquier función que devuelve una imagen binaria lo hace como un vector lógico de clase *uint8*.

b4) Imágenes RGB.

Al igual que en una imagen indexada, en una imagen RGB el color de cada píxel se representa por un conjunto de **tres valores**, que corresponden con el rojo, el verde y el azul.

A diferencia de una imagen indexada, estos valores de intensidad **se almacenan directamente en la matriz de la imagen**, y no, de forma indirecta, en un mapa de color.

Los tres componentes del color se almacenan en una **matriz de tres dimensiones**, $m \times n \times 3$, siendo m y n el número de filas y columnas de los píxeles de la imagen, consistiendo la tercera dimensión en tres planos que contienen los valores del rojo, verde y azul. Para cada píxel de la imagen, es necesario combinar los valores del rojo, verde y azul para obtener el color del píxel.

Ejemplo.

Para determinar el color del píxel (112, 86), se busca en los tres valores asociados a esta posición (112, 86, 1:3). Si se supone que:

(112, 86, 1) contiene el valor 0.1238

(112, 86, 2) contiene el valor 0.9874

(112, 86, 3) contiene el valor 0.2543

El color para este píxel vendría dado por sus componentes RGB:

0.1238, 0.9874, 0.2543

Una matriz RGB puede ser de clase:

double: contiene valores en el rango [0, 1].

uint8: contiene valores en el rango [0, 255].

b) Códigos en Matlab.

En la sintaxis de Matlab, cualquier texto que vaya precedido por un signo de porcentaje %, implica que se trata de un comentario.

01) Código para la binarización de las imágenes.

```
clear;  
close all;
```

% Se borran todas las variables y funciones antes utilizadas y se cierran todas las ventanas.

```
L = imread('naranjas/naranja 002.jpg');  
figure(1), imshow(L);
```

% Se carga la imagen original, se le asigna la letra "L" y se muestra en una ventana.

```
G = rgb2gray(L);  
figure(2), imshow(G);
```

% Se transforma la imagen de original (a color) a una imagen en escala de grises y se le asigna la letra "G".

```
F = [1 1 1; 1 9 1; 1 1 1];  
P = conv2(G,F);
```

% Se declara un filtro de tres por tres (paso bajas), se hace la convolución de la imagen con el filtro F y a la resultante se le asigna la letra "P"

```
figure(3);imshow(P);
```

```
B = im2bw(P);  
figure(4), imshow(P);
```

% Se transforma de la imagen P a una imagen binarizada, se le asigna la letra "B" y se muestra en otra ventana el resultado.

```
BW = edge(P, 'canny', 0.15);  
figure(4);imshow(BW);
```

% Se extrae el contorno de la imagen binarizada aplicando el algoritmo de Canny y el resultado se muestra en otra ventana.

02) Código para la obtención del ancho y alto del contorno.

```
clear;  
close all;
```

% Se borran todas las variables y funciones antes utilizadas y se cierran todas las ventanas.

```
[ap,count]=fopen('binarizadas/manza25blJ.jpg', 'r');
```

% Se abre el archivo de la imagen que muestra el contorno del objeto (fruta) y se le asigna el nombre "ap" para poder tener acceso de lectura en forma binaria.

```
fruta=fread(ap);
```

% Se lee la información binaria de la imagen y se reescribe dentro de la matriz "fruta"

```
fruta=fruta';  
figure(1);  
imshow(fruta);  
title ('imagen original');
```

% Se transpone la matriz con los datos de la imagen y el resultado se muestra en una ventana.

```
t=size(fruta);  
disp(t);
```

% Se obtiene el tamaño de la matriz (que es una matriz renglón de dos elementos), se guarda en la variable "t" y se despliega en pantalla.

```
alto=t(1,1);  
disp(alto);
```

% Al primer elemento de "t" se le asigna el nombre alto, es decir, el alto de la imagen.

```
ancho=t(1,2);  
disp(ancho);
```

% Al segundo elemento de "t" se le asigna el nombre ancho, es decir, el ancho de la imagen.

% La siguiente rutina recorre los píxeles de la imagen o elementos de la matriz, para obtener el extremo izquierdo, esto es, recorre los elementos de cada columna desde la que se encuentra más a la izquierda, hasta encontrar el primer elemento diferente de cero, es decir, el elemento de la izquierda del contorno de la imagen.

```
for c=1:ancho
```

```

for r=1:alto
    pix(r,c) = fruta(r,c);
    if pix(r,c) == 0
        x1 = r;
        y1 = c;
        %A = pix(r,c);
        A = [x1, y1];

        break;
        break;
        break;
    end
    %    pix=impixel(fruta,c,r);
    %    pixeles((r-1)*ancho+c,1)=pix(1);

end
end

```

% La siguiente rutina recorre los píxeles de la imagen o elementos de la matriz, para obtener el extremo derecho, haciendo lo mismo que la anterior, sólo que partiendo de la columna que está más a la derecha.

```

for c=ancho:-1:1
    for r=alto:-1:1
        pix(r,c) = fruta(r,c);
        if pix(r,c) == 0
            x2 = r;
            y2 = c;
            %B = pix(r,c);
            B = [x2, y2];

            break;
            break;
            break;
        end
        %    pix=impixel(fruta,c,r);
        %    pixeles((r-1)*ancho+c,1)=pix(1);

    end
end
end

```

% Los valores arrojados por las dos rutinas "A" y "B" corresponden al píxel del extremo izquierdo y al del extremo derecho respectivamente.

```
diam = x2 - x1;
```

% Dado que A y B son coordenadas, lo que se necesita para obtener el valor de la anchura del contorno del objeto es la distancia (en píxeles), con respecto al eje x, la cual se almacenada en la variable "diam".

% La siguiente rutina recorre los píxeles de la imagen o elementos de la matriz, para obtener el extremo superior, esto es, recorre los elementos de cada renglón desde el que se encuentra más arriba, hasta encontrar el primer elemento diferente de cero, es decir, el elemento superior del contorno de la imagen.

```

for r=1:alto
    for c=1:ancho
        pix(r,c) = fruta(r,c);
        if pix(r,c) == 0
            x3 = r;
            y3 = c;
            %A = pix(r,c);
            C = [x3, y3];

            break;
            break;
            break;
        end
        % pix=impixel(fruta,c,r);
        % pixeles((r-1)*ancho+c,1)=pix(1);

    end
end

```

% La siguiente rutina obtiene el extremo inferior del contorno, de manera similar a la rutina anterior, sólo que partiendo del renglón que se encuentra más abajo.

```

for r=alto:-1:1
    for c=ancho:-1:1
        pix(r,c) = fruta(r,c);
        if pix(r,c) == 0
            x4 = r;
            y4 = c;
            %B = pix(r,c);
            D = [x4, y4];

            break;
            break;
            break;
        end
        % pix=impixel(fruta,c,r);
        % pixeles((r-1)*ancho+c,1)=pix(1);

    end
end

```

C
D

% Los variables obtenidas para este caso son "C" y "D" que corresponden a los valores del punto superior y del punto inferior del contorno respectivamente.

$$H = y_4 - y_3;$$

% Para poder obtener la altura del contorno sólo es necesario tomar los valores, con respecto al eje y, de las variables C y D obtenidas antes, lo cual se almacena en "H".

$$E = (y_1 + y_2)/2;$$

% Se obtiene el promedio de la altura de los dos puntos extremos con respecto al eje x, para así determinar a que altura se encuentra la zona más ancha del contorno y se le asigna el nombre "E".

%Valor de la X

$$X = y_4 - E;$$

% Ahora, para poder obtener el valor de X, es decir, la porción de altura que va desde la parte superior del contorno hasta la parte más ancha del mismo, es necesario obtener la distancia (en píxeles) entre el valor de "E" y la ordenada del punto superior, esto es y_4 .

$$CA = X/H;$$

% Para determinar el coeficiente de altura sólo es necesario obtener la razón X/H , a la cual se le asigna el nombre "CA".

diam
CA

03) Código para la obtención de las tres clases con el algoritmo de K-Medias, usando las variables de color y forma.

%Limpia pantalla, elimina las variables y carga el archivo con los datos.

```
clear;  
close all;  
load datos.txt;
```

%Se crean los centros de masa y se les asigna un valor a cada uno.

```
z11 = [0,0];  
z11(1) = datos(1,2);  
z11(2) = datos(1,3);
```

```
z21 = [0,0];  
z21(1) = datos(2,2);  
z21(2) = datos(2,3);
```

```
z31 = [0,0];  
z31(1) = datos(3,2);  
z31(2) = datos(3,3);
```

%Se declara el número de iteraciones a realizar y se inicia el ciclo el ciclo for para calcular las distancias de los datos a los centros de masa.

```
n = 10;  
for k = 1:n,
```

```
    a = 0;  
    b = 0;  
    c = 0;
```

```
    sumac1_x = 0; sumac1_y = 0;  
    sumac2_x = 0; sumac2_y = 0;  
    sumac3_x = 0; sumac3_y = 0;
```

```
    clase1 = [0,0];  
    clase2 = [0,0];  
    clase3 = [0,0];
```

```
    for i = 1:90,
```

```
        d11(i) = sqrt( (z11(1) - datos(i,2))^2 + (z11(2) - datos(i,3))^2 );  
        d21(i) = sqrt( (z21(1) - datos(i,2))^2 + (z21(2) - datos(i,3))^2 );  
        d31(i) = sqrt( (z31(1) - datos(i,2))^2 + (z31(2) - datos(i,3))^2 );
```

%Comparación de la distancia para asignar a que clase pertenece.


```

if (d11(i) <= d21(i)) & (d11(i) <= d31(i))

    a = a + 1;
    clase1(a,1) = datos(i,2);
    clase1(a,2) = datos(i,3);
    sumac1_x = sumac1_x + clase1(a,1);
    sumac1_y = sumac1_y + clase1(a,2);

elseif (d21(i) <= d11(i)) & (d21(i) <= d31(i))

    b = b + 1;
    clase2(b,1) = datos(i,2);
    clase2(b,2) = datos(i,3);
    sumac2_x = sumac2_x + clase2(b,1);
    sumac2_y = sumac2_y + clase2(b,2);

else

    c = c + 1;
    clase3(c,1) = datos(i,2);
    clase3(c,2) = datos(i,3);
    sumac3_x = sumac3_x + clase3(c,1);
    sumac3_y = sumac3_y + clase3(c,2);

end
end

```

%Se inicializan los centros de masa secundarios para calcular su nuevo valor.

```

z12 = [0,0];
z22 = [0,0];
z32 = [0,0];

z12(1) = (sumac1_x/a); z12(2) = (sumac1_y/a);
z22(1) = (sumac2_x/b); z22(2) = (sumac2_y/b);
z32(1) = (sumac3_x/c); z32(2) = (sumac3_y/c);

```

%Incrementa el contador de iteraciones.

```
n = n + 1;
```

%Compara si el nuevo centro de masa es igual al anterior, si son diferentes, asigna el valor del centro de masa nuevo al inicial y regresa al ciclo for, sino termina el ciclo.

```

if (z12 == z11) & (z22 == z21) & (z32 == z31)
    break;
else
    z11 = z12;

```

```

        z21 = z22;
        z31 = z32;
    end
end

```

%Gráfica las clases obtenidas.

```

plot(clase1(1:30,1),clase1(1:30,2), 'b.')
hold;
plot(clase2(1:30,1),clase2(1:30,2), 'rx')
plot(clase3(1:30,1),clase3(1:30,2), 'g*')
hold off;
xlabel('Cociente de alturas');
ylabel('Cociente de colores');
title('GRAFICA DE LOS DATOS');

```

04) Código para la obtención de las fronteras de decisión de Bayes con una variable (cociente del color)

%Se limpia la pantalla y se borran todas las variables.

```

clear;
close all;

```

% se cargan los datos de las manzanas amarillas, y se les asigna una variable.

```

load datosmam.txt;
y = datosmam(1:30,1);
x = datosmam(1:30,2);

```

%Se inicializan todas las variables en cero.

```

n = length(x);
sumaxm=0;
sumaym=0;
sumaxd=0;
sumayd=0;
sumaxy=0;

```

%Ciclo que calcula la media de clase.

```

for i=1:30,
    sumaxm = sumaxm + x(i);
    sumaym = sumaym + x(i);
end
media = [(sumaxm/n),(sumaym/n)];

```

%Ciclo que calcula la varianza y la desviación estándar.

```

for i=1:30,

```

```

    sumaxd = (x(i)-media(1))^2;
    sumayd = (y(i)-media(2))^2;
end

```

```

var = [sumaxd/n,sumayd/n];
desv = [sqrt(sumaxd/n),sqrt(sumayd/n)];

```

%Declaración de una variable: t

```
t=sym('t');
```

(Los pasos anteriores se repiten para las naranjas, datosnar; y las manzanas rojas, datosmro)

%Se declaran las propiedades de la gráfica (título principal y nombre del eje).

```

xlabel('COLOR')
title('DATOS')
hold

```

%Se grafican los datos de last res clases.

```

load datosmam.txt;
x = datosmam(1:30,2);

```

```
y=0;
```

```
plot(datosmam(1:30,2),y,'yh')
```

```

load datosmro.txt;
x = datosmro(1:30,2);

```

```
plot(datosmro(1:30,2),y,'yh')
```

```

load datosnar.txt;
x = datosnar(1:30,2);

```

```
plot(datosnar(1:30,2),y,'yh')
```

%Se calculan las funciones gaussianas para cada clase.

```

x=[0.4:0.001:1];
gxmam=1/(0.027*sqrt(2*pi)) * exp(- ( (x-0.9121).^2) / (2*0.0007) );
plot(x,gxmam,'y')

```

```

gxmro=1/(0.0362*sqrt(2*pi)) * exp(- ( (x-0.5259).^2) / (2*0.0013) );
plot(x,gxmro,'r')

```

```

gxnar=1/(0.0426*sqrt(2*pi)) * exp(- ( (x-0.7681).^2) / (2*0.0018) );

```

```
plot(x,gxnar,'b')
```

%Se trazan las líneas que dividen las tres clases.

```
y=[0:0.001:3];  
x=0.6379;  
plot(x,y,'k');  
x=0.8547;  
plot(x,y,'k');
```

c) Ejemplo de ejecución del Código.

Ejemplo de la extracción del contorno a través de Matlab:



Fig. A.2 Imagen original.

La primera parte del programa, consiste en cargar la imagen original, es decir, seleccionar la imagen a la cual se le aplicará el algoritmo para la extracción de la información, en este caso la imagen de la figura 2.

El siguiente paso consiste en “preparar” la imagen para que pueda extraérsele dicha información por medio de técnicas del procesamiento de imágenes, ésta preparación consiste en transformarla en una imagen con niveles de gris, o lo que regularmente conocemos como blanco y negro, lo cual se puede ver en la siguiente imagen.

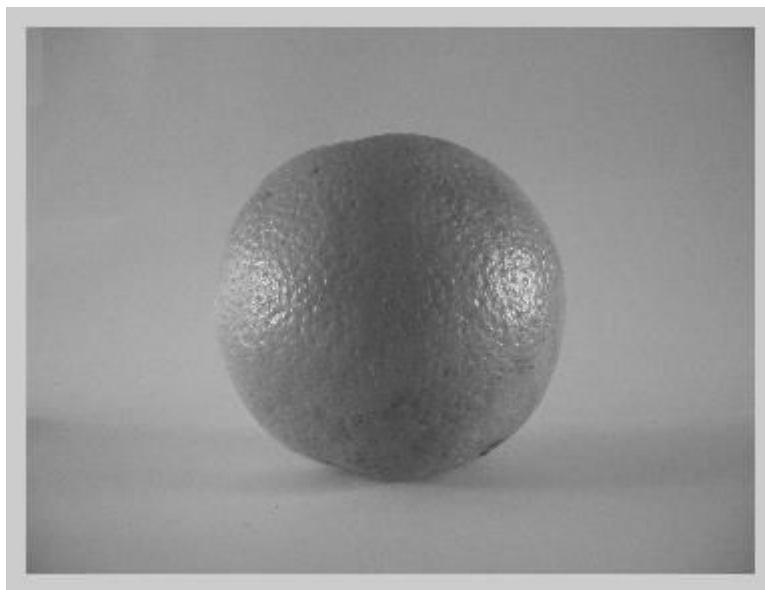


Fig. A.3 Imagen a niveles de gris.

Posteriormente se transforma en una imagen binaria, es decir, con sólo dos colores o niveles, blanco y negro, donde el blanco está dado por el número o nivel 0 y el negro por el 1, y de esta manera se le puede extraer el contorno según el método de Canny obteniéndose el resultado siguiente.

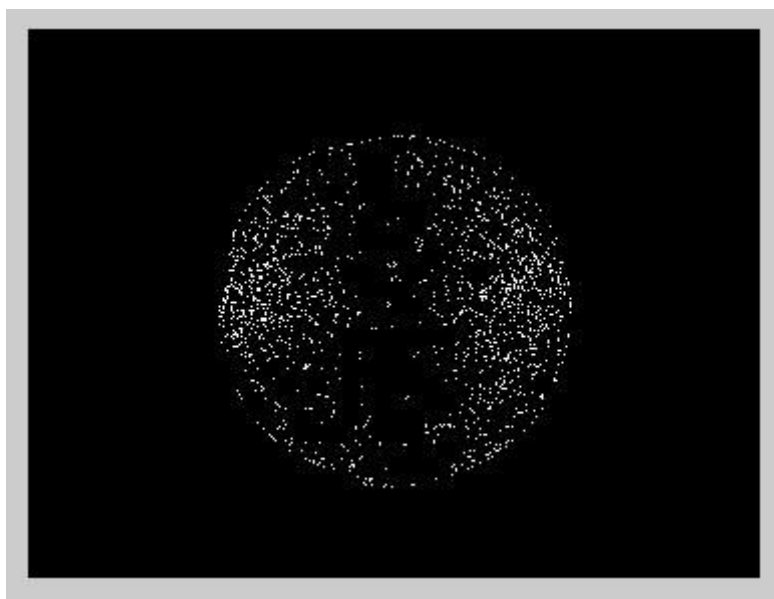


Fig. A.4 Contorno de la imagen.

Bibliografía

- [1] <http://ccc.inaoep.mx/~ariel/recpat.pdf>
- [2] **Image Based Measurement Systems: Object Recognition and Parameter Estimation**
Heijden Ferdinand van der
Ed. John Wiley & Sons
- [3] <http://www.tldp.org/HOWTO/Speech-Recognition-HOWTO>
- [4] <http://www.cs.toronto.edu/~dmac/images/ProjectFiles/sraf/srafdoc/>
- [5] **Pattern Recognition and Image Preprocessing**
Bow Sing-Tze
Mircel Drekker, Inc., Second Edition, 2002.
- [6] **Notas de Reconocimiento de Patrones**
Vázquez Vargas Ana María
- [7] **Digital Image Processing**
Castleman Kenneth R.
Prentice Hall
- [8] **Procesamiento Digital de Imagen: fundamentos y práctica con Matlab**
Alegre Gutiérrez Enrique [et al.]
Universidad de León, 2003.
- [9] **Tratamiento Digital de Imágenes**
González Rafael C., Woods Richard E.
Ed. Addison – Wesley / Díaz de Santos, 1992.
- [10] **Visión por computador. Imágenes digitales y aplicaciones.**
Pajares Gonzalo, De la Cruz Jesús M.
Ed. Alfaomega, Ra-Ma, 2002.
- [11] <http://www.aavbae.net/bol12/12camccd.htm>
- [12] **Guía del usuario de la cámara PowerShot A530**
2006 Canon Inc.
- [13] www.ii.uam.es/~siguenza/Imagenes.ppt
- [14] **Pattern Recognition**
Marques de Sá. J. P.
Ed. Springer, 2000
- [15] <http://es.wikipedia.org/wiki/MATLAB>
- [16] <http://www.mathworks.com/products/matlab/>
- [17] <http://www.mathworks.com/products/matlab/description1.html>