



Reconocimiento de Imágenes

III Unidad

Ms. Ing. Liz Sofia Pedro H.



Contenidos.

1. Segmentación.
2. Reconocimiento
3. Biometria, OCR.



2. RECONOCIMIENTO



2.1. Reconocimiento

□ Posibles aplicaciones:

- Microbiología (cromosomas, contar células, detectar patologías)
- Teledetección (extracción y análisis de la información presente en imágenes de la Tierra y la atmósfera cronológicamente tal vez las primeras)
- Reconocimiento de huellas dactilares
- Reconocimiento de iris
- Reconocimiento de rostros
- Búsqueda de imágenes, etc.



2.1. Reconocimiento (Cont.)

- Adquisición
- Preprocesamiento
- Obtención de regiones o fronteras
- Detección de objetos
- Descripción de objetos
- Reconocimiento de objetos



2.2. Análisis de imágenes

- Permite obtener una descripción de los objetos (medidas)
- Se puede medir los siguientes parámetros:
 - Densitométricos: Cuantifican el nivel de gris y parámetros derivados.
 - Morfológicos: tamaño, forma, orientación y relaciones espaciales.
 - Globales. Porcentaje del volumen de un componente, número de objetos por unidad de volumen
 - Por objeto. Área, perímetro, aspect ratio (relación entre longitud y anchura)



2.2.1. Conteo

- Requerido en muchas aplicaciones.
- Para realizar el conteo se requerirá de una **imagen binarizada**.
- Algoritmos:
 - **Formula de Euler**
 - **Componentes convexas**
 - 4-conectividad
 - 8-conectividad



2.2.1.1. Formula de Euler

- Se debe **eliminar previamente el ruido de la imagen.**
- La imagen debe estar binarizada.
- Se emplea la siguiente



$$g_{i,j} = b_{i+1,j} + 2b_{i+1,j+1} + 4b_{i,j} + 8b_{i,j+1}$$



2.2.1.1. Formula de Euler (Cont.)

- Condición: $0 \leq g_{i,j} \leq 15$
- Se debe construir una matriz de salida G con los valores calculados.
- Considerando los valores de la matriz G se construirá un vector h mediante un conteo muy similar al histograma.

$$h = \{h_0, h_1, h_2, h_3, \dots, h_{14}, h_{15}\}$$



2.2.1.1. Formula de Euler (Cont.)

- Emplearemos los valores del vector para contar los elementos considerando información de vértices, caras y aristas.

numElementos=Vertices-aristas+caras

$$\text{Vértices} = \text{num} \begin{pmatrix} ? & ? \\ 1 & ? \end{pmatrix}$$

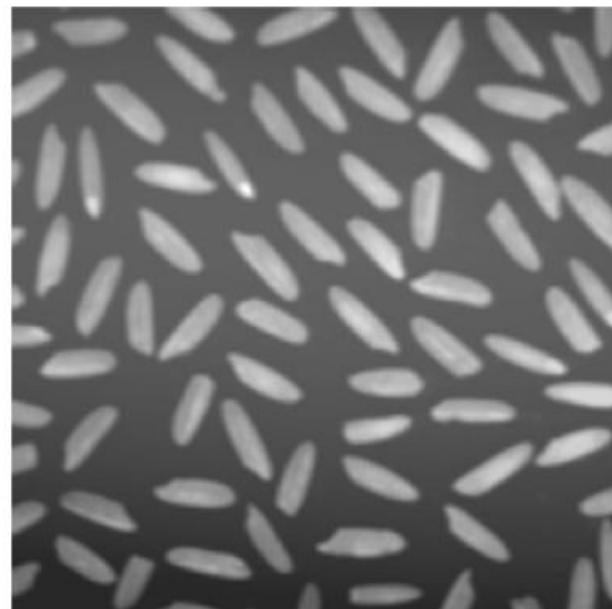
$$\text{Caras} = \text{num} \begin{pmatrix} 1 & ? \\ 1 & 1 \end{pmatrix} + \text{num} \begin{pmatrix} 1 & 1 \\ ? & 1 \end{pmatrix}$$

$$\text{Aristas} = \text{num} \begin{pmatrix} 1 & ? \\ 1 & ? \end{pmatrix} + \text{num} \begin{pmatrix} ? & ? \\ 1 & 1 \end{pmatrix} + \text{num} \begin{pmatrix} 1 & ? \\ ? & 1 \end{pmatrix}$$

$$\chi(\alpha) = h_1 + h_9 - h_6 - h_7$$



1.2.1.1. Formula de Euler (Cont.)



$$\chi(\alpha) = 69$$



2.2.1.2. Conteo de Componentes Conexas

- Se basa en propiedades topológicas como la adyacencia y conectividad independiente del tamaño y forma.
- Son útiles para segmentar regiones conexas, detectar bordes, etc.
- Emplea conceptos de:
 - 4 vecinos de un pixel (horizontales y verticales) (4 adyacentes)
 - 8 vecinos de un pixel (horizontales, verticales y diagonales) (8 adyacentes)
 - Ruta de P a Q: Secuencia de pixeles adyacentes (vecinos) desde P hasta Q



2.2.1.2. Conteo de Componentes Conexas (Cont.)

- Conexidad: Se dice que dos píxeles son conexos en S :
 - S es un subconjunto no vacío de I
 - Existe una ruta entre P y Q que contiene sólo puntos de S .
 - Las rutas entre P y Q pueden ser 4-ruta (4 vecinos) y 8-ruta (8 vecinos).
 - Si S tiene **solo un componente conexo** (Conjunto conexo):
 - S tiene un complemento.
 - El complemento es el único que contiene el borde y se llama fondo de S .
 - Si hubiera otros adicionales al borde entonces son agujeros en S .
 - Si no hay otros entonces S es simplemente C



2.2.1.2. Conteo de Componentes Conexas (Cont.)

- La imagen empleada debe estar binarizada.
- Se considera dos variantes:
 - 4-conectividad en negro
 - 8-conectividad en negro
 - Primera forma
 - Segunda forma



2.2.1.2. Conteo de Componentes Conexas (Cont.)



□ **4-conectividad:**

1. Recorrer la imagen para cada pixel negro P, examinar a los vecinos de **arriba y a izquierda** de P (**si son 1's**, ya están etiquetados) y compararlos
 - Si ambos son 0's, dar a P una nueva etiqueta;
 - Si tan sólo uno es 0, dar a P la etiqueta del otro;
 - Si ninguno es 0's, dar a P la etiqueta de cualquiera (**la izquierda por ejemplo**) , y si sus **etiquetas son diferentes**, registrar el hecho de que **son equivalentes** (es la la misma componente).



2.2.1.2. Conteo de Componentes Conexas (Cont.)

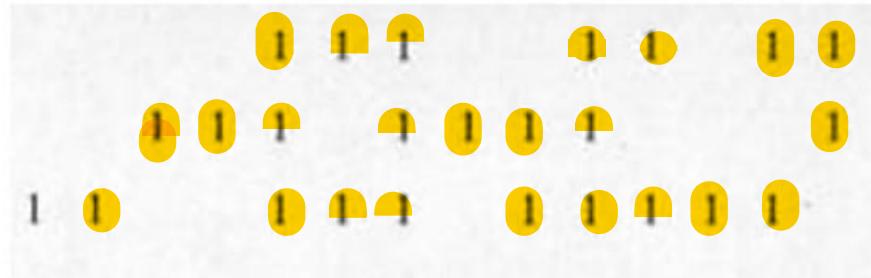
2. Ordenar las parejas equivalentes (pertenecen a la misma clase de equivalencia) y asignarles una etiqueta que represente a la clase.
3. Contar el número final de etiquetas asignadas porque corresponde al número de componentes conexas.



2.2.1.2. Conteo de Componentes Conexas (Cont.)



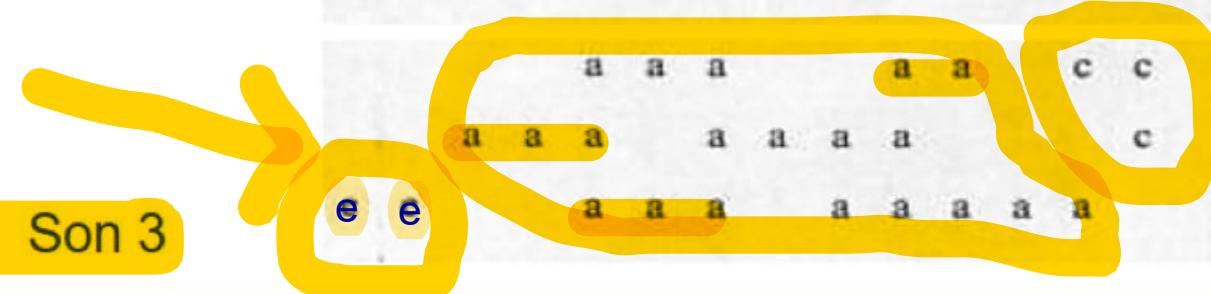
□ Sea la imagen:



□ Paso 1:



□ Paso 2:



□ Paso 3: Son 3

Etiquetas:
a
c
e
Son 3



2.2.1.2. Conteo de Componentes Conexas (Cont.)

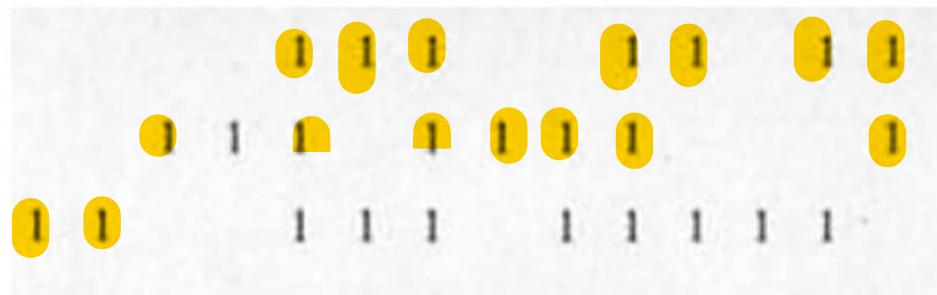
- 8-conectividad (*Primera forma: Comparando 4 pixeles*)
 1. Recorrer la imagen para cada pixel negro P, examinar a los vecinos de *arriba y a izquierda* de P (si son 1's, ya están etiquetados) y compararlos.... (4 pixeles)
 - Si ambos son 0's, dar a P una nueva etiqueta;
 - Si tan sólo uno es 0, dar a P la etiqueta del otro;
 - Si ninguno es 0's, dar a P la etiqueta de cualquiera (la izquierda por ejemplo), y si sus etiquetas son diferentes, registrar el hecho de que son equivalentes (es la misma componente).
 2. *Reetiquetado igual que en 4-conectividad*



2.2.1.2. Conteo de Componentes Conexas (Cont.)

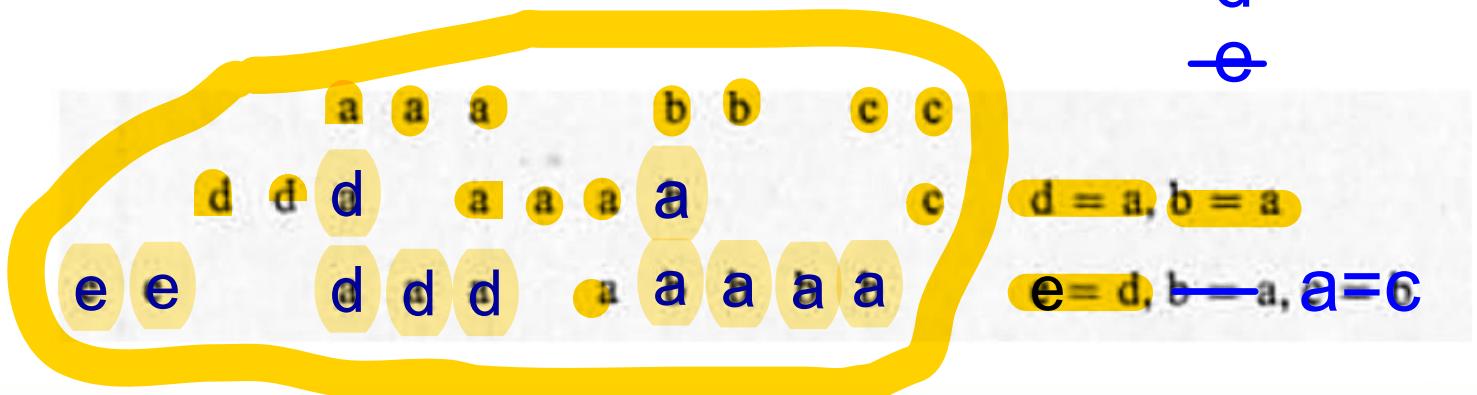
3. Solo hay 1 componente conexa.

- Sea la imagen:



a = b = c = d = e
b
c
d
e

- Paso 1:





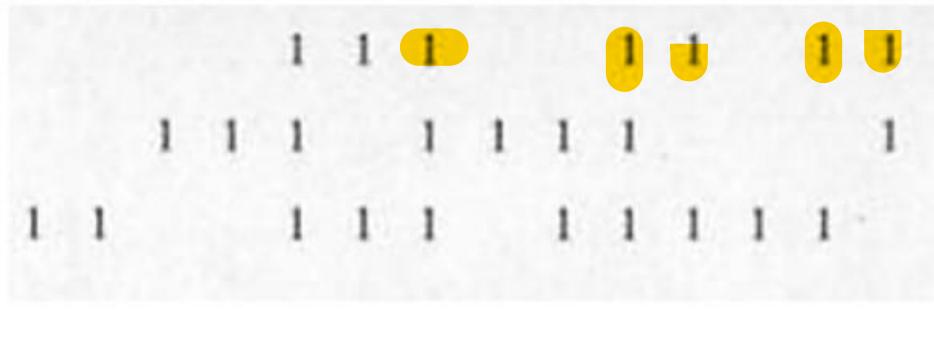
2.2.1.2. Conteo de Componentes Conexas (Cont.)

- 8-conectividad (*Segunda forma: Comparado 3 pixeles*)
 1. Recorrer la imagen para cada pixel negro P, examinar solo vecinos de arriba de P (si son 1's, ya están etiquetados) y compararlos.... (3 pixeles)
 - Si ambos son 0's, dar a P una nueva etiqueta;
 - Si tan sólo uno es 0, dar a P la etiqueta del otro;
 - Si ninguno es 0's, dar a P la etiqueta de cualquiera (la izquierda por ejemplo), y si sus etiquetas son diferentes, registrar el hecho de que son equivalentes (es la misma componente).
 2. Reetiquetado igual que en 4-conectividad



2.2.1.2. Conteo de Componentes Conexas (Cont.)

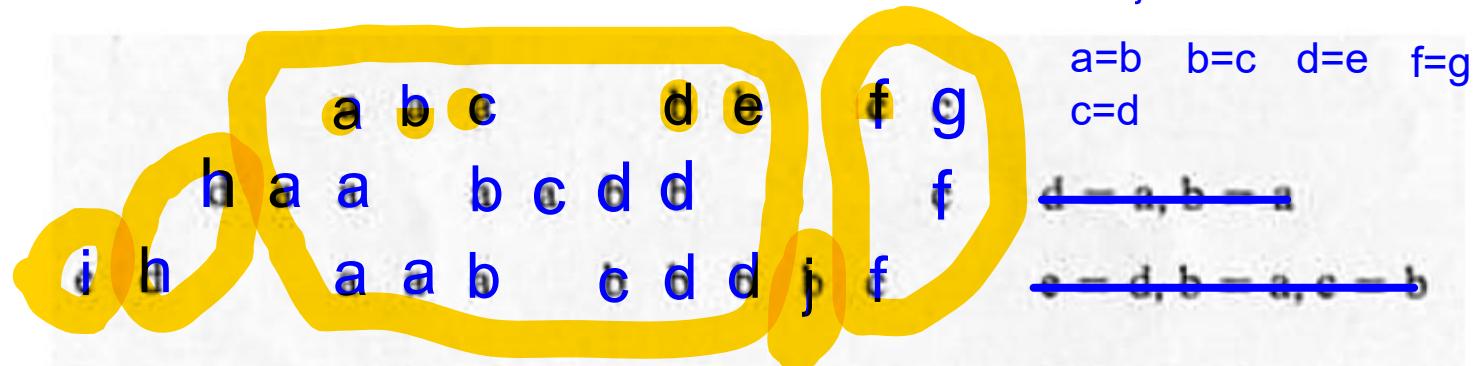
- Sea la imagen:



a =b=c=d=e
b
c
d
e
f =g
g
h
i
j

3. Son 5 cc

- Paso 1:





2.2.1.2. Conteo de Componentes Conexas (Cont.)

□ Comparación:

- Primera forma:

a	a	a	b	b	c	c
d	d	a	a	a	b	c
c	c	a	a	a	a	b

$d = a, b = a$

$e = d, b = a, c = b$

- Segunda Forma:

a	a	a	b	b	c	c
d	a	a	a	b	b	c
e	d	a	a	b	b	c

$d = a, b = a$

$e = d, b = a, c = b$



2.2.2. Seguimiento de bordes

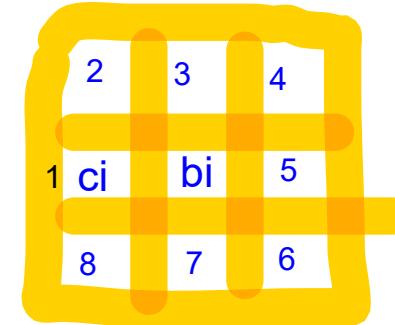
- El algoritmo elegido requieren que los pixeles sean recorridos en el sentido de las agujas del reloj.
- La salida es una secuencia ordenada de puntos.
- La **entrada** debe ser una **imagen binaria (objeto y fondo)**.
- Las imágenes se llenan con un borde de ceros para evitar la fusión de objetos con los bordes de la imagen.
- Se considera **regiones individuales**, pero se puede extender.



2.2.2. Seguimiento de bordes (Cont.)

□ Pasos:

1. Sea el punto de partida b_0 ,
(puede ser el punto
superior más hacia la
izquierda de la región del
objeto).
2. Denote por c_0 al vecino
izquierdo de b_0 donde
 c_0 debe ser un pixel de
fondo.



	1	1	1	1
1				
	1			
	1			
	1	1	1	1

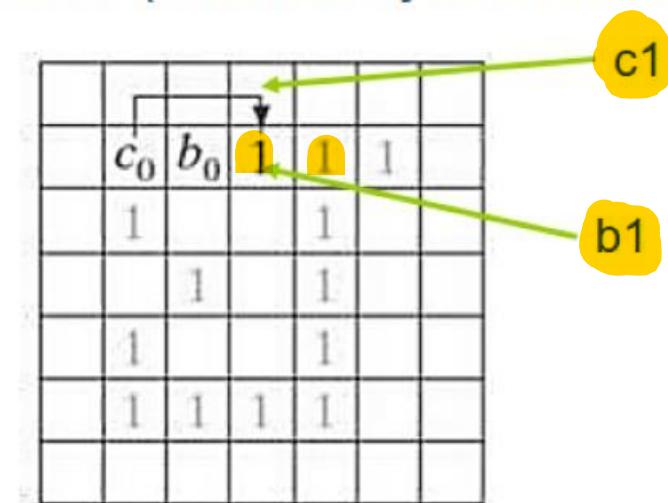
c_0	b_0	1	1	1
1				
	1			
	1			



2.2.2. Seguimiento de bordes (Cont.)

3. Examinar los 8 vecinos de b_0 , comenzando en c_0 en el sentido de las agujas del reloj, hasta encontrar al primer vecino con pixel de objeto b_1 , c_1 el punto de fondo (recorrido justo antes de b_1).

		1	1	1	1
1			1		
	1		1		
1			1		
1	1	1	1		

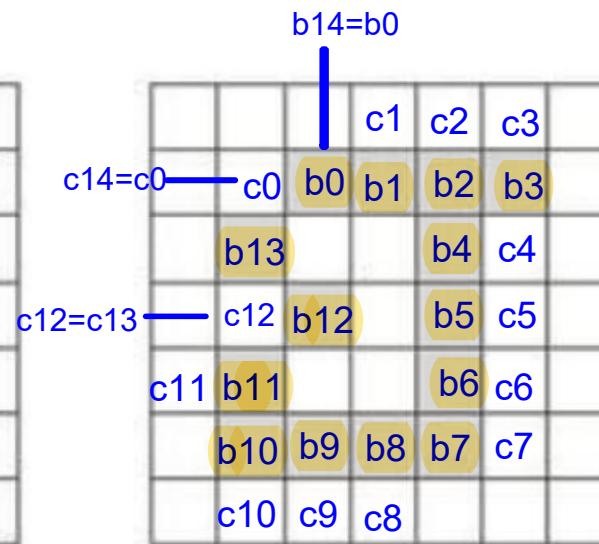




2.2.2. Seguimiento de bordes (Cont.)

3. La operación de elegir nuevos bi y ci , se repetirá hasta que $bi = b0$.

		c1	c2	
c0	b0	b1	b2	1
1				1
	1			1
1				1
1	1	1	1	1



Retorna la secuencia de bi

b0 = (2 , 3)
b1 = (2 , 4)
b2 = (2 , 5)
b3 =
b4 =
b5 =
b6 =
b7 =
b8 =
b9 =
b10 =
b11 =
b12 =
b13 =
b14 =

NOTA: Algunos pixeles podrían repetirse, en la secuencia.



2.2.2. Seguimiento de bordes (Cont.)

□ Tarea:

- Método de Cadenas de código (Chain codes)
- ¿Cómo se puede extender el algoritmo de seguimiento de bordes?



3. RECONOCIMIENTO DE PATRONES



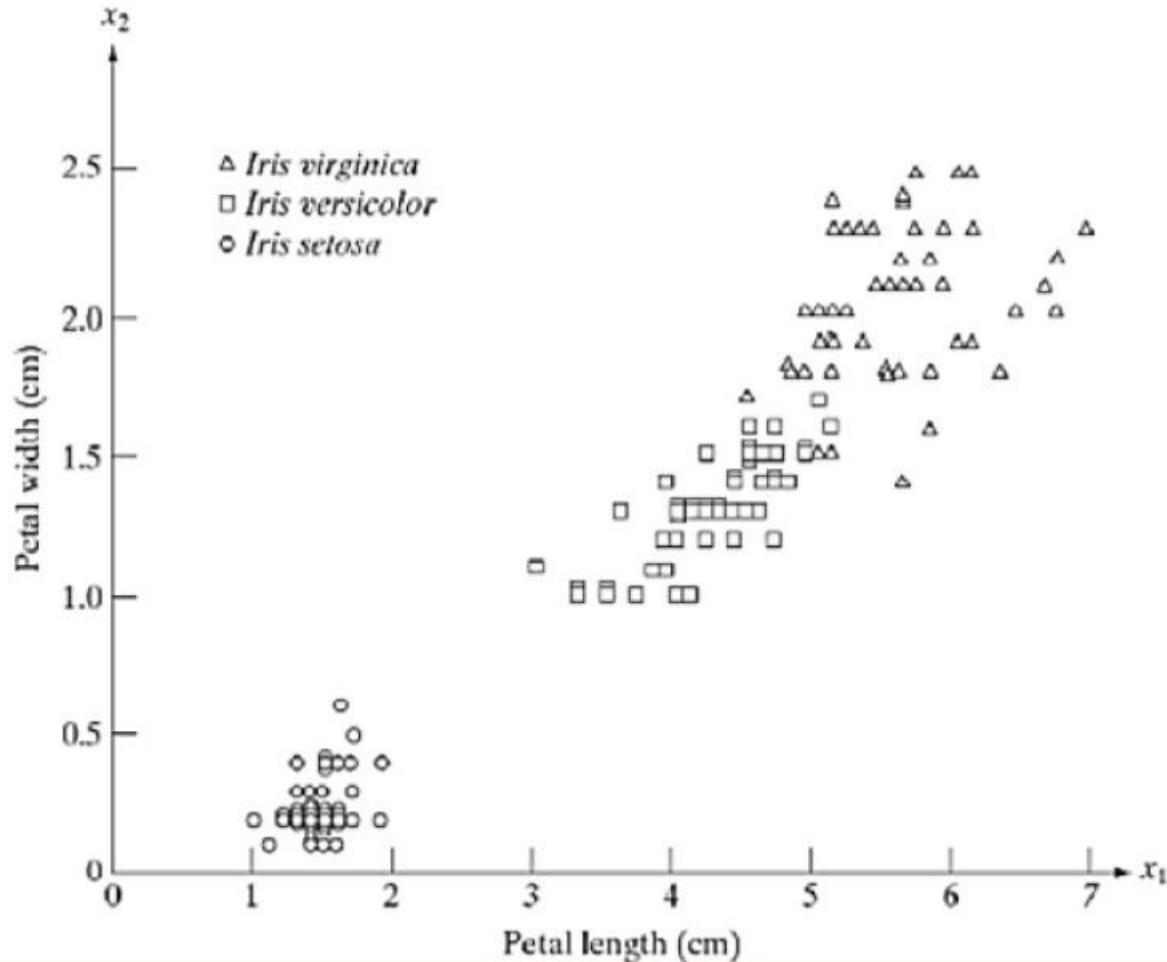
3.1. Reconocimiento de Patrones

- Un *patrón* es un conjunto de características.
- Una *clase de patrones* es un conjunto de patrones “similares”.
- El objetivo del reconocimiento de patrones es asignar un patrón a la clase a la que pertenece.
- Existen varios formatos de representación siendo el más común el vector de características cualitativas X y el árbol para características estructurales.



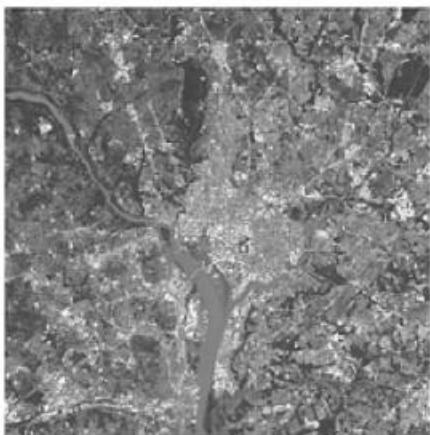
3.1. Reconocimiento de Patrones (Cont.)

- Ejemplo: representación de vector $x = [longitud\ petalos,\ anchura\ petalos]$

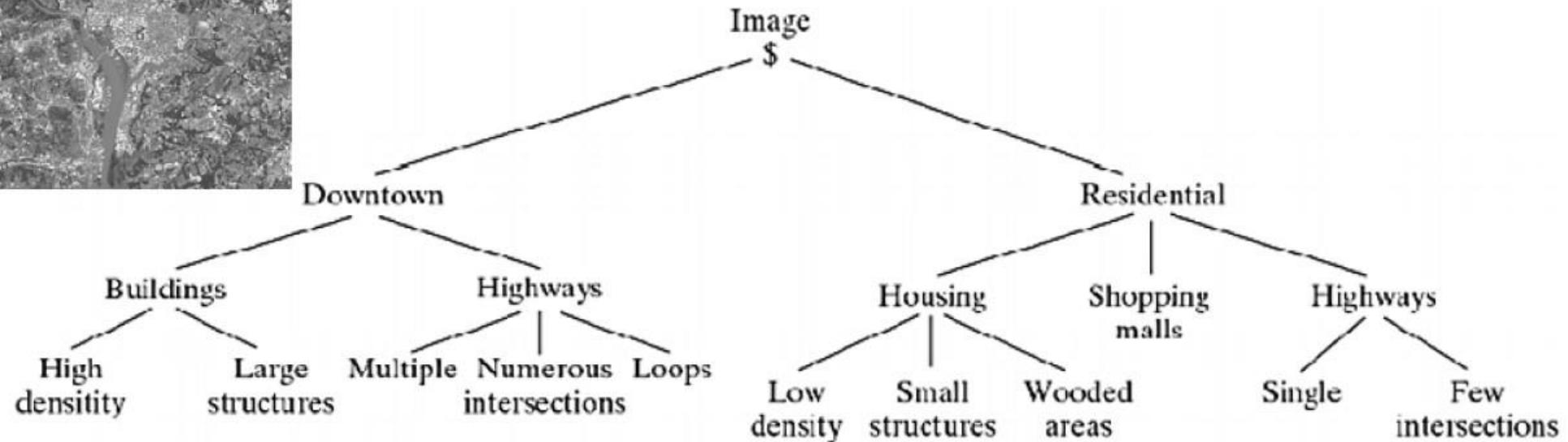




3.1. Reconocimiento de Patrones (Cont.)



□ Ejemplo: representación de árbol





3.1. Reconocimiento de Patrones (Cont.)

- La clasificación es muy usada en reconocimiento de patrones y puede ser utilizar procedimientos:
 - **Estadística** (toma de decisiones) basado en las características estadísticas de los patrones.
 - **Sintáctica (o estructural)**, basado en las relaciones estructurales de las características.



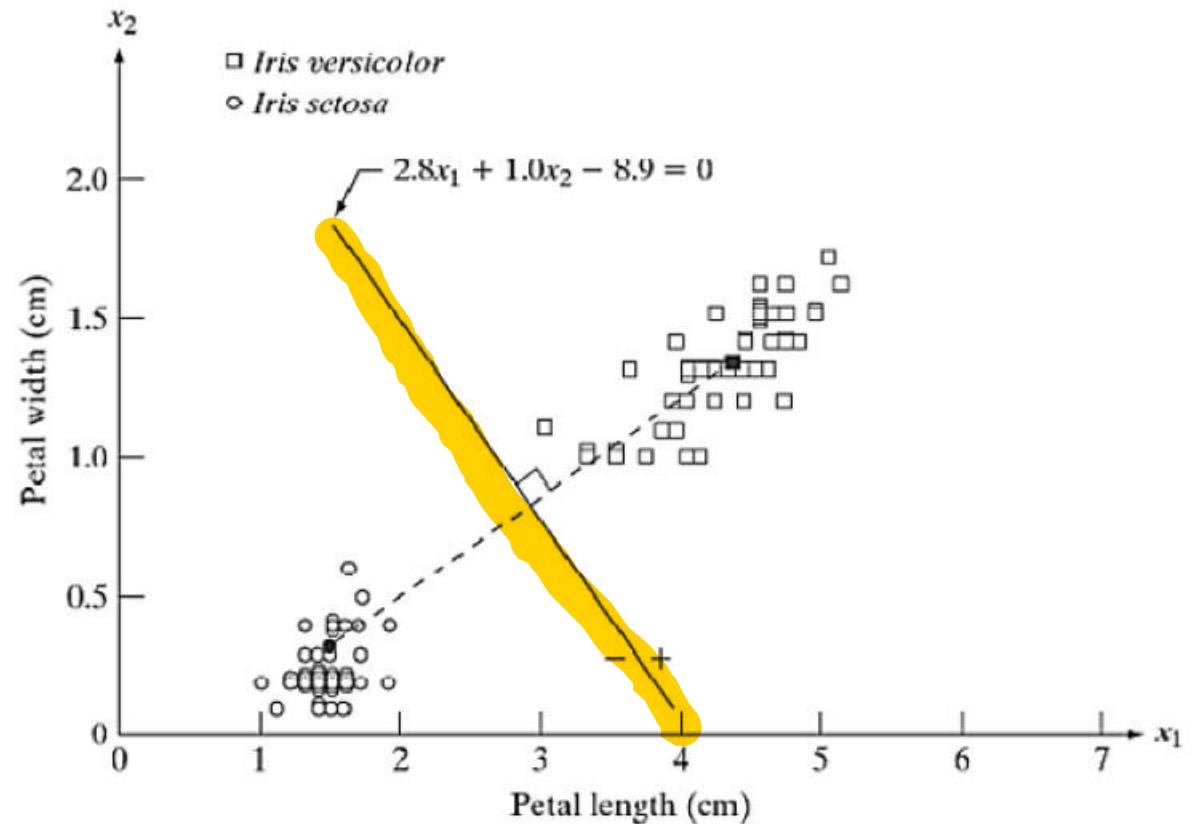
3.1.1. Reconocimiento de patrones basado en métodos de decisión

- Sea x un patrón y x_i una característica de x .
- Para cada clase w de patrones se debe encontrar una función dw (llamado clasificador) tal que si x pertenece a la clase w y no a la clase v , entonces $dw(x) > dv(x)$.
- La frontera de decisión son aquellos vectores x tales que $dw(x) = dv(x)$.
- La función frontera $dw(x) - dv(x)$, toma valores positivos cuando x pertenezca a la clase w y negativos cuando pertenezca a v



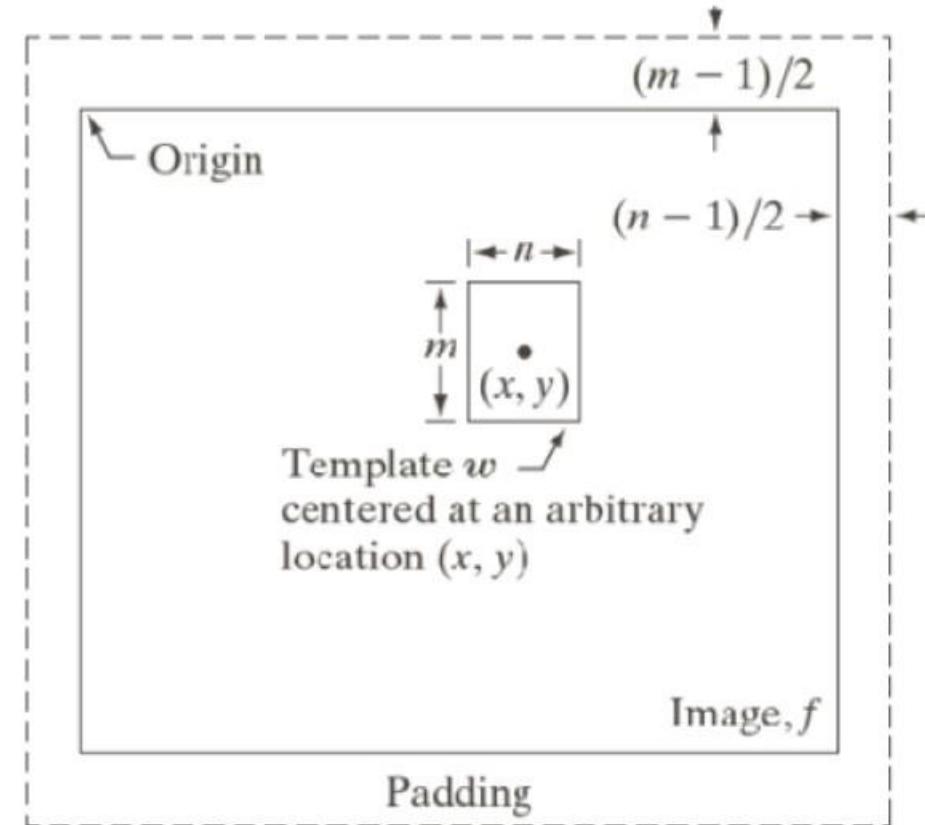
3.1.1. Reconocimiento de Patrones basado en métodos de decisión (Cont.)

- Basado en distancia mínimas:



3.1.1. Reconocimiento de Patrones basado en métodos de decisión (Cont.)

- Basado en correlación





3.1.1. Reconocimiento de Patrones basado en métodos de decisión (Cont.)

- Otros métodos:
 - Mediante árboles de decisión
 - Algoritmos genéticos
 - Bayes.
 - Redes neuronales.



Gracias...