

PRÁCTICA 2

“Extracción de información básica de objetos a partir de imágenes binarias: implementación de funciones de interés”

Objetivo:

Implementación de funciones de interés en la extracción de información de objetos que han sido previamente localizados a través de la generación de matrices binarias. Estas funciones son:

- Algoritmo iterativo de etiquetado.
- Cálculo de área y centroide de objetos.
- Eliminación de componentes conexas cuyo número de píxeles es inferior a un valor dado.

Imagen a utilizar: “ImagenBinaria.tif”.

Ejercicios

1. Lee la imagen del fichero “ImagenBinaria.tif”. Comprueba que es una imagen binaria en la que están localizados los píxeles que integran los objetos que se muestran en la imagen.
2. Genera una matriz binaria tipo *double* donde: el valor 0 signifique píxel de fondo; el valor 1 signifique píxel de objeto.
3. Genera una imagen en color, donde se visualice con un color diferente los objetos presentes en la imagen (al haber 6 objetos, pueden utilizarse los colores primarios y secundarios).

Para ello, se debe implementar la siguiente función (ver documentación adjunta):

```
[Matriz_Etiquetada N] = Funcion_Etiquetar (Matriz_Binaria)
```

donde:

- Matriz_Binaria: matriz binaria con dos posibles valores, 0's y 1's.
 - N: número de agrupaciones conexas de 1's presentes en la matriz (número de objetos de la imagen).
 - Matriz_Etiquetada: matriz tipo double, de las mismas dimensiones que Matriz_Binaria, con $N+1$ posibles valores, 0 para identificar píxel de fondo y un valor de etiqueta i , de 1 a N (números enteros), que se le asigna a todos los píxeles de un mismo objeto.
4. Genera una imagen donde se localicen, a través de su centroide, los objetos de mayor y menor área (ver documentación para la definición de área y centroide).

Para ello, se deben implementar las siguientes funciones:

`Areas = Calcula_Areas (Matriz_Etiquetada)`

donde:

- Matriz_Etiquetada: matriz generada con función_Etiquetar.
- Areas: matriz de N filas y una columna (vector fila) con los valores de áreas de los objetos etiquetados con el valor de la posición de su fila.

`Centroides = Calcula_Centroides (Matriz_Etiquetada)`

donde:

- Matriz_Etiquetada: matriz generada con función_Etiquetar.
- Centroides: matriz de N filas y dos columnas; cada fila contiene la coordenada x (primera columna) e y (segunda columna) del centroide del objeto etiquetado con el valor de la posición de su fila.

5. Genera una imagen binaria donde sólo se visualicen los dos objetos de área mayor.

Para ello, se debe implementar la siguiente función:

`Matriz_Binaria_Filtrada = Filtra_Objetos (Matriz_Binaria , NumPix)`

donde:

- `Matriz_Binaria`: matriz binaria con dos posibles valores, *0*'s y *1*'s.
- `NumPix`: se eliminarán todas las agrupaciones de *1*'s de `Matriz_Binaria` que tengan un área menor a este valor.
- `Matriz_Binaria_Filtrada`: matriz binaria, de las mismas dimensiones que `Matriz_Binaria`, con dos posibles valores, *0*'s y *1*'s, con el valor *1* indicando los píxeles de los objetos cuya área es mayor o igual a `NumPix`.

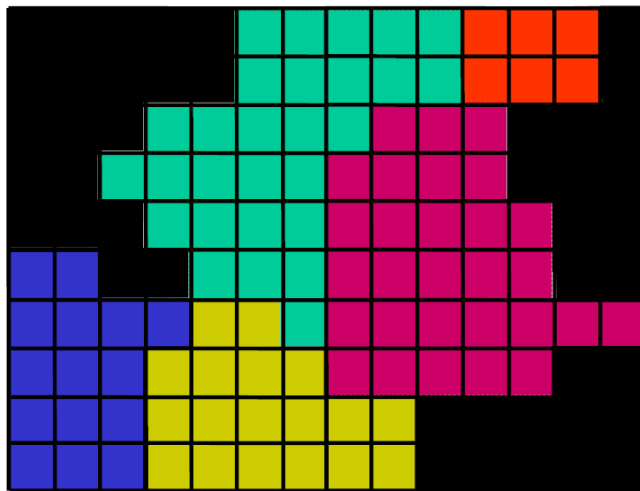
Observación: para implementar esta función deben utilizarse las funciones implementadas en esta práctica.

DOCUMENTACIÓN PRÁCTICA 2: ETIQUETADO

Etiquetado

➤ **Binarización por umbralización:** los píxeles de los distintos objetos se localizan y se distinguen del fondo. Sin embargo, los píxeles de diferentes objetos no están diferenciados (todos tienen el mismo nivel de gris alto) \Rightarrow **Etapas de Etiquetado (“labelling”):** a cada objeto se le asigna una etiqueta, un nivel de gris distinto.

➤ **Objetivo:** crear un mapa de ocupación de regiones (o plano de etiquetado), un conjunto de etiquetas de regiones relacionadas con la imagen, $f(x,y)$, que las diferencie.



Imagen



1	1	1	1	1	2	2	2	2	2	3	3	3	4
1	1	1	1	1	2	2	2	2	2	3	3	3	4
1	1	1	2	2	2	2	2	5	5	5	4	4	4
1	1	2	2	2	2	2	5	5	5	5	4	4	4
1	1	1	2	2	2	2	5	5	5	5	5	4	4
6	6	1	1	2	2	2	5	5	5	5	5	4	4
6	6	6	6	7	7	2	5	5	5	5	5	5	5
6	6	6	7	7	7	7	5	5	5	5	5	8	8
6	6	6	7	7	7	7	7	7	8	8	8	8	8
6	6	6	7	7	7	7	7	7	8	8	8	8	8

Mapa de ocupación o plano de etiquetado

DOCUMENTACIÓN PRÁCTICA 2: ETIQUETADO

➤ **Idea básica:** píxeles vecinos de acuerdo a alguna clase de conectividad pertenecen al mismo objeto y deben ser etiquetados de la misma forma.

→ **Entrada:** imagen binaria donde los “unos” representan los píxeles de los distintos objetos presentes en la imagen.

→ Asumiendo *conectividad-4*, con un orden de exploración de izquierda a derecha y de arriba-abajo se recorre la imagen.

→ Cuando se encuentra el primer píxel que esté a nivel alto se le asocia la etiqueta (“*label*”) 1 para identificarlo como primer objeto.

→ Se examinan sus vecinos: aquellos que también estén a nivel alto recibirán la misma etiqueta.

→ Cuando un píxel no sea vecino de uno etiquetado pero esté a nivel alto se le asocia la siguiente etiqueta, 2, y así sucesivamente.

➤ **Ejemplo:** imagen *a* con regiones segmentadas y resultado final tras el etiquetado (imagen *b*):

$$a \equiv \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}; \quad b \equiv \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 1 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 & 1 & 1 & A \end{bmatrix}$$

→ El píxel *A* tiene un píxel etiquetado con 1 a la izquierda y con 2 por encima, y conecta a las regiones 1 y 2.

→ Todos los píxeles etiquetados con 1 y con 2 pertenecen realmente a la misma región y deben ser etiquetados de la misma forma.

DOCUMENTACIÓN PRÁCTICA 2: ETIQUETADO

Algoritmo iterativo de etiquetado

→ **Entrada: Imagen simple binaria:** $a \equiv \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$

1.- Paso de inicialización:

→ **A cada píxel que esté a nivel alto le asocia una etiqueta distinta:** $b \equiv \begin{bmatrix} 0 & 1 & 2 & 0 & 3 & 4 & 0 \\ 0 & 5 & 6 & 0 & 7 & 8 & 0 \\ 0 & 9 & 10 & 11 & 12 & 13 & 0 \end{bmatrix}$

2.- Primera pasada de arriba-abajo:

→ **Recorrer la imagen, empezando por el primer píxel (extremo superior izquierda), de izquierda a derecha y de arriba abajo.**

→ **Para cada píxel etiquetado, seleccionar la etiqueta mínima de sus vecinos distintos de cero en la misma línea o en la previa.**

$$c \equiv \begin{bmatrix} 0 & 1 & 1 & 0 & 3 & 3 & 0 \\ 0 & 1 & 1 & 0 & 3 & 3 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

3.- Primera pasada de abajo-arriba:

→ **Proceder de la misma forma que en 2, pero empezando por el último píxel (extremo inferior derecha) y recorriendo la imagen de derecha a izquierdas y de abajo a arriba.**

$$d \equiv \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

4.- Repetir pasos 2 y 3 iterativamente hasta que no se produzcan cambios de etiqueta.

Algoritmo iterativo de etiquetado: pseudocódigo

→ Imagen binaria I de $NLINEAS$, cada una con $NPIXELES$ y la etiqueta $ETIQUETA$.

Funciones:

→ $NUEVAETIQUETA$: genera una nueva etiqueta con valor entero cada vez que es llamada.

→ $VECINOS$: devuelve el conjunto de los vecinos ya etiquetados de un determinado píxel en su misma línea o en la línea previa.

→ $ETIQUETAS$: cuando se le proporciona el conjunto de los píxeles ya etiquetados, devuelve el conjunto de sus etiquetas.

→ MIN : cuando se le proporciona un conjunto de etiquetas devuelve la mínima etiqueta.

Procedimiento iterar

“inicialización de cada píxel de valor 1 a una única etiqueta”

```
for L=1 to NLINEAS
  for P=1 to NPIXELES
    if I(L,P) == 1
      then ETIQUETA(L,P) = NUEVAETIQUETA()
    else ETIQUETA(L,P) = 0
  endfor
endfor
```

“Iteración de arriba-abajo seguida por pasos de abajo-arriba”

Repetir hasta que la variable CAMBIO sea FALSO

“Paso de arriba-abajo”

CAMBIO = FALSO

for L=1 to NLINEAS

for P=1 to NPIXELES

if ETIQUETA(L,P) \neq 0

then

M = MIN (ETIQUETAS(VECINOS((L,P)) \cup (L,P)))

if M \neq ETIQUETA(L,P)

then CAMBIO = VERDADERO

ETIQUETA(L,P) = M

endif

endif

endfor

endfor

“Paso de abajo-arriba”

for L=NLINEAS to 1 (por -1)

for P=NPIXELES to 1 (por -1)

If ETIQUETA(L,P) \neq 0

then

M = MIN (ETIQUETAS(VECINOS((L,P)) \cup (L,P)))

if M \neq ETIQUETA(L,P)

then CAMBIO = VERDADERO

ETIQUETA(L,P) = M

endif

endif

endfor

endfor

Fin Iterar

➤ **Tamaño o Área (A): número de píxeles comprendidos en el contorno del objeto.**

❖ Si $f(x,y)$ es una imagen binaria donde los píxeles del objeto tienen todos valor “1” y los del fondo valor “0”:

$$A = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)$$

➤ **Centroide o centro de gravedad: se determina promediando las coordenadas x e y de los píxeles que conforman el contorno.**

$$\bar{x} = \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} xf(x,y)}{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)} \quad ; \quad \bar{y} = \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} yf(x,y)}{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)}$$