Tarea #2: Fundamentos de la imagen digital

Dan Williams Robledo Cruz

02 de octubre del 2013

Procesamiento Digital de Imágenes

Cinvestav

1. Introducción

La conectividad es un concepto muy importante para poder delimitar los limites de los objetos en regiones dentro de una imagen digital.

Para decir que 2 pixeles están conectados entre si se debe cumplir que sean adyacentes en algún sentido(8 o 4 adyacencia) y que sus niveles de grises sean iguales, y así poder sacar el contorno de algún objeto presente en la imagen, para poder detectar múltiples objetos se usa el etiquetado seguido de la detección de contorno.

Es esta practica se vera cada una de los pasos para detección de contornos de uno o mas objetos en una imagen.

2. Desarrollo

2.1. Descripción de la practica

1. Hacer una función en Matlab para la decteccion de contorno de una imagen binaria usando 4-adyacencia.
2. Hacer una función en Matlab para el etiquetado de las regiones conexas disjuntas de una imagen binaria.
3. Realizar el seguimiento de múltiples contornos en una imagen binaria.

2.2. Seguimiento de contorno con 4 adyacencia

Su algoritmo es el siguiente

1. Se busca el primer pixel P_0 de la región conexa a partir de la esquina superior izquierda.
2. Se define una variable d que almacena la dirección del recorrido del pixel. d=1.
3. Calcular N_4 alrededor del pixel actual P_i el inicio de la búsqueda es en la posición actual es q=(d+3) mod 4.
4. A partir de q buscar en sentido del reloj el primer pixel con valor 1 para determinar el nuevo elemento del contorno P_i.
5. Actualizar la dirección del recorrido d al el pixel actual.
6. Si el píxel actual P_i es igual al segundo elemento del borde P_1 y si el elemento previo P_{i-1} es igual al punto inicial del borde P_0 entonces terminar el algoritmo sino regresar al paso 1.

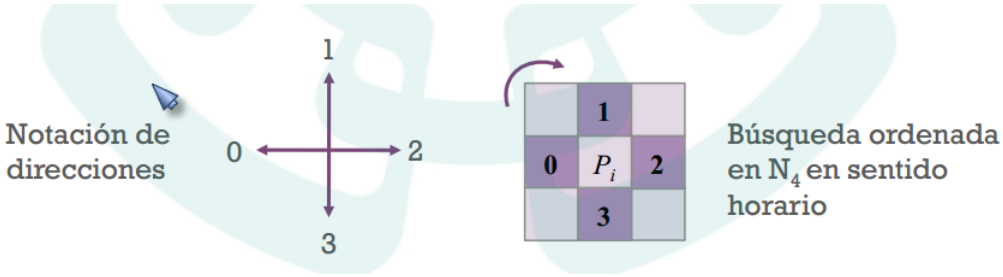

Figura 1: Seguimiento de contornos 4 adyacencia[1]

2.2.1. Se empieza a buscar los pixeles adyacentes y se guarda en una matriz temporal

```
1          q =mod ((d+3),4);
2          switch(q)
3          case 3
4          if(temp(a+1,b)~=0)%3
5          a=a+1;
6          M_temp(a-1,b-1)=temp(a,b); %se guarda el pixel
               encontrado
7          p=[p [a ;b]];
8          d=3;
9          else
10         d=1;
11         end
```

```
12
13          case 0
14          if(temp(a,b-1)~=0)%0
15          b=b-1;
16          M_temp(a-1,b-1)=temp(a,b);
17          p=[p [a ;b]];
18          d=0;
19          else
20          d=2;
21          end
22
23          case 1
24          if(temp(a-1,b)~=0)%1
25          a=a-1;
26          M_temp(a-1,b-1)=temp(a,b);
27          p=[p,[a ;b]];
28          d=1;
29          else
30          d=3;
31          end
32
33          case 2
34          if(temp(a,b+1)~=0)%2
35          b=b+1;
36          M_temp(a-1,b-1)=temp(a,b);
37          p=[p,[a ;b]];
38          d=2;
39          else
40          d=0;
41          end
```

2.2.2. Si el píxel actual P_i es igual al segundo elemento del borde P_1 y si el elemento previo P_{i-1} es igual al punto inicial del borde P_0 entonces terminar

```
1           if(p(:,end-1)==p(:,1) )
2                   if(p(:,end)==p(:,2))
3                   flag=1;
4                   end
5           end
```

2.3. Etiquetado de regiones conexas

El etiquetado consiste en asignar que identifique las distintas regiones disjuntas de una imagen, es muy usado en imágenes binarias para identificar los diferentes objetos en la imagen.

El algoritmo clásico de etiquetado consta de 3 pasos.

1. Propagación de las etiquetas.

2. Resolver tablas de equivalencias

3. Asignación de etiquetas usando clases de equivalencia.

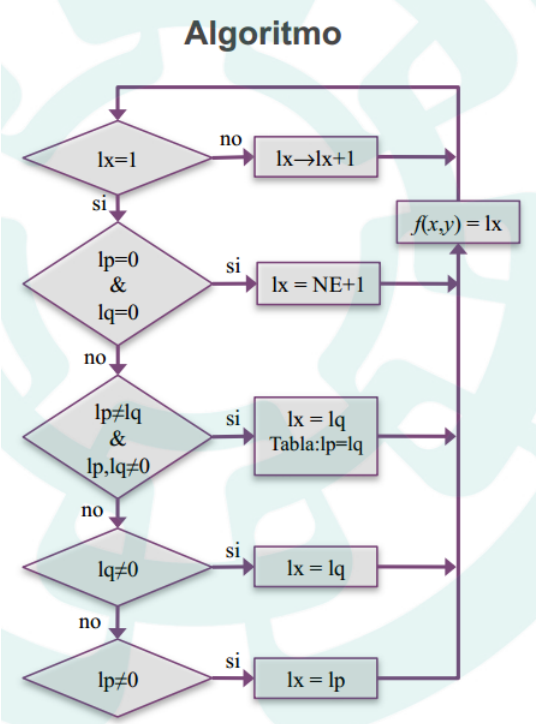

Figura 2: Algoritmo clásico de propagacion de etiquetas para 4-conectividad[2]

2.3.1. Propagación de las etiquetas

Se empieza a analizar el pixel actual si este es igual a 0 se pasa a analizar el siguiente pixel, si es igual a 1 se comprueba si su pixel superior e izquierdo es igual a 0, si cumple la condición se asigna una etiqueta y se incrementa el numero de etiquetas mas 1.

Si la condición anterior no se cumple se analiza si su pixel superior e izquierdo son diferentes y diferentes de 0, si la condición se cumple se añade a la tabla de equivalencia la relación entre el pixel superior e izquierdo y se pone la etiqueta del pixel izquierdo.

Si el pixel izquierdo es diferente de 0 se añade la etiqueta del pixel izquierdo a pixel actual

```
1           lx=I(i,j);
2           if(lx~=0)
3                   if(I(i,j-1)==0&&I(i-1,j)==0)
4                   Ne=Ne+1;
5                   lx=Ne;
6                   I(i,j)=lx;
7                   elseif(I(i,j-1)~=I(i-1,j)&&I(i,j-1)~=0&&I(i-1,
                        j)~=0)
8                   tabla=[tabla;I(i-1,j),I(i,j-1)];
9                   lx=I(i,j-1);
10                  I(i,j)=lx;
11                  elseif(I(i,j-1)~=0)
12                  lx=I(i,j-1);
13                  I(i,j)=lx;
14                  elseif(I(i-1,j)~=0)
15                  lx=I(i-1,j);
16                  I(i,j)=lx;
17                  end
18          end
```

2.3.2. Resolver tabla de equivalencias

A partir de los pares de la tabla de equivalencias generar la matriz binaria B y se convierte en reflexiva y simétrica .

```
1           b=eye(length(clases));
2           aux=length(tabla(:,1));
3           for i=2:aux
4                   b(tabla(i,1),tabla(i,2))=1;
5                   b(tabla(i,2),tabla(i,1))=1;
6           end
```

Se resuelve resuelve la tabla de equivalencia con el algoritmo de Warshall.

```
Bm=zeros(length(clases));
          for i=1:length(clases)
                  for j=1:length(clases)
                          if(b(i,j)==1)
                          for k=1:length(clases)
                                  Bm(i,k)=b(i,k)|b(k,j);
                          end
                          end
                  end
          end
```

2.3.3. Asignación de etiquetas usando clases de equivalencias

A partir de Bm se determinan las clases de equivalencias y se sustituyen las etiquetas equivalentes en la matriz de salida de la imagen.

```
for i=1:length(clases)
        for j=1:length(clases)
                if(Bm(i,j)==1)
                        clases(j)=clases(i);
                end
        end
end

idx = unique(Salida(:));
for i = 2:length(idx)
        ind = Salida==idx(i);
        Salida(ind)=clases(i-1);
end
```

2.4. Resultados: Realización del seguimiento de múltiples contornos en una imagen binaria.

Esta practica se concluye usando los métodos de etiquetado de regiones conexas combinado con el de detección de bordes, para poder localizar múltiples objetos en una imagen binaria.

El argumento de entrada es la imagen binaria de los objetos y el argumento de salida es la imagen binaria del contorno de los objetos.

1. Se lee la imagen binaria con la función 'readbmp'.
2. Se pasa la imagen leída a la función 'bwlabeling' para hacerle el etiquetado.
3. Después de etiquetar la imagen se pasa a la función 'bwedge' para detectar los bordes de los objetos uno a uno, guiándose por el etiquetado de cada objeto.
4. Se manda a imprimir las imágenes resultantes para ver los resultados

```
1  i1 = fopen('bw1.bmp','r');
2  if(i1~=-1)
3          [Isalida,flag]=readbmp(i1);
4          Salida=bwlabeling(Isalida);
5          Salida2=bwedge(Salida);
6          Salida=label2rgb(Salida);
```

```
7           if(flag~=0)
8                   figure('Name','Practica 2.3: Seguimiento del
                        contorno de objetos múltiples');
9                   subplot(2,2,1);
10                  imshow(Isalida,[]); %Muestra la Imagen 1
11                  title('Imagen original')
12                  subplot(2,2,3);
13                  imshow(Salida,[]); %Muestra la imagen 2
14                  title('Imagen Etiquetada por regiones')
15                  subplot(2,2,4);
16                  imshow(Salida2,[]); %Muestra la imagen 2
17                  title('Contorno de objetos múltiples')
18                  fclose(i1);
19  end
```

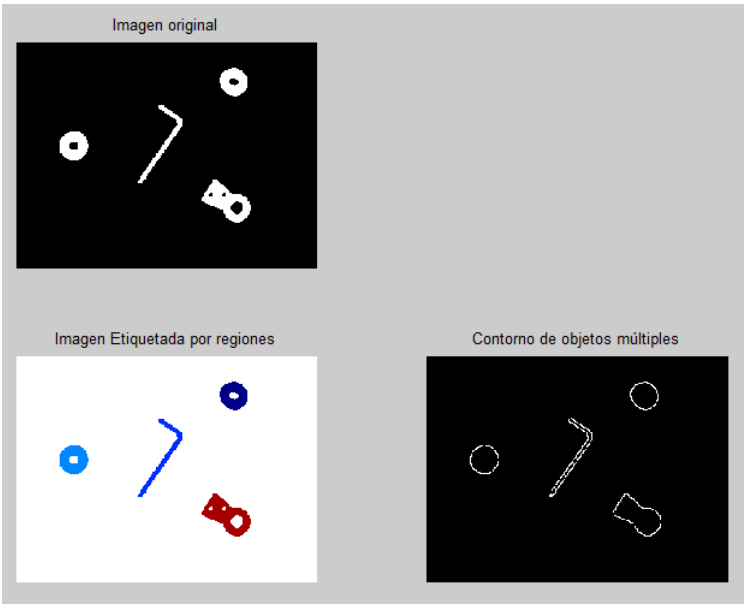

Figura 3: Resultado después de aplicar el metodo de etiquetado y deteccion de bordes

Referencias

[1] Wilfrido Gómez Flores———Clase03a_PDI_2013.pdf

[2] Wilfrido Gómez Flores———Clase03b_PDI_2013.pdf