Práctica 1: Vecindad y regiones

Visión por Computadora

Elaborado por:
José Baltazar Ramírez Rodríguez

Dra María Susana Ávila García

05 de febrero del 2019

I. DESCRIPCIÓN DEL PROBLEMA

Una imagen está compuesta por muchos píxeles. Los niveles de intensidad de estos píxeles irán variando en un rango de 0 a 255, forma mediante la cual se definirán los colores de una imagen. En una imagen binaria, sólo se pueden tener dos valores: 0 ó 1.

Los píxeles pueden ser manipulados para incrementar o decrementar su valor. Para esto una imagen puede y debe ser tratada como una matriz para poder accesar a cada una de sus posiciones y modificar el valor del píxel.

Se puede tener relaciones básicas entre píxeles. Entre los expuestos en clase de Visión por Computadora están:

- Vecindad
- Adyacencia
- Conectividad
- Rutas
- Regiones y bordes

De la presentación de clase, cualquier pixel P(x,y) tiene dos verticales y dos horizontales vecinos, dados por:

$$(x+1,y)(x-1,y),(x,y+1)(x,y-1)$$

Este conjunto de pixeles es llamado 4-vecinos de P.

Existe también los vecinos en diagonal de P, dados por:

$$(x+1,y+1)(x+1,y-1),(x-1,y+1)(x-1,y-1)$$

La suma de estos dos vecinos se denomina 8-vecinos.

Respecto a bordes y regiones, un subconjunto de pixeles en una imagen es llamado **región**, si ese subconjunto está conectado entre sus pixeles. El **borde** de una región son los pixeles en la región que tienen uno o más vecinos que no están en esa región.

El problema a resolver consiste en que se determine en una imagen binaria, las regiones que se encuentren dentro de ella, en este caso donde se ubiquen 1’s verificando que tenga 8-vecinos.

En una segunda instancia, se deben detectar los bordes de una región con esta misma matriz de la imagen binaria.

II. ALGORITMO UTILIZADO PARA RESOLVER EL PROBLEMA

Para completar esta tarea de verificar las regiones y que existan vecinos alrededor de ciertos pixeles, se utilizó el algoritmo siguiente:

Recorrer Matriz de 1's y 0' hasta encontrar un 1

Si encuentra un 1 verificar si en alguna de sus vecinos existe un 1

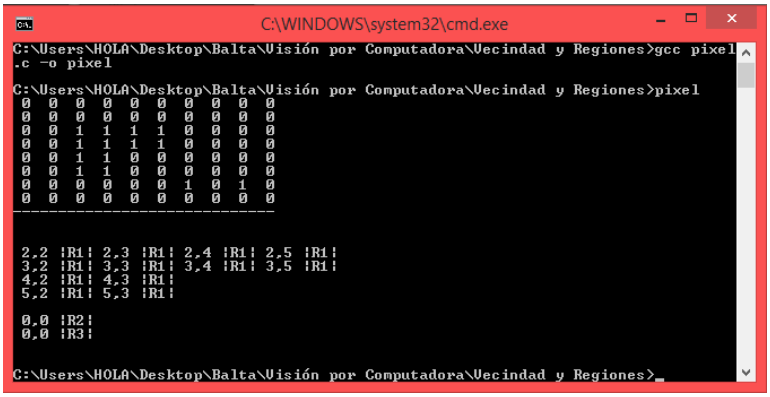
Existe vecino y guardar valores de fila y columna en un arreglo

De lo contrario se marca que terminó de buscar un 1 y se incrementa la región

En el primer caso, había que detectar la coordenada y la región a la que pertenecía tal posición, se presentó un inconveniente que no se supo resolver. Al momento de detectar la región a la que pertenece, se realizaba de manera correcta, así como su almacenamiento en la estructura. El problema surgió al mostrar las coordenadas de una región de la imagen. Verificaba correctamente que existieran vecinos, pero no se pudo concretar al 100% la funcionalidad del programa. Los casos de prueba fueron con una matriz como la siguiente:

```
0000000000
0000000000
0011110000
0011110000
0011000000
0011000000
0000001010
0000000000
```

Obteniendo la salida



Y para definir si existen bordes, se utiliza el algoritmo:

Recorrer Matriz de 1's y 0' hasta encontrar un 1

Si encuentra 1, verifica que en alguno de sus vecinos exista un 0

Se guardan las coordenadas de fila y columna en un arreglo

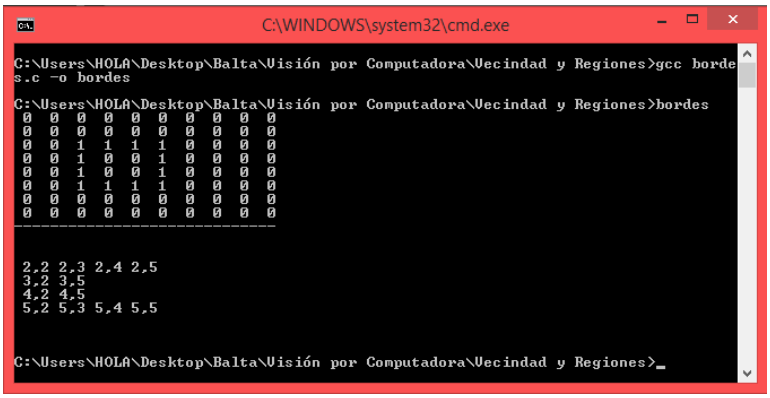
De lo contrario, sigue buscando 0's alrededor de los sig 1's para encontrar bordes.

Se declaró en ambos programas una estructura para almacenar las posiciones donde se encontraban los elementos que nos interesaba obtener, para mostrarse posteriormente las coordenadas con las posiciones. La matriz en ambos escenarios fue de 8 filas y 10 columnas, llena de 0's y 1's.

En este programa para detectar bordes se cumple la funcionalidad al 100%. Despliega las coordenadas en donde se encuentran los bordes. Se realizaron distintas pruebas.

El primer caso para probar era cuando nuestra matriz

```
0000000000
0000000000
0011110000
0010010000
0010010000
0011110000
```



También se realizó la prueba con una matriz para al borde de una imagen. Obteniendo la salida.