

Tarea 11

Axel Báez Ochoa

5 de noviembre de 2020

Se implemento el algoritmo de conjunto conexos en imágenes pgm.

1. Introducción

Una componente conexa es un conjunto de píxeles, C , tal que para cualquier par de píxeles del conjunto, existe un camino digital que los une, donde este camino esta dentro de C . Sea S una imagen digital, podemos verla en un mallado cuadrado o una matriz, donde cada entrada es un píxel de la imagen y el valor de esta representa el color del píxel. Lo que se busca es encontrar los diferentes conjuntos conexos de la imagen, esto se logra considerando los vecinos de un píxel, Si un píxel se encuentra en la posición (x, y) , definimos a sus ocho vecinos como

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

Para los píxeles de la matriz se puede definir una relación que define dos píxeles como conectados cuando son vecinos y sus valores son similares desde algún punto de vista por lo que para ver que píxeles están conectados, recorreremos la imagen de izquierda a derecha, de arriba hacia abajo, para cada píxel visitamos a sus vecinos y para cada vecino visitamos a los vecinos de este, así sucesivamente, considerando que no debemos visitar píxeles ya visitados, de esta forma podemos establecer una relación para un conjunto de píxeles, de acuerdo al valor de estos, cuando los todos los vecinos se han visitado decimos que este conjunto de píxeles es un conjunto conexo y continuamos con el recorrido de la imagen hasta encontrar un nuevo píxel que no forme parte de este conjunto que ya definimos, al encontrar un nuevo píxel se repite el proceso hasta delimitar una conjunto de píxeles relacionados entre si, este proceso continua hasta haber recorrido toda la imagen.

2. Descripción

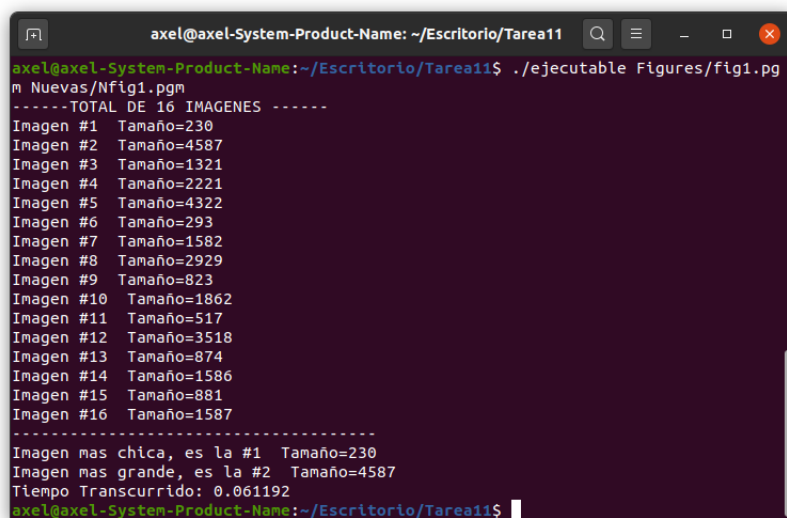
Se utilizo el sistema de 8 vecinos para separar los conjuntos, el programa requiere que se corra con dos argumentos, la ruta de la imagen a procesar y la ruta donde se guardara la nueva imagen, por ejemplo para la imagen número uno se ejecuta de la forma

`./ejecutable Figures/fig1.pgm Nuevas/Nfig1.pgm`

donde **Figures** es la carpeta donde se encuentran todas las 5 imágenes de prueba y **Nuevas** es la carpeta para las nuevas imágenes generadas.

Comenzamos leyendo el archivo de la imagen.pgm , creamos una matriz de datos que contenga el valor de cada píxel, y cuando el valor era mas cercano a 0 que a 255 lo guardábamos como 0 en caso contrario se guardaba como 255. Para llevar a cabo el proceso para ver cuantas figuras hay en la imagen, creamos una matriz adicional con las mismas dimensiones de la primera, cuyas entradas son puros ceros, recorremos la matriz principal empezando en 0,0 y cuando un valor sea diferente de 0 visitamos todos los vecinos de esta celda y los vecinos de los vecinos, así sucesivamente hasta que ya no halla mas por visitar, cada que nos colocamos en una nueva celda, volvemos 1 el lugar correspondiente en la segunda matriz, esto nos indica que este vecino ya se visito para evitar que se vuelva a ir a el, después de esto continuamos el recorrido de la matriz original y cuando un valor sea diferente a 0 y su valor correspondiente en la segunda matriz sea 0, empezamos a visitar los vecinos de esta celda ahora marcando con un 2 el lugar correspondiente en la segunda matriz, continuamos este proceso hasta que recorramos toda la matriz principal, de esta forma al final tendremos nuestra matriz secundaria llena de 0 y 1,2,...,n donde n es el número de figuras encontradas en el archivo .pgm , por lo que si el archivo .pgm tiene 4 conjuntos conexos entonces nuestra matriz secundaria estará llena de números 0,1,2,3 y 4. Ahora contamos cuantas veces se repite cada uno de estos números en nuestra matriz secundaria y tenemos el tamaño de cada conjunto, de aquí vemos cual es el mas grande y mas pequeño, y al escribir la nueva imagen nos fijamos en la matriz secundaria , recorremos esta y cuando el número de la celda sea igual al de nuestro conjunto mas pequeño o mas grande, le asignamos el valor de píxel que tenia en la imagen original y a los demás les asignamos 0, de esta forma nuestra nueva imagen solo tendrá a la figura mas grande y mas pequeña.

3. Resultados



```
axel@axel-System-Product-Name: ~/Escritorio/Tarea11
axel@axel-System-Product-Name:~/Escritorio/Tarea11$ ./ejecutable Figuras/fig1.pg
m Nuevas/Nfig1.pgm
-----TOTAL DE 16 IMAGENES -----
Imagen #1 Tamaño=230
Imagen #2 Tamaño=4587
Imagen #3 Tamaño=1321
Imagen #4 Tamaño=2221
Imagen #5 Tamaño=4322
Imagen #6 Tamaño=293
Imagen #7 Tamaño=1582
Imagen #8 Tamaño=2929
Imagen #9 Tamaño=823
Imagen #10 Tamaño=1862
Imagen #11 Tamaño=517
Imagen #12 Tamaño=3518
Imagen #13 Tamaño=874
Imagen #14 Tamaño=1586
Imagen #15 Tamaño=881
Imagen #16 Tamaño=1587
-----
Imagen mas chica, es la #1 Tamaño=230
Imagen mas grande, es la #2 Tamaño=4587
Tiempo Transcurrido: 0.061192
axel@axel-System-Product-Name:~/Escritorio/Tarea11$
```

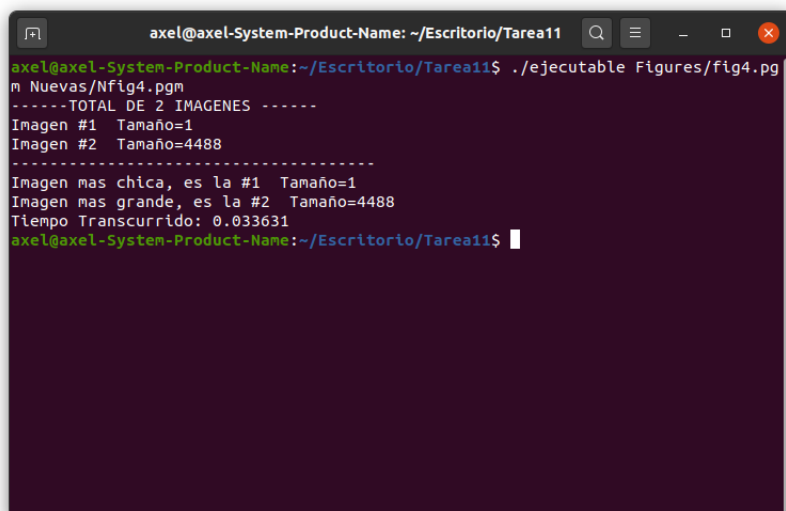
Para la primera imagen se encontraron 16 figuras ajenas, donde la mas grande tenia 4587 píxeles y la mas pequeña tenia 230, el tiempo de ejecución fue de 0.0611 segundos.

```
axel@axel-System-Product-Name: ~/Escritorio/Tarea11
Imagen #5 Tamaño=587
Imagen #6 Tamaño=587
Imagen #7 Tamaño=406
Imagen #8 Tamaño=441
Imagen #9 Tamaño=425
Imagen #10 Tamaño=419
Imagen #11 Tamaño=318
Imagen #12 Tamaño=612
Imagen #13 Tamaño=521
Imagen #14 Tamaño=758
Imagen #15 Tamaño=703
Imagen #16 Tamaño=513
Imagen #17 Tamaño=456
Imagen #18 Tamaño=556
Imagen #19 Tamaño=492
Imagen #20 Tamaño=800
Imagen #21 Tamaño=431
Imagen #22 Tamaño=672
Imagen #23 Tamaño=470
-----
Imagen mas chica, es la #11 Tamaño=318
Imagen mas grande, es la #20 Tamaño=800
Tiempo Transcurrido: 0.089732
axel@axel-System-Product-Name:~/Escritorio/Tarea11$
```

En la segunda imagen se encontraron 23 figuras ajenas, donde la mas grande tenia 800 píxeles y la mas pequeña tenia 318, el tiempo de ejecución fue de 0.0897 segundos.

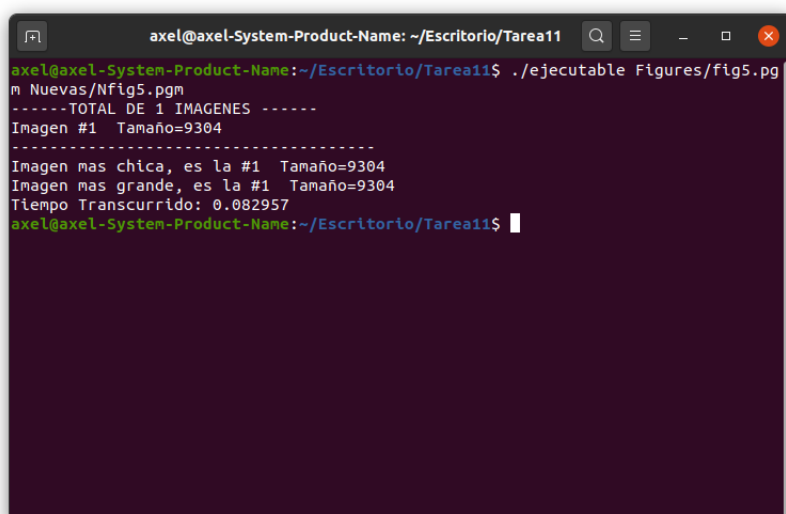
```
axel@axel-System-Product-Name: ~/Escritorio/Tarea11
axel@axel-System-Product-Name:~/Escritorio/Tarea11$ ./ejecutable Figures/fig3.pg
m Nuevas/Nfig3.pgm
-----TOTAL DE 9 IMAGENES -----
Imagen #1 Tamaño=39800
Imagen #2 Tamaño=4896
Imagen #3 Tamaño=5700
Imagen #4 Tamaño=17292
Imagen #5 Tamaño=1440
Imagen #6 Tamaño=576
Imagen #7 Tamaño=10201
Imagen #8 Tamaño=27390
Imagen #9 Tamaño=13920
-----
Imagen mas chica, es la #6 Tamaño=576
Imagen mas grande, es la #1 Tamaño=39800
Tiempo Transcurrido: 0.13686
axel@axel-System-Product-Name:~/Escritorio/Tarea11$
```

Para la tercera imagen se encontraron 9 figuras ajenas, donde la mas grande tenia 39800 píxeles y la mas pequeña tenia 576, el tiempo de ejecución fue de 0.1368 segundos.



```
axel@axel-System-Product-Name: ~/Escritorio/Tarea11
axel@axel-System-Product-Name:~/Escritorio/Tarea11$ ./ejecutable Figures/fig4.pg
m Nuevas/Nfig4.pgm
-----TOTAL DE 2 IMAGENES -----
Imagen #1  Tamaño=1
Imagen #2  Tamaño=4488
-----
Imagen mas chica, es la #1  Tamaño=1
Imagen mas grande, es la #2  Tamaño=4488
Tiempo Transcurrido: 0.033631
axel@axel-System-Product-Name:~/Escritorio/Tarea11$
```

En la quinta imagen se encontraron 2 figuras ajenas, donde la mas grande tenia 4488 píxeles y la mas pequeña tenia 1, el tiempo de ejecución fue de 0.0336 segundos.



```
axel@axel-System-Product-Name: ~/Escritorio/Tarea11
axel@axel-System-Product-Name:~/Escritorio/Tarea11$ ./ejecutable Figures/fig5.pg
m Nuevas/Nfig5.pgm
-----TOTAL DE 1 IMAGENES -----
Imagen #1  Tamaño=9304
-----
Imagen mas chica, es la #1  Tamaño=9304
Imagen mas grande, es la #1  Tamaño=9304
Tiempo Transcurrido: 0.082957
axel@axel-System-Product-Name:~/Escritorio/Tarea11$
```

La ultima imagen tuvo una sola figura de 22064 píxeles, el tiempo de ejecución fue de 0.0829 segundos.

4. Discusión

Para las imágenes dadas, estas eran diferentes conjuntos conexos en color blanco, sobre un fondo negro, por lo que la matriz de los píxeles tenia en su mayoría valores de 0 y 255, donde el 0 representa al color negro y el 255 al color blanco, había algunos valores intermedios que eran tonos de grises que se usaban para las curvaturas de algunas figuras. Por lo que la relación que había entro los píxeles para determinar si el vecino de uno pertenecía al conjunto conexo o no, es que tuvieran un valor diferente a 0, de esta forma nuestros conjuntos conexos serian las figuras blancas que había sobre el fondo. Como dijimos las imágenes no eran de solo dos colores sino que había algunos píxeles de color gris esto no representaba un problema al momento de encontrar cuantas figuras conexas había dentro de la imagen, pues estos píxeles eran vecinos de píxeles blancos, por lo que

formaban parte del conjunto conexo, excepto para la imagen 5, ya que si tomábamos los diferentes valores de los grises el algoritmo encontraba aproximadamente 199 figuras conexas, la mayoría de estas de tamaño 1, lo que no era deseado, pues se esperaba encontrar un solo conjunto (la vena del ojo), por lo que al leer los valores de cada píxel, se tenía en cuenta a que valor eran mas cercanos, si al 0 que representa el color negro o al 255 que era el color blanco, y al que fueran mas cercanos seria el valor que tomarían, haciendo que nuestra imagen fuera de solo dos colores, esto no afecto el numero de conjuntos conexos encontrados para las otras imágenes pero si sus tamaños, finalmente para la imagen numero 5 se logro encontrar un solo conjunto conexo.

5. Conclusión

En esta ocasión se trabajo con el sistema de 8 vecinos, aunque también existe un algoritmo considerando solo 4 vecinos que son los píxeles de arriba, abajo, izquierda y derecha del píxel, este algoritmo podría delimitar mas conjuntos conexos en una imagen de los que encuentra el sistema de 8 vecinos, esto no implica que el sistema de 8 vecinos falle o no encuentre algún conjunto sino que la forma en la que trabaja el sistema de 4 vecinos implica que conjuntos que están unidos solamente por las esquinas son tomados como dos conjuntos diferentes, mientras que para 8 vecinos son solo uno, así que dependiendo como queramos verlo deberíamos ocupar uno o el otro.

En la implementación del algoritmo se ocupo una matriz adicional no solo para ver los píxeles ya visitados sino, para ver a cual de los diferentes conjuntos hallados pertenecía, esto facilito la implementación del algoritmo, puesto que el uso de pilas o alguna otra estructura para llevar un registro de los píxeles visitados nos haría buscar dentro de toda esta por las coordenadas de un píxel para saber si ya se había visitado o no.