

Universidad de Guadalajara.

Centro Universitario de Ciencias Exactas e Ingenierías.

División de Electrónica y Computación.

Sem. de solución de problemas de algoritmia.

Planteamiento.

Diseñe un sistema computacional con la capacidad de seleccionar y analizar imágenes que contienen círculos negros. Asuma que en las imágenes solamente hay círculos, y que todos son negros. Considere también, que las imágenes a analizar tienen ruido, los extremos de los círculos no son absolutamente negros. Todos los círculos están separados por, al menos, 5 pixeles blancos.

El sistema debe ser capaz de identificar circulos, para cada uno, es imprescindible conocer la posición (cartesiana) en la que se encuentra. Los círculos identificados deberán tener un identificador único (un número, o una letra, por ejemplo) para relacionar los círculos de la imagen y la lista con la información de los círculos. La lista de círculos podrá clasificarse, de mayor a menor area, de menor a mayor posición, respecto al eje x, de menor a mayor posición, respecto al eje y.

Hay que tomar en cuenta que se puede analizar cualquier cantidad de imágenes en una sola ejecución (secuencialmente, no simultáneamente), por lo que el programa puede repetir su análisis, sin necesidad de tener que volver a abrir el programa.

La secuencia de ejecución es la siguiente:

- 1. Seleccionar imagen a analizar.
- 2. Mostrar imagen seleccionada.
- 3. Analizar la imagen.
- Mostrar gráficamente la identificación de los círculos y mostrar la lista de círculos, con sus características: Posición del centro del círculo y radio.

Requerimientos funcionales

Paradigma de programación: P.O.O.

El sistema debe contener una interfaz gráfica intuitiva.

- Seleccionar imagen. (10 puntos)
- Mostrar la imagen a analizar. (10 puntos)
- Mostrar gráficamente los círculos identificados y la posición de sus centroides. (30 puntos)
- Almacenar y mostrar las características de los círculos. (20 puntos).
- Ordenar por tamaño (bubble sort). (10 puntos)
- Ordenar por eje x (selection sort). (10 puntos)
- Ordenar por eje y (insertion sort). (10 puntos)

Objetivos

- 1. Crear interface grafica para la interacción del usuario
- 2. Detectar círculos negros en una imagen con fondo blanco.
- 3. Clasificar círculos:

Por tamaño

Por orden de aparición en eje x

Por orden de aparición en eje y

Marco teórico

Representación de imágenes en un sistema computacional.



Imagen

Imagen con detección de circulo y centro



Algoritmos de ordenamiento:

for i in range(1, len(circles)): for j in range(i - 1, -1, -1):

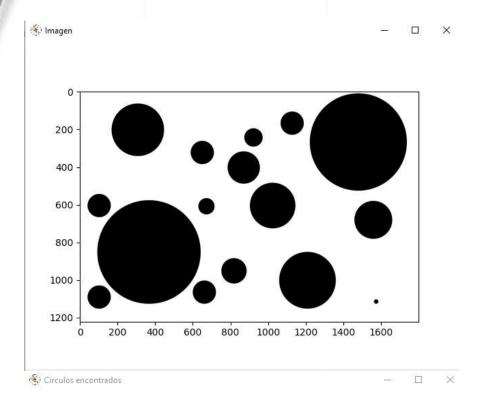
```
Selection sort.
def sort_by_x(circles):#Selection sort
  for i in range(len(circles)):
     min_idx = i
     for j in range(i+1, len(circles)):
        if circles[min_idx][1] > circles[j][1]:
           min_idx = j
     circles[i], circles[min_idx] = circles[min_idx], circles[i]
  x = [ (circles[i][0],circles[i][1]) for i in range(len(circles)) ]
  x = [ str(f''\{x[i][0]\} Eje X = \{x[i][1]\}'') for i in range(len(x))]
  x = "\n".join(x)
  return x
       Bubble sort
def sort_by_area(circles):
  for i in range(len(circles)-1):
     for j in range(0, len(circles)-i-1):
        if circles[j][3] > circles[j+1][3] :
           circles[j], circles[j+1] = circles[j+1], circles[j]
  areas = list(reversed([ (circles[i][0],round((math.pi*(circles[i][3]**2)),2)) for i in
range(len(circles))]))
  areas = [str(f"{areas[i][0]} Area = {areas[i][1]}") for i in range(len(circles))]
  areas = "\n".join(areas)
  return áreas
       Insertion sort
def sort_by_y(circles):#Insertion sort
```

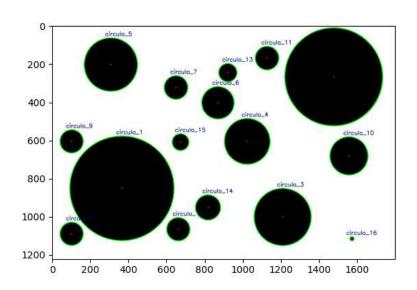
Desarrollo

Iniciamos un cona ventana la cual tiene dos botones los cuales uno sirve para seleccionar una imagen y mostrarla al usuario, en caso de que no se seleccione una imagen le dira al usuario por medio de otra ventana que no selecciono una imagen, en caso de que si la selecciono la mostrara al usuario, consiguiente podrá presionar el botón de analizar el cual mostrara la imagen selección resaltando los centros de los círculos asi como su contorno y también asignara una etiqueta a cada uno de los círculos para identificar cual es el circulo del cual estamos hablando en los ordenamientos.

Pruebas y resultados







Ordenamientos		– 🗆 ×
Area	Eje X	Eje Y
Circulo 1 Area = 235858.21 Circulo 2 Area = 207499.05 Circulo 3 Area = 70685.83 Circulo 5 Area = 61575.22 Circulo 4 Area = 45238.93 Circulo 10 Area = 31415.93 Circulo 6 Area = 22698.01 Circulo 14 Area = 13684.78 Circulo 11 Area = 12076.28 Circulo 7 Area = 12076.28 Circulo 7 Area = 11689.87 Circulo 9 Area = 11689.87 Circulo 8 Area = 11689.87 Circulo 13 Area = 7238.23 Circulo 15 Area = 6082.12 Circulo 16 Area = 380.13	Circulo 8 Eje X = 102 Circulo 9 Eje X = 102 Circulo 5 Eje X = 308 Circulo 1 Eje X = 366 Circulo 7 Eje X = 650 Circulo 12 Eje X = 662 Circulo 15 Eje X = 674 Circulo 14 Eje X = 818 Circulo 6 Eje X = 870 Circulo 13 Eje X = 922 Circulo 4 Eje X = 1024 Circulo 11 Eje X = 1128 Circulo 3 Eje X = 1210 Circulo 2 Eje X = 1478 Circulo 10 Eje X = 1558 Circulo 16 Eje X = 1574	Circulo 11

Conclusiones

En esta actividad logre detectar círculos encontrando su centro y marcando su perímetro así como también sacar su perímetro y agregar a una lista el perímetro de cada uno de ellos para después sacar su área y en la lista ordenarlo de ascendente a descendente por medio de uno de los tres métodos deseados por el profesor, así como también en base al eje x, y el eje y ordenarlos por medio de otros dos métodos de ordenamiento.