CC5508: Procesamiento y Análisis de Imágenes
Tarea 3: Detección de Objetos

Prof. José M. Saavedra

Octubre 2020

1. Objetivo

El objetivo de esta tarea es familiarizarse con la manipulación de imágenes a través de algoritmos básicos para detectar objetos basados en atributos de color y forma.

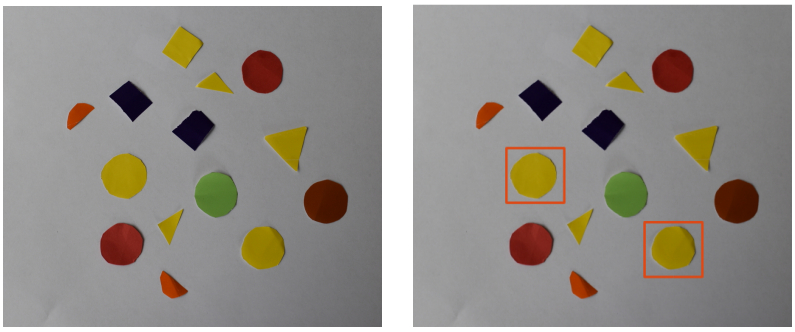
2. Descripción

La tarea consiste de múltiples etapas con el objetivo final de detectar cierto tipo de objetos a partir de características de color y forma. Dada una imagen de entrada, como la que aparece en Figura 1-A, que contiene diversos objetos diferenciados por color y forma, se pide desarrollar un programa en *Python* que permita identificar o detectar objetos de color AMARILLO y forma CIRCULAR, así como se muestra en la Figura 1-B.

Para resolver el problema planteado se pide desarrollar las siguientes etapas:

1. **Discriminar objetos amarillos**. Deberá crear una función que reciba como entrada una imagen RGB y generé una imagen binaria donde los objetos de color amarillo se representen con 1's y el resto de la imagen quede representada con 0's.

2. **Extraer componentes conexos**. Crear una función que reciba una imagen binaria y genere una lista de componentes conexos, en donde cada componente ζ quede representado por:



(A) (B) ’

Figura 1: (A) Imagen de entrada. (B) Imagen resultante.

a) Un identificador $id \in \{1, \cdots, C\}$, donde C es la cantidad total de componentes detectados.

b) Una lista $lpoints_\zeta = \{(i,j)|(i,j) \in \zeta\}$. Es decir $lpoints_\zeta$ es la lista con los puntos que forman el componente ζ

c) Una lista $lboundary_\zeta = \{(i,j)|(i,j) \in \Gamma\zeta\}$, donde $\Gamma(\cdot)$ es la función de extracción de bordes de un componente conexo.

d) Una lista $lbb_\zeta = [x, y, w, h]$, representando el rectángulo mínimo que cubre a ζ, donde (x, y) es el punto inicial, w es el ancho y h es el alto del rectángulo.

Nota: Para implementar esta sección debe seguir los algoritmos de detección de componentes basados en el recorrido de pixeles vecinos, que ocupan estrategias como *Depth First Search* o *Breadth First Search*. Asimismo, para extraer el borde de un componente deberán seguir el algoritmo de topología digital visto en la sala de clases.

3. **Reconocimiento de objetos circulares**. Se pide escribir una función que determine si un componente tiene forma circular o no. **Sugerencia**: Utilizando tres puntos del borde de la componentes, se puede estimar una circunferencia. Con esto podemos determinar el error entre la circunferencia estimada y el componente (hint: utilice la distancia del centro estimado al borde). Si el $error < t$, entonces la componente se etiqueta como circular. La ecuación general de la circunferencia se define como $x^2 + y^2 + Dx + Ey + F = 0$, siendo el centro $= (-D/2, -E/2)$ y el radio $= \sqrt{(D^2 + E^2)/4 - F}$.

3. Detalle Técnico

- Crear un programa en Python que reciba como entrada una imagen RGB.
- El programa debe generar los siguiente archivos:
  - Un imagen de salida marcando los objetos detectados (amarillos circulares) con su respectivo bounding-box.
  - Una imagen binaria representando los componentes amarillos.
  - Un archivo de texto con toda la información de las componentes detectadas, según lo indicado en la sección anterior.

4. Esquema de Informe

1. **Abstract o Resumen**: es el resumen del trabajo.
2. **Introducción**: se describe el problema y el contexto de aplicación. (10 %)
3. **Desarrollo**: se describe el diseño e implementación del programa. (40 %)
4. **Resultados Experimentales y Discusión**: se debe presentar los resultados y hacer un análisis de los mismos. (40 %)
5. **Conclusiones** (10 %)

5. Restricciones y Condiciones

1. NO se aceptan tareas sin informe.
2. **Todas las funciones antes indicadas deben ser implementadas por el propio alumno, no se aceptan uso de librerías externas que resuelvan las tareas solicitadas. Pueden utilizar todas el código disponible en el github del ramo, incluyendo las relacionadas con componentes conexos.**
3. NO hay atrasos.
4. La tarea es individual.

5. Poner mucho esfuerzo en la redacción del informe.

6. La implementación se realizará en Python (de preferencia 3.6).

7. Junto a esta tarea se deja disponible un cojunto de imágenes de pruebas, pero ustedes pueden agregar más ejemplos.

6. Entrega

La entrega se realiza por u-cursos hasta el domingo 15 de noviembre, 2020, 23:50 hrs. Se debe incluir:

1. Código fuente (en Ptyhon)

2. Informe