

Modelado y programación

Segundo proyecto

Integrantes:

- Arturo Gonzalez Peñaloza.
- Asahel Said Main Cerezo.
- Emilio Arsenio Raudry Rico

Definición del problema

Entender el problema: Queremos catalogar las figuras geométricas presentes en un archivo de mapa de bits. Esto soluciona el problema

Arsenal: Lenguaje de programación: python3

Requisitos funcionales: Como salida queremos el valor RGB en hexadecimal de cada figura seguido de la categoría a la que pertenece. Se debe llevar a cabo la detección de cada figura presente en la imagen

Requisitos no funcionales: El programa debe ser eficiente y resistente a fallas.

Selección de la mejor alternativa

Optamos por python3 más que nada por la sencillez de su sintaxis y por las funciones ya implementadas dentro de sus bibliotecas que nos permitieron ahorrar cierto trabajo. Entre ellas OpenCv fue una de las más importantes, ya que nos permitió pasar el mapa de bits a un arreglo que nos permitió tener un control sobre los píxeles de la imagen a lo largo del programa.

Análisis del problema

Inicialmente el programa lee el mapa de bits dado y lo pasa a un arreglo de tres dimensiones, correspondientes a la fila, columna y valor BGR de cada píxel.

Posteriormente recorreremos el arreglo de izquierda a derecha y de arriba a abajo, de manera que, al encontrar un píxel cuyo color sea distinto al fondo, calculamos su centro a partir del promedio de sus coordenadas.

A partir de aquí cada resultado que obtengamos por figura en los siguientes métodos lo asociamos a su color correspondiente por medio de un diccionario.

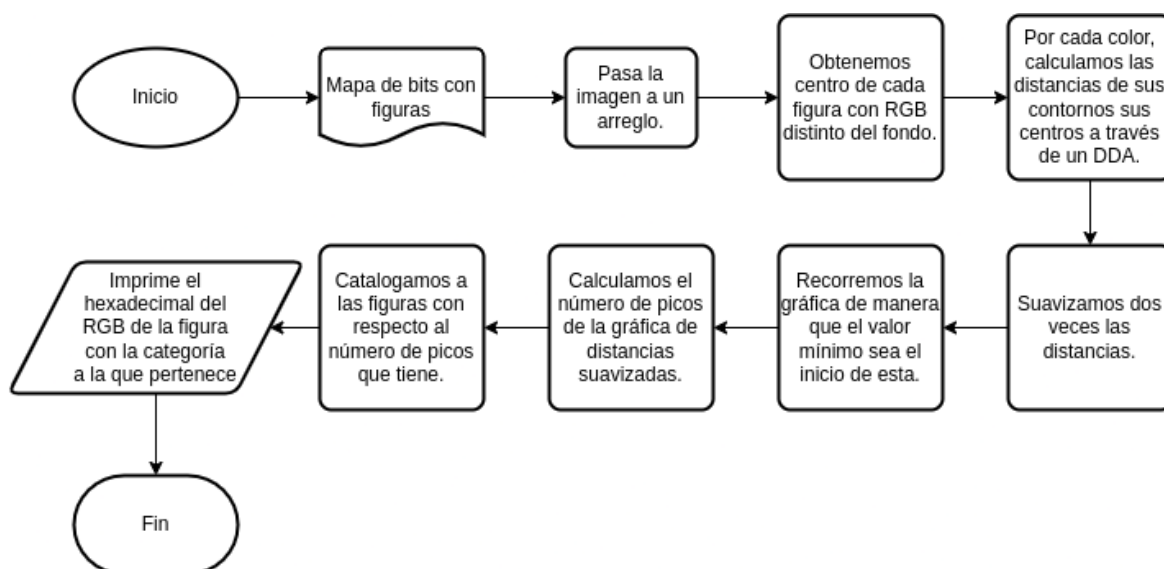
Por cada color identificado en la imagen obtenemos las distancias de su centro con los píxeles que forman parte del contorno de la figura correspondiente a través de un DDA.

Ejecutamos un proceso de suavizado dos veces por cada lista de distancias, de manera que aquellos valores que estén sobre el promedio multiplicado por 1.45 tengan un margen de suavizado menor que al de los que están por debajo (7 y 12 respectivamente).

Tomamos el valor mínimo de las distancias suavizadas para que esté a la cabeza de la lista, y que así no haya cortes en su gráfica.

Luego recorremos las distancias ya suavizadas valor por valor, de manera que al encontrar un pico (el cual se identifica por medio de un método auxiliar que compara los valores contiguos a la izquierda y a la derecha del valor actual dentro de un determinado rango) añadimos su índice correspondiente, a fin de que al llegar al final, contemos los índices almacenados que corresponden a los vértices de las figuras.

Finalmente, a cada figura le asignamos su categoría correspondiente de acuerdo al número de vértices que tiene.



Piensa a futuro.

Consideramos que el suavizado de las distancias entre el contorno de la figura con su centro puede ser digno de cierto mantenimiento. Si tenemos figuras no regulares de tipo 'X', puede que la gráfica correspondiente a sus distancias se vea alterada durante el proceso de suavizado, de manera que se pierdan ciertos picos, dándonos una cantidad de vértices no tan precisa.

Para nosotros un precio entre \$21,000-\$24,000 puede cubrir el tiempo y esfuerzo empleados en el proyecto, de esta manera a cada integrante del equipo le corresponderá un pago entre \$7,000-\$8,000. Adicionalmente, durante la elaboración del proyecto nosotros estuvimos investigando, ideando y probando qué algoritmo sería el más eficaz para ciertos métodos, un apartado que forma parte del proceso de programación y que se debería de valorar.