Segundo Parcial

Ajuste de una imagen para visualizarla en un arreglo de LEDs con menor o mayor resolución

David Correa Ochoa

Y

Francis David Roa Bernal

Departamento de Ingeniería Electrónica y Telecomunicaciones
Universidad de Antioquia
Medellín
Septiembre de 2021

Índice

1. Descripción del problema

Se debe extraer la información RGB de una imagen y modificar dicha información para que por medio de varios arreglos de LEDs RGB o NeoPixel en el simulador TinkerCad se pueda visualizar la misma imagen con una menor o mayor resolución.

2. Propuesta de solución

A continuación se proponen posibles soluciones al problema descrito en la sección anterior.

- Para el submuestreo se propone;
  - Dividir la información RGB de la imagen en una cantidad de bloques igual a la resolución del ancho y el alto que se desea lograr, para luego tomar la información RGB en los bloques y extraer un promedio de la misma. En la figura 1(1) se muestra un ejemplo meramente ilustrativo en la que se usan bloques de 5 píxeles para reducir una ïmagen"de 5*3 píxeles a un único píxel.
  - En caso de que la información RGB de la imagen no se pueda dividir en la cantidad deseada de bloques, se superpondrán los primeros bloques(el inicio de un bloque está dentro del bloque anterior) para obtener la cantidad de bloques requerida
- Para sobremuestreo se propone;
  - Copiar las filas y columnas de la información RGB de la imagen y clonar la información de cada fila y columna hasta alcanzar la resolución deseada.

3. Clases implementadas

Luego de una revisión de los ejemplos proporcionados en clase y una breve discusión se optó por utilizar las clases QImage, Adafruit_NeoPixel y vector.

- La clase QImage del entorno QT se utilizará para obtener los datos de la imagen a modificar. Se utilizaran métodos como el height(), width(), pixelcolor(), red(), blue() y green().
- La clase vector del entorno QT se utilizará para guardar los datos proporcionados por QImage en vectores, generando en el algoritmo 6 vectores, 3 de uso temporal y 3 que guardaran los datos de la imagen ya modificada, estos últimos son los que se imprimirán en un archivo de texto para usarlos en Arduino.

- La clase Adafruit_NeoPixel del entorno arduino se utilizará en el código de Arduino donde permitirá a través del método setPixelColor() para manipular de manera sencilla de las tiras de neopixel solo necesitando 4 datos; posición (x,y) del led y los valores de intensidad de cada RGB(rojo,verde,azul).

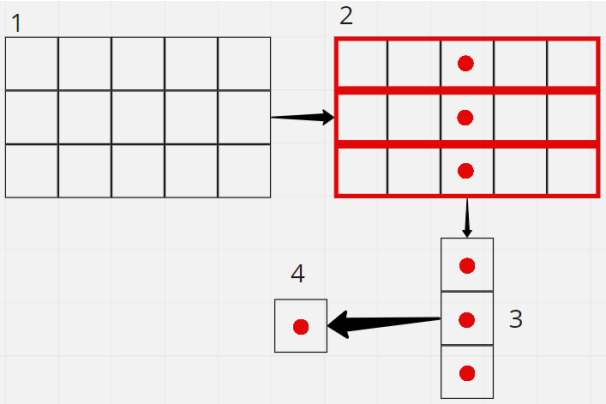

Figura 1: Ejemplo

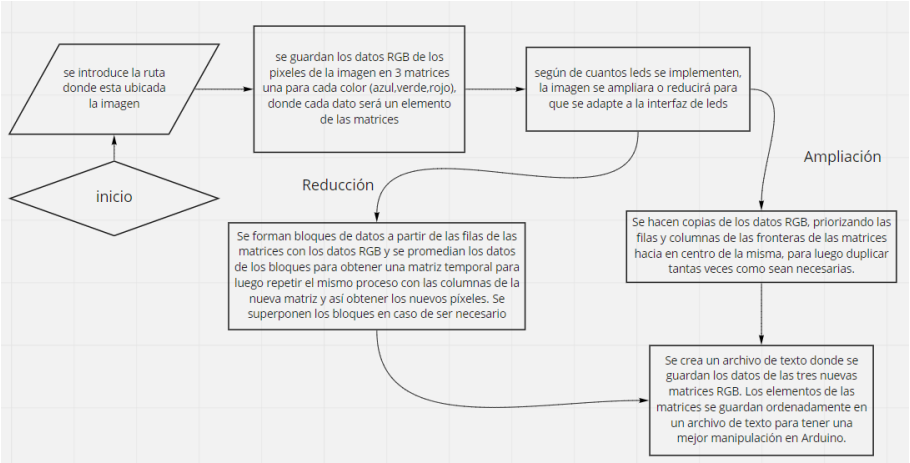

Figura 2: Esquema de tareas

4. Consideraciones a tomar en cuenta

Luego de una discusión se propusieron las siguientes consideraciones a tener en cuente a la hora de realizar la implementación

- La reducida capacidad de memoria de la tarjeta Arduino.
- dependiendo de la clase de leds a implementar aumentará o disminuirá la complejidad de código.
- el segmento de código creado en Qt debe ser adaptado para el entorno Arduino.

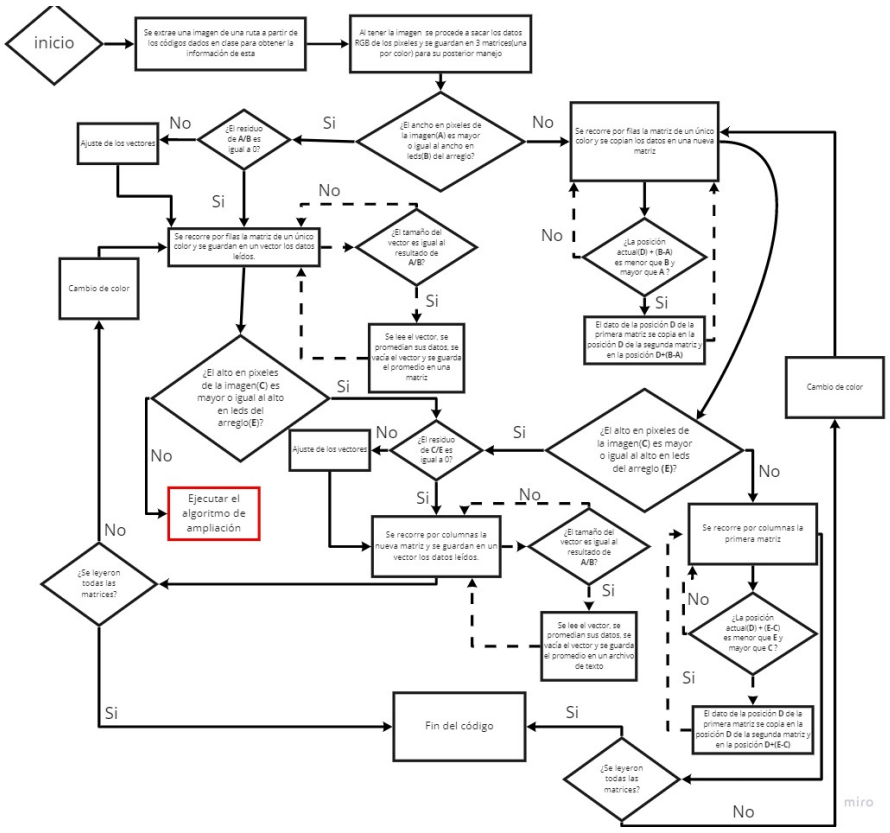

Figura 3: Algoritmo