

Informe de implementacion

Parcial 2 de informatica.

Cristian Martinez De La Ossa

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

Índice

1. Introducción	2
2. implementación de las ideas	2
3. Clases y Contenedores Usados	3
4. Estructura del circuito	4
5. Conclusión	4

1. Introducción

Después de haber analizado el problema a resolver en este parcial, pude percibir varias estrategias que podían darle solución a dicho problema, así que comencé primeramente a realizar pruebas de escritorio para validar mis ideas. Entre las diferentes ideas que se me pasaban por la mente, dos de ellas fueron muy acertada a lo que estaba buscando. A continuación, voy a describir cuales fueron esas ideas y como las implemente.

2. implementación de las ideas

SUBMUESTREO.

La primera idea que tuve fue con respecto al submuestreo, mi propuesta fue cargar la imagen primeramente, para ello hago uso de la clase `QImage`, acto seguido cargo cada una de las tres capas de colores rojo, verde y azul en una matriz con dimensiones $m \times n$, donde m y n las puedo determinar gracias a los métodos `height()` y `width()` de la clase `QImage`.

Una vez que ya tengo la matriz de los colores rojo, verde y azul, le aplico la siguiente técnica a cada una.

Primero divido la matriz en pequeños bloques que tengan un área rectangular con las siguientes dimensiones:

Alto = (CANTIDAD DE FILAS DE LA MATRIZ)/16

Ancho = (CANTIDAD DE COLUMNAS DE LA MATRIZ)/16

Tal como se muestra en la siguiente figura.

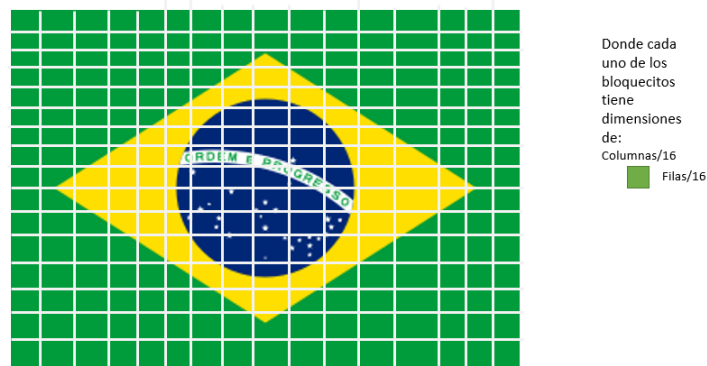


Figura 1: Imagen para ilustrar metodo

Luego recorro cada bloquecito con un doble `for()` y cuento cuantos pixeles tiene. Al mismo tiempo voy sumando las intensidades de cada color en dicho bloque para así sacar el promedio de ese respectivo color en ese bloque. Finalmente el promedio que obtenga para cada color en cada bloque, será lo que voy a colocar en mi matriz de leds de tinkercad.

SOBREMUESTREO.

Para el sobremuestreo lo que hago es tomar las matrices anteriormente descritas, la del rojo, verde y azul, y duplicar sus filas y columnas hasta que tanto las filas como las columnas sean mayores a 16(Puesto que mi matriz de leds será de 16x16). Eso lo hago con usando bucles while y for, una vez que se ha cumplido ese objetivo entonces aplico el método de submuestreo descrito anteriormente, el cual me arroja la matriz final de 16x16.

3. Clases y Contenedores Usados

En mi código de qt hice uso de las siguientes clases y contenedores para realizar las diferentes tareas que implicó el proceso descrito anteriormente.

Use la clase `QImage` para cargar y leer la imagen, de esa clase utilice el método `pixelcolor`, también hice uso del contenedor `vector`, con el cual cree un vector que contiene a otros vectores, lo cual me permite tener una matriz donde las filas son cada vector que hay dentro del vector superior.

Además utilice la clase `iostream` que siempre es necesaria y la clase `fstream` para crear el archivo de texto donde se guarda la matriz para los leds.

De igual forma hice uso de una clase que yo mismo cree llamada `escalarimagen`, en esa clase tengo tres atributos (dos `int` y uno de tipo `QImage`), un método llamado `reducirmatrices` que se encarga de leer los píxeles de la imagen, contruir una matriz para los colores, reducir dicha matriz utilizando el método anteriormente descrito y al final construye el archivo con los datos que van para `tikercad`.

En mi clase hay un constructor que recibe la imagen que es cargada en el `main` y la almacena en un atributo de la clase para que pueda ser usada por el método `reducirmatrices`, también se inicializan los demás atributos, se crea el archivo de texto para tenerlo listo para ser usado más adelante y se invoca a la función `reducirimagen` para que haga su trabajo.

4. Estructura del circuito

Mi circuito lo arme con tiras de 16 neopixeles para así sacar una matriz de 16X16, básicamente utilice 16 tiras, un arduino el cual configure en el puerto 0, para salida. Cada tira de led está unida a la anterior de acuerdo a la documentación que hay sobre las tiras y de acuerdo a lo que he aprendido en clases con los profesores.

Con respecto a inconvenientes que haya tenido al resolver este trabajo, creo que es único inconveniente más relevante fue que al principio no me encendían los leds de la matriz, pero me di cuenta que era porque debía cambiar el tipo de datos de mi matriz, que lo tenía en int y debía ponerlo en byte, de esa forma se solucionó el inconveniente.

5. Conclusión

Pienso que este parcial ha sido un reto para demostrar mis conocimientos adquiridos hasta el momento y considero que es muy importante para mi crecimiento profesional. Creo que no estuvo tan fácil, pero tampoco muy difícil, además me motivó mucho a investigar sobre el tema de cómo están hechas las imágenes. En general considero que hice un buen trabajo.