

Parcial 2.

Informa2 S.A.S.

Holman Londoño Restrepo

Análisis y diseño.

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

Índice

1. Análisis y consideraciones.	2
2. Esquema para el desarrollo del algoritmo.	2
3. Algoritmo.	3
4. Consideraciones.	3

1. Análisis y consideraciones.

Para la solución de este parcial primero tuve que comprender más a fondo como era que funcionaba el procesamiento de una imagen computacionalmente y como se representaba la información contenida en las imágenes de una forma más nativa para así poder jugar con estos datos y lograr modificar el tamaño satisfactoriamente, considero que el mayor problema que representa este reto es hacer el submuestreo de las imágenes debido a que es un proceso mediante el cual se pierde información, por lo tanto debemos encontrar la forma de hallar los pixeles menos valiosos para así poder comprimirlos y no terminar eliminando los pixeles con más relevancia para que de esta forma la imagen tenga el menor cambio posible, en el caso del sobre muestreo seria lo contrario, no se perdería información sino que habría que multiplicar los datos contenidos en cada pixel X cantidad de veces para de esta forma lograr aumentar el tamaño de manera exitosa. En cuanto al montaje del circuito y el código del Arduino no he entrado mucho en detalles ya que quiero solucionar primero el problema del procesamiento de las imágenes, por el momento solo tengo definido la forma en como almacenaré la información arrojada por el programa hecho en Qt y será mediante una matriz de $3 \times N$, donde N será el número de leds que tendrá mi montaje, de esta manera guardar en cada fila el código RGB de cada pixel, por ejemplo en la columna 0 se almacenara el valor de la tonalidad del rojo "R", en la columna 1 el valor de la tonalidad verde "G" y en la columna 2 o última columna el valor de la tonalidad en azul "B", así el Arduino leerá esta matriz fila por fila y enviara estos datos a los leds respectivamente; En relación a el montaje, aún no he definido el tamaño de mi matriz de leds pero tengo claro que será rectangular, ósea que tendrá más columnas que filas, debido a que esta es la proporción en la que están diseñadas la mayoría de banderas nacionales.

Por ahora solo he considerado el uso de una clase la cual será "Imagen", en esta clase se implementaran las funciones necesarias para hacer la lectura y escritura de los datos de la imagen, como lo puede ser el valor RGB de cada pixel y el tamaño, además de que contendrá las funciones que requiera para hacer el submuestreo o el sobremuestreo de cada imagen, este código lo implementaré usando programación orientada a objetos para de esta manera poder crear en el main un puntero que haga referencia a la clase imágenes y así poder usar todos los atributos de esta clase y así tener un código más optimizado.

2. Esquema para el desarrollo del algoritmo.

El esquema que definí para el desarrollo del algoritmo fue el siguiente:

1. Reconocimiento del problema: En esta parte me enfoqué en entender a fondo cual era el problema o desafío que me estaban planteando para de esta manera poder saber que era lo que tenía que hacer.
2. Búsqueda de información: Esta sección del desarrollo consistía en buscar por diferentes medios como internet, libros y revistas científicas, como funcionaba el procesamiento digital de imágenes y como era implementada esta técnica para modificar el tamaño de las imágenes, así una vez identificada la lógica podía implementarla en mi programa.
3. Desarrollo del algoritmo: Acá procedo a definir la lógica que usare en mi programa para la solución del problema, como por ejemplo cual será el método que usaré para realizar el submuestreo o el sobremuestreo y que clases implementaré.
4. Codificación: Este es de los últimos pasos, acá procedo a plasmar todo lo que hice durante el desarrollo del algoritmo y es uno de los pasos más importantes ya que es donde me doy cuenta de que fallos tiene la lógica que quiero implementar y tendré que buscar la forma de solucionar esto, ya sea modificando pequeños aspectos o cambiando la lógica por completo.
5. Montaje del circuito y codificación de Arduino: El último paso y también uno de los que tiene más relevancia ya que aquí es donde realizo la conexión entre el código hecho en Qt y la parte de hardware, en esta sección debo poner suma importancia a la carga computacional ya que el Arduino cuenta con menos capacidad de procesamiento que un computador y si lo sobrecargo podría desencadenarse algunos errores que no permitirían que mi programa funcione correctamente.

3. Algoritmo.

El algoritmo que diseñe consiste en primeramente leer la imagen, esto tengo pensado hacerlo recorriendo pixel por pixel la imagen y almacenar sus datos en un contenedor, una vez tenga estos datos guardados procederé a comparar el tamaño original de la imagen con el tamaño final para saber si debo aplicar la técnica de submuestreo o sobremuestreo para luego llamar a la función correspondiente, en el caso del sobremuestreo pienso multiplicar por X uno a uno los pixeles de la imagen original para así lograr aumentar el tamaño de esta, esto siempre que el tamaño final sea divisible entre el tamaño original y X tomara el valor del resultado de esta operación, si los tamaños no son divisibles se hace el proceso anterior y para rellenar los pixeles faltantes se duplicaran los pixeles del medio en forma de X hasta que se llegue al tamaño deseado; la técnica de submuestreo es un poco más complicada ya que es un proceso mediante el cual se perderá información de la imagen, por lo tanto hay que seleccionar correctamente los pixeles que se van a eliminar para que de esta forma no cambie la información que se quiere reflejar en la imagen, para desarrollar esta técnica correctamente debo empezar a comprimir información por los extremos de la imagen, esto lo haré tomando X cantidad de pixeles vecinos y sacando un promedio del valor RGB, este promedio será el valor RGB del nuevo pixel en la imagen resultante, hay que tener en cuenta que se debe hacer este proceso para cada valor del código RGB, por último se guardan los datos de cada pixel de la imagen resultante en un archivo "TXT" para que esta información pueda ser utilizada en el Arduino.

4. Consideraciones.

El método que voy a usar para el sobremuestreo y submuestreo tiene varios aspectos a considerar, en especial el de submuestreo ya que mediante este proceso se perderá bastante información de la imagen por lo tanto hay que ser extremadamente cautelosos a la hora de definir que pixeles omitir, para lidiar con esto tengo pensado iniciar el proceso por los bordes ya que allí, en la mayoría de banderas, es donde hay menos variedad en los datos de los pixeles, aparte debo realizar muy bien el código para que seleccione correctamente los pixeles con menos relevancia para que de esta manera no elimine un pixel crucial de la imagen a modificar. En otro aspecto que debo poner suma atención, es a la hora de realizar el código de Arduino hacerlo de la manera más optima posible para no sobrecargar el procesador de este, debido a que el procesador que tiene la tarjeta Arduino Uno no es tan avanzado como el de los computadores es muy fácil sobrecargarlo y así provocar fallos en el funcionamiento de este o hasta podría llegar al extremo de no correr el código, por esto es que no haré una matriz de leds muy grande ya que esto implicaría tener una matriz de información RGB muy grande aparte se debe tener en cuenta que cada led de las tiras de neo pixel necesita de tres valores para funcionar y esto triplica el espacio que usa en memoria comparado con un led convencional.