

Laboratorio #1

Laura Camila Blanco Gómez
Escuela de Ingeniería
Universidad Sergio Arboleda
Bogotá, Colombia
laura.blanco01@usa.edu.co

Santiago Cáceres Linares
Escuela de Ingeniería
Universidad Sergio Arboleda
Bogotá, Colombia
santiago.caceres01@usa.edu.co

I. INTRODUCCIÓN

En este informe se puede ver el proceso de carga y manipulación de imágenes BMP. Inicialmente se podrá ver como se carga el archivo de imagen BMP en la memoria, donde los datos de los pixeles se copiarán a una matriz del tamaño de la imagen, una vez resuelto este tema se debe mostrar un menú donde la primera opción debe transformar la imagen en una escala de grises y guardar el resultado en un nuevo archivo BMP, la segunda opción debe permitir realizar la operación de convolución en la imagen y el resultado también se debe guardar en un archivo BMP.

A través de este informe se busca aprender y aplicar el procesamiento de imágenes BMP junto con conocimientos adquiridos en la programación en C.

II. MARCO TEÓRICO

Como recapitulación de ciertos de los tópicos tratados durante el laboratorio se parte para el correcto entendimiento de algunos conceptos mencionados en la sección anterior.

Formato BMP: Son archivos de mapa de bits en los que cada dirección es acorde a un código de color, por lo que cada cuadro de la matriz de pixeles cuenta con un color determinado, son documentos poco eficientes pero capaces de mantener buena calidad.

Cada píxel puede ocupar 1 o varios bits dependiendo de la profundidad del color. Estos documentos cuentan con una cabecera los cuales son los primeros 54 Bytes, los primeros dos contiene las letras BM que permiten identificar el archivo en formato BMP, en la cabecera también se encuentran datos como el tamaño del archivo, el alto y ancho de la imagen, numero de planos, tamaño de cada punto, la resolución, etc. Estos bytes usan el sistema little-endian.

La distribución de bytes dentro de la cabecera es de la siguiente manera:

Bytes	Información
0, 1	Tipo de fichero "BM"
2, 3, 4, 5	Tamaño del archivo
6, 7	Reservado
8, 9	Reservado
10, 11, 12, 13	Inicio de los datos de la imagen
14, 15, 16, 17	Tamaño de la cabecera del bitmap
18, 19, 20, 21	Anchura (píxels)
22, 23, 24, 25	Altura (píxels)
26, 27	Número de planos
28, 29	Tamaño de cada punto
30, 31, 32, 33	Compresión (0=no comprimido)
34, 35, 36, 37	Tamaño de la imagen
38, 39, 40, 41	Resolución horizontal
42, 43, 44, 45	Resolución vertical
46, 47, 48, 49	Tamaño de la tabla de color
50, 51, 52, 53	Contador de colores importantes

Escala de grises: Es una escala que se usa de manera digital la cual se logra haciendo que cada uno de los bits de una imagen RGB tengan valores cercanos al gris, este tipo de imagenes se compone principalmente de sombras y grises

Convolución: Es una operación matemática que a partir de dos señales genera una tercera, esta se expresa de la siguiente manera:

$$(f * g)(t) \approx \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$$

El filtrado convolucional sirve en el procesamiento de

imágenes para detectar bordes, para el aumento de nitidez o el desenfoque de las imágenes, para esto se hace uso de la matriz de kernel.

Operador Sobel: Es utilizado en procesamiento de imágenes, especialmente en algoritmos de detección de bordes. Es un operador diferencial discreto que calcula una aproximación al gradiente de la función de intensidad de una imagen. Para cada punto de la imagen a procesar, el resultado del operador Sobel es tanto el vector gradiente correspondiente como la norma de este vector.

Este hace uso de las fórmulas que se encuentran en la figura 3 y 4 para sus cálculos matemáticos, donde lo que busca es calcular aproximaciones a las derivadas, por esto se hace uso de un Kernel vertical y uno horizontal.

III. PROCEDIMIENTO

Se inicia el laboratorio definiendo una estructura de la cabecera del archivo BMP, para esto tomamos en cuenta la tabla del marco teórico, se define también la estructura para pixeles de imágenes BMP donde se crearán variables con respecto al RGB.

Con esto procedemos a hacer uso de la función **fopen** que nos sirve para abrir la imagen en formato BMP, se crea una condición en la que, si el valor que trae es nulo, muestre un mensaje indicando que no se logró abrir el archivo BMP y otra condición en la que se confirma si el archivo es de hecho un BMP, esto se hace revisando los dos primeros Bytes y confirmando que la posición 0 se encuentra una **B** y en la posición 1 se encuentra una **M**.

Una vez abierto el archivo podemos extraer el resto de los datos de la cabecera como el ancho y alto de la imagen, estos nos servirán para poder separar un espacio en memoria para una matriz con el tamaño de la imagen que le cargamos, una vez separado este espacio lee los pixeles del archivo BMP y los va copiando dentro de la matriz que creamos, en este caso es la matriz pixels, una vez copiados los datos se debe cerrar el archivo de la imagen que usamos, esto se hace por medio de la función **fclose**.

Por medio de un **switch** realizamos un menú para poder acceder a las dos opciones de variantes de la imagen que podemos crear, la primera opción llamara la función de escala de grises y la segunda opción llamara la de convolución.

Escala de grises: Para este apartado del laboratorio se hace uso de YUV el cual es un sistema de procesamiento de imagen en color, este modelo YUV define un espacio de color en términos de dos componentes los cuales son:

- Luma
- Cromancia

Los cuales se dividen de la siguiente manera, la letra Y representa el Luma y el UV representa la cromancia, para sistemas monocromáticos se usa generalmente el luma y por esto se decidió usar esta parte del sistema YUV para general

la escala de grises. la fórmula que se usó para calcular la escala de grises es la siguiente.

$$Y = 0,299R + 0,587G + 0,144B$$

Esta fórmula se sacó de las fórmulas de conversión de RGB a YUV [Figure 1], adicionalmente en este método se le da más prioridad al color verde sobre los demás como se demuestra en la formula.

$$\begin{cases} Y = 0,299R + 0,587G + 0,144B \\ U = 0,493(B - Y) \\ = -0,147R - 0,289G + 0,436B \\ V = 0,877(R - Y) \\ = 0,615R - 0,515G - 0,100B \end{cases}$$

Esto equivale a escribir, en forma matricial:

$$\begin{bmatrix} Y' \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,144 \\ -0,147 & -0,289 & 0,436 \\ 0,615 & -0,515 & -0,100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Figure 1: Formulas de conversión de RGB a YUV.

Convolución: En este punto de laboratorio se decidió usar dos funciones más aparte de la que genera la convolución la primera fue **createMatrix** la cual, como indica su nombre, crea y retorna una matriz 3x3 la cual será usada como Kernel de convolución, la otra función adicional es **printKernel** la cual antes de general la convolución muestra el Kernel ingresado. Para este método de procesamiento de imagen se necesita parte del mapa de bit que se maneja al ingresar la imagen punto BMP un apuntador auxiliar el cual guardara la información sobre los cálculos realizados junto a la matriz de convolución para posteriormente pasar estos cambios sobre el mapa de bits original. la fórmula que se usó para este punto del laboratorio fue la siguiente.

$$(j - u + 1) * width + (i - v + 1)$$

Esta fórmula se usa para poder calcular el index que sirve para poder acceder a los pixels de la vecindad del pixel central, en esta formula se tiene en cuenta el ancho de la imagen las coordenadas del mapa de bits [j, i] y las coordenadas de la matriz de convolución [u,v]. Cuando se realiza una convolución de 3D se debe tener en cuenta que se debe dejar el borde exterior del mapa de bits sin cambios como se ve en la siguiente imagen

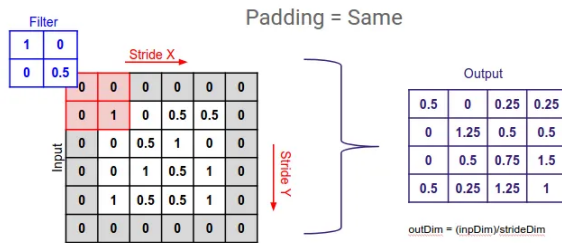


Figure 2: Convolución 3D.

Operador Sobel: Para esta sección, se hace uso de la convolución como base, y se envían dos matrices, una en x y una en y, estas dos actuarán como filtros discretos

$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} -1 & 0 & +1 \end{bmatrix}$$

Figure 3: Primera matriz sobel

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ +1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

Figure 4: Segunda matriz sobel

Se deben declarar estas matrices como arreglos dentro del código, después en el menú agregamos una nueva opción que permita crear la imagen con el operador de Sobel, si esta opción es elegida se llama a la función sobelOperator donde se separa un espacio de memoria y luego se envían las matrices ala función de convolución de Sobel, una vez se tengan los datos se envían al guardado de imagen con la ruta que se quiere del nombre del archivo, por último se debe liberar memoria.

IV. RESULTADOS Y ANÁLISIS DE RESULTADOS

Esta primera imagen es la imagen original que se utilizó para este ejercicio con la que se hará el procesamiento de imágenes con diferentes métodos.

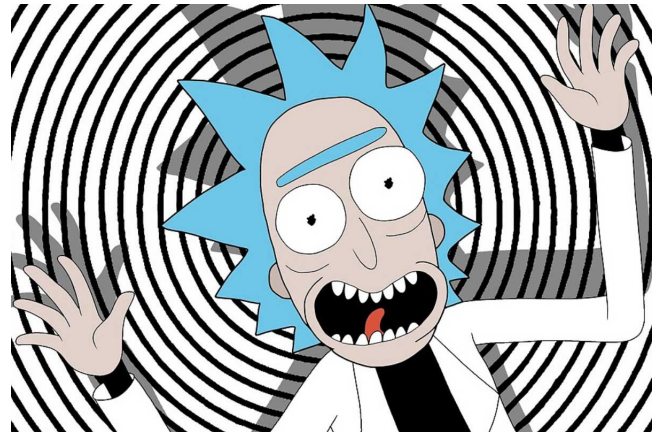


Figure 5: imagen original usada.

Escala de grises: Al realizar el procedimiento anteriormente mencionado para realizar la escala de grises podemos ver que se toma bit por bit RGB y se les asigna el mismo valor el cual sale de la formula para Y del sistema de YUV como se ve en el siguiente código

```
1   grayValue = (unsigned char) ...
   ((pixelGrayScale->red * 0.299) + ...
   (pixelGrayScale->green * 0.587) + ...
   (pixelGrayScale->blue * 0.114));
2   pixelGrayScale->red = grayValue;
3   pixelGrayScale->green = grayValue;
4   pixelGrayScale->blue = grayValue;
```



Figure 6: Escala de grises resultados.

Convolución: Para esta cambio sobre la imagen se usaron dos Kernels los cuales fueron las siguientes.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Este primer kernel lo que logra es un desenfoco sobre la imagen después de realizar el cambios en su mapa de bits. La segunda matriz que se uso fue.

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

El segundo Kernel transforma la imagen a como se ve en la imagen [Figure 7], esta convolución genera que la imagen quede totalmente en negro y resalta lo que ya se encontraba en negro en líneas resaltadas, este proceso se logra primero aplicando la escala de grises a la imagen y posterior mente agregando la convolución con el segundo Kernel.



Figure 7: Convolución.

Operador Sobel: Podemos ver que al tomar la imagen de escala de grises y someterla al operador sobel esta intensifica los bordes dentro de la imagen, esto se hace por medio de la diferenciación de intensidad, donde a partir de los cálculos se pueden generar cambios drásticos dentro de esta intensidad para poder mostrarlos con mayor contraste.

En el caso de que el resultado de aplicar el operador sobel sobre una región con intensidad constante es un vector cero, en caso de ser un borde el resultado es un vector que cruza el borde, por esto se generan unos puntos más claros y otros más oscuros.

Este tipo de imágenes nos pueden ayudar en la segmentación, reconocimiento y seguimiento de objetos principalmente.

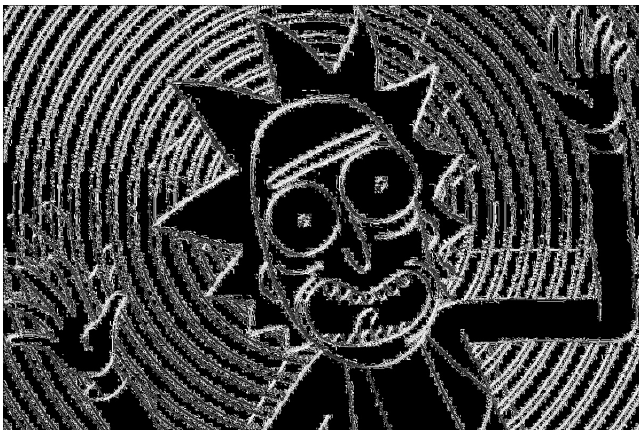


Figure 8: Operador de Sobel.

V. CONCLUSIONES

- A pesar de que se utilizó el sistema YUV para la escala de grises se podría usar una convolución usando su respectivo Kernel.
- A pesar de que hay cambios en las imágenes esta nunca cambia su formato ya que estos cambios se generan en la imagen debido a cambios sobre el mapa de bits de la imagen

REFERENCES

- [1] I. politécnico nacional. Introducción al procesamiento de imágenes bmp con ansi c. [Online]. Available: https://docencia.eafranco.com/materiales/sistemasoperativosii/06_Introduccion_al_procesamiento_de%20imagenes_BMP_con_ANSI_C.pdf
- [2] wikipedia. (2023) Windows bitmap. [Online]. Available: https://docencia.eafranco.com/materiales/sistemasoperativosii/06_Introduccion_al_procesamiento_de%20imagenes_BMP_con_ANSI_C.pdf
- [3] MathWorks. Convolución. [Online]. Available: <https://la.mathworks.com/discovery/convolution.html>
- [4] wikipedia. (2023) stdio.h. [Online]. Available: <https://es.wikipedia.org/wiki/Stdio.h>
- [5] Wikipedia. (2022) stdlib.h. [Online]. Available: <https://es.wikipedia.org/wiki/Stdlib.h>
- [6] W. enciclopedia. (2021) Yuv. [Online]. Available: <https://es.wikipedia.org/wiki/YUV>
- [7] J. Cuartas. (2021) El concepto de la convolución en gráficos, para comprender las convolutional neural networks (cnn) o redes convolucionadas. [Online]. Available: <https://josecuartas.medium.com/el-concepto-de-la-convoluci%C3%B3n-en-gr%C3%A1ficos-para-comprender-las-convolutional-neural-networks-cnn-519d2eee009c>
- [8] P. Pérez. (2018) Fundamentos básicos del procesamiento de imágenes. [Online]. Available: <https://www.famaf.unc.edu.ar/~pperez1/manuales/cim/cap2.html>
- [9] M. Sotaquirá. (2019) La convolución en las redes convolucionales. [Online]. Available: <https://www.codificandobits.com/blog/convolucion-redes-convolucionales/>
- [10] wikipedia. (2021) Operador sobel. [Online]. Available: https://es.wikipedia.org/wiki/Operador_Sobel

VI. ANEXOS