Analisis Desafío 1

Jorim de Jesus Saltarin Villamizar
Duvian Alexander Flores Munera

Informática II
Augusto Enrique Salazar Jiménez

Universidad de Antioquia

12 de abril de 2025

Análisis Desafío

El objetivo de este desafío es la reconstrucción de una imagen BMP, la cual se encuentra distorsionada, todo esto a través de una lógica inversa para recuperar su forma original. Una correcta reconstrucción implica analizar las transformaciones que se le aplicaron en un inicio, con el fin de revertir esos cambios. Lo que significa que se requiere entender la lógica detrás de la distorsión para luego aplicar el proceso contrario.

Cómo podríamos reconocer cada color

Existen varias paletas de colores, por las cuales hacen un poco complicado en aplicar una forma de generar imágenes, pero entre esas descubrimos una forma rentable de evitar extender tanto la forma en la que se ubica cada color en el pixel, es decir que se necesita un byte para cada píxel (donde en color verdadero necesita 3 veces más).

También se puede considerar que los colores en las imágenes generalmente están representados por una tabla de colores o por colores rgb con valores que van desde 0–255 o 0–65535, esto significa que cada color está representado por una combinación de números específica. Además también podría juzgar la forma en que se generan los colores de una manera tan interesante como la mezcla de colores primarios de la luz(rojo, verde y azul). Los colores secundarios de la luz (cian, amarillo y magenta). Haciendo uso de estas podría acortarse la sintaxis a utilizar en el código y una posible mejor facilidad de reconocer los colores en cada pixel.

Posibles soluciones

- Para llevar a cabo este desafío debemos saber cómo funcionan los archivos BMP y la memoria dinámica.
- Revisando el código proporcionado, descubrimos que el almacenamiento de cada píxel después del enmascaramiento sobrepasa el intervalo de 0 a 255. Lo que nos hace pensar que una posible solución, se resuelve en este apartado del código.
- Investigando encontramos información que puede ser útil, como aplicar el uso de números hexadecimales.
- Además también consideramos la investigación acerca de tarjetas gráficas y circuitos integrados, lo que convierte los 1 y 0 de una imagen que se envía a la pantalla para producir colores predefinidos.
- Aplicar el uso de una paleta de colores que facilite y disminuya la dificultad de este proceso en la aplicación con los colores primarios y secundarios de la luz que es comúnmente utilizado para imágenes digitales.

Ideas para reconstrucción de imagen de BMP

Lógica de enmascaramiento de imagen: Buscar la mejor forma de facilitar y acelerar el proceso de enmascaramiento según la exigencia de los parámetros de la imagen que sea ingresada sin el riesgo de perder calidad de la imagen o píxeles.

- Dividir los parámetros de la imagen, mediante un proceso de división
- Hacer una iteración con la memoria dinámica o binaria

Lógica para ubicar los píxeles de forma correcta: Analizar la mejor forma de garantizar el buen seguimiento respecto a la ubicación de cada pixel.

- Basarse mediante su ubicación vectorial pero en forma de solo 2 dimensiones, es decir filas y columnas.
- Basarse en asignar la ubicación de cada pixel, basándose en la idea, de que cada 8 bits equivale a un pixel.

Ideas para denominar el color/colores de cada pixel

Lógica de lectura del RGB del píxel: Pasar el término binario de cada píxel a hexadecimal, la cual es la forma más conocida de representación de colores.

- Haciendo uso de librerías especializadas en el término de identificación de colores.
- Hacer uso de la forma peculiar de un motor gráfico de jugar con los colores primarios de la luz y sus mezclas.

Ideas para la exportación por fases (enmascaramiento, ubicación, lectura de color, memoria dinamica - binario - hexadecimal y resultado final)

Lógica para exportar enmascaramiento: Guardar el archivo de enmascaramiento en su respectivo .txt con la información de cada pixel.

Lógica para exportar ubicación y color de cada píxel: En un archivo de texto se guarda la información del enmascaramiento transformada a binario para la ubicación de cada píxel. Luego pasa de binario a hexadecimal para guardar en otro archivo txt la información del color de cada pixel**.**

Lógica para exportar el resultado final: Guardar archivo en formato BMP con la imagen reconstruida según las dimensiones o ubicación guardadas en el archivo txt de ubicaciones y agregando el color guardado en el archivo txt de color de pixel.

De acuerdo con lo anterior, entre otras cosas, usted deberá:

[15%] Escribir funciones para realizar operaciones a nivel de bit, tales como XOR, desplazamiento y rotación de bits. El máximo número de bits a rotar o desplazar es de 8.

[10%] Realizar experimentos con las diferentes operaciones a nivel de bit y analizar el efecto de las transformaciones sobre la integridad de los datos y su utilidad en un escenario de encriptación básica de información.

[10%] Implementar un algoritmo que permita verificar el resultado del enmascaramiento, comparando la imagen transformada y la máscara contra los archivos de resultados.

[65%] Implementar un algoritmo que permita identificar qué tipo de operaciones a nivel de bits fueron realizadas (y en qué orden), con el fin de reconstruir la imagen original.