

Informe Desafío I

Por:

Isabela Aguirre Ceballos

Juan Felipe Higuera Pérez

Informática II

Universidad de Antioquia

2025

Análisis del problema

En este desafío nos entregan una imagen final (ID) que fue modificada usando transformaciones a nivel de bits. Estas transformaciones pueden incluir operaciones XOR con otra imagen (IM), rotaciones o desplazamientos de bits. El problema es que no sabemos qué transformaciones se hicieron, ni en qué orden se aplicaron.

Después de cada transformación, se realizó un enmascaramiento. Esto significa que, desde una posición aleatoria en la imagen transformada, se tomó una porción de datos y se sumó píxel a píxel con una máscara de color (M). El resultado de esa suma se guardó en archivos .txt. Cada archivo tiene dos datos clave: el número de desplazamiento (s) y los valores RGB que resultaron de la suma con la máscara.

Ese número (s) indica desde qué punto del arreglo de píxeles de la imagen transformada se comienza a tomar la porción. Luego, se suman los valores RGB de esa porción con los de la máscara M. El total de datos que se toman depende del tamaño de la máscara, que generalmente es mucho más pequeña que la imagen completa.

La tarea principal es reconstruir la imagen original (IO) usando solamente la imagen final ID, la imagen IM (utilizada en los pasos de XOR), la máscara M y los archivos .txt que quedaron como rastro después de cada transformación.

Como no se conoce el orden de los pasos, la solución consiste en analizar y probar diferentes combinaciones posibles de transformaciones. Después de cada intento, se aplica el enmascaramiento usando la máscara M y el desplazamiento (s), y se verifica si el resultado coincide con lo que aparece en el archivo .txt. Si coincide, quiere decir que se aplicó correctamente esa transformación. Si no, hay que intentar con otra opción. De esta forma, paso a paso, se puede ir deshaciendo todo lo que se hizo hasta llegar a la imagen original.

Diseño

Para resolver este problema, lo primero que realizaremos es cargar en el programa la imagen final (ID), la imagen IM, la máscara M y los archivos .txt que contienen el resultado del enmascaramiento.

Luego, se probarán diferentes combinaciones de transformaciones (XOR, rotación de bits, desplazamiento de bits), aplicándolas en orden inverso, una por una. Después de cada transformación, se utilizará el valor de desplazamiento s del archivo correspondiente para tomar una parte de la imagen transformada.

Una vez se haya verificado que cada transformación fue correcta y ya no queden archivos .txt por procesar, significa que se ha logrado recuperar la imagen original. En ese momento, el programa entregará al usuario un archivo .bmp llamado **“IO.bmp”**, como resultado final del proceso.

Para poder aplicar y revertir las transformaciones en la imagen, se van a implementar las siguientes funciones:

- **Función de XOR:**

Aplica una operación XOR entre dos imágenes del mismo tamaño (por ejemplo, entre ID e IM), píxel por píxel y canal por canal (R, G, B). Esta operación también es útil para revertir una transformación, ya que aplicar XOR dos veces con la misma imagen devuelve la original.

- **Función de rotación de bits:**

Rota los bits de cada byte hacia la izquierda o hacia la derecha. Se usará para revertir

transformaciones de rotación. Por ejemplo, si se rotaron 3 bits a la derecha, se puede revertir rotando 3 a la izquierda.

- **Función de desplazamiento de bits:**

Realiza desplazamientos a la izquierda o a la derecha de cada byte. A diferencia de la rotación, los bits que se salen se pierden, por lo que este tipo de transformación puede ser más difícil de revertir si se utilizó.

- **Función de enmascaramiento:**

Suma los valores RGB de una porción de la imagen transformada con los de la máscara M, comenzando desde una posición S. Se usa para simular el proceso de enmascaramiento original y generar un resultado comparable con los datos del archivo .txt.

- **Función de verificación contra archivo .txt:**

Compara el resultado del enmascaramiento con los datos guardados en el archivo correspondiente. Verifica si cada valor coincide exactamente. Si todos los valores son iguales, significa que la transformación aplicada fue correcta y se puede continuar con el siguiente paso. Si no, hay que probar otra opción.

- **Función obtención de cantidad de .txt a trabajar:**

Analizará y contará la cantidad de archivos .txt que se proporcionan en el caso a resolver, con el fin de conocer cuántas pistas están disponibles. Esta función nos entregará el número total de estos archivos, lo cual permitirá saber desde cuál

comenzar y llevar el control del proceso, disminuyendo en uno la cantidad restante cada vez que se realice una transformación correcta.

- **Función NOT (Inversor):**

Invierte todos los valores de los bits de cada canal RGB (1->0 / 0->1). Una transformación bastante sencilla que con aplicarla dos veces en un valor este regrese al valor original, siendo posible usarla dentro de la función XOR la operación que se realiza dentro de esta.