

Reconstrucción de Imagen BMP mediante Funciones de Transformación, Enmascaramiento y Verificación

Desafío I - Informática II, Semestre 2025-1

Daniel Rúa, Tomás Mesa

Ingeniería Electrónica
Universidad de Antioquia

Resumen—Este documento presenta el desarrollo de un proyecto cuyo objetivo es reconstruir una imagen original a partir de una imagen BMP distorsionada. La metodología se basa en la aplicación secuencial de distintas funciones en C++ que permiten revertir operaciones a nivel de bits ejecutadas sobre bloques de 8 bits. Entre las transformaciones se incluyen desplazamientos (derecha e izquierda), rotaciones (derecha e izquierda) y operación XOR. Posteriormente, se aplica un enmascaramiento con una máscara específica y se verifica que el resultado coincida con datos de referencia contenidos en un archivo .txt. El proceso se repite para cada bloque de datos hasta lograr reconstruir completamente la imagen original.

Index Terms—Imagen BMP, Transformaciones a nivel de bits, Ingeniería inversa, C++, Enmascaramiento, Verificación, Bloques de 8 bits.

I. INTRODUCCIÓN

La protección y encriptación de información visual requieren técnicas que a menudo implican operaciones a nivel de bits sobre la imagen. En este proyecto se propone reconstruir una imagen original a partir de una imagen BMP distorsionada mediante la reversión de las transformaciones aplicadas. La solución se implementa en C++ utilizando el framework Qt y se enfoca en el uso de punteros, arreglos y memoria dinámica, evitando el uso de estructuras y la STL.

II. ANÁLISIS DEL PROBLEMA

El reto consiste en:

- Analizar una imagen BMP de 24 bits distorsionada, que ha sido sometida a diversas transformaciones.
- Aplicar una serie de funciones para revertir dichas transformaciones. Las funciones se dividen en tres etapas: transformación, enmascaramiento y verificación.
- Las transformaciones incluyen desplazamiento a la derecha, desplazamiento a la izquierda, rotación a la derecha, rotación a la izquierda y operación XOR, todas aplicadas a bloques de 8 bits.
- La función de enmascaramiento suma el resultado de la transformación con una máscara específica.
- La función de verificación compara el resultado final con datos de referencia almacenados en un archivo .txt, determinando si la operación fue correcta.

III. DISEÑO Y METODOLOGÍA DE LA SOLUCIÓN

III-A. Estructura del Algoritmo

La solución está organizada en las siguientes funciones:

1. **Función de Transformación:** Se implementa una función encargada de aplicar una de las operaciones sobre un bloque de 8 bits. Las operaciones disponibles son:
 - Desplazamiento a la derecha.
 - Desplazamiento a la izquierda.
 - Rotación a la derecha.
 - Rotación a la izquierda.
 - Operación XOR.
2. **Función de Enmascaramiento:** Una vez aplicada la transformación, esta función suma el resultado obtenido con una máscara específica.
3. **Función de Verificación:** Se compara el resultado del enmascaramiento contra los datos de referencia contenidos en un archivo .txt. Si los datos coinciden, se guarda el bloque transformado. En caso contrario, se prueba con otra operación de transformación.

El proceso se repite de manera iterativa para cada bloque de datos hasta que todos los bloques hayan sido verificados y validados, culminando con la reconstrucción completa de la imagen original.

III-B. Herramientas y Restricciones

La implementación se realiza en C++ haciendo uso del framework Qt, privilegiando punteros, arreglos y memoria dinámica. No se utilizan estructuras (structs) ni la STL. Además, se mantiene un control riguroso de la evolución del proyecto mediante commits en un repositorio público.

III-C. Función de Transformación

Se implementan las operaciones de desplazamiento, rotación y XOR sobre bloques de 8 bits. Cada operación se aplica de manera que, al variar la transformación, se busque lograr la coincidencia con los datos de referencia.

III-D. Función de Enmascaramiento

El enmascaramiento se realiza sumando el resultado de la transformación con una máscara predefinida, permitiendo la recuperación de la secuencia de bits original.

III-E. Función de Verificación

En esta etapa se compara el resultado final de cada bloque (tras transformación y enmascaramiento) con los datos de referencia contenidos en un archivo .txt. Si la comparación es exitosa, se guarda el bloque; de lo contrario, se intenta con otra transformación.