



UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERIA Y TECNOLOGIAS

AVANZADAS – IPN

Prácticas de Laboratorio

Formato BMP y Esteganografía

Prof. Noé Sierra Romero



MATERIA

Multimedia

ALUMNO

Hernández Arteaga Itzel

TEMA

Esteganográfica

Grupo:3TM2 Práctica 1

Fecha de asignación:

17/Febrero/2026



Introducción General

El formato BMP (Bitmap) es uno de los formatos de imagen más simples y utilizados en el estudio de procesamiento de imágenes digitales, debido a su estructura sin compresión (en su variante más común) y su cabecera bien documentada. Esta característica lo convierte en el soporte ideal para aprender técnicas de esteganografía, es decir, el arte de ocultar información dentro de un medio digital de forma que sea imperceptible al observador humano. Las dos prácticas de este módulo están diseñadas con progresión de dificultad: toma como base el conocimiento de la estructura del archivo BMP, y se avanza hacia la técnica clásica de Bit Menos Significativo (LSB), y finalmente se exploran variantes más robustas orientadas a la seguridad.

PRÁCTICA 1 — Esteganografía LSB en Imágenes BMP

Objetivo general: Implementar desde cero la técnica de ocultamiento de información mediante sustitución del bit menos significativo (Least Significant Bit – LSB) en imágenes BMP de 24 bits por píxel.

Competencias a desarrollar: Manipulación de bits, diseño de algoritmos de codificación/decodificación, análisis de capacidad de almacenamiento y evaluación de calidad de imagen (PSNR).

Herramientas sugeridas: Python 3 (módulo struct, Pillow opcional solo para comparación visual), imagen BMP 24 bpp de al menos 200×200 px.

Duración estimada: 60 minutos

2.1 Marco Teórico

La técnica LSB consiste en reemplazar el bit menos significativo de cada canal de color (B, G, R) de cada píxel con los bits del mensaje secreto. Dado que un bit de cambio en un byte corresponde a un cambio máximo de 1 unidad de intensidad (de 256 posibles), la diferencia es imperceptible a simple vista. Capacidad de carga: Para una imagen de $W \times H$ píxeles a 24 bpp, utilizando 1 LSB por canal se pueden ocultar $(W \times H \times 3)$ bits = $(W \times H \times 3) / 8$ bytes de mensaje. Para una imagen de 512×512 píxeles: $(512 \times 512 \times 3) / 8 = 98,304$ bytes ≈ 96 KB. Para señalar el fin del mensaje se puede utilizar un terminador (por ejemplo, el carácter nulo '\0'), un encabezado con la longitud del mensaje almacenado en los primeros bits, o una combinación de ambos. En esta práctica se utilizará un encabezado de 32 bits al inicio de los datos de píxel para almacenar la longitud del mensaje en bytes.

2.2 Procedimiento



Fig.1 Imagen que se utilizó para la práctica 1



Código:

```
import struct
import math
def leer_bmp(filepath):
    """Retorna (header_bytes, pixels, width, height, row_size)"""
    with open(filepath, 'rb') as f:
        data = f.read()
        offset = struct.unpack_from('<I', data, 10)[0]
        width = struct.unpack_from('<i', data, 18)[0]
        height = struct.unpack_from('<i', data, 22)[0]
        row_size = (width * 3 + 3) & ~3
        header = bytearray(data[:offset])
        pixels = bytearray(data[offset:])
        return header, pixels, width, height, row_size
def guardar_bmp(filepath, header, pixels):
    with open(filepath, 'wb') as f:
        f.write(header)
        f.write(pixels)
def embed_lsb(src_path, dst_path, mensaje):
    header, pixels, width, height, row_size = leer_bmp(src_path)
    msg_bytes = mensaje.encode('utf-8')
    msg_len = len(msg_bytes)
    datos = struct.pack('>I', msg_len) + msg_bytes
    bits = []
    for byte in datos:
        for i in range(7, -1, -1):
            bits.append((byte >> i) & 1)
    if len(bits) > len(pixels):
        raise ValueError('Mensaje demasiado largo para esta imagen')
    pixels_mod = bytearray(pixels)
    for idx, bit in enumerate(bits):
        pixels_mod[idx] = (pixels_mod[idx] & 0xFE) | bit
    guardar_bmp(dst_path, header, pixels_mod)
    print(f'[OK] Mensaje de {msg_len} bytes incrustado en {dst_path}')
def extract_lsb(stego_path):
    _, pixels, _, _, _ = leer_bmp(stego_path)
    len_bits = [pixels[i] & 1 for i in range(32)]
    msg_len = 0
    for b in len_bits:
        msg_len = (msg_len << 1) | b
    total_bits = 32 + msg_len * 8
    msg_bits = [pixels[i] & 1 for i in range(32, total_bits)]
    msg_bytes = bytearray()
    for i in range(0, len(msg_bits), 8):
        byte = 0
        for bit in msg_bits[i:i+8]:
            byte = (byte << 1) | bit
        msg_bytes.append(byte)
    return msg_bytes.decode('utf-8')
def calcular_psnr(original_path, stego_path):
    _, pix_orig, w, h, _ = leer_bmp(original_path)
    _, pix_steg, _, _, _ = leer_bmp(stego_path)
    mse = sum((a - b)**2 for a, b in zip(pix_orig, pix_steg)) / (w * h * 3)
```



```
if mse == 0:  
    return float('inf')  
psnr = 10 * math.log10(255**2 / mse)  
print(f'MSE: {mse:.6f}')  
print(f'PSNR: {psnr:.2f} dB (>40 dB: cambio imperceptible)')  
return psnr  
# --- PRUEBA ---  
embed_lsb('imagen.bmp', 'stego.bmp', 'TELEMÁTICA SECRETA 2025')  
  
recuperado = extract_lsb('stego.bmp')  
print(f'Mensaje recuperado: {recuperado}')  
  
assert recuperado == 'TELEMÁTICA SECRETA 2025', '¡Error en la extracción!'  
print('Prueba exitosa.')  
  
calcular_psnr('imagen.bmp', 'stego.bmp')
```

Resultados

```
[OK] Mensaje de 24 bytes incrustado en stego.bmp  
Mensaje recuperado: TELEMÁTICA SECRETA 2025  
Prueba exitosa.  
MSE: 0.000942  
PSNR: 78.39 dB (>40 dB: cambio imperceptible)  
78.39183163432115
```

Modificación del código para cambiar el tamaño del mensaje

```
# --- PRUEBA ---  
base = "TELEMATICA SECRETA 2026"  
faltan = 50 - len(base.encode('utf-8'))  
mensaje = base + " " * faltan  
  
embed_lsb('imagen.bmp', 'stego.bmp', mensaje)  
  
recuperado = extract_lsb('stego.bmp')  
print(f'Mensaje recuperado: {recuperado}')  
  
assert recuperado == mensaje, '¡Error en la extracción!'  
print('Prueba exitosa.')  
  
calcular_psnr('imagen.bmp', 'stego.bmp')
```

```
[OK] Mensaje de 50 bytes incrustado en stego.bmp  
Mensaje recuperado: TELEMATICA SECRETA 2026  
Prueba exitosa.  
MSE: 0.001900  
PSNR: 75.34 dB (>40 dB: cambio imperceptible)  
75.3432675991508
```



```
[OK] Mensaje de 500 bytes incrustado en stego.bmp
Mensaje recuperado: TELEMATICA SECRETA 2026
Prueba exitosa.
MSE: 0.017217
PSNR: 65.77 dB (>40 dB: cambio imperceptible)
65.77131289731933
```

```
[OK] Mensaje de 5000 bytes incrustado en stego.bmp
Mensaje recuperado: TELEMATICA SECRETA 2026
Prueba exitosa.
MSE: 0.025499
PSNR: 64.07 dB (>40 dB: cambio imperceptible)
64.06562183726032
```

```
ValueError                                     Traceback (most recent call last)
/tmp/ipython-input-350720051.py in <cell line: 0>()
      91 mensaje = base + " " * faltan
      92
---> 93 embed_lsb('imagen.bmp', 'stego.bmp', mensaje)
      94
      95 recuperado = extract_lsb('stego.bmp')

/tmp/ipython-input-350720051.py in embed_lsb(src_path, dst_path, mensaje)
      35
      36     if len(bits) > len(pixels):
---> 37         raise ValueError('Mensaje demasiado largo para esta imagen')
      38
      39     pixels_mod = bytearray(pixels)

ValueError: Mensaje demasiado largo para esta imagen
```

Realice las siguientes pruebas y complete la tabla de resultados:

Imagen (px)	Tamaño mensaje	PSNR obtenido	¿Imperceptible?
200x200	50 bytes	75.34 dB	Si
200x200	500 bytes	65.77 dB	Si
512x512	5000 bytes	64.07 dB	Si
512x512	Capacidad Maxima	No se puede obtener	No

2.4 Preguntas de Análisis

1. ¿Qué sucede si se intenta ocultar un mensaje más largo que la capacidad de la imagen? Implemente un control de error apropiado.



UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERIA Y TECNOLOGIAS

AVANZADAS – IPN

Prácticas de Laboratorio

Formato BMP y Esteganografía

Prof. Noé Sierra Romero



No alcanzan los pixeles para cubrir la cantidad de bits que tiene el mensaje, es por eso que se tiene una capacidad máxima de tamaño para que no sobrepase el límite, se utilizó la siguiente función para el control de error para verificar que no halla sobrepasado el límite de la imagen.

```
if len(bits) > len(pixels):  
    raise ValueError('Mensaje demasiado largo para esta imagen')
```

2. Compare visualmente (o mediante histograma) la imagen original y la estenografiada. ¿Hay diferencias observables en el histograma de intensidades?

El histograma presenta ligeras variaciones entre valores consecutivos debido a la modificación del bit menos significativo. Sin embargo, no se observan cambios significativos en la distribución global de intensidades.

3. Proponga una estrategia para aumentar la capacidad de ocultamiento a 2 LSB's por canal. ¿Cómo afecta esto al PSNR?

De acuerdo a la formula general de LSB en RGB tenemos que

Capacidad en bits = ancho × alto × 3 × 1 bit por canal

Ya que el normalmente se utiliza R = 10101101 el bit más significativo

Si se modifica los dos últimos bits R = 1010110, la capacidad de bits sería así:

Capacidad = ancho × alto × 3 × 2 bits lo que implica que el PSNR disminuya un poco.

4. ¿Por qué el formato BMP es preferible al JPEG para esteganografía LSB? ¿Qué ocurriría si se guarda la imagen stego como JPEG?

Las principales diferencias entre estos dos archivos radican en la forma de comprimir los datos de la imagen, mostrar el color y mantener la calidad de la imagen, además de cómo se usan normalmente.