**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО”**

**НАВЧАЛЬНО НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**

Комп’ютерна графіка

Лабораторна робота №6

Виконав:

Студент 2 курсу

Групи ФІ-21

Поштаренко Сергій

Київ 2024

# «Цифрова стеганографія засобами найменш значущого біта»

***Мета роботи***: навчитись вбудовувати цифровий водяний знак у зображення методом найменшого значущого біта.

***Завдання:***

1. Ознайомитись із усіма теоретичними відомостями до лабораторної роботи.
2. Виконати проєктування програмного додатку для PC, який:

a. виконує вбудовування водяного знаку зображення. Для процесу вбудовування водяний знак перетворити до бінарного (чорно-білого, не сірого!). Використати для вбудовування метод найменшого значущого біта. Параметр «номер бітової площини» (натуральне число, менше за 9) вивести на форму головного вікна для відображення різниці в результатах вбудовування. Водяний знак вбудовувати у канал Blue. Передбачити випадок різних розмірів вхідного зображення (контейнера) та водяного знаку. Якщо вхідне зображення більше за водяний знак, виконати вбудовування циклічно та періодично для всього вхідного зображення (замостити вхідне зображення водяним знаком). Якщо вхідне зображення менше за водяний знак, вбудувати частину водяного знаку для всього вхідного зображення.

b. Здійснити вбудовування текстової інформації (яка представлена побітово) у вхідне зображення на основі методу найменшого значущого біта.

c. Додатковий бал передбачається з встворення методу, який виконує вилучення вонядного знаку із заповненого конейнега (зображення з водяним знаком) для завдання 7 даної лабораторної роботи.

3. Усі пункти завдання лабораторної роботи виконати в одному багатовіконному програмному додатку.

4. Навести основні етапи проєктування додатку, зокрема, вказати діаграму класів, діаграму пакетів та діаграму прецедентів користування додатком.

***Результат:***

from PIL import Image

import numpy as np

def convert\_to\_binary\_image(image\_path):

image = Image.open(image\_path).convert('1') # Convert to binary

return np.array(image)

def embed\_watermark(input\_image\_path, watermark\_binary, bit\_plane):

image = Image.open(input\_image\_path)

image\_array = np.array(image)

watermark\_rows, watermark\_cols = watermark\_binary.shape

img\_rows, img\_cols, \_ = image\_array.shape

for i in range(img\_rows):

for j in range(img\_cols):

watermark\_pixel = watermark\_binary[i % watermark\_rows, j % watermark\_cols] # Ensures that the indices i and j wrap around when they exceed the dimensions of the watermark

bit = 1 if watermark\_pixel else 0

# Embed bit in the specified bit plane of the blue channel

image\_array[i, j, 2] = (image\_array[i, j, 2] & ~(1 << (bit\_plane - 1))) | (bit << (bit\_plane - 1))

return Image.fromarray(image\_array)

def text\_to\_bits(text):

return ''.join(format(ord(char), '08b') for char in text)

def embed\_text(input\_image, text):

image\_array = np.array(input\_image)

binary\_text = text\_to\_bits(text)

data\_index = 0

text\_length = len(binary\_text)

for i in range(image\_array.shape[0]):

for j in range(image\_array.shape[1]):

if data\_index < text\_length:

image\_array[i, j, 2] = (image\_array[i, j, 2] & 0xFE) | int(binary\_text[data\_index])

data\_index += 1

return Image.fromarray(image\_array)

def remove\_watermark(input\_image\_path, bit\_plane):

image = Image.open(input\_image\_path)

image\_array = np.array(image)

rows, cols, \_ = image\_array.shape

for i in range(rows):

for j in range(cols):

# Clear the specified bit plane of the blue channel

image\_array[i, j, 2] = image\_array[i, j, 2] & ~(1 << (bit\_plane - 1))

return Image.fromarray(image\_array)

watermark\_binary = convert\_to\_binary\_image('WM.png')

watermarked\_image = embed\_watermark('img.png', watermark\_binary, 6) # Specify the bit plane (1-8)

embedded\_text\_image = embed\_text(watermarked\_image, "Hidden text here")

embedded\_text\_image.save('final\_img.png')

watermarked\_image\_path = 'final\_img.png'

unwatermarked\_image = remove\_watermark(watermarked\_image\_path, bit\_plane=6)

unwatermarked\_image.save('unwatermarked\_img.png')

8 – 1 bit

Зображення, що містить просто неба, дерево, транспортний засіб, Наземний транспортний засіб

Автоматично згенерований описЗображення, що містить риф, природа, транспортний засіб, акваріум

Автоматично згенерований описЗображення, що містить дерево, транспортний засіб, просто неба, Наземний транспортний засіб

Автоматично згенерований описЗображення, що містить просто неба, дерево, транспортний засіб, Наземний транспортний засіб

Автоматично згенерований описЗображення, що містить просто неба, дерево, транспортний засіб, Наземний транспортний засіб

Автоматично згенерований описЗображення, що містить просто неба, дерево, транспортний засіб, Наземний транспортний засіб

Автоматично згенерований описЗображення, що містить просто неба, дерево, транспортний засіб, Наземний транспортний засіб

Автоматично згенерований описЗображення, що містить просто неба, дерево, транспортний засіб, Наземний транспортний засіб

Автоматично згенерований описЗображення, що містить просто неба, дерево, транспортний засіб, Наземний транспортний засіб

Автоматично згенерований опис