Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики» (СибГУТИ)

Институт информатики и вычислительной техники

Кафедра прикладной математики и кибернетики

# Лабораторная работа №4 по дисциплине

**Прикладная стеганография**

Выполнил:

студент гр.МГ-411 Шевельков П.С.

ФИО студента

«27» марта 2025 г.

Новосибирск 2025 г.

# Задание на лабораторную работу:

1. Составить краткий обзор методов RDH на основе предложенных научных работ либо новых работ и выбрать метод для реализации.
2. Реализовать систему сокрытия данных для 8-битных изображений формата bmp по выбранному методу.
3. Оценить ёмкость встраивания и величину искажения PSNR на примере контейнера из набора. Провести качественный анализ стегоконтейнеров.

# Результаты работы программы:

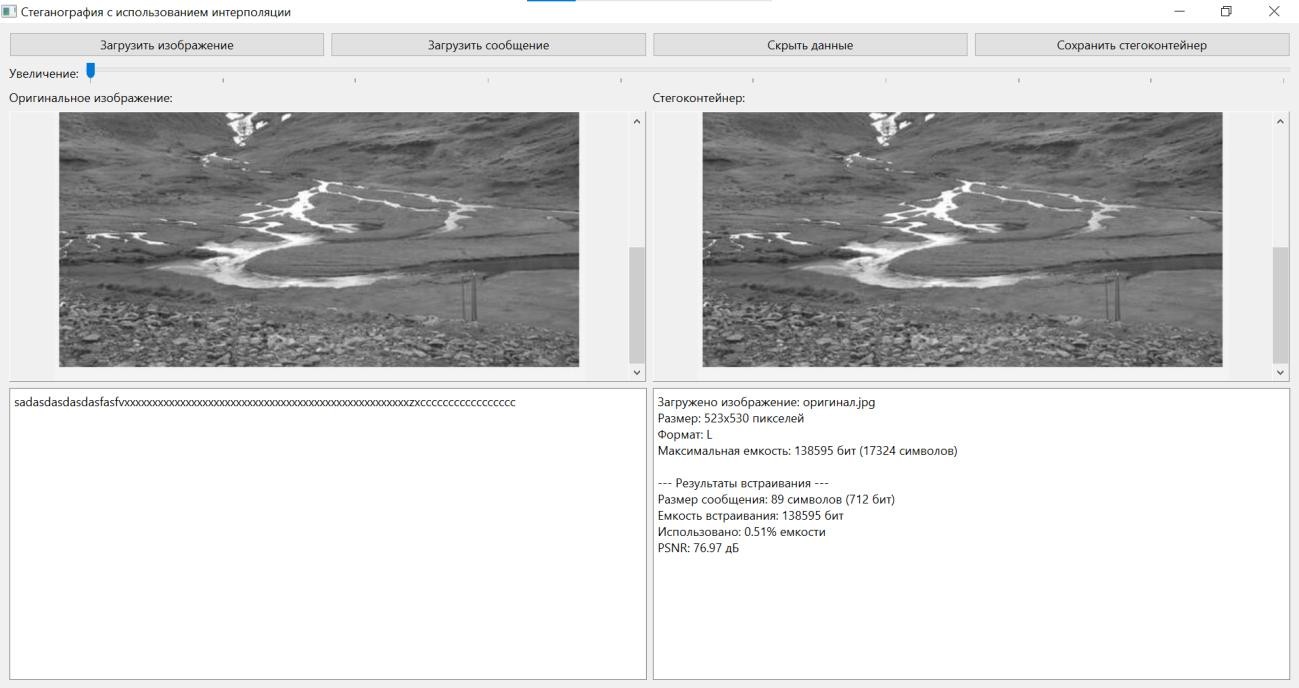
****

рисунок 1. Результат внедрения скрытых данных.

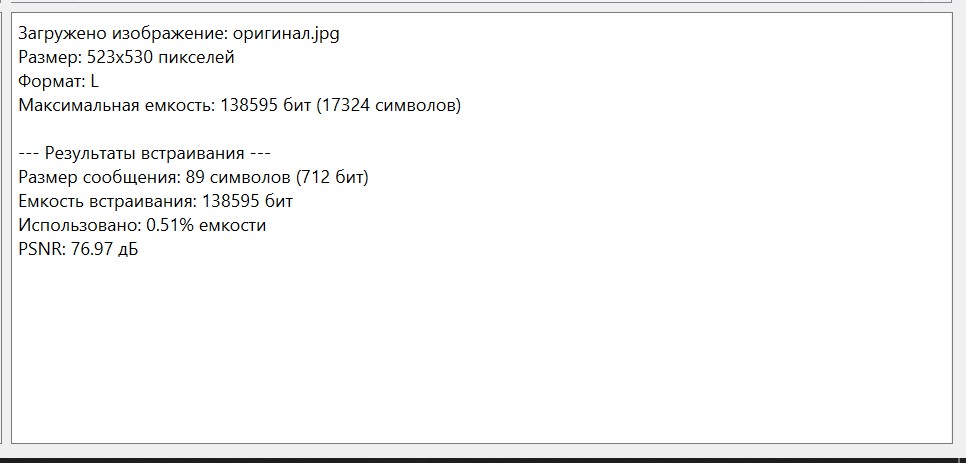


рисунок 2. Результат оценки встраивания данных.

## Описание метода сокрытия данных с использованием интерполяции изображений

Метод сокрытия данных с использованием интерполяции изображений (Data Hiding Method Using Image Interpolation) представляет собой технику стеганографии, которая позволяет скрывать информацию в изображениях, изменяя значения пикселей на основе интерполяции. Этот метод обеспечивает более высокую степень скрытности и устойчивости по сравнению с традиционными методами, такими как встраивание в наименьшие значащие биты (LSB).

## Описание работы приложения:

Приложение "Стеганография с использованием интерполяции" предназначено для скрытия текстовых сообщений в изображениях с использованием метода интерполяции. Пользователь может загружать изображения и текстовые сообщения, а затем встраивать эти сообщения в загруженные изображения, изменяя значения пикселей на основе

интерполированных значений соседних пикселей.

Приложение также позволяет сохранять полученные стегоконтейнеры и отображать информацию о процессе встраивания, включая емкость

встраивания и PSNR (Peak Signal-to-Noise Ratio). Благодаря использованию интерполяции, изменения в изображении менее заметны, что обеспечивает более высокую степень скрытности и устойчивости по сравнению с традиционными методами.

## Основные компоненты кода:

* 1. Метод **load\_image**: Реализует функционал выбора и загрузки изображения из файловой системы. Проверяет, является ли изображение в градациях серого, и отображает его в интерфейсе.
  2. Метод **load\_message**: Позволяет пользователю загружать текстовые сообщения из файловой системы. Проверяет, не превышает ли длина сообщения максимальную емкость для встраивания в загруженное изображение.
  3. Метод **hide\_data**: Внедряет текстовое сообщение в загруженное изображение, используя метод LSB Matching Revisited. Преобразует сообщение в битовую последовательность и встраивает ее в пиксели изображения.
  4. Метод **text\_to\_bits**: Преобразует текстовое сообщение в

последовательность бит, где каждый символ представляется в виде 8 бит.

* 1. Метод **interpolation\_method**: Реализует алгоритм сокрытия данных с использованием интерполяции для встраивания битов сообщения в пиксели изображения. Метод вычисляет интерполированное значение на основе соседних пикселей и изменяет значение текущего пикселя в зависимости от четности интерполированного значения и значения бита сообщения. Это позволяет скрывать информацию в изображении, минимизируя визуальные изменения и обеспечивая более высокую степень скрытности.
  2. Метод **calculate\_psnr**: Рассчитывает PSNR между оригинальным и стегоконтейнером, что позволяет оценить качество встраивания.
  3. Метод **display\_image**: Отображает изображение в интерфейсе с учетом заданного масштаба. Преобразует изображение в формат QImage для отображения в QLabel.
  4. Метод **update\_zoom**: Обновляет масштаб отображаемых изображений в зависимости от значения слайдера.
  5. Метод **save\_stego\_image**: Позволяет пользователю сохранять стегоконтейнер в файловой системе, проверяя, что файл имеет правильное расширение.
  6. Метод **extract\_data**: Извлекает скрытое сообщение из стегоконтейнера, используя алгоритм, аналогичный тому, что использовался для встраивания.
  7. Метод **bits\_to\_text**: Преобразует последовательность бит обратно в текст, группируя биты по 8 и преобразуя их в символы.

## Ссылка на программу:

https://github.com/bothyD/steganograf

# Листинг:

import sys import os

import numpy as np from PIL import Image

from PyQt6.QtWidgets import (QApplication, QMainWindow, QWidget, QVBoxLayout, QHBoxLayout,

QPushButton, QLabel, QFileDialog, QScrollArea,

QSlider,

TextEdit, QMessageBox) from PyQt6.QtGui import QPixmap, QImage

from PyQt6.QtCore import Qt import math

class SteganographyApp(QMainWindow): def init (self):

super(). init ()

self.setWindowTitle("Стеганография с использованием интерполяции") self.setGeometry(100, 100, 1200, 800)

# Инициализация переменных self.original\_image = None self.stego\_image = None self.zoom\_factor = 1

# Создание интерфейса self.init\_ui()

def init\_ui(self): central\_widget = QWidget()

main\_layout = QVBoxLayout(central\_widget)

# Верхняя панель с кнопками button\_layout = QHBoxLayout()

self.load\_image\_btn = QPushButton("Загрузить изображение") self.load\_image\_btn.clicked.connect(self.load\_image)

self.load\_message\_btn = QPushButton("Загрузить сообщение") self.load\_message\_btn.clicked.connect(self.load\_message) self.load\_message\_btn.setEnabled(False)

self.hide\_data\_btn = QPushButton("Скрыть данные") self.hide\_data\_btn.clicked.connect(self.hide\_data) self.hide\_data\_btn.setEnabled(False)

self.save\_stego\_btn = QPushButton("Сохранить стегоконтейнер") self.save\_stego\_btn.clicked.connect(self.save\_stego\_image) self.save\_stego\_btn.setEnabled(False)

button\_layout.addWidget(self.load\_image\_btn) button\_layout.addWidget(self.load\_message\_btn) button\_layout.addWidget(self.hide\_data\_btn) button\_layout.addWidget(self.save\_stego\_btn)

main\_layout.addLayout(button\_layout) zoom\_layout = QHBoxLayout() zoom\_label = QLabel("Увеличение:")

self.zoom\_slider = QSlider(Qt.Orientation.Horizontal)

self.zoom\_slider.setMinimum(1) self.zoom\_slider.setMaximum(10) self.zoom\_slider.setValue(1)

self.zoom\_slider.setTickPosition(QSlider.TickPosition.TicksBelow) self.zoom\_slider.setTickInterval(1) self.zoom\_slider.valueChanged.connect(self.update\_zoom)

zoom\_layout.addWidget(zoom\_label) zoom\_layout.addWidget(self.zoom\_slider)

main\_layout.addLayout(zoom\_layout) images\_layout = QHBoxLayout() original\_layout = QVBoxLayout()

original\_label = QLabel("Оригинальное изображение:") self.original\_image\_label = QLabel() self.original\_image\_label.setAlignment(Qt.AlignmentFlag.AlignCenter)

self.original\_scroll\_area = QScrollArea() self.original\_scroll\_area.setWidgetResizable(True) self.original\_scroll\_area.setWidget(self.original\_image\_label)

original\_layout.addWidget(original\_label) original\_layout.addWidget(self.original\_scroll\_area)

# Стегоконтейнер stego\_layout = QVBoxLayout()

stego\_label = QLabel("Стегоконтейнер:") self.stego\_image\_label = QLabel()

self.stego\_image\_label.setAlignment(Qt.AlignmentFlag.AlignCenter)

self.stego\_scroll\_area = QScrollArea() self.stego\_scroll\_area.setWidgetResizable(True) self.stego\_scroll\_area.setWidget(self.stego\_image\_label)

stego\_layout.addWidget(stego\_label) stego\_layout.addWidget(self.stego\_scroll\_area)

images\_layout.addLayout(original\_layout) images\_layout.addLayout(stego\_layout)

main\_layout.addLayout(images\_layout) text\_layout = QHBoxLayout()

self.message\_text = QTextEdit() self.message\_text.setPlaceholderText("Введите сообщение для скрытия или

загрузите файл...")

self.info\_text = QTextEdit() self.info\_text.setReadOnly(True)

self.info\_text.setPlaceholderText("Информация о стеганографическом процессе будет отображаться здесь...")

text\_layout.addWidget(self.message\_text) text\_layout.addWidget(self.info\_text)

main\_layout.addLayout(text\_layout)

self.setCentralWidget(central\_widget)

def load\_image(self):

file\_path, \_ = QFileDialog.getOpenFileName(self, "Выберите изображение", "", "Image Files (\*.bmp \*.jpg \*.jpeg \*.png)")

if file\_path: try:

img = Image.open(file\_path) if img.mode != "L":

img = img.convert('L')

self.original\_image = img

self.display\_image(img, self.original\_image\_label) width, height = img.size self.info\_text.setText(f"Загружено изображение:

{os.path.basename(file\_path)}\n"

// 2} бит "

f"Размер: {width}x{height} пикселей\n" f"Формат: {img.mode}\n"

f"Максимальная емкость: {(width \* height) f"({(width \* height) // 16} символов)")

self.load\_message\_btn.setEnabled(True) self.message\_text.setEnabled(True) self.hide\_data\_btn.setEnabled(True)

except Exception as e:

QMessageBox.critical(self, "Ошибка", f"Не удалось загрузить изображение: {str(e)}")

def hide\_data(self):

if not self.original\_image:

QMessageBox.warning(self, "Ошибка", "Сначала загрузите изображение!") return

message = self.message\_text.toPlainText() if not message:

QMessageBox.warning(self, "Ошибка", "Введите сообщение для скрытия!") return

message\_bits = self.text\_to\_bits(message) width, height = self.original\_image.size max\_bits = (width \* height) // 2

if len(message\_bits) > max\_bits: QMessageBox.warning(self, "Предупреждение",

f"Сообщение слишком длинное! Будет сохранено только {max\_bits // 8} символов.")

message\_bits = message\_bits[:max\_bits] img\_array = np.array(self.original\_image)

stego\_array = self.interpolation\_method(img\_array, message\_bits) self.stego\_image = Image.fromarray(stego\_array) self.display\_image(self.stego\_image, self.stego\_image\_label)

psnr = self.calculate\_psnr(np.array(self.original\_image), stego\_array) embedding\_capacity = (width \* height) // 2

actual\_embedded = len(message\_bits)

self.info\_text.append(f"\n--- Результаты встраивания ---\n"

f"Размер сообщения: {len(message)} символов

({actual\_embedded} бит)\n"

f"Емкость встраивания: {embedding\_capacity} бит\n" f"Использовано: {actual\_embedded /

embedding\_capacity \* 100:.2f}% емкости\n"

f"PSNR: {psnr:.2f} дБ")

self.save\_stego\_btn.setEnabled(True)

def load\_message(self):

file\_path, \_ = QFileDialog.getOpenFileName(self, "Выберите текстовый файл", "", "Text Files (\*.txt)")

if file\_path: try:

with open(file\_path, 'r', encoding='utf-8') as file: message = file.read()

self.message\_text.setText(message) if self.original\_image:

width, height = self.original\_image.size max\_chars = (width \* height) // 16

if len(message) > max\_chars: QMessageBox.warning(self, "Предупреждение",

f"Сообщение слишком длинное!

Максимальная длина: {max\_chars} символов.\n"

f"Сообщение будет обрезано.")

except Exception as e:

QMessageBox.critical(self, "Ошибка", f"Не удалось загрузить сообщение: {str(e)}")

def interpolation\_method(self, img\_array, message\_bits):

"""Реализация метода сокрытия данных с использованием интерполяции""" height, width = img\_array.shape

stego\_array = img\_array.copy() message\_length = len(message\_bits) length\_bits = format(message\_length, '032b')

full\_message = [int(bit) for bit in length\_bits] + message\_bits

# Встраивание данных idx = 0

for i in range(height): for j in range(width):

if idx < len(full\_message):

# Получаем текущий пиксель pixel\_value = stego\_array[i, j]

# Применяем интерполяцию if j < width - 1:

next\_pixel\_value = stego\_array[i, j + 1] interpolated\_value = (pixel\_value + next\_pixel\_value) //

2

if full\_message[idx] == 1:

if interpolated\_value % 2 == 0: stego\_array[i, j] += 1

else:

if interpolated\_value % 2 == 1: stego\_array[i, j] -= 1

idx += 1

if idx >= len(full\_message): break

if idx >= len(full\_message): break

return stego\_array

def save\_stego\_image(self): """Сохранение стегоконтейнера""" if not self.stego\_image:

QMessageBox.warning(self, "Ошибка", "Стегоконтейнер не создан!") return

file\_path, \_ = QFileDialog.getSaveFileName(self, "Сохранить стегоконтейнер", "", "BMP Files (\*.bmp)")

if file\_path: try:

# Убедимся, что файл имеет расширение .bmp if not file\_path.lower().endswith('.bmp'):

file\_path += '.bmp'

self.stego\_image.save(file\_path)

QMessageBox.information(self, "Успех", f"Стегоконтейнер успешно сохранен в {file\_path}")

except Exception as e:

QMessageBox.critical(self, "Ошибка", f"Не удалось сохранить стегоконтейнер: {str(e)}")

def update\_zoom(self):

"""Обновление масштаба изображений""" self.zoom\_factor = self.zoom\_slider.value()

if self.original\_image:

self.display\_image(self.original\_image, self.original\_image\_label) if self.stego\_image:

self.display\_image(self.stego\_image, self.stego\_image\_label)

def text\_to\_bits(self, text):

"""Преобразует текст в последовательность бит""" bits = []

for char in text:

# Преобразование символа в 8 бит byte = format(ord(char), '08b') for bit in byte:

bits.append(int(bit)) return bits

def display\_image(self, img, label): """Отображение изображения с учетом масштаба""" if img:

width, height = img.size

scaled\_width = int(width \* self.zoom\_factor) scaled\_height = int(height \* self.zoom\_factor) img\_array = np.array(img)

h, w = img\_array.shape bytes\_per\_line = w

q\_img = QImage(img\_array.data, w, h, bytes\_per\_line, QImage.Format.Format\_Grayscale8)

pixmap = QPixmap.fromImage(q\_img)

pixmap = pixmap.scaled(scaled\_width, scaled\_height, Qt.AspectRatioMode.KeepAspectRatio)

label.setPixmap(pixmap) label.resize(pixmap.size())

def calculate\_psnr(self, original, stego): """Расчет PSNR (Peak Signal-to-Noise Ratio)""" mse = np.mean((original - stego) \*\* 2)

if mse == 0:

return float('inf') max\_pixel = 255.0

psnr = 20 \* math.log10(max\_pixel / math.sqrt(mse)) return psnr

def main():

app = QApplication(sys.argv) window = SteganographyApp() window.show() sys.exit(app.exec())

if name == " main ": main()