**Министерство науки и высшего образования Российской Федерации**

**ФГАОУ ВО «Севастопольский государственный университет»**

Институт информационных технологий

Кафедра «Информационные системы»

**Отчет по лабораторной работе №7**

по дисциплине «Технологии обработки информации»

Выполнил: студент группы

ИС/б-21-2-о

Мельничук В.В.

Принял:

г. Севастополь

**Лабораторная работа №7**

Тема: Алгоритмы сжатия данных

Цель работы: реализовать алгоритмы сжатия для графической информации RLE, алгоритм сжатия с потерями.

Ход работы:

1. Алгоритм RLE

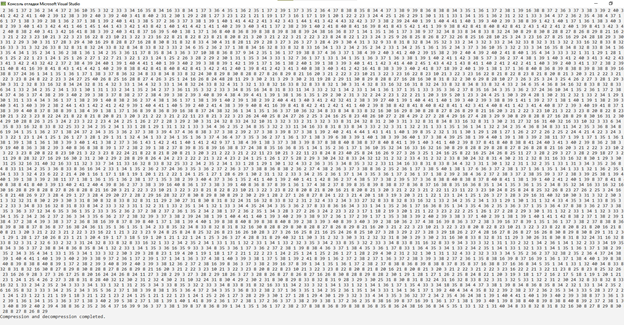


Рисунок 1 – Работа программы



Рисунок 2 – Входные данные



Рисунок 3 – Выходные данные

Листинг:

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Drawing.Imaging;

using System.IO;

public static class RleCompression

{

public static byte[] Compress(Bitmap image)

{

List<byte> compressedData = new List<byte>();

for (int y = 0; y < image.Height; y++)

{

byte currentPixel = image.GetPixel(0, y).R;

int count = 1;

for (int x = 1; x < image.Width; x++)

{

byte pixel = image.GetPixel(x, y).R;

if (pixel == currentPixel && count < 255)

{

count++;

}

else

{

compressedData.Add((byte)count);

compressedData.Add(currentPixel);

currentPixel = pixel;

count = 1;

}

}

compressedData.Add((byte)count);

compressedData.Add(currentPixel);

}

return compressedData.ToArray();

}

public static Bitmap Decompress(byte[] compressedData, int width, int height)

{

Bitmap decompressedImage = new Bitmap(width, height);

int dataIndex = 0;

for (int y = 0; y < height; y++)

{

for (int x = 0; x < width; x++)

{

byte count = compressedData[dataIndex++];

byte pixel = compressedData[dataIndex++];

for (int i = 0; i < count; i++)

{

decompressedImage.SetPixel(x + i, y, Color.FromArgb(pixel, pixel, pixel));

}

x += count - 1;

}

}

return decompressedImage;

}

public static void Main(string[] args)

{

// Load an image for compression

Bitmap originalImage = new Bitmap("D:/15\_.jpg");

// Compress the image

byte[] compressedData = Compress(originalImage);

// Print the compressed data

Console.WriteLine("Compressed data:");

foreach(byte data in compressedData)

{

Console.Write(data + " ");

}

Console.WriteLine();

// Decompress the data

Bitmap decompressedImage = Decompress(compressedData, originalImage.Width, originalImage.Height);

// Save the decompressed image

decompressedImage.Save("D:/16\_.jpg");

Console.WriteLine("Compression and decompression completed.");

}

}

1. Алгоритм сжатия с потерями



Рисунок 4 – Входные данные



Рисунок 5 – Выходные данные

Листинг:

using System;

using System.Drawing;

using System.Drawing.Imaging;

class JPEGCompression

{

static void Main()

{

string inputFile = "D:/15.jpg";

string compressedFile = "D:/17.jpg";

Compress(inputFile, compressedFile);

Console.WriteLine("Файл успешно сжат!");

}

static void Compress(string inputFile, string compressedFile)

{

using (Bitmap inputImage = new Bitmap(inputFile))

{

// Создаем объект кодека для формата JPEG

ImageCodecInfo jpegCodec = GetEncoderInfo(ImageFormat.Jpeg);

// Создаем параметры сжатия для JPEG

EncoderParameters encoderParams = new EncoderParameters(1);

encoderParams.Param[0] = new EncoderParameter(Encoder.Quality, 5L); // Здесь можно настроить качество сжатия

// Сохраняем сжатое изображение в файл

inputImage.Save(compressedFile, jpegCodec, encoderParams);

}

}

// Вспомогательный метод для получения информации о кодеке изображения

static ImageCodecInfo GetEncoderInfo(ImageFormat format)

{

ImageCodecInfo[] codecs = ImageCodecInfo.GetImageEncoders();

foreach(ImageCodecInfo codec in codecs)

{

if (codec.FormatID == format.Guid)

{

return codec;

}

}

return null;

}

}

**Вывод**: в ходе выполнения лабораторной работы были реализовали два способа сжатия графической информации на языке С++. Реализованы алгоритмы сжатия для графической информации RLE, а также алгоритм сжатия с потерями.