Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий

Кафедра ИС

ОТЧЁТ

по лабораторной работе №7

АЛГОРИТМЫ СЖАТИЯ ДАННЫХ

Выполнил:

ст. гр. ИС/б-21-2-о

Шевелёв К. С.

Проверил:

Минкин С. И.

Севастополь

2024

1. ЦЕЛЬ РАБОТЫ

Реализовать алгоритмы сжатия для графической информации RLE, алгоритм сжатия с потерями.

1. ХОД РАБОТЫ
   1. Алгоритм RLE

Листинг 1 – Код программы

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)

{

Bitmap originalImage = new Bitmap("C:/University/ТОИ/lab7/example1.1.jpg");

byte[] compressedData = Compress(originalImage);

Console.WriteLine("Сжатые данные:");

foreach (byte data in compressedData)

{

Console.Write(data + " ");

}

Console.WriteLine();

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

decompressedImage.Save("C:/University/ТОИ/lab7/example2.1.jpg");

Console.WriteLine("Сжатие и распаковка завершены.");

}

}



Рисунок 1 – Изображение до сжатия



Рисунок 2 – Изображение после сжатия

Изначальный размер файла был 237 КБ, а стал 1069 КБ. Это произошло из-за того, алгоритм RLE не всегда эффективен для сжатия изображений, особенно если на изображении много различных цветов и нет больших областей одного и того же цвета.

Алгоритм RLE работает лучше всего на изображениях, где есть большие области одного и того же цвета. Если на изображении много различных цветов и нет больших областей одного и того же цвета, то сжатие RLE может даже увеличить размер файла.

Так, на рис. 3 показано изображение, которое весило 227 КБ.

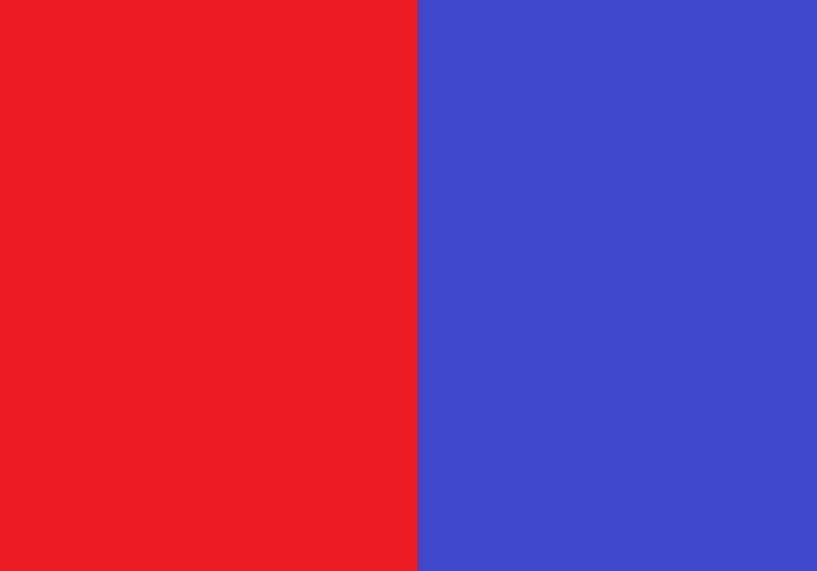


Рисунок 3 – Изображение до сжатия

На рис. 4 показано изображение, которое прошло сжатие и стало весить 76 КБ.

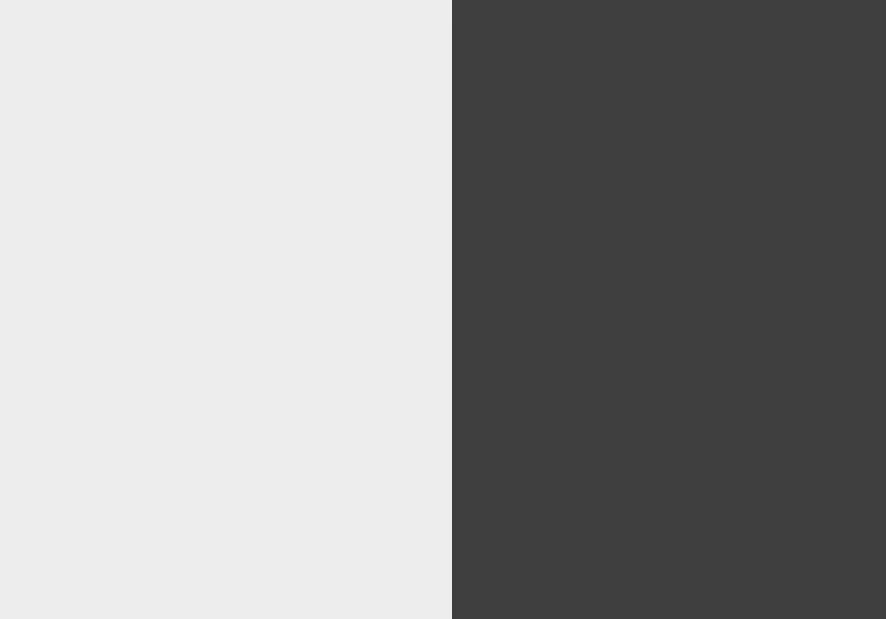


Рисунок 4 – Изображение после сжатия

* 1. Алгоритм сжатия с потерями

Листинг 2 – Код программы

using System;

using System.Drawing;

using System.Drawing.Imaging;

class JPEGCompression

{

static void Main()

{

string inputFile = "C:/University/ТОИ/lab7/example.jpg";

string compressedFile = "C:/University/ТОИ/lab7/example0.1.jpg";

Compress(inputFile, compressedFile);

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

}

static void Compress(string inputFile, string compressedFile)

{

using (Bitmap inputImage = new Bitmap(inputFile))

{

ImageCodecInfo jpegCodec = GetEncoderInfo(ImageFormat.Jpeg);

EncoderParameters encoderParams = new EncoderParameters(1);

encoderParams.Param[0] = new EncoderParameter(Encoder.Quality, 5L);

inputImage.Save(compressedFile, jpegCodec, encoderParams);

}

}

static ImageCodecInfo GetEncoderInfo(ImageFormat format)

{

ImageCodecInfo[] codecs = ImageCodecInfo.GetImageEncoders();

foreach (ImageCodecInfo codec in codecs)

{

if (codec.FormatID == format.Guid)

{

return codec;

}

}

return null;

}

}

При сжатии с потерей качество цветовая палитра остается почти в таком же виде, но заметно ухудшается качество изображения.



Рисунок 5 – Изображение до сжатия



Рисунок 6 – Изображение после сжатия

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.