МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
КАФЕДРА «ЭВМ и системы»

ОТЧЁТ

по лабораторным работам № 3 и № 4

Фильтрация изображения от импульсных помех, нелинейные методы контрастирования

Листов **15**

Выполнил студент группы Э-56
Козей Д. А.

Проверил Дубицкий А. В.

Брест 2020

Цель работы: Изучить методы выделения конутрных признаков изображения, а также методы фильтрации изображения от импульсных помех.

Задание: Составить программу, выполняющую фильтрацию изображения от импульсных помех медианным фильтром. окно 1xN И Nx1. Составить программу, выполняющую выделение контурных признаков изображения с помощью оператора Кирша.

Код программы:

```
using System;
using System.Drawing;
using System.Windows.Forms;
using System.IO;

namespace FormApp
{
    public partial class Form1 : Form
    {
        public int N;
        public Bitmap original, work;
        public String bfType;
        public Int32 bfSize;
        public Int16 bfReserved1;
        public Int16 bfreserved2;
        public Int32 bfOffBits;
        public Int32 bfSizeheader;
        public Int32 bfShirinaImage;
        public Int32 bfVisotaImage;
        public Int16 bfNumberPlosk;
        public Int16 bfBitPixel;
        public Int32 bfCompress;
        public Int32 bfSizeRastMass;
        public Int32 bfGorSize;
        public Int32 bfVertSize;
        public Int32 bfNumberColors;
```

```
public Int32 bfMainColors;

public Form1()
{
    InitializeComponent();
}

private void button1_Click(object sender,
  EventArgs e)
{
    openFileDialog1.Filter = "bmp |*.bmp";
    openFileDialog1.ShowDialog();
    BinaryReader bReader = new BinaryReader(File.
      Open(openFileDialog1.FileName, FileMode.Open
      ));
    bfType = new string(bReader.ReadChars(2));
    bfSize = bReader.ReadInt32();
    bfReserved1 = bReader.ReadInt16();
    bfreserved2 = bReader.ReadInt16();
    bfOffBits = bReader.ReadInt32();
    bfSizeheader = bReader.ReadInt32();
    bfShirinaImage = bReader.ReadInt32();
    bfVisotaImage = bReader.ReadInt32();
    bfNumberPlosk = bReader.ReadInt16();
    bfBitPixel = bReader.ReadInt16();
    bfCompress = bReader.ReadInt32();
    bfSizeRastMass = bReader.ReadInt32();
    bfGorSize = bReader.ReadInt32();
    bfVertSize = bReader.ReadInt32();
    bfNumberColors = bReader.ReadInt32();
    bfMainColors = bReader.ReadInt32();
    bReader.Close();
    String CompressType = 0.ToString();
```

```
if (bfCompress == 0 || bfCompress == 3 ||
   bfCompress == 6)
     CompressType = "Без сжатия";
else if (bfCompress == 1 || bfCompress == 2)
     CompressType = "RLE";
else if (bfCompress == 4)
     CompressType = "JPEG";
else if (bfCompress == 5)
     CompressType = "PNG";

Bitmap original_image = new Bitmap(
   openFileDialog1.FileName);
original = new Bitmap(openFileDialog1.FileName
   );
work = new Bitmap(openFileDialog1.FileName);
pictureBox1.Image = original_image;
pictureBox1.Show();
pictureBox2.Image = work;
pictureBox2.Show();

#region fileInfo
String message = "Сигнатура файла: " + bfType
   + "\n Размер файла: " + bfSize.ToString() +
                     "\n Местонахождение данных
                        растрового массива: " +
                        bfOffBits.ToString() +
                     "\n Длина заголовка
                        растрового массива: " +
                        bfSizeheader.ToString() +
                     "\n Ширина изобрадения: " +
                        bfShirinaImage.ToString() +
                         "\n Высота изображения: "
                       +
```

```
                bfVisotaImage.ToString() + "\
                    n Число цевтовых плоскостей
                    : " + bfNumberPlosk +
                "\n Бит/пиксел: " +
                    bfBitPixel + "\n Метод
                    сжатия: " + CompressType +
                "\n Длина растрового массива:
                     " + bfSizeRastMass + "\n
                    Горизонтальное разрешение:
                    " +
                bfGorSize + "\n Вертикальное
                    разрешение: " + bfVertSize
                    +
                "\n Количество цветов
                    изображения: " +
                    bfNumberColors + "\n
                    Количество основных цветов:
                     " +
                bfMainColors;

        MessageBox.Show(message);
        #endregion

    }
    #region Shum
    private void button2_Click(object sender,
        EventArgs e)
    {
        int lvlshum = 0;
        Random rand = new Random();
        if (radioButton1.Checked == true)
            lvlshum = 2000;
        else if (radioButton2.Checked == true)
```

```
            lvlshum = 7000;
        else if (radioButton3.Checked == true)
            lvlshum = 20000;
        for (int i = 0; i < lvlshum; i++)
        {
            work.SetPixel(rand.Next(work.Width), rand.
                Next(work.Height), Color.White);
        }
        pictureBox2.Refresh();
    }
    #endregion
    private void radioButton6_CheckedChanged(object
        sender, EventArgs e)
    {

    }

    private void button3_Click(object sender,
        EventArgs e)
    {
        N = Convert.ToInt32(textBox1.Text);
        if (radioButton4.Checked == true)
        {
            for (int i = 0; i < work.Width - N; i++)
            {
                for (int j = 0; j < work.Height; j++)
                {
                    median_filter(work, i, j);
                }
            }
            pictureBox2.Refresh();
        }
        if (radioButton5.Checked == true)
```

```
        {
            for (int i = 0; i < work.Width; i++)
            {
                for (int j = 0; j < work.Height - N; j
                    ++)
                {
                    median_filter2(work, i, j);
                }
            }
            pictureBox2.Refresh();
        }
        if (radioButton6.Checked == true)
            for (int i = 0; i < work.Width - N; i++)
            {
                for (int j = 0; j < work.Height - N; j
                    ++)
                {
                    median_filter(work, i, j);
                    median_filter2(work, i, j);
                }
            }
        pictureBox2.Refresh();
    }
    private void median_filter(Bitmap my_bitmap, int x
        , int y)
    {
        int cR_, cB_, cG_;
        int k = 0;

        int n = N;
        int[] cR = new int[n + 1];
        int[] cB = new int[n + 1];
        int[] cG = new int[n + 1];
```

```
        for (int i = 0; i < n + 1; i++)
        {
            cR[i] = 0;
            cG[i] = 0;
            cB[i] = 0;
        }
        for (int i = x; i < x + N; i++)
        {

            System.Drawing.Color c = my_bitmap.
               GetPixel(i, y);
            cR[k] = System.Convert.ToInt32(c.R);
            cG[k] = System.Convert.ToInt32(c.G);
            cB[k] = System.Convert.ToInt32(c.B);
            k++;
        }

        Array.Sort(cR);
        Array.Sort(cG);
        Array.Sort(cB);
        int n_ = (int)(n / 2) + 1;

        cR_ = cR[n_];
        cG_ = cG[n_];
        cB_ = cB[n_];

        my_bitmap.SetPixel(x, y, System.Drawing.Color.
           FromArgb(cR_, cG_, cB_));

    }
    private void median_filter2(Bitmap my_bitmap, int
       x, int y)
```

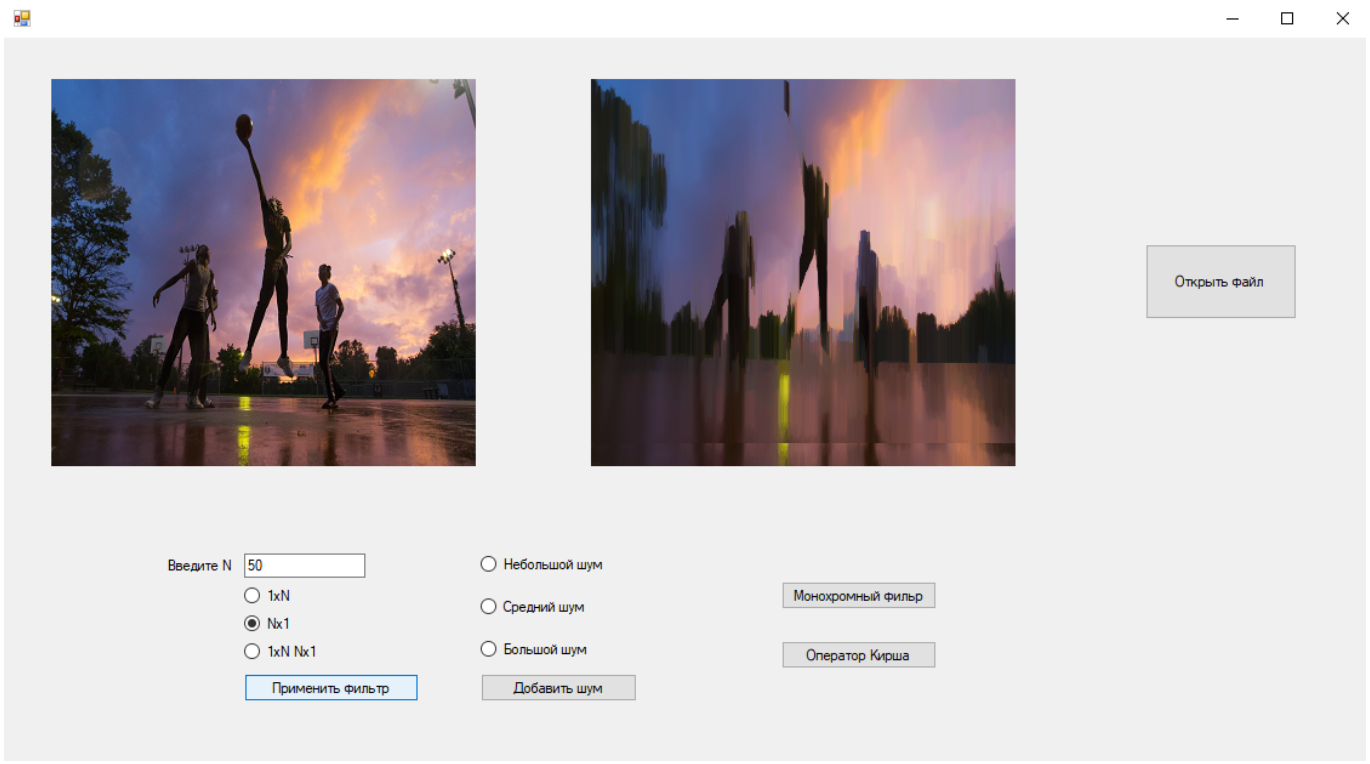
```
{
    int n;
    int cR_, cB_, cG_;
    int k = 0;

    n = N;

    int[] cR = new int[n + 1];
    int[] cB = new int[n + 1];
    int[] cG = new int[n + 1];

    for (int i = 0; i < n + 1; i++)
    {
        cR[i] = 0;
        cG[i] = 0;
        cB[i] = 0;
    }

    for (int j = y; j < y + N; j++)
    {
        System.Drawing.Color c = my_bitmap.
            GetPixel(x, j);
        cR[k] = System.Convert.ToInt32(c.R);
        cG[k] = System.Convert.ToInt32(c.G);
        cB[k] = System.Convert.ToInt32(c.B);
        k++;
    }



    Array.Sort(cR);
    Array.Sort(cG);
    Array.Sort(cB);
    int n_ = (int)(n / 2) + 1;
```

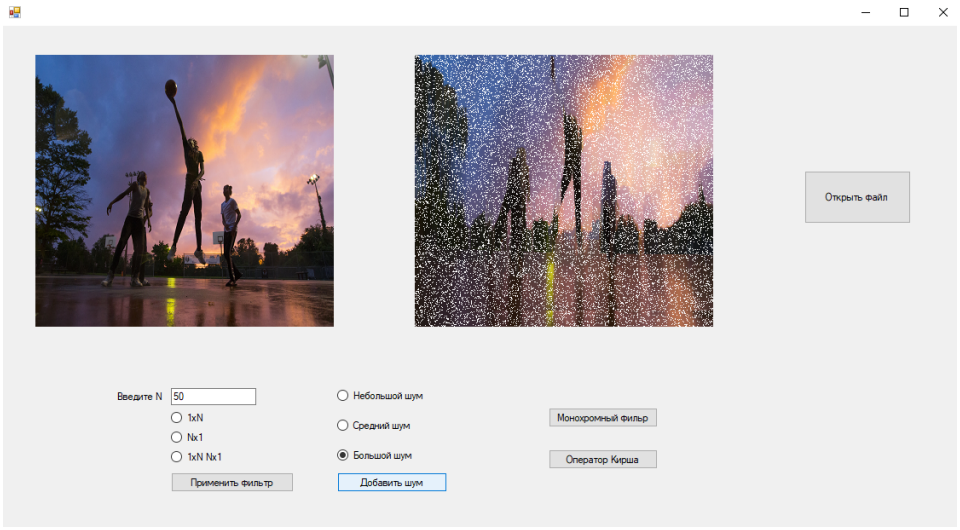
```
        cR_ = cR[n_];
        cG_ = cG[n_];
        cB_ = cB[n_];
        my_bitmap.SetPixel(x, y, System.Drawing.Color.
            FromArgb(cR_, cG_, cB_));


    }

    private void button4_Click(object sender,
        EventArgs e)
    {
        int R = 0;
        int G = 0;
        int B = 0;

        for (int i = 0; i < work.Width; i++)
        {
            for (int j = 0; j < work.Height; j++)
            {
                Color c = work.GetPixel(i, j);
                R = Convert.ToInt32(c.R);
                G = Convert.ToInt32(c.G);
                B = Convert.ToInt32(c.B);
                int sr = (R + G + B) / 3;
                work.SetPixel(i, j, Color.FromArgb(sr,
                    sr, sr));
            }
        }
        pictureBox2.Refresh();
    }
```

```
private void radioButton4_CheckedChanged(object
  sender, EventArgs e)
{

}

private void label1_Click(object sender, EventArgs
   e)
{

}

private void radioButton6_CheckedChanged_1(object
  sender, EventArgs e)
{

}

private void radioButton5_CheckedChanged(object
  sender, EventArgs e)
{

}

private void button5_Click(object sender,
  EventArgs e)
{
    int[] sb = new int[8];
    var mas = new[] { 0, 1, 2, 3, 4, 5, 6, 7 };
    for (int i = 1; i < work.Width - 1; i++)
        for (int j = 1; j < work.Height - 1; j++)
        {
            int ti = i, tj = j;
```

```
Color c = work.GetPixel(ti - 1, tj -
    1);
sb[0] = Convert.ToInt32(c.
    GetBrightness());
c = work.GetPixel(ti - 1, tj);
sb[0] = Convert.ToInt32(c.
    GetBrightness());
c = work.GetPixel(ti - 1, tj + 1);
sb[0] = Convert.ToInt32(c.
    GetBrightness());
c = work.GetPixel(ti, tj - 1);
sb[0] = Convert.ToInt32(c.
    GetBrightness());
c = work.GetPixel(ti, tj + 1);
sb[0] = Convert.ToInt32(c.
    GetBrightness());
c = work.GetPixel(ti + 1, tj - 1);
sb[0] = Convert.ToInt32(c.
    GetBrightness());
c = work.GetPixel(ti + 1, tj);
sb[0] = Convert.ToInt32(c.
    GetBrightness());
c = work.GetPixel(ti + 1, tj + 1);
sb[0] = Convert.ToInt32(c.
    GetBrightness());

int[] MAX = new int[8];
for (int l = 0; l < 8; l++)
{
    int S = sb[mas[0]] + sb[mas[1]] +
        sb[mas[2]];
```

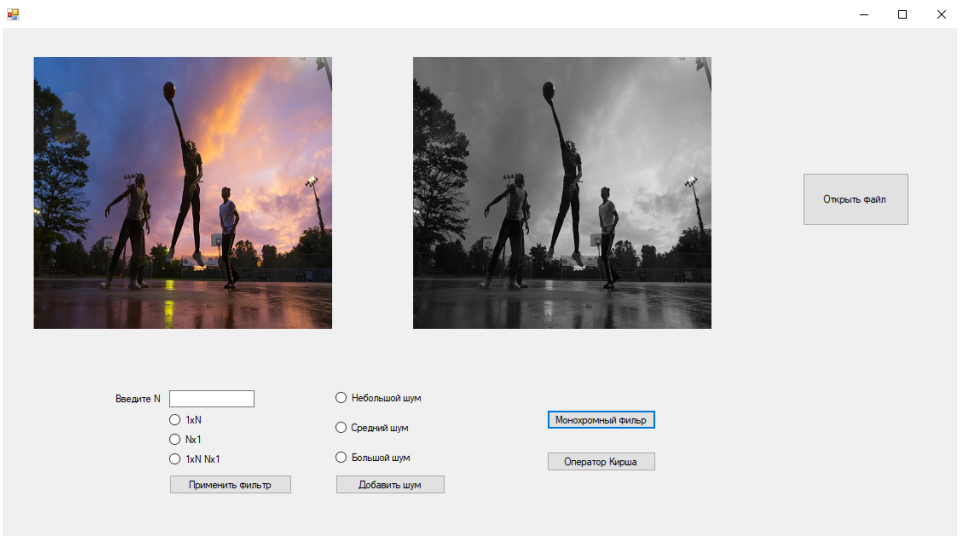
```
                        int T = sb[mas[3]] + sb[mas[4]] +
                            sb[mas[5]] + sb[mas[6]] + sb[mas
                            [7]];
                        MAX[1] = Math.Abs(5 * S - 3 * T);
                            //Формула Кирша
                        int temp = mas[7];
                        for (int z = 7; z > 0; z--)
                        {
                            mas[z] = mas[z - 1];
                        }
                        mas[0] = temp;
                    }
                    Array.Sort(MAX);
                    if (MAX[7] > 1)
                        work.SetPixel(i, j, Color.FromArgb
                            (0, 0, 0));
                    else
                        work.SetPixel(i, j, Color.FromArgb
                            (255, 255, 255));

                }
            pictureBox2.Refresh();
        }
    }
}
```

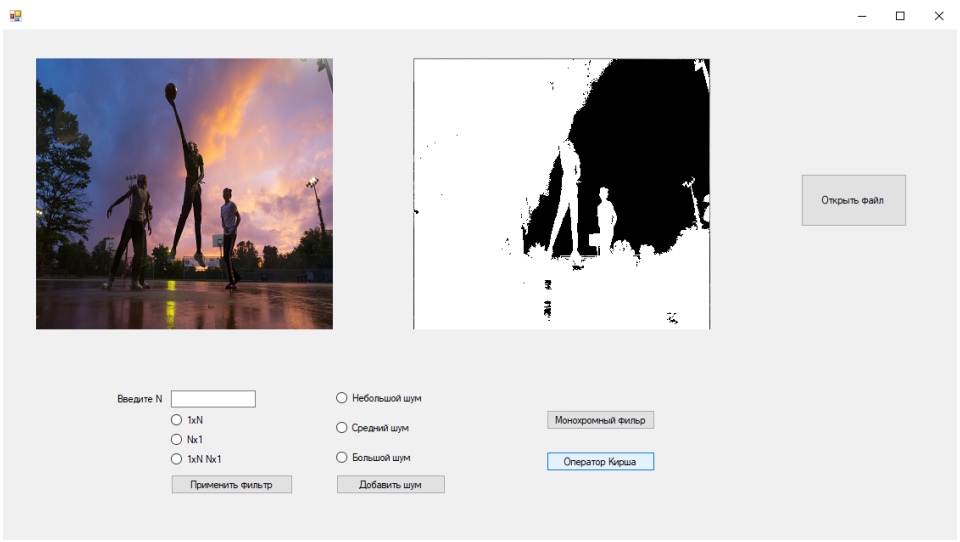




По результатам работы программы можно увидеть, что медианный фильтр представляет собой оконный фильтр, последовательно скользящий по массиву сигнала, и возвращающий на каждом шаге один из элементов, попавших в окно (апертуру) фильтра.

Обнаружение границ этим методом Кирша. Алгоритм основан на использовании всего одной маски, которую вращают по восьми главным направлениям: север, северо-запад, запад, юго-запад, юг, юго-восток, восток и северо-восток. Величина границы определена как максимальное значение, найденное с помощью маски. Определенное маской направление выдает максимальную величину.





Вывод: Составили программу, выполняющую фильтрацию изображения от импульсных помех медианным фильтром. окно 1xN И Nx1. Составили программу, выполняющую выделение контурных признаков изображения с помощью оператора Кирша.