

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ
КАФЕДРА КОМП'ЮТЕРНИХ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ І МЕРЕЖ

Протокол

Лабораторна робота №5

На тему “Построение цветowych гистограмм изображения. Гистограммная коррекция изображений”

По предмету: «Цифрова обробка сигналів та зображень»

Виконав:

студент групи АМ-182

Борщов М.І.

Перевірив:

Защолкін К. В.

Одеса 2022

ЗАДАНИЯ ДЛЯ РОБОТЫ

Мета. Основы программной обработки растровых изображений.

Задания. В данной лабораторной работе необходимо разработать приложение, реализующее три рассмотренные метода коррекции изображения (гистограммная линейная коррекция; метод «Идеальный отражатель», метод усреднения) и изменение яркости изображения. Рекомендуется последовательно реализовать следующие функциональности приложения.

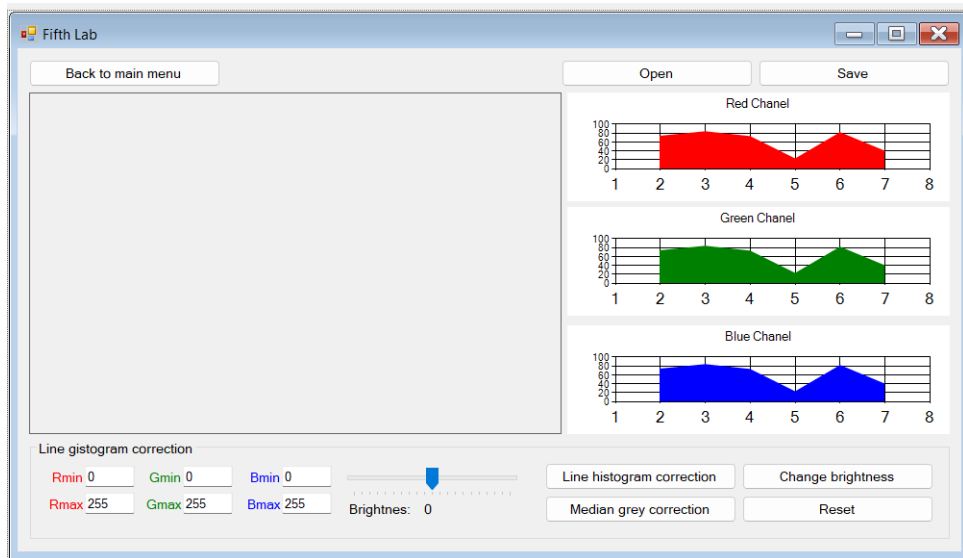
Функциональность №1: по нажатию кнопки «Открыть» приложение активизирует диалог открытия графических файлов; после выбора файла, изображение отображается на форме приложения посредством компонента `pictureBox`; одновременно с этим на форме приложения посредством компонентов `Chart` отображаются три цветовые гистограммы изображения; процесс вычисления значений гистограмм должен сопровождаться отображением доли выполненной работы при помощи компонента `progressBar`.

Функциональность №2: пользователь имеет возможность ввести значения `xmin` и `xmax` для выражения (5.1) по каждому цветовому каналу через компоненты `textBox`. По нажатию кнопки «Выполнить» приложение выполняет линейную гистограммную коррекцию изображения, выводит откорректированное изображение на форму приложения и отображает цветовые гистограммы откорректированного изображения. Для отображения откорректированного изображения и его гистограмм можно использовать те же компоненты `pictureBox` и `Chart`, что и для исходного изображения. Возможный интерфейс приложения, реализующего данную функциональность показан на рис. 5.8.

Функциональность №3: пользователь имеет возможность выполнить коррекцию по методу «Медианный серый цвет» (метод усреднения) путем нажатия на соответствующую кнопку. При этом приложение выполняет коррекцию изображения, выводит откорректированное изображение на форму приложения и отображает цветовые гистограммы откорректированного изображения.

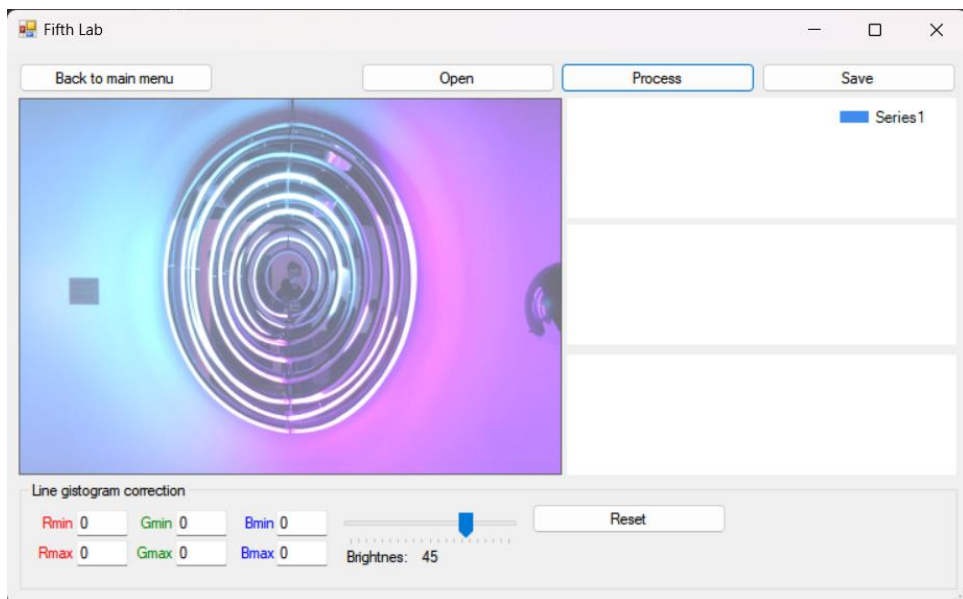
Функциональность №4: пользователь имеет возможность выполнить изменение яркости изображения при помощи компонента `TrackBar`, диапазон значений которого определен как $-150 \dots 150$.

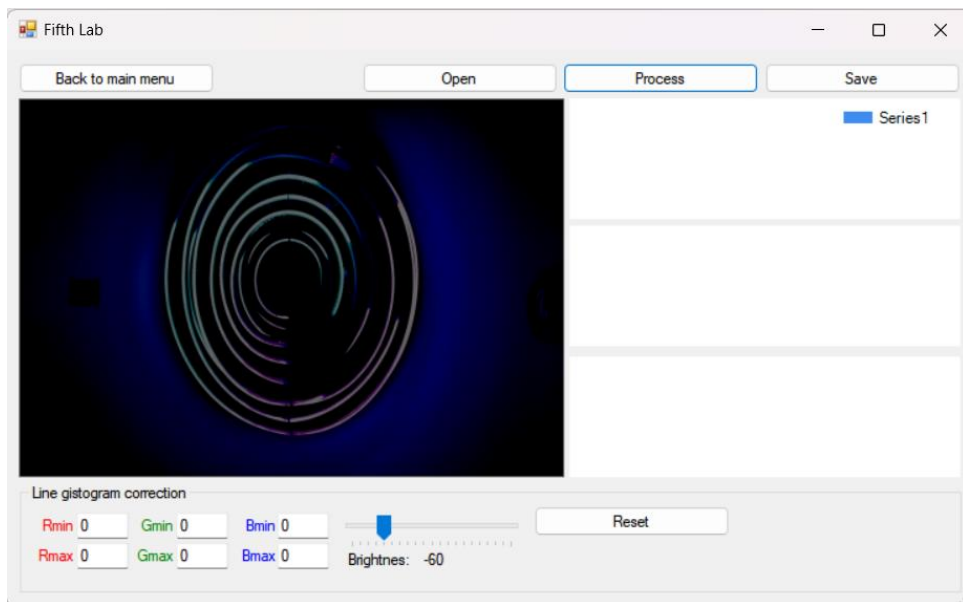
Хід роботи



Дизайн програми

Діапазон яскравості буде задано від -100 до 100 відсотків для більшого діапазону результату.





Результат зміни яскравості

```
for (var i = 0; i < _originalImage.Width; i++)
for (var j = 0; j < _originalImage.Height; j++)
{
    var col1:Color = _originalImage.GetPixel(x:i, y:j);

    var r:byte = col1.R;
    var g:byte = col1.G;
    var b:byte = col1.B;

    //future color processing

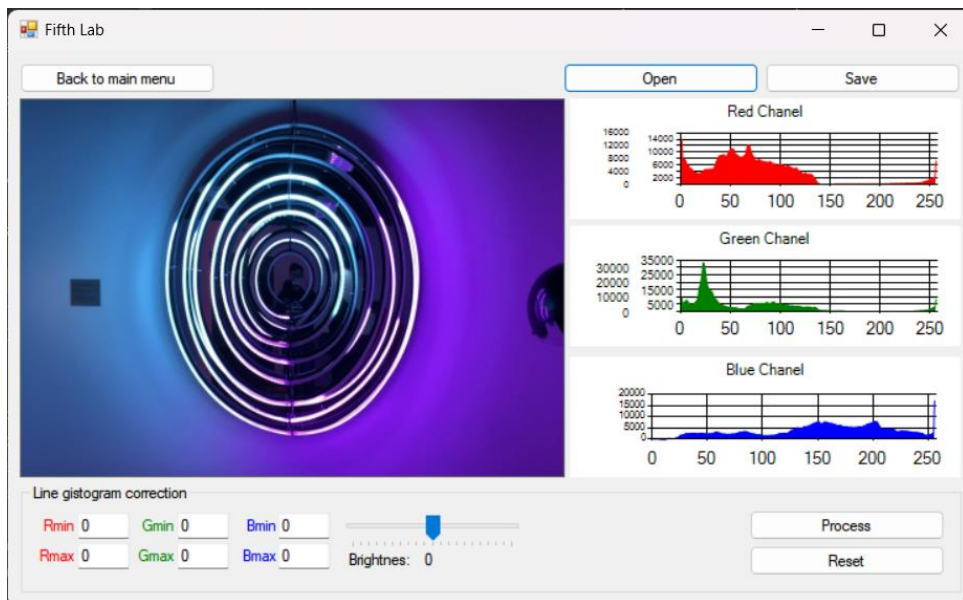
    if (brightnessTrackBar.Value != 0)
    {
        r = (byte)AdjustColorBrightness(r);
        g = (byte)AdjustColorBrightness(g);
        b = (byte)AdjustColorBrightness(b);
    }

    _generatedImage.SetPixel(x:i, y:j, Color.FromArgb(alpha:255, red:r, green:g, blue:b));
}
```

Код обробки зображення

```
private int AdjustColorBrightness(int pixelColor)
{
    var brightness:double = brightnessTrackBar.Value * 2.55; // trackbar value is in percents
    var pixelResult:double = pixelColor + brightness; // if brightness < 0 then + (- value)
    if (pixelResult < 0)
        pixelResult = 0;
    else if (pixelResult > 255)
        pixelResult = 255;
    return (int)pixelResult;
}
```

Код функції зміни яскравості



Вивід гістограм

```
private void openButton_Click(object sender, EventArgs e)
{
    if (openFileDialog.ShowDialog() != DialogResult.OK) return;
    _originalImage = new Bitmap(stream: openFileDialog.OpenFile());
    originalPicture.Image = _originalImage;
    GenerateHistograms(_originalImage);
}
```

Запуск виводу

```
private void GenerateHistograms(Bitmap targetPicture)
{
    for (var i = 0; i < targetPicture.Width; i++)
        for (var j = 0; j < targetPicture.Height; j++)
            FillHistograms(targetPicture.GetPixel(x: i, y: j));

    SetHistograms(redChanelChart, _histogramRed);
    SetHistograms(greenChanelChart, _histogramGreen);
    SetHistograms(blueChanelChart, _histogramBlue);
}
```

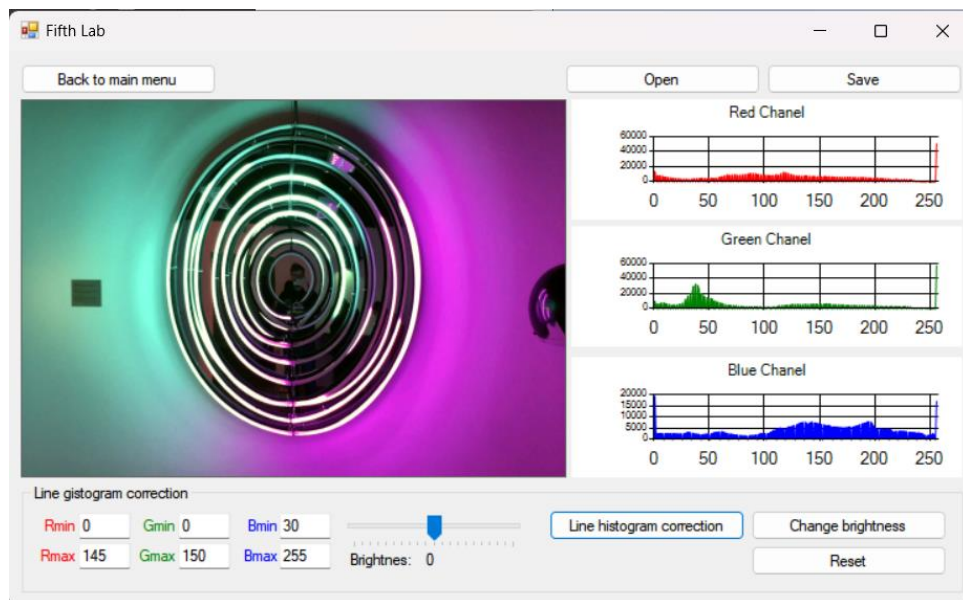
Функція генерація гістограм

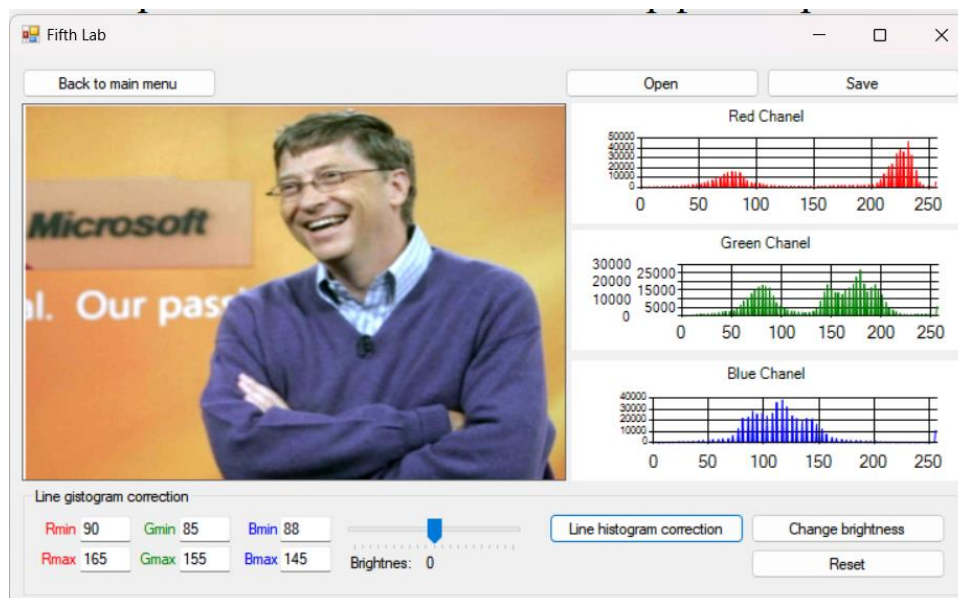
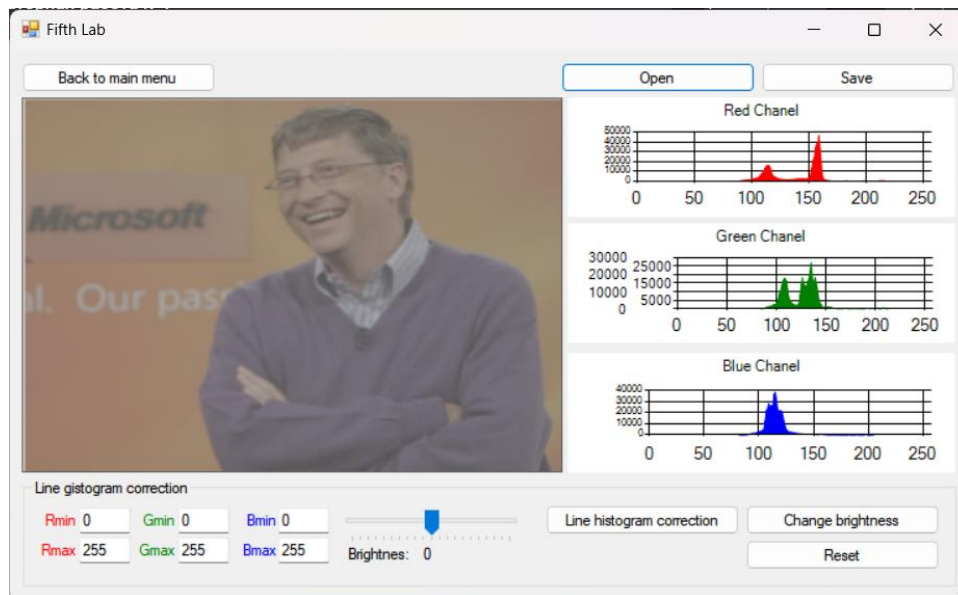
```
private void FillHistograms(Color pixel)
{
    _histogramRed[pixel.R]++;
    _histogramGreen[pixel.G]++;
    _histogramBlue[pixel.B]++;
}
```

Функція заповнення масивів

```
private static void SetHistograms(Chart colorChart, IList<int> colorArray)
{
    colorChart.Series["Series1"].Points.Clear();
    foreach (var t:int in colorArray)
        colorChart.Series["Series1"].Points.AddY(t);
}
```

Функція виводу гістограм на екран





Результат лінійної корекція

```

if (_originalImage == null) return;

_generatedImage = new Bitmap(_originalImage);

for (var i = 0; i < _originalImage.Width; i++)
for (var j = 0; j < _originalImage.Height; j++)
{
    var col1:Color = _originalImage.GetPixel(x:i, y:j);

    var r:int = CalculateColorByHistogram(
        color: col1.R,
        minValue: Convert.ToInt32(redMinTextBox.Text),
        maxValue: Convert.ToInt32(redMaxTextBox.Text));
    var g:int = CalculateColorByHistogram(
        color: col1.G,
        minValue: Convert.ToInt32(greenMinTextBox.Text),
        maxValue: Convert.ToInt32(greenMaxTextBox.Text));
    var b:int = CalculateColorByHistogram(
        color: col1.B,
        minValue: Convert.ToInt32(blueMinTextBox.Text),
        maxValue: Convert.ToInt32(blueMaxTextBox.Text));

    _generatedImage.SetPixel(x:i, y:j, Color.FromArgb(alpha: 255, red: r, green: g, blue: b));
}
GenerateHistograms(_generatedImage);
originalPicture.Image = _generatedImage;

```

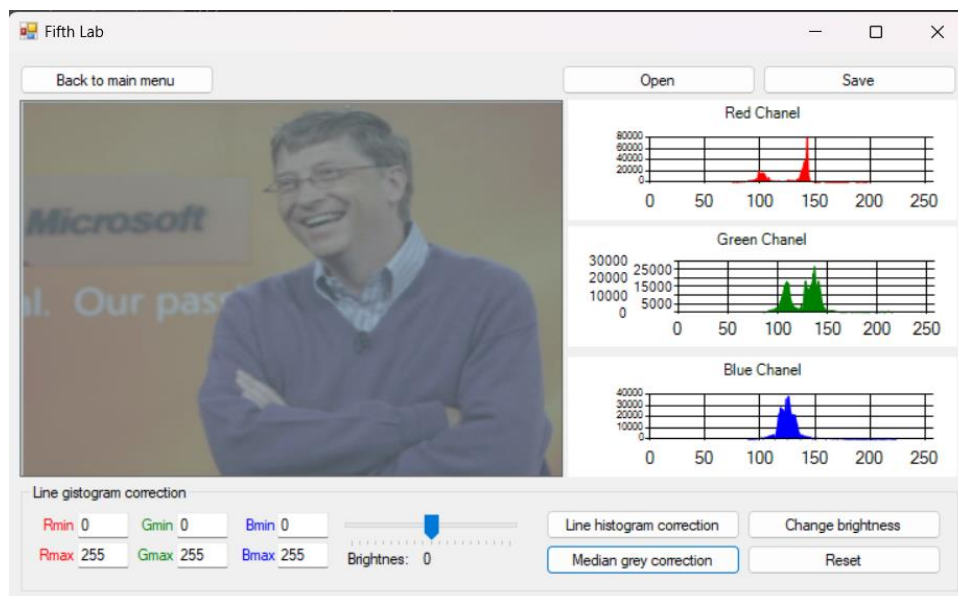
Код лінійної корекції

```

private static int CalculateColorByHistogram(int color, int minValue, int maxValue)
{
    var result:int = (color - minValue) * 255 / (maxValue - minValue);
    if (result < 0) return 0;
    return result > 255 ? 255 : result;
}

```

Функція розрахунку корекції



Результат корекції по сірому медіану


```

private void medianGreyCorrection_Click(object sender, EventArgs e)
{
    if (_originalImage == null) return;
    if (_generatedImage == null) _generatedImage = new Bitmap(_originalImage);

    var averageGreyLong = (_redMedium + _greenMedium + _blueMedium) / 3;

    for (var i = 0; i < _originalImage.Width; i++)
    for (var j = 0; j < _originalImage.Height; j++)
    {
        var col1:Color = _originalImage.GetPixel(x:i, y:j);

        var r:int = CalculateGreyMediumColor((int)averageGrey, targetColor.col1.R, (int)_redMedium);
        var g:int = CalculateGreyMediumColor((int)averageGrey, targetColor.col1.G, (int)_greenMedium);
        var b:int = CalculateGreyMediumColor((int)averageGrey, targetColor.col1.B, (int)_blueMedium);

        _generatedImage.SetPixel(x:i, y:j, Color.FromArgb(alpha:255, red:r, green:g, blue:b));
    }

    GenerateHistograms(_generatedImage);
    originalPicture.Image = _generatedImage;
}

```

Функція обробки по сірому медіану

```

private void FillMediumValues()
{
    _redBucket = 0;
    _greenBucket = 0;
    _blueBucket = 0;
    _pixelCount = 0;

    for (var i = 0; i < _originalImage.Width; i++)
    for (var j = 0; j < _originalImage.Height; j++)
    {
        var color = _originalImage.GetPixel(x:i, y:j);

        _pixelCount++;
        _redBucket += color.R;
        _greenBucket += color.G;
        _blueBucket += color.B;
    }

    _redMedium = _redBucket / _pixelCount;
    _greenMedium = _greenBucket / _pixelCount;
    _blueMedium = _blueBucket / _pixelCount;
}

```

Код розрахунку середнього значення кольору

```
private void openButton_Click(object sender, EventArgs e)
{
    if (openFileDialog.ShowDialog() != DialogResult.OK) return;
    _originalImage = new Bitmap(stream: openFileDialog.OpenFile());
    originalPicture.Image = _originalImage;
    GenerateHistograms(_originalImage);
    FillMediumValues();
}
```

Середнє значення усіх кольорів рахується при відкритті файлу

```
private static int CalculateGreyMediumColor(int greyColor, int targetColor, int targetColorMedium)
{
    int result;
    if (targetColorMedium == 0)
        result = targetColor * greyColor;
    else result = targetColor * greyColor / targetColorMedium;
    if (result < 0) return 0;
    return result > 255 ? 255 : result;
}
```

Код розрахунку по сірому медіану

Також були оброблені ситуації, коли зображення ще не було відкрите, але користувач намагається викликати інші функції

Код програми: <https://github.com/zeinlol/image-converter>