Дисципліна "Комп'ютерний синтез та обробка зображень"

тема "Оптимізація палітри і псевдо тонування зображення"

**Мета роботи:** вивчення методів оптимізації палітри зображення і методів псевдо тонування.

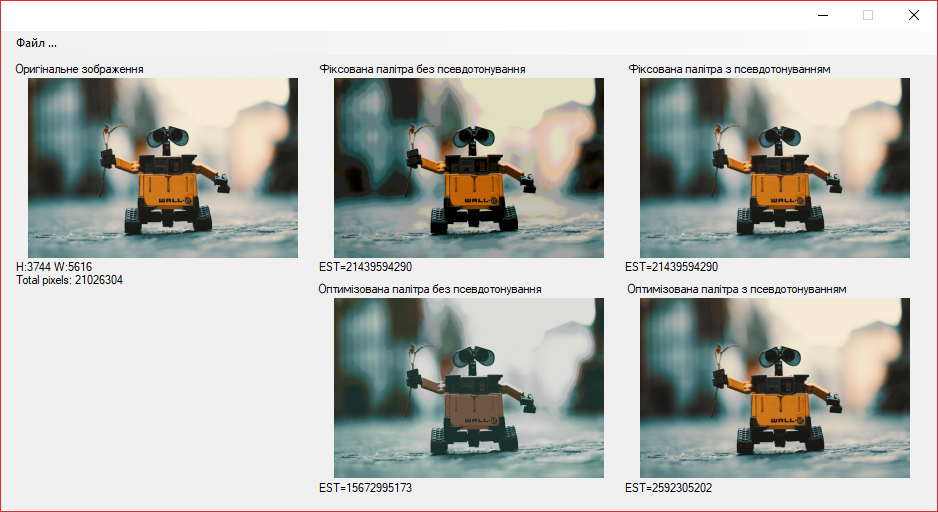
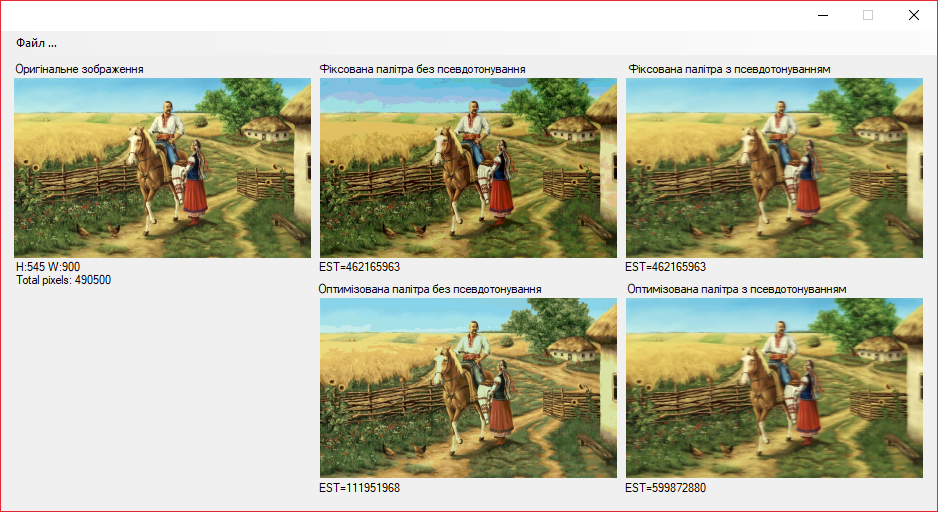
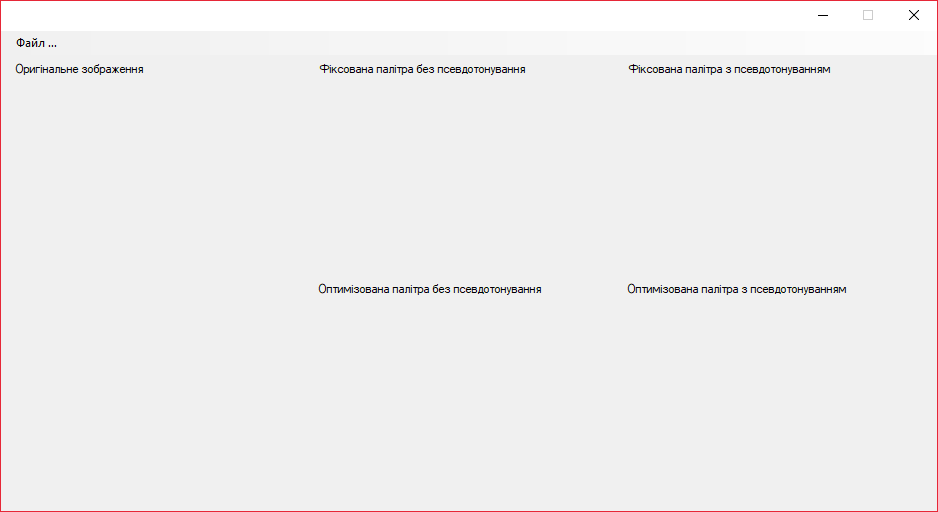
**Постановка задачі лабораторної роботи**

Реалізувати програмно перетворення початкового повно кольорового зображення (24 біта/піксель) в зображення з фіксованою і оптимізованою палітрою і псевдо тонуванням. При отриманні фіксованої палітри використовувати лінійне квантування, при отриманні оптимізованої палітри – метод популярності, при псевдо тонуванні – фільтр Флойда –Штейнбеpга (прохід по всіх рядках зліва направо).

Результати слід представити так, щоб була можливість візуально оцінити якість оптимізації палітри і псевдо тонування і порівняти реалізовані методи. У вікні програми одночасно повинні бути представлені початкове зображення і результати його перетворення: зображення з фіксованою палітрою з псевдо тонуванням і без нього, зображення з оптимізованою палітрою з псевдо тонуванням і без нього. Для формальної оцінки оптимізації палітри слід обчислити помилку в отриманому зображенні в порівнянні з початковим зображенням, використовуючи формулу квадратичного відхилення.

Кількість кольорів в палітрі після перетворення повинна дорівнювати 64.

**Інтерфейс розробленої програми**



**Вихідний код розробленої програми**

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace lab5

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new MainForm());

}

}

}

QBitmap.cs

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab5

{

static class QBitmap

{

public static Bitmap PrepareBitmap(Bitmap bitmap)

{

for (int x = 0; x < bitmap.Width; x++)

{

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

{

bitmap.SetPixel(x, y, Color.FromArgb(1, bitmap.GetPixel(x, y)));

}

}

return bitmap;

}

public static long ColorEST(Color lc, Color rc)

{

return (long)(Math.Pow(lc.R - rc.R, 2.0) + Math.Pow(lc.G - rc.G, 2.0) + Math.Pow(lc.B - rc.B, 2.0));

}

public static long BitmapEST(Bitmap lb, Bitmap rb)

{

long error = 0;

for (int x = 0; x < lb.Width; x++)

{

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

{

error += ColorEST(lb.GetPixel(x, y), rb.GetPixel(x, y));

}

}

return error;

}

public static Bitmap LinearQuantization(Bitmap bitmap, int numberOfColor)

{

Bitmap result = new Bitmap(bitmap.Width, bitmap.Height);

byte trimDegree = (byte)Math.Pow(2, ((24 - (byte)Math.Log(numberOfColor, 2)) / 3));

for (int i = 0; i < bitmap.Width; i++)

{

for (int j = 0; j < bitmap.Height; j++)

{

Color color = bitmap.GetPixel(i, j);

color = Color.FromArgb(color.A, color.R - color.R % trimDegree, color.G - color.G % trimDegree, color.B - color.B % trimDegree);

result.SetPixel(i, j, color);

}

}

return result;

}

public static Bitmap PopularQuantization(Bitmap bitmap, int numberOfColor)

{

return ApplyPalette(bitmap, GetPopularColors(bitmap, numberOfColor));

}

public static List<Color> GetPopularColors(Bitmap bitmap, int numberOfColor)

{

Dictionary<Color, long> colors = new Dictionary<Color, long>();

for (int x = 0; x < bitmap.Width; x++)

{

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

{

Color color = bitmap.GetPixel(x, y);

if (colors.Keys.Contains(color)) colors[color]++;

else colors.Add(color, 1);

}

}

return colors.OrderBy(entry => entry.Value).Select(x => x.Key).Take(numberOfColor).ToList();

}

public static Bitmap ApplyPalette(Bitmap bitmap, List<Color> palette)

{

Bitmap result = new Bitmap(bitmap.Width, bitmap.Height);

for (int x = 0; x < bitmap.Width; x++)

{

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

{

Color nowColor = bitmap.GetPixel(x, y);

Color newColor = palette[0];

double est = ColorEST(newColor, nowColor);

foreach (Color tryColor in palette)

{

double tryEST = ColorEST(nowColor, tryColor);

if (tryEST < est)

{

newColor = tryColor;

est = tryEST;

}

}

result.SetPixel(x, y, newColor);

}

}

return result;

}

public static Bitmap Tonning(Bitmap baseImage, Bitmap picture2)

{

Bitmap editImage = new Bitmap(baseImage.Width, baseImage.Height);

for (int x = 0; x < baseImage.Width; x++)

{

editImage.SetPixel(x, 0, picture2.GetPixel(x, 0));

}

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

{

for (int x = 0; x < baseImage.Width; x++)

{

Color color1 = baseImage.GetPixel(x, y);

Color color2 = editImage.GetPixel(x, y);

int errorR = color1.R - color2.R;

int errorG = color1.G - color2.G;

int errorB = color1.B - color2.B;

if (y != baseImage.Height - 1)

{

if (x != 0)

{

editImage.SetPixel(x - 1, y + 1, Color.FromArgb(color2.A, (int)(color2.R + 0.25 \* errorR),

(int)(color2.G + 0.25 \* errorG),

(int)(color2.B + 0.25 \* errorB)));

}

editImage.SetPixel(x, y + 1, Color.FromArgb(color2.A, (int)(color2.R + 0.25 \* errorR),

(int)(color2.G + 0.25 \* errorG),

(int)(color2.B + 0.25 \* errorB)));

}

if (x != baseImage.Width - 1)

{

if (y != baseImage.Height - 1)

{

editImage.SetPixel(x + 1, y + 1, Color.FromArgb(color2.A, (int)(color2.R + 0.25 \* errorR),

(int)(color2.G + 0.25 \* errorG),

(int)(color2.B + 0.25 \* errorB)));

}

editImage.SetPixel(x + 1, y, Color.FromArgb(color2.A, (int)(color2.R + 0.25 \* errorR),

(int)(color2.G + 0.25 \* errorG),

(int)(color2.B + 0.25 \* errorB)));

}

}

}

return editImage;

}

}

}

MainForm.Designer.cs

namespace lab5

{

partial class MainForm

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

this.MenuStrip = new System.Windows.Forms.MenuStrip();

this.fileToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.sourceImagePB = new System.Windows.Forms.PictureBox();

this.FPTYPicture = new System.Windows.Forms.PictureBox();

this.OPTYPicture = new System.Windows.Forms.PictureBox();

this.OPTNPicture = new System.Windows.Forms.PictureBox();

this.BasePictureLabel = new System.Windows.Forms.Label();

this.FPTNPictureLabel = new System.Windows.Forms.Label();

this.FPTYPictureLabel = new System.Windows.Forms.Label();

this.OPTNPictureLabel = new System.Windows.Forms.Label();

this.OPTYPictureLabel = new System.Windows.Forms.Label();

this.FPTNPictureESTLabel = new System.Windows.Forms.Label();

this.OPTNPictureESTLabel = new System.Windows.Forms.Label();

this.FPTYPictureESTLabel = new System.Windows.Forms.Label();

this.OPTYPictureESTLabel = new System.Windows.Forms.Label();

this.OpenFileDialog = new System.Windows.Forms.OpenFileDialog();

this.BasePicturePropertys = new System.Windows.Forms.Label();

this.FPTNPicture = new System.Windows.Forms.PictureBox();

this.MenuStrip.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.sourceImagePB)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.FPTYPicture)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.OPTYPicture)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.OPTNPicture)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.FPTNPicture)).BeginInit();

this.SuspendLayout();

//

// MenuStrip

//

this.MenuStrip.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.fileToolStripMenuItem});

this.MenuStrip.Location = new System.Drawing.Point(0, 0);

this.MenuStrip.Name = "MenuStrip";

this.MenuStrip.Size = new System.Drawing.Size(936, 24);

this.MenuStrip.TabIndex = 0;

this.MenuStrip.Text = "menuStrip1";

//

// fileToolStripMenuItem

//

this.fileToolStripMenuItem.Name = "fileToolStripMenuItem";

this.fileToolStripMenuItem.Size = new System.Drawing.Size(60, 20);

this.fileToolStripMenuItem.Text = "Файл …";

this.fileToolStripMenuItem.Click += new System.EventHandler(this.OpenToolStripMenuItem\_Click);

//

// sourceImagePB

//

this.sourceImagePB.Location = new System.Drawing.Point(12, 47);

this.sourceImagePB.Name = "sourceImagePB";

this.sourceImagePB.Size = new System.Drawing.Size(300, 180);

this.sourceImagePB.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;

this.sourceImagePB.TabIndex = 1;

this.sourceImagePB.TabStop = false;

//

// FPTYPicture

//

this.FPTYPicture.Location = new System.Drawing.Point(624, 47);

this.FPTYPicture.Name = "FPTYPicture";

this.FPTYPicture.Size = new System.Drawing.Size(300, 180);

this.FPTYPicture.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;

this.FPTYPicture.TabIndex = 3;

this.FPTYPicture.TabStop = false;

this.FPTYPicture.Click += new System.EventHandler(this.SaveOnPictureBox\_Click);

//

// OPTYPicture

//

this.OPTYPicture.Location = new System.Drawing.Point(624, 267);

this.OPTYPicture.Name = "OPTYPicture";

this.OPTYPicture.Size = new System.Drawing.Size(300, 180);

this.OPTYPicture.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;

this.OPTYPicture.TabIndex = 4;

this.OPTYPicture.TabStop = false;

this.OPTYPicture.Click += new System.EventHandler(this.SaveOnPictureBox\_Click);

//

// OPTNPicture

//

this.OPTNPicture.Location = new System.Drawing.Point(318, 267);

this.OPTNPicture.Name = "OPTNPicture";

this.OPTNPicture.Size = new System.Drawing.Size(300, 180);

this.OPTNPicture.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;

this.OPTNPicture.TabIndex = 5;

this.OPTNPicture.TabStop = false;

this.OPTNPicture.Click += new System.EventHandler(this.SaveOnPictureBox\_Click);

//

// BasePictureLabel

//

this.BasePictureLabel.AutoSize = true;

this.BasePictureLabel.Location = new System.Drawing.Point(12, 31);

this.BasePictureLabel.Name = "BasePictureLabel";

this.BasePictureLabel.Size = new System.Drawing.Size(135, 13);

this.BasePictureLabel.TabIndex = 6;

this.BasePictureLabel.Text = "Оригінальне зображення";

//

// FPTNPictureLabel

//

this.FPTNPictureLabel.AutoSize = true;

this.FPTNPictureLabel.Location = new System.Drawing.Point(315, 31);

this.FPTNPictureLabel.Name = "FPTNPictureLabel";

this.FPTNPictureLabel.Size = new System.Drawing.Size(214, 13);

this.FPTNPictureLabel.TabIndex = 7;

this.FPTNPictureLabel.Text = "Фіксована палітра без псевдотонування";

//

// FPTYPictureLabel

//

this.FPTYPictureLabel.AutoSize = true;

this.FPTYPictureLabel.Location = new System.Drawing.Point(624, 31);

this.FPTYPictureLabel.Name = "FPTYPictureLabel";

this.FPTYPictureLabel.Size = new System.Drawing.Size(210, 13);

this.FPTYPictureLabel.TabIndex = 8;

this.FPTYPictureLabel.Text = "Фіксована палітра з псевдотонуванням";

//

// OPTNPictureLabel

//

this.OPTNPictureLabel.AutoSize = true;

this.OPTNPictureLabel.Location = new System.Drawing.Point(315, 251);

this.OPTNPictureLabel.Name = "OPTNPictureLabel";

this.OPTNPictureLabel.Size = new System.Drawing.Size(230, 13);

this.OPTNPictureLabel.TabIndex = 9;

this.OPTNPictureLabel.Text = "Оптимізована палітра без псевдотонування";

//

// OPTYPictureLabel

//

this.OPTYPictureLabel.AutoSize = true;

this.OPTYPictureLabel.Location = new System.Drawing.Point(624, 251);

this.OPTYPictureLabel.Name = "OPTYPictureLabel";

this.OPTYPictureLabel.Size = new System.Drawing.Size(226, 13);

this.OPTYPictureLabel.TabIndex = 10;

this.OPTYPictureLabel.Text = "Оптимізована палітра з псевдотонуванням";

//

// FPTNPictureESTLabel

//

this.FPTNPictureESTLabel.AutoSize = true;

this.FPTNPictureESTLabel.Location = new System.Drawing.Point(315, 229);

this.FPTNPictureESTLabel.Name = "FPTNPictureESTLabel";

this.FPTNPictureESTLabel.Size = new System.Drawing.Size(0, 13);

this.FPTNPictureESTLabel.TabIndex = 11;

//

// OPTNPictureESTLabel

//

this.OPTNPictureESTLabel.AutoSize = true;

this.OPTNPictureESTLabel.Location = new System.Drawing.Point(315, 450);

this.OPTNPictureESTLabel.Name = "OPTNPictureESTLabel";

this.OPTNPictureESTLabel.Size = new System.Drawing.Size(0, 13);

this.OPTNPictureESTLabel.TabIndex = 12;

//

// FPTYPictureESTLabel

//

this.FPTYPictureESTLabel.AutoSize = true;

this.FPTYPictureESTLabel.Location = new System.Drawing.Point(621, 229);

this.FPTYPictureESTLabel.Name = "FPTYPictureESTLabel";

this.FPTYPictureESTLabel.Size = new System.Drawing.Size(0, 13);

this.FPTYPictureESTLabel.TabIndex = 13;

//

// OPTYPictureESTLabel

//

this.OPTYPictureESTLabel.AutoSize = true;

this.OPTYPictureESTLabel.Location = new System.Drawing.Point(621, 450);

this.OPTYPictureESTLabel.Name = "OPTYPictureESTLabel";

this.OPTYPictureESTLabel.Size = new System.Drawing.Size(0, 13);

this.OPTYPictureESTLabel.TabIndex = 14;

//

// OpenFileDialog

//

this.OpenFileDialog.FileName = "OpenFileDialog";

//

// BasePicturePropertys

//

this.BasePicturePropertys.AutoSize = true;

this.BasePicturePropertys.Location = new System.Drawing.Point(12, 229);

this.BasePicturePropertys.Name = "BasePicturePropertys";

this.BasePicturePropertys.Size = new System.Drawing.Size(0, 13);

this.BasePicturePropertys.TabIndex = 6;

//

// FPTNPicture

//

this.FPTNPicture.Location = new System.Drawing.Point(318, 47);

this.FPTNPicture.Name = "FPTNPicture";

this.FPTNPicture.Size = new System.Drawing.Size(300, 180);

this.FPTNPicture.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;

this.FPTNPicture.TabIndex = 2;

this.FPTNPicture.TabStop = false;

this.FPTNPicture.Click += new System.EventHandler(this.SaveOnPictureBox\_Click);

//

// MainForm

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(936, 480);

this.Controls.Add(this.OPTYPictureESTLabel);

this.Controls.Add(this.FPTYPictureESTLabel);

this.Controls.Add(this.OPTNPictureESTLabel);

this.Controls.Add(this.FPTNPictureESTLabel);

this.Controls.Add(this.OPTYPictureLabel);

this.Controls.Add(this.OPTNPictureLabel);

this.Controls.Add(this.FPTYPictureLabel);

this.Controls.Add(this.FPTNPictureLabel);

this.Controls.Add(this.BasePicturePropertys);

this.Controls.Add(this.BasePictureLabel);

this.Controls.Add(this.OPTNPicture);

this.Controls.Add(this.OPTYPicture);

this.Controls.Add(this.FPTYPicture);

this.Controls.Add(this.FPTNPicture);

this.Controls.Add(this.sourceImagePB);

this.Controls.Add(this.MenuStrip);

this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedDialog;

this.MainMenuStrip = this.MenuStrip;

this.MaximizeBox = false;

this.Name = "MainForm";

this.ShowIcon = false;

this.Text = "Lab5";

this.MenuStrip.ResumeLayout(false);

this.MenuStrip.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.sourceImagePB)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.FPTYPicture)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.OPTYPicture)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.OPTNPicture)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.FPTNPicture)).EndInit();

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.MenuStrip MenuStrip;

private System.Windows.Forms.ToolStripMenuItem fileToolStripMenuItem;

private System.Windows.Forms.PictureBox sourceImagePB;

private System.Windows.Forms.PictureBox FPTYPicture;

private System.Windows.Forms.PictureBox OPTYPicture;

private System.Windows.Forms.PictureBox OPTNPicture;

private System.Windows.Forms.Label BasePictureLabel;

private System.Windows.Forms.Label FPTNPictureLabel;

private System.Windows.Forms.Label FPTYPictureLabel;

private System.Windows.Forms.Label OPTNPictureLabel;

private System.Windows.Forms.Label OPTYPictureLabel;

private System.Windows.Forms.Label FPTNPictureESTLabel;

private System.Windows.Forms.Label OPTNPictureESTLabel;

private System.Windows.Forms.Label FPTYPictureESTLabel;

private System.Windows.Forms.Label OPTYPictureESTLabel;

private System.Windows.Forms.OpenFileDialog OpenFileDialog;

private System.Windows.Forms.Label BasePicturePropertys;

private System.Windows.Forms.PictureBox FPTNPicture;

}

}

MainForm.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace lab5

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

private void OpenToolStripMenuItem\_Click(object sender, EventArgs e)

{

OpenFileDialog ofd = new OpenFileDialog {

Filter = "Файли зображень (\*.bmp,\*.jpg,\*.jpeg,\*.gif,\*.png)|\*.bmp;\*.jpg;\*.jpeg;\*.gif;\*.png|Усі файли (\*.\*) | \*.\* "

};

if (ofd.ShowDialog() == DialogResult.OK) {

sourceImagePB.Load(ofd.FileName);

Bitmap sourceBitmap = (Bitmap)sourceImagePB.Image;

BasePicturePropertys.Text = "H:" + sourceBitmap.Height + " W:" + sourceBitmap.Width + "\r\n"

+ "Total pixels: " + sourceBitmap.Height \* sourceBitmap.Width;

Application.DoEvents();

int numberOfColors = 256;

FPTNPicture.Image = QBitmap.LinearQuantization((Bitmap)sourceImagePB.Image, numberOfColors);

FPTNPictureESTLabel.Text = "EST=" + QBitmap.BitmapEST((Bitmap)sourceImagePB.Image, (Bitmap)FPTNPicture.Image).ToString();

Application.DoEvents();

FPTYPicture.Image = QBitmap.Tonning((Bitmap)sourceImagePB.Image, (Bitmap)FPTNPicture.Image);

FPTYPictureESTLabel.Text = "EST=" + QBitmap.BitmapEST((Bitmap)sourceImagePB.Image, (Bitmap)FPTNPicture.Image).ToString();

Application.DoEvents();

OPTNPicture.Image = QBitmap.PopularQuantization((Bitmap)sourceImagePB.Image, numberOfColors);

OPTNPictureESTLabel.Text = "EST=" + QBitmap.BitmapEST((Bitmap)sourceImagePB.Image, (Bitmap)OPTNPicture.Image).ToString();

Application.DoEvents();

OPTYPicture.Image = QBitmap.Tonning((Bitmap)sourceImagePB.Image, (Bitmap)OPTNPicture.Image);

OPTYPictureESTLabel.Text = "EST=" + QBitmap.BitmapEST((Bitmap)sourceImagePB.Image, (Bitmap)OPTYPicture.Image).ToString();

Application.DoEvents();

}

}

private void SaveOnPictureBox\_Click(object sender, EventArgs e)

{

PictureBox thisPictureBox = (PictureBox)sender;

if (thisPictureBox.Image != null)

{

SaveFileDialog sfd = new SaveFileDialog

{

Filter = "Файли зображень (\*.bmp,\*.jpg,\*.jpeg,\*.gif,\*.png)|\*.bmp;\*.jpg;\*.jpeg;\*.gif;\*.png|Усі файли (\*.\*) | \*.\* "

};

if (sfd.ShowDialog() == DialogResult.OK)

{

((PictureBox)sender).Image.Save(sfd.FileName);

}

}

}

}

}