Ministerul Educaţiei din Republica Moldova

Universitatea Liberă Internaţională din Moldova

Facultatea Informatică şi Inginerie

Catedra Tehnologii Informaţionale şi Inginerie

**RAPORT**

la lucrarea de laborator № 7

Disciplina: Prelucrarea semnalelor

*"*  *2D signal’s edge detection "*

**A efectuat**

**studentul gr.IA- 33 Semnătura Cușnariov Ruslan**

**A verificat**

**Dr.hab., prof.univ Semnătura Perju Veaceslav**

**Chişinău 2015**

2. **Describe the algorithm and the software.**

using System;

using System.IO;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

namespace Vaneok

{

public partial class FormPrincipala : Form

{

Bitmap bitmap,bitmap2;

int step;

public FormPrincipala()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "") MessageBox.Show("Introduceti adresa imaginii...", "Camp gol");

else if (textBox2.Text == "") MessageBox.Show("Introduceti pasul.", "Camp gol");

else

{

FileStream fStream;

try

{

fStream = File.Open(textBox1.Text, FileMode.Open, FileAccess.Read);

}

catch

{

MessageBox.Show("Calea incorecta!", "Eroare...");

bitmap.Dispose();

return;

}

if (fStream != null)

{

try

{

bitmap = new Bitmap(fStream);

bitmap2 = new Bitmap(fStream);

}

catch

{

MessageBox.Show("Aceast fisier nu reprezinta o Imagine");

bitmap.Dispose();

fStream.Close();

fStream.Dispose();

}

try { step = Convert.ToInt32(textBox2.Text) \* 10000; }

catch { MessageBox.Show("Nu ati introdus pasul \'PAS\'"); return; }

EdgeProcess(radioButton1.Checked);

// Bitmap bitmap2 = new Bitmap(fstream);

Form PictureForm = new Form();

PictureForm.Size = new Size(bitmap.Width\*2 + 20, bitmap.Height + 40);

PictureForm.Name = "Imaginea prelucrata";

PictureBox pbox = new PictureBox();

pbox.Size = new Size(bitmap.Width, bitmap.Height);

pbox.Image = bitmap;

PictureBox pbox2 = new PictureBox();

pbox2.Size=new Size(bitmap.Width+20, bitmap.Height+20);

pbox2.Image = bitmap2;

pbox2.Location = new Point(bitmap.Width + 10, 0);

PictureForm.Controls.Add(pbox2);

PictureForm.Controls.Add(pbox);

PictureForm.ShowDialog();

bitmap.Dispose();

PictureForm.Dispose();

fStream.Close();

fStream.Dispose();

}

}

}

// Prelucrarea imaginii

private void EdgeProcess(bool roberts)

{

//Merge verificarea

// Algoritmul lui Roberts

if (roberts == true)

{

Bitmap tempBMP = new Bitmap(bitmap);

int imgLatime = tempBMP.Width;

int imgInaltime = tempBMP.Height;

for (int h = 0; h < imgInaltime - 1; h++)

for (int w = 0; w < imgLatime - 1; w++)

{

if (Math.Abs(tempBMP.GetPixel(w, h).ToArgb() + tempBMP.GetPixel(w, h + 1).ToArgb()) -

Math.Abs(tempBMP.GetPixel(w + 1, h).ToArgb() + tempBMP.GetPixel(w + 1, h + 1).ToArgb()) >= step)

tempBMP.SetPixel(w, h, Color.Black);

else tempBMP.SetPixel(w, h, Color.White);

}

// Distrugerea obiectelor

bitmap.Dispose();

bitmap = tempBMP;

}

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

}

private void FormPrincipala\_Load(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

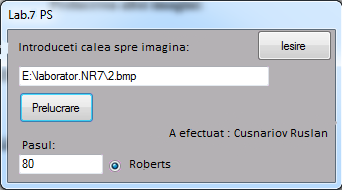
Application.Exit();

}

}

}

***Rezultatele executării:***

******

***Fig.2- Prelucrarea imaginii cu pasul 10***



***Fig.3 - Prelucrarea imaginii cu pasul 50***



***Fig.4 - Prelucrarea imaginii cu pasul 80***



**Prelucrrea altei imagini:**

***Fig.5 - Prelucrarea imaginii cu pasul 10***



***Fig. 6 – Prelucrarea imaginii cu pasul 50***

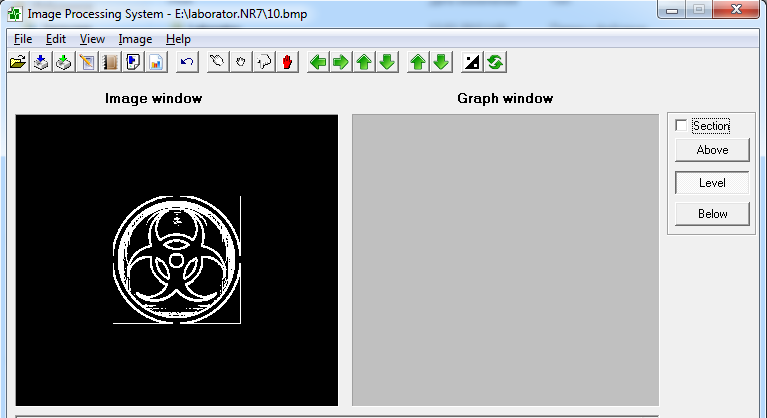


***Fig. 7 – Prelucrarea imaginii cu pasul 80***

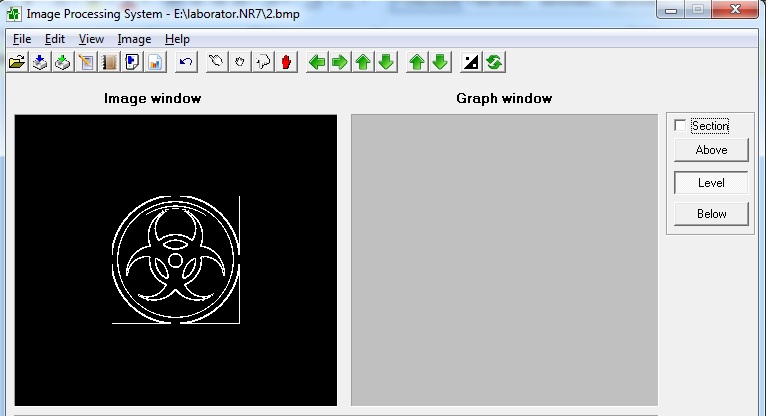


**Comparam rezultatele obtinute cu IPS**

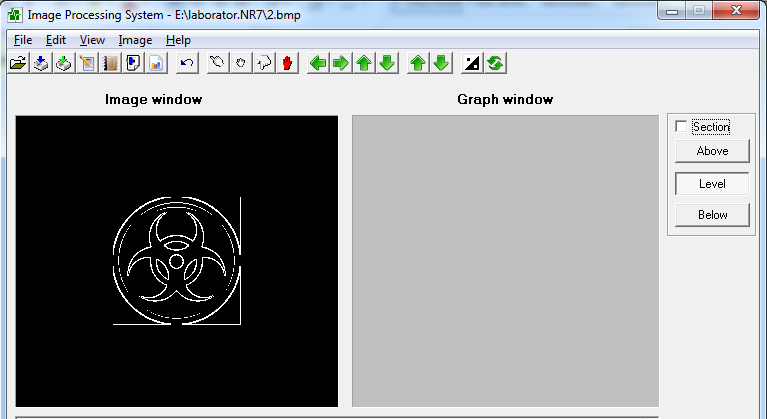
***Fig.8 - Prelucrarea imaginii cu pasul 10***

****

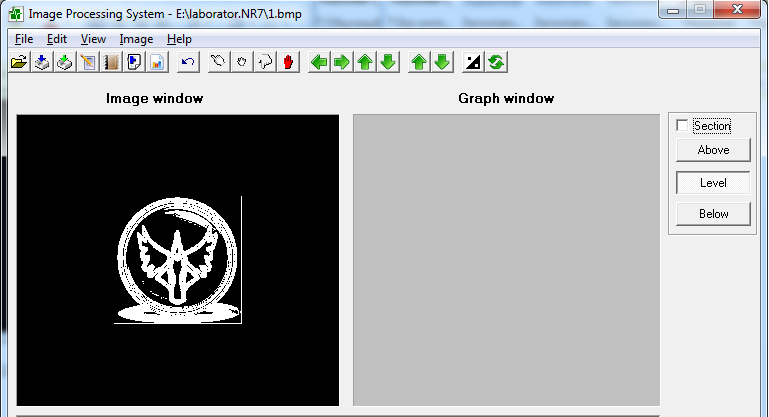
***Fig. 9 - Prelucrarea imaginii cu pasul 50***

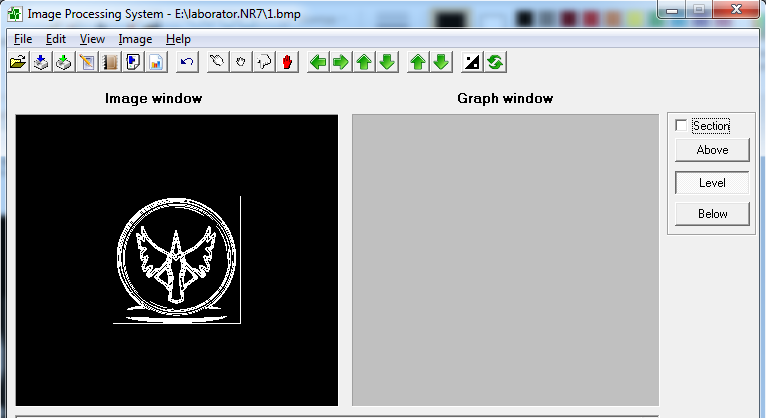


***Fig.10 - Prelucrarea imaginii cu pasul 80***

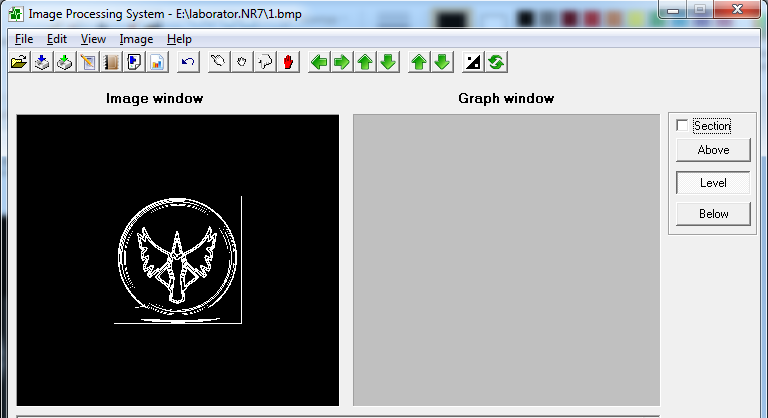


***Fig.11 - Prelucrarea imaginii cu pasul 10***

******

***Fig.12 - Prelucrarea imag***

***Fig.13 - Prelucrarea imaginii cu pasul 80***



**Concluzii:**