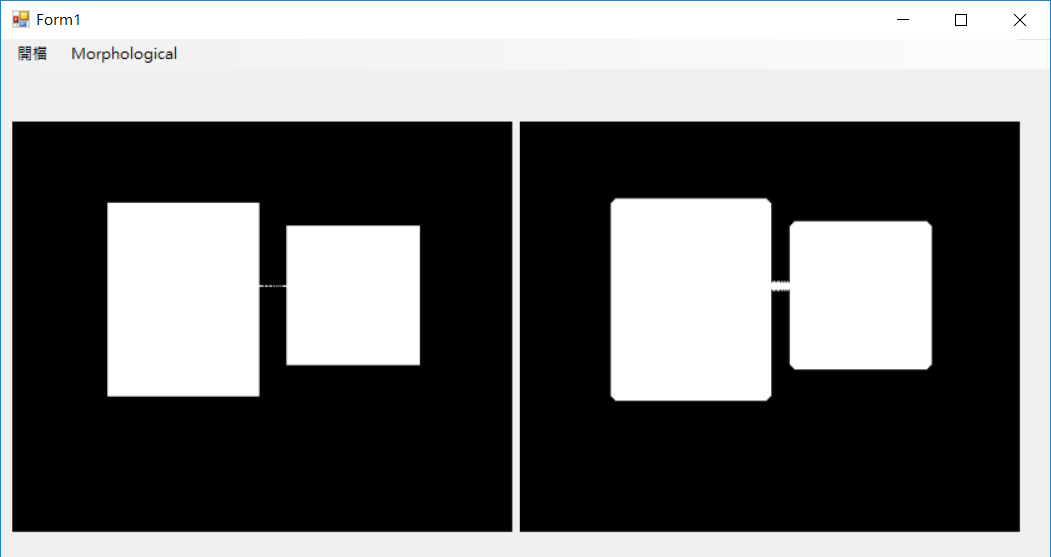
姓名：林佑恩 班級：四子三丙  
學號：1103105336老師：陳昭和老師

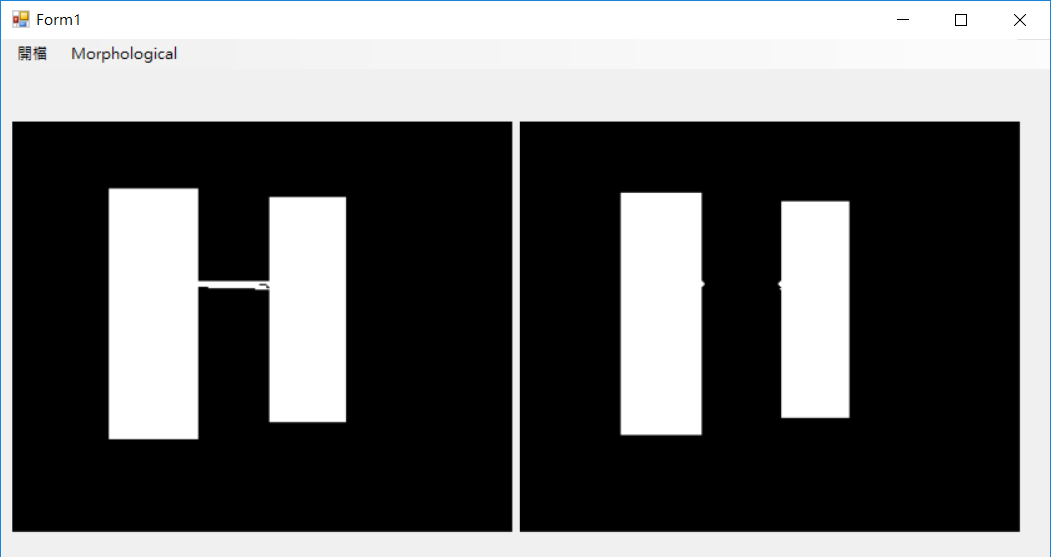
Morphological operations, make a program that realizes Dilation, Erosion, Opening, and Closing, and show the results of using test images. (Please use C++ based programming language, can’t use Matlab and LabVIEW )

結果：

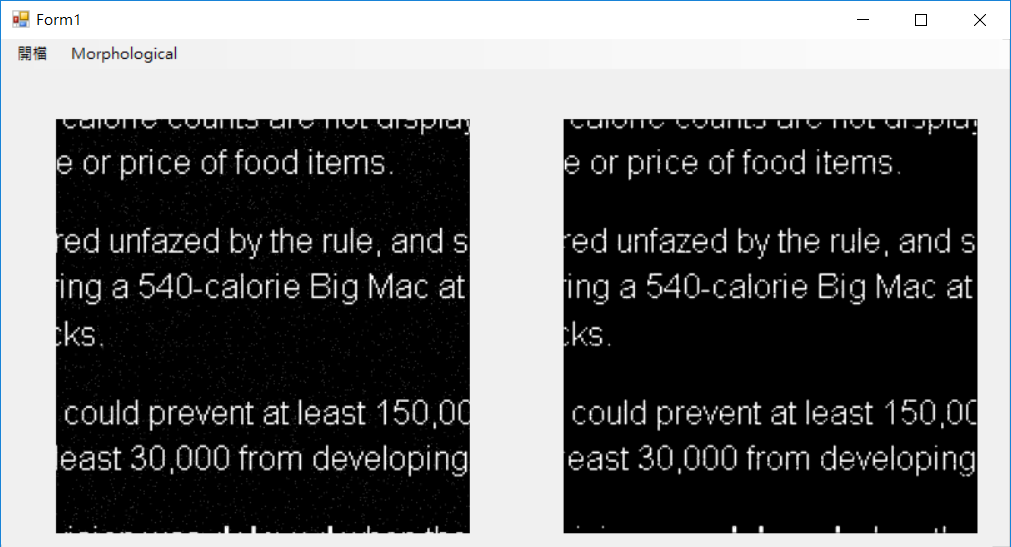
Dilation：膨脹x3



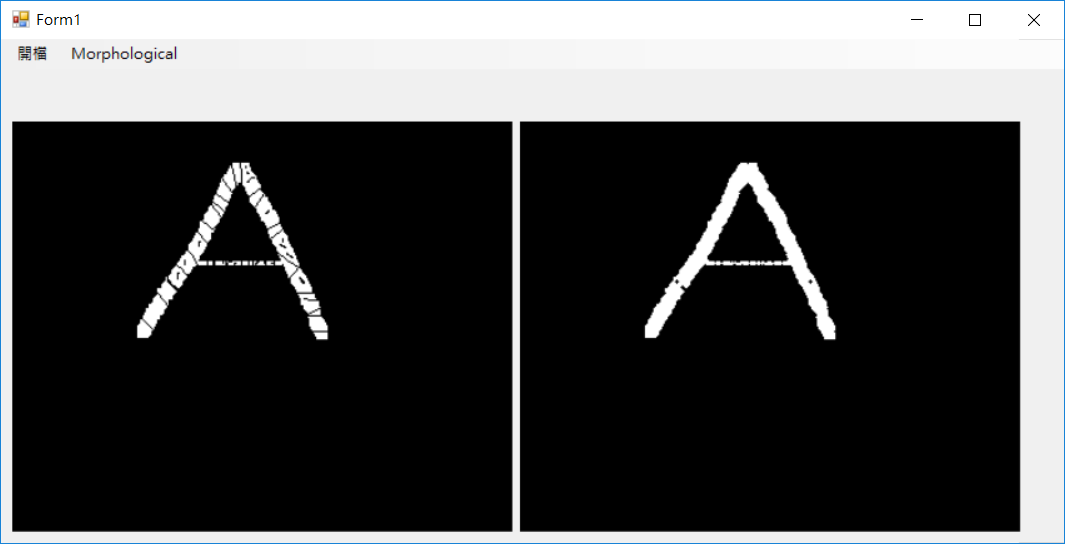
Erosion：侵蝕x3



Opening(先侵蝕再膨脹，斷開就是(AΘB)⊕B)：



Closing(先膨脹再侵蝕，閉合就是(A⊕B)ΘB)：



主要程式碼：

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;

using System.Drawing.Imaging;

namespace \_20170531\_work05\_MorphologicalImageProcessing

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

Bitmap ori\_image;

BitmapData ori\_data;

Rectangle imageRect;

byte[] ori\_Values;

int ori\_bytes;

bool check\_input = false;

private byte[,] shape

{

get

{

return new byte[,]

{

{ 0, 1, 0 },

{ 1, 1, 1 },

{ 0, 1, 0 }

};

}

}

private void 開檔ToolStripMenuItem\_Click(object sender, EventArgs e)

{

OpenFileDialog ofd = new OpenFileDialog();

ofd.Filter = "Image Files(\*.BMP;\*.JPG;\*.GIF;\*.PNG)|\*.BMP;\*.JPG;\*.GIF;\*.PNG";

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

{

pictureBox1.Image = Image.FromFile(ofd.FileName);

}

ori\_image = pictureBox1.Image as Bitmap;

if (ori\_image != null)

{

imageRect = new Rectangle(0, 0, ori\_image.Width, ori\_image.Height);

ori\_data = ori\_image.LockBits(imageRect, ImageLockMode.ReadWrite, ori\_image.PixelFormat);

ori\_bytes = ori\_data.Stride \* ori\_image.Height;

ori\_Values = new byte[ori\_bytes];

IntPtr ori\_Ptr = ori\_data.Scan0;

System.Runtime.InteropServices.Marshal.Copy(ori\_Ptr, ori\_Values, 0, ori\_bytes);//複製RGB信息到byte數組

GrayLevel(ori\_image.Height, ori\_image.Width, ori\_data.Stride, ori\_Values);

System.Runtime.InteropServices.Marshal.Copy(ori\_Values, 0, ori\_Ptr, ori\_bytes); //複製byte陣列到RGB

ori\_image.UnlockBits(ori\_data);

check\_input = true;

pictureBox1.Image = ori\_image;

}

}

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

{

if (check\_input)

{

ori\_image.LockBits(imageRect, ImageLockMode.ReadWrite, ori\_image.PixelFormat);

var D\_Values = new byte[ori\_bytes];

Array.Copy(ori\_Values, 0, D\_Values, 0, ori\_bytes); //把ori\_Values陣列的數值複製到D\_Values陣列

ori\_Values = Dilation\_Erosion(D\_Values, true, 3);

System.Runtime.InteropServices.Marshal.Copy(ori\_Values, 0, ori\_data.Scan0, ori\_bytes); //複製byte陣列到RGB

ori\_image.UnlockBits(ori\_data);

pictureBox2.Image = ori\_image;

}

else MessageBox.Show("未先選擇圖檔", "ERROR",MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

}

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

{

if (check\_input)

{

ori\_image.LockBits(imageRect, ImageLockMode.ReadWrite, ori\_image.PixelFormat);

var E\_Values = new byte[ori\_bytes];

Array.Copy(ori\_Values, 0, E\_Values, 0, ori\_bytes); //把ori\_Values陣列的數值複製到D\_Values陣列

ori\_Values = Dilation\_Erosion(E\_Values, false, 3);

System.Runtime.InteropServices.Marshal.Copy(ori\_Values, 0, ori\_data.Scan0, ori\_bytes); //複製byte陣列到RGB

ori\_image.UnlockBits(ori\_data);

pictureBox2.Image = ori\_image;

}

else MessageBox.Show("未先選擇圖檔", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

}

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

{ //先侵蝕再膨脹，斷開就是(AΘB)⊕B

if (check\_input)

{

ori\_image.LockBits(imageRect, ImageLockMode.ReadWrite, ori\_image.PixelFormat);

var O\_Values = new byte[ori\_bytes];

Array.Copy(ori\_Values, 0, O\_Values, 0, ori\_bytes); //把ori\_Values陣列的數值複製到D\_Values陣列

O\_Values = Dilation\_Erosion(O\_Values, false, 3); //侵蝕

ori\_Values = Dilation\_Erosion(O\_Values, true, 3); //膨脹

System.Runtime.InteropServices.Marshal.Copy(ori\_Values, 0, ori\_data.Scan0, ori\_bytes); //複製byte陣列到RGB

ori\_image.UnlockBits(ori\_data);

pictureBox2.Image = ori\_image;

}

else MessageBox.Show("未先選擇圖檔", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

}

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

{ //先膨脹再侵蝕，閉合就是(A⊕B)ΘB

if (check\_input)

{

ori\_image.LockBits(imageRect, ImageLockMode.ReadWrite, ori\_image.PixelFormat);

var O\_Values = new byte[ori\_bytes];

Array.Copy(ori\_Values, 0, O\_Values, 0, ori\_bytes); //把ori\_Values陣列的數值複製到D\_Values陣列

O\_Values = Dilation\_Erosion(O\_Values, true, 3); //膨脹

ori\_Values = Dilation\_Erosion(O\_Values, false, 3); //侵蝕

System.Runtime.InteropServices.Marshal.Copy(ori\_Values, 0, ori\_data.Scan0, ori\_bytes); //複製byte陣列到RGB

ori\_image.UnlockBits(ori\_data);

pictureBox2.Image = ori\_image;

}

else MessageBox.Show("未先選擇圖檔", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

}

public void GrayLevel(int h, int w, int stride, byte[] ori\_bmp)

{

for (int i = 0; i < h; i++) //轉成gray level

{

for (int j = 0; j < w; j++)

{

int p\_Index = i \* stride + j \* 3;

byte pixel = Convert.ToByte((ori\_bmp[p\_Index + 0] + ori\_bmp[p\_Index + 1] + ori\_bmp[p\_Index + 2]) / 3);

ori\_bmp[p\_Index + 0] = pixel; //R

ori\_bmp[p\_Index + 1] = pixel; //G

ori\_bmp[p\_Index + 2] = pixel; //B

}

}

}

public byte[] Dilation\_Erosion(byte[] Values, bool Type, int Size)

{

byte[] tmp\_Values = new byte[ori\_bytes];

int Offset = (Size - 1) / 2;

byte value;

for (int i = Offset; i < ori\_image.Height - Offset; i++)

{

for (int j = Offset; j < ori\_image.Width - Offset; j++)

{

if (Type) value = 0; //膨脹

else value = 255; //侵蝕

int index = i \* ori\_data.Stride + j \* 3;

for (int ii = -Offset; ii <= Offset; ii++)

{

for (int jj = -Offset; jj <= Offset; jj++)

{

if (shape[ii + Offset, jj + Offset] == 1)

{

int calcOffset = index + ii \* ori\_data.Stride + jj \* 3;

if (Type) value = Math.Max(value, Values[calcOffset]);

else value = Math.Min(value, Values[calcOffset]);

}

else

continue;

}

}

tmp\_Values[index] = value;

tmp\_Values[index + 1] = value;

tmp\_Values[index + 2] = value;

}

}

return tmp\_Values;

}

}

}