**Image Detail only**

**[Testing Image]**

****

Fig1. Gray level image

I have chosen a picture of lake that I took inside my university campus (shown as Figure 1). I want to use the sharpen method to separate the details and noises. Only the detail parts of the image are left after being processed with Gaussian Blur.

**[Output Image]**

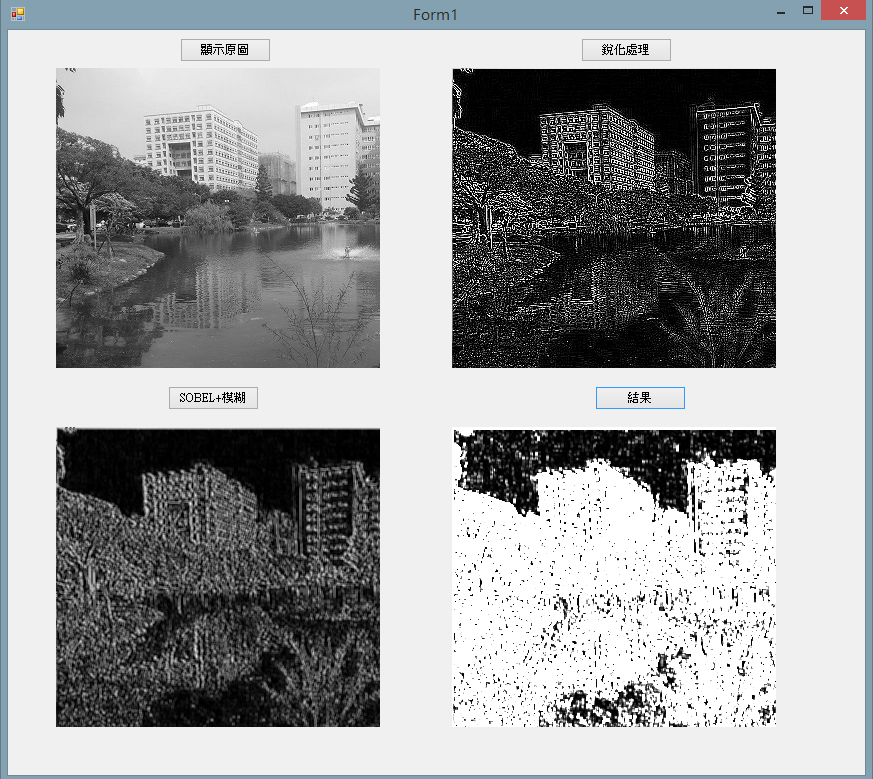


Fig2. Output

Figure 2 is the result. The top-left image is the original one; the top-right image is the image after being sharpen; the bottom-left image has been processed with sobel edge detection and Gaussian Blur; and the bottom-right image is the outcome.

**[Processing and Code]**  
1st step→Image sharpening， is mask

[C# code]

public Image sharp(Image org)

{

Bitmap sharp = (Bitmap)org;

int Height = sharp.Height;

int Width = sharp.Width;

int[, ,] rgbData = getRGBData(org);

for (int y = 1; y < Height-1; y++)

{

for (int x = 1; x < Width-1; x++)

{

int pixel = ((rgbData[x, y, 0] \* 8) - (rgbData[(x - 1), (y - 1), 0] + rgbData[x, (y - 1), 0] + rgbData[(x + 1), (y - 1), 0] + rgbData[(x - 1), y, 0] + rgbData[(x + 1), y, 0] + rgbData[(x - 1), (y + 1), 0] + rgbData[x, (y + 1), 0] + rgbData[(x + 1), (y + 1), 0]));

if (pixel <= 0) { pixel = 0; }

else if (pixel >= 255) { pixel = 255; }

sharp.SetPixel(x, y, Color.FromArgb(pixel,pixel,pixel));

}

}

return sharp;

}

第二步→Edge detection by Sobel，and are masks.

[C# code]

public Image sobel(Image sharp)

{

Bitmap sobel = (Bitmap)sharp;

int Height = sobel.Height;

int Width = sobel.Width;

int[, ,] rgbData = getRGBData(sobel);

for (int y = 1; y < Height - 1; y++)

{

for (int x = 1; x < Width - 1; x++)

{

int pixel = ((rgbData[(x - 1), (y + 1), 0] + (rgbData[x, (y + 1), 0] \* 2) + rgbData[(x + 1), (y + 1), 0]) - (rgbData[(x - 1), (y - 1), 0] + (rgbData[x, (y - 1), 0] \* 2) + rgbData[(x + 1), (y - 1), 0]));

if (pixel <= 0) { pixel = 0; }

else if (pixel >= 255) { pixel = 255; }

sobel.SetPixel(x, y, Color.FromArgb(pixel, pixel, pixel));

}

}

for (int y = 1; y < Height - 1; y++)

{

for (int x = 1; x < Width - 1; x++)

{

int pixel = ((rgbData[(x + 1), (y - 1), 0] + (rgbData[(x + 1), y, 0] \* 2) + rgbData[(x + 1), (y - 1), 0]) - (rgbData[(x - 1), (y - 1), 0] + (rgbData[(x - 1), y, 0] \* 2) + rgbData[(x - 1), (y + 1), 0]));

if (pixel <= 0) { pixel = 0; }

else if (pixel >= 255) { pixel = 255; }

sobel.SetPixel(x, y, Color.FromArgb(pixel, pixel, pixel));

}

}

return sobel;

}

3rd step→Image blurred /9 is mask。

[C# code]

public Image blur(Image sobel)

{

Bitmap blur = (Bitmap)sobel;

int Height = blur.Height;

int Width = blur.Width;

int[, ,] rgbData = getRGBData(blur);

for (int y = 1; y < Height - 1; y++)

{

for (int x = 1; x < Width - 1; x++)

{

int pixel = ((rgbData[(x - 1), (y - 1), 0] + rgbData[x, (y - 1), 0] + rgbData[(x + 1), (y - 1), 0] + rgbData[(x - 1), y, 0] + rgbData[(x + 1), y, 0] + rgbData[x, y, 0] + rgbData[(x - 1), (y + 1), 0] + rgbData[x, (y + 1), 0] + rgbData[(x + 1), (y + 1), 0]) / 9);

if (pixel <= 0) { pixel = 0; }

else if (pixel >= 255) { pixel = 255; }

blur.SetPixel(x, y, Color.FromArgb(pixel, pixel, pixel));

}

}

return blur;

}

4th →Sharp image multiplied by blurred image。

[C# code]

public Image result(Image sharp,Image blur)

{

Bitmap sharpresult = (Bitmap)sharp;

Bitmap blurresult = (Bitmap)blur;

int Height = sharp.Height;

int Width = sharp.Width;

int[, ,] rgbDatasharp = getRGBData(sharpresult);

int[, ,] rgbDatablur = getRGBData(blurresult);

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

{

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

{

int pixel = (rgbDatasharp[x, y, 0] \* rgbDatablur[x, y, 0]);

if (pixel <= 0) { pixel = 0; }

else if (pixel >= 255) { pixel = 255; }

sharpresult.SetPixel(x, y, Color.FromArgb(pixel, pixel, pixel));系

}

}

return sharp;

}