Computer Vision - Homework 10

開發環境

- OS: Windows 10 Pro
- Program Language: C# (with .Net Core 3.1)
- IDE: Visual Studio 2019
- Project: Console Application

程式說明

程式碼主要寫在 Program.cs，各題目程式皆已實作個別方法，由 Main entry 進行呼叫，答案結果儲存於 answers 資料夾。

各題目相關演算法說明如下：

各題目作法相同，步驟及 Code Snippet 參考如下：

1. 取得 Laplacian Array
2. 藉由 Lapliacian Array 取得 ZeroCrossingImage

```
private static int[,] GetLaplacianArray(Bitmap srcImg, int threshold, int[][]
kernel, int weight = 1)
{
    var kWidth = kernel.GetLength(0);
    var kHeight = kernel[0].GetLength(0);

    var result = new int[srcImg.Width, srcImg.Height];

    for (var x = 0; x < srcImg.Width; x++)
    {
        for (var y = 0; y < srcImg.Height; y++)
        {
            #region get xn, yn

            var xn = new int[kWidth];
            var yn = new int[kHeight];

            var distance = -kWidth / 2;

            for (var i = 0; i < xn.Length; i++)
            {
                if (distance <= 0)
                {
                    xn[i] = Math.Max(x + distance, 0);
                }
                else
                {
                    xn[i] = Math.Min(x + distance, srcImg.Width - 1);
                }

                distance += 1;
            }
```

```
            distance = -kHeight / 2;

            for (var i = 0; i < yn.Length; i++)
            {
                if (distance <= 0)
                {
                    yn[i] = Math.Max(y + distance, 0);
                }
                else
                {
                    yn[i] = Math.Min(y + distance, srcImg.Height - 1);
                }

                distance += 1;
            }

            #endregion

            #region get neighbors

            var neighbors = new int[kWidth, kHeight];

            for (var i = 0; i < kWidth; i++)
            {
                for (var j = 0; j < kHeight; j++)
                {
                    neighbors[i, j] = srcImg.GetPixel(xn[i], yn[j]).R;
                }
            }

            #endregion

            var magnitude = 0;

            for (var i = 0; i < kWidth; i++)
            {
                for (var j = 0; j < kHeight; j++)
                {
                    magnitude += kernel[j][i] * neighbors[i, j];
                }
            }

            magnitude /= weight;

            if (magnitude >= threshold)
                result[x, y] = 1;
            else if (magnitude <= -threshold)
                result[x, y] = -1;
            else
                result[x, y] = 0;
        }
    }

    return result;
}

private static Bitmap GetZeroCrossingDetectorImage(int[,] gradient, int width,
int height)
```

```
{
    var result = new Bitmap(gradient.GetLength(0), gradient.GetLength(1));

    for (var x = 0; x < result.Width; x++)
        for (var y = 0; y < result.Height; y++)
        {
            var cross = 1;
            if (gradient[x, y] == 1)
                for (var x1 = -width / 2; x1 <= width / 2 + 1; x1++)
                    for (var y1 = -height / 2; y1 <= height / 2 + 1; y1++)
                    {
                        var destX = x + x1 < 0 ? 0 : x + x1;
                        destX = destX > result.Width - 1 ? result.Width - 1 :
destX;
                        var destY = y + y1 < 0 ? 0 : y + y1;
                        destY = destY > result.Height - 1 ? result.Height - 1 :
destY;

                        if (gradient[destX, destY] == -1)
                            cross = 0;
                    }

            result.SetPixel(x, y, cross == 1 ? Color.White : Color.Black);
        }

    return result;
}
```

(A). Laplace Mask1 (0, 1, 0, 1, -4, 1, 0, 1, 0)

- Threshold: 15

```
GetZeroCrossingDetectorImage(GetLaplacianArray(image, 15, LaplacianMask1Kernel),
3, 3);
```

(B). Laplace Mask2 (1, 1, 1, 1, -8, 1, 1, 1, 1)

- Threshold: 15

```
GetZeroCrossingDetectorImage(GetLaplacianArray(image, 15, LaplacianMask2Kernel,
3), 3, 3);
```

(C). Minimum variance Laplacian

- Threshold: 20

```
GetZeroCrossingDetectorImage(GetLaplacianArray(image, 20,
MinimumVarianceLaplacianKernel, 3), 3, 3)
```

(D). Laplace of Gaussian

- Threshold: 3000

```
GetZeroCrossingDetectorImage(GetLaplacianArray(image, 3000,
LaplacianOfGaussiankernel), 11, 11)
```

(E). Difference of Gaussian

- Threshold: 1

```
GetZeroCrossingDetectorImage(GetLaplacianArray(image, 1,
DifferenceOfGaussianKernel), 11, 11)
```

結果圖片

A. Laplace Mask1, Threshold: 15	B. Laplace Mask2, Threshold: 15	C. Minimum variance Laplacian, Threshold: 20
		
D. Laplace of Gaussian, Threshold: 3000	E. Difference of Gaussian, Threshold: 1