

# 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 Laplacian 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[,] laplacianArray)

```

```

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

    for (var x = 0; x < result.Width; x++)
    {
        for (var y = 0; y < result.Height; y++)
        {
            var cross = true;

            // 若 pixel 為 1
            if (laplacianArray[x, y] == 1)
            {
                var x0 = Math.Max(x - 1, 0);
                var y0 = Math.Max(y - 1, 0);
                var x1 = x;
                var y1 = y;
                var x2 = Math.Min(x + 1, result.Width - 1);
                var y2 = Math.Min(y + 1, result.Height - 1);

                var p1 = laplacianArray[x0, y0];
                var p2 = laplacianArray[x1, y0];
                var p3 = laplacianArray[x2, y0];
                var p4 = laplacianArray[x0, y1];
                var p5 = laplacianArray[x2, y1];
                var p6 = laplacianArray[x0, y2];
                var p7 = laplacianArray[x1, y2];
                var p8 = laplacianArray[x2, y2];

                // 周圍的鄰居其中有一個為 -1
                if (p1 <= -1 || p2 <= -1 || p3 <= -1 || p4 <= -1 || p5 <= -1 ||
                    p6 <= -1 || p7 <= -1)
                {
                    cross = false;
                }

                result.SetPixel(x, y, cross ? 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));

```

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

- Threshold: 15

```

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

```

(C). Minimum variance Laplacian

- Threshold: 20

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

(D). Laplace of Gaussian

- Threshold: 3000




```
GetZeroCrossingDetectorImage(GetLaplacianArray(image, 3000,
LaplacianOfGaussianKernel))
```

(E). Difference of Gaussian

- Threshold: 1

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

## 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	
