

影像處理程式設計

內容

- 影像檔讀取與顯示
- 像素值讀取與設定
- 色彩轉換
- 影像直方圖計算
- 影像品質的改變
- 影像測邊

GDI+

- .NET Framework 繪圖相關類別庫GDI+
 - Graphic Device Interface
 - E.g.

```
Graphic G = this .CreateGraphics();  
G.DrawLine(Pens.Black, 0,0,100,100);
```
- 透過**GDI+**類別庫可以完成直線、曲線、橢圓行、矩形、多邊行、字形、圖形等和繪圖有關的程式。
- 其中在命名空間裡面**System.Drawing.Imaging**支援對圖形影像的處理與呈現

影像檔讀取與顯示

- **Image** 類別是一個影像類別，提供基本讀取、與儲存影像功能
- **Bitmap** 點陣圖類別繼承自**Image** 類別，另外提供對像素處理的功能

- E.g.

```
Image img = Image.FromFile("myPicture.jpg");
```

```
int W = img.Width;
```

```
int H = img.Height;
```

```
e.Graphics.DrawImage(img, 10, 50, img.Width, img.Height);
```

影像檔讀取與顯示

```
public partial class Form1 : Form
{
    Image img; // Image 影像
    public Form1()
    {
        InitializeComponent();
    }
    // 開啟檔案按鈕
    private void button1_Click(object sender, EventArgs e)
    {
        if (openFileDialog1.ShowDialog() == DialogResult.OK) // 開啟影像檔
        {
            String input = openFileDialog1.FileName;
            img = Image.FromFile(input); // 產生一個Image物件
            // 呈現 影像 的寬高資訊
            label1.Text = "(" +img.Width.ToString() + ", " + img.Height.ToString() + ")";
            this.Invalidate(); // 要求重畫
        }
    }
}
```

影像檔儲存

// 儲存檔案按鈕

```
private void button3_Click(object sender, EventArgs e)
{
    string strFilter = "BMP(*.bmp)|*.bmp|" + "GIF(*.gif)|*.gif|" +
    "JPEG(*.jpg)|*.jpg|" + "PNG(*.png)|*.png|" + "所有檔案(*.*)|*.*";
    saveFileDialog1.Filter = strFilter;
    if (saveFileDialog1.ShowDialog() == DialogResult.OK) // 儲存影像檔
    {
        String output = saveFileDialog1.FileName;
        pictureBox2.Image.Save(output,
        System.Drawing.Imaging.ImageFormat.Png); // .Png 格式
    }
}
```


表單重畫事件

```
private void Form1_Paint(object sender,  
    PaintEventArgs e)  
    {  
        if (img != null)  
            e.Graphics.DrawImage(img, 10, 50,  
                img.Width, img.Height);  
    }
```

影像旋轉

- Image img2;//global variable
private void button2_Click(object sender, EventArgs e)
{
img2=(Image) img.Clone(); //copy到img2
img2.RotateFlip(RotateFlipType.Rotate90FlipNone);
// 影像旋轉90度
 this.Invalidate(); // 要求重畫
}
- img.RotateFlip(RotateFlipType.Rotate180FlipNone);
// 影像旋轉180度
- img.RotateFlip(RotateFlipType.Rotate270FlipNone);
// 影像旋轉270度

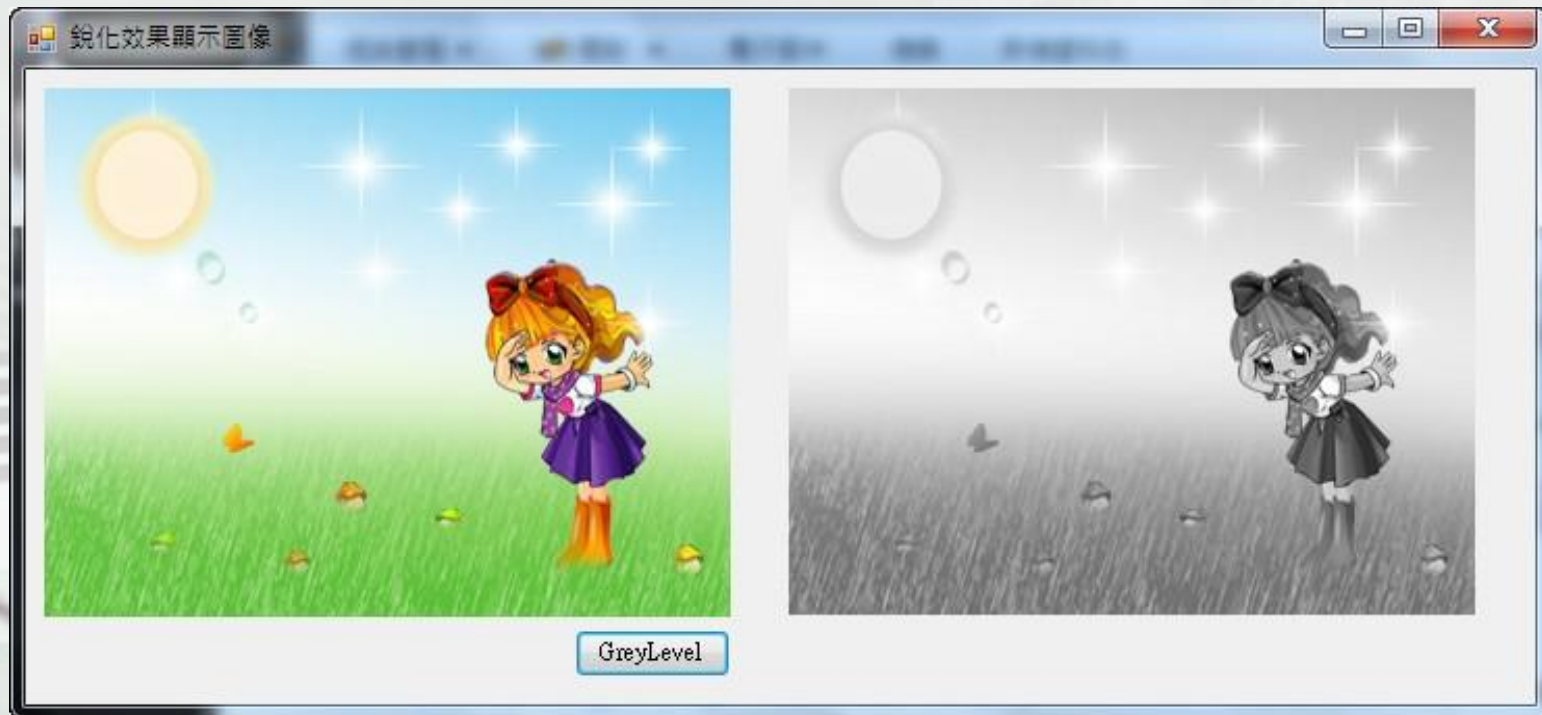
像素值讀取與設定

- `Bitmap img;`
- 取得影像位置(x,y)的像素值
 - `Color pixelColor = img.GetPixel(x, y);`
- 將影像位置(x,y)的像素值 設為newColor
 - `img.SetPixel(x, y, newColor);`
- Color 類別操作

```
int newR = pixelColor.R;//獲取R色值
int newG = pixelColor.G;//獲取G色值
int newB = pixelColor.B;//獲取B色值
Color newColor = Color.FromArgb(newR, newG, newB);
```

彩色轉灰階程式範例

```
private void button1_Click(object sender, EventArgs e)
{
    Bitmap Var_bmp = (Bitmap) pictureBox1.Image.Clone();
    pictureBox2.Image = BlackandWhiteEffect(Var_bmp);
}
```



彩色轉灰階程式範例

```
public Image BlackandWhiteEffect(Bitmap Var_bmp)
{
    int Var_H = Var_bmp.Height;//獲取圖像的高度
    int Var_W = Var_bmp.Width;//獲取圖像的寬度
    Bitmap Var_SaveBmp = (Bitmap) Var_bmp.Clone(); //copy輸入之bimap

    for (int i = 0; i < Var_W; i++) //遍歷圖像的象素
        for (int j = 0; j < Var_H; j++)
        {
            Color tem_color = Var_SaveBmp.GetPixel(i, j);//獲取當前象素的顏色值
            int tem_r, tem_g, tem_b, tem_Value = 0;//定義變量
            tem_r = tem_color.R;//獲取R色值
            tem_g = tem_color.G;//獲取G色值
            tem_b = tem_color.B;//獲取B色值
            tem_Value = ((tem_r + tem_g + tem_b) / 3);//用平均值法產生黑白圖像
            Var_bmp.SetPixel(i, j, Color.FromArgb(tem_Value, tem_Value, tem_Value));//改變當前象素的顏色
        }
    return Var_bmp;
}
```

彩色轉灰階

- **YIQ** color space
- **YIQ** is the color space used by the NTSC color TV system
- NTSC
 - National Television System Committee
 - is the analog television system used in most of North America, most countries in South America, Burma, South Korea, Taiwan, Japan, Philippines

RGB轉YIQ

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.595716 & -0.274453 & -0.321263 \\ 0.211456 & -0.522591 & 0.311135 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$Y = 0.299R + 0.587G + 0.114 * B$$

反白

$$T(f) = 255 - f$$



直方圖

- 像素值分布圖

8	10	10	8	10
10	11	12	9	12
9	10	10	8	10
10	12	9	9	10
9	9	10	12	9

圖4-33 5X5灰階像素影像

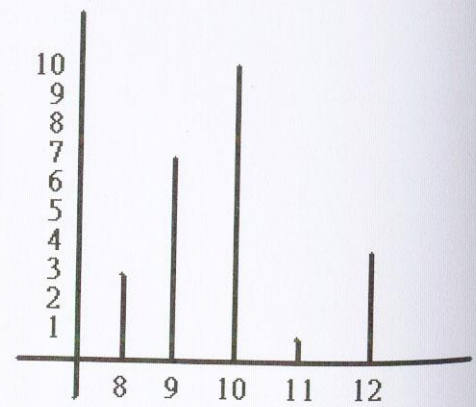


圖4-34 直方圖表示

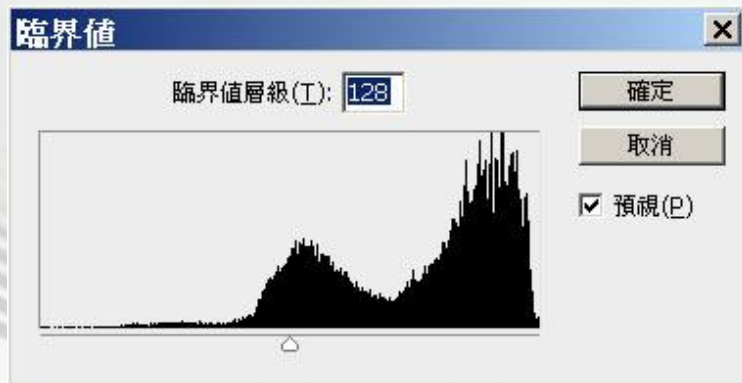
```
int freq[256]={0};  
for(x=0;x<w;x++)  
    for(y=0;y<h;y++)  
        freq[f(x,y)]++;
```

二值化

- 將影像轉為黑白兩色

$$T(f) = \begin{cases} 0, & f < T_0 \\ 255, & f \geq T_0 \end{cases}$$

T_0 為落在0~255的一門檻值



影像品質的改變

- 平滑法 去雜訊
 - 取鄰近9個像素平均值

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

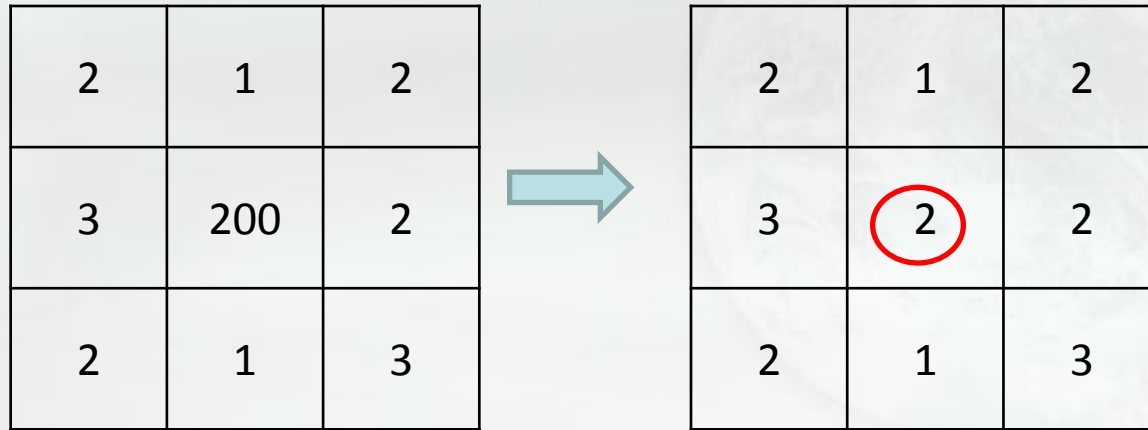
- 9個像素的相對位置

x-1,y-1	x-1, y	x-1, y+1
x,y-1	x,y	x, y+1
x+1, y-1	x+1, y	x+1,y+1

$$g(x, y) = \frac{1}{9} (f(x-1, y-1) + f(x-1, y) + f(x-1, y+1) + f(x, y-1) + f(x, y) + f(x, y+1) + f(x+1, y-1) + f(x+1, y) + f(x+1, y+1))$$

影像品質的改變


- 中值法 去雜訊
 - 取鄰近8個像素值的中間值



影像品質的改變

- 最大值濾波器 把較白的像素突顯出來
 - 取鄰近8個像素值的最大值

190	120	190
200	150	190
220	200	185




190	120	190
200	220	190
220	200	185

影像品質的改變

- 最小值濾波器 把較黑的像素突顯出來
 - 取鄰近8個像素值的最小值

90	90	80
100	180	190
220	200	185



90	90	80
100	80	190
220	200	185

影像品質的改變

- 銳化影像
 - 保留影像低頻部分，強化高頻邊緣。
 - 使影像看起來更清晰
- 對比遮罩

-1	-1	-1
-1	9	-1
-1	-1	-1

0	-a	0
-a	4a+1	-a
0	-a	0

a為介於0~1之間的數字

影像測邊

- Sobel測邊算子

1	2	1
0	0	0
-1	-2	-1

測水平方向的灰階變化

$$v(x, y) = \text{abs}(f(x-1, y-1) + (-1)f(x-1, y+1) + 2f(x, y-1) + (-2)f(x, y+1) + f(x+1, y-1) + (-1)f(x+1, y+1))$$

-1	0	1
-2	0	2
-1	0	1

測垂直方向的灰階變化

$$h(x, y) = \text{abs}((-1)f(x-1, y-1) + (-2)f(x-1, y) + (-1)f(x-1, y+1) + f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1))$$

Let $g(x,y)=h(x,y)+v(x,y)$