

影像處理程式設計2

2012/10/24

GDI 畫圓與線

```
namespace WindowsApplication1
{
    public partial class Form1 : Form
    {
        List<Point> points = new List<Point>(); // 動態陣列 (點的座標)
        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_MouseDown(object sender, MouseEventArgs e)
        {
            Graphics G = this.CreateGraphics();

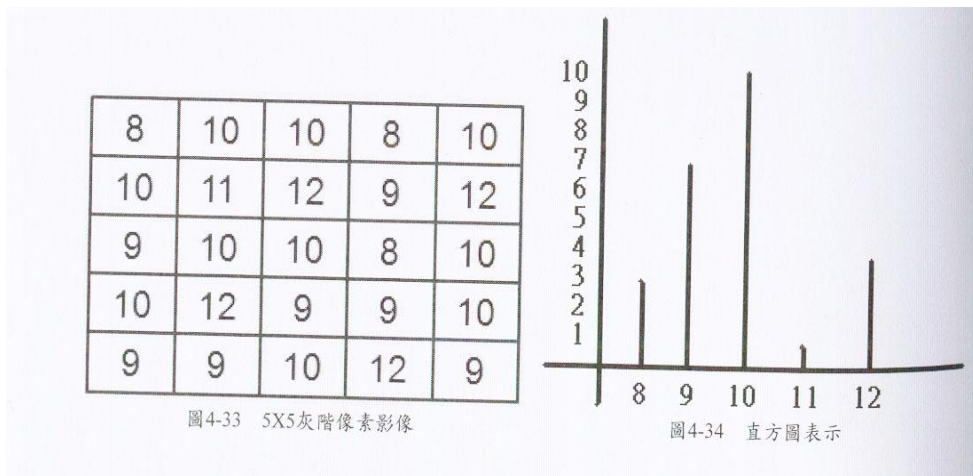
            if (e.Button == MouseButton.Left) // 滑鼠左鍵
            {
                if (points.Count == 0) // 第一個點 畫出 小橢圓形
                    G.DrawEllipse(Pens.Black, e.Location.X - 3, e.Location.Y - 3, 6, 6);
                else // 第二個點 以後 就和上一個點 相連
                    G.DrawLine(Pens.Black, points[points.Count - 1], e.Location);

                points.Add(e.Location); // 把點的資料 加到 動態陣列中
                label1.Text = "x= " + e.Location.X.ToString() + " y=" + e.Location.Y.ToString();
            }
        }
    }
}
```

```
private void button1_Click(object sender, EventArgs e)
{
    points.Clear(); // 清空 動態陣列
    this.Invalidate(); // 要求重畫
}
// 重畫事件函數
private void Form1_Paint(object sender, PaintEventArgs e)
{
    if (points.Count == 0) return; // 沒有點 就不用畫了
    // 第一個點 畫出 小橢圓形
    e.Graphics.DrawEllipse(Pens.Black, points[0].X - 3, points[0].Y - 3, 6, 6);
    if (points.Count > 1) // 如果至少有兩個點
    {
        Point[] PA = new Point[points.Count]; // 建構一個點Point的陣列
        for (int i = 0; i < points.Count; i++)
            PA[i] = points[i];
        e.Graphics.DrawLines(Pens.Black, PA); // 畫出一連串的線條
    }
}
```

直方圖

- 像素值分布圖



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

畫直方圖

```
public Image BlackandWhiteEffect(PictureBox Pict)
{
    int Var_H = Pict.Image.Height;//獲取圖像的高度
    int Var_W = Pict.Image.Width;//獲取圖像的寬度
    Bitmap Var_bmp = new Bitmap(Var_W, Var_H);//根據圖像的大小實例化Bitmap類
    Bitmap Var_SaveBmp = (Bitmap)Pict.Image;//根據圖像實例化Bitmap類
    //把所有曲線點歸零
        for (int a = 0; a < 256; a++)
            mypoint[a] = 0;
    //遍歷圖像的象素
    for (int i = 0; i < Var_W; i++)
        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
                    mypoint[tem_Value]++;
                Var_bmp.SetPixel(i, j, Color.FromArgb(tem_Value, tem_Value));//改變當前象素的顏色值
            }
    return Var_bmp;
}
```

直方圖結果

▶ 0	65874	▶ 16	2059
▶ 1	8795	▶ 17	2052
▶ 2	4332	▶ 18	2206
▶ 3	3038	▶ 19	2365
▶ 4	2520	▶ 20	2202
▶ 5	2282	▶ 21	2164
▶ 6	2204	▶ 22	2227
▶ 7	2232	▶ 23	2581
▶ 8	2049	▶ 24	3061
▶ 9	1983	▶ 25	2785
▶ 10	1859	▶ 26	2672
▶ 11	1923	▶ 27	2608
▶ 12	1837	▶ 28	2339
▶ 13	1913	▶ 29	2068
▶ 14	1898	▶ 30	1886
▶ 15	2112	▶ 31	1728
		▶ :	
		▶ :	
		▶ :	
		▶ :	
		▶ 255	



```
private void button4_Click(object sender, EventArgs e)
{
    pictureBox3.Refresh();
    Graphics myGraphic = pictureBox3.CreateGraphics();
    int max = 0;

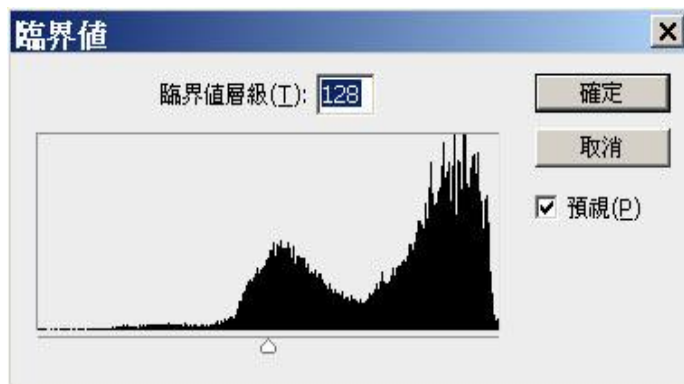
    for (int i = 0; i < 256; i++)
        if (mypoint[i] > max) max = mypoint[i];
    label5.Text = max.ToString();
    float y;
    for (int b = 0; b < 256; b++){
        y= ((float) mypoint[b])/max;
        y = 450 * y;
        myGraphic.DrawLine(new Pen(Color.Blue), b , 450 -(int)y, b, 450);
    }
}
```

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



二值化

```
public Image TwoValued(PictureBox Pict)
{
    int Var_H = Pict.Image.Height; //獲取圖像的高度
    int Var_W = Pict.Image.Width; //獲取圖像的寬度
    Bitmap Var_bmp = new Bitmap(Var_W, Var_H); //根據圖像的大小實例化Bitmap類
    Bitmap Var_SaveBmp = (Bitmap)Pict.Image; //根據圖像實例化Bitmap類
    int threshold = trackBar1.Value;

    //遍歷圖像的象素
    for (int i = 0; i < Var_W; i++)
        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
                if (tem_Value > threshold) tem_Value = 255;
                else tem_Value = 0;
                Var_bmp.SetPixel(i, j, Color.FromArgb(tem_Value, tem_Value, tem_Value)); //改變當前象素的顏色值
            }
    return Var_bmp;
}
```