

暑期課程 基本影像處理 day 1

指導教授：顏淑惠、林慧珍

<http://163.13.127.10>

<http://pria.cs.tku.edu.tw>

指導教授：涂靜琰

<http://mail.tku.edu.tw/cttu>

2016.06

Summer Course Outline

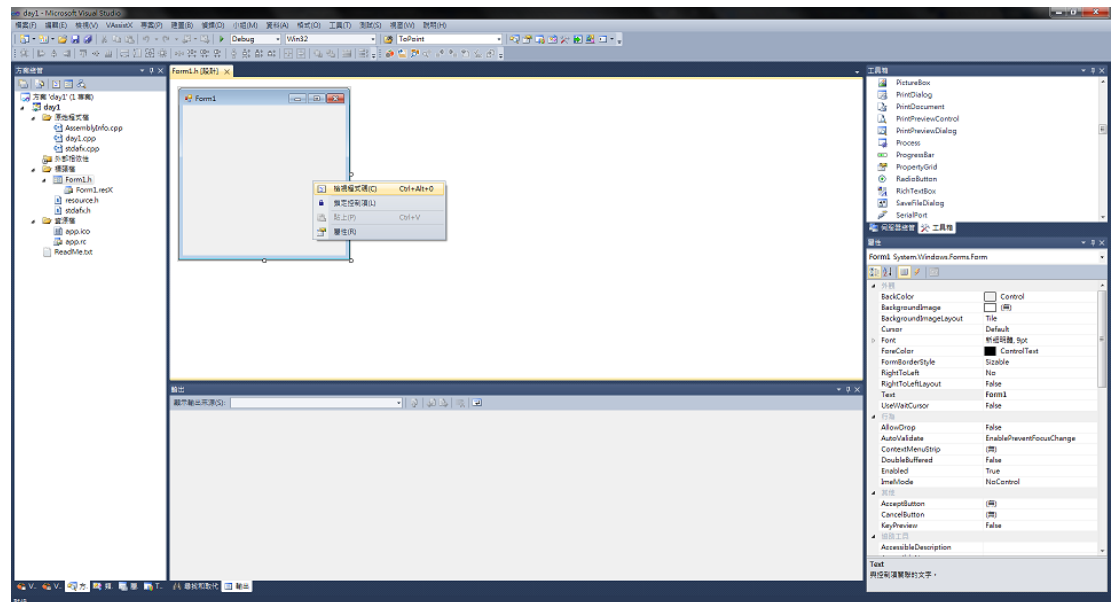
- 開發環境
 - VS2010 Windows Form + OpenCV
- 影像處理
 - Grayscale, Color Image (灰階與彩色影像)
 - Filter (濾波器)
 - Color Space (色彩空間)
 - Histogram (直方圖)
 - Image Morphology (型態影像學)
 - Others

Course Outline – Day1

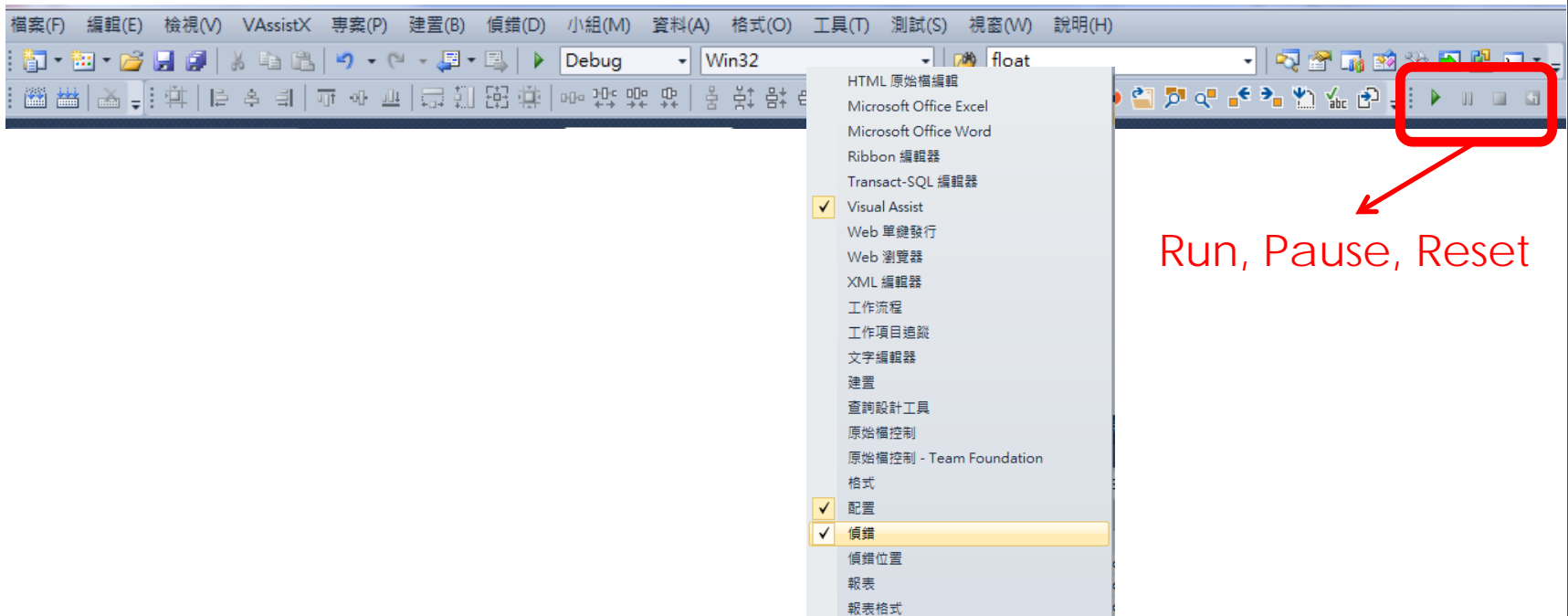
- 開發環境
 - VS2010 Windows Form + OpenCV
- 影像處理
 - Grayscale, Color Image (灰階與彩色影像)
 - Filter (濾波器)
 - Color Space (色彩空間)
 - Histogram (直方圖)
 - Image Morphology (型態影像學)
 - Others

Windows form環境主要可以分為四個部分

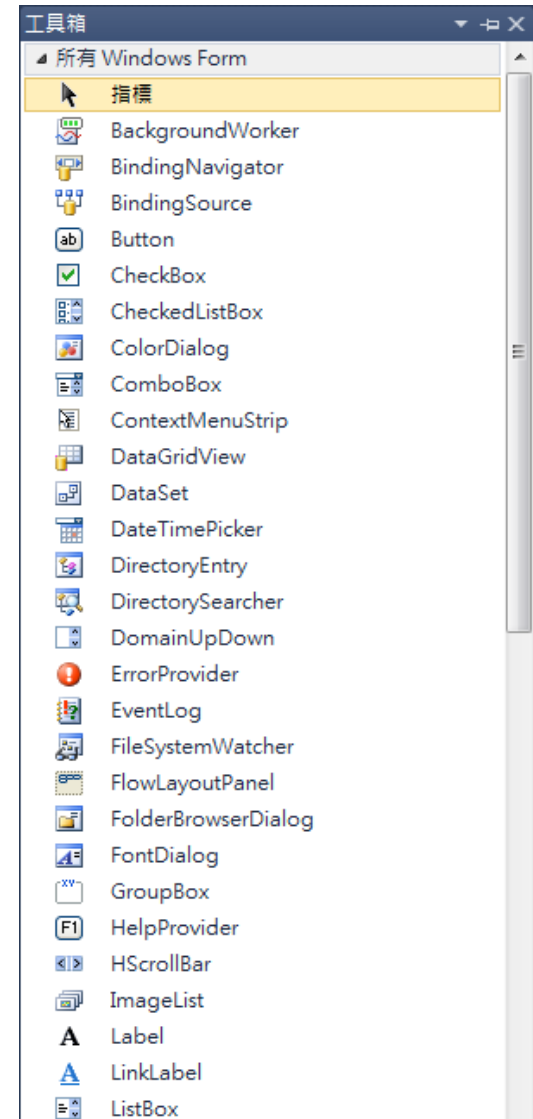
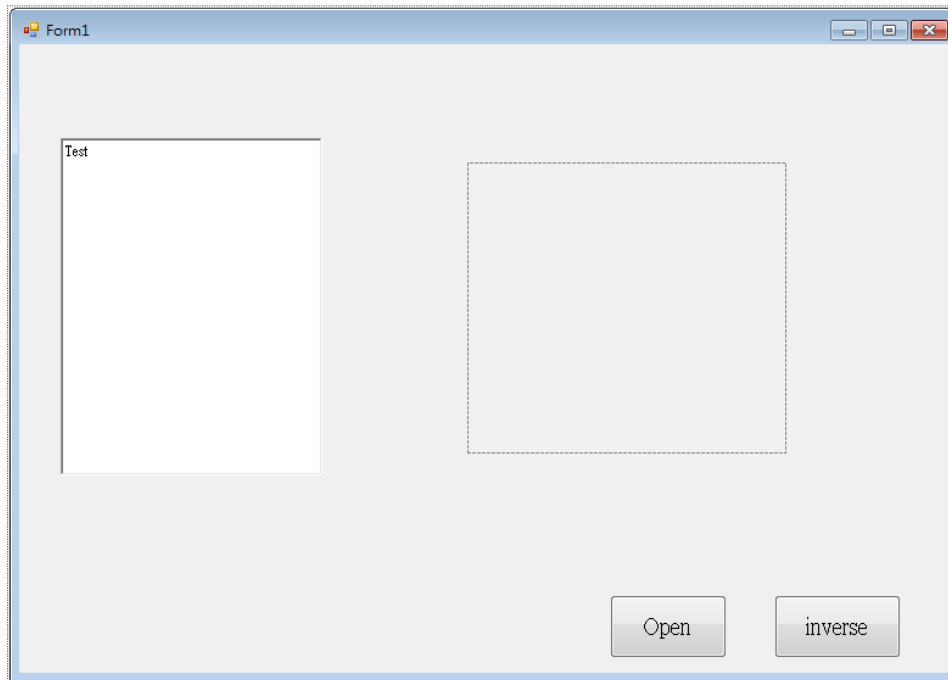
- 主選單&工具列
- 視窗設計&工具箱
- 屬性檢視器
- 程式編輯器
- 方案總管



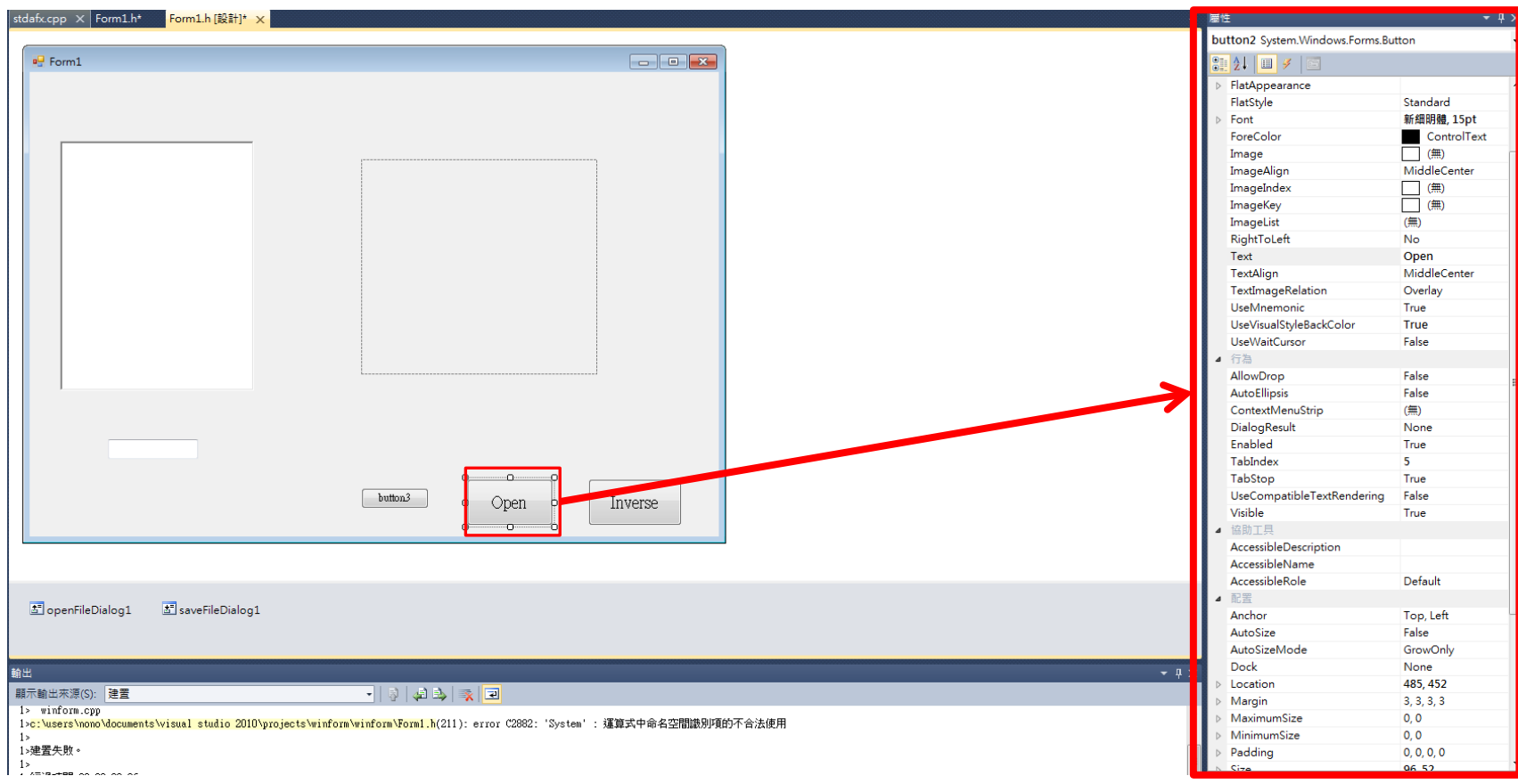
主選單&工具列



視窗設計&工具箱



屬性檢視器



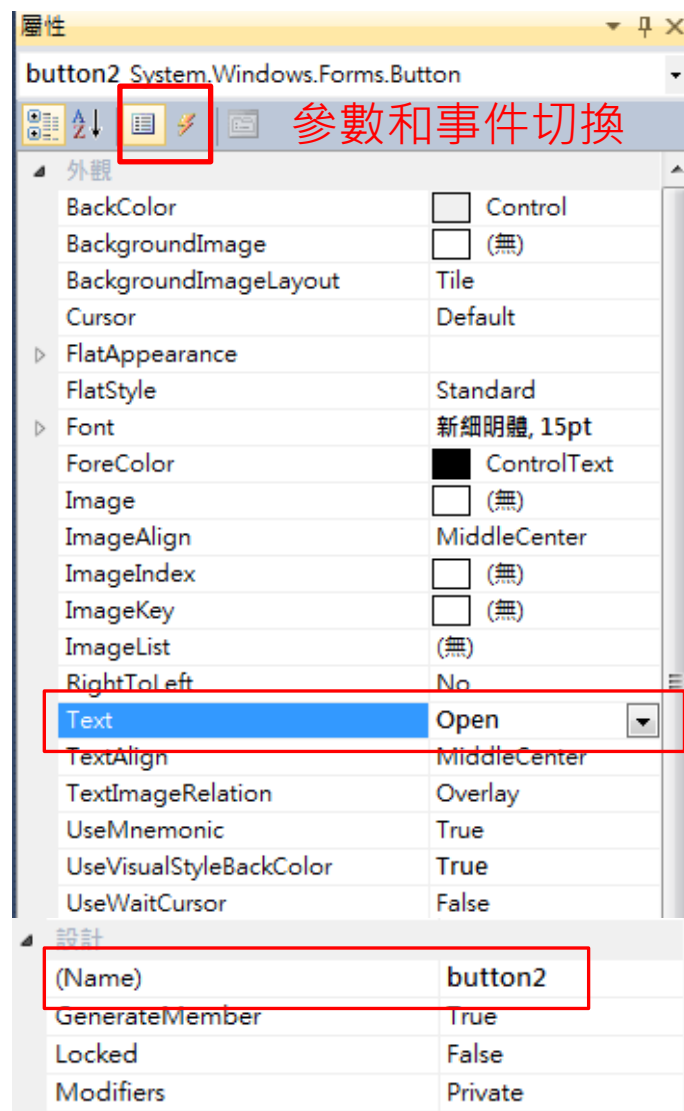


屬性檢視器

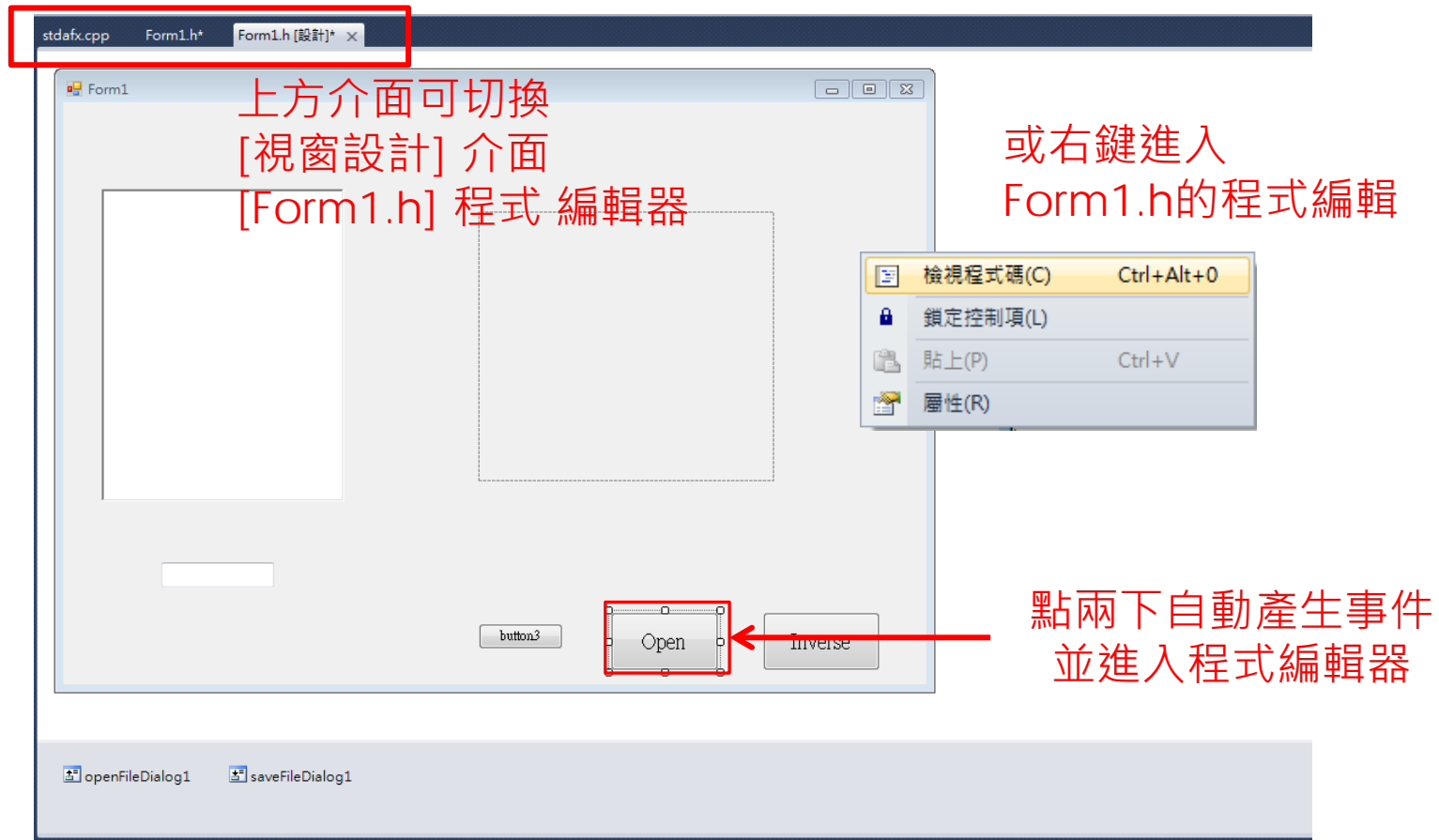
● 參數設定

Text: 物件表面顯示的文字

(Name): 物件的程式名稱(非必要勿更改)




程式編輯器



程式編輯器

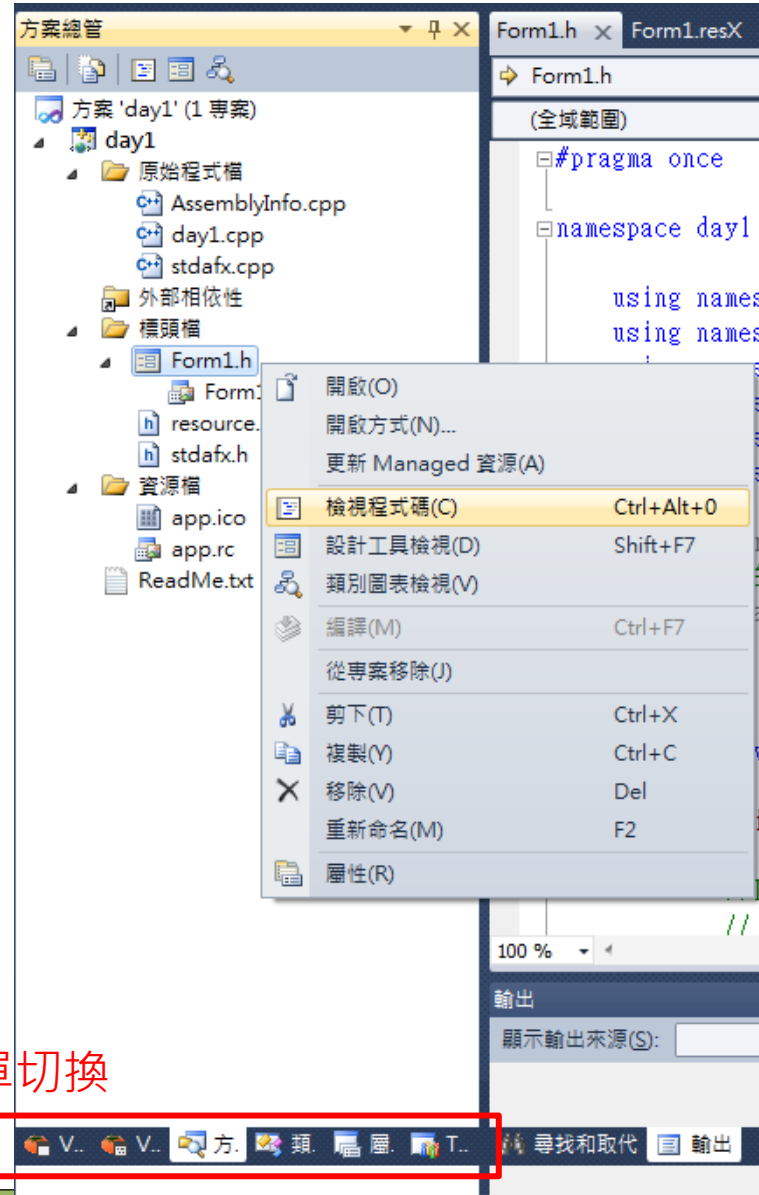
● Form1.h

若點兩下元件則自動建立與此相對應的event function
[Name]_Click



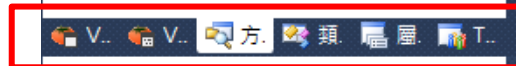
```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {  
    MessageBox::Show("hello, world", "title",  
        MessageBoxButtons::OK, MessageBoxIcon::Asterisk);  
}  
  
};
```

方案總管



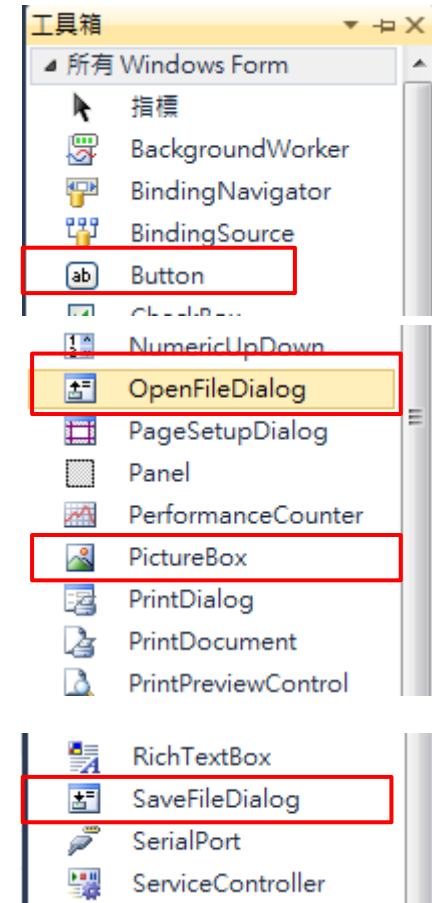
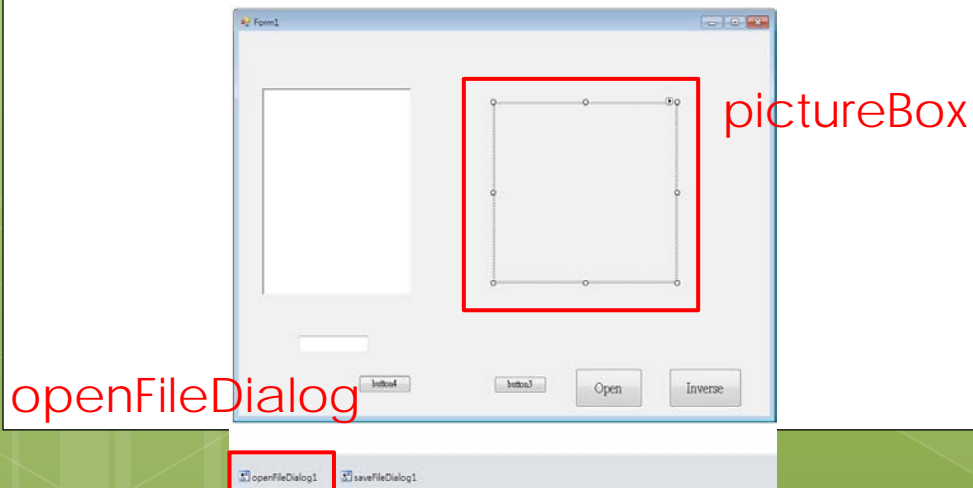
也可由此進入切換
Form1.h的程式編輯

各類清單切換



影像讀取

- 讀取圖片的元件
 - button
 - pictureBox
 - openFileDialog
 - saveFileDialog



影像讀取

□ #pragma once

需要使用的
opencv library

```
#include <opencv2/core/core_c.h>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/highgui/highgui.hpp>
```

```
#include <msclr/marshal_cppstd.h>
```

```
#include <iostream>
#include <math.h>
```

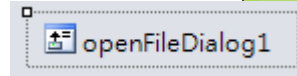
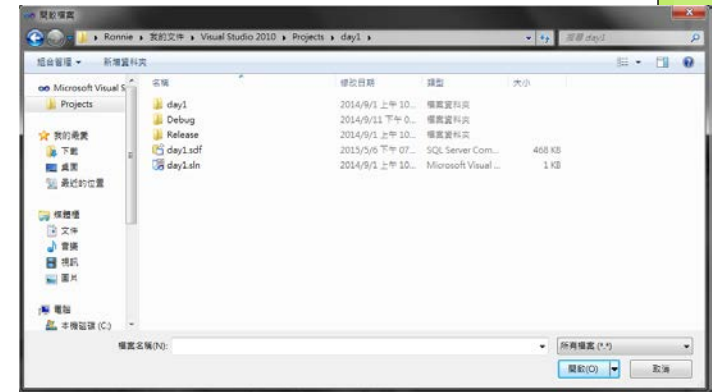
```
using namespace cv;
using namespace std;
```

如要使用opencv的function，需指定是cv函式庫裡的function。
如未宣告則Ex. cv::Mat。
Std同理。

Mat Img;

陣列宣告
(global變數)

影像讀取 Open



```
private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e) {
```

```
    if(openFileDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK){
```

```
        //pictureBox為把圖show在form上的元件
```

```
        pictureBox1->Load(openFileDialog1->FileName);
```

以VS2010 Windows Form 顯示影像
透過 openFileDialog 取得檔案路徑

```
        //imread為opencv的語法, 而 msclr 是為取得 Windows Forms 中 openFileDialog的檔名
```

```
        Img = imread(msclr::interop::marshal_as<std::string>(openFileDialog1->FileName), 0);
```

```
        //imshow 是另開視窗把 Img show在名為 "show1" 的視窗上  
        imshow("show1", Img);
```

若使用opencv imread()讀圖
需#include <msclr/marshal_cppstd.h>
用來轉換string

```
    }
```

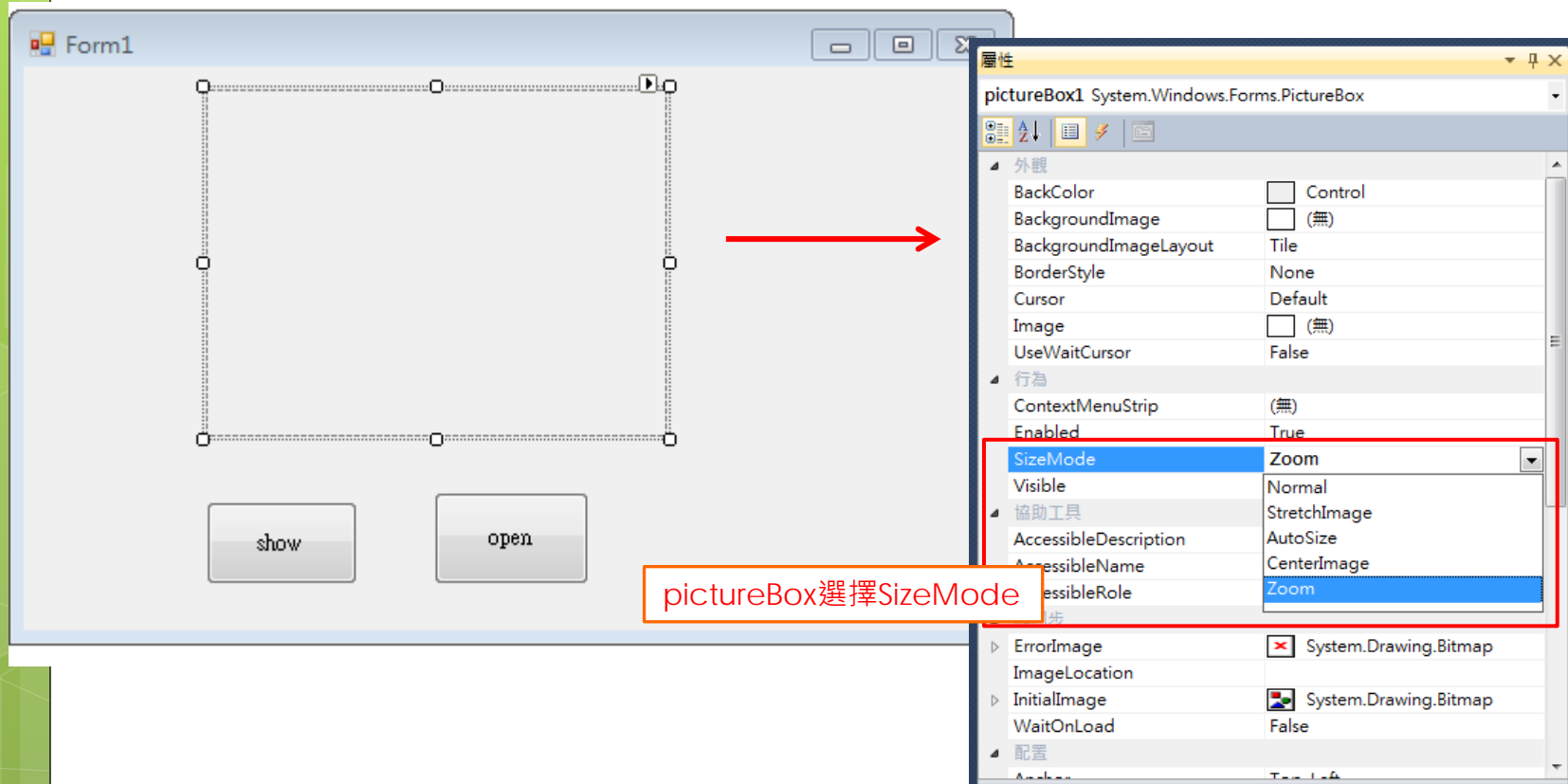
```
    else{
```

```
        return;
```

```
    }
```

```
}
```

影像讀取 Open





openFileDialog1



menuStrip1



saveFileDialog1

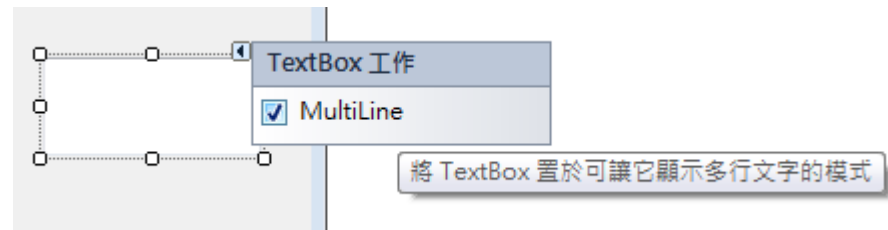
影像儲存 Save

Opencv save image sample

```
private: System::Void button9_Click(System::Object^ sender, System::EventArgs^ e) {  
  
    if(saveFileDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK){  
  
        //需注意filename之副檔名, 可配合 saveFileDialog1 取得  
        imwrite( msclr::interop::marshal_as<std::string>(saveFileDialog1->FileName), Img);  
  
    }  
    else{  
        //NO Filename  
        return;  
    }  
  
}  
};
```

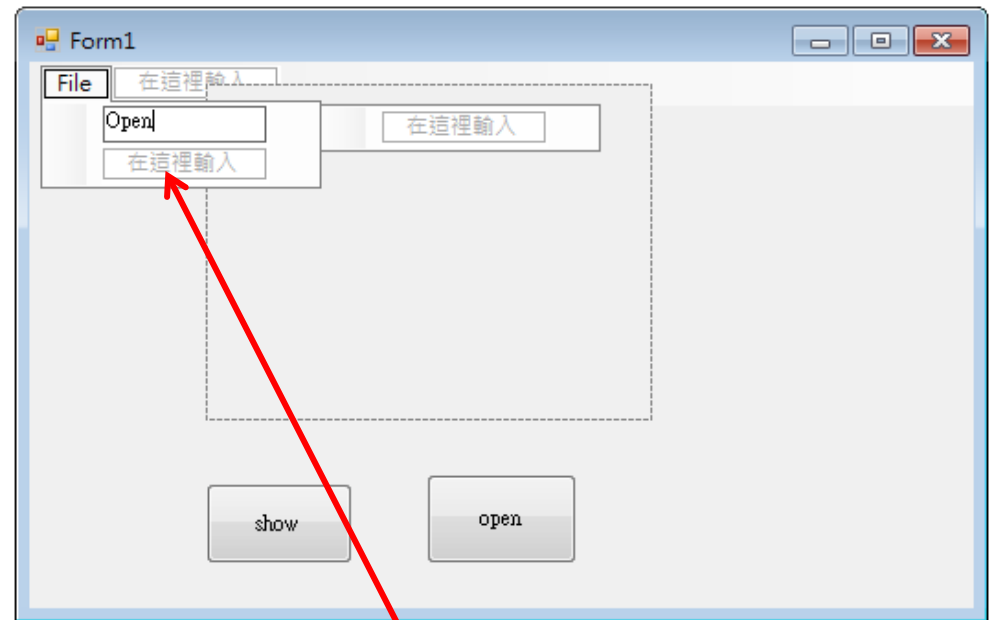

常用元件

- Button
- TextBox
- PictureBox
- OpenFileDialog
- MenuStrip



常用元件

- Button
- TextBox
- PictureBox
- OpenFileDialog
- MenuStrip



openFileDialog1

menuStrip1

影像處理



Course Outline – Day1

- 開發環境
 - VS2010 Windows Form + OpenCV
- 影像處理
 - Grayscale, Color Image (灰階與彩色影像)
 - Filter (濾波器)
 - Color Space (色彩空間)
 - Histogram (直方圖)
 - Image Morphology (型態影像學)
 - Others

Grayscale & Color Image

- 數位影像

將現實影像經由各種媒介(如數位相機)做量化，利用有限數值和數量的像素點(pixels)來描述二維影像。

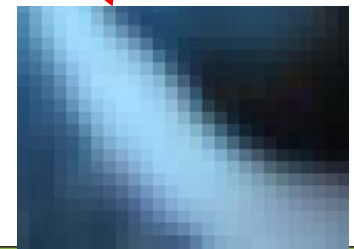
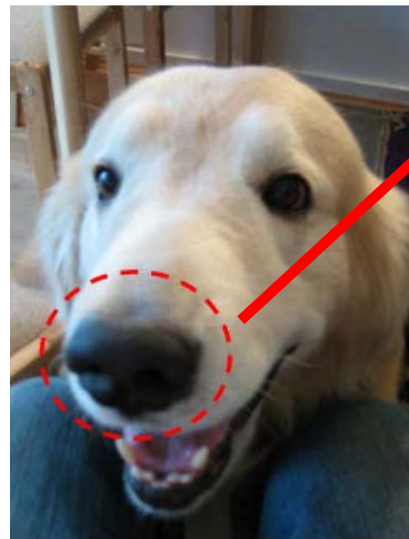
- Pixel (像素)

影像上的小方格單位

Ex: 200x300影像

⇒ 寬200像素

⇒ 高300像素



○ 數位影像

0	1	0	1	1	1	0	1
---	---	---	---	---	---	---	---

➤ 1 byte = 8 bits

⇒ $2^8 = 256$

⇒ 0 ~ 255(10) = 0x00 ~ 0xFF(16)

○ 灰階影像 (grayscale)

➤ 1 channel (亮度資訊)

➤ 或稱灰階值 (gray values)

• 1 pixel 為 1 byte 大小

• 即每 1 pixel 可存的值為 0~255 (256種顏色)

➤ 意即將亮度量化成0~255之間

• 越接近 0 代表亮度越低, 視覺上呈現黑色

• 越接近 255 代表亮度越高, 視覺上呈現白色



255	255	255	255	255	255	255	255
77	77	77	77	77	77	77	77
255	255	128	128	0	255	255	255
255	255	128	128	0	0	0	255
255	255	128	128	0	0	0	255
255	255	128	128	0	0	0	255
255	255	255	255	255	255	255	255

● 彩色影像 (Color Image)

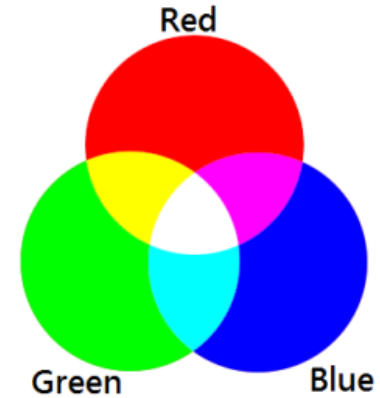
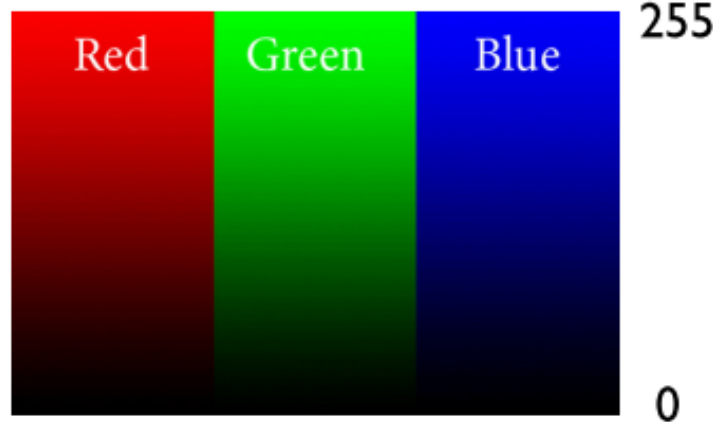
➤ 3 channels (R, G, B)

- 將色光三原色做量化
- 1 pixel 為 3 bytes 大小
- 即每 1 pixel 可存的值為 $0 \sim 256^3 - 1$

⇒ $0 \sim 0x\text{FFFFFF}$ (pixel values) ⇒ 共 256^3 種顏色




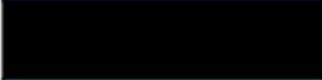




➤ 意即將三原色各別量化成 $0 \sim 255$ 之間

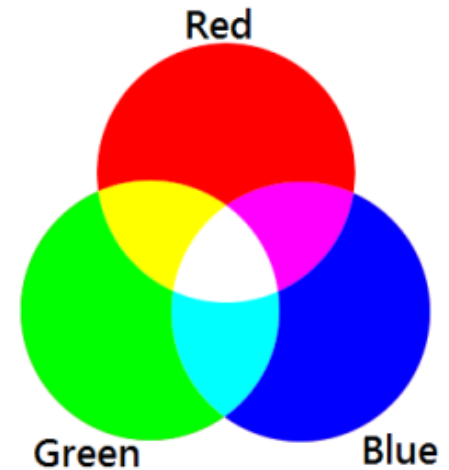
- 越接近 0 代表該顏色亮度越低
- 越接近 255 代表該顏色亮度越高



● 彩色影像 (Color Image)

- 3 channels (R, G, B)
- 小畫家 => 調色盤
- For example

	(R, G, B)	Pixel值
	(255, 0, 0)	0xFF0000
	(0, 255, 0)	0x00FF00
	(0, 0, 255)	0x0000FF
	(0, 0, 0)	0x000000
	(0, 255, 255)	0x00FFFF
	(255, 255, 0)	0xFFFF00
	(172, 0, 0)	0xAC0000
	(255, 192, 0)	0xFFC000



● 灰階與彩色比較

	Channels	Size per Pixel	Range (10)	Range (16)
Gray Scale	1	1 byte	0~255	0~0xFF
RGB	3	3 bytes	0~256 ³ -1	0~0xFFFFFFFF

● 影像大小

➤ EX: 一張320x320未壓縮的RGB影像，其像素總量為320(寬) X 320(高) = 102,400 (pixels)

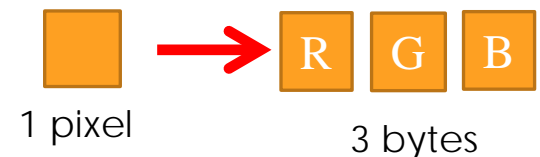
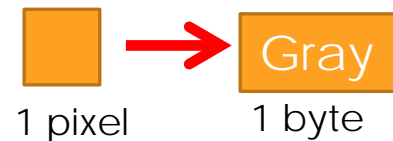
➤ 則其檔案大小為：

320 X 320 X 3 (bytes)

= 102,400 X 3 (bytes)

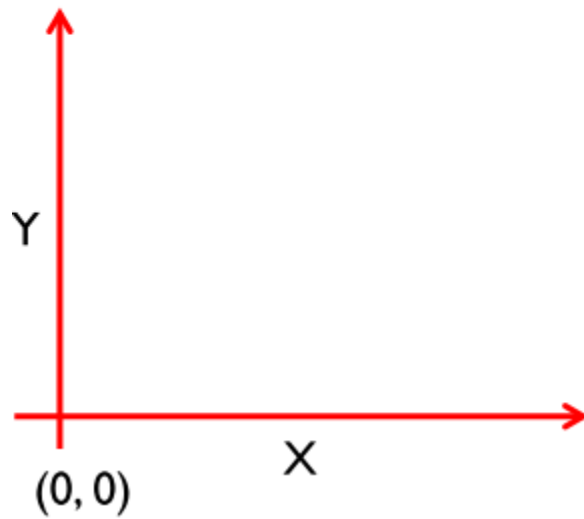
= 100 X 3 (KB)

= 300 (KB)

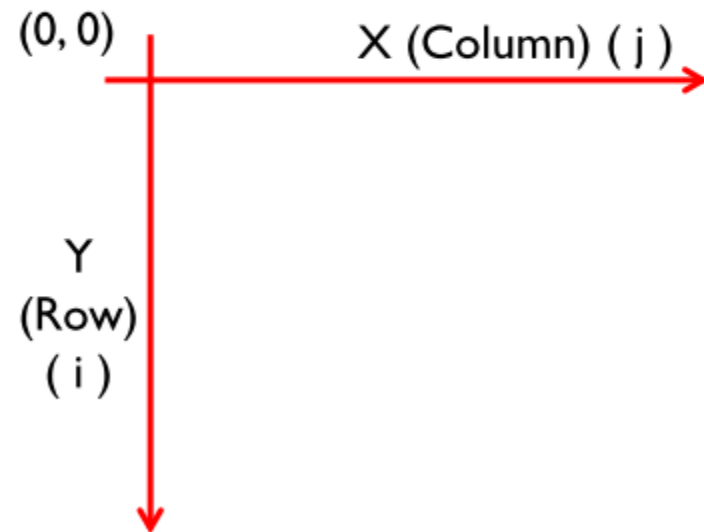


○ 影像座標

1. 傳統座標



2. 影像座標



Bitmap

- Bitmap 格式可分為三部分
 - 檔頭 (File Header)
 - 調色盤 (Palette) (except 24-bit)
 - 資料 (Image Data Array)
- 常見深度為8bit, 24bit
 - 其他
 - 1bit, 2bit, 4bit...etc



Header

Platte

Image
Data

Bitmap

- 兩種常見轉灰階公式
 - $\text{gray value1} = (R+G+B)/3$
 - $\text{gray value2} = 0.299R + 0.587G + 0.114B$ // (YCbCr)
- 如何以24-bit Bitmap 表現灰階影像
 - 讓 RGB 3 channels 的值都相同
 - Ex: 0xF2F2F2 顯示即為亮度 0xF2 的灰階影像
(242, 242, 242)

Bitmap

- 調色盤 (Palette)
- Index 索引值

0	1	- - - -	- - - -	- - - -	- - - -	63
64	65					⋮
128	129					⋮
192	193	- - - -	- - - -	- - - -	- - - -	255

- 8-bit調色盤為大小256格的Color Table, 每一格4bytes存一種顏色, 即Index所代表的顏色
- 顏色第一個byte保持為0
 - 例如灰階影像中 $\text{Color}[255] = 0x00FFFFFF$
 - $\text{Color}[0] = 0x00000000$

Bitmap

- 資料 (Image Data Array)
 - 即pixel值, 但pixel所存的為索引值, 再從調色盤中找出其對應的顏色

pixel data

100



Palette[100] -->

Palette

0	0x00000000
	0x00000001
100	0x00RRGGBB
255	0x00FFFFFF

Homework #1

- 第一支影像程式: Inverse

- Inverse 就是所謂的反白或是底片效果。
- 此處理過程將影像原來為亮的像素變暗,而原來暗的像素即變亮。




```
#pragma once
```

需要使用的
opencv library

```
#include <opencv2/core/core_c.h>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/highgui/highgui.hpp>
```

```
#include <msclr/marshal_cppstd.h>
```

```
#include <iostream>
#include <math.h>
```

如要使用opencv的function，需指定
是cv函式庫裡的function。
如未宣告則Ex. cv::Mat。
Std同理。

```
using namespace cv;
using namespace std;
```

```
Mat Img;
```

陣列宣告
(global變數)

```
private: System::Void button10_Click(System::Object^ sender, System::EventArgs^ e) {  
    //宣告矩陣的寬高  
    int height = Img.rows, width = Img.cols;  
  
    //opencv的顯示圖片語法  
    imshow("input", Img);  
  
    //灰階反白即為 最大值255-每個pixel的值  
    for (int i=0;i<height;i++){  
        for (int j=0;j<width;j++){  
            Img.at<uchar>(i,j) = 255 - Img.at<uchar>(i,j);  
        }  
    }  
    imshow("output", Img);  
    waitKey(0);  
  
    //opencv的儲存圖片語法  
    imwrite("inverse.jpg", Img);  
}  
};
```