

# 第十五屆旺宏科學獎

## 成果報告書

參賽編號：SA15-298

作品名稱：簡易光譜儀與即時顯示程式之開發

姓名：陳芊卉

關鍵字：DVD 光柵、簡易光譜儀、視訊處理

## 研究動機與目的：

在學校實驗室課程時，老師曾為大家介紹光譜儀的原理以及其在分析物質上的多種應用，但同時也提及其動輒數十萬甚至上百萬的昂貴價格，以及每逢故障就必須等待廠商派專人進行修繕的缺點。研讀其他自製光譜儀的相關研究後，發現都是在拍攝光譜影像後，使用現有的應用軟體進行分析，不只需要多道的人工處理程序外，還不見得能夠確定得到的頻譜波長數據是否正確。

有鑑於此，是否能夠利用生活中隨手可得之材料，自行製造出價格低廉、操作方便、易於攜帶且有一定可信度的簡易光譜儀，並搭配自行撰寫的程式來完成光源頻譜檢測及溶液的透光率、吸光度的檢測？這個構想非常具有趣味性，決定由此著手實驗，並進行進一步的探討。本研究的主要目的有下列三點：

- 一、自製光譜儀，找出光柵角觀測角的最佳組合。
- 二、自行撰寫程式，可用最簡便的方式校正頻譜，並能即時呈現光譜及頻譜曲線。
- 三、檢測水溶液的透光率及吸光度。

## 器材及軟體：

DVD-R 空白光碟片、0.1mm 狹縫、0.15mm 狹縫、各式光源燈泡燈管、筆記型電腦、紅外線網路攝影機（200 萬畫素、手動對焦）、光譜儀 PG Instruments Ltd. Spectrophotometer T80、塑膠製比色管、數位相機 CANON EOS 100D。

## 研究過程：

### 一、光譜儀的分光原理

可見光是電磁波譜中人類眼睛可以看得見、感受得到的部分，波長的範圍一般是落在 380nm 到 780nm 之間，人眼對於不同波長的電磁波會有不同顏色的視覺神經感覺。光譜儀的主要的功用是將各種波長組合而成的光分解為光譜線的科學儀器，主要是由稜鏡或繞射光柵等構成，當光通過分光元件（如光柵、稜鏡）進行分光後，即會依照波長的大小順次排列，配合著偵測器，即可測量物體表面對於各種波長光線的反射情形，或是光線穿透物體時，對於各種波長光線的吸收情形。

光譜儀可對物質的結構和成分進行觀測、分析和處理，具有分析精度高、測量範圍大、速度快等優點。所以廣泛應用於冶金、地質、石油化工、醫藥衛生、環境保護等部門，也是軍事偵察、宇宙探索、資源和水文勘測所必不可少的儀器[8]。

光譜儀依分光元件的不同分為稜鏡光譜儀、光柵光譜儀和干涉光譜儀，其中以光柵光譜儀最為常見。光柵可分為透射與反射兩種，透射光柵是在光學平玻璃上刻劃出一道道等間距的刻痕，刻痕處不透光，未刻處則是透光的狹縫；反射光柵則是在金屬反射鏡上刻劃一道道刻痕，刻痕處不反光，未刻處則可以反光。當光線照射到光柵時會產生繞射現象，由於不同波長的光線會有不同的繞射角而達到分光的效果。不同波長  $\lambda$  的光線照射到光柵後其入射角  $\theta$  及繞射角  $\alpha$  的關係遵循著下列的光柵

方程式[9]：

$$d(\sin \theta + \sin \alpha) = m\lambda \Rightarrow \alpha = \sin^{-1}(m\lambda/d - \sin \theta)$$

其中  $d$  為光柵的狹縫間距， $m = 0, \pm 1, \pm 2, \dots$  為繞射的級數，由此方程式可以看出，當光柵的狹縫間距為定值時，不同波長的光線會有不同的繞射角，藉以達到分光的效果。

以圖 2 為例，波長 550nm 的綠光以  $\theta = 14^\circ$  的入射角，穿透每 mm 有 1000 條線（狹縫間距為 1000nm）的透射光柵後，所產生的第 1 級繞射角為

$$\alpha = \sin^{-1}(1 \times 550/1000 - \sin 14^\circ) = 17.9^\circ$$

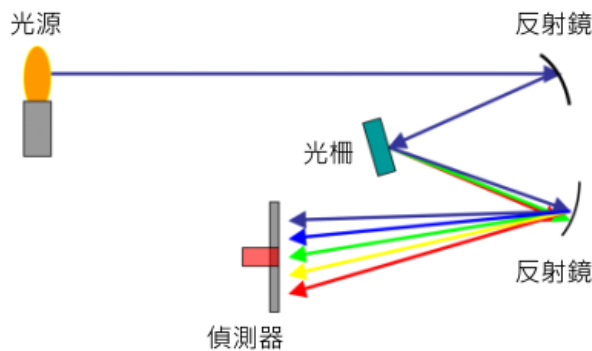


圖 1 維基百科網站上光譜儀的簡圖

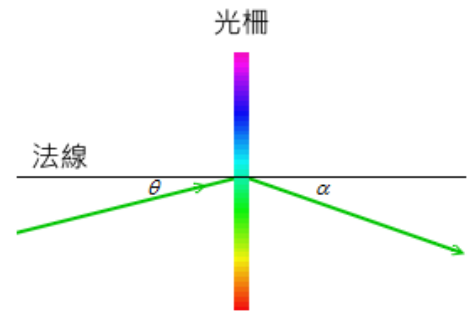


圖 2 光線的人射角與繞射角

## 二、簡易光譜儀的相關研究

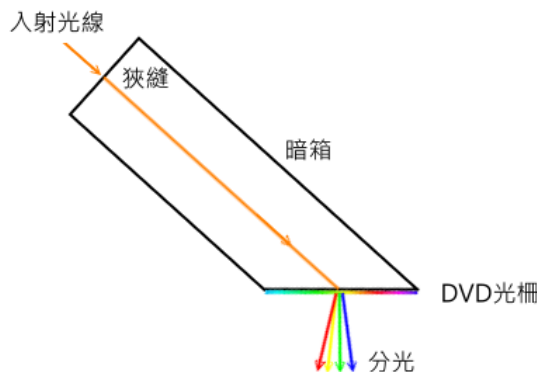
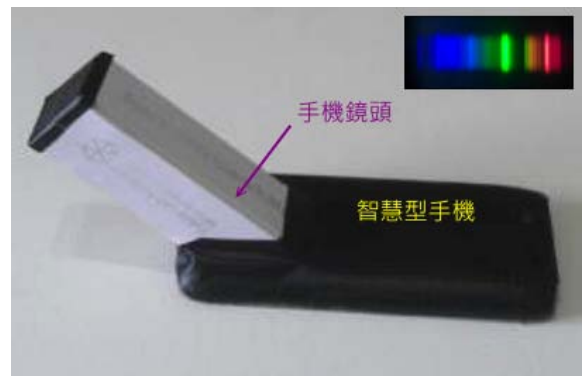


圖 3 參考資料[4]所使用的簡易光譜儀及使用智慧型手機拍攝光譜的情形



參考資料[1][2][3][4]利用自製的光譜儀分別進行不同領域的研究，甚至簡易光譜儀 DIY 也變成了近年科普教學重要的活動項目[5][6][7]。圖 3 為參考資料[4]依據公共實驗室網站[14]的方法而自製的簡易光譜儀，除了可以人眼觀察光譜外，還可以搭配智慧型手機來拍攝光譜。這些簡易光譜儀的組成元件大致可分為暗箱、狹縫、光柵、觀測器、分析軟體等五個部份，分述如下：

### 1. 暗箱

主要的用途就是避免外界的光線，影響了光譜的呈現與觀察，在製作暗箱時，要注意暗箱內部不可以有漏光的情形。

### 2. 狹縫

可以阻擋外界雜亂的光線，僅讓平行於狹縫的光線入射到光柵上。狹縫的寬度影響著進光量以及光譜的解析度，狹縫越寬，光譜的解析度就越差，但有較大

的進光量；狹縫越細，光譜的解析度就越好，但進光量也跟著減少。在選用或製作時，狹縫要儘可能的讓寬度勻稱且平直，依據上述參考資料的說明，狹縫寬度大約介於 0.1mm 到 0.5mm 之間為佳。

### 3. 光柵

可以使用專業的光學光柵，但價格較為昂貴，也可以使用大家隨手可取得的 VCD 或 DVD 光碟片來自製，花費則是非常的平民化。由於線條越密集的光柵分光的效果越好，所以使用每 mm 有 1200 條線的光學光柵，或是每 mm 約有 1350 條線的 DVD-R 光碟片是較好的選擇，由參考資料的實驗結果看來，專業的光學光柵所產生的光譜會有較佳的品質。

### 4. 觀測器

總共有三種觀測器來檢視光譜，第一種就是用眼睛直接觀看，第二種是用相機拍攝（如 CANON 5D、骨董級的 NIKON FM2、智慧型手機），第三種就是用網路攝影機（webcam）直接置於暗箱中。依據實際操作及研讀參考資料的心得，人眼直接觀察的光譜最為清晰與漂亮，再來是高檔相機所拍攝的光譜，廉價的網路攝影機則品質不佳。雖然如此，但就取像與分析光譜的方便性而言，網路攝影機最為便捷，至於人類眼睛所看到的美麗光譜，目前還不知該如何取像與別人分享呢。

### 5. 分析軟體

大多數是先將光譜影像數位化，接著使用影像軟體 ImageJ 或是公共實驗室[14]提供的 Java 程式，分析影像後輸出頻譜資料，再將頻譜資料放到 Excel 中繪出各種波長強度的光譜分析圖，不僅如此，為了使得數據更為正確，一定還得找些已知波長的光源來當作校準參考，例如紅光雷射（635nm）、綠光雷射（532nm）等。

## 三、光譜儀的開發測試

研讀了參考資料[1][2][3][4][5][6][7]的成果與經驗，希望利用一般人也容易取得材料，能夠發展出僅使用這些材料可以達到最佳效果的自製光譜儀，另外也嚐試著自行撰寫光譜的分析程式，可與本研究的光譜儀結合著使用，這樣才能達到即時分析的便利性。

### 1. 構想

與許多自製的光譜儀相同，也是使用暗箱、狹縫及 DVD-R 光碟片剪裁而成的光柵。至於觀測器部分，打算採用較高規格的網路攝影機，最好是紅外線攝影機，不然就得自行移除紅外線濾片，這樣取得的波長範圍可以更寬廣些，而且透過 USB 與電腦連線後，自行撰寫的程式就可以將攝影機傳送過來的光譜視訊串流做即時的分析處理與頻譜呈現。

圖 4 為光譜儀構想圖，光源入口到狹縫之間，預留了放置比色管的位置，如果需要測量溶液的吸收光譜時，會方便許多。

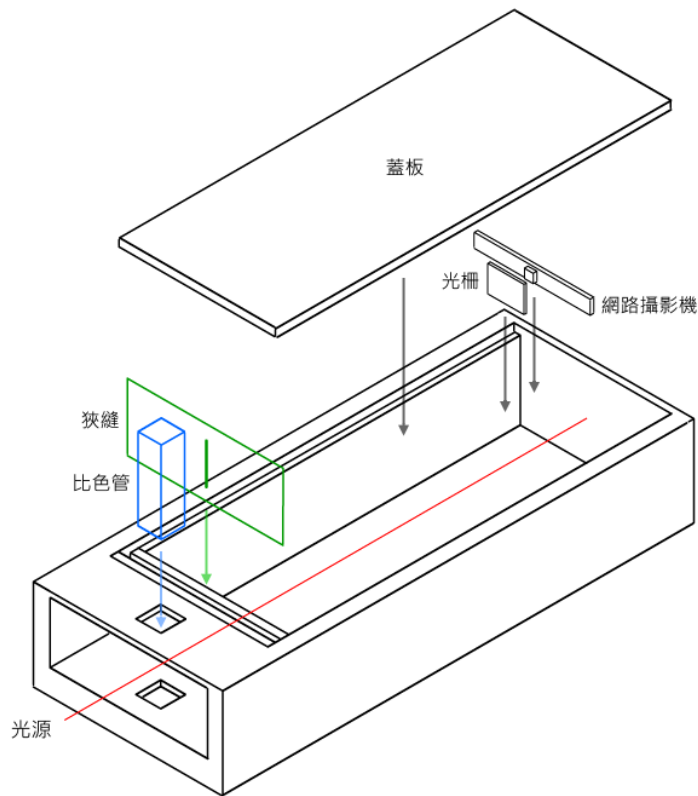


圖 4 光譜儀構想圖

## 2. 光柵角與觀測角的理論推導

綜觀參考資料中的自製光譜儀，光柵的放置方式可分為兩種類型，第一種方式是與入射光垂直，然後由第 1 級繞射角的地方觀測光譜，以 DVD 光柵及波長位於可見光中間值的 550nm 綠光為例，DVD 光柵的狹縫間距約為 740nm，利用光柵方程式，當入射角為  $0^\circ$  時，可以計算出繞射角為  $48.0^\circ$ ，如圖 5-1 所示。第二種方式為慢慢旋轉光柵，當繞射角為  $0^\circ$  時，這時可由光柵的法線處觀測到第 1 級繞射的光譜，同樣地以 DVD 光柵及波長為 550nm 綠光為例，利用光柵方程式，當繞射角為  $0^\circ$  時，光柵順時旋轉了  $36.6^\circ$ ，如圖 5-2 所示。

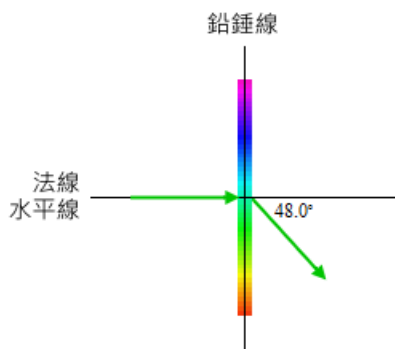


圖 5-1 入射角為  $0^\circ$

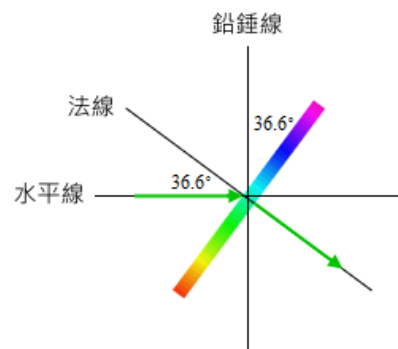


圖 5-2 繞射角為  $0^\circ$

由參考資料的研究結果看來，若非使用高階相機，而採用網路攝影機來拍攝光譜，似乎都很難有品質較佳的影像，是否可藉由旋轉光柵及調整網路攝影機的觀測角度，來找到最佳的組合，以提升網路攝影機拍攝光譜影像的品質呢？先定義光柵順時旋轉時與鉛錘線的夾角，稱之為光柵角，其實也等於入射角  $\theta$ ；第 1 級繞射角  $\alpha$  加上光柵角  $\theta$  之和，為水平線與第 1 級繞射光線的夾角，稱為

水平繞射角  $\alpha + \theta$ ；另外，網路攝影機拍攝光譜時與水平線的夾角，稱之為觀測角  $\beta$ ，如圖 6 所示。

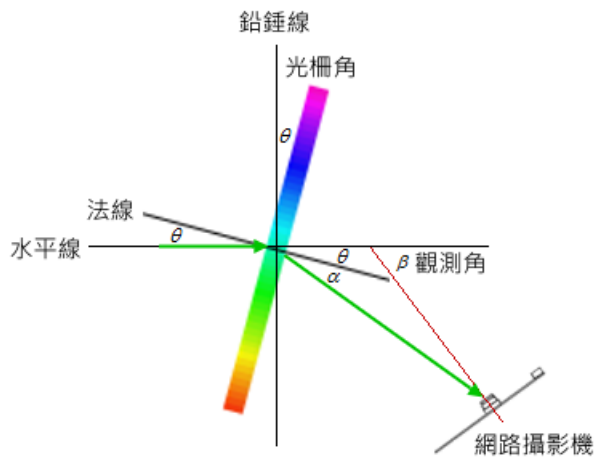


圖 6 光柵角與觀測角

在進行此項實驗前，先利用光柵方程式  $m\lambda = d(\sin\theta + \sin\alpha)/\cos\theta$ ，來推算以 DVD 為光柵 ( $d = 740nm$ )，波長位於可見光中間值的 550nm 綠光第 1 級繞射 ( $m = 1$ ) 之理論水平繞射角 ( $\alpha + \theta$ )，這樣可以方便實驗的順利進行，必免花費時間於不可能有光譜的地方尋找光譜。

光柵角	入射角 $\theta$	$\cos\theta$	$\sin\theta$	$\sin\alpha$	繞射角 $\alpha$	水平繞射角 $\alpha + \theta$
0°	0°	1.000	0.000	0.743	48.0	48.0
10°	10°	0.985	0.174	0.558	33.9	43.9
20°	20°	0.940	0.342	0.356	20.9	40.9
30°	30°	0.866	0.500	0.144	8.3	38.3
40°	40°	0.766	0.643	-0.073	-4.2	35.8
50°	50°	0.643	0.766	-0.288	-16.8	33.2
60°	60°	0.500	0.866	-0.494	-29.6	30.4

表格 1 光柵角及利用光柵方程式推導的理論水平繞射角

### 3. 測試平台

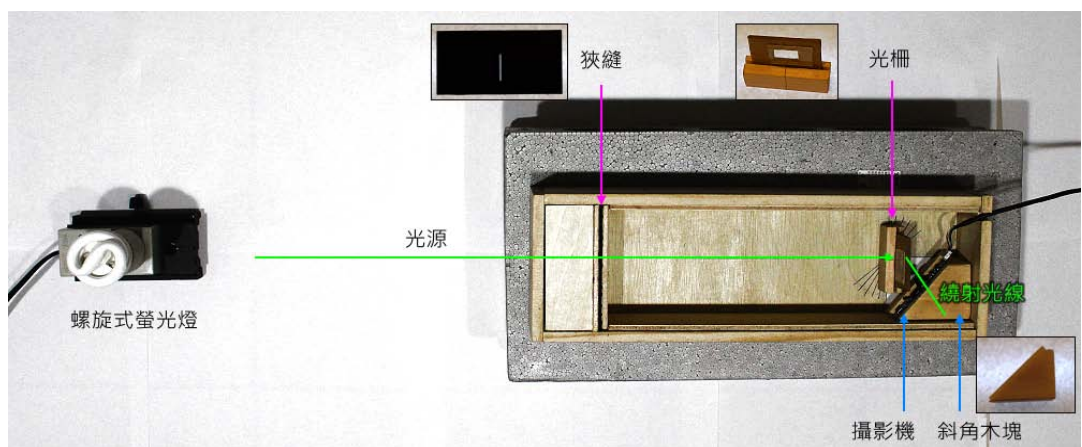


圖 7-1 測試平台

實驗的重點著重於如何尋找出光譜儀中 DVD 光柵的光柵角及網路攝影機的觀測角之最佳角度組合，依據光譜儀的構想圖製作了測試平台，如圖 7-1 所示。

其中光柵是先固定在厚紙板上，然後插入一個可以旋轉的木製插槽上，這樣可以方便調整光柵角。網路攝影機則使用雙面膠黏貼在斜角木塊上，雙面膠可方便於重新黏貼網路攝影機，使其鏡頭在光柵方程式推導出來的第一級繞射光線的路徑上。而斜角木塊的主要功能是穩固網路攝影機，藉由更換不同斜角的木塊，網路攝影機也會有不同的觀測角。圖 7-2 是了光柵角、理論水平繞射角、木塊斜角、觀測角的說明示意圖。

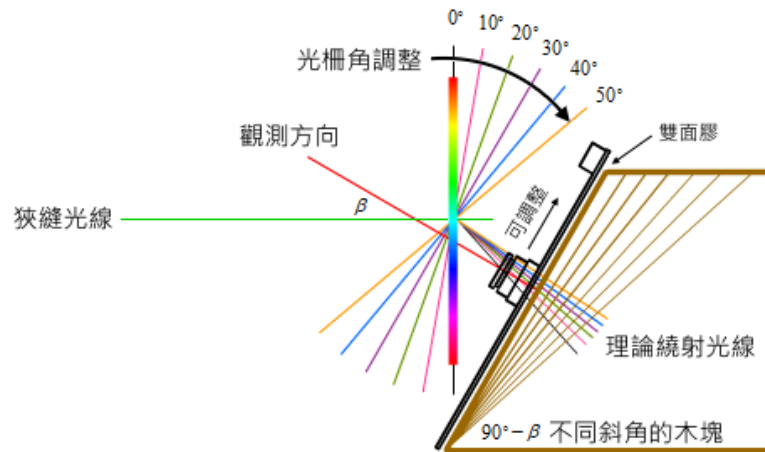


圖 7-2 光柵角、理論水平繞射角、木塊斜角、觀測角的說明示意圖

#### 4. 光譜的拍攝

紅外線網路攝影機是採用威聚科技的 UM17，先將焦距手動調整至 22cm 的地方，也就是狹縫光源到網路攝影機的距離，若是焦距太長會影響到光譜儀的大小尺寸。

當光柵角介於  $0^\circ \sim 50^\circ$  時，由光柵方程式推導 550nm 波長的綠光第 1 級繞射之理論水平繞射角約介於  $33^\circ \sim 48^\circ$  之間，所以準備了角度為  $42^\circ$ 、 $45^\circ$ 、 $48^\circ$ 、 $51^\circ$ 、 $54^\circ$ 、 $57^\circ$ 、 $60^\circ$  的斜角木塊，固定在其上的網路攝影機分別會有  $48^\circ$ 、 $45^\circ$ 、 $42^\circ$ 、 $39^\circ$ 、 $36^\circ$ 、 $33^\circ$ 、 $30^\circ$  的觀測角，這樣應該可以看到光柵角介於  $0^\circ \sim 50^\circ$  時所產生的光譜。

光柵角的測試範圍限定於  $0^\circ \sim 50^\circ$ ，主要是因為較大的光柵角，其理論水平繞射角也就越小，這樣斜角木塊的斜角就會變的越大，這樣會使得光柵與斜角木塊上的攝影機相衝突碰撞，無法將鏡頭移動至光譜產生的位置。若是增加兩者間的距離，因為不同波長的光線有不同的繞射角（色散），網路攝影機的前端小透鏡，將無法觀測到全部的光譜。

各種光柵角及觀測角的光譜拍攝步驟如下，拍攝結果如表格 2 所示。

步驟 1：先將  $42^\circ$  的斜角木塊（觀測角為  $48^\circ$ ）置入測試平台。

步驟 2：光柵角由  $0^\circ$  開始。

步驟 3：將網路攝影機用雙面膠黏貼在斜角木塊上，鏡頭必須在以目前光柵角推算出來的理論繞射光線路徑上。

步驟 4：將網路攝影機傳送過來的光譜影像擷取後存檔。

步驟 5：調整光柵角，增加  $10^\circ$ ，然後重覆步驟 3、步驟 4，直到光柵角到  $50^\circ$  完成為止。



步驟 6：依序更換 45°、48°、51°、54°、57°、60° 的斜角木塊（觀測角分別為 45°、42°、39°、36°、33°、30°），重覆步驟 2 至步驟 5，直到各角度的斜角木塊完成為止。

光柵角 觀測角	0°	10°	20°	30°	40°	50°
48°						
45°						
42°						
39°						
36°						
33°						
30°						

表格 2 各種光柵角及觀測角所拍攝的光譜

## 5. 光柵角與觀測角的最佳組合

要從表格 2 的 42 幅光譜影像中尋找最佳的光柵角與觀測角組合，是件非常困難的工作，有下列兩項基本要求必須符合：

- 所拍得的光譜圖上，必須能夠呈現光源的光譜特徵線條。
- 各光譜特徵線條所代表的波長，必須與其在影像中的像素位置是一線性比例的關係。

以日常生活最常見的螢光燈為例，參考資料[10]維基百科的網站有其精準的光譜圖，如圖 8 所示，其光譜圖中的特徵線條為編號 2（波長 436nm）、5（波長 546nm）、12（波長 611nm）的頻譜峰值處，分別由水銀、水銀、鎔元素所發出。

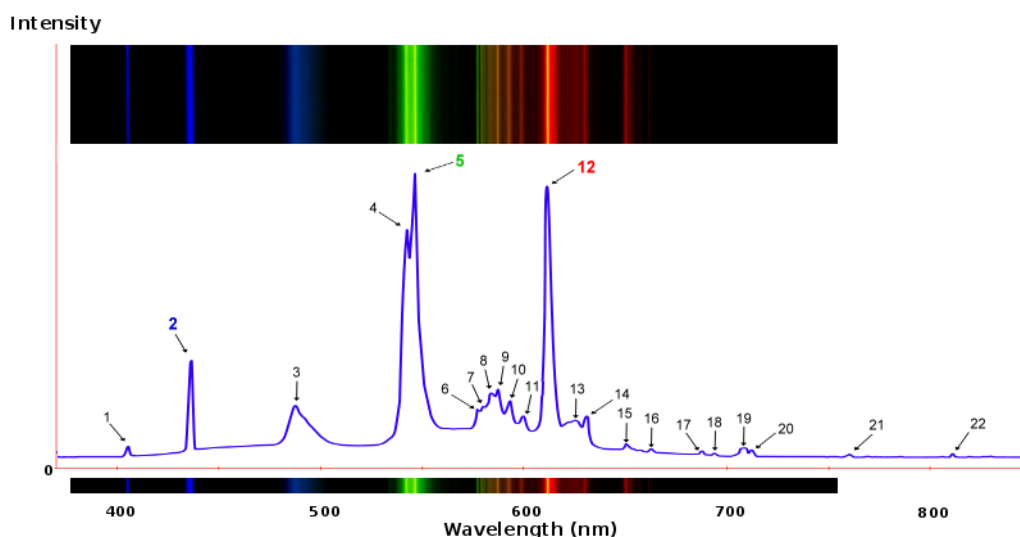


圖 8 螢光燈標準光譜圖

先以人工肉眼比對影像中光譜的清晰度及特徵線條為依據，經過兩輪的篩選，挑選出品質最佳的五幅影像，如圖 9 所示。再以 Photoimpact 軟體，將這五幅影像與參考資料[10]的精準螢光燈光譜圖做比對，觀察光譜中的特徵線條與其在影像中的像素位置是否是線性比例的關係，結果發現針對此網路攝影機，最佳的角度組合為光柵角 30°、觀測角 42°，圖 10 為光柵角 30° 觀測角 42° 及光柵角 50° 觀測角 33° 的光譜比對情形，光柵角 30°、觀測角 42° 有較佳的線性比例關係。



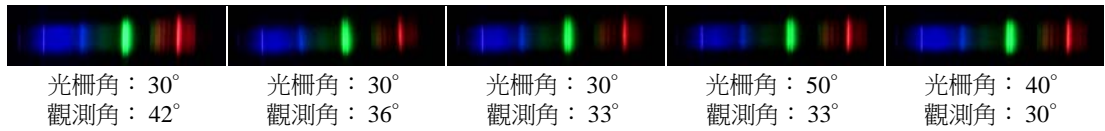


圖 9 經過人工肉眼篩選出來五幅品質較佳的光譜影像

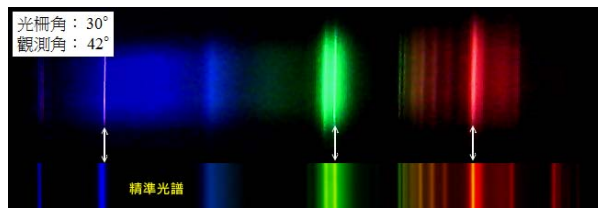


圖 10-1 光柵角 30° 觀測角 42° 與精準螢光燈光譜的比對

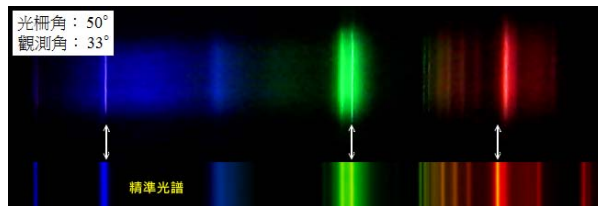


圖 10-2 光柵角 50° 觀測角 33° 與精準螢光燈光譜的比對

#### 四、自製光譜儀實作

依據構想圖及實驗所得的最佳光柵角與觀測角組合，以木工實作出光譜儀，為了方便更換狹縫及調校光柵及網路攝影機，並沒有用膠封死光譜儀，而採用上蓋板的方式，經觀察，光譜儀內部幾乎不會有透光的情形，但為了安全保險起見，再加層黑色粉彩紙做的紙蓋，這樣除了由狹縫進入的光線外，完全不會受其他外界光源的影響（圖 11）。

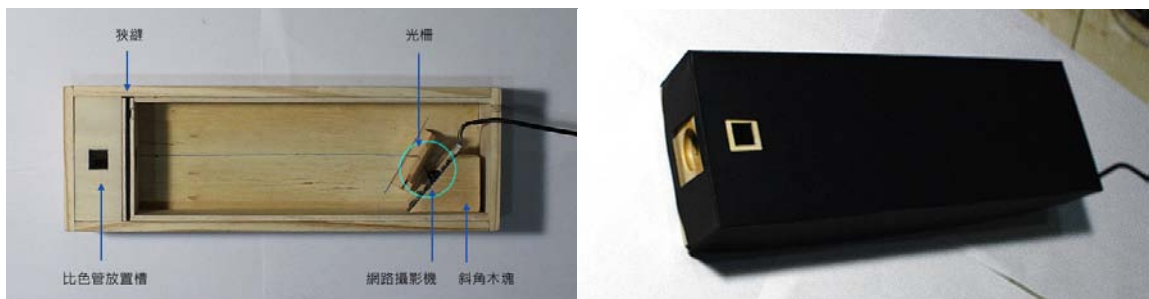


圖 11 自製光譜儀實作

#### 五、光譜分析軟體的開發

在 .Net 平台可以取得網路攝影機透過 USB 介面傳送過來的視訊方法有四種[15]，分別為 Windows API 的 AVICap32.dll、DirectShow、Windows Image Acquisition (WIA)、EmguCV，其中 EmguCV 在撰寫程式的時候較為簡潔，而且當取得網路攝影機的視訊後，還會進行一些影像的處理程序，所以採用 VB.Net 及 EmguCV 當做軟體程式的開發平台。

著名的 OpenCV 是一個跨平台之電腦視覺函式庫，由英特爾公司發起並參與開發，可用於開發即時的圖像處理、模式識別、電腦視覺等程式，而 EmguCV 是將 OpenCV 包裝起來供 .NET Framework 使用的函式庫，EmguCV 函式庫中與網路攝影機有關的 Capture Class 類別，所使用到的類別成員如表格 3 所列[12]。

Constructors ( 建構子 )	
Capture (String)	Create a capture from file or a video stream
Method ( 方法 )	
GetCaptureProperty (CAP_PROP)	Obtain the capture property
SetCaptureProperty (CAP_PROP, Double)	Sets the specified property of video capturing
Dispose	Release the capture
Start	Start the grab process in a sperate thread
Pause	Pause the grab process if it is running
Stop	Stop the grabbing thread
RetrieveBgrFrame()	Retrieve a Bgr image frame after an image is grabbed
Properties ( 屬性 )	
FlipHorizontal	Get or Set if the image should be flipped horizontally
FlipVertical	Get or Set if the image should be flipped vertically
Height	The height of this capture
Width	The width of this capture
Events ( 事件 )	
ImageGrabbed	The event to be called when an image is grabbed

表格 3 Capture Class 的成員

所開發的程式都用到表格 4 的基本架構，當 `_capture = New Capture()` 將選定的網路攝影機實體化後，使用 `_capture.Start()` 啟動一個新的執行緒來負責網路攝影機，並以 `AddHandler _capture.ImageGrabbed, AddressOf GetFrame` 告訴網路攝影機當發生 `ImageGrabbed` 事件時的處理常式 `GetFrame`，而影像處理的程序則都放在 `GetFrame` 常式中。

```

'程式開始執行
Private Sub Camera_Capture_Load
    Dim _Cameras As DsDevice() = DsDevice.GetDevicesOfCat(FilterCategory.VideoInputDevice)
    webCams = New Video_Device(_Cameras.Length - 1) {}
    For i As Integer = 0 To _Cameras.Length - 1 '將系統中的 webcam 列在 ComboBox
        webCams(i) = New Video_Device(i + 1, _Cameras(i).Name)
        cobSelectCamera.Items.Add(webCams(i).ToString())
    Next
End Sub

'關閉程式
Private Sub Camera_Capture_FormClosed
    If _capture IsNot Nothing Then
        If _capture.GrabProcessState = System.Threading.ThreadState.Running Then
            _capture.Pause()
            _capture.Stop()
            RemoveHandler _capture.ImageGrabbed, AddressOf GetFrame '移除ImageGrabbed事件處理常式
        End If
        _capture.Dispose()
    End If
End Sub

'開啟/關閉 Camera
Private Sub btnStart_Click
    If Not _captureInProgress Then '按下開啟視訊按鈕
        If cobSelectCamera.SelectedIndex <> CameraDevice Then '與選取後的 webcam 連接
            SetupCapture(cobSelectCamera.SelectedIndex)
        End If
        _capture.Start() '開始 capture
    Else '按下關閉視訊按鈕
        _capture.Pause() '暫停 capture
    End If
End Sub

'_capture 與選取的 webcam 連接後初始化
Private Sub SetupCapture
    CameraDevice = Camera_SelectedIndex
    If _capture IsNot Nothing Then '若_capture 已取得 webcam, 則先釋放_capture

```

```

If _capture.GrabProcessState = System.Threading.ThreadState.Running Then
    _capture.Pause()
    _capture.Stop()
    RemoveHandler _capture.ImageGrabbed, AddressOf GetFrame '移除 ImageGrabbed 事件處理常式
End If
_capture.Dispose()
End If
Try
    _capture = New Capture(CameraDevice) 'Set up capture device
    _capture.SetCaptureProperty(CAP_PROP.CV_CAP_PROP_FRAME_WIDTH, 1280) '寬
    _capture.SetCaptureProperty(CV_CAP_PROP_FRAME_HEIGHT, 720) '高
    AddHandler _capture.ImageGrabbed, AddressOf GetFrame '加入 ImageGrabbed 事件處理常式
Catch ex As NullReferenceException
    MessageBox.Show(ex.Message)
End Try
End Sub

'擷取影像
Private Sub btnCapture_Click(sender As Object, e As EventArgs) Handles btnCapture.Click
    DisplayImage(outFrame, imgCapture)
End Sub

'儲存擷取的影像
Private Sub btnSave_Click
    SaveFileDialog1.ShowDialog()
End Sub
Private Sub SaveFileDialog1_FileOk
    imgCapture.Image.Save(SaveFileDialog1.FileName)
End Sub

'_capture.ImageGrabbed 每當獲得 webcam 視訊畫面
Private Sub GetFrame
    inFrame = _capture.RetrieveBgrFrame() '獲得一幅新的影像畫面
    ImageProcessing() '處理獲得的影像畫面 Input :I nFrame, Output : outFrame
    DisplayImage(outFrame, imgCamera) '顯示 outFrame 影像
End Sub

```

表格 4 光譜儀程式主要架構

### 1. 網路攝影機視訊接收與擷圖程式

程式執行後，先由 `cobSelectCamera` 選擇網路攝影機，按下開啟視訊後，即可在圖形框 `imgCamera` 中看到網路攝影機傳送過來的視訊，每幅畫面的尺寸設定為  $1280 \times 720$  畫素，由標籤 `lblStopWatch` 可以看出，網路攝影機在此畫面大小的運作狀態下，每秒鐘僅可以傳送 6 至 7 幅的畫面。按下按鈕 `btnCapture` 即可將目前的視訊畫面擷取至圖形框 `imgCapture`，若想儲存圖檔，按下按鈕 `btnSave` 即可。

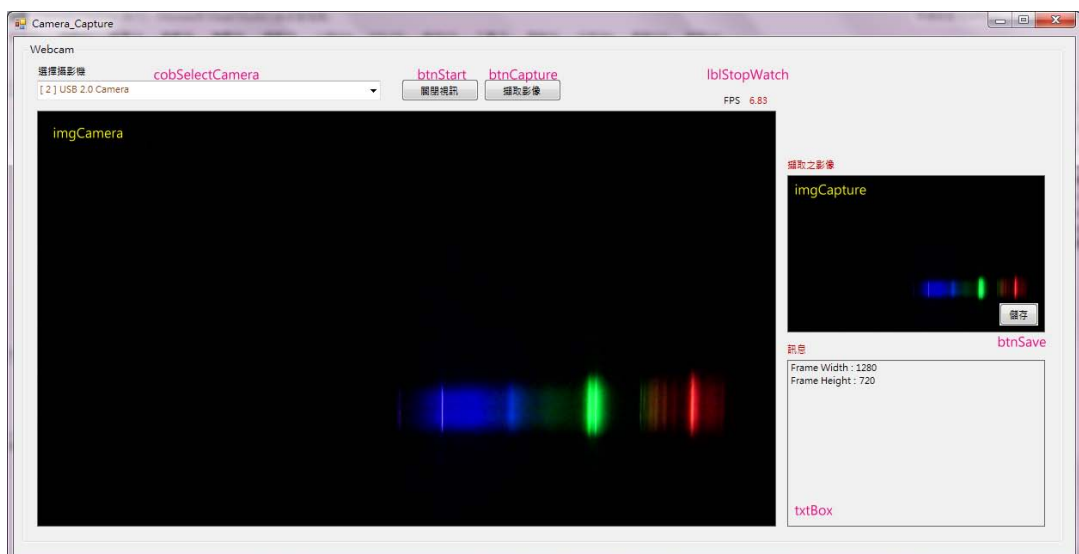


圖 12 網路攝影機視訊接收與擷圖程式

## 2. 單張影像的光譜顯示程式

本程式可以分析影像的光譜，影像的來源可由網路攝影機的視訊中擷圖，也可以是影像檔案的載入，以滑鼠點選圖形框 `imgCapture` 中光譜的位置，即會產生黃色的光譜掃描線，然後將掃描線上的光譜影像填入 `imgSpectrum` 圖形框中，同時也進行頻譜分析並在 `picSpectrum` 中繪出頻譜曲線，若滑鼠移動到此圖形框中，可顯示滑鼠游標所在的位置的詳細頻譜資訊，如圖 13-1 所示。

另外，本程式也提供了非常方便的頻譜校正模式，當擷取或載入此光譜儀所拍攝的螢光燈光譜影像後，按下 `radCalibration` 的單選按鈕後，拖曳 `trackB2` 與 `trackEu` 捲軸即可進行，如圖 13-2，主要是利用螢光燈中水銀 (Hg) 及銻 (Eu) 的兩道發射光譜[10]，波長分別為 436nm 及 611nm，就利用它們在光譜影像中的位置，以等比例的方式推得其他色光的波長。

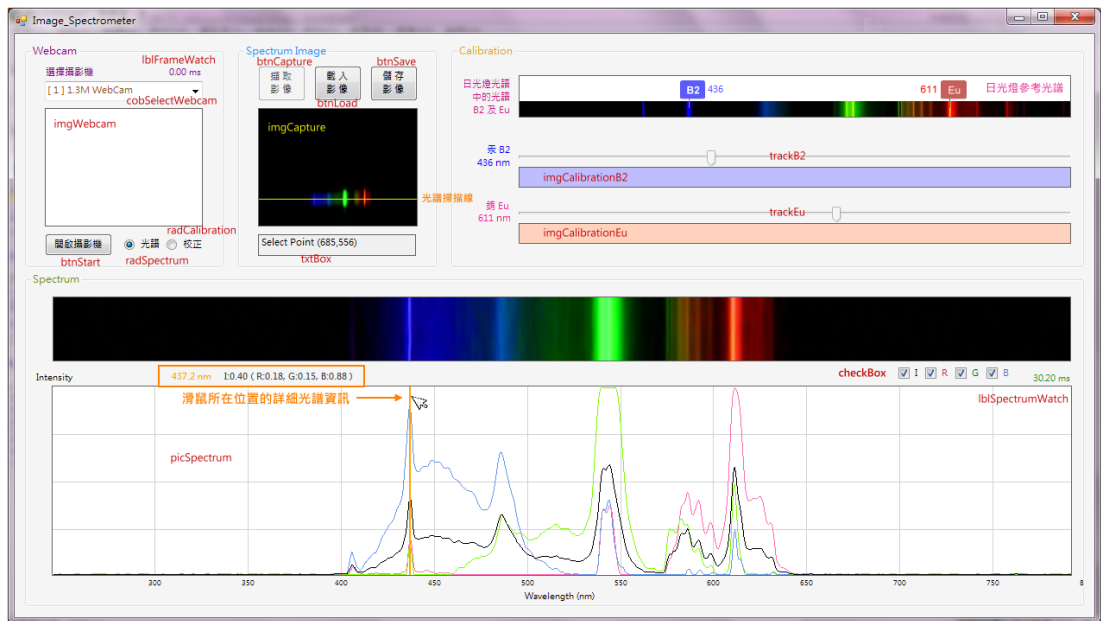


圖 13-1 單張影像的光譜顯示程式

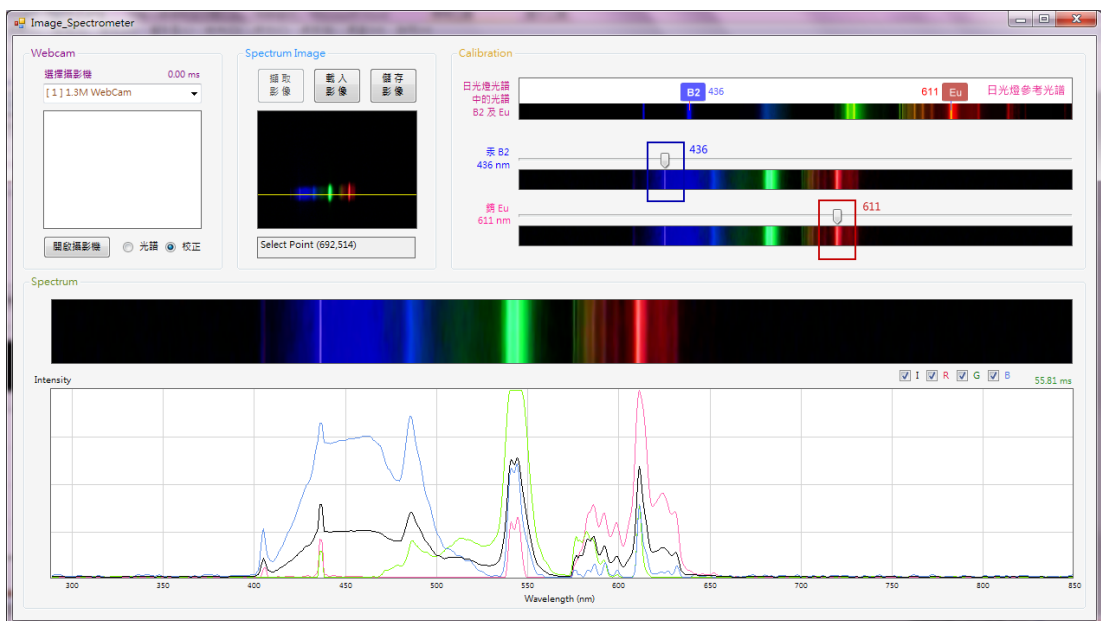


圖 13-2 利用螢光燈光譜影像進行頻譜校正

### 3. 即時光譜顯示程式

第三個程式嘗試著將網路攝影機傳送過來的光譜視訊，即時地繪製其頻譜，如圖 14-1 所示，頻譜的曲線有紅色、綠色、藍色及強度等四種，設有檢核方塊，可依使用者選擇顯示各種頻譜曲線。

本程式也提供了頻譜的校正方法，若是拍攝的是螢光的光譜，就可選擇螢光燈光譜的特徵線條進行校正，藍色線條為水銀 (Hg) 所發出，其波長為 436nm，紅色線條為銻 (Eu) 所發出，波長為 611nm，如圖 14-2 所示。由程式的實做看來，頻譜的分析與繪製都可以在 20ms 內完成，速度遠比每秒 30 幅畫面的正常視訊串流之 33ms 要求，還要快上許多，確實可以達到即時呈現頻譜的要求。

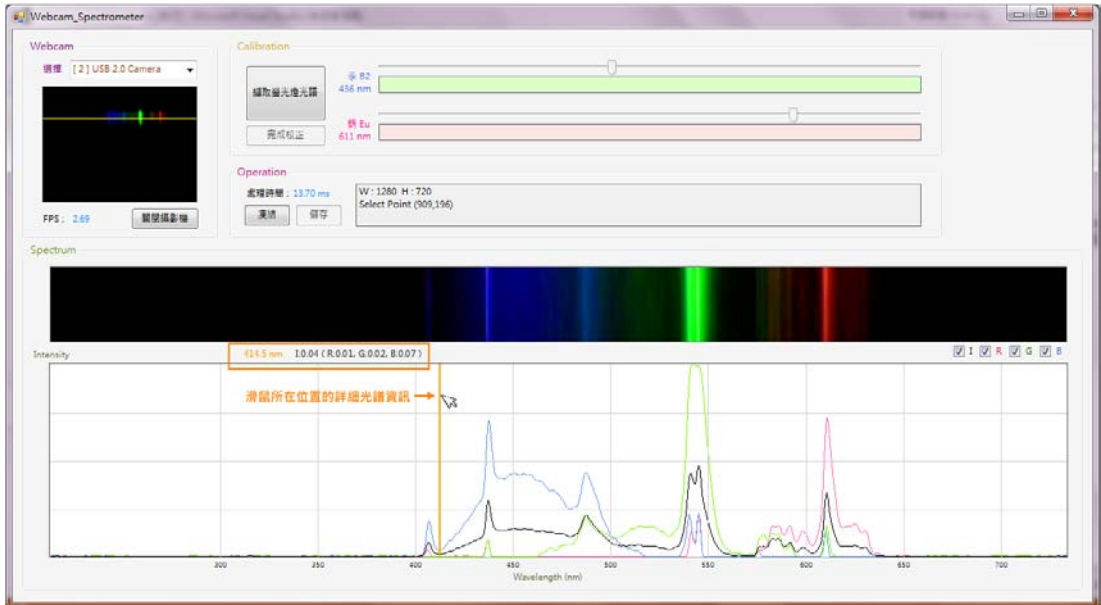


圖 14-1 即時光譜顯示

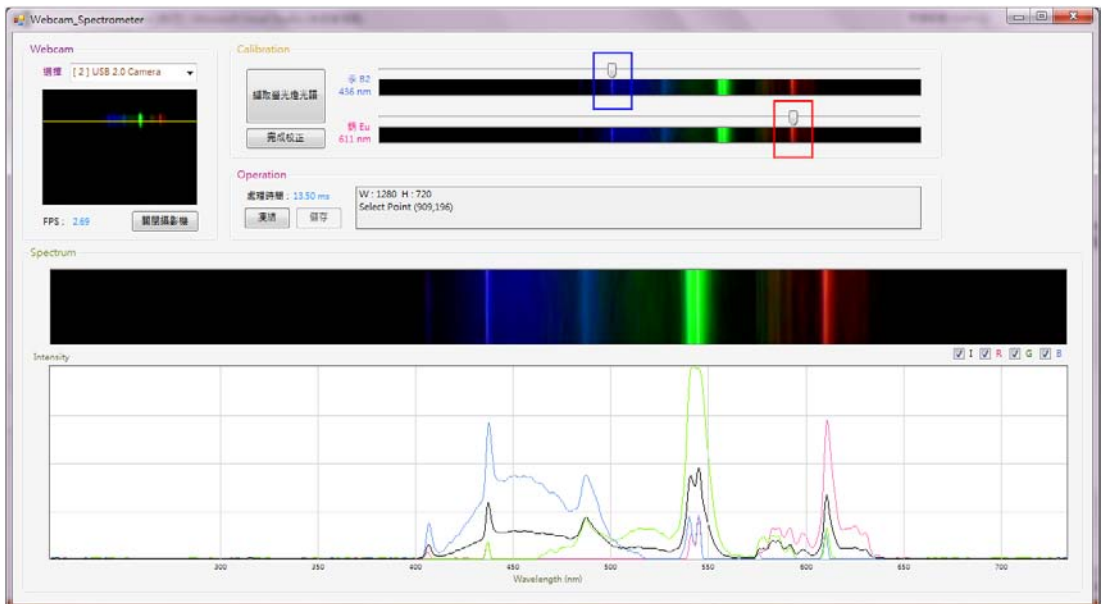


圖 14-2 拍攝螢光燈光譜進行頻譜校正

## 六、自製光譜儀的準確度測試

藉由拍攝螢光燈光譜來進行校正的光譜儀，其顯示的頻譜波長數值是否正確呢？以



擁有固定波長的綠光及紅光雷射來進行測試，裝置圖如圖 15-1 所示，當綠光及紅光雷射穿過狹縫後所產生的頻譜圖分別如圖 15-2 及圖 15-3 所示。經過影像軟體求得峰值位置及內插法的計算，得到綠光及紅光雷射的波長分別為 531.3nm 及 659.6nm，這與雷射筆的說明書上記載的 532nm 及 660nm 相當的接近，這也意謂著利用螢光燈光譜的特徵線條進行校正的方法確實可行，而且準確度也相當地不錯。



圖 15-1 自製光譜儀的準確度測試裝置圖

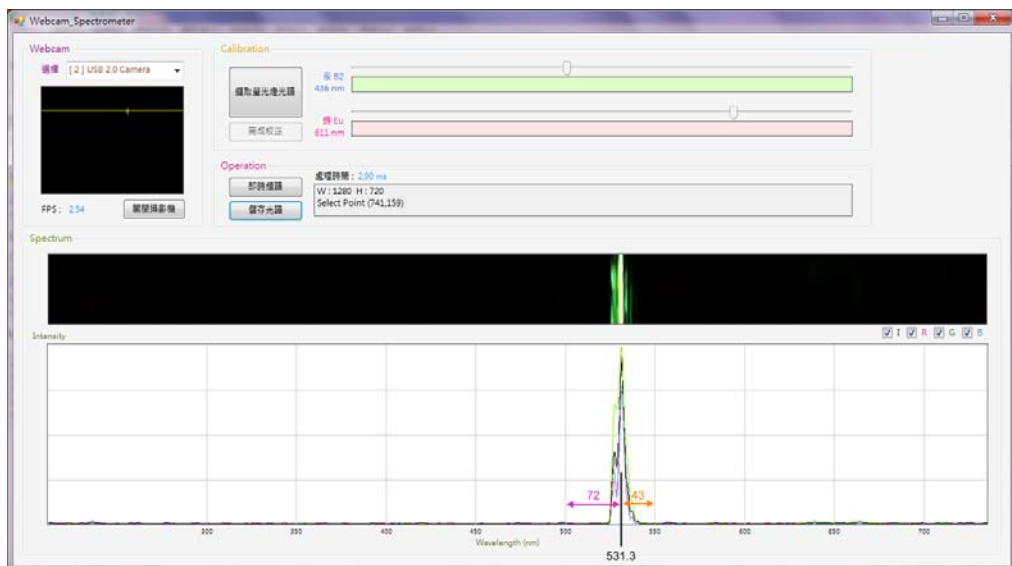


圖 15-2 綠光雷射頻譜圖中的峰值約在 531.3nm 處

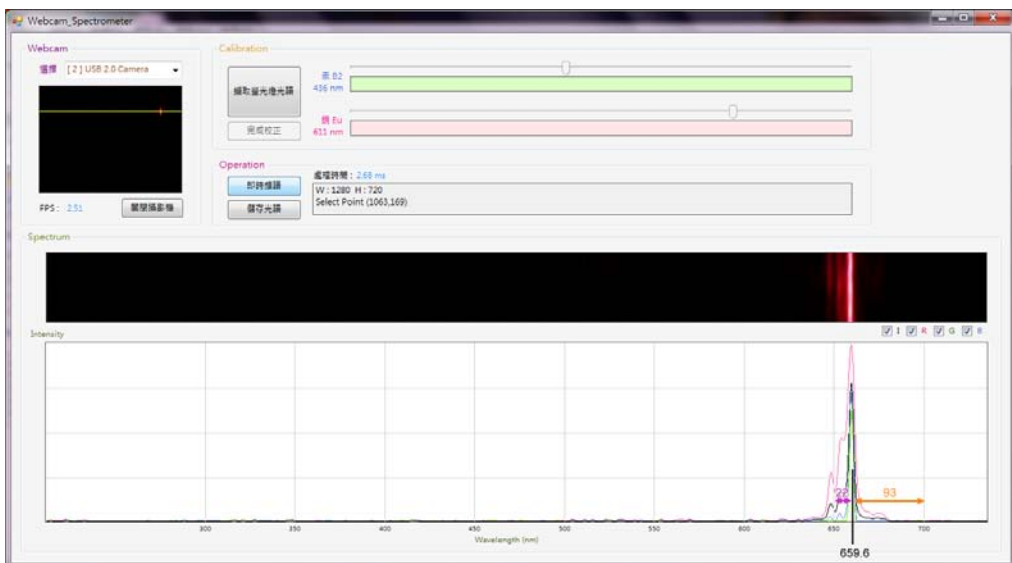


圖 15-3 紅光雷射頻譜圖中的峰值約在 659.6nm 處



## 研究結果：

### 一、觀測各式光源的頻譜

不同的光源會發出不同的光譜，生活中常見的燈泡是發出什麼樣發的光譜呢？嘗試著使用自製的光譜儀來觀測它們。觀測裝置如圖 16 所示，各式光源計有 6500K 螺旋省電燈泡（白光）、2700K 螺旋省電燈泡（黃光）、6500K 3U 省電燈泡（白光）、2700K 3U 省電燈泡（黃光）、6500K LED 燈泡（白光）、3000K LED 燈泡（黃光）、清光燈泡、靜電塗裝燈泡、Philips Argenta K 真柔燈泡，各式光源的頻譜如圖 17-1 至 17-9 所示。

觀測的結果有點令人訝異，各種波長頻譜最完整最齊全的竟然是傳統燈泡，就是耗電且又壽命不長的清光燈泡、靜電塗裝燈泡及飛利浦真柔燈泡，目前比較流行的燈泡中只有 LED 燈泡比較接近，但是頻譜中波長的分佈範圍比較狹隘。所以想要使用自製光譜儀來檢測溶液的吸收光譜時，光源的首選應該是傳統燈泡，再來才是白光 LED 燈泡，至於以螢光粉為原料製成的螢光燈泡，應該是無法達到要求的。

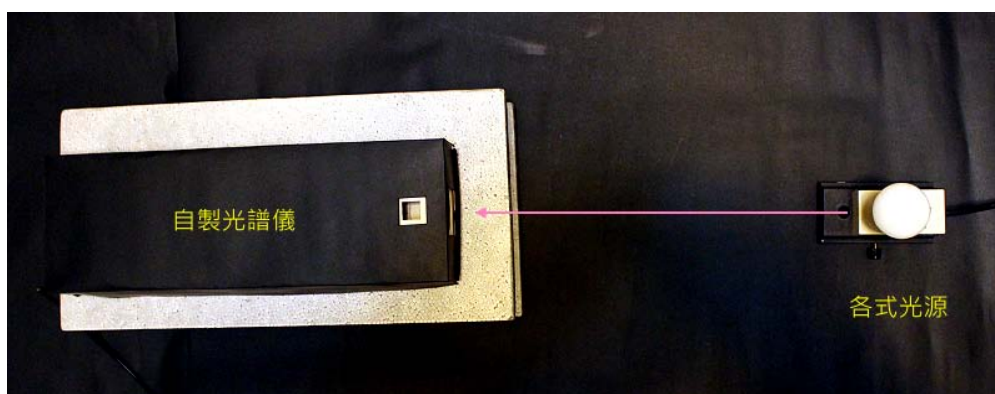


圖 16 拍攝各種光源頻譜之裝置

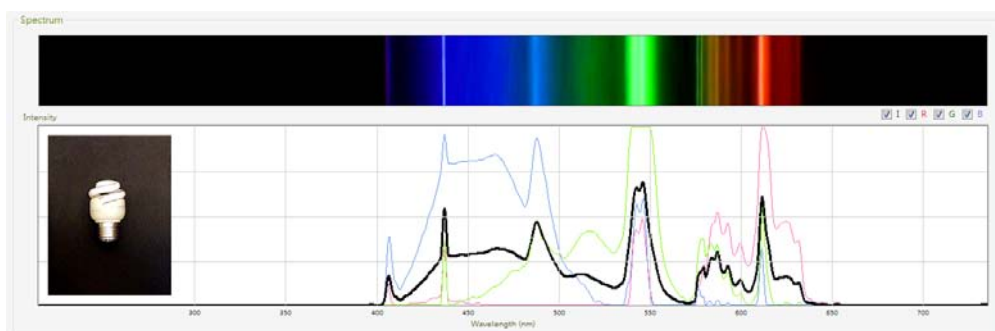


圖 17-1 5W 螺旋省電燈泡 6500K

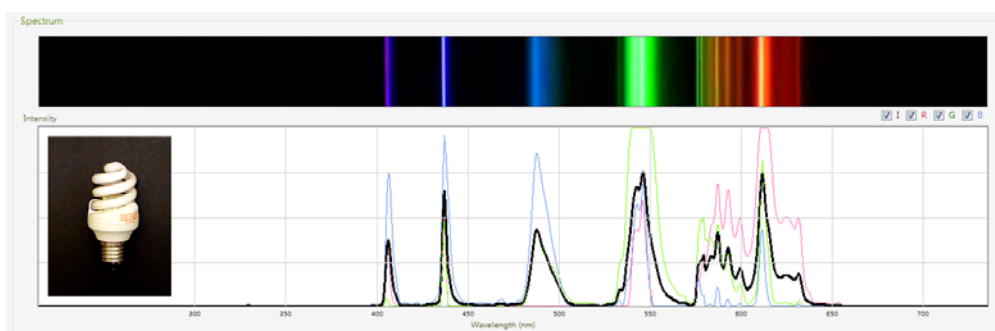


圖 17-2 11W 螺旋省電燈泡 2700K

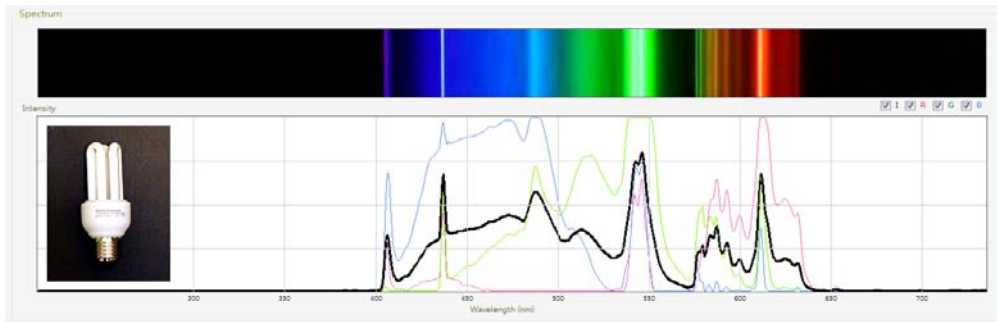


圖 17-3 11W 3U 省電燈泡 6500K

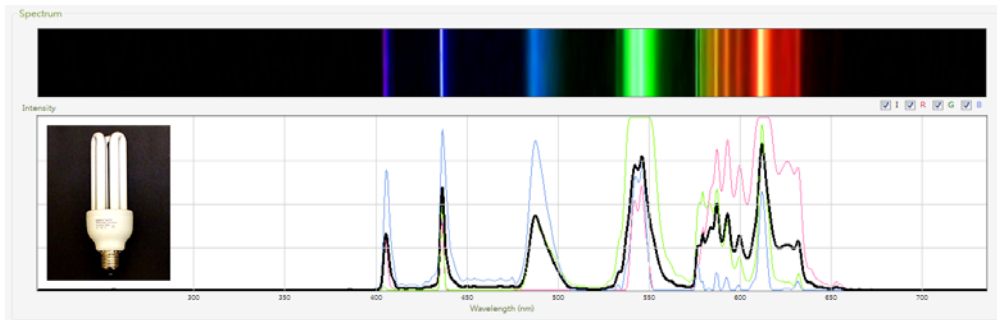


圖 17-4 18W 3U 省電燈泡 2700K

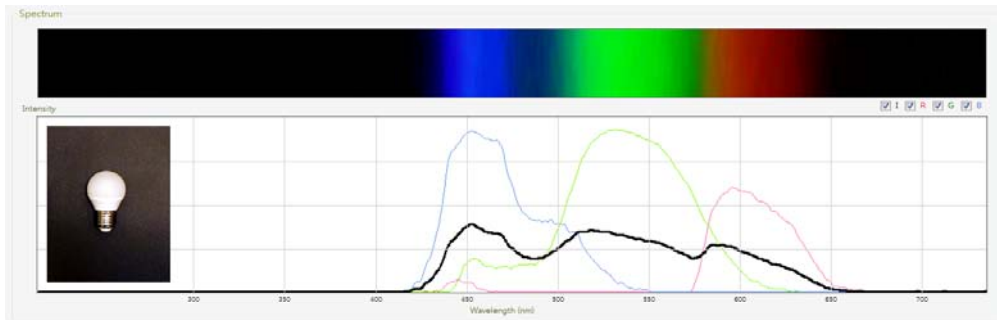


圖 17-5 3W LED 6500K

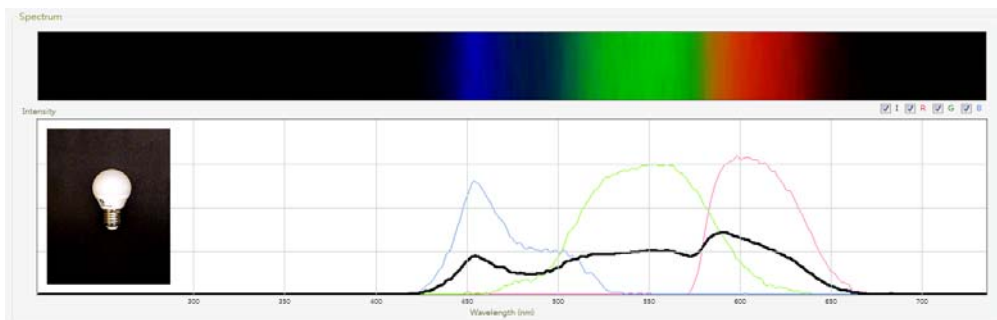


圖 17-6 3W LED 3000K

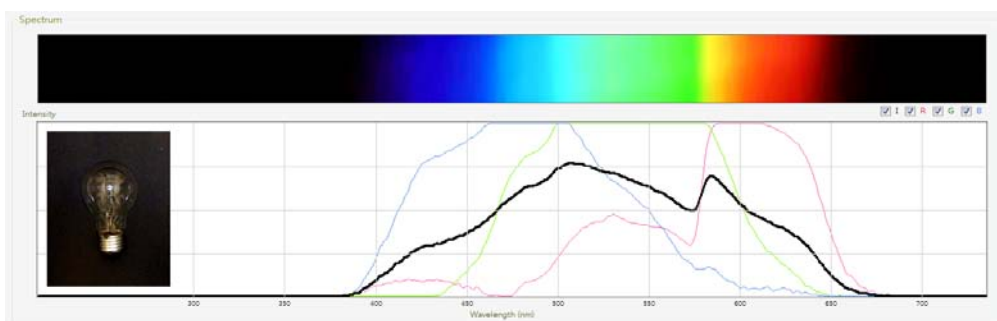


圖 17-7 60W 清光燈泡

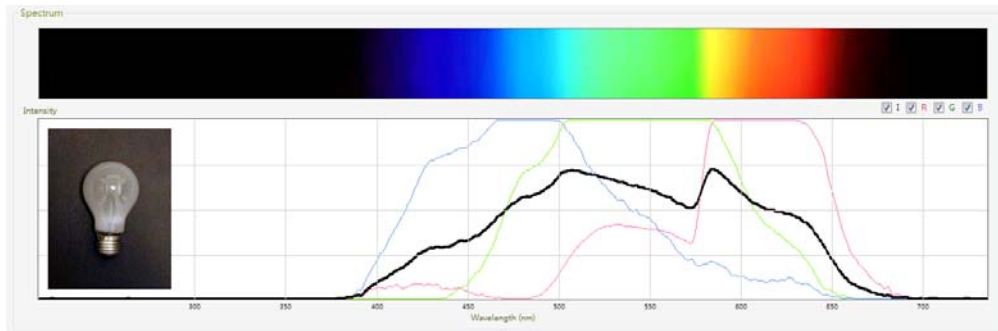


圖 17-8 100W 靜電塗裝燈泡

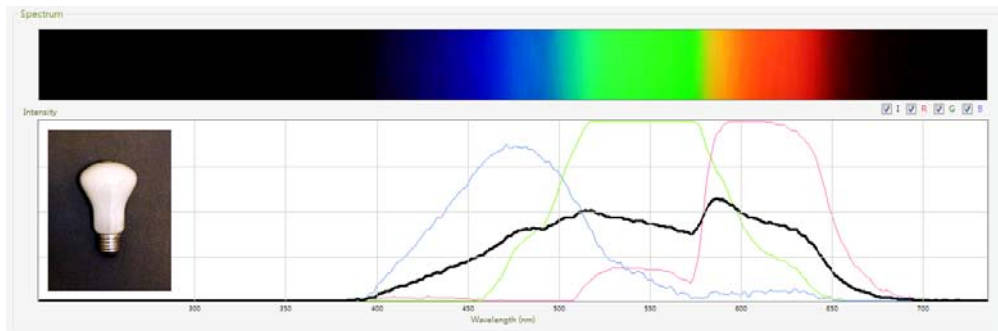


圖 17-9 100W Philips Argenta K 真柔燈泡

## 二、檢測不同濃度溶液的頻譜變化

檢測溶液的穿透率及吸光度是專業光譜儀的主要工作之一，是否自製的光譜儀也能勝任這項工作呢？以高中化學實驗室中最常見的硫酸銅溶液為例，以自製光譜儀搭配自行撰寫的程式，來檢測 0.1M、0.2M、0.3M、0.4M、0.5M 濃度溶液的穿透率及吸光度。

檢測裝置如圖 19 所示，光源則選擇飛利浦的真柔燈泡，比色管則採用一般光譜儀用的塑膠製替代品，操作的過程如下：

步驟 1：啟動程式後，使用光譜儀取得螢光燈的光譜影像後，先進行頻譜校正（圖 20-1）。

步驟 2：依圖 19 設定好檢測裝置。

步驟 3：比色管內裝蒸餾水後置入自製光譜儀的插槽中，以程式取得基準穿透光譜影像（圖 20-2）。

步驟 4：依專業光譜儀標準操作程序，依序取得不同濃度的硫酸銅溶液之穿透光譜影像（圖 20-3）。

步驟 5：按下程式使用者介面中的頻譜繪製、透光率繪製、吸光度繪製，即可取得各濃度硫酸銅溶液的頻譜（圖 20-4）、透光率（圖 20-5）、吸光度（圖 20-6）等重要曲線。

將所測得的各濃度硫酸銅溶液的吸光度與專業光譜儀所測得的數據曲線（圖 20-7）做比對，相似度還蠻高的，表示自製光譜儀尚可勝任工作。

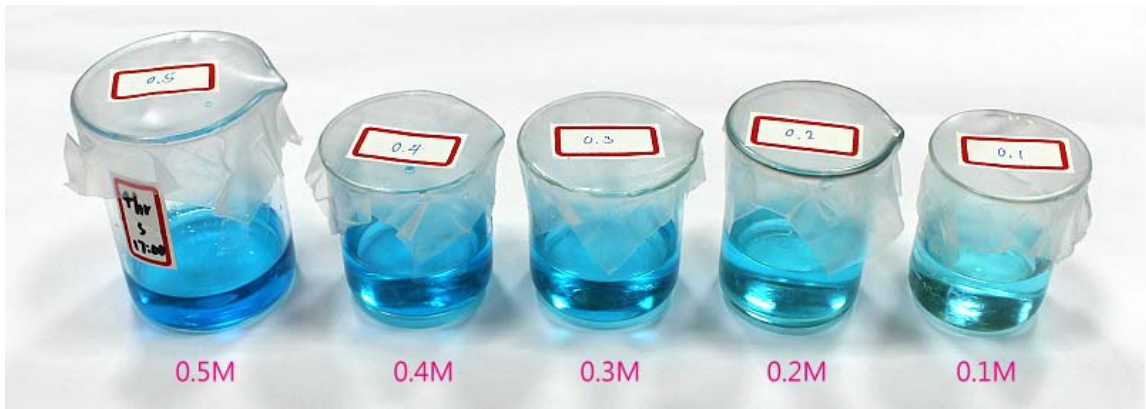


圖 18 不同濃度的硫酸銅溶液

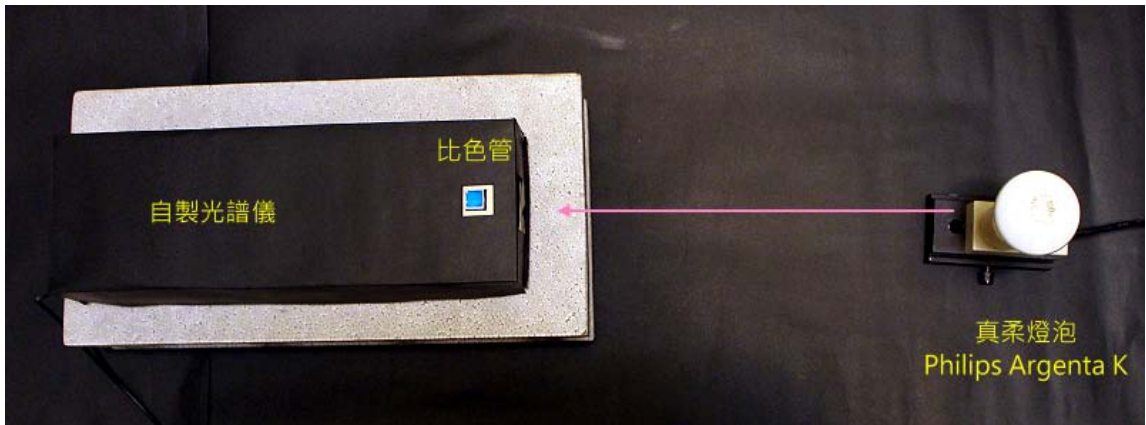


圖 19 檢測裝置

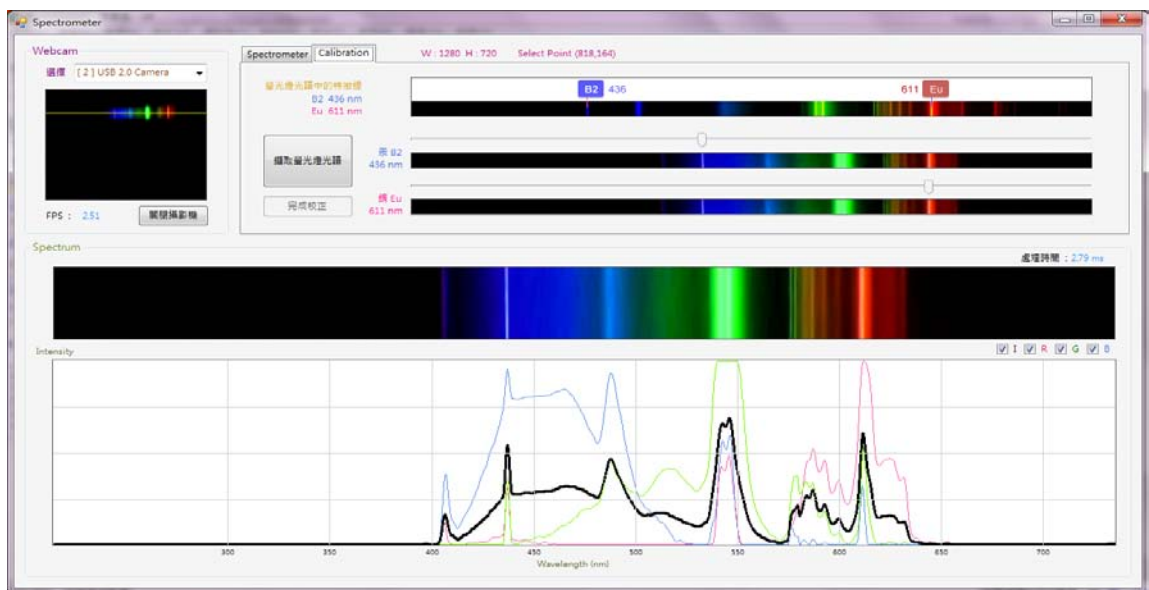


圖 20-1 利用螢光燈光譜影像進行頻譜校正



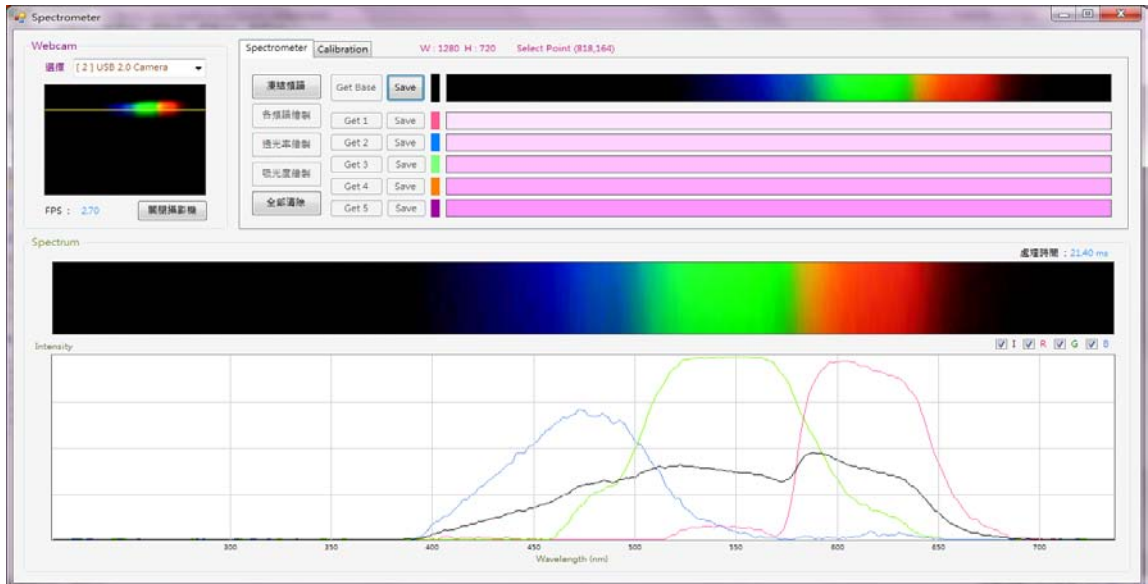


圖 20-2 比色管內裝蒸餾水當作參考溶液以取得基準穿透光譜影像

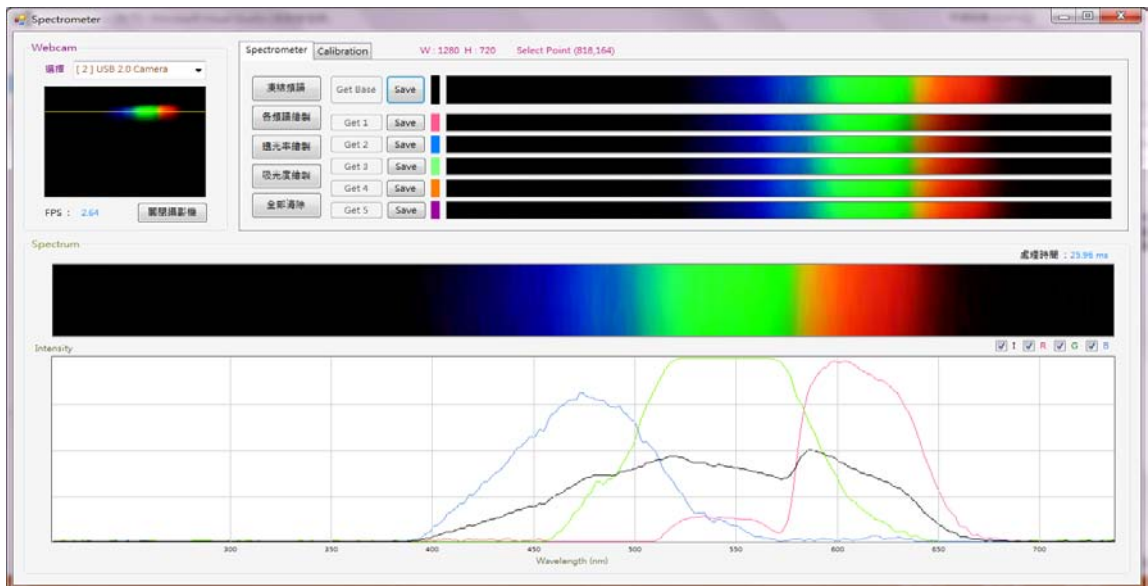


圖 20-3 依序取得不同濃度的硫酸銅溶液之穿透光譜影像

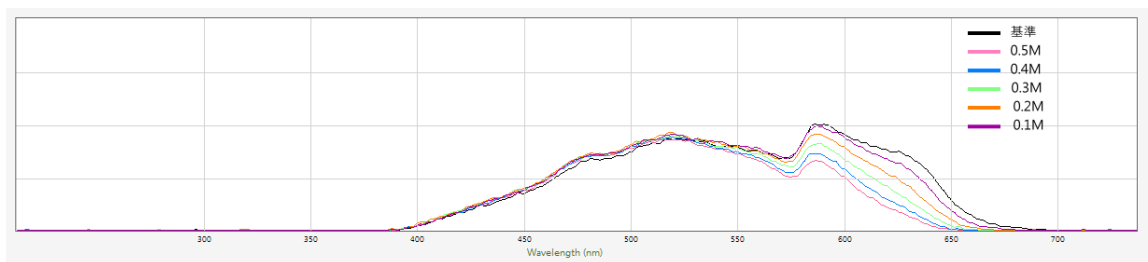


圖 20-4 繪製不同濃度硫酸銅溶液的穿透頻譜

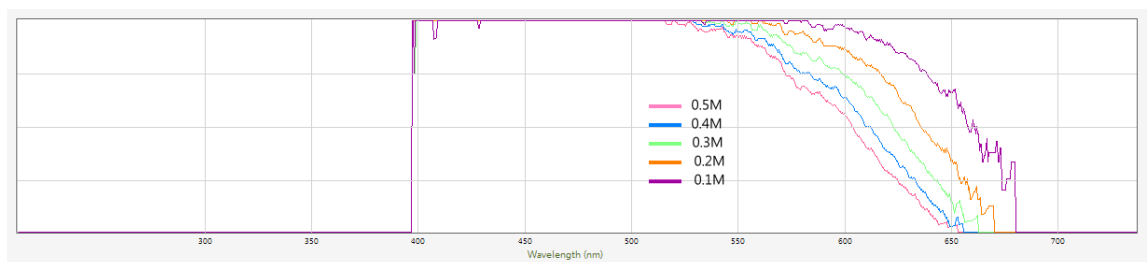


圖 20-5 繪製不同濃度硫酸銅溶液的穿透率

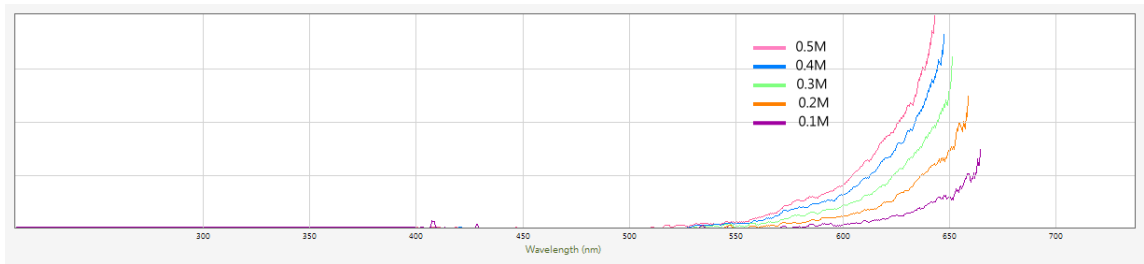


圖 20-6 繪製不同濃度硫酸銅溶液的吸光度

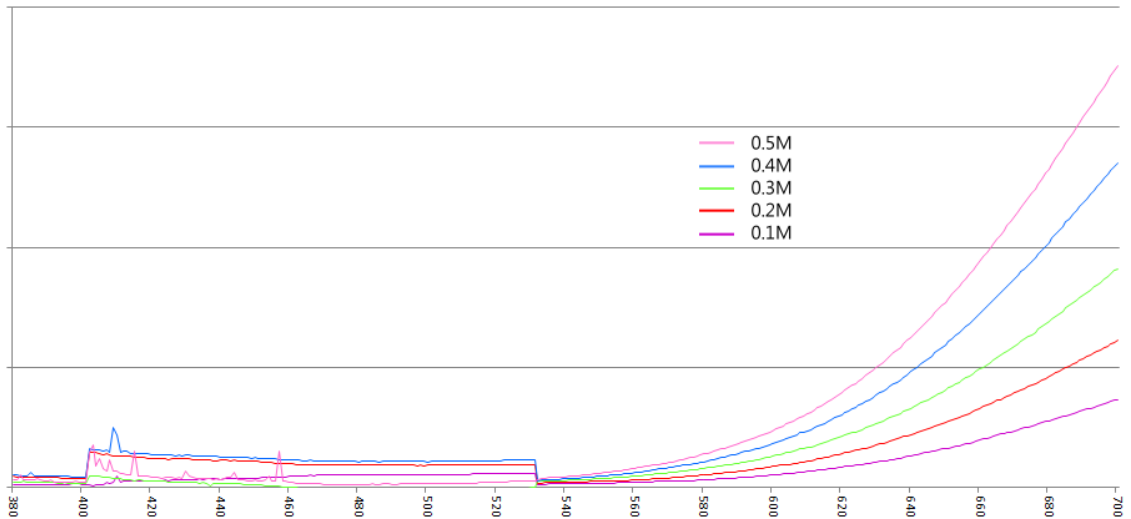


圖 20-7 專業光譜儀所測得的各濃度硫酸銅溶液吸光度 (PG Instruments Ltd. Spectrophotometer T80)

## 討論及結語：

第一次看到網路攝影機傳送過來的光譜影像時，品質糟糕的令人訝異，直覺地認為是不是那裡出錯了，先用眼睛觀看光譜，咦！光譜非常的漂亮啊，所以不是簡易的DVD光柵問題，而是在取像設備。使用入門的單眼相機 CANON EOS 100D 來拍拍看，先用智慧型模式拍攝，是比網路攝影機好，但是曝光過度了，再改用手動設定B快門模式，試了好多張，最好的影像是在光圈5，快門1/3秒狀況下拍到的，所以可以了解到，拍攝光譜，較高檔的相機會比廉價的網路攝影機效果好上許多，但與人類眼睛還是有很大的差距。雖然網路攝影機無法顯示較高品質的光譜，但是在光譜影像的即時分析處理及頻譜呈現上有相當地便捷性，藉由尋找光柵角及觀測角的最佳組合，還是可以將網路攝影機盡其可能的發揮其能力所及範圍的光譜影像之品質極限。



圖 21-1 智慧型模式拍攝的光譜



圖 21-2 手動快門光圈拍攝的光譜

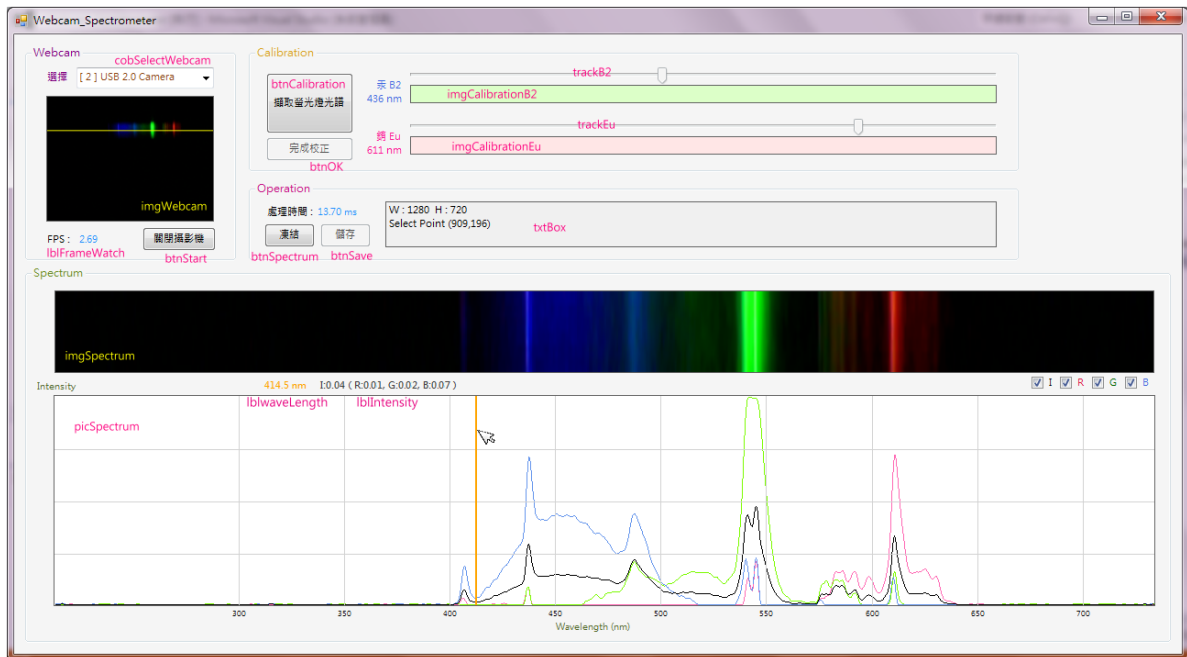


利用自製光譜儀來檢測溶液的穿透率及吸光度時，光源的選擇非常的重要，因為許多燈泡所發出的光線中並非各種波長的光都齊全（全頻譜），這樣會使得檢測溶液時欠缺部分波長的頻譜資訊，進而影響了相關數據的計算與曲線繪製。藉由自製光譜儀觀測各式光源的頻譜，發現最好的光源是傳統的燈泡及飛利浦真柔燈泡，至於目前比較常用的燈泡中，白光LED燈泡是備用的選擇。

## 參考資料：

- 一、吳詩芳、李佳杰、張哲睿、劉子裕著，“數位影像分析可見光光譜，” 中華民國第 45 屆中小學科學展覽會高中組物理科第二名作品，2005。
- 二、蘇玟瑜、林欣曄、陳盈均著，“察「焰」觀「色」-以自製分光儀檢視焰色實驗，” 中華民國第 48 屆中小學科學展覽會高中組生活與應用科學科佳作作品，2008。
- 三、葉國掄著，“解「色」－探究自製光譜儀之校正方法與應用於水溶液濃度分析，” 中華民國第 53 屆中小學科學展覽會國中組物理科佳作作品，2013。
- 四、李姍姍、吳宗諱著，“河川中的彩虹光譜－製作行動光譜儀分析河川水色，” 中華民國第 54 屆中小學科學展覽會國小組地球科學科佳作作品，2014。
- 五、傅學海教授（國立台灣師範大學地球科學系）著，“用自製光譜儀觀察各型光譜，”  
<http://www3.cns.hk.edu.tw/~eao/aveplan/99-2/Documents/ForTeachers/99-2-t5.pdf>。
- 六、[http://a-chien.blogspot.tw/2014/04/blog-post\\_29.html](http://a-chien.blogspot.tw/2014/04/blog-post_29.html), 阿簡生物筆記（部落格）。
- 七、游大立老師（國立彰化高級中學）著，“簡易摺紙光譜儀，”  
[http://www.sec.ntnu.edu.tw/Monthly/95\(286-295\)/295-pdf/04.pdf](http://www.sec.ntnu.edu.tw/Monthly/95(286-295)/295-pdf/04.pdf)。
- 八、<https://zh.wikipedia.org/zh-tw/光譜儀>, 維基百科。
- 九、<https://zh.wikipedia.org/zh-tw/繞射光柵>, 維基百科。
- 十、<https://zh.wikipedia.org/zh-tw/螢光燈>, 維基百科。
- 十一、資訊教育研究室著，“Visual Basic 2012 從零開始，” 博碩文化，2013。
- 十二、[http://www.emgu.com/wiki/index.php/Main\\_Page](http://www.emgu.com/wiki/index.php/Main_Page), Emgu CV : OpenCV in .NET。
- 十三、<http://yy-programer.blogspot.tw/>, yy's Program。
- 十四、<https://spectralworkbench.org/>, Public Lab。
- 十五、<https://msdn.microsoft.com/>, Microsoft Developer Network 微軟開發者網路。

## 附錄：即時光譜顯示程式



```
Imports System.Collections.Generic
Imports System.ComponentModel
Imports System.Data
Imports System.Drawing
Imports System.Drawing.Imaging
Imports System.Text
Imports System.IO
Imports System.Windows.Forms
Imports System.Threading
```

'EMGU

```
Imports Emgu.CV
Imports Emgu.CV.Structure
Imports Emgu.Util
Imports Emgu.CV.CvEnum
```

'DirecstShow

```
Imports DirectShowLib
```

'Structure VideoDevice

```
Structure Video_Device
    Public Device_Name As String
    Public Device_ID As Integer
    Public Sub New(ID As Integer, Name As String)
        Device_ID = ID
        Device_Name = Name
    End Sub
    Public Overrides Function ToString() As String
        Return [String].Format("[ {0} ] {1}", Device_ID, Device_Name)
    End Function
End Structure
```

Public Class Webcam\_Spectrometer

'定義常數

```
Const enable As Boolean = True
Const disable As Boolean = False
Const scanLine_heightScale = 1 / 200 '光譜掃描線的粗細
Const SpectrumHeight = 12 '光譜分析影像的高度
Const smoothWindow As Integer = 5 '光譜曲線平滑視窗大小
Const mercuryB2length = 435.833
Const mercuryG2length = 546.074
```

```

Const europiumEulength = 611
Const webcamWidth = 1280
Const webcamHeight = 720

```

'定義變數

```

Dim scanLinePosition As Double '光譜掃描線位置
Dim mercuryB2, Eu As Integer '著名光譜在影像中的位置
Dim webCams As Video_Device() 'List containing all the camera available
Dim CameraDevice As Integer '全部 webcam 列表
Dim _capture As Capture
Dim _captureInProgress As Boolean
Dim stopFrameWatch, stopSpectrumWatch As New Stopwatch
Dim inFrame, outFrame As Image(Of Bgr, Byte) 'webcam 影像
Dim tmpImg, spectrumImg As Image(Of Bgr, Byte)
Dim freeze As Boolean
Structure spectrumPoint '繪製頻譜用的點
    Public _waveLength As Double
    Public _value As Double
End Structure
Dim _Rpts(), _Gpts(), _Bpts(), _Ipts() As spectrumPoint
Dim mouseImg As Image(Of Bgr, Byte) 'mouseMove 用影像

```

'程式開始執行

```

Private Sub Webcam_Spectrometer_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    '讀取檔案參數 scanLinePosition, mercuryB2, Eu
    Dim sr As StreamReader = New StreamReader(CurDir() & "\parameter.txt")
    Dim no As Integer = 1
    Do While Not sr.EndOfStream
        Dim parameter As String = sr.ReadLine()
        Select Case no
            Case 1
                scanLinePosition = CDb(Trim(parameter))
            Case 2
                mercuryB2 = CInt(Trim(parameter))
            Case 3
                Eu = CInt(Trim(parameter))
        End Select
        no = no + 1
    Loop
    sr.Close()
    trackB2.Value = mercuryB2
    trackEu.Value = Eu
    '---- 前置處理
    Me.Location = New Point(3, 3) '視窗開啟位置
    freeze = False
    Update_btn(disable, "擷取螢光燈光譜", btnCalibration)
    cobSelectWebcam.Items.Clear()
    Update_txb(txtBox, "")
    picSpectrum.Width = imgSpectrum.Width + 2 '光譜曲線視窗略大於光譜影像
    Dim _Cameras As DsDevice() = DsDevice.GetDevicesOfCat(FilterCategory.VideoInputDevice)
    webCams = New Video_Device(_Cameras.Length - 1) {}
    For i As Integer = 0 To _Cameras.Length - 1 '將所有 webcam 列於選單中
        webCams(i) = New Video_Device(i + 1, _Cameras(i).Name)
        cobSelectWebcam.Items.Add(webCams(i).ToString())
    Next
    If cobSelectWebcam.Items.Count > 0 Then
        cobSelectWebcam.SelectedIndex = 0
        cobSelectWebcam.Enabled = True
        Update_btn(enable, "開啟攝影機", btnStart)
        _captureInProgress = False
    Else
        Update_txb(txtBox, "沒有連接任何視訊設備 !")
    End If
End Sub

```

'關閉程式視窗

```

Private Sub Webcam_Spectrometer_FormClosed(sender As Object, e As FormClosedEventArgs) Handles MyBase.FormClosed
    If _capture IsNot Nothing Then
        If _capture.GrabProcessState = System.Threading.ThreadState.Running Then
            _capture.Pause()
            _capture.Stop()
            RemoveHandler _capture.ImageGrabbed, AddressOf GetFrame
        End If
        _capture.Dispose()
    End If
End Sub

```

### 開啟/關閉 Webcam

```
Private Sub btnStart_Click(sender As Object, e As EventArgs) Handles btnStart.Click
    If _capture Is Nothing Then
        SetupCapture(cobSelectWebcam.SelectedIndex) ' _capture 與選取的 webcam 連接
        btnStart_Click(Nothing, Nothing) 'Recall this method to start camera
    Else
        If Not _captureInProgress Then '按下開啟視訊按鈕
            If cobSelectWebcam.SelectedIndex <> CameraDevice Then ' _capture 與變更選取後的 webcam 連接
                SetupCapture(cobSelectWebcam.SelectedIndex)
            End If
            Update_btn(enable, "關閉攝影機", btnStart) '按鈕變為關閉視訊
            _capture.Start() 'Start the capture
            _captureInProgress = True '開始取像
            Update_btn(enable, "擷取螢光燈光譜", btnCalibration)
            Update_btn(disable, "完成校正", btnOK)
            Update_btn(disable, "儲存光譜", btnSave)
            Update_btn(enable, "凍結頻譜", btnSpectrum)
            Update_btn(disable, "儲存光譜", btnSave)
            freeze = False
            Dim tmpWidth As Integer =
                _capture.GetCaptureProperty(Emgu.CV.CvEnum.CAP_PROP_CV_CAP_PROP_FRAME_WIDTH)
            Dim tmpHeight As Integer =
                _capture.GetCaptureProperty(Emgu.CV.CvEnum.CAP_PROP_CV_CAP_PROP_FRAME_HEIGHT)
            Update_txb(txtBox, "W : " + tmpWidth.ToString + "   H : " + tmpHeight.ToString + vbCrLf + "Select Point (" +
                CStr(CInt(tmpWidth / 2)) + ", " + CStr(CInt(scanLinePosition * tmpHeight)) + ")")
        Else '按下關閉視訊按鈕
            Update_btn(enable, "開啟攝影機", btnStart) '按鈕變更為開啟視訊
            _capture.Pause() 'pause the capture
            _captureInProgress = False
            freeze = True
            Update_btn(disable, "凍結頻譜", btnSpectrum)
            Update_btn(enable, "儲存光譜", btnSave)
        End If
    End If
End Sub
```

---

### 影格光譜分析統計繪製核心

---

### '1 webCam 影像上繪製掃描線

```
Private Function GenerateScanLineOutFrame(ByVal inImg As Image(Of Bgr, Byte), ByVal scanLine_position As Double) As
    Image(Of Bgr, Byte)
    '產生 outImg
    Dim outImg As Image(Of Bgr, Byte) = New Image(Of Bgr, Byte)(inImg.Width, inImg.Height)
    CvInvoke.cvCopy(inImg, outImg, System.IntPtr.Zero)
    '繪製光譜橫線位置
    Dim yellowLine As New LineSegment2D(New Point(0, CInt(scanLine_position * outImg.Height)), New Point(outImg.Width - 1,
    CInt(scanLine_position * outImg.Height)))
    Dim lineHeight As Integer = CInt(Math.Round(outImg.Height * scanLine_heightScale))
    lineHeight = Math.Max(1, lineHeight)
    outImg.Draw(yellowLine, New Bgr(Color.Yellow), lineHeight)
    Return outImg
End Function
```

### '2 產生光譜影像 spectrumImg '1280x12

```
Private Function GenerateSpectrumImg(ByVal inImg As Image(Of Bgr, Byte), ByVal scanLine_position As Double, ByVal
    img_height As Integer) As Image(Of Bgr, Byte)
    '產生 outImg
    Dim outImg As Image(Of Bgr, Byte) = New Image(Of Bgr, Byte)(inImg.Width, img_height)
    inImg.ROI = New Rectangle(0, CInt(Math.Round(scanLine_position * inImg.Height - img_height / 2)), inImg.Width,
    img_height)
    inImg.CopyTo(outImg)
    inImg.ROI = New Rectangle()
    Return outImg
End Function
```

### '3 分析統計 spectrumImg 各色頻強度

```
Private Sub Analysis_Channel_Intensity(ByVal inImg As Bitmap, ByRef R() As Long, ByRef G() As Long, ByRef B() As Long,
    ByRef intensity() As Long, ByVal smooth As Integer)
    Dim row, col As Integer
    Dim tmpR(inImg.Width), tmpG(inImg.Width), tmpB(inImg.Width), tmpI(inImg.Width) As Double
    Dim inData As BitmapData = inImg.LockBits(New Rectangle(0, 0, inImg.Width, inImg.Height), ImageLockMode.ReadOnly,
```

```

        PixelFormat.Format24bppRgb)
Dim ptrIn As IntPtr = inData.Scan0
Dim bytesIn As Long
Dim rgbvalIn() As Byte
bytesIn = inData.Stride * inImg.Height
ReDim rgbvalIn(bytesIn - 1)
System.Runtime.InteropServices.Marshal.Copy(ptrIn, rgbvalIn, 0, bytesIn)
'計算光譜影像中各 column 的顏色
For i As Integer = 1 To inImg.Width
    R(i) = 0 : G(i) = 0
    B(i) = 0 : intensity(i) = 0
Next
For row = 1 To inImg.Height
    For col = 1 To inImg.Width
        If (row <> 1) And (row <> inImg.Height) Then
            R(col) = R(col) + rgbvalIn(inData.Stride * (row - 1) + (col - 1) * 3 + 2)
            G(col) = G(col) + rgbvalIn(inData.Stride * (row - 1) + (col - 1) * 3 + 1)
            B(col) = B(col) + rgbvalIn(inData.Stride * (row - 1) + (col - 1) * 3 + 0)
            intensity(col) = intensity(col) + rgbvalIn(inData.Stride * (row - 1) + (col - 1) * 3 + 2) + rgbvalIn(inData.Stride *
                (row - 1) + (col - 1) * 3 + 1) + rgbvalIn(inData.Stride * (row - 1) + (col - 1) * 3 + 0)
        End If
    Next
Next
inImg.UnlockBits(inData) ' Unlock the bits.
For i As Integer = 1 To inImg.Width
    R(i) = CInt(Math.Round(R(i) / (inImg.Height - 2)))
    G(i) = CInt(Math.Round(G(i) / (inImg.Height - 2)))
    B(i) = CInt(Math.Round(B(i) / (inImg.Height - 2)))
    intensity(i) = CInt(Math.Round(intensity(i) / 3 / (inImg.Height - 2)))
Next
'smooth 曲線
For i As Integer = 1 To inImg.Width
    If i > (smooth \ 2) And i < inImg.Width - (smooth \ 2) Then
        tmpR(i) = 0 : tmpG(i) = 0 : tmpB(i) = 0 : tmpI(i) = 0
        For j = -1 * (smooth \ 2) To (smooth \ 2)
            tmpR(i) = tmpR(i) + R(i + j) / smooth
            tmpG(i) = tmpG(i) + G(i + j) / smooth
            tmpB(i) = tmpB(i) + B(i + j) / smooth
            tmpI(i) = tmpI(i) + intensity(i + j) / smooth
        Next
    Else
        tmpR(i) = R(i) : tmpG(i) = G(i) : tmpB(i) = B(i) : tmpI(i) = intensity(i)
    End If
Next
For i As Integer = 1 To inImg.Width
    R(i) = CInt(tmpR(i)) : G(i) = CInt(tmpG(i)) : B(i) = CInt(tmpB(i)) : intensity(i) = CInt(tmpI(i))
Next
End Sub

```

#### 4 繪製光譜

```

Private Function drawSpectrum(ByVal _width As Integer, ByVal _height As Integer, ByVal R() As Long, ByVal G() As Long, ByVal
    B() As Long, ByVal Intensity() As Long) As Image(Of Bgr, Byte)
Dim outImg As New Image(Of Bgr, Byte)(_width, _height, New Bgr(Color.White))
outImg.CopyBlank()
Dim a As Integer = UBound(R)
Dim x1, y1, x2, y2 As Long
Dim colorLine As LineSegment2D
Dim intensityScale As Double = (_height - 4) / 255
For i As Integer = 1 To UBound(Intensity)
    _Rpts(i)._value = R(i) / 255
    _Gpts(i)._value = G(i) / 255
    _Bpts(i)._value = B(i) / 255
    _Ipts(i)._value = Intensity(i) / 255
    If i <> 1 Then
        x1 = i - 1 : x2 = i
        If cbR.Checked = True Then
            y1 = CInt(Math.Round(outImg.Height - R(i - 1) * intensityScale)) - 3 : y2 = CInt(Math.Round(outImg.Height - R(i)
                * intensityScale)) - 3
            colorLine = New LineSegment2D(New Point(x1, y1), New Point(x2, y2))
            outImg.Draw(colorLine, New Bgr(Color.HotPink), 1)
        End If
        If cbG.Checked = True Then
            y1 = CInt(Math.Round(outImg.Height - G(i - 1) * intensityScale)) - 3 : y2 = CInt(Math.Round(outImg.Height - G(i)
                * intensityScale)) - 3
            colorLine = New LineSegment2D(New Point(x1, y1), New Point(x2, y2))
            outImg.Draw(colorLine, New Bgr(Color.LawnGreen), 1)
        End If
        If cbB.Checked = True Then

```

```

        y1 = Cint(Math.Round(outImg.Height - B(i - 1) * intensityScale)) - 3 : y2 = Cint(Math.Round(outImg.Height - B(i)
        * intensityScale)) - 3 * B
        colorLine = New LineSegment2D(New Point(x1, y1), New Point(x2, y2))
        outImg.Draw(colorLine, New Bgr(Color.CornflowerBlue), 1)
    End If
    If cbI.Checked = True Then
        y1 = Cint(Math.Round(outImg.Height - Intensity(i - 1) * intensityScale)) - 3 : y2 =
        Cint(Math.Round(outImg.Height - Intensity(i) * intensityScale)) - 3 * Intensity
        colorLine = New LineSegment2D(New Point(x1, y1), New Point(x2, y2))
        outImg.Draw(colorLine, New Bgr(Color.Black), 2)
    End If
End If
Next
Return outImg
End Function

```

'\_capture.ImageGrabbed 取得 webcam 影像

Private Sub GetFrame(sender As Object, arg As EventArgs)

```

    Try
        stopFrameWatch.Reset()
        stopFrameWatch.Start()
        inFrame = _capture.RetrieveBgrFrame()
        stopFrameWatch.Stop()
        Update_lbl(lblFrameWatch, String.Format("{0:n2}", 1000 / stopFrameWatch.Elapsed.TotalMilliseconds),
            lblFrameWatch.Location)
        stopSpectrumWatch.Reset()
        stopSpectrumWatch.Start()
        '1 webcam 影像上繪製掃描線
        outFrame = GenerateScanLineOutFrame(inFrame, scanLinePosition)
        DisplayImage(outFrame, imgWebcam) 'outFrame 填入 imgWebcam
        If freeze = True Then
            stopSpectrumWatch.Stop()
            Update_lbl(lblSpectrumWatch, String.Format("{0:n2}", stopSpectrumWatch.Elapsed.TotalMilliseconds) + " ms",
                lblSpectrumWatch.Location)
        End If
        Exit Sub
    End If
    '2 產生光譜影像 spectrumImg 1200x12
    spectrumImg = GenerateSpectrumImg(inFrame, scanLinePosition, SpectrumHeight)
    DisplayImage(spectrumImg.Resize(imgSpectrum.Width, imgSpectrum.Height, INTER.CV_INTER_LINEAR, False),
        imgSpectrum) 'spectrumImg 填入 imgSpectrum 1200x90
    '3 分析統計 spectrumImg 各色頻強度 1200x12
    Dim analysis_spectrumImg As Bitmap = spectrumImg.Resize(imgSpectrum.Width, SpectrumHeight,
        INTER.CV_INTER_LINEAR, False).ToBitmap() '將 spectrumImg 轉成 bitmap 1200x12
    Dim R(analysis_spectrumImg.Width), G(analysis_spectrumImg.Width), B(analysis_spectrumImg.Width),
        Intensity(analysis_spectrumImg.Width) As Long
    Analysis_Channel_Intensity(analysis_spectrumImg, R, G, B, Intensity, smoothWindow)
    '4 繪製光譜 1202x230
    Dim outImg As Image(Of Bgr, Byte)
    'Dim outImg As New Image(Of Bgr, Byte)(picSpectrum.Width, picSpectrum.Height, New Bgr(Color.White))
    outImg = drawSpectrum(picSpectrum.Width, picSpectrum.Height, R, G, B, Intensity)
    '5 計算光譜波長位置
    Dim waveLengthPosition(13) As Double '光譜波長 300nm~900nm ( 250 + 50*i nm )
    Dim displayScale As Double = imgSpectrum.Width / imgCalibrationB2.Width '1200 / 640
    Dim pixelPERnm As Double = (Eu - mercuryB2) / (europiumEulength - mercuryB2length)
    Dim position600nm As Double = mercuryB2 + (600 - mercuryB2length) * pixelPERnm 'in 640 pixel imgCalibration
    ImageBox
    Dim displayPosition600nm As Double = position600nm * displayScale 'in 1200 pixel imgSpectrum ImageBox
    For i As Integer = 1 To 13
        waveLengthPosition(i) = displayPosition600nm + (i - 7) * 50 * pixelPERnm * displayScale 'in 1200 pixel
        imgSpectrum ImageBox
    Next
    '6 繪製波長座標線於 outImg 上
    For i As Integer = 1 To UBound(waveLengthPosition)
        Dim x1 As Long = waveLengthPosition(i) : Dim x2 As Long = waveLengthPosition(i)
        Dim y1 As Long = 0 : Dim y2 As Long = outImg.Height
        Dim colorLine As LineSegment2D = New LineSegment2D(New Point(x1, y1), New Point(x2, y2))
        outImg.Draw(colorLine, New Bgr(Color.LightGray), 1)
    Next
    '繪製 50%Intensity 線
    Dim ix1 As Long = 0 : Dim ix2 As Long = picSpectrum.Width
    Dim iy1 As Long = picSpectrum.Height * 0.5 : Dim iy2 As Long = picSpectrum.Height * 0.5
    Dim icolorLine As LineSegment2D = New LineSegment2D(New Point(ix1, iy1), New Point(ix2, iy2))
    outImg.Draw(icolorLine, New Bgr(Color.LightGray), 1)
    iy1 = picSpectrum.Height * 0.25 : iy2 = picSpectrum.Height * 0.25
    icolorLine = New LineSegment2D(New Point(ix1, iy1), New Point(ix2, iy2))
    outImg.Draw(icolorLine, New Bgr(Color.LightGray), 1)

```



```

iy1 = picSpectrum.Height * 0.75 : iy2 = picSpectrum.Height * 0.75
icolorLine = New LineSegment2D(New Point(ix1, iy1), New Point(ix2, iy2))
outImg.Draw(icolorLine, New Bgr(Color.LightGray), 1)
7 顯示光譜
DisplayImage(outImg, picSpectrum)
mouseImg = outImg.Clone
8 波長標籤定位
Dim shiftPixel As Integer = picSpectrum.Location.X
For i As Integer = 1 To UBound(waveLengthPosition)
    If i = 1 Then Update_lbl(lbl300, "300", New Point(shiftPixel - 8 + waveLengthPosition(i), lbl300.Location.Y))
    If i = 2 Then Update_lbl(lbl350, "350", New Point(shiftPixel - 8 + waveLengthPosition(i), lbl350.Location.Y))
    If i = 3 Then Update_lbl(lbl400, "400", New Point(shiftPixel - 8 + waveLengthPosition(i), lbl400.Location.Y))
    If i = 4 Then Update_lbl(lbl450, "450", New Point(shiftPixel - 8 + waveLengthPosition(i), lbl450.Location.Y))
    If i = 5 Then Update_lbl(lbl500, "500", New Point(shiftPixel - 8 + waveLengthPosition(i), lbl500.Location.Y))
    If i = 6 Then Update_lbl(lbl550, "550", New Point(shiftPixel - 8 + waveLengthPosition(i), lbl550.Location.Y))
    If i = 7 Then Update_lbl(lbl600, "600", New Point(shiftPixel - 8 + waveLengthPosition(i), lbl600.Location.Y))
    If i = 8 Then Update_lbl(lbl650, "650", New Point(shiftPixel - 8 + waveLengthPosition(i), lbl650.Location.Y))
    If i = 9 Then Update_lbl(lbl700, "700", New Point(shiftPixel - 8 + waveLengthPosition(i), lbl700.Location.Y))
    If i = 10 Then Update_lbl(lbl750, "750", New Point(shiftPixel - 8 + waveLengthPosition(i), lbl750.Location.Y))
    If i = 11 Then Update_lbl(lbl800, "800", New Point(shiftPixel - 8 + waveLengthPosition(i), lbl800.Location.Y))
    If i = 12 Then Update_lbl(lbl850, "850", New Point(shiftPixel - 8 + waveLengthPosition(i), lbl850.Location.Y))
    If i = 13 Then Update_lbl(lbl900, "900", New Point(shiftPixel - 8 + waveLengthPosition(i), lbl900.Location.Y))
Next
stopSpectrumWatch.Stop()
Update_lbl(lblSpectrumWatch, String.Format("{0:n2}", stopSpectrumWatch.Elapsed.TotalMilliseconds) + " ms",
    lblSpectrumWatch.Location)
Catch ex As Exception
    MessageBox.Show(ex.Message)
End Try
End Sub

```

#### 滑鼠所在位置的波長與強度

```

Private Sub picSpectrum_MouseMove(sender As Object, e As MouseEventArgs) Handles picSpectrum.MouseMove
    Dim mouseX As Integer = e.X
    Dim mouseY As Integer = e.Y
    Dim indexX As Integer = mouseX
    If indexX > 0 And indexX < picSpectrum.Width Then
        Update_lbl(lblwaveLength, String.Format("{0:n1}", _Ipts(indexX)._waveLength) + " nm", New
            Point(lblwaveLength.Location.X, lblwaveLength.Location.Y))
        Update_lbl(lblIntensity, "I:" + String.Format("{0:n2}", _Ipts(indexX)._value) + " (R:" + String.Format("{0:n2}",
            _Rpts(indexX)._value) + ", G:" + String.Format("{0:n2}", _Gpts(indexX)._value) + ", B:" + String.Format("{0:n2}",
            _Bpts(indexX)._value) + ")") , New Point(lblIntensity.Location.X, lblIntensity.Location.Y))
    End If
    Dim mouseLine As LineSegment2D
    Dim x1, y1, x2, y2 As Long
    Dim mouseOutImg As Image(Of Bgr, Byte) = mouseImg.Clone
    x1 = mouseX : x2 = mouseX
    y1 = 0 : y2 = mouseOutImg.Height
    mouseLine = New LineSegment2D(New Point(x1, y1), New Point(x2, y2))
    mouseOutImg.Draw(mouseLine, New Bgr(Color.Orange), 1)
    DisplayImage(mouseOutImg, picSpectrum)
End Sub

```

```

Private Sub picSpectrum_MouseLeave(sender As Object, e As EventArgs) Handles picSpectrum.MouseLeave
    Dim mouseOutImg As Image(Of Bgr, Byte) = mouseImg.Clone
    DisplayImage(mouseOutImg, picSpectrum)
End Sub

```

#### 點選光譜橫線位置

```

Private Sub imgWebcam_MouseDown(sender As Object, e As MouseEventArgs) Handles imgWebcam.MouseDown
    Dim p As Point
    Dim tmpX, tmpY As Integer
    If imgWebcam.Image Is Nothing Then Exit Sub
    p.X = e.X : p.Y = e.Y '取得滑鼠敲擊點
    tmpX = CInt(Math.Round(p.X / imgWebcam.Width * inFrame.Width))
    tmpX = Math.Min(inFrame.Width - 1, tmpX) : tmpX = Math.Max(0, tmpX)
    tmpY = CInt(Math.Round(p.Y / (imgWebcam.Height - 1) * inFrame.Height))
    tmpY = Math.Min(inFrame.Height - 2 - CInt(Math.Round(SpectrumHeight / 2)), tmpY) : tmpY =
        Math.Max(CInt(Math.Round(SpectrumHeight / 2)) + 1, tmpY)
    scanLinePosition = Math.Round(tmpY / inFrame.Height, 4)
    Dim sw As StreamWriter = New StreamWriter(CurDir() & "\parameter.txt")
    sw.WriteLine(CStr(scanLinePosition)) : sw.WriteLine(CStr(mercuryB2)) : sw.WriteLine(CStr(Eu))
    sw.Close()
    Update_txb(txtBox, "W : " + CStr(inFrame.Width) + " H : " + CStr(inFrame.Height) + vbCrLf + "Select Point (" + CStr(tmpX)
        + "," + CStr(tmpY) + ")")
End Sub

```

### 擷取日光燈光譜

```
Private Sub btnCalibration_Click(sender As Object, e As EventArgs) Handles btnCalibration.Click
    If imgSpectrum.Image IsNot Nothing Then
        DisplayImage(spectrumImg.Resize(imgCalibrationB2.Width, imgCalibrationB2.Height, INTER.CV_INTER_LINEAR,
            False), imgCalibrationB2) 'spectrumImg 填入 imgCalibrationB2 640x20
        DisplayImage(spectrumImg.Resize(imgCalibrationEu.Width, imgCalibrationEu.Height, INTER.CV_INTER_LINEAR,
            False), imgCalibrationEu) 'spectrumImg 填入 imgCalibrationEu 640x20
        Update_tb(trackB2, enable, trackB2.Minimum, mercuryB2, trackB2.Maximum)
        Update_tb(trackEu, enable, trackEu.Minimum, Eu, trackEu.Maximum)
        Update_btn(enable, "完成校正", btnOK)
    End If
End Sub
```

### '光譜 B2(436) / Eu(611) 位置調整

```
Private Sub track_Scroll(sender As Object, e As EventArgs) Handles trackB2.Scroll, trackEu.Scroll
    mercuryB2 = trackB2.Value
    Eu = trackEu.Value
    Eu = Math.Max(mercuryB2 + 5, Eu)
    Update_tb(trackEu, enable, trackEu.Minimum, Eu, trackEu.Maximum)
    Dim sw As StreamWriter = New StreamWriter(CurDir() & "\parameter.txt")
    sw.WriteLine(CStr(scanLinePosition))
    sw.WriteLine(CStr(mercuryB2))
    sw.WriteLine(CStr(Eu))
    sw.Close()
End Sub
```

### '完成光譜校正

```
Private Sub btnOK_Click(sender As Object, e As EventArgs) Handles btnOK.Click
    Update_tb(trackB2, disable, trackB2.Minimum, trackB2.Value, trackB2.Maximum)
    Update_tb(trackEu, disable, trackEu.Minimum, trackEu.Value, trackEu.Maximum)
    Update_btn(disable, "完成校正", btnOK)
End Sub
```

### '凍結/即時 光譜

```
Private Sub btnSpectrum_Click(sender As Object, e As EventArgs) Handles btnSpectrum.Click
    If picSpectrum.Image IsNot Nothing Then
        If btnSpectrum.Text = "凍結頻譜" Then '即時改為凍結 顯示光譜
            Update_btn(enable, "即時頻譜", btnSpectrum)
            Update_btn(enable, "儲存光譜", btnSave)
            freeze = True
        Else '凍結改為及時 顯示光譜
            Update_btn(enable, "凍結頻譜", btnSpectrum)
            Update_btn(disable, "儲存光譜", btnSave)
            freeze = False
        End If
    End If
End Sub
```

### '儲存光譜影像 spectrumImg 1280x12

```
Private Sub btnSave_Click(sender As Object, e As EventArgs) Handles btnSave.Click
    SaveFileDialog1.FileName = DateTime.Now.ToString("MMddhhmmss")
    SaveFileDialog1.FileName = Mid(SaveFileDialog1.FileName, 1, 4) & "-" & Mid(SaveFileDialog1.FileName, 5, 4) & "-" &
        Mid(SaveFileDialog1.FileName, 9, 2) & ".png"
    SaveFileDialog1.ShowDialog()
End Sub
Private Sub SaveFileDialog1_FileOk(sender As Object, e As CancelEventArgs) Handles SaveFileDialog1.FileOk
    spectrumImg.Save(SaveFileDialog1.FileName)
End Sub
```

----- 更新 Imagebox, Button, Label -----

### 按鈕設定 btnStart, btnProduce, btnSave

```
Private Delegate Sub Update_btn_Delegate(ByVal ableStatus As Boolean, ByVal inText As String, ByRef Controls As Button)
Public Sub Update_btn(ByVal ableStatus As Boolean, ByVal inText As String, ByRef Controls As Button)
    If Controls.InvokeRequired Then
        Try
            Dim btnD As New Update_btn_Delegate(AddressOf Update_btn)
            Me.BeginInvoke(btnD, New Object() {ableStatus, inText, Controls})
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    Else
        Controls.Enabled = ableStatus
        If inText <> "" Then Controls.Text = inText
    End If
End Sub
```

```
End If
End Sub
```

顯示影像 imgWebcam, imgSpectrum, imgCalibrationB2, imgCalibrationEu

```
Private Delegate Sub DisplayImage_Delegate(ByVal outImg As Image(Of Bgr, Byte), ByRef Controls As UI.ImageBox)
```

```
Private Sub DisplayImage(ByVal outImg As Image(Of Bgr, Byte), ByRef Controls As UI.ImageBox)
```

```
    If Controls.InvokeRequired Then
        Try
            Dim displayimageD As New DisplayImage_Delegate(AddressOf DisplayImage)
            Me.BeginInvoke(displayimageD, New Object() {outImg, Controls})
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    Else
        Controls.Image = outImg
    End If
```

```
End Sub
```

'光譜分析繪製時間 lblWatch 的顯示 Delegate

```
Private Delegate Sub Update_lbl_Delegate(ByRef Controls As Label, ByVal inStr As String, ByVal location As Point)
```

```
Public Sub Update_lbl(ByRef Controls As Label, ByVal inStr As String, ByVal location As Point)
```

```
    If Controls.InvokeRequired Then
        Try
            Dim lblD As New Update_lbl_Delegate(AddressOf Update_lbl)
            Me.BeginInvoke(lblD, New Object() {Controls, inStr, location})
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    Else
        Controls.Text = inStr
        Controls.Location = location
    End If
```

```
End Sub
```

'textBox 的 Delegate

```
Private Delegate Sub Update_txb_Delegate(ByRef Controls As TextBox, ByVal inStr As String)
```

```
Public Sub Update_txb(ByRef Controls As TextBox, ByVal inStr As String)
```

```
    If Controls.InvokeRequired Then
        Try
            Dim txbD As New Update_txb_Delegate(AddressOf Update_txb)
            Me.BeginInvoke(txbD, New Object() {Controls, inStr})
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    Else
        Controls.Text = inStr
    End If
```

```
End Sub
```

'trackB2, trackEu 拉桿的設定 Delegate

```
Private Delegate Sub Update_tb_Delegate(ByRef Controls As TrackBar, ByVal ableStatus As Boolean, ByVal minimum As Long,
    ByVal value As Long, ByVal maximum As Long)
```

```
Public Sub Update_tb(ByRef Controls As TrackBar, ByVal ableStatus As Boolean, ByVal minimum As Long, ByVal value As Long,
    ByVal maximum As Long)
```

```
    If Controls.InvokeRequired Then
        Try
            Dim tbD As New Update_tb_Delegate(AddressOf Update_tb)
            Me.BeginInvoke(tbD, New Object() {Controls, ableStatus, minimum, value, maximum})
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    Else
        Controls.Enabled = ableStatus
        Controls.Minimum = minimum
        Controls.Maximum = maximum
        value = Math.Min(Controls.Maximum, value)
        value = Math.Max(Controls.Minimum, value)
        Controls.Value = value
    End If
```

```
End Sub
```

```
-----
----- 一般副程式
-----
```

'\_capture 與選取的 webcam 連接

```
Private Sub SetupCapture(Camera_SelectedIndex As Integer)
```

```

CameraDevice = Camera_SelectedIndex
If _capture IsNot Nothing Then '若_capture 已取得 webcam, 則先釋放 _capture
    If _capture.GrabProcessState = System.Threading.ThreadState.Running Then
        _capture.Pause()
        _capture.Stop()
        RemoveHandler _capture.ImageGrabbed, AddressOf GetFrame '停止事件_capture.ImageGrabbed 處理
    End If
    _capture.Dispose()
End If
Try
    _capture = New Capture(CameraDevice) 'Set up capture device
    _capture.SetCaptureProperty(Emgu.CV.CvEnum.CAP_PROP.CV_CAP_PROP_FRAME_WIDTH, webcamWidth)
    _capture.SetCaptureProperty(Emgu.CV.CvEnum.CAP_PROP.CV_CAP_PROP_FRAME_HEIGHT, webcamHeight)
    AddHandler _capture.ImageGrabbed, AddressOf GetFrame '啟動事件_capture.ImageGrabbed 處理
Catch ex As NullReferenceException
    MessageBox.Show(ex.Message)
End Try
End Sub
End Class

```