

2005 年臺灣國際科學展覽會

研究報告

區別：

科別：工程學科

作品名稱：魚眼鏡頭影像之變形校正

關鍵詞：魚眼鏡頭、變形校正、_____

編號：

英文摘要(Abstract) :

Distortion Correction of Fisheye Lenses

Owing to their wide viewing angle, fisheye lenses are adopted in many territories, such as computer vision, forest ecology, astronomy, medical image, optical inspect and so on. But the images taken with the fisheye lenses often have severe distortions. If we wish to derive the metric information from these distorted images, they have to be transformed into distortion-free images.

The previous approach on image distortion correction often uses the polynomial curve fitting method. The accuracy of this method largely depends on whether the found polynomial function fits the lens system or not. But it is usually difficult to find a suitable polynomial function in many cases. To exempt from such a problem, in this report, we propose an interpolation method for fisheye images calibration. Our method is based on the perspective projection mechanism and the relation-mapping table. It can fit the real distortions of the lens system better if there are more sampling points in inner surface of the cylindrical pattern. Experimental results with real fisheye images are presented, and the correction results appear satisfactory.

中文摘要：

魚眼鏡頭影像之變形校正

魚眼鏡頭因視角較大可取得大範圍的影像，所以被廣泛地運用在各種領域，例如電腦視覺、光學檢測、醫療內視鏡影像、森林生態、天文學等，由於魚眼鏡頭影像有很大的變形失真，若要利用影像中的資訊作為量測或度量方面的用途，必需先將其轉換成人類所習慣的透視投影影像。

目前校正相機鏡頭影像變形失真的做法，大都是採用多項式函數曲線擬合法，其校正成果的好壞，端賴於尋找出的多項式函數與真實相機系統成像投射函數的擬合程度，不過，在許多時候，找到的多項式函數並無法與相機系統做完全地擬合。在本篇報告中，我們提出了一種使用插值法的魚眼鏡頭影像之變形校正方法，這方法主要是利用透視投影機制以及入射光線與其成像點間之關係對應表來完成變形影像的校正，它可以改善曲線擬合法無法保證多項式函數一定會經過取樣點的缺點，尤其當圓柱形校正樣板內佈有更多的取樣點時，其符合真實相機系統投射方式的程度會更為精準。我們以程式實作並用來校正真實的魚眼鏡頭影像，由實驗的結果看來，變形校正的成果還不錯。

一、研究動機與目的：

目前最流行的商品，當屬來自於日本的大頭狗（圖 1-1），不論是玩偶或是文具、馬克杯，甚至衣服、包包上都可以看到這些狗狗的變形大頭照，其逗趣可愛的模樣，令人愛不釋手。類似的變形影像，也可由家中大門上防盜用的警眼看到（圖 1-2），為什麼會有中間影像放大但周圍影像壓縮的變形失真呢？經過查詢相關資料後，發現是魚眼鏡頭所造成的，由於其誇張的變形饒富趣味性，使得我們產生了極大的研究興趣。

與傳統的鏡頭比較起來，具有大視角的魚眼鏡頭可取得較大範圍的影像，現今廣泛地運用在各種領域，例如電腦視覺、光學檢測、醫療內視鏡影像、全景影像、森林生態、天文學等，由於魚眼鏡頭影像有很大的變形失真，若要利用影像中的資訊作為量測或度量方面的用途，必需先將其轉換成人類所習慣的透視投影影像，因此，一套良好的魚眼鏡頭影像變形失真校正技術是非常重要的，雖然目前已有很多的文獻在探討這方面的問題，但我們希望以較為直觀的方式發展出魚眼鏡頭光學失真校正方法，能夠輕易地辨認出實物相對尺寸的大小。



圖 1-1 最新超人氣玩偶「大頭狗」

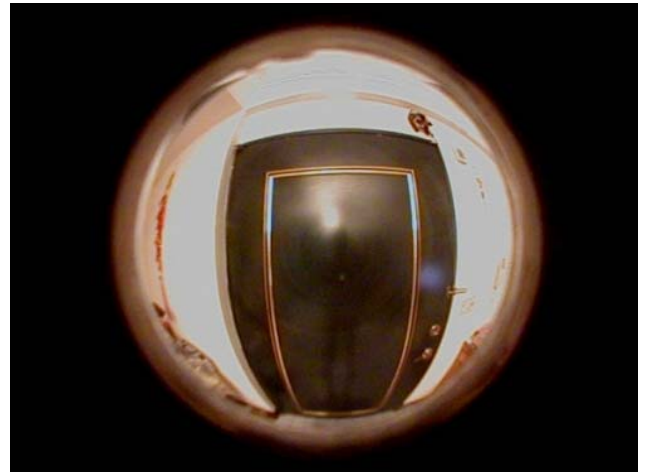


圖 1-2 由防盜用廣角警眼所看到的景象

二、研究過程：

1. 符合人類習慣的透視投影成像機制

傳統相機乃是採用針孔成像幾何模型[2]，其成像機制是根據透視投影的關係所建立(如圖 2 所示)，其中 X 、 Y 、 Z 三軸構成相機的三維空間直角座標，原點 C 為相機鏡頭的投射中心， Z 軸與相機的光軸重合； x 、 y 兩軸構成影像平面的二維平面直角座標，原點 c 點為影像平面的中心點，影像平面與 XY 平面平行且兩者的距離為 f （相機鏡頭之焦距）。我們可由圖 2 推導出相機三維空間中的一點 $P(X, Y, Z)$ 與其影像平面上的成像點 $p(x, y)$ 的關係式

$$x = \frac{f}{Z} X$$

$$y = \frac{f}{Z} Y$$

說明：

$$\ominus \Delta PAB \sim \Delta pac$$

$$\therefore \frac{\overline{AB}}{pa} = \frac{\overline{PA}}{ca} = \frac{\overline{BP}}{cp} \dots\dots\dots (1)$$

$$\ominus \Delta PBC \sim \Delta pcC$$

$$\therefore \frac{\overline{BP}}{cp} = \frac{\overline{CB}}{Cc} \dots\dots\dots (2)$$

由(1)(2)可得

$$\frac{\overline{AB}}{pa} = \frac{\overline{PA}}{ca} = \frac{\overline{BP}}{cp} = \frac{\overline{CB}}{Cc}$$

$$\Rightarrow \frac{Y}{y} = \frac{X}{x} = \frac{\overline{BP}}{cp} = \frac{Z}{f}$$

$$\text{故 } x = \frac{f}{Z} X$$

$$y = \frac{f}{Z} Y$$

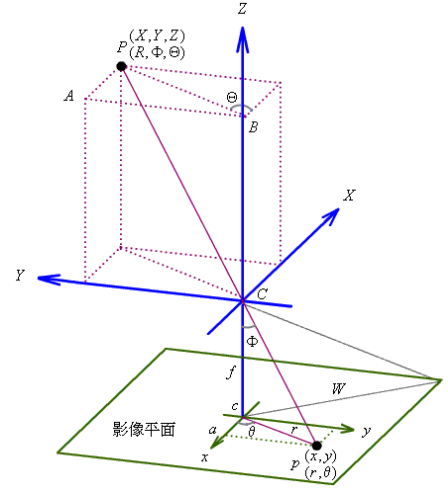


圖 2 針孔成像幾何模型

若是將三維空間直角座標以球面座標來代替，則球面上一點 $P(R, \Phi, \Theta)$ 與其影像平面上成像點的極座標 $p(r, \theta)$ 和平面直角座標 $p(x, y)$ 的關係式分別為

$$r = f \tan \Phi ; \theta = \Theta$$

$$x = r \cos \theta = f \tan \Phi \cos \Theta ; y = r \sin \theta = f \tan \Phi \sin \Theta$$

鏡頭視角的大小由影像平面的尺寸與鏡頭焦距所決定，其視角 FOV 為

$$2 \tan^{-1} \frac{W}{f}$$

2. 相關文獻探討

在針孔成像的幾何模型中，似乎可以很輕易地調整鏡頭視角的大小，但在真實的相機系統中，由於影像平面的尺寸有其大小上的限制，所以針孔成像幾何模型就不適用於有較大視角的廣角鏡頭和魚眼鏡頭，必須採用廣角成像幾何模型來設計，常用的有 stereographic、equidistant、sine law、equi-solid angle 等投射方法[3]。

設計出來的魚眼鏡頭結構是相當地複雜，通常由好幾片透鏡組合而成，其設計的目的在使得三維空間來的入射光線經由這些透鏡折射後，能夠遵循著上述的某種投射函數而成像在影像平面上，圖 3 顯示的是 Miyamoto 於 1964 年所設計的魚眼鏡頭平面圖，共有 9 片透鏡， $r = \alpha \times \sin(\beta\Phi)$ 為其成像投射函數，其中 r 為影像平面中心點至成像點的距離、 α 是影像縮放因子、 β 為投射函數之參數、 Φ 是入射光線與相機光軸的夾角[9]。

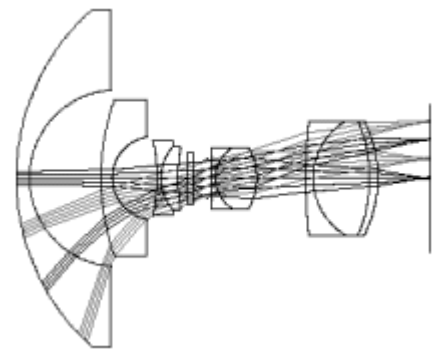


圖 3 Miyamoto 的魚眼鏡頭

在真實的相機系統中，不論鏡頭的視角大小如何，由於在設計與成本上的考量及鏡頭

製作技術上的瓶頸，任何鏡頭均無法完全依循設計時所採用的投射函數來成像，或多或少都有光學失真的情形，尤其在價格越便宜與視角越大的鏡頭上更為明顯。關於鏡頭光學失真校正的問題有相當多的文獻報告提出[1][2][5][6][7][8][10]，這些報告大部分都使用一些精密的儀器或是校正樣板，利用多項式型態的校正函數模擬出真實三維空間座標與影像平面二維座標間之對應關係（投射函數），然後將影像平面上的成像點重新移動到符合理想透視投影機制的位置，進而達成鏡頭光學失真校正的目的。其校正成果的好壞，端賴於一些重複性測試與最佳化的步驟以尋找最適當的多項式校正係數，在完成鏡頭光學失真校正後，同時也建立影像平面上校正前與校正後成像點的對應表，以後新的影像只需要經由查表的方式就可得到校正後的影像，其中最有名且被其他文獻廣為引用的報告是 1987 年與 1992 年分別由 Tsai[1]與 Weng et al. [2]所提出。

3. 校正魚眼鏡頭變形影像的新構想

設計魚眼鏡頭時所依循的投射函數有非常多的種類，加上成本上的考量及鏡頭製作技術上的瓶頸，使得真實魚眼鏡頭的成像會變得相當地複雜多變，若是以一個「簡潔漂亮」的多項式校正函數來模擬此成像投射函數，進而用來將影像平面上的每一個成像點正確無誤地校正到符合透視投影機制的位置，實在是一件不太容易令人信服的事，我們可從 Fleck et al. [4]的實驗結果中來驗證這個說法，Fleck et al. 以一個視角 116.5° 、焦距 2.6 mm的鏡頭來作測試，我們可由圖 4 中看到 Tsai 的多項式校正函數與成像投射函數並沒有曲線擬合的很好。

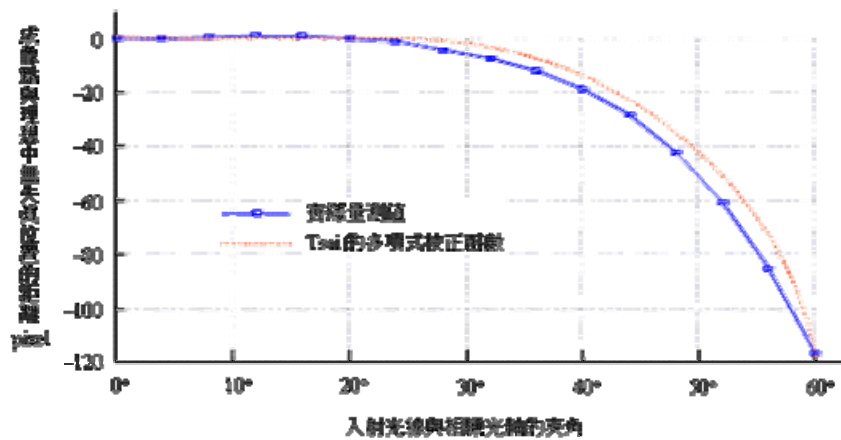


圖 4 Fleck et al.的實驗測試結果

由於魚眼鏡頭影像均是由空間中入射光線經鏡頭折射後的成像點所構成，我們可將魚眼鏡頭看成一個「黑盒子」，不用去理會入射光線在鏡頭中折射的精確路徑以及魚眼鏡頭的透鏡光學設計，只要找到入射光線與其影像平面上成像點的關係對應表即可，以後，當我們要校正由此魚眼鏡頭所拍攝之影像而產生變形校正後影像的時候，變形校正後影像上的任何一點 $p'(x', y')$ ，均可以藉由透視投影機制推算出其入射光線 $Ray(\Phi, \Theta)$ ，再透過關係對應表從魚眼鏡頭影像中取出該入射光線的對應點 $p(x, y)$ 來填入，以達成影像變形校正的目的，圖 5 可用來簡單地表示我們魚眼鏡頭影像變形校正之構想。

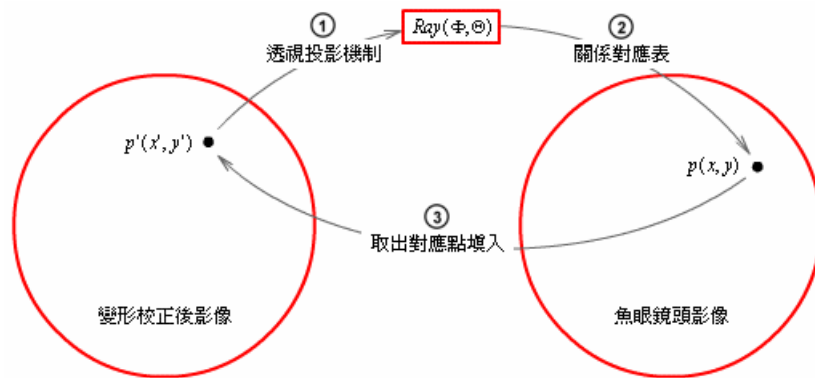


圖 5 變形校正影像透過透視投影機制與關係對應表從魚眼鏡頭影像中取點

我們的關係對應表是採用入射光線 $Ray(\Phi, \Theta)$ 與其影像平面成像點 $p(x, y)$ 的對應關係 (其入射光線方向角的定義可見圖 6)，而非使用三維空間中的點 $P(R, \Phi, \Theta)$ 之對應關係，主要是因為我們的校正構想中，需使用透視投影機制推算出變形校正後影像上各點的入射光線，而在透視投影成像機制中，空間中的一點 $P(R, \Phi, \Theta)$ 被映射到影像平面上的成像點為 $p(x, y)$ ，其中 $x = f \tan \Phi \cos \Theta$ ； $y = f \tan \Phi \sin \Theta$ ，與空間中的座標 R 無關，這意味著空間中的點只要其 Φ 與 Θ 座標相同就會被映射到影像平面上的同一點，而空間中 Φ 與 Θ 座標相同的點均在入射光線 $Ray(\Phi, \Theta)$ 上。

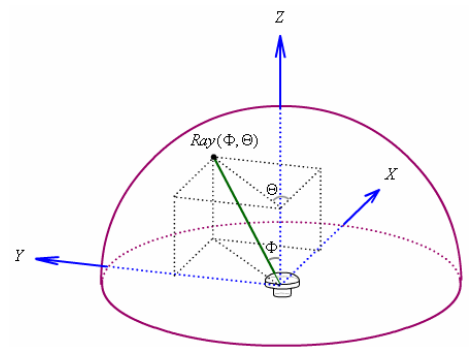


圖 6 入射光線的方向角度

4. 空間中入射光線與其魚眼鏡頭影像成像點之對應關係

我們可以三個步驟來說明入射光線與其魚眼鏡頭成像點關係對應表的建立方式，首先，我們必須尋找一個適合於魚眼鏡頭的校正樣板，並在其適當地位置上設置一些取樣點用來代表入射光線；然後，以魚眼鏡頭來拍攝這個佈有取樣點的校正樣板；最後，根據取樣點所代表的入射光線及其所對應的成像點位置建立起對應關係，當然，取樣點僅能代表所有入射光線的一部份，我們可以使用插值法將整個關係對應表補齊。

A. 校正樣板的選擇與製作

在以往的文獻報告中，均是採用平面式的校正樣板來尋找三維空間座標與影像平面二維座標間之對應關係，但是對於視角很大甚至超過 180° 的魚眼鏡頭，平面式的校正樣板就不是很恰當的選擇了，必須能夠涵蓋整個鏡頭的視角才算是符合需求，我們採用了內徑 $D = 17\text{cm}$ 的壓克力圓管，依其內徑繪製了如圖 7-1 所示的網狀格子圖，繪製完成後將其環接成圓柱形放入壓克力圓管內，並在壓克力圓管的另端開口封貼上在圓心繪有黑色圓點的圓形紙板，如此就完成了我們的圓柱形校正樣板。

網狀格子圖中每條橫線與直線的交會點以及圓形紙板上的黑色圓點我們稱之為取樣點，若是將魚眼鏡頭的投射中心置於圓柱形校正樣板中心點 (如圖 7-2 所示之 C 點，為網狀格子圖底部算起第二條橫線環接後形成的圓之圓心) 拍攝影像的話，對於鏡頭來說，每個取樣點所代表的入射光線其方向角是已知的，例如圖 7-2

中的 P 點，假設其位於橫線 $\frac{1}{2}D \cot 20^\circ$ 與直線 $\frac{1}{24}\pi D$ 的交會點上，那麼 P 點所代表的入射光線為 $Ray(\Phi, \Theta)$ ，其中 $\Phi = 20^\circ$ ， $\Theta = \frac{1}{12}\pi$ 。

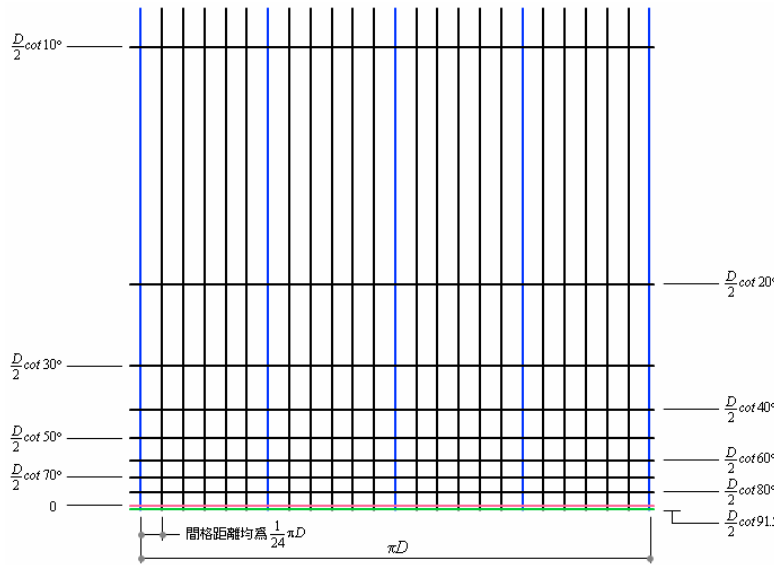


圖 7-1 依據壓克力圓管內徑繪製的網狀格子圖

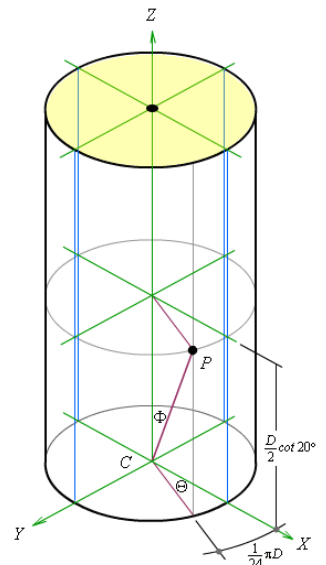


圖 7-2 圓柱形校正樣板

B. 拍攝設備與裝置

整個魚眼鏡頭拍攝設備與裝置如圖 8-1 所示，我們所使用的器材有：數位相機 Nikon CoolPix 4500、魚眼鏡頭 Nikon FC-E8（視角 183° 、焦距 8 mm）、線控拍攝器、天文望遠鏡腳架（可微調赤經與赤緯角度）、USB 傳輸線、圓柱形校正樣板、水平儀、固定腳架、筆記型電腦 HP Evo N800v，所使用的軟體有：Ulead PhotoImpact8、Nikon View6。圖 8-2 是我們所拍的校正樣板影像，在正式拍攝前有三件事情是必須特別注意的地方，第一，置放校正樣板的腳架平台以及相機機身必須做好水平調校；第二，相機的光軸須與圓柱形校正樣板的軸心重合；第三、相機的投射中心需置於圓柱形校正樣板的中心點（ C 點）。



圖 8-1 拍攝裝置示意圖

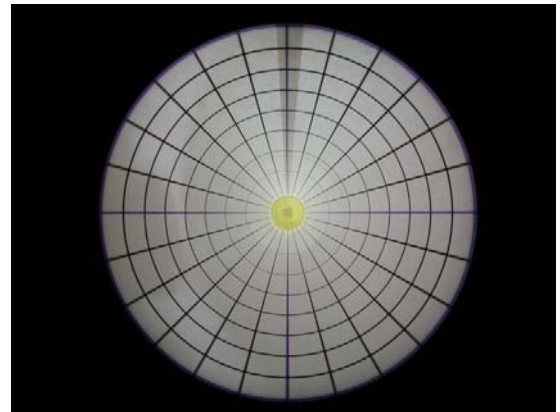


圖 8-2 校正樣板的魚眼鏡頭影像

C. 入射光線與其魚眼鏡頭成像點間之關係對應表

由於相機的投射中心位於圖 7-2 中的 C 點，所以每一個取樣點所代表的入射光線 $Ray(\Phi, \Theta)$ 之方向角可由網狀格子圖中推得，至於入射光線所對應的成像點 $p(x, y)$ 在影像平面上的二維直角座標，我們可使用影像軟體 PhotoImpact 在所拍得的影像

中取得（取得資料詳見附錄一）。因為有四個變數（ Φ 、 Θ 、 x 、 y ），我們可以分別建立 (Φ, Θ, x) 與 (Φ, Θ, y) 的關係對應表，建立好取樣點所代表的入射光線及其相對應成像點位置的對應關係後，再使用 MATLAB 軟體的二維三次多項式插值法將整個關係對應表補齊，其程式及執行結果可見程式 1-1、程式 1-2、圖 9-1 與圖 9-2，除了繪製出關係對應表的圖形外，我們也將結果存成兩個二維陣列資料檔，供以後的魚眼鏡頭影像校正查表用。

```
clear; '清除記憶體中的變數與函數
xdata; '載入取樣點 $(\Phi, \Theta, x)$ 資料
[X,Y]=meshgrid(0:1:90, 0:1:360); '用 XY 陣列建立三維圖形
Z=griddata(x,y,z,X,Y,'cubic'); '插值產生網格資料
mesh(X,Y,Z); '繪製三維網格圖形
hold on; '鎖定已繪製的圖形
plot3(x,y,z, '.'); '繪出取樣點
hold off; '已繪製的圖形解除鎖定
box on; '繪出座標外框
fid=fopen('datax.txt','w');
fprintf(fid,'%8.2f,Z); '將插值法求出的資料儲存於檔案
fclose(fid);
```

程式 1-1 插值法補齊 (Φ, Θ, x) 對應表

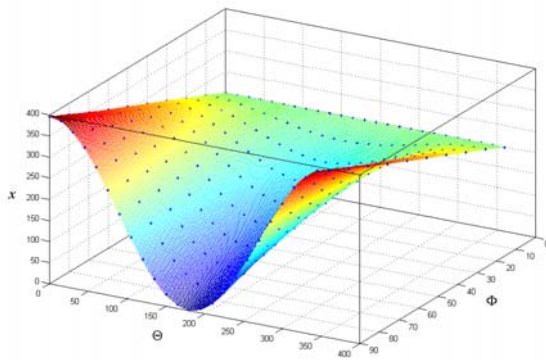


圖 9-1 關係對應表 (Φ, Θ, x)

```
clear; '清除記憶體中的變數與函數
ydata; '載入取樣點 $(\Phi, \Theta, y)$ 資料
[X,Y]=meshgrid(0:1:90, 0:1:360); '用 XY 陣列建立三維圖形
Z=griddata(x,y,z,X,Y,'cubic'); '插值產生網格資料
mesh(X,Y,Z); '繪製三維網格圖形
hold on; '鎖定已繪製的圖形
plot3(x,y,z, '.'); '繪出取樣點
hold off; '已繪製的圖形解除鎖定
box on; '繪出座標外框
fid=fopen('datax.txt','w');
fprintf(fid,'%8.2f,Z); '將插值法求出的資料儲存於檔案
fclose(fid);
```

程式 1-2 插值法補齊 (Φ, Θ, y) 對應表

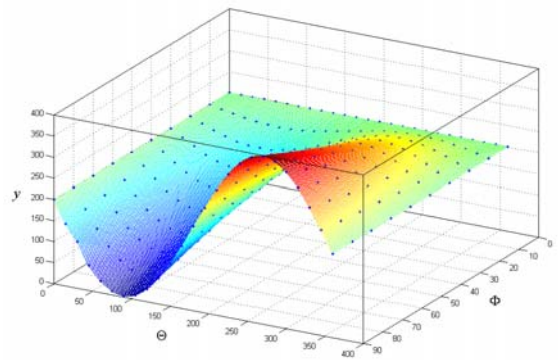


圖 9-2 關係對應表 (Φ, Θ, y)

5. 魚眼鏡頭影像之變形校正程序

當空間中入射光線與其魚眼鏡頭影像成像點之關係對應表完成後，再加上透視投影成像機制，就可以用來校正任何由此魚眼鏡頭所拍攝的變形影像了。在啟動校正程序之前必須先給予兩個參數（變形校正後影像的半徑 W 及所能呈現的視角 FOV ），校正程序先依照參數 W 建立一個新的影像空間用來存放變形校正後之影像，然後，針對變形校正後影像上的每一點 $p'(x', y')$ 的位置，經由透視投影機制推算出其空間中的入射光線 $Ray(\Phi, \Theta)$ ，推算的結果如下，

若是 $x'^2 + y'^2 = 0$ ，則

$$\Phi = 0^\circ ; \Theta = 0^\circ$$

若是 $x'^2 + y'^2 \neq 0$ ，則

$$\Phi = \tan^{-1} \frac{\tan(FOV/2) \sqrt{x'^2 + y'^2}}{W}$$

$$\Theta = \sin^{-1} \frac{|y'|}{\sqrt{x'^2 + y'^2}} \quad \text{當 } x' \geq 0, y' \geq 0$$

$$\Theta = \pi - \sin^{-1} \frac{|y'|}{\sqrt{x'^2 + y'^2}} \quad \text{當 } x' \leq 0, y' \geq 0$$

$$\Theta = \pi + \sin^{-1} \frac{|y'|}{\sqrt{x'^2 + y'^2}} \quad \text{當 } x' \leq 0, y' \leq 0$$

$$\Theta = 2\pi - \sin^{-1} \frac{|y'|}{\sqrt{x'^2 + y'^2}} \quad \text{當 } x' \geq 0, y' \leq 0$$

最後，利用入射光線與其魚眼鏡頭影像之關係對應表從魚眼鏡頭影像中取出該入射光線的對應點 $p(x, y)$ 來填入，當所有變形校正後影像上的點均填入後，就完成了魚眼鏡頭影像之變形校正程序。另外，補充說明校正程序所需的兩個參數在給定時的一些限制，第一，能夠呈現變形校正後影像的半徑 W 必須依據顯示影像設備的大小尺寸來決定；第二，因為透視投影機制本身的成像特性使然，變形校正後的影像所能呈現的視角 FOV 必須小於 180° 。

6. 程式實作

以 VB.NET 程式來實作是最方便不過了，雖然有執行速度稍慢的缺點，但它提供了適用於多媒體與繪圖的 GDI+繪圖裝置介面 (Graphics Device Interface)，可讓程式設計人員不需要注意顯示裝置的硬體詳細資料，只要透過呼叫 GDI+類別所提供的方法，便可以開發出與裝置無關的應用程式並在螢幕顯示資訊。

我們在表單視窗中插入兩個 PictureBox、兩個 TextBox、一個 Button 按鈕，右邊的 PictureBox 用來載入原始的魚眼鏡頭影像，變形校正後的影像存放在左邊的 PictureBox，TextBox 可供使用者輸入變形校正後影像的半徑 W 及所能呈現的視角 FOV ，輸入完成後，按下 Button 按鈕來啟動變形校正程序。

變形校正程式啟動後，先建立兩個二維陣列用來存放 MATLAB 軟體輸出的資料檔，這兩個陣列就是用來校正魚眼鏡頭影像的關係對應表 (Φ, Θ, x) 與 (Φ, Θ, y) ，然後依照使用者輸入的兩個參數變形校正後影像半徑 W 及所能呈現視角 FOV 來完成魚眼鏡頭影像的變形校正，整個程式的運作流程如圖 10 所示，詳細原始程式碼可參見附錄二。

圖 11-1 為變形校正程式執行時的情形，右邊為用來校正的原始魚眼鏡頭影像，我們可以看到影像中的輕鋼架與地磚有很明顯

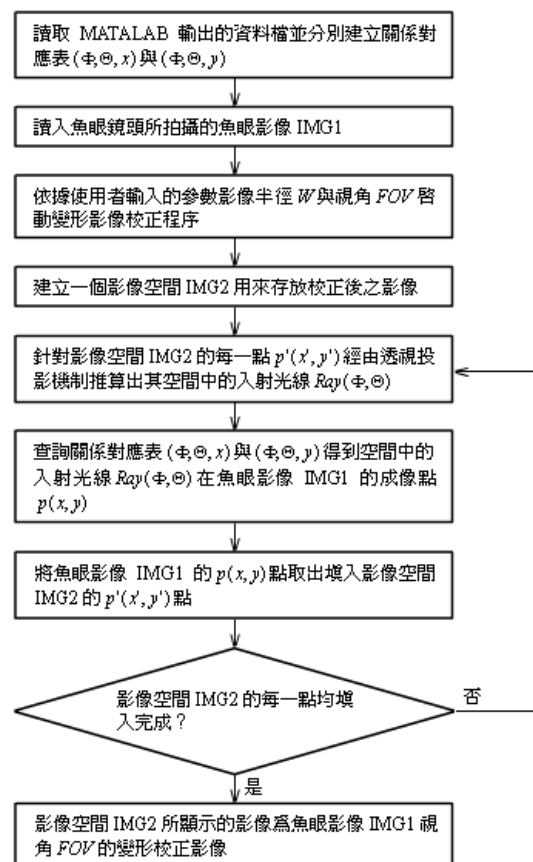


圖 10 變形校正程式運作流程圖

的彎曲變形，綠色圓圈內部是依照使用者輸入的視角參數 FOV 取出來校正的部份，我們可以從左邊校正後的影像看出，校正的成果尚稱理想，輕鋼架與地磚的線條經過校正後都變得比較直了。圖 11-2 是另外一幅魚眼鏡頭影像之變形校正情形，由樑柱與門窗的線條可以看出校正的成果還不錯。

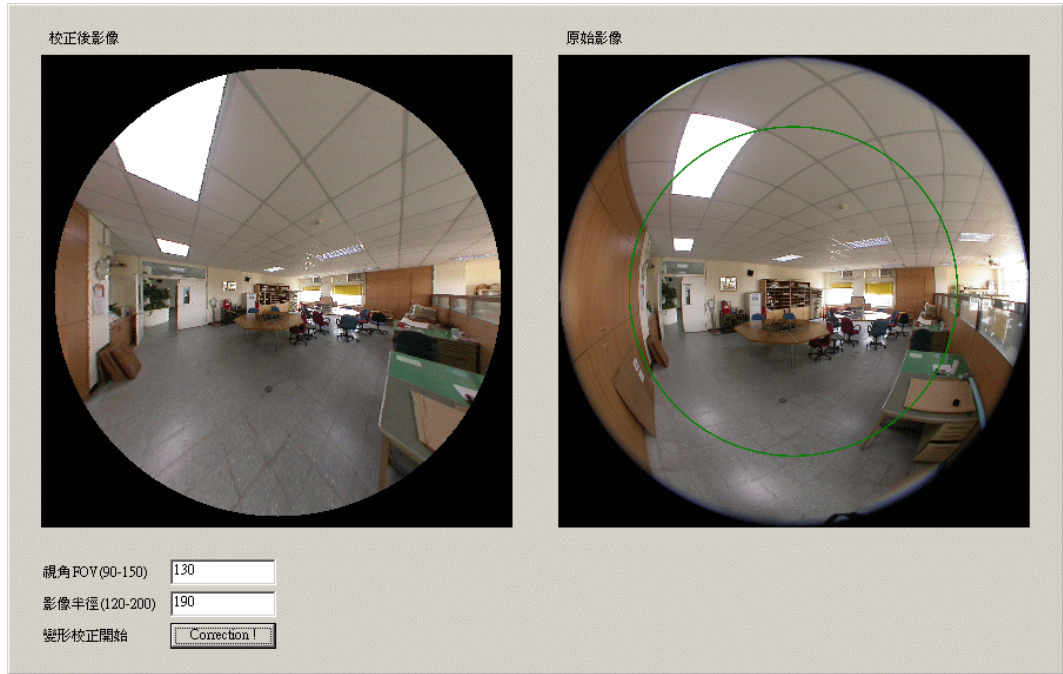


圖 11-1 魚眼鏡頭影像之變形校正式執行視窗

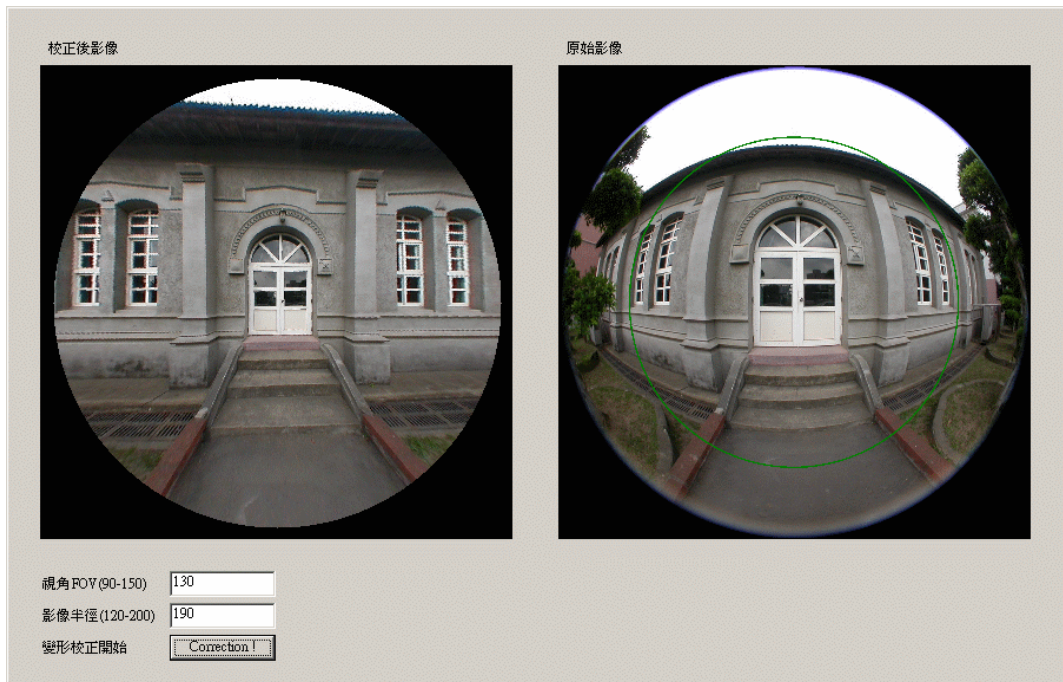


圖 11-2 另一幅魚眼鏡頭影像之變形校正

圖 11-3 是拿校正樣板的魚眼鏡頭影像來作變形校正，我們可以在校正樣板上設計一些測試點（取樣點除外），由其所代表的入射光線 $Ray(\Phi, \Theta)$ 經透視投影機制推算出其在完全無變形失真影像上的位置 (x_p, y_p) ，其中

$$x_p = \frac{W}{\tan(FOV/2)} \tan \Phi \cos \Theta ; y_p = \frac{W}{\tan(FOV/2)} \tan \Phi \sin \Theta$$

將其與校正後所在位置 (x', y') 間的誤差值來檢驗校正成果的好壞（見附錄三）。

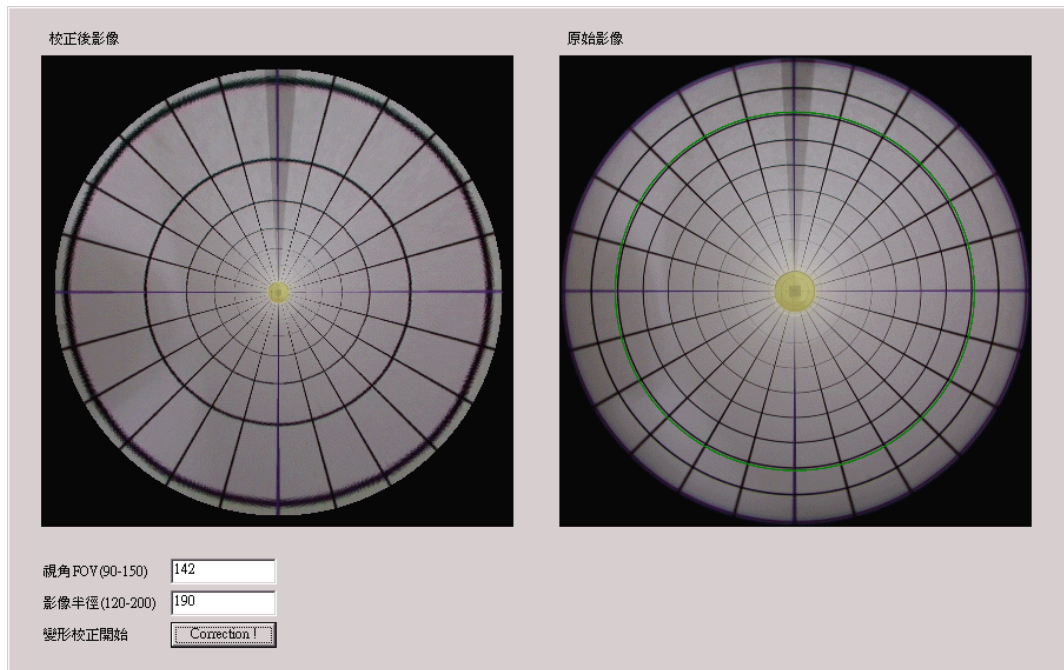


圖 11-3 校正樣板的魚眼鏡頭影像之變形校正

三、討論與應用：

1. 討論

目前校正相機鏡頭影像變形失真的做法，多是採用多項式型態的校正函數來模擬相機鏡頭的投射函數，進而達成鏡頭影像變形校正的目的，其校正成果的好壞，端賴於尋找出的校正函數與真實相機系統成像投射函數的擬合程度，不過，由 Fleck et al.[4]的研究報告中知道，要找到與相機系統完全擬合的校正函數是非常困難的一件事。由於無論採用何種校正方法，在完成鏡頭光學失真校正後，均會建立影像平面上成像點在校正前與校正後位置的對應表，以後只要是利用此鏡頭拍攝的影像，僅需經由查表的方式就可很快速地得到完成校正後的影像，所以，我們的校正方法主要就是針對關係對應表的建立，而避免去尋找費時又不見得精準的校正函數。

我們根據一些已知的取樣點數據，並利用 MATLAB 軟體三次多項式插值法來建立整個入射光線與其成像點位置間之關係對應表，由於取樣點是很均勻地分佈在校正樣板內部，所以可視為入射光線對於整個鏡頭面都做了探測，如此而得到的關係對應表會比多項式型態的校正函數更能符合真實相機系統的投射方式，尤其當校正樣板內佈有更多的取樣點時，其符合真實相機系統投射方式的程度會更為精準。

在建立入射光線與其成像點位置間之關係對應表時，我們遭遇到最大的困擾就是相機鏡頭與圓柱形校正樣板間的定位問題，主要有兩點，第一，因為相機的投射中心需置於校正樣板的中心點，我們該如何找到相機投射中心的正確位置？第二，相機的光軸須與校正樣板的軸心重合，我們如何知道相機光軸是否已和校正樣板的軸心重合了沒

有？我們所採取的可行方案分述如下。

A. 相機投射中心的位置

對於相機系統而言，空間中來的入射光線均會通過投射中心而成像在影像平面上，所以只要找到兩道已知的入射光線，其交會點就是投射中心的位置了，我們如圖 12 所示般，在相機的近處與遠處分別放置與地面垂直的矮柱 \overline{AC} 與高柱 \overline{BD} ，矮柱 \overline{AC} 放好後，調整高柱 \overline{BD} 的位置使得高柱 \overline{BD} 的頂端 B 與矮柱 \overline{AC} 的頂端 A 在影像平面上為同一點（重合），如此高柱 \overline{BD} 與矮柱 \overline{AC} 的頂端連線 \overline{BA} 就可視為一道已知的入射光線，以同樣得方式，我們可求得另一道入射光線 \overline{FE} 。假設兩道入射光線在地面投影 \overline{DC} 及 \overline{HG} 之交點為 O' ，相機鏡頭面在地面的投影為 O ，那麼 $\overline{OO'}$ 即為相機投射中心與鏡頭面的距離。

上述的方法理論上應該可以找到相機投射中心的位置，經過 10 次的測量後， $\overline{OO'}$ 的平均值為 21.4mm ，但是所使用的器材精密度會影響到量測值甚巨，還好查閱鏡頭說明書後知道 Nikon FC-E8 魚眼鏡頭的視角為 183° ，網狀格子圖上繪製的綠色橫線 $\frac{1}{2}D \cot 91.5^\circ$ ，在調整相機的投射中心使其位於校正樣板的中心點時，給予我們很大的輔助。

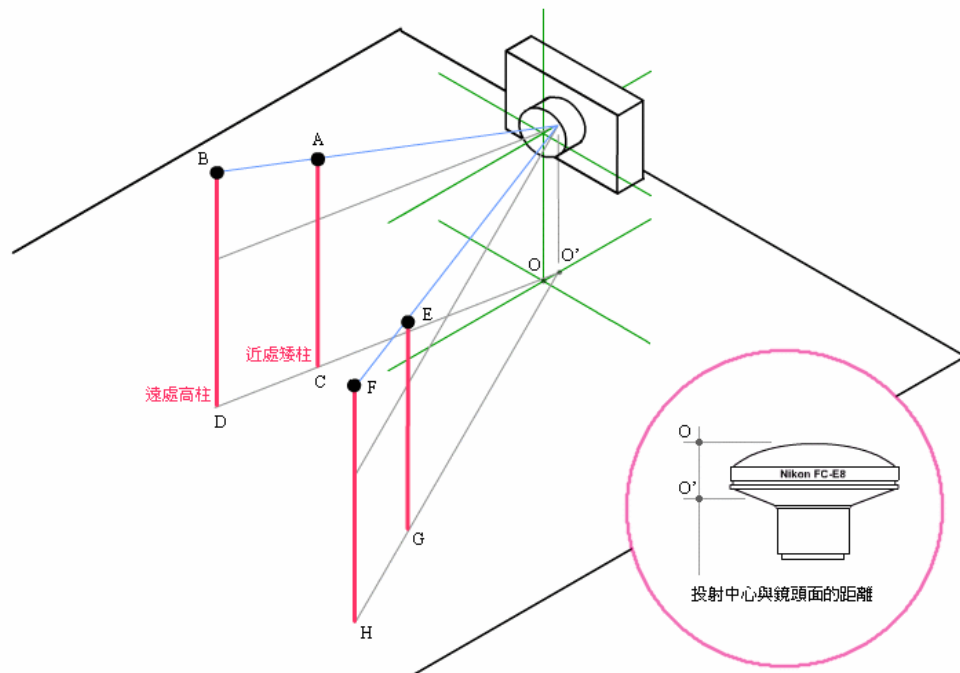


圖 12 尋找相機投射中心的位置

B. 相機光軸與校正樣板的軸心重合

相機光軸與影像平面的交點為影像中心點，我們可藉著觀察所拍攝之影像的中心點來判別相機光軸是否已和校正樣板的軸心重合了沒有，其實，只要將上述尋找相機投射中心位置的方法稍作修改，就可以用來尋找影像中心點，如圖 13 所示，平面 P 為與地面垂直的平面，我們在平面 P 上分別插上與其垂直的長柱 \overline{BD} 與短柱 \overline{AC} ，長柱 \overline{BD} 設定好後，調整短柱 \overline{AC} 的位置使得長柱 \overline{BD} 的頂端 B 與短柱 \overline{AC} 的頂端 A 在影像平面上為同一點（重合），如此短柱 \overline{AC} 與長柱 \overline{BD} 的頂端連線 \overline{AB}

可視為一道與相機光軸相交的入射光線，以同樣得方式，我們可求得另一道入射光線 \overline{EF} ，假設兩道入射光線在平面 P 上的投影 \overline{CD} 及 \overline{GH} 之交點為 I ，那麼 I 點在影像平面上的成像點就是影像中心點。

我們將相機拍攝的影像設定為 1600×1200 像素，經過 10 次的測量後，影像中心點位置的平均值大約為 $(813.3, 610.2)$ ，同樣地，理論上我們的方法是可行的，但是，除非使用非常精密的 XYZ 三軸平台來量測，不然很難去控制影像中心點量測位置與真實位置間的誤差值。網路上有則關於「光斑」的報導引起了我們的注意，當相機面對強光拍攝時，光線射到鏡頭內，會產生複雜的光學折射、反射或散射現象，會使得影像中出現光斑 (Lens Flare)，光斑由強光處發出，會成一直線延伸，一定會通過這張影像的中心點 (<http://w3.nctu.edu.tw/~u9112507/032/>)。我們將圖 14-1、圖 14-2、圖 14-3 等三張影像的光斑延伸線重疊在圖 14-4 上，三條線的交點分別為 $(814, 605)$ 、 $(808, 610)$ 、 $(819, 613)$ ，平均值為 $(813.7, 609.3)$ ，與我們測量的影像中心點的位置平均值倒是蠻接近的。

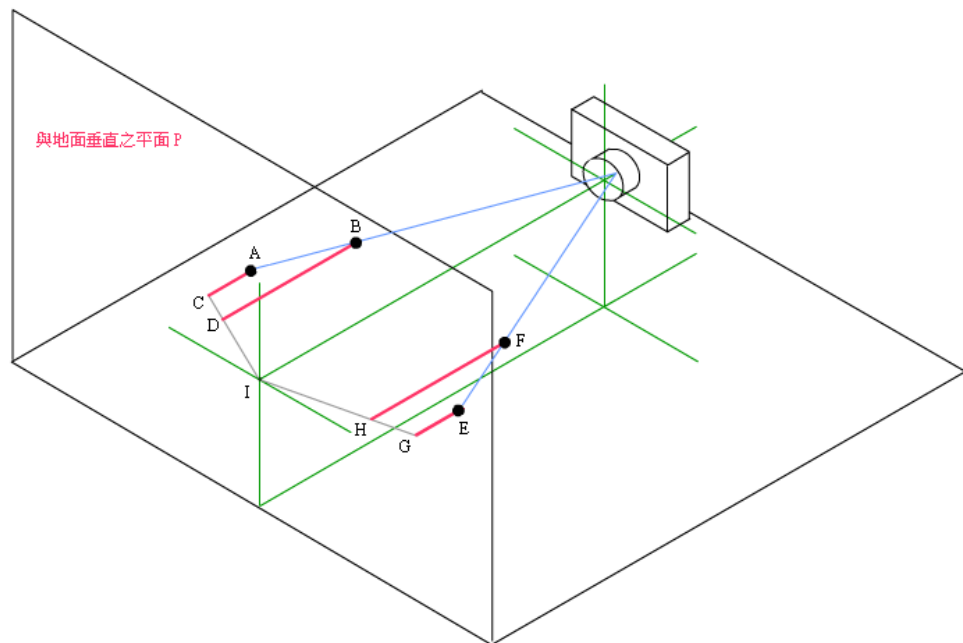


圖 13 尋找相機光軸及影像平面中心點



圖 14-1 具有光班的影像 1



圖 14-2 具有光班的影像 2



圖 14-3 具有光班的影像 3

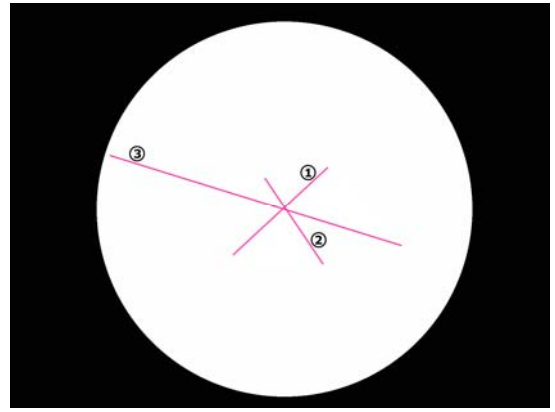


圖 14-4 利用光斑尋找影像中心點

2. 應用

取出影像中的資訊作為量測方面用途的應用已日益普遍，不過，在使用這些資訊之前需將影像的變形失真做好校正[11][12][13][14][15][16][17][18][19]，任何鏡頭由於設計與成本上的考量及製作技術上的瓶頸，或多或少都有光學失真變形的情形，尤其在價格越便宜的鏡頭上更為明顯，由於我們的校正樣板能夠涵蓋的鏡頭視角範圍很大（甚至可以超過 180° ），而且入射光線可以對於整個鏡頭面都做探測（因為取樣點很均勻地分佈在校正樣板內部），所以我們的方法只要經過適當的修改，就可以用來校正任何的鏡頭影像。

四、結語：

為了校正相機鏡頭影像的變形失真，可由一些已知的取樣點資料來推知整個鏡頭的成像方式，可以採用的方法有兩種：一是以往文獻使用的多項式函數曲線擬合法；二是我們所採用插值法。曲線擬合的做法是找到一條光滑的多項式函數曲線和這些已知的取樣點作最佳的擬合，使得每個取樣點與此曲線的距離平方和為最小，其優點是做法相當的簡潔，但是此條光滑曲線不見得會經過每個取樣點，所以其變形失真校正成果的好壞，端賴於尋找出的多項式函數與真實相機系統成像投射函數的擬合程度。

我們採用插值法來改善曲線擬合法無法保證多項式函數一定會經過取樣點的缺點，尤其當校正樣板內佈有更多的取樣點時，其符合真實相機系統投射方式的程度會更為精準。雖然需要建立入射光線與其成像點間之關係對應表，但是可用來取代以往文獻採用的成像點在校正前後位置的對應表，事實上並沒有花費較多的記憶體儲存空間。

五、參考資料：

1. R. Y. Tsai, "A Versatile Camera calibration technique for high-accuracy 3D machine Vision Metrology using off-the shelf TV Cameras and Lenses," IEEE Journal of Robotics and Automation, Vol. 3, No. 4, pp. 323-344, 1987. °
2. J. Weng, P. Cohen, and M. Herniou, "Camera Calibration with Distortion Models and Accuracy Evaluation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 10, pp. 965-980, 1992. °

3. M. M. Fleck, "Perspective Projection : The Wrong Imaging Model," Technique Report 95-01, Computer Science, University of Iowa, 1995. ◦
4. D. E. Stevenson, and M. M. Fleck, "Robot Aerobics: Four Easy Steps to a More Flexible Calibration," International Conf. on Computer Vision, pp. 34-39, 1995. ◦
5. S. S. Beauchemin, R. Bajcsy, and G. Givaty, "A Unified Procedure for Calibrating Intrinsic Parameters of Spherical Lenses," International Conference on Vision Interface '99, pp. 272-279, 1999. ◦
6. S. S. Beauchemin, R. Bajcsy, and G. Givaty, "Modelling and Removing Radial and Tangential Distortions in Spherical Lenses," International Workshop on Theoretical Foundations of Computer Vision Dagstuhl Castle, pp. 1-21, 2000. ◦
7. R. Swaminathan, and S. K. Nayar, "Nonmetric Calibration of Wide-Angle Lenses and Polycameras," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 10, pp. 1172-1178, 2000. ◦
8. James P. Helferty, Chao Zhang, Geoffrey McLennan, and William E. Higgins, "Videoendoscopic Distortion Correction and Its Application to Virtual Guidance of Endoscopy," IEEE Transactions on Medical Imaging, Vol. 20, No. 7, pp. 605-617, 2001. ◦
9. J. Kumler, and M. Bauer, "Fisheye Lens Designs and Their Relative Performance," <http://360vr.com/fisheye41/coastal-fisheyep.pdf>, 2003. ◦
10. L. Ma, Y. Q. Chen, and K. L. Moore, "A New Analytical Radial Distortion Model for Camera Calibration," Report of arXiv.org e-Print Archive arXiv:cs.CV/0307051 http://www.arxiv.org/PS_cache/cs/pdf/0307/0307051.pdf, 2003. ◦
11. 謝瑞斌, "低價位數位相機空間量測實用性之研究 An Investigation on the Usage of a Low Cost Digital Camera in Measuring the Objects' 3D Coordinates," 台灣大學土木工程研究所碩士論文, 2000. ◦
12. 林志鴻, "內視鏡影像之扭曲校正與病兆面積之量測 Correction of Endoscopic Images and Measurement of Lesion Area," 大業大學電機工程學系碩士論文, 2001. ◦
13. 張哲銘, "利用魚眼鏡頭所取得的影像進行半球環場的接圖 An Automatic Registering System for Constructing Hemi-Sphere Panoramic Views Using Fish-Eye Lens," 元智大學資訊工程學系碩士論文, 2001. ◦
14. 簡大淵, "內視鏡影像序列之自動校正重構與病灶量測 Auto-Calibration Reconstruction and Assessment of Clinieal Lesions from Endoscopic Image Sequence," 成功大學資訊工程學系碩士論文, 2002. ◦
15. 林登秋, 江智民, "半球面影像在森林生態研究的應用," 台灣林業科學, Vol. 17, No. 3, pp. 387-400, 2002. ◦
16. 黃添成, "水下雷射掃瞄量測系統 CCD 攝影機之校正 ,Calibration of CCD Camera for Underwater Laser Scanning System," 中山大學海下技術研究所碩士論文, 2002. ◦
17. 劉俊延, "內視鏡影像序列之變形校正與三維重構 3D Calibration and Reconstruction from Endoscopic Image Sequence," 成功大學資訊工程學系碩士論文, 2003. ◦

18. 陳柏安, “利用電腦視覺作自走車之障礙物定位與環境掃描 Obstacle Detection and Environment Scanning by Computer Vision Technique for Automatic Vehicle System,” 成功大學工程科學系碩士論文, 2003.。
19. 羅啓維, “影像處理技術於表面粘著取置機定位檢測 Application of Image Processing Techniques in SMD Locating and Checking as well as SMT Machine Auto-calibration,” 台灣科技大學電機工程學系博士論文, 2003.。

六、附錄：

1. 附錄一：入射光線與影像平面成像點之對應資料

$(\Phi = 0^\circ, \Theta) \rightarrow (x, y)$	$(\Phi = 10^\circ, \Theta) \rightarrow (x, y)$	$(\Phi = 20^\circ, \Theta) \rightarrow (x, y)$	$(\Phi = 30^\circ, \Theta) \rightarrow (x, y)$	$(\Phi = 40^\circ, \Theta) \rightarrow (x, y)$
(0 , 0) --> (199 , 200)	(10 , 0) --> (222 , 200)	(20 , 0) --> (243 , 200)	(30 , 0) --> (265 , 199)	(40 , 0) --> (286 , 199)
(0 , 15) --> (199 , 200)	(10 , 15) --> (221 , 194)	(20 , 15) --> (242 , 188)	(30 , 15) --> (263 , 182)	(40 , 15) --> (283 , 177)
(0 , 30) --> (199 , 200)	(10 , 30) --> (219 , 188)	(20 , 30) --> (237 , 177)	(30 , 30) --> (256 , 167)	(40 , 30) --> (274 , 156)
(0 , 45) --> (199 , 200)	(10 , 45) --> (215 , 184)	(20 , 45) --> (230 , 168)	(30 , 45) --> (245 , 153)	(40 , 45) --> (260 , 138)
(0 , 60) --> (199 , 200)	(10 , 60) --> (210 , 180)	(20 , 60) --> (221 , 161)	(30 , 60) --> (231 , 142)	(40 , 60) --> (242 , 124)
(0 , 75) --> (199 , 200)	(10 , 75) --> (205 , 178)	(20 , 75) --> (210 , 157)	(30 , 75) --> (215 , 136)	(40 , 75) --> (221 , 116)
(0 , 90) --> (199 , 200)	(10 , 90) --> (199 , 177)	(20 , 90) --> (199 , 156)	(30 , 90) --> (198 , 134)	(40 , 90) --> (198 , 113)
(0 , 105) --> (199 , 200)	(10 , 105) --> (193 , 178)	(20 , 105) --> (187 , 156)	(30 , 105) --> (181 , 136)	(40 , 105) --> (176 , 115)
(0 , 120) --> (199 , 200)	(10 , 120) --> (187 , 180)	(20 , 120) --> (176 , 161)	(30 , 120) --> (166 , 143)	(40 , 120) --> (155 , 124)
(0 , 135) --> (199 , 200)	(10 , 135) --> (183 , 184)	(20 , 135) --> (167 , 168)	(30 , 135) --> (152 , 153)	(40 , 135) --> (137 , 138)
(0 , 150) --> (199 , 200)	(10 , 150) --> (179 , 189)	(20 , 150) --> (160 , 177)	(30 , 150) --> (142 , 167)	(40 , 150) --> (123 , 157)
(0 , 165) --> (199 , 200)	(10 , 165) --> (177 , 194)	(20 , 165) --> (156 , 188)	(30 , 165) --> (135 , 183)	(40 , 165) --> (114 , 177)
(0 , 180) --> (199 , 200)	(10 , 180) --> (177 , 200)	(20 , 180) --> (154 , 200)	(30 , 180) --> (133 , 200)	(40 , 180) --> (112 , 200)
(0 , 195) --> (199 , 200)	(10 , 195) --> (177 , 205)	(20 , 195) --> (156 , 211)	(30 , 195) --> (135 , 217)	(40 , 195) --> (115 , 222)
(0 , 210) --> (199 , 200)	(10 , 210) --> (180 , 211)	(20 , 210) --> (161 , 222)	(30 , 210) --> (142 , 233)	(40 , 210) --> (124 , 244)
(0 , 225) --> (199 , 200)	(10 , 225) --> (183 , 216)	(20 , 225) --> (168 , 231)	(30 , 225) --> (152 , 246)	(40 , 225) --> (137 , 261)
(0 , 240) --> (199 , 200)	(10 , 240) --> (188 , 219)	(20 , 240) --> (177 , 238)	(30 , 240) --> (166 , 257)	(40 , 240) --> (156 , 275)
(0 , 255) --> (199 , 200)	(10 , 255) --> (193 , 221)	(20 , 255) --> (188 , 243)	(30 , 255) --> (182 , 263)	(40 , 255) --> (177 , 284)
(0 , 270) --> (199 , 200)	(10 , 270) --> (199 , 222)	(20 , 270) --> (199 , 244)	(30 , 270) --> (199 , 266)	(40 , 270) --> (199 , 287)
(0 , 285) --> (199 , 200)	(10 , 285) --> (205 , 221)	(20 , 285) --> (210 , 242)	(30 , 285) --> (216 , 263)	(40 , 285) --> (222 , 284)
(0 , 300) --> (199 , 200)	(10 , 300) --> (210 , 220)	(20 , 300) --> (221 , 238)	(30 , 300) --> (232 , 267)	(40 , 300) --> (243 , 275)
(0 , 315) --> (199 , 200)	(10 , 315) --> (216 , 216)	(20 , 315) --> (231 , 231)	(30 , 315) --> (246 , 247)	(40 , 315) --> (261 , 262)
(0 , 330) --> (199 , 200)	(10 , 330) --> (219 , 212)	(20 , 330) --> (237 , 222)	(30 , 330) --> (256 , 233)	(40 , 330) --> (274 , 244)
(0 , 345) --> (199 , 200)	(10 , 345) --> (221 , 206)	(20 , 345) --> (242 , 211)	(30 , 345) --> (263 , 217)	(40 , 345) --> (283 , 222)
(0 , 360) --> (199 , 200)	(10 , 360) --> (222 , 200)	(20 , 360) --> (243 , 200)	(30 , 360) --> (265 , 199)	(40 , 360) --> (286 , 199)

$(\Phi = 50^\circ, \Theta) \rightarrow (x, y)$	$(\Phi = 60^\circ, \Theta) \rightarrow (x, y)$	$(\Phi = 70^\circ, \Theta) \rightarrow (x, y)$	$(\Phi = 80^\circ, \Theta) \rightarrow (x, y)$	$(\Phi = 90^\circ, \Theta) \rightarrow (x, y)$
(50 , 0) --> (307 , 199)	(60 , 0) --> (328 , 200)	(70 , 0) --> (350 , 199)	(80 , 0) --> (373 , 200)	(90 , 0) --> (396 , 200)
(50 , 15) --> (303 , 172)	(60 , 15) --> (324 , 166)	(70 , 15) --> (345 , 160)	(80 , 15) --> (367 , 154)	(90 , 15) --> (390 , 148)
(50 , 30) --> (292 , 146)	(60 , 30) --> (311 , 135)	(70 , 30) --> (330 , 124)	(80 , 30) --> (349 , 113)	(90 , 30) --> (369 , 110)
(50 , 45) --> (275 , 123)	(60 , 45) --> (290 , 108)	(70 , 45) --> (305 , 92)	(80 , 45) --> (321 , 76)	(90 , 45) --> (338 , 59)
(50 , 60) --> (253 , 106)	(60 , 60) --> (263 , 88)	(70 , 60) --> (274 , 69)	(80 , 60) --> (285 , 49)	(90 , 60) --> (297 , 28)
(50 , 75) --> (226 , 96)	(60 , 75) --> (232 , 75)	(70 , 75) --> (237 , 54)	(80 , 75) --> (243 , 32)	(90 , 75) --> (249 , 9)
(50 , 90) --> (198 , 92)	(60 , 90) --> (198 , 70)	(70 , 90) --> (198 , 49)	(80 , 90) --> (198 , 26)	(90 , 90) --> (198 , 3)
(50 , 105) --> (170 , 85)	(60 , 105) --> (165 , 75)	(70 , 105) --> (159 , 54)	(80 , 105) --> (153 , 33)	(90 , 105) --> (147 , 10)
(50 , 120) --> (144 , 107)	(60 , 120) --> (134 , 88)	(70 , 120) --> (123 , 69)	(80 , 120) --> (111 , 50)	(90 , 120) --> (100 , 30)
(50 , 135) --> (122 , 124)	(60 , 135) --> (107 , 109)	(70 , 135) --> (91 , 93)	(80 , 135) --> (75 , 78)	(90 , 135) --> (59 , 61)
(50 , 150) --> (105 , 146)	(60 , 150) --> (87 , 136)	(70 , 150) --> (68 , 124)	(80 , 150) --> (49 , 113)	(90 , 150) --> (28 , 101)
(50 , 165) --> (94 , 172)	(60 , 165) --> (74 , 166)	(70 , 165) --> (53 , 161)	(80 , 165) --> (31 , 155)	(90 , 165) --> (9 , 149)
(50 , 180) --> (91 , 200)	(60 , 180) --> (70 , 200)	(70 , 180) --> (48 , 200)	(80 , 180) --> (25 , 200)	(90 , 180) --> (2 , 200)
(50 , 195) --> (94 , 228)	(60 , 195) --> (74 , 233)	(70 , 195) --> (53 , 239)	(80 , 195) --> (31 , 245)	(90 , 195) --> (9 , 250)
(50 , 210) --> (106 , 254)	(60 , 210) --> (87 , 264)	(70 , 210) --> (68 , 275)	(80 , 210) --> (48 , 287)	(90 , 210) --> (28 , 298)
(50 , 225) --> (122 , 276)	(60 , 225) --> (107 , 291)	(70 , 225) --> (92 , 306)	(80 , 225) --> (76 , 322)	(90 , 225) --> (59 , 339)
(50 , 240) --> (145 , 293)	(60 , 240) --> (134 , 312)	(70 , 240) --> (123 , 331)	(80 , 240) --> (112 , 350)	(90 , 240) --> (100 , 371)
(50 , 255) --> (171 , 304)	(60 , 255) --> (165 , 325)	(70 , 255) --> (160 , 346)	(80 , 255) --> (154 , 368)	(90 , 255) --> (147 , 391)
(50 , 270) --> (199 , 308)	(60 , 270) --> (199 , 329)	(70 , 270) --> (199 , 351)	(80 , 270) --> (199 , 374)	(90 , 270) --> (199 , 398)
(50 , 285) --> (227 , 304)	(60 , 285) --> (232 , 325)	(70 , 285) --> (238 , 346)	(80 , 285) --> (244 , 368)	(90 , 285) --> (250 , 391)
(50 , 300) --> (253 , 293)	(60 , 300) --> (264 , 312)	(70 , 300) --> (275 , 331)	(80 , 300) --> (286 , 351)	(90 , 300) --> (298 , 371)
(50 , 315) --> (276 , 277)	(60 , 315) --> (291 , 292)	(70 , 315) --> (306 , 307)	(80 , 315) --> (322 , 324)	(90 , 315) --> (340 , 341)
(50 , 330) --> (293 , 255)	(60 , 330) --> (311 , 265)	(70 , 330) --> (330 , 276)	(80 , 330) --> (350 , 287)	(90 , 330) --> (370 , 299)
(50 , 345) --> (304 , 228)	(60 , 345) --> (324 , 233)	(70 , 345) --> (345 , 239)	(80 , 345) --> (367 , 245)	(90 , 345) --> (390 , 251)
(50 , 360) --> (307 , 199)	(60 , 360) --> (328 , 200)	(70 , 360) --> (350 , 199)	(80 , 360) --> (373 , 200)	(90 , 360) --> (396 , 200)

2. 附錄二：魚眼鏡頭影像之變形校正式碼

```
Imports System.IO
Public Class Form1
    Inherits System.Windows.Forms.Form
    Const Ratio = 3
    Const CenterX = 201
    Const CenterY = 201
    Dim tmpImg1, tmpImg2, tmpImg3 As Bitmap
    Dim MapMatX(90, 360) As Double
    Dim MapMatY(90, 360) As Double

    Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Dim objReaderX As New StreamReader("datax.txt", System.Text.Encoding.ASCII)
        Dim objReaderY As New StreamReader("datay.txt", System.Text.Encoding.ASCII)
```

```

Dim Chararray(8) As Char
Dim s As String
Dim i, j As Integer
i = 0
j = 0
While objReaderX.Peek <> -1
    objReaderX.Read(Chararray, 0, 8)
    s = Chararray
    MapMatX(i, j) = CDbI(s)
    j = j + 1
    If j > 360 Then
        i = i + 1
        j = 0
    End If
End While
objReaderX.Close()
i = 0
j = 0
While objReaderY.Peek <> -1
    objReaderY.Read(Chararray, 0, 8)
    s = Chararray
    MapMatY(i, j) = CDbI(s)
    j = j + 1
    If j > 360 Then
        i = i + 1
        j = 0
    End If
End While
objReaderY.Close()
tmpImg3 = PictureBox1.Image.Clone
End Sub

Private Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim PixelColor, BlackColor As Color
    Dim ImgWidth, ImgHeight As Integer
    Dim FOV, W As Integer
    Dim Setx, Sety, Getx, Gety, i, j As Integer
    Dim Z, A As Integer
    Dim Calx, Caly, CalZ, CalA As Double
    Dim LT, LB, RT, RB, tmpL, tmpR As Double
    Dim Distance, Radius As Double
    FOV = CInt(TextBox1.Text)
    W = CInt(TextBox2.Text)
    tmpImg1 = tmpImg3.Clone
    ImgHeight = Int(tmpImg1.Height / Ratio)
    ImgWidth = Int(tmpImg1.Width / Ratio)
    tmpImg2 = tmpImg1.Clone
    For Setx = 1 To ImgWidth
        For Sety = 1 To ImgHeight
            tmpImg2.SetPixel(Setx - 1, Sety - 1, BlackColor.Black)
        Next
    Next
    For Setx = 1 To ImgWidth
        For Sety = 1 To ImgHeight
            Distance = Math.Sqrt((Setx - CenterX) ^ 2 + (Sety - CenterY) ^ 2)
            If Distance <= W Then
                If Distance = 0 Then
                    CalZ = 0
                    CalA = 0
                Else
                    CalZ = Math.Atan(Math.Tan((FOV * Math.PI / 180) / 2) * Distance / W) * 180 / Math.PI
                    If (Setx - CenterX >= 0) And (Sety - CenterY <= 0) Then

```

```

    CalA = Math.Asin(Math.Abs(Sety - CenterY) / Distance) * 180 / Math.PI
End If
If (Setx - CenterX <= 0) And (Sety - CenterY <= 0) Then
    CalA = (Math.PI - Math.Asin(Math.Abs(Sety - CenterY) / Distance)) * 180 / Math.PI
End If
If (Setx - CenterX <= 0) And (Sety - CenterY >= 0) Then
    CalA = (Math.PI + Math.Asin(Math.Abs(Sety - CenterY) / Distance)) * 180 / Math.PI
End If
If (Setx - CenterX >= 0) And (Sety - CenterY >= 0) Then
    CalA = (2 * Math.PI - Math.Asin(Math.Abs(Sety - CenterY) / Distance)) * 180 / Math.PI
End If
End If
If (CalZ = 0) Or (CalZ = 90) Or (CalA = 0) Or (CalA = 360) Then
    If CalZ = 0 Then
        If CalA = 0 Then
            Calx = MapMatX(0, 0)
            Caly = MapMatY(0, 0)
        Else
            If CalA = 360 Then
                Calx = MapMatX(0, 360)
                Caly = MapMatY(0, 360)
            Else
                tmpL = MapMatX(0, Int(CalA))
                tmpR = MapMatX(0, Int(CalA) + 1)
                Calx = tmpL * (1 - (CalA - Int(CalA))) + tmpR * (CalA - Int(CalA))
                tmpL = MapMatY(0, Int(CalA))
                tmpR = MapMatY(0, Int(CalA) + 1)
                Caly = tmpL * (1 - (CalA - Int(CalA))) + tmpR * (CalA - Int(CalA))
            End If
        End If
    End If
End If
If CalZ = 90 Then
    If CalA = 0 Then
        Calx = MapMatX(90, 0)
        Caly = MapMatY(90, 0)
    Else
        If CalA = 360 Then
            Calx = MapMatX(90, 360)
            Caly = MapMatY(90, 360)
        Else
            tmpL = MapMatX(90, Int(CalA))
            tmpR = MapMatX(90, Int(CalA) + 1)
            Calx = tmpL * (1 - (CalA - Int(CalA))) + tmpR * (CalA - Int(CalA))
            tmpL = MapMatY(90, Int(CalA))
            tmpR = MapMatY(90, Int(CalA) + 1)
            Caly = tmpL * (1 - (CalA - Int(CalA))) + tmpR * (CalA - Int(CalA))
        End If
    End If
End If
If CalA = 0 Then
    If CalZ = 0 Then
        Calx = MapMatX(0, 0)
        Caly = MapMatY(0, 0)
    Else
        If CalZ = 90 Then
            Calx = MapMatX(90, 0)
            Caly = MapMatY(90, 0)
        Else
            tmpL = MapMatX(Int(CalZ), 0)
            tmpR = MapMatX(Int(CalZ) + 1, 0)
            Calx = tmpL * (1 - (CalZ - Int(CalZ))) + tmpR * (CalZ - Int(CalZ))
            tmpL = MapMatY(Int(CalZ), 0)

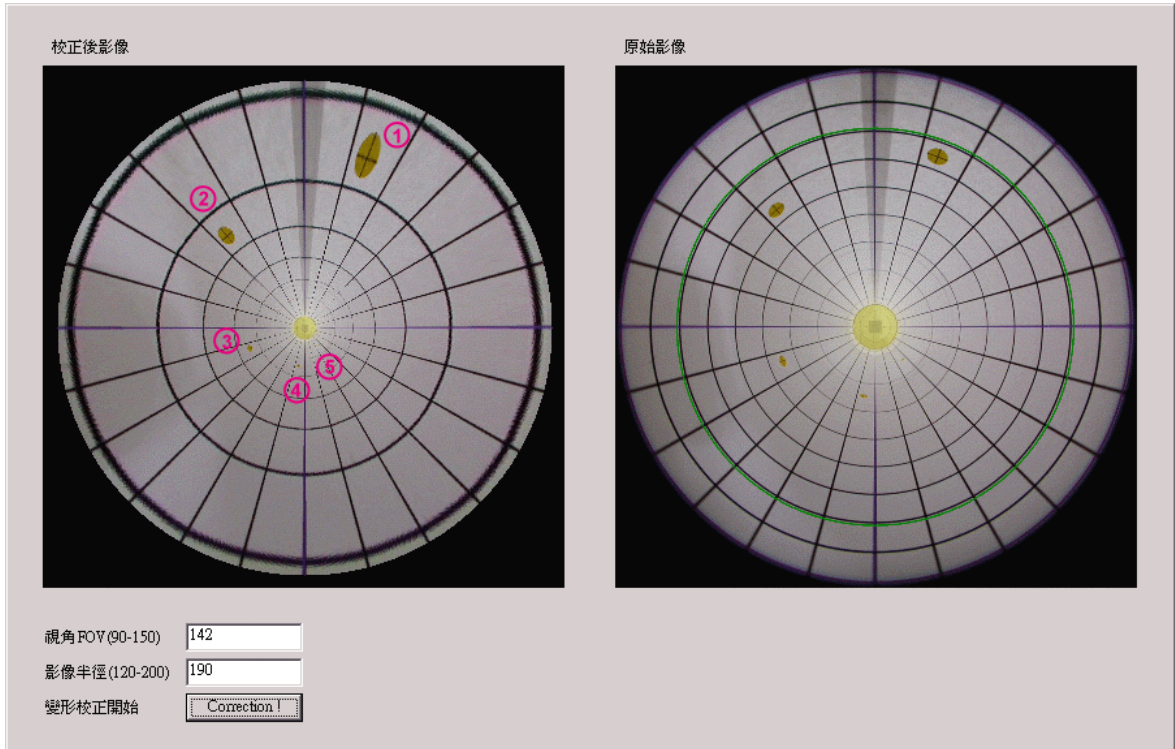
```

```

        tmpR = MapMatY(Int(CalZ) + 1, 0)
        Caly = tmpL * (1 - (CalZ - Int(CalZ))) + tmpR * (CalZ - Int(CalZ))
    End If
End If
End If
If CalA = 360 Then
    If CalZ = 0 Then
        Calx = MapMatX(0, 360)
        Caly = MapMatY(0, 360)
    Else
        If CalZ = 90 Then
            Calx = MapMatX(90, 360)
            Caly = MapMatY(90, 360)
        Else
            tmpL = MapMatX(Int(CalZ), 360)
            tmpR = MapMatX(Int(CalZ) + 1, 360)
            Calx = tmpL * (1 - (CalZ - Int(CalZ))) + tmpR * (CalZ - Int(CalZ))
            tmpL = MapMatY(Int(CalZ), 360)
            tmpR = MapMatY(Int(CalZ) + 1, 360)
            Caly = tmpL * (1 - (CalZ - Int(CalZ))) + tmpR * (CalZ - Int(CalZ))
        End If
    End If
End If
Getx = Int(Ratio * Calx)
Gety = Int(Ratio * Caly)
Else
    LT = MapMatX(Int(CalZ), Int(CalA))
    LB = MapMatX(Int(CalZ) + 1, Int(CalA))
    RT = MapMatX(Int(CalZ), Int(CalA) + 1)
    RB = MapMatX(Int(CalZ) + 1, Int(CalA) + 1)
    tmpL = LT * (1 - (CalZ - Int(CalZ))) + LB * (CalZ - Int(CalZ))
    tmpR = RT * (1 - (CalZ - Int(CalZ))) + RB * (CalZ - Int(CalZ))
    Calx = tmpL * (1 - (CalA - Int(CalA))) + tmpR * (CalA - Int(CalA))
    LT = MapMatY(Int(CalZ), Int(CalA))
    LB = MapMatY(Int(CalZ) + 1, Int(CalA))
    RT = MapMatY(Int(CalZ), Int(CalA) + 1)
    RB = MapMatY(Int(CalZ) + 1, Int(CalA) + 1)
    tmpL = LT * (1 - (CalZ - Int(CalZ))) + LB * (CalZ - Int(CalZ))
    tmpR = RT * (1 - (CalZ - Int(CalZ))) + RB * (CalZ - Int(CalZ))
    Caly = tmpL * (1 - (CalA - Int(CalA))) + tmpR * (CalA - Int(CalA))
    Getx = Int(Ratio * Calx)
    Gety = Int(Ratio * Caly)
End If
PixelColor = tmpImg1.GetPixel(Getx - 1, Gety - 1)
tmpImg2.SetPixel(Setx - 1, Sety - 1, PixelColor)
End If
Next
Next
Radius = Ratio * Math.Sqrt((MapMatX(Int(FOV / 2), 0) - CenterX) ^ 2 + (MapMatY(Int(FOV / 2), 0) - CenterY) ^ 2)
ImgHeight = tmpImg1.Height
ImgWidth = tmpImg1.Width
For Setx = 1 To ImgWidth
    For Sety = 1 To ImgHeight
        Distance = Math.Sqrt((Setx - Ratio * CenterX) ^ 2 + (Sety - Ratio * CenterY) ^ 2)
        If (Distance >= Radius + 1) And (Distance <= Radius + 5) Then
            tmpImg1.SetPixel(Setx - 1, Sety - 1, BlackColor.White)
        End If
    Next
Next
PictureBox2.Image = tmpImg2
PictureBox1.Image = tmpImg1
End Sub
End Class

```

3. 附錄三：魚眼鏡頭影像之變形校正成果檢驗



控制點編號	1	2	3	4	5
控制點所代表的入射光線 (Φ, Θ)	$(65^\circ, 70^\circ)$	$(55^\circ, 130^\circ)$	$(35^\circ, 200^\circ)$	$(25^\circ, 260^\circ)$	$(15^\circ, 310^\circ)$
完全無失真的位置 (x_p, y_p)	(249.0, 69.2)	(140.9, 129.4)	(158.0, 216.7)	(195.7, 231.0)	(212.3, 214.4)
校正後所在的位置 (x', y')	(249, 70)	(141, 130)	(159, 217)	(196, 230)	(212, 214)
誤差距離 Pixels	0.84	0.58	1.10	1.08	0.51

控制點編號	1	2	3	4	5
控制點所代表的入射光線 (Φ, Θ)	(75°,25°)	(67°,335°)	(65°,70°)	(62°,290°)	(55°,110°)
完全無失真的位置 (x_p, y_p)	(361.2,126.3)	(302.1,248.2)	(235.7,105.5)	(231.5,284.7)	(177.9,137.4)
校正後所在的位置 (x', y')	(249,70)	(141,130)	(159,217)	(196,230)	(212,214)
誤差距離 Pixels	0.84	0.58	1.10	1.08	0.51

控制點編號	6	7	8	9	10
控制點所代表的入射光線 (Φ, Θ)	(53°,260°)	(45°,160°)	(18°,190°)	(35°,215°)	(27°,35°)
完全無失真的位置 (x_p, y_p)	(190.1,262.9)	(156.5,184.8)	(185.8,203.7)	(173.8,220.0)	(220.8,187.2)
校正後所在的位置 (x', y')	(249,70)	(141,130)	(159,217)	(196,230)	(212,214)
誤差距離 Pixels	0.84	0.58	1.10	1.08	0.51