

臺灣二〇〇八年國際科學展覽會

科 別：電腦科學

作品名稱：倒車攝影機影像之變形校正

得獎獎項：佳作

學校 / 作者：新竹市立光華國民中學
 新竹市立光華國民中學

陳彥豪
劉育全

作者簡介



我是光華國中三年級的劉育全。我的興趣是下棋、游泳、溜冰，也對數理方面有著相當程度的好奇心。我的家裡有一個媽媽、一個爸爸和一個哥哥。爸爸是證券交易所的營業員，媽媽是營造公司的業務員，哥哥現在則是中正理工學院的學生。在一個偶然的機會下，看到了到車攝影機拍攝時的變形，便想邀請我的同學陳彥豪和我一起研究，並製作成科展。

作者簡介



我是陳彥豪，現在是光華國中三年級的學生。我的興趣是算一些數學的題目和打籃球，空閒的時候便算一下，我覺得有益頭腦發展。家中有爸爸、媽媽及一個弟弟，爸爸是資深工程師，媽媽則是高中的英文老師。我是在二年級時，受同學劉育全的邀請，發現科展好像十分有趣，因此步入科展的研究領域。歷經一年的時間，或許遇上許多瓶頸，但我們都一一克服，希望這篇報告能在國際科展的舞臺上，發光發熱。

Summary

Compared with the traditional lens, wide-angle lens can make images of larger range, so they are widely used in a lot of places, such as move backward cine-cameras, laparoscopes, and burglarproof eyes. The drawback is that the images distort a lot and may cause the illusion to human eyes and lead to erroneous judgments. This reserch takes the backward cine-camera as example and utilizes multinomial function to correct the deformation and distortion of the image. The results show that the method we use are quite effective and the model can be applied to other equipment with wide-angle lens.

摘要

與傳統的鏡頭比較起來，廣角鏡頭可取得較大範圍的影像，所以廣泛地運用在很多地方，例如倒車攝影機、內視鏡、防盜警眼等，但是其影像會有很大的變形失真，容易造成人類眼睛的錯覺與誤判。本篇研究以倒車攝影機為例，使用了多項式函數來校正其影像的變形與失真，從程式實作的校正結果看來，我們的方法之效果還不錯，相信同樣的模式可用來校正其他使用廣角鏡頭的儀器設備之變形影像。

壹、研究動機與目的：

老爸的車上裝有倒車攝影機，每當倒車的時候，駕駛座旁的液晶螢幕就會顯示出車子後方的影像，可是原本是直線條的地磚接縫，在倒車攝影機的拍攝下，很明顯地變彎曲了，而且越靠近影像的周邊，就變形的越嚴重。經過尋找資料後，發現造成這種影像彎曲變形的元兇就是倒車攝影機使用了廣角鏡頭。

與傳統的鏡頭比較起來，廣角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、觀測監視等，由於廣角鏡頭會造成影像有不小的變形失真，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，一套良好的廣角鏡頭影像變形失真校正技術是非常重要的。我們希望藉此研究，尋找出造成倒車攝影機變形的模式，並進而發展出其校正的方法。



圖 1-1 倒車攝影系統



圖 1-2 倒車攝影所拍攝的地磚

貳、研究設備器材及軟體：

- 一、個人電腦、筆記型電腦
- 二、Ulead PhotoImpact
- 三、Visual Basic.Net
- 四、Microsoft Excel
- 五、倒車攝影機 HPB（合盈光電）VN-2105
- 六、天文望遠鏡腳架

參、研究過程與方法：

一、無失真的針孔成像機制

如圖 2 所示，若三維空間中的一點 $P(X, Y, Z)$ 與其影像平面上的成像點 $p(x, y)$ 的關係式如下，則是符合無失真的透視投影成像方式

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

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

其中， X 、 Y 、 Z 三軸為鏡頭的三維空間直角座標， C 點為相機鏡頭的中心點， x 、 y 兩軸構成影像平面的二維平面直角座標， 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鏡頭之焦距 f 。

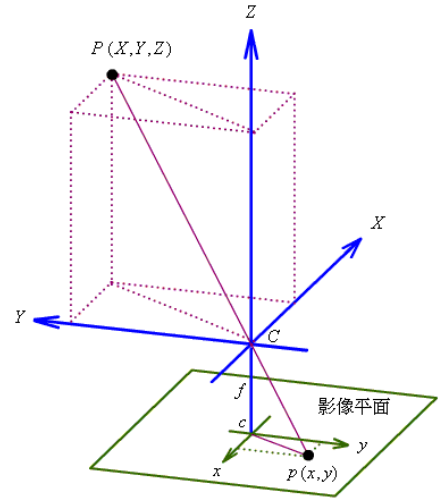


圖 2 無失真的針孔成像機制

二、鏡頭的種類

在針孔成像的模型中，似乎可以很輕易地藉著調整焦距的大小來改變鏡頭的視角，並產生無失真的影像。但是在真實的鏡頭設計過程中，為了克服曝光量、色差、球面差、像場畸變等問題，製作出來的鏡頭結構是相當地複雜，通常多由好幾片透鏡組合而成，因為成本上的考量及鏡頭製作技術上的瓶頸，任何鏡頭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。若以鏡頭焦距的長短來分類，有望遠鏡頭、標準鏡頭、廣角鏡頭等三大類，分別介紹如下：

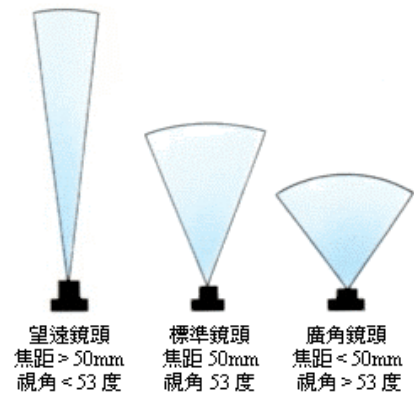


圖 3 無失真的針孔成像機制

1. 望遠鏡頭

望遠鏡頭為焦距 50mm 以上的長焦距鏡頭，能夠拉近遠距離的景物，會壓縮空間感，視角較小，景深也會變短，適合拍攝遠距離的大自然景觀，自然生態的動植物及人物的特寫鏡頭。

2. 標準鏡頭

標準鏡頭的焦距為 50mm，其呈現的影像與肉眼所見的景物相似（視角為 53 度），因為最大光圈值較大，所以適合拍攝一般風景或人物肖像照片。

3. 廣角鏡頭

廣角鏡頭為焦距 50mm 以下的短焦距鏡頭，其拍攝視野較寬（即視角較大），

從觀景窗看到的景物呈現於照片上會變小，不過景深較深長。適合拍攝寬闊的全景、風景照、俯瞰景觀等景物。

三、倒車攝影機影像之變形與校正

廣角鏡頭的成像方式，常用的有立體投影、等距投影、等角投影、垂直投影等投射法，但是，到目前為止，無論製作技術有多麼地精良，廣角鏡頭都會有光學失真的情形，倒車攝影機採用了廣角鏡頭而取得較大範圍的影像，當然也無可避免的會有影像變形的缺點，為了減少變形影像所引起的駕駛人員之誤判，一套良好的廣角鏡頭影像變形校正方法是非常重要的，我們打算先設計些校正樣板，然後，以倒車攝影機來拍攝這些校正樣板，希望能夠找到倒車攝影機影像之變形模式，並進而發展出其變形校正的方法。

1. 校正樣板的製作

校正樣板可用來尋找三維空間座標與影像平面二維座標間之對應關係，圖 4-1 的網狀格子點是最直覺的選擇，經過倒車攝影機的拍攝後，我們發現了一些變形的特性，於是又設計了圖 4-2 與圖 4-3 的校正樣板來驗證這些特性。

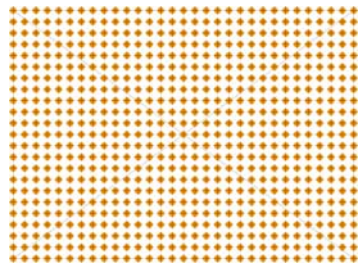


圖 4-1 校正樣板 1

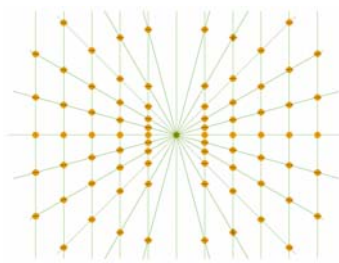


圖 4-2 校正樣板 2

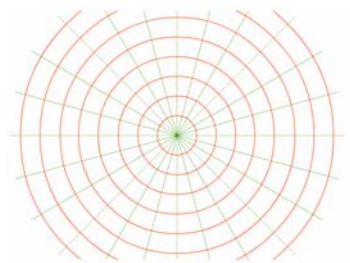


圖 4-3 校正樣板 3

2. 拍攝設備與裝置

整個拍攝設備與裝置如圖 5 所示，所使用的器材與軟體有：倒車攝影機、天文望遠鏡腳架（可微調赤經與赤緯角度）、校正樣板、水平儀、鉛垂線、筆記型電腦、影像擷取器、*Ulead PhotoImpact*，所拍攝出來的三個校正樣板影像分別為圖 6-1、圖 6-2 與圖 6-3。在正式拍攝前有兩件事情是必須特別注意的地方，第一

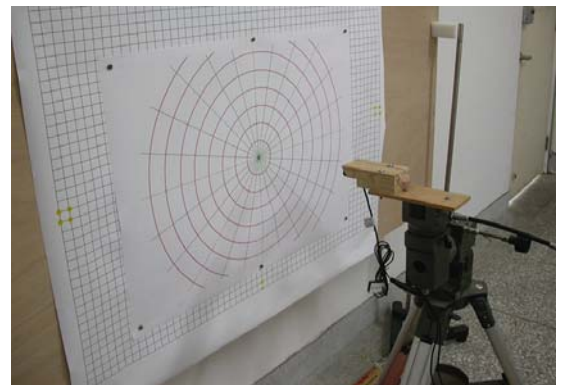


圖 5 拍攝裝置圖

，置放校正樣板的腳架平台以及相機機身必須做好水平與垂直調校；第二，倒車攝影機的中心點須與校正樣板的中心點重合。

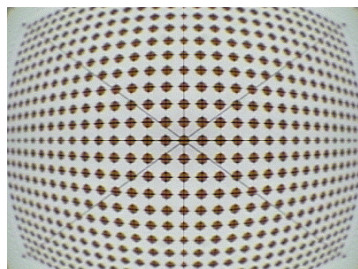


圖 6-1 校正樣板 1 的影像

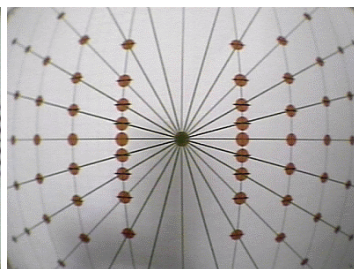


圖 6-2 校正樣板 2 的影像

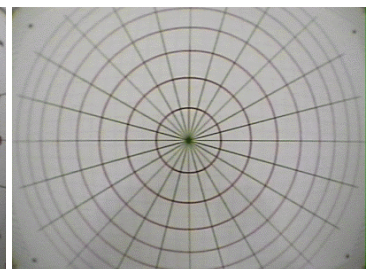


圖 6-3 校正樣板 3 的影像

3. 變形特性

觀察圖 6-1 後我們可以發現，如圖 7-1 所示，校正樣板 1 上通過中心點的直線，經過倒車攝影機成像後仍為通過中心點的直線，其他沒有通過中心點的直線則會彎曲，而且離中心點越遠彎曲的越嚴重。由圖 7-2 中可以了解造成直線彎曲的原因，校正樣板 2 中同一直線上的點，經過倒車攝影機成像後，離中心點越遠的點，其距離被縮減的越多。

校正樣板 3 是由等間隔距離的同心圓及等夾角的直線所構成，我們可從圖 7-3 中發現，間隔距離相等的同心圓經過倒車攝影機成像後仍為同心圓，但是間隔距離有所改變，離中心點越近，間隔越大，離中心點越遠，則間隔距離越小；從圖 7-4 中發現，通過中心點等夾角的直線經過倒車攝影機成像後仍為通過中心點的直線，且此直線與水平軸的夾角維持不變。

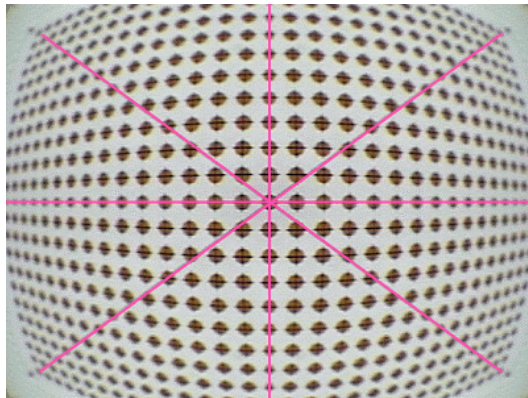


圖 7-1 通過中心點的直線

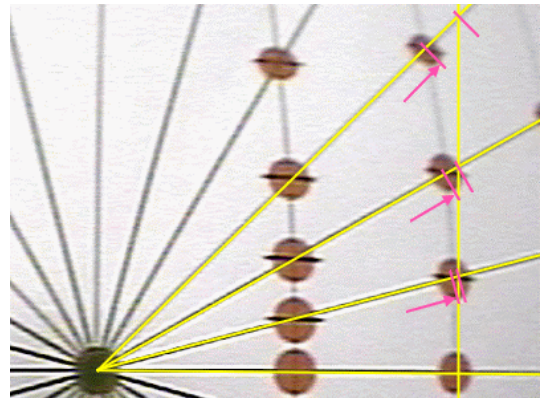


圖 7-2 離中心點越遠距離縮減越多

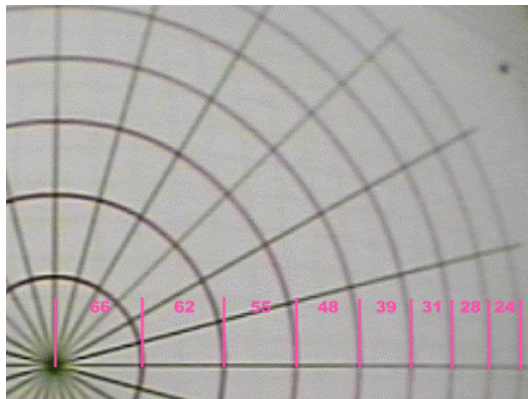


圖 7-3 同心圓間隔距離的改變情形

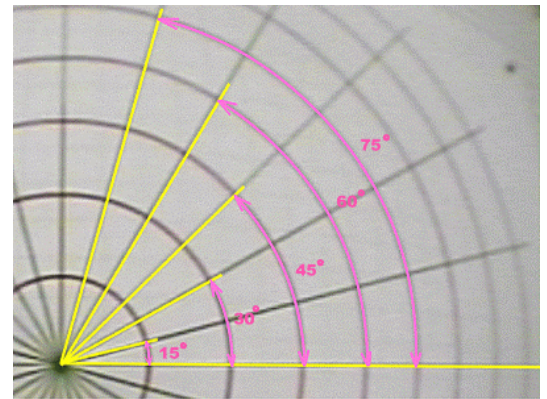


圖 7-4 直線與水平軸的夾角維持不變

4. 校正方法的構想

由於原始影像中各點與中心點的連線與水平軸的夾角經過倒車攝影機成像後仍維持不變，僅是與中心點的距離有所改變，所以極座標系統比較適合來表示倒車攝影機的物與像關係函數。 P 點為原始影像的某一點，在以中心點 O 點為極的極座標系統中其座標為 (r, θ) ，經過倒車攝影機成像後映射至 $P'(r', \theta)$ ，現在我們的首要工作就是要找出 $r' = f(r)$ 成像函數，有了 $f(r)$ 成像函數後，我們就可以利用其反函數 $f^{-1}(r')$ 來校正倒車攝影機所拍攝的變形影像，整個倒車攝影機變形影像校正構想如圖 8 所示。

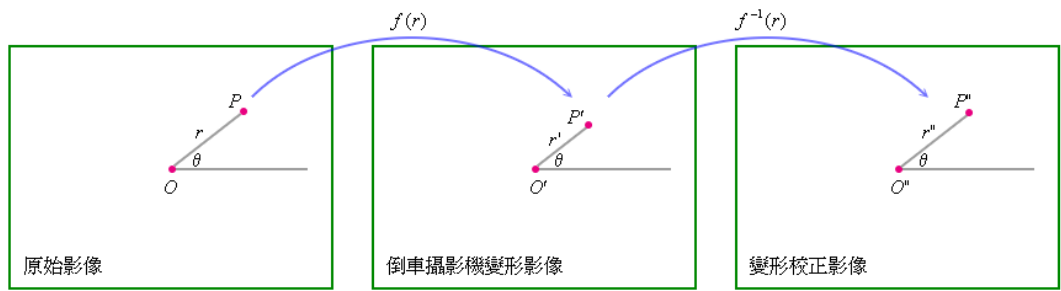


圖 8 校正倒車攝影機變形影像構想

四、倒車攝影機變形影像之校正程序

1. 尋找成像函數

我們可由圖 7-3 看出等間隔距離的同心圓在倒車攝影機拍攝後間隔距離的改變情形，這種間隔距離的改變，是否遵循著某種數學函數呢？我們嘗試著繪製出其函數圖形，以便了解倒車攝影機鏡頭的成像方式，由圖 9 看來， n 階多項式函數似乎可以滿足需求

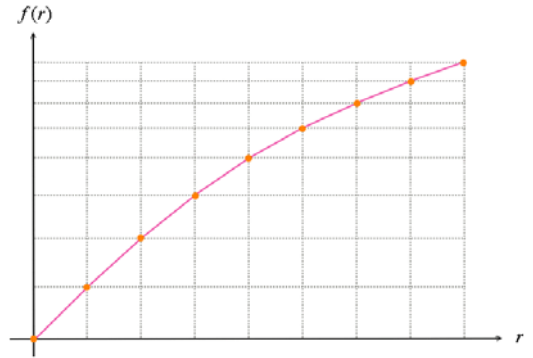


圖 9 同心圓間隔距離的改變情形

$$f(r) = a_1 r + a_2 r^2 + \dots + a_n r^n$$

為了求得 n 階多項式函數的各項係數，必須利用校正樣板 2 上的 18 個取樣點及其成像點當作已知資料，然後再使用 *Microsoft Excel* 的趨勢線來尋找成像函數，由取樣點的平均誤差看來，較高階的多項式函數會比較精準。

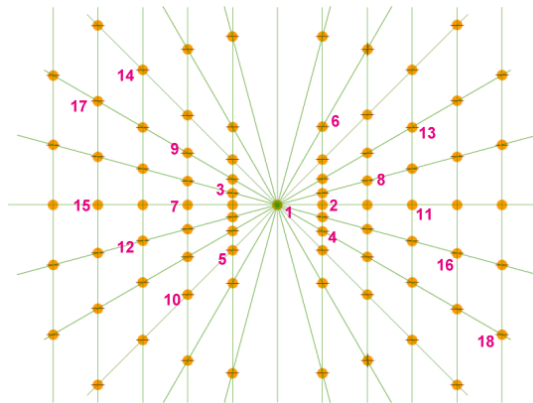


圖 10-1 校正樣板 2 上的取樣點

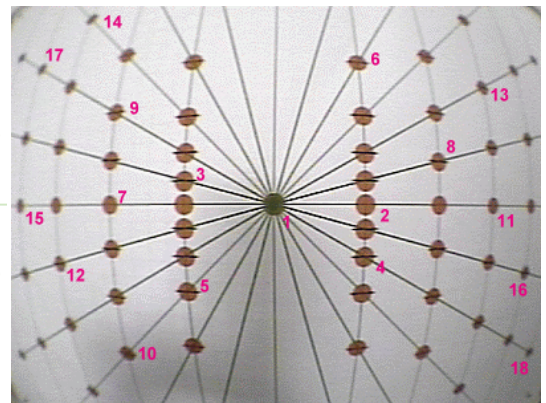


圖 10-2 各取樣點的成像點

取樣點編號	取樣點位置	取樣點與中心點距離 r	成像點位置	成像點與中心點距離 $f(r)$
1	(0, 0)	0.00	(0, 0)	0.00
2	(45, 0)	45.00	(68, 0)	68.00
3	(-45, 12)	46.57	(-68, 39)	78.39
4	(45, -26)	51.97	(67, -39)	77.52
5	(-45, -45)	63.64	(-66, -66)	93.34
6	(45, 79)	90.92	(62, 107)	123.66
7	(-91, 0)	91.00	(-125, 0)	125.00
8	(91, 25)	94.37	(123, 33)	127.35
9	(-90, 53)	104.45	(-121, 69)	139.29

取樣點編號	取樣點位置	取樣點與中心點距離 r	成像點位置	成像點與中心點距離 $f(r)$
10	(-91, -90)	127.99	(-112, -113)	159.10
11	(136, 0)	136.00	(165, -1)	165.00
12	(-136, -36)	140.68	(-163, -44)	168.83
13	(136, 78)	156.78	(156, 88)	179.11
14	(-136, 136)	192.33	(-138, 140)	196.58
15	(-181, 0)	181.00	(-193, -1)	193.00
16	(181, -48)	187.26	(188, -52)	195.06
17	(-181, 105)	209.25	(-177, 102)	204.29
18	(226, -130)	260.72	(192, -111)	221.78

n	多項式成像函數 $f(r)$	取樣點平均誤差
2	$1.6410r - 3.1177 \times 10^{-3} r^2$	1.93
3	$1.7679r - 4.7435 \times 10^{-3} r^2 + 4.6752 \times 10^{-6} r^3$	1.20
4	$1.7396r - 4.1129 \times 10^{-3} r^2 + 5.5070 \times 10^{-7} r^3 + 8.1909 \times 10^{-9} r^4$	1.08
5	$1.8987r - 9.5450 \times 10^{-3} r^2 + 6.1943 \times 10^{-5} r^3 - 2.7186 \times 10^{-7} r^4 + 4.4327 \times 10^{-10} r^5$	1.00

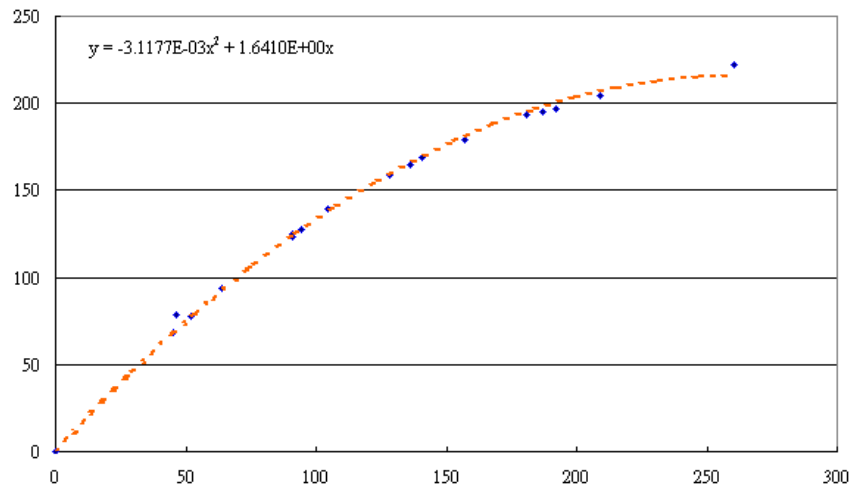


圖 11-1 二階多項式函數

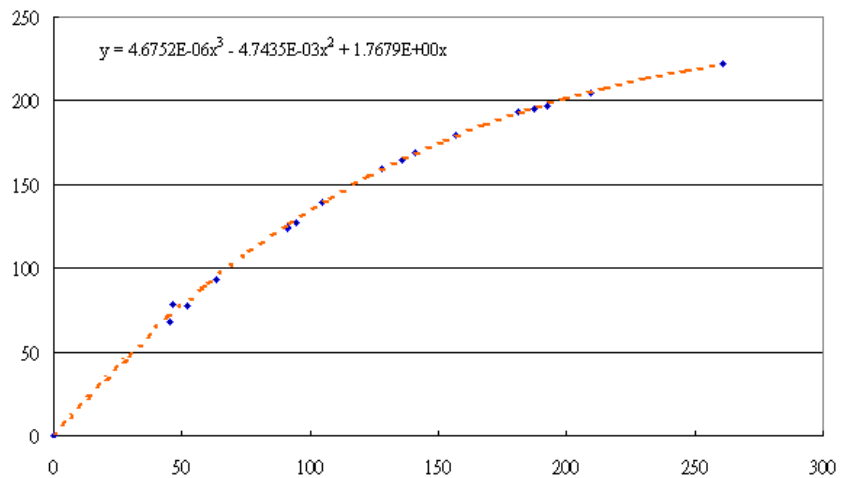


圖 11-2 三階多項式函數

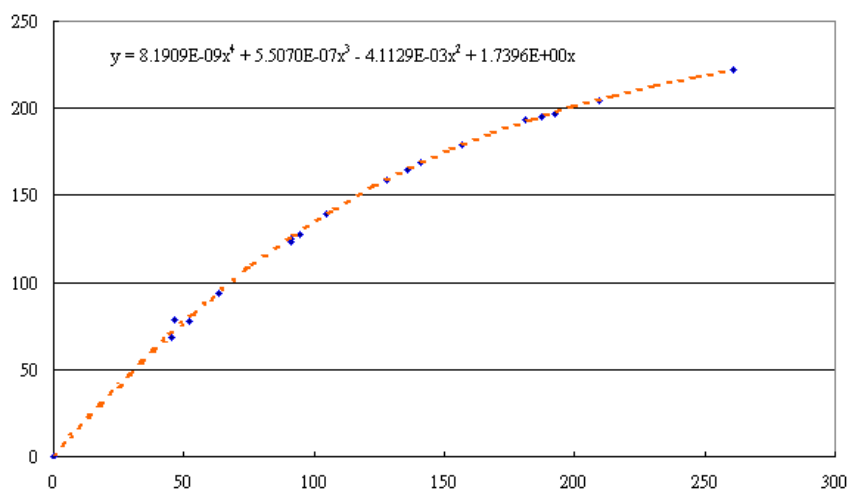


圖 11-3 四階多項式函數

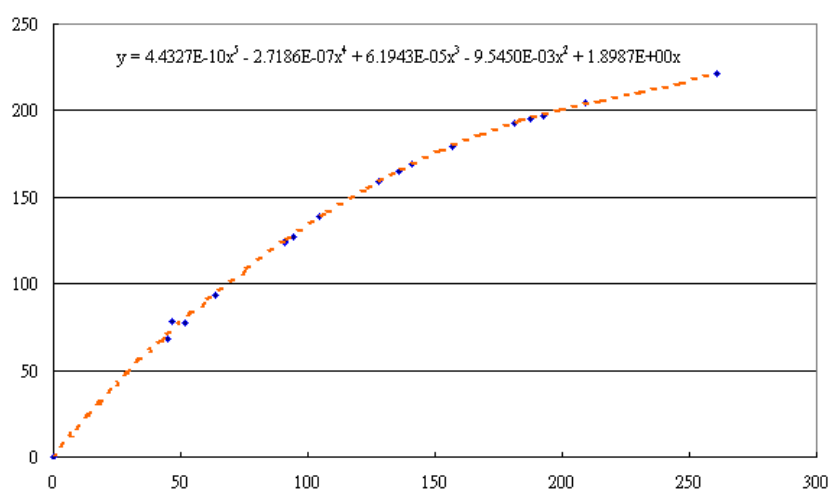


圖 11-4 五階多項式函數

2. 校正步驟

有了 $f(r)$ 成像函數後，我們就可以如圖 8 所示，利用其反函數 $f^{-1}(r)$ 來校正倒車攝影機所拍攝的變形影像，不過這樣的校正方式會有個缺點，當變形影像上的每一個像素映射到校正後影像後，可能有些校正後影像上的像素沒有被映射到而會形成一些空洞，為了避免這種現象，我們將倒車攝影機變形影像校正方式做了些修正，整個變形校正步驟如下：

- 建立一個新的影像空間用來存放變形校正後之影像。
- 針對變形校正後影像上的每一點 $p(x, y)$ 的位置，推算出其極座標 $p(r, \theta)$ ，推算的方式如下：

若是 $x^2 + y^2 = 0$ ，則

$$r = 0 ; \theta = 0^\circ$$

若是 $x^2 + y^2 \neq 0$ ，則

$$r = \sqrt{x^2 + y^2}$$

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

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

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

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

- 利用成像函數 $f(r)$ ，算出 p 點在變形影像上的成像點 $p'(r', \theta')$ ，其中
 - $r' = f(r)$
 - $\theta' = \theta$
- 將變形影像上的成像點 $p'(r', \theta')$ 轉換成平面直角座標 $p'(x', y')$ ，其中
 - $x' = r' \cos \theta'$
 - $y' = r' \sin \theta'$
- 將變形影像上的 $p'(x', y')$ 取出並填入變形校正後影像的 $p(x, y)$ 。
- 當所有變形校正後影像上的點均填入後，就完成了倒車攝影機影像之變形校正。

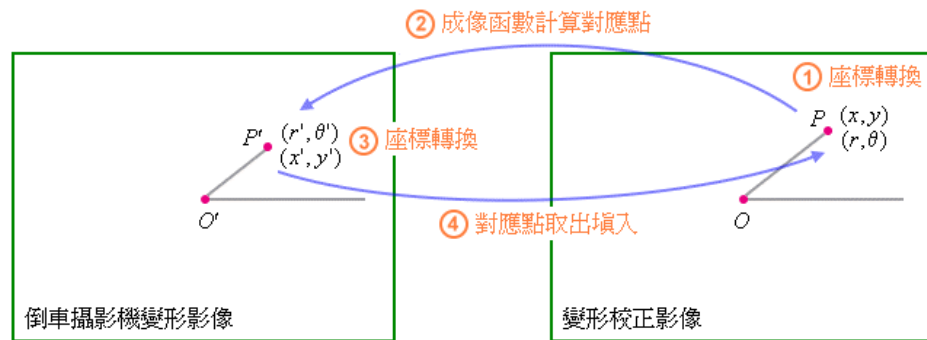


圖 12 修正後的倒車攝影機變形影像校正方法

肆、研究結果：

一、單張影像的變形校正

我們以 Visual Basic.Net 來實作單張影像的變形校正文式，在表單的左邊插入了一個 *PictureBox1*（原始影像）及一個 *Button1*（讀入原始影像）按鈕，用來載入並顯示倒車攝影機所拍攝的單張變形影像，表單的右邊則插入了 *PictureBox2*（校正後影像）、*TextBox1*（顯示範圍）、*TextBox2*（花費時間）及一個 *Button2*（開始校正）按鈕，由 *TextBox1* 輸入變形影像的校正範圍後，按下按鈕開始校正，*PictureBox2* 就用來存放校正後的影像，並於 *TextBox2* 顯示電腦運算處理的時間。圖 13-1 與圖 13-2 分別為利用三階與四階多項式函數校正的情形，由校正後的影像看來，彎曲的圓柱與地磚接縫都變直了，校正的成果還不錯。



圖 13-1 利用三階多項式函數校正單張倒車攝影機變形影像



圖 13-2 利用四階多項式函數校正單張倒車攝影機變形影像

二、即時影像的變形校正

在網路上找到了一個使用 Windows API (應用程式設計介面) `avicap32.dll` 與 `user32.dll` 寫的 USB 介面視訊擷取 Visual Basic.Net 程式，將其稍作修改並套用四階多項式校正函數而完成了變形影像的即時校正式，為了方便使用者的操作，加入了兩個 *TrackBar*，分別可用來調整校正後影像的亮度以及顯示視角。程式執行後，發現視訊影像會有慢動作般的格放現象，為了改善這種現象，我們先利用校正函數建立起變形影像與校正後影像每個像素間的對應表，再利用查表法來取代影像中每個像素在校正時的多項式的運算，如此可以節省大量的運算時間，由修正後的即時校正式之執行情形看來 (圖 14-1 與圖 14-2)，視訊影像的慢動作格放現象確實有獲得改善。



圖 14-1 視角 150 度時變形影像的即時校正



圖 14-2 視角 84 度時變形影像的即時校正

伍、結論：

廣角鏡頭由於視角比普通鏡頭還大，所以可取得較大範圍的畫面，但底片大小是固定的，因此必須將影像壓縮，就容易造成光學失真的現象。拍攝物（樣板）通過中心點的直線，經過倒車攝影機成像後會仍為通過中心點的直線，且此直線與水平軸的夾角維持不變，只有距離會產生改變，距離中心點越遠時，直線變形的曲率就會越大，致使圖片變的像球體。

我們校正時所使用的校正多項式函數越高階時，校正的效果就越好，平均誤差也越小。校正後影像的上下左右邊會有白色的空洞，其原因是廣角鏡頭所能拍攝的畫面範圍本身就不是矩形，為了填滿矩形而產生了變形影像，因此將之校正還原後，

就會在影像的上下左右邊產生白色的空洞。

校正程式方面，不同的廣角鏡頭之成像特性與視角都會有所不同，因此變形校正函數也有所不同，但也都可以藉著同樣的模式，找到取樣點在拍攝前和拍攝後的座標改變情形，代入 *Microsoft Excel* 軟體中算出其多項式校正函數。我們所實作出來的即時影像變形校正程式，可以將倒車攝影機所拍攝出來的視訊做及時校正並呈現於螢幕上。

陸、未來展望：

當我們將倒車攝影機的視訊做即時校正時，無論如何改善程式，都還是會有慢動作格放的現象，或許是因為我們對於應用程式設計介面 *avicap32.dll* 了解的還不夠透徹，是未來值得努力的地方，假如真的要在車輛上安裝具有變形校正功能的倒車攝影機，必須將校正程式製作成硬體，那麼在校正處理上應該會快很多。此外，由於車輛上的倒車攝影機都是固定式的，我們若是能在顯示校正後影像的螢幕上同時也顯示類似尺規的輔助線，那麼對於駕駛人在距離的估測上會更有幫助。

在製作程式時總是遇到很多難題不曉得如何去解決，三不五時就必須去請教他人，尤其是在製作即時影像變形校正的程式時總是遇到困難，讓我們深深覺得要做好一個研究，不論是哪一個科別的，都需要電腦程式設計來輔助，當我們在做這一個科展專題時，若我們本身就懂得如何去製作程式，那麼不但不需要麻煩他人，也可以省下更多的時間。做完這次的科展專題，令我們獲益良多，不但對攝影機、相機這類的物品，了解的更透徹，並且在程式製作的方面有很大的進步，甚至作出了一個有用的程式呢！

柒、參考資料：

- 一、國中自然與生活科技
- 二、鍾國亮，“影像處理與電腦視覺”，台灣東華書局股份有限公司，2006。
- 三、彭明柳，“Visual Basic.NET 中文專業版徹底研究”，博碩文化股份有限公司，2004。
- 四、張隱泉等著，“攝影原理與實用”，東方圖書公司，第 2 章，1963。
- 五、<http://andrew.csie.ncyu.edu.tw/Doc3A.htm> 國立嘉義大學 資訊工程學系 洪燕竹教授網站。

捌、附錄：即時影像的變形校正式

```
Imports System.Runtime.InteropServices
Public Class Form1
    Inherits System.Windows.Forms.Form
    '----- 視訊部分 Code By Ken Tucker -----
    Const WM_CAP As Short = &H400S
    Const WM_CAP_DRIVER_CONNECT As Integer = WM_CAP + 10
    Const WM_CAP_DRIVER_DISCONNECT As Integer = WM_CAP + 11
    Const WM_CAP_EDIT_COPY As Integer = WM_CAP + 30
    Const WM_CAP_SET_PREVIEW As Integer = WM_CAP + 50
    Const WM_CAP_SET_PREVIEWRATE As Integer = WM_CAP + 52
    Const WM_CAP_SET_SCALE As Integer = WM_CAP + 53
    Const WS_CHILD As Integer = &H40000000
    Const WS_VISIBLE As Integer = &H10000000
    Const SWP_NOMOVE As Short = &H2S
    Const SWP_NOSIZE As Short = 1
    Const SWP_NOZORDER As Short = &H4S
    Const HWND_BOTTOM As Short = 1
    Const TV_FIRST = &H1100
    Const TVM_SETBKCOLOR = TV_FIRST + 29
    Dim iDevice As Integer = 0 'Current device ID
    Dim hHwnd As Integer 'Handle to preview window
    Declare Function SendMessage Lib "user32" Alias "SendMessageA" _
        (ByVal hwnd As Integer, ByVal wParam As Integer, ByVal lParam As Object) As Integer
    Declare Function SetWindowPos Lib "user32" Alias "SetWindowPos" (ByVal hwnd As Integer, _
        ByVal hWndInsertAfter As Integer, ByVal x As Integer, ByVal y As Integer, _
        ByVal cx As Integer, ByVal cy As Integer, ByVal wFlags As Integer) As Integer
    Declare Function DestroyWindow Lib "user32" (ByVal hwnd As Integer) As Boolean
    Declare Function capCreateCaptureWindowA Lib "avicap32.dll" _
        (ByVal lpszWindowName As String, ByVal dwStyle As Integer, _
        ByVal x As Integer, ByVal y As Integer, ByVal nWidth As Integer, _
        ByVal nHeight As Short, ByVal hWndParent As Integer, _
        ByVal nID As Integer) As Integer
    Declare Function capGetDriverDescriptionA Lib "avicap32.dll" (ByVal wDriver As Short, _
        ByVal lpszName As String, ByVal cbName As Integer, ByVal lpszVer As String, _
        ByVal cbVer As Integer) As Boolean
    Declare Function CreateCompatibleDC Lib "gdi32" Alias "CreateCompatibleDC" (ByVal hdc As Long) As Long
    Declare Function CreateCompatibleBitmap Lib "gdi32" Alias "CreateCompatibleBitmap" (ByVal hdc As Long, _
        ByVal nWidth As Long, ByVal nHeight As Long) As Long
    Dim ImgData As IDataObject
    Dim PreviewImg As Image

    '----- 我們的變數宣告 -----
    Const a5 = 0
    Const a4 = 0.4723
    Const a3 = 0.0926
    Const a2 = -1.9768
    Const a1 = 2.412
    Const Interval = 66 '每隔 0.066 秒取像一次
    Const ImgWidth = 320
    Const ImgHeight = 240
    Const MaxVal = 255
    Const MinVal = 0
    Dim APL As Integer
    Dim View As Double
    Dim PixelColor As Color

```



```

Dim ViewImg, tmpImg As Bitmap
Structure position
    Dim GetRow As Integer
    Dim GetCol As Integer
End Structure
Dim MapTable(ImgHeight, ImgWidth) As position
Dim row, col As Integer

Public Sub LoadDeviceList() '列出所有的視訊設備
    Dim strName As String = Space(100)
    Dim strVer As String = Space(100)
    Dim bReturn As Boolean
    Dim no As Integer = 0
    Do
        bReturn = capGetDriverDescriptionA(no, strName, 100, strVer, 100) '取得視訊設備名稱
        If bReturn Then LstDevices.Items.Add(strName.Trim) '將視訊設備加入 ListBox
        no = no + 1
    Loop Until bReturn = False
End Sub

Private Sub CreateMapTable() '建立變形校正對應表
    Dim CenterX, CenterY As Double
    Dim Radius, NormRadius, NormfRadius, fRadius, MaxRadius, Angle As Double
    Dim GetRow, GetCol As Integer
    MaxRadius = Math.Sqrt(ImgWidth ^ 2 + ImgHeight ^ 2) / 2
    CenterX = (ImgWidth + 1) / 2 : CenterY = (ImgHeight + 1) / 2
    For row = 1 To ImgHeight
        For col = 1 To ImgWidth
            Radius = Math.Sqrt((col - CenterX) ^ 2 + (CenterY - row) ^ 2)
            If Radius = 0 Then
                fRadius = 0
                Angle = 0
            Else
                NormRadius = Radius / MaxRadius
                NormfRadius = a4 * NormRadius ^ 4 + a3 * NormRadius ^ 3 + a2 * NormRadius ^ 2 + a1 * _
                NormRadius
                fRadius = MaxRadius * NormfRadius
                If (col - CenterX >= 0) And (CenterY - row >= 0) Then '第一象限
                    Angle = Math.Asin(Math.Abs(CenterY - row) / Radius)
                End If
                If (col - CenterX <= 0) And (CenterY - row >= 0) Then '第二象限
                    Angle = Math.PI - Math.Asin(Math.Abs(CenterY - row) / Radius)
                End If
                If (col - CenterX <= 0) And (CenterY - row <= 0) Then '第三象限
                    Angle = Math.PI + Math.Asin(Math.Abs(CenterY - row) / Radius)
                End If
                If (col - CenterX >= 0) And (CenterY - row <= 0) Then '第四象限
                    Angle = 2 * Math.PI - Math.Asin(Math.Abs(CenterY - row) / Radius)
                End If
            End If
            GetRow = Int(CenterY - fRadius * Math.Sin(Angle) + 0.5)
            GetCol = Int(fRadius * Math.Cos(Angle) + CenterX + 0.5)
            MapTable(row, col).GetRow = GetRow
            MapTable(row, col).GetCol = GetCol
        Next
    Next
End Sub

Private Sub OpenPreviewWindow() '開啓原始影像預覽視窗
    Dim iHeight As Integer = PicCapture.Height

```

```

Dim iWidth As Integer = PicCapture.Width
'Open Preview window in picturebox
hHwnd = capCreateCaptureWindowA(iDevice, WS_VISIBLE Or WS_CHILD, 0, 0, ImgWidth, _
ImgHeight, PicCapture.Handle.ToInt32, 0)
'Connect to device
If SendMessage(hHwnd, WM_CAP_DRIVER_CONNECT, iDevice, 0) Then
    SendMessage(hHwnd, WM_CAP_SET_SCALE, True, 0) '允許 preview 視訊的縮放比例
    SendMessage(hHwnd, TVM_SETBKCOLOR, 0, RGB(0, 0, 0))
    SendMessage(hHwnd, WM_CAP_SET_PREVIEWRATE, Interval, 0) '在 preview 模式顯示時設定速度
    SendMessage(hHwnd, WM_CAP_SET_PREVIEW, True, 0)
    SetWindowPos(hHwnd, HWND_BOTTOM, 0, 0, PicCapture.Width, PicCapture.Height, _
    SWP_NOMOVE Or SWP_NOZORDER)
    BtnStop.Enabled = True
    BtnStart.Enabled = False
Else
    DestroyWindow(hHwnd)
End If
End Sub

Private Sub ClosePreviewWindow() '停止原始影像預覽視窗
    SendMessage(hHwnd, WM_CAP_DRIVER_DISCONNECT, iDevice, 0)
    DestroyWindow(hHwnd)
End Sub

Private Sub CopyClipboard() '從預覽視窗取出影像
    SendMessage(hHwnd, WM_CAP_EDIT_COPY, 0, 0)
    ImgData = Clipboard.GetDataObject()
    If ImgData.GetDataPresent(GetType(System.Drawing.Bitmap)) Then
        PreviewImg = CType(ImgData.GetData(GetType(System.Drawing.Bitmap)), Image)
    End If
End Sub

Private Sub Correction() '產生變形校正影像 ViewImg
    Dim tmpR, tmpG, tmpB As Integer
    Dim tmpGetCol, tmpGetRow As Integer
    Dim WidthGap, HeightGap As Double
    WidthGap = (ImgWidth - ImgWidth * View) / 2
    HeightGap = (ImgHeight - ImgHeight * View) / 2
    For row = 1 To ImgHeight
        For col = 1 To ImgWidth
            tmpGetCol = MapTable(Int(row * View + HeightGap + 0.5), Int(col * View + WidthGap + 0.5)).GetCol
            tmpGetRow = MapTable(Int(row * View + HeightGap + 0.5), Int(col * View + WidthGap + _
            0.5)).GetRow
            If (tmpGetCol >= 1) And (tmpGetCol <= ImgWidth) And (tmpGetRow >= 1) And _
            (tmpGetRow <= ImgHeight) Then
                PixelColor = tmpImg.GetPixel(tmpGetCol - 1, tmpGetRow - 1)
                tmpR = PixelColor.R + APL
                tmpG = PixelColor.G + APL
                tmpB = PixelColor.B + APL
                If tmpR < MinVal Then tmpR = MinVal
                If tmpR > MaxVal Then tmpR = MaxVal
                If tmpG < MinVal Then tmpG = MinVal
                If tmpG > MaxVal Then tmpG = MaxVal
                If tmpB < MinVal Then tmpB = MinVal
                If tmpB > MaxVal Then tmpB = MaxVal
                ViewImg.SetPixel(col - 1, row - 1, Color.FromArgb(tmpR, tmpG, tmpB))
            Else
                ViewImg.SetPixel(col - 1, row - 1, Color.FromArgb(MaxVal, MaxVal, MaxVal))
            End If
        Next
    Next

```

```

Next
tmpImg.Dispose()
End Sub

```

```

Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles MyBase.Load
'程式開始執行時的初始化動作

```

```

    Call LoadDeviceList() '列出所有的視訊設備
    If LstDevices.Items.Count > 0 Then
        LstDevices.SelectedIndex = 0
        BtnStart.Enabled = True
    Else
        LstDevices.Items.Add("沒有連接任何視訊設備")
        BtnStart.Enabled = False
    End If
    Me.Timer1.Stop()
    BtnStop.Enabled = False
    PicCapture.SizeMode = PictureBoxSizeMode.StretchImage
    APL = TrackBar1.Value '取得亮度調整值
    View = TrackBar2.Value / 100 '取得視角調整值
    Call CreateMapTable()

```

```

End Sub

```

```

Private Sub BtnStart_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles BtnStart.Click
'開始接收視訊按鈕按下

```

```

    iDevice = LstDevices.SelectedIndex
    Call OpenPreviewWindow() '開啓原始影像預覽視窗
    Me.Timer1.Start()

```

```

End Sub

```

```

Private Sub BtnStop_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles BtnStop.Click
'停止接收視訊按下

```

```

    ClosePreviewWindow() '停止原始影像預覽視窗
    BtnStart.Enabled = True
    BtnStop.Enabled = False
    PicCorrect.Image = Nothing
    Me.Timer1.Stop()

```

```

End Sub

```

```

Private Sub TrackBar1_MouseUp(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs)
Handles TrackBar1.MouseUp

```

```

    APL = TrackBar1.Value '取得亮度調整值

```

```

End Sub

```

```

Private Sub TrackBar2_MouseUp(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs)
Handles TrackBar2.MouseUp

```

```

    View = TrackBar2.Value / 100 '取得視角調整值

```

```

End Sub

```

```

Private Sub Timer1_Tick(ByVal sender As Object, ByVal e As System.EventArgs) Handles Timer1.Tick
'固定間格時間到需更新視訊設備影像

```

```

    Call CopyClipboard()
    tmpImg = PreviewImg.Clone
    ViewImg = PreviewImg.Clone
    PreviewImg.Dispose()
    Call Correction()
    PicCorrect.Image = ViewImg.Clone
    ViewImg.Dispose()
    TextBox1.Text = "亮度=" & APL & "    " & "視角=" & Int(150 * View + 0.5) & "度"

```

```

End Sub

```

```

End Class

```

評語

作品極具實用價值，且具備關鍵核心技術。由於此主題並不是很新的研究，建議應加強更多的 **survey work**。此外應增加演算法效能評估的部分，進一步的能提出「演算法優化」。