

3DTV 作業四

I. 作業題目：使用附檔之記事本檔案進行影像校正，詳細步驟與格式如投影片所述，算出內部參數(Alpha、Gamma)、外部參數（平移、旋轉矩陣），詳細數值至少顯示到小數第六位。

II. 編譯/執行環境

- i. Visual Studio 2008，OpenCV2.1。
- ii. Windows 7，64 位元作業系統。

III. 程式碼(附件在壓縮檔內)

main calibration.cpp。

IV. 實驗流程與結果

i. 實驗流程

1. 使用 fopen()函式開啟並讀入 Nonconplanar_pts.txt 檔。
2. 目標求解下式未知參數。

$$x_{im} - o_x = -f_x \frac{X^c}{Z^c} = -f_x \frac{r_{11}X^w + r_{12}Y^w + r_{13}Z^w + T_x}{r_{31}X^w + r_{32}Y^w + r_{33}Z^w + T_z}$$
$$y_{im} - o_y = -f_y \frac{Y^c}{Z^c} = -f_y \frac{r_{21}X^w + r_{22}Y^w + r_{23}Z^w + T_y}{r_{31}X^w + r_{32}Y^w + r_{33}Z^w + T_z}$$

經過整理可得：

$$x_i X_i^w v_1 + x_i Y_i^w v_2 + x_i Z_i^w v_3 + x_i v_4 - y_i X_i^w v_5 - y_i Y_i^w v_6 - y_i Z_i^w v_7 - y_i v_8 = 0$$

其中

$$v_1 = r_{21}, v_2 = r_{22}, v_3 = r_{23}, v_4 = T_y, v_5 = \alpha r_{11}, v_6 = \alpha r_{12}, v_7 = \alpha r_{13}, v_8 = \alpha T_x$$

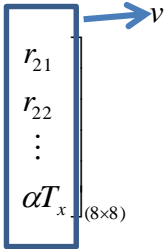
$$\alpha = \frac{f_x}{f_y}$$

假設我們有 N 組點，可得以下關係：

$$Av = 0$$

其中

$$A = \begin{bmatrix} x_1 X_1^w & x_1 Y_1^w & x_1 Z_1^w & x_1 & -y_1 X_1^w & -y_1 Y_1^w & -y_1 Z_1^w & -y_1 \\ x_2 X_2^w & x_2 Y_2^w & x_2 Z_2^w & x_2 & -y_2 X_2^w & -y_2 Y_2^w & -y_2 Z_2^w & -y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N X_N^w & x_N Y_N^w & x_N Z_N^w & x_N & -y_N X_N^w & -y_N Y_N^w & -y_N Z_N^w & -y_N \end{bmatrix}_{(N \times 8)}$$

$$V = \begin{bmatrix} \cdots & \cdots & r_{21} \\ \cdots & \cdots & r_{22} \\ \vdots & \vdots & \vdots \\ \cdots & \cdots & \alpha T_x \end{bmatrix}_{(8 \times 8)}$$


3. 將讀入的資料轉換成 opencv 的陣列結構。
4. 使用 cvSVD()、cvmMul() 等函式算出 V 矩陣，並經由以下運算式

解出 $\alpha |\gamma|$ ：

$$\sqrt{\bar{v}_1^2 + \bar{v}_2^2 + \bar{v}_3^2} = \sqrt{\gamma^2(r_{21}^2 + r_{22}^2 + r_{23}^2)} = |\gamma|$$

$$\sqrt{\bar{v}_5^2 + \bar{v}_6^2 + \bar{v}_7^2} = \sqrt{\gamma^2 \alpha^2 (r_{11}^2 + r_{12}^2 + r_{13}^2)} = \alpha |\gamma|$$

$|\gamma|$ 由下列判別式決定正負：

$$r_{11}X^w + r_{12}Y^w + r_{13}Z^w + T_x$$

判別式為正， $\gamma < 0$ ；

判別式為負， $\gamma > 0$ 。

5. 由以下運算將旋轉矩陣完整求出：

$$r_{31} = \left\| \begin{bmatrix} r_{12} & r_{13} \\ r_{22} & r_{23} \end{bmatrix} \right\| \quad r_{32} = - \left\| \begin{bmatrix} r_{11} & r_{13} \\ r_{21} & r_{23} \end{bmatrix} \right\| \quad r_{33} = \left\| \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \right\|$$

6. 由以下運算將平移矩陣完整求出：

$$A \begin{pmatrix} T_z \\ f_x \end{pmatrix} = b \text{ or } Ax = b$$

$$x = \begin{pmatrix} T_z \\ f_x \end{pmatrix} = (A^T A)^{-1} A^T b$$

其中

$$A = \begin{bmatrix} x_1 & r_{11}X_1^W + r_{12}Y_1^W + r_{13}Z_1^W + T_x \\ x_2 & r_{11}X_2^W + r_{12}Y_2^W + r_{13}Z_2^W + T_x \\ \vdots & \vdots \\ x_N & r_{11}X_N^W + r_{12}Y_N^W + r_{13}Z_N^W + T_x \end{bmatrix}$$

$$b = \begin{bmatrix} -x_1(r_{31}X_1^W + r_{32}Y_1^W + r_{33}Z_1^W) \\ -x_2(r_{31}X_2^W + r_{32}Y_2^W + r_{33}Z_2^W) \\ \vdots \\ -x_N(r_{31}X_N^W + r_{32}Y_N^W + r_{33}Z_N^W) \end{bmatrix}$$

7. 由以求出的 α 及 f_x 利用以下關係求出 f_y ：

$$\alpha = \frac{f_x}{f_y}$$

8. 將運算完的結果輸出至螢幕上。

ii. 實驗結果

```
c:\Users\Milk_Lab_15\Desktop\Dropbox\3D\Project\3D hw4\Debug\3D hw4.exe
Parameters
Gama: 0.0105282
Alpha: 0.998987
Translation:
1.3041 -94.9632 927.049
Rotation:
0.582966 0.0024239 -0.812493
-0.0286409 0.999421 -0.0183621
0.811978 0.033975 0.582698
Focal length:
fx: 2394.03
fy: 2396.45
請按任意鍵繼續 . . .
```

V. 結論與心得

一開始看投影片的時候對於這次的作業其實沒有頭緒，主要是因為對於線性代數一些定理都忘記了，加上對於 opencv 一些矩陣運算的函式不是很熟悉所致。後來上網查了一些相關資料，才知道如何實作這次作業。除此之外，對於投影片中一些參數的定義不是很確定，花了一些時間驗證。例如影像原點我一開始認為它指的是像素(0,0)的位置，但計算出來的結果都不正確，後來改成影像的正中心就對了。

關於 opencv 的矩陣資料結構 `CvMat`，它可以儲存很多種資料型態，像是 `int`、`float`、`double` 等等。當要對它取值的時候要相當注意，假設給它 `int` 的資料，資料結構的設計中，對 `float` 或其他型態同樣可以取值，但它沒有做型態轉換的動作，所以值會是不正確的。這次作業中我曾使用一些 opencv 已經有的函式去對我的矩陣取值，但常常出問題，我想是資料型態的關係，因此後來我都改用直接對資料結構的特定空間取值，例如 `Matrix->data.i[0]`，不依靠 opencv 的函式，以確保正確性。