

結合 Visual studio 與 ObjectARX 之客製化式 AutoCAD 外掛插件

黃敬群博士

國立中正大學

電機工程學系

應用運算與多媒體實驗室

<http://acm.ee.ccu.edu.tw/>

聯絡電話：05-2720411#33223

電子郵件：chingchun.huang6@gmail.com

目錄

● 研究內容.....	3
● 整體流程圖.....	5
● 演算法說明.....	11
● 環境安裝.....	16
● 使用方法.....	20
● 程式功能說明.....	30

圖目錄

● 系統流程圖.....	5
● 輸入資料.....	6
● 圖 1，線性模型與參考模型的示意圖.....	13
● 圖 2，參考模型的初始化步驟示意圖.....	14
● 圖 3，參考模型用來預測點的示意圖.....	14
● 圖 4，座標轉換示意圖.....	15
● 結果圖.....	28

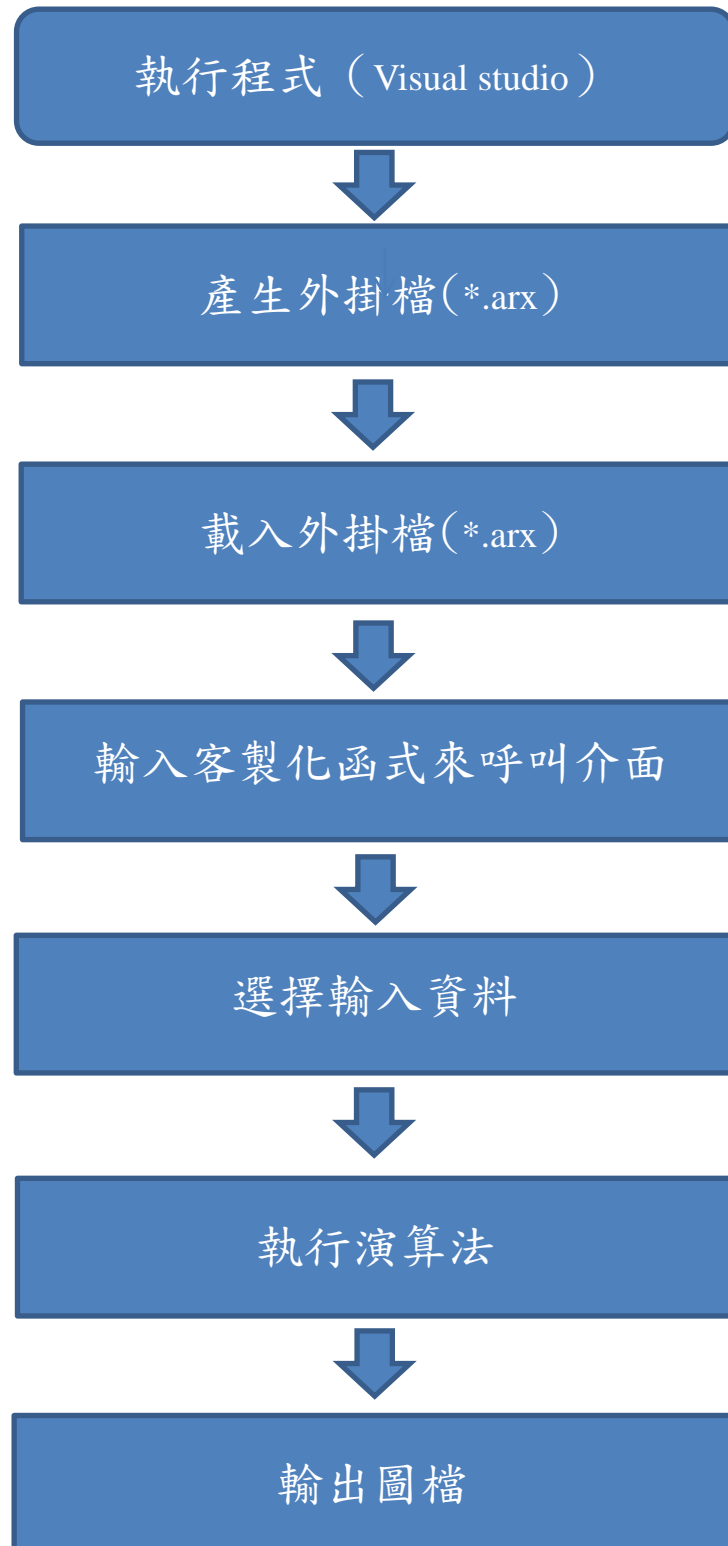
➤ 研究內容：

- 本計畫期望能夠在太陽能板上印製屬於自己的標籤以及圖案，倘若在不允許的位置上印製可能會影響到太陽能板的使用導致功能上的缺失，因此必須得知太陽能板上網格可列印部分的精確位置，之後再將真實世界坐標系轉換到理想坐標系上（AutoCAD），以此來製作標籤及圖案。然而太陽能板在生產製造的過程中或多或少都會產生不同程度的扭曲，所以如何在扭曲情況得知精確的網格點中心，找出其所屬線段的扭曲曲線，並由真實世界坐標系轉到理想坐標系（設計之 AutoCAD 坐標系）上，最後結合演算法，客製化產生符合真實網格扭曲變化之 AutoCAD 圖檔為本計畫核心目標，針對此計畫目標，我們設計之雛型系統可分為下面兩部分。
- ◆ 第一部分（演算法）：將經由掃描機循邊取得的真實世界座標點轉換到理想的坐標系上，然而掃描過程中可能會有估計錯誤的情況產生，因此我們使用演算法設計三個嚴謹的條件對每一個點進行篩選，將錯誤的點濾除並根據設計的模型合理的預測新的點來取代錯誤的點，最後再透過循邊掃描歸類的資訊將同一類別的線段連接起來，並透過所估算之 Homography 空間轉換矩陣，將真實坐標系轉為

(AutoCAD) 理想坐標系。

- ◆ 第二部分 (AutoCAD 製圖): 演算法主要在 Visual studio 的環境上實作, 其產出之資訊需進一步與 AutoCAD 結合, 以便產生 AutoCAD 圖檔。為完成此目標, 我們使用了 ObjectARX (Visual studio 與 AutoCAD) 來溝通兩者, 並將第一部分所設計之客製化演算法包裝為 AutoCAD 指令, 並設計可視化界面讓使用者選取資料, 之後將資料輸入演算法進行運算, 最後輸出並串聯所有經篩選與預測過後的點資訊且自動出圖。

➤ 整體流程圖



■ 輸入資料:

◆ 下圖為產生一組圖檔所需的輸入資料



GetPointsC
ADForWar
p.txt



GetPointsF
orWarp.txt



testdata15
0L.H.G2X.t
xt



testdata15
0L.H.G2Y.t
xt



testdata15
0L.V.G2X.t
xt



testdata15
0L.V.G2Y.t
xt

■ 檔案 1: 循邊掃描點資訊(.txt)

資料名稱	單位	範例
所有掃描點資訊	X 座標/Y 座標	0.001



testdata15
OL.H.G2X.t
xt



testdata15
OL.H.G2Y.t
xt



testdata15
OL.V.G2X.t
xt



testdata15
OL.V.G2Y.t
xt

備註：

105LinesHorizontal.txt(105LinesVertical.txt 格式相同)

存點座標資訊，四個檔案分別為水平線 X 座標、Y 軸座標資訊・垂直

線 X 座標資訊，Y 軸座標資訊，每個資訊間使用空白隔開。

```
-0.145349 -0.0813953 -0.0174419
-0.151163 -0.0872093 -0.0232558
-0.162791 -0.0930233 -0.0290698
-0.168605 -0.104651 -0.0348837
-0.174419 -0.110465 -0.0465116
-0.180233 -0.116279 -0.0523256
-0.186047 -0.122093 -0.0581395
```

■ 檔案 3:掃描點坐標系的校正點(.txt)

檔案 5		
資料名稱	單位	範例
實際網格校正點	左下,	0,0
	右下,	200.041,0.0581395
	左上,	-0.0639535,200.006
	右上	199.977,200.07



GetPointsF
orWarp.txt

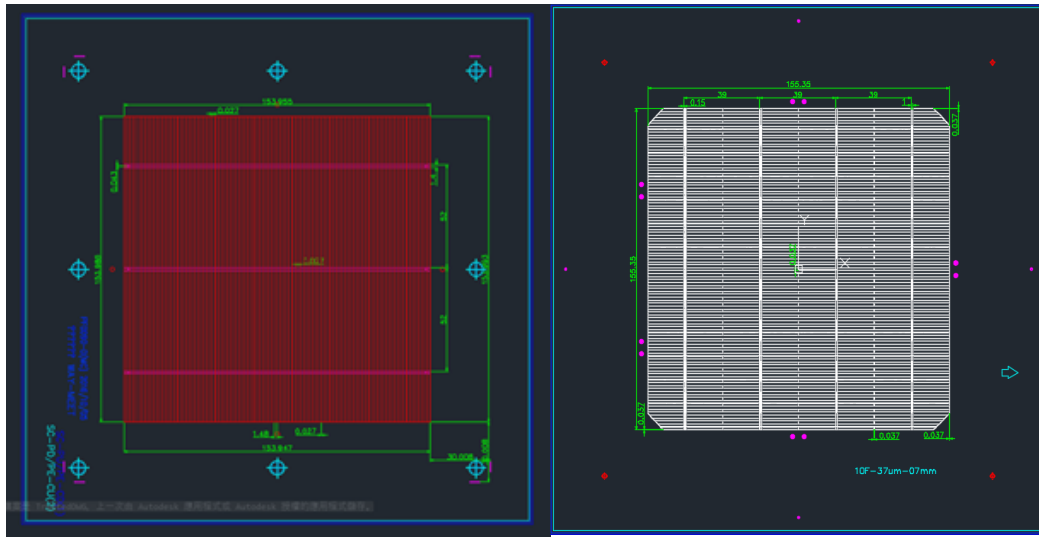
備註：

校正點 (X Y) 座標資訊，提供四個座標點作為輸入 XY 資訊用空

白間隔，每個點換行儲存。

```
0 0
200.041 0.0581395
-0.0639535 200.006
199.977 200.07
```


■ 檔案 4:CAD 圖檔(.DWG/.DXF)



■ 檔案 5:CAD 上的校正點(.txt)

檔案 5		
資料名稱	單位	範例
.Dwg 檔在 AutoCAD 上的校正點	左下,	
	右下,	219.3799,151.2251; 419.3799,151.2251;
	左上,	219.3799,351.2251; 419.3799,351.2251;
	右上	



GetPointsC
ADForWar
p.txt

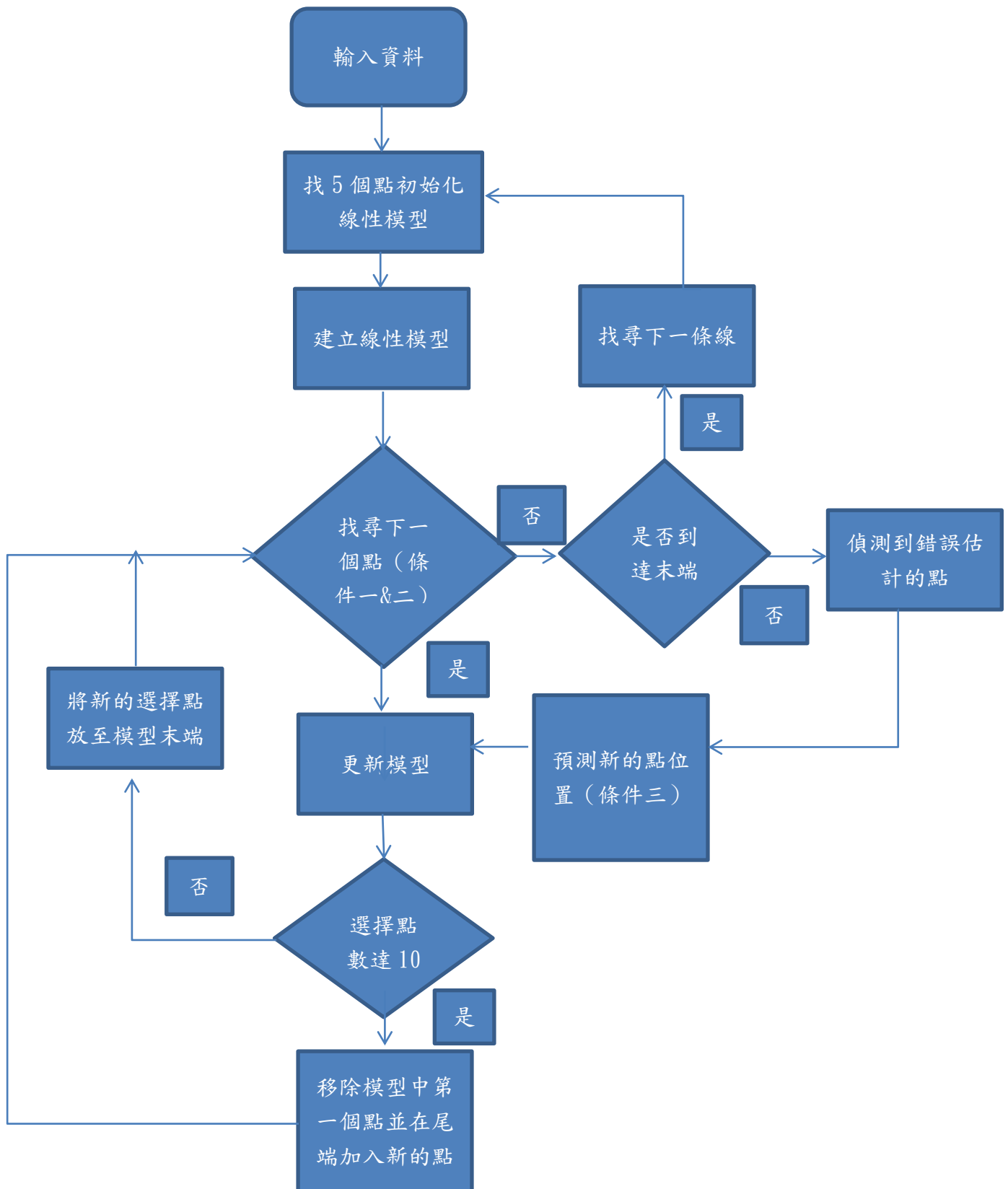
備註：

校正點 (X Y) 座標資訊，提供四個座標點作為輸入 XY 資訊用空

白間隔，每個點換行儲存。

```
219.3799 151.2251
419.3799 151.2251
219.3799 351.2251
419.3799 351.2251
```

➤ 演算法說明：



■ 為了找出掃描機錯誤估計的點資訊，因此我們設計了 3 個條件來判斷每一個點是否合乎標準。我們得知彼此接連的網格點中心的理想的大小為 0.065mm，因此我們設計了以下的條件：

◆ 條件一:假設相鄰點距離為 d 則 d 不可小於 0.04mm 且不可大於 0.073mm:

● $0.04 < d < 0.073$

此項條件可以由使用者進行改變，經由實驗過後這是我們認為比較合適的參數。

◆ 條件二：由幾個點組成一直線方程式(線性模型)來評估下一個點是否有錯誤估計的問題，因為這條直線是由前幾個點所組成的，因此可以預期下一個點與這條線性模型之間的距離必須足夠的接近，此距離參數我們預設為 0.03mm

◆ 條件三:當我們找到掃描點被錯誤估計的點時，設計參考模型與理想距離（0.065mm）去預測下一個點位置

■ 參考模型（如圖 1）：參考模型分為兩步驟，首先必須對參考模型進行初始化，從第一條線開始若下一個點符合條件一和條件二時將點與點之間的距離保存下來，若遇到了錯誤估計的點時，跳到下一條線進行更新參考模型，直到預測到整條完整的參考模型，

再回過頭檢視錯誤估計的點並產生預測點（如圖 2 和圖 3）。

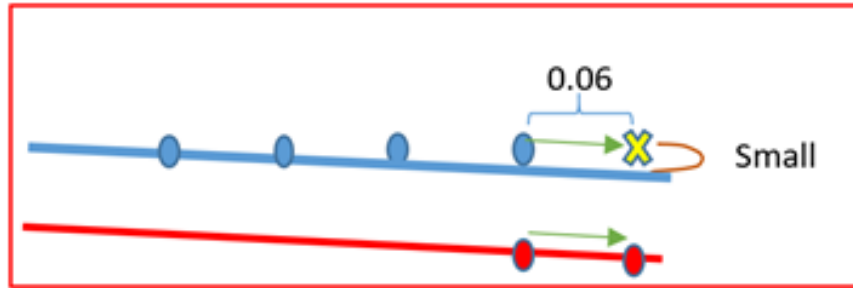


圖 1．表示線性模型與參考模型的示意圖，紅色點為參考模型；藍色線為線性模型，透過參考線性模型與理想點距離來產生的預測點仍然需要符合條件一和條件二。

◆ 假設當前 t 時的輸入點為 (X_t, Y_t) ，參考模型的參考向量為 (V_x, V_y) ，若在 $t+1$ 時遇到錯誤估計的點 (X_{t+1}, Y_{t+1}) 則預測點 (X'_{t+1}, Y'_{t+1}) 的方程式如下：

■ $(X'_1, Y'_1) = (X_t + 0.065, Y_t)$ ，.....(1)

■ $(X'_2, Y'_2) = (X_t + V_x, Y_t + V_y)$ ，.....(2)

■ $(X'_{t+1}, Y'_{t+1}) = (X'_1, Y'_1)/2 + (X'_2, Y'_2)/2$ ，.....(3)

● : Input points ● : Incomplete lines ● : Output points ● : Predict points

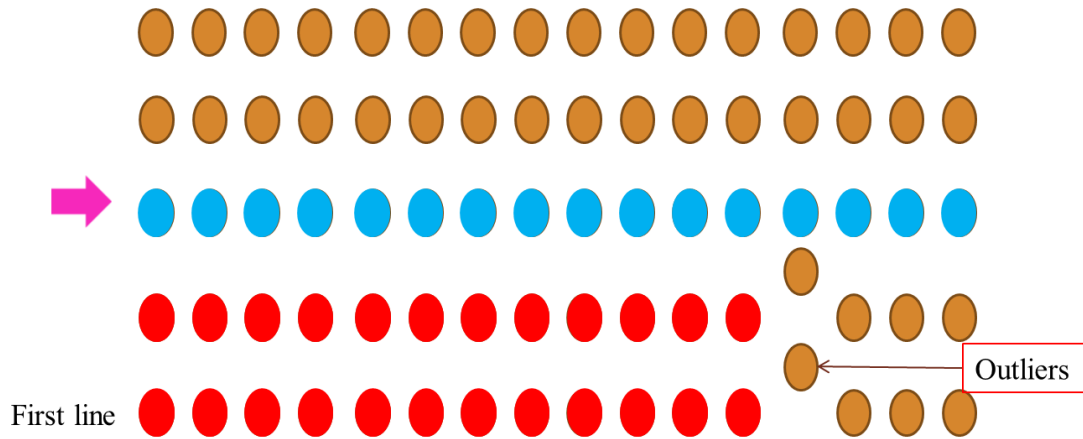


圖 2 · 參考模型的初始化步驟示意圖

● : Input points ● : Incomplete lines ● : Output points ● : Predict points

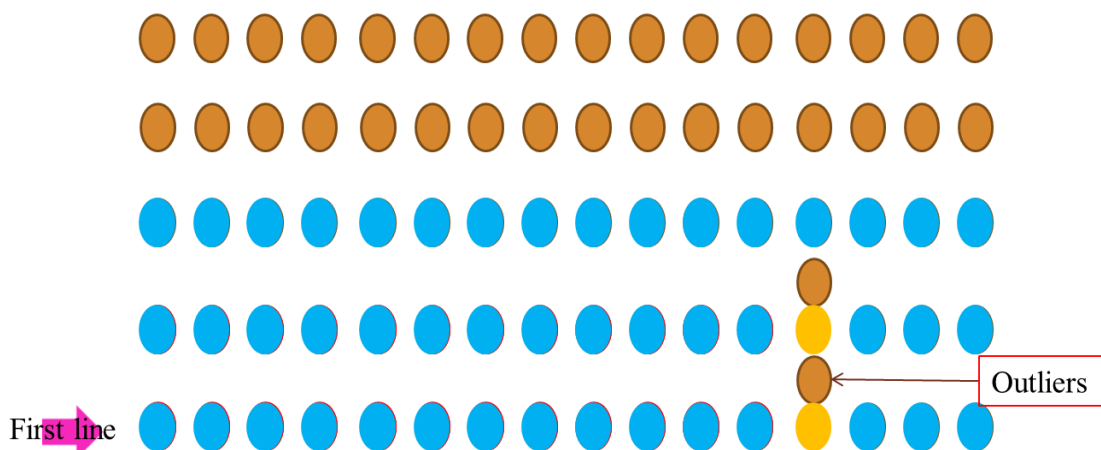


圖 3 · 參考模型用來預測點的示意圖

◆ 座標轉換: 估算 Homography 轉換矩陣

■ 要將真實座標轉換到理想座標上我們需要至少 4 個

對應的校正點，用來作為計算 Homography 矩陣的參

數，當解出矩陣後之後將所有點透過矩陣由真實做標

系轉換到理想做標系上（如圖 4）。

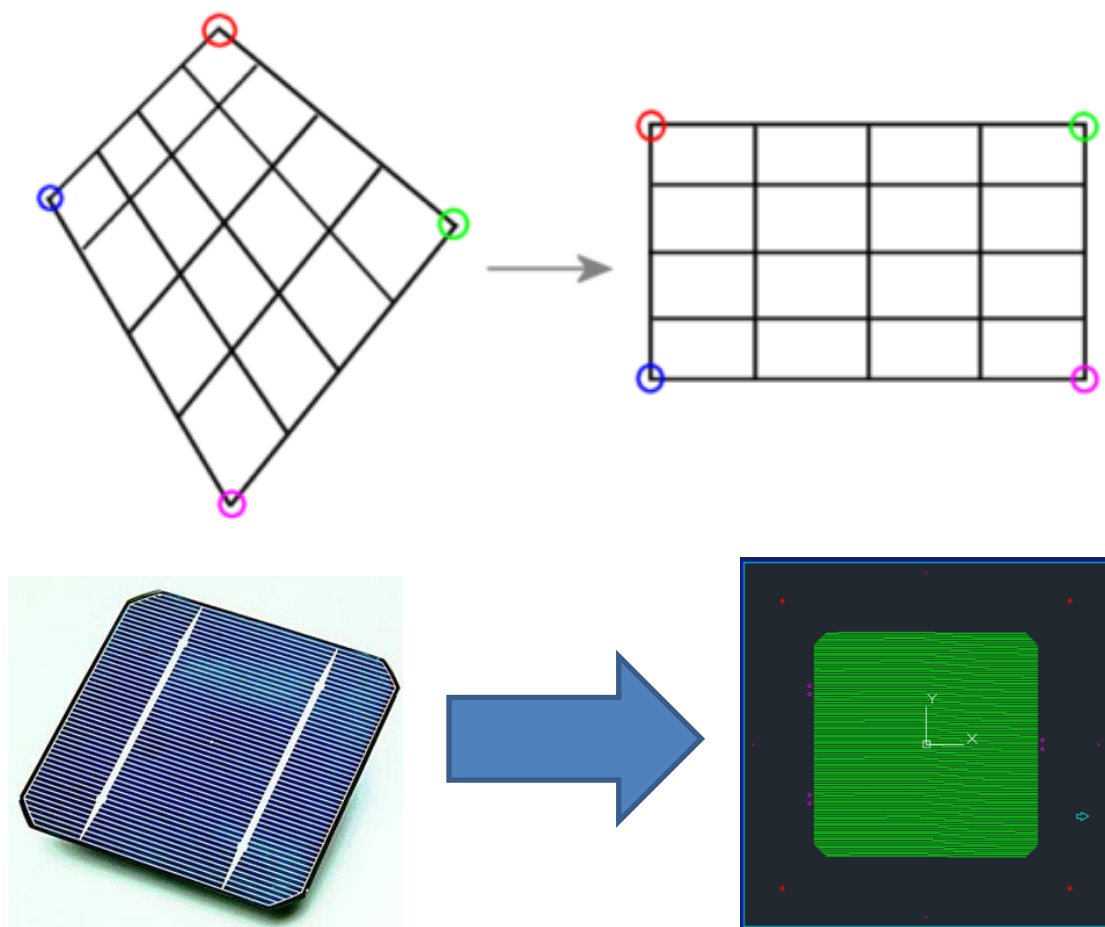


圖 4．上方為座標轉換示意圖，左下圖為太陽能板線條示意圖，經由掃描機掃描太陽能板得到真實作標系中網格中心點資訊再透過轉換矩陣轉換到理想做標系並繪製在 AutoCAD 上，右下圖為轉到理想做標上的示意圖。

➤ 環境安裝：

■ 下載並安裝

- ObjectARX SDK

-ObjectARX_2012_Win_64_and_32Bit.exe

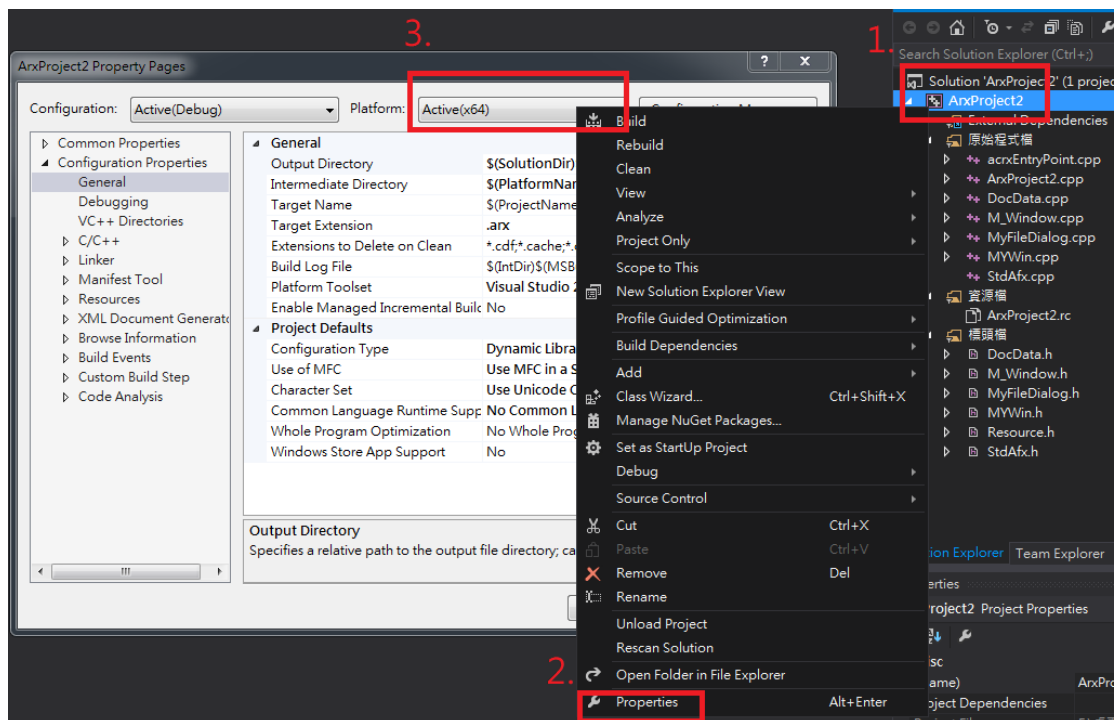
(<http://www.objectarx.net/forum.php?mod=viewthread&tid=1173>)

- AutoCAD_2012
- Visual Studio2013

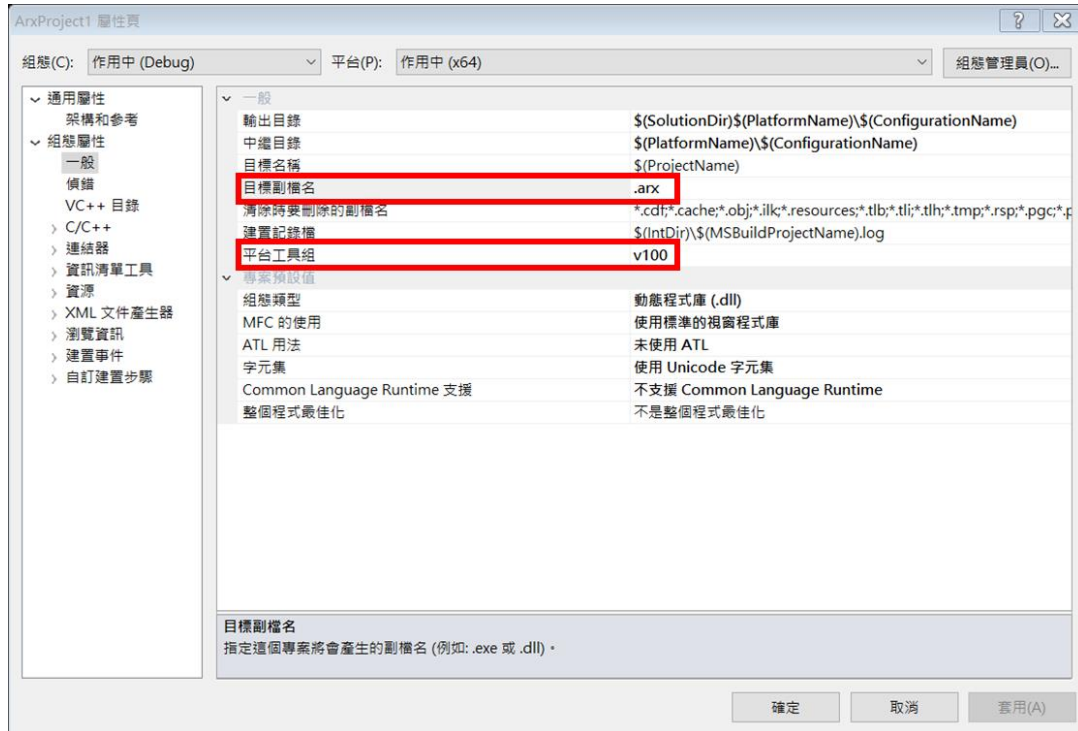
■ Visual studio 環境設置

- 1.對專案點擊右鍵->2.點擊屬性(properties)->3.更改編

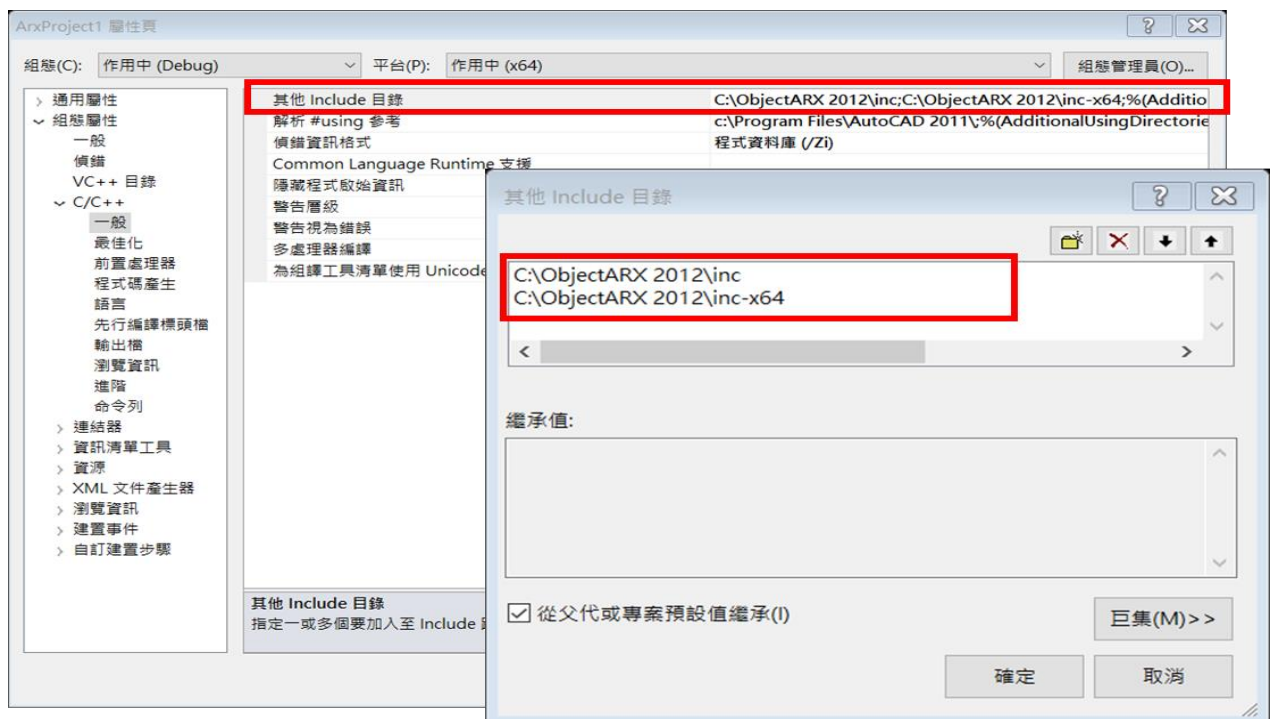
譯環境為 X64



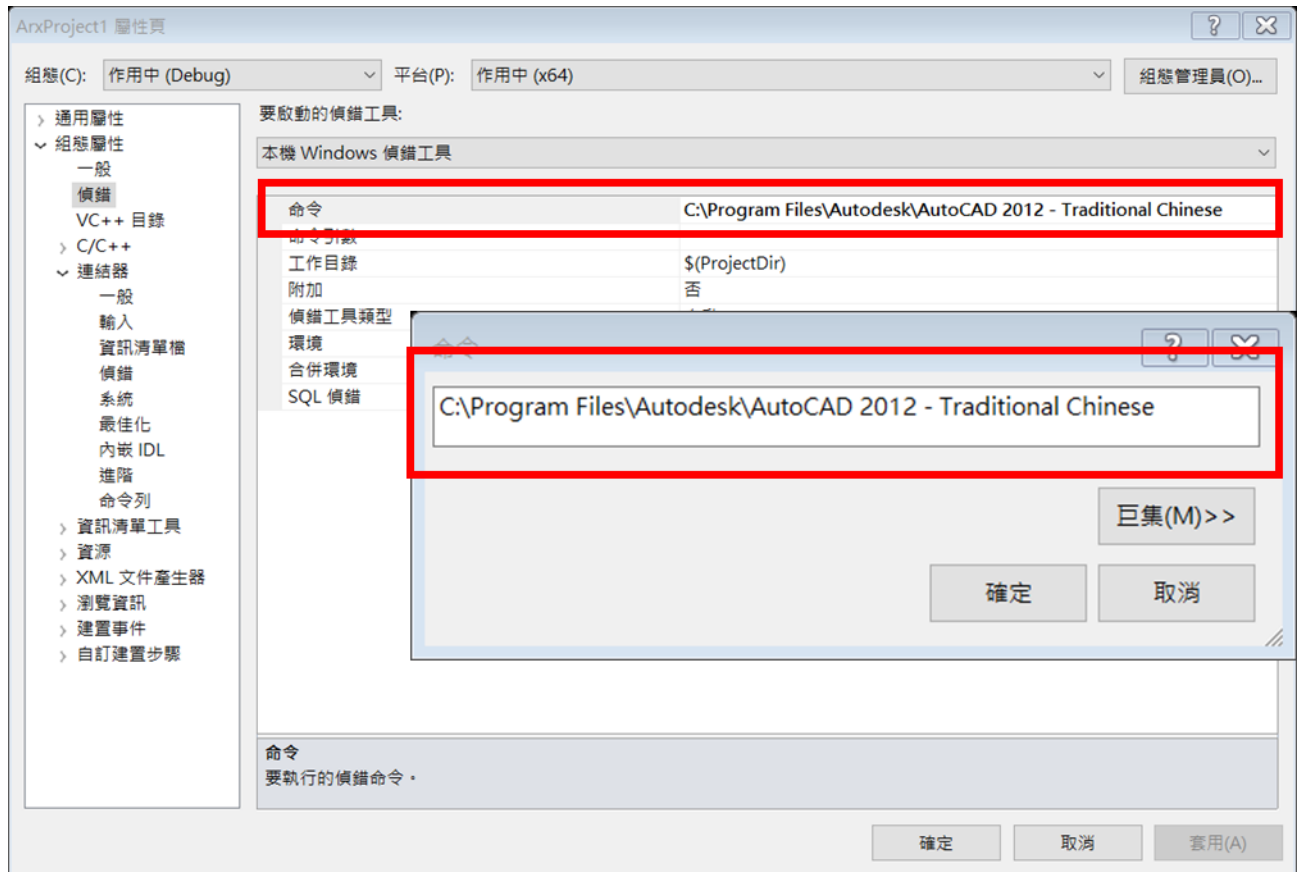
- 如文件擴展名改為 arx，平台工具集高位 V120



■ 加載頭文件，加載你下載的 objectarx 文件中的 inc 和 64 位的 inc

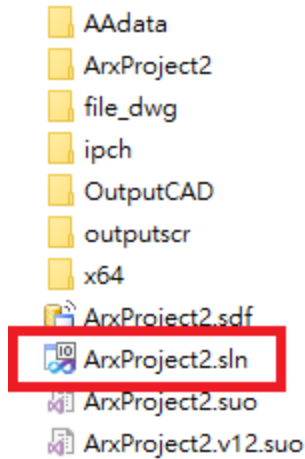


■ 加載 lib 文件

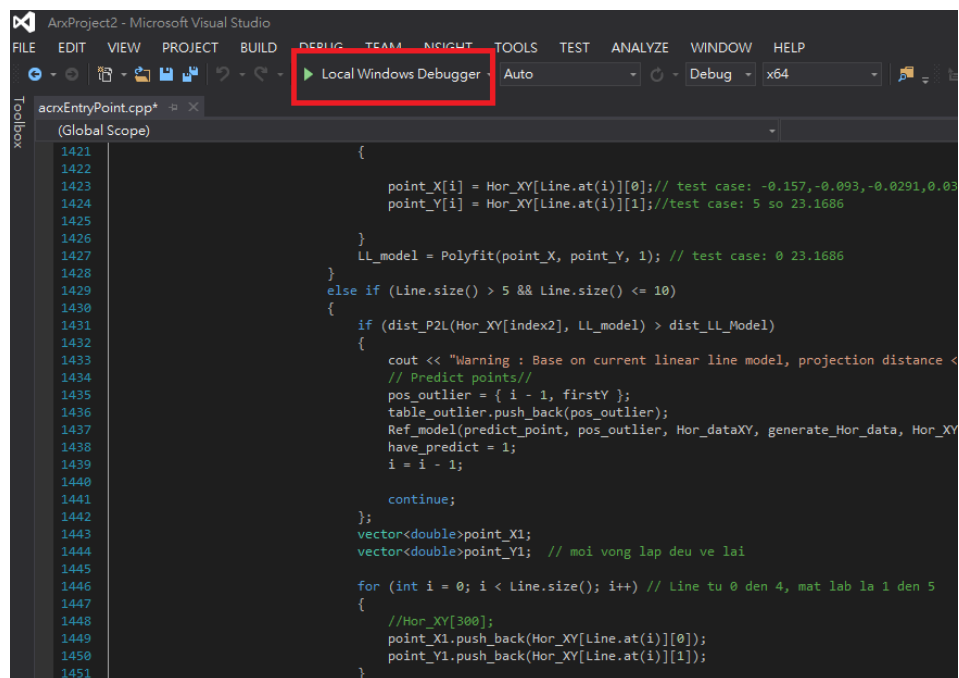


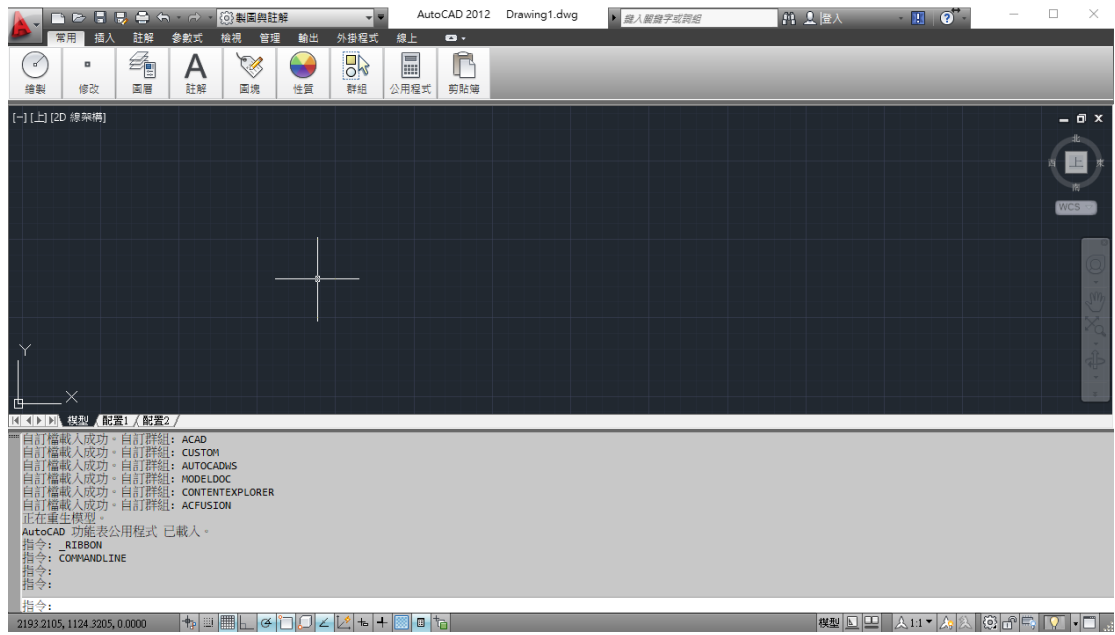
➤ 使用方法:

■ 開啟程式 ArxProject2



■ 執行 Visual studio2013 會自動開啟程式 AutoCAD



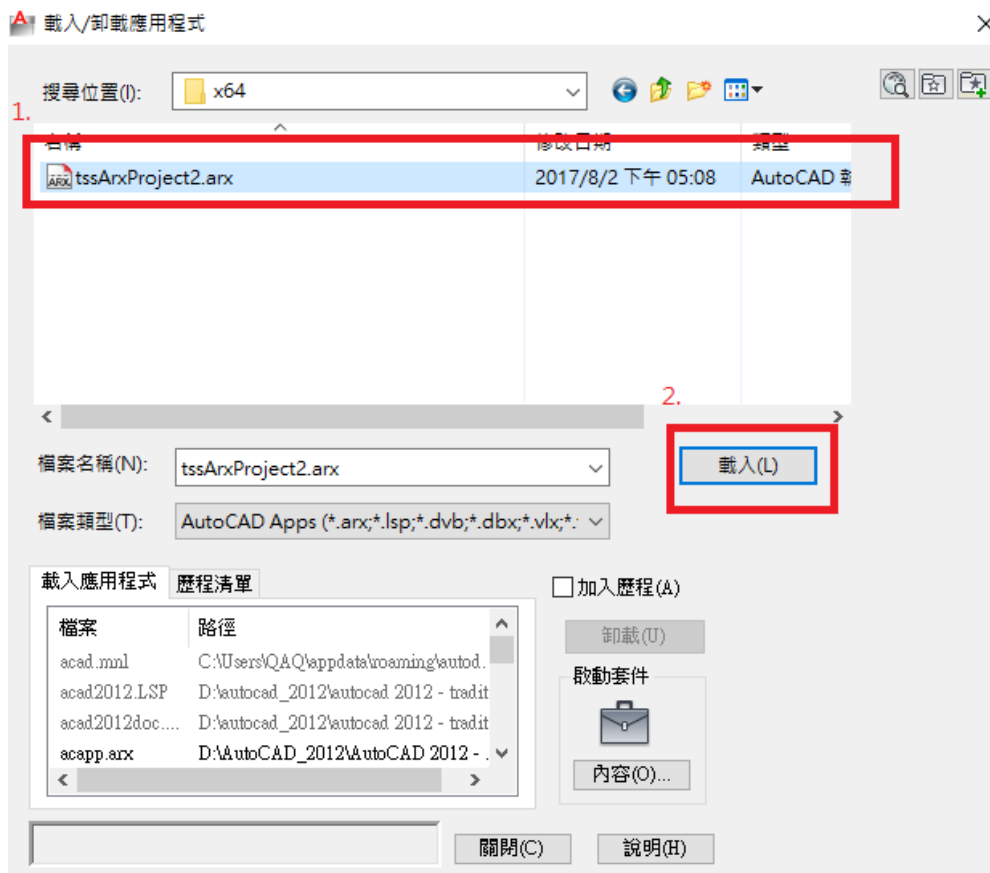


■ 將外掛檔載入

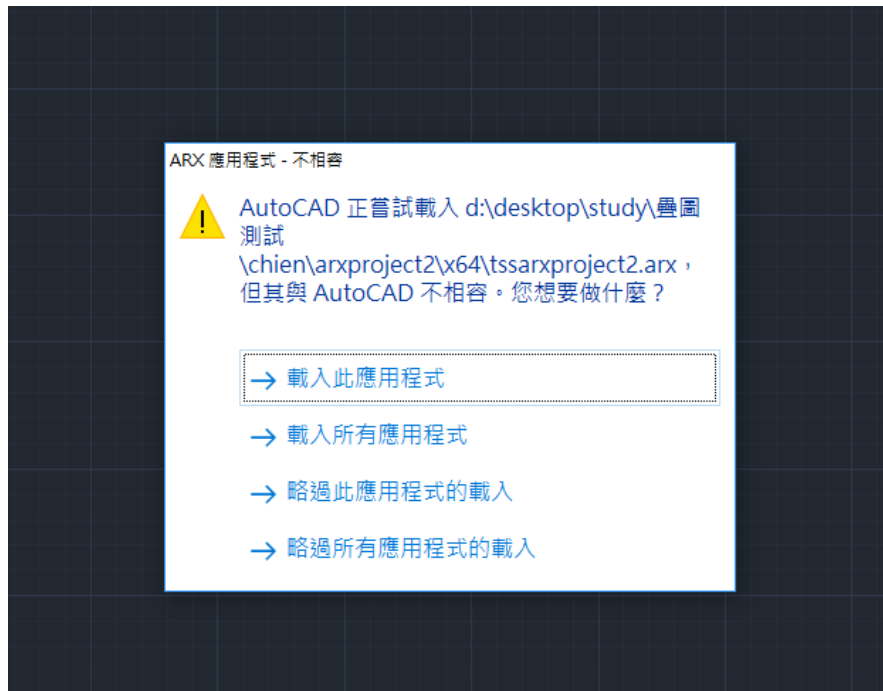
◆ 管理->應用程式->載入應用程式



◆ 選擇檔案(*.arx)



■ 案下”載入此應用程式”

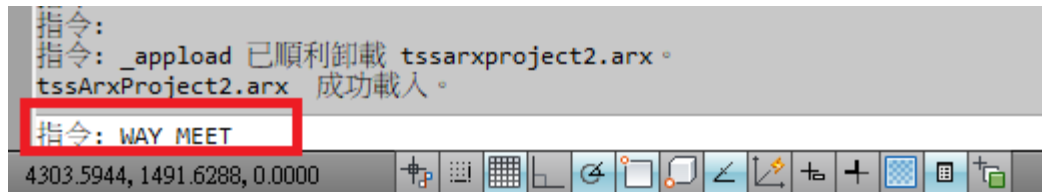


■ 載入成功

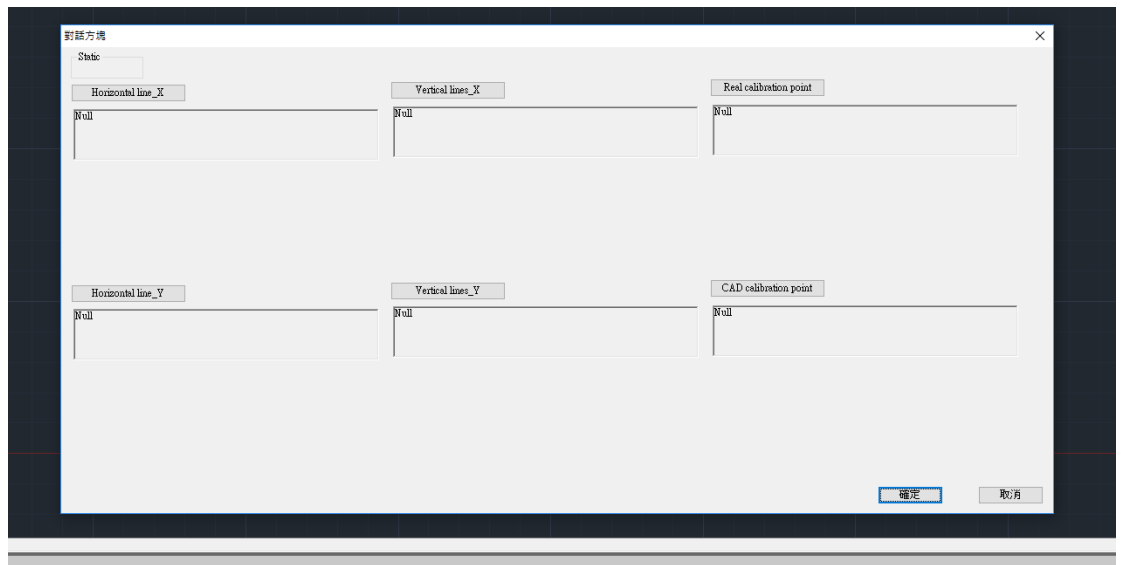
```
指令:
指令:
指令: _appload 已順利卸載 tssarxproject2.arx。
指令:
指令: (ARXLOAD "D:/Desktop/study/疊圖測試/Chien/ArxProject2/x64/tssArxProject2.arx")
"D:/Desktop/study/疊圖測試/Chien/ArxProject2/x64/tssArxProject2.arx"
指令:
指令:
指令: _appload 已順利卸載 tssarxproject2.arx。
tssArxProject2.arx 成功載入。
指令:
```

■ 輸入客製化指令：WAY_MEET

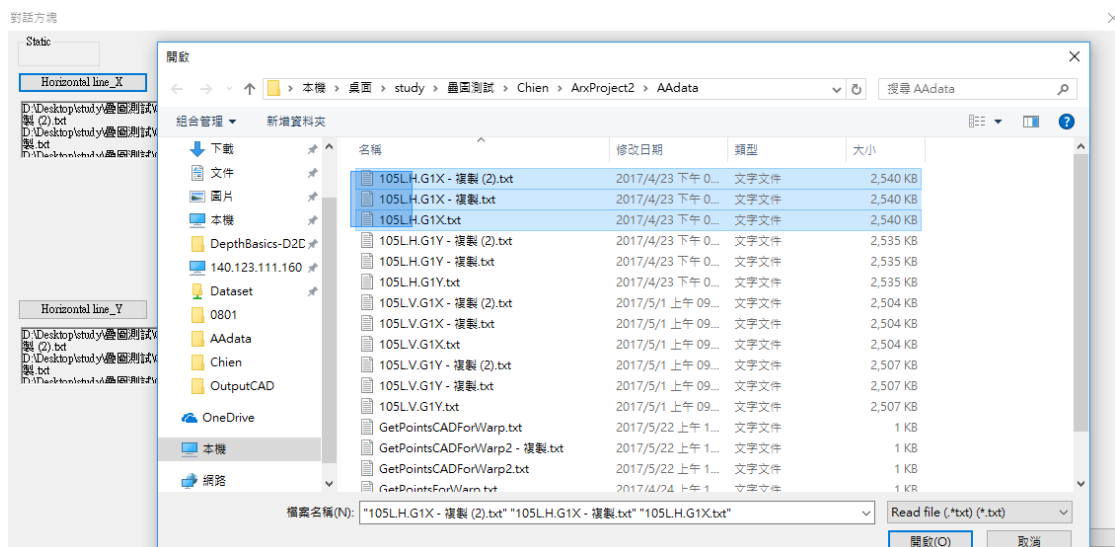
◆ 在命令視窗輸入函式,呼叫介面



成功呼叫介面:



■ 選取要處理的檔案



■ 確認

封裝方塊

<p>Static</p> <p>Horizontal line_X</p> <p>D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\105L H G1X - 複製 (2).bt D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\105L H G1X - 複製.btd D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\105L H G1Y - 複製 (2).bt D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\105L H G1Y - 複製.btd</p>	<p>Vertical lines_X</p> <p>D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\105L V G1X - 複製 (2).bt D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\105L V G1X - 複製.btd D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\105L V G1Y - 複製 (2).bt D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\105L V G1Y - 複製.btd</p>	<p>Real calibration point</p> <p>D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\GetPointsForWarp.btd D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\GetPointsForWarp2 - 複製.btd D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\GetPointsForWarp2 - 複製.btd</p>
<p>Horizontal line_Y</p> <p>D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\105L H G1Y - 複製 (2).bt D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\105L H G1Y - 複製.btd D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\105L H G1Y - 複製 (2).bt D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\105L H G1Y - 複製.btd</p>	<p>Vertical lines_Y</p> <p>D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\105L V G1Y - 複製 (2).bt D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\105L V G1Y - 複製.btd D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\105L V G1Y - 複製 (2).bt D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\105L V G1Y - 複製.btd</p>	<p>CAD calibration point</p> <p>D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\GetPointsCADForWarp.btd D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\GetPointsCADForWarp2 - 複製.btd D:\Desktop\study\疊圖測試\Chien\AndProject\AA\Adata\GetPointsCADForWarp2 - 複製.btd</p>

確定 取消

■ 程式執行中

模型 配置1 配置2

```

指令: Way
WAY_MEET COMMANDLINE
指令: Our_Code COMMANDLINE
指令: Our_Code COMMANDLINE
指令: Our_Code COMMANDLINE
指令: Our_Code COMMANDLINE
指令: Our_Code COMMANDLINE
指令: Our_Code COMMANDLINE
指令: Our_Code COMMANDLINE
指令: Our_Code COMMANDLINE
指令: Our_Code COMMANDLINE
指令: Our_Code COMMANDLINE

```

■ 自動產生腳本以及圖檔(結果圖)

◆ 腳本存放在資料夾(*/outputscr)

◆ 圖檔存放在資料夾(*OutputCAD)

名稱	修改日期	類型
outputHor_XY.scr	2017/8/3 下午 03:29	AutoCAD 腳本
outputHor_XY1.scr	2017/8/3 下午 03:28	AutoCAD 腳本
outputHor_XY2.scr	2017/8/3 下午 03:29	AutoCAD 腳本
outputHor_XY3.scr	2017/8/2 下午 04:10	AutoCAD 腳本
outputVER_XY.scr	2017/8/3 下午 03:29	AutoCAD 腳本
outputVER_XY1.scr	2017/8/3 下午 03:28	AutoCAD 腳本
outputVER_XY2.scr	2017/8/3 下午 03:29	AutoCAD 腳本
outputVER_XY3.scr	2017/8/2 下午 04:11	AutoCAD 腳本

產生N組水平&垂直
線腳本檔案

```

outputHor_XY.scr  outputVER_XY.scr
Pline  AUTOCAD指令
242.441,174.282
242.505,174.282
242.569,174.282
242.633,174.282
242.697,174.282
242.761,174.282
242.825,174.282
242.889,174.281
242.953,174.281
243.017,174.281
243.081,174.281
243.145,174.276
243.209,174.275
243.272,174.275
243.336,174.275
243.4,174.275
243.464,174.275
243.528,174.275
243.592,174.275

```

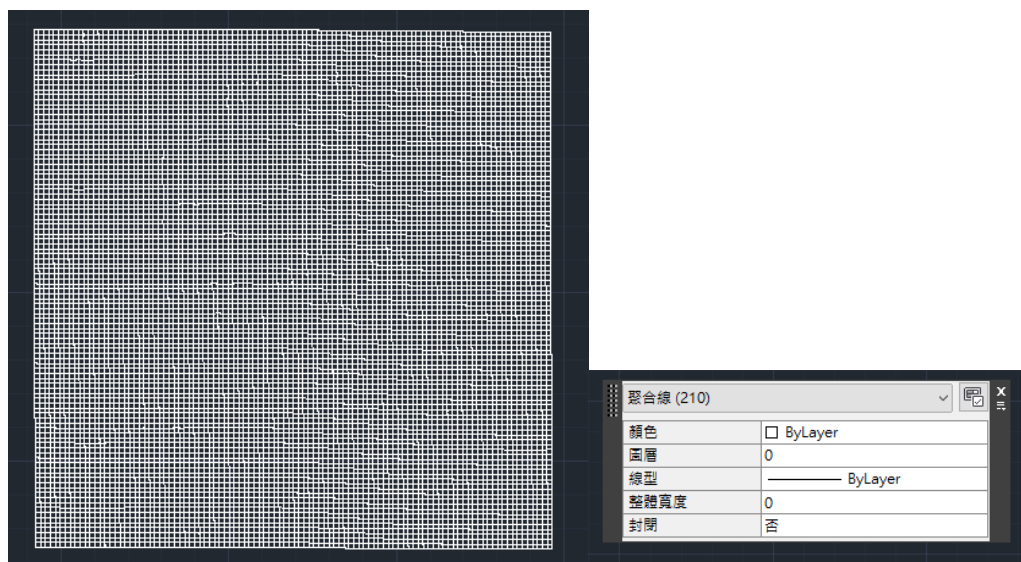
理想點座標 (X , Y)

名稱	修改日期	類型	大小	產生N組圖檔
CADfile1.bak	2017/8/3 下午 03...	BAK 檔案	2,	
 CADfile1.dwg	2017/8/3 下午 03...	DWG 檔案	2,	
CADfile2.bak	2017/8/3 下午 03...	BAK 檔案	2,	
 CADfile2.dwg	2017/8/3 下午 03...	DWG 檔案	2,	
CADfile3.bak	2017/8/3 下午 03...	BAK 檔案	2,	
 CADfile3.dwg	2017/8/3 下午 03...	DWG 檔案	2,	
CADfile3.dwl	2017/8/3 下午 03...	DWL 檔案		
CADfile3.dwl2	2017/8/3 下午 03...	DWL2 檔案		

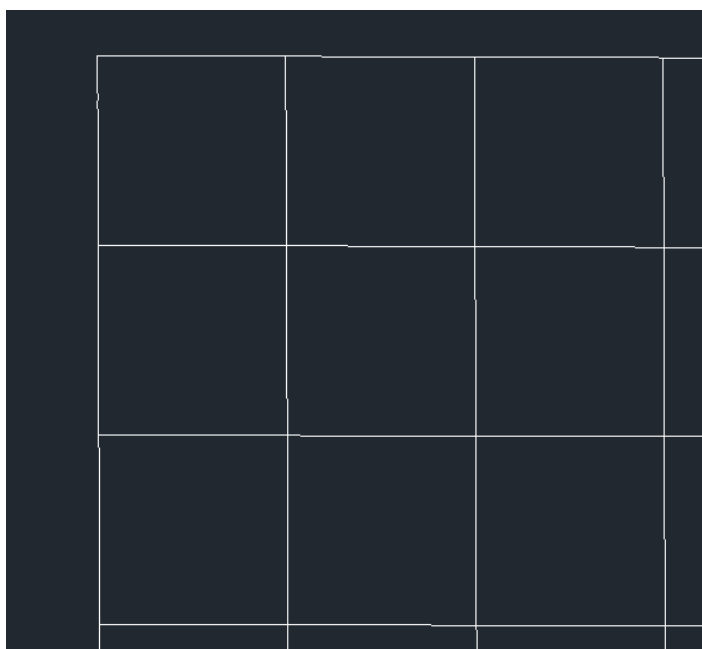
結果圖

輸入資料:210 條線, 360 條線, 420 條線

210 條線:

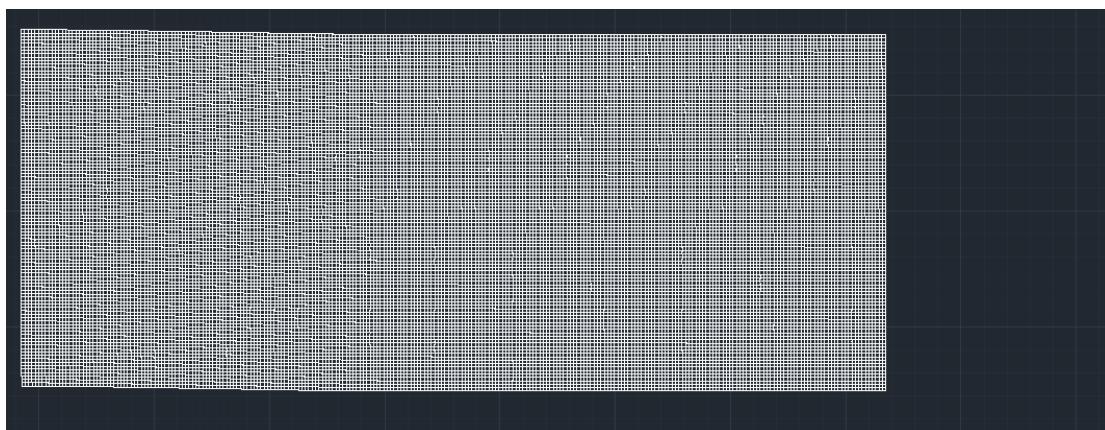


邊緣資訊:



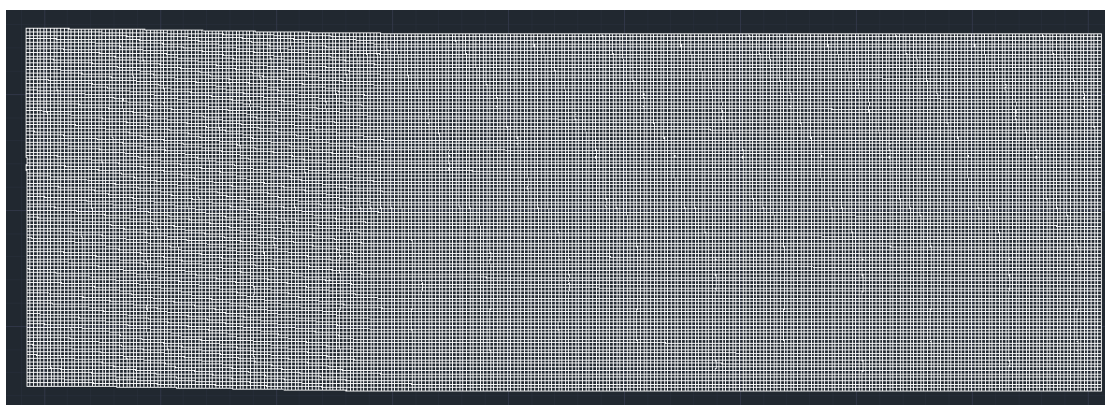
360 條線

聚合線 (360)	
顏色	<input type="checkbox"/> ByLayer
圖層	0
線型	———— ByLayer
整體寬度	0
封閉	否



420 條線

聚合線 (420)	
顏色	<input type="checkbox"/> ByLayer
圖層	0
線型	———— ByLayer
整體寬度	0
封閉	否

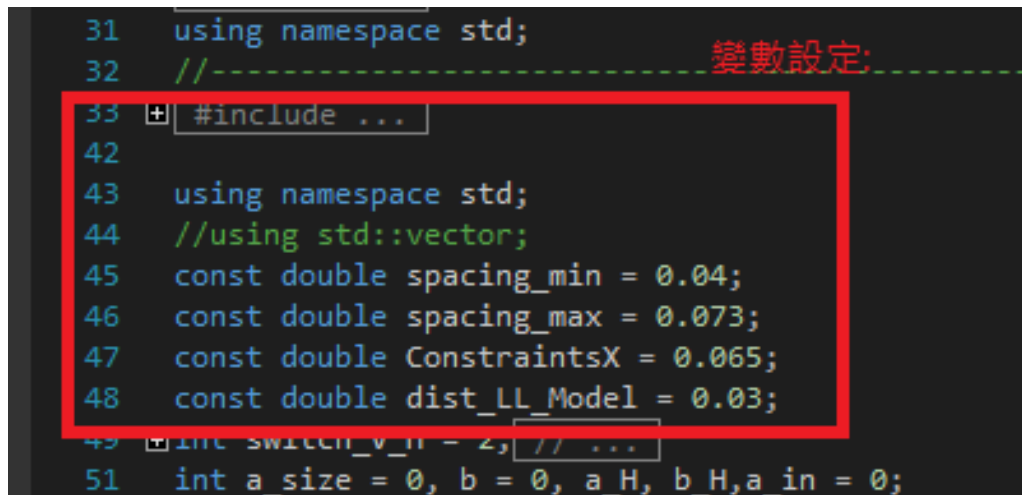


➤ 程式碼功能說明：

■ 變數設定

◆ 理想的網格大小設定為 0.065,設計閾值讓網格

- spacing_min ,spacing_max 是條件一的限制項參數
- ConstraintsX 為理想的網格距離;
- dist_LL_Model 為點和線性模型的距離限制;

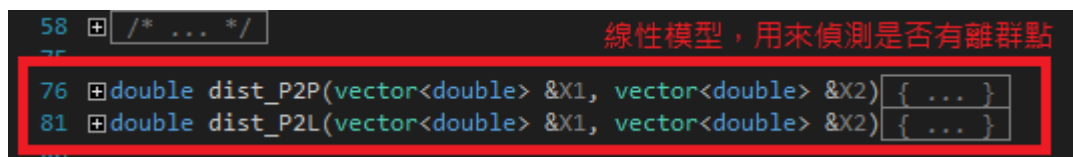


```
31 using namespace std;
32 //----- 變數設定:
33 + #include ...
42
43 using namespace std;
44 //using std::vector;
45 const double spacing_min = 0.04;
46 const double spacing_max = 0.073;
47 const double ConstraintsX = 0.065;
48 const double dist_LL_Model = 0.03;
49 + int switch_v_H = 2; // ...
51 int a_size = 0, b = 0, a_H, b_H, a_in = 0;
```

◆ 線性模型

■ Dist_P2P 為計算點到點的距離

■ Dist_P2L 為計算點到線的距離



```
58 + /* ... */
76 + double dist_P2P(vector<double> &X1, vector<double> &X2) { ... }
81 + double dist_P2L(vector<double> &X1, vector<double> &X2) { ... }
89
```

◆ Polyfit：用來建立線性模型的函式，輸入 10 個選擇點來

計算線性方程式，若超過 10 個點則移除最前面的點並將新取得的點加入最後面，一直保持 10 個點為單位來計算線性方程式。

- ◆ Ref_model：參考模型，用來作為計算預測點的條件之一
- ◆ InitialRef_model：參考模型的初始化步驟，用來初始化參考模型。

```
⊕vector<double> Polyfit(vector<double> &X_point, vector<double> &Y_point, int n){ ... }  
⊕void Ref_model(vector<double> &s, vector<int> &pos_outlier, vector<vector<int>> &dataXY,  
⊕void InitialRef_model(vector<double> &s, vector<int> &pos_outlier, vector<vector<int>> &dataXY,
```

◆ 存檔與座標轉換

- readFile：用來讀取真實作標點資訊,輸入檔案為 *.txt。
- gaussian_elimination：用來求解 Homography 矩陣，在 findHomography 函式中被呼叫，findHomography 則是用來計算與儲存 Homography 矩陣的函式。

```
⊕int * readFile(char* filename, int* a, int*b){ ... }  
⊕void gaussian_elimination(float *input, int n){ ... }  
⊕void findHomography(vector<vector<double>> src, vector<vec
```

◆ 客製化函式

- 函式執行順序為 2->3->1->4->5

- 透過呼叫函式 WAY_MEET 來呼叫檔案選取界面，用以選取檔案（可多選），要產生一張圖必須要有 6 個資料，分別為水平線的 X 做標以及 Y 做標,垂直線的 X 做標以及 Y 做標以及真實作標系的校正點與理想做標系的校正點。
- 將選取到的資料輸入進去 Our_Code 函式內執行演算法，輸出經過篩選與預測後的點資訊，並將其串連起來製作成腳本檔案 (*.SCR)。
- 透過 SETWIND 讀取腳本檔案自動繪製圖檔並使用 Savefile 來儲存圖檔最後由呼叫 Clear_renge 來清理版面，等待下一次的繪圖。

```

ACED_ARXCOMMAND_ENTRY_AUTO(CArxProject2App,TsdArxProject1,MyCommand1,SETWIND,ACRX_CMD_TRANSPARENT,NULL)
ACED_ARXCOMMAND_ENTRY_AUTO(CArxProject2App,TsdArxProject2,MyCommand2,WAY_MEET,ACRX_CMD_TRANSPARENT,NULL)
ACED_ARXCOMMAND_ENTRY_AUTO(CArxProject2App,TsdArxProject3,MyCommand3,Our_Code,ACRX_CMD_TRANSPARENT,NULL)
ACED_ARXCOMMAND_ENTRY_AUTO(CArxProject2App,TsdArxProject4,MyCommand4,Savefile,ACRX_CMD_TRANSPARENT,NULL)
ACED_ARXCOMMAND_ENTRY_AUTO(CArxProject2App,TsdArxProject5,MyCommand5,Clear_renge,ACRX_CMD_TRANSPARENT,NULL)

```

static void TsdArxProject1MyCommand1(void){ ... }	讀取腳本檔案
static void TsdArxProject2MyCommand2(void){ ... }	檔案選取介面
// ...	
static void TsdArxProject3MyCommand3(void){ ... }	座標轉換與誤差偵測之演算法
static void TsdArxProject4MyCommand4(void){ ... }	儲存圖檔
static void TsdArxProject5MyCommand5(void){ ... }	清除版面