Final project

ORB-SLAM & COLMAP

作業內容

錄製自己的影片，使用兩種不同的視覺定位方法(ORB-slam & COLMAP)
建立相機軌跡並比較。

繳交檔案：將以下檔案包成zip檔，group_0.zip

1. 一組一份10頁以內的報告，轉成pdf檔
2. code
3. 實驗過程影片

繳交期限：6/21(一) 晚上 23:59前

ORB-SLAM2

- Windows
  - https://github.com/phdsky/ORBSLAM24Windows?fbclid=IwAR2Q3DCAgMn-gzNp_E_5fMk1aBld_rPPW99qojWJm62c7fSUdPfT6lEpmaY
  - Prerequisite from github
    - OpenCV: newer than 2.4.13
    - Cmake: at least be 2.8
    - Visual Studio VS2013 (corresponding to opencv's vc12)
  - Prerequisite from TAs
    - OpenCV: 3.4.6-vc14_vc15
    - Cmake: newest
    - Visual Studio VS2017 (corresponding to opencv's vc15)
    - Git: newest

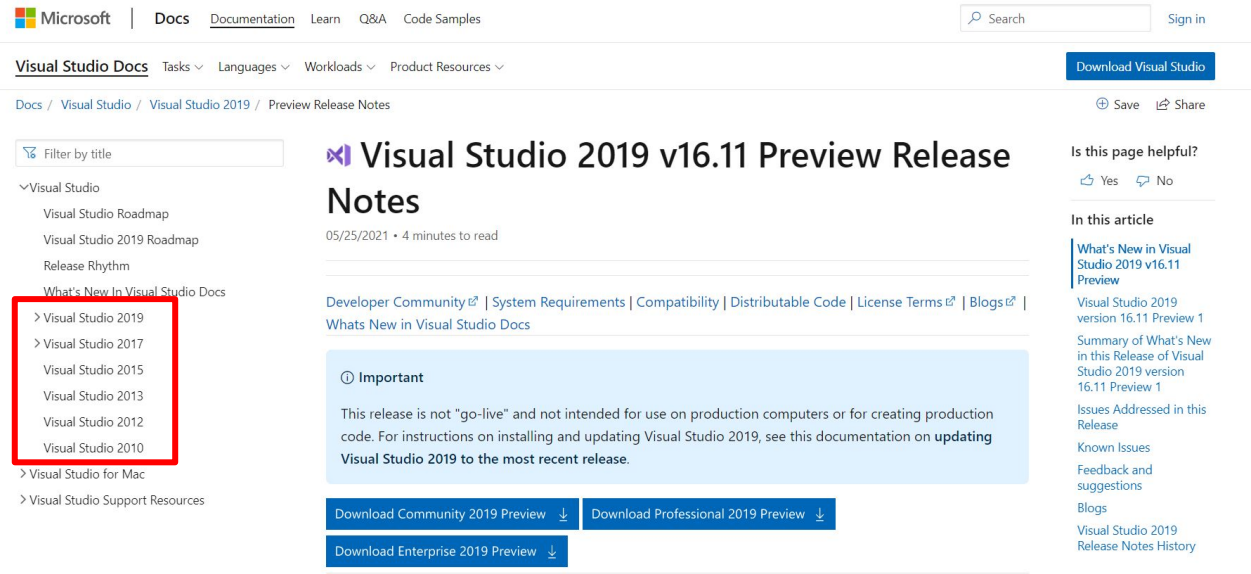
Prerequisite

- OpenCV: newer than 2.4.13
  - https://sourceforge.net/projects/opencvlibrary/files/opencv-win/2.4.13/opencv-2.4.13.exe/download
  - Add environment variable “PATH”
    - *YOUR_OWN_PATH\opencv\build*
    - *YOUR_OWN_PATH\opencv\build\x64\vc12\bin*

- Cmake: at least be 2.8
  - https://cmake.org/download/
- Git
  - https://git-scm.com/downloads

Prerequisite

- Visual Studio 2013
  - 可以從交大的filezilla載
  - 官網 : https://docs.microsoft.com/en-us/visualstudio/releases/2019/release-notes-preview

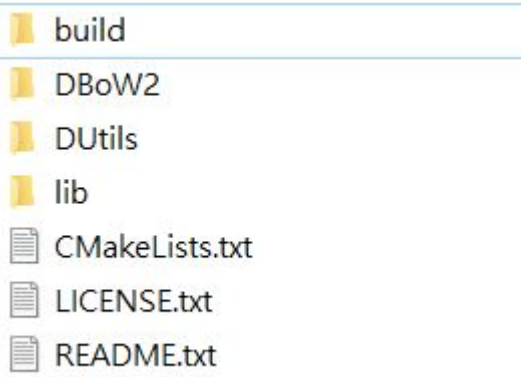


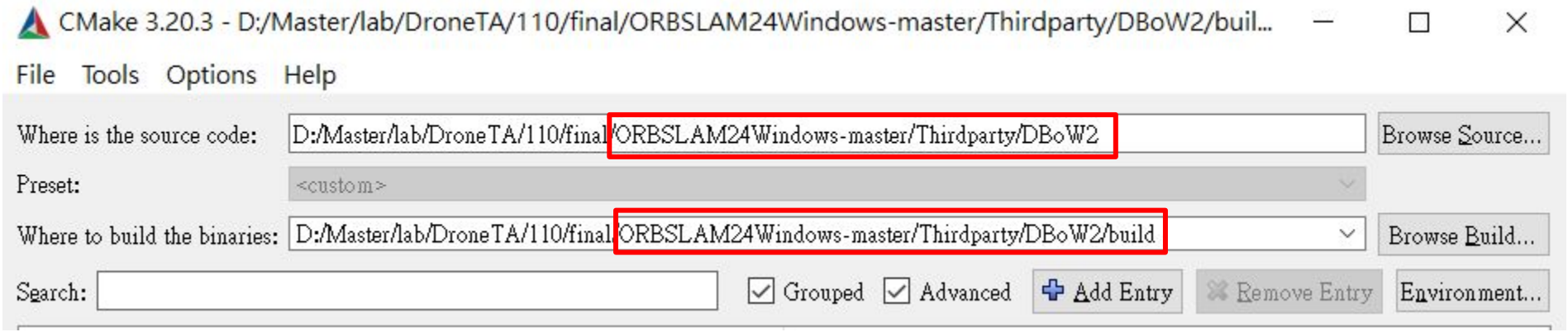
Steps

- Compile the projects in **Thirdparty** folder
  - DBoW2
  - eigen(not need to build)
  - g2o
  - Pangolin
- Build ORBSLAM24Windows

Steps - DBoW2

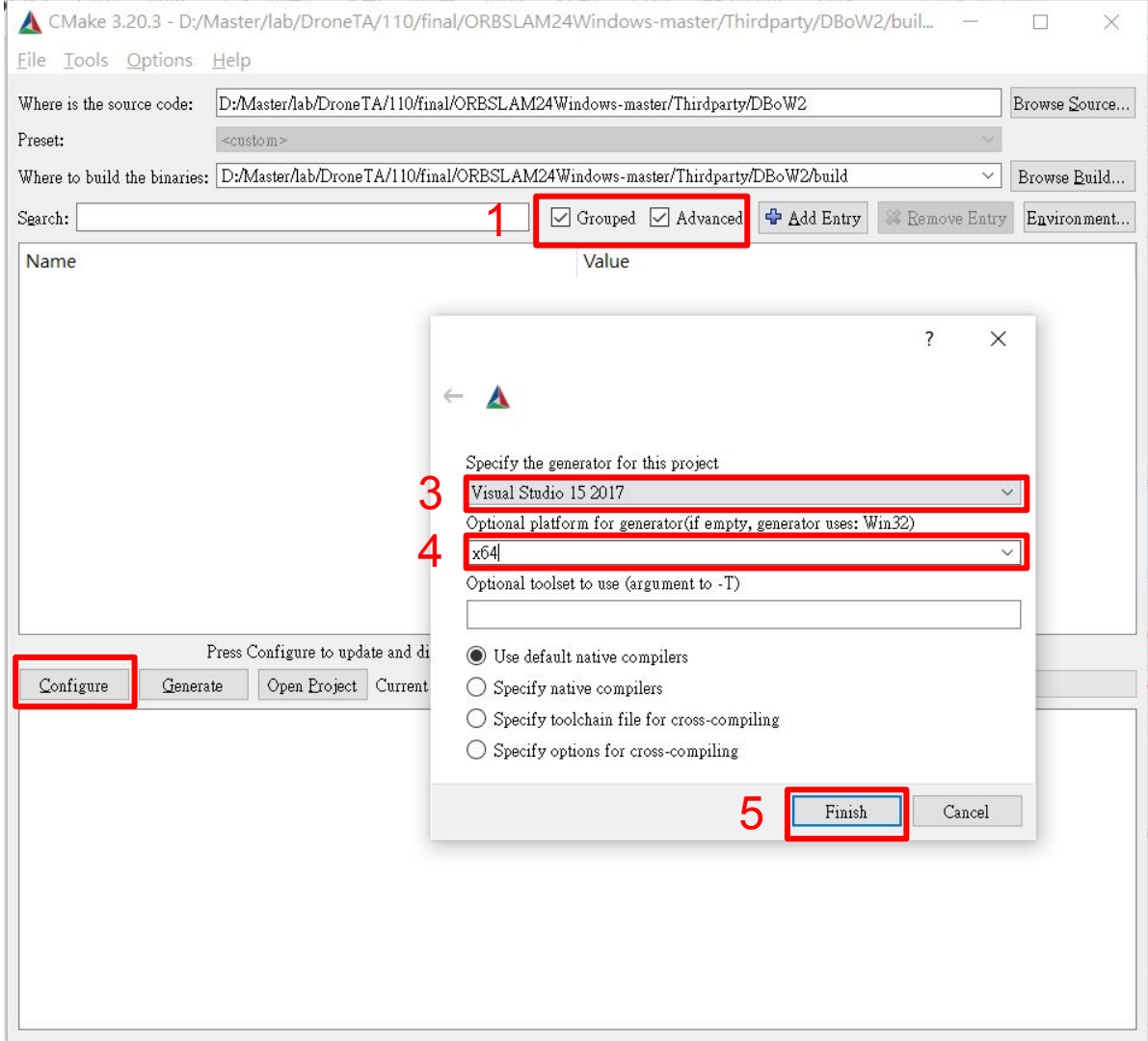
- 新增folder “build”
- Cmake
  - Browse Source.. 選*DBoW2*
  - Browse Build.. 選*DBoW2/build*





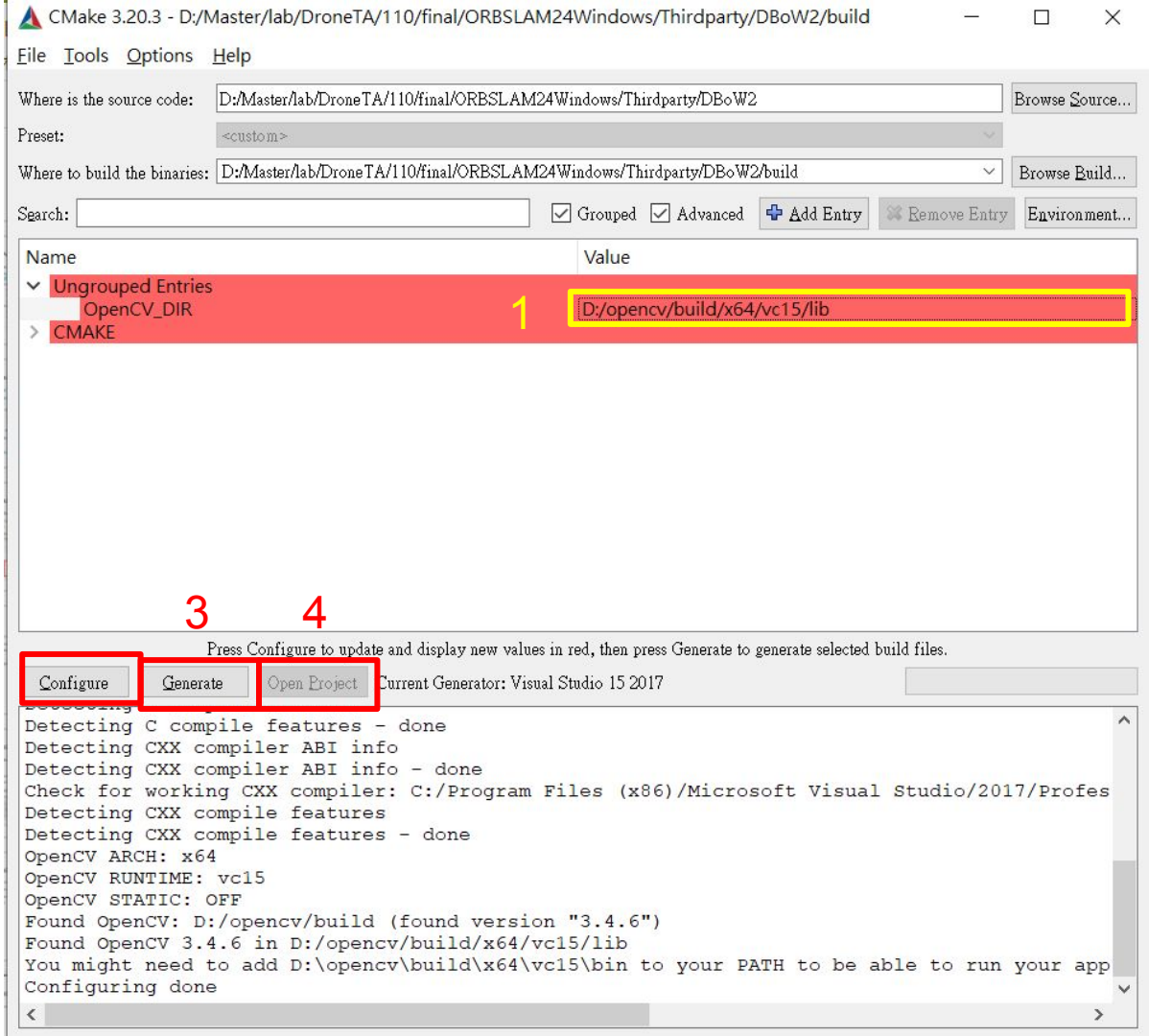
Steps - DBoW2

1. 把Grouped和Advanced勾起來
2. 點configure
3. 依自己的VS選版本
4. 選x64版本
5. 點Finish



Steps - DBoW2

1. 把opencv的lib path填上去
2. 再Configure一次
3. Generate
4. Open Project



Steps - DBoW2

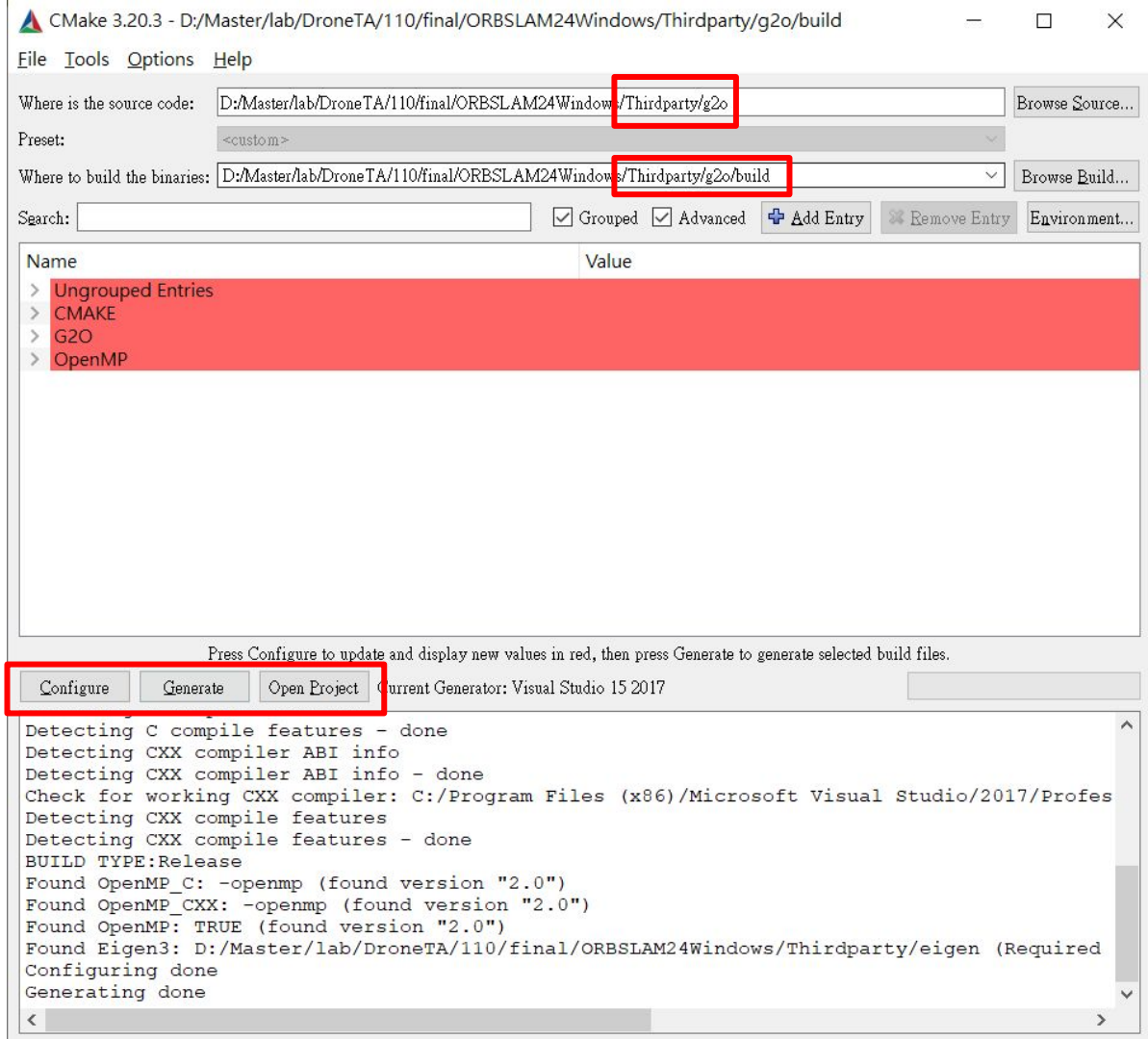
1. 選Release模式
2. 在ALL_BUILD項目點右鍵選擇"建置"
3. DBoW2 build完成！

```
========== 建置: 3 成功、0 失敗、0 最新、0 略過 ==========
```

Steps - g2o

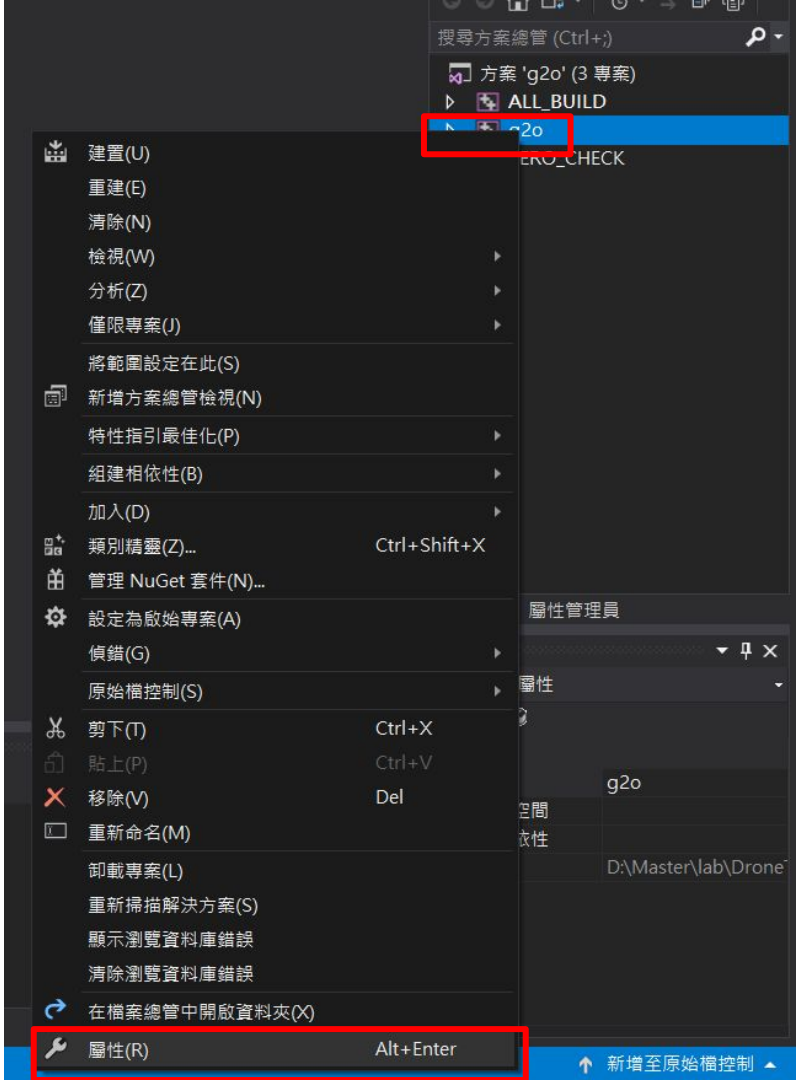
- Cmake和DBoW2一樣

(不用理中間的紅框)



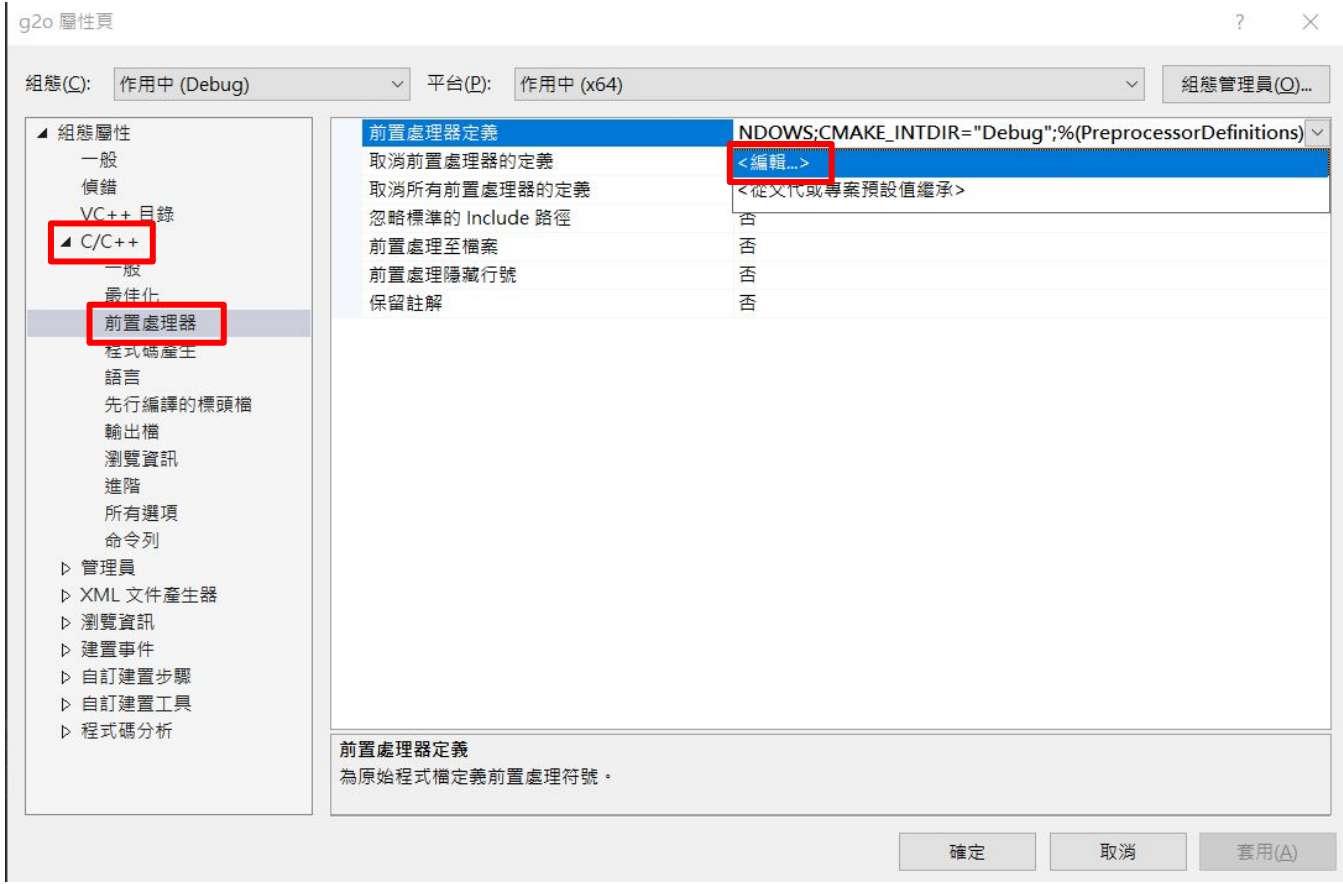
Steps - g2o

1. 點g2o項目右鍵
2. 選屬性

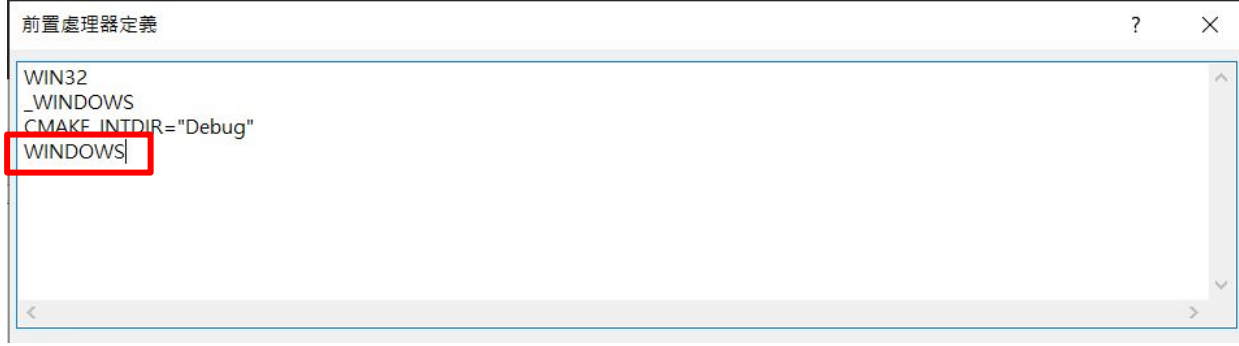


Steps - g2o

1. C/C++
2. 前置處理器
3. 編輯



Steps - g2o

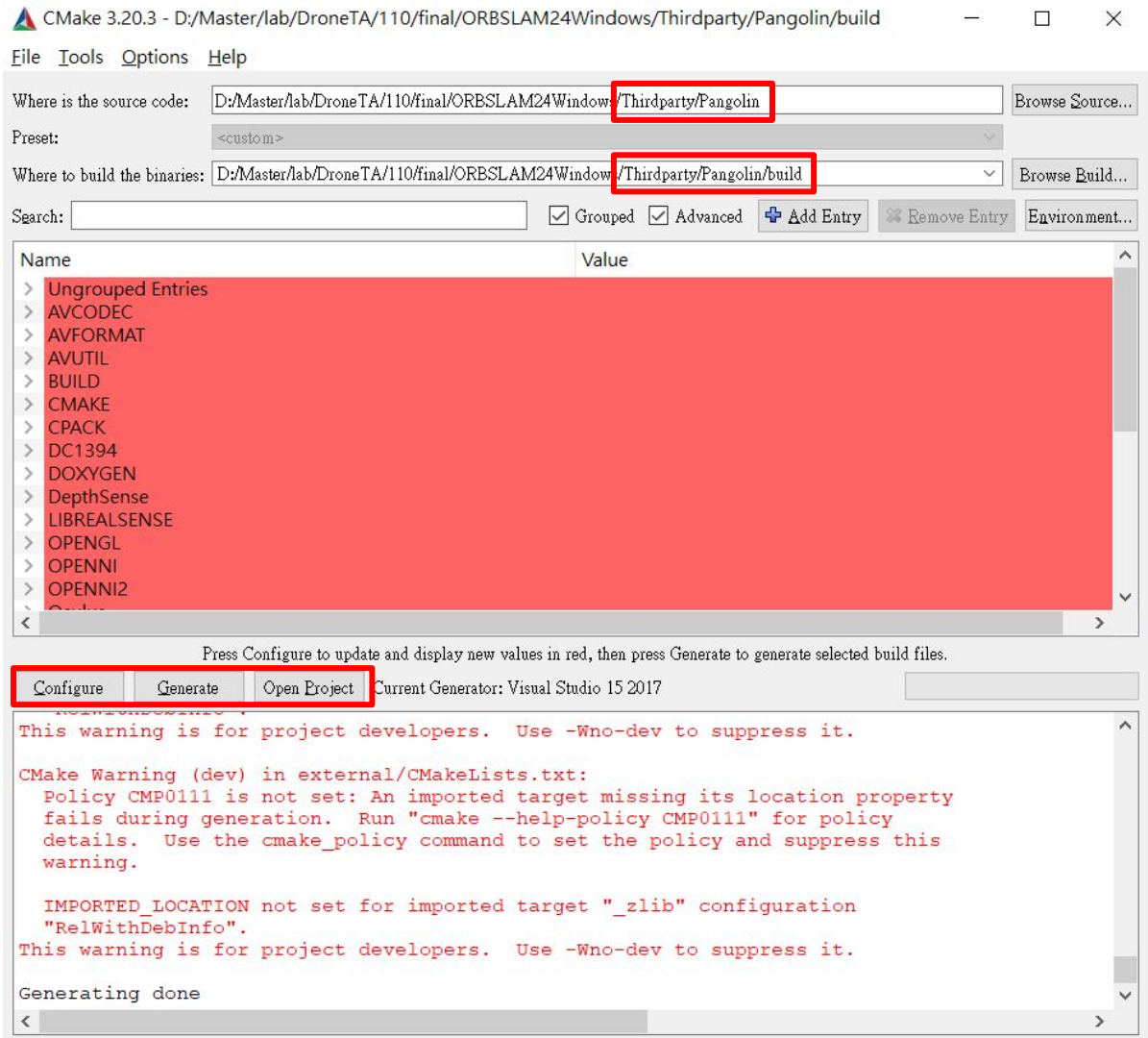


1. 加WINDOWS在最下層
2. 選Release模式
3. 在ALL_BUILD項目點右鍵選擇"建置"
4. 會有一個失敗
5. 一樣的動作再加WINDOWS，再建置一次
6. g2o build完成！

```
========== 建置: 1 成功、0 失敗、2 最新、0 略過 ==========
```

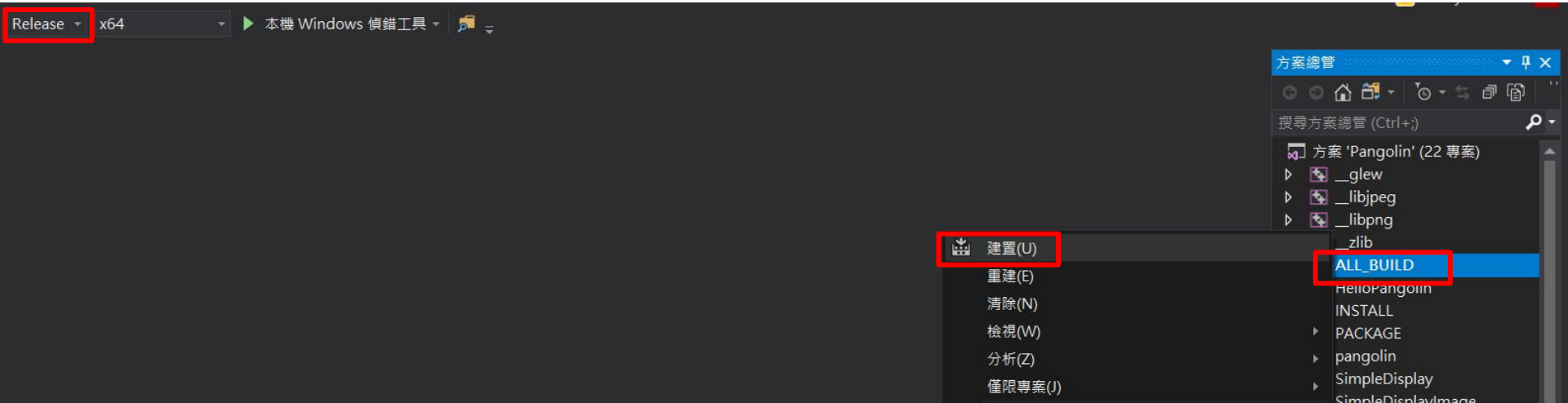
Steps - Pangolin

1. Cmake同上
2. 有很多紅框不理他→



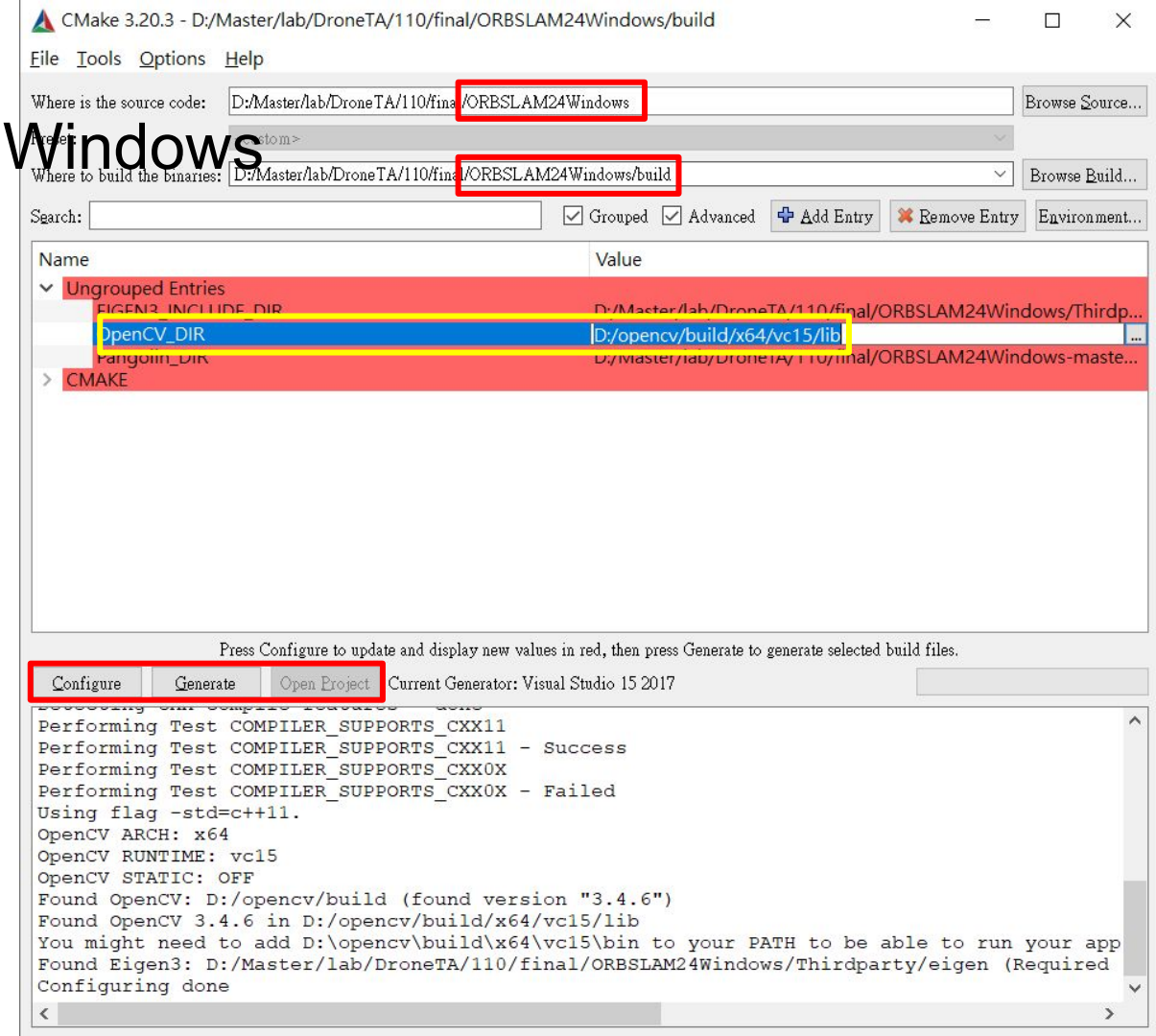
Steps - Pangolin

1. 選Release模式
2. 在ALL_BUILD項目點右鍵選擇"建置"
3. pthread.lib的失敗不用理他
4. Pangolin build完成！



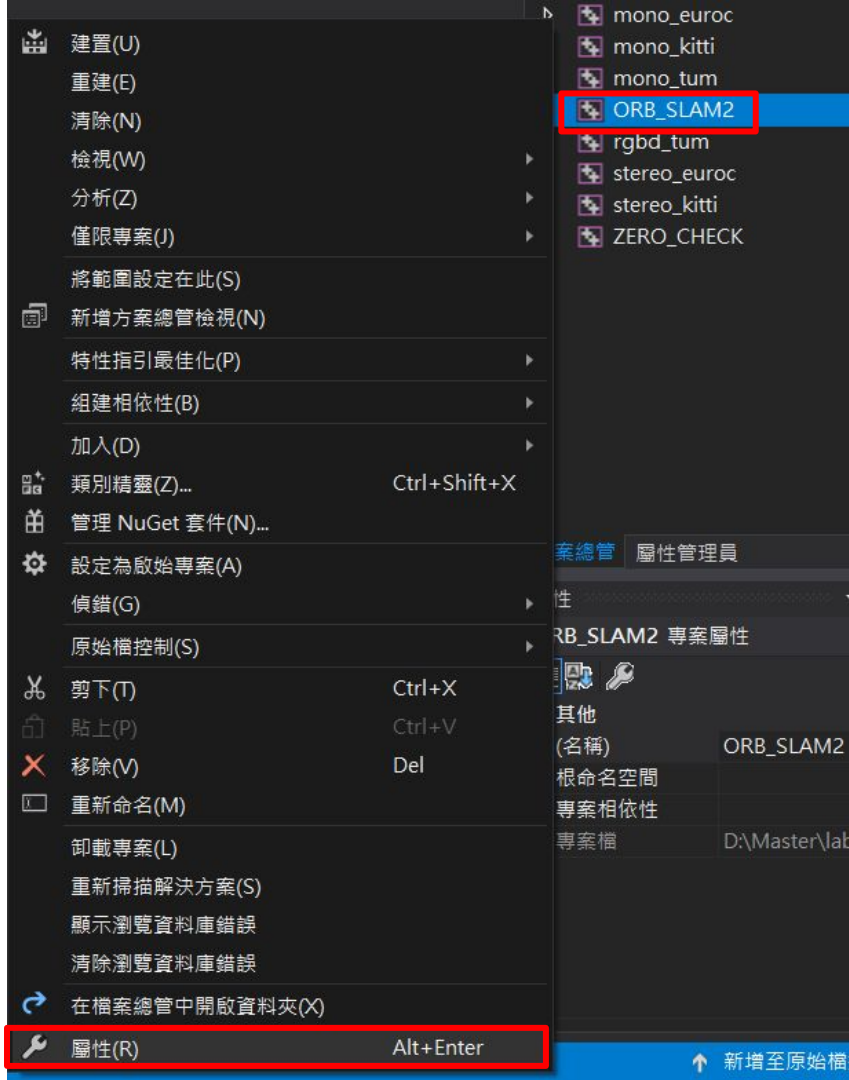
Steps - ORBSLAM24Windows

1. Cmake同上
2. 會報錯
3. 填上opencv的lib path
4. Configure
5. Generate
6. Open project



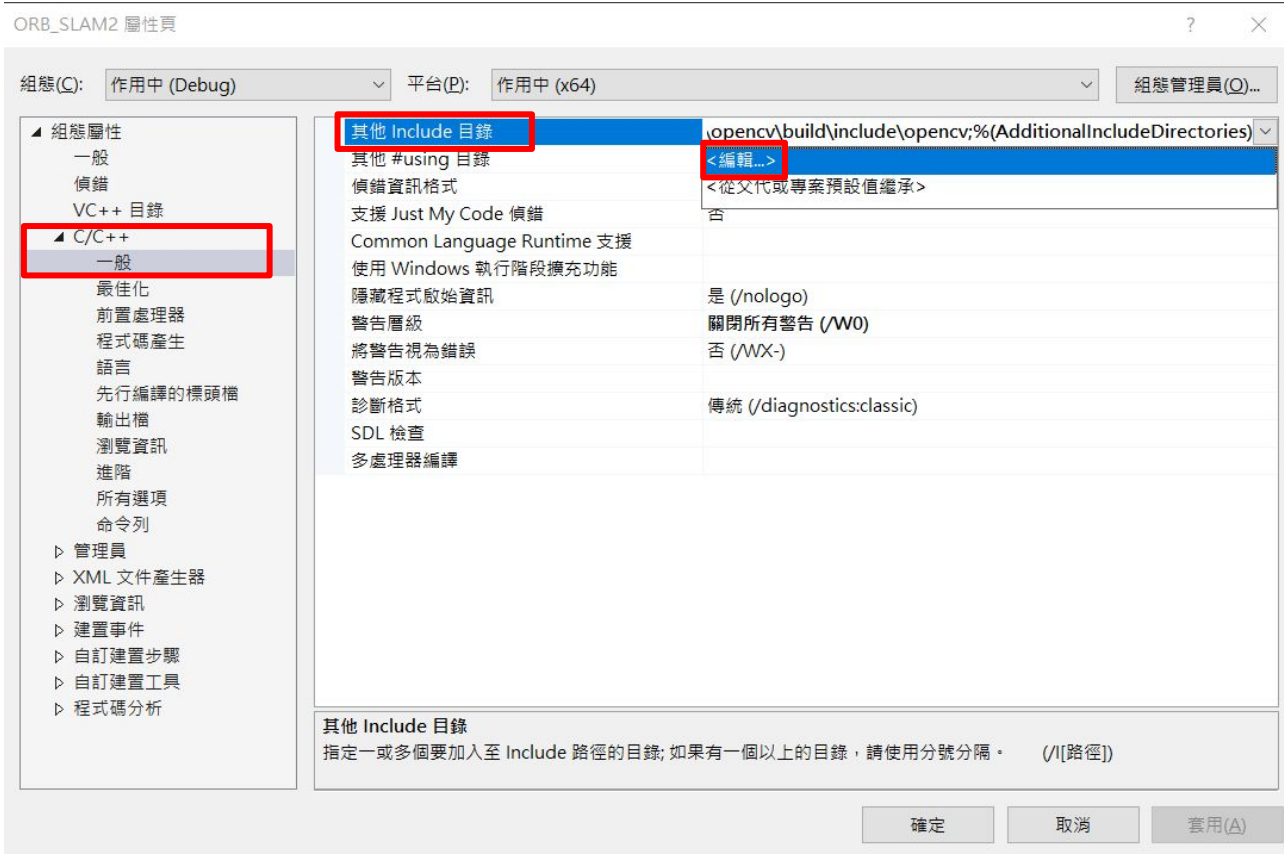
Steps - ORBSLAM24Windows

1. 選Release模式
2. 在ORB_SLAM2項目點右鍵
3. 屬性



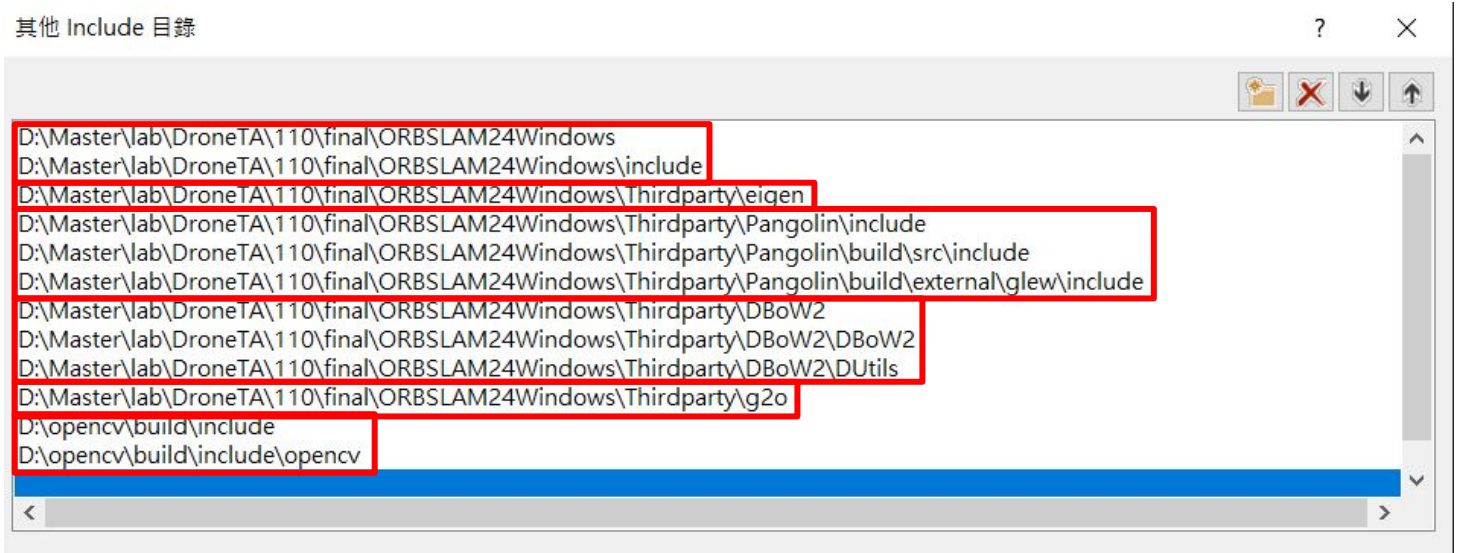
Steps - ORBSLAM24Windows

1. C/C++ → 一般
2. 其他include目錄
3. 編輯



Steps - ORBSLAM24Windows

1. 把缺的include path補上
2. 總共12個



Steps - ORBSLAM24Windows

1. 選Release模式
2. 在ORB_SLAM2項目點右鍵選擇"建置"

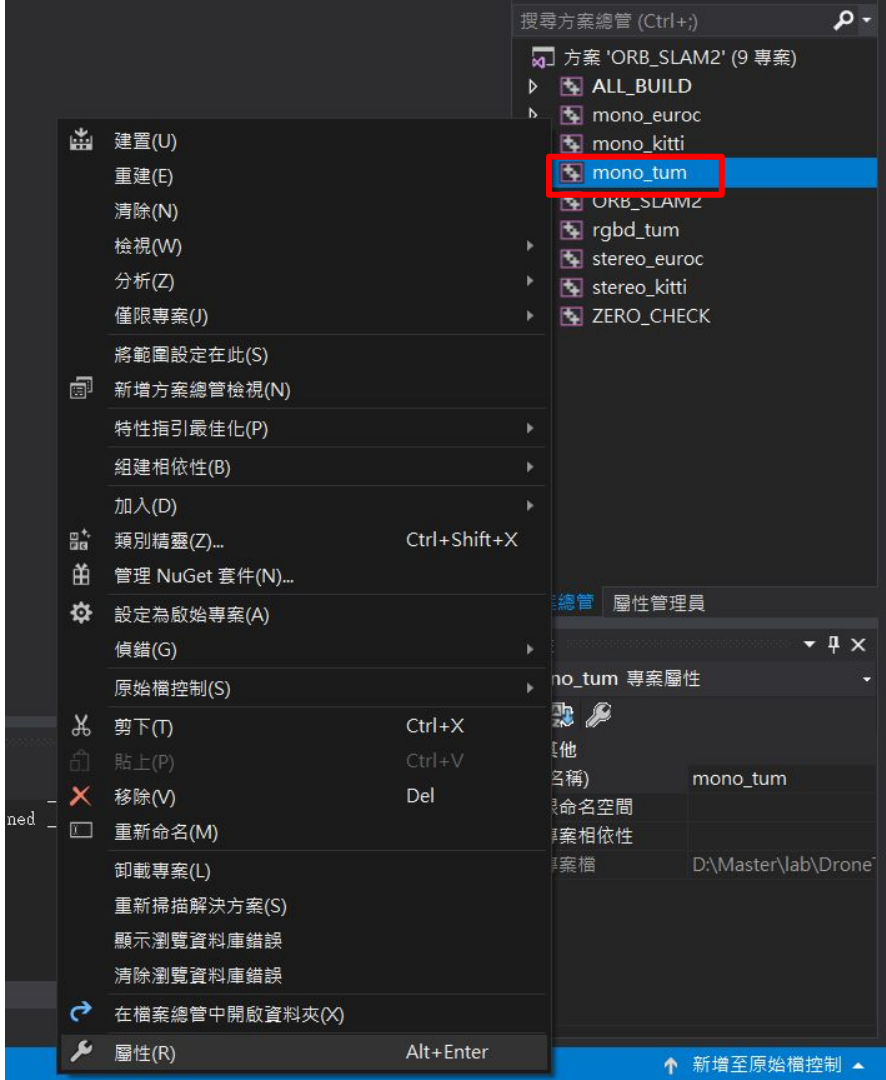
```
========== 建置: 2 成功、0 失敗、0 最新、0 略過 ==========
```

Steps - ORBSLAM24Windows

1. 在mono_tum項目同上再做一次

   (補include)

   (建置)

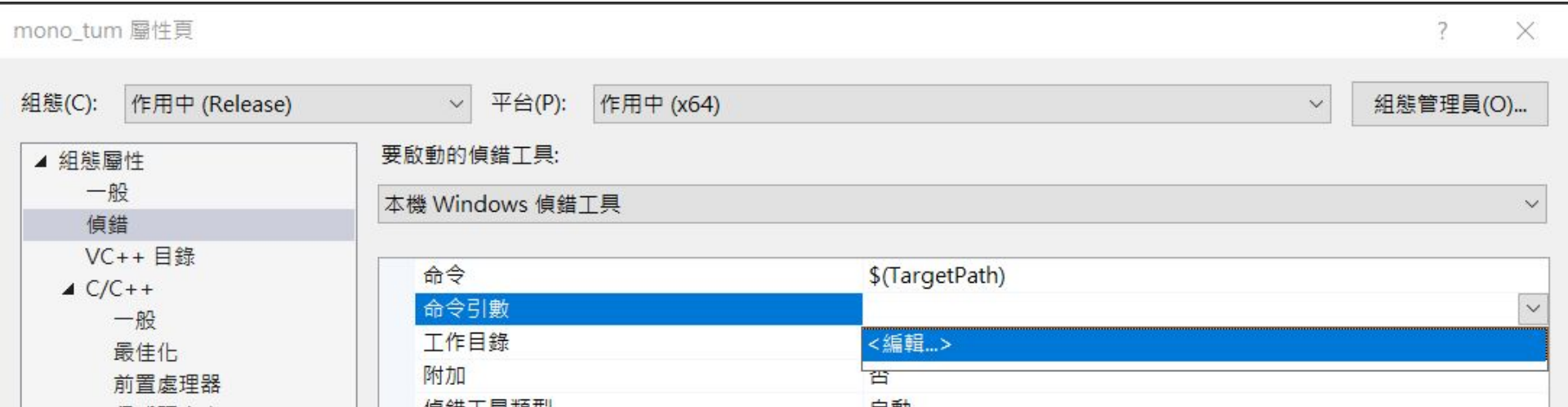


Test

1. 開啟ORBSLAM24Windows/build/ORB_SLAM2.sln
2. 選Release模式
3. 下載dataset: https://vision.in.tum.de/rgbd/dataset/freiburg2/rgbd_dataset_freiburg2_desk.tgz
4. 右鍵mono_tum項目 → 屬性

Test

5. 組態屬性 → 偵錯 → 命令引數 → 編輯

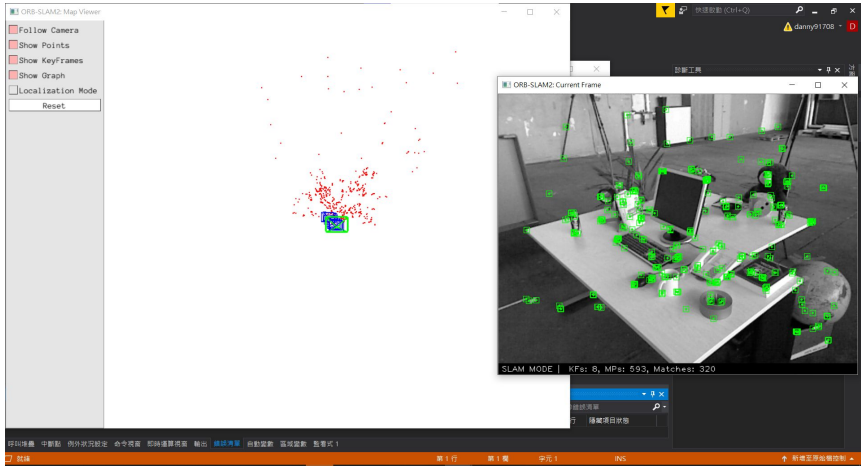


Test

6. 輸入三個parameters, 用空格隔開:

   - path_to_vocabulary: ORBSLAM24Windows/Vocabulary/ROBvoc.txt
   - path_to_settings: ORBSLAM24Windows/Examples/Monocular/TUM2.yaml, 包含相機參數和orb slam的參數
   - path_to_sequence: rgbd_dataset_freiburg2_desk資料夾的path

7. 右鍵mono_tum項目 → 偵錯 → 開始執行個體

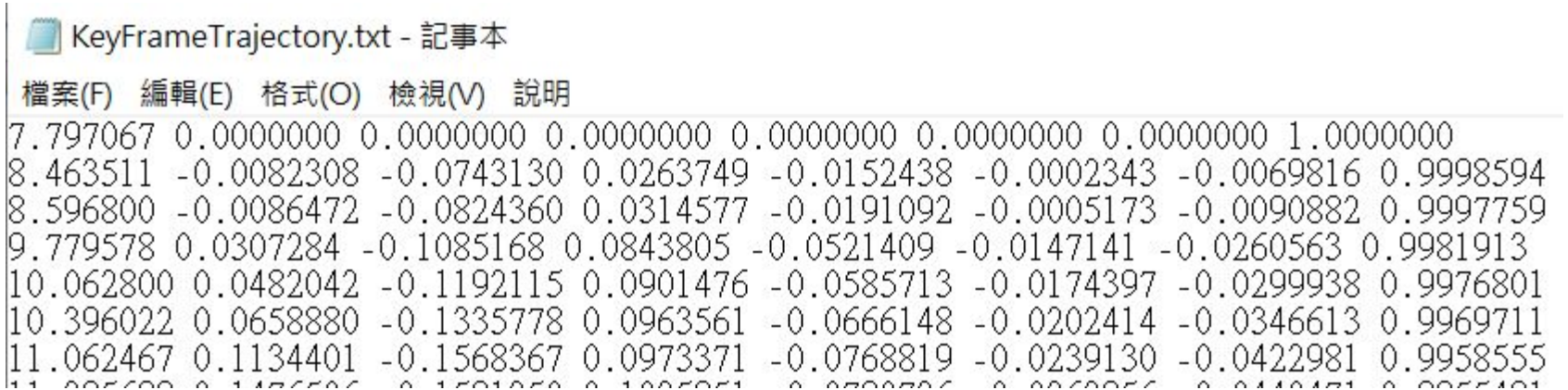


ORB-SLAM2

- Linux
  - https://github.com/raulmur/ORB_SLAM2
  - Prerequisite from github
    - C++11 or C++0x Compiler
    - Pangolin
    - Opencv: Required at leat 2.4.3.
    - Eigen3: at least 3.1.0.
    - DBoW2 & g2o in the Thirdparty folder

ORB-slam的Output

在 build 資料夾中 KeyFrameTrajectory.txt



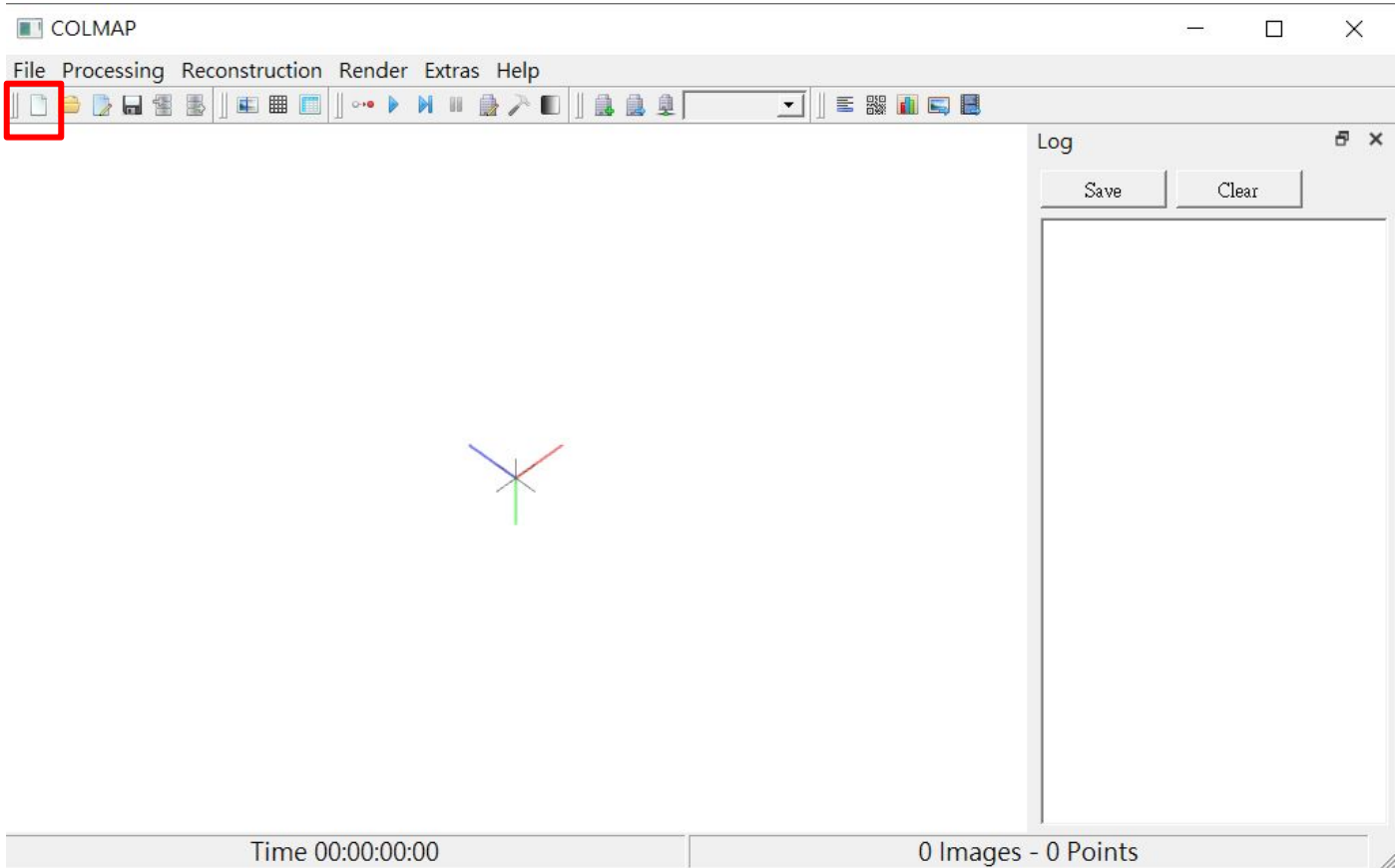
依序為 timestamp, 相機(x, y, z), 相機旋轉(q_x, q_y, q_z, q_w)

COLMAP

- https://github.com/colmap/colmap/releases?fbclid=IwAR38THauVythCkkbdUs4fcjv85muGyr34wHMIUqREK9v5dZrSsEKMHZbinQ
- COLMAP-3.6-windows-no-cuda.zip即為windows免安裝版
- 先建模型，再用測試影片重建相機位置

COLMAP

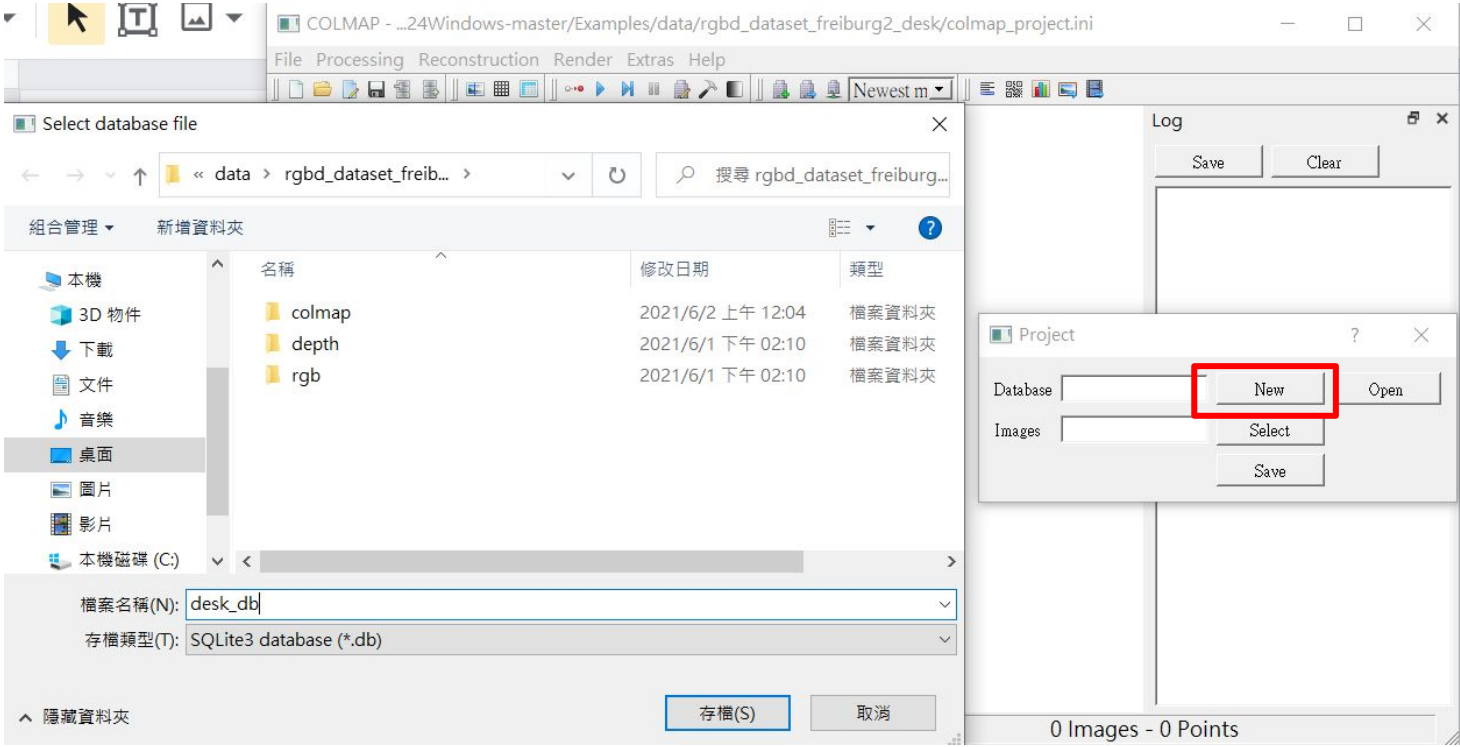
1. new project



COLMAP

2. Database

在要存的資料夾打一個名稱，會存成.db檔

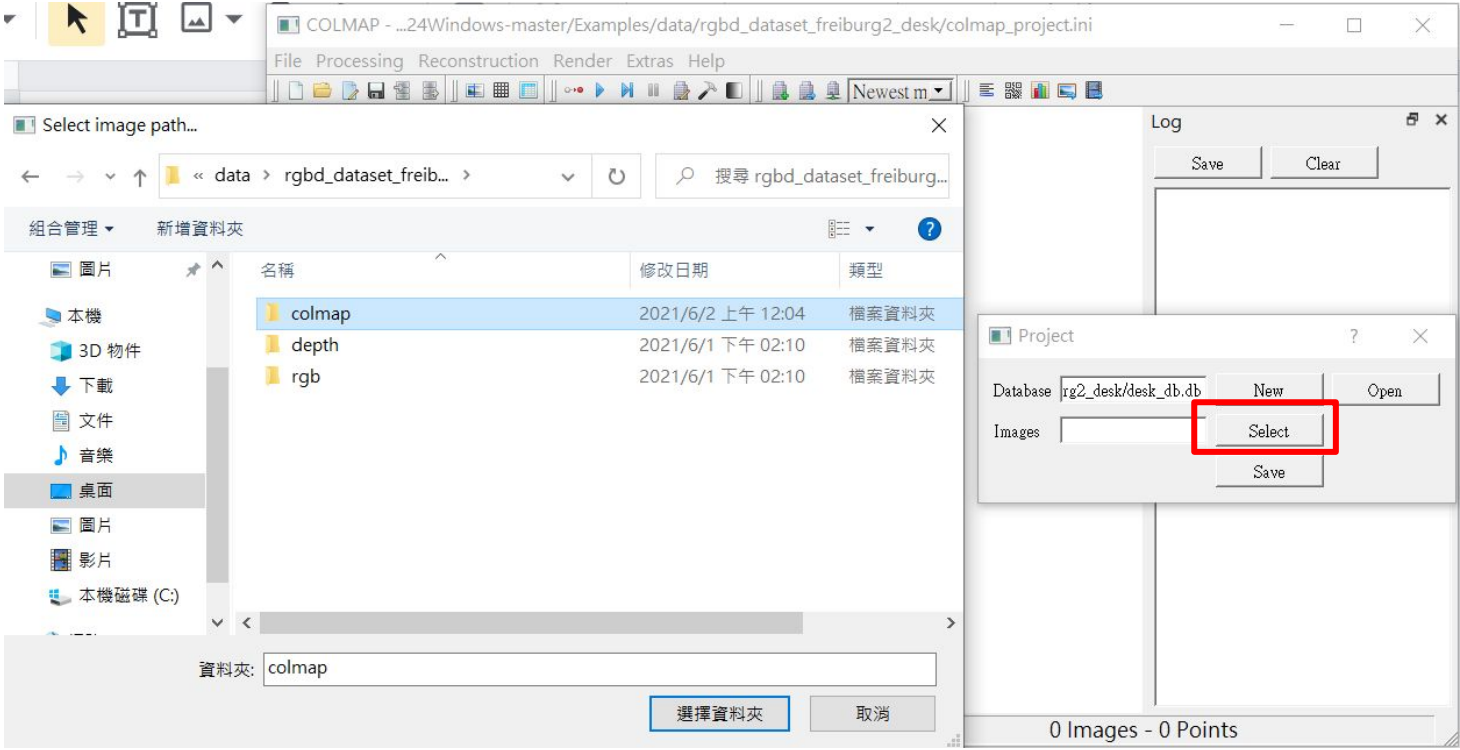


COLMAP

3. Image

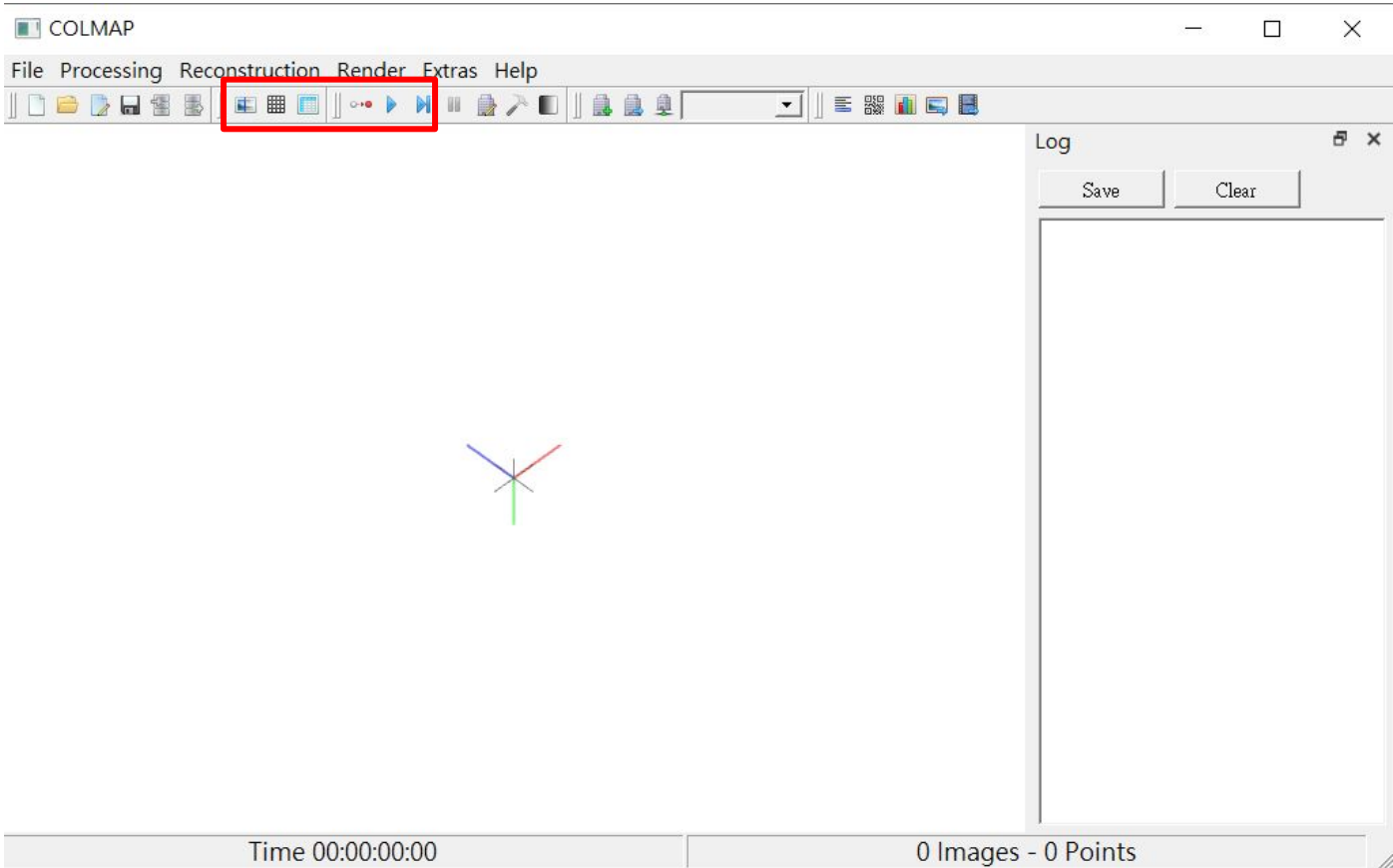
選擇輸入圖片資料夾

然後按Save



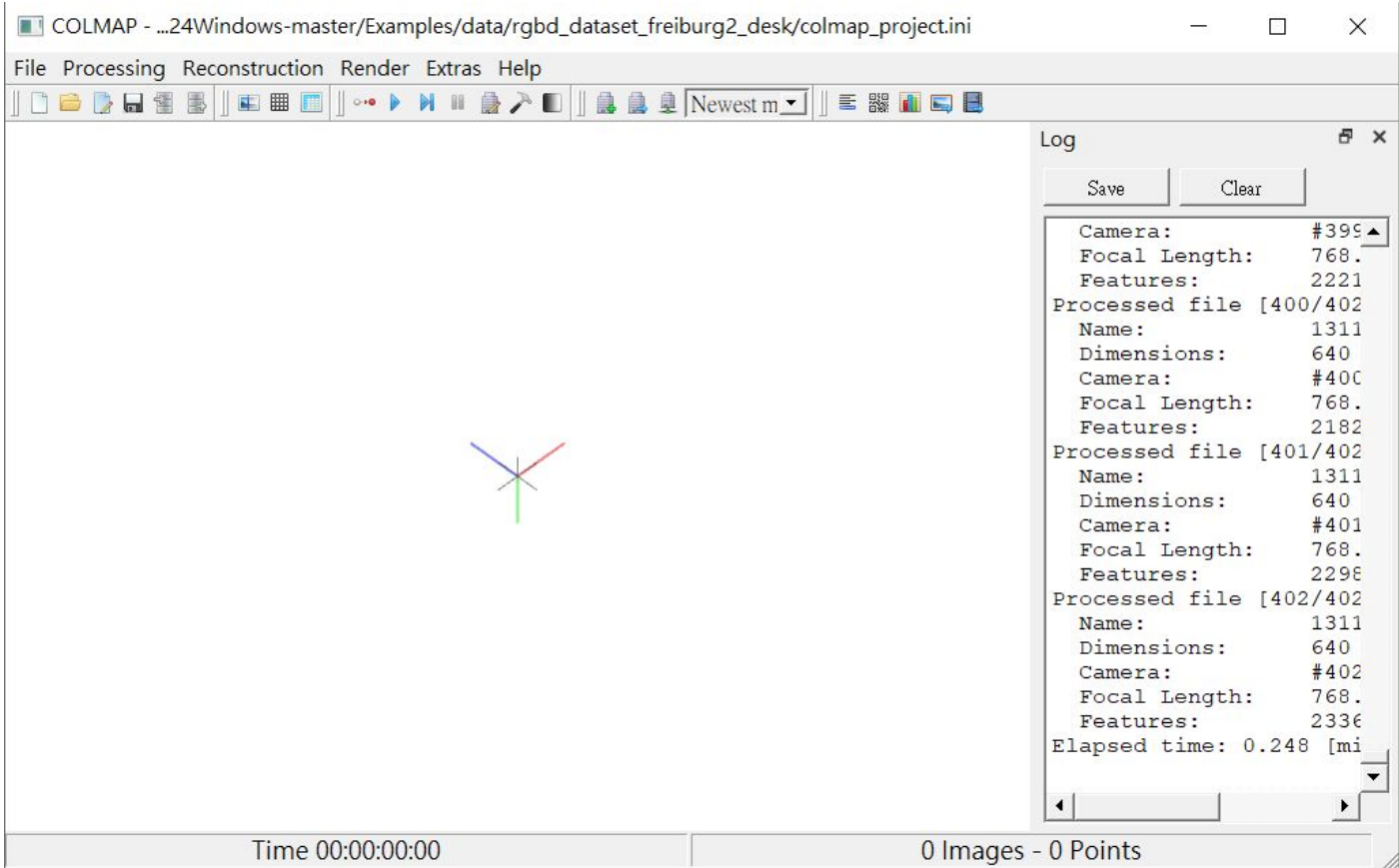
COLMAP

4. Feature Extraction



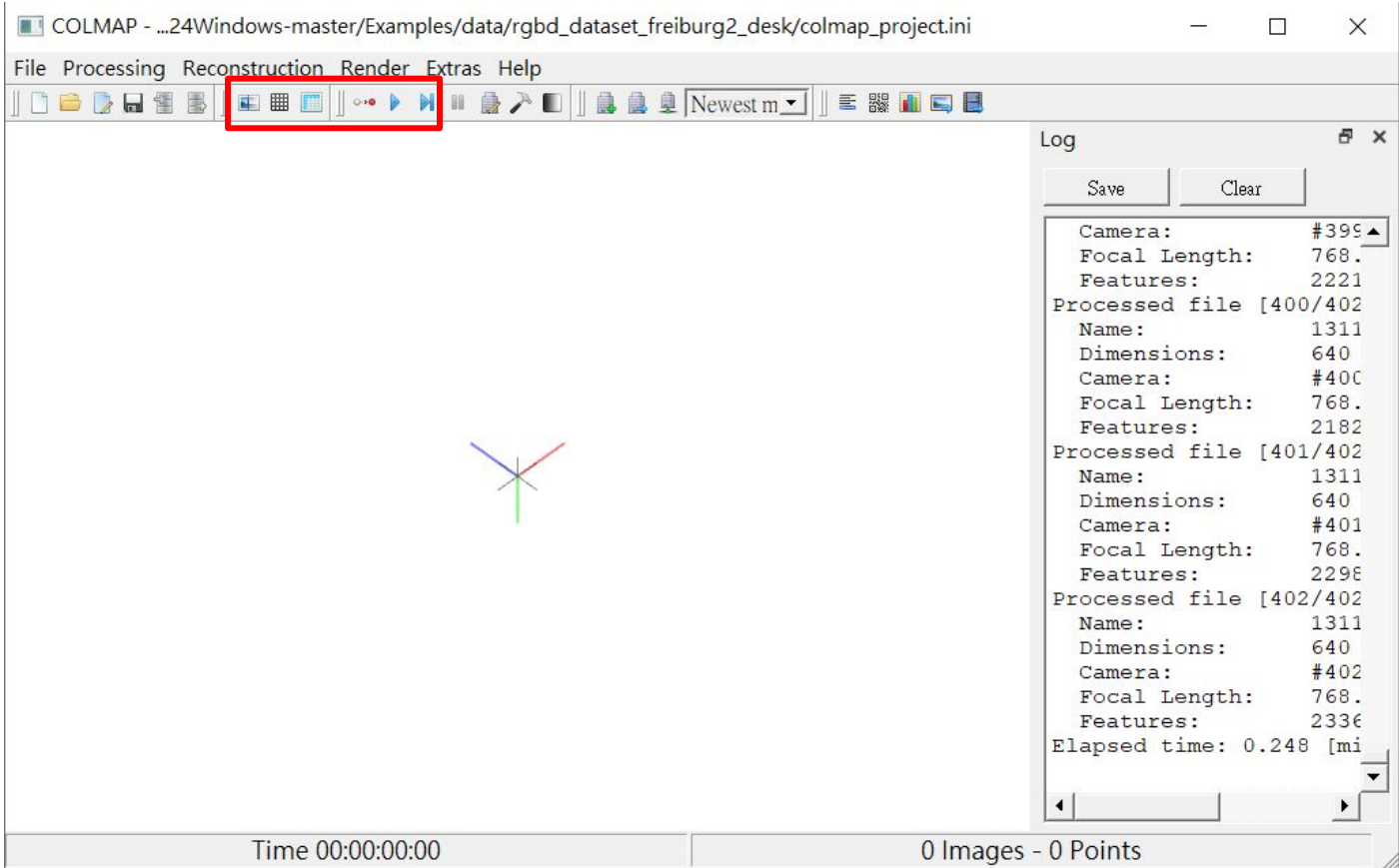
COLMAP

4. Feature Extraction



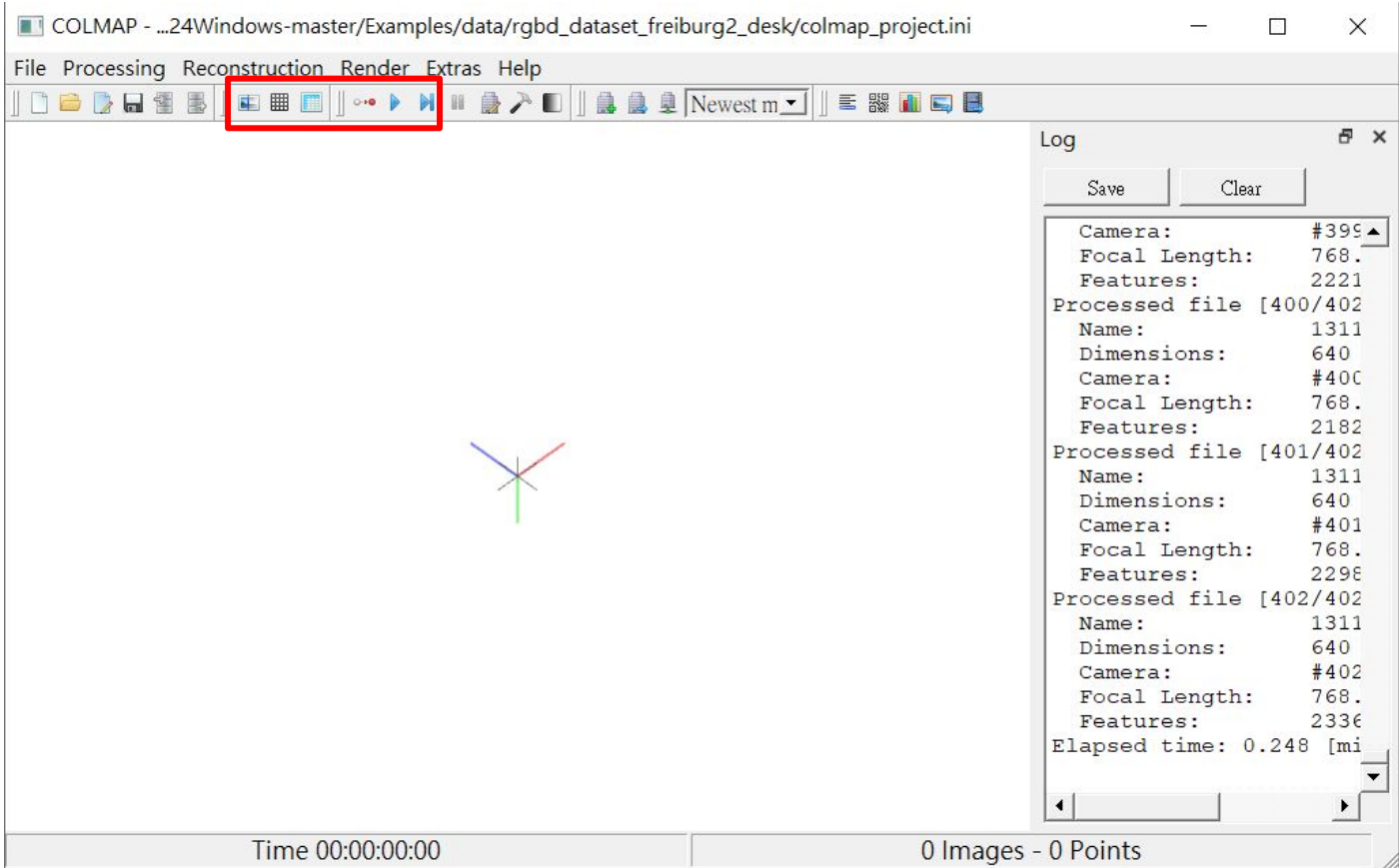
COLMAP

5. Feature Matching



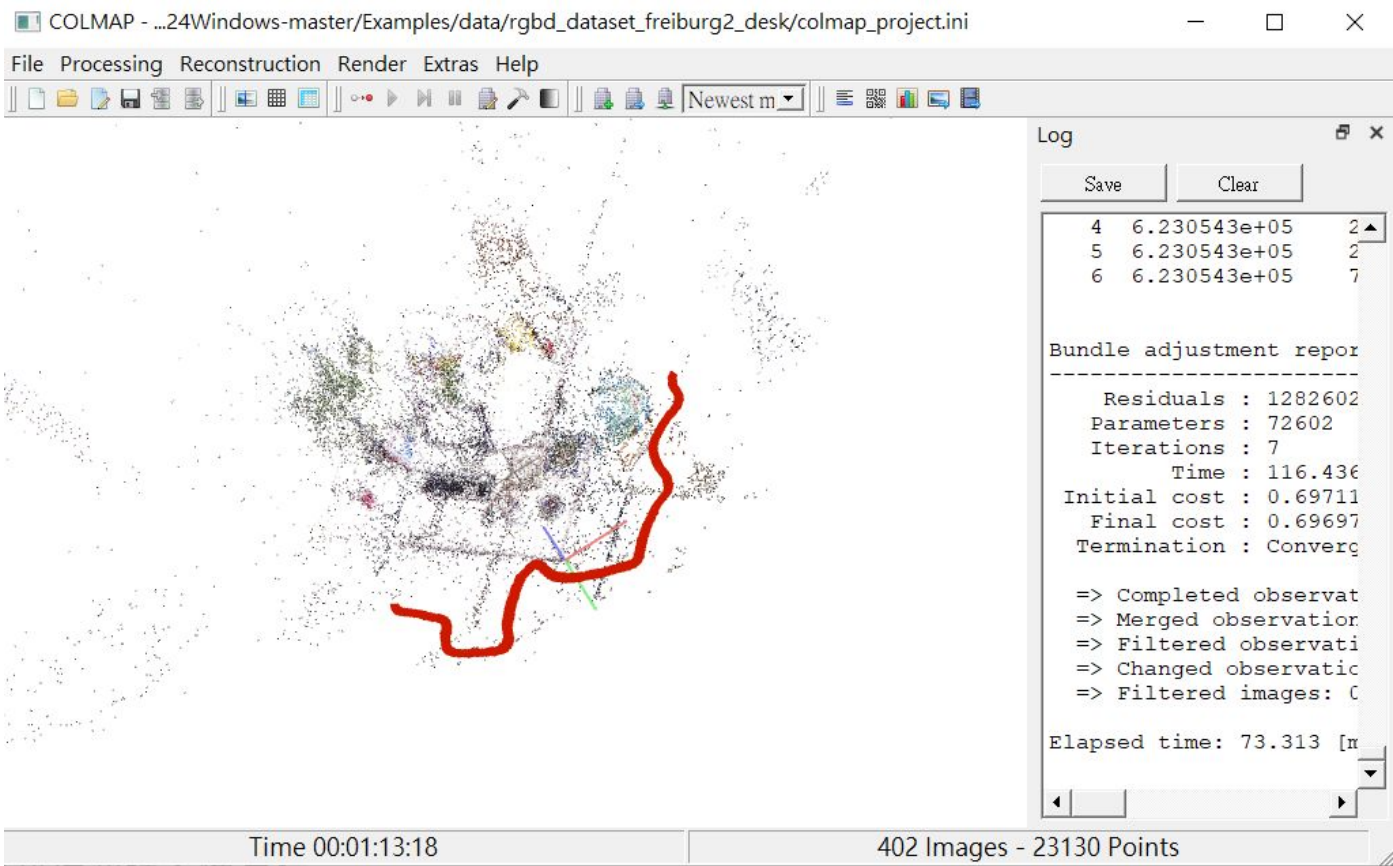
COLMAP

6. Start Reconstruction



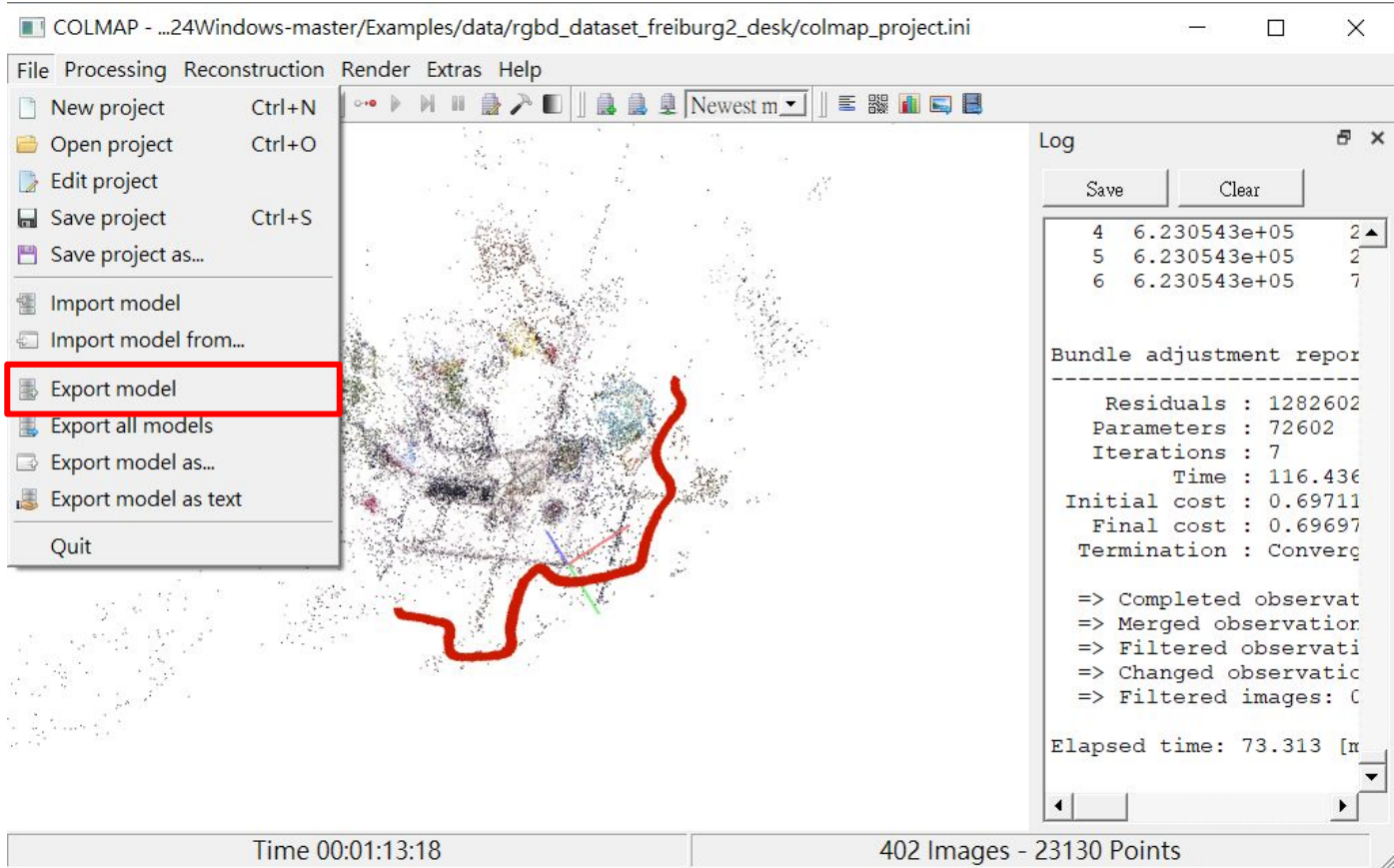
COLMAP

6. Start Reconstruction



COLMAP

7. Export model



COLMAP

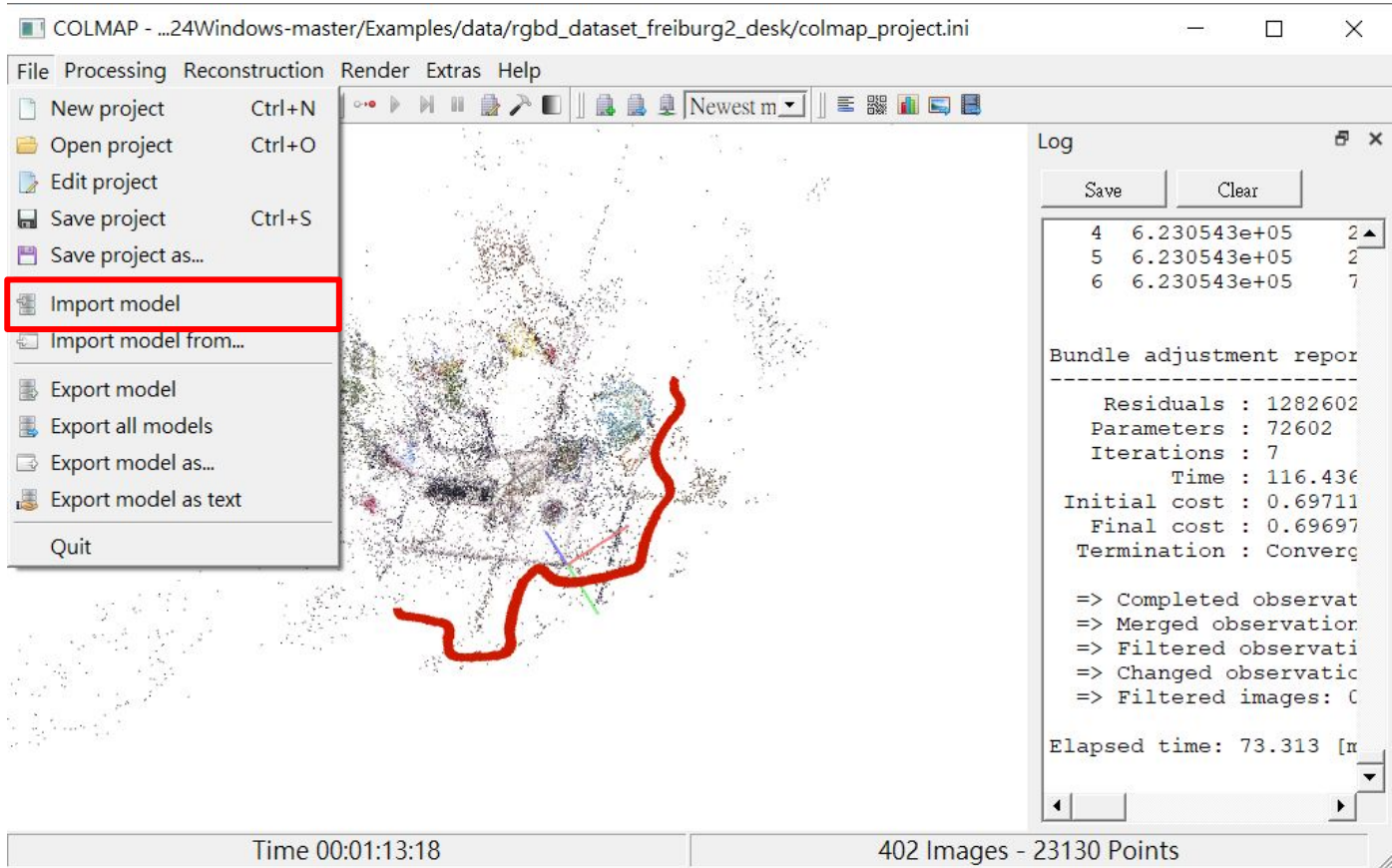
8. 把測試影像丟到原始影像資料夾裡

可以分成兩個資料夾

COLMAP

9. Import model

把剛剛Export的模型讀入



COLMAP

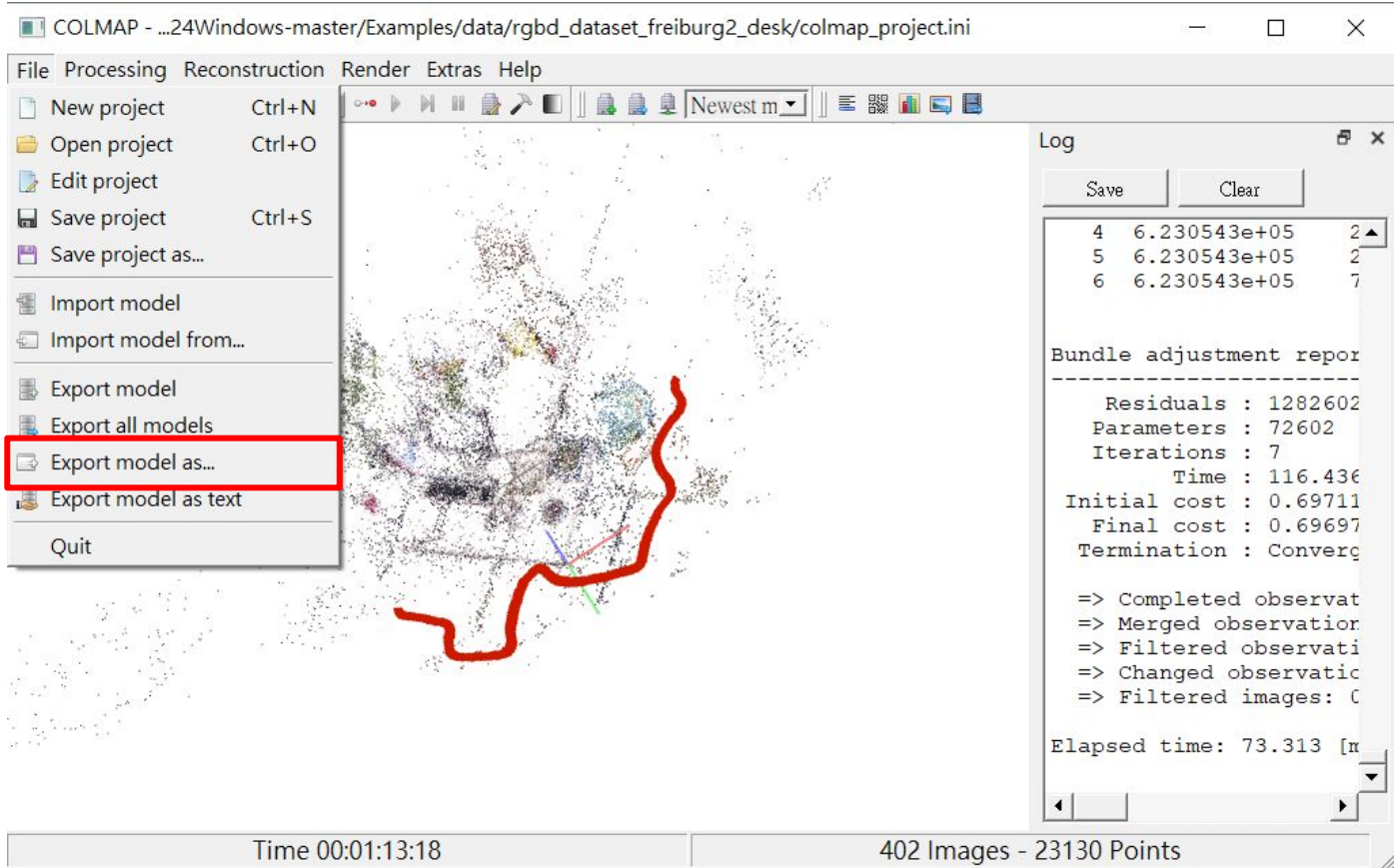
10. 重複4~6步驟

COLMAP會自動找還沒處理過的影像，也就是第8步新增的那些測試影像

COLMAP

11. Export model as

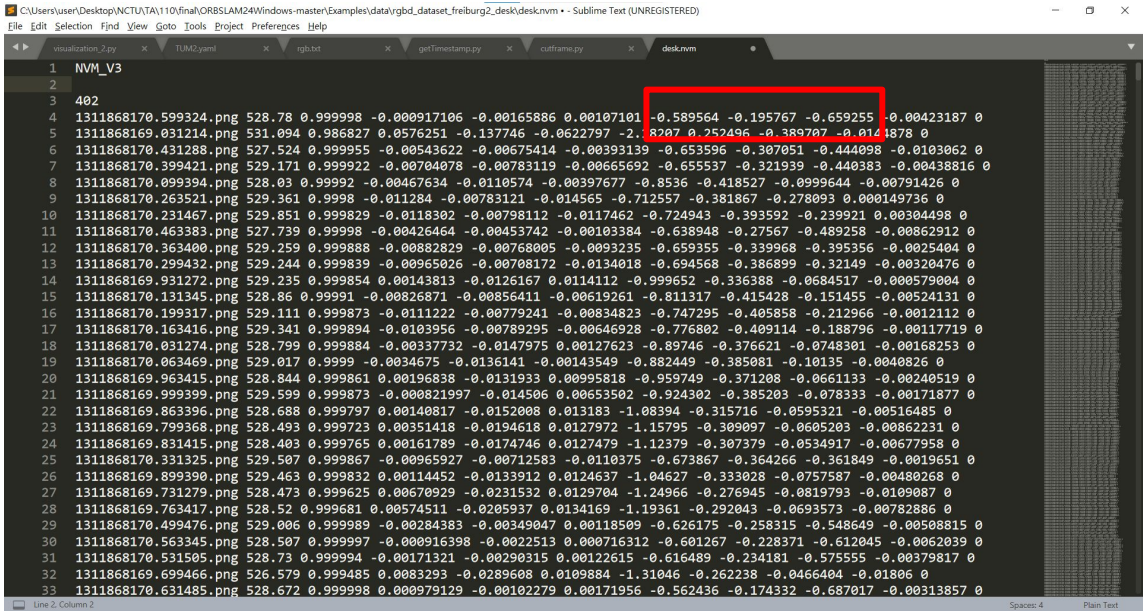
存成.nvm檔



COLMAP的output

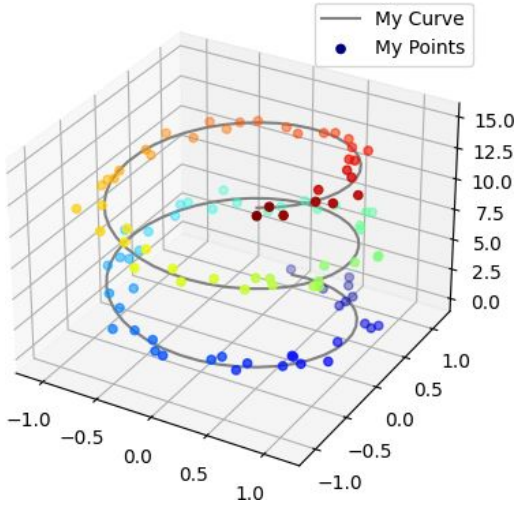
.nvm file

<Camera> = <File name> <focal length> <quaternion WXYZ> <camera center> <radial distortion> 0



Evaluation

- 把COLMAP重建的相機軌道當作Ground truth
- ORB-SLAM的坐標系和COLMAP的坐標系不一樣
- 把ORB-SLAM重建的相機軌道轉換到COLMAP的坐標系，和Ground truth計算誤差
- 把兩個相機軌道plot出來



報告內容

- 比較兩種方法之間的差異
- 比較建出來的模型還有各自的相機定位
- 兩個相機軌道的圖
- ORB-SLAM實驗過程影片