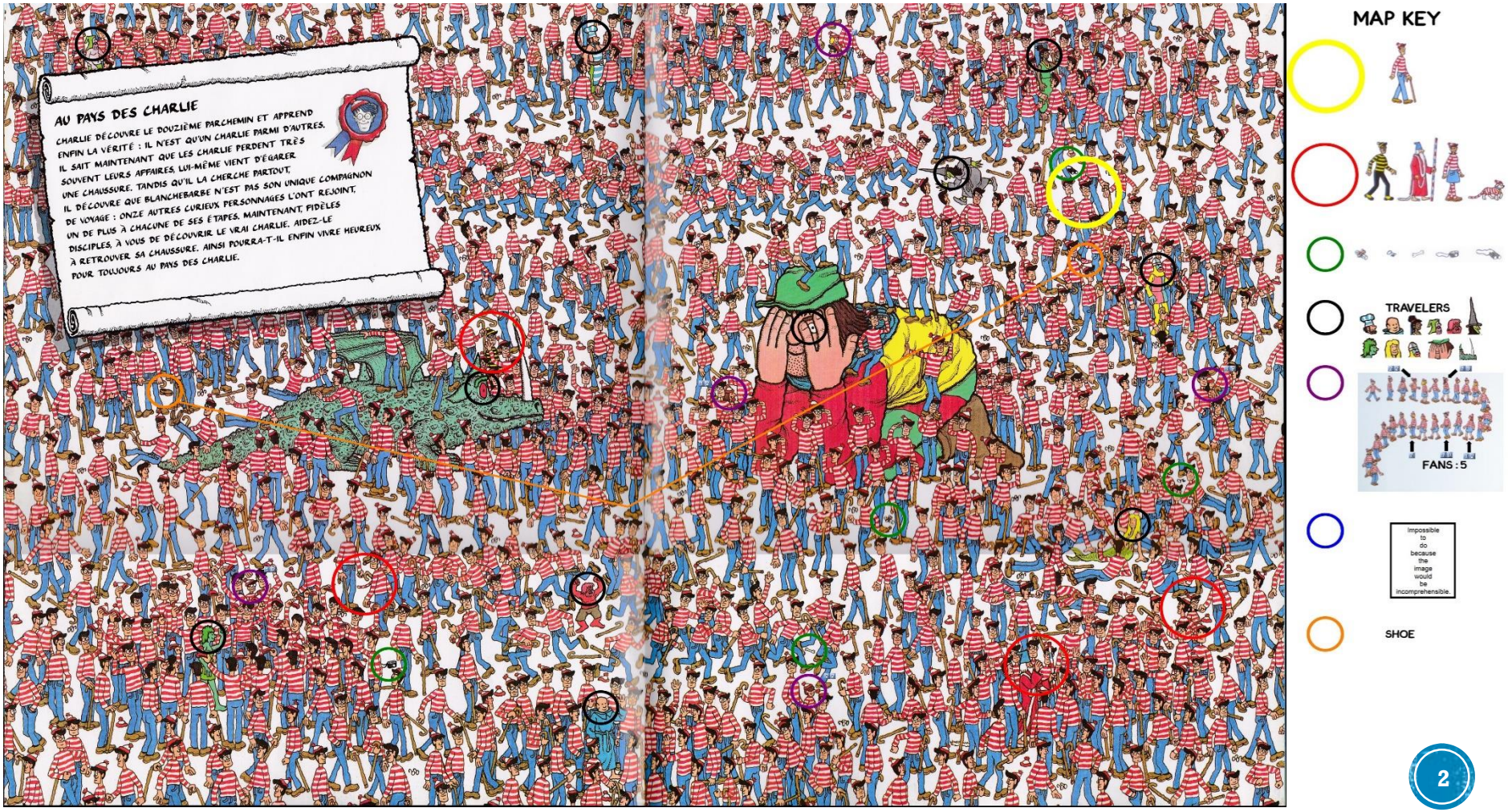
Computer Vision-Homework 2

Object Detection & Image Warping

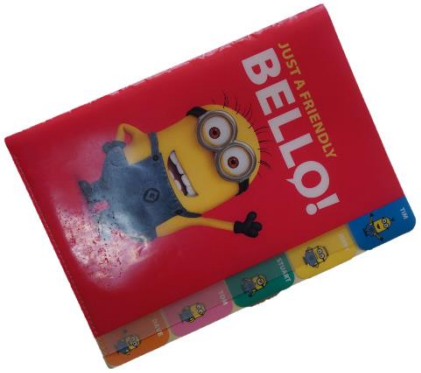
About “ Object Detection “

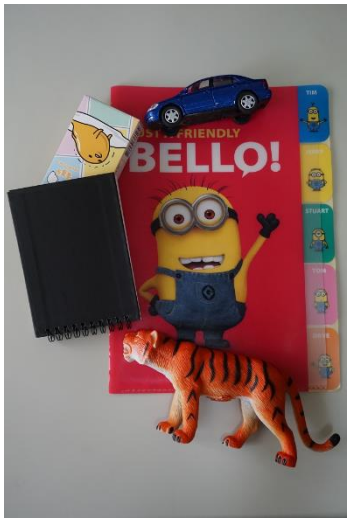
- For example, “Where's Wally?”



In our Homework 2

- Given the pictures of multiple objects taken from some different angles.
- Detect the object in the pictures.
  - Part 1: **Feature Detection** (with SIFT or etc.)
  - Part 2: **RANSAC**
- Warp the photo of the object to the pictures using the homography matrix you estimate.





Feature Detection

- find features correspondences/compute homography matrix.
- SIFT – Scale Invariant Feature Detection
  - detect key points in the image and describe the points as 128-dimensional features.
- Check Ch.6、7 for more details of SIFT.

SIFT in OpenCV

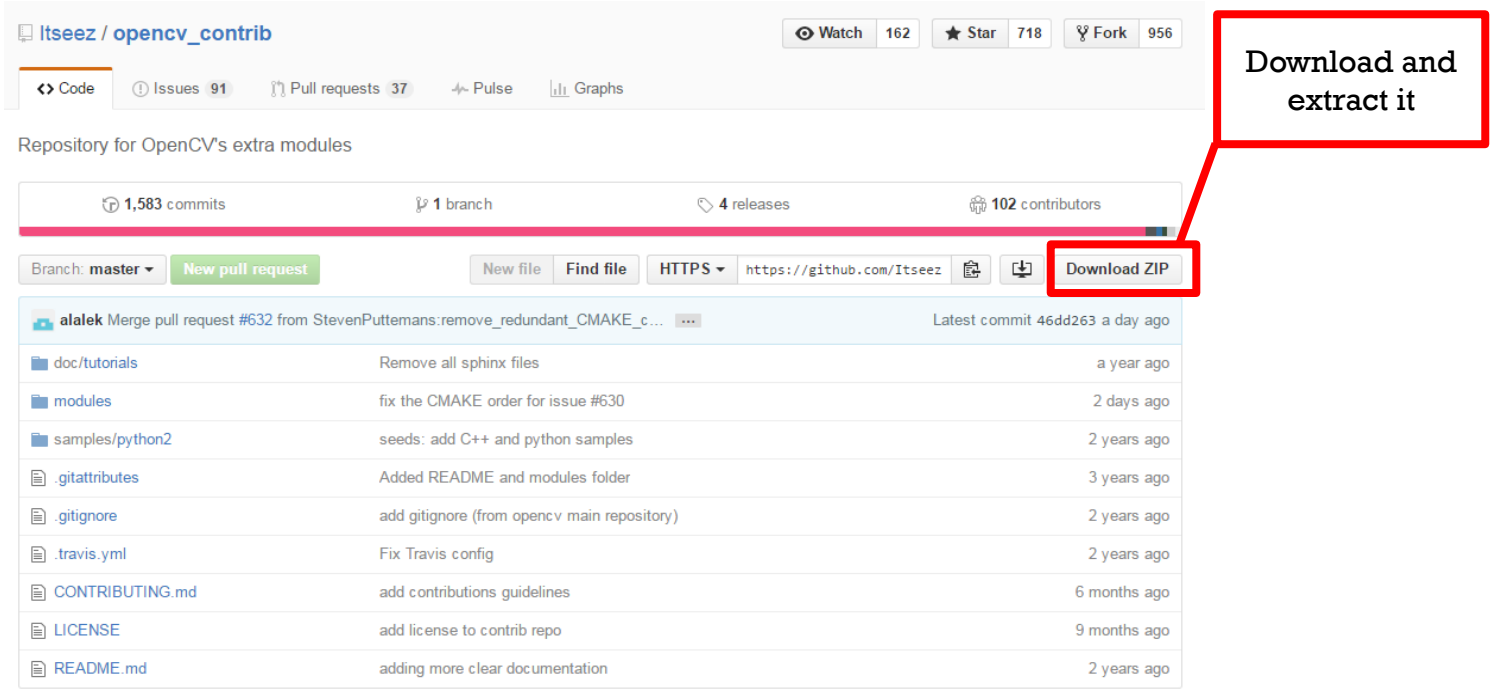
- We provide SIFT working on OpenCV 2.4.X and OpenCV 3.X

- For OpenCV2.4.X, you may need to add： Linker – Additional Dependency

  opencv_core24Xd.lib
  opencv_calib3d24Xd.lib
  opencv_contrib24Xd.lib
  opencv_features2d24Xd.lib
  opencv_highgui24Xd.lib
  opencv_imgproc24Xd.lib
  opencv_nonfree24Xd.lib

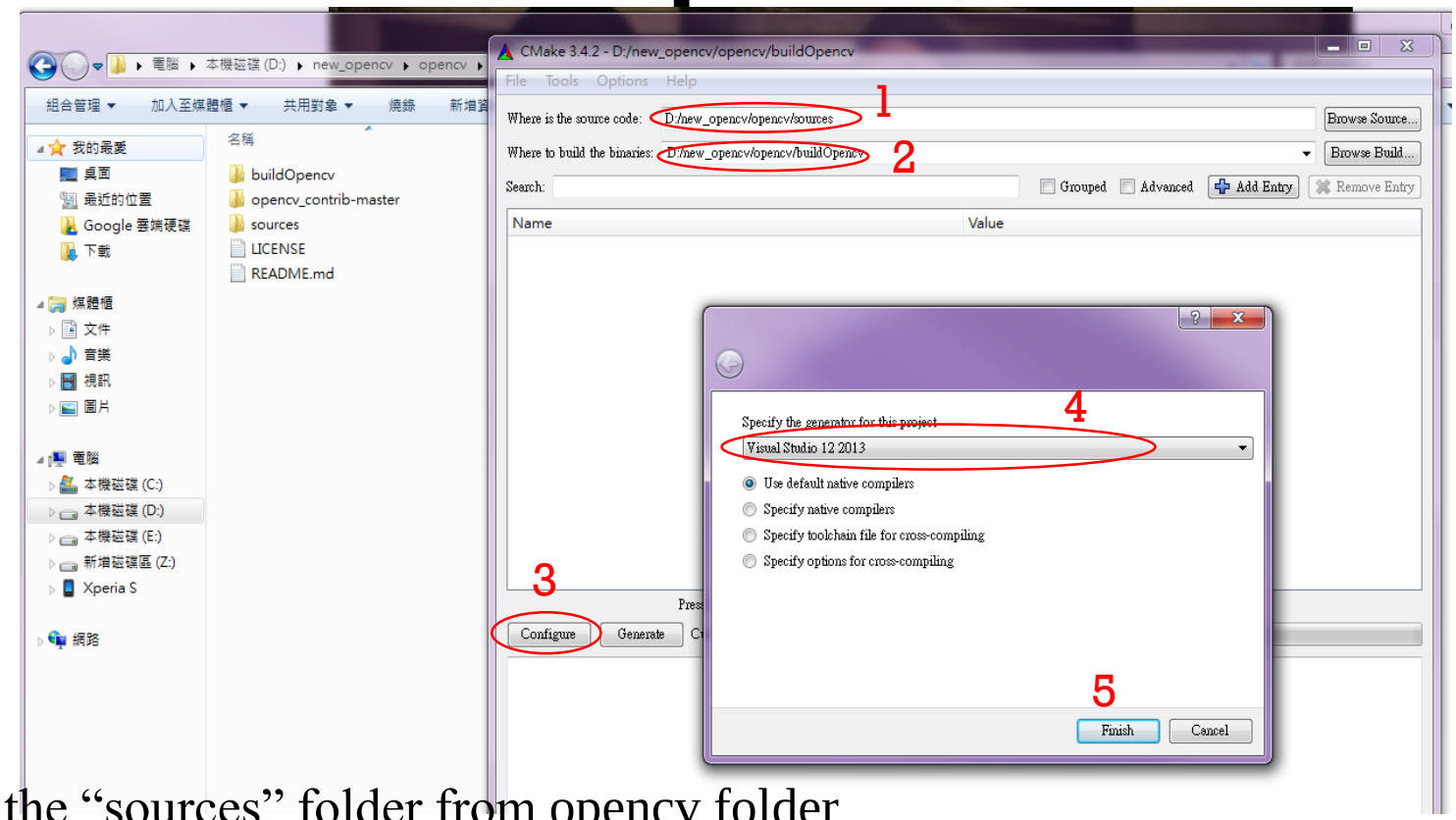
SIFT in OpenCV

- For OpenCV 3.X
  - you need to compile **opencv_contrib-master**
  - Download opencv_contrib-master to your opencv folder
- opencv_contrib-master download linker : https://github.com/Itseez/opencv_contrib



SIFT in OpenCV

- Open Cmake



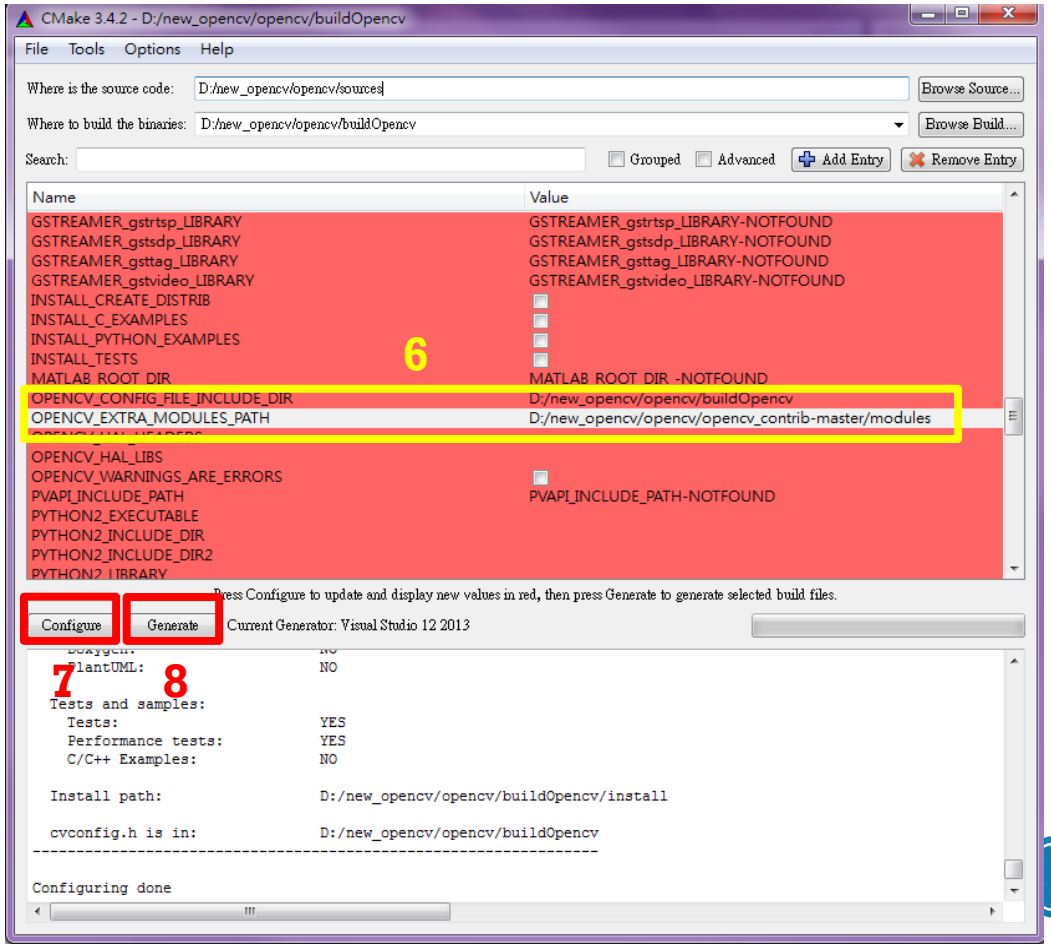
1. Choose the “sources” folder from opencv folder (D:/new_opencv/opencv/sources)
2. Create a new folder (buildOpencv) in the opencv folder
3. Click “Configure”
4. Choose your version of visual studio
5. Click “Finish”

SIFT in OpenCV

6. "**OPENCV_EXTRA_MODULE_PATH**"
   add the path of the "modules" folder in opencv_contrib-master (D:/new_opencv/opencv/ opencv_contrib-master/modules)

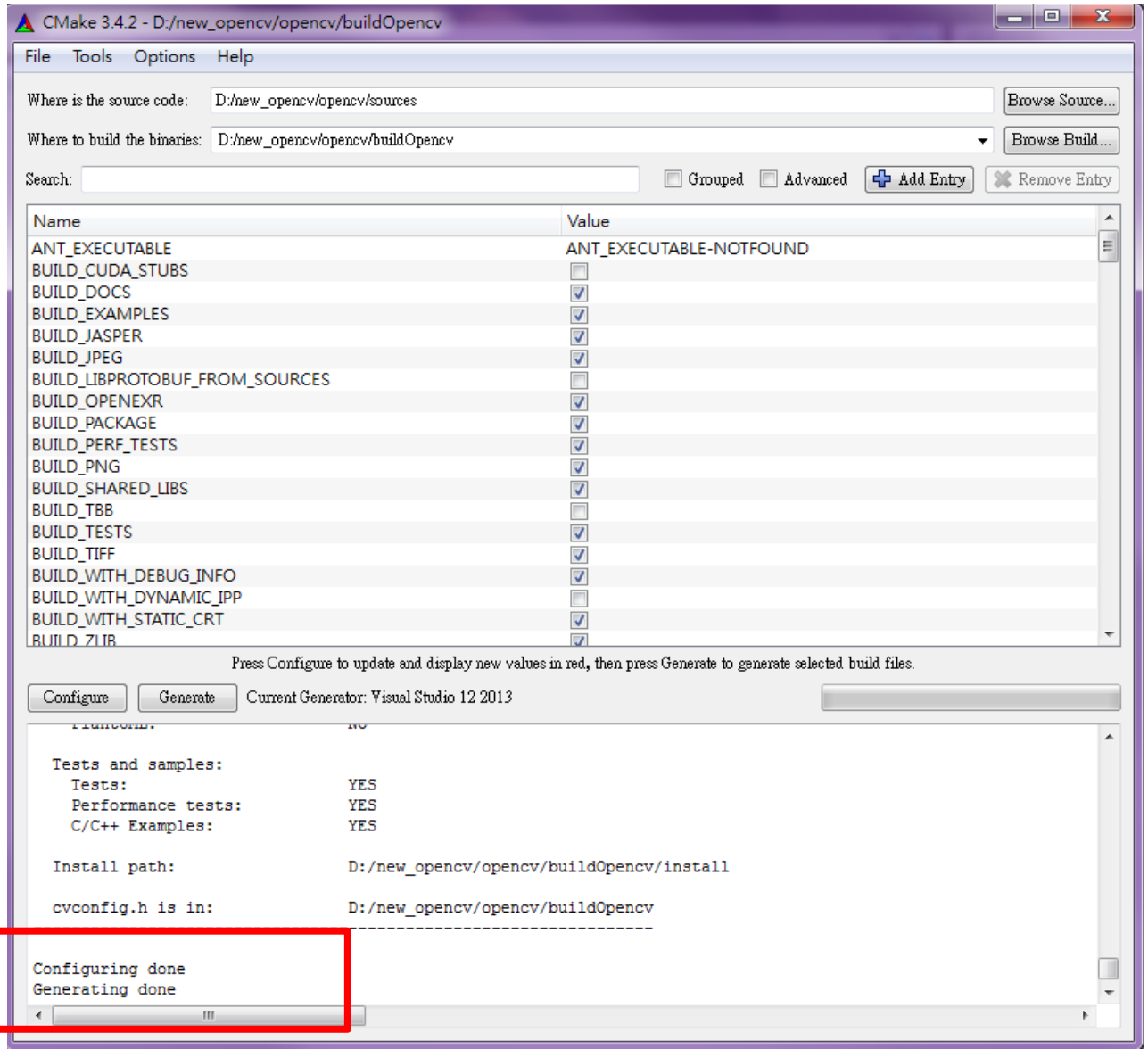
7. Click "**Configure**" , repeat click it until the window shows "Configure finish" and the window is not red

8. Click "**Generate**"



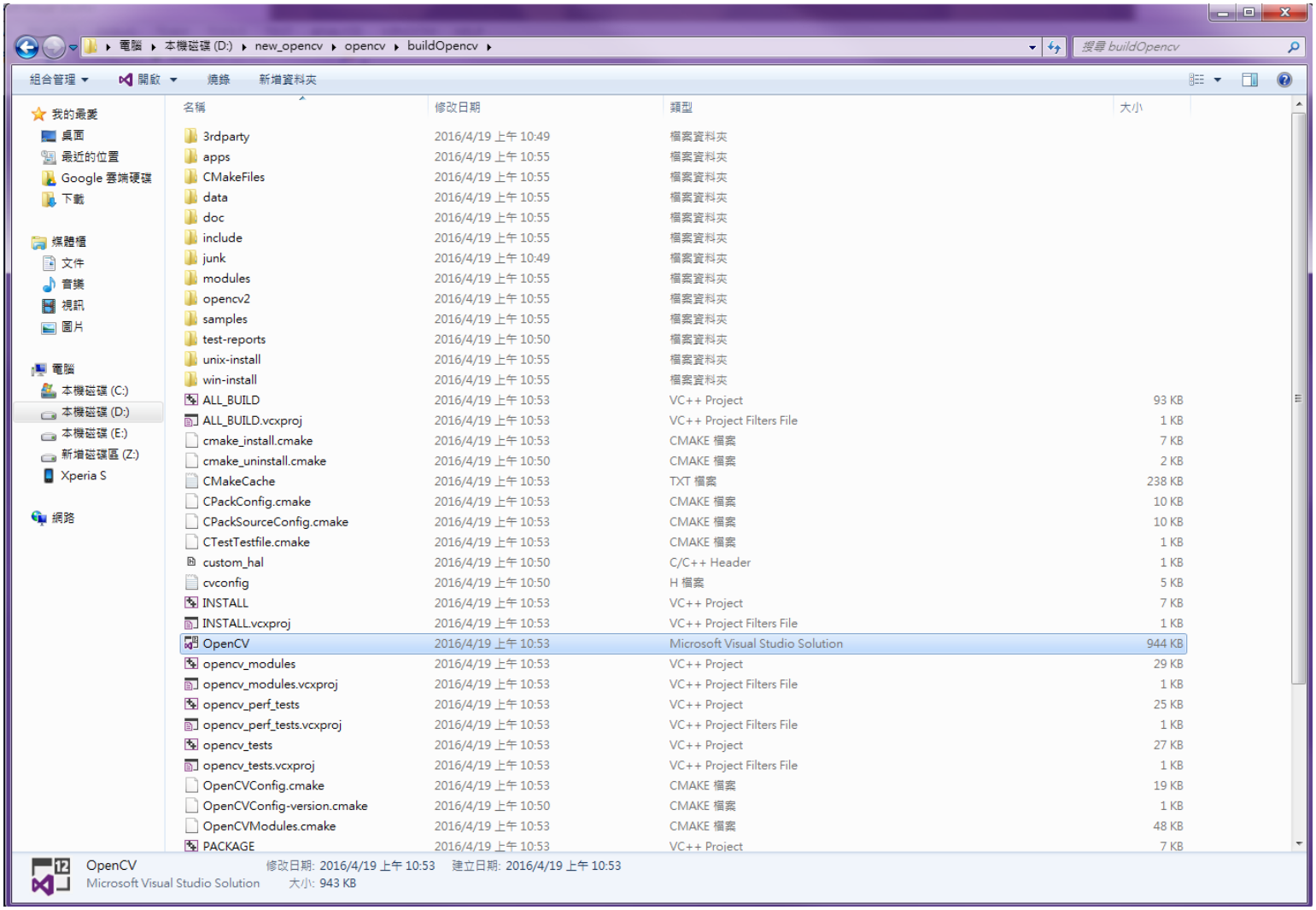
SIFT in OpenCV

- IF you success above, you will see like this :



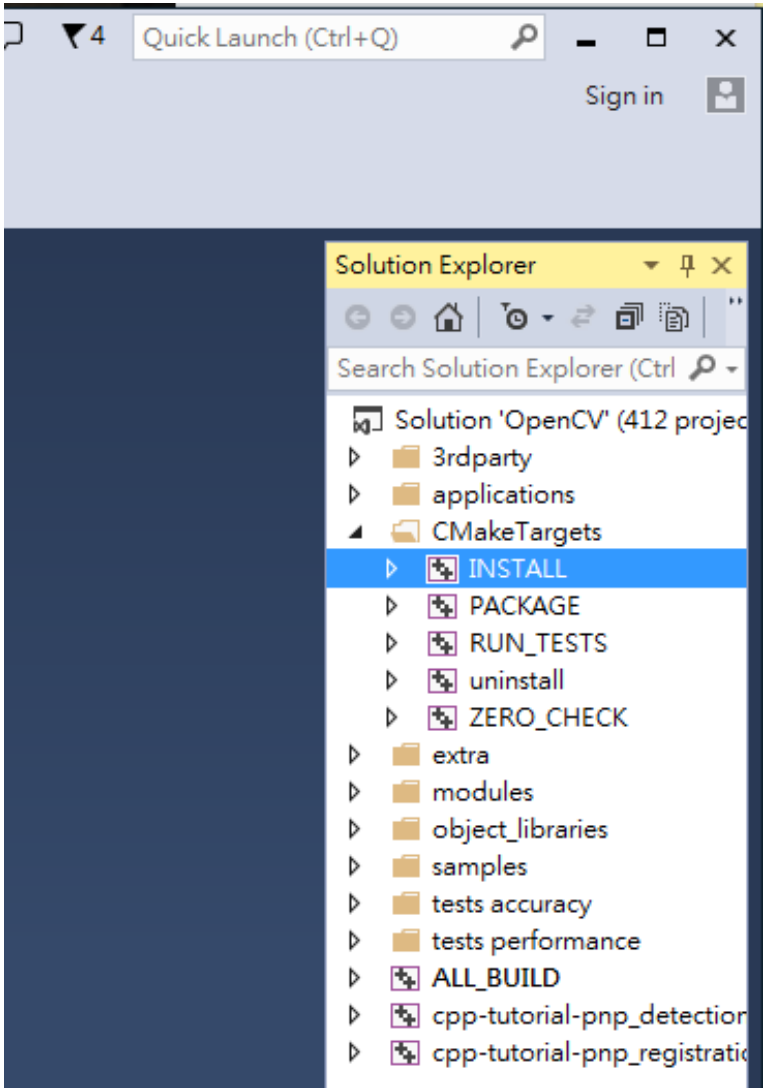
SIFT in OpenCV

9. Open “OpenCV.sln” in the folder you create in step2 (buildOpencv)



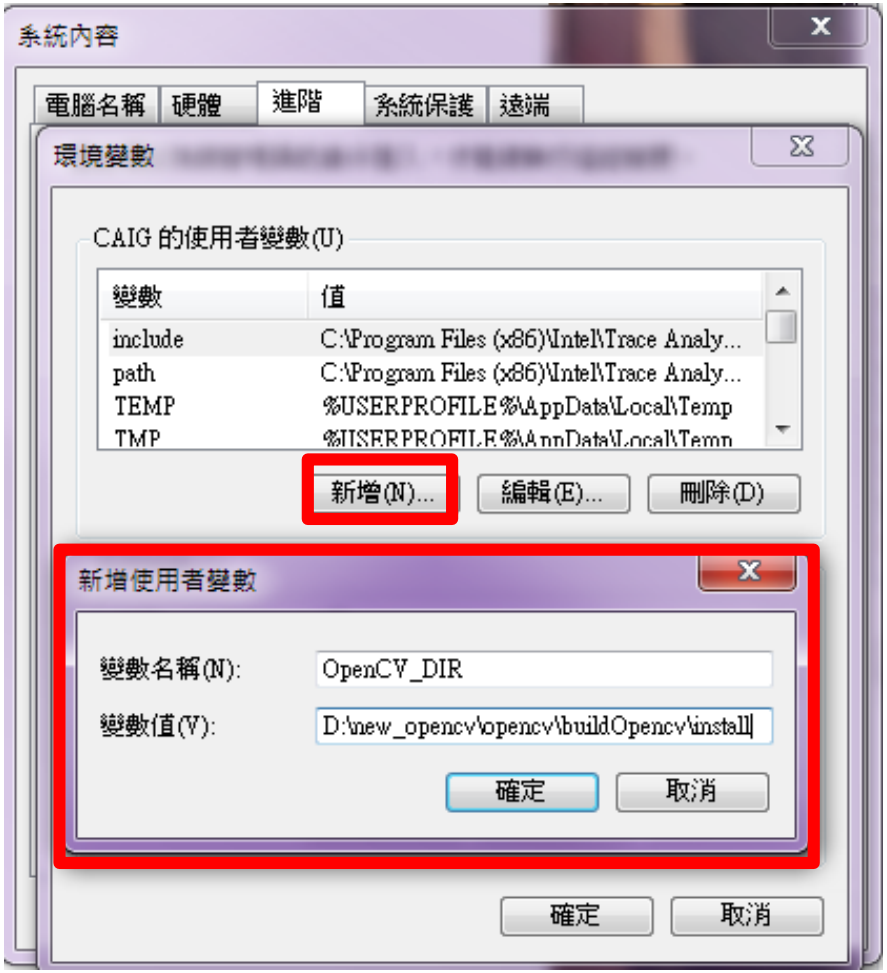
SIFT in OpenCV

10. Build “INSTALL”



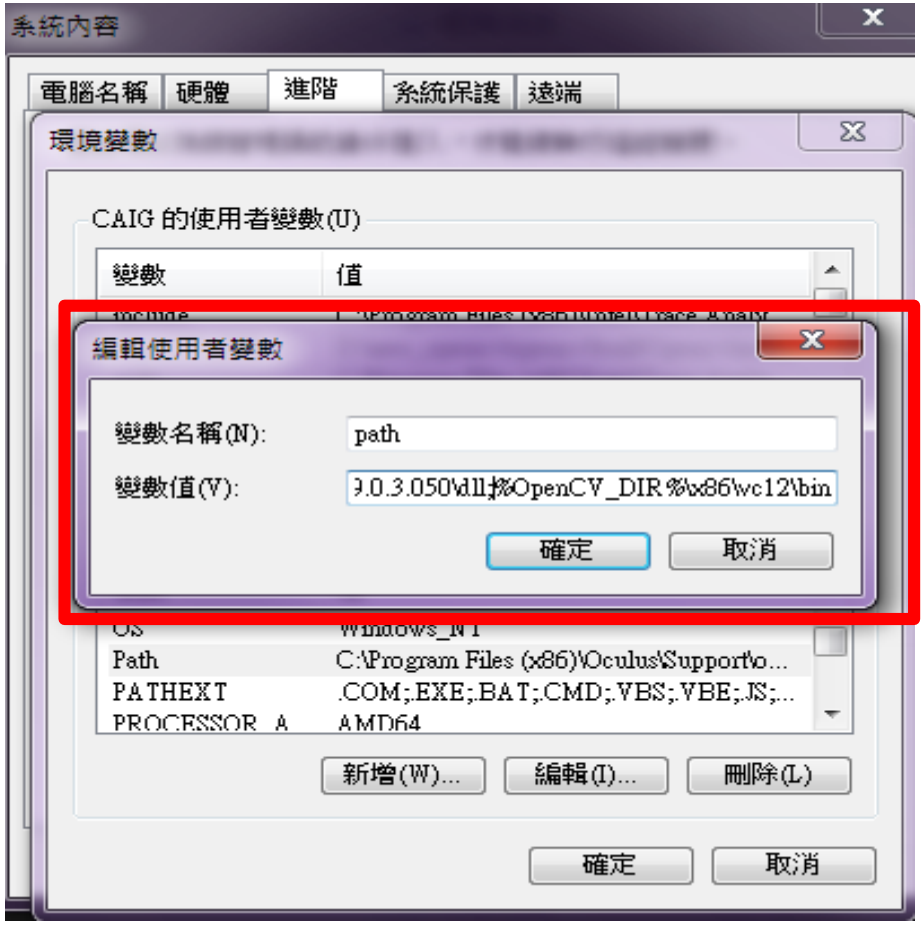
SIFT in OpenCV

11. Set the “Environment Variable(環境變數)”
    Add a new variable and set the path to the “install” in the folder you create in step2 (buildOpencv)
    (Add -> OpenCV_DIR -> D:\new_opencv\opencv\buildOpencv\install)



SIFT in OpenCV

12. Edit “Path”
    Add ：%OpenCV_DIR%\x86\vc12\bin
    Change above：
    OpenCV_DIR -> your variable name in step 12
    x86\vc12 -> your version of visual studio



SIFT in OpenCV

13. Follow the steps in HW1 to create a new project
14. Try the SIFT code

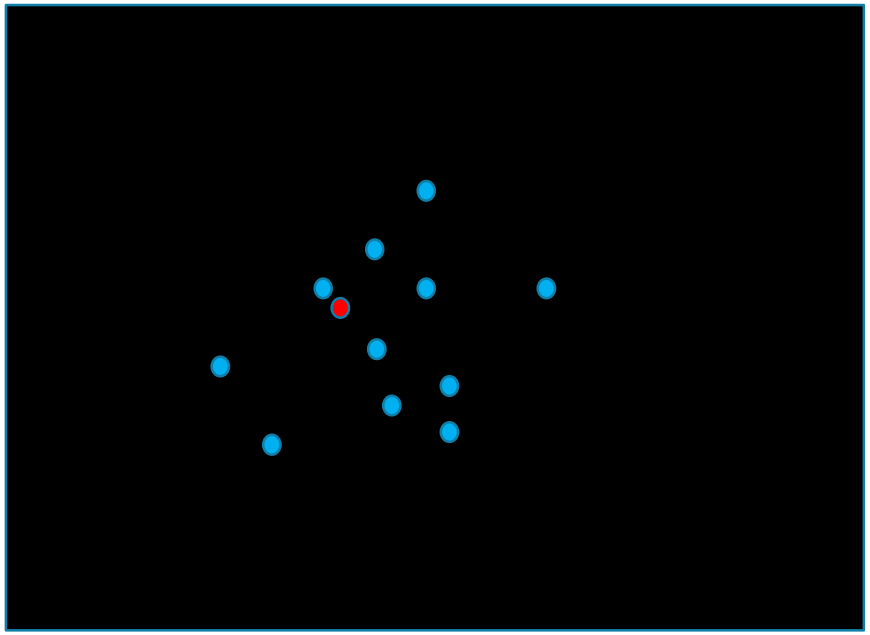
- Result
  - Feature image output in Feat1.bmp and Feat2.bmp
  - Feature points stored in keypoints1 and keypoints2
  - Descriptors stored in descriptor1 and descriptor2

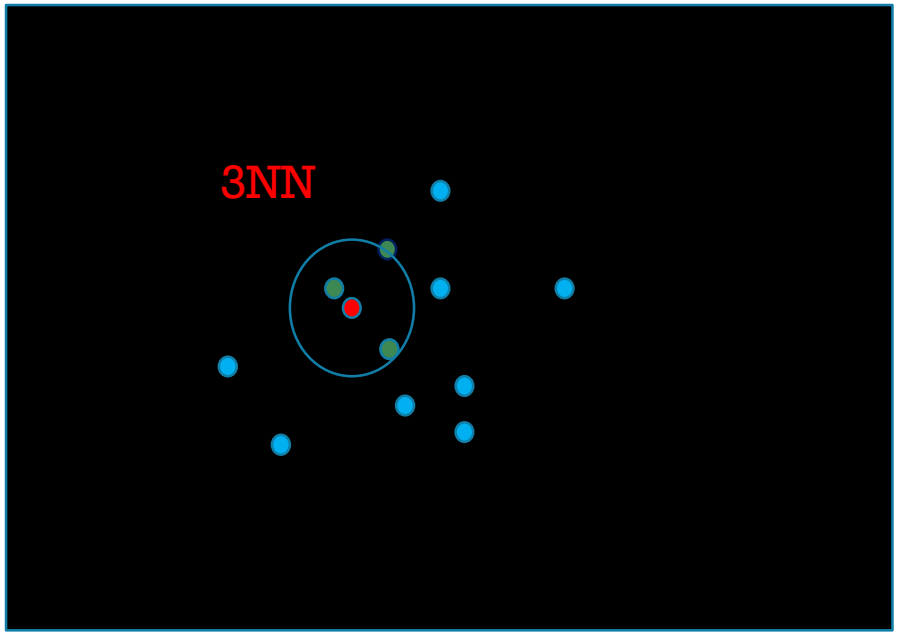
openCV 2.4.X

openCV 3.X

KNN

- K-Nearest Neighbor
  - Find the K closest neighbors to the target.





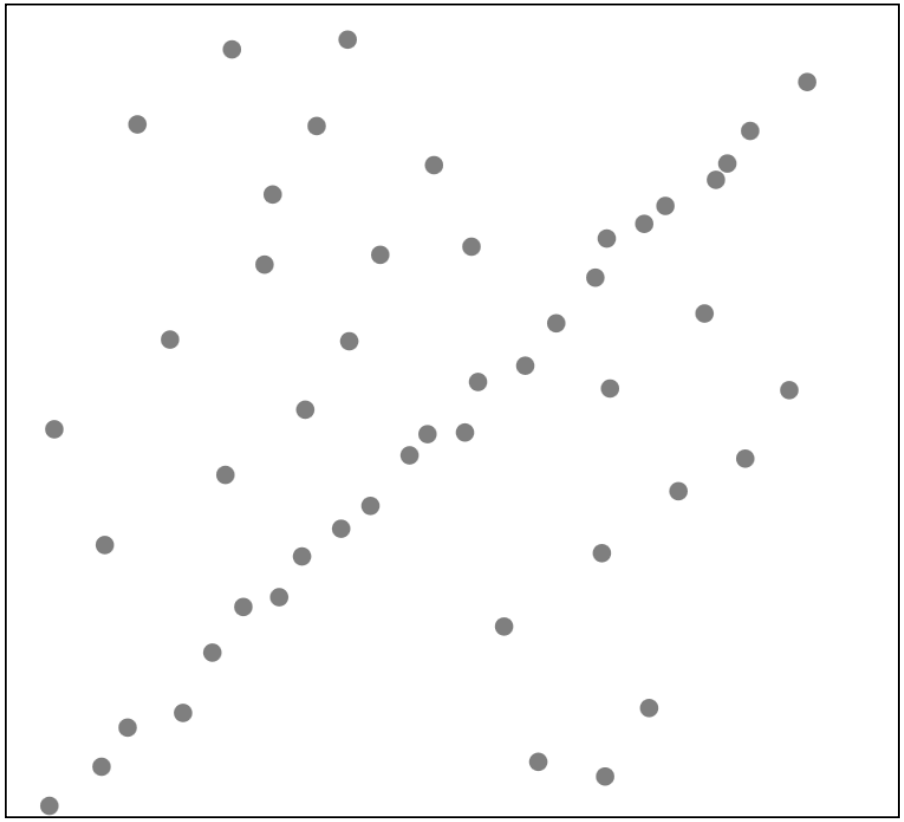
RANSAC

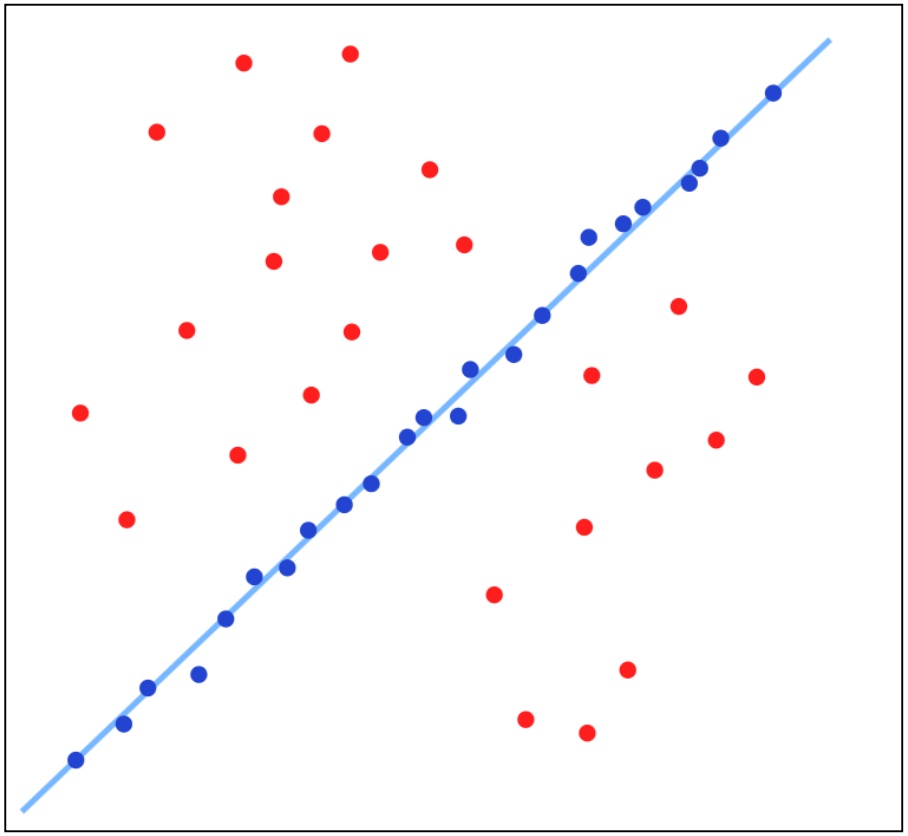
Random Sample Consensus

- Input: M data points;
  1. Randomly select N data points as inliers S. ($N \ll M$)
  2. Fit a model $\mathcal{M}$ to S.
  3. Test all data points against $\mathcal{M}$, add the points consistent with $\mathcal{M}$ to S, which is called a consensus set.
  4. If $|S|$ is larger than ever, mark $\mathcal{M}$ as the best estimated model $\mathcal{M}^*$.
  5. If some stopping criterion is satisfied, end
  6. Else go to step 1.

  Note that you can re-estimate the models with the consensus sets.

RANSAC





Recover Homographies

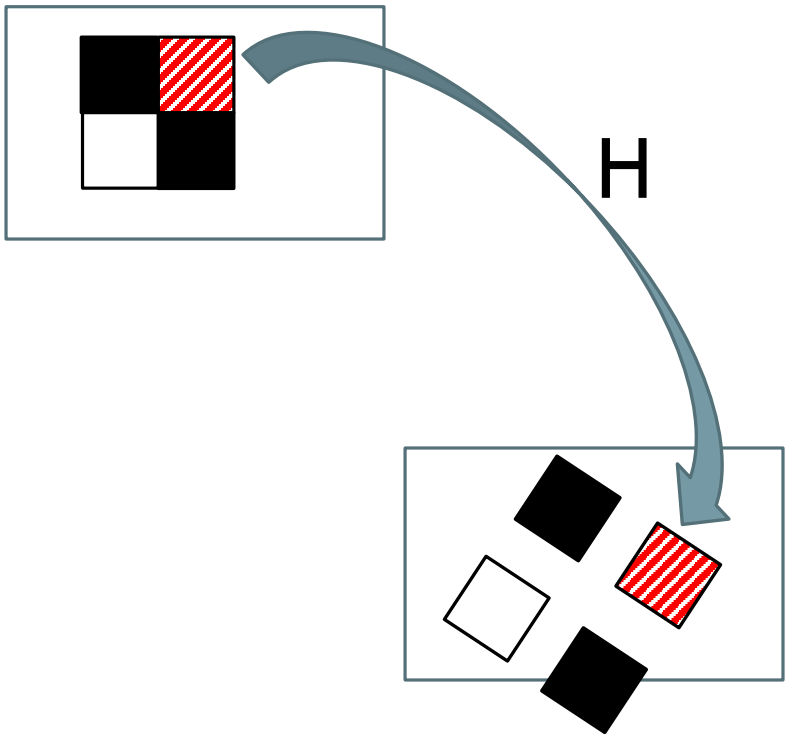
- Construct a linear system as: p'=Hp, where p' and p are correspondence points.
- Follow the Lecture 7 page 6~8. You may try Affine mappings(DOF=6) or Projective mappings(DOF=8).
- Solve Ax=0
- cv::eigen (用法可查詢 http://docs.opencv.org/2.4/modules/core/doc/operations_on_arrays.html#eigen)

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} * \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

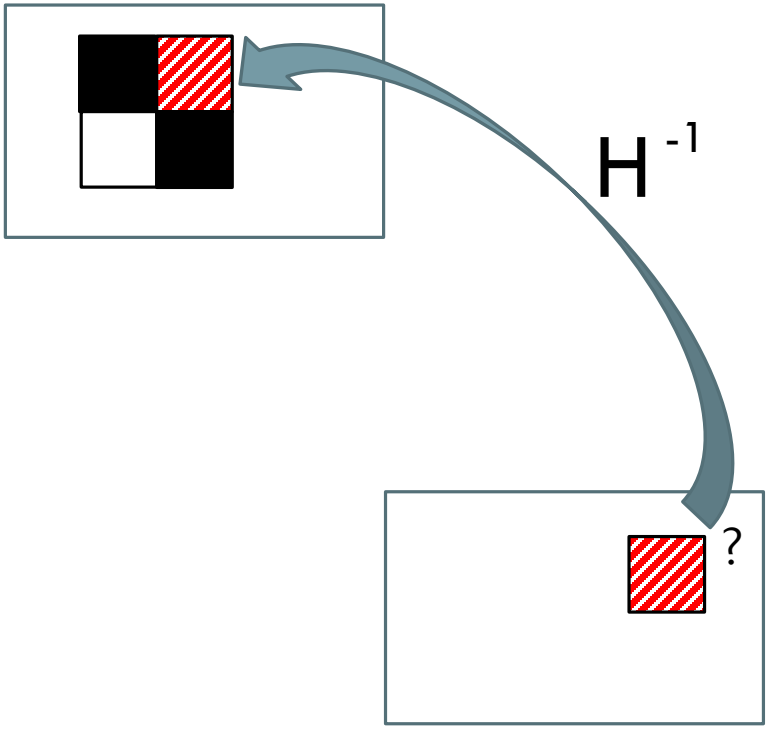
$$\begin{bmatrix} X_1 & Y_1 & 1 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1 \\ 0 & 0 & 0 & X_1 & Y_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1 \\ X_2 & Y_2 & 1 & 0 & 0 & 0 & -x_2X_2 & -x_2Y_2 & -x_2 \\ 0 & 0 & 0 & X_2 & Y_2 & 1 & -y_2X_2 & -y_2Y_2 & -y_2 \\ X_3 & Y_3 & 1 & 0 & 0 & 0 & -x_3X_3 & -x_3Y_3 & -x_3 \\ 0 & 0 & 0 & X_3 & Y_3 & 1 & -y_3X_3 & -y_3Y_3 & -y_3 \\ X_4 & Y_4 & 1 & 0 & 0 & 0 & -x_4X_4 & -x_4Y_4 & -x_4 \\ 0 & 0 & 0 & X_4 & Y_4 & 1 & -y_4X_4 & -y_4Y_4 & -y_4 \end{bmatrix} * \begin{bmatrix} A_{11} \\ A_{12} \\ A_{13} \\ A_{21} \\ A_{22} \\ A_{23} \\ A_{31} \\ A_{32} \\ A_{33} \end{bmatrix} = 0$$

Warp the Images

- Forward



- Backward



Warp the Images

- Warp the object image without the image background.
- In this assignment, you may cut the background by：

```
if ( pixel-color != white ){
    warping;
}
else{
    skip;
}
```

Algorithm(for reference)

- 0. SIFT feature detection
- 1. For each feature point in the object image:

  Find KNN points in the target image
  (according to the descriptors)
- 2. Randomly select 4 (or more) points in the object image:

  For all k*k*k*k possible match sets:

  Construct the Homography matrix

  Compute inliers and outliers

  If > Threshold, output

  else repeat 2
- 3. Use the recovered Homogrphy matrix to warp the object image to the target image.

Requirements

- Input：Enter file name of object image and target image

- Output：
  - find the object in the target image
  - Warp the object image with right orientation to the target image

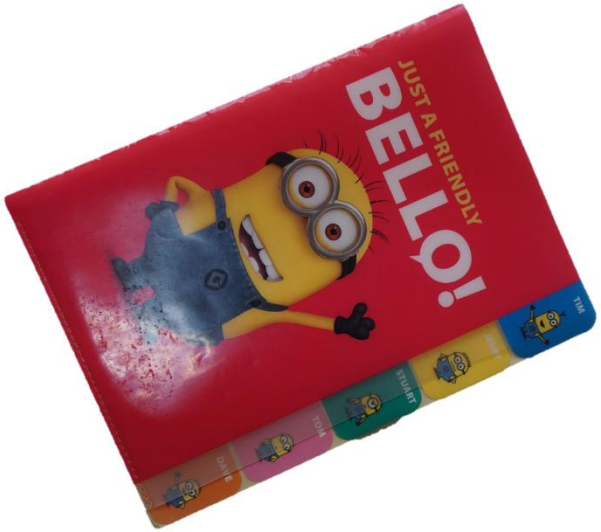
Requirements

- ☆★此次作業只可以使用image read/save, pixel set/get, matrix basic operation(+-*/,反矩陣等), 內建eigenvalue及eigenvector函式，請勿使用其他功能太強大的函式
- Homography matrix 要自己找，KNN、RANSAC、Warping 都要自己寫，請勿呼叫內建函式直接完成
- 可使用C/C++以外的語言，但限制同上

Example

Target Image

Object Image



Result Image

Grading

- Find object in target image and warp it. 100 %
  - 1 object successful 75 %
  - 2 objects successful 85 %
  - Another testing dataset when demo 95 %
  - Use backward warping +5 %

- Other interesting things (ex. Speed up……)

+10% at most

Deadline

- Deadline： 2016/05/17(二) 11:59:59 pm
- 請將作業包含code/report/result image 壓縮並以學號命名: ex 0987654.zip
- 上傳至E3！
- Report包含程式流程說明 & 執行方式 & 自己多做了哪些功能&其他你想寫的…