Homework 2

Computer Vision 2017 Spring

2017.4.27

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.

OpenCV

- Highly optimized C++ implementation of many CV algorithms
- Cross platform
- Many bindings (Python, Matlab, Java . . .)

http://opencv.org

Install

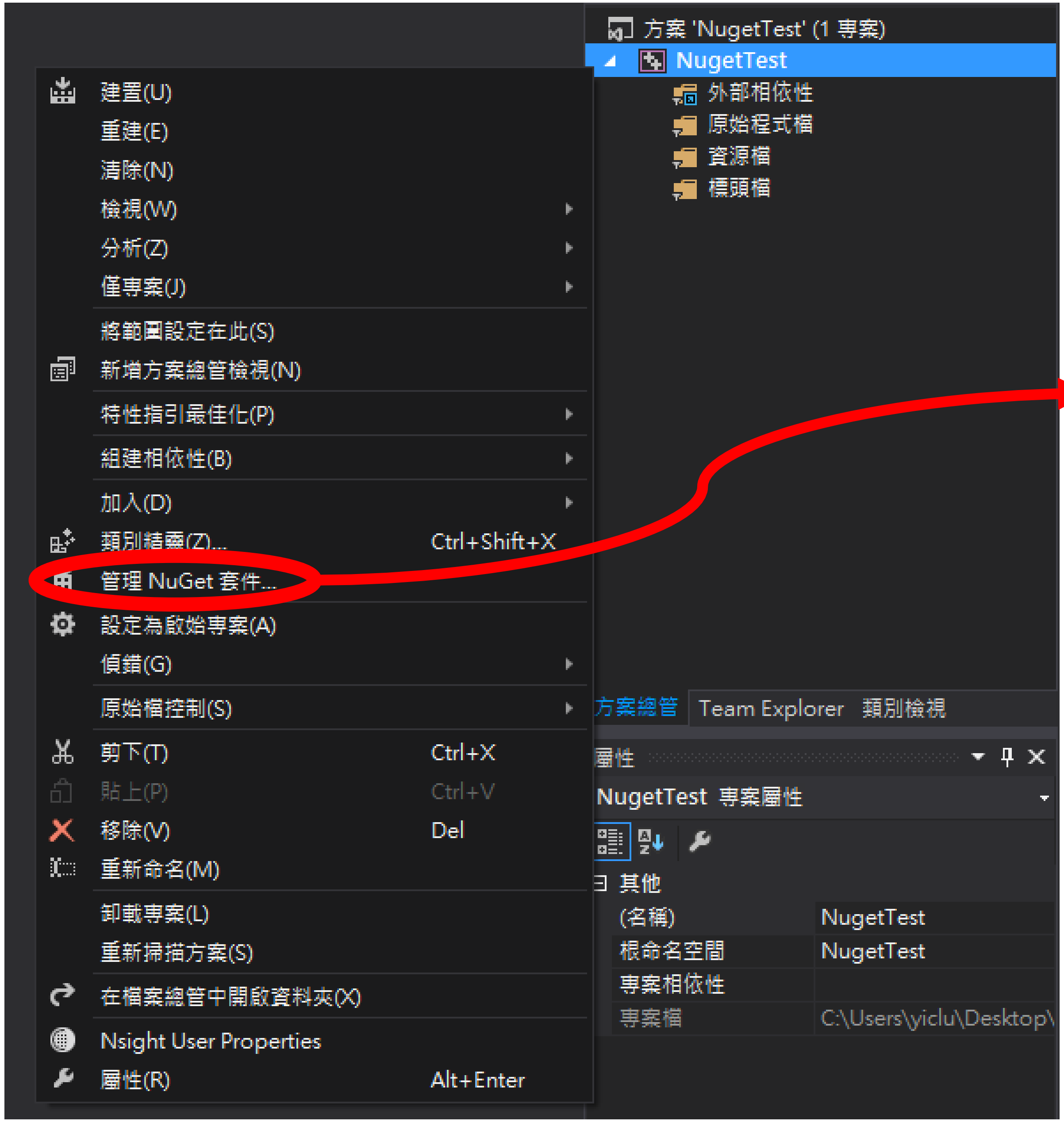
- VS 2013、VS 2015

1 , Open your visual Studio

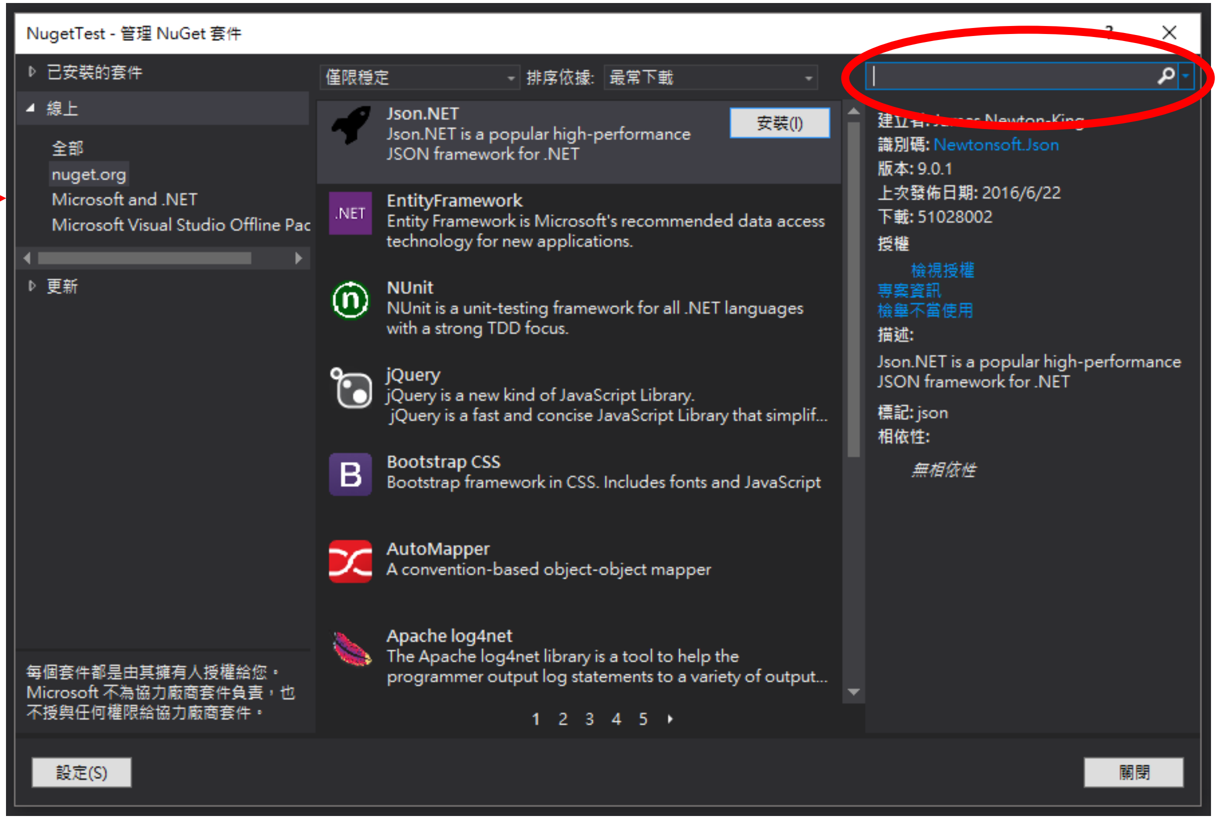
2 , New a Empty Project

3 , Install OpenCV

VS 2013

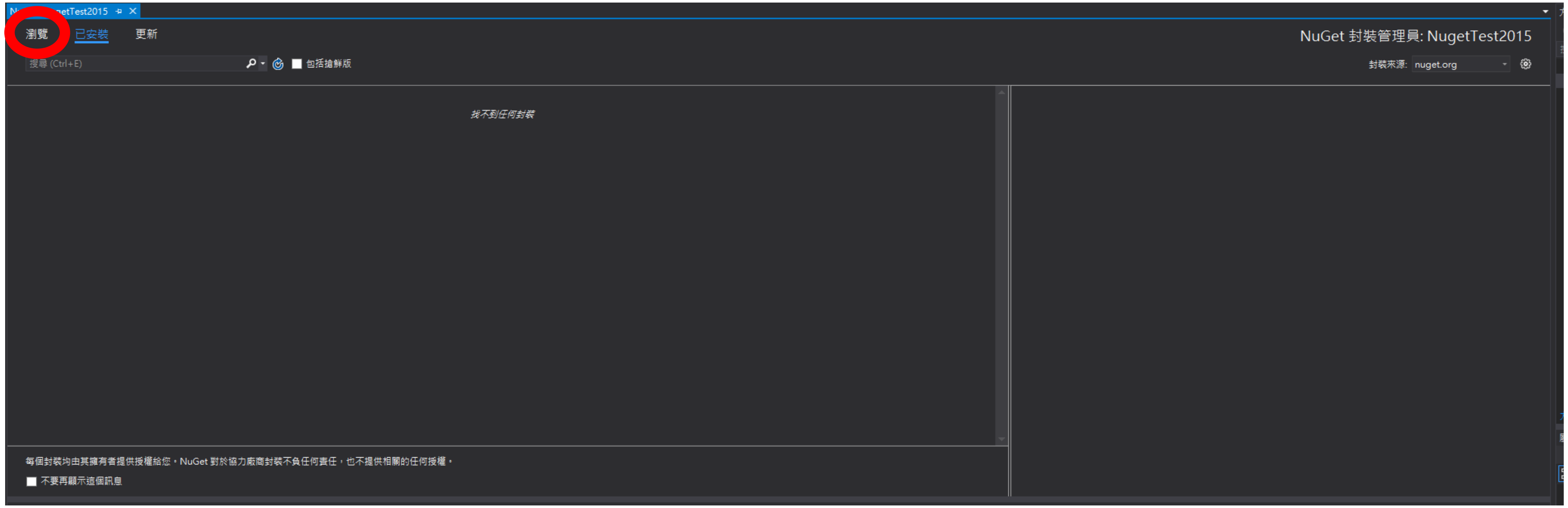


方案 'NugetTest' (1 專案)
NugetTest
外部相依性
原始程式檔
資源檔
標頭檔
建置(U)
重建(E)
清除(N)
檢視(W)
分析(Z)
僅專案(J)
將範圍設定在此(S)
新增方案總管檢視(N)
特性指引最佳化(P)
組建相依性(B)
加入(D)
類別精靈(Z)... Ctrl+Shift+X
管理 NuGet 套件...
設定為啟始專案(A)
偵錯(G)
原始檔控制(S)
剪下(T) Ctrl+X
貼上(P) Ctrl+V
移除(V) Del
重新命名(M)
卸載專案(L)
重新掃描方案(S)
在檔案總管中開啟資料夾(X)
Nsight User Properties
屬性(R) Alt+Enter
方案總管
Team Explorer
類別檢視
屬性
NugetTest 專案屬性
其他
(名稱) NugetTest
根命名空間 NugetTest
專案相依性
專案檔 C:\Users\yiclu\Desktop\

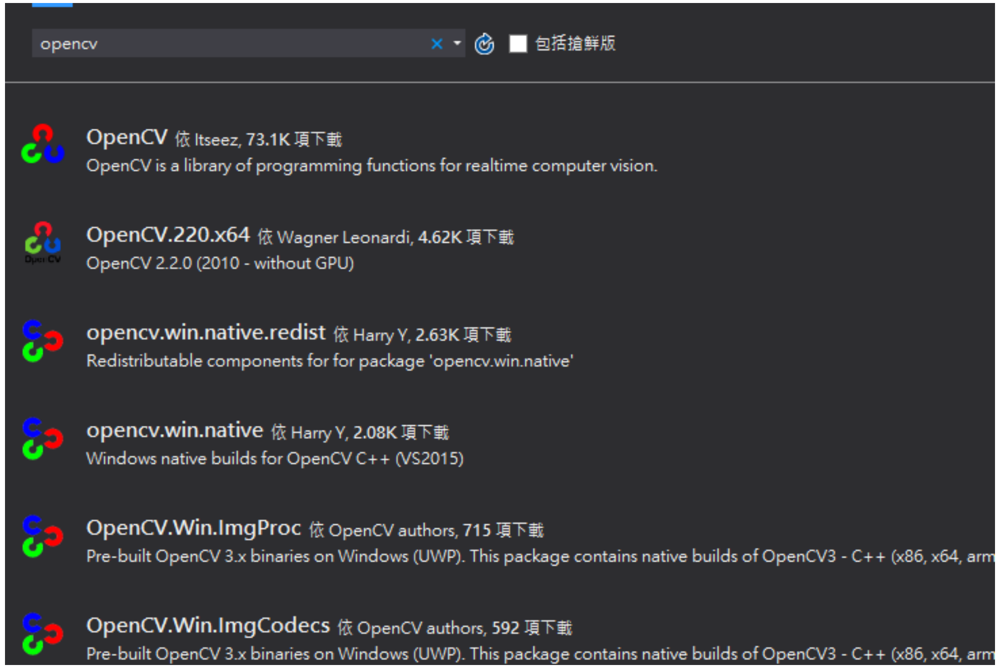


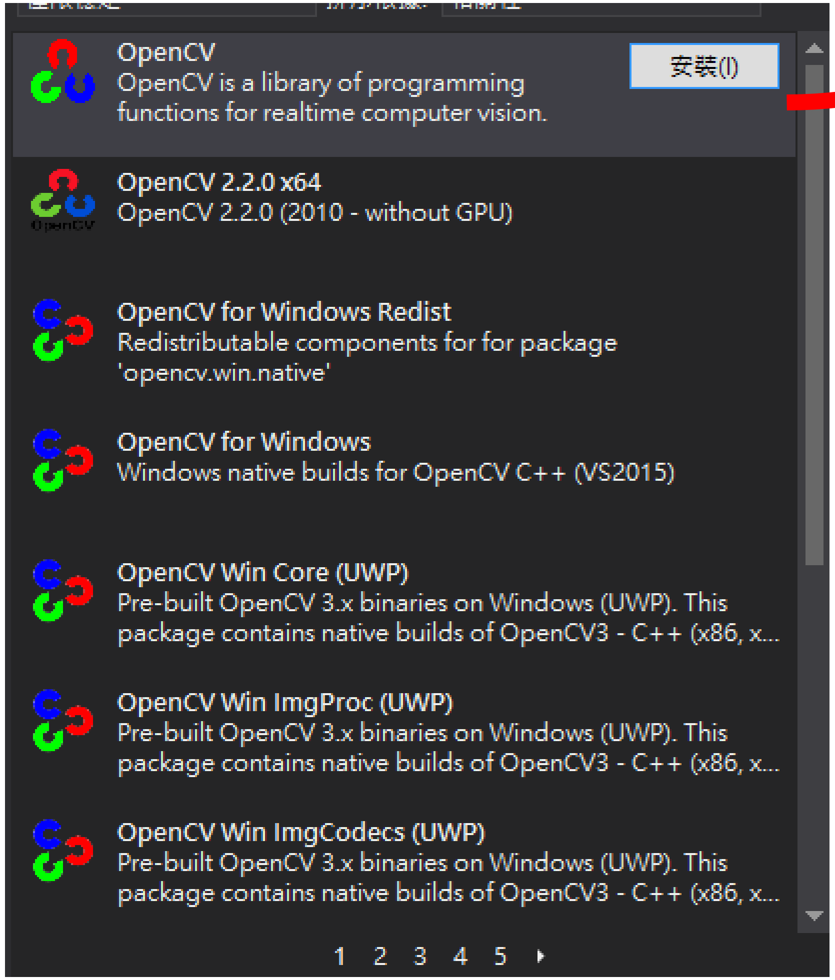
NugetTest - 管理 NuGet 套件
已安裝的套件
線上
全部
nuget.org
Microsoft and .NET
Microsoft Visual Studio Offline Pac
更新
僅限穩定
排序依據: 最常下載
Json.NET
Json.NET is a popular high-performance JSON framework for .NET
安裝(I)
EntityFramework
Entity Framework is Microsoft's recommended data access technology for new applications.
NUnit
NUnit is a unit-testing framework for all .NET languages with a strong TDD focus.
jQuery
jQuery is a new kind of JavaScript Library.
jQuery is a fast and concise JavaScript Library that simplif...
Bootstrap CSS
Bootstrap framework in CSS. Includes fonts and JavaScript
AutoMapper
A convention-based object-object mapper
Apache log4net
The Apache log4net library is a tool to help the programmer output log statements to a variety of output...
1 2 3 4 5
識別碼: Newtonsoft.Json
版本: 9.0.1
上次發佈日期: 2016/6/22
下載: 51028002
授權
檢視授權
專案資訊
檢舉不當使用
描述:
Json.NET is a popular high-performance JSON framework for .NET
標記: json
相依性:
無相依性
每個套件都是由其擁有人授權給您。Microsoft 不為協力廠商套件負責，也不授與任何權限給協力廠商套件。
設定(S)
關閉

VS 2015

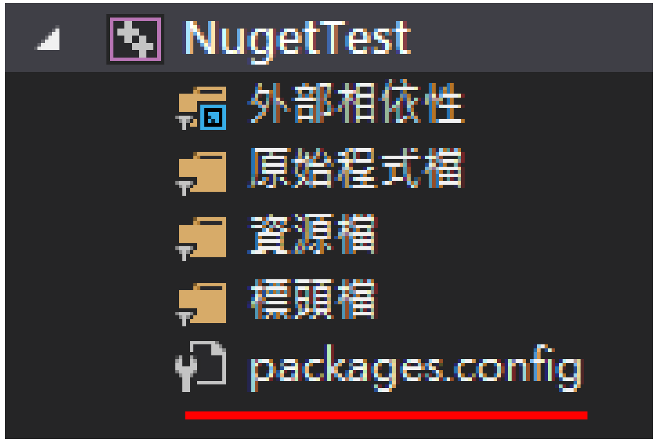

瀏覽
已安裝
更新
NuGet 封裝管理員: NugetTest2015
搜尋 (Ctrl+E)
包括搶鮮版
封裝來源: nuget.org
找不到任何封裝
每個封裝均由其擁有者提供授權給您。NuGet 對於協力廠商封裝不負任何責任，也不提供相關的任何授權。
不要再顯示這個訊息


瀏覽
已安裝
更新
搜尋 (Ctrl+E)
包括搶鮮版
Newtonsoft.Json 依 James Newton-King, 50.9M 項下載
Json.NET is a popular high-performance JSON framework for .NET
EntityFramework 依 Microsoft, 27.6M 項下載
Entity Framework is Microsoft's recommended data access technology for new applications.
NUnit 依 Charlie Poole, 8.62M 項下載
NUnit is a unit-testing framework for all .NET languages with a strong TDD focus.
jQuery 依 jQuery Foundation, Inc., 32.5M 項下載
jQuery is a new kind of JavaScript Library.
jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, a
bootstrap 依 Twitter, Inc., 9.22M 項下載
Bootstrap framework in CSS. Includes fonts and JavaScript
AutoMapper 依 Jimmy Bogard, 7.48M 項下載
A convention-based object-object mapper


opencv
包括搶鮮版
OpenCV 依 Itseez, 73.1K 項下載
OpenCV is a library of programming functions for realtime computer vision.
OpenCV.220.x64 依 Wagner Leonardi, 4.62K 項下載
OpenCV 2.2.0 (2010 - without GPU)
opencv.win.native.redist 依 Harry Y, 2.63K 項下載
Redistributable components for for package 'opencv.win.native'
opencv.win.native 依 Harry Y, 2.08K 項下載
Windows native builds for OpenCV C++ (VS2015)
OpenCV.Win.ImgProc 依 OpenCV authors, 715 項下載
Pre-built OpenCV 3.x binaries on Windows (UWP). This package contains native builds of OpenCV3 - C++ (x86, x64, arm
OpenCV.Win.ImgCodecs 依 OpenCV authors, 592 項下載
Pre-built OpenCV 3.x binaries on Windows (UWP). This package contains native builds of OpenCV3 - C++ (x86, x64, arm



VS 2013
VS 2015



SIFT in OpenCV

```
//#include <opencv/highgui.h> // For VS2013
#include <opencv2/highgui/highgui.hpp> // For VS2015
#include <opencv2/nonfree/features2d.hpp>
```

SIFT in OpenCV

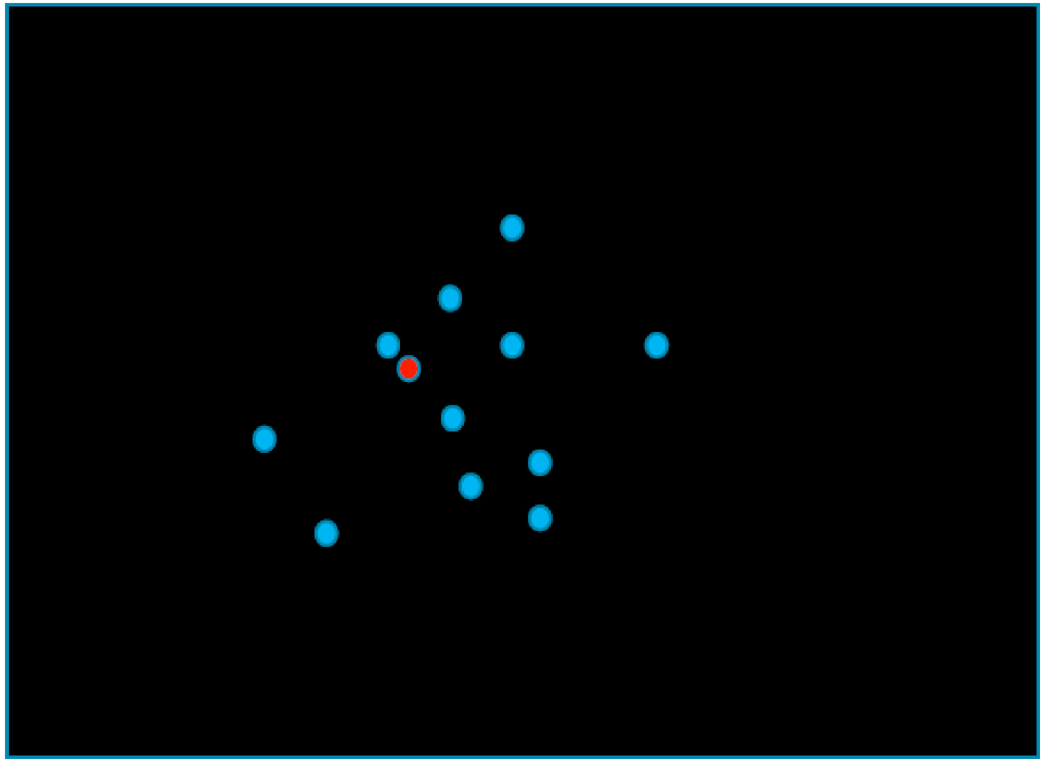
```
// SIFT feature detector and feature extractor
SiftFeatureDetector detector;
SiftDescriptorExtractor extractor;

// Feature detection
vector<KeyPoint> keypoints;
detector.detect(Image, keypoints);
cout << "Keypoints' number = " << keypoints.size() << endl;

// Feature descriptor computation
Mat descriptor;
extractor.compute(Image, keypoints, descriptor);
cout << "Descriptor's size = " << descriptor.size() << endl;

// Feature display on image
Mat feature;
drawKeypoints(Image, keypoints, feature, Scalar(255, 255, 255), DrawMatchesFlags::DRAW_RICH_KEYPOINTS);
imwrite("result.bmp", feature);
```

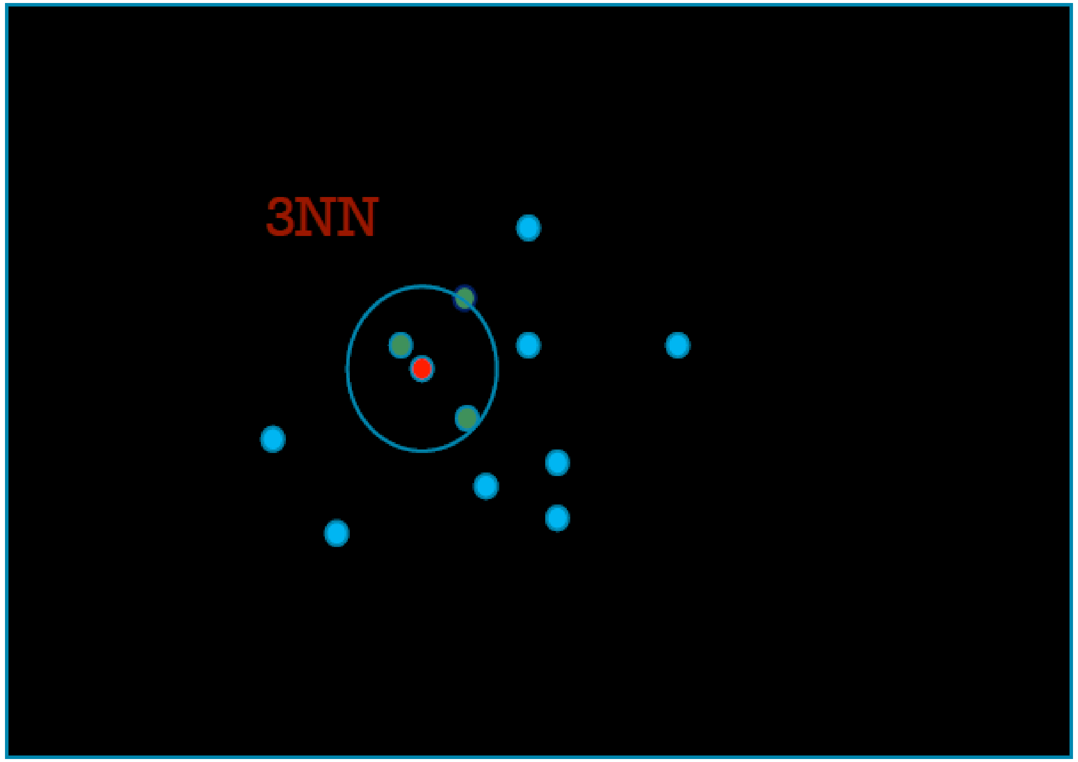
SIFT

- Result
  - Feature image output in result.bmp
  - Feature points stored in keypoints
  - Descriptors stored in descriptor

KNN

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





RANSAC

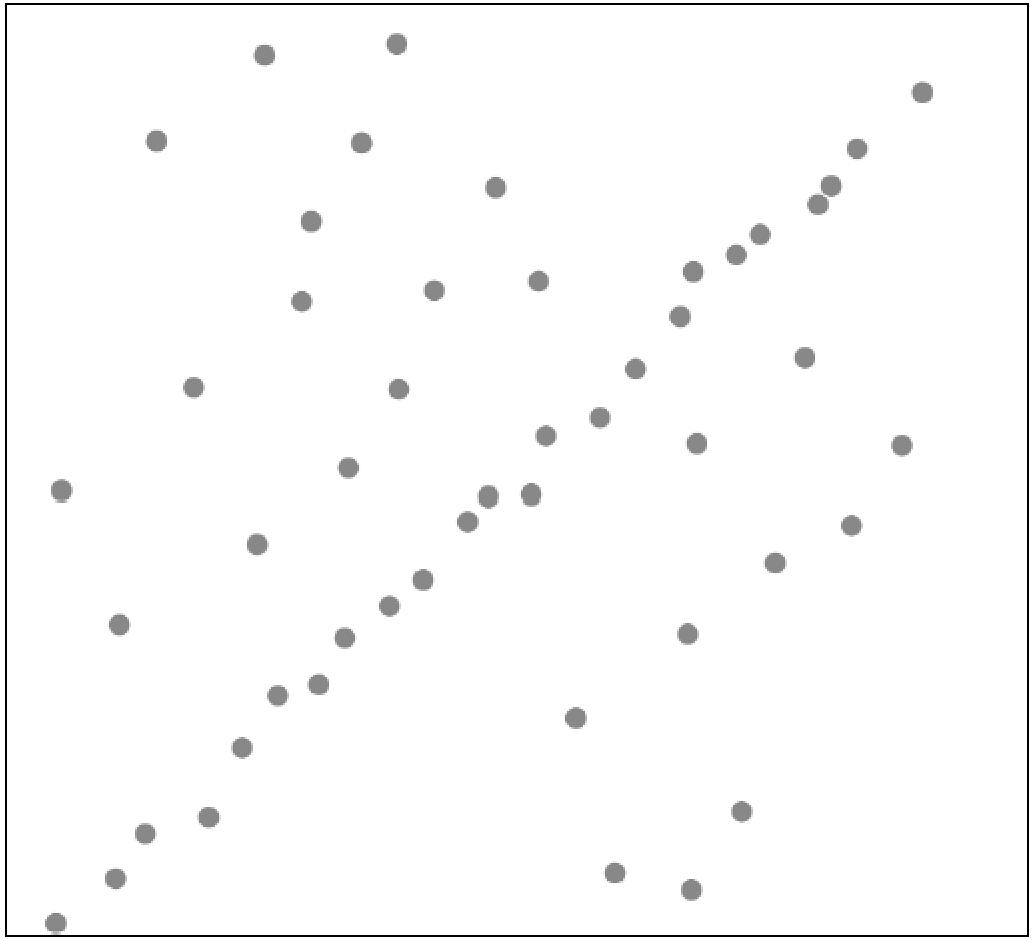
Random Sample Consensus

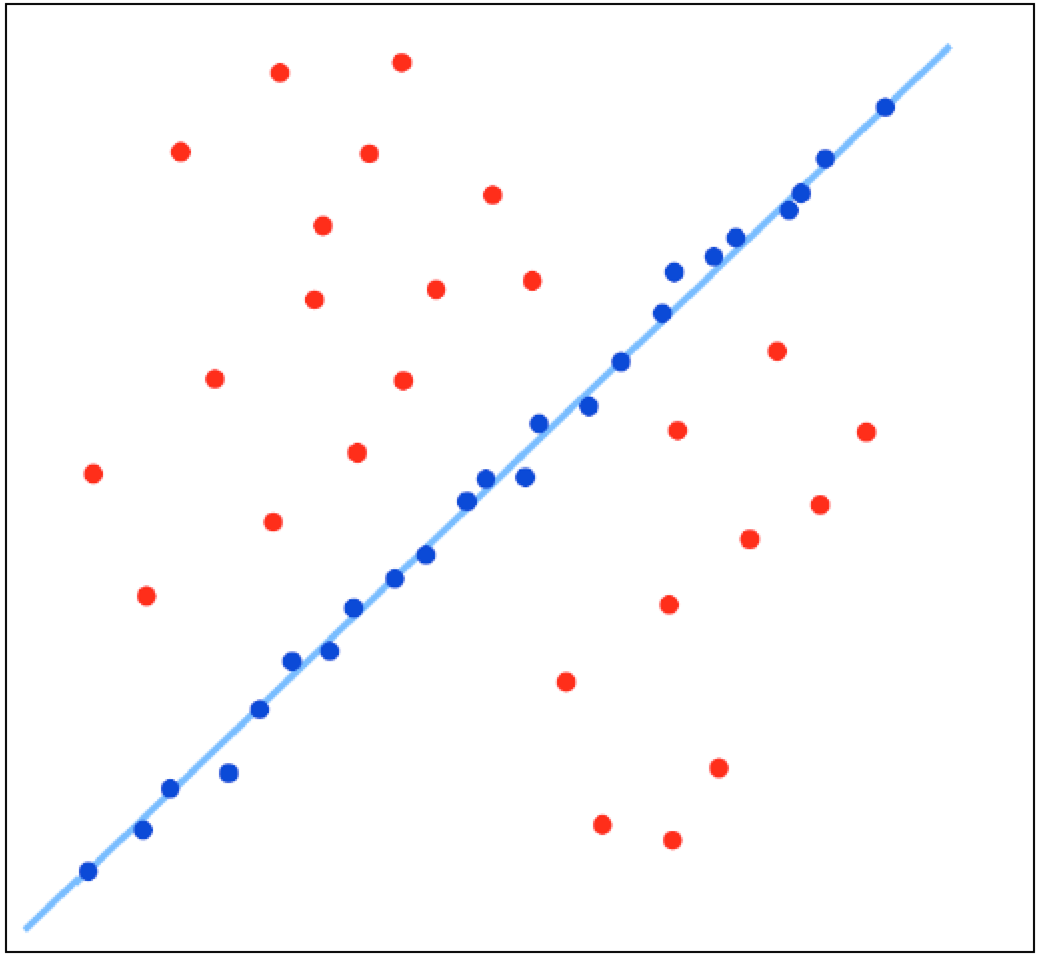
Input : M data points;

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 Homography

- 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).

Recover Homography

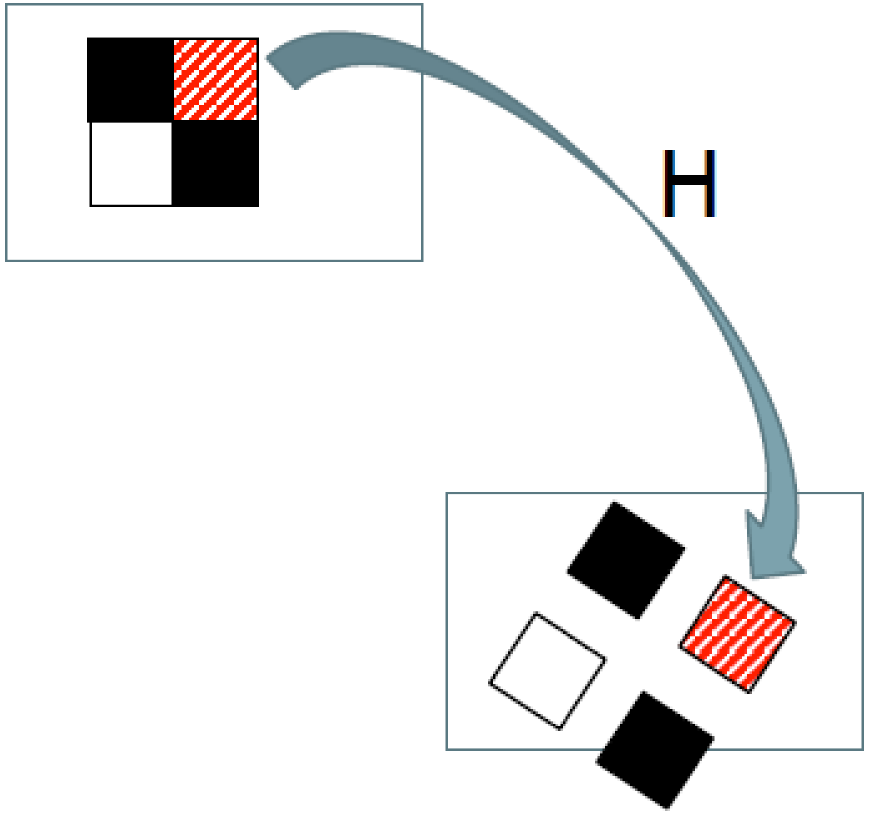
- Solve Ax=0
- cv::eigen
    - (用法可查詢http://docs.opencv.org/2.4/modules/core/doc/operations_on_arrays.html#eigen)

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

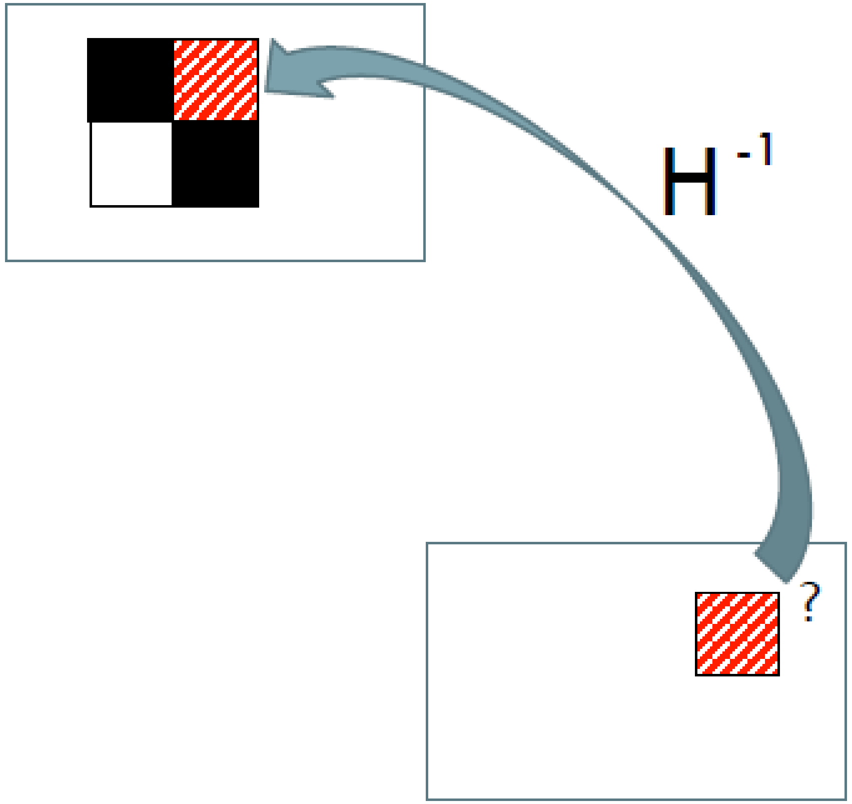
$$\begin{bmatrix} X_1 & Y_1 & 1 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1 \\ 0 & 0 & 0 & X_1 & Y_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1 \\ X_2 & Y_2 & 1 & 0 & 0 & 0 & -x_2X_2 & -x_2Y_2 & -x_2 \\ 0 & 0 & 0 & X_2 & Y_2 & 1 & -y_2X_2 & -y_2Y_2 & -y_2 \\ X_3 & Y_3 & 1 & 0 & 0 & 0 & -x_3X_3 & -x_3Y_3 & -x_3 \\ 0 &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

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 != black ){
        warping;
}
else{
        skip;
}
```

Algorithm(for reference)

1. SIFT feature detection
2. For each feature point in the object image:
    Find KNN points in the image (according to the descriptors)
3. 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 3
4. Use the recovered Homogrphy matrix to warp the image.

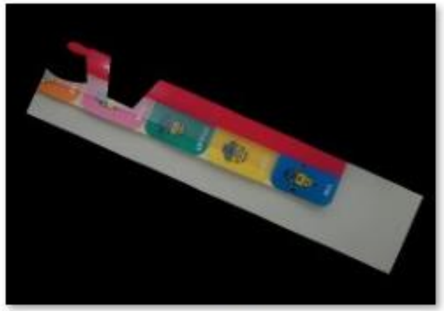
Requirements

- Input :

puzzle1

puzzle2

puzzle3



puzzle4

puzzle5

puzzle6

puzzle7

sample

target

Requirements

- Output :

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(如有要使用，不確定的，再詢問助教)
- 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

Scores

- Find object in target image and warp it.
  - 1 object successful 75 %
  - 2 objects successful +10 %
  - Another testing dataset when demo +10 %

- Other interesting things (ex. Speed up……) +15 % at most

Deadline

- 期限: 2017/05/18 (四) 11:59:59 pm
- 請將作業壓縮並以學號命名: ex 0987654-hw2.zip
  - 資料夾內包含:
    - 1. 學號-hw2-report.pdf
    - 2. code 或 完整專案
    - 3. Result image
- 上傳至e3
- Code需要加上對應流程註解
- 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…