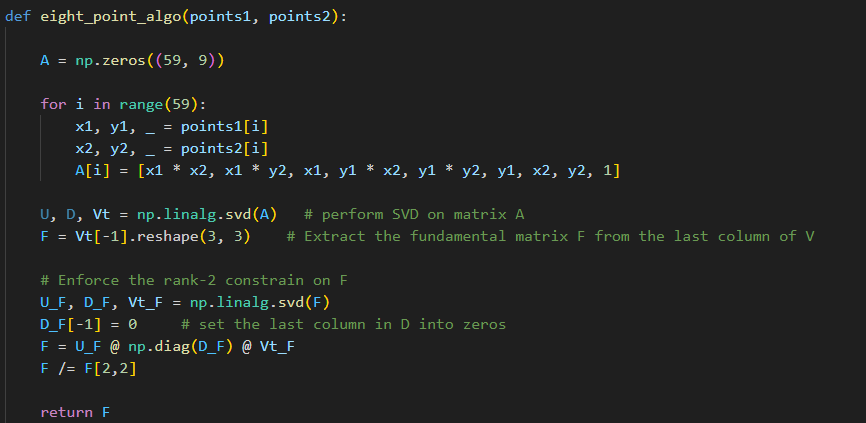
**Q1 Fundamental Matrix Estimation from Point Correspondences**

**Implementation**

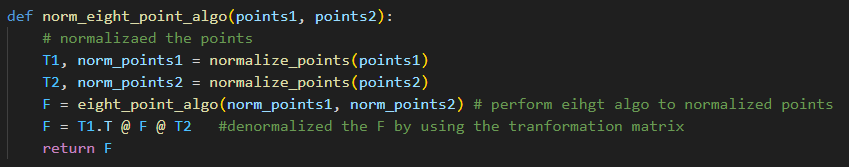
Step 1 of (a.)

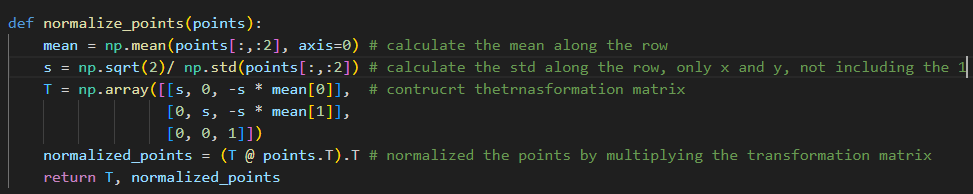
* Code
* Explanation

首先將對應matching point根據公式去計算矩陣A，再對A做SVD分解成三個矩陣U、D、Vt的相乘。取Vt最後一行在reshape成3x3的大小作為fundamental matrix F，再對F做SVD分解成三個矩陣U\_F、D\_F、V\_F。透過將D\_F最後一行設成0，讓F確保可以是一個2-rank的矩陣。最後在將改變後的D\_F乘回去得到最終的F。

Step 2 f (b)

* Code

# normalized eight point algorithm

# normalization

首先將對應matching point根據公式去計算矩陣A，再對A做SVD分解

Step 2 f (b)

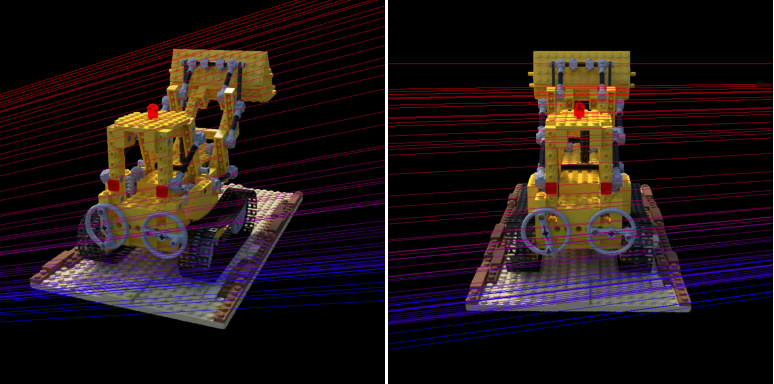
* Explanation

首先對點做normalization，normalization是將每個點減去mean乘上根號2再除與std，根據這樣的想法創建出一個transformation matrix T1、T2。將normalized完的點輸入進eight point algorithm中到F。最後再將F乘上T1的轉置以及T2完成denormalization得到最終的F。

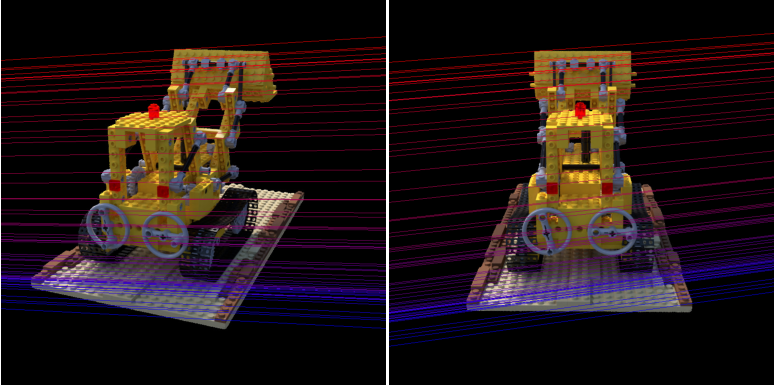
Step 3 of (c)

* Reuslt

Distance: 25.64771846800745 Distance: 25.260657259241153

****

Distance: 0.9005558198177118 Distance: 0.9154447659757211

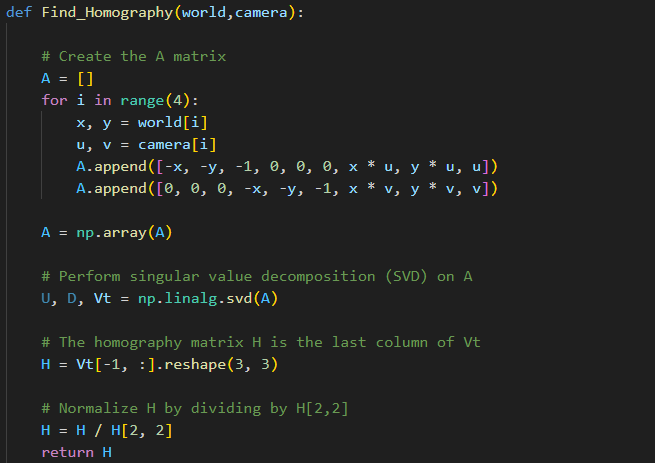
****

**Q2 Homography Transform**

**Implementation**

Step 1 of (a.)

* Code



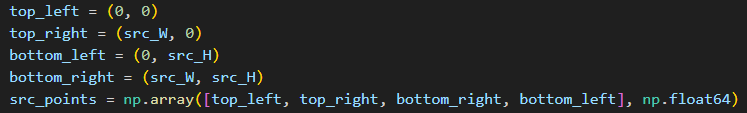
* Explanation

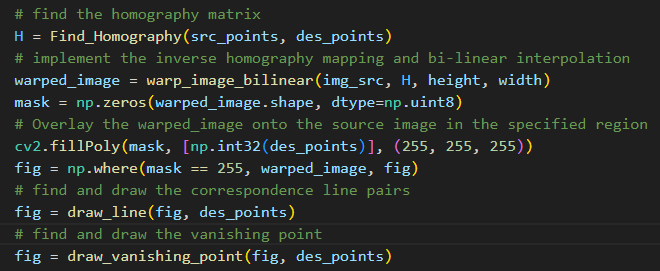
首先將對應點根據homography的公式去計算矩陣A，再對A做SVD分解成三個矩陣U、D、Vt的相乘。取Vt最後一行在reshape成3x3大小

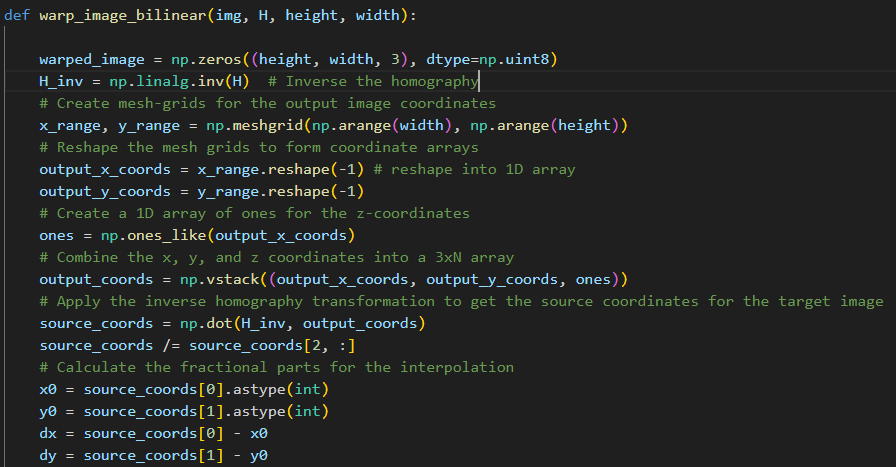
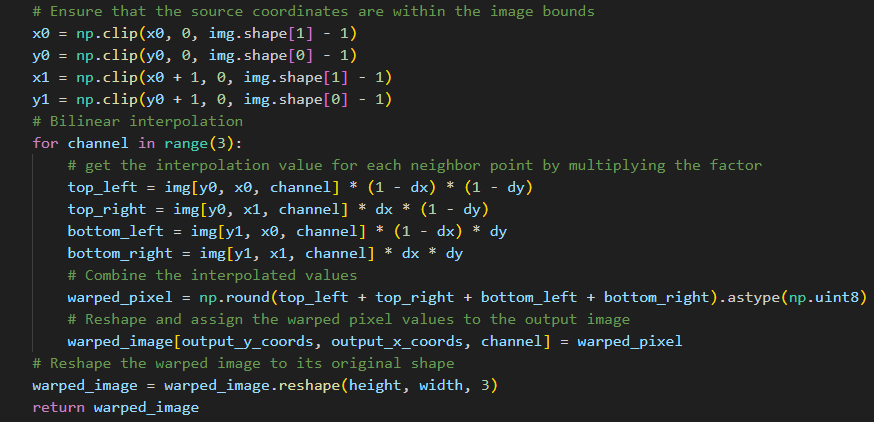
作為homography matrix H。最後就是利用右下角的數值對矩陣H做normalization。

Step 1 of (b.)

* Code

# points for the target image

# implement steps

# detail of function: wrap\_image\_bilinear

Step 2 of (b.)

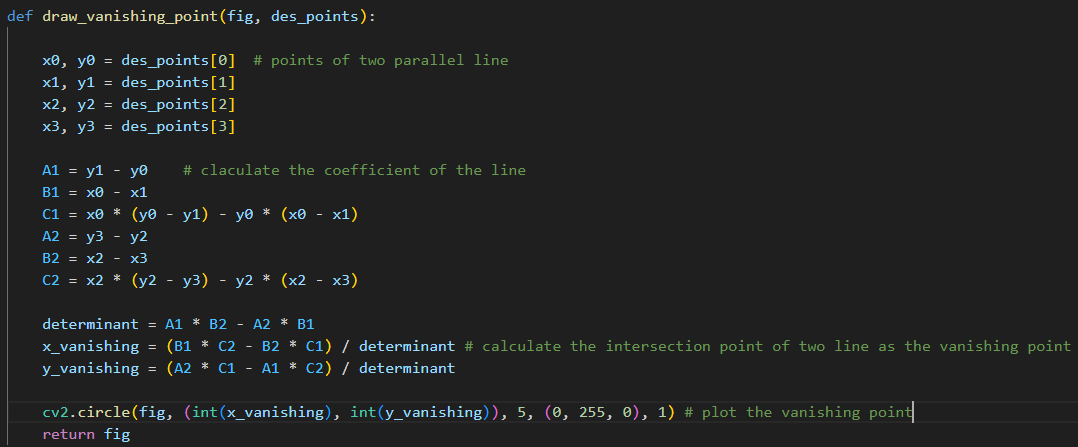
* Explanation

首先對H做inverse再乘上target的點座標，得到target在source image上mapping的座標點。在計算出interpolate的小數部分通過乘以小數部分來取得每個相鄰點的插值值。最後相加並將相加結果分配給輸出圖像，最後將所得到的結果覆蓋在figure上

* Result



Step 2 of (c.)

Code

* Explanation

利用兩條線延伸求出交集的點作為vanishing paoint。