

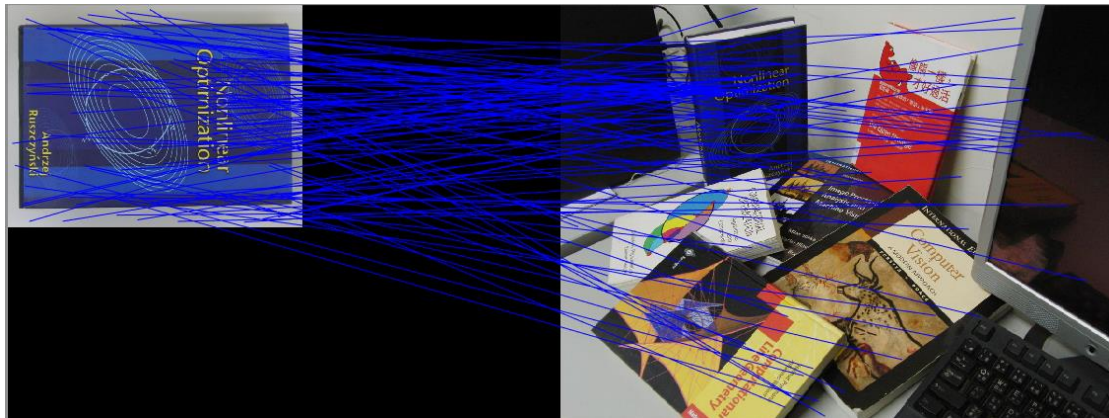
# Computer Vision HW4 Report

102062209 邱政凱

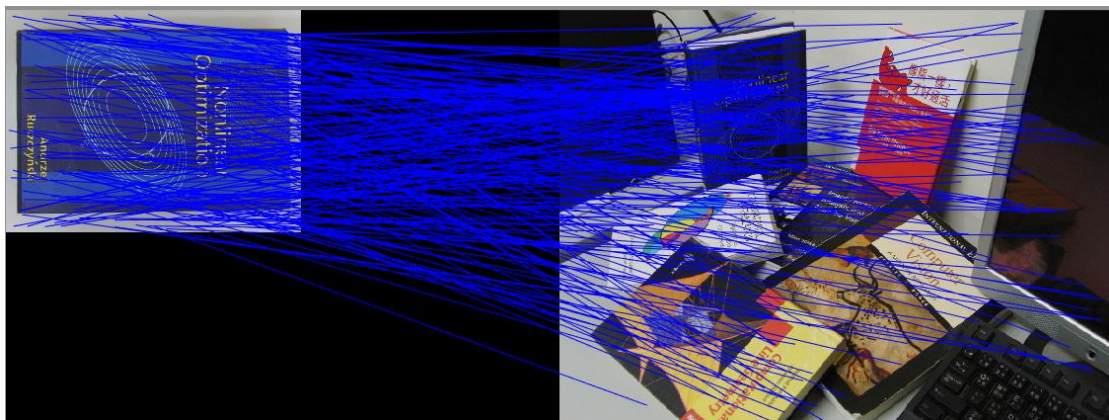
這次的作業要求我們實作的是用 SIFT 擷取出來的特徵點做特徵匹配，並利用 RANSAC 做指定物件的 Object Detection。

第一題要我們做的就是用 VLFeat 的 API，分別針對每本書的圖片和混雜了很多書的場景圖做 SIFT 的特徵擷取。SIFT 擷取出來的特徵是 128 維度的向量，兼具 translation、rotation、scale invariant 的性質。我們只要把欲匹配的物件上的每個特徵點，去對場景上所有的特徵點做 L2 的距離計算，如果在場景上找到距離最小的特徵點的距離小於某個 THRESHOLD，則我們便可以認為這兩點是物體上的同一點。Threshold 的值越小，匹配到的特徵點數量就會越少，反之亦然。另外就是 VLFeat 的 SIFT 裡面的 Magnif 參數。這個參數可以用來設定對應每個特徵點的 SIFT Descriptor 取樣時的範圍大小，越大的 Magnif 表示 Descriptor 的取樣範圍越大。實作方式不困難，用 VLFeat 的 SIFT 找出特徵點之後就只要去計算各個特徵點之間的 norm 就好了。

以下是每本書對應不同參數值的結果：

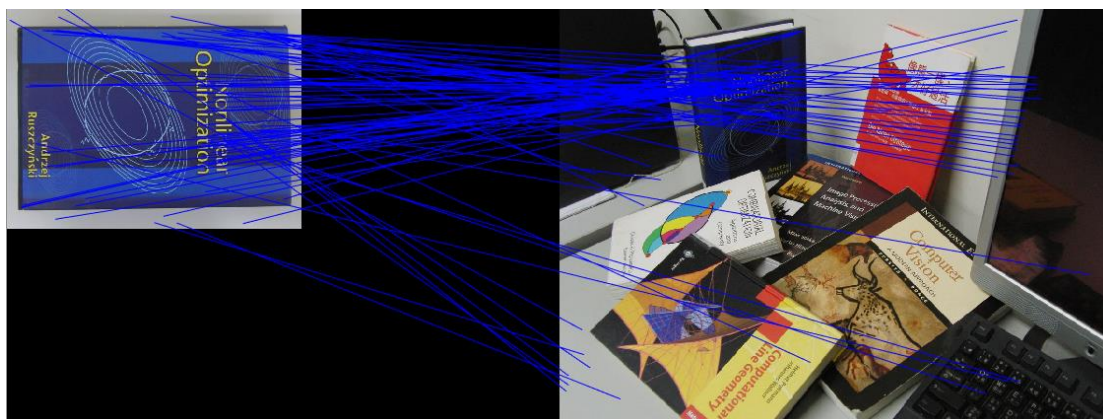


BOOK1, Magnif = 3 , threshold = 150 。



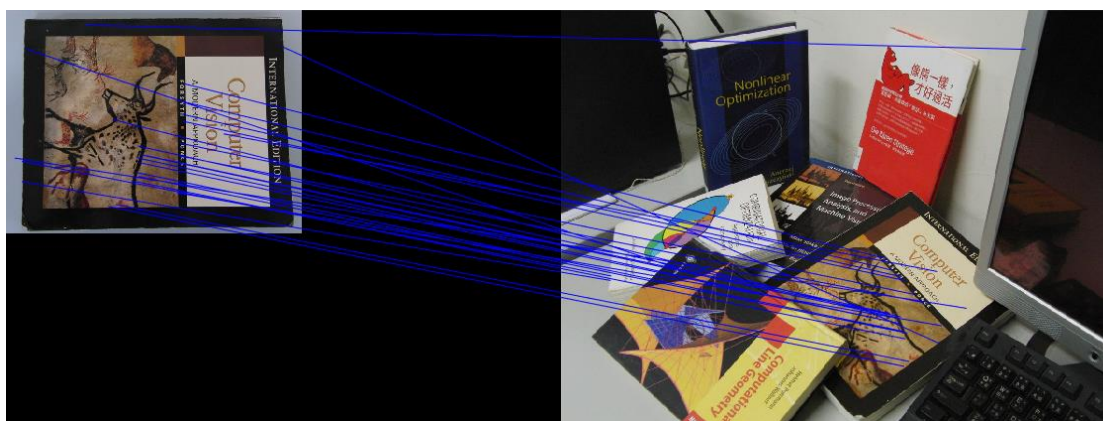
BOOK1, Magnif = 3 , threshold = 230 。

可以很明顯的看到高 Threshold 值找出來的特徵點數量明顯少於低 threshold 值

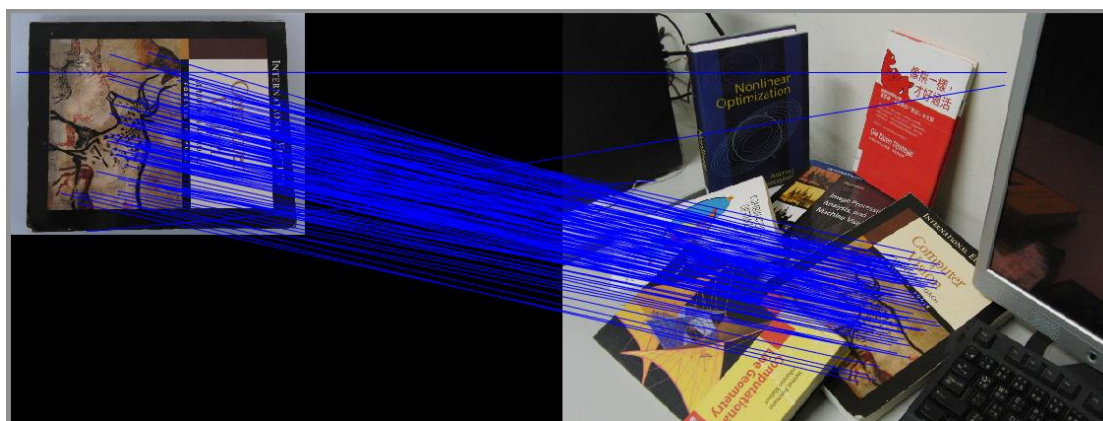


BOOK1, Magnif = 10, threshold = 230 。

同樣的 threshold，可以發現較高的 Magnif 值有過濾的效果，可以讓匹配點更集中在場景中對應的書上，少了許多雜訊(匹配錯誤)。(best result)



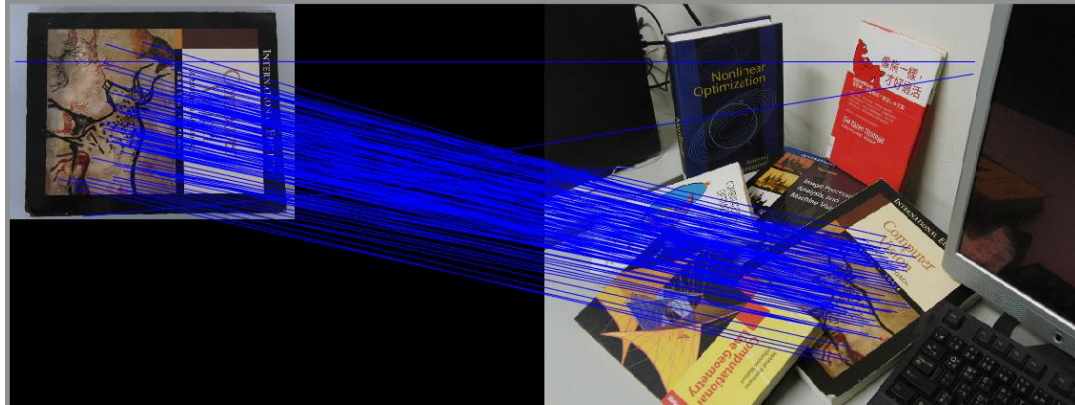
BOOK2, Magnif=3, threshold=70 。



Book2, Magnif=3, threshold=150 。

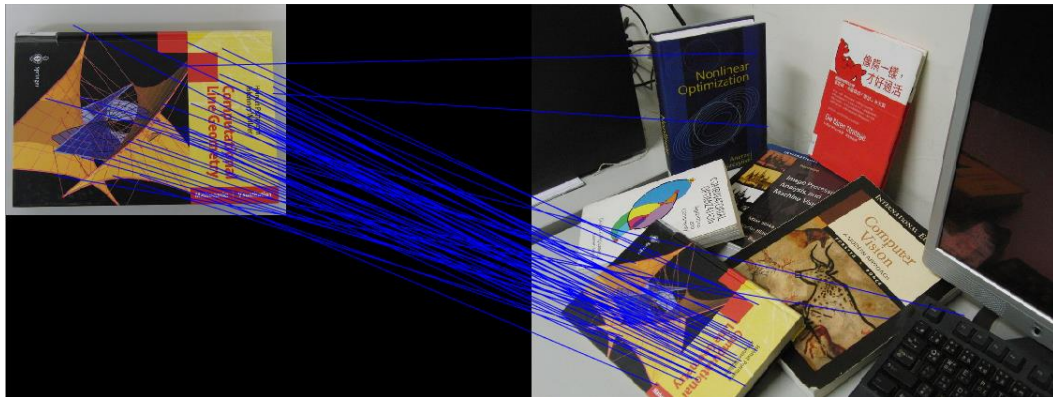
這張圖片相比於前一張 book1，即使 Magnif 值較低場景匹配的特徵點仍比較聚集在對應的書本上，較少雜訊。



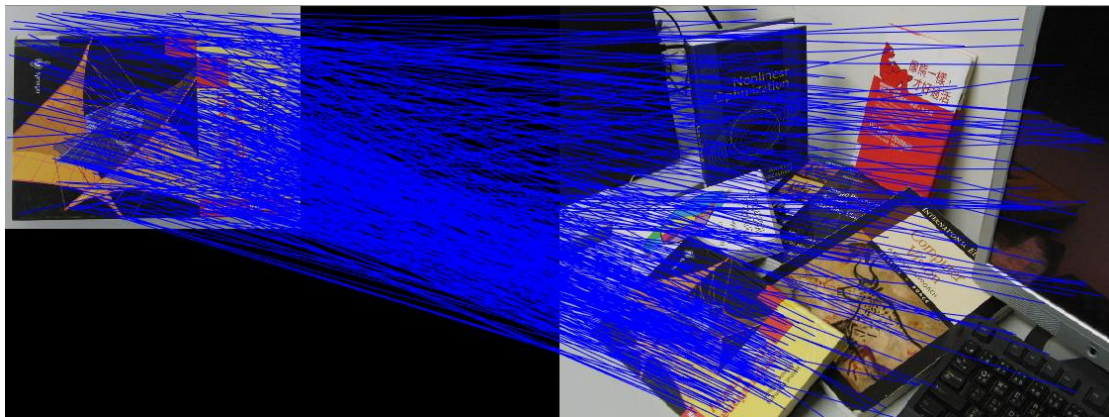


Book2, Magnif=10, threshold=150。

增加 Magnif 之後不意外的可以看到已經幾乎所有特徵點都集中在場景中對應的書上了。可以推測這本書的封面應該是屬於特徵很明顯的。(best result)。

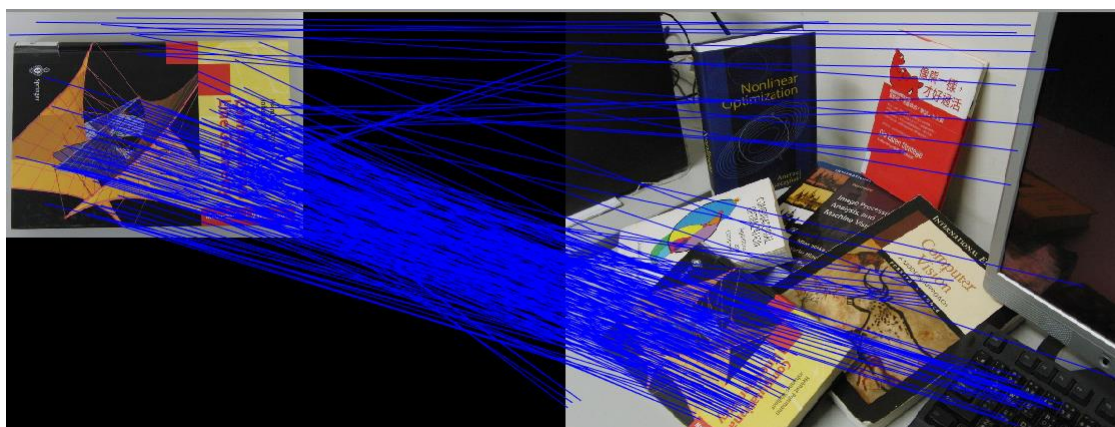


Book3, Magnif=3, threshold = 100.



Book3, Magnif=3, threshold=250。

BOOK3 的特徵點數量頗多，threshold 值一高場景上的雜訊也就馬上變多很多。



Book3, Magnif=15, threshold=250。

不過我們還是可以用調整 Magnif 值來讓特徵點匹配集中於對應的書上。雖然到目前為止看來好像 Magnif 值高一點可以減少雜訊，但是若是 Magnif 值太高也會讓場上可以使用的特徵點數量減少，從而增加 Object Detection 的困難度，因此抓取適當的 Magnif 值是相當重要的。

第二題的部分是這次作業的重頭戲，也就是要求我們藉由 Random sample consensus(RANSAC)重複隨機取樣的方式讓程式自己找出對應的 best Model。而這次題目要求的 Model 是把單一書本圖片用 Projective transformation 的方式轉換到場景圖片中同樣書本的對應位置上的 Homography Matrix。

因為 Homography 的計算至少需要 4 組對應點，所以 RANSAC 的初始 putative matching 取樣數(n)就設為  $n=4$  (不論哪一題我都這樣設)。用隨機的四組點算出對應的 Homography 之後把書上的其他特徵點都用這個 Homography 去做 Projective transformation，如果書上的特徵點轉換後的結果跟原本場景中對應的書上的特徵點的距離夠小的話，我們就把這組特徵點設為 inlier，否則設為 outlier。如果 inlier 的數量夠大( $\geq d$ )，且大於 iterate 到目前為止的最大 inlier 數的話，就用這些 inlier 再去計算一次 best Homography matrix。Iterate 以上的步驟到 k 次為止，並保留最好的 best Homography matrix。之後我的程式會讓使用者自行抓取單本書圖片上，對應書封面的四個角落的四個點(必須順或逆時針選取)，之後就會用紅、綠、藍、白四色的線在原圖和場景圖中把書的對應邊用相同顏色標記出來。作法很簡單，就是把四個點用 best Homography 轉換。之後我還會再次把所有特徵點匹配用 line 畫出來，inlier 用藍色，outlier 用綠色。以及 inlier feature point 和 corresponding (transformed) point 之間會用 deviation vector(我用 matlab quiver)的方式把偏差向量畫出來並用 circle 標記(紅色圈是正確特徵點位置，綠色圈是 transformed 特徵點的位置)。

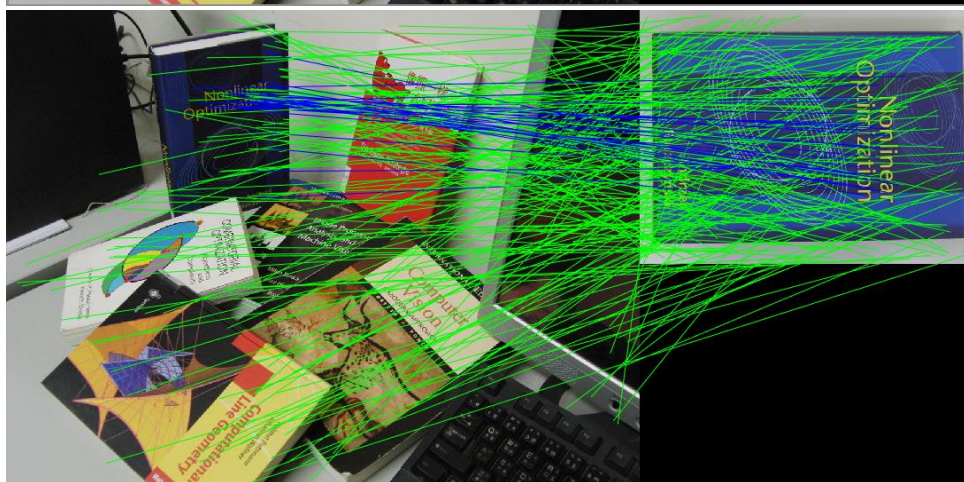
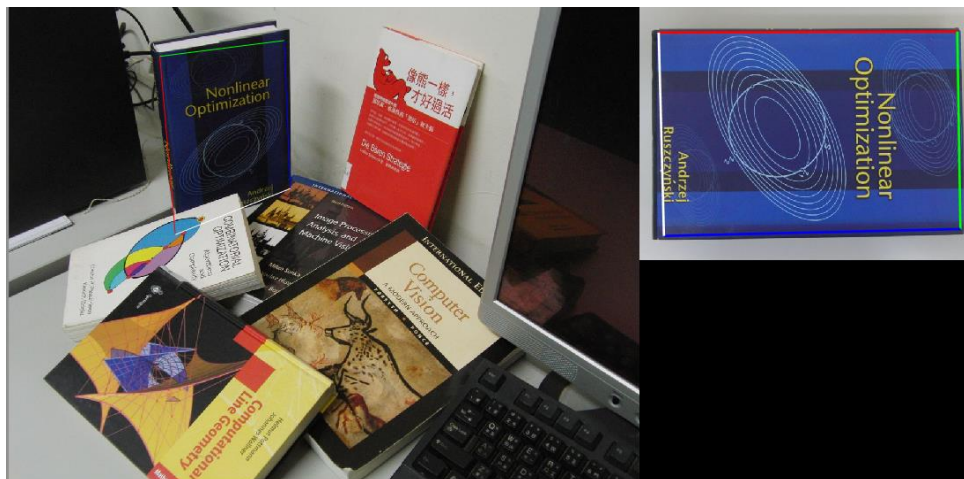
以下是我的結果：

(每一張圖都會用兩組 RANSAC 參數來進行所有操作)

(每本書的第一組參數都是我認為較好的結果)



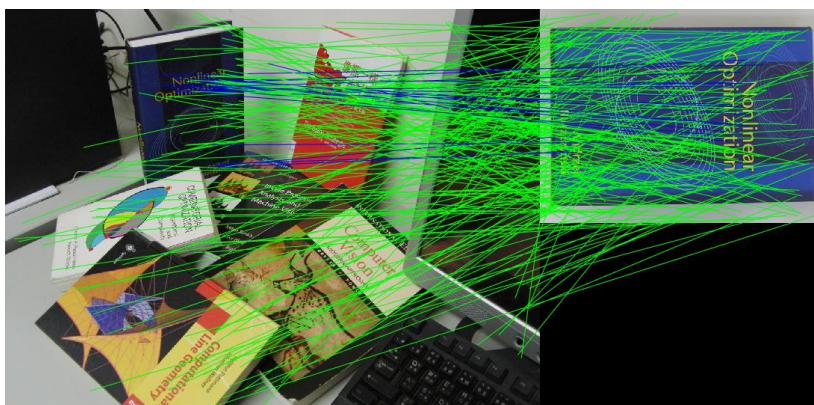
以下是 BOOK1 用  $k=3000$ ,  $t=10$ ,  $d=13$  的參數做出來的結果



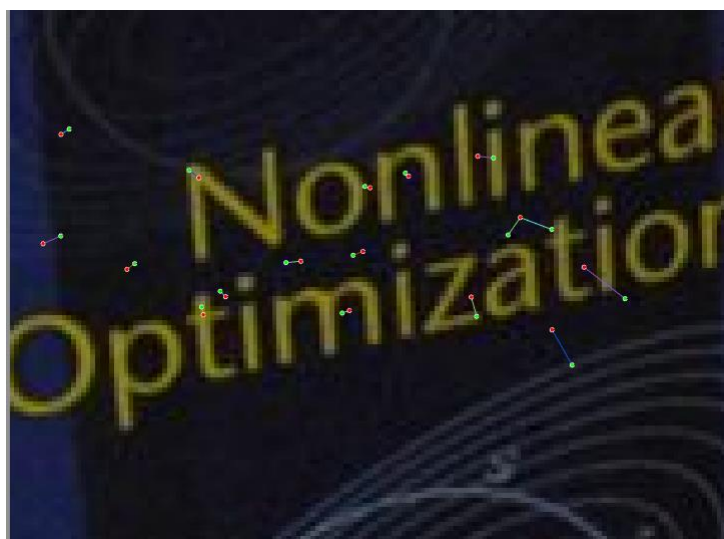
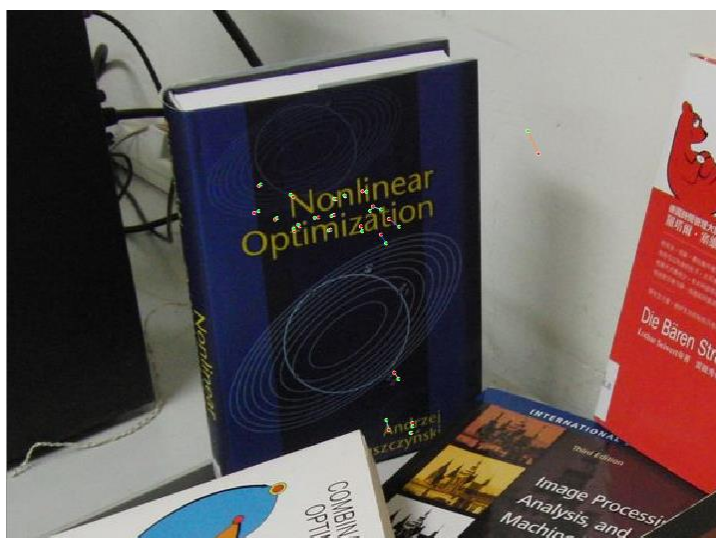


第一本書的特徵點其實比較少，如果  $t$  值設太小或  $d$  值設太大會容易找不到可以使用的 best homography。

換一組參數： $k=100, t=15, d=5$ 。

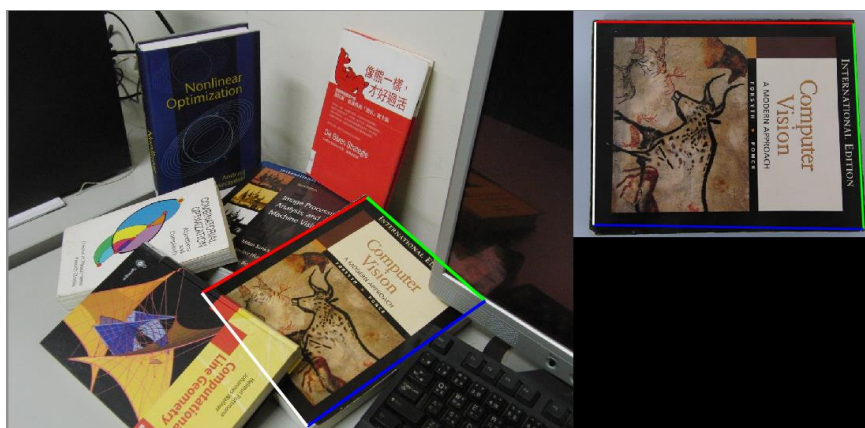


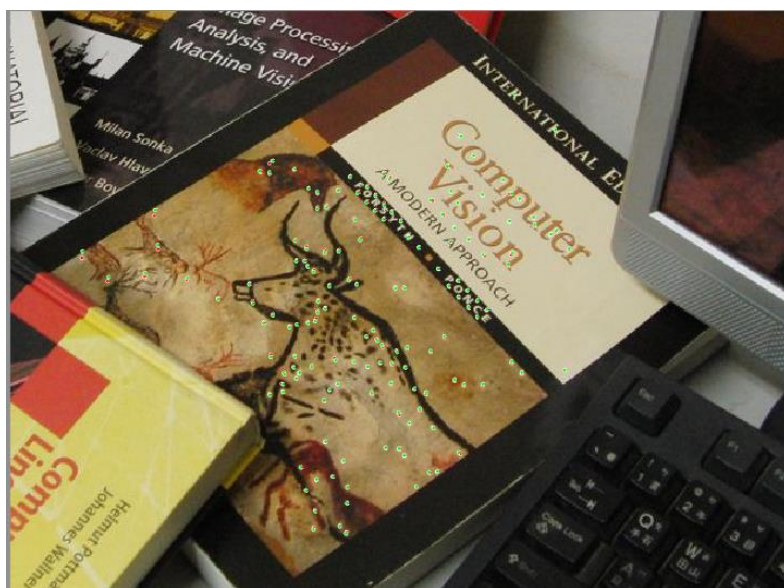
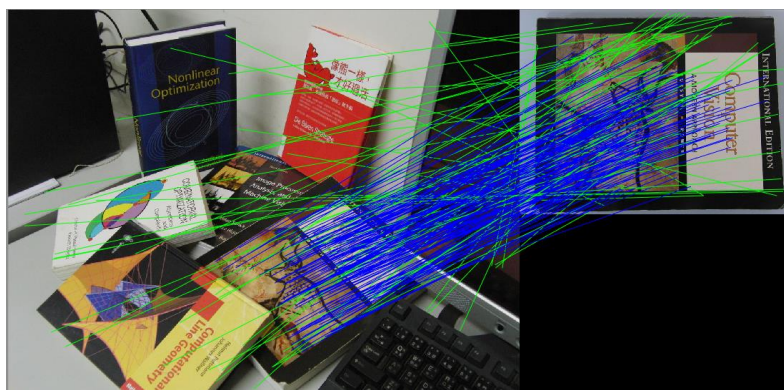




減少 iteration 次數( $k$ )，增大 inlier 距離標準(可能會抓到不希望的特徵點匹配)以及減少最小 inlier 數( $d$ )，我們可以看到算出來的 transformation 是會偏離的，從 deviation vector 也可以看出對應的 deviation vector 偏移量平均有變大。不過在這本書還沒有很明顯。

以下是第二本書的第一組參數： $k=1000, t=5, d=10$ 。

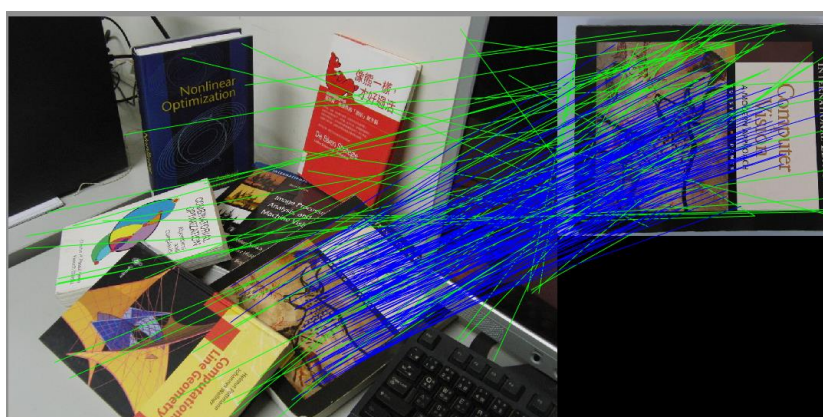
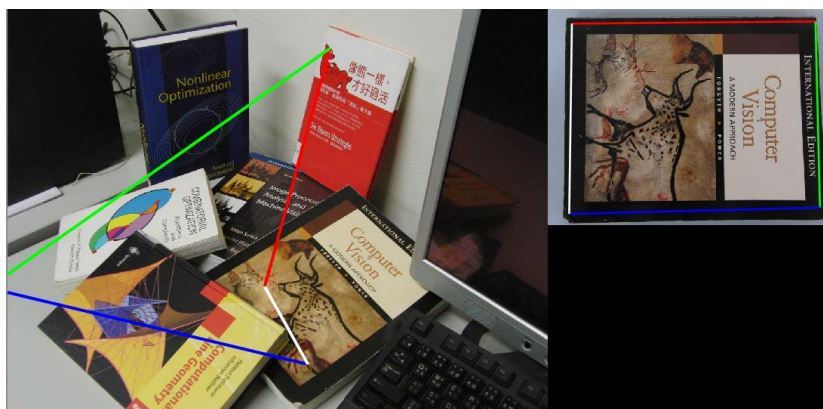




第二本書因為書本特徵比較明顯，其實 inlier 占有所有特徵點的比例算是滿高的。整體而言算是(我覺得)很棒的效果。就算書在場景中好幾個邊被其他物體擋住還是可以找出它的位置。

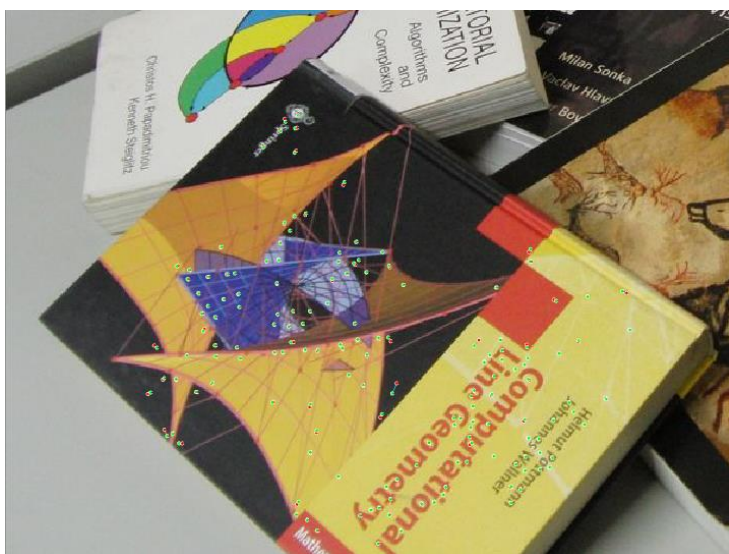
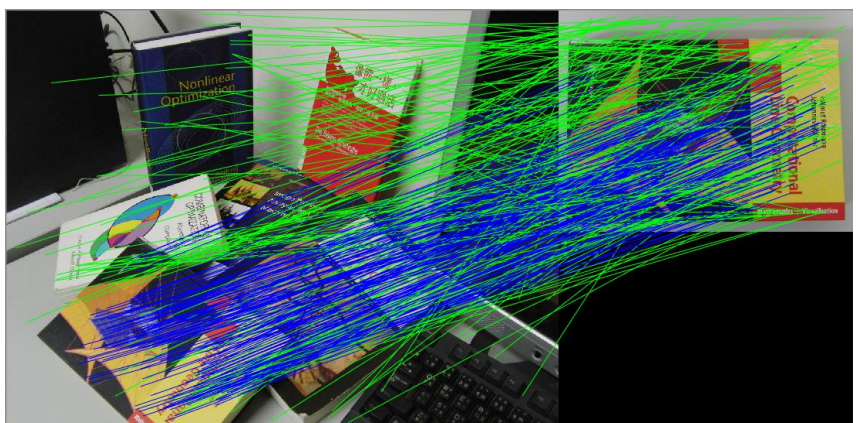
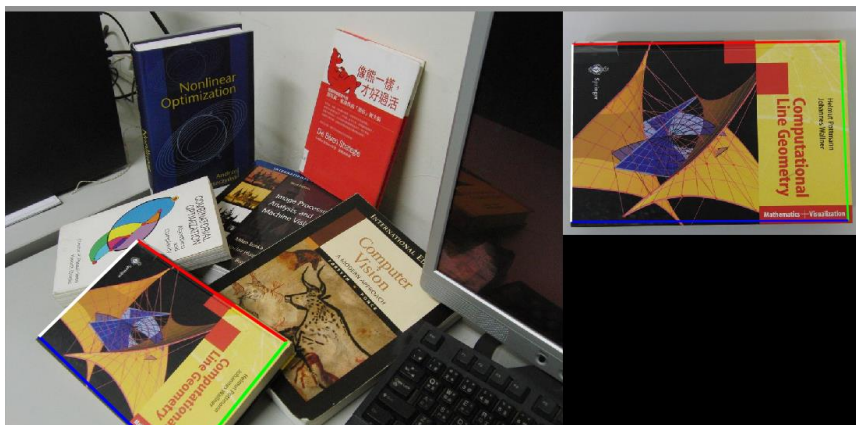
第二組參數： $k = 1000, t=80, d=10$ 。





這組參數很明顯的展示出了  $t$  值設太大的結果。就算是 Book2 這樣子特徵明顯的物件，若是把 inlier 的條件設太低(distance 下限值太大)，也仍然會抓到許多不是出現在書上的特徵點，造成結果偏離(deviation vector 應該畫得很清楚 QQ)。

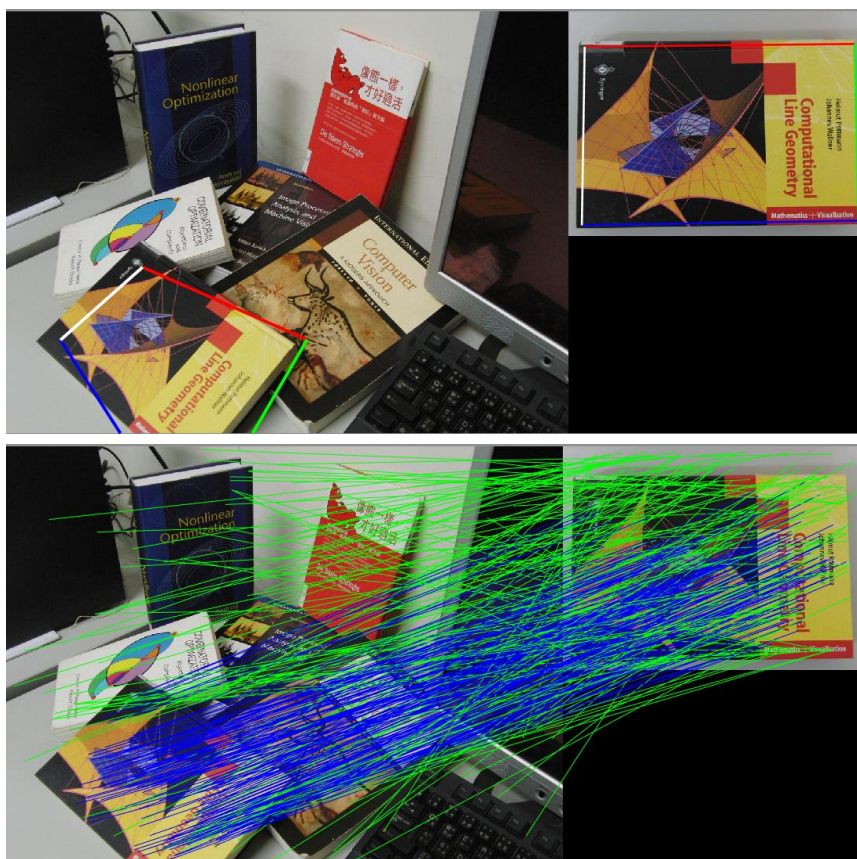
接著是最後一本 Book3。首先是第一組參數： $k=1000, t=10, d=20$ 。







Book3 也算是特徵明顯的書，可以發現 inlier 幾乎都落在 BOOK3 的書本上。  
第二組參數： $k=100$ 、 $t=50$ 、 $d=5$ 。





結果應該跟前面都是吻合的，不論是  $k$ ， $t$  還是  $d$  值都會影響到 fit 出來的 model 的結果。