

影像處理與電腦視覺 作業五

統計所 李權恩 R26094022

Introduction

在本篇作業當中主要使用 Scale Invariant Feature Transform (SIFT) 來進行圖像對齊，在雜亂的場景中找到書籍圖像與輸入圖像的對齊方式，並實作 RANSAC-based 的方法來進行關鍵點配對。下圖為本次作業使用到的影像資料。

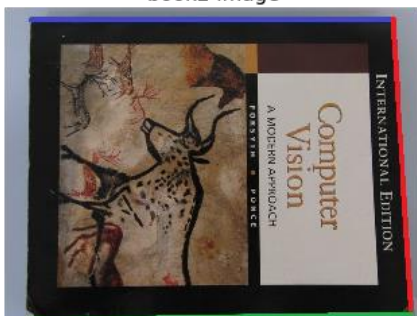
scene Image



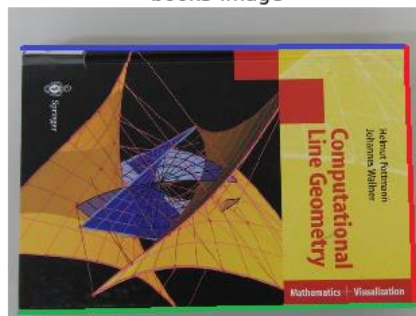
book1 Image



book2 Image



book3 Image



程式碼於連結中，主要使用 python 完成

<https://github.com/Quan-En/DIPCV/tree/main/assignment5>

另外在 SIFT 上使用 opencv 套件，由於最新版的 opencv 在 SIFT 或 SURF 有版權問題需要降級。使用的 python 版本與 opencv 版本如下

Python 3.6.5

opencv-contrib-python==3.3.0.10

Methodology

1. Scale Invariant Feature Transform (SIFT)

SIFT 演算法主要可以分成 4 大步驟，分別為

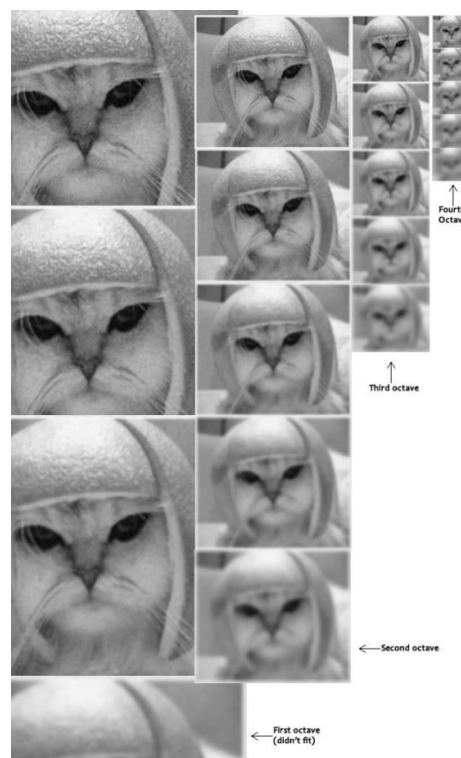
- 選擇尺度空間峰值 (Scale-space peak selection)
- 定位關鍵點 (Keypoint localization)
- 方向配置 (Orientation Assignment)
- 關鍵點描述 (Keypoint descriptor)

最終再藉由關鍵點的位置與描述進行兩兩圖片之間的配對，下方將簡短描述各步驟的作法及目的。

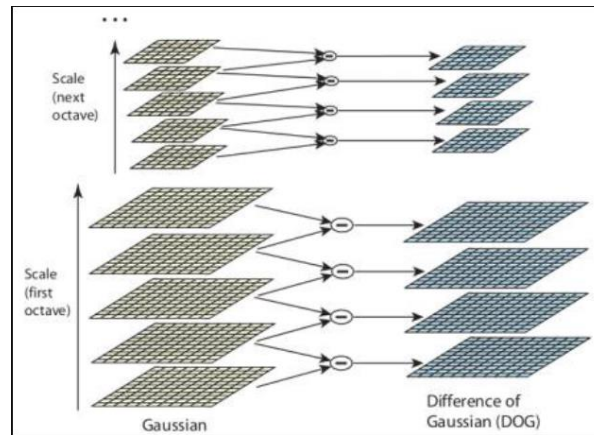
選擇尺度空間峰值 (Scale-space peak selection)

最一開始的步驟主要是希望能在圖像當中找到特徵的潛在位置，除此之外，成像位置的遠近、相機的縮放等等都會影像圖片的尺度大小，因此在搜尋潛在特徵時須同時考量影像尺度的問題。

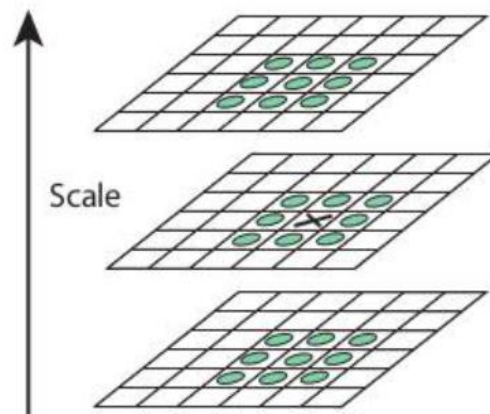
為了考量尺度問題會先將圖片進行縮放，根據原始圖片的大小，通常會分成 8 種層級，每層級的比例是前一層級的一半，如下圖所示



再來為了可以取得影像的潛在特徵點，會利用不同 σ 大小的高斯核 (Gaussian kernel) 進行卷積 (Convolution)，最後再將這些影像相減，此步驟主要是為了計算 Laplacian of Gaussian approximation (LoG)，主要 LoG 的計算成本較高，如下圖所示。



接著就會在圖像上搜索尺度和空間上的局部極值。例如，將圖像中的一個像素與其 8 個相鄰像素以及下一個比例的 9 個像素和前一個比例的 9 個像素進行比較。如果是局部極值，就是潛在的關鍵點。這基本上意味著關鍵點在該比例下得到最好的表示。如下圖所示：



定位關鍵點 (Keypoint localization)

一旦找到潛在的關鍵點位置，就必須對其進行細化以獲得更準確的結果。主要是對泰勒級數來獲得更準確的極值位置，如果該極值處的強度小於閾值(根據論文為 0.03)，則將其拒絕。DoG 對邊緣的響應較高，因此邊緣也需要去

除。為此使用 2×2 Hessian 矩陣 (\mathbf{H}) 來計算主曲率。如果比率大於閾值，在 OpenCV 中稱為 `edgeThreshold`，則該關鍵點將被丟棄。主要消除了任何低對比度的關鍵點和邊緣關鍵點。

- Reject flats:

- $|D(\hat{x})| < 0.03$

- Reject edges:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

Let α be the eigenvalue with larger magnitude and β the smaller.

$$\text{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$

$$\text{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

Let $r = \alpha/\beta$.
So $\alpha = r\beta$

$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r},$$

$(r+1)^2/r$ is at a min when the 2 eigenvalues are equal.

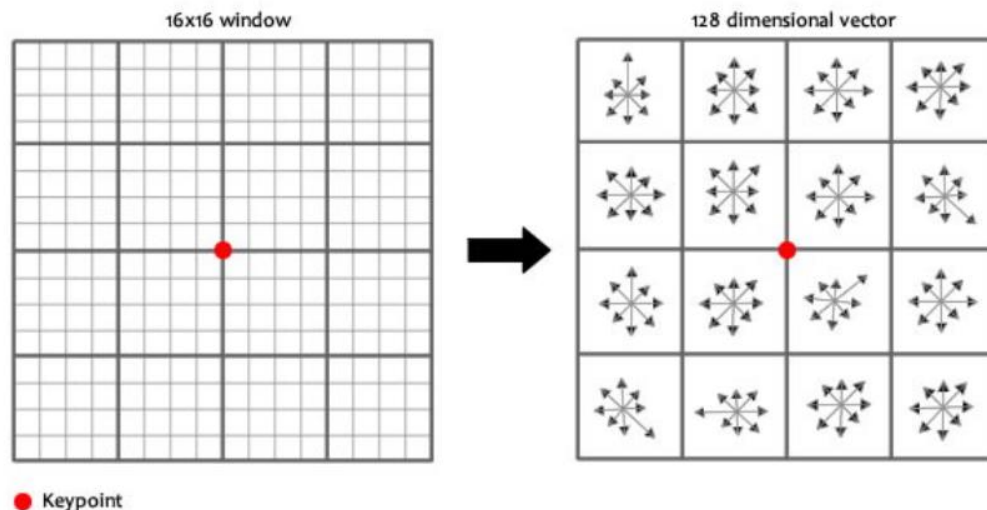
- $r < 10$

方向配置 (Orientation Assignment)

再來為了考慮圖片的旋轉方向，我們需要對每個關鍵點配置一個方向，來達成圖像旋轉的不變性。根據比例在關鍵點位置周圍取一個鄰域，並在該區域計算梯度大小和方向。創建具有 36 個 bin 覆蓋 360 度的方向直方圖。取直方圖中的最高峰。如此便可創建具有相同位置和比例但方向不同的關鍵點，它有助於匹配的穩定性。

關鍵點描述 (Keypoint descriptor)

在關鍵點描述的部分，先取關鍵點周圍的 16×16 鄰域並分成 16 個 4×4 大小的子塊。對於每個子塊，創建 8 個 bin 方向直方圖。因此共有 128 個 bin 值可用。這些值會以一個向量來表示關鍵點的描述。



在完成上述步驟後，通過識別它們最近的鄰居來匹配兩個圖像之間的關鍵點。另外在某些情況下，第二個最接近的匹配可能非常接近第一個，這可能是由於噪音或其他一些原因而發生的。屆時採用最近距離與次近距離的比率，如果大於 0.8，則拒絕。

2. RANdom SAmple Consensus (RANSAC)

RANSAC 算法是一種通過對觀察到的數據進行隨機抽樣來估計模型參數的學習技術。給定一個數據元素同時包含內點和異常點的數據集，RANSAC 使用投票方案來找到最佳擬合結果。數據集中的數據元素用於為一個或多個模型投票。該投票方案的實施基於兩個假設：噪聲特徵不會一致地對任何單個模型進行投票，並且有足夠的特徵可以就一個好的模型達成一致。

RANSAC 算法本質上由迭代重複的兩個步驟組成：

第一步，從輸入數據集中隨機選擇包含最少數據項的樣本子集。僅使用該樣本子集的元素計算擬合模型和相應的模型參數。樣本子集的基數最小，足以確定模型參數。

第二步，算法檢查整個數據集的哪些元素與第一步獲得的估計模型參數實例化的模型一致。如果數據元素在某個誤差閾值內不適合由估計模型參數集實例化的擬合模型，則該數據元素將被視為異常值，該誤差閾值定義了可歸因於噪聲影響的最大偏差。

為擬合模型獲得的內點集合稱為共識集。RANSAC 算法會反復重複上述兩個步驟，直到在某次迭代中得到的共識集有足夠的內點。

在本次作業當中 RANSAC 主要用來篩選適合的成對關鍵點，考慮兩張圖像對應點的關係如下

$$\begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} \cong \mathbf{H} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

則可表示成 $\mathbf{A}\mathbf{h} = \mathbf{0}_{2 \times 1}$ ，其中 $\mathbf{h} = [\mathbf{H}[0,:], \mathbf{H}[1,:], \mathbf{H}[2,:]]^T$ 為未知參數，而

$\mathbf{A} = \begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -y'_i x_i & -y'_i y_i & -y'_i \end{bmatrix}$ ，考慮有 n 組配對的關鍵點時， $\mathbf{A} \in \mathbf{R}^{2n \times 6}$ 。

求解 \mathbf{h} 為一最佳化問題，目標如下：

$$\underset{\mathbf{h}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{h}\|^2, \text{ where } \|\mathbf{h}\| = 1$$

則 $\hat{\mathbf{h}}$ 為 $\mathbf{A}^T \mathbf{A}$ 最小 eigenvalue 的 eigenvector，上述的估計過程至少需要 4 組樣本點，而 RANSAC 便是在每次從所有成對的關鍵點當中隨機抽取四組進行 \mathbf{h} 的估計，將越符合估計結果的樣本歸類為 Inlier 反之則為 Outlier，透過反覆迭代抽樣、估計及樣本歸類，如果 Inlier 達到一定比例(實作設定為 60%)或者達到迭代次數上限則停止。

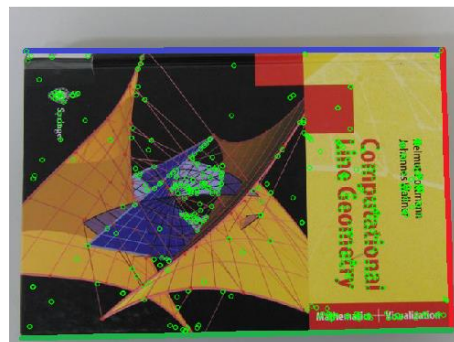
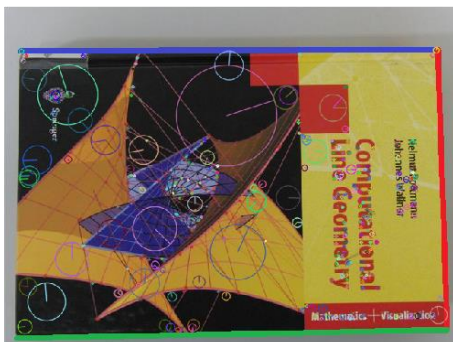
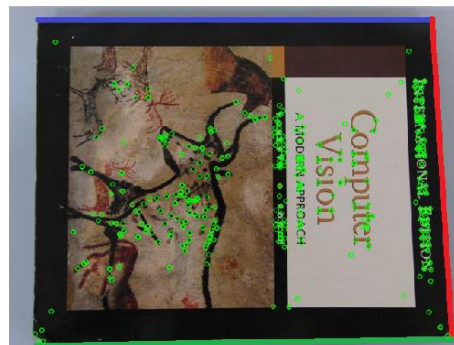
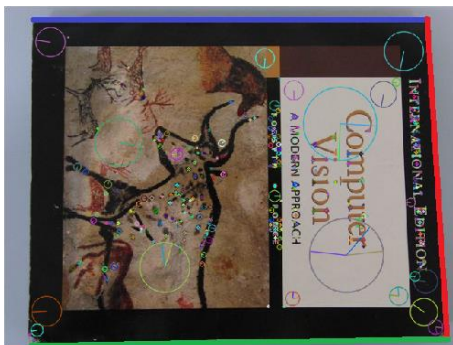
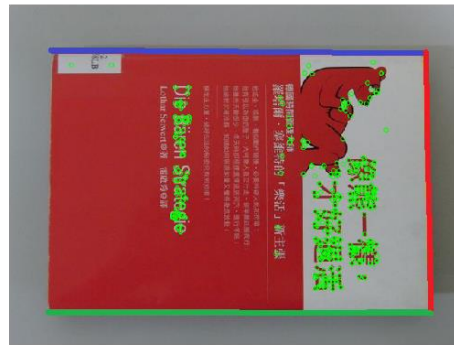
Implement result

1. 首先第一張圖是所有的圖片進行關鍵點偵測的結果，圖片右行僅呈現關鍵點的位置，左行額外體現了強度與梯度方向。

Train keypoints With Size



Train keypoints Without Size

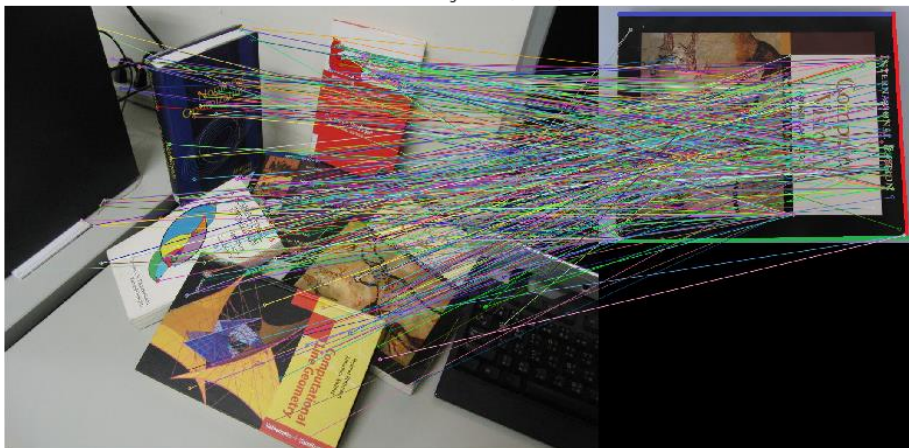


2. 下列三張圖分別是書堆的場景圖與不同的三本書各自進行關鍵點配對的結果，對於全部匹配的效果其實並不好，可以發現配對的許多對其實並不是同樣的位置。

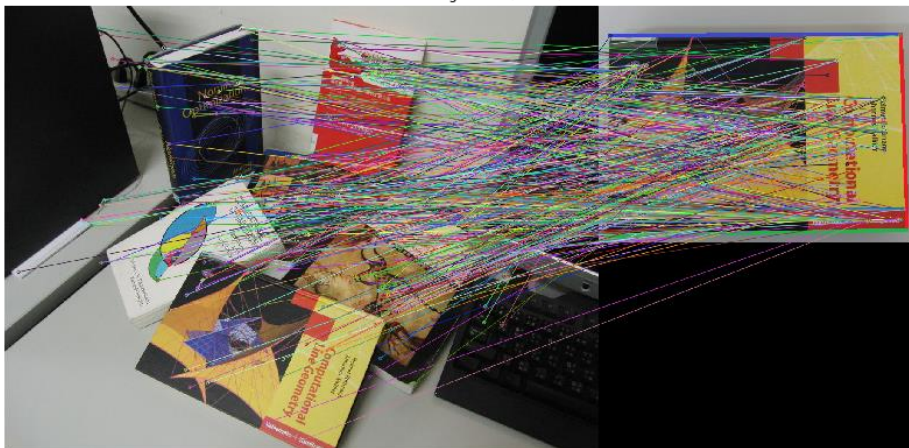
Best Matching Points (book1)



Best Matching Points (book2)



Best Matching Points (book3)



3. 下列三張圖是使用 RANSAC 演算法的結果，綠色配對線代表 Inlier 藍色則代表 Outlier，可以發現配對效果較上一階段直接配對所有關鍵點來的正確，綠線都有匹配到對應的書本上。

