

VFX2023-Group26_HW2

Wei Ting-Yu 魏廷宇 R11942104

Charles Huang 黃湛元 R11942180

程式實作Colab連結

[https://colab.research.google.com/drive/1iJm-fhG_4kIrWzssW5LvgprrlhWjIMxI?
usp=sharing](https://colab.research.google.com/drive/1iJm-fhG_4kIrWzssW5LvgprrlhWjIMxI?usp=sharing)

Introduction

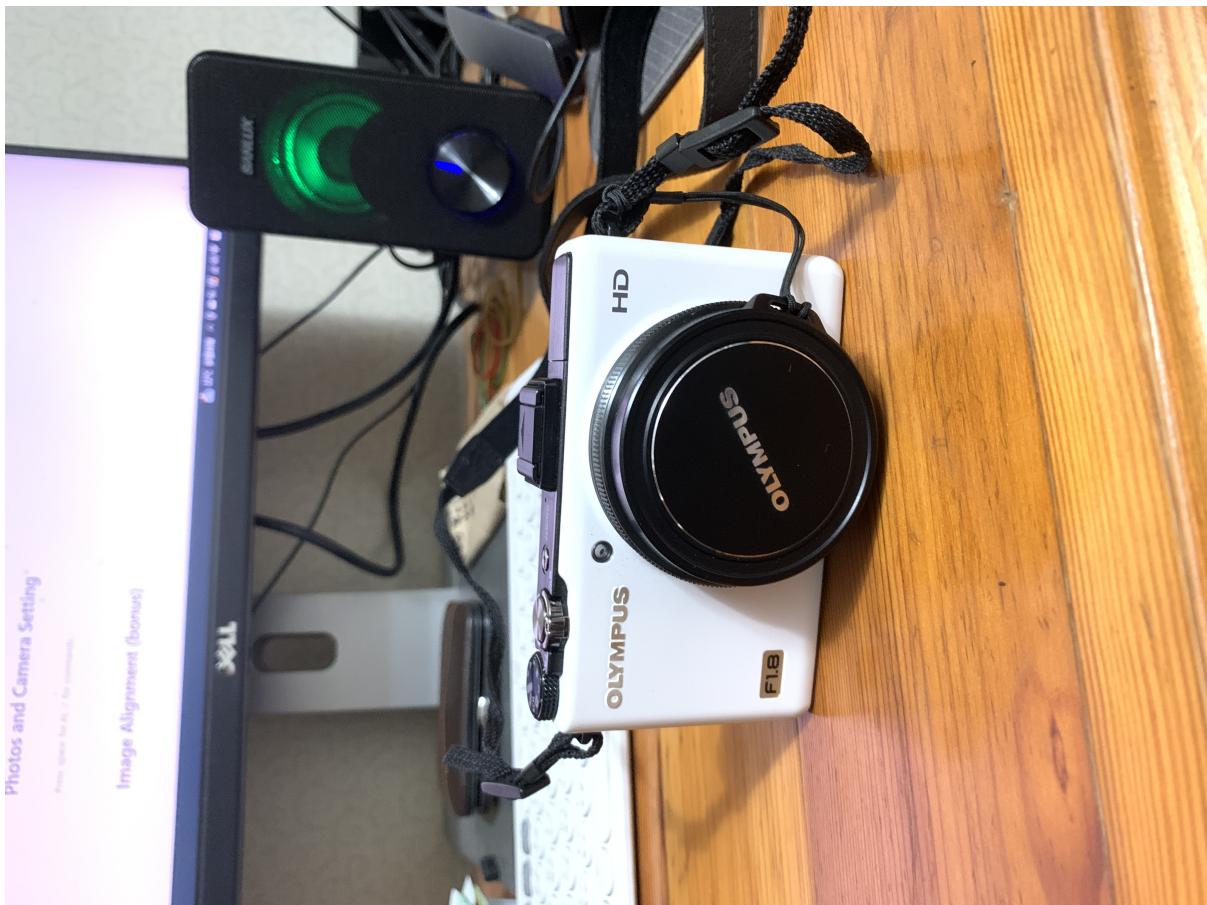
這次的作業需要實作全景影像的縫合，主要會分成 4 個部分：

1. feature detection
2. feature description
3. feature matching
4. image blending

在本次作業中，我們將參考論文 Recognising Panoramas 中提出的方法來實現全景影像的縫合。我們使用 Harris Corner Detector 來檢測圖像中的特徵點，並使用 SIFT Descriptor 的方法描述符來計算特徵點的特徵向量。通過 feature matching 和 image blending，我們可以將多張圖片拼接成一張無縫的全景影像。在本報告中，我們將描述我們的方法的各個步驟，包括 feature detection、feature description、feature matching 和 image blending 等。最後我們會展示我們的實驗結果，以及討論方法的優點和不足之處。

Photos and Camera Setting

使用的相機：Olympus XZ-1

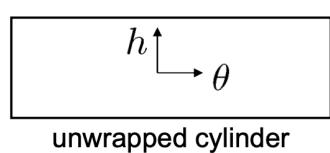


Cylindrical projection

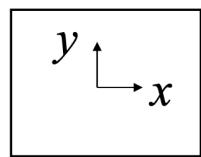
我們首先透過下方的公式將原始照片依據focal length投影到圓柱體的座標上，這麼一來越靠近邊緣的圖片才不會被扭曲。我們計算focal length的方式是用autostitch協助計算。

Cylindrical projection

DigiVFX



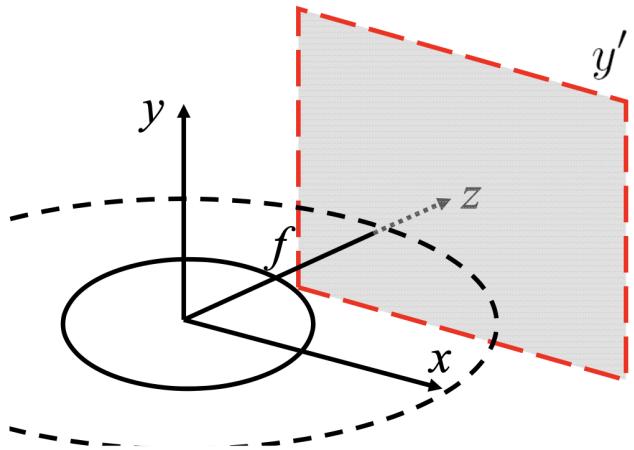
unwrapped cylinder



$$x' = s\theta = s \tan^{-1} \frac{x}{f}$$

$$y' = sh = s \frac{y}{\sqrt{x^2 + f^2}}$$

$s=f$ gives less distortion



投影過後的相片如下方所示。



Feature detection

Feature detection 是全景影像縫合的第一步，目的是從圖片中找出具有代表性的特徵點，以便進一步進行 feature matching。在這個部分，我們使用了 Harris Corner Detector，它是一種經典的feature detector，用於檢測圖片中的角點。具體步驟如下：

1. 計算圖像的 x 和 y 方向的梯度，可以使用 `np.gradient` 來計算。
2. 計算 Ix^2 、 Iy^2 和 Ixy 三個矩陣，這些矩陣是梯度的乘積，用於計算圖片每個像素的 Harris matrix M。 Ix^2 和 Iy^2 代表每個像素在 x 和 y 方向的梯度的平方， Ixy 則代表 x 和 y 方向梯度的乘積。可以使用 numpy 中的 `multiply` 函數來實現。
3. 使用高斯加權函數對 Ix^2 、 Iy^2 和 Ixy 三個矩陣進行平滑處理。
4. 計算每個像素的 Harris response，公式為： $R = \det(M) - k * (\text{trace}(M))^2$ ，其中 M 是 Harris Matrix， $\det(M)$ 和 $\text{trace}(M)$ 分別代表矩陣 M 的行列式和迹，k 是一個常數，通常取值為 0.04~0.06。

5. 根據一定的 Threshold (這邊使用harris response 的平均值當作 threshold value) 對 Harris response 進行 Non-maximum suppression，得到圖像中的角點，這些角點就是特徵點。

Feature description

完成 feature detection 後，我們使用SIFT的 feature descriptor 描述特徵點。

SIFT 的特徵描述步驟如下：

1. 我們先使用sobel kernel得出x與y方向的梯度。此處選擇使用sobel kernel計算梯度，是因為我們認為sobel kernel得出的梯度較不受雜訊干擾。
2. 計算每個點的梯度大小及方向。
3. 接著計算上階段Harris corner偵測到的特徵點的major orientation。Major orientation則是透過將360度切成36個bins，取出特徵點周圍 11×11 的範圍經過 Gaussian kernel加權進行投票選出。
4. 根據特徵點的major orientation，將圖片旋轉至orientation指向0度處，接著取出特徵點周遭 16×16 的pixel，並分成 4×4 個區域，再把每個區域中的360度角切成8個bins，分別紀錄每個bins的票數作為描述。此處為了節省運算量，旋轉major orientation的方法，我們是利用重新排序bins的方式完成，舉例來說若是major orientation是10度，則我們就將第一個bin挪到最後一個bin的後面，由第二bin開始，則就達到旋轉的效果了。
5. 蒐集完一共128個參數後，最後進行 clip 以及 normalize。

Feature matching

我們計算 A 圖的一個特徵點描述與B圖的所有特徵點描述的 L2 距離，並排序，最後取出距離最近的兩個配對。若是最近距離的配對，沒有小於第二近距離的 0.8 倍，則我們認為此結果不夠具代表性，因此此配對不取用。反之若此條件成立，則留下第一名的配對。

Image matching

在找到兩張圖片的對應點之後，我們透過計算 homography 來得到兩張圖片中特徵點之間的轉換關係。在這邊是使用計算 $Ah = b$ 的 least square solution。

另外在 matching 的過程中，會有一些 match 錯誤的點，我們需要把錯誤的點篩選出來並去除以免對結果產生影響。這個部分我們使用 RANSAC algorithm 來完成。流程

如下：

1. 從所有 match 中隨機選擇 4 個 match。
2. 確認這 4 個 match 是否可以組成一個四邊形。
3. 計算出這 4 個 match 對應的 Homography 矩陣 H。
4. 對於所有的 match，使用 H 將左側的特徵點座標進行投影變換，得到右側的投影後的點。
5. 計算左右對應的特徵點之間的距離誤差，如果距離小於 threshold，就將其視為 inlier，否則視為 outlier。
6. 如果目前的 inlier 數量大於 best_inliers，或者 inlier 數量相同但總誤差更小，則更新 best_inliers、best_H 和 best_mask。
7. 最後回傳 best homography matrix。

而針對相機中心不變的資料集，我們則透過另一種方式來完成 matching 的實作，流程如下：

1. 先對於原本的多張照片進行圓柱體座標投影。
2. 使用投影後的照片進行 scaling + translation matrix 的計算。
3. 一樣透過 RANSAC algorithm 篩選錯誤的點並計算出最好的 translation and scaling matrix。
4. 透過計算出來的 matrix 進行後續的拼接照片。

Blending

當我們要將一張圖片貼到另一張圖片上時，我們需要將原始圖片做 backward warping 以將其對齊。這種方法需要計算 Homography 的 inverse matrix 並將其乘以目標圖片中的每個點的坐標得到原始圖片中對應的坐標。這些坐標需要做 normalized 以便於計算，並在原始圖片邊界外的區域使用 mask 以避免生成缺陷。接著在原始圖片上應用 Bilinear interpolation 以獲得目標圖片上每個點的 pixel value，最後將這些 pixel value 分配給目標圖片中的對應點，兩張重疊的部分則保留最大值。

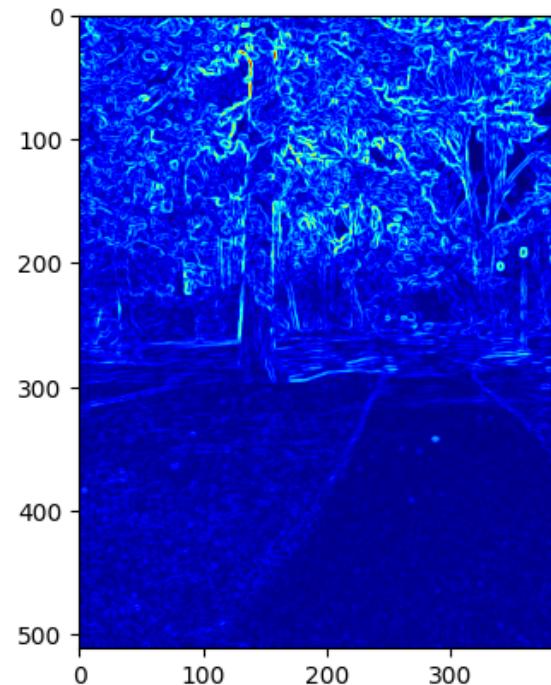
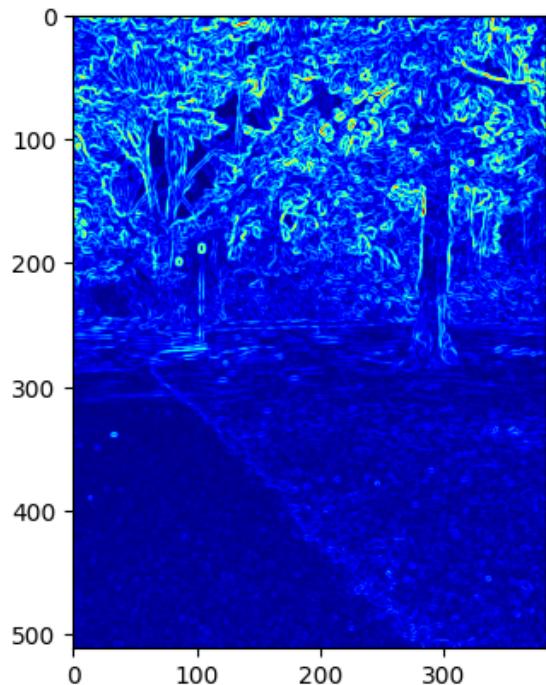
全景圖的拼接則是透過事先設定好全景圖的 height 和 width，height 是由所有圖片中取 maximum，width 則是所有圖片的 width 總和。

Results and Discussion

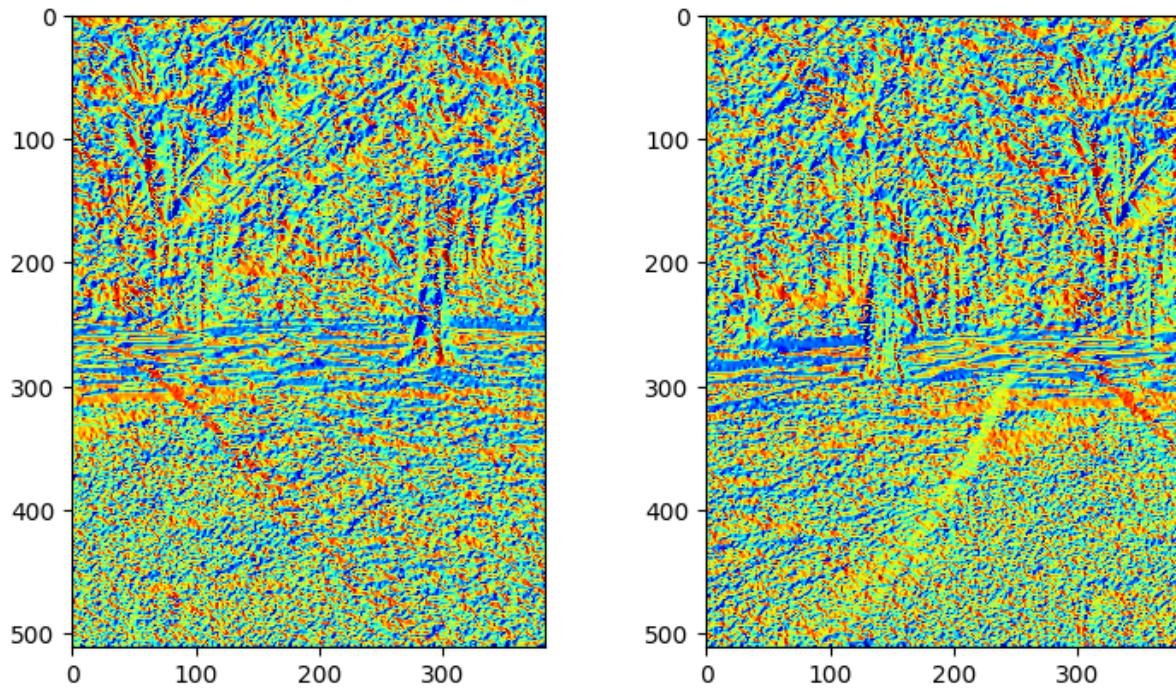
feature description

由下列可以看出同樣的場景，其梯度magnitude大小在edge處都相當一致，此外edge處的orientation一致性也較高。

Gradient magnitude

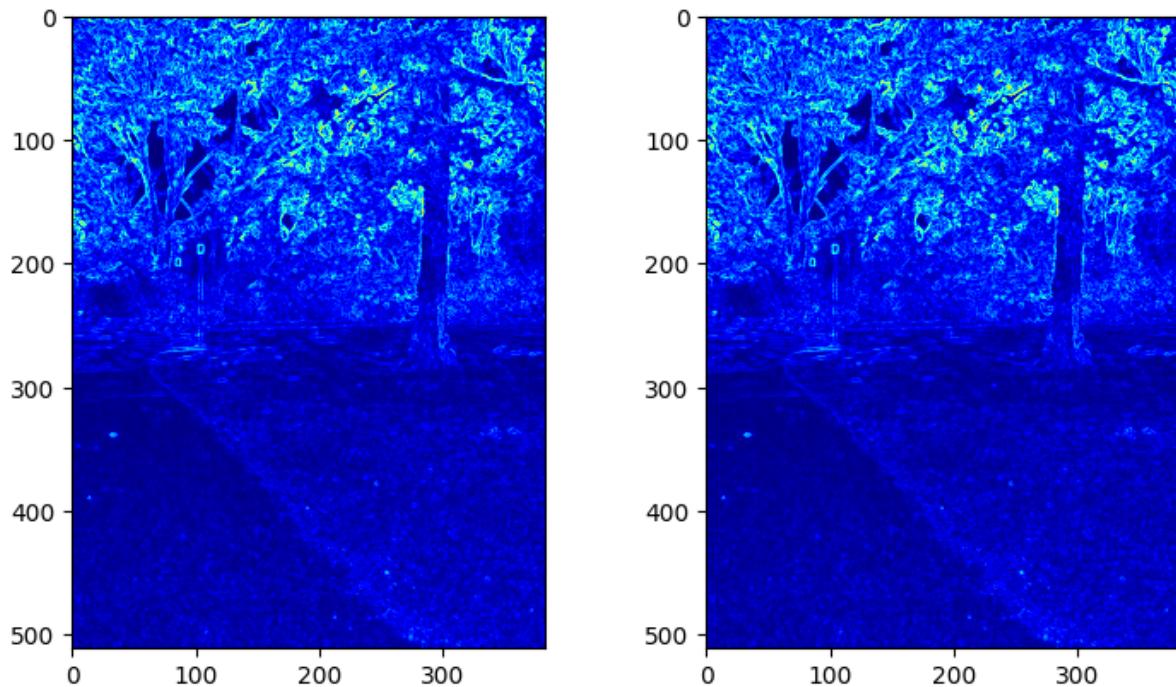


Gradient orientation



此外我們也比較使用一般的 gradient 而非 sobel kernel 的結果。可以發現，其邊緣處受到雜訊的結果影響較大，其對應的 orientation 也會較為紊亂難以描述，容易使得 feature descriptor 失敗。

Gradient magnitude



Gradient orientation

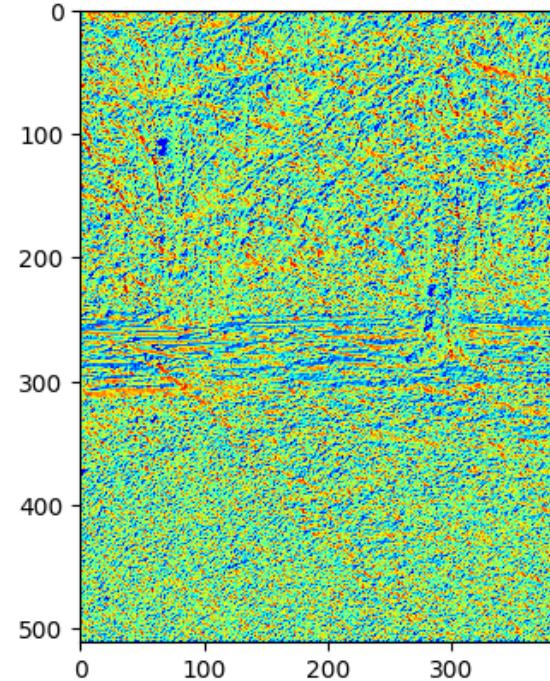
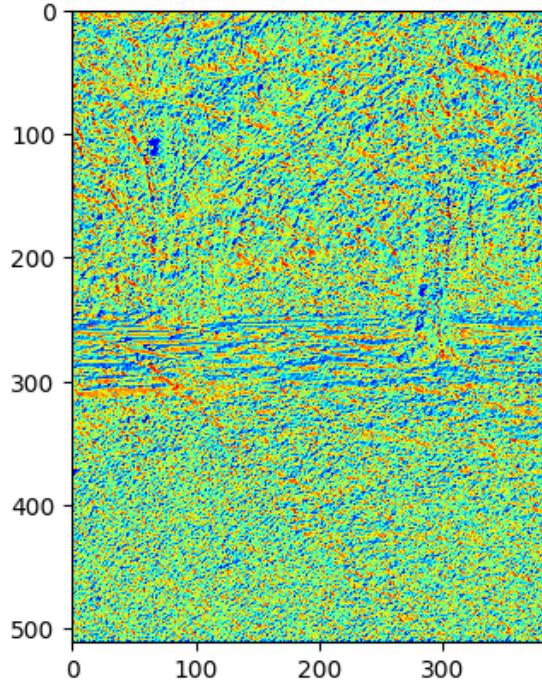


Image Matching

最一開始時，我們使用最 powerful 的 homography 預測座標轉換矩陣，然而發現對於相機位置不變的資料集來說，homography 過於厲害，容易過度擬合，造成最後全景扭曲。因此最後我們改成僅模擬縮放與位移。結果好很多。

Homography



Scale + translation



反之對於相機中心有些許移動的圖像來說，僅模擬縮放與位移又不夠，會導致影像和的不準，最後造成鬼影的產生。反而用 homography 效果會更好。

Scale + Translation

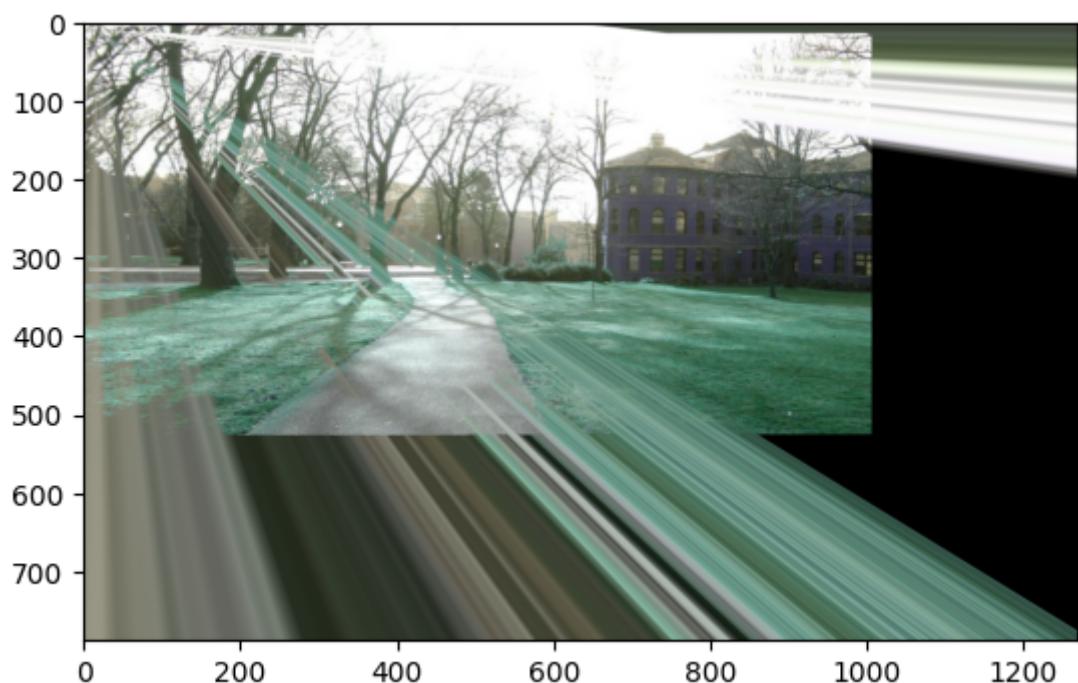


Homography



Blending

我們一開始使用的方法是先將兩張圖片 warping 後合併，將形成的圖片進行特徵偵測後，再和下一張圖片合併，但後來我們發現這個問題會造成進行到第四或第五張照片的合成時，程序就會失敗。我們推測主因是 warping 後會使得 gradient 的方向被扭曲，由於 homography 無法保存直線，使得gradient方向也會受影響，因此特徵間的描述就算受影響，使得 match 數量大量減少，使用不夠充足的 match 估算 homography 就會使得圖片破裂無法視讀，甚至程式崩潰。下圖中線條狀的 artifact 即是 homogrphy 失敗的結果，後方則是前四張圖片拼接的結果，原本預期第五張可以和前四張拼接成功。



因此後來我們採用圖片仍與相鄰的下一張配對即可。於拼接時，再與之前的已經累積起來的座標轉換式互乘，使得可以直接接於已經拼接完的影像後。最後使用這個方法可以順利完成全景圖的拼接。

Conclusion

本次專案我們成功實作 feature detection (Harris corner detector), feature description (SIFT descriptor), feature matching 和 image blending 等步驟將多張照片縫合成全景照片，並且我們也針對相機位置是否變化進行不同的縫合方式，以達到更好的全景圖縫合效果。

Reference

- [1] Brown and Lowe, "Recognising panoramas," Proceedings Ninth IEEE International Conference on Computer Vision, Nice, France, 2003, pp. 1218-1225 vol.2, doi: 10.1109/ICCV.2003.1238630.
- [2] C. Harris and M. Stephens, "A combined corner and edge detector," in Proc. 4th Alvey Vis. Conf., 1988, pp. 147–151.