DIPCV Assignment V
Homograph Estimation and Correspondence

Student: Chung-Chia Cheng (鄭中嘉)

Student ID: L46104020

Email: l46104020@gs.ncku.edu.tw

Space Weather Lab, Department of Earth Science, National Cheng Kung University, Taiwan

***Abstract* – In this report, we will use SIFT to find the key points for provided images. Then we will find match points based on the key points. And finally, we will project images to target plane by using homograph matrix. As for source code, please check [1].**

Index Terms – SIFT, Homograph Estimation, RANSAC

I. Introduction

In this assignment, we are provided the following 4 pictures, including 3 pictures of different books and 1 picture that has 3 of the books inside (see Figure 1). Our goal is to find match points and warp the pictures of books to the target plane.

Fig 1. Provided pictures

This report is designed as follows, in section II, we will show how SIFT will help us to find the key points of the provided image; in section III, match points will be used to plot the correspondence results. In section IV, RANSAC will be used to find best match points and we will see how to warp the images of book to the target domain by using those best match points.

II. SIFT

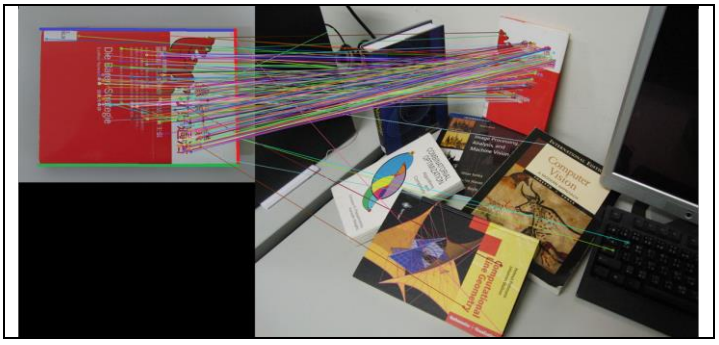
A. *Figures and Tables*

SIFT is used for finding the key points from a given image. With different external conditions (ex: strength of light, rotation), this method can still find the same key points. Therefore, it's a good choice when it comes to the topic like finding match points or image stitching. Figure 2 shows the corresponding result after applying SIFT. There are 969, 1650, 1010, and 3295 key points found in each picture respectively.

Fig 2. Result after applying SIFT

C. *Correspondence*

Once we derived key points from SIFT for each image, we can use the key points from a given image of book and try to match the key points onto the key points from the scene image. We've mentioned, when SIFT consider whether the candidate is key point of not, it will ignore the external conditions like strength of light and effect of rotation, so, in theory, the same book shown in different pictures should have similar key points. Figure 3 shows how key points are the same in two different pictures.



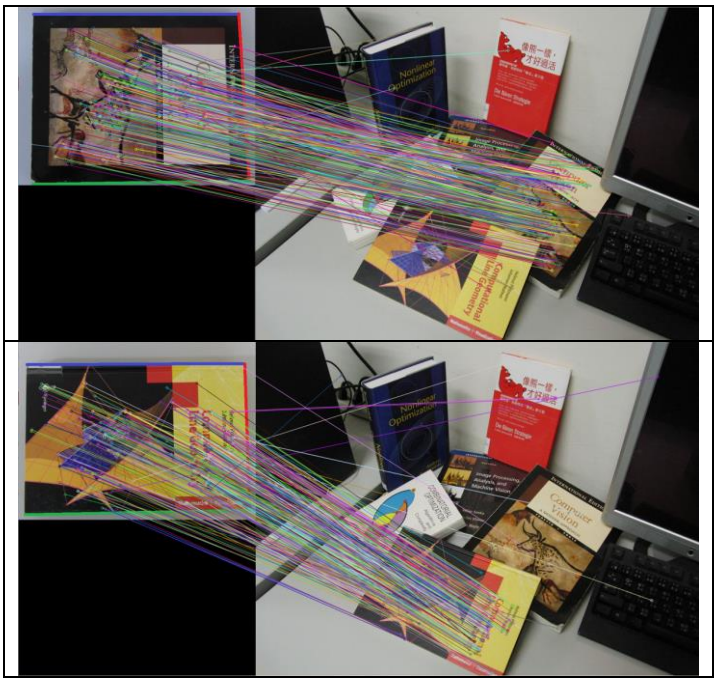


Fig 3. Correspondence of two different pictures

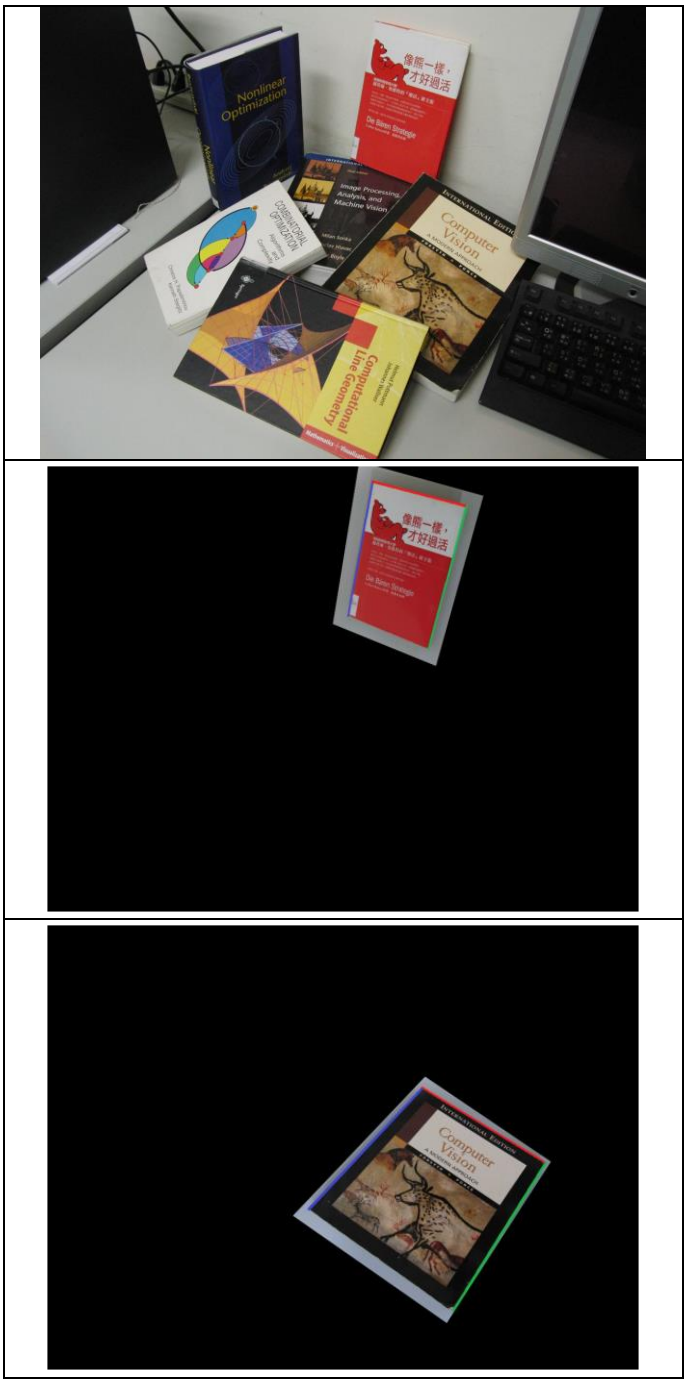
It's clear that most key points on the books are correctly matched to the scene image.

D. RANSAC and Warp

RANSAC stands for "Random sample consensus". It is an iterative method for estimating a mathematical model from a data set that contains outliers. The outliers we are talking about is the so-called "good matches". Here, when performing RANSAC, we need to first pick 4 matches randomly. Then, see these 8 matches performs well (by determining the number of outliers). Repeat the procedures mentioned above until it finds the best 4 matches.

The reason that we need to find 4 matches instead of other numbers is because our goal is to warp one image to the other domain. To do so, we need to find the "homograph matrix". We need 4 matches to form 8 equations that solve this problem.

Once finding the homograph matrix, we can warp the image of book onto the plane that exist in the scene image (see Figure 4). As can be seen, the warped images indeed lie on the proper location.



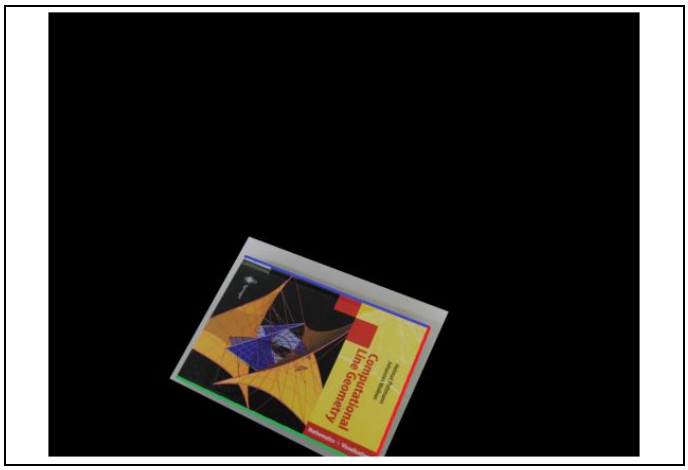


Fig 4. Warped results

ACKNOWLEDGMENT

When performing SIFT by using OpenCV, I found [2] and [3] are helpful. When trying to find matches of key points, I found [4], [5], [6] and [7] are helpful.

REFERENCES

[1] Source Code: https://github.com/Ratherman/Computer-Vision/blob/main/HW5_Python/main.ipynb
[2] SIFT: https://www.youtube.com/watch?v=USl5BHFq2H4
[3] SIFT: https://www.youtube.com/watch?v=nnH55-zD38I
[4] Computing Homography: https://www.youtube.com/watch?v=l_qjO4cM74o
[5] 3x3 Image Transform https://www.youtube.com/watch?v=B8kMB6Hv2eI
[6] How to get pixel coordinates from Feature Matching in OpenCV python https://stackoverflow.com/questions/30716610/how-to-get-pixel-coordinates-from-feature-matching-in-opencv-python
[7] Feature matching and Homography to find Objects https://www.programcreek.com/python/?CodeExample=find+homography