

Project Report 2: CSE534

Xuanyu Dong
xuanyudo
50112003
xuanyudo@buffalo.edu

Abstract

This report includes the implementation of visual search system and some of the test result on paris building dataset. In specific, this report is going to explain the key implementation detail of this visual search system and the performance analysis.

1. Github Link

The implementation of this visual search system can be found on github:
https://github.com/xuanyudo/Visual_Search_System

2. Implementation

2.1. Key point extraction

The first step is performing sift to detect key point from the image along with the descriptor of the image for further key point matching. Below is one sample of extracted key point image.

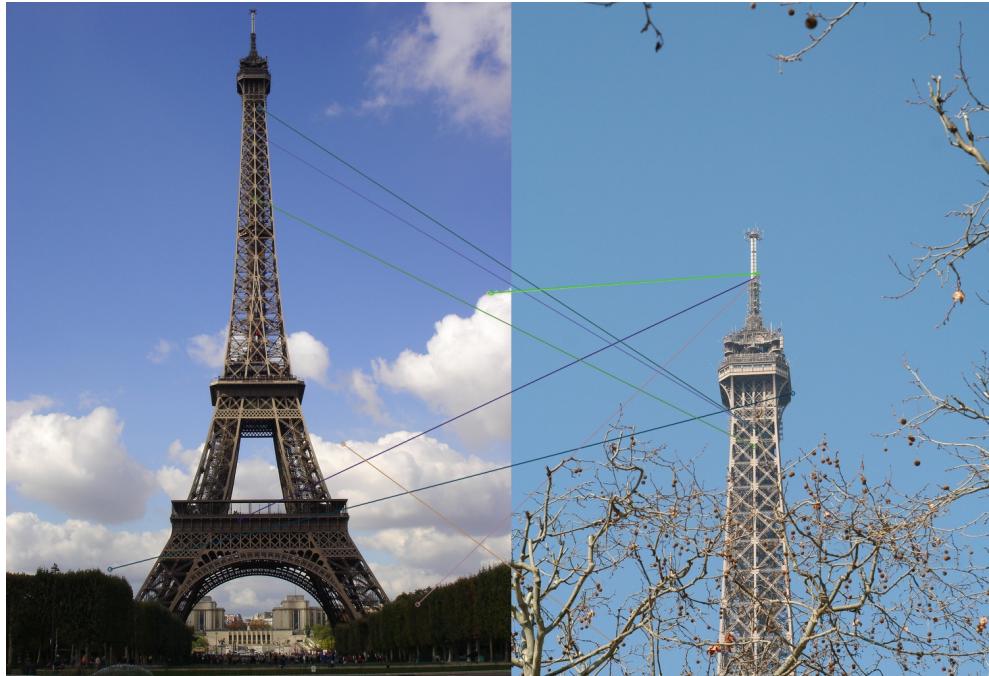




2.2. Search for pairs of match

Comparing the descriptor of two image and use k nearest neighbor algorithm to connect point to points and form different pair.

However, this sometime is not enough for finding a good match of the image as example below.



As shown above, sometime we will get lots of outlier that can confuse our match. Therefore, I decide to find homography between two image with those pairs of key points we found before. With this step, I can filter out the outlier and keep the inlier.

Lastly, I set a threshold of 10 to decide whether or not the object is found on a specific image.

2.3. Final Result of Match



Figure 1: Query Image

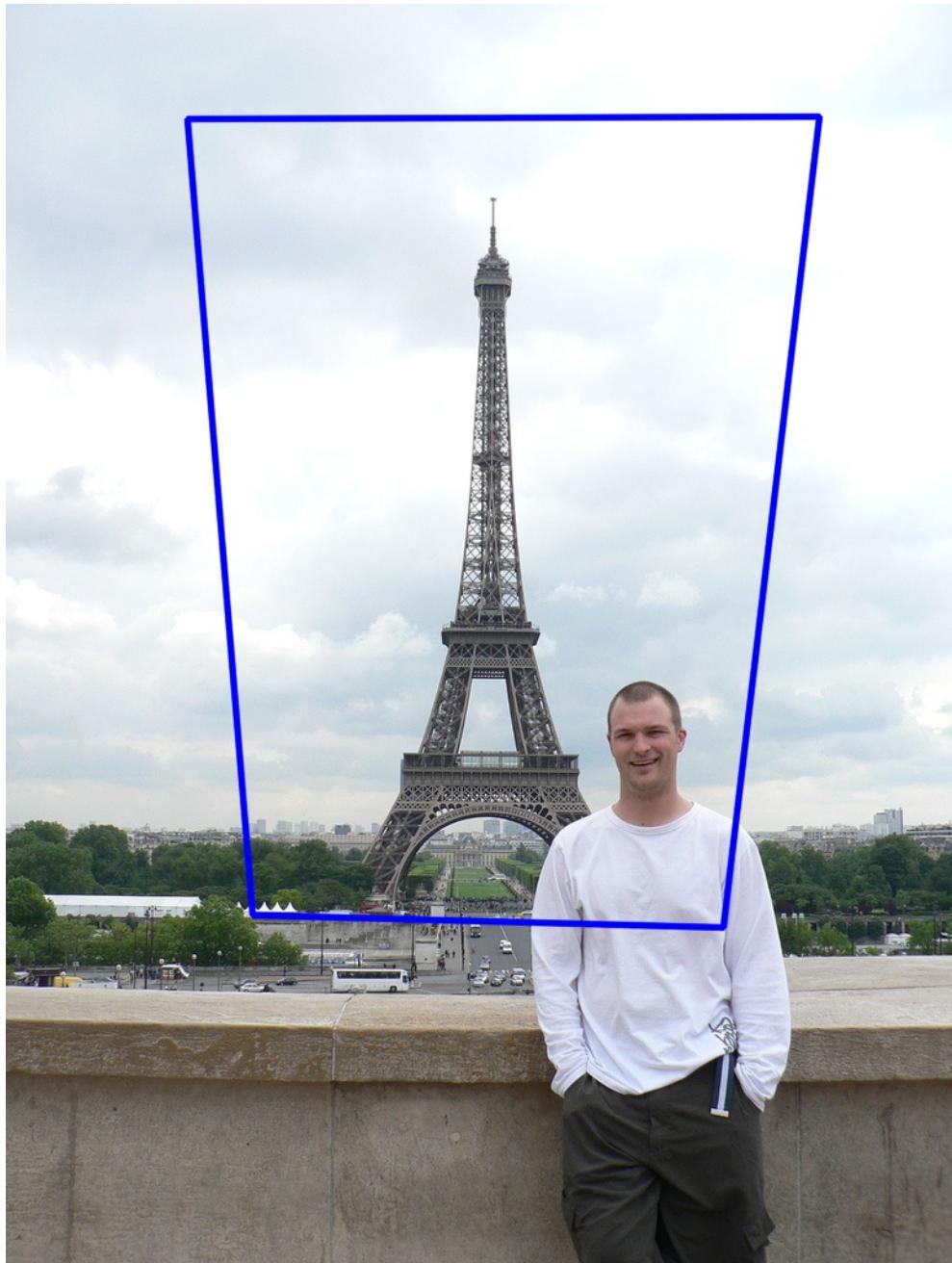


Figure 2: Result one

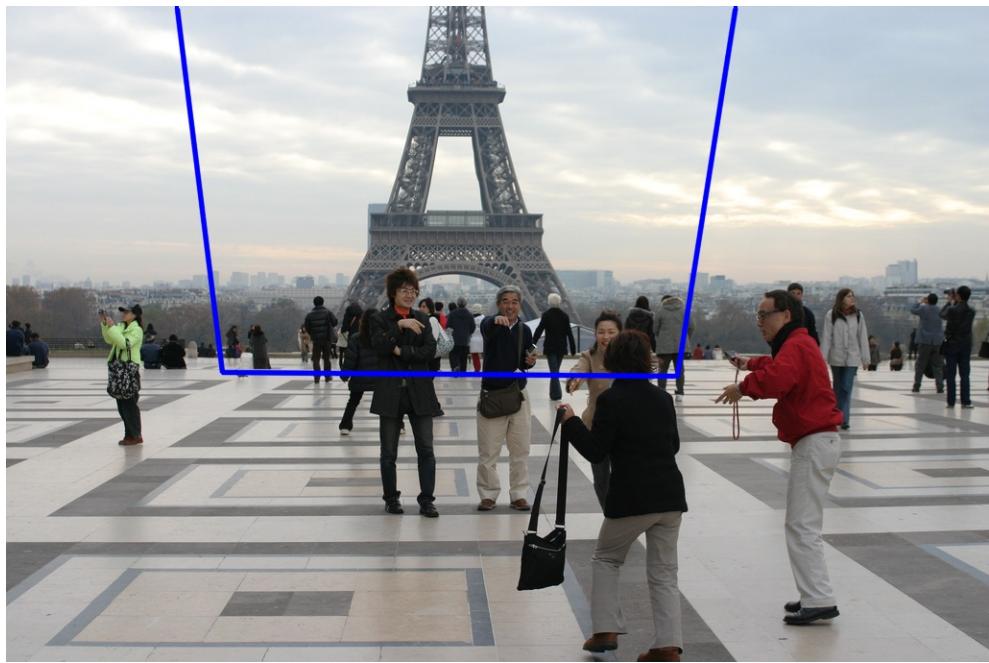


Figure 3: Result one

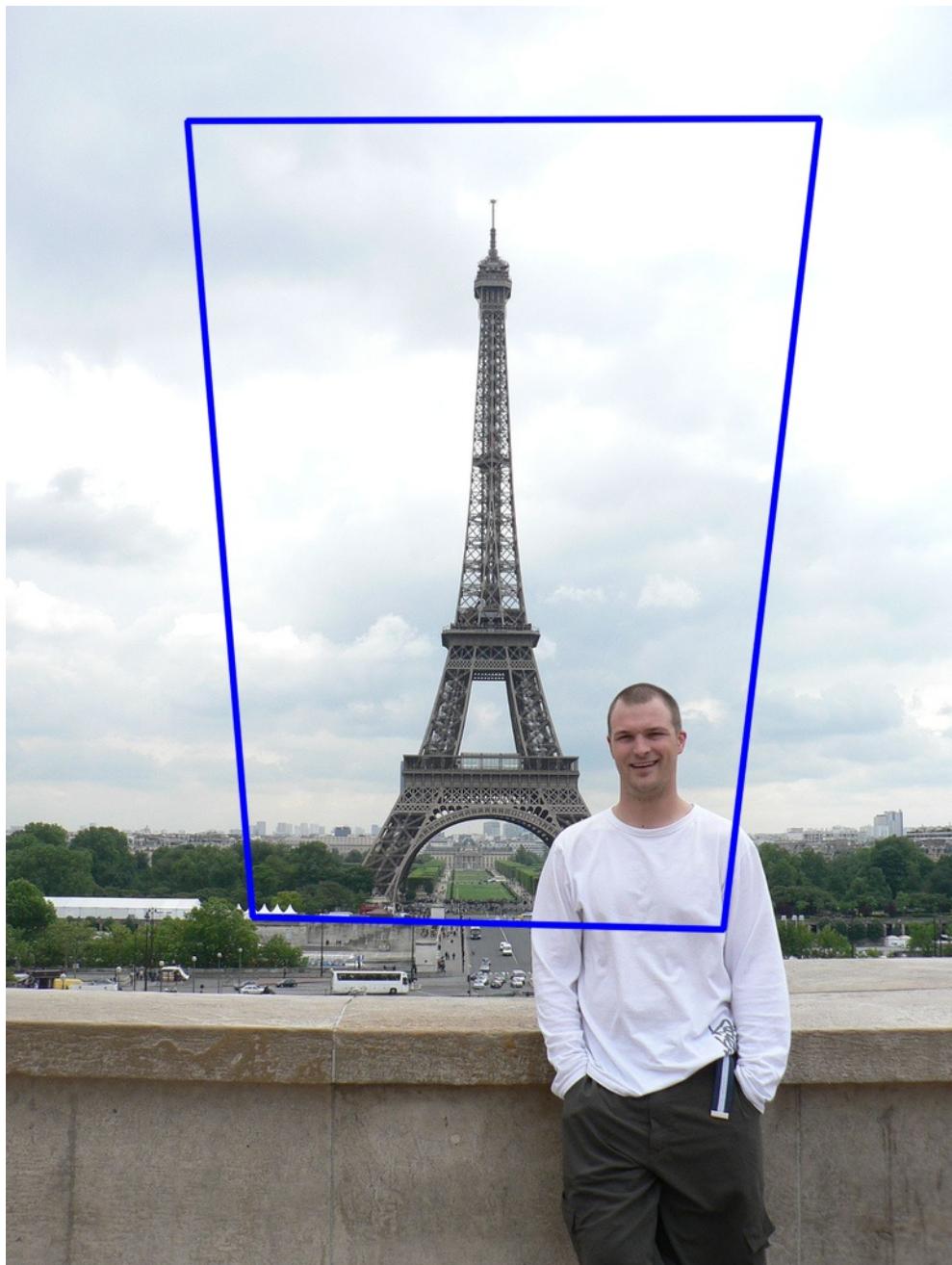


Figure 4: Result one



Figure 5: Result one

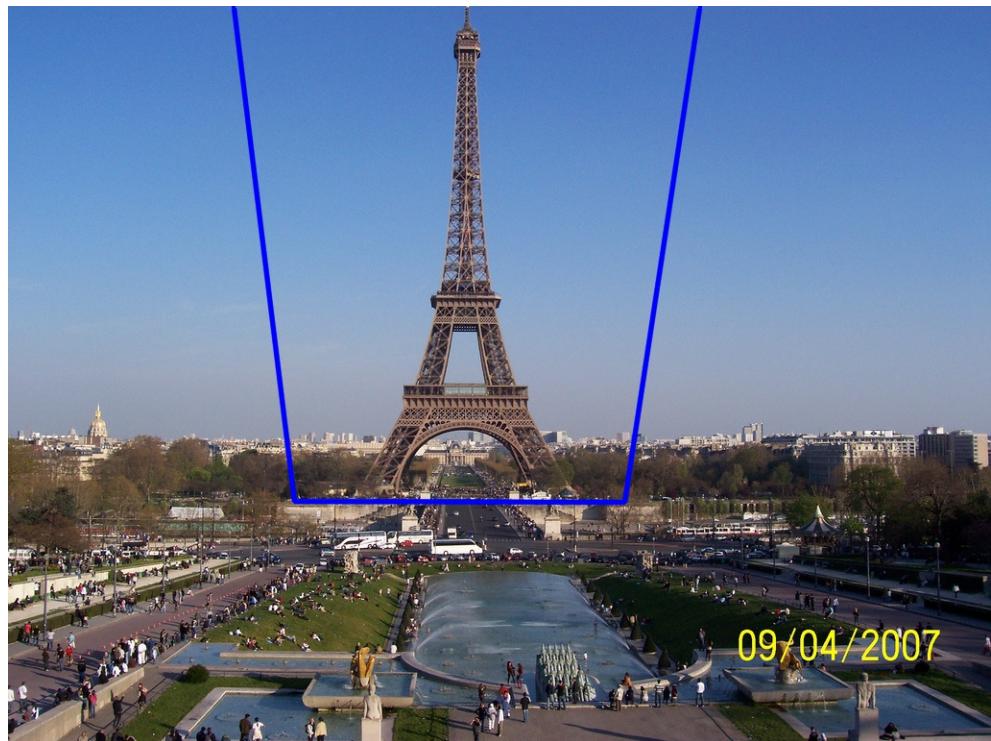


Figure 6: Result one

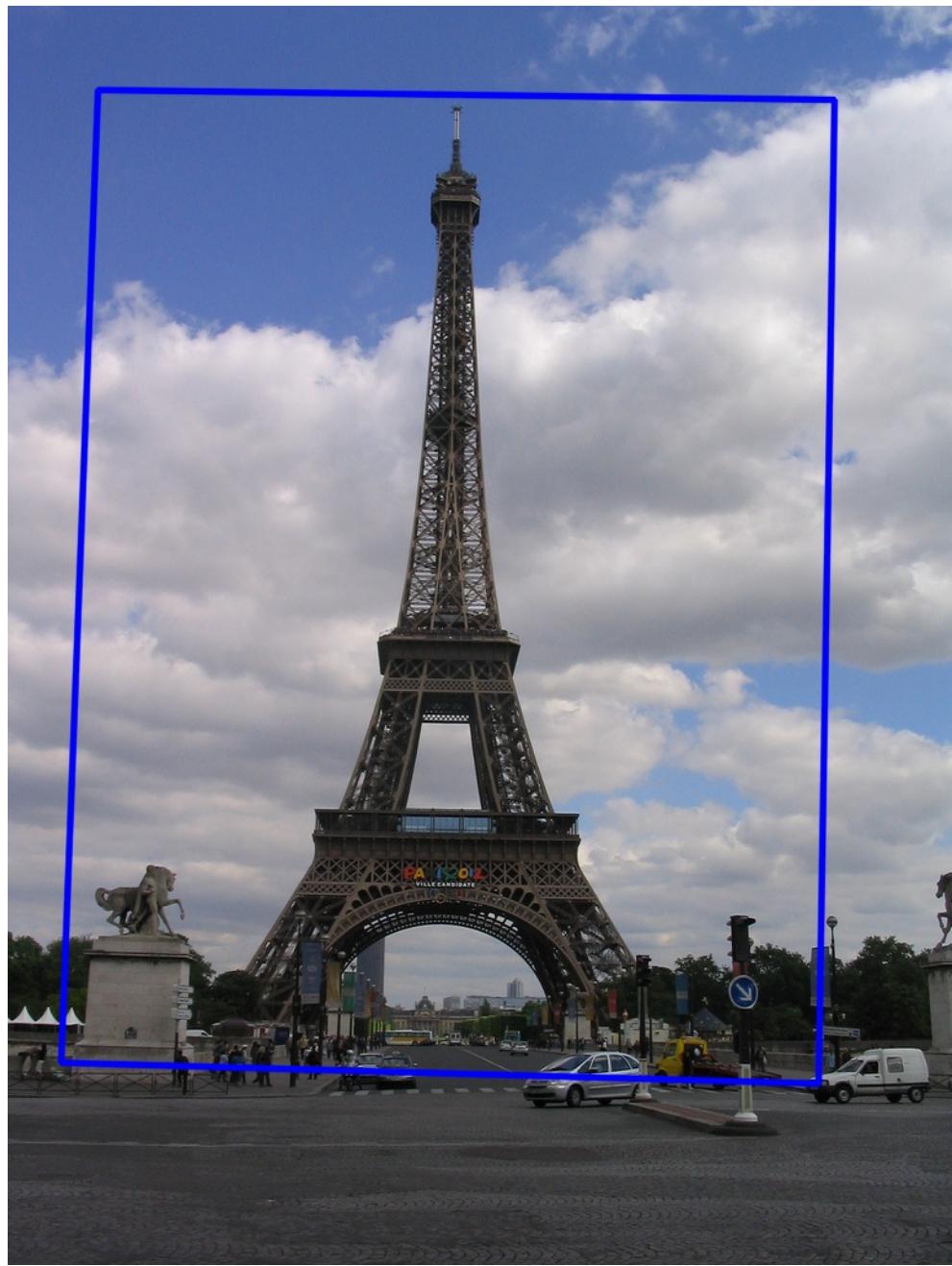


Figure 7: Result one



Figure 8: Result one



Figure 9: Result one



Figure 10: Result one

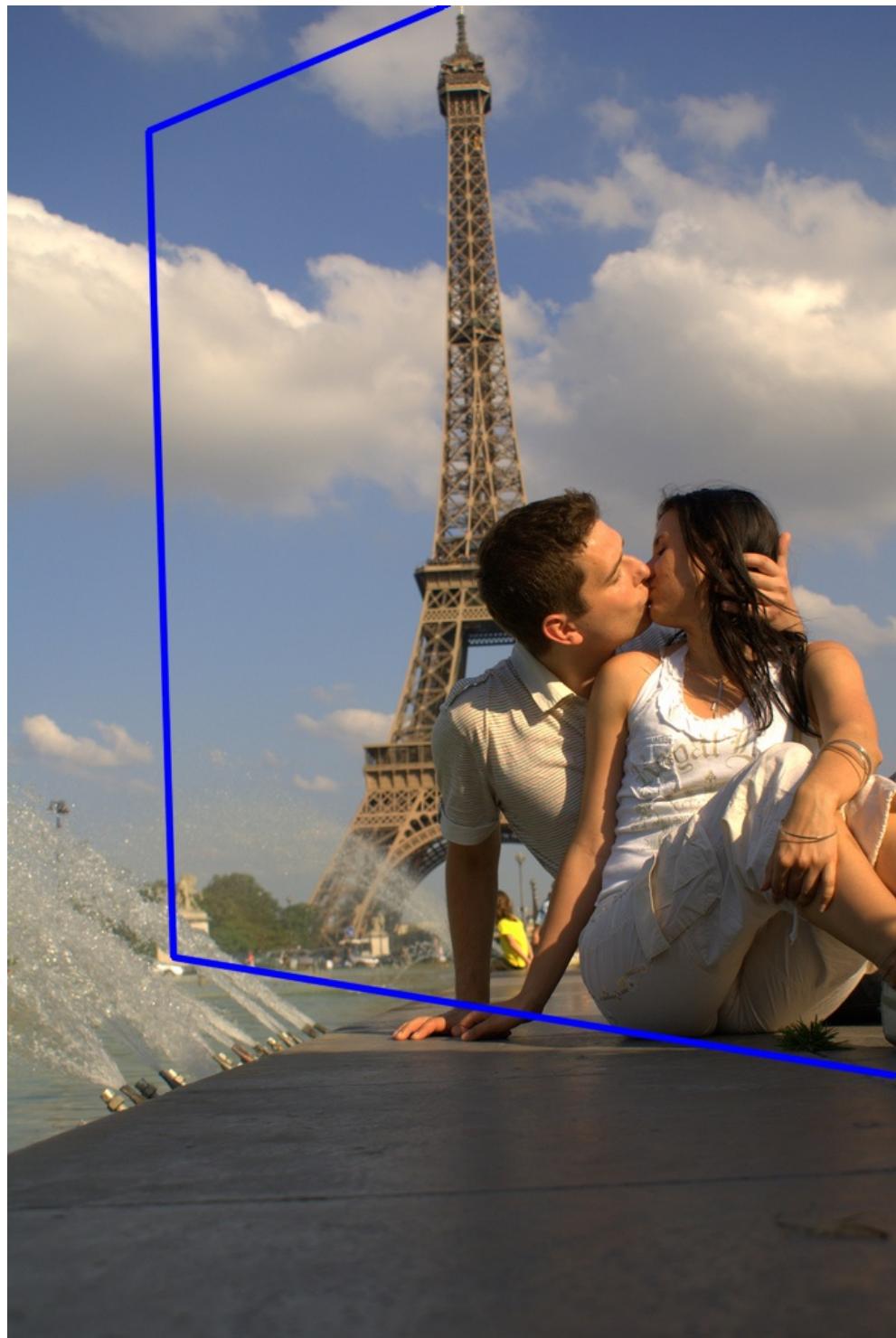


Figure 11
15 Result one

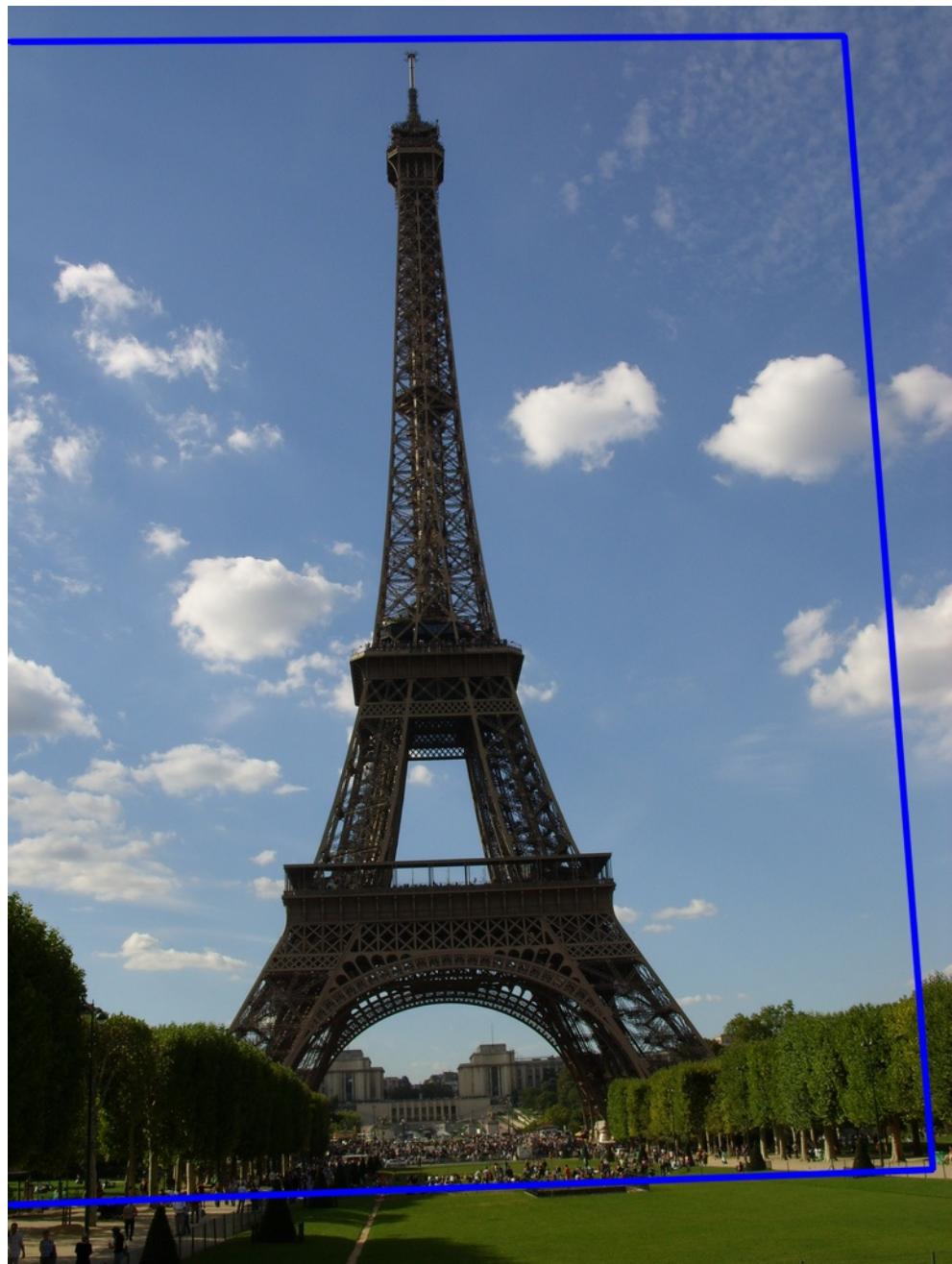


Figure 12: Result one



Figure 13: Result one

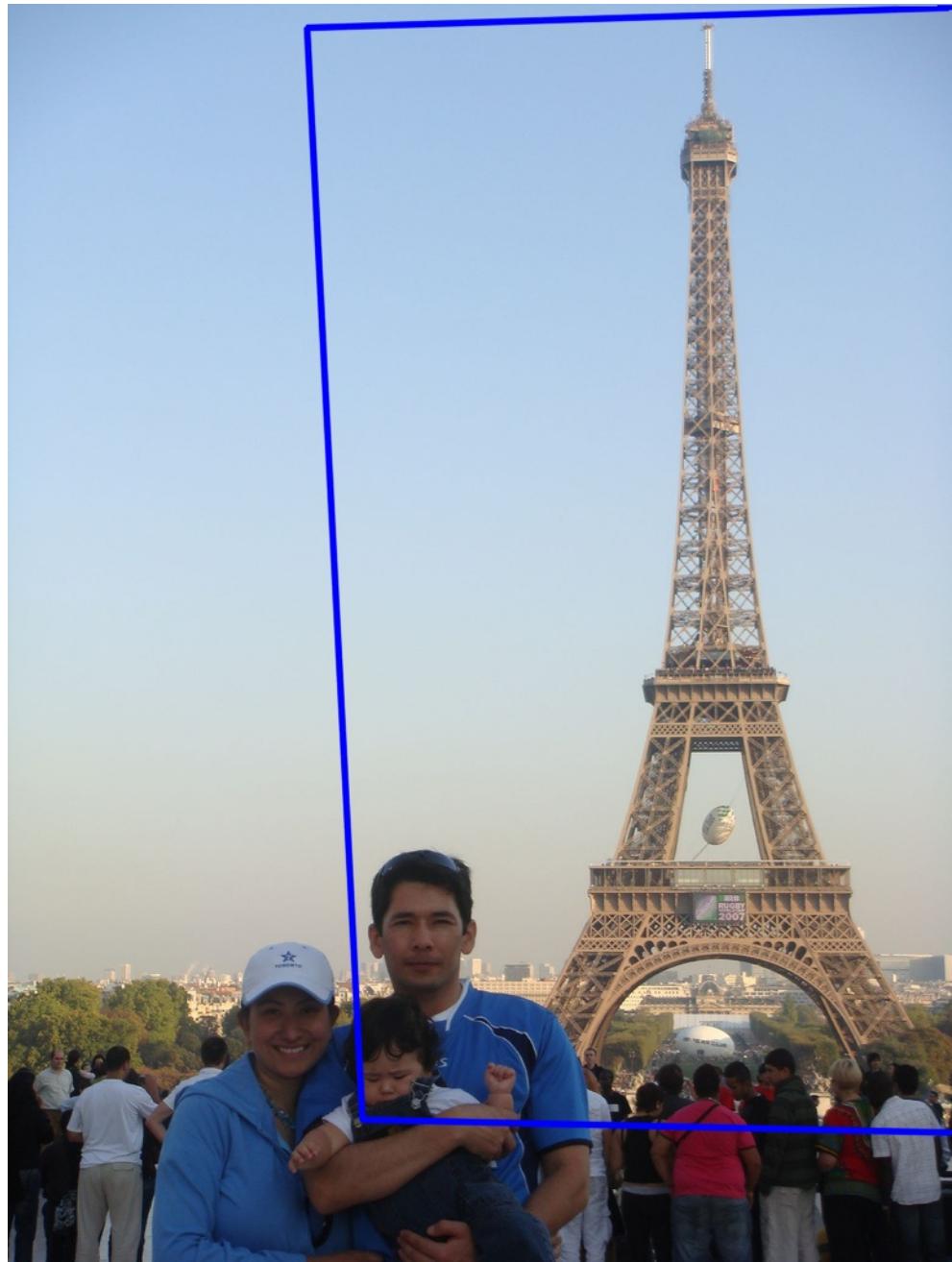


Figure 14: Result one

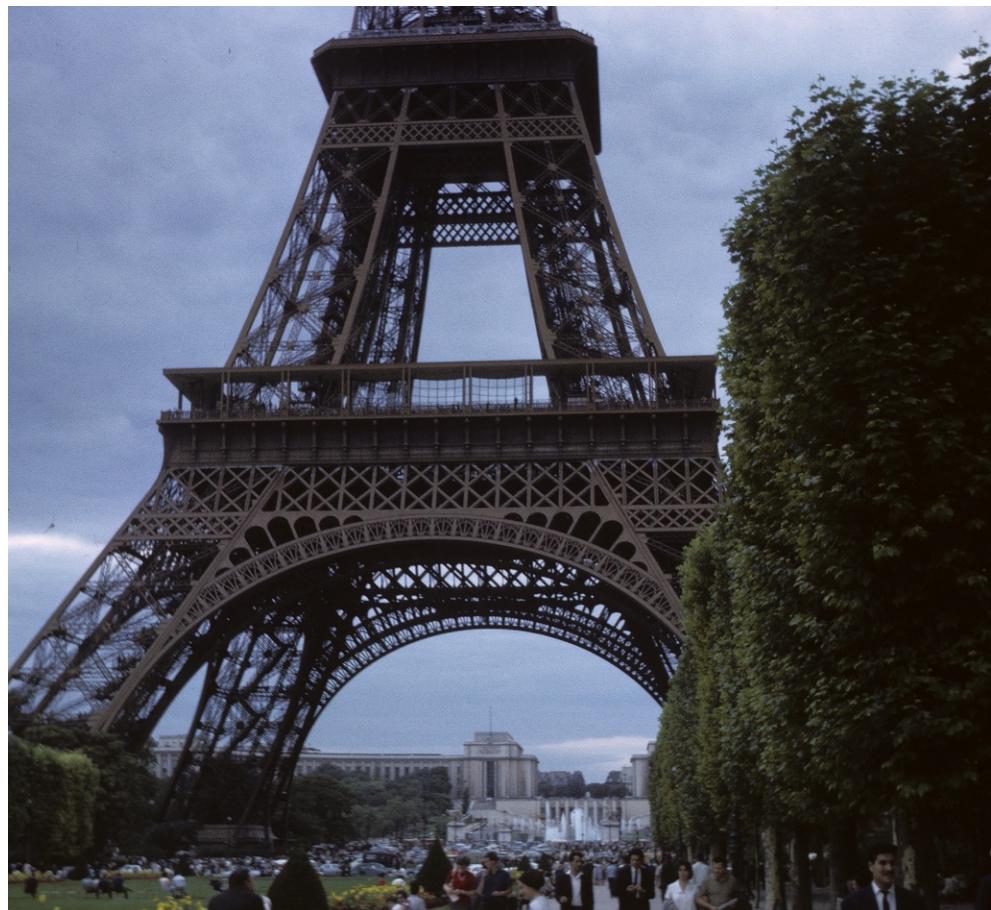


Figure 15: Result one

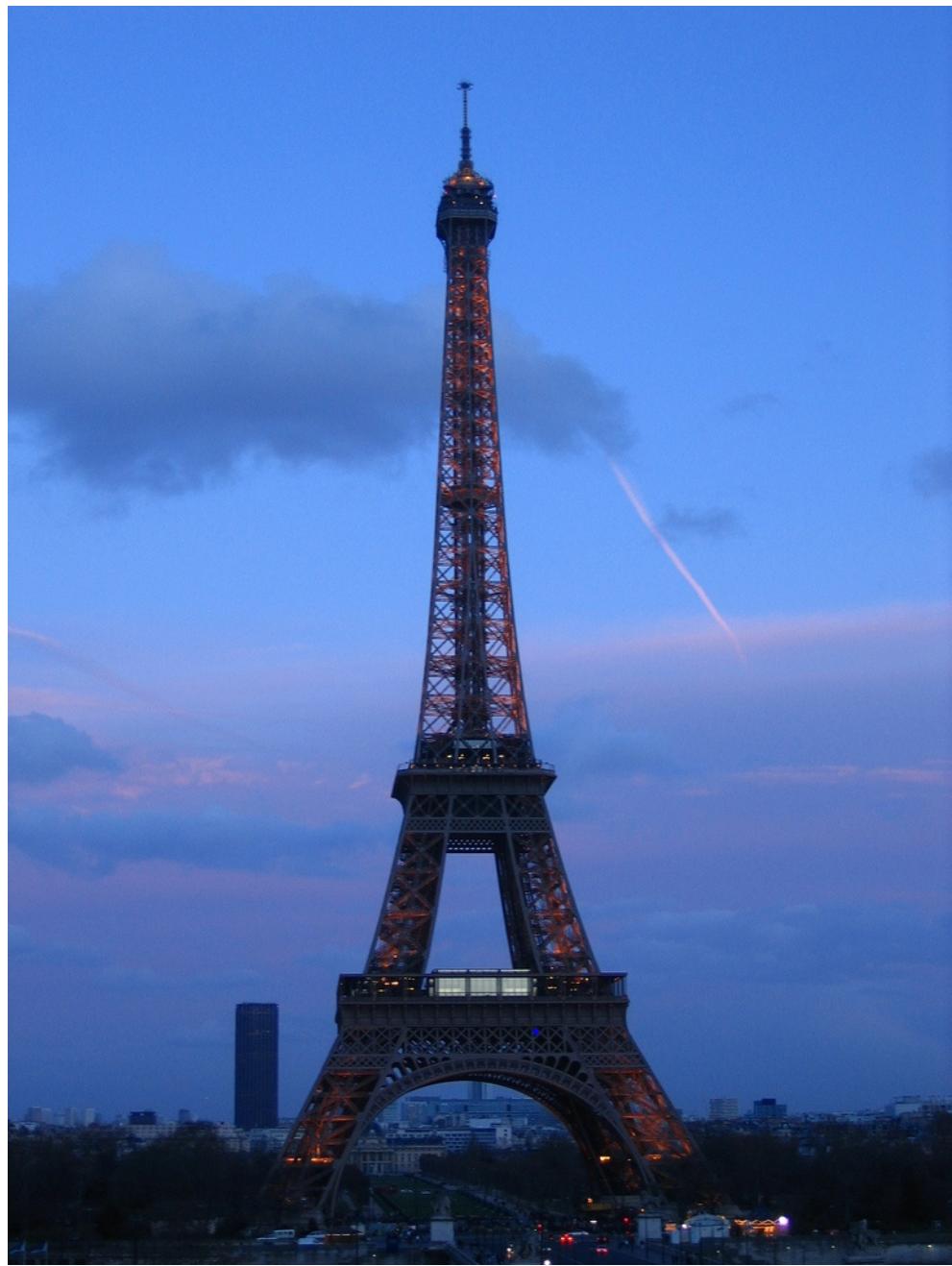


Figure 16: Result one

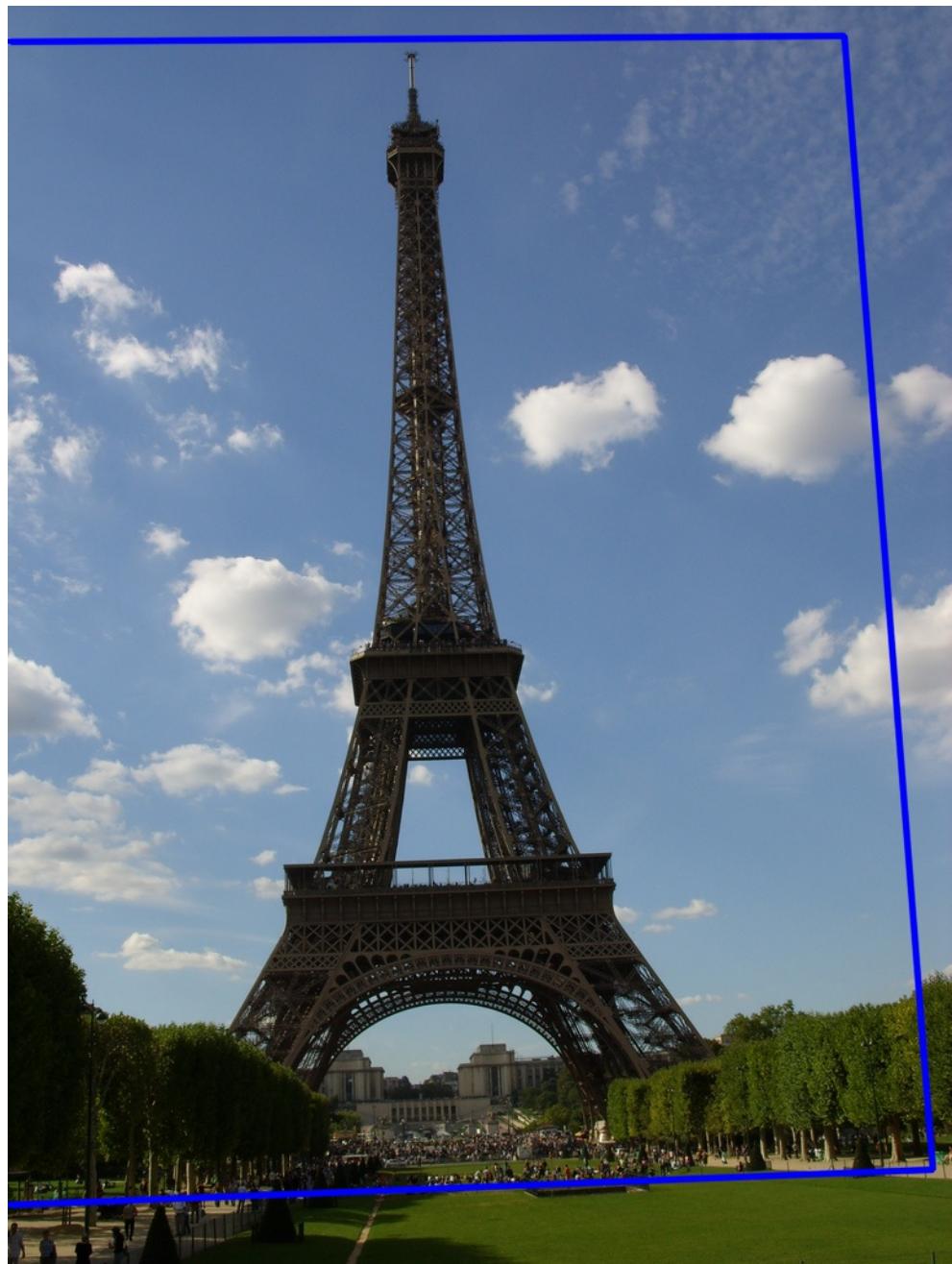


Figure 17: Result one

3. Performance Anaysis

Current program perform extremely well on accuracy of correct object in the database because of the high threshold value(10) in order to decide whether a specific image fall into the true result. In other word, this query image should have at least 10 inlier in between homography of two images.

However, my implementation is doing bad on the speed of query. Since paris building database has lots of image, each image comparison takes about 0.7 second which is really bad for a large image database.

3.1. way to improve performance

As I test the program, I found that the key reason which slows down the process is computing key point and descriptor. This step can be discord in query step if we pre-compute the key point and descriptor for each image.