November 17, 2020

PROJECT 2 REPORT

Eric Miao
CS-7390
Fall 2020
hongyum@smu.edu

1. INTRODUCTION AND MOTIVATION

A feature is a piece of information which is relevant for solving the computational task related to a certain application, especially in computer vision and image processing. Features may be specific structures in the image such as points, edges or objects. Features may also be the result of a general neighborhood operation or feature detection applied to the image. In this project, I implement code for detecting discriminating features in an image and attempt to find the best matching features in another image.

2. PROGRAM OVERVIEW

There are 3 main components of the program:

Feature Detection: Identify the feature point. Feature point is the point at which the direction of the boundary of the object changes abruptly or intersection point between two or more edge segments. In this project, I used Harris Corner Detection algorithm, as well as FAST Corner Detection Algorithm.

Feature Description: A feature descriptor is an algorithm which takes an image and outputs feature descriptors/feature vectors. Feature descriptors encode interesting information into a series of numbers and act as a sort of numerical "fingerprint" that can be used to differentiate one feature from another. In this project, I define my descriptors using the pixel intensity values in a 5x5 neighborhood of the corners detected from Part 1(Using Harris Corner Detector).

Feature Matching: Features matching or generally image matching. Descriptors are compared across the images, to identify similar features. For two images we may get a set of pairs (Xi, Yi) ↔ (Xi`, Yi`), where (Xi, Yi) is a feature in one image and (Xi`, Yi`) its matching feature in the other image. In this project, given 2 vectors of descriptors from two images, I compared each pair of features individually, and calculated a scalar distance between them.

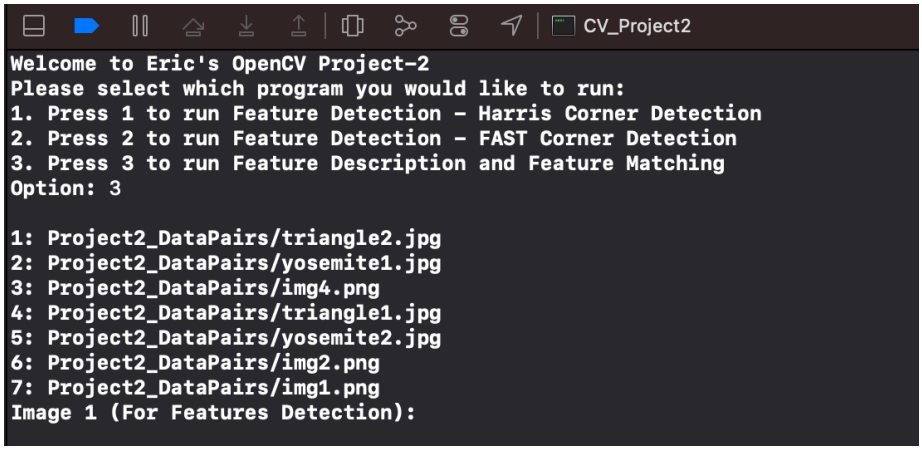
3. IMPLEMENTATION:

My implementation to each component of the program is very straightforward.

A. User Interface

Similar to my Program 1, I designed a friendly user interface for my program. The following screenshot is my user interface. User is allowed to choose between 3 different modes of the program:

1. Mode 1: Run the Feature Detector - Harris Corner Detection
2. Mode 2: Run the Feature Detector - FAST Corner Detection
3. Mode 3: Run the Feature Description and Feature Matching

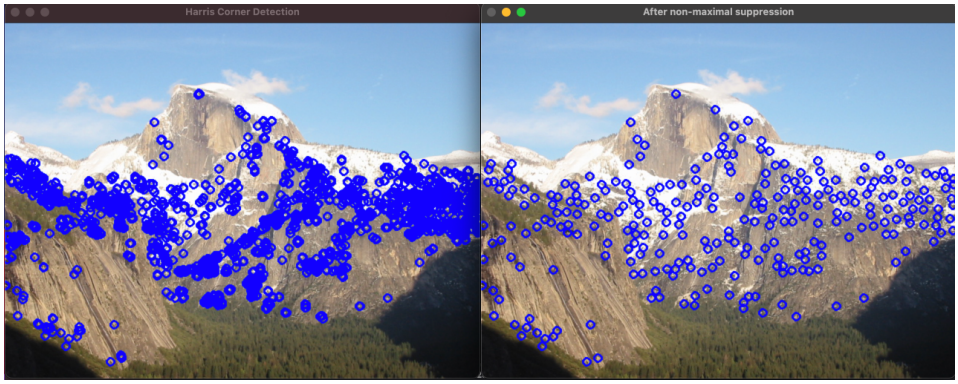


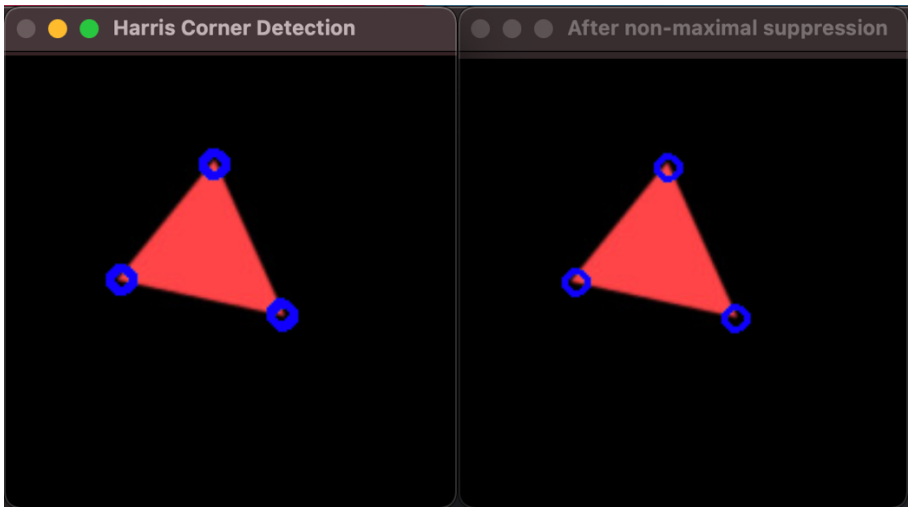
The program also utilizes C++ 14's filesystem function to find and list all the file name under the given directory. User can easily select which image(s) to use without typing any additional input.

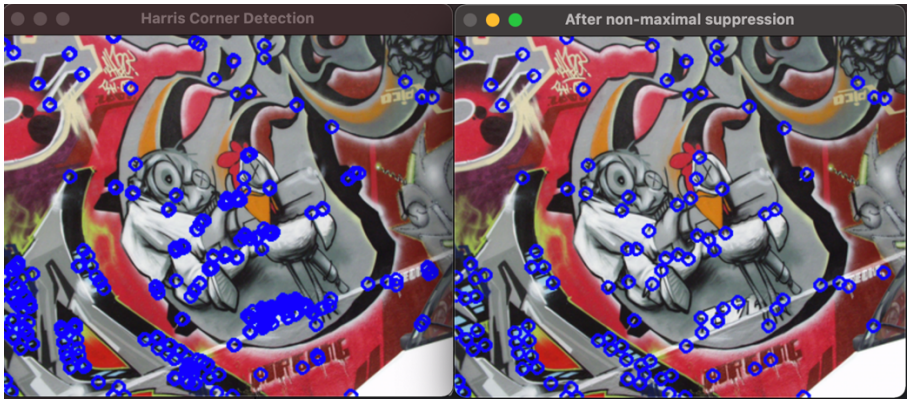
B. Mode 1: Feature Detector - Harris Corner Detection

This mode utilizes the power of Harris Corner Detection algorithm, an algorithm that is commonly used for extracting corners and infer features of an image.

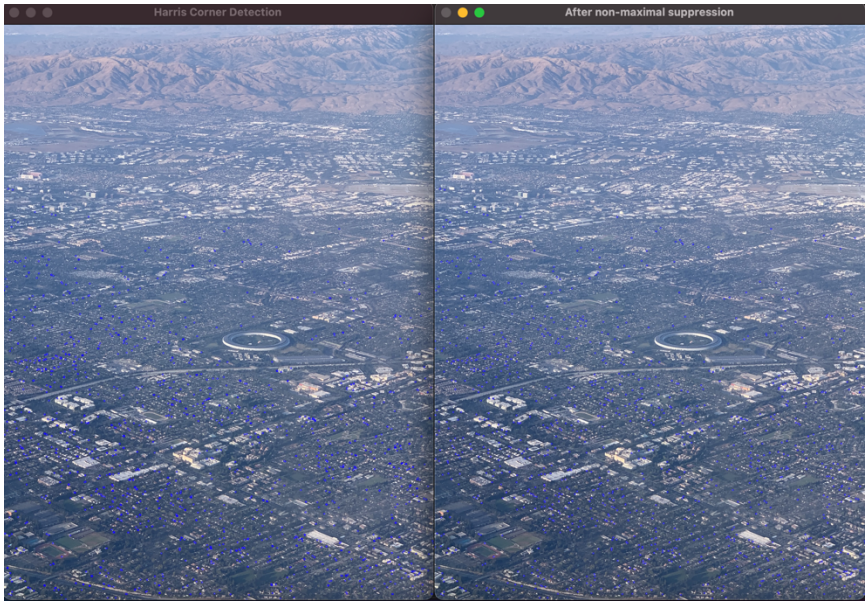
Note: Below examples are produced using different threshold values.





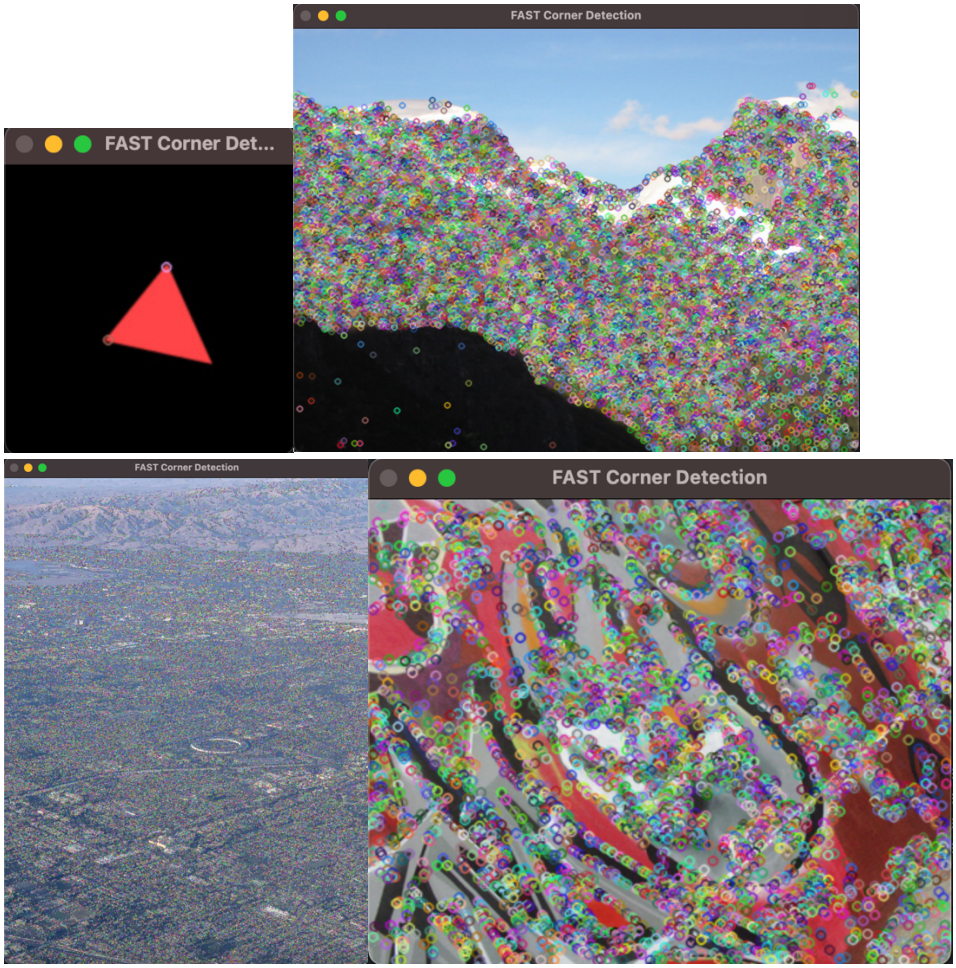


Below are two pictures I took in an airplane when I was traveling to California (Did you see the Apple Park?) All the very small blue dots represent the corner detected.



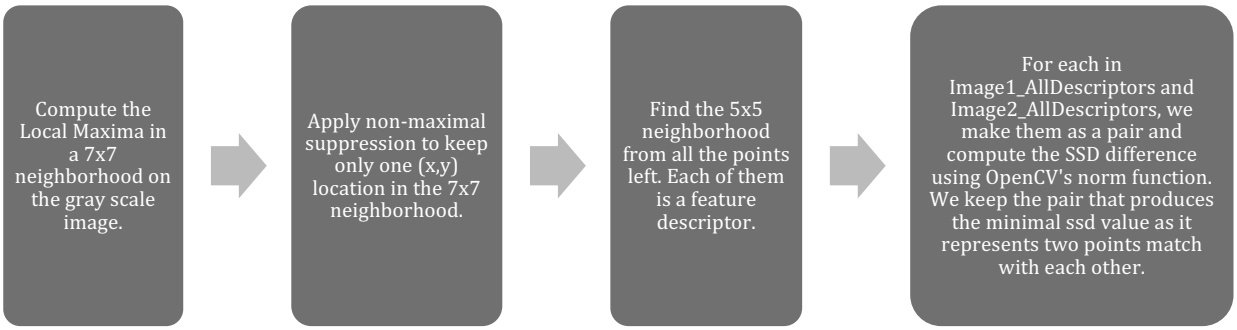
C. Mode 2: Feature Detector - FAST Corner Detection

This mode utilizes the power of FAST Corner Detection algorithm, an algorithm that could not only extracts feature points but also could be used to track and map objects in many computer visions tasks.

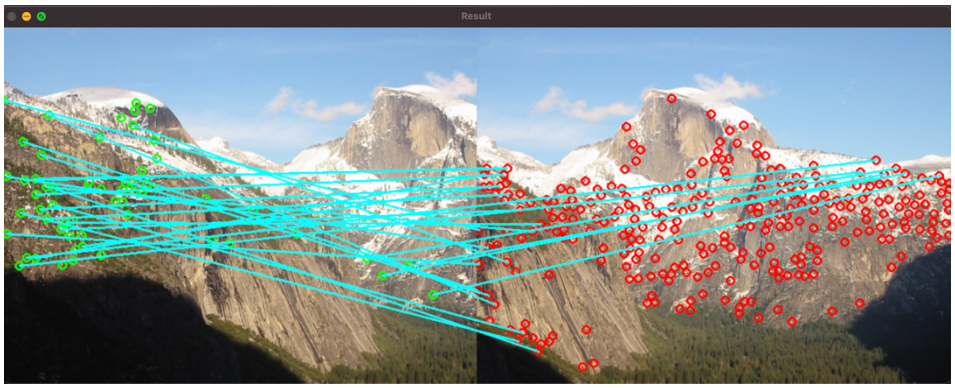


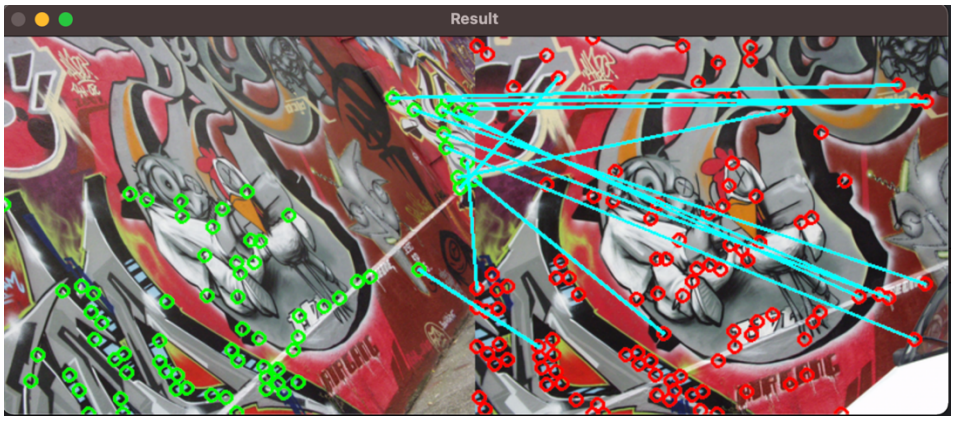
D. Mode 3: Feature Description and Feature Matching

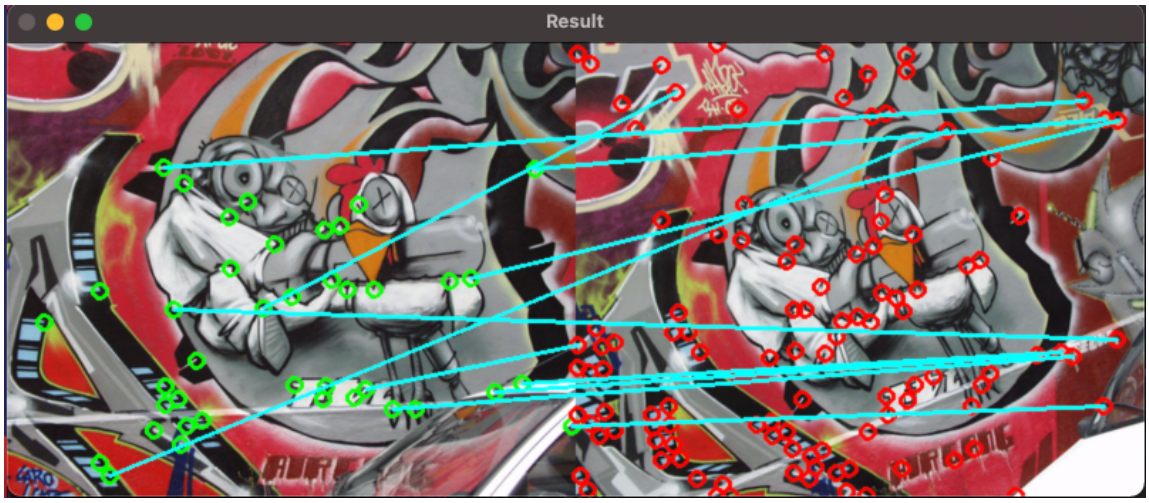
After finding all corners using Harris Corner Detection in Part 1, we now have 2 vectors of features from 2 input images. The feature description and feature matching processes follow the rules below:

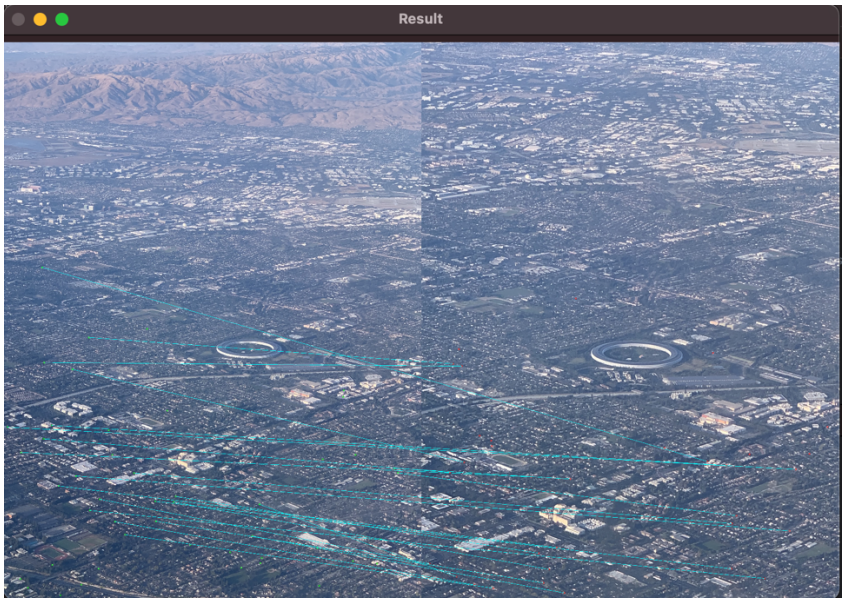


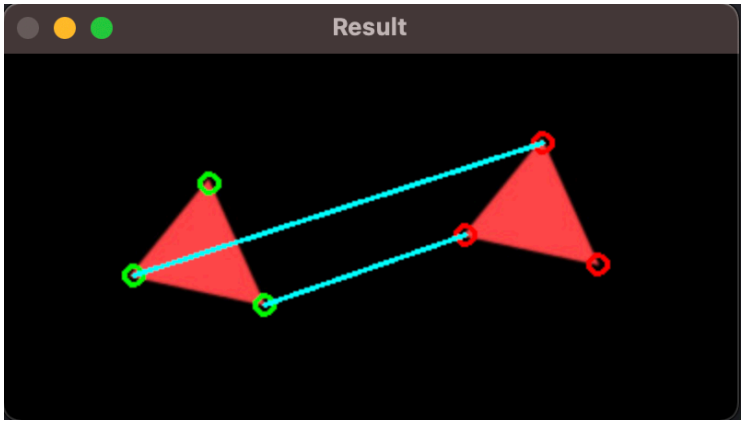
4. RESULT






Result


Result


Result

5. SUMMARY TO ICCV PAPER

(Brown, & Lowe. (2003). Recognising panoramas. Proceedings Ninth IEEE International Conference on Computer Vision. doi:10.1109/iccv.2003.1238630)

Panoramic image has become increasingly popular. In 2020, most of the smart phone can take fantastic panoramic photos. However, the mathematical and computational process behind the scene is way complicated than most of us think. It turns out that when shooting a panoramic photo, the camera is actually constantly taking many photos and uses algorithm to detect and match object on each image, and thus connect images all together.

The first step in the panoramic recognition algorithm is to extract and match SIFT features between all of the images. After we do all the features matching, we start working on the image matching to bring photos together. This step is crucial, and it needs us to build a probabilistic model to do pairwise matches. Once the matches are done, we use bundle adjustment to solve for all of the camera parameters jointly.

After reading this paper, I have a new understanding of modern photography. As a photographer myself, I love looking at the world through my lens. However, as I am learning and reading more toward image processing, I realized that there is way more knowledge hidden underneath my camera. I believe, the future of photography is all about computation.

6. FAST VS HARRIS CORNER DETECTOR

Like we mention above, Harris' corner detector takes the differential of the corner score into account with reference to direction directly, instead of using shifting patches for every 45-degree angle and has been proved to be more accurate in distinguishing between edges and corners. Similarly, features from accelerated segment test (FAST) is a corner detection method, which could be used to extract feature points and later used to track and map objects in many computer visions tasks. Using the same input image, we can see that FAST algorithm detects way more corners than Harris algorithm in almost half of the time Harris takes.

I did more research and I also found that there are limitations to the FAST algorithm as well. In FAST algorithm, n usually represents the number of contiguous pixels in the circle. When n<12, the algorithm does not work well in because when n < 12, the number of interest points detected are very high. Besides, since the detected corner must have a ring of darker or lighter pixel values around the center that includes both edges of the corner, crisp images do not work well.

Algorithm	Advantage	Disadvantage
Harris	Accurate in distinguishing between edges and corners	Need to set a threshold value.
FAST	Computational Efficiency (faster than many other well-known feature extraction methods)	Crisp, or machine generated images do not work well using FAST.
SIFT	can generates large numbers of features that densely cover the image over the full range scales and locations.	The opposite to FAST, SIFT is mathematically complicated and computationally heavy

7. SUMMARY AND FUTURE WORK

Visually, I think the accuracy of my feature matching is not as accurate as I would expect. Since I was not happy with the matching result, I did some research on how I could better my application in the future.

A. Optimize Harris Corner Detection Algorithm

Started from the very beginning, Harris corner detection is not the best detection algorithm to use. A 2009 paper proposes an improved algorithm based on Harries corner detection. The improved algorithm is based on the neighboring point eliminating method. It reduces the time of the detection, and makes the corners distributing more homogenous so that avoids too many corners stay together.

B. Better Detection Algorithms

There are other detection algorithms that are widely used in the real world.

Below are a few similar algorithms that can possibly produce a better result.

- SIFT (Scale Invariant Feature Transform)
- SURF (Speeded Up Robust Feature)
- FAST (Features from Accelerated Segment Test)
- ORB (Oriented FAST and Rotated BRIEF)

C. Optimize Non-Maximal Suppression Algorithm

Second, I believe that there is a better way to do the non-maximal suppression too. According to a newly published paper, the traditional Adaptive Non-Maximal

Suppression (ANMS) algorithm is rarely used in real life is due to its high computational complexity. To overcome this limitation, the researchers propose three novel approaches called Range Tree ANMS (RT ANMS), K-d Tree ANMS (K-dT ANMS), and Suppression via Square Covering (SSC). All three approaches can not only speed up the computing efficiency, but also homogeneously distribute detected features in the image.

D. Optimize Feature Descriptor

Using Multi-Scale Oriented Patches (MOPS) could help us get a better result. We can implement orientation normalization to use the dominant image gradient direction to normalize the orientation of the feature. Besides MOPS, there are also other feature descriptors we can use such as Haar Wavelets.

Reference

Bailo, O., Rameau, F., Joo, K., Park, J., Bogdan, O., & Kweon, I. S. (2018). Efficient adaptive non-maximal suppression algorithms for homogeneous spatial keypoint distribution. *Pattern Recognition Letters, 106*, 53-60. doi:10.1016/j.patrec.2018.02.020

Brown, & Lowe. (2003). Recognising panoramas. Proceedings Ninth IEEE International Conference on Computer Vision. doi:10.1109/iccv.2003.1238630

Zhang, X., & Ji, X. H. (2012). An Improved Harris Corner Detection Algorithm for Noised Images. *Advanced Materials Research, 433-440*, 6151-6156. doi:10.4028/www.scientific.net/amr.433-440.6151