ECEN 765 Machine Learning with Networks
Mid-Term Report

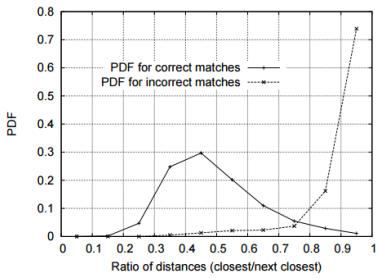
Jicheng Gong

November 10, 2017

1 Sift matching approach

1.1 David Lowe's original method

The approach Sift method used for matching is finding each key point's nearest neighbor from training images. The nearest neighbor is defined as the keypoint with minimum Euclidean distance of the 128 dimension descriptor.
And instead of using a global threshold, a more effective measure is by comparing the distance of the closest neighbor to that of the second-closest neighbor. According to Lowe's experiment, he chose the thresh hold to be 0.8, which has the best overall performance.



1.2 Inner Product method

To eliminate the affection produced by illumination, all the SIFT descriptors are normalized to size of 512. This feature brings us much convenience to calculate its inner product value to get the Euclidean distance.

$$x_1 \cdot x_2 = |x_1| \cdot |x_2| \cdot cos\theta = 512 \cdot 512 \cdot cos\theta = 262144 \cdot cos\theta$$

From the formula above we can easily find that the dot product value will be larger if two vectors are close to each other, which, in other words, have a smaller Euclidean distance between each other. So we may also be able to use inner product as a criteria. In fact, in some modified SIFT algorithms, inner product has already been used to do putative matches before calculating the Euclidean distance.

1.3 Dimension Reduction

Despite SIFT has high robustness for feature detection, its use is somehow limited in online and real-time applications. Because the features take a relatively long time to compute, and finding correspondences for the 128-dimensional SIFT vectors is computationally expensive given that matches are determined from the Euclidean distance metric.
Many approaches have been made to solve this problem and there are mainly two kinds of approaches, either do clustering of keypoints to eliminate similar keypoints so that

the number of keypoints being matching will reduce or do reduction on descriptor itself. One of the most famous improvement for latter approach is PCA-SIFT, which can effectively reduce dimension while remaining a good accuracy.

However, since PCA needs large amount of data to extract important features, it may not suitable for real-time application either. Inspired by SIFT-HHM algorithm proposed by Geoffrey, we may find that certain dimensions in SIFT descriptors will have larger contribution. It may be possible to just use part of 128 dimensions to calculate the inner product.

2 Algorithm

2.1 Inner product

The image dataset I used is the KITTI Vision Benchmark Suite. It is a group of consecutive images shot while a car was driving. I modified SIFT algorithm to be able to export descriptors and coordinates for each keypoint in one image. Also export match pairs between two consecutive images as a ground truth to be compared with. The following figures show two groups of descriptors from two consecutive image, they respectively contains m and k descriptors in total.

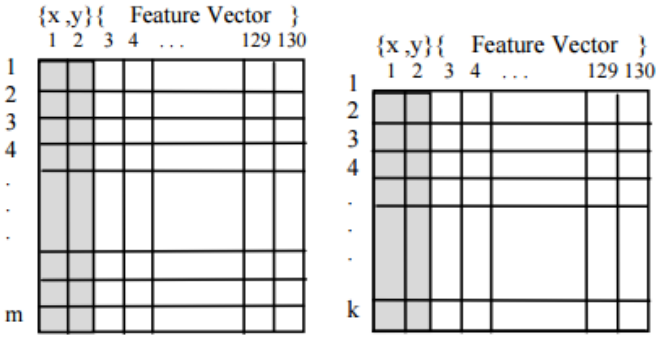


Figure 1: Decriptors extracted by SIFT

The first and second column above represent the coordinate of the keypoint while the remaining 128 columns are the descriptors. Suppose we set matrix M to represent descriptors for first image, matrix K for second image. The inner product matrix P will be:

$$P = M \cdot K^T$$

$P[i,:]$ represents the dot product value of i^{th} keypoint in first image with all the keypoints in second image. Similarly, $P[:,j]$ represents the dot product value of j^{th} keypoint in second image with all the keypoints in first image. The largest dot product in each row will be extracted and compared with the thresh hold. If it is larger than thresh hold then we will claim there is a match.

2.2 PCA based Inner product

In this part we want to see the performance if some of the dimensions are reduced. We combine every two consecutive descriptor matrix into one matrix and perform principle component analysis for each combined matrix. Then perform PCA transform for each matrix.

2.3 Evaluation Metrics

To evaluate the matching,we use recall vs. 1-precision. Recall will measure the ratio between the number of correct matches retrieved over the total of commit matches. As

we can achieve a 100% of recall by returning a set with all possible matches, we notice that the recall measure is not enough; therefore, we also calculate the imprecision(1-precision). The precision measures the ratio between the quantity of correct retrieved matches over the number of retrieved matches, and the imprecision measures the ratio between the number of false retrieved matches over the total number of retrieved matches. Consequently we will realize that the Recall vs. 1-precision curve shows adequately the trade off to obtain.

$$Recall = \frac{Correctmatchesretrieved}{Totalnumberofcorrectmatches}$$

$$1 - Precision = \frac{Incorrectmatchesretrieved}{Totalofmatchesretrieved}$$

3 Experiment result

3.1 Inner product

In the experiment I used 114 consecutive images to do SIFT matching. Figure 2 3 8 shows Error rate and Recall rate of the original inner product method with different threshold. We can see that 245000 threshold has almost the same accuracy as 250000 while it has a higher Recall rate.

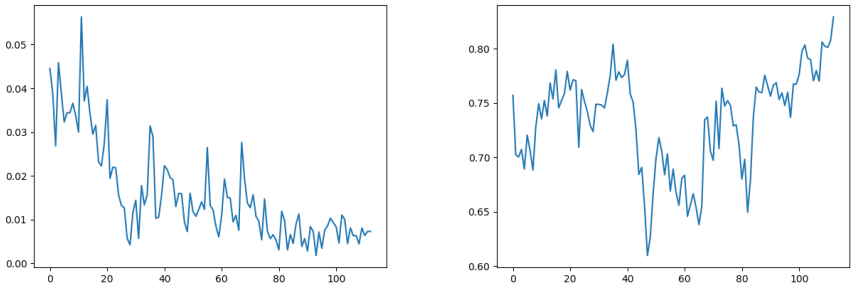


Figure 2: Error rate and Recall with threshold 250000

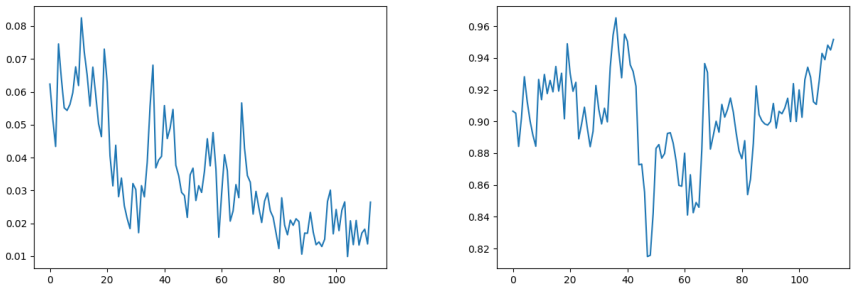


Figure 3: Error rate and Recall with threshold 245000

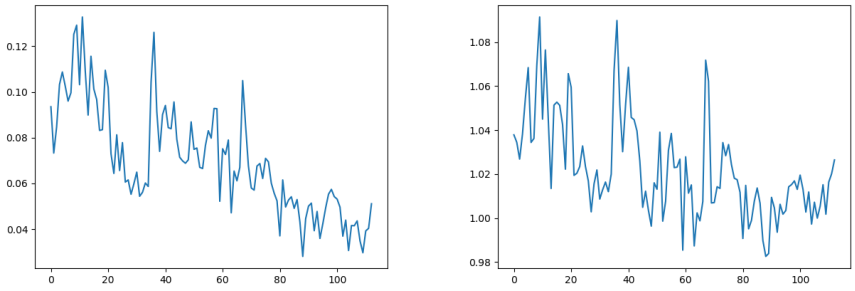


Figure 4: Error rate and Recall with threshold 240000

3.2 PCA

In this part I try to use PCA to reduce the dimension first. However, with the reduction of dimension, we should also decrease the threshold, too.

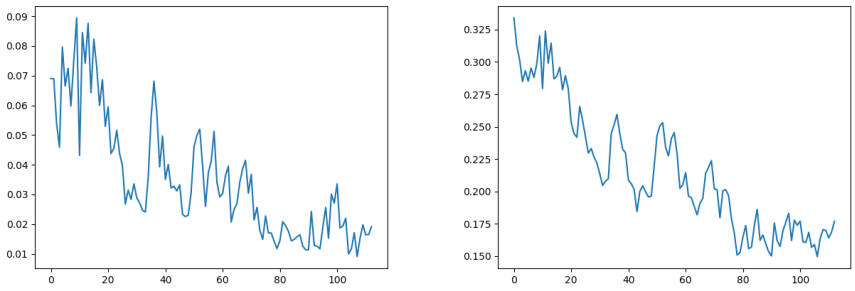


Figure 5: 80 dimension Error rate and Recall with threshold 160000

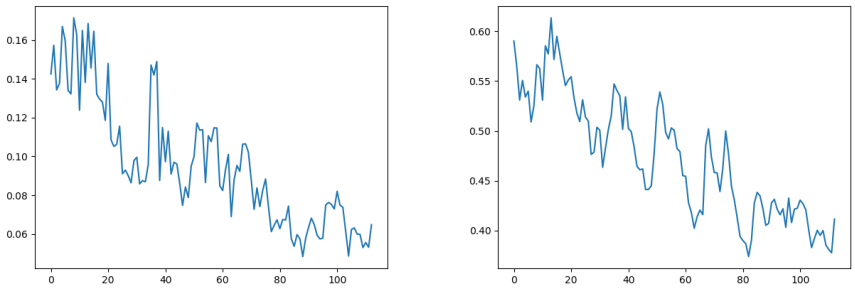


Figure 6: 64 dimension Error rate and Recall with threshold 140000

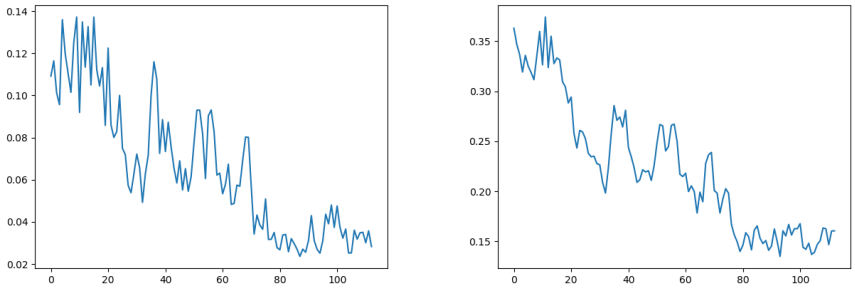


Figure 7: 32 dimension Error rate and Recall with threshold 140000

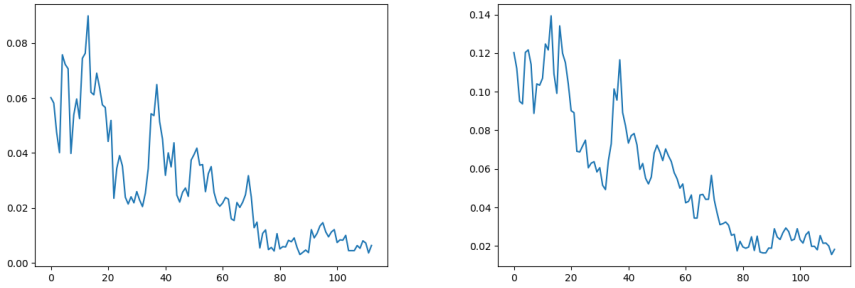


Figure 8: 16 dimension Error rate and Recall with threshold 140000

We may find that with fewer dimensions, the error rate will increase, otherwise the recall rate will drop vastly. What's more, the data set seems to have some bias because both error rate and recall rate drops in last a few images. Which indicates that a universal threshold may not work for this problem.

4 Future plan

4.1 Result analysis

So far I have used PCA to do dimension reduction. And the results showed that I should have a better threshold strategy. Also due to PCA will transform original vectors into another coordinate space, it will be not intuitively clear for us to find the relation between important dimensions and image property.

5 Future plan

In next two weeks I will try SOM or other feature selection algorithm to extract important dimensions which can minimize the error rate but maximize the recall rate and use a good threshold function. Besides, use the other two weeks try to find out the possible connection between image property to important dimensions.