

A Fast Image Matching Approach for Indoor Image Localization

by

Hongyi Fan

B. Engineering, University of Electronic Science and Technology of China, China,
2011

Thesis

Submitted in partial fulfillment of the
requirements for the Degree of Master of Science
in the School of Engineering at Brown University

Providence, Rhode Island

May 2016

AUTHORIZATION TO LEND AND REPRODUCE THESIS

As the sole author of this thesis, I authorize Brown University to lend it to other institutions or individuals for the purpose of scholarly research.

Date: _____

Signature: _____

Hongyi Fan, Author

I further authorize Brown University to reproduce this thesis by photocopying or other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Date: _____

Signature: _____

Hongyi Fan, Author

This thesis by Hongyi Fan is accepted in its present form
by the School of Engineering as satisfying the
thesis requirement for the degree of Master of Science.

Date _____

Signature: _____

Dr. Benjamin Kimia, Adviser

Recommended to the Graduate Council

Date: _____

Signature: _____

Dr. , Reader

Date: _____

Signature: _____

Dr. , Reader

Approved by the Graduate Council

Date: _____

Signature: _____

Dr. Peter Weber, Dean of the Graduate School

Vitæ

Hongyi Fan was born on May 3rd, 1989, in Beijing. After completing high school at Beijing 101 High School in 2007, he chose to attend University of Electronic Science and Technology of China(UESTC) to pursue a Bachelor degree in Electronic Science and Engineering. After that, he got a Master of Engineering degree in Electronic and Communication Engineering from UESTC in May 2014. In August 2014, he entered the School of Engineering at Brown University.

Dedicated to my family

Acknowledgements

I would like to thank the many people who helped me out along the way to completing this thesis. In particular, I would like to thank Daniel Keenan for early troubleshooting, training, and design of the experimental setup. I would like to thank Paul Waltz and Brian Corkum for helpful suggestions and guidance on updating and altering the experimental setup. Ron Szalkowski and Team Wendy provided the NOCSAE headform, football helmet, and drop tower test data and were an invaluable asset throughout the course of this project. Everyone in the Franck Lab was very helpful and always provided me guidance when I needed it. In particular I would like to thank Eyal Bar-Kochba, Jon Estrada, and Mark Scimone for their input and help. I would like to thank my advisor, Christian Franck, for working with me and supporting my interests. Finally, I would like to thank my family for their support and love.

Contents

Vitæ	iii
Dedication	iv
Acknowledgements	v
Chapter 1. Introduction	1
1.1 Introuction	1
1.2 Related Works	4
Chapter 2. Our Method	6
2.1 Deep Features from Convolutional Neural Network	6
2.2 Bag of Features Model	9
2.3 Post Processing	11
Chapter 3. Implementation Details	14
Chapter 4. Experimental Results	16
4.1 Environment, Datasets and Measurements	16
4.2 Results of deep feature with bag of features model	18
4.3 Different features from different layers	24
4.4 Examination of post processing	25
Chapter 5. Conclusion	31
Bibliography	32
Bibliography	32

Abstract of A Fast Image Matching Approach for Indoor Image Localization

by Hongyi Fan, Sc.M., Brown University, May 2016

Localization is an important and challenging task for robots and wearable navigation systems. In this paper, We present a multi-stage method using both local and global feature of images. We use mid-layer output from convolutional neural network(CNN) as a kind of general local feature to deal with the lacking feature problem. Also, CNN feature could grab more structural appearance-based information than other local features(etc. SIFT, SURF). Such feature works well with hierarchical bag-of-features model. Several post-processing strategies is examined to lower the ambiguity of the place location, include global features based re-ranking and geometric verification. We test such method on multiple different datasets, captured from both robot-mounted camera and wearable camera. The proposed approach has been integrated into an indoor vision-based wearable navigation system that can reach real time performance in use.

CHAPTER 1

Introduction

1.1 Introduction. In navigation systems, while navigating in an environment, the system should have the ability to recognize its current position. Multiple techniques was widely used in localization problems, e.g. GPS, WiFi positioning and ceiling positioning [1]. However, those techniques are hard to use in indoor environment where GPS signal cannot reach, which constrain the use of robot platform or wearable navigation system [2].

For indoor environment, a requirement for a wearable navigation system or a robot platform is the ability to localize itself within a environment which prior structure and layouts was known. Different traditional sensors was widely used in localization techniques, e.g. laser and sonar [2]. Though they can achieve high precision, however, due to high costs of the sensor and sensitivity to occlusions and moving objects, such traditional sensors cannot be used for all the environment. Vision based systems have gained focus recently for two mean reasons: (1) vision based system has lower cost and more portable than laser or sonar, which potentially offers the ability to build low cost wearable systems. (2) vision can provide more information than traditional sensor readings: laser or other traditional sensors only provides us the basic geometric informations of the environment, such limited information may provide ambiguity in similar structures, e.g. the similar room layouts and sizes will be treated as the same place [3]. With vision information, it is easy to input various information, includes geometric, appearance and contextual informations.

For vision based navigation system, image localization means for each query image, we can estimated the position in space, such position can be either 2D or 3D. For robot or wearable navigation systems, 2D positioning is widely used. And for quadcopters or other aerial vehicles, 3D positioning is more reliable. There are two

main approaches in achieving this with vision: (1) the prior knowledge of environment is represented with a series of tagged images, the position was estimated by retrieving the most similar image. This approach treats the position tag of the most similar image as the position of query image [4], (2) the prior knowledge of environment is represented with a 3D model [5]. Such model can be sparse or dense. The query image was registered into this 3D model, the registered results is the position of the query image. Both approaches have their own pros and cons for indoor image localization tasks. For the first type of method, the resolution of the result highly depends on the resolution of the dataset. If there is an image which has a very different view angle with the dataset, then the image localization will fail. On the other hand, this type of approach is efficiency in memory, which makes it more appropriate to scalable situation[6]. For the second type of methods, the prior map was represented by 3D models of the environment, so the performance of the method depends on the precision of the 3D models. It is possible to get a near precise 3D model from numbers of image with the recent structure of motion(SfM) research[7], then one can register the image into the 3D model. The common approach is to use the feature descriptors(e.g. SIFT), for the 3D points computed during the structure from motion reconstruction, formulating the correspondence as a keypoints matching problem. Once the correspondences are established, the pose of the camera was obtained by solving a perspective-n-points problem [8]. Such approach can reach high precision. However, the performance depends on the correctness of the 3D model. For outdoor environments, we always have enough keypoints to build a good 3D model of urban area, but for indoor environment, multiple reasons can make reconstruction fail: large homogeneous area(e.g. long hall way), image blurring(e.g. bad capturing) or similar structure in environment(e.g. same posters in room or same room layouts). Some sample images when structure from motion will fail are shown in Figure 1.1. Experiments on 3D reconstructions on our dataset can be found in Chapter 4, which is really bad. So for indoor image localization, we choose the approach which use image dataset to represent the map.



(a)



(b)



(c)



(d)

FIGURE 1.1. The sample images that will make SfM fail: (a) Image blurring, (b) large homogeneous area, (c)&(d) similar layouts but in different location.

In this work, we propose an approach that use multi-stage approach to retrieval the most similar image in the dataset with geo-tagged position information. For the dataset, a vocabulary tree [9] model was trained with the local deep feature from datasets. Our approach uses deep features comes from the convolutional layers of deep convolutional neural network(CNN)[10]. We do not set fully connected layers after CNN because we want to preserve the localization information of features. Each dataset image is indexed by term frequencyinverse document frequency (TF-IDF) vectors [11, 12]. For each query image, the TF-IDF vectors representation is calculated by sending the image's deep feature into the same vocabulary tree. A list of localization candidates can be obtained from comparing TF-IDF vectors. Due to

feature ambiguity, image blurring and other reasons, such localization result list may not be fully satisfied. In our method, we examine multiple different strategies to re-rank the result list in order to refine the final results. Two strategies are examined. One is re-ranking with the global features of query image and dataset images. Local features cannot represent the images fully. In bag-of-features model, the spatial layouts of local features is not checked. So the global representation can grab the information of the whole image. Two types of global features are examined: (1) gist feature [13] and (2) deep feature after fully connected layer [10]. More than that, Geometric verification approach is also examined in this paper. It tries to check the scale change and the affine transformation between patches of images. To get the best localization results, we need to select the smallest scaling and transformation as the best localization results.

This paper is structured as follows. The related works will be discussed in the rest part of this chapter. Chap. 2 explains the basic theory and methods conducted in this thesis. In Chap. 3 we introduces the implementation detail of our approach. Chap. 4 shows the evaluation of our method in both robot roaming dataset and wearable navigation system dataset. And the conclusions and future works are presented in Chap. 5.

1.2 Related Works. The vision based image localization was received focus recently. Bag-of-features model was widely used in this area, and achieved good performance. Schindler et al. [14] firstly use vocabulary tree model [9], achieved a scalable image localization system using a dataset with 30,000 images. They also announced the concept of "informative features" which let the algorithm to use the very distinctive features. [15] presents an approach that uses global likelihood and human travel prior. FAB-MAP system[16], implemented an usable image localization and loop detection for navigation for omnidirection camera using bag-of-features model and Chow-Liu tree to learn distribution of bag-of-feature vectors. But bag-of-features model ignores the spatial layouts of features. To solve that, Gálvez-López et al. [11] also used bag-of-features model, but using binary descriptor to raise the temporal

performance. Also, their work introduces the geometric verification strategy with random sample consensus(RANSAC) approach, which preserve the spatial layouts of the image. Also, repetitive feature points will influence the efficiency of TF-IDF indexing. Torii et al. [17] detected repetitive features and modified the weights of indexing to get rid of the negative effect of repetitive features. To enlarge the ability of Bag-of-features model, strategies was introduced into this field.[18] introduced hamming embedding into large scale image retrieval method to get more precise visual words. Zamir et al. [4] uses graph model to unify the local features and global features, treating the image localization problem as a optimization problem.

Approaches other than bag-of-features-based approaches were also presented. [1] presents an approach using simple but effective feature voting scheme. And it has the ability to process image sequence in real time. With the growth of SfM technique, it is possible to get good reconstruction model of environment. [19, 5, 20] uses different approaches to matching keypoints between query image and 3D models to register camera into the world coordinate system. All these methods uses more disk spaces for storage of models, which makes them not that useful for scalable environments. The performances are also depend on the quality of reconstruction. Lacking of features makes them not useful for indoor environments.

With the development of GPU computing and deep learning, convolutional neural network(CNN) plays a important role in computer vision community. CNN representation has achieved tremendous performance in scene recognition[21]. It shows the ability to model the scene image into an compact representation, which could be useful for image localization. Ali et al. [22] announced that the mid-layer output of CNN can be treated as a kind of local features and descriptors, which gives the potential to using CNN features in image localization task. With CNN, Lin et al. [23] created a similar metric between ground image and aerial image to do localization.

CHAPTER 2

Our Method

Our approach consists with 3 main parts: dataset preparation, query retrieval and post processing. Dataset preparation indexing the whole dataset with a vocabulary tree model. We feed each image in dataset into the convolutional neural network to get global features and local features from neural network. Then build a vocabulary tree to cluster all the features from dataset. All the centroids are indexed with TF-IDF indexing strategy. The query image is also fed into the same network on the air, then calculated the TF-IDF vector of query image. We can get a result list from vocabulary tree model. The result list is refined by post processing. The procedures of our method is shown in Figure 2.1.

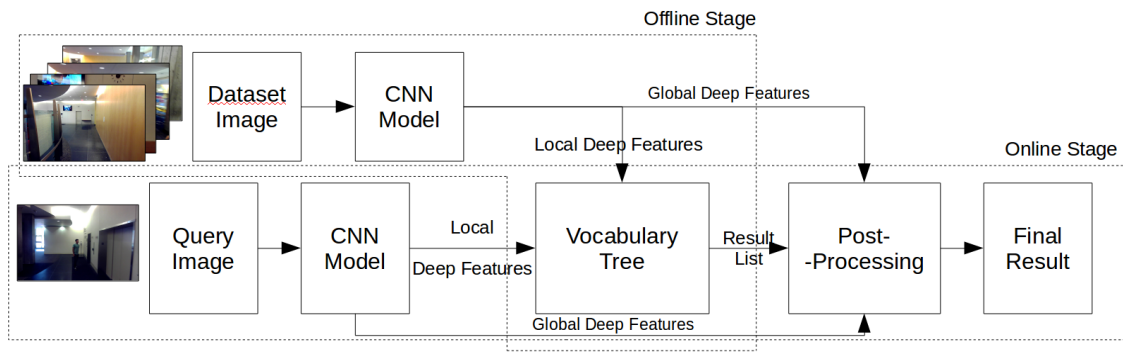


FIGURE 2.1. The framework of proposed approach. The framework consists with offline stage and online stage.

2.1 Deep Features from Convolutional Neural Network. Extracting local features is critical in image retrieval and corresponding image localizations. Previously, SIFT, SURF and other keypoints and descriptors generation method were used in extracting keypoints and the descriptors of the patches around keypoints.

Such point-based descriptor is too local to describe the whole image. However, convolutional neural network(CNN) consists with several convolution layers in different scales. The convolution layer has the view field which covers the whole image. So there is less information loss than other point-based keypoints and descriptors.

The architecture of CNN contains multiple layers, such architecture is shown in Figure 2.2. Convolutional layers consist of a series of filters. Each filters can be seen as a neuron. Each filter takes inputs from a feature map of the previous layer; the weights for each neuron(the convolutional filters) are the same in the convolutional layer. After each convolutional layer, there may be a pooling layer. The pooling layer takes small rectangular blocks from the convolutional layer and downsamples it to produce a single output from that block. There are several ways to do this pooling, such as taking the average or the maximum, or a learned linear combination of the neurons in the block. Our pooling layers will always be max-pooling layers; that is, they take the maximum of the block they are pooling. Finally, after several convolutional and max pooling layers, the high-level reasoning in the neural network is done via fully connected layers. A fully connected layer takes all neurons in the previous layer (be it fully connected, pooling, or convolutional) and connects it to every single neuron it has. Fully connected layers are not spatially located anymore (outputs of fully connected layer is an one-dimensional vector), so there can be no convolutional layers after a fully connected layer.

For recognition task, fully connected layer creates a conventional classifier to give the score of each categories. But for our situation, each of our dataset image is one category. It is impossible to have enough images to train a fully connected layer for image localization considering about the data hungry property of CNN. So in proposed approach, we treat pre-trained CNN as a general feature extractor. So the proposed approach cancel the fully connected layer. The feature consists with the responses from convolutional layer concatenating the response from same rectangular area. Such feature corresponding one rectangular area from the original image, which covers the whole image. The way to get one mid-layer feature is shown in Figure

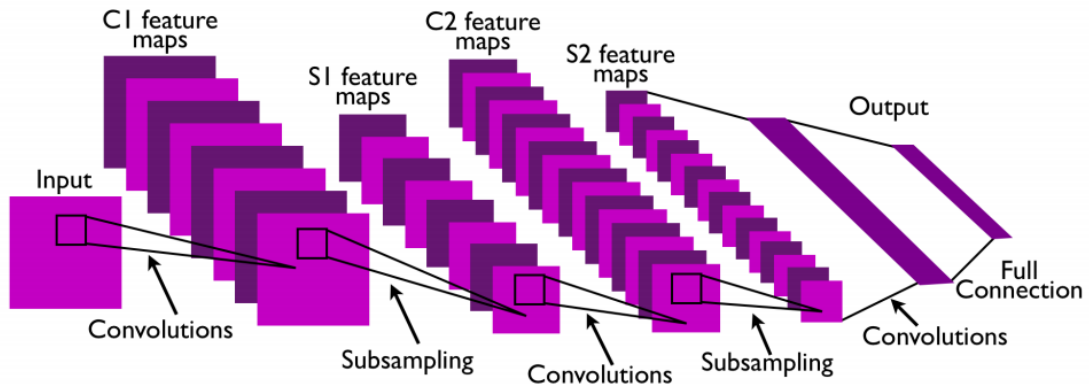


FIGURE 2.2. The architecture of a typical CNN with two convolutional layers and two pooling layers. For lower layers, multiple convolution layer and pooling layers are stacked together. The upper-layers are fully connected layer implemented by conventional neural network. The input to the first fully-connected layer is the set of all features maps at the layer below. This figure comes from [24].

2.3. For lower level feature maps, the responses of pre-trained CNN are low level features such as blobs, corners or edges. Those lower level features cannot represent the difference between different images fully. The higher level features give us an abstract, compact representation of each scene or position. In experiment part, we examine the influence of extracting features from different convolutional layers.

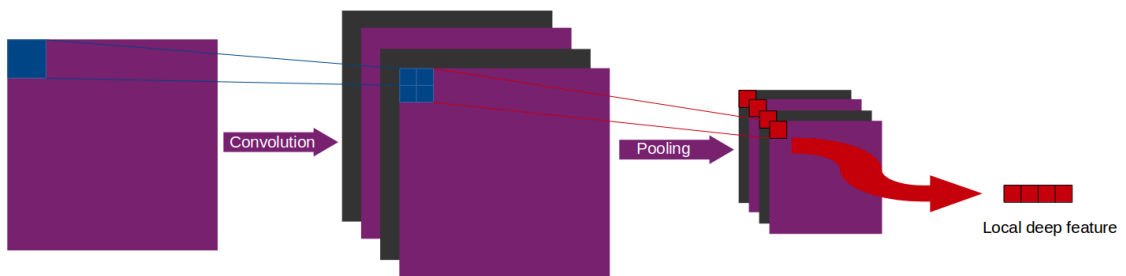


FIGURE 2.3. The procedure to get each deep features. The responses of a specific pooling layer from the same area are concatenate together to build feature vectors.

Extracting local features (keypoints and their descriptor vectors) has usually high cost of computation in terms of computation time when comparing images. This is often the bottleneck when these kinds of techniques are applied in real time. Regularly, binary features and descriptors(e.g. FAST keypoints [25] and BRIEF descriptors [26]) were used to make the feature extraction fast. Computing CNN features of an image is expensive in CPU. But with the help of developed GPU computing technique, one image feed into the network will take lower than 20ms. More discussion and experiments on computing time will be presented in Chap. 4.

2.2 Bag of Features Model. The bag of words is a technique that uses a visual vocabulary to index an image into a very sparse numerical vector, allowing us to deal with scalable image set. The visual vocabulary is created offline by discretizing the descriptor space into N visual words. In the case of the hierarchical bag of words, the vocabulary is structured as a tree. To build it, we extract large amount of deep features from images. The extracted deep features are firstly clustered into k clusters by performing k-means or k-medians clustering. These clusters form the first level of nodes in the vocabulary tree. Subsequent levels are created by repeating this operation with the descriptors associated with each node, up to L times. We can obtain W leaves, called visual words. Each word is given a weight according to its relevance in the training corpus, decreasing the weight of those words which are very frequent and, thus, less discriminative. For this, we use the term frequencyinverse document frequency (TF-IDF). Term frequency(TF) represents the frequency of the word appear in the image. Such weight is:

$$weight_{tf} = 1 + \log(N_w)$$

where w_{tf} is the word weight based on TF, N_W is the number of W th visual words occur in one image. inverse document frequency(IDF) represens by:

$$weight_{idf} = \log\left(\frac{N}{|\{w \in N\}|}\right)$$

where N is the number of image in the dataset, the denominator is the number of images where the word w appears. The TF-IDF weight of the words can be obtained as $weight_{tf-idf} = weight_{tf} \times weight_{idf}$. or a given query image, the deep features is firstly extracted. Each feature is fed into the vocabulary tree. For each level of the tree, the features feeds to the tree selecting the closest centroid that minimize the Euclidean distance. The same minimization is done in each level of the tree. The feature will be clustered into one leaves. After all the features in one image are all clustered into corresponding words, one can obtain a sparse histogram of words. For each sparse histogram, TF-IDF weights will be calculated for each words in one query image. Then we get a vector of weights of words. That is the representation of one image. The similarity between is measured by Euclidean distance between two sparse vectors.

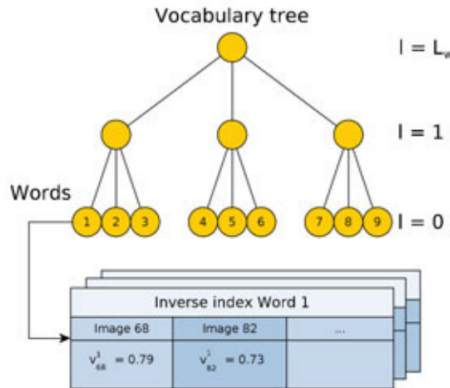


FIGURE 2.4. Example of vocabulary tree and direct and inverse indexes that compose the image database. The vocabulary words are the leaf nodes of the tree. The inverse index stores the weight of the words in the images in which they appear. The direct index stores the features of the images and their associated nodes at a certain level of the vocabulary tree. This image comes from [11]

Other than bag of features model, An inverse index table are maintained in the approach. The inverse index structure stores all the parents image of each word w_i in the vocabulary. Such structure has the ability to access the dataset images quickly.

Similar to [11]. We extend the inverse index to store both image index and its weights for corresponding words. Then one can give each image a score by their weights in real time. This inverted index can be updated on the air by adding new images and its features into the index structure. The architecture of two structures is shown in Figure 5.

2.3 Post Processing. Bag of features model works fine with deep features. One can get an initial result list. It can reaches a good performance. Post processing is applied in the Top N candidates to eliminate false feature matches. There are two main strategies to do post processing: one is verifying with global features, another way is verifying geometric consistency. In this paper, we examine these two categories of post processing methods.

2.3.1 Reranking with Global features. A global descriptor describes the whole image. They are generally not very robust as a change in part of the image may cause it to fail as it will effect the resulting descriptor. But our result list from previous stage is good enough. So for eliminating false results, global features is a good choice to eliminate results that is differ from query image. Two types of global descriptors are examined in this paper: gist global descriptor and deep global descriptor from CNN. Some samples of all these two global descriptors are shown in Figure 6. Once the global descriptor is extracted, the Euclidean distance between query image and images in result list can be used to re-rank the list.

Gist global descriptor, proposed by Oliva et al. [13], represents the dominant spatial structure of a scene by modelling perceptual dimensions(e.g. naturalness, openness, roughness, expansion and ruggedness). However, for indoor environment, those perceptual dimensions may be the similar for different positions with same structural layouts(e.g. different hallway or different living room). More discussions and experimental results will be shown in Section 5.

Deep features comes from the fully connect layer of a specific CNN, which use a conventional neural network to compact all the feature maps from previous convolutional layers. The visualization of deep global features is shown in Fig.6(a). It

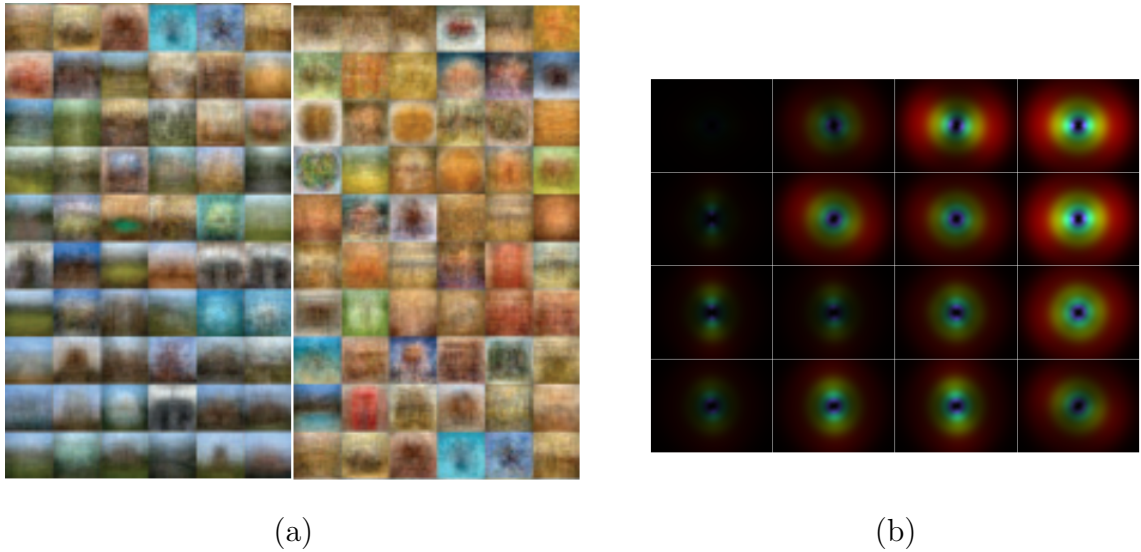


FIGURE 2.5. Examples of global features (a)Deep global features of multiple images from fc7 layer in AlexNet. Each block is the global feature of one image. This image comes from [21]. (b)Gist global feature of one image.

shows that the deep global feature can grab a outline or a sketch of a scene to distinguish different positions. More discussions and experimental results will be shown in Section 5.

2.3.2 Reranking with Geometric Verification. Bag of features model ignores the spatial layouts of local features. That may generate false results. For two same position, there should be now translation and scaling in spatial layouts of all the local features. Previously, for keypoints, RANSAC was widely used for estimating the translation and scaling between two different images. If the geometric transformation is consistent, then it has a higher score. However, for local deep features, we don't have specific keypoints. So we cannot use RANSAC to estimate the transformation. We estimate such transformation by using appearance, with the fast match method proposed by Korman et al. [27].

Such method trying to minimize the optimization problem:

$$\min \frac{1}{n^2} \sum_{p \in I_1} |I_1(p) - I_2(T(p))|$$

where I_1 and I_2 are two images, p is the point in two images, T is the affine transformation, and n is the size of the image (we assume that the image is n -by- n). By minimize the Sum-of-Absolute-Differences(SAD) error with respect to T , we can get the affine transformation between one patch and one image. The searching space is huge. But this method sampling the pixel space and the solution space to make the algorithm efficiency. It is good for geometric verification since it is the only method existed who can deal with arbitrary affine transformations. Some sample results of this method is shown in Figure 7.



FIGURE 2.6. Example of one path and its corresponding matching path in another image.

With this method, geometric verification is achieved as follows. For each query image, similar to RANSAC, we pick several random image patch from query image. For each image in top N of result list, we estimate translation between image patches and candidate images. Once transformation is got. We firstly remove the image which random patches have no translation consistency which means that the difference of normal of translation vectors or rotation vector is larger than a threshold. After that, the rest result list is re-ranked by the product of length of translation vector and rotation vector. the effectiveness of such geometric verification will be discussed in Chapter 4.

CHAPTER 3

Implementation Details

In the implementation, we use pre-trained CNN model (called Place205-AlexNet) trained with 2.5 millions of scene images [21]. It is the only CNN trained with scene images, which is useful for image localization task. To get the local and global deep features, we cancel the last recognition layer. The CNN we use has 5 convolutional layers, and 2 fully convolutional layers. The input image will be resized into $227 \times 227 \times 3$. The first convolutional layer filters the input image with 96 kernels of size $11 \times 11 \times 3$ with a stride of 4 pixels. The second convolutional layer takes the input of previous layer, convoluted it with 256 kernel of size $5 \times 5 \times 96$. There are two max-pooling layers which follow the first and second convolutional layers, respectively. For the third, fourth and fifth layers, there is no pooling layer in between, the kernel size of all these three layers are all $5 \times 5 \times \text{Kernel numbers of previous layers}$. The kernel number of these three layers are 384, 384 and 256 respectively. Two fully connected layers all have 4096 neurons to convert the feature map into 1D descriptors. So for each local feature from fifth convolutional layer, the size is 1×256 . For other layers, the dimension of each local feature is their kernel numbers.

As for bag of features model, we implement hierarchy clustering by implementing k-means algorithm along with the k-means++ initialization strategy [28]. For level of the tree and clusters in different layers, we tried many different configurations. Then we selected the best one with 9 levels which has 7 clusters in each level. In total we have $7^9 = 40353607$ visual words. The result list is truncated with soft threshold: the result scores that lower than 90% of the highest score will be truncated. The final result list will be sent to post processing block.

We implemented two schemes of post processing block: one with global feature based reranking, another with geometric verification reranking. For global feature

based reranking, we need to obtain the nearest features in terms of query image. k-d tree seems the best choice, but it faces dimension curse and also it will took a long time to partition the space. In our case, linear search is a simple but effective solution. For geometric verification scheme, considering about processing time, one cannot select too many patches. We select only two randomly selected patches to do geometric verification. It works smoothly on portable laptop computer. For the patch size, fast match method announced that the bigger the patch, the faster it estimate the transformation. So we select 80 % of original image as patch size which can deal with zooming in and zooming out.

Experimental Results

In this section, we explain the numerical results of our method in different aspects. This section is structured as follows: Sec. 4.1 introduces our experiment environment, the datasets we used and the evaluation measurements. In Sec. 4.2, the experiment results of bag of features and deep features is shown, along with the comparison with different baselines and different parameter sets. In Sec. 4.3, different features from different layers is examined. In Sec 4.4, two post processing schemes were checked.

4.1 Environment, Datasets and Measurements. We test this method on a laptop with i7 2.5Ghz CPU, 16GB memory, and GeForce GTX 860M GPU with 6GB graphic memory. Such laptop is very popular in market nowadays, which could be widely used in robots and wearable systems.

We test this method on two different datasets. KTH-IDOL2 datasets is a dataset for robot localization[29]. The image sequences were captured by a robot platform roaming in the lab environment at KTH. The location tag is acquired by laser scans and odometry technique. Each image sequence has near 1000 images and they all covered the same indoor area in different time. In our experiments, we merged different sequences into one big dataset which contains 10,382 images. And we use another image sequence as test set which contains 917 images. In this dataset, the image size is 320×240 . And images are captured with 5fps. Some sample images are shown in Figure 4.1. And the roaming path of two paths are shown in Figure 4.2.

Another dataset we used is a dataset captured with wearable camera mounted on a glasses. We captured this dataset in the lobby of Barus & Holly building at Brown University. Unlike the smooth moving of a robot platform, the images captured by wearable camera has more vibration and blurring due to non-smooth motion of walking. Some sample images from this dataset is shown in Figure 4.3. In total, There

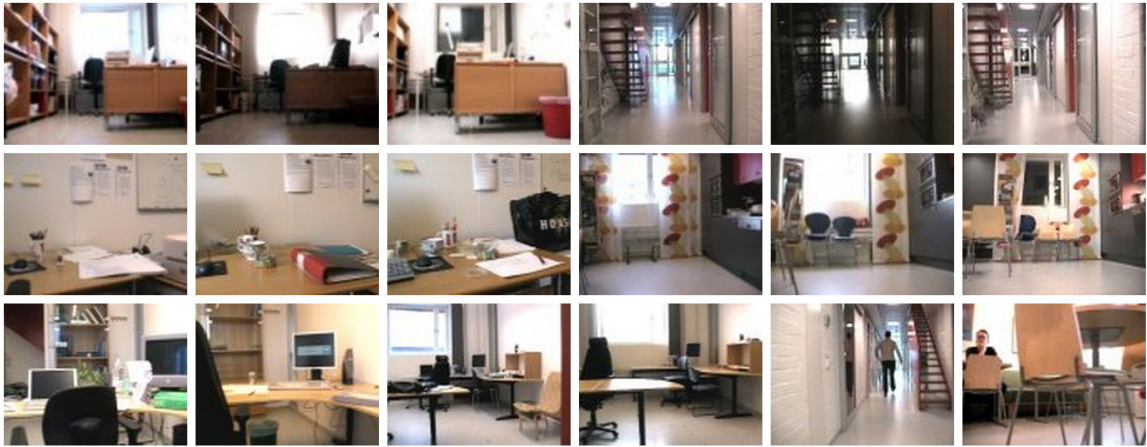


FIGURE 4.1. Sample images from KTH-IDOL2 datasets. Same areas at different times are shown.

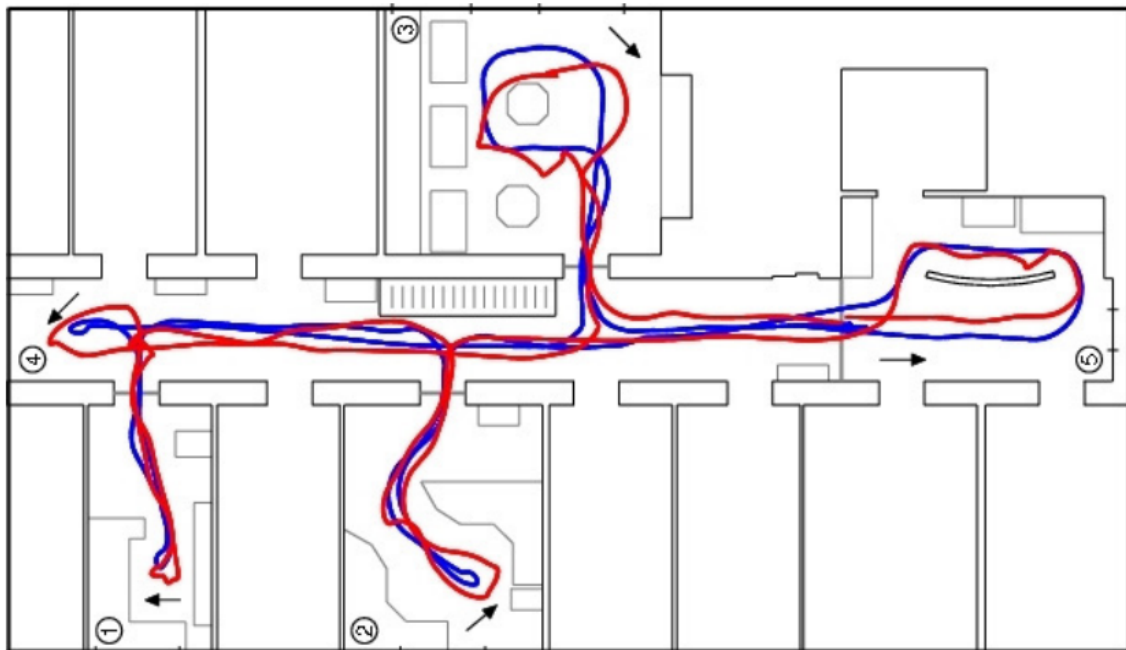


FIGURE 4.2. The difference between robot path for two acquisitions. They cover nearly the same area in the environment. This image comes from [29].

are 52,910 images in the dataset, captured with 30fps. The test set is a sequence contains 1442 images. The image size is 1280×800 . The location tag was measured by recording the walking path of the person. Then the location tag is marked by

re-measure the keypoints on the path, e.g.start point, turning point and end points. The localization tag is marked by interpolate the coordinates of all these keypoints on the path.



FIGURE 4.3. Sample images from B&H dataset. The whole lobby is covered by our dataset.

We treat location tag as ground truth. If the ground truth of the query image is within 1 meter to the top N result in the result list. Then we say our image localization is successful, since that the image within 1 meter is nearly the same for indoor environment. The distances between two image are described by L2-norm between two 2D position vectors.

4.2 Results of deep feature with bag of features model. We begin test on KTH-IDOL2 dataset with only deep local features and bag of features model, without any post processing strategy. We use deep local features from last convolutional layers, conv5 layers. For comparison, we use the same bag-of-features with SURF feature as one baseline of image localization task. The result of such results is shown in Figure 4.4.

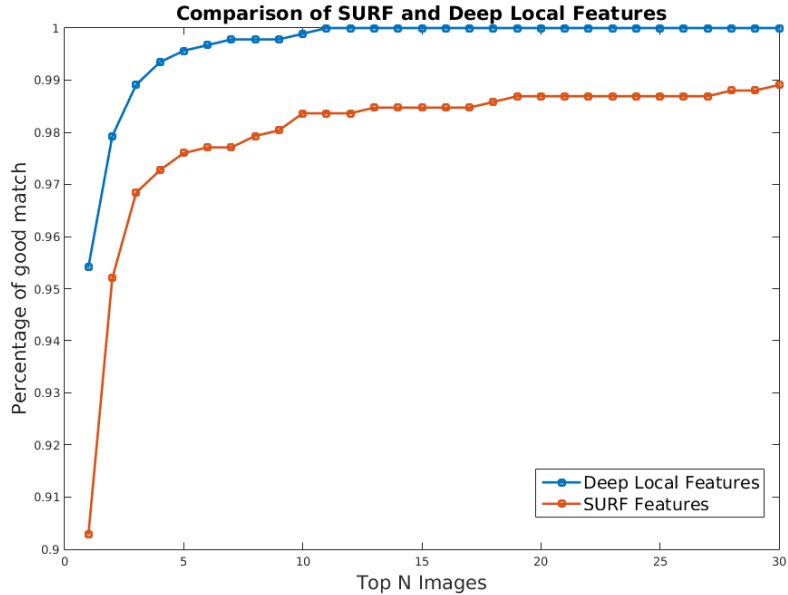


FIGURE 4.4. The comparison of the performance between our method and baseline method. The horizontal axis is the Top N result candidates. The vertical axis represents the percentage of queries which has good matching.

As one can see from Figure 4.5, the performance of deep features and SURF can all reach a good performance on this dataset. But deep features has a better performance. If we only check the top candidate, the deep feature can reach more than 95% precision, but SURF features can only reach 90.29%. Also, deep features can reach 100% precision with top 15 images, in the mean time for SURF feature can't reach 100% precision with top 30 candidates. Then we check the position between ground truth and the estimated position. Such result is shown in Figure 4.6.

As we mentioned before, the result is highly depend on the resolution of datasets. If the path of datasets is not similar to the query set. Then the results will be really bad. In this dataset, the roaming robot covers all the area of the building, so the result is acceptable except there is small drift in some area, due to the difference between query path and dataset path.

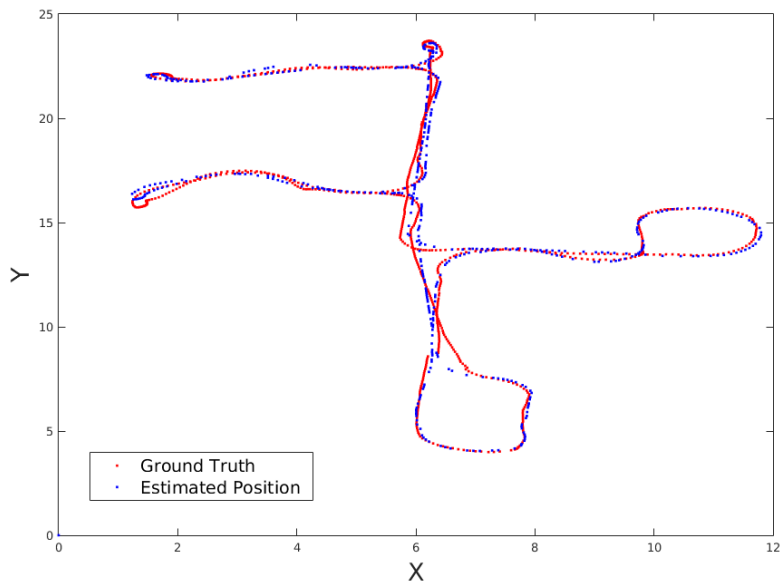


FIGURE 4.5. The corresponding positions between ground truth and estimated position. The unit of two axis is meter.

TABLE 4.1. The Error Distribution of Three Approach

Methods	Mean Error	Max Error	Min Error
Proposed Method	0.4429m	15.2387m	0.0064m
SURF and BoF model	0.6549m	15.7392m	0.0042m
Sattler et al.[5]	0.1933m	5.2813m	0.0018m

Another baseline is Sattler’s 3D model based image localization method. The dataset images are reconstructed into one sparse 3D model with VisualSfm software[30]. The 3D model created by VisualSfM is shown in Figure 4.7.

Such 3D model can correctly represent the spatial structure of the environment. We can run Sattler’s image localization method on this 3D model as another baseline. There is no candidate list in this method, we change the comparison from percentage to average error distance between ground truth and estimated position. The comparison of these methods are shown in Table 1. Here we use only top results from result lists.

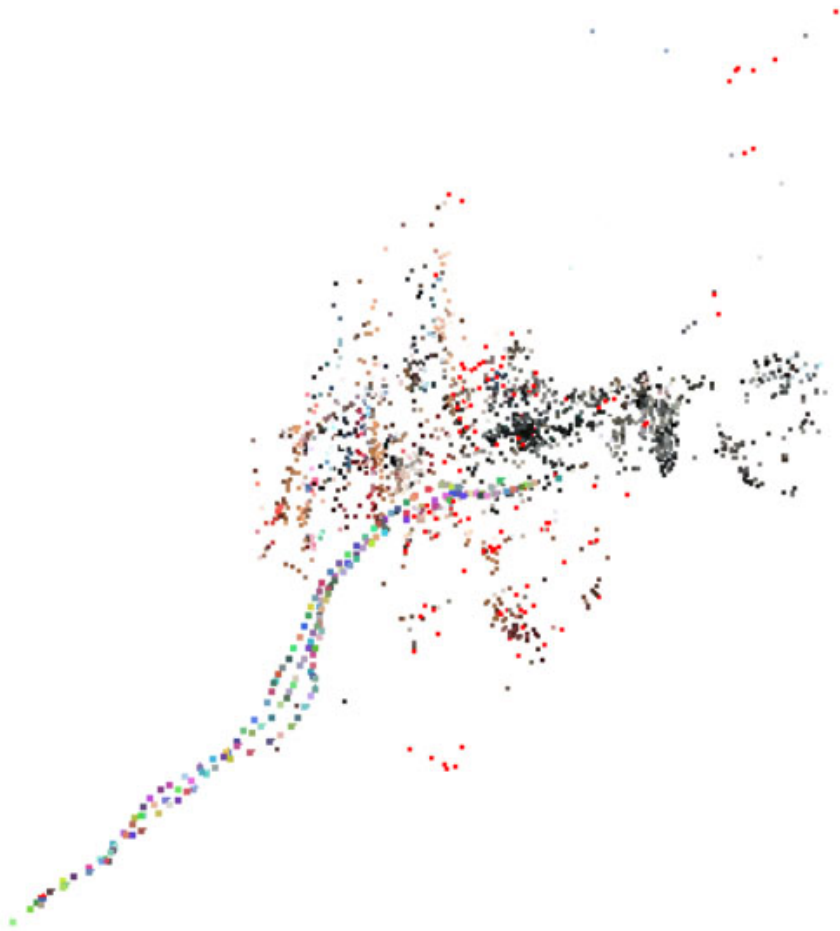


FIGURE 4.6. Part of the 3D model generated from KTH-IDOL2 dataset. Such Model correctly represent the layout of sparse keypoint and the positions of frames.

From the table above, the 3D model based localization approach has the best performance compared with retrieval approach. It is normal based on what we have introduced in Sec. 1. The precision of 3D based approach should be higher than retrieval based approach. However, here is no post processing stage, the result with post processing will be presented in Sec. 4.4. The experimental result of Barus & Holley Lobby dataset is shown in Figure 4.8.

As we can see from the Figure 4.9, the performance of proposed method is still better than SURF feature. Because of the size of images is larger than KTH-IDOL2

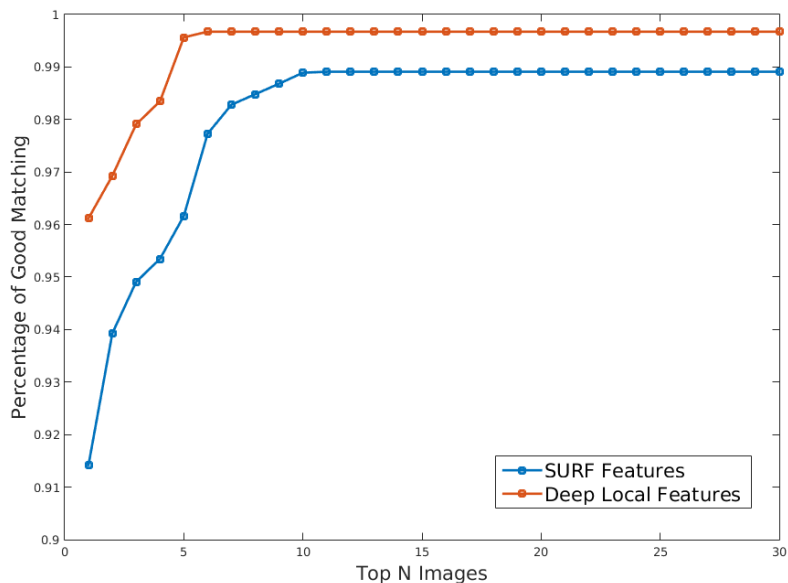


FIGURE 4.7. The comparison of the performance between our method and baseline method. The horizontal axis is the Top N result candidates. The vertical axis represents the percentage of queries which has good matching.

dataset, which makes images contain more information, the results of SURF features become better compared with IDOL2 dataset. The reason why the curve cannot reach 100% is that there are several images do not have corresponding ground truth matching which within 1 meter from the query image.

For near homogeneous image, our proposed approach has a dramatic improvement on those images. In Figure 4.10, we shown one near homogeneous image and its result candidates with both SURF based and proposed approach.

Actually, there is no exact same image for such query image in the dataset, so the method tend to search for the nearest image at that position. From Fig. 4.8, the proposed approach returns the same homogeneous area(the same white wall). But SURF based approach gives us some irrelevant results, which shows our method is efficient.

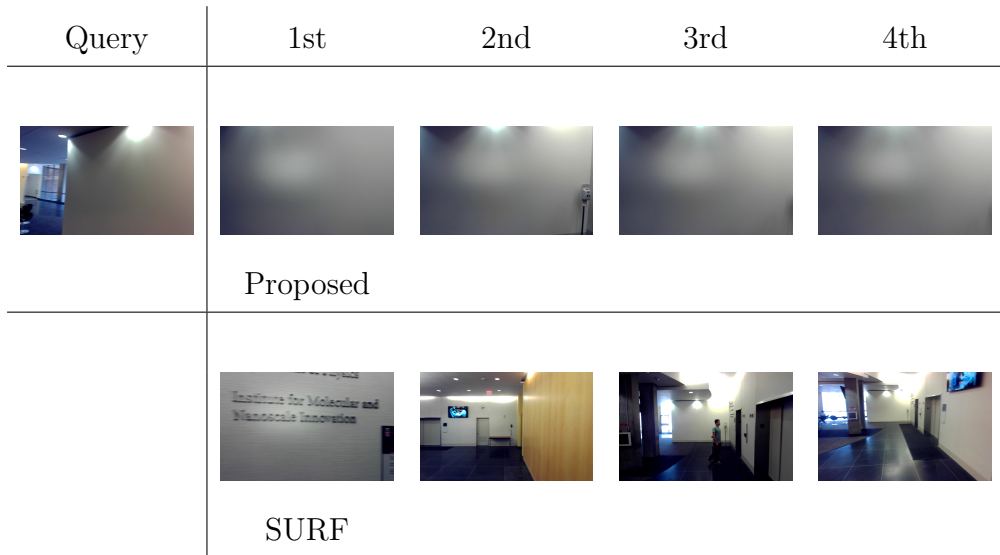


FIGURE 4.8. The result list with near homogeneous image.

TABLE 4.2. The running time comparison

Methods & image Size	Mean Running time per Image
320×240 , SURF-based Method	0.96s
1280×800 , SURF-based Method	1.76s
320×240 , Proposed Method	0.65s
1280×800 , Proposed Method	0.67s

As for 3D model based method, we build 3D model with VisualSfM software, but there are several homogeneous area. We cannot get a perfect 3D model. Such 3D model is shown in Figure 16. In such figure, all the colorful pyramids are position of cameras it estimated. This 3D model is bad, because of lack of features in some images, the feature matching is not good enough, which makes the position estimation fail. It shows that 3D model based localization method sometimes does not fit for indoor environment image localization task.

As for time complexity, extracting local features is always time consuming, for CPU computing, SIFT or SURF features will take more than half seconds. It is not usable in real time. Even for GPU based SIFT, it also needs near 100ms for

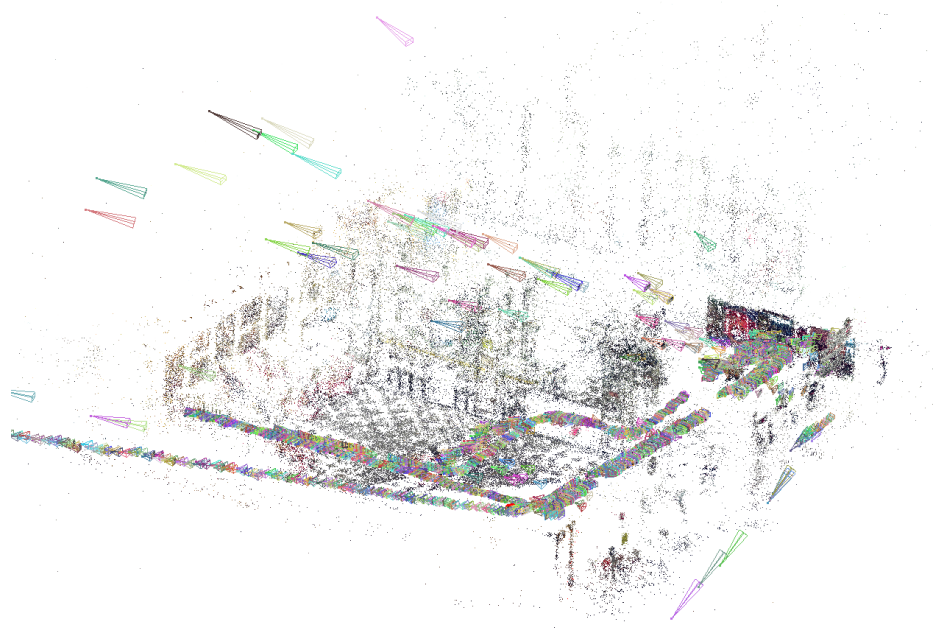


FIGURE 4.9. The bad reconstruction model of lobby of the Barus and Holley building. All the points are space points, colorful pyramids are estimated camera positions. Such model is really bad. Some of the cameras are floating on the air, which is impossible. Also, lots of cameras are in the wall.

extraction. But with CNN features, GPU has a good structure for such repetitive convolution operation. So for feature extraction, CNN feature can be extracted in 20ms. Bag of feature retrieval take near the same time. So the proposed approach is more efficiency than conventional methods. The running time comparison is shown in Table 2. The proposed method is faster and can be used in real time.

4.3 Different features from different layers. In CNN, one have multiple different convolutional layers, different layers have different feature map outputs. Different outputs has different interpretations. Though the exact interpretations of mid-layer output of CNN is not fully understand, we already have some vague understanding about middle layer feature maps. For the lower level layers, the lower level features was detected, which trying to detect edges and blobs. The trained filters from first convolutional layer is shown in Figure 4.10. And for the higer layers (e.g.fourth

or fifth layers in AlexNet structure), it tends to grab the representation of abstract representation of different training sets. Those several layers are task specific, so it is hard to tell which layer perform the best in theory, due to not fully understand of mid-layer feature maps. So to study which layer is the best for our method, we

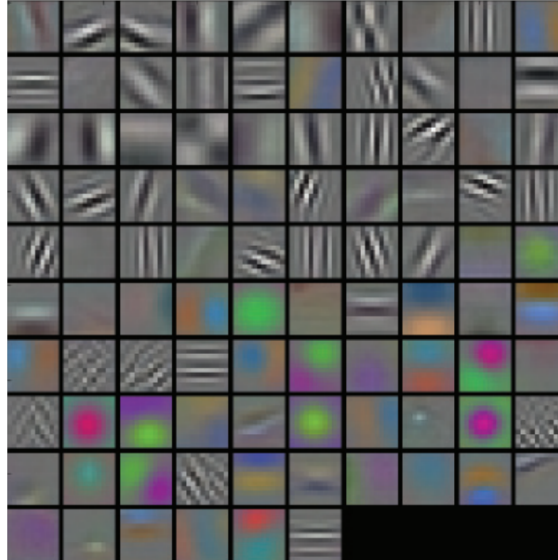


FIGURE 4.10. The filter bank of first convolutional layer. These filters detect edges and blobs.

select last 3 convolutional layers: conv5 and conv4. And we compare the performance for each kind of features with KTH-IDOL2 dataset. The reason we did not use low level layer is those edges and blobs is not efficient to retrieve exact same scenes. The results of all these features is shown in Figure 4.11. From the results we can see that the higher layer do performs better than lower layer. Especially with less candidates results, the difference between results of conv4 is much lower than its counterparts. But with more candidates results, two results tend to get the same results. In the following experiments about post processing, we all use the result list generated from conv5 layer.

4.4 Examination of post processing. The shown results expressed that CNN feature maps can get a good localization performance along with the bag of features model. However, we may got some false matching when only running deep feature

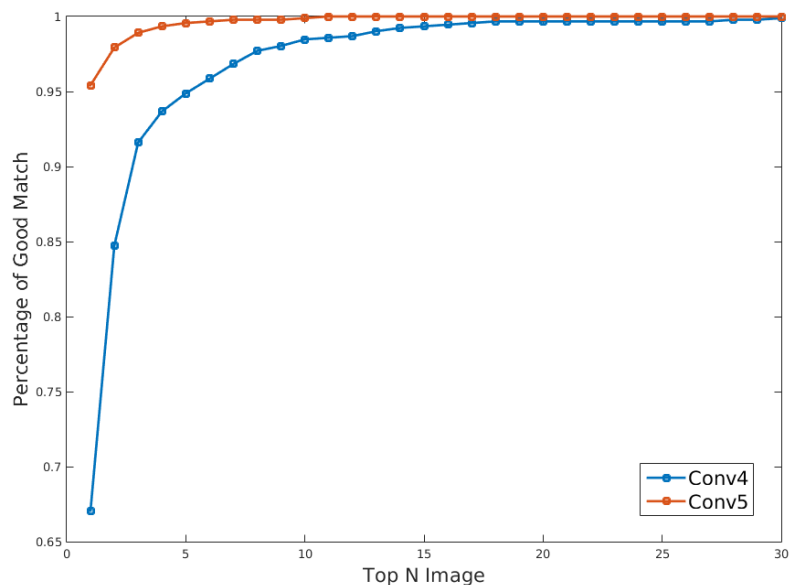


FIGURE 4.11. The results of different layers.

based localization method. From the result of Sec. 4.2, we can see that for two datasets, first stage will reach 100% precision within top 15 candidates. The scores of those candidates are not less than 90% of top score, so we select the candidates whose score is not less than 90% of top score in the same list. Some sample candidates list are shown in Figure 4.12. From such figure, we also can see that our method has the ability to deal with different illumination conditions in some level.

We examined two schemes for post processing stage. For KTH-IDOL2 dataset, the results of applying global features to re-rank result list is shown in Figure 4.13. From the figure, the post processing stage do improve the performance. Geometric Verification has better performance due to its appearance based check. Since global features compress the whole image into one single compact vector, the performance is worse than geometric verification scheme. Meanwhile, CNN based global feature and GIST global feature nearly has the same performance. But for top candidate, CNN based features has better performances. A re-ranking result sample with CNN global features is shown in Figure 4.14.



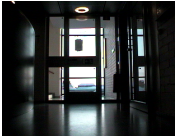
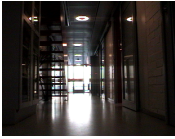



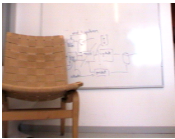




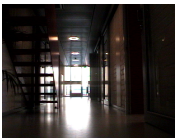
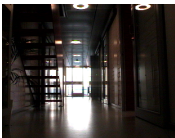
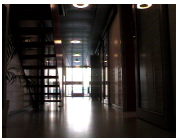
Query	1st	2nd	3rd	4th
 (a)	 bad	 good	 bad	 bad
 (b)	 good	 bad	 bad	 bad
 (c)	 bad	 good	 good	 good

FIGURE 4.12. Sample candidate lists from different positions. Query(a) and Query(c) did not have good matching in top candidate, but has good results in top 2. Meanwhile, Query(b) has a good match in the top ranking.

In Fig. 4.13, the re-ranking involves 15 images. Some of the better results is re-ranked to top results, which presents the efficiency of our scheme.

As for Barus & Holley Lobby dataset, we do the same experiments. In previous experiment, we can see that our approach has the worse performance on this dataset due to homogeneous area in the lobby area and blurring caused by walking vibration. In this case, post processing is more critical and important. Some re-ranking results from CNN global features of one image are shown in Figure 4.15. compared with

The comparison between different post processing method is shown in Figure 23. From such figure we still can see the same tendency with KTH-IDOL2 dataset.

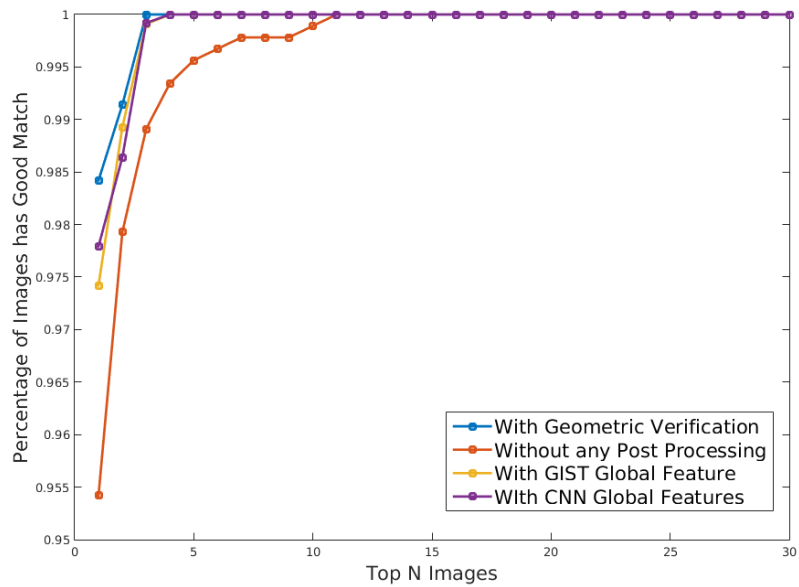


FIGURE 4.13. The experimental results of different post processing schemes, comparing with the results without any post processing.








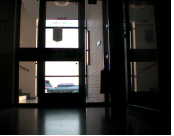
Query	1st	2nd	3rd	4th
				
	Previous			
				
	Reranked			

FIGURE 4.14. The result list before reranking and after reranking.

The geometric verification has the best performance. Global features has worse performance but still has the appreciable improvement. From previous experiments, geometric verification has the best performance, however, it has the largest running

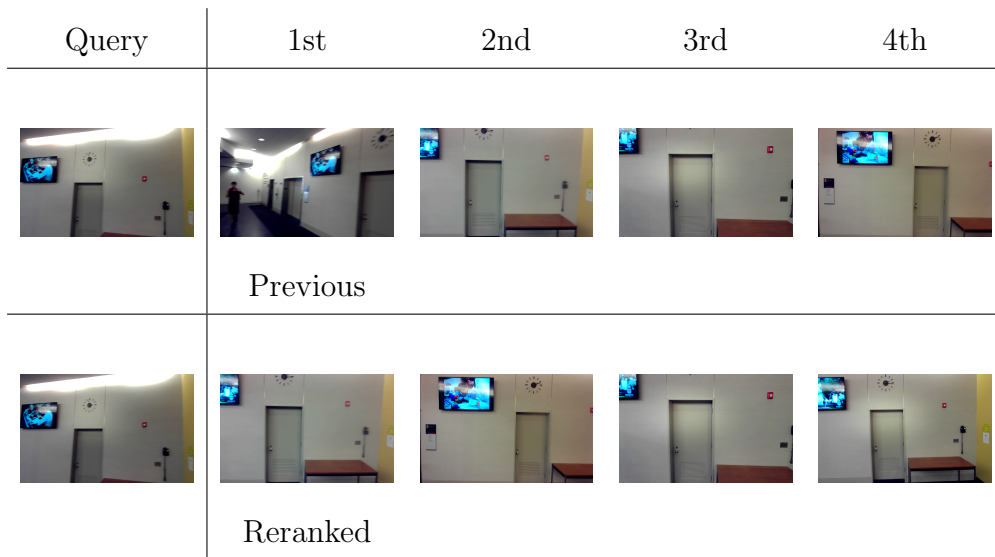


FIGURE 4.15. The result list before reranking and after reranking.

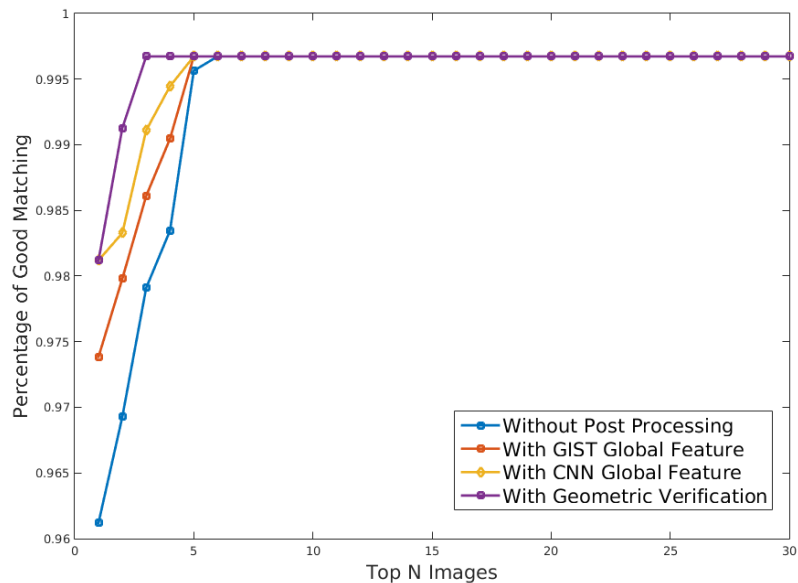


FIGURE 4.16. The experimental results of different post processing schemes, comparing with the results without any post processing in B&H dataset.

time, which make it not applicable in some high demand of running time condition. The average running time of all the methods in shown in Table 4.3. Image localization with geometric verification can rich 0.5Hz running frequency, and global CNN

TABLE 4.3. Image Localization Methods Running Time

Methods & image Size	Average Time per Image
320×240 , with Geometric Verification	1.52s
1280×800 , with Geometric Verification	1.98s
320×240 , with CNN re-ranking	0.85s
1280×800 , with CNN re-ranking	0.86s
320×240 , with GIST re-ranking	0.97s
1280×800 , with GIST re-ranking	1.12s

based re-ranking can rich more than 1Hz. Due to extracting one more features, GIST, compared with CNN, has no benefits in running time. So, these two schemes, fast matching based geometric verification and global CNN based re-ranking, can be used in different conditions.

CHAPTER 5

Conclusion

Bibliography

- [1] A. R. Zamir, S. Ardeshir, and M. Shah, “Gps-tag refinement using random walks with an adaptive damping factor,” in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 4280–4287.
- [2] M. M. Ullah, A. Pronobis, B. Caputo, J. Luo, P. Jensfelt, and H. I. Christensen, “Towards robust place recognition for robot localization,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 530–537.
- [3] J. Luo, A. Pronobis, B. Caputo, and P. Jensfelt, “Incremental learning for place recognition in dynamic environments,” in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 721–728.
- [4] A. R. Zamir and M. Shah, “Image geo-localization based on multiplenearest neighbor feature matching using generalized graphs,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 8, pp. 1546–1558, 2014.
- [5] T. Sattler, B. Leibe, and L. Kobbelt, “Fast image-based localization using direct 2d-to-3d matching,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 667–674.
- [6] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt, “Image retrieval for image-based localization revisited.” in *BMVC*, vol. 1, no. 2, 2012, p. 4.
- [7] D. Nistér, “Preemptive ransac for live structure and motion estimation,” *Machine Vision and Applications*, vol. 16, no. 5, pp. 321–329, 2005.
- [8] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [9] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree,” in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, vol. 2. IEEE, 2006, pp. 2161–2168.
- [10] Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series,” *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.
- [11] D. Gálvez-López and J. D. Tardos, “Bags of binary words for fast place recognition in image sequences,” *Robotics, IEEE Transactions on*, vol. 28, no. 5, pp. 1188–1197, 2012.

- [12] J. Sivic and A. Zisserman, “Video google: A text retrieval approach to object matching in videos,” in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 1470–1477.
- [13] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.
- [14] G. Schindler, M. Brown, and R. Szeliski, “City-scale location recognition,” in *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*. IEEE, 2007, pp. 1–7.
- [15] E. Kalogerakis, O. Vesselova, J. Hays, A. A. Efros, and A. Hertzmann, “Image sequence geolocation with human travel priors,” in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 253–260.
- [16] M. Cummins and P. Newman, “Fab-map: Probabilistic localization and mapping in the space of appearance,” *The International Journal of Robotics Research*, vol. 27, 2008.
- [17] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, “Visual place recognition with repetitive structures,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 883–890.
- [18] M. Jain, H. Jégou, and P. Gros, “Asymmetric hamming embedding: taking the best of our bits for large scale image search,” in *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 2011, pp. 1441–1444.
- [19] T. Sattler, B. Leibe, and L. Kobbelt, “Improving image-based localization by active correspondence search,” *Computer Vision–ECCV 2012*, pp. 752–765, 2012.
- [20] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua, “Worldwide pose estimation using 3d point clouds,” in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 15–29.
- [21] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, “Learning deep features for scene recognition using places database,” in *Advances in neural information processing systems*, 2014, pp. 487–495.
- [22] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “Cnn features off-the-shelf: an astounding baseline for recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813.
- [23] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays, “Learning deep representations for ground-to-aerial geolocation,” in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 5007–5015.
- [24] Y. LeCun, K. Kavukcuoglu, C. Farabet *et al.*, “Convolutional networks and applications in vision.” in *ISCVS*, 2010, pp. 253–256.

- [25] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 430–443.
- [26] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” *Computer Vision–ECCV 2010*, pp. 778–792, 2010.
- [27] S. Korman, D. Reichman, G. Tsur, and S. Avidan, “Fast-match: Fast affine template matching,” in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 1940–1947.
- [28] D. Arthur and S. Vassilvitskii, “k-means++: The advantages of careful seeding,” in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.
- [29] J. Luo, A. Pronobis, B. Caputo, and P. Jensfelt, “The kth-idol2 database,” *KTH, CAS/CVAP, Tech. Rep.*, vol. 304, 2006.
- [30] C. Wu, “Visualsfm: A visual structure from motion system,” 2011.