Image Based Localization

BY

MAHDI SALARIAN
B.Sc., University of Guilan, 2004
M.Sc., University of Mazandaran, 2006

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Chicago, 2018

Chicago, Illinois

Defense Committee:

Rashid Ansari, Chair and Advisor
Miloš Žefran
Ahmet Enis Cetin
Hulya Seferoglu
Ajay Kshemkalyani, Computer Science

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor Prof. Rashid Ansari for the continuous support and trust in my Ph.D related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study. Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Dan Schonfeld, Prof. Miloš Žefran Prof. Hulya Seferoglu, Prof. Ahmet Enis Cetin and Prof. Ajay Kshemkalyani, for their insightful comments and encouragement. Their supportive guidance helped me navigate through the problems and incented me to widen my research from various perspectives. Many thanks to the Center for Urban Transportation Research at the University of South Florida and the Elizabeth Morse Genius Charitable Trust for supporting part of my research. I had great experience as a member of Multimedia Communications lab at the University of Illinois at Chicago (UIC). The collaborations and working over the years helped me grow, personally and intellectually. I thank my co-author, Nick Iliev, for his contributions in the projects. Many of the ideas in our work emerged from our discussions and teamwork. Last but not the least, I would like to thank my family: my parents and my brother and sisters for supporting me spiritually throughout my work in UIC and my life in general.

MS

PREFACE

This dissertation is an original intellectual product of the author, M. Salarian. All of the work presented here was conducted in the Multimedia Communications Lab at the University of Illinois at Chicago.

The results of our research have been previously published (or will be published) as an article in the IEEE Transactions on Multimedia and several conference and symposium publications: IEEE International Symposium on Multimedia (ISM'17) (Salarian et al., 2017), ICASSP'18, ISM'16 (Salarian and Ansari, 2016), (Salarian et al., 2016), IntelliSys'15 (Salarian et al., 2015). The copyright permissions for reusing the published materials are given in Appendix C.

Mahdi Salarian
August 16, 2018

CONTRIBUTION OF AUTHORS

The overall contribution of this research is the development of algorithms and efficient implementation to compensate for GPS ineffectiveness in dense areas of cities and especially for accurate localization of a pedestrian walking on a street sidewalk where GPS accuracy suffers due to proximity to walls and buildings. Specific contributions and their publication in different outlets are described below.

The first major contribution that is described in Chapter 4 is a method based on utilizing uncertainty of the GPS to narrow down the search space for a query and thereby reduce the number of images that should be searched for a particular query. Research described in Chapter 4 has been published in IEEE International Symposium on Multimedia, ISM16 (Salarian and Ansari, 2016) and IntelliSys'15 (Salarian et al., 2015). I was the main investigator for this research. In order to evaluate our method, we created a dataset of images extracted from Google Street View (GSV). I also collected a set of query images captured with different type of cell phones along with other required information such as GPS tag, GPS uncertainty, and camera heading. Other researchers previously investigated limiting the search space by considering a fixed radius obtained from the maximum error of the GPS, however the use of GPS uncertainty in adapting search space was not reported before.

The second main contribution consists of methods proposed for optimal selection of the images for the Structure From Motion method along with an algorithm for estimating the query location by employing a new method of coordinate transformation covered in Chapter 5. Different parts of this chapter have been published in IEEE International Symposium on Multimedia (Salarian et al., 2016) and International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2018) and IEEE

CONTRIBUTION OF AUTHORS (Continued)

Transactions on Multimedia. I was the lead investigator of the research in those papers. Nick Ileiv was involved in Camera coordinate transformation section. I was responsible for designing the image retrieval system and optimal selection of images as well as formulating coordiante transfer and all simulations in this work. A. Enis Cetin was adviser in formulating the image selection framework.

The third major contribution described in Chapter 6 is the improvement of the image retrieval system by utilizing the scale of feature descriptors in the vector representing an image in BOF framwork. Some of the content of this chapter has been published in IEEE International Symposium on Multimedia, ISM 2017 (Salarian et al., 2017). We have also submitted a journal manuscript based on the content of this chapter to IEEE Transactions on Pattern Analysis and Machine Intelligence. I was the main investigator for this research. Mehdi Sharifzade was involved in the early stage of the problem formulation. All mentioned research in the contributions are supervised by my advisor, Rashid Ansari.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF TABLES

‘

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

FIGURE **PAGE**

LIST OF ABBREVIATIONS

CBIR	Content Based Image Retrieval
BOF	Bag Of Feature
SFM	Structure From Motion
GMCP	Generalized Minimum Clique Problem
SF	San Francisco
GSV	Google Street View
RANSAC	Random sample consensus
GPS	Global Positioning System
UIC	University of Illinois at Chicago

SUMMARY

The proposed research was motivated by the need for developing assistive technology to provide the capability for people who are blind or visually impaired to navigate outdoors, un-assisted, as sighted people can do. This goal was set by a problem posed to us by some members of the National Federation of the Blind. Numerous navigation methods are available to aid the blind and they are primarily based on the use of GPS technology. However, they are often ineffective in very dense area of cities due to GPS signal degradation. The research in this dissertation seeks to devise a key ingredient to support the objective of developing a system to enable ease of outdoor navigation especially for crossing streets. While the overall problem of navigation involves many issues such as traffic and pedestrian signal recognition, vehicle recognition and their velocity estimation and finding their distance, this dissertation is largely focused on research to enhance accurate global position estimation that is essential for navigation. While most of the recently proposed methods to estimate position utilize GPS for outdoor applications, some methods consider the use of additional sensors such as compass and camera to achieve better results. One thrust of this dissertation research is to develop robust methods for accurate localization using mobile devices. The main goal in this effort is to develop algorithms and efficient implementation for compensating GPS ineffectiveness in dense areas of cities and especially for accurate localization of a pedestrian walking on street sidewalks where GPS accuracy suffers due to proximity to walls and buildings.

SUMMARY (Continued)

We seek to overcome the shortcomings of current methods by adopting approaches that venture beyond available pure retrieval methods. We augment our knowledge of the approximate position of query image by incorporating additional sensor information to adaptively select the search area.

We use the hitherto unutilized information of the Error Positioning Estimation (EPE) as an aid to address issues of complexity and accuracy in the large-scale geo-tagged dataset to narrow down the search space for a query and thereby improve the efficiency by reducing the number of images that should be searched for a particular query. To compare our proposed method based on adaptive selection of search radius, we created our query for the Chicago dataset by including GPS uncertainty, so that each query image is provided with the associated EPE information. To test the system in a real-world situation a client server application is developed, where the client is an Android Application responsible for acquiring all necessary information such as query image, position data, EPE, Pitch, Yaw and Roll which is sent to a server through TCP-IP protocol.

In our second contribution we exploit availability of multiple images of the scene captured from different perspectives to perform the location coordinate estimation. A new approach based on Structure From Motion (SFM) is proposed that shows better performance in terms of accuracy. Since we know that even successful retrieval returns only the position of the vehicle from which the image was captured and not the actual position of the query, we propose a method to refine the query position relative to some retrieved images. While the query image in a successful retrieval may be similar to the stored dataset images, positions from which the images are captured are not necessarily close. The error range has been reported to be as large as 30 meters that would not be useful for navigation especially for the blind.

SUMMARY (Continued)

For selecting multiple images with similar content we examine different strategies. We propose a method to optimally select images to achieve higher convergence rate in the SFM process. The criterion for selecting image in our proposed method is that the selection should be similar not only to the query image but also to all images in the final selected set. To evaluate and compare our proposed algorithm with other results the San Francisco dataset is used. The final localization error in most of the cases is less than five meters which is significantly better than other reported results and suitable for navigation.

In our third major contribution to improve the performance of the system the image retrieval engine is modified by considering more spatial information in vectors representing the images. In this context, instead of frequency of visual words we consider the scale of feature descriptors in a vector representing images. The new vector, called Bag Of Scale-Indexed Features (BOSIF), does not impose higher memory usage than that required in Soft Assignment while significantly improving the recall rate. We also propose a hybrid method that combines our proposed method with the well-known Adaptive Assignment algorithm and show how the hybrid method provides recall performance that is better than either method while maintaining the level of memory usage.

CHAPTER 1

INTRODUCTION

In this chapter the motivation and research goal of the dissertation research are first presented. An overview of the research approach is then provided. Next the main contributions of the thesis are summarized. Finally, the organization of the content of the dissertation is described.

1.1 Motivation and Research Goal

The estimate of the location of a mobile device is valuable in a variety of different applications such as navigation and augmented reality. Accurate location is especially critical for blind or visually impaired pedestrians to safely navigate outdoors in cities. As a result accurate localization has recently been a very active area of research (Schroth et al., 2011), (Liu et al., 2012). Even though traditional approaches that utilize the GPS module data or the distance from cellular towers are useful for performing this task, the adequacy of this performance depends mostly on satisfactory access to the satellite signal. GPS information is usually satisfactory in many applications when the device has a clear view of the sky to get the signal from at least 4 satellites. For example, it is difficult to obtain accurate localization using a GPS-equipped device carried by a pedestrian who is moving slowly on a street sidewalk close to tall walls and buildings. This difficulty is generally most pronounced in dense urban areas. However, the GPS errors of mobile phone are rarely greater than 100 meters (Schroth et al., 2011). Figure 1 shows a pedestrian trajectory in downtown Chicago in which only the GPS data has been utilized. The red trace

shows the trajectory recorded by a cell phone while the blue trace is ground truth. It is clear that the peak error (maximum difference between red and corresponding blue traces) is more than even a city block causing difficulty in navigation. As a result, significant research effort has been directed at finding solutions to compensate GPS shortcomings. This is because accuracy is vital for applications in which people are highly dependent on Location-Based Services (LBS) technologies, such as blind and visually impaired people. This effort has sought to exploit as much information from the sensors as is available in mobile devices (Liu et al., 2012), (Guan et al., 2013a). A large part of this effort has focused on using a camera that is an essential part of any smart phone or mobile device. It relies on the notion of getting accurate position of a query image generated by the camera using image retrieval methods. Researchers have investigated techniques for seeking the best match for a query image in a database of Geo-tagged images with accurate GPS coordinates. Some of these investigations have relied on the use of datasets created by researchers, while others have used databases of images which are publicly available such as Google Street View (GSV) and Flicker. Sample images extracted from GSV service are shown in Figure 3 and Figure 4. The purpose is to search for the best match as shown in Figure 2. Then another step can be applied to improve location accuracy which is discussed in detail later.

1.2 Overview of the Research Approach

Since the core of the system is image retrieval, we tried to implement some of successful algorithms for image retrieval while keeping in mind how the other sources of data can be utilized to improve the results. We can look at the problem from a different angle to get better results in terms of accuracy, time, memory usage, channel etc. To simulate a real-world situation, different prototype systems were devised. For example, to understand how a wearable system can be used, we developed a system such

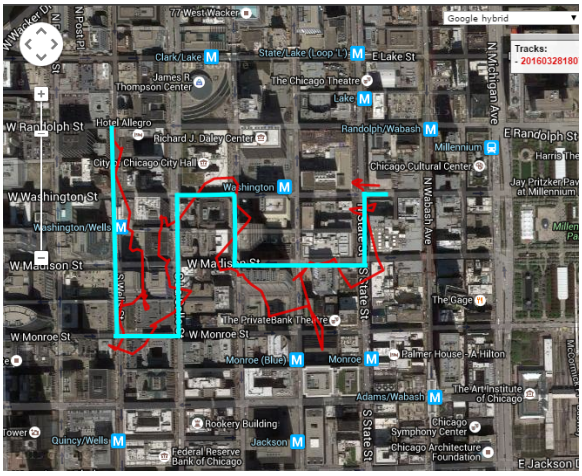


Figure 1: Sample trajectory saved by a smart phone (pure GPS) shown by red and ground truth by Blue

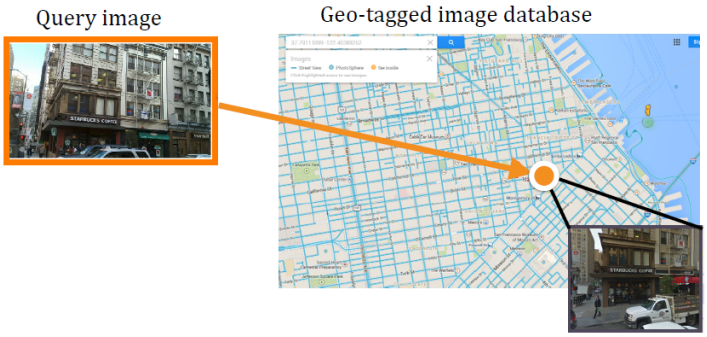


Figure 2: procedure of localization through image

as that shown in Figure 5. It contains a camera, an Inertial Measurement Unit (IMU) including Gyro, Accelometer and Compass, GPS module and Arduino to read related data and send them to a laptop. Before sending those sensor data to laptop, Data from IMU should be fused by an algorithm such as

Figure 3: A sample panorama Google Street View

Figure 4: A sample panorama Google Street View

Kalman filter in Arduino to acquire the pitch, yaw and roll of the camera and relayed to a labtop. We use a code provided by IMU factory for fusing. Then, a Matlab-based program reads the data. Figure 6

Figure 5: A wearing prototype for acquiring different sources of data

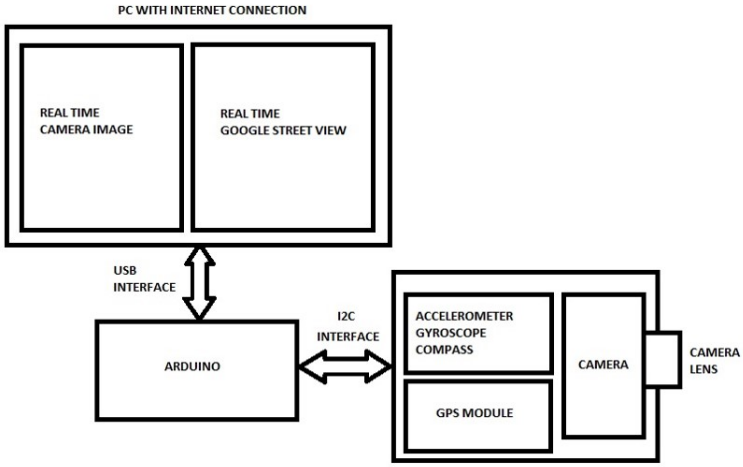


Figure 6: Block diagram of the prototype

shows the block diagram of the system. After testing this prototype, we found that sensors and GPS are reliable most of the time, however, their shortcoming especially in urban areas with tall buildings makes navigation a difficult task for pedestrians.

Consider a scenario where a person is walking in an area that has a clear view of the sky. In this situation the GPS returns position data with minimum error. For example consider Figure 7 and Figure

Figure 7: Image taken by our wearing system while other sensors data is recording

Figure 8: Image extracted from Google Map API by sending GPS and sensors data

8. The first picture is taken automatically using the prototype while the second picture is obtained when sensor data including pitch, yaw and roll along with GPS data is fed to a Google street view API simultaneously. It can be seen that images are captured from similar locations. The only difference is the GPS error that seems to be less than 10 m. However, achieving this result in dense areas of cities such as Chicago or San Francisco is usually not possible .

So our goal in this research is to incorporate all available data and use it in conjunction with a powerful image retrieval system to improve localization accuracy. Following the dataset search, the best match for a query image is found which leads to better estimates of the coordinates of query. Existing image retrieval approaches have been found to be powerful in getting successful matches especially when images have adequate texture. The key tools employed in these methods are based on scale-invariant features such as SIFT (Lowe, 2004) and SURF (Bay et al., 2008a). Other variants of SIFT such as Dense SIFT or ASIFT (Morel and Yu, 2009) that are proposed recently may lead to better results in terms of accuracy but impose more computational complexity. This would affect the performance of the entire system when we are dealing with a huge number of images in the data set. Suppose we are working on a city-scale data set such as San Francisco dataset with more than one million images while the size of each feature description is around 300K bytes. To have all of those features, about 300 Gigabytes of RAM is needed which is not possible for most computers. Searching among such a huge volume of data is a time-consuming task and makes it impossible for applications such as ours that are seeking fast feedback. So, one aspect of our work is to consider a cloud-based service to overcome the time and memory limitations. Although some of past research has focused on implementation of this type of applications in mobile devices, the majority of research utilizes a server to which data is sent for processing. This is acceptable since new generations of network are fast and reliable for this application. So in our research we have designed a client-server program to use the power of a server. The client that is an Android application is responsible for sending the query image along with data from other sources to a server through a reliable protocol (TCP-IP) while the server can find the best match for the query, find the related position, and send the result back to client. This result can be used

in navigation or similar applications. In the process of finding the best match, comparing features of a given query directly in a dataset is not recommended due to the time constrains. Therefore a more sophisticated approach needs to be employed as an image search engine. Most image search engines are based on the Bag of Words (BOW) algorithm. In this method each feature is quantized to a visual word so each image can be represented by a vector of visual words. To create a visual word all or a portion of features of all training images in the dataset should be fed to a clustering algorithm such as K-Means (Hartigan and Wong, 1979). Due to the huge size of the data, we were unable to utilize K-Means. So an alternative form called Hierarchical K-Means which is more practical for large scale data is used . The same procedure should be used for a given query image. Then the simplest method for image selection is to compare the vector of the query with all vectors of data set images. The common metric to verify the similarity of those vectors is cosine distance or inner product of two vectors after normalization. Since this process may need considerable time, an alternative algorithm called Inverted Index can be employed to return the result faster. In the Inverted index description, all images that contain a particular visual word are known. At query time, dataset images that have common visual words with the query receive higher weights. For each dataset image the cumulative weight from present visual words of the query is a measure for similarity evaluation. Besides image retrieval pipline, data from other sources have been extracted from available sensors. We are trying to leverage all relevant information that is available since the performance of the overall system degrades with increasing the size of the data set. On the other hand, increasing the size of dataset may lead to reducing the chances of finding the correct match. This can be overcome by having prior information about the approximate position which can be used to narrow the search space down. For example, in (Zhang et al., 2011) the database is split into

overlapping regions for the search. That method is attractive since even clustering features from city-scale database is very time-consuming and needs a large memory. Relying purely on image retrieval techniques for outdoor navigation is limiting. Instead, we may leverage additional media information including available sensor data such as GPS coordinates available in the new generation of phones. In addition, noting the fact that the GPS error can be as large as 100 meters in some locations, we may use query images generated by smart phone cameras to aid in refining the localization where the GPS is not accurate enough. One piece of information that is very valuable in location refinement is the uncertainty in GPS-based location that is accessible using the method described in (Massoud Sharif and others,). In our proposed method, we have exploited additional media information about Estimated Positional Error (EPE) to limit the search area for image retrieval. Our method is also adaptive in a sense that it decides about the context in which it is better to use GPS data by itself and to use retrieval-assisted refinement to achieve a better result. One other key piece of accessible sensor information we use is the camera heading to limit the search to lie within a selected angle of view. Unfortunately, publicly available datasets cannot be considered for evaluating the idea of using EPE since the EPE is not available in those datasets and has not been exploited in previous research. Therefore we created a dataset of images captured the Chicago mostly from dense regions of the downtown where the localization error is high. We employed Google Street View (GSV) service API to download images. Then, we developed an image retrieval system and showed how our proposed method based on EPE improves sytem performance in recognizing relevant image for the query by limiting the search area and consequently the number of the dataset images fed to the image retrieval pipeline. In addition to methods we used to narrow down search area, we also worked on image retreival system to could get better matches for a query image.

For example, spatial information of features contains information that is important for finding similar images. In regular BOF representation of images no spatial information is inserted. We re-rank dataset images by applying a geometry verification process between the query and dataset images by employing algorithm such as RANSAC. This step usually should be employed as a complementary step at the end of the retrieval process and on only limited number of the images with higher similarity.
Another proposed method we discuss is incorporating scale information of the feature descriptors directly in the vector representing an image. In this method we create a vector called BOSIF containing the scale of visual words (associated features) instead of the frequency of the visual word without increasing the memory usage. The performance of our algorithm has been compared with some of the newly proposed methods in recent research and shown to perform better in terms of recall.

Although image retrieval is a vital part in our research, we need a method to compute the location of the query precisely and not just by simply considering GPS tag of the most similar image in the dataset. We propose a method to optimally select a group of images returned by the image retrieval pipeline with the highest intra-similarity. To extract this set, we define a similarity measure that not only takes into account similarity between each individual image and the query, but the similarity between all members of the set. We then feed a selected set of more than three images to the Structure From Motion (SFM) procedure. Upon convergence of the SFM, we can compute the query GPS tag easily by a a method proposed in this research. Experiment results show that the location accuracy of our proposed method for most of the sample queries is in the range that is acceptable for our localization purpose.

Different chapters of this dissertation cover different aspects of our work as described in detail in the section below.

1.3 Summary of Research Contributions

The overall contribution of this research is the development of algorithms and efficient implementation to compensate for GPS ineffectiveness in dense areas of cities and especially for accurate localization of a pedestrian walking on street side. The first major contribution that is described in Chapter 4 is the utilization of uncertainty of the GPS to narrow down the search space for a query and thereby reduce the number of images that should be searched for a particular query. Research described in Chapter 4 has been published in IEEE International Symposium on Multimedia, ISM16 (Salarian and Ansari, 2016) and IntelliSys'15 (Salarian et al., 2015).

The second main contribution consists of methods for optimal selection of the images for the SFM process along with an algorithm for estimating query location by employing a new method of coordinate transfer covered in Chapter 5. Different parts of this chapter have been published in IEEE International Symposium on Multimedia (Salarian et al., 2016) and International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2018) and IEEE Transactions on Multimedia (Salarian et al., 2018).

The third major contribution described in Chapter 6 is the improvement of image retrieval system by utilizing the scale of feature descriptors in the vector representing an image in BOF framwork. Some of the content of this chapter has been published in IEEE International Symposium on Multimedia, ISM 2017 (Salarian et al., 2017). We have also submitted a manuscript based on the content of this chapter for publication in a journal.

1.4 Dissertation Content Organization

Our thesis is organized into three parts and seven chapters. Further details of each part and each chapter is presented below.

Part I, Preliminaries: This part presents an introduction to the research problem and goal and a review of related works.

Chapter 1: This chapter gives a general overview of the problem and motivation to the reader. Then the challenges for our particular goal in existing approaches are explained. Later, we present our contribution for solving the addressed problem in this chapter.

Chapter 2: In this chapter relevant related works about determining device location using image retrieval techniques are presented. The most popular feature descriptors and their properties are then discussed. Next, common methods for clustering, indexing, and related concepts are described. We also explain how an image can be represented as a vector and how those vectors should be compared. We review multiple methods for finding similarity between vectors. Later, we show how the geometric content of two images using RANSAC can be verified to double check the similarity between a query and a limited number of most similar images in dataset.

Chapter 3: This chapter provides insight on the core of the system used in different chapters as an image retrieval engine by discussing recent image retrieval techniques and the results of our implementation on the Oxford dataset. We have applied a basic image retrieval technique by utilizing most recent algorithms for each part and we evaluate the influence of the different parameters such as number of clusters (visual words) and similarity measure on the final retrieval results.

Part II, Proposed solutions: This part contains our contributions for improving image based localization

Chapter 4: This chapter introduces our proposed method for utilizing GPS uncertainty and camera heading in the image retrieval pipeline. After describing the procedure for creating the dataset for the city of Chicago and extracting GPS error for our query images, we find the optimal search radius, and show how an adaptive search radius improves the performance of image search task based on the fixed radius extracted from maximum GPS error on two datasets with different size.

Chapter 5: This chapter introduces a method to compute a more accurate location of the query camera. To be able to compare our final result to other reported results, a publicly available dataset for landmark recognition has been used. Then the result by considering prior knowledge about the query position is shown. We have emphasized that for our special problem it is recommended to use any source of data that is available and enhance the accuracy and system reliability. Although there are different schemes for image retrieval and we believe they will improve the result, our proposed algorithm based on SFM can be considered as a complementary method for all of those to achieve higher accuracy in the estimated location. This chapter contains two contributions. First, we propose a system with integrating different ideas to get even higher localization accuracy. This method that employs Structure From Motion (SFM) shows highest accuracy in comparison with other proposed methods. In order to accurately estimate the query location, we proposed a method that utilizes more than three images along with the query for the SFM process. A subset

of images returned by image retrieval engine should be incorporated in the SFM process. Since an image retrieval engine returns multiple images, we need to determine which of images are the best option for the next step. It is important to select a set of images that leads to a higher convergence rate in the SFM process. Another point that should be taken into account is incorporating only images with distinct GPS tags. This is necessary since the coordinate transfer method formulated in our work needs more than three images with distinct GPS tags. So we propose an optimal solution for selecting the best subset from set returned by image retrieval.

Chapter 6: This chapter describes a new method to improve the image retrieval engine. Our proposed method incorporates spatial information of the image in the vector that represent an image. Unlike available approaches, we embed the scale information of feature descriptors directly in the vector without increasing the memory usage. After comparing our proposed method with available algorithms, we propose a hybrid method that achieves higher recall rate while the memory usage remains comparable.

Part III, Conclusion and future work: This part contains open problems and concluding remarks.

Chapter 7: A concluding summary of the thesis is presented in this chapter. We discuss a few possible extensions of our work as well.

1.5 Reader's Guide

The thesis is organized in such a way that each chapter provides foundation for the following chapters. As such, it is expected to be read in the order presented. However, in some parts of the dissertation, a brief preview of the results has been provided to motivate the approach and provide the reader with additional insight.

CHAPTER 2

BACKGROUND AND NOTATION

Recent computer vision advances have made it possible to search for an image similar to a query in social sharing websites like Flickr or user generated datasets with sufficient reliability and for many applications (Li et al., 2013), (Guan et al., 2013b), (C. Yan, 2018). A noteworthy application of this capability is searching a massive number of Geo-tagged images on the internet to find the location of a query image (Song et al., 2016), (Yu et al., 2011), (Salarian et al., 2016). A variety of methods have been proposed to do this. For instance Torralba(Torralba et al., 2003) employs low-dimensional global image representation along with Hidden Markov Model. The shortcoming of this approach is the lack of robustness to clutter and orientation. Other newer research such as Cipolla (Cipolla et al., 1999) utilize vanishing points in 2D on a small set of images. Although this approach rectifies features based on vanishing points, It suffers from loss of 3D information.

Reitmayr and Drummond (Reitmayr and Drummond, 2007) utilized an edge-based method to get street facades based on a 3-dimensional method. The most efficient and accurate approach uses Content-Based Image Retrieval (CBIR) techniques relying on features such as SIFT and its variants. Some effective approaches frequently used in CBIR systems are Bag of Features (BOF) (Sivic and Zisserman, 2003), (Philbin et al., 2007), (Nister and Stewenius, 2006), Fisher Vector (FV) (Jegou et al., 2012), (Jegou et al., 2011) and vector of locally aggregated descriptors(VLAD) (Jégou et al., 2010). In these methods all feature descriptors are quantized to visual words with a clustering algorithm like K-Means. An image is represented by a histogram of a number of visual words and each image in a database has

its own histogram. For finding the best match, the histogram of a query image is compared with all histograms in database.

There are different measures for finding similarity such as the inner product of two BOF vectors or specific distance functions (Jing et al., 2013), but a widely used procedure is the inverted file (Witten et al., 1999). Some researchers have focused on clustering to find an efficient quantization technique for assigning each feature descriptor to a visual word (Schindler et al., 2007).

For example, Soft Assignment (SA) instead of Hard Assignment (HA) has been proposed to compensate for incorrect assignment of a sample feature (Philbin et al., 2008). Some have tried to select more distinctive features (Knopp et al., 2010), (Gronat et al., 2013), (Chen et al., 2013), (Turcot and Lowe, 2009) while others have evaluated how repetitive structures influence the ultimate result (Park et al., 2009), (Torii et al., 2013). This is not necessarily the last step in the retrieval. Most of the methods select more than one candidate for a match in this step. An additional step, called homography verification performed by applying algorithm such as RANSAC (Fischler and Bolles, 1981) and its variants, (Torr and Zisserman, 2000), (Moisan et al., 2012) is used to re-rank candidates. In fact, this step compensates for the weakness of image retrieval schemes based on BOF where the geometric information of images is ignored.

Other studies such as (Liu et al., 2012), (Guan et al., 2013a) have proposed a method of using inertial sensor information and BOF to get more accurate results. Specifically in (Guan et al., 2013a) prior knowledge of the approximate location from the cell towers is used to limit the search to the cellular area. Specifically in (Guan et al., 2013a), the rough prior knowledge of location is inferred from the cell towers close to the mobile phone and that is used to limit the search to the cellular area. The common

approach to using prior knowledge of position is to partition the city scale database into suitable smaller sub-regions. To avoid issues at sub-region boundaries, those sub-regions are partitioned in a way to have overlap. Search for an image closest to the query for the region between sub-regions should be done for all involved regions. Although a unique training can be considered for creating a vocabulary tree, it would be better to exploit different vocabulary trees for each region. This approach imposes more computational cost and memory. To avoid those issues, a single vocab tree and smaller regions for the search should be selected. So work like (Zhang et al., 2011) has considered sub-regions with different areas and proportional to density of regions in terms of number of images. This is because usually the number of images in dense regions of cities is more than the number in an urban area. Although this method has shown reasonable performance it should however be noted that the cellular area is much larger than the area corresponding to uncertainty of the GPS. We emphasize that all the above-mentioned approaches do not exploit the information that is accessible about the uncertainty of GPS and as a result they consider a much larger region for the search. So one thrust of our work is to extrac this available value either from our prototype or from smart phones and find the optimum extent of the search space that is proportional to this value (Salarian et al., 2015) and (Salarian and Ansari, 2016). To be able to implement our proposed method we need a query together with related GPS uncertainty that is not available in other famous landmark recognition datasets. So we have created our database in a way that addresses our need..

After finding the best match we can do better than just assigning the GPS tag of the best match to the query coordinate. To get more accurate result suitable complementary methods should be utilized.

Most of those methods employ structure from motion (SFM) for 3D scene reconstruction followed by camera coordinate registration.

Besides the approaches mentioned earlier, learning-based methods can be employed in each part of our project. For example the image search engine can be developed by deep learning and convolutional neural network (Krizhevsky and Hinton, 2011).

In work that is not part of this dissertation, we have been working on training an architecture based on Siamese model to extract features and use them for the image retrieval pipeline. Work is underway on the model to reach reasonable accuracy that fits our needs for the localization. Moreover, Deep learning based methods can be used in camera pose estimation (Kendall et al., 2015) and SFM methods to estimate structure of 3D objects (Tung et al., 2017), as well.

However, CBIR and SFM methods based on engineered feature such as SIFT used in this dissertation are found to work well for the accurate location estimation.

2.1 Image search

Image search is the key problem that we address in this research. Figure 9 represents an example of an image search application. This problem is an ongoing and recurrent one and there is a vast body of work describing solutions and efforts to improve their performance in terms of accuracy, memory usage, time and channel. Given a query image, consider the problem of finding an image closest to the query among a dataset with enormous number of images. The query image can be an object or scene or even a vehicle. The goal can be to recognize an object, scene category or type of vehicle. In our project the goal is to find the best match for a query image taken in an arbitrary place and use it to estimate the global position of the camera. A generalized version of this problem is to find the trajectory of a

moving camera in street. This task that is known Visual Odometry can be fused by landmark recognition to achieve better result. In Visual Odometry the system uses a sequence of images taken by a unique camera periodically every fraction of a second. So the process for finding features and matching is easier. This is because the content of image is going to share some common features even in abrupt changes by increasing the number of frames in a second. For place or landmark recognition we need to deal with different images, query and dataset that are taken with different cameras, illumination, season and so on. So more confusing features occur and the matching process would be more difficult. It is better to acquire the query image with the camera capturing the front view of a building rather than along the length of a street. This is because the information and texture is inadequately distinctive when the image is captured along the street. Usually there many distracting non-permanent objects on the street that make it difficult to get a good match. The pipeline for finding the similarity between a query and a set of dataset images is usually similar to the procedure described in subsequent sections.

2.2 Image Based Localization Pipeline

The main task in image-based localization is to search for the best match for a query image among a huge image dataset. This process is called image retrieval and is closely related to an object recognition problem. Although extensive effort was undertaken during last decade, a successful strategy proposed by Zisserman and Sivic (Sivic and Zisserman, 2003) is considered as a standard and most acceptable scheme for related tasks. Unlike earlier approaches that have tried to directly find the match between a query image and a dataset containing omnidirectional panorama image, the standard pipeline employed a model from text processing called Bag Of Words. In this model, described in the following section, each document (here image) is represented by a vector. The process that is applied on all database im-

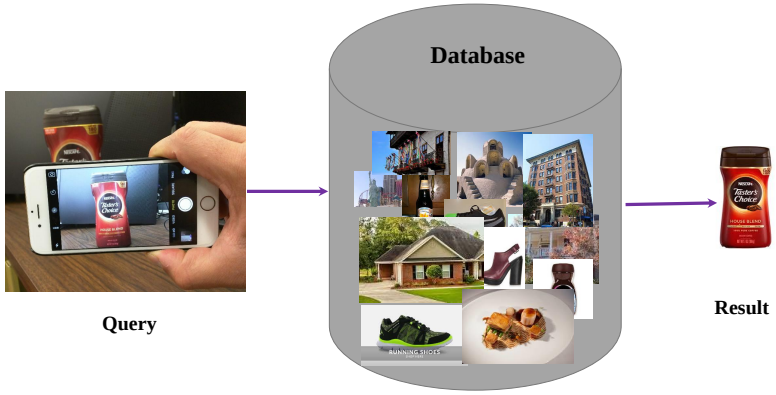


Figure 9: Basic image search system

ages (training) and query includes feature extraction, clustering and quantization of feature descriptors to visual words, creating a Bag of Word vector, finding similarity using a metric followed by geometry verification. Different algorithms have been proposed for each of mentioned steps. Since the number of images in a database is huge for covering a big city, the retrieval process should be efficient in terms of time. Moreover it should be accurate and reliable since we are going to use it for compensating GPS shortcoming to help blind and visual impaired people for crossing street with more safety. So we may consider other sources of information such as different sensors to improve the result. As mentioned before, a complementary step for achieving a better estimate of the position would be required. Those algorithms need to find the Fundamental matrix between a query and the best matches or create a 3D

model of scene using Structure From Motion (SFM). The next sections describe the necessary steps in a standard pipeline.

2.2.1 Feature Extraction

To have an efficient description for an image we have to express each image through its features. Typically each image can have thousands of features. In this scenario, comparing two images consist of comparing their features. Here the assumption is that if two images contain the same objects, they exhibit the same features and the overall similarity measure would be proportional to the number of common features in the two images. For tasks such as landmark recognition, strong features should be utilized since the result should be robust to a change in view point, scale, illumination difference, and so on. We note that a query image is acquired mostly with cellphone cameras while dataset image are acquired with cameras mounted on a vehicle with a different camera model from a query camera. Moreover representing each image by features is more reliable since two image may just have a common region. Although earlier methods based on color histogram showed reasonable result for changes such as view point, they were not efficient enough in dealing with large databases. This is because color histogram is not distinctive enough for selecting among a huge number of images. Also it is better to consider geometry information of images. So besides global image descriptors, different local feature descriptors have been proposed to address the issues mentioned. An important methods proposed in the late 1990s by Schmid and Mohr (Schmid and Mohr, 1977), first finds interest points in the image and then computes invariant features in the neighborhood. In this work the first step is to consider small patches of the image. Such regions are well-constrained and less affected by perspective change. Even with changes in the view point or lighting conditions, those patches are likely to be identifiable. These

TABLE I: SOME KEYPOINT DETECTORS WHICH ARE NOT SCALE INVARIANT

Harris detector (Harris and Stephens, 1988)
Smallest univalue segment assimilating nucleus (SUSAN) (Smith and Brady, 1997)

so-called key points should be extracted after the raw image is preprocessed with techniques such as re-normalization and smoothing or even more sophisticated algorithms. Those keypoints are more robust to change and can be used for verifying geometry consistency between 2 images. When the image experiences changes such as missing a region or contains dynamic objects, only related features in that region would be affected. So representing an image through its feature descriptors returns more reliable results. Doing so the process for checking similarity reduced to just comparing corresponding common features.

2.2.2 Feature Patch Detection

The first step for finding the key points is searching for salient regions such as corners and blobs by selecting sparsely from salient regions [(Fei-Fei and Perona, 2005) (Nowak et al., 2006) (Tuytelaars and Schmid, 2007) (Tola et al., 2008)] or by sampling grid as described in (Nowak et al., 2006) (Tuytelaars and Schmid, 2007) (Makar et al., 2013) (Tsai et al., 2014). Some of researchers employed a combination of mentioned approaches (von Hundelshausen and Sukthankar, 2012) to take advantage of both approaches.

TABLE II: SOME SCALE INVARIANT SPARSE KEYPOINT DETECTORS IN CHRONOLOGICAL ORDER

Scale-normalized Laplacian and Hessian (Lindeberg, 1998)
Scale-invariant feature transform (SIFT) (Lowe, 1999), (Lowe, 2004)
Maximally stable extremal regions (MSER) (Matas et al., 2004), (Nistér and Stewénius, 2008)
Geometry-based affine-invariant detector (Tuytelaars and Van Gool, 2004)
Speeded up robust features (SURF) (Bay et al., 2008b), (Bay et al., 2006)
Binary robust invariant scalable keypoint (BRISK)(Leutenegger et al., 2011)
Rotation-invariant fast features (RIFF) (Takacs et al., 2012)

The commonly used algorithms used in visual search usually employ the method of sparse selection of patches. One reason is that these methods are faster and more efficient for feature extraction in applications such as image retrieval. Table I and Table II present some popular methods in chronological order for detecting scale variant and scale invariant interest point, respectively. After key-points (interest points) detection , related features should be represented to make them invariant to transformation. Those features will be invarient with respect to a transformation if the featres vector stays intact after applying that transformation. A variety of local descriptores have been proposed during the last decade. Table III shows some of the main works in this area. One of the most papular one is SIFT (Lowe, 2004), that is invariant to scale, affine transformation, and illumination. A samples of the SIFT descriptor is shown in Figure 10

TABLE III: SOME WELL KNOWN FEATURE DESCRIPTORS IN CHRONOLOGICAL ORDER

Scale-invariant feature transform (SIFT)(Lowe, 1999), (Lowe, 2004)
Speeded up robust features (SURF) (Bay et al., 2008b), (Bay et al., 2006)
Dual-tree complex wavelet transform (DTCWT) (Kingsbury, 2006)
Compressed histogram of gradients (CHoG) (Chandrasekhar et al., 2012)
Rotation-invariant fast features (RIFF) (Takacs et al., 2012)
Binary robust independent elementary features (BRIEF) (Calonder et al., 2010)
Fast retinal keypoint descriptor (FREAK)(Calonder et al., 2010)
Local quantized pattern (LQP) (ul Hussain and Triggs, 2012)

In most cases, the best performance in terms of retrieval accuracy obtained with the use of Hessian-affine detectores. SIFT needs 128 bytes for each descriptor while a more recent descriptor, SURF, (Bay et al., 2008a) is faster and needs only 64 bytes for each feature. Those feature descriptors have shown to give better performance than algorithms such as MSER (Matas et al., 2004). In most of recent research SIFT and SURF variant have been adopted since they provide state-of-the-art performance in recognition. A version of SIFT called Affine-SIFT (ASIFT) (Morel and Yu, 2009) has been proposed and is shown to be more robust to affine transformation. A sample ASIFT based matching is presented in Figure 11. Due to its higher computational cost and memory requirements, ASIFT was not used in our work. Another method that needs reduced memory is Compressed Histogram Of Gradients (CHOG).

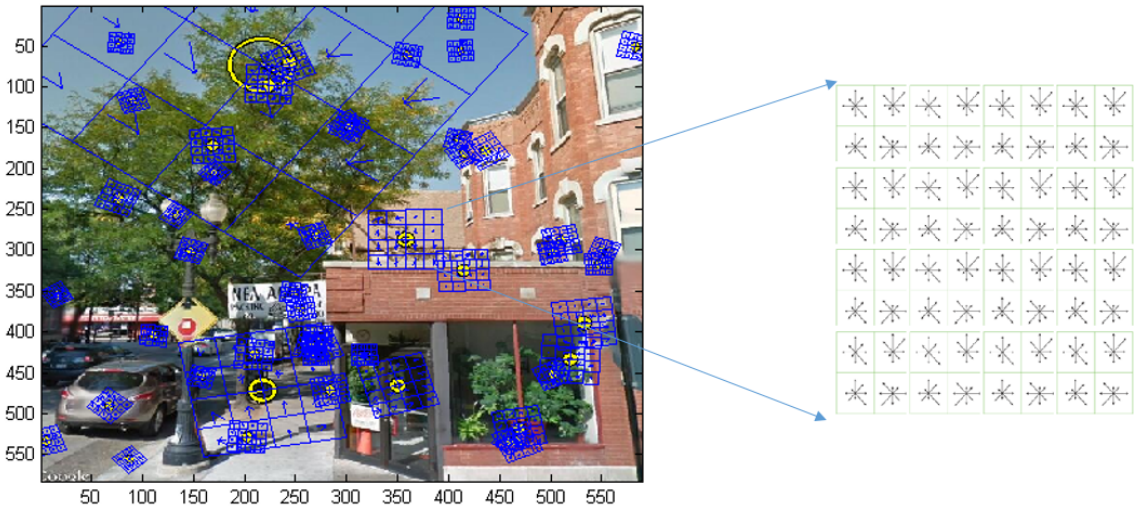


Figure 10: SIFT descriptor

This descriptor needs 60 bits while providing performance comparable to SURF. Later Calonder et al. proposed binary descriptors in a method called Binary robust independent elementary features (BRIEF) (Calonder et al., 2010). His algorithm is based on comparing intensities between different region in the patches and encoding them in a binary vector. The last method in the table, FREAK is based on comparing the intensity of pixels by employing a model inspired by the human vision system. Although binary feature descriptors are faster than SIFT and SURF but researchers such as Heinly et al. showed the performance of gradient based descriptors are better than Binary descriptors in terms of recognition and matching (Heinly et al., 2012). Meanwhile the best reported result in case of object recognition and landmark recognition is RootSIFT, which is implemented as a ℓ_1 normalized version of SIFT. In

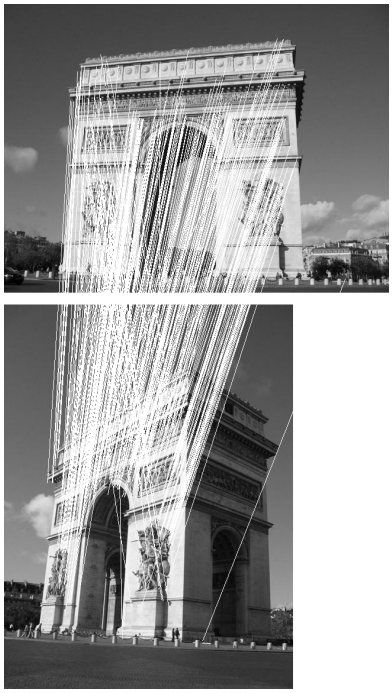


Figure 11: Matching two images using ASIFT

this research we have mostly used Dense version of SIFT called DSIFT and RootSIFT. The next section covers Bag-Of-Features based image retrieval and related concepts.

2.2.3 Bag-Of-Word apporoch

As mentioned before, image search in a huge database is more successful when local features are employed. The simplest search algorithm for image search considers the full representation of each image in terms of features and directly comparing features of query and datbase images. Due to the huge number of images in a database, time complexity for comparing a query to each of those images is really high and not practical for most applications especially those with mobile devices. So a

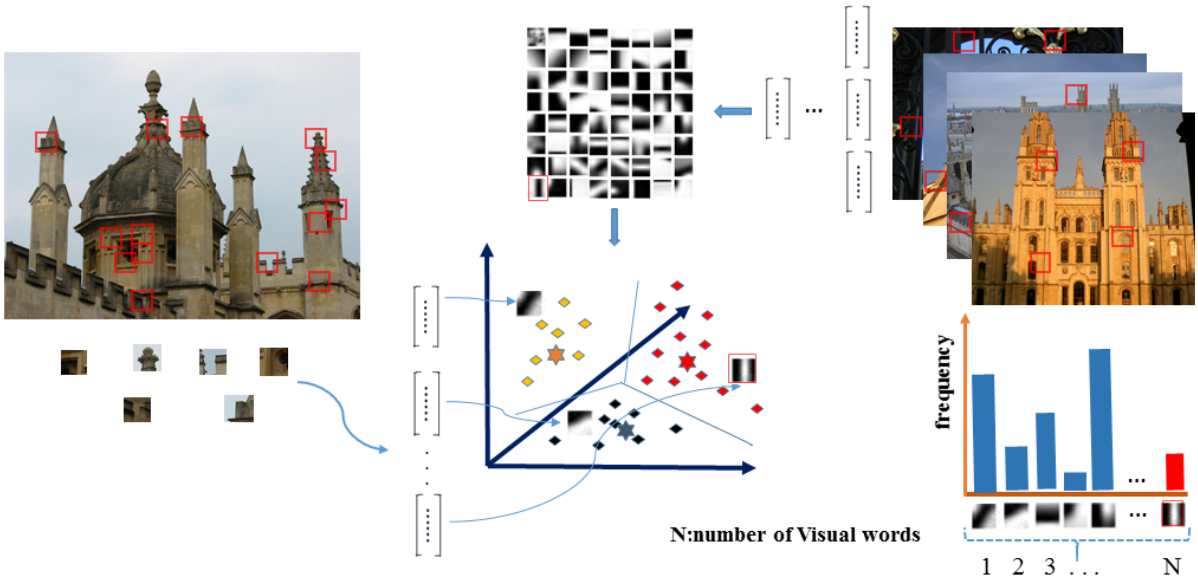


Figure 12: Simple BOF implementation

technique borrowed from text retrieval, proposed by Sivic and Zisserman (Sivic and Zisserman, 2003), reformulates the image retrieval in a way that avoid a majority of candidates even without calculating their structural similarity. The main part of this technique called Bag-Of-Word is clustering features by the popular algorithm of K-Means. In order to achieve a higher performance Nister and Stewenius (Nistér and Stewénius, 2008) have utilized Hierarchical K-Means to create more cluster centers. Those centers are called visual words. The next step is assigning each feature to the closest cluster center. In the Bag-Of-Word (BOW) method, each image should be represented as a histogram showing occurrence frequency of each visual word. Since this histogram does not contain any structural information about the position of the features in an image, BOW is not able to verify the geometric information of

features. To do that, an extra time-consuming step should be applied to a limited number of the best candidates that is discussed in following sections. One of the key steps in BOW-based approaches is the method used for clustering. This is because clustering introduces quantization error. Also assigning a particular feature to a cluster center may be done incorrectly. In this research 2 different techniques for clustering that have proved powerful in large scale datasets have been employed. The first one, Hierarchical K-Means, is technically similar to regular K-Means and Approximate Nearest Neighbor (ANN) search and randomized forest. After examining these methods, we found the ANN techniques is suited for our application. This conclusion was also reached in other works such as (Torii et al., 2013). So after studying its performance on our dataset, ANN is adopted in the rest of our work. The process for implementing BOW is described below:

1. Find the features for all images in a database.
2. Cluster features using one of the clustering algorithm with branching factor of n and depth of m (n^m visual words).
3. Find the closest visual word (cluster center) for each feature in database images
4. Represent each image by a histogram showing the frequency of each visual word
5. Re-weight each histogram by methods described in section 2.3.3 to consider importance of each visual word in terms of distinctiveness.

The process is shown in Figure 12. At query time exactly the same procedure should be applied. Then a similarity metric such as Euclidean distance can return the closest image in database. To make the

search process faster, different data structures have been proposed. The most commonly used method is Inverted Index (file) that is covered in next sections.

2.3 Clustering

To avoid direct similarity comparison between image descriptors due to time complexity, images should be represented as a vector of their local features. Since the number of features is high, features should be mapped to some fixed centers. Hierarchical K-Means and Approximate Nearest Neighbor search are two common methods in most of related works.

2.3.1 Hierarchical K-means (HKM)

The principle of HKM is similar to K-Means. Since K-Means is not suitable for large scale database with millions of features, a hierarchical decomposition (HKM) of features is considered. In this approach, features are clustered to K nodes using K-Means algorithm. Features belonging to each node must then be clustered to K clusters and this procedure should be applied for all nodes and continue recursively to reach to the desired depth. The process at query time is traveling by query features to find the best leaf in a way the feature belong to it are nearest neighbor (NN). The number of clusters is a function of two parameters:

1. Branching factor: the branching factor of the tree has a crucial role in clustering. Although increasing the branching factor for fixed number of features leads to consuming more memory, it reduces the running time.

2. Maximum depth: Let n be the branching factor of a particular tree. The number of clusters by setting maximum depth m is going to be n^m. For example for n=10 and m=5 we get 10^5

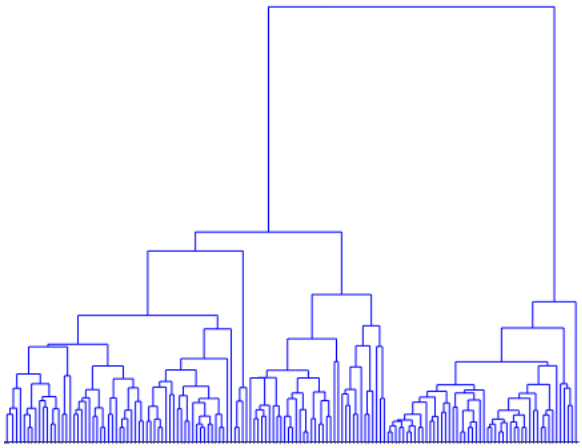


Figure 13: HKM clustering

cluster centers. Although a larger *m* requires more memory, the time complexity reduces due to decreasing the number of features in each particular leaves.

2.3.2 Approximate Nearest Neighbor (ANN)

ANN is a newer method for searching nearest neighbor in high-dimensional space where the complexity of search is significantly reduced by accepting some degree of error. Since it has shown better performance in image retrieval applications, most of the recent research has employed this algorithm for clustering and fast indexing of features. A variety of tools are available for implementation. A commonly used framework for computer vision application is Vlfeat (Vedaldi and Fulkerson, 2008). Although HKM is used in the earlier part of our research, it is replaced by ANN based on our experience and similar related research.

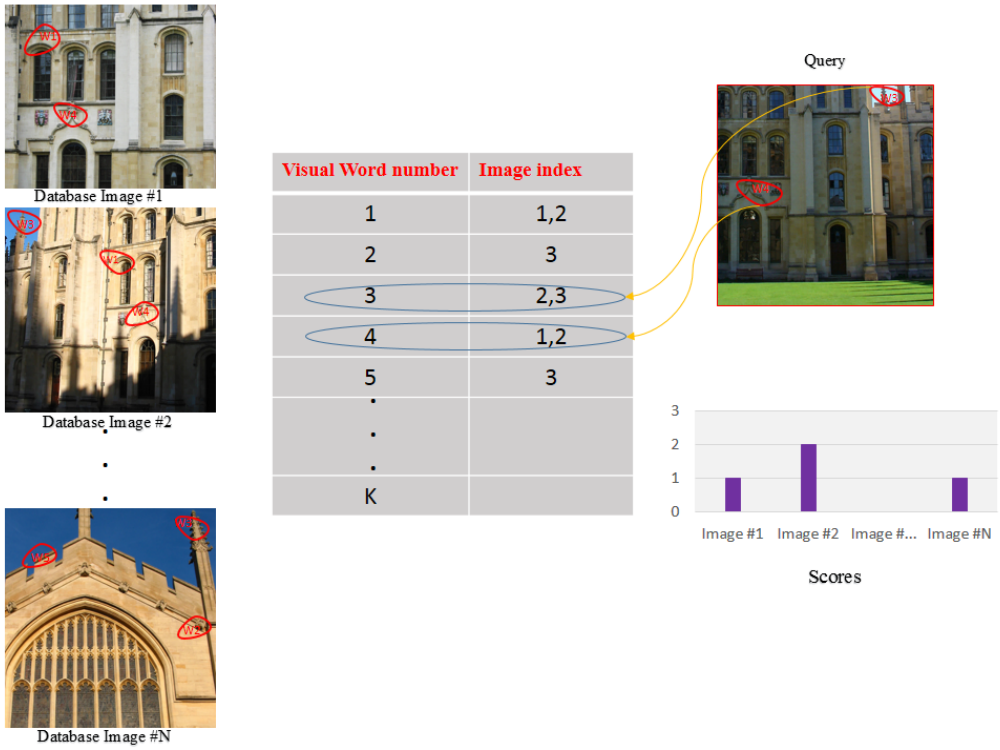


Figure 14: Inverted index approach

2.3.3 Inverted Index (file)

The idea of using inverted file comes from text retrieval. This technique creates a structure that lists all images containing feature f_i, $i = 1,2 : n$, when n is the number of vocabulary words. The advantages of inverted file at the query time is reduced run-time. Here instead of directly measuring similarity of each query to all images in database by methods such as Cosine distance, only an image which has common features (visual word) with the query should be considered. Figure 14 describes Inverted file.

To obtain a better result, some options such as the procedure for creating visual dictionary, number of visual words, TF-IDF weighting, normalization, distance metric and so on must be suitably chosen. Some of the mentioned parameters are described below for clarity.

1. TF-IDF:

   TF-IDF stands for the Term Frequency-Inverse Document Frequency. A variety of weighting schemes have been proposed to assign different weight to visual words to account for relative importance. Each word has a particular weight showing its importance and how distinctive it is. The experimental result shows it can be successfully applied for filtering stop-words in various applications such as classification and text summariztion.

   But there is a difference in image retrieval and text retrieval in creating BOF. Unlike text retrieval, in image retrieval we have to utilize a clustering method to assign each feature to a cluster center. So the quantization process has to contend with error (quantization error) and would change the final result. In this research several TF-IDF schemes were examined in our image search engine but the results were almost identical. So we decided to use the basic and most popular scheme of eq 2.1 in all of our study. The common formula for TF-IDF contains two terms: the first is the Term Frequency (TF) that is the number of times a visual word exists in a document (here image) and the second term, the Inverse Document Frequency (IDF), is logarithm of the number of the images in the dataset divided by the number of images which contains a specific term.

$$w_{xd} = tf_{xd} \times \log\left(N_d / df_x\right) \tag{2.1}$$

where N_d is the number of images in database, df_x is number of images containing x and tf_{xd} is the frequency of term x in image d.

TF: Term Frequency measures how many times a visual word occurs in an image. In order to consider the ratio of word frequency to all features of an image, term frequency is often normalized by the total number of the features.

TF(t) = (Number of times word t appears in an image) / (Total number of visual words in the image).

Besides considering importance of a particular word in a particular image, the importance among all images in a dataset should also be considered. For example some features appear in most of the images. Similarly in text retrieval, words such as 'is', 'are' appear all the time and are not as important as words with lower frequency of occurrence. So Inverse Document Frequency (IDF) is employed to address this concept. As a result with applying IDF we attempt reduce the weight of the frequent visual words and increase the weight of rare words.

2.3.4 Soft Assignment

To obtain a histogram showing features in a particular image a vocabulary tree should be created. Suppose the vocab tree has branching factor k and depth d. The total number of visual words is going to be k^d. For example considering 10 and 5 for k and d respectively lead to 10^5 visual words that is used in this research. The next step called Hard-Assignment assigns each feature to the closest visual word as is shown in Figure 16. In this figure the closest cluster center for the descriptor 2 is cluster 1. So all cluster weights except W_2 are considered zero. The typical method for quantization of a descriptor

using vocab tree is the greedy algorithm. In general this process cannot attain the best solution in terms of the distance between the feature and the best visual word. Also the feature descriptors are inherently noisy leading to the selection of an incorrect visual word. As a result, the same feature in the same region in 2 different images may be quantized to two different visual words. To reduce the effect of quantization error in hard assignment, Soft Assignment has been proposed (Philbin et al., 2008). In this method n nearby visual words are assigned partial weights based on their distance to the feature. Figure 17 illustrates this method. Unlike the approach for Hard-Assignment, 3 closest clusters are found. A function is then used to assign a weight proportional to the distance between feature descriptor and the cluster center. The Euclidean distance between descriptor 2 and the cluster centers are denoted with $Dist_k$ for the k-th cluster. This weight can be normalized as shown by the sum over $k = 1,2,4$.

Different weight functions can be considered such as $F(x) = \exp(x^2/\sigma^2)$ where σ is the standard deviation extracted from the distribution of distance between features and related nearest nodes. In a recent work by Torii (Torii et al., 2013) instead of considering a function of distance, a simple weight is assigned to kth nearest neighbour cluster by $Weight(k) = \frac{1}{2^{k-1}}$ for $k = 1,2,...m$ where m is the maximum number of assignments. In our work we selected the second function since it is easier to implement and it returns the same result, as has been emphasized in (Torii et al., 2013).

In addition to using the basic scheme, some researchers have been working on different extensions. Those efforts have mostly focused on better ways of learning the vocabulary tree (Philbin et al., 2010), better quantization (Jegou et al., 2011), query expansion (Chum et al., 2011), use of graph or 3D structure of the dataset (Li et al., 2010), (Philbin et al., 2011) or considering the spatial relation between visual words (Jegou et al., 2008), and multiple architectures that are based on damping features with

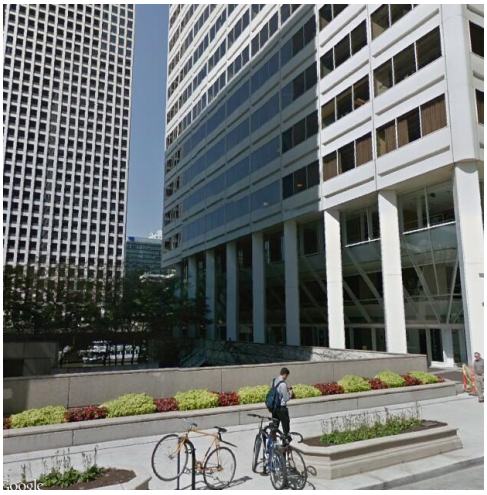


Figure 15: Image with maximum visual word occurrence equal to 241

higher occurrence. One of the best architectures that shows better performance in terms of recall is called Burstiness (Jégou et al., 2009). This method down-weight the number of occurrences of each visual word. It is based on this fact that in place and landmark recognition problems, the frequency of some of visual words is more than one. In some cases the frequency may even be more than 100.

For example in Figure 15 that shows a Chicago street view the frequency of one visual word is 241. Such a dominant weight may degrade the algorithm performance. This is because this image will receive a high score in retrieval for any query with the common visual word . This common feature may result from a wrong assignment (quantization error). Also the number of common features between 2 relevant images is usually less than 100. In the presence of such a feature with high occurrence frequency, all the other features would not be significant. So the Burstiness method reduces the weight of visual words

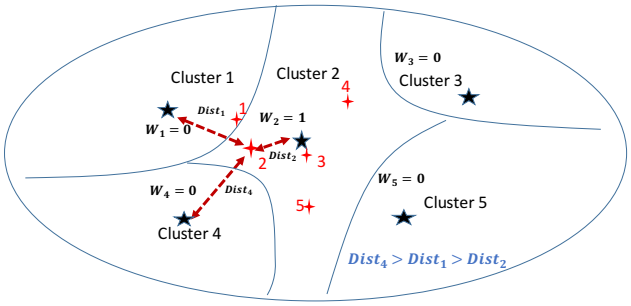


Figure 16: Hard Assignment to nearest cluster

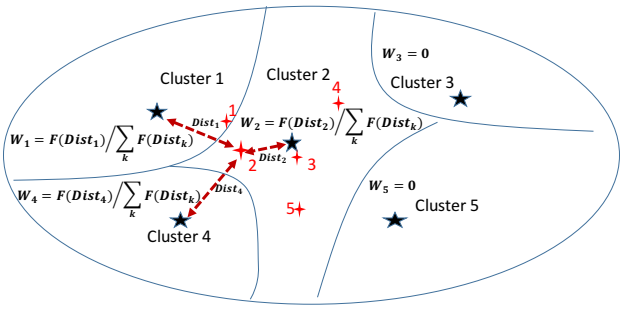


Figure 17: Soft Assignment to 3 closest clusters

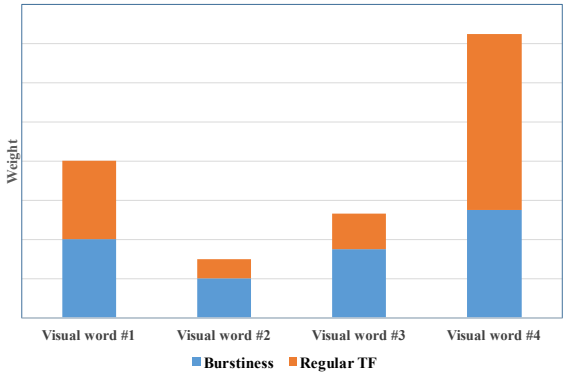


Figure 18: visual words before and after burstiness normalization

by using a suitable scheme as shown in Figure 18. We note that limiting the frequency of visual words below a threshold such as six works almost the same as Burstiness.

Recently Torrii proposed an apporoach to improve Soft Assignment (SA) (Torii et al., 2013). Although SA shows better performance versus Hard Assignment, considering a fixed number of visual words for all features would not lead to the best solution. The main concern of their work is to find repetitive structures (features) since they observed that such structures often occur in man-made environments specially in fences, streets and building facades. Therefore they reduced the weight of the related index in the BOF representation. More details about their approach are covered here since their result shows better performance with the higher recall rate when applied to place recognition datasets. The first step in their approach is finding spatially localized groups of visual words with the same appearance. Then the weight of repeated visual words in the BOF model is modified based on the notion that multiple occurrences of repeated structures yield a natural soft assignment to visual words. To find

a repeated structure as shown in Figure 19 the image is represented by a graph. Each feature is a node and edge exists between two nodes when multiple criteria are satisfied. Each feature should then be assigned to a different number of visual words according to the number of members in the group they belong to. In this work it is shown that the presence of a visual word is important and not the related occurrence frequency. In their work the system considers Adaptive Assignment where a different number of assignments for different features are considered. To find repetitive structures and suitable number of assignments to a particular visual word they considered three parameters:

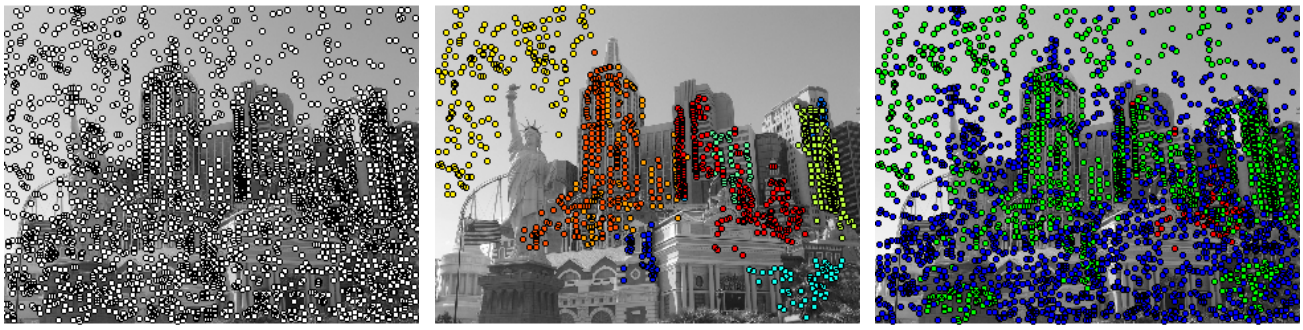
1. Euclidean distance between features.
2. Scale ratio between features (this ratio should be between .5 and 2)
3. Presence of a common visual word in the first 50 nearest neighbor visual words.

Based on these criteria and by creating a graph they found repetitive structures called Repttile as shown in Figure 19.b. To determine which repttile each feature belongs to, different number of assignments are considered. A sample result in Figure19.c shows features with different number of assignments with different colors.

2.3.5 Similarity measure

At the query time, a metric should be employed showing the similarity of two BOF vectors. This metric should be a suitable distance function. The Euclidean distance is a good candidate since it is easy to work with. It even could be more convenient if normalized Euclidean vectors (equal to one) are used. In general the Euclidean distance between two vectors A and B can be expressed by eq 2.2.

$$d^2(A,B) = \|A-B\|^2 = \|A\|^2 + \|B\|^2 - 2A.B. \tag{2.2}$$



a: Detected features

b: Detected repttiles

c: Features with one (red), two (green), and three (blue) assignment

Figure 19: Repttile and Adaptive Assignment simulation results based on Torii et al work

Since A and B are normalized, $\|A\|^2$ and $\|B\|^2$ are equal to 1. So $d^2(A,B) = 2 - 2A.B$.

As shown in Figure 2.2 the distance between the normalized versions of the two vectors can be obtained by calculating the scalar product of the two vectors, which can be used directly as a similarity metric. Defining a simple similarity measure make it easy to compare the query feature descriptors to all features descriptors of all images in dataset. Although we represent each image with a vector, it is not possible to do so in the feature space. Instead an identifier (ID) should be assigned to each feature. Each ID represents multiple similar feature descriptors and as a result makes it possible to transfer from high-dimensional space to small set of integers (visual words). Each image can then be expressed in terms of presence of each visual words in a vector. It should be noted that in most of the related research the inner product of two histograms has been considered as a similarity measure. Here one of the most important consideration is the number of visual words. In most of the cases having a larger set of visual words leads to a higher recognition rate. We have tested the retrieval performance for different size of

visual word sets on some well known image retrieval datasets. A higher number of visual words also damps the high repetitive features that would change the result. This is not the only option for finding the best match. We may also use another version of similarity measure by considering the weight of each visual word as described later in 3. Moreover an algorithm such as Inverted file(Inverted Index) shows a better performance in terms of run time. That algorithm uses sparsity of the vectors. In a large scale image retrieval task with a huge number of visual words the number of non-zero visual words in a BOF vector is much less than the total number of visual words. So the BOW vector is sparse and this property can be utilized to obtain better performance in terms of memory and time. Another method called Histogram-Intersection works well in our retrieval system. One of the main advantage of this metric is convenient algorithm when is used in conjunction with Inverted index approach.

Suppose V_Q=[$V_1^q, V_2^q, ...V_{n^m}^q$] and V_D=[$V_1^d, V_2^d, ...V_{n^m}^d$] are vectors denoting the frequency of the visual word $w_0, w_1, ...w_{n^m}$ for the query and the dataset images, respectively. Each element of the vectors is the number of times each feature descriptor belonging to a query or dataset image has been assigned to a visual word. Representation of images according to those vectors can be used either for Hard or Soft Assignment. The only difference is that Soft Assignment assigns 1 to the nearest neighbor visual word and a fraction less than 1 to other neighbors. These values can be a predefined as in eq 2.3 or calculated from the distance between feature descriptor and the assigned visual word (Philbin et al., 2008).

$$W_{depth} = \frac{1}{2^{Depth-1}} \quad \text{for} \quad Depth = 1, 2, ..., M \tag{2.3}$$

In order to assign weight to each visual word, IDF should be applied in the similarity metric formula. The complete formula for measuring the similarity between two images (vectors) is shown in eq 2.4.

In our research we consider upto $M = 3$ closest visual words for Soft-Assignment.

$$SIM = \frac{\sum_{i=1}^{n^m} IDF(i)\ min(V_i^q, V_i^{db})}{(\sum_{i=1}^{n^m} IDF(i)V_i^q)(\sum_{i=1}^{n^m} IDF(i)V_i^{db})} \tag{2.4}$$

where $IDF(i) = log(\frac{N_{db}}{N_i})$.

N_{db} is the total number of images in the dataset while N_i is the number of images with at least one visual word i. The value for IDF depends on multiple variables such as the number of images in the dataset, the number of visual words, and the average number of features of images. The typical value of IDF in our research varies between 4 and 15. As can be inferred from the IDF formula in eq 2.4, the more distinctive visual words receive higher weight.

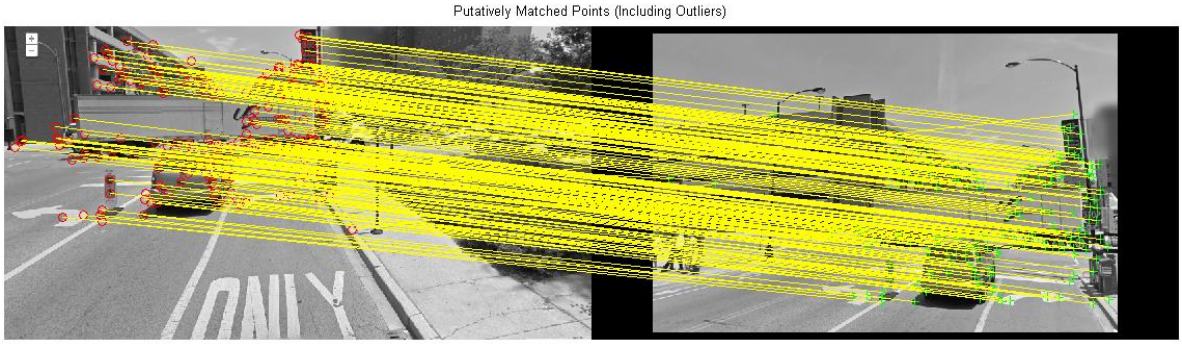
2.3.6 Min Hash

In this method the first step is to extract some of locality-sensitive hash function from the binarized histograms of images in the database and creating a set of Min-Hash tables. At the query time, Min-hash tables should be checked to get the nearest neighbor in Min-Hash tables. Spectral hashing method (Weiss et al., 2009) is based on this notion that the sum of Hamming distances between pairs of codes should be minimized. Some other approaches like Locality-sensitive hashing (LSH) (Charikar, 2002), also has been proposed for this task. In our work, as mentioned, we use regular Bag-Of-Word method for the retrieval engine.

2.3.7 Homography Verification

When the retrieval process considers only the vectors that represent the visual content of images, sometimes it fails to produce accurate results. As mentioned before the retrieval process works based only on measuring the distance between vectors showing the visual content of the images. Two images may have similar BOW vectors obtained from the features extracted from different spatial layouts. Therefore 2 images may have similar BOW representations while they are totally different in content and this difference is not considered in measuring the similarity metric. Ignoring the geometry of features in BOW model may lead to failure in the retrieval task. Substantial effort has been focused on addressing this issue. The most common solution that yield reasonable results is to check the consistency of the feature position. To do that another step called Homography verification is usually applied. The process seeks to find a set of pairs of similar features in the query and the dataset images and then to find homograghy between them.

The most popular algorithm adopted for homography verification is RANSAC (**?**) that returns the number of the inliers. RANSAC estimates the homography so as to reach to a higher number of inliers. This number in turn can be used as a metric to figure out whether a returned image is a correct match or not. Although the time complexity of the RANSAC algorithm is high, it is not necessary to apply it to all images in the dataset. A common way in image retrieval is to consider K best matches from the BOW results and re-rank them by RANSAC. Depending on the application K can vary from 10 to even more than 100. In most of our simulations K is set to a value of up to 50 to make the whole process fast. Figure 20 shows the putative match between corresponding features of 2 images. Here only the strong features are considered using Low (Lowe, 2004) law. Suppose $d(F_i, G_1)$ is distance between feature i



/

Figure 20: Putative match between corresponding features

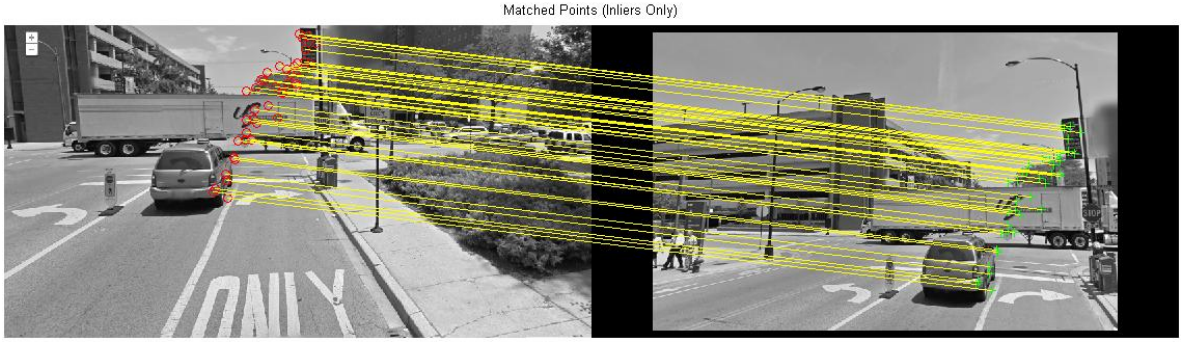


Figure 21: Inliers found after applying Homography verification

of image F and closest feature in image G, and $d(F_i, G_2)$ is second closest feature in image G, then the pairs of features (F_i, G_1) should satisfy

$$\frac{d(F_i, G_1)}{d(F_i, G_2)} < T \tag{2.5}$$

for a suitable threshold T. T should be greater than one to select more strong feature pairs. The commonly chosen value for T is about 1.5. Figure 21 shows the resulting feature match after applying RANSAC. Here we seek to keep just inliers and remove outliers which are not compatible with homography between two images. The number of inliers is the best way to re-rank the results returned by the retrieval engine. After images are re-ranked by RANSAC, we can assign the GPS tag of the best match or average of closest matches as a rough estimate of the location for the query that gives us a middle-of-the-street level accuracy. The resulting location error can be large when the query camera position is actually on the sidewalk. To achieve a higher accuracy, alternate methods such as those that utilize similarity matrix from two query-matching images (trifocals tensor) can be applied as done in (Xu et al., 2012), (Vishal et al., 2015), and (Roshan Zamir et al., 2014). These methods utilize Structure From Motion (SFM) to estimate three camera positions: two matching-image camera positions and the query camera position. This step turns to be important for our work since our ultimate goal is to estimate location of the camera not just enhancing the image retrieval algorithm performance.

In the above approaches SFM is used to estimate three camera positions: two positions of two cameras corresponding to the two best-matching images and the query camera position, yielding a triplet of reconstructed 3D camera positions. Numerous triplets are typically taken into account and subsequently processed by a least-squares fitting routine to compute the similarity matrix and generate a unique estimate of the query's location. Dimensionality reduction techniques such as PCA should be used to reduce the 3D position vectors to 2D position vector. A key limitation of currently used methods is multiple application of SFM processing on pairs of images returned by the retrieval pipeline along with the query which is computationally expensive. Also, the selected pair of images may not have

enough common features. This is because images are ranked based on the similarity to the query while all images should be sent to the SFM process. In following chapters we show how only a single SFM process can be applied to the optimally selected images. Then we propose a method to directly find camera coordinate transformation parameters between relative camera centers returned from SFM and bundle adjustment to real world coordinates.

CHAPTER 3

ASSESSING PERFORMANCE OF IMAGE RETRIEVAL TECHNIQUES

Content-based image retrieval (CBIR) provides a framework that can be used for organizing digital images based on their visual content. CBIR systems are useful in a wide variety of applications.

In mobile applications, usually the main task of a CBIR systems is to examine a query image and search for a similar image among a set of candidates. The core of the procedure is the definition and extraction of feature descriptors that work robustly in different scenarios such as the presence of the noise, occlusion, or any moderate change in perspective view. In applications such as navigation, the images are captured from street level, and usually in a dynamic environment. The use of robust feature descriptors enables us to extract an image similar to a query from a huge dataset. Since different steps and related concepts for image retrieval were reviewed in the previous chapter 2, in this chapter we confine our attention to performance of the state of the art image retrieval schemes. Different algorithms for each step of the image retrieval are used in the following chapters that are introduced in this chapter. We applied this methods in our pipeline when a available dataset such as San-Francisco is used to be able to compare our result with the other similar research. Although image retrieval process is powerful in general, some issues make outdoor visual based navigation that is based on image retrieval more difficult task. As described before, there are variety sources of distraction in outdoor from bus stations and trees to dynamic scene. Moreover, factors such as the different seasons can influence the result. Also in some special cases, we have found multiple images in dataset that almost completely were covered by a truck. Moreover, pedestrian, light reflection and shadow, glass windows and symmetric

buildings cause negative retrieval result. In our project that is mostly focused on helping blind or visually impaired pedestrians, the main goal is to achieve high accuracy in retrieval. While Pure image based position recognition shown reasonable performance for city-scale datasets such as San-Francisco or Pittsburgh, the result are not satisfactory for many real-world applications. This is because queries in the available datasets usually cover the frontal view of buildings while in real-word samples we may have a dark picture for example of a building wall taken from below a rail track. Therefore to achieve higher accuracy in our work, information from other sources has been extracted and used to limit the search space. This technique is discussed in next chapter in detail. To implement our image retrieval pipeline a variety of feature descriptions such as SIFT, SURF, ASIFT have been considered. Unlike others, ASIFT needs more time and returns more features that is not suitable for large datasets. Initially we used SIFT and later DSIFT in our work, and we eventually replaced these with a new version called Root SIFT (RSIFT). RSIFT shown better performance in terms of accuracy with the almost identical computational cost to SIFT. The next section presents more details about different factors and their influences on the result.

To find the similarity between images, both inner product (Cosine distance) and Histogram intersection have been used. After comparing the performance of well-known algorithms reported in recent research, we found the use of RootSift and ANN search returns better results. To evaluate the results multiple criteria such as MAP, Recall, Precision or recall-precision graph are used. Those factors make it possible to compare our result with other reported research.

3.0.1 Implementation and result for Oxford 5K dataset

Although most of our algorithms are implemented based on the SIFT feature, we have used RootSift that is the newer and more powerful feature descriptor. For creating a visual word, ANN is used to return different sizes of the clusters in a reasonable time. Most of the recent research has shown that it is superior to method such as HKM (Torii et al., 2013), (Chen et al., 2011) . We have created different sizes of the vocabulary trees from 100k to 500k and verified the results with metrics such as MAP, recall and precision curves. Also techniques such as Soft-Assignment are used to provide the highest possible performance in terms of the recall. Regarding the effect of similarity distance different methods of measuring distance between BOW vectors of image are used and the results are presented in next section. Before examining the results we need to define some concepts as follow:

3.0.1.1 Precision and Recall

To assess the results of image retrieval, a variety of metrics may be considered. An important metric often used in information Retrieval (IR) research is Recall. Recall, defined in eq 3.1, is a measure of the fraction of all relevant images in the datasets that are retrieved while precision, defined in eq 3.2, is a measure of the fraction of retrieved images that are relevant to the query.

$$Recall = \frac{Number\ of\ relevant\ images\ retrieved}{Number\ of\ relevant\ images\ in\ dataset} \tag{3.1}$$

$$Precision = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ number\ of\ images\ retreived} \tag{3.2}$$

3.0.1.2 MAP

Another metric used widely in information retrieval is Mean Average Precision (MAP), given by eq 3.3.

$$MAP(Q) = \frac{1}{|Q|}\sum_{n=1}^{|Q|}\frac{1}{m_n}\sum_{k=1}^{m_n} Precision(R_{nk}) \tag{3.3}$$

$\{d_1, d_2, ...d_{m_n}\}$ are relevant images for q_n

and R_{nk}: ranked retrieved images set from top to d_k and $q_n \in \mathbf{Q}$

In order to see whether the retrieved documents are relevant, precision can be calculated as shown in eq 3.2. In this work we have employed Recall curve to compare our results with other available results. To plot this curve a criteria may apply. The most common criteria is the number of inliers in the last step of retrieval that is actually homography verification. It means a threshold can be considered to decide whether the result for a query should be retrieved or not. For example we discuss in future that, the probability of successful retrieval will be high when the number of inliers is more than 30. It means setting different value for the threshold changes the curve. To avoid this and have a metric without influence of the number of inliers, another curve called Precision-Recall curve may be used.

3.0.1.3 Precision-Recall curve

Another common method used by the Information Retrieval community for showing and evaluating image retrieval performance is Precision-Recall curve. One reason for popularity of this curve is that the result can be interpreted easily. There are some other similar representations such as Recall vs precision or recall vs $1 - precision$. Also partial representation of this curve (Partial PR curve) have been introduced to show a region of this graph in greater detail. It is recommended to use partial

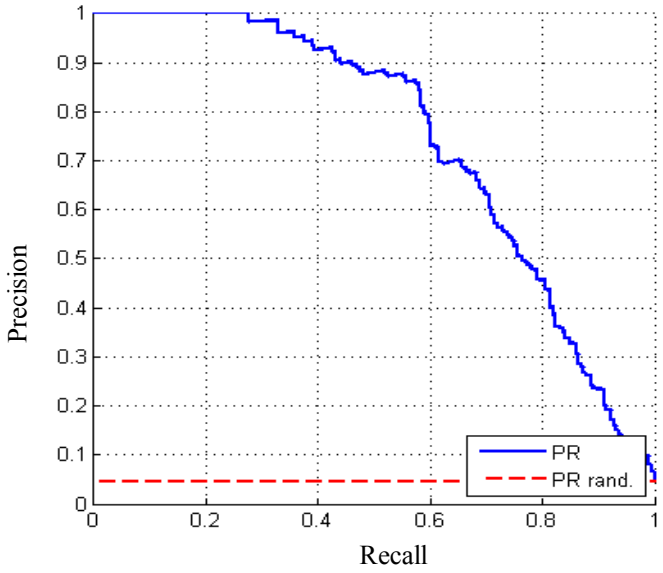


Figure 22: Precision-Recall curve for 100k visual word (MAP=53.9).

graph along with full graph to avoid any wrong interpretation due to lack of poor performance regions. Fig 22 and Fig 23 show the result of retrieval for 100k and 300k visual words. For both results we have considered inner product for measuring similarity. MAP for those two results are 53.9 and 75.9, respectively.

Another point that has been investigated is effect of IDF. Since we found IDF range varies and just one visual word with corresponding high IDF value can cause error in retrieval process, other versions of IDF such as square root(IDF) has been applied. Although according to Fig 24 histogram intersection of two BOF vectors with square root of IDF for a fixed number of visual words (300k) outperforms regular IDF in terms of MAP but the two results are very close. So it is unlikely to acheive a significantly

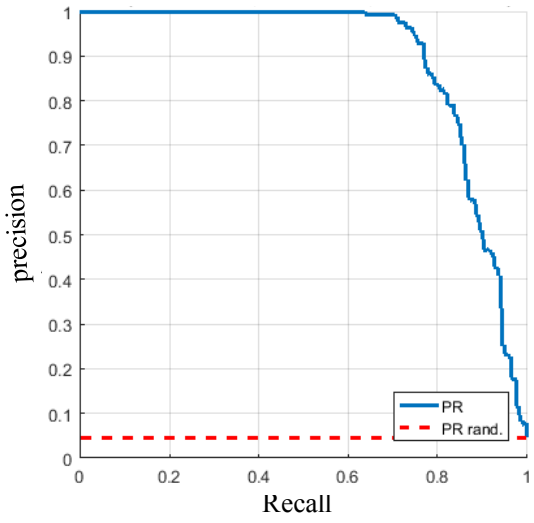


Figure 23: Precision-Recall curves for 300k visual word (MAP=75.92)

	100k	200k	300k	400k	500k
MAP	53.91	69.35	75.92	81.11	84.29

TABLE IV: MAP FOR DIFFERENT SIZE OF VISUAL WORDS USING INNER PRODUCT

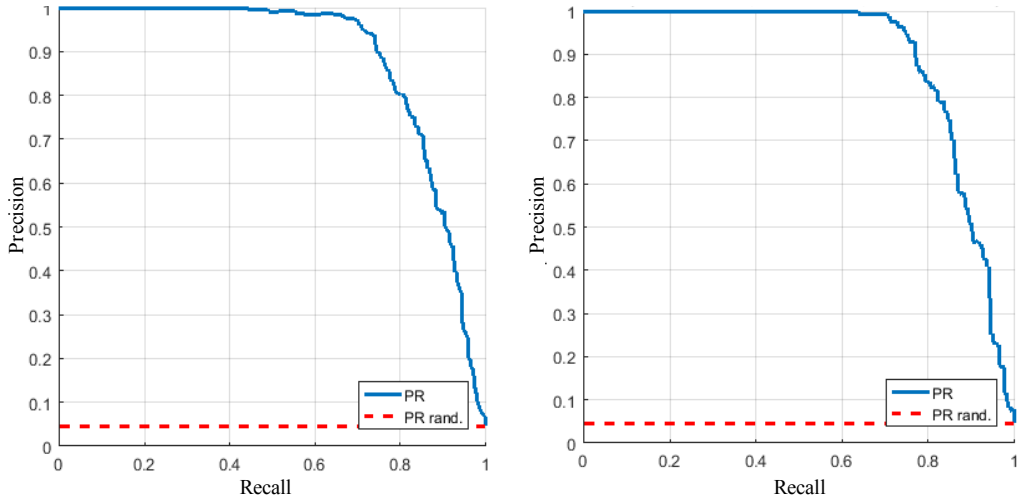


Figure 24: Precision-Recall curves for 300k visual words and inner product for a: regular IDF(MAP=75.92), b: square IDF (MAP=76.12)

better result by changing IDF. Soft Assignment yielded the worst result. As also mentioned in (Torii et al., 2013) the performance of their method based on repttile detection and burstiness was close to the performance by hard assignment for Oxford dataset. One reason for this result is that the query and dataset images are exactly the same for Oxford dataset. The only difference is that the query is a cropped part of an image in the dataset as shown in appendix A.1 by a yellow rectangular. So quantization error unlikely to be the source of error since the query and the corresponding dataset image have exactly identical features except at the border of the query images. So considering multiple visual word for each feature descriptor as employed in Soft Assignment approach cannot improve the result. Table 25

provides the results of the comparison for the different approaches with the same vocabulary size (300k).

	Inner product	Inner product + Square (IDF)	Histogram intersection	Soft assignment
MAP	75.92	76.12	79.1	74.69

Figure 25: MAP for different approaches with 300k visual words.

For larger vocab size as shown in Figure 26 the highest possible performance is achieved with the 500k dataset. MAP value for the two different methods of similarity metrics is 83.76 for histogram intersection while is 84.29 for inner product. This is different from our expectation since for all previous vocab size from 100k to 400k histogram intersection as shown in section 2.4 outperforms inner product. One plausible explanation of this result is that by increasing the size of vocab tree the frequency of visual word occurrence in BOF vectors is going to decrease so the performance of histogram intersection can not reach to similarity measure based on inner product. In chapters 4 and 5, 200k vocab size for the San-Francisco and Chicago datasets have been utilized. For such large datasets, it would not be reasonable to select very large vocabulary tree. This is because the model would need huge memory to keep all BOF vectors.

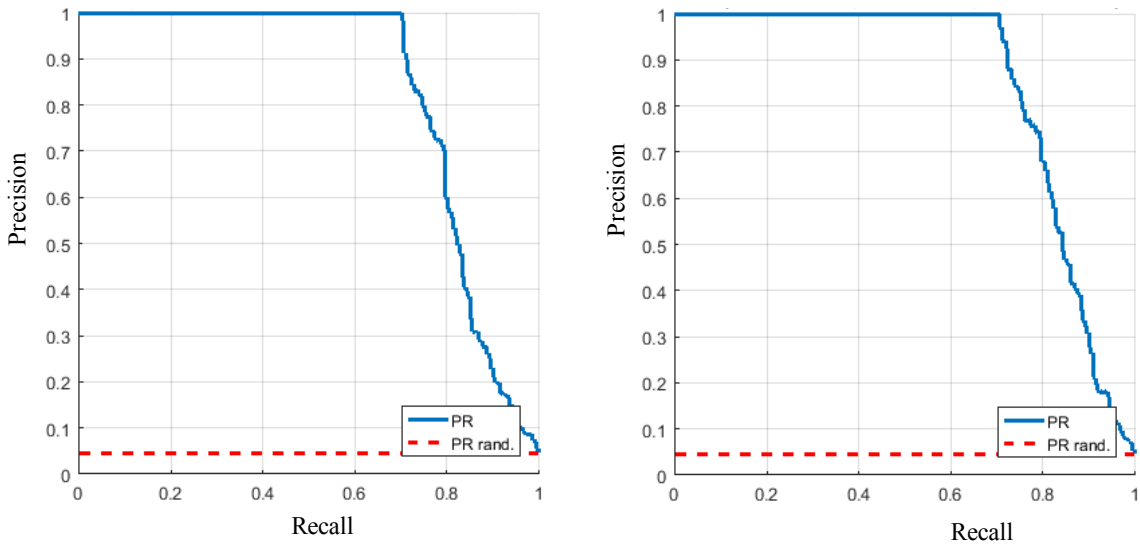


Figure 26: Precision-Recall for Histogram intersection (MAP=83.76) and inner product(MAP= 84.29) with 500k visual words.

CHAPTER 4

EFFICIENT POSITION RECOGNITION BY NARROWING DOWN THE SEARCH REGION USING GPS DATA

In this chapter we are going to verify and compare the result of basic image retrieval either with initial estimation of position or just with pure image retrieval. As mentioned before the GPS error of mobile phones are rarely more than 100 meters (Schroth et al., 2011). Actually this level of error may be negligible in some applications, but it is crucial in others. Consider a blind person in a dense region of a city with tall buildings and towers. Experiencing 100 meters error may cause street level error in navigation. To address the problem of inaccurate localization, significant research effort has been directed at finding solutions to compensate GPS shortcomings mostly by considering available sensors in smart phones and other mobile devices. Using a query image taken by camera and comparing it with a large database of images that covers cities is a solution for finding the place. To do that for a real-world application we investigate the use of estimates of GPS error to limit the search space. Also we found the GPS error is usually less than 100 meters. Although image retrieval techniques based on Bag-Of-Features (BOF) can be exploited, we can just use the last step of retrieval that is Homograpgy verification. In retrieval apporoches most of methods select more than one candidates for a match in this step. An additional step called Homography verification performed by applying the popular algorithm of RANSAC (Fischler and Bolles, 1981) is used to choose the best match among the plausible candidates.

In fact this step compensates for the weakness of image retrieval schemes based on BOF where the geometric information of images is ignored. Since we predict the number of images in a small region is limited the Homography verification can be applied directly to our candidates extracted by distance constraint. In the following section the running time and limitation of this approach is discussed. We need to keep in mind that the time for running RANSAC is high, So this approach can be used when the number of the candidates is not high. Otherwise full image retrieval pipeline should be considered. To check the possibility of this method we need to have a sense from the number of images returned by distance constrain drived from GPS error. Since other available dataset doesn't consider GPS error, we have tried to build our dataset containing GPS error for queries. Next section covers creation of our dataset and proposed techniques used to improve the result.

4.1 Our dataset for Chicago

Available databases are not suited for the purpose of some part of our study since their query images do not contain all the sensor data used in our proposed methods. The dataset for evaluating our method was created by Google Street View (GSV) service. Since the introduction of GSV, some researchers have used it to create datasets. The GSV service covers most of US cities and more than 10 countries in 4 continents. Although it is available free for access, the creation of a dataset that contains a large collection, for example thousands, of images requires that permission be sought from Google for downloading images. With approval for such access and by utilizing the GSV API, the images can be downloaded directly from a URL. The first step before is to find all the coordinates containing GSV images by creating and running a JavaScript code. After storing all possible coordinates, a URL request created with a Matlab script is used to download images for each unique set of coordinates. Unlike method used in

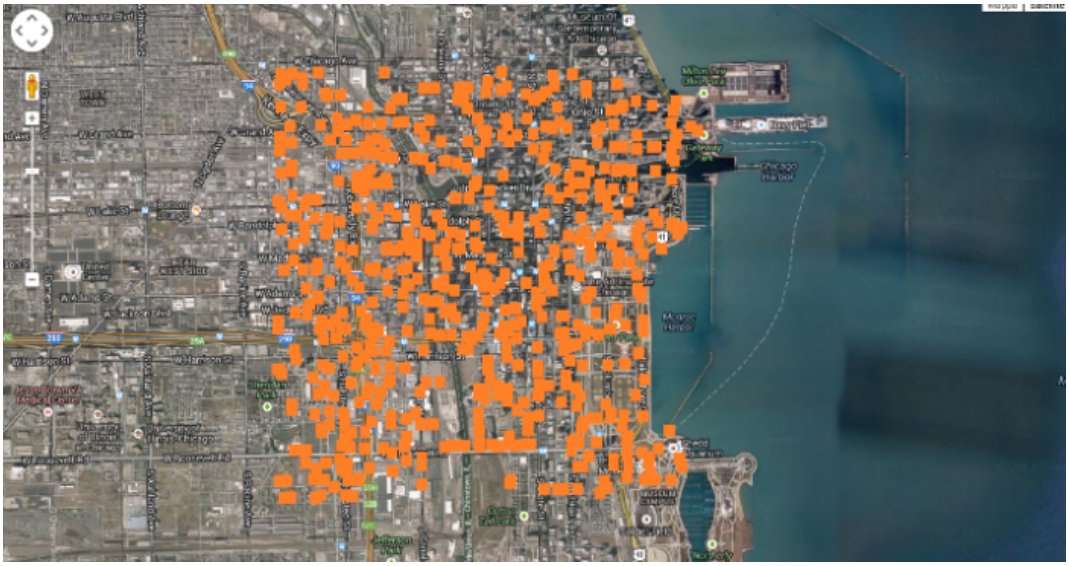


Figure 27: Region in downtown Chicago covered by our database

the Pittsburg dataset, we are not downloading panorama image but a regular split version directly by sending a request containing heading, FOV, pitch, Yaw, Roll, lat, long, and so on.

Since the aim of this research is localization by a pedestrian, pitch and FOV are assigned to 10 and zero to get images similar to what a pedestrian would observe. For each set of coordinates we considered 12 images with 30 degree separation in heading. The field of view is 60 degree so the image overlap is 50 percent. The database contains coordinates and heading of all images and these coordinates have high accuracy because Google vehicles are equipped with high-precision instrumentations. To evaluate the effect of size of dataset, two datasets A and B have been created. They contain 54k and 92k images respectively around the center of orange region in Figure 27 which is the densest region of the Chicago.

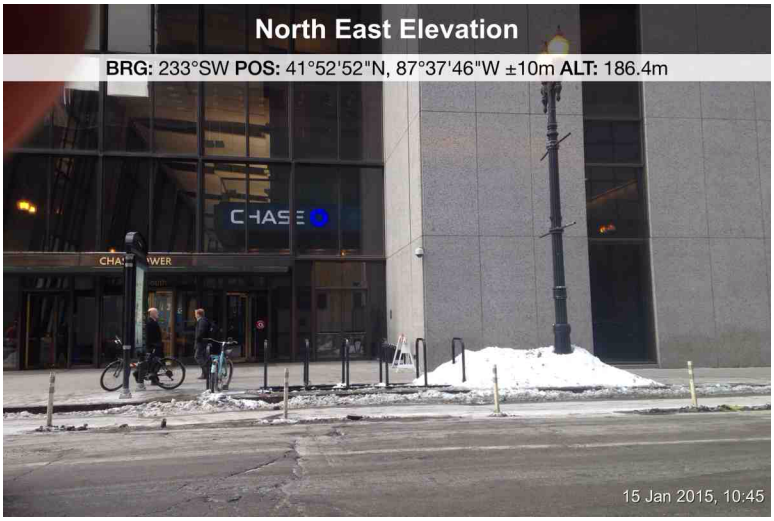


Figure 28: Sample query by Solocator

In fact larger dataset (B) covers all images in A. Both datasets are available and can be sent to other researchers upon request. It is clear that for a city scale database, multiple sub-databases can be created. On the other hand, given the enormous of images associated with a big city, it is advisable to split the database to smaller overlapping regions as in (Zhang et al., 2011) while our proposed method still can be used to improve the accuracy. For the query part as a first step in our research an available application called Solocator is employed to acquire image along with sensors data. We collected 273 query images with EPE and heading tags in the most densely inhabited areas of Chicago and in different conditions such as low illumination and from under rail tracks by different phones such as Nexus 6, Samsung Galaxy S4 and iPhone 5C.

Also in our query database we removed images with low position error since the GPS was accurate enough for localization. Only 228 images remained in our database with highest error. Later an Android application is developed to be used for real-time application. More details is covered in next subsection.

4.1.1 Creating client-server app for Android cellphones

As is mentioned earlier we used available application (Solocator) to create our query dataset. A sample image by Solocator shown in Figure 28. In order to send image and related sensors data directly to a server an Android application and a receiver for the server have been developed to do below steps.

1. Acquire latitude, longitude, and Estimation of GPS error for every second. Android GPS uncertainty (Error) can be extracted by location.getAccuracy() function.
2. Assign a name including all above data to an image when it is taken.
3. Send image to a server based on TCP-IP protocol.
4. Server is listening to get new image. Upon receiving, the image retrieval engine should be lunched.
5. The best match should pass a criteria to be considered as a correct match. Then its position can be considered as estimation of the query position. Further steps can compensate resulting error by using epipolar geometry, structure from motion and so on.

Having access to 4G network, it takes couple of seconds to send image to the server. To make it faster re-sizing of image can be considered. We found image with resolution 400×600 is enough.

4.2 Using GPS error and Homography verification

4.2.1 Estimated Position Error (EPE) defined search space

In recent years, several methods have been proposed to make the position data more reliable for mobile device users. Besides receiving position data directly from GPS module, other sources such as WiFi and nearby cell towers are used to compensate some shortcoming of GPS. However usually the highest accuracy comes from GPS and not from cellular towers. In fact GPS needs time to achieve high accuracy while using rough position through cell towers and algorithm such as triangulation make it possible for GPS to focus on smaller region and give fast result. On this observation we decided to take the uncertainty of the GPS into consideration. According to (Wikipedia contributors, 2018) the standard deviation of the error range of a receiver position estimate is the product of the appropriate Dilution Of Precision (DOP) and User Equivalent Range Errors (UERE). The Estimated Position Error (EPE) is defined as:

$$EPE = DOP \times UERE \tag{4.1}$$

Standard DOP consists of HDOP, VDOP, PDOP and TDOP which are respectively Horizontal, Vertical, Position (3-D) and Time Dilution of Precision. In our scenario Horizontal DOP (HDOP) become our major concern and can be simplified as:

$$EPE \approx HDOP \times UERE \tag{4.2}$$

In the real-world applications, HDOP and UERE can be directly obtained from GPS module.

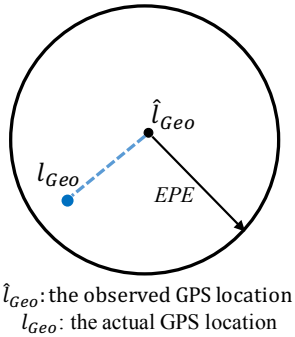


Figure 29: Estimated Position Error (EPE) Defined Search Space

4.2.2 Search Space Analysis

Let $\hat{l}_{Geo} = [l_{lat}\ l_{long}\ l_{height}]$, where l_{lat} refers to the geodetic latitude, l_{long} to geodetic longitude and l_{height} to height of a target point respectively. The geo distance D_{Geo} between two locations or images l_1 and l_2 is calculated as follows:

$$D_{Geo}(l_1,l_2) = \cos^{-1}(\sin(l_{lat1})\sin(l_{lat2}) + \cos(l_{lat1}) \cos(l_{lat2})\cos(l_{long2} - l_{long1})) \times R \tag{4.3}$$

where R is earth radius that is approximately 6371 kilometers. In the meantime, we also define $D_{Heading}$ as a heading difference between query and dataset images. Also, denote the geometry information obtained from GPS as l_{Geo}. Paper (Massoud Sharif and others,) shows the distance between the estimated GPS location $\hat{l}_{Geo}$ and the actual GPS location l_{Geo} can be controlled within a certain limited

error range with sufficient confidence level shown in Figure 29, such that the Euclidean norm satisfies the condition:

$$\|\hat{l}_{Geo} - l_{Geo}\| < EPE \tag{4.4}$$

Nowadays, most devices such as smart phones can acquire a rough position estimation through cell tower, which facilitates GPS to focus on a smaller region and results in a faster and more accurate localization. In fact, in the real-world applications, the smart phone uses cellular network or WiFi for the initial settings. Considering the different mobile operation system platforms, our experiment shows the maximum position error is close to 100 meters as claimed in (Wikipedia contributors, 2018).

Algorithm 1 Image Candidate Selection Algorithm

Input: All the images $\mathbf{S} = \{s_1, s_2, \ldots, s_{N_d}\}$ with size N_d, Geo and Heading data of a query image s_q.
Get EPE of a query image
for $s_i \in \mathbf{S}$ **do**
    if $D_{Geo}(s_i, s_q) < EPE'$ **then**
        if $D_{Heading}(s_i, s_q) < \phi$ **then**
            add s_i to $\tilde{\mathrm{S}}$
        end if
    end if
end for
Output: find $\tilde{\mathrm{S}}$ as the set of image matching candidates for the query image s_q

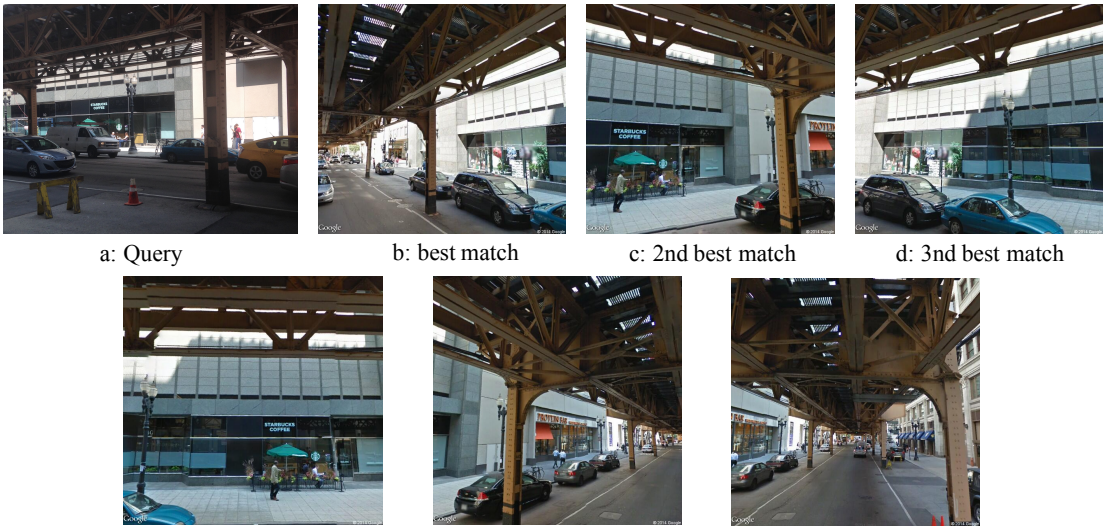


Figure 30: 6 sample retrieved dataset images

Although we know that Google is using really accurate system for collecting data, for example Differential-GPS, but it would be better to consider bigger region to compensate any unlikely error. So we defined EPE' that is more than EPE. Here we considered

$$EPE' = EPE + 10 \tag{4.5}$$

The number of resulting coordinates are directly associated to EPE that representing GPS accuracy.

Another parameter used here is querys heading. The only images with heading in the range of ϕ degree have been selected for next step. Algorithm 1 describes different steps for finding the best

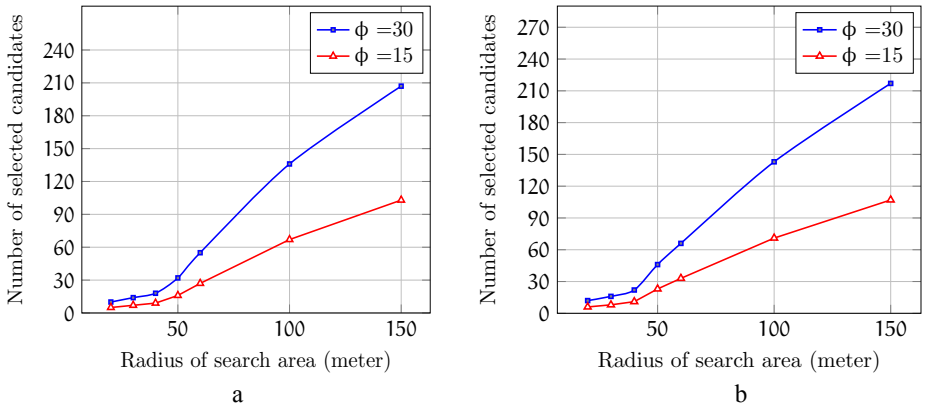
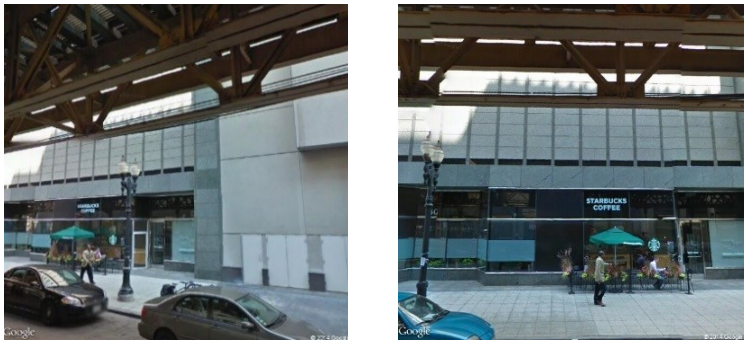


Figure 31: The Number of Candidate Images VS. Distance for ϕ =30 (blue) and 60 (red) for a: Urban area of Chicago and b: dense area of Chicago

candidates. For example six closest candidates for the query image of Figure 30 are shown in Figure 30.b to f.

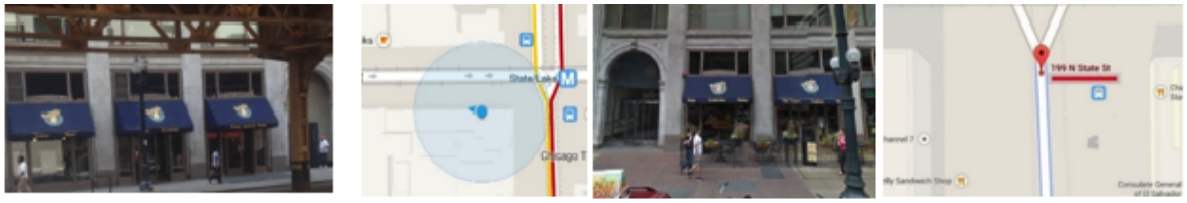
In next step all of those candidate images are fed to Homography verification algorithm. Before applying RANSAC, the best match features have been found with considering Low criteria (Lowe, 2004) when $T = 1.5$

Also different values for ϕ is considered. Plots in Figure 31 show the relationship between number of candidate images and search radius with two different value of ϕ for both a dense and sparse urban area of Chicago. It can be infer that the number of image is almost the same for this two regions. So if an approximation of heading is available, ϕ=15 can be selected for limiting number of images feeding to Homography verification. Figure 32 presents the result of retrieval for identical EPE and different



a: Result for EPE=40 and ∅ =60 b: Result for EPE=40 and ∅ =15

Figure 32: Sample results for the identical EPE and different ϕ, a for $\phi = 60$ and b:$\phi = 15$



(a) query (b) Coordinate of query (c) Best match (d) Modified position

Figure 33: Sample position correction for a successful retrieval

ϕ for query in Figure30.a. As it is clear both results for ϕ =15 and ϕ =60 are correct. Although the quality of images in our database is not really high but it seems enough for our research. To generate features we used Vlfeat library (Vedaldi and Fulkerson, 2008) and DSIFT(Vedaldi and Fulkerson, 2008) as a descriptors. Before applying feature extraction all images are re-sized to 300×400 pixel. To exceed

our algorithm parallel toolbox of MATLAB is used to accelerate feature extraction procedure. By using proposed method, finding the best match is the same for limited regions and city scale problem.

Consider Figure 33.a. There are 53 candidate images for this query. Applying Homography verification on those candidates gives us the result in Figure33.c. If we use directly coordinate of the best match, we have the position in Figure 33.d that is really close to our real position. Another refinement step based on fundamental matrix would be applied when higher accuracy is demanding.

Considering data from GPS along with heading from sensors, make it possible to modified Coordinate in even difficult conditions like sample image shown in Figure 33. Although our proposed method seems to be successful in most of cases, but it fails in some of samples especially when the quality of images are not good in dataset. Another source of error is the limited search space due to opting search radius close to EPE. In fact we found the process for some cases are not successful because of Homography verification but for narrow search space. This is because either EPE may be noisy or not set for 100 percent confidence in our mobile device. In addition Considering higher value for search space returns higher number of candidates that make it difficult to apply Homography verification for all of them. So in the next step we are going to show how the optimum search space must be selected. Then, recent image retrieval approaches based on BOF by employing state-of-the-art feature description, clustering and distance metric have been studied. To evaluate effectiveness of our method some sample images with different cameras have been taken in dense area of Chicago where there were train tracks in top of us. Also heading and coordinate of camera are saved simultaneously.

As formentioned in 4.2.1, considering search space almost equal to EPE is not always the best solution. Also by considering bigger search space, the number of images is going to be increased

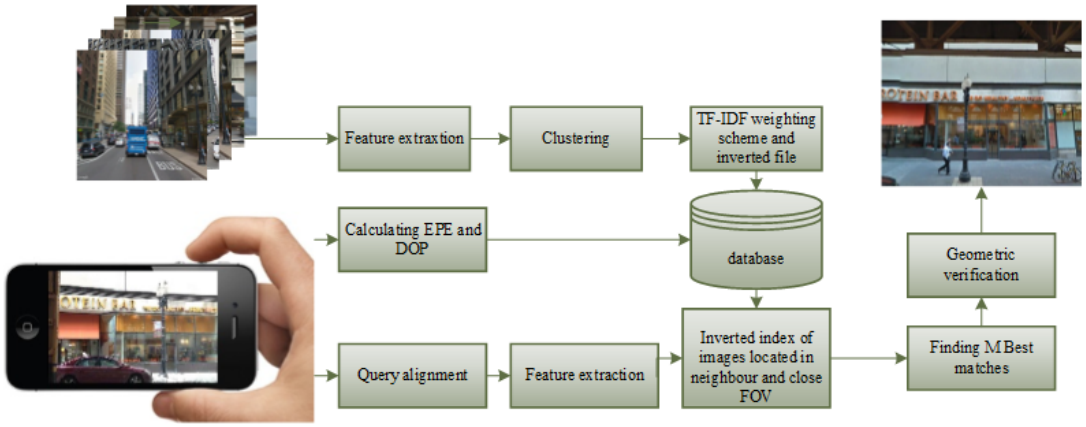


Figure 34: Proposed System for localization using EPE-assisted image retrieval

that make it difficult to directly apply Homography verification. To achieve better performance we have employed common image retrieval approaches while tried to find the optimum search space. The algorithm is similar to Algorithm 1. Here different search space have been studied by introducing a new parameter, Th. Algorithm 2 shows the steps we take initially to preprocess the image search space. After acquiring the EPE from the query image s_q, the most relevant image candidates in the database $\mathbf{S} = \{s_1, s_2, \ldots, s_{N_d}\}$ with size N_d should be selected. ϕ is used to filter out image based on heading. For example $\phi = 30$ means only considering images in database when the heading difference between query and database image is lass than 30°. Th should be considered more than 1 to cover a region bigger than EPE defined region (we already have result for $Th = 1$). This is because the EPE returned from mobile devices is not for 100 percent confidence. This chapter is focused on finding the best value for Th by considering EPE from a mobile device such as Android cell phone. Figure 31 shows the number of

candidate images versus search radius when $\phi = 30$. Experimental results show that the computational cost for filtering images by approximate GPS coordinates and heading is negligible.

Algorithm 2 Image Candidate Selection Algorithm

Input: All the images $\mathbf{S} = \{s_1, s_2, \ldots, s_{N_d}\}$ with size N_d, Geo and Heading data of a query image s_q.
Get EPE of a query image
for $s_i \in \mathbf{S}$ **do**
    if $D_{Geo}(s_i, s_q) < Th \times EPE$ **then**
        if $D_{Heading}(s_i, s_q) < \phi$ **then**
            add s_i to $\tilde{\mathrm{S}}$
        end if
    end if
end for
Output: find $\tilde{\mathrm{S}}$ as the set of image matching candidates for the query image s_q

4.3 Image Matching Procedure

There are variety of approaches for finding similarity between images. Although in this chapter the focus was finding the best value for the search radius, we have tried to apply newest techniques in the image retrieval engin of our place recognition system. It is clear that the result of this work can be

improved with any state of the art image retrieval techniques such as (Torii et al., 2013). Our task for implementation of BOF is as follow.

1. Find the RootSIFT feature for all images in a database.
2. Cluster features using Approximate Nearest Neighbor (ANN) with branching factor of 10 and depth of 5 that returns 10^5 clusters (visual words).
3. Find the closest visual word (cluster center) for each feature in database images and represent each image by a vector showing the frequency of each visual word
4. Find the best match based on the score obtained for the database image using inverted file algorithm.

Figure 34 shows our proposed system block diagram.

In our implementation of tasks 3 and 4, multiple methods, Regular TF-IDF, Binary TF and Burstiness (Jégou et al., 2009) are adopted. As mentioned before query images in other available datasets such as San-Francisco do not have information about uncertainty of the GPS. So we are not able to compare our proposed methods to other reported results for those particular datasets. Indeed we have implemented algorithm the same as Burstiness to show how our proposed method can improve other algorithm. The whole process can be described with below steps. As has been mentioned before we build a quantizer map, which quantizes the value of a feature x to $q(x)$.

Then a match function is defined as:

$$match_{x,y} = \begin{cases} 1 & \text{if } q(x) = q(y) \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

Given a query image and its feature descriptors, we search each image in the database and compare the similarity of their features by examining :

$$match_{q(x_{i,d}),q(y_m)} \tag{4.7}$$

where $x_{i,d}$ is the i-th feature of the image d in the database and y_m is the m-th feature of the query image. The score of image d will be evaluated as:

$$score_d = \sum_{m=1}^{n'} \sum_{i=1}^{n_d} match_{q(x_{i,d}),q(y_m)} \tag{4.8}$$

where n' and n_d are the total number of features of the query image and database image d respectively. Here TF-IDF weighting scheme should be applied to consider importance of each visual word. For TF-IDF we have used the most common formula of:

$$w_{xd} = tf_{xd} \times \log\left(N_d / df_x\right) \tag{4.9}$$

where N_d is the number of images in database, df_x is number of images containing x and tf_{xd} is the frequency of term x in image d.

Although our contribution in this work is about adaptive selection of search space not new image retrieval algorithm, but multiple approaches have been used to evaluate the result. So besides regular TF-IDF method, two other methods have been utilized to damp the effect of the repetitive features. The first one is the same as Burstiness algorithm (Jégou et al., 2009) while the last one that is called Binary weighting just consider presence of each visual word not the occurrence frequency. Note that when we adopt the inverted file method the algorithm uses the sparsity property of visual words in the query. Therefore the formula above can be simplified as:

$$score_d = \sum_{x=1}^{w_n} n'_x w_{xd} \tag{4.10}$$

where w_n is the total number of distinctive visual words of query, n'_x is the occurrence frequency of a visual word x in query. For Binary TF-IDF and burstiness tf_{xd} and n'_x are defined as:

$$tf_{xd} = \begin{cases} K_t & \text{when image } d \text{ has more than } K_t \text{ visual word } x \\ tf_{xd} & \text{otherwise} \end{cases} \tag{4.11}$$

$$n'_x = \begin{cases} K_n & \text{when query image has more than } K_n \text{ visual word } x \\ n'_x & \text{otherwise} \end{cases} \tag{4.12}$$

K_n and K_t should be limited to damp the effect of features with higher frequency of occurance. They can be set to a value like 6 to act such as Burstiness algorithm or 1 to have binary TF-IDF. The Binary

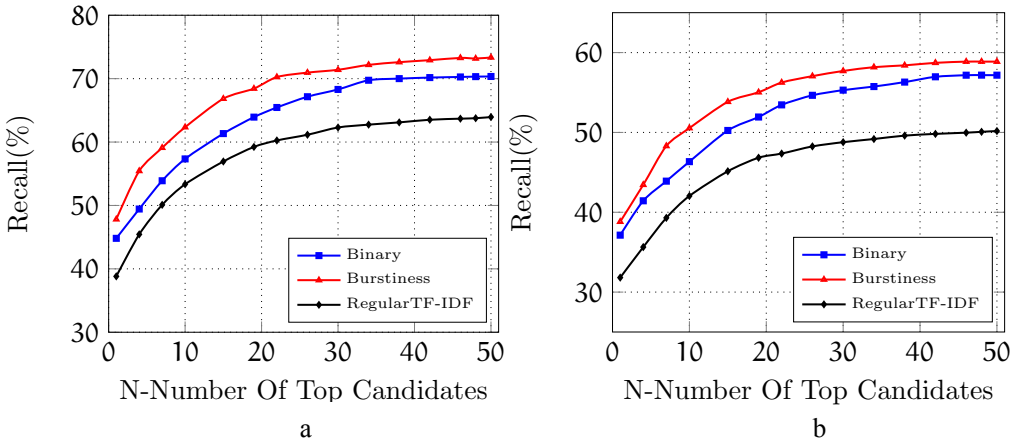


Figure 35: Recall vs number of best candidates for a: dataset A and b: dataset B

TF-IDF just considers 1 for visual words that are exist. As shown in Figure 35 the performance of this method is lower than Burstiness, But it is better in terms of memory.

The next step for all schemes is to choose the image with the maximum score, that is:

$$\underset{d}{\text{argmax}}\ score_d \tag{4.13}$$

This score should be calculated just for images from the subset $\tilde{S}$. Next we select 50 image candidates with the highest score. Finally the images with the highest score are fed to another step for Homography verification.

Next section discuses about performance of different methods and optimizing the search space to achieve better result.

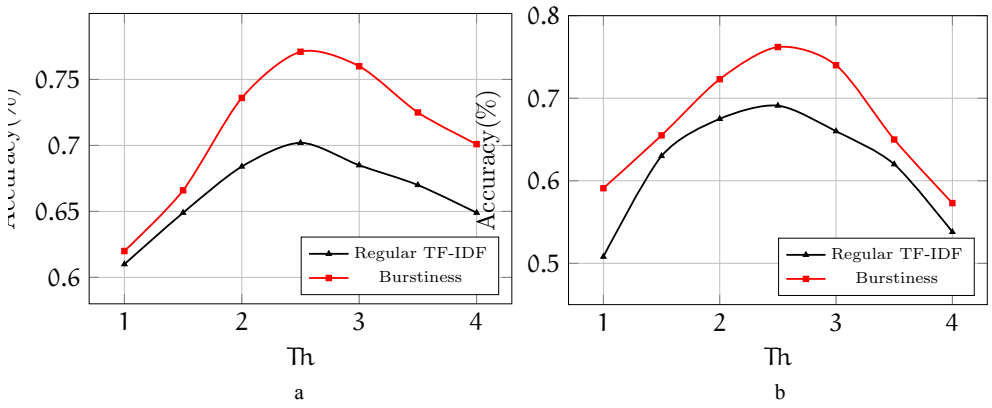


Figure 36: Accuracy vs Th for a: dataset A and b: dataset B

4.4 Performance Evaluation

Position of a smart phone is determined with reasonable accuracy in less than 5 seconds. As mentioned before, we have used 2 databases to evaluate how size of the database affect the performance. The ANN is built from the 54k images in dataset A. This vocabulary tree also is used for evaluation of dataset B. All of queries are taken from regions which are covered by both datasets. Figure 35 shows the recall vs the number of selected candidates. As can be seen the highest recall for both datasets achieved by Burstiness while even Binary scheme is better than regular TF-IDF. Now we are going to improve the recall by considering adaptive search space.

To find the optimum value of Th, retrieval performance for different values of Th was investigated. To evaluate the retrieval performance recall for $N = 1$ has been considered to see whether the retrieval was successful for the best match or not. Here we have defined Recall for $N = 1$ as a accuracy of the pipeline. The first step here is finding the best match for each query image that contains 228 images. The

next step is finding how many of those best matches are relevant to their corresponding query. Suppose only 137 images in best match are relevant to their corresponding queries. So the Accuracy is going to be $\frac{100}{228} = 0.6$. Figure 36 shows the accuracy versus Th for dataset A and B for Burstiness method respectively while ϕ=30. For both datasets the best performance is achieved around Th=2.5 while the performance for Th=1 is low. This can be attributed to the level of confidence of GPS accuracy used in the cellphones. The error occurs because either the correct match is outside the search area or an irrelevant image within the search areas produces the best match. Thus there is a trade off between (i) searching too few images in a small search area with a higher likelihood that the correct match is outside the search area and (ii) searching too images in a large area with increased likelihood of having the best match yield an incorrect result based on the search criteria. Note that it is highly unlikely that the query image is identical to any specific image in the database. An enlarged region yields an increased likelihood of a match up to a point beyond which too many matching candidates may lower the chances of a correct match.

The errors occur because the best match in some cases is in error since the query images do not have a perfect match in the database and an incorrect location may yield the best match. It should be noted that the accuracy of the retrieval system without considering prior knowledge form GPS is actually equall to accuracy while $Th = \infty$. This value for dataset B and for burstiness algorithm is less than 0.4. Also considering a fixed radius for search such as (Chen et al., 2011) returns lower recall in comparison with dynamic search space extracted from optimum value from EPE. It is important to note that the accuracy of retrieval is not significantly different in the two databases of different sizes so long as the optimum threshold value Th is used. As a result, our method allows the use of a single-vocabulary tree

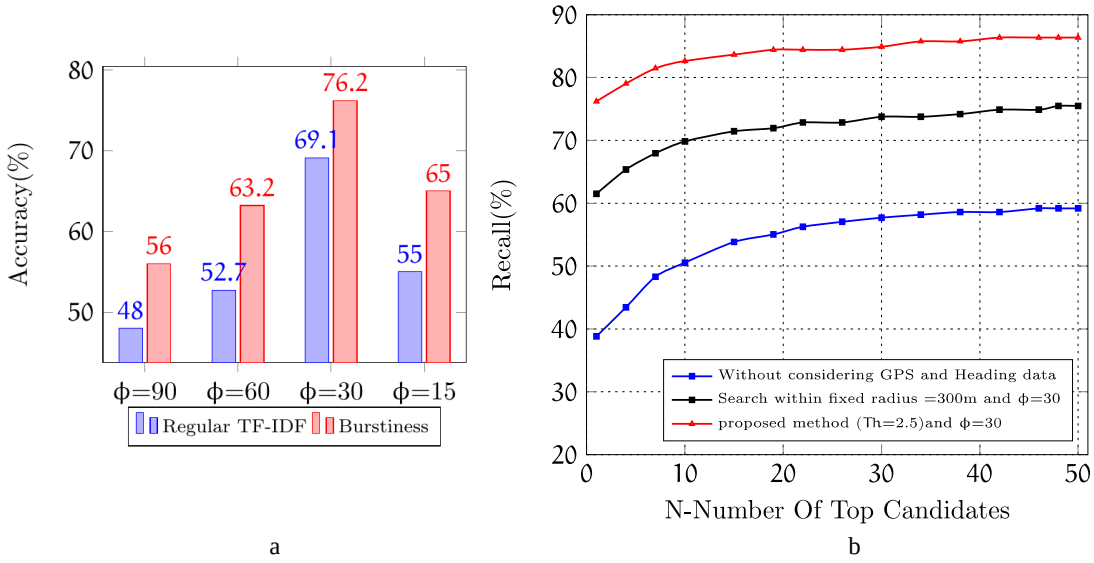


Figure 37: a: Accracy vs ϕ for different algorithms, b:Recall vs number of best candidates for dataset B for burstiness algorithm

for a city-scale database. In other works prior knowledge from GPS or cell towers is used to find out in which partition the search should be done. For example (Ankita et al., 2008) used GPS signal to search local vocabulary trees for visual loop closing problem. Those methods may need to search more than one region by splitting the database to smaller regions. These sub-regions are in the range of kilometers resulting in a much larger number images to be searched. Since the radius of our search area is usually less than 200m, the number of images searched in our method is much less than in other methods.

Dynamic adjustment of search radius in our method allows for better performance since a fixed radius would require the worst-case choice of radius. In addition to search space, different values of ϕ have been evaluated while $Th = 2.5$. The accuracy results are given in Figure 37a. We found that when ϕ is set to 30 degrees, all schemes show the highest accuracy. That is why we have used this value in our system. Since our idea of considering EPE is new, we were not able to use other available databases for landmark recognition for comparison. This is because the query of those databases have nothing related to GPS error or EPE. This necessitated making our own database and creating an app to save related data of query. Those datasets are available upon request. In Figure 38 two results from dataset B, both with successful retrieval for 2 different queries and related EPE are shown. For the Figure 38.a although the retrieval is successful, it is not necessary to use visual search since the GPS is accurate enough. For this sample EPE is about 15 m so the pure GPS data is reliable. But for the Figure 38.b EPE is about 86 (since it is taken below the rail track), so it is better to use visual based location estimate. Here the system search a region with radius of $86 \times 2.5 = 215m$ that contains 589 images. This is almost similar for most of the cases with high EPE. So by utilizing GPS uncertainty and heading for this particular case, only 589 images should be searched instead of $92k$ images. This reduction in the number of the images for the search is not possible for other proposed methods. Moreover we have tried to see for how many of the queries the actual position is out of the EPE defined search space. We found the actual position of only 2 queries from 228 were out of the search region that is equivalent to less than 2 percent. It means most of the failures are from retrieval part. So considering more sophisticated image retrieval algorithms along with utilizing our adaptive search region would result in great performance.

a: Sample query 1 with EPE=15 b: Best match

c: Sample query 2 with EPE=86 d: Best match

Figure 38: Sample queries in the urban area and their matched

Although we reached to street level accuracy, we are working on extracting actual position of query by considering camera model and location of one or couple of best retrieved images.

4.5 Conclusion

In this chapter a method based on GPS uncertainty is proposed to improve image retrieval performance in visual localization applications. Noting that useful sensors are embedded in most of mobile devices, a new solution for localization based on restriction on search area by using EPE from a smart

Phone is exploit to improve the performance of visual recognition engine. Experimental results show the optimum search space and related effectiveness of our proposed method. The performance would be better if we use higher quality images in database. Modifying coordinate by fundamental matrix between query and best match or by using SFM is another steps that can be taken into account. The ultimate goal of our research is finding an approach to help more accurate device for localization. This research is primary step toward our goal and may need other techniques such as Visual Odometry as a complementary method.

CHAPTER 5

IMPROVED IMAGE-BASED LOCALIZATION USING SFM AND MODIFIED COORDINATE SYSTEM TRANSFER

5.1 Introduction

As described in last chapters, although the main problem in finding accurate location of a mobile device based on visual information is designing a reliable image retrieval engine, but accurate localization also requires an extra procedure that considers the transformation between the query image and dataset image candidates. This is because relying on only the best matching image returns a middle-of-the street level of accuracy and not the actual query camera location where is typically on the sidewalk. In this chapter we propose a new method for reliable estimation of the actual query camera location by optimally utilizing structure from motion (SFM) for 3D camera position reconstruction, and introducing a new approach for applying a linear transformation between two different 3D Cartesian coordinate systems. Since the success of SFM hinges on effectively selecting among the multiple retrieved images, we propose an optimization framework to do this using the criterion of the highest intra-class similarity among images returned from retrieval pipeline to increase SFM convergence rate. The selected images along with the query are then used to reconstruct a 3D scene and find the relative camera positions by employing SFM. In the last processing step an effective camera coordinate transformation algorithm is introduced to estimate the query's geo-tag. The influence of the number of images involved in SFM on

the ultimate position error is investigated by examining the use of three and four dataset images with different solution for calculating the query world coordinate. We have evaluated our proposed method on query images with known accurate ground truth. Experimental results are presented to demonstrate that our method outperforms other reported methods in terms of average error.

5.2 Problem

Suppose we have an image retrieval engine and have access to sensory data to improve the retrieval task. An additional step, called homography verification performed by applying algorithm such as RANSAC (Fischler and Bolles, 1981) and its variants, (Torr and Zisserman, 2000), (Moisan et al., 2012) are used to re-rank candidates returned by search engine. This step compensates for the weakness of image retrieval schemes based on BOF where the geometric information of images is ignored. By applying this procedure, more relevant candidates will be returned in the first N retrieval images.

After a small set of best-matching images has been collected for a given query image, the next task is to estimate the query's accurate location. Untill now the best estimate for the query location is the GPS tag of the best match. In order to acheive higher location accuray, more images should be incorporated. This is because each image has its GPS tag and likely some common features with the query to be used for coordinate transform. Multi-view Structure From Motion (SFM) for the reconstruction of 3D camera poses from 2D-2D correspondences, or from 3D-2D correspondences can be used in this case (Min et al., 2014). Recent state-of-the-art approaches in this field such as (Xu et al., 2012), (Vishal et al., 2015), and (Zamir and Shah, 2010), find a similarity matrix from two query-matching images. These approaches utilize SFM to estimate three camera positions: two positions of two cameras corresponding to the two best-matching images and the query camera position, yielding a triplet of reconstructed 3D

camera positions. Numerous triplets are typically generated (multiple matches with the query) and are subsequently processed by a least-squares fitting routine in order to compute the similarity matrix and generate a unique estimate of the query's location. They also reduce the 3D position vectors to 2D position vectors by dimensionality reduction techniques such as PCA. Based on their results the ultimate error range is still high which makes its use difficult in navigation. For example, we noticed that for some queries in different intersections, the estimated positions are found to be on the opposite side of the street from the actual position which makes navigation hard. A key limitation of currently used methods is using multiple SFM processing on pair of images returned by the retrieval pipeline along with the query which is computationally expensive. Our focus is using a single SFM on a subset of images from the retrieval with the highest similarity. So we formulate the image selection as an optimization problem. Then we proposed a method to directly find camera coordinate transformation parameters between camera relative centers from SFM to real world coordinates as described in the following sections.

5.3 Problem formulation for optimal selection of images for SFM

We now consider the framework for formulating the problem of optimally selecting a subset of retrieved images as input to SFM process. We first briefly describe the method we use for image retrieval to obtain N matching images from which a trimmed subset of k images is optimally selected for SFM implementation. Typically N may range between 10 to 50 whereas the choice of k is either three or four.

5.3.1 Retrieval of N Images

We first obtain N images that best match a query image. For this purpose several image retrieval methods may be employed. The main component of most image retrieval methods is the Bag Of Features

(BOF) technique. In this approach, each image is represented with a vector containing the occurrence frequency of features (visual words). There are a variety of features such as SIFT, SURF or a normalized version of the SIFT called RootSIFT that have shown better performance. The query vector should be compared with all dataset vectors to find the most similar image. It is important to mention that the goal of our research is primarily on finding a better estimate of query position extracted from multiple matches from the dataset, and, not on improving the image retrieval engine itself. Any suitable method with good retrieval performance can be used for this stage.

As mentioned earlier, images can be represented by visual words, but the importance of the words varies. This importance is captured in the assigned weights using the Term Frequency-Inverse Document Frequency (TF-IDF). The weight of the visual word α in image i is

$$t_{\alpha,i} = f_{\alpha i} \times \log(\frac{N_{db}}{N_{\alpha}}) \tag{5.1}$$

where $f_{\alpha i}$ is the frequency of term α in image i, N_{db} is the number of images in the dataset and N_{α} is number of images containing visual word α. For each visual word α, note that the Inverse Document Frequency (IDF) is defined as

$$IDF(\alpha) = log(N_{db}/N_{\alpha}) \tag{5.2}$$

Let η be the number of visual words and F_q=[$f_1^q f_2^q ... f_\eta^q$] and F_{db}=[$f_1^{db} f_2^{db} ... f_\eta^{db}$] be the frequency of visual words $\alpha_1, \alpha_2, ..., \alpha_\eta$ for query and a dataset image, respectively. The jth, element F_q or F_{db} are the number of times feature descriptors of the query and a dataset image have been assigned to visual word α_j. The similarity between query and a given dataset image (vectors) can be computed by Eq. 5.3.

$$SIM(I_q, I_{db}) = \frac{\sum_{\alpha=1}^{\eta} IDF(\alpha)\ min(f_\alpha^q, f_\alpha^{db})}{(\sum_{\alpha=1}^{\eta} IDF(\alpha) f_\alpha^q)(\sum_{\alpha=1}^{\eta} IDF(\alpha) f_\alpha^{db})} \tag{5.3}$$

The above similarity measure is different from the commonly used Cosine similarity measure. It is experimentally observed that it produces more robust results than the Cosine similarity measure. Our procedure for implementation of the basic image retrieval engine consists of the following steps:

1. Find the RootSIFT features for all images in a database.
2. Cluster features using the Approximate Nearest Neighbor algorithm (ANN) into η clusters (visual words).
3. Find the closest visual word (cluster center) for each feature in database images and represent each image by a vector showing the frequency of each visual word.
4. Apply TF-IDF using Eq. 6.2 and normalize the vectors.
5. Find the best *N* matches based on the score obtained for the dataset image using distance criteria.
6. Re-rank *N* closest images based on homography verification by applying RANSAC.

In order to achieve higher recall we used the Adaptive Assignment algorithm (Torii et al., 2013). This algorithm, which assigns different number of visual words to different features improves recall. It is worth mentioning that any method could be used for the image retrieval pipeline. We can further improve the result by considering prior knowledge of the location from GPS as described in 5.3.2.

5.3.2 Considering Prior Knowledge Of Location From GPS

As mentioned in previous sections, the result of retrieval should be fed to our proposed method for query geo-tag estimation. One option to achieve a better result is taking prior knowledge from the query

position into account. Some reported studies have used noisy location data. For example in (Zhang et al., 2011) coarse location data from cellular tower and triangulation are used to limit the search region. Another option for narrowing down the search space is considering maximum error of the GPS which is denoted here as R_{max}. Suppose the GPS coordinates of two images I_1 and I_2 are given by (θ_1, ϕ_1) and (θ_2, ϕ_2) where θ_i and ϕ_i are the latitude and the longitude for image i. The Geo-distance between locations of these two images is computed by Eq. (5.4).

$$D_{Geo}(I_1, I_2) = \cos^{-1}(\sin(\theta_1)\sin(\theta_2) + \cos(\theta_1)\cos(\theta_2)\cos(\phi_2 - \phi_1)) \times R_e \tag{5.4}$$

where R_e is the radius of the earth that is approximately 6371 kilometers. The search space can be limited to those images located in a circle with the radius of R_{max}. The procedure to limit the search space for the query image I_q is described in Algorithm 3.

For the San Francisco dataset the maximum reported error is 300 meters. Based on our experience in Chicago, the error in the position estimated by a smart phone such as iPhone 5, iPhone 6, Nexus 6, or Galaxy S6, is typically less than 100 meters. This is because those phones benefit from other sources of data such as cellular towers and inertial measurement unit (IMU). The search space therefore turns out to be smaller for real-world applications.

The final common step in most of image retrieval algorithms is the application of geometry verification based on RANSAC to re-rank the limited number of candidate based on the number of inlier

Algorithm 3 Image Candidate Selection Algorithm

1: **Input:** Set of N_{db} images $\mathbf{S} = \{I_1^{db}, I_2^{db}, \ldots, I_{N_{db}}^{db}\}$, R_{max}, I_q

2: **Output:** $\tilde{\mathrm{S}}$ which is the set of all images in the region limited by R_{max}

3: **for** $I_i^{db} \in \mathbf{S}$ **do**

4: **if** $D_{Geo}(I_i^{db}, I_q) < R_{max}$ **then**

5: add I_i^{db} to $\tilde{\mathrm{S}}$

6: **end if**

7: **end for**

8: **Output:** find $\tilde{\mathrm{S}}$ as the set of image matching candidates for the query image I_q

features. This step mitigates the weakness of systems based on BOF which ignore the geometric information of features. To go forward and estimate the actual position of the query, more than one image is needed. This is because estimating the camera position by using just a single image and considering fundamental matrix between the query and the best match, even when models for both cameras are available, would not be accurate enough for our purpose. For our proposed 3D coordinate transformation method at least three candidates with distinct GPS tags are required. To acquire candidates that are most similar to a query, criteria such as the number of inliers between the query features and the candidate features can be considered for the re-ranking and removing irrelevant candidates. Along with this criterion, another suitably devised step should be applied to ensure return of the best candidate

images with distinct GPS tags and highest intra-class similarity. The procedure for optimally selecting candidates is discussed in the next sub-section.

5.3.3 Optimum Selection among the Retrieved Images

Suppose N images with location coordinates $g_i, i = 1, ..., N$ are selected after re-ranking. We wish to select k images out of N, where k is preferably four. If that is infeasible, then k equal to three images may be selected if possible. A simple way to select k images is to find images with distinct GPS tag and select k images with the highest number of inliers. Such a set of images is not necessarily the best choice for SFM processing since the selection relies only on the number of pairwise matches (inliers) between the query and all candidates while the number of matched features between each pair of candidate images is not taken into account.

It is important to note that a set of candidate images is the best choice when each member of this set shares the highest number of common features with the other members. In our case, while multiple images per location exist, we seek a method that optimally selects the set so that each member of the set has the highest consensus on common features with other members as well as with the query image. The solution is facilitated by defining a pairwise dissimilarity measure, w_{ij}, between distinct image i and j. An undirected graph $G = (V, E, w)$ with vertices $V = 1, 2, ..., N$ corresponding to image $I_1, I_2, ..., I_N$ with location $g_1, g_2, ..., g_N$, the set E of edges, and the set w of weights can then be created. By this definition, the more similar images will have the lower w_{ij}. Now the problem is to find a subset $G^\star = (V^\star, E^\star, w^\star)$, $V^\star \subset V$, $E^\star \subset E$, with k vertices, $k < N$, that minimize the total weights:

$$V^{k\star} = \underset{V^k \subset V}{\operatorname{argmin}} \sum_{\substack{i,j \in V^k \\ g_i \neq g_j \\ i \neq j}} w_{ij} \tag{5.5}$$

Here V can be partitioned into clusters with distinct GPS-tags. We now devise a solution to the problem of optimal selection of k images using the framework just described.

5.4 Implementing Solution To Optimal Image Selection For SFM

The problem of finding an optimal subset from a set has been studied extensively during last years (Katoh et al., 1981), (Fischetti et al., 1994). Since there likely to be a chance of multiple images per location, the algorithm should only select one image per location. We therefore employ the General Minimum Clique Problem (GMCP) to select one image in each cluster containing images with identical GPS-tags. In the following subsection we describe how our problem is formulated and solved by GMCP.

5.4.1 Candidates Selection By GMCP

In order to formulate and solve our optimal selection problem using GMCP, we start with N best retrieved images with world coordinates (GPS-tag) g_j, $j \in \{1,...,N\}$, not necessarily distinct. Let h be the total number of distinct or unique location coordinates. The N candidates are then grouped into clusters $\{V_1,...V_h\}, h \leq N$ with an identical GPS tag. So an arbitrary cluster V_r, $1 \leq r \leq h$ contains different number of images associated with world coordinate g_r. With this partition of images, some clusters may contain only a single image meaning that the retrieval returned only one image for that location. Also h is usually larger than k (k is preferably 4) which exceeds our need of images for the

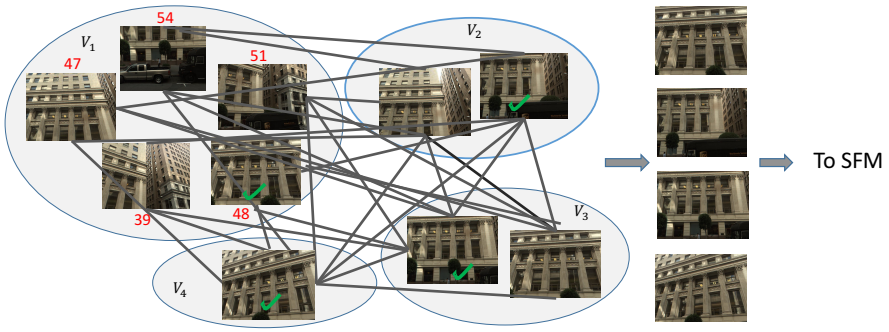


Figure 39: Candidate selection by GMCP. Images with similar GPS-tags are placed to the same cluster. For cluster V_1, the number of inliers between each member and the query is shown in red. For each cluster only one image is returned by GMCP and is shown with a green check mark (edge weights are not shown in this figure). Note that for the cluster V_1, in our approach the two images with higher number of inliers (54 and 51) were not selected unlike the scenario in which the maximum number of inliers is the only criteria for image selection in each location

next step. One possible solution is to keep first $k = 4$ clusters and find all images with the highest similarity. We choose to keep more clusters and then select only k images with the highest score from the result of GMCP. In order to solve our problem, for each member of all clusters, a similarity measure between image $i \in V_x$ and $j \in V_y$ where $x \neq y$ should be calculated. The number of inliers between a pair of images derived from geometry verification is a strong indicator of similarity. In the last steps, we only found the number of inliers between the query and limited number of candidate images. Applying geometry verification between each pair of candidates would be practically infeasible since it would require an unacceptable amount of time. In order to avoid this time complexity we propose the use of vectors containing frequency of visual words of images as defined in Section 5.3. It is also important

to incorporate the query visual words in computing the similarity between two images. This is because images selected in this stage along with the query should be fed to the SFM pipeline. So a desirable similarity measure should take into account those visual words that are common to two images as well as to the query image. We therefore introduce a query-contextualized image similarity measure. Suppose the vector of visual words for image I is represented by $F_I = \{f_1^I, f_2^I, ..., f_\eta^I\}$. In order to incorporate query visual words in computing similarity, the indices of non-zero visual words of the query are extracted and represented by $I_{nz}^q = \{u_1, ..., u_d\}$ where d is the number of non-zero visual words. We define the similarity between any pair of images i and j by Eq. 5.6:

$$\psi_{ij} = \sum_{k=1}^{d} \Delta(f_{u_k}^i)\Delta(f_{u_k}^j)/(\sum_{k=1}^{d} \Delta^2(f_{u_k}^i))^{1/2}(\sum_{k=1}^{d} \Delta^2(f_{u_k}^j))^{1/2} \tag{5.6}$$

where, for $x \in \mathbb{R}$,

$$\Delta(x) = \begin{cases} 1 & \text{if } x \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

Since $\Delta^2(x) = \Delta(x)$ the denominator in Eq. 5.6 can be reduced to

$$(\sum_{k=1}^{d} \Delta(f_{u_k}^i) \sum_{k=1}^{d} \Delta(f_{u_k}^j))^{1/2} \tag{5.8}$$

This measure calculates the similarity between two images while taking into account the non-zero features of the query. In the next step, all selected images along with the query image should be fed to SFM step. The complexity of computing Eq. 5.6 is low since vectors are already available and summation is

applied for the non-zero features of the query. A convenient measure of dissimilarity between image i and j can be defined by Eq. 5.9.

$$w_{ij} = 1 - \psi_{ij} \tag{5.9}$$

The next step is to find the subgraph $G^\star = (V^\star, E^\star, w^\star)$ with nodes $V^\star = \{v_1^\star, ..., v_h^\star\} \subset V$ where only one node is selected from each cluster, for instance $v_1^\star$ from V_1 and $v_h^\star$ from V_h, and subset of edges $E^\star \subset E$ that minimizes the total dissimilarity that for a feasible solution is:

$$T_{Dissimilarity}(V^\star) = \sum_{m=1}^{h} \sum_{l=m+1}^{h} w_{V^\star(m)V^\star(l)} \tag{5.10}$$

Fig. 39 shows the process of clustering images with only four clusters where the costs of edges are not shown. For the members of cluster one, V_1, the number of inliers between the query and each member is shown in red. In this case, clusters contain different numbers of images. The result of GMCP is shown with green check marks. As shown in cluster V_1, an image with 48 inliers with the query is selected as a best candidate. Note that for the cluster V_1, the two images with a higher number of inliers (54 and 51) were not selected by our proposed method based on GMCP. This is different from the scenario in which the maximum number of inliers is the only criteria for image selection in each location. Without use of GMCP, the candidate selected from the cluster V_1 is the image with 54 inliers. We compare the SFM convergence rate in Section 6.4 for both methods.

5.4.2 Generalized Minimum Clique Problem (GMCP)

Generalized Minimum Clique Problem (GMCP) can be used when the costs of edges are non-negative and graph is $|K|$-partite complete. Unlike a minimum clique problem, GMCP substitutes nodes

with cluster of nodes. In this problem nodes of a given graph are partitioned into disjoint clusters. The goal is to find a subgraph with minimum cost while selecting only one node from each cluster. Each cluster furnishes only one of its nodes to the subgraph. This algorithm has been used recently in Computer Vision for multi-object tracking (Zamir et al., 2012). Suppose we are given a graph $G = (V, E, w)$ with nodes $V = \{v_1, ..., v_N\}$ and these N nodes are grouped into h sets of nodes called clusters $V_1, V_2, ..., V_h$. Note that $V = V_1 \cup V_2 \cup ... \cup V_h$ and $V_x \cap V_y = \varnothing$ for all $x, y \in \{1, ..., h\}$ where $h \in \mathbb{Z}$: $1 \leq h \leq N$ and $x \neq y$. As mentioned earlier, a cost w_{ij} is assigned to the edge between nodes $i \in V_x$ and $j \in V_y$, for $x \neq y$. Now the objective is to find a subgraph $G^\star = (V^\star, E^\star, w^\star)$ with nodes $V^\star = \{v_1^\star, ..., v_h^\star\}$ $\subset V$ which is composed of only one node from each cluster together with associated subset of edges $E^\star \subset E$ that is minimized the total edge cost. For such a problem GMCP can find a feasible solution with minimum cost which is in fact the total weights of all edges in $E^\star$. So based on the formulation of our problem in Section 5.4.1, GMCP can return the subset with highest intra-cluster similarity which leads to a higher convergence rate in the SFM step. In the next section we discus how the selected candidate images are used to estimate the query camera position.

5.5 Query Camera Position Estimation

5.5.1 Estimate Query Location By Four Dataset Images

The image retrieval process selects multiple matching images for a specific query. Each of the matching images has a known GPS tag which is used in our novel procedure for estimating the query camera's location. The proposed method is illustrated in Fig. 40. A key concept in our approach for query GPS tag estimation is the selection of a subset of images with the highest inter-class similarity using GMCP as described in 5.4.1 and then obtaining a $3D - 3D$ coordinate transformation from one

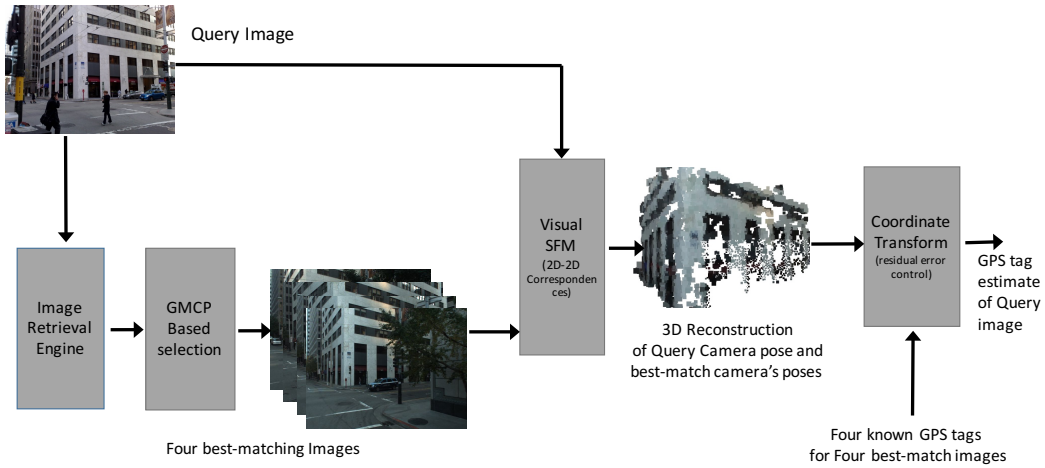


Figure 40: Proposed pipeline for image-based localization (using four matching images from dataset)

$3D$ coordinate system (eg. camera centers in camera $3D$ space as reconstructed from multi-view SFM) to another $3D$ coordinate system (eg. GPS tags in absolute world $3D$ Cartesian space for the same cameras). Fig. 41 illustrates the concept with four cameras (images), with centers C_1, C_2, C_3, and C_4, using four images obtained in the previous step, with the query camera as the fifth camera with center at C_5. C_1 to C_5 represent camera 3D center coordinates which have been reconstructed by multi-view SFM. We use the VisualSFM package (Wu et al., 2011) for this task and extract the coordinates C_1 to C_5 based on four matching images (dataset images) for the given query image. The details of camera center localization with SFM are as follows. Assume that for a given query image I_q a set of h images, $T = \left\{I_{v_1^\star}, I_{v_2^\star}, ..., I_{v_h^\star}\right\}$ $h >= 4$ is returned by the GMCP. Here $I_{v_1^\star}$ is image corresponding to node $v_1^\star$. The corresponding GPS tags for those h images are denoted with the set of locations $L = \left\{P_{v_1^\star}, P_{v_2^\star}, ..., P_{v_h^\star}\right\}$.

The set $\{I_{v_1^\star}, I_{v_2^\star}, I_{v_3^\star}, I_{v_4^\star}, I_q\}$ should then be processed with VisualSFM. Upon convergence to five camera center locations, $C_1, ..., C_5$, the quintuplet $C = \{C_1, .., C_5\}$ is used to obtain absolute world coordinate locations. If fewer than five relative camera centers are returned, SFM does not converge. It is worth mentioning that there would be a possibility to re-run the process using three best candidates as described in Section 5.5.2.

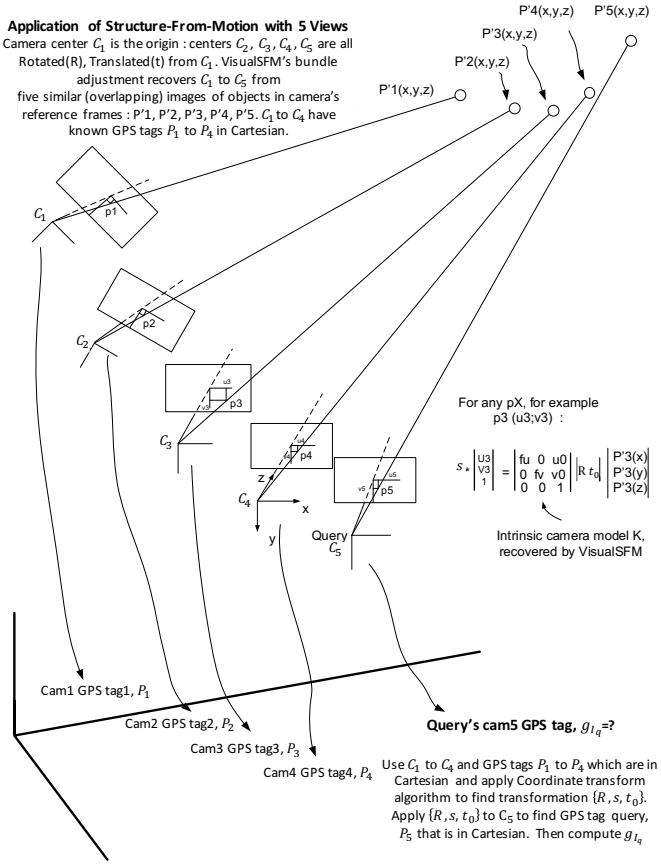


Figure 41: Transformation from camera-referenced 3D coordinate system based on SFM to real world-referenced 3D Cartesian location using four dataset images

In the following, without loss of generality, we have adopted the convention that C_5 in C corresponds to the camera center location for query image I_q. Locations $C_1, ..., C_4$ correspond to the cameras for the matching dataset images. Each camera center location in C is specified with $3D$ Cartesian coordinates in camera referenced space. Before computing the transformation the GPS tag of dataset images should be converted to Cartesian coordinates. The conversion equations are as follows. Assuming the GPS tag contains latitude and longitude pair (θ, ϕ), the coordinates x, y, z are computed by:

$$\begin{aligned} x &= R_e \cos(\theta)\cos(\phi) \\ y &= R_e \cos(\theta)\sin(\phi) \\ z &= R_e \sin(\theta) \end{aligned} \tag{5.11}$$

where R_e is the radius of Earth. Suppose the GPS tags for the four dataset images used in SFM are represented as $P_1, ..., P_4$ in Cartesian coordinates. Algorithm 4 is used for deriving the transformation from camera-referenced to absolute reference coordinates. It uses the values for the matching dataset images, P_1 to P_4 , and their relative locations C_1 to C_4 derived from SFM.

Two final steps are applied after Algorithm 4:

1) compute query's location P_5 (GPS tag in Cartesian coordinates) as $P_5 = t_0 + sRC_5$.

and

2) convert P_5 back to GPS latitude/longitude g_{I_q}.

In step 3 of algorithm 4, points C_1 to C_4 can be considered as points in the left coordinate system. This is a 3D Cartesian coordinate system for all reconstructed camera centers with origin at C_1. We

Algorithm 4 Camera Referenced Coordinate System to World Coordinate System Transformation

1: **Input:** Camera center coordinates $C_1, ..., C_4$ from quintuplet C and their corresponding GPS tags in 3D spherical coordinates

2: Convert GPS tags $P_{v_1^\star}$ to $P_{v_4^\star}$ for dataset images $I_{v_1^\star}$ to $I_{v_4^\star}$ to 3D Cartesian coordinates P_1 to P_4 by Eq. 5.11

3: Use C_1 to C_4 and P_1 to P_4 as inputs in computing the rotation matrix R, translation vector t_0, and scaler s.

4: Compute the residual error evaluated for the current values of R, t_0, and s. If the error is less than a desired threshold, the localization error is acceptable.

5: **Output:** Matrix R, column vectors t_0 and s, defining the linear transformation from the camera referenced coordinate system to the world coordinate system.

label these as $y_{l,i}$ with $i = 1$ to 4. Locations P_1 to P_4 can be considered as points in the right coordinate system. This is a 3D Cartesian coordinate system representing the GPS tags for the same cameras. We label these as $y_{r,i}$ with $i = 1$ to 4. The transformation we seek, from the left to right coordinate systems, is given by:

$$y_r = sRy_l + t_0 \tag{5.12}$$

where $s \in \mathbb{R}$ is a scale factor, $t_0 \in \mathbb{R}^3$ is the translational offset, and $R \in \mathbb{R}^{3\times3}$ is a rotation matrix applied to 3×1 column vector y_l.

Because of measurement errors, we are unlikely to compute rotation matrix, a translation vector, and a scale factor, such that the transformation equation **??** is satisfied for each point exactly. Alternately there will be a residual error given by:

$$e_i = y_{r,i} - sRy_{l,i} - t_0 \tag{5.13}$$

For general coordinate transformation problem and two sets of k points in left and right, the problem can be formulated as a Least Squares problem. The objective is to find a match matrix or correspondences m which represents the corresponding points in the left and right coordinates and transformation parameters R, s, t_0 which minimize mapping error from one set of points y_l onto another set of points y_r.

$$(t_0^\star, s^\star, R^\star, m^\star) = \underset{t_0,s,R,m}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{j=1}^{n} m_{ij} \|y_{r,i} - sRy_{l,j} - t_0\|^2 \tag{5.14}$$

In our application we know the GPS-tags of all images (for example four images) in dataset which are fed to SFM. Upon convergence of SFM, the camera centers corresponding to those four images but in camera referenced coordinate system can be extracted. Therefore correspondences are known from left to right systems which obviates the need to keep match matrix in Eq. 5.14. So we have:

$$(t_0^\star, s^\star, R^\star) = \underset{t_0,s,R}{\operatorname{argmin}} \sum_{i=1}^{n} \|y_{r,i} - sRy_{l,i} - t_0\|^2 \tag{5.15}$$

In our method we ideally use four (or three if four is infeasible) corresponding points. Each point has three variables. Therefore more than eight equations are available which makes it feasible to find the transformation parameters with a closed form method described in (Horn et al., 1988) in $O(n)$ time. So we directly calculate R, s, t_0 as shown below. The first step according to (Horn et al., 1988) is computing the centriod of y_l and y_r.

$$\bar{y_l} = \frac{1}{n}\sum_{i=1}^{n} y_{l,i} \quad \bar{y_r} = \frac{1}{n}\sum_{i=1}^{n} y_{r,i} \tag{5.16}$$

Then points should be shifted with respect to the centroids:

$$y_{l,i}^{'} = y_{l,i} - \bar{y_l} \quad y_{r,i}^{'} = y_{r,i} - \bar{y_r} \tag{5.17}$$

Now by using $y_{l,i}^{'}$ and $y_{r,i}^{'}$ in the error e_i we have:

$$e_i = y_{r,i}^{'} - sRy_{l,i}^{'} - t_0^{'} \tag{5.18}$$

$$t_0^{'} = t_0 - \bar{y_r} + sR\bar{y_l} \tag{5.19}$$

The square of error in Eq. 5.15 can be minimized when $t_0^{'}$ is equal to zero. This yields

$$t_0 = \bar{y_r} - sR\bar{y_l} \tag{5.20}$$

Now for finding the translation, t_0, s and R should be computed. From (Horn et al., 1988) s can be computed as follows:

$$s = \sqrt{\sum_{i=1}^{n} \|y'_{r,i}\|^2 / \sum_{i=1}^{n} \|y'_{l,i}\|^2} \tag{5.21}$$

Now R can be calculated using the steps below. First compute M :

$$M = \sum_{i=1}^{n} y'_{r,i} (y'_{l,i})^{\intercal} \tag{5.22}$$

which is a 3×3 matrix. Then compute $B = (M^{\intercal} M)$ and find the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and eigenvectors $\hat{v_1}, \hat{v_2}, \hat{v_3}$ and express B using eigen-decomposition as follows:

$$B = \lambda_1 \hat{v_1} \hat{v_1}^{\intercal} + \lambda_2 \hat{v_2} \hat{v_2}^{\intercal} + \lambda_3 \hat{v_3} \hat{v_3}^{\intercal} \tag{5.23}$$

$$R = M\left(\frac{1}{\sqrt{\lambda_1}} \hat{v_1} \hat{v_1}^{\intercal} + \frac{1}{\sqrt{\lambda_2}} \hat{v_2} \hat{v_2}^{\intercal} + \frac{1}{\sqrt{\lambda_3}} \hat{v_3} \hat{v_3}^{\intercal}\right) \tag{5.24}$$

By substituting s and R from Eq. 5.21 and Eq. 5.24 into Eq. 5.20, the translation vector, t_0 can be computed. Now each point from left coordinate including the query point can be transformed to right side by:

$$y_r = sRy_l + t_0 \tag{5.25}$$

The total residual error (E_{Total}) resulting from the transformation is equal to

$$E_{Total} = \sum_{i=1}^{n} \|e_i\|^2 \tag{5.26}$$

We describe the result and the related error range for some samples in Section 6.4. Notice that (Zamir and Shah, 2010) and (Vishal et al., 2015) utilized multiple estimates of the query position derived from multiple running of SFM. Then an optimization approach (Random Walk) is employed to estimate query location. In order to avoid time complexity of multiple SFM, our proposed method runs SFM only once to compute the coordinate transformation parameters as mentioned above. Since four relevant images with distinct GPS-tags may not always be available for all queries, we have also examined the use of only three matching images. In order to adapt Algorithm 4 to three dataset images, two methods have been proposed as described below.

5.5.2 Estimate Query Location Using Three Dataset Images

Four relevant images may not be found in every case. We therefore seek to recover the query GPS-tag using a smaller number of images. Until now we considered the use of four images since three unknown coordinate variables should be determined. In general for computing transformation parameters between m-dimensional vectors using the method described above, $m+1$ corresponding points are required. So if three images are used, only two unknown coordinates such as x and y can be recovered. The advantage of this approach is that it can be applied to more query images since some of them do not have four relevant candidates with distinct GPS-tags. Although finding the transformation between the camera coordinate system and the real-world system in Cartesian coordinate is almost the same for four or three images, transfer from Cartesian coordinates to GPS tag (Lat, Long) is not possible

without having corresponding 3D position vectors. To address that, we propose two different methods described below.

5.5.2.1 Finding Third Component By Averaging z

Since z values would be close for the query and dataset images, we seek to only recover x and y by computing coordinate transformation. We use x and y in left and right coordinates (camera-referenced coordinates and real-world coordinates) for three images to compute the transformation parameters, R, s, t_0. Then the transformation should be applied to the query location in left system to obtain the query location in real-world coordinate (only x and y). By having x and y , only calculating longitude is possible since z is required for calculating latitude. In order to have a reasonable estimate, an average of z, as shown in Eq. 5.27 below

$$Z_q \approx \frac{\sum_{i=1}^{3} Z_{db_i}}{3} \tag{5.27}$$

for those three candidates is computed. This is because we assume that there would not be an abrupt change in the z coordinate values among the selected images and the query. All three components of the location vector of the query, $[x, y, z]$, can be used for computing query's latitude and longitude.

5.5.2.2 Position Vector Reduction

Another method employed is dimension reduction. We applied Principal Component Analysis (PCA) to the 3D position vectors for transfer to 2D space. A coordinate transformation is then applied between those 2D vectors and their associated GPS tags. The query GPS tag can then be calculated directly. To examine how the whole process affects the ultimate accuracy, the same query images as used with the four images approach in Section 5.5 have been used for evaluation. Also, the experimental

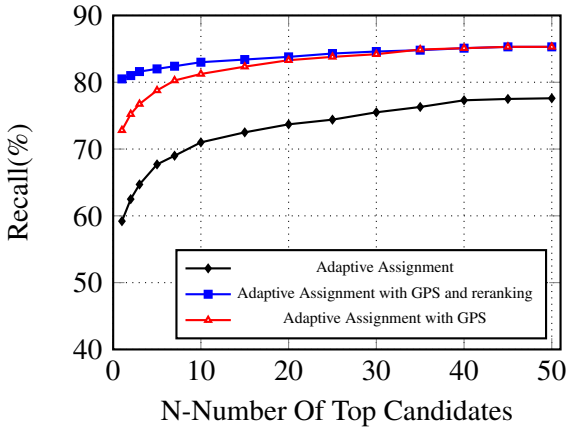


Figure 42: Recall versus the number of top candidates for San Francisco dataset for different scenarios. Limiting the search area using Algorithm 3 improves recall

results for those queries which were not evaluated due to the lack of four relevant images are covered in the Section 6.4.

5.6 Performance Evaluation

In this research, we evaluated the performance of our method using the publicly available San Francisco dataset from (Chen et al., 2011) containing more than one million images.

The reason for using this dataset is that the location error for its queries is high since the images are captured mostly in downtown San Francisco. Also, it contains more images per area which is necessary in our research which is based on atleast three images with distinct GPS-tag in SFM process. We have used only perspective central images, PCIs, from the dataset since they are less likely to cause distortions during the VisualSFM 3D reconstruction. The San Francisco dataset provides a set of 803 query images, usually taken from a pedestrian's perspective at street level. Each query image is also annotated with

a) Best candidates from Retrieval after re-ranking by Geometry verification

b) Query

c) Four best images from GMCP

d) Four best images by finding images with unique GPS-tag

Figure 43: a) Sample set of images returned by retrieval pipeline for query image shown in b. c) Images selected by proposed method based on GMCP. d) Images selected by finding images with highest number of inliers with query and distinct GPS-tag. Although images in two sets (c) and (d) look similar, only the set returned by GMCP let to convergence in the SFM pipeline.

a ground truth GPS tag which is noisy. Since accurate ground truth is required for evaluating the final results, we have used ground truth from (Xu et al., 2012) and compared our result with the results provided in this article. We also used Adaptive Assignment (Torii et al., 2013) while $\eta = 200k$ for the image retrieval engine. To assess the performance, recall as used in (Chen et al., 2011), (Sattler et al., 2012), and described in Eq. 6.9 has been used. To further improve recall rate, rough position and maximum GPS error are used for narrowing down the search space. For the San Francisco dataset $R_{max} = 300$ is reported. By considering R_{max} in Algorithm 3, recall has improved as shown in Fig. 42.

Figure 44: Sample set of images returned by retrieval pipeline in a case where neither GMCP nor distinct GPS-tag led to convergence. Images selected by GMCP and/or distinct (unique) GPS-tag are denoted as G and U, respectively, while images that were not selected are denoted as NS.

This figure also covers the recall curves after re-ranking. Note that we have used San Francisco 2011 ground truth which does not cover all the query images. As a result perfect recall is not attainable, where

$$recall = \frac{\#\ of\ relevant\ retreived\ images}{\#\ of\ retreived\ images} \tag{5.28}$$

We found that relevant images typically have more than 20 inliers. So candidate images with fewer than 20 inliers have been filtered out directly. From 803 original queries, our retrieval pipeline finds candidates which have at least 20 inliers for 453 queries. For 398 queries, more than four images are

TABLE V: QUERY IMAGES AND CORRESPONDING FOUR MATCHES FOR SPARSE 3D CAMERA POSE RECONSTRUCTION

Query ID	PCI_ sp (best-matching) image ID				Query image	Transform: R, t_0, s	$E_{Total} = \sum_{i=1}^{n} \|e_i\|^2 (km^2)$
14	9276	9277	9279	9275	14		
3D cam positions	0.316227 -0.010382 1.740068	1.024494 -0.011895 1.0341255	2.418441 -0.000215 -0.439624	-0.391183 -0.000558 2.459574	1.148572 -0.000714 -1.207395	-0.892 -0.573 -0.137 0.403 -0.881 -0.655 -0.194 0.852 -0.747 t_0= [-2.697 -4.250 3.904] e+03 s = 0.0078	1.7399 e-06
26	4535	4534	4533	19128	26		
3D cam positions	-0.615103 -0.000913 2.005482	-1.189003 -0.001757 2.662914	-1.737672 -0.001110 3.313662	0.3774560 .040469 1.761960	-1.085159 -0.000942 -0.231594	0.602 -0.006 -0.678 0.297 0.056 0.681 0.740 0.044 0.274 t_0= [-2.697 -4.250 3.904] e+03 s = 0.0046	1.8707 e-07
320	13255	13256	13257	4819	320		
3D cam positions	0.003372 -0.005780 0.200270	0.733618 -0.035297 0.6269572	1.396704 -0.045083 -1.526042	1.332279 -0.063366 -1.287443	0.410683 0.349956 -3.633897	-0.27 0.01 -0.86 0.73 0.01 0.09 0.61 0.03 -0.49 t_0= [-2.697 -4.250 3.904] e+03 s = 0.0035	4.1369 e-07

found. Although retrieval curves for $N = 50$ are shown, we have selected 15 images for the GMCP ($N = 15$). The reason is that the recall is almost flat for the $N > 15$. Subset of four images is then selected with two different approaches discussed in Section 5.3.3. For queries for which the number

of retrieved candidates is less than 15, all retrieved images proceed in the next step. Fig. 43 shows a query with multiple candidates returned from the retrieval pipeline and four images opted by two approaches. Although images appear to be similar in both sets, the set returned by GMCP converged in SFM processing while the other did not. Fig. 44 represents a sample which did not converge for both methods while they contain different images. In Fig. 44, G represents images selected by GMCP and U by distinct GPS tag. Images which are not selected are shown by NS.

For the 277 queries from 398, both approaches, returned identical subsets. Among those sets, 141 of them converge and produce 3D coordinates. For the reminding 121 queries we got different subsets with 42 convergences for the method based on finding distinct GPS tag and 61 convergences for the GMCP based approach. It is worth mentioning that GMCP based selection converged for all samples which distinct based method converged. We also found that localization error is low and acceptable for our application when the SFM converges with five images including the query. This is because the amount of error introduced by the approach we have used for coordinate transformation is low. Therefore the total location error is acceptable upon convergence of the SFM. Although some queries could find more than three candidates, the number of candidates with distinct tag is less than four. We have evaluated our method based on three candidates as discussed in 5.5.2 and found success in 47 more cases.

Table. V illustrates several query images from the San Francisco dataset, the corresponding four matching images for 3D camera pose reconstruction, and the residual error $\sum_{i=1}^{n} \|e_i\|^2$ in squre kilometers. The reconstructed best-matching camera center positions from SFM are listed for each query. Note that we rely on VisualSFM (Wu et al., 2011) for convergence i.e. estimate camera locations for cameras including the query camera. When convergence is not achieved with four images, the same approaches

with three images can be applied. For each query, the amount of residual error obtained from closed-form formula for the transformation is listed in the rightmost column. Those values of errors which are introduced by coordinate transformation are acceptable for all quintuplets considered in testing. This error is sum of errors for coordinate transformation of four images from camera coordinates to real world coordinates system through the computed R, s, and t_0 and is acceptable for our application specially when we know the precision of the ground truth is in the range of a meter. The specific parameters of the corresponding transformation, scale factor s, translational offset t_0, and rotation matrix R are listed in the adjacent column. Details are presented in the Section 6.4.

Table. VI depicts an illustrative random subset of query images and the distance error in meters between the estimated GPS tag and the ground truth tag for each query when four dataset images are used. When VisualSFM does not converge using four candidate images, we considered the result for that query to be unsuccessful. It is however possible to consider using three candidates for that query.

The coordinate transformation pipeline was found to converge with acceptable error for all successful cases of convergence in SFM. According to the Table. V, the maximum residual error for transferring four points from left to right coordinate system was about three meters in the worst case.

It is worth mentioning that for all of the samples we got less than this level of error for residual error and it was an order of magnitude times smaller for most of the cases. The resulting estimation error of each query camera's GPS tag is shown in the Fig. 45.

As can be seen, the best result is obtained using the method with four dataset images. Moreover, the plot shows that 59.4% of the query estimated locations have an error of less than 5 meters and 32.6% have an error between 5 and 10 meters. For that scenario and for some samples shown in Table. VI, the

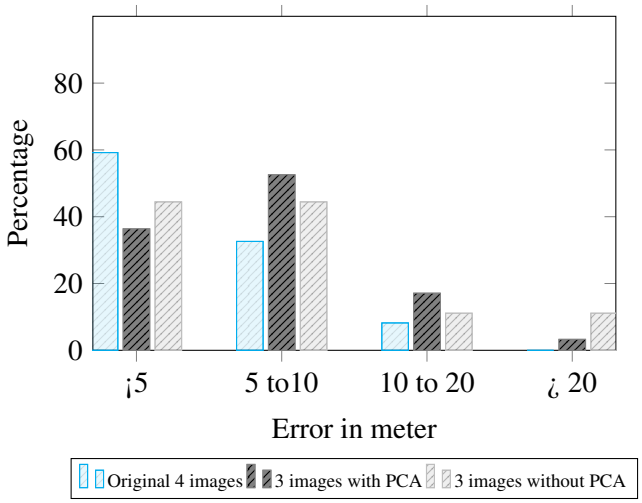


Figure 45: Distribution of estimation error in meters of query cameras GPS tag using our proposed method for the cases of four original images (blue), three images without PCA (dark gray), three images with PCA (light gray). Localization error for about 59% of query images is less than 5m using four images (blue).

ultimate localization error for most of the samples is less than three meters. This level of accuracy is not achievable for other two methods based on three dataset images. In fact for three images, PCA-based method is slightly better while it is inferior in a scenario with four images.

Also, these results represent a marked improvement over the errors reported in (Xu et al., 2012), (Liu et al., 2012). In (Xu et al., 2012) only errors less than 20 meters are reported while no statistics for errors less than five meters or between five and 10 meters is presented. In (Liu et al., 2012) only 15% of the errors are less than five meters compared to 59.4% achieved with our approach. Achieving an error in the range of 20 meters was not our goal since this level of error can be obtained by just considering the location of the best match from retrieval for most of the queries. The average of estimation error of

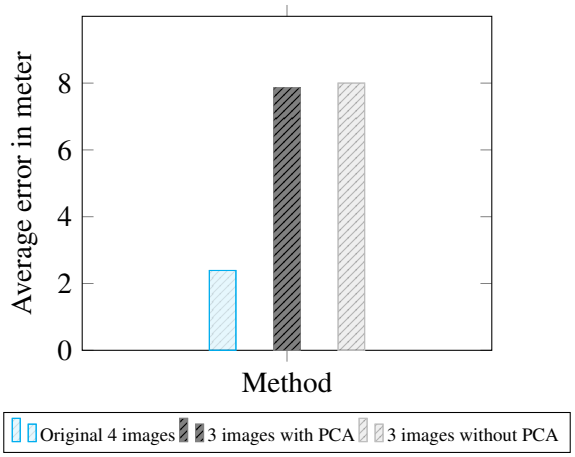


Figure 46: Overall average of estimation error (in meters) of query cameras GPS tag using our proposed method for the cases of 4 original images (blue), 3 images with PCA (dark gray), 3 images without PCA (light gray).

the query cameras GPS tag using our proposed method is shown in Fig. 46 for the cases of four original images (blue), three images with PCA (light gray) and three images without PCA (dark gray).

It is important to note that we do not incur any increase in computational burden in our method with four or three images. This is because the time required for retrieving and re-ranking images for an arbitrary query is almost the same for all approaches. Also, image selection based on GMCP with only $N = 15$ nodes does not require a large amount of computation and adds up less than 10% to the time required for a single SFM. Moreover, the computational cost for coordinate transformation based on the proposed closed-form approach is even less than 1% of required time for a single SFM. So, the total running time is dependent mainly on the number of times SFM is executing. Unlike other mentioned research, we run SFM only once that leads to significantly reduced running time. It is worth mentioning

(a) query (b) match 1 (c) match 2

(d) match 3 (e) match 4

Figure 47: Sample image set 1 of query and 4 best matching images considered in the position estimation shown in Fig. 48.

that considering four images instead of three images for the visualSFM process has a negligible effect on the processing time as discussed in (Wu et al., 2011) but reduces the mean squared reprojection error of the estimated five camera coordinates.

In order to show how our proposed method improved the localization, two samples are provided with more details. The two sample image sets considered here are shown in Fig. 47 and Fig. 49, and the positions of the retrieved images (more than four), are shown, respectively, in Fig. 48 and 50 with red icons. Also Fig. 47 and Fig. 49 show four images that are used in SFM for each query.

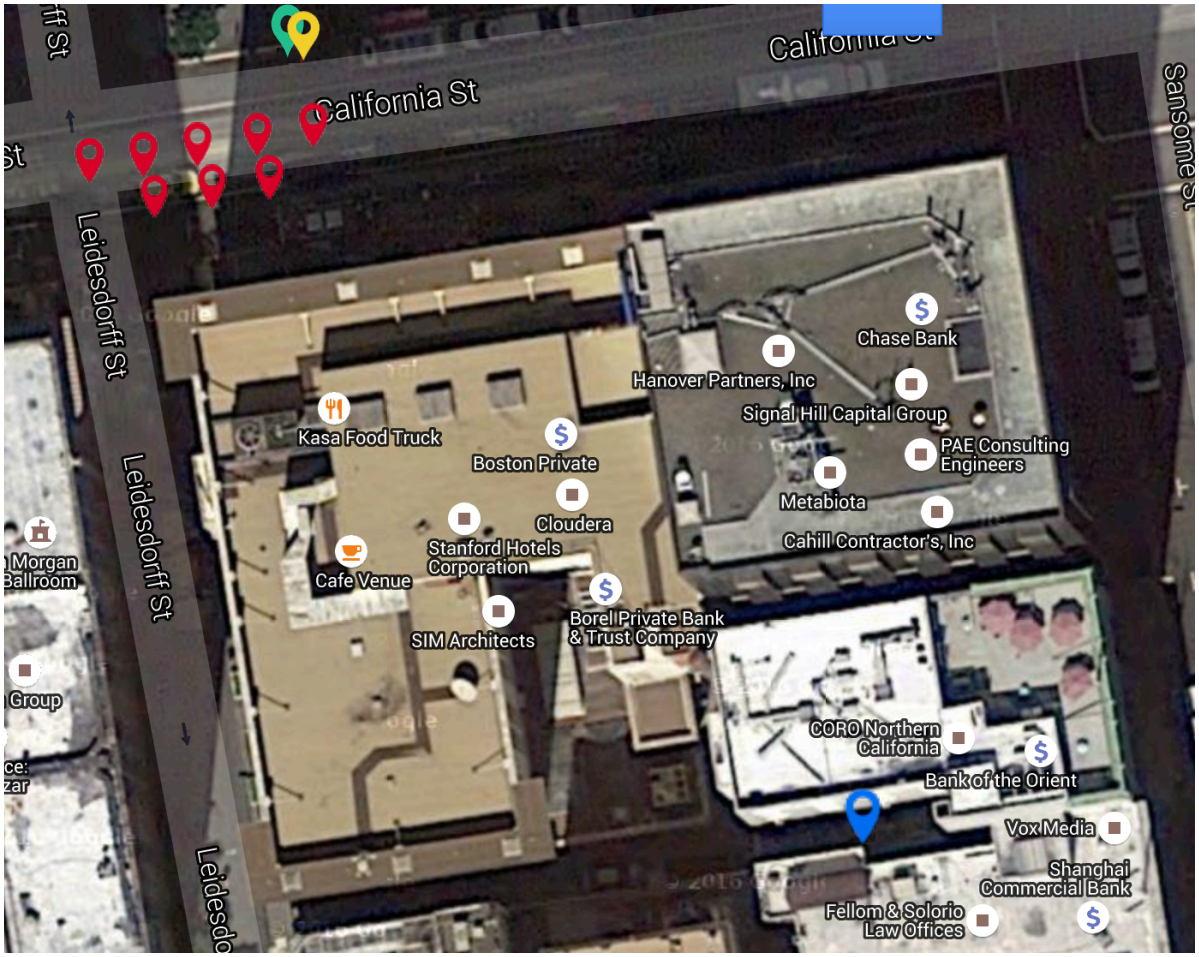


Figure 48: Sample localization result for query image in set 1 in Fig. 47: Noisy query position from GPS (blue), Position of the best matches (red), actual (green) and estimated positions by proposed method (yellow)

To evaluate the performance of our proposed method, the noisy query position is shown in blue while the actual and estimated positions are shown in green and yellow, respectively. As can be seen, the actual and estimated positions are close, especially in Fig. 50 where the distance between the two is less than two meters.

(a) query (b) match 1 (c) match 2

(d) match 3 (e) match 4

Figure 49: Sample image set 2 of query and 4 best matching images considered in the position estimation shown in Fig. 50

Although we have considered prior knowledge of the position along with maximum GPS error for the San Francisco dataset, the localization errors for new cellphones are usually within 100 meters even in the worst case in cities such as San Francisco or Chicago. So the retrieval engine can search in an even smaller region specified by the range of GPS error. By applying our proposed method this level of error would be reduced to a range of a couple of meters.

5.7 Conclusion

In this chapter, we used a state of the art image retrieval algorithm and the GPS tag and maximum error of the GPS to improve retrieval task. Then, we proposed a method to optimally select the best

TABLE VI: GEO DISTANCE ERROR BETWEEN GROUND TRUTH AND ESTIMATED GPS TAGS FROM FIVE IMAGES

Geo Distance Error	Query Number						
	14	26	52	115	189	233	524
meters	1.81	7.37	5.1	1.29	0.97	2.2	1.89

subset of images selected with the highest similarity to be used in reconstructing a 3D scene by using SFM. In order to compute the query location, we introduced a coordinate transformation between dataset images location in camera referenced coordinate system and their corresponding real-world locations. The advantage of this method is that the transformation parameters and consequently query location can be computed directly from the results obtained with only a single execution of SFM. Although four images from retrieval are employed, for most of the samples we showed that transformation parameters can also be computed with three images. Experimental results show that our approach is able to reduce the error in the estimates of query's GPS tag from more than 20 meters (distance between actual query position and best match) to less than five meters in a high percentage of the considered test cases which is suitable for localization application of interest to us. Also we observed that our proposed method will produce an improved performance (SFM convergence for a larger set of query images) if the original database has more images per location and a higher degree of overlap between images from similar locations.

Figure 50: Sample localization result for query image in set 2 in Fig. 49: Noisy query position from GPS (blue), Position of the best matches (red), actual (green) and estimated positions by proposed method (yellow)

CHAPTER 6

SCALE CONSISTENCY MODEL

In this chapter we propose a new approach to improve our image retrieval system. As mentioned before, the core of our system is BOF model. This model proposed by Zisserman and Sivic (Sivic and Zisserman, 2003) is adapted from text retrieval techniques to image retrieval is an effective approach widely used in computer vision.

In this method all feature descriptors are quantized to visual words with a clustering algorithm like K-Means. An image is represented by a histogram of a number of visual words and each image in the dataset has its own histogram. In order to find the best match, different methods can be applied. For example, the histogram of the query image should be compared with all histograms in the dataset. There are different measures for finding similarity such as utilizing query-specific distance functions (Jing et al., 2013), the inner product of two BOF vectors, ℓ_1 distance, and inverted index method (Witten et al., 1999). Other researchers have focused on different aspects of the image retrieval system, such as creating a larger vocabulary (Nister and Stewenius, 2006), training a classifier such as SVM per location (Gronat et al., 2013) to extract more distinctive features by considering their geographical distribution, and considering geometric consistency (Turcot and Lowe, 2009). The problem in most of the methods mentioned is that they are not practical for large-scale datasets especially in our application with millions of images. Besides the approaches mentioned earlier, other researchers have worked on approaches for

more accurate descriptor matching (Jegou et al., 2008), removing confusing features (Knopp et al., 2010), Soft Assignment (SA) instead of Hard Assignment (Philbin et al., 2008), Burstiness removal (Jégou et al., 2009) and Adaptive Soft Assignment by considering repetitive structures (Torii et al., 2013). Although the Recall for the Adaptive Soft Assignment outperforms Recall in other mentioned approaches, it imposes time complexity for creating the BOF vector.

Moreover, some new research (Arandjelović and Zisserman, 2014) has focused on employing distinctiveness of features which is different from common methods based on inverse document frequency (IDF). Such methods use Hamming Embedding (HE) to mitigate quanization error (Jégou et al., 2009), (Arandjelović and Zisserman, 2014). In HE a binary signature for each descriptor should be saved in the dataset. BOF augmented by HE imposes greater memory and computational cost, but it significantly improves performance in terms of Recall.

Methods mentioned earlier are not necessarily the last step in the image retrieval. Most methods select more than one candidate for a match in this step. An additional step called Homography verification performed by applying the popular algorithm of RANSAC (Fischler and Bolles, 1981) is used to choose the best match among the plausible candidates. This step compensates for the weakness of image retrieval schemes based on BOF where the geometric information of images is ignored. Time complexity of this process is high which prohibits its application to large set retrieved images. In order to re-rank a larger set of images, a weighting scheme which considers visual burstiness has been proposed (Jégou et al., 2009). This method takes more images into account than does the regular RANSAC which lead to higher recall with the same running time. However, that post processing approach does not improve performance of retrieval engine based on BOF. In this paper we seek to improve the SA

approach by considering scale consistency between corresponding feature descriptors between query and dataset images. Unlike the regular BOF method which uses frequency of visual words, our proposed method instead uses the scale of feature descriptors in the BOF vector. We refer to this vector as Bag Of Scale-Indexed Features (BOSIF). The creation of the model and retrieval of the best matches is described in the following sections.

6.1 Image Matching Procedure

The core of many image retrieval engines is BOF. In this approach each image is represented by a vector containing frequency of occurrence of features (visual words). Since the importance of the visual words varies, the term frequency-inverse document frequency (TF-IDF) weighting scheme in Eq. 6.2 below is used to assign a weight to each visual word. Term frequency (TF) considers the importance of each visual word (term) in each image while Inverse Document Frequency (IDF) reflects importance of each term in the whole dataset. Suppose we have a vocabulary of ρ visual words. Each dataset image d can be represented by:

$$V_d = [v_1^d, v_2^d, ..., v_\rho^d]^\top \tag{6.1}$$

where the vector entry is defined by

$$v_x^d = \frac{f_{xd}}{f_d} \times \log(\frac{N_{ds}}{N_x}) \tag{6.2}$$

where N_{ds} is the number of images in database, N_x is the number of images containing visual word x, f_{xd} is the frequency of term x in image d and f_d is the number of visual words in image d.

The next step is to find the most similar images based on the score obtained using a criterion such as Cosine distance. Let $V_q = [v_1^q, v_2^q, ...v_\rho^q]$ and $V_d = [v_1^d, v_2^d, ...v_\rho^d]$ denote vectors showing the frequency

of visual words for the query image q and dataset image d, respectively. Each element of the vectors represents the number of times feature descriptors of an image are assigned to this visual word. Images either for HA or SA can be represented with similar vectors. The only difference is that SA considers different weights for different assignments. A measure of similarity between q and d images (vectors) can be expressed as Eq. 6.3.

$$SIM(q,d) = \frac{\sum_{x=1}^{\rho} v_x^q v_x^d}{\sqrt{\sum_{x=1}^{\rho}(v_x^q)^2}\sqrt{\sum_{x=1}^{\rho}(v_x^d)^2}} \tag{6.3}$$

The processing during query time is fast since we do not need to calculate Eq. 6.3. All vectors for the dataset images can be normalized earlier. At query time, we only need to compute the normalized vector for the query. Then inner product between two normalized vectors needs to be computed.
In this research we propose a new algorithm to filter out some of the incorrect assignments caused by SA at query time. Also, our proposed method is combined with a recent algorithm, Adaptive Assignment (Torii et al., 2013) to remove incorrect assignments. In order to proceed, we need to introduce the concept of scale consistency which is done in the next section.

6.2 Scale consistency

Most image-based localization systems employ BOF-based approaches for the retrieval task. Those approaches do not factor the geometric information in the image into the BOF vector. In our work, the goal is to improve retrieval results by changing the content of BOF vector. As evidenced in the SA approach (Philbin et al., 2008), the chance of assigning a feature to a correct visual word increases when more than one cluster (visual word) per feature are considered. Most of methods use three assignments

Figure 51: Removing outliers by evaluating the scale consistency between query and dataset images. The scale ratios for the features assigned to the same visual word are shown above each line. The process consists of counting the number of scale ratios located in each interval. Images are then ranked based on these numbers.

for both the query and dataset images (Torii et al., 2013), while some others employ four or five assignments only at query time to keep memory usage low. Although employing SA improves the retrieval performance in terms of recall, a new source of error is introduced because of more visual words per feature. For example, by considering three assignments for an image with $1k$ features, almost $3k$ visual words in the BOF vector will be non-zero. Although that process increases the chance for correct assignment, it can result in a higher number of incorrect assignments. Since BOF does not consider any spatial information of the image, distinguishing between correct and incorrect visual word matches is not pos-

sible with base BOF scheme. In order to remove incorrect visual word matches, we propose a method which examines the scale ratio between features of the query and dataset images assigned to the same visual word. The rationale for our proposed approach is that for two similar images the scale ratio of most of corresponding feature descriptors of common visual words should be in the same range. Therefore, the ratio of scale of the descriptors can be evaluated to see how many of common visual words are in the same range. Since the range of the ratio values is not readily predictable, suitable intervals for evaluating scale consistency should be considered. For example, for two images taken from almost close locations with the same camera covering similar objects, the scale ratios of common features are around one. So by choosing the interval between .75 and 1.5, we would get consistent corresponding features. In our application, more than one interval should be examined to cover all possibilities. This is because images covering a particular building or scene are taken from different locations and angles of view. Fig. 51 demonstrates the scale consistency evaluation procedure considering only three dataset images. First, the scale ratios for corresponding feature descriptors between the query and dataset images are calculated. Then, we count the number of scale ratios located in each interval for each dataset image and sort images based on those numbers. When the dominant interval for each image is found, other visual words whose scale ratio lies outside that interval should be removed, allowing relevant correspondences be extracted. To proceed further, we need to create the BOSIF vectors discussed in section 6.2.2.

6.2.1 Creating BOSIF vector

Suppose $[f_1 f_2 ... f_m]^\top$ is the feature vector of an image with m features and known scales. After assigning each feature to a visual word, a vector of scale can be created. Let $S_q = [S_1^q, S_2^q, ... S_\rho^q]$ and

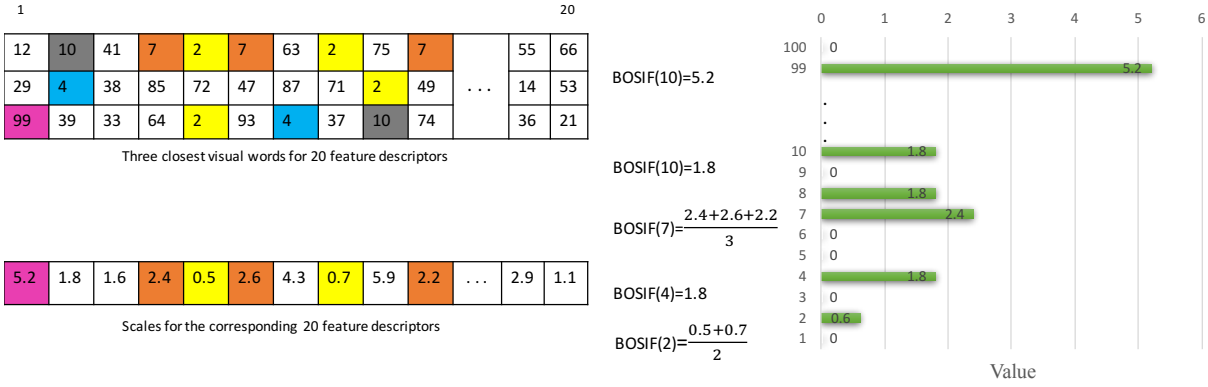


Figure 52: BOSIF vector creation for an image according to Algorithm 5. For simplicity it is assumed that image has only 20 features while the number of visual words is 100. For each feature three closest visual words and scale of their descriptors are provided. As shown the priority for visual words decreases from top (first row) to bottom (third row)

$S_d = [S_1^d, S_2^d, ...S_\rho^d]$ be vectors showing the scale of visual word descriptors for query and dataset images, respectively. In order to create those vectors, we need a procedure described in Algorithm 5 . Suppose we want to create BOSIF for HA. A BOSIF vector contains scale of feature descriptors for words with frequency of one. For words with frequency more than one, the mean of scale of their associated features is used in the BOSIF vector. Based on our observation, the scale of descriptors assigned to the same visual word are close for a high percentage of samples.

The algorithm for creating BOSIF based on SA is more elaborate as described in Algorithm 5.

In Algorithm 5 the depth, k, denotes the number of closest visual words used for creating BOF or BOSIF vectors. One difference with HA is that a particular visual word may receive different scales from different features while there is only one memory location to store the scale of each visual word.

This issue is addressed by initially assigning higher priority to a feature whose closest assignment is that particular visual word, and then continuing to the second, third, . . . and ending with the $k-th$ closest assignment. Only the scale of the feature with the highest priority is then saved. Suppose second closest visual word of feature x and third closest visual word of feature y are visual word M. We only select the scale of feature x for the visual word M in the BOSIF vector. This process is described in Algorithm 5 and visualized in Fig. 52 for clarity . In the next section we demonstrate how scale consistency is taken into account when images are represented by the Algorithm 5.

6.2.2 Finding the best matches by evaluating scale consistency

Algorithm 6 demonstrates how our retrieval system works. Multiple images with the highest scores should be selected for each scale interval. Therefore, we have to search among all images in the dataset multiple times for each scale interval. Considering $V_q = [v_1^q, v_2^q, ...v_\rho^q]$ as a regular BOF vector for the query image obtained from SA and a predefined scale interval $scale = (a_s, b_s)$, a similarity estimate is obtained using eq. 6.4 below where.

$$SIM(q,d)_{scale} = \sum_{x=1}^{\rho} (v_x^q) \times \mu(S_x^q, S_x^d)_{scale} \tag{6.4}$$

Algorithm 5 Create BOSIF vector for an image with m features and related scale of feature descriptors

Input: D: visual words dictionary, I: Given image, k:maximum number of assignment, m number of the feature of I, each feature with s_i :scale and d_i:descriptor

Initialize $BOSIF$ as a zero vector of size ρ

for $i = 1, \ldots, m$ **do**

    Find the k closest visual words for descriptor d_i, $VW_i^k = \{vw_i^j\}, j = 1, \ldots, k$.

end for

let $VW^1 = \{vw_i^1\}, i = 1, \ldots, m$

Create H: histogram of visual words for the set $\{vw_i^1\}, i = 1, \ldots, m$ with ρ bins

for $i = 1, \ldots, m$ **do**

    for $z = 1, ..., k-1$ **do**

        $BOSIF(vw_j^{k-z+1}) = s_j$

    end for

end for

Find set of visual words $n_x,\ x \in X\ s.t.\ H_x > 1$

for $x \in X$ **do**

    $\Delta =$ index of feaure when $(VW^1 == n_x)$

    $BOSIF(\Delta) = Mean(s_\Delta)$

end for

Output: BOSIF vector of I

$$\mu(x,y)_{scale} = \begin{cases} 1 & \text{if } (a_{scale} < x/y < b_{scale}) \\ 0 & \text{otherwise} \end{cases} \tag{6.5}$$

$\mu(x,y)$ is a function that returns one when the ratio of its inputs is located between two given scale boundaries and IDF is computed from the dataset images. Since the frequency of features are not available for the dataset images, we only apply it during query time and for the query image. All of those best images from different scales with their scores should be re-ranked to return the best matches among all scales.

Computational cost for Algorithm 6 is high due to executing a loop for different scales which contains all dataset images. We know that the element-wise scale ratio between BOSIF vectors of the query and dataset image j, S_d^j/S_q, $j = 1,2,...N_{ds}$, should be computed for only the nonzero elements of S_q. The resulting vectors are sorted based on the number of non-zero elements. Those values indicate the visual words that are common to the query and dataset images which is a good indicator for image similarity. To reduce complexity, we found that the second loop in Algorithm 6 operation 3 should be run only on a subset of dataset images with the higher number of visual words common with the query. Other images do not have enough common visual words and would not be suitable matches. Section 6.4 discusses the percentage of images in the subset and its influence on the performance of our proposed approach.

6.3 BOSIF in Adaptive Assignment

In order to improve our results in terms of recall, we employ Adaptive Assignment(AA) which has been shown to outperform other methods in most recent research (Torii et al., 2013). In this method, instead of a fixed number of assignments, a variable number of assignments for each feature is taken into

Algorithm 6 Image Retrieval Based On Scale Consistency Algorithm

Input *Model* : Containing all the vectors for dataset images generated in Algorithm 5

Z_1: Number of scale intervals

N_{ds}:Number of images in the dataset

q: query image,

N: Number of images for recall curve

(V^q): BOF vector of query, (S^q): BOSIF vector of query

for $scale = 1, ..., Z_1$ **do**

  for $vector \in Model$ **do**

    Find $\psi(q, vector) = SIM(q, vector)_{scale}$

  end for

  Sort ψ

  $Res(scale, 1:N) = \psi$ for N highest scores

  $ID(scale, 1:N)$ =Image ID for N highest scores

end for

Reshap *Res* and *ID* to a vector

Sort *Res* and arrange *ID* according to it

Find unique image *ID* in the list

Output: Select N first *ID* as the set of best matches for the query image q

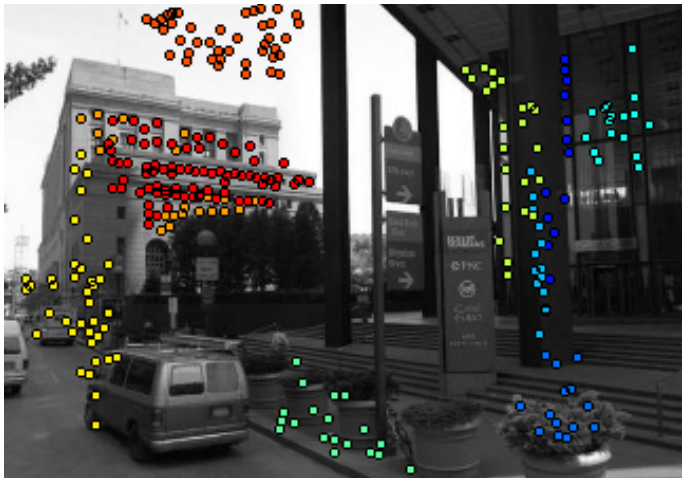


Figure 53: Result of repttiles detection. Different groups (repttiles) are shown with different colors. For creating the BOF vectors for all features shown in orange and red three and two closest visual word are considered, respectively.

account. The number of assignments is obtained after grouping features into multiple clusters. We first discuss Adaptive Assignment (AA) briefly, and then we demonstrate how this method can be combined with our scale consistency algorithm to improve the recall rate.

6.3.1 Adaptive Assignment

One of the main challenges in creating BOF vectors is the quantization and assignment of feature descriptors to the visual words. This process often introduces quantization error especially when SA is applied. In order to address this problem, Torri et al. (Torii et al., 2013) proposed a method to adjust feature weights in the BOF model. The goal in their solution was to categorize features in group

of repetitive patterns and assign different number of assignments according to the group each feature belongs to. The core of this method is the creation of an undirected feature graph $G = (V,E)$ with N vertices that represent all features of an image. Two arbitrary vertices V_i and V_j are connected by an edge e_{ij} when corresponding features satisfy three conditions. First, the scales of their descriptors are in a similar range. Second, they are spatially close in the image. Third, they share at least one common visual word in their top visual word assignments usually 50 or sometimes only 20. The next step is to segment vertices (features) by finding connected vertices that are called repttiles. Figure 53 shows a sample result for detecting repttiles (Torii et al., 2013). Only structures containing higher number of features are shown with different colors. The next step after detecting a repetitive structure is to apply a quantization procedure that adaptively assigns a different number of visual words for a particular feature descriptor based on membership in each repttile. The idea behind this method is that a particular feature belonging to bigger repttile needs to be assigned to a fewer number of visual words since they will receive naturally *soft-assignment* from members of their repttile. Let r_x be the weight of xth visual word in image d that can be computed by adding all weights from features assigned to the visual word in images while closeness to visual word is taken into account. The BOF vector by AA method can be computed as

$$\mathbf{z}_d = (z_1, z_2, ..., z_\rho)^\top \tag{6.6}$$

where

$$z_x = \begin{cases} r_x & \text{if } (0 \leq r_x < T) \\ T & \text{if } T \leq r_x \end{cases} \tag{6.7}$$

The threshold T down-weights repeating visual words. Suppose each feature f_i in the given image is assigned to β_i nearest visual words and w_i^k, $1 \leq k \leq \beta_i$, is the index for kth nearest visual word of feature f_i. According to (Torii et al., 2013), β_i for each feature can be computed with a formula which considers maximum number of assignments, β_{max}, and number of features in associated repttile. Then the weight r_x is defined by

$$r_x = \sum_{i=1}^{N} \sum_{k=1}^{\beta_i} 1[w_i^k = x] \frac{1}{2^{x-1}} \tag{6.8}$$

where the function $1[w_i^k = x]$ is equal to one when visual word x is available in kth nearest visual word of feature i. It means the weight r_x is computed by adding contribution of assignments from all features while order of visual word, k, is taken into account. It is worth mentioning that other types of weighting such as the method employed in original SA algorithm can be used here. After computing $\mathbf{z}_d$ for all d in the dataset, equation 6.3 can be used to find the best matches for a particular query. In section 6.3.2 we discuss the proposed scale consistency in Adaptive Assignment algorithm.

6.3.2 Scale consistency in Adaptive Assignment algorithm

In order to improve performance of scale consistency, we propose a hybrid method which combines the idea of scale consistency and Adaptive Assignment and is represented in Algorithm 7. This algorithm only takes into account β_j as is considered in the Adaptive Assignment algorithm. The number of

assignments should therefore be determined in the first step by detecting repttiles. This approach needs to be applied both at query time and in creating the model for the dataset images. The next step for finding the images closest to a query is the same as Algorithm 6 as shown in Algorithm 7. Since $\beta_j \leq \beta_{max}$, the total memory usage for the hybrid algorithm is less than that for the pure BOSIF approach.

Algorithm 7 Create BOSIF vector for an image with Adaptive Assignment

7: steps 1 to 7 in Algorithm 5

8: **for** $i = 1, \ldots, m$ **do**

9: Find number of Assignment, β_i, for d_i based on repttile detection

10: **for** $z = 1, \ldots, \beta_i$ **do**

11: $BOSIF(vw_i^{\beta_{max}-z+1}) = s_i$

12: **end for**

13: **end for**

14: Find set of visual words $n_x,\ x \in X\ s.t.\ H_x > 1$

15: **for** $x \in X$ **do**

16: $\Delta = Find(VW^1 == n_x)$

17: $BOSIF(\Delta) = Mean(s_\Delta)$

18: **end for**

19: **Output:** BOSIF vector of I for hybrid method based on our scale consistency and Adaptive Assignment

6.4 Performance Evaluation

To evaluate our proposed method, two available datasets, San Francisco (Chen et al., 2011) and Pittsburgh (Torii et al., 2013) are used. Each of those datasets contains hundred thousands of images and the corresponding tags show their locations. For creating the vocabulary tree, all features of 20% of the images in each dataset are used. The datasets also provide query images used for evaluating the performance.

6.4.1 City-scale San Francisco dataset

This dataset contains 1.06M images which have been created from splitting 150k panorama images from street level of San Francisco. Each image has its own 'carto id' which shows the building covered in the image. The San Francisco dataset also provides a set of 803 query images, usually taken from a pedestrian's perspective at street level. For each query image, usually there are some relevant dataset images with ground truth indicated by carto id. Each query image is also annotated with a ground truth GPS tag which is noisy. Since our goal is to evaluate the image retrieval engine, we do not need the GPS tags of images.

6.4.2 Pittsburgh dataset

This dataset contains 254k perspective images extracted from 10.6k panorama images. Unlike the San Francisco dataset, panorama images have been downloaded directly from Google Street View. So this dataset is sparser and has fewer images in a fixed area. The number of query images which are from another GSV dataset is 24k which makes the retrieval slower than in the case of San Francisco dataset. All images in both query and dataset have accurate GPS tags to be used for evaluation. Retrieval process for a query is considered successful when the retrieved dataset image is within 25m of the query location.

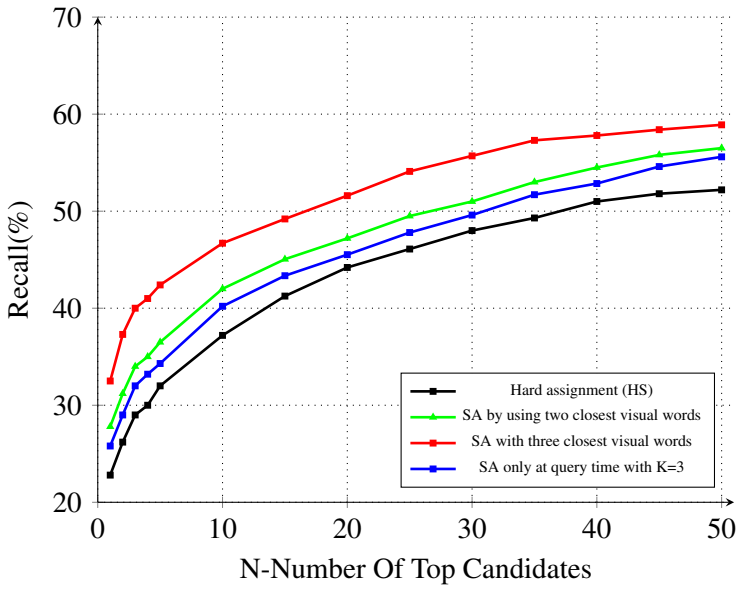


Figure 54: Performance of HA and SA methods with two and three closest visual words along with SA just at query time on the San Francisco dataset. Plot represents the recall vs. the number of top N retrieved dataset images.

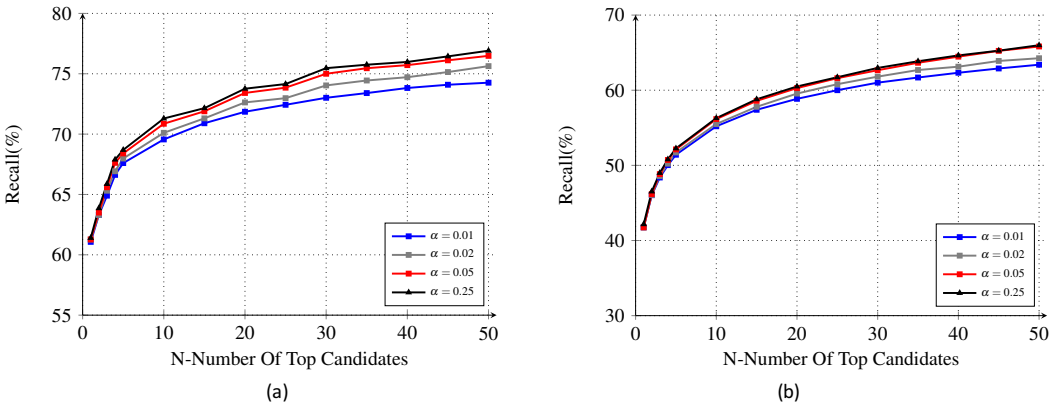


Figure 55: Recall vs number of best candidates for different value of α which indicates percentage of dataset images involved in scale verification for a): San Francisco and b): Pittsburgh datasets.

In order to find the correct visual word matches, scale consistency should be examined in a limited number of scale intervals. To select the proper intervals, we carried out some investigative experiments on the datasets. We found that the scale ratio of feature descriptors between two images captured from close distances is less than two for the majority of features (assuming the larger of the scales of corresponding image features is in the numerator). Another point that should be taken into account is that queries are taken by different cameras from the dataset images and their features would have different scales. Being aware that all incorrect correspondence assignments could not be detected, we tried to remove the wrong assignments to the extent possible. From the experiments on different pairs of sample images, we found that we can achieve this goal of pruning when the end of the interval is about two times greater than the starting value. We also found that by selecting narrower intervals, some correct assignments would be lost. We therefore created intervals $(a_1,b_1),(a_2,b_2),\ldots(a_m,b_m)$,

where $b_r = 2a_r$ and $a_{(r+1)} = 0.5(a_r + b_r)$ starting from $a_1 = 0.3$ with number of intervals, $m = 5$. For both datasets in this research the same setting is used. Although we could consider more intervals with some cost in additional execution time, the result does not improve significantly since the features scale ratio between the query and its matches is rarely outside the intervals considered. Later, we show the number of intervals can be reduced to only three without degrading the result in terms of recall.

[scale=1]

In order to determine the best number of assignments, different values of k for creating the model and at query time are examined. Figure 54 represents the result for the San Francisco dataset. As shown, SA with three visual words returns the best results when it is applied to all images in the dataset and to the query as well. Results for the same setting applied only to the query shows inferior performance in terms of recall. We therefore considered three visual words and sought to remove some of the wrong assignments based on scale consistency. Regarding visual words with higher number of occurrences, a method such as burstiness (Jégou et al., 2009) is employed. This procedure was only applied at query time and the resulting vector is shown by V^q in Algorithm 6.1. Creating the model is fast with execution time close to that for the regular SA model. At query time, the query should be compared with all images at different scales. But as mentioned before, the scale loop need not be applied on all images in the dataset. It is enough to just examine a fraction of the dataset images with the highest number of common visual words with the query images. So the scale vector of the query image should be divided element-wise by all vectors of the dataset in the model for the nonzero elements of the query's scale vector. Then vectors are sorted based on the number of nonzero elements. Those numbers provide a measure for selecting candidates. The next step is to evaluate those candidates with the scale loop in the

Algorithm 6. In this algorithm α represents the percentage of the dataset images which are employed in the loop of scale. For evaluating the performance of the system, Recall as described in (6.9) is used.

$$Recall = \frac{\#\ of\ relevant\ retreived\ images}{\#\ of\ retreived\ images} \tag{6.9}$$

Figure 55 shows the performance of the proposed method for different values of α for Pittsburgh and San Francisco datasets. The result is smoother for the Pittsburgh dataset since more query images are available for this dataset. The recall in both cases for lower values of N, for example $N < 7$, is almost the same for different values of α. Moreover, the curves for the $\alpha = 0.05$ and 0.25 are almost identical. So if we want to achieve a higher recall, $\alpha = 0.05$ would be enough. This implies for the San Francisco and Pittsburgh datasets with almost 1.06 million and 254k images, only 53k and 12.7k images respectively should pass the loop in Algorithm 6. For a smaller value of N, a lower value for α can be selected to reduce the total execution time.

To have better assessment about the performance of different algorithms, we compare our scale consistency method to some recent popular approaches and show that our approach outperforms most of them in terms of recall. It is worth mentioning that we utilized exactly the code used in Adaptive Assignment (Torii et al., 2013) provided by the authors to generate the BOF vectors and compare our proposed method.

Figure 57 presents the recall for both datasets when $\alpha = 0.05$. According to these plots, the recall for our proposed method is much better than SA for both datasets. For example, the recall improvement on San Francisco dataset for BOSIF and for $N = 1$ is almost 29 %. For other values of N, the difference is more than 15%. The improvement for the Pittsburgh dataset is similar. The result shows our proposed

Figure 56: Sample retrieved images for three given queries. The scale ratio interval number is shown below each image. Most of those are from interval number three

method works better for smaller N which is desirable since we only need a couple of relevant images to find the place or location of the query image. Although recall increases with lower slope for larger values of N in our approach, the performance is better than SA for all values of N. Figure 56 shows multiple retrieved images and their scale intervals number for three query images. Most of the retrieved images have been selected in scale interval number three which is defined between 0.675 and 1.35 and indicates the scale ratio between features of the query and best matches are around one. It confirms our earlier claim that additional scale intervals are not required in our application.

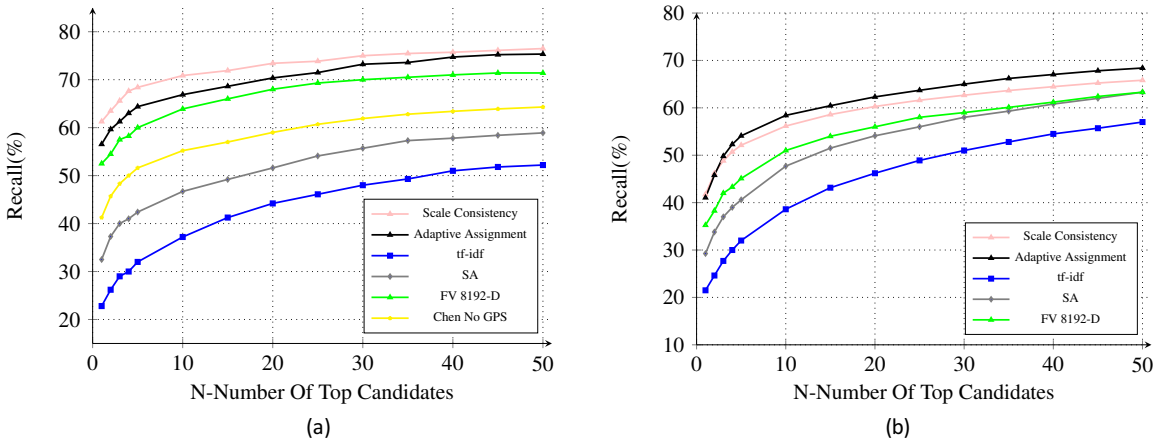


Figure 57: Performance of different methods on the a): San Francisco and b): Pittsburgh datasets. For the proposed method $\alpha = .05$ is considered.

6.5 Scale consistency with Adaptive Assignment

Evaluating similarity for the five level of scales imposes time complexity. In order to mitigate this issue, we normalize the scale of feature descriptors of an image by the median of scales such that:

$$\hat{s_i} = \frac{s_i}{median\{s_1, \ldots, s_N\}} \tag{6.10}$$

Different number of scale levels are then considered to evaluate the result. Figure 58 shows the result for one, three and four levels on the San Francisco dataset. So after normalizing the scale by the median, three level of scale intervals would be enough for our Scale Consistency with Adaptive Assignment (SCAA) method.

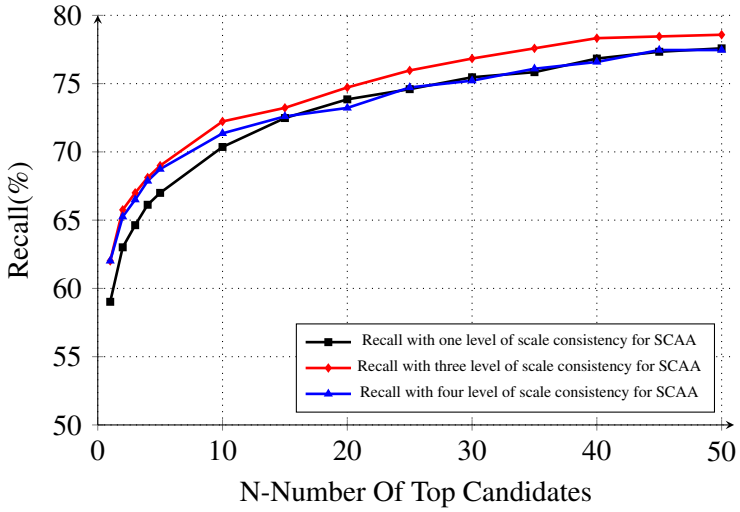


Figure 58: Recall when different number of levels are considered in verifying the scale consistency of features after scale normalization by median for the San Francisco dataset

Figure 59 represents the recall for different algorithms for both datasets. The proposed hybrid method performs the best in terms of recall. Even for the Pittsburgh dataset, which shows lower recall for the scale consistency vs Adaptive Assignment, the SCAA outperforms all methods. Table VII and VIII present more information including memory usage and Recall@N for both datasets. For example, SCAA achieved the highest Recall@1 rate for both datasets without increasing the memory usage. The difference is more than five in both datasets. Figures 60 and Figure 61 depict the recall vs memory for these two datasets. All results confirm that our scale consistency idea can be utilized for removing wrong assignments and can be integrated with other methods to improve the result. Another step that needs to be applied at the end of this level is homography verification. The well-known RANSAC

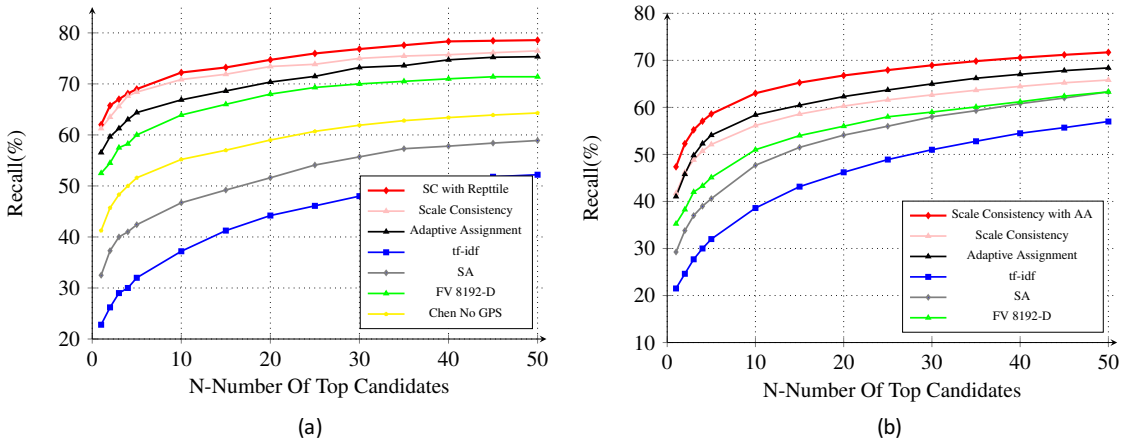


Figure 59: Performance of different methods on the a): San Francisco and b): Pittsburgh datasets. For the proposed method $\alpha = .05$ is considered

algorithm can be used to re-rank images based on their spatial information. Since the computational cost for re-ranking is high, this steps should be applied to only a limited number of images, such as 50 or 100.

6.6 Conclusion

In this chapter we proposed a new method to use scale of feature descriptors in the BOF vector without increasing the memory usage. Instead of inserting the frequency of the visual word in the vector, a method for utilizing the scale of the feature descriptors has been introduced. In order to improve the result, Adaptive Assignment algorithm has been integrated to our work to remove incorrect assignments. At query time, ratio of BOSIF vectors of the query image and dataset images examined to find correct assignments. In each interval of scale, some of the corresponding visual words have

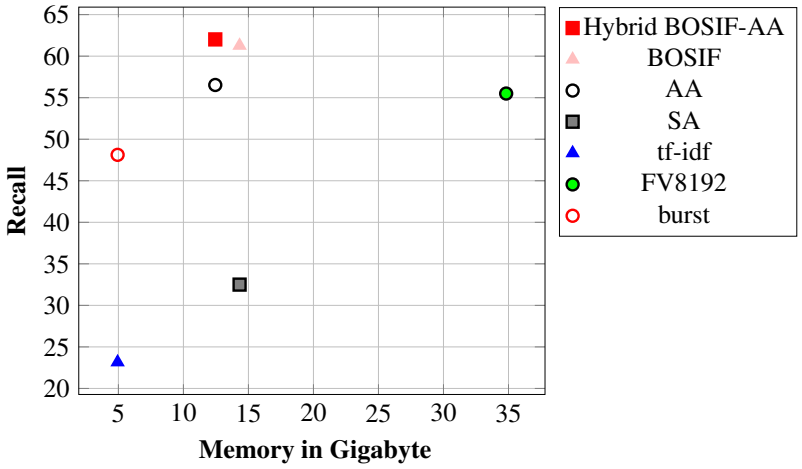


Figure 60: San Francisco dataset

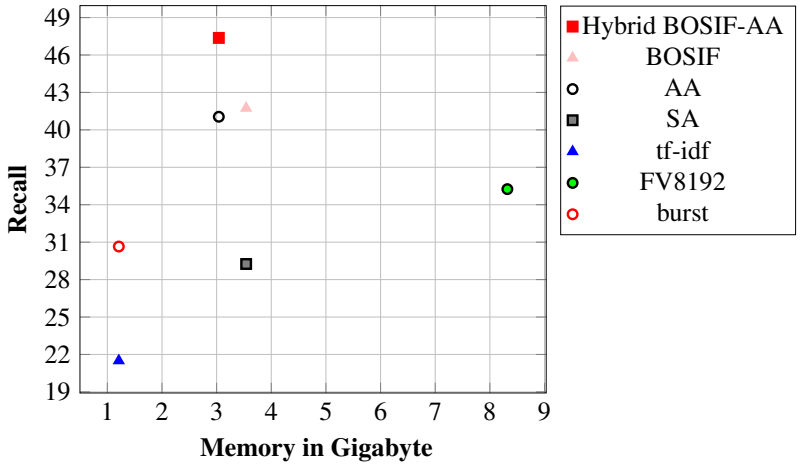


Figure 61: Pittsburgh dataset

Method	Memory (GB)	Recall@1	Recall@2	Recall@5	Recall@10	Recall@50
Hybrid BOSIF-AA	12.45	62.017	65.75	68.991	72.29	78.58
BOSIF	14.32	61.270	65	68.2	70.859	76.53
AA thr-idf	12.45	56.53	59.65	64.38	66.87	75.34
FV 8092	34.82	55.5	57.5	63	66.9	74.06
brst-idf	4.94	48.12	52.51	55.21	62.35	72.41
tf-idf	4.94	23.16	27.12	32	37.2	52.72
SA	14.32	32.5	37.3	42.4	46.7	58.9

TABLE VII: RECALL RATE AND MEMORY USAGE FOR DIFFERENT METHODS ON THE SAN FRANCISCO DATASET

been removed due to scale inconsistency. So the system finds set of candidates per scale interval which would contain similar images. The result after finding unique images shows that our proposed method improves the performance of the retrieval engine in terms of Recall. The recall is higher than SA and even Adaptive assignment especially for lower value of N in the Recall curve. Also, the memory usage in our approach is slightly less than the SA. In future work we are going to find a solution to use visual word correspondences after passing the scale consistency procedure in geometry verification to make it faster and more reliable.

Method	Memory (GB)	Recall@1	Recall@2	Recall@5	Recall@10	Recall@50
Hybrid BOSIF-AA	3.04	47.370	52.283	58.60	63	71.71
BOSIF	3.54	41.74	46.170	52.125	56.158	65.88
AA thr-idf	3.04	41.05	45.8	54.1	58.4	68.4
FV 8092	8.32	35.25	38.28	45.1	51	63.1
brst-idf	1.21	30.65	35.3	42	49	64.4
tf-idf	1.21	21.5	24.62	32	38.6	57
SA	3.54	29.25	33.8	40.6	47.7	63.2

TABLE VIII: RECALL RATE AND MEMORY USAGE FOR DIFFERENT METHODS ON THE PITTSBURGH DATASET

CHAPTER 7

CONCLUSION AND FUTURE WORK

In our research new methods were proposed to achieve improved performance in terms of recall in image retrieval. We considered the use of prior knowledge of query location and the hitherto unutilized sensor information about GPS uncertainty in refining our search. Specifically, Estimation of Position Error (EPE) was found to be useful to dynamically narrow down the search space. Other researchers used a fixed radius for the search space such as 300 meters in San-Francisco dataset. Experimental results on our Chicago dataset have shown that an EPE-assisted selection of dynamic search space outperforms other methods which are based on fixed search space. We also propose a new approach based on Structure From Motion (SFM) that shows better performance in terms of accuracy. Since we know that even successful retrieval only returns the location of the vehicle from which the image was captured and not the actual position of the query, we tried to estimate the query position relative to selected retrieved images. The approach uses SFM on multiple images with similar content selected by our proposed method which enhances the process for finding 2D-2D feature correspondences and estimating multiple camera (images) locations in the camera coordinate system. The camera locations are used to find a 3D coordinate transformation method directly from the camera coordinate system to the world coordinate system (GPS tag). We conjecture that Enhanced Soft-Assignment or using other types of information such as scale of feature descriptors improves regular Bag of Features (BOF) based on SA. Similar methods have been proposed for considering geometry relations between feature positions. Those methods impose larger memory and time requirements and usually difficult to apply to a large scale image re-

trieval problem. In our research we enhance Soft Assignment through considering scale consistency of features. The indexing process for a particular feature descriptor usually incorporates more than one cluster (visual word) to boost the chance of selecting correct visual word. This process usually gives more votes to irrelevant images at query time. To address this, we exploit the scale information of the feature descriptors in a way that avoid increasing the memory usage. The goal is to verify scale consistency between corresponding features of the query and dataset images. Experimental results show that our proposed method is highly effective in improving the recall rate.

As part of ongoing and future work, we explored the idea of utilizing Convolutional Neural Network (CNN) or Deep learning for feature extraction and image retrieval. Extensive research has been focused to extract as mush information content of an image as possible by deep learning techniques. Unlike local feature like SIFT, features extracted from deep learning model are global features and rely on training a model We selected a base model trained on other task such as object recognition and created a Siamese model. The new model can be trained on our dataset for the landmark recognition. The main reason for employing CNN is that this method shows higher accuracy or recall in most of recent computer vision tasks such as object recognition. Also, the size of the feature vector of an image can be reduced to about 4k that is suitable for mobile device.

In almost all similar systems, retrieval task should be done on a server or cloud with enough memory. So another area that is worth to explore is to verify what is the best solution for deploying such a client-server system.

Appendices

Appendix A

AVAILABLE DATABASE FOR IMAGE RETRIEVAL

There are multiple available datasets used by image retrieval and landmark recognition communities such as San-Francisco, Pittsburgh, and Oxford datasets. We used Oxford dataset to compare our image retrieval engine with other related works. We have built our own dataset from the downtown area of Chicago to have all necessary data such as EPE and noisy GPS position for evaluating proposed landmark recognition system. For the method we developed using SFM, more images per location are required than those available. So, the dataset created from Google Street View such as Pittsburgh and our dataset with at least 12 meters distance between two panorama images would not be suitable. The San Francisco dataset which contains images for every 4 meters was therefore employed for this task. further details about each dataset are provided in following sections.

A.1 Oxford 5K dataset

The entire dataset contains 5062 images with resolution 1024×768.

This dataset covers 11 different Oxford buildings collected from Flicker with different distractors. For each building 5 different query images are available giving a total of 55 query images. Images are labeled by 4 possible conditions such as Good, OK, Junk and Absent based on object visibility, absent and occlusion level. The result should be assessed by the average of the performance for each group of query images. Other datasets such as 100K and 1M dataset are also available for Oxford.

Appendix A (Continued)

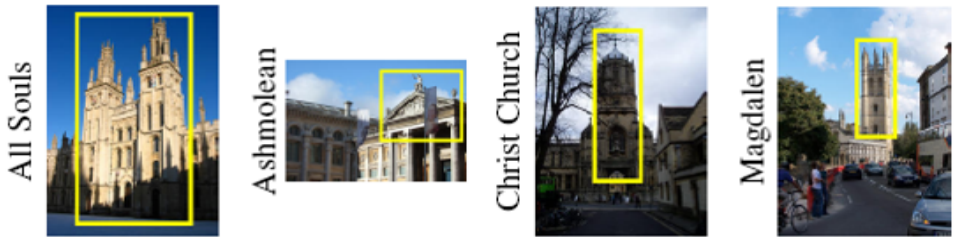


Figure 62: Sample Oxford dataset image and query as a regioon of image shown by yellow box

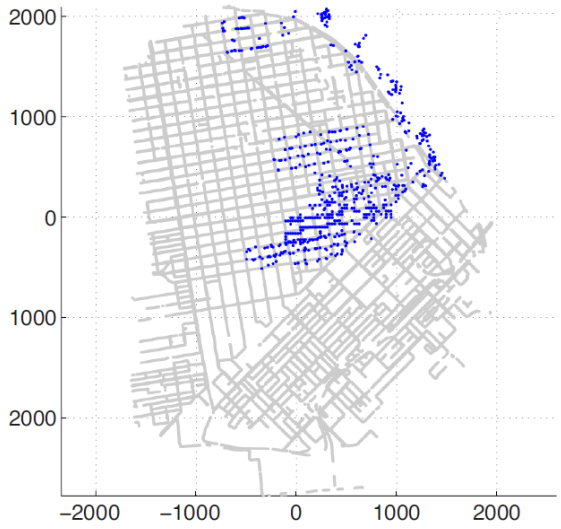


Figure 63: Area covered by San Francisco dataset

A.2 San Francisco dataset

This dataset is formed by 1062468 perspective central image (PCI) and 638090 perspective frontal images (PFI). The area covered by the dataset is shown in Fig. 63 The query set has 803 images taken

Appendix A (Continued)

with different cell phones. This dataset is suitable for use in our research since the query images are taken by pedestrian from street sidewalk. All information is embedded as a name of the image that contains GPS position, building label in the picture (ID), and so on. The performance can be evaluated by checking the ID of the query and the best match.

A.3 Pittsburg Dataset

This dataset is created by splitting 10586 Google Street View panoramas of Pittsburg, PA. Each panorama image is split into 24 images resulting in 254064 perspective images. For testing purposes, 1k panorama images are used to make 24000 query images. Similar to San Francisco dataset the query is captured with different devices and different viewpoints, seasons, and illuminations. The GPS position of the query images is known and can be used as ground truth.

Appendix B

VISUAL SFM

This appendix presents some information about a tool called VisualSFM employed in our work for creating a 3D model and in particular for coordinate system registration.

VisualSFM is a tool created by Chang Chang Wu for the creation of 3D point clouds and meshes. The output of this application can be read directly from CAD packages. This open source software contains other modules from Dr Wu's work such as SiftGPU, Multicore, Bundle adjustment, and Linear time incremental structure from motion. Unlike other tools VisualSFM makes it easy to keep all the images on a PC and not a server while is able to work with dozens of images to reconstruct a very dense set of point clouds. It runs fast by exploiting multicore parallelism that accelerates feature detection, matching, and bundle adjustment. This project started when its inventor was working on 3D modeling of urban areas in the University of North Carolina at Chapel Hill and improved later at the University of Washington where he was a postdoc in GRAIL lab. The GUI of this application can show the result of hundred of images and is the reason for selecting its name, VisualSFM.

Appendix C

COPYRIGHT PERMISSIONS

This appendix presents the copyright permissions for the articles, whose contents were utilized in our thesis. The list of the articles include several conference and symposium publications: ISM'17 (Salarian et al., 2017), ICASSP'18, ISM'16 (Salarian and Ansari, 2016), (Salarian et al., 2016), IntelliSys'15 (Salarian et al., 2015) and a paper in IEEE Transactions on Multimedia (Salarian et al., 2018). The copyright permissions for reusing the published materials are presented in Appendix C.

Title: Image Based Localization Based on Feature Scale Consistency in BOF Vector
Conference Proceedings: 2017 IEEE International Symposium on Multimedia (ISM)
Author: Mahdi Salarian
Publisher: IEEE
Date: Dec. 2017

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

CITED LITERATURE

[Ankita et al. , 2008] Ankita, K., Philippe, T. J., and Roy, A.Anati andRoy, D.: on visual loop closing using vocabulary tree. In Computer Vision and Pattern Recognition Workshops, CVPRW'08, pages 1–8, 2008. organization=IEEE.

[Arandjelović and Zisserman, 2014] Arandjelović, R. and Zisserman, A.: Dislocation: Scalable descriptor distinctiveness for location recognition. In Asian Conference on Computer Vision, pages 188–204. Springer, 2014.

[Bay et al. , 2008a] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L.: Speeded-up robust features (surf). Computer vision and image understanding, 110(3):346–359, 2008.

[Bay et al. , 2008b] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L.: Speeded-up robust features (surf). Computer vision and image understanding, 110(3):346–359, 2008.

[Bay et al. , 2006] Bay, H., Tuytelaars, T., and Van Gool, L.: Surf: Speeded up robust features. In European conference on computer vision, pages 404–417. Springer, 2006.

[C. Yan, 2018] C. Yan, H. Xie, D. Y. J. Y. Y. Z. Q. D.: Supervised hash coding with deep neural network for environment perception of intelligent vehicles. IEEE Transactions on Intelligent Transportation Systems, 19:284–295, January 2018.

[Calonder et al. , 2010] Calonder, M., Lepetit, V., Strecha, C., and Fua, P.: Brief: Binary robust independent elementary features. In European conference on computer vision, pages 778–792. Springer, 2010.

[Chandrasekhar et al. , 2012] Chandrasekhar, V., Takacs, G., Chen, D. M., Tsai, S. S., Reznik, Y., Grzeszczuk, R., and Girod, B.: Compressed histogram of gradients: A low-bitrate descriptor. International journal of computer vision, 96(3):384–399, 2012.

[Charikar, 2002] Charikar, M. S.: Similarity estimation techniques from rounding algorithms. In Proceedings of the thiry-fourth annual ACM symposium on Theory of computing, pages 380–388. ACM, 2002.

[Chen et al. , 2011] Chen, D., Baatz, G., Koser, K., Tsai, S., Vedantham, R., Pylvanainen, T., Roimela, K., Chen, X., Bach, J., and Pollefeys, M.: City-scale landmark identification on mobile devices. CVPR IEEE, 2011.

[Chen et al. , 2013] Chen, Y., Dick, A., Li, X., and Van Den Hengel, A.: Spatially aware feature selection and weighting for object retrieval. Image and Vision Computing, 31(12):935–948, 2013.

[Chum et al. , 2011] Chum, O., Mikulik, A., Perdoch, M., and Matas, J.: Total recall ii: Query expansion revisited. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 889–896. IEEE, 2011.

[Cipolla et al. , 1999] Cipolla, R., Drummond, T., and Robertson, D. P.: Camera calibration from vanishing points in image of architectural scenes. In BMVC, volume 99, pages 382–391, 1999.

[Fei-Fei and Perona, 2005] Fei-Fei, L. and Perona, P.: A bayesian hierarchical model for learning natural scene categories. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 2, pages 524–531. IEEE, 2005.

[Fischetti et al. , 1994] Fischetti, M., Hamacher, H. W., Jørnsten, K., and Maffioli, F.: Weighted k-cardinality trees: Complexity and polyhedral structure. Networks, 24(1):11–21, 1994.

[Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM, 24(6):381–395, June 1981.

[Gronat et al. , 2013] Gronat, P., Obozinski, G., Sivic, J., and Pajdla, T.: Learning and calibrating per-location classifiers for visual place recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 907–914, 2013.

[Guan et al. , 2013a] Guan, T., He, Y., Gao, J., Yang, J., and Yu, J.: On-device mobile visual location recognition by integrating vision and inertial sensors. Trans. Multi., 15(7):1688–1699, November 2013.

[Guan et al. , 2013b] Guan, T., He, Y., Gao, J., Yang, J., and Yu, J.: On-device mobile visual location recognition by integrating vision and inertial sensors. IEEE Transactions on Multimedia, 15(7):1688–1699, 2013.

[Harris and Stephens, 1988] Harris, C. and Stephens, M.: A combined corner and edge detector. In Alvey vision conference, volume 15, page 50. Citeseer, 1988.

[Hartigan and Wong, 1979] Hartigan, J. A. and Wong, M. A.: Algorithm as 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1):100–108, 1979.

[Heinly et al. , 2012] Heinly, J., Dunn, E., and Frahm, J.-M.: Comparative evaluation of binary features. In Computer Vision–ECCV 2012, pages 759–773. Springer, 2012.

[Horn et al. , 1988] Horn, B. K. P., Hilden, H., and Negahdaripour, S.: Closed form solution of absolute orientation using orthonormal matrices. Optical Society of America A, 1988.

[Jegou et al. , 2008] Jegou, H., Douze, M., and Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In European conference on computer vision, pages 304–317. Springer, 2008.

[Jégou et al. , 2009] Jégou, H., Douze, M., and Schmid, C.: On the burstiness of visual elements. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 1169–1176. IEEE, 2009.

[Jegou et al. , 2011] Jegou, H., Douze, M., and Schmid, C.: Product quantization for nearest neighbor search. IEEE transactions on pattern analysis and machine intelligence, 33(1):117–128, 2011.

[Jégou et al. , 2010] Jégou, H., Douze, M., Schmid, C., and Pérez, P.: Aggregating local descriptors into a compact image representation. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 3304–3311. IEEE, 2010.

[Jegou et al. , 2012] Jegou, H., Perronnin, F., Douze, M., Sánchez, J., Perez, P., and Schmid, C.: Aggregating local image descriptors into compact codes. IEEE transactions on pattern analysis and machine intelligence, 34(9):1704–1716, 2012.

[Jing et al. , 2013] Jing, Y., Covell, M., Tsai, D., and Rehg, J. M.: Learning query-specific distance functions for large-scale web image search. IEEE Transactions on Multimedia, 15(8):2022–2034, 2013.

[Katoh et al. , 1981] Katoh, N., Ibaraki, T., and Mine, H.: An algorithm for finding k minimum spanning trees. SIAM Journal on Computing, 10(2):247–255, 1981.

[Kendall et al. , 2015] Kendall, A., Grimes, M., and Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings of the IEEE international conference on computer vision, pages 2938–2946, 2015.

[Kingsbury, 2006] Kingsbury, N.: Rotation-invariant local feature matching with complex wavelets. In Proc. European Conference on Signal Processing (EUSIPCO), pages 901–904, 2006.

[Knopp et al. , 2010] Knopp, J., Sivic, J., and Pajdla, T.: Avoiding confusing features in place recognition. In European Conference on Computer Vision, pages 748–761. Springer, 2010.

[Krizhevsky and Hinton, 2011] Krizhevsky, A. and Hinton, G. E.: Using very deep autoencoders for content-based image retrieval. In ESANN, 2011.

[Leutenegger et al. , 2011] Leutenegger, S., Chli, M., and Siegwart, R. Y.: Brisk: Binary robust invariant scalable keypoints. In 2011 International conference on computer vision, pages 2548–2555. IEEE, 2011.

[Li et al. , 2013] Li, H., Wang, Y., Mei, T., Wang, J., and Li, S.: Interactive multimodal visual search on mobile device. IEEE transactions on multimedia, 15(3):594–607, 2013.

[Li et al. , 2010] Li, Y., Snavely, N., and Huttenlocher, D. P.: Location recognition using prioritized feature matching. In European Conference on Computer Vision, pages 791–804. Springer, 2010.

[Lindeberg, 1998] Lindeberg, T.: Feature detection with automatic scale selection. International journal of computer vision, 30(2):79–116, 1998.

[Liu et al. , 2012] Liu, H., Mei, T., Luo, J., Li, H., and Li, S.: Finding perfect rendezvous on the go: Accurate mobile visual localization and its applications to routing. In Proceedings of the 20th ACM International Conference on Multimedia, MM '12, pages 9–18, New York, NY, USA, 2012. ACM.

[Lowe, 1999] Lowe, D. G.: Object recognition from local scale-invariant features. In Computer vision, 1999. The proceedings of the seventh IEEE international conference on, volume 2, pages 1150–1157. Ieee, 1999.

[Lowe, 2004] Lowe, D. G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2):91–110, 2004.

[Makar et al. , 2013] Makar, M., Tsai, S. S., Chandrasekhar, V., Chen, D., and Girod, B.: Interframe coding of canonical patches for low bit-rate mobile augmented reality. International Journal of Semantic Computing, 7(01):5–24, 2013.

[Massoud Sharif and others,] Massoud Sharif, A. et al.: Integrated approach to predict confidence of GPS measurement.

[Matas et al. , 2004] Matas, J., Chum, O., Urban, M., and Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and vision computing, 22(10):761–767, 2004.

[Min et al. , 2014] Min, W., Xu, C., Xu, M., Xiao, X., and Bao, B.-K.: Mobile landmark search with 3d models. IEEE Transactions on Multimedia, 16(3):623–636, 2014.

[Moisan et al. , 2012] Moisan, L., Moulon, P., and Monasse, P.: Automatic homographic registration of a pair of images, with a contrario elimination of outliers. Image Processing On Line, 2:56–73, 2012.

[Morel and Yu, 2009] Morel, J.-M. and Yu, G.: Asift: A new framework for fully affine invariant image comparison. SIAM Journal on Imaging Sciences, 2(2):438–469, 2009.

[Nister and Stewenius, 2006] Nister, D. and Stewenius, H.: Scalable recognition with a vocabulary tree. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, pages 2161–2168. IEEE, 2006.

[Nistér and Stewénius, 2008] Nistér, D. and Stewénius, H.: Linear time maximally stable extremal regions. In European Conference on Computer Vision, pages 183–196. Springer, 2008.

[Nowak et al. , 2006] Nowak, E., Jurie, F., and Triggs, B.: Sampling strategies for bag-of-features image classification. In European conference on computer vision, pages 490–503. Springer, 2006.

[Park et al. , 2009] Park, M., Brocklehurst, K., Collins, R. T., and Liu, Y.: Deformed lattice detection in real-world images using mean-shift belief propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(10):1804–1816, 2009.

[Philbin et al. , 2007] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, pages 1–8. IEEE, 2007.

[Philbin et al. , 2008] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE, 2008.

[Philbin et al. , 2010] Philbin, J., Isard, M., Sivic, J., and Zisserman, A.: Descriptor learning for efficient retrieval. In European Conference on Computer Vision, pages 677–691. Springer, 2010.

[Philbin et al. , 2011] Philbin, J., Sivic, J., and Zisserman, A.: Geometric latent dirichlet allocation on a matching graph for large-scale image datasets. International journal of computer vision, 95(2):138–153, 2011.

[Reitmayr and Drummond, 2007] Reitmayr, G. and Drummond, T. W.: Initialisation for visual tracking in urban environments. In Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on, pages 161–172. IEEE, 2007.

[Roshan Zamir et al. , 2014] Roshan Zamir, A., Ardeshir, S., and Shah, M.: Gps-tag refinement using random walks with an adaptive damping factor. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4280–4287, 2014.

[Salarian et al. , 2018] Salarian, M., Iliev, N., Ansari, R., and Cetin, A. E.: Improved image-based localization using sfm and modified coordinate system transfer. IEEE Transactions on Multimedia, pages 1–1, 2018.

[Salarian and Ansari, 2016] Salarian, M. and Ansari, R.: Improved image retrieval for efficient localization in urban areas using location uncertainty data. In IEEE International Symposium on Multimedia (ISM),. IEEE, 2016.

[Salarian et al. , 2016] Salarian, M., Ileiv, N., and Ansari, R.: Accurate image based localization by applying sfm and coordinate system registration. In Multimedia (ISM), 2016 IEEE International Symposium on, pages 189–192. IEEE, 2016.

[Salarian et al. , 2015] Salarian, M., Manavella, A., and Ansari, R.: Accurate localization in dense urban area using google street view images. In SAI Intelligent Systems Conference (IntelliSys), 2015, pages 485–490. IEEE, 2015.

[Salarian et al. , 2017] Salarian, M., Sharifzadeh, M., and Ansari, R.: Image based localization based on feature scale consistency in bof vector. In Multimedia (ISM), 2017 IEEE International Symposium on, pages 31–37. IEEE, 2017.

[Sattler et al. , 2012] Sattler, T., Weyand, T., Leibe, B., and Kobbelt, L.: Image retrieval for image-based localization revisited. In BMVC, volume 1, page 4, 2012.

[Schindler et al. , 2007] Schindler, G., Brown, M., and Szeliski, R.: City-scale location recognition. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, pages 1–7. IEEE, 2007.

[Schmid and Mohr, 1977] Schmid, C. and Mohr, R.: Local grayvalue invariants for image retrieval. IEEE Trans. Pattern Anal. Mach. Intell., 1977.

[Schroth et al. , 2011] Schroth, G., Huitl, R., Chen, D., Abu-Alqumsan, M., Al-Nuaimi, A., and Steinbach, E.: Mobile visual location recognition. Signal Processing Magazine, IEEE, 28(4):77–89, 2011.

[Sivic and Zisserman, 2003] Sivic, J. and Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, pages 1470–1477. IEEE, 2003.

[Smith and Brady, 1997] Smith, S. M. and Brady, J. M.: Susana new approach to low level image processing. International journal of computer vision, 23(1):45–78, 1997.

[Song et al. , 2016] Song, Y., Chen, X., Wang, X., Zhang, Y., and Li, J.: 6-dof image localization from massive geo-tagged reference images. IEEE Transactions on Multimedia, 18(8):1542–1554, 2016.

[Takacs et al. , 2012] Takacs, G., Chandrasekhar, V., Tsai, S., Chen, D., Grzeszczuk, R., and Girod, B.: Rotation-invariant fast features for large-scale recognition. SPIE Optical Engineering and Applications, pages 84991D–84991D, 2012.

[Tola et al. , 2008] Tola, E., Lepetit, V., and Fua, P.: A fast local descriptor for dense matching. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE, 2008.

[Torii et al. , 2013] Torii, A., Sivic, J., Pajdla, T., and Okutomi, M.: Visual place recognition with repetitive structures. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 883–890, 2013.

[Torr and Zisserman, 2000] Torr, P. H. and Zisserman, A.: Mlesac: A new robust estimator with application to estimating image geometry. Computer Vision and Image Understanding, 78(1):138–156, 2000.

[Torralba et al. , 2003] Torralba, A., Murphy, K. P., Freeman, W. T., and Rubin, M. A.: Context-based vision system for place and object recognition. In Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, pages 273–280. IEEE, 2003.

[Tsai et al. , 2014] Tsai, S. S., Chen, H., Chen, D., and Girod, B.: Word-hogs: Word histogram of oriented gradients for mobile visual search. In 2014 IEEE International Conference on Image Processing (ICIP), pages 3968–3972. IEEE, 2014.

[Tung et al. , 2017] Tung, H.-Y. F., Harley, A. W., Seto, W., and Fragkiadaki, K.: Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision. In Proceedings of the IEEE International Conference on Computer Vision (ICCV17), volume 2, 2017.

[Turcot and Lowe, 2009] Turcot, P. and Lowe, D. G.: Better matching with fewer features: The selection of useful features in large database recognition problems. In Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, pages 2109–2116. IEEE, 2009.

[Tuytelaars and Schmid, 2007] Tuytelaars, T. and Schmid, C.: Vector quantizing feature space with a regular lattice. In 2007 IEEE 11th International Conference on Computer Vision, pages 1–8. IEEE, 2007.

[Tuytelaars and Van Gool, 2004] Tuytelaars, T. and Van Gool, L.: Matching widely separated views based on affine invariant regions. International journal of computer vision, 59(1):61–85, 2004.

[ul Hussain and Triggs, 2012] ul Hussain, S. and Triggs, B.: Visual recognition using local quantized patterns. In Computer Vision–ECCV 2012, pages 716–729. Springer, 2012.

[Vedaldi and Fulkerson, 2008] Vedaldi, A. and Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[Vishal et al. , 2015] Vishal, K., C.V., J., and Visesh, C.: Accurate localization by fusing images and gps signals. CVPR IEEE, 2015.

[von Hundelshausen and Sukthankar, 2012] von Hundelshausen, F. and Sukthankar, R.: D-nets: Beyond patch-based image descriptors. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 2941–2948. IEEE, 2012.

[Weiss et al. , 2009] Weiss, Y., Torralba, A., and Fergus, R.: Spectral hashing. In Advances in neural information processing systems, pages 1753–1760, 2009.

[Wikipedia contributors, 2018] Wikipedia contributors: Error analysis for the global positioning system — Wikipedia, the free encyclopedia, 2018. [Online; accessed 27-July-2018].

[Witten et al. , 1999] Witten, I. H., Moffat, A., and Bell, T. C.: Managing gigabytes: compressing and indexing documents and images. Morgan Kaufmann, 1999.

[Wu et al. , 2011] Wu, C., Agarwal, S., Curless, B., and Seitz, S. M.: Multicore bundle adjustment. CVPR IEEE, 2011.

[Xu et al. , 2012] Xu, X., Mei, T., Zeng, W., Yu, N., and Luo, J.: Amigo: Accurate mobile image geotagging. In Proceedings of the 4th International Conference on Internet Multimedia Computing and Service, ICIMCS '12, pages 11–14, New York, NY, USA, 2012. ACM.

[Yu et al. , 2011] Yu, F. X., Ji, R., and Chang, S.-F.: Active query sensing for mobile location search. In Proceedings of the 19th ACM International Conference on Multimedia, MM '11, pages 3–12, New York, NY, USA, 2011. ACM.

[Zamir and Shah, 2010] Zamir, A. and Shah, M.: Accurate image localization based on google maps street view. ECCV, 2010.

[Zamir et al. , 2012] Zamir, A. R., Dehghan, A., and Shah, M.: Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In Computer Vision–ECCV 2012, pages 343–356. Springer, 2012.

[Zhang et al. , 2011] Zhang, J., Hallquist, A., Liang, E., and Zakhor, A.: Location-based image retrieval for urban environments. In Image Processing (ICIP), 2011 18th IEEE International Conference on, pages 3677–3680. IEEE, 2011.

VITA

Mahdi Salarian

Education

Ph.D. Electrical and Computer Engineering University of Illinois at Chicago	2013 – 2018
M.S. Electrical Engineering University of Mazandaran	2004 – 2006
B.S. Electrical Engineering (Electronics) University of Guilan	1999 – 2004

Publications

Mahdi Salarian and Nick Iliev and Rashid Ansari and A. E. Cetin, Improved Image-Based Localization Using SFM and Modified Coordinate System Transfer, IEEE Transaction on Multimedia, May 2018.

Mahdi Salarian and Rashid Ansari and Mehdi Sharifzadeh, Image-based Localization based on feature scale consistency in BOF vector, IEEE International symposium on Multimedia (ISM2017).

Mahdi Salarian and Nick Iliev and Rashid Ansari, Accurate Image based Localization by applying SFM and Coordinate System Registration, IEEE International symposium on Multimedia (ISM2016), San Jose, USA, December 2016.

Mahdi Salarian and Rashid Ansari, An Efficient refinement of GPS-based localization in urban areas using visual information and sensor parameter, IEEE International Symposium on Multimedia (ISM2016), San Jose, USA, December 2016.

Mahdi Salarian and Andrea Manavell and Rashid Ansari, A vision based system for Traffic lights recognition, IEEE SAI Intelligent Systems Conference (IntelliSys 2015).

Mahdi Salarian and Rashid Ansari and Andrea Manavell, Accurate Localization in Dense Urban Area Using Google Street View Image, IEEE SAI Intelligent Systems Conference (IntelliSys 2015).

Mahdi Salarian and Rashid Ansari and Justin Wanek and Mahnaz Shahidi, Accurate automatic segmentation of retina layers with emphasis on first layer, IEEE International Conference on Electro/Information technology, EIT2015.

Awards — Travel award for presentation in ISM 2016

Presentations — Conference Presentations at EIT 2015, ISM 2016, ISM 2017
Poster Presentations at ICASSP 2018

Memberships — IEEE student membership

Experience — Graduate Assistant at Lions of Illinois Eye Research Institute
Research Assistant at UIC
Intern at CCC information service
Instructor at UIC
Teaching Assistant at UIC