VISUAL GEO-LOCALIZATION AND LOCATION-AWARE IMAGE UNDERSTANDING

by

AMIR ROSHAN ZAMIR
M.S. University of Central Florida, 2013

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2014

Major Professor: Mubarak Shah

ABSTRACT

Geo-localization is the problem of discovering the location where an image or video was captured. Recently, large scale geo-localization methods which are devised for ground-level imagery and employ techniques similar to image matching have attracted much interest. In these methods, given a reference dataset composed of geo-tagged images, the problem is to estimate the geo-location of a query by finding its matching reference images. In this dissertation, we address three questions central to geo-spatial analysis of ground-level imagery: **1) How to geo-localize images and videos captured at unknown locations? 2) How to refine the geo-location of already geo-tagged data? 3) How to utilize the extracted geo-tags?**

We present a new framework for geo-locating an image utilizing a novel multiple nearest neighbor feature matching method using Generalized Minimum Clique Graphs (GMCP). First, we extract local features (e.g., SIFT) from the query image and retrieve a number of nearest neighbors for each query feature from the reference data set. Next, we apply our GMCP-based feature matching to select a single nearest neighbor for each query feature such that all matches are globally consistent. Our approach to feature matching is based on the proposition that the first nearest neighbors are not necessarily the best choices for finding correspondences in image matching. Therefore, the proposed method considers multiple reference nearest neighbors as potential matches and selects the correct ones by enforcing the consistency among their global features (e.g., GIST) using GMCP. Our evaluations using a new data set of 102k Street View images shows the proposed method outperforms the state-of-the-art by 10 percent.

Geo-localization of images can be extended to geo-localization of a video. We have developed a novel method for estimating the geo-spatial trajectory of a moving camera with unknown intrinsic parameters in a city-scale. The proposed method is based on a three step process: 1) individual geo-localization of video frames using Street View images to obtain the likelihood of the location (latitude and longitude) given the current observation, 2) Bayesian tracking to estimate

the frame location and videos temporal evolution using previous state probabilities and current likelihood, and 3) applying a novel Minimum Spanning Trees based trajectory reconstruction to eliminate trajectory loops or noisy estimations.

Thus far, we have assumed reliable geo-tags for reference imagery are available. However, crowdsourced images are well known to suffer from the acute shortcoming of having inaccurate geo-tags. We have developed the first method for refinement of GPS-tags which automatically discovers the subset of corrupted geo-tags and refines them. We employ Random Walks to discover the uncontaminated subset of location estimations and robustify Random Walks with a novel adaptive damping factor that conforms to the level of noise in the input.

In location-aware image understanding, we are interested in improving the image analysis by putting it in the right geo-spatial context. This approach is of particular importance as the majority of cameras and mobile devices are now being equipped with GPS chips. Therefore, developing techniques which can leverage the geo-tags of images for improving the performance of traditional computer vision tasks is of particular interest. We have developed a location-aware multimodal approach which incorporates business directories, textual information, and web images to identify businesses in a geo-tagged query image.

To my parents,

for their love and sacrifices.

$\sim$

To my brother and sister-in-law,

for their continual support.

ACKNOWLEDGMENTS

First and foremost, I would like to thank Dr. Mubarak Shah for his advice over the past few years. I feel very fortunate for having been part of Dr. Shah's research group and for being given the opportunity to be involved with and learn from a wide range of projects. I would like to thank all of the past and present members of the Center for Research in Computer Vision (CRCV), particularly Dr. Vladimir Reilly, Dr. Subhabrata Bhattacharya, Dr. Yang Yang, Dr. Ramin Mehran, Dr. Paul Scovanner, Dr. Omar Oreifej, Dr. Pavel Babenko, Dr. Imran Saleemi, Dr. Kishore Reddy, Dr. Asaad Hakeem, Dr. Berkan Solmaz, Dr. Mikel Rodriguez, Dr. Niels Lobo, Dr. Enrique Ortiz, Gonzalo Vaca, Afshin Dehghan, Shayan Modiri, Shervin Ardeshir, Yusuf Aytar, Alejandro Torroella, Rui Hou, Khurram Soomro, Mahdi Kalayeh, Dong Zhang, Salman Khokhar, Alexander Darino, Neil Gealy, Corey Pittman, Stephanie Morris, Ruben Villegas, Ryan Patrick, Soumyabrata Dey, Behnaz Nojavan, Haroon Idrees, Cherry Place, Tonya LaPrarie, Brittany Kaval, Oliver Nina, and Jonathan Poock for their support, the good times, and the great memories. Also, I was honored to work with Dr. Jingen Liu, Dr. Yu-Gang Jiang, Dr. Massimo Piccardi, Dr. Hamed Pirsiavash, and Dr. Sumit K. Jha and benefit from their support. Lastly, I want to thank Dr. Rahul Sukthankar for the kindness he extended towards me and for his selfless guidance. This journey has been long but I am so thankful for those that assisted me through it.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

The geo-location where an image or video was captured plays a key role in several fundamental tasks across the fields of Computer Vision, Multimedia, and Photogrammetry, and therefore, have a momentous importance. For instance, the prominent systems for organization and analysis of aerial and satellite imagery, e.g. USGS[9] and ArcGIS[9], structure their data based on their geo-locations, or the most popular online photo repositories, e.g. Panroamio[10], organize their databases using geo-locations and present them to the user in a geographically structured manner.

However, the geo-locations[1] of a considerable percentage of images and videos are often not recorded at the time of collection. Thus, a substantial amount of attention has been placed on developing automatic techniques for identifying the location of an image or video, commonly referred to as *visual geo-localization*, using a geo-referenced dataset. Until early 2000s, the majority of automatic visual geo-localization methods were targeted towards airborne and satellite imagery. That is, the furnished datasets were mainly composed of images with nadir or limb views, and the query data was captured from either satellites or aircrafts. Towards this end, many successful methods [11, 12, 13] were developed which were primarily based on planar-scene registration techniques and often utilized subsidiary models, such as Digital Elevation Model (DEM) and Digital Terrain Model (DTM), along with regular aerial imagery and Digital Orthophoto Quadrangles (DOQ).

However, during the past decade, the production of visual data has undergone a major shift with the sudden surge of consumer imagery which mainly retains a *ground-level* viewpoint. The shift was mostly due to the plummeting cost of the photographic devices as well as the increasing convenience of sharing the multimedia material including pictures. Currently, the ground-level

[1] also known as *geo-tags*.

images and videos are primarily produced either through systematic efforts by governments and the private sector (e.g. Google Street View) or directly by consumers (crowdsourcing). Google Street View, which provides dense spherical views of public roadways, and Panoramio Collection, which is composed of crowdsourced images, are two notable examples of such structured and unstructured databases, respectively. The coverage of these two resources for Pittsburgh, PA is shown in Fig. 1.1.

Street View Images Crowdsourced Images

Figure 1.1: The coverages of Street View and crowdcourced images, two main resources of ground-level data, are shown for Pittsburgh, PA. Each marker represents one Street View place mark (shown in green) or a user shared image (shown in various colors).

This considerable shift in the production of visual data poses new challenges in the context of automatic geo-localization and demands novel techniques which are capable of coping with this substantial change. The following are the main challenges which have emerged as a result of this shift:

- **Immensity of the data**: The amount of produced data is prohibitively massive and is increasing sharply. This makes devising techniques which are efficient in preprocessing (leveraging

the large amount of reference data) and query processing (sifting through the preprocessed data) essential.

- **Necessity of an accurate geo-location**: many of the procedures which use geo-tags as their input require a precise geo-location, particularly in the urban areas. In general, extracting a coarse geographical location, e.g. which continent the image was captured at or distinguishing between desert and coast, has limited applications, and performing the geo-localization with an accuracy comparable to, or better than, handheld GPS devices is desirable.

- **Ambiguity and excessive similarly of visual features**: Unless the data includes distinctive objects, such as landmarks, discovering the location merely based on visual information is often challenging due to the significant similarity between man-made structures. This issue becomes critical when city or country scale geo-localization is of interest.

- **Undesirable photography effects**: Unwanted effects, such as suboptimal lighting, frequent occlusions by moving objects, lens distortions, or stitching artifacts, often introduce additional complexities.

This dissertation targets answering the three foremost questions central to geo-spatial analysis of ground-level visual data:

1) How to geo-locate images and videos taken from unknown locations? What are the main unresolved challenges?

2) How to refine the geo-location of already geo-tagged images? particularly when the geo-tags are expected to include inaccuracies (e.g. when obtained through crowdsourcing).

3) How useful are the geo-tags? How can content analysis be enhanced using the discovered geo-tags?

In the coming sections, we elaborate on each of these questions, discuss the fundamental unresolved difficulties, and overview our developed solutions.

1.1 Image-Matching based Geo-localization

The conventional techniques devised for geo-registration and geo-localization of airborne imagery[11, 13, 12] fail to localize ground level data due to the significantly dissimilar characteristics of this problem in the ground-level and aerial views: Non-planarity of the scene, limited value of top-view subsidiary data (e.g. DEM, DOQ), frequent occlusions, and the complexity of the pictured scene are some of these differences. This highlights the importance of developing techniques customized for geo-localization of the ground-level data.

Since no single reference image similar to DOQ or Google Earth imagery is available for ground-level geo-localization purposes, a *collection* of geo-tagged ground-level images can be employed as the reference data. Using such dataset, a coarse geo-localization can be performed, i.e. a single-spot geo-location is assigned to the whole image instead of the pixel-wise geo-localization in the traditional geo-registration using DOQ. The discovered singe-spot geo-location is typically expected to be the location of the camera at the time of collection.

Given such reference dataset, one approach to finding the geo-location of a query image is to employ an image-matching based strategy. Image matching and image geo-localization share a great deal of similarity in terms of the problem definition and the challenges faced, such as the volume of the data, necessity of dealing with undesirable photography effects, non-planarity of the scene, and frequent occlusions by irrelevant objects. Inspired by the outstanding advances in the area of Internet-scale image matching, the image-matching based approach to image geo-localization has recently attracted a lot of interest in the Computer Vision community [14, 15, 2]. In this approach, the geo-localization is performed by finding the references image(s) with the highest amount of similarity to the query and estimating the query's location based on the geo-tags

of the found images. This approach particularly suits geo-localization of ground-level data due to the aforementioned mutually faced challenges.

However, image geo-localization and image matching differ in a few fundamental aspects: In image matching, the ultimate goal is to find *all* of the images which match the query with different amounts of similarity. On the contrary, in image geo-localization, the goal is to propose the best location for the query which does not necessarily require finding a large number of matching images. For instance, estimating the location of the query is deemed easier having a few geo-tagged images with relatively similar viewpoints (and probably substantially similar in content) as compared to a large number of not-so-similar images. Additionally, all forms of resemblance, such as semantic similarity (e.g. sharing generic objects), is typically in the interest of image matching. In contrast, the primary objective in image geo-localization is to find the images which indeed show the *same* location, and not just a similar one. Such divergences signify that adopting image matching techniques off-the-shelf is not an adequate solution for geo-localization, and devising methods specifically intended for the task of localization, even if based on matching, is vital.

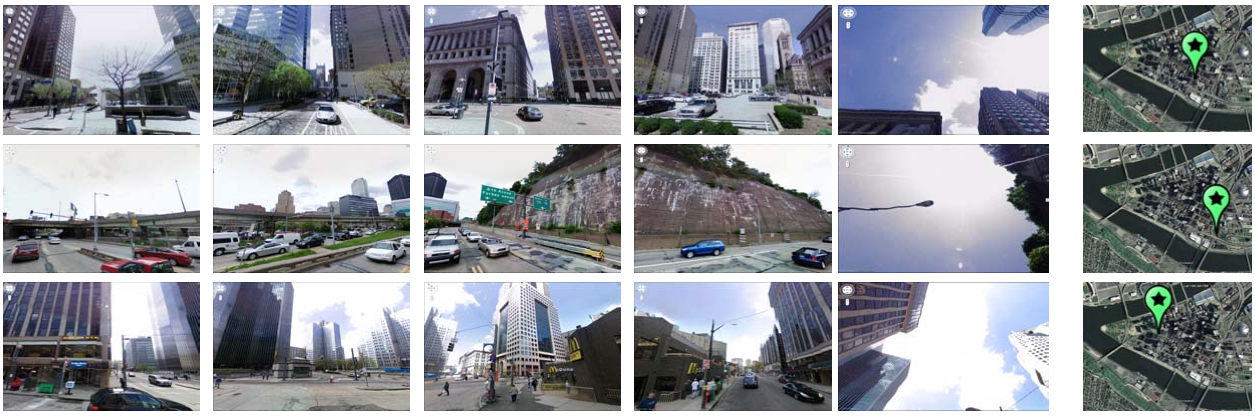

Figure 1.2: Sample Street View Images from Pittsburgh, PA. Each row shows one place mark's side views, top view and map location.

1.2 Summary of the dissertation

The following sections provide an overview of the novel methods described in this dissertation for the problems of automatic geo-localization of images and videos, refinement of geo-tags, and location-aware image understanding.

As the reference data of our geo-localization methods, we use a set of Google Street View images covering the three cities of Pittsburgh, PA; Orlando, FL and partially Manhattan, NY (downtown and the neighboring area of each). We break the 360° spherical view of the Street View place marks into five images composed of four side views and one top view, as shown in Fig. 1.2. We assign the GPS-tag of the center of the place mark, i.e. the camera location, as the geo-location of each image.

1.2.1 Image geo-localization based on local feature matching and geo-spatial pruning

We developed a new framework for accurate geo-localization of ground-level images which adopts the image matching-based approach, yet we put forth remedies for the challenges peculiarly faced in localization. The developed method is capable of estimating the location with an accuracy comparable to handheld GPS devices.

To preprocess the reference data, we extract SIFT features from the Street View images and organize them in a k-means tree to enable performing a timely search at the query time. In order to geo-localize a query image, SIFT features are extract from the image and the NNs of each query feature are retrieved from the k-means tree. As thoroughly discussed in the literature [16], pruning the correspondences established in this manner is essential as many of such NNs are incorrect. We argue that traditional approaches to correspondence pruning, such as SIFT ratio proposed by Lowe in the SIFT paper [16], have a suboptimal performance when geo-localization in the urban area is the problem at hand. That is primarily due to the repetitive patterns in the architectural features of man-made structures which makes the 1^{st} and 2^{nd} NNs of local features significantly similar and

consequently invalidates measuring their level of distinctiveness by comparing them. Therefore, we developed a novel feature pruning method which incorporates the geo-spatial location of each NN in order to find the proper ones for the distinctiveness test. In this context, the proper NN is the one which does not belong to the same geo-spatial region as the 1^{st} NN's, and therefore, ensures the repetitive architectural structures are not contributing to the similarity of the features. We observed that this approach typically yields a few, but mostly correct, correspondences which is the main reason behind its superior final geo-localization results.

We use the geo-location of the reference NNs which survive the pruning step to form a likelihood map for the location of the query image (see Fig. 1.3). The likelihood map is compiled by employing a voting scheme in which each reference NNs votes for the location where its parent image was captured. Finally, a 2D Gaussian smoothing step, intended to suppress the noise in the likelihood map, is applied before the geo-location with the highest likelihood is selected as the location of the query.

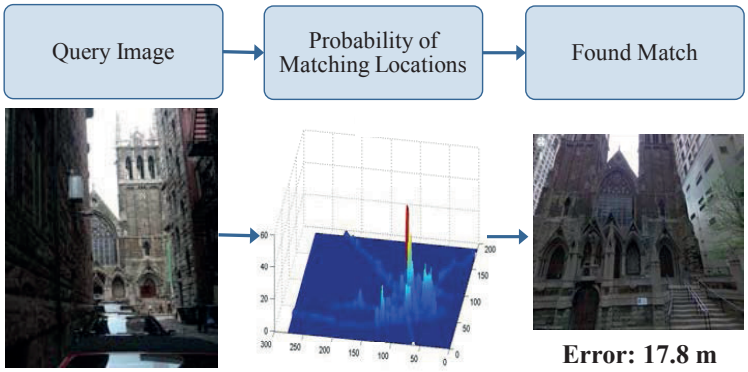

Figure 1.3: We devise an image-matching based approach to image geo-localization which yields a likelihood map for the location of the query.

We will demonstrate that this framework effectively leverages the structured nature of the Street View data in discovering the precise location of a query image in a city scale. We also avoid

using any operation which involves an information loss, e.g. the quantization distortion in Bag or Visual Words (BoVW) model [17, 14, 18], in the features. That is again due to the excessive similarity and repetitive architectural features in an urban area which makes the fine differences between local features the actual cues to the right location, while these differences are lost by quantization. This requires working with raw features that essentially increases the complexity of the search, which we alleviate by using NN trees (e.g. k-means or k-d), but yields a significantly higher accuracy of localization. We will experimentally demonstrate this in chapters 3 and 4.

In addition, we developed a novel approach to localizing *groups* of images, as opposed to a single image, in a hierarchical manner. Our method is based on the cue that the images which appear in one album or the ones which have close timestamps are often captured at geo-spatially nearby areas. In our method, each image is localized individually in the first step; then, the rest of the images in the group are matched against the neighboring area found for the first image. This process is repeated for all of the images in the group. Each neighborhood, and the individual locations within, which are found to have the highest overall *confidence*, are selected as the final estimation for the locations. The developed image group localization technique can deal with challenging queries which are not capable of being geo-localized individually.

1.2.2 Image geo-localization using multi-NN feature matching

Local features, such as SIFT, have been heavily used in the Computer Vision literature due to their good performance in being view invariant, robust to partial occlusion, and their relative compactness. However, limiting the scope of a feature to a local patch makes discovering feature correspondences essentially prone to mismatches, which is commonly discussed as *ambiguity of local features* in the literature. Fig. 1.4 exemplifies this where the local features, marked with red, are misleadingly similar while they belong to totally dissimilar objects in a larger scope. This shortcoming of local features becomes tragic in image-matching based localization in urban area, as the architectural features of man-made structures often share substantial local similarities. For

instance, all buildings often possess vertical/horizontal structures or all windows usually have a rectangular form.

Moreover, establishing a correspondence between two local features is usually performed by finding the 1^{st} NN feature (i.e. using NN classifier). Due to the ambiguity of local features, this approach is considerably challenged when the NN search is performed among millions of local features, as the right match is often not the 1^{st} NN. A example is illustrated in Fig. 1.5 where the top four NNs of five local query features retrieved from more than 10 million reference features are shown. The correct NN, i.e. the one that actually belongs to an image of the same building, is marked with the yellow border; this signifies that the correct NNs are not necessarily ranked 1^{st}.

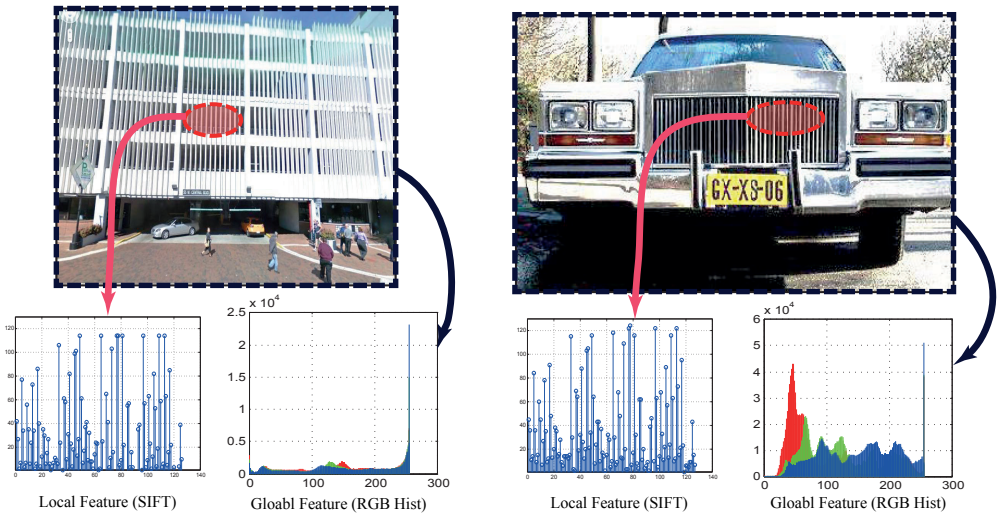

Figure 1.4: Two images with similar local features and dissimilar global feature. The local descriptor shows a misleadingly significant resemblance while the global features reveal that, in a larger scope, the local features can not be matching.

Assisting local feature matching with features which possess a larger scope (e.g. global features) and incorporating multiple NNs, as opposed to using only the 1^{st} NN, are potential remedies

to the two major drawbacks of the systems built upon local features discussed earlier. However, in that case, the fundamental question to answer is: *how the correct NN among the top retrieval NNs can be identified.*

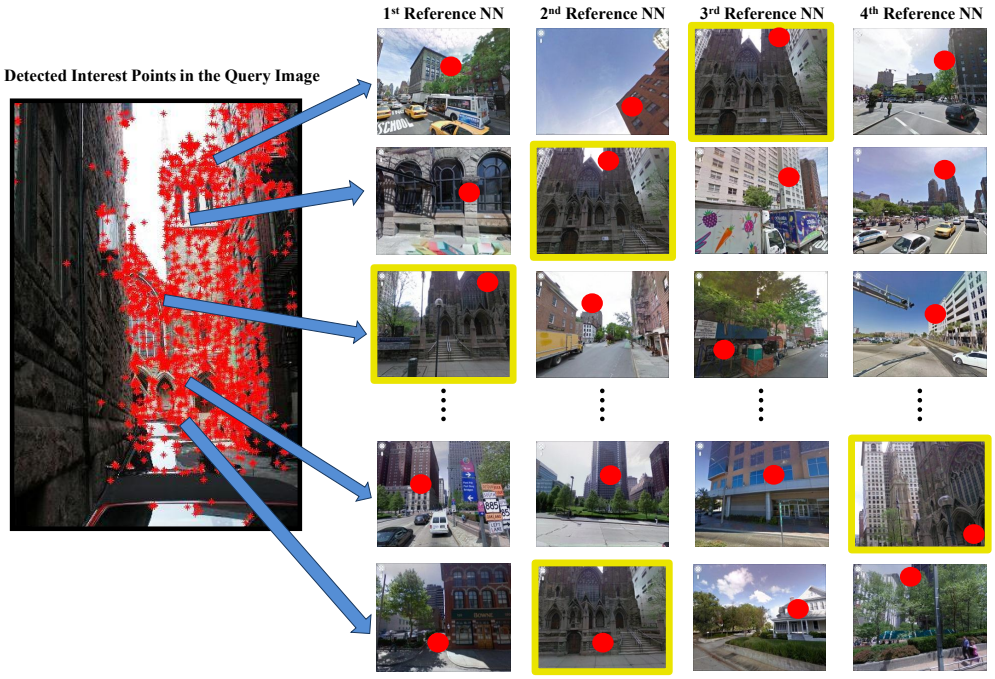

Figure 1.5: The top four reference NNs shown for five sample query features. The correct NNs are marked with the Yellow borders and signify that the 1^{st} NNs are not necessarily the correct ones.

To answer this question, we developed a new formulation for feature matching to address these two issues in a unified manner. Our approach has two main characteristics: *it utilizes local and global features simultaneously in the process of feature matching*, and *establishes the feature correspondences exploiting multiple-NNs*. In order to realize this, we use the Generalized Minimum Clique Problem (GMCP) at the core of our feature matching method. GMCP is useful in situations where there are multiple potential solutions for a number of subproblems, as well as a global criterion among the subproblems to be satisfied. In our framework, each subproblem is

matching a local query feature to the reference features, the potential solutions are the NNs, and the global criterion is the consistency of global features of the NNs. We apply our GMCP-based feature matching to select a single NN for each query feature so that all matches are globally consistent. In other words, the developed method considers multiple reference NNs as potential matches and selects the correct ones by enforcing consistency among their global features (e.g. color histogram) using GMCP. Generalize Subgraph Selection problems, including GMCP, are formally shown to be $\mathcal{NP}$-hard [19]. Thus, we developed an approximate combinatorial method for solving GMCP based on Local Neighborhood Search.

In this context, we argue that using a robust distance function for finding the similarity between the global features is essential for the cases where the query image matches multiple reference images with dissimilar global features. For this purpose, we develop a robust distance function based on the Gaussian Radial Basis Function (G-RBF). The proposed robustification can be viewed as finding the distances between the global features in a space transformed using the Gaussian Radial Basis Function (G-RBF) kernel. We will show that the geo-localization obtained by employing the proposed feature matching approach significantly outperforms the state-of-the-art methods.

1.2.3 Video geo-localization

Thus far, we discussed how images can be automatically geo-localized. However, videos have a significant share of the rapid surge of the consumer visual data. As of August 2013, more than 2400 hours of video were being uploaded to YouTube each day; over 20 times more than the year 2008. Therefore, developing geo-localization methods specifically devised for videos is particular interest. Even though some forms of the problem of geo-localizing a video has previously appeared in Robotics (e.g. SLAM, robot navigation[20, 21]) or Structure from Motion [22], it is still a significantly young and undeveloped area of research compared to image geo-localization. This can be mainly attributed to the conventional limitations in computational re-

sources, and the limited availability of consumer videos in the past which lead to a lesser demand for geo-localization of videos. However, with the popularity of social media and video sharing websites, such as YouTube, the necessity of developing techniques for geo-localization of videos is beyond doubt.

A video is in fact a sequence of images (i.e. video frames) which includes temporal information for each frame. Therefore, a video geo-localization system can be built on the techniques developed for image geo-localization with added mechanisms for utilization of the temporal information. Moreover, a geo-spatial trajectory is a more appropriate identifier for the geo-location of a video, as compared to a single-spot location, since the user may not remain stationary over the course of the video.

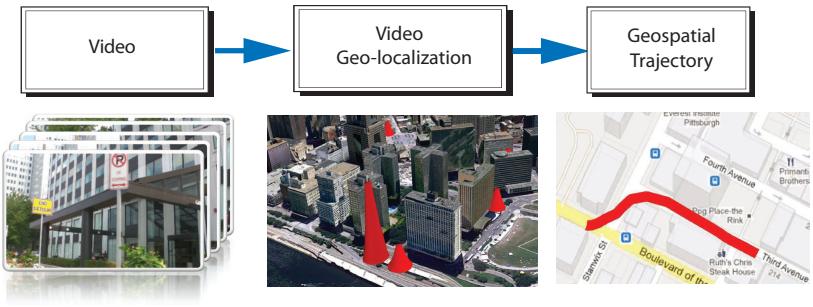

Figure 1.6: We developed a novel approach to geo-localizing videos based on Bayesian filtering and a novel curve reconstruction method which is free of any parametric-mode to cope with the typically stochastic motion of human.

A conventional approach to extracting the geo-spatial trajectory of a camera is to establish a geo-metric relationship between the consecutive frames, often using Structure from Motion [22]. However, this approach is significantly challenged when it comes to consumer videos, e.g. YouTube clips, as they do not completely handle the undesirable cinematographic effects, such as abrupt moves, blurred frames or the absence of metadata. In addition, we will argue that estab-

lishing such convoluted relationship between all of the video frames, even if plausible, may not be necessary for estimating a geo-spatial trajectory.

The availably of large scale ground-level reference data empowers a different approach to video geo-localization which can be viewed as a dual of the image-matching based image geo-localization, but specifically targeted for videos. That is because the geo-locations of video frames can be *roughly* hypothesized through matching individual frames to a reference dataset of images. This approach does not suffer from a high rate of failure in geo-localizing a single frame (unlike SfM) and paves the way for incorporating the temporal information in a forthcoming step.

To be more specific, our video geo-localization method is based on a three step process that includes: 1) Finding the best visual matches of individual frames in the reference dataset of Street View images. This yields a probability distribution, not a single-spot, for the location of each frame which is interpreted as the likelihood (latitude and longitude) given the current observation (i.e. video frame/segment). 2) Bayesian Tracking to estimate the frame location and enforcing the temporal consistency based on the previous state probabilities and the current likelihood. 3) Finally, an offline trajectory reconstruction which eliminates the remaining noisy estimations. This step is particularly essential for consumer videos since human motion, when recording a video, often does not follow a parametric motion model, such as constant velocity. Therefore, the proposed trajectory reconstruction method is free of any parametric-model to remedy the rather stochastic human motion.

We will demonstrate that performing merely frame-by-frame geo-localization is not an adequate solution to video localization as it completely ignores the temporal information. We will show that the developed method effectively enforces the temporal consistency and is capable of coping with complexities of consumer videos, yet it enjoys the advantages of a matching-based approach.

1.2.4 Geo-tag refinement on crowdsourced ground-level images

As extensively discussed in the previous sections, automatic visual geo-localization requires a geo-tagged reference dataset. Potentially, aerial imagery could serve as the reference data with a vast coverage. However, the performance of the state-of-the-art localization techniques which are robust to radically different viewpoints, commonly termed *cross-view matching*, is still far from satisfactory. Therefore, a reference resource composed of ground-level or near-ground-level data is essential for precise localization of ground-level queries.

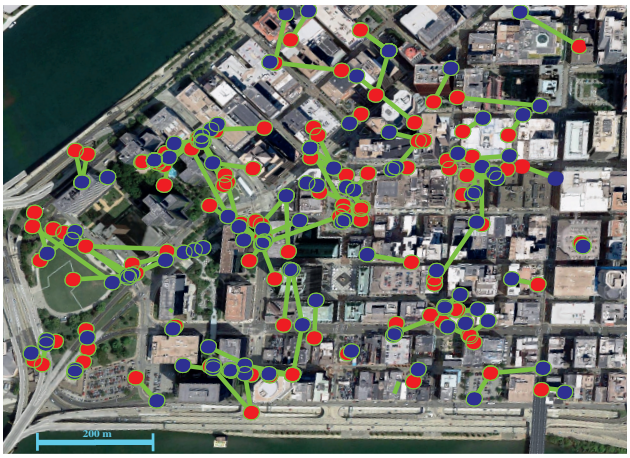

Figure 1.7: The user-specified GPS-tags (blue) of about 100 images from Pittsburgh along with their correct GPS-locations (red). The green line connects the user-specified location to the ground truth. Significant inaccuracies in the GPS labels can be observed.

As we will qualitatively and quantitatively demonstrate, Street View is a decent candidate with a dense place-mark-to-place-mark coverage and precise geo-tags. However, the coverage of Street View is yet limited to only a few percent of the populated areas of the world (currently 45 out of 195 countries with an insignificant coverage for many of them). On the contrary, crowdsourced images are available for most of the inhabited regions, and their density is climbing up rapidly even

for underdeveloped countries. Therefore, they can be potentially adopted as a reference dataset of geo-tagged ground-level images.

However, crowdsourced images are well known to suffer from a major drawback: *the user provided geo-tags often include significant inaccuracies* (see Fig. 1.7). To alleviate this issue, we developed the first method for refinement of GPS-tags. We assume a large dataset of GPS-tagged images which includes an unknown subset with inaccurate tags is available. We developed a robust method for identification and refinement of the subset with contaminated tags using the rest of the images in the dataset. In other words, the proposed method is capable of performing *self-refinement* and does not need to be supplied with uncontaminated data.

Robustness is a key trait of the proposed method which we effectuate using Random Walks. Random Walks have been applied to a wide range of problems, such as document retrieval or web image search [23, 24, 25]. A random walk is a special case of Markov Chain which, in high level, can be used for discovering a subset of highly consistent elements out of a superset. This can be imagined by assuming a person is to walk from one node of a graph to another and count the number of times each node is visited while the probability of selecting the next node to travel to is acquired from a predefined consistency between the nodes. Thus, after a sufficiently large number of walks, the nodes which are more consistent to one another are visited more often and consequently have a higher final relevance score. We leverage this property to evaluate the given GPS-tags and adjust the inaccurate ones to a better location.

To be more specific, for each image in the dataset, we retrieve a set of matching images from the rest of the dataset. We form image triplets composed of the query and two of the retrieved images. The triplet is then used for estimating the location of the query utilizing Structure from Motion. We generate a large number of such estimations, which may include inaccurate ones due to the noisy GPS-tags in the dataset, and perform random walks on them in order to identify the subset with the maximal agreement. This subset typically corresponds to the uncorrupted estimations, and we use it for refining the GPS-tag of the query image by finding the weighted

mean of the geo-location of the subset elements.

Random Walk formulation includes a term called damping factor which is primarily intended to inject prior knowledge about the data into the diffusion process. We argue that the Random Walks with the conventional constant damping factor are prone to noise in their input and develop an *adaptive damping factor* which conforms to the estimated level of noise in the input; consequently, the diffusion process is robustified even further. We empirically demonstrate that the proposed approach achieves desirable characteristics in terms of the formal measures of Robust Statistics, e.g. high Breakdown Point or descending Influence Function [26], and is capable of refining signifiant amounts of noise in the geo-tags of images.

1.2.5 Location-Aware Image Understating: Enhancing content analysis using geo-tags

Thus far, we discussed how the geo-location of an untagged image or video can be automatically extracted. In the last part of this dissertation, we wish to address a principal question in the area of visual geo-spatial analysis: *How useful are the extracted geo-tags? How can the geo-tags assist analyzing the visual content?*

These questions are particularly important as the majority of cameras, cell phones, and mobile devices are now being equipped with internal localization chips, such as GPS. Thus, a notable part of the visual data produced in the future is going to be associated with a coarse or precise geo-tag at the time of collection. This becomes even more inclusive with the emergence of new, and often inexpensive, localization techniques such as WiFi Position System (WPS) [27] or mobile phone tracking using cell phone signals.

Even though the geo-locations of images and videos have been previously utilized for various applications, e.g. geographical database organization[10] or Photo Tourism[28], we argue that a considerable part of their potential usages has yet to be explored. We promote the function of the geo-tags to a higher level and argue that *they can be utilized to enhance understating the image content*. We show that performing image analysis in a *location-aware* manner can assist

traditional computer vision problems, e.g. objection detection, or enable solving new applications, e.g. Business Recognition.

In order to demonstrate this, we devise a new problem called Visual Business Recognition which is defined as precise identification of the storefronts in a query image (see Fig. 1.8). This is an interesting and practical task with plenty of potential applications especially for mobile device users. We develop a location-aware multimodal approach to this problem which incorporates business directories, textual information, and web images in a unified framework.

Figure 1.8: We present a location-aware framework capable of identification of businesses in images. Sample results of our method are show in the above figure.

We assume the query image is geo-localized, use the GPS-tag for searching through the business directories and extract an over-complete list of nearby businesses which may be visible in the image. In the first modality, we form search keywords based on the names of the nearby businesses in order to automatically collect a set of relevant images from the web. We perform image matching between the retrieved images and the query, which yields a distribution for the probability of each nearby business to be visible in the image. For the second modality, we developed a text processing method customized for business recognition assisted by a lexicon formed based on the names of nearby businesses. We formulate our text recognition approach as an optimization prob-

lem in which we consider multi potential hypotheses for the recognition of each character detected in the image and minimize the edit distance between the permutations they induce and the lexicon using Local Search. This also yields a distribution specifying a probability of each nearby business to be visible in the image. Finally, we fuse the distributions acquired from the two modalities (i.e. image matching and text processing) in a probabilistic framework to recognize the business(es).

We will demonstrate that the developed method, which is built upon the assumption of the availability of a geo-tag for the image, can precisely recognize the businesses in complex user-shared images; a task which is deemed extremely difficult, if not impossible, in the absence of the geo-tags. We will also show that the novel multi hypothesis formulation for character recognition can effectively cope with the complex appearance of text in the storefronts.

1.3 Thesis Organization

The rest of the dissertation is structured as follows: In Chapter 2, an overview of existing methods for image geo-localization, leveraging the global context of images, trajectory extraction from videos, and geo-spatial analysis of images is provided. In Chapters 3 and 4, we present our approach to geo-localizing images and describe our GMCP-based multi-NN feature matching. In Chapter 5, we discuss our method for discovering the geo-location of a video and extracting its geo-spatial trajectory. In Chapter 6, we present our robust method for refinement of image geo-tags. In Chapter 7, we present our location-aware image understanding framework for Visual Business Recognition. Finally, Chapter 8 consists of the conclusion and a discussion on the identified directions for future works.

CHAPTER 2: LITERATURE REVIEW

In this chapter, we review the prominent approaches in the area of geo-spatial analysis of images and videos and explain the differences between the existing techniques and the methods introduced in this thesis. First, we provide an overview of the current methods for automatic image geo-localization and categorize them based on their accuracy and employed approach. Next, we discuss the existing methods for extracting the geo-spatial trajectory of a camera which can be employed for video geo-localization. A discussion on the current methods for tag refinement, tag ranking and re-tagging of images is also provided, and the differences dealing with a numerical tag, i.e. GPS, poses compared to textual tags is discussed. Finally, we provide an overview of the current applications of having geo-tagged data and discuss the existing methods which can be employed for location-aware recognition of businesses in images.

2.1 Image Geo-Localization

The earliest approaches to estimating the location of an image were devised for aerial imagery. In this context, several successful methods were developed [11, 12, 13] which were primarily based on geo-registration techniques and leveraging the constraints that the planarity of the scene in the aerial view warrants. Such methods commonly fail if the scene is not largely planar or an approximate location from the metadata cannot be acquired.

In contrast, the methods devised for geo-localizing ground-level imagery mainly employ an approach similar to image matching [14, 29, 15]. Several methods in this context have been developed: Schindler et al. [14] presented a method for city scale localization based on the bag of visual words model [30] using a dataset of street side images. They proposed a greedy algorithm for improving the accuracy of searching a vocabulary tree. Knopp et al. [29] proposed an approach to generating a codebook which discards the words which are identified to be non-discriminative

for geo-localization purposes. Doersch et al. [31] discover a set of visual features which are exclusive to a geo-graphical area. Sattler et al. [32] developed a framework similar to [33] for identifying 2D-to-3D correspondences between the query and the reference dataset with a large number of user shared images. In [34], they presented an efficient method for the same purpose based on both 2D-to-3D and 3D-to-2D matching. Compared to geo-registration techniques, these approaches typically provide a higher robustness with respect to partial occlusion, non-planarity of the scene and existence of moving objects in the image which are essential for localization of ground-level imagery.

The aforementioned methods can be further categorized based their accuracy of localization. For instance, Lin et al. [35] estimate a rough location for a ground-level query by performing cross-view matching using a training set of ground-aerial pairs, or Hays and Efros [15] extract coarse geographical information from a query image with the nominal accuracy of hundreds of kilometers. In contract, [14, 29, 36, 37, 38] aim at performing the localization with an error typically in the order of few tens of meters.

In addition, the existing approaches can be classified into methods which are based on epipolar geometry techniques [22, 32, 33, 39] or the ones which attempt to discover the most similar reference images without the need to explicit incorporation of the geometry of the scene [15, 14, 29, 37, 38, 35, 40]. The geometry based techniques are capable of providing a higher accuracy at the expense of a larger failure rate, while the other methods handle the variations in the images and the challenges of establishing point correspondences in a more robust manner.

Our approach to image localization (Chapters 3 and 4) is similar to some of the aforementioned methods in the sense that we adopt the image-matching based approach and perform the geo-localization with an error of few tens of meters. The high-level differences between our approach and the majority of aforementioned ones are: performing the matching based on raw local features instead of quantizing them into a vocabulary, incorporating the geo-spatial information in pruning the feature correspondences, and leveraging the structure of organized dataset such as

Street View in the localization process. We also utilize a novel multi-NN feature matching method which uses both local and global features in the process of localization.

2.1.1 Contextual and Global Image Descriptors

As discussed in the previous chapter, most of the existing methods for location recognition only utilize local features which ignore the global context of the image, and consequently, make the established correspondences prone to mismatches. Several methods for embedding contextual information in local descriptors have been developed as a remedy to this shortcoming: Mortensen et al. [41] proposed an extension to SIFT by augmenting it with global curvilinear shape information. Mikolajczyk et al. [42] leveraged local feature and edge based information along with a geometric consistency verification for object class recognition. Cao et al. [43] present an approach similar to [41] to make SIFT affine invariant. Hao et al. [44] and Zhang et al. [45] proposed two methods for incorporating the geometry of the scene in image matching using bundles of local features generally termed "visual phrases".

In Chapter 4, we introduce a multiple-NN feature matching method which uses both local and global features to address the shortcoming of local features in leveraging the context. Despite the shared similarities in the high level goal, the aforementioned methods for leveraging the global context are fundamentally different from ours in four aspects: 1- Unlike most of the existing approaches which capture one particular type of contextual information [41, 46, 45], our method is capable of leveraging arbitrary global features such as the global color histograms or geo-location. 2- We do not embed the global context in the local feature vector. Therefore, the space in which local and global features are matched are kept separate, and different metrics can be used for each. 3- Our method matches all the features of one image simultaneously which essentially means they contribute to each others' match. This is different from the existing methods which perform feature matching on an individual basis [43, 41, 42]. 4- A number of methods perform geometric verification by fitting the fundamental matrix to a set of initially discovered correspondences in order

to remove the incorrect matches [17, 22]. Such methods are different from ours as we use global features in establishing the initial correspondences rather than pruning a set of already found correspondences. Moreover, the type of contextual information leveraged in such methods is limited to the spatial arrangement of features.

In addition, Robust estimation techniques, such as RANSAC, are commonly used in computer vision for performing a robust model estimation where the input data includes outliers. Such methods were adopted for discovering feature correspondences [47] and have been robustified by modified cost functions [48, 49]. However, despite the similarity in the overall goal, there is a difference between such methods and ours: we nominate multiple NNs as the potential matches for a query feature. By definition, GMCP enforces picking *one and only one* candidate for each query features, whereas in the basic RANSAC formulation, the aim is to select the inlier correspondences given a set of one-to-one matches.

2.2 Video Geo-Localization

Several methods for extracting the trajectory of the camera from a video, particularly in the context of robot localization, has been developed to date. Visual Odometry (VO) and Visual SLAM (V-SLAM) are the two main research topics in this area. Visual odometry (VO) is the process of estimating the egomotion of an agent using the single or multiple cameras connected to it. The term was originally coined by Nister in [50] due to the similarity of this concept to the wheel odometer on vehicles. Visual odometry is concerned only with local consistency (typically, over the last n poses) of the trajectory. Most methods assume some simplifying constraints such as having the camera attached to a vehicle, the availability of additional sensors (e.g. IMU), or the use of omnidirectional cameras. For instance, Scaramuzza [51] used the Non-Holonomic constraint to reduce the number of correspondences in the Structure from Motion problem. Tardif et al. [52] presented an approach for VO on a car using an omnidirectional camera which decoupled

the rotation and translation estimation. Howard [53] proposed a method for simultaneous visual odometry and localization of the camera which is based on estimating the relative motion from successive stereo image pairs. Less constrained approaches, such as Mouragnon and Lhuillier et al. [54], assume the use of a calibrated camera, which is not available in most of the consumer recorded videos.

In Visual SLAM (V-SLAM) the objective is to incrementally build a consistent map of the environment while simultaneously determining its location on the map [55]. Two main categories of V-SLAM methods include: 1) those that use filtering (like EKF) to fuse the information from all the images with a probability distribution [56, 57], and 2) keyframe methods that retain the optimization of batch techniques, like global bundle adjustment to selected keyframes [58]. These methods also depend on calibrated cameras and are highly sensitive to outliers, such as those caused by vehicles or pedestrians, that effect the consistency of the map. Even assuming ideal conditions, such as calibrated cameras and static scenery, the requirement of having robust point correspondences between frames, low accumulative error, removing the scale ambiguity, and the necessity of extracting a global position would render such methods unusable for extracting the geo-spatial trajectory of the camera from a consumer video in a city scale.

Moreover, some of these approaches require finding a geometric relationship between either different frames of the query video or the query frames and some reference data [59, 53, 54]. Establishing this geometric frame-to-frame or frame-to-reference data relationship may be feasible for controlled environments. However such methods achieve a limited success when applied to typical user-uploaded videos where difficulties such as frequent abrupt changes in camera motion, existence of uninformative or blurred frames, and lack of metadata, is taken into account.

On the contrary, our video geo-localization method features a Bayesian formulation to fuse the temporal information across the frames, and consequently, is capable of handling the aforementioned challenges the videos in the wild manifest. Cummins and Newamn also proposed a method (FAB-MAP) [20, 21] to appearance-based place recognition based on a Bayesian framework. The

main differences between FAB-MAP and our approach are: 1) FAB-MAP uses bag of visual words model to find similarity between images, while we use the localization method of Chapter 3 which has a better performance in the urban area. 2) We utilize an offline curve reconstruction algorithm to handle the inaccurate estimations that are mainly due to the presumptions about the camera motion in the Bayesian formulation which do not generalize well to human motion in consumer videos.

2.3 Tag Refinement, Tag Ranking and Re-tagging

Even though improving GPS-tags of images has not been deeply explored before, various operations on textual tags such as re-labeling or ranking, which have natural connections to GPS-tag refinement, have been extensively studied in the literature. For instance, Li et al. [60] proposed a method for finding the relevance of a textual tag to the image content based on accumulating the votes from visually similar images in their dataset. Liu et al. [24] developed an algorithm for ranking the textual tags based on their relevance to the image content. Zhu et al. [61] presented a textual tag refinement method based on decomposition of the user specified tags to a low-rank and a sparse component.

Similar to our method for GPS-tag refinement, all of these techniques utilize the image content for verifying the credibility of the tag associated with the image. What differentiates our method from these is keeping robustness a key factor in the design of our framework, and refining GPS-tags which are *numerical* and consequently pose a problem with different properties compared to textual tags.

2.4 The applications of geo-tags and Visual Business Recognition

The majority of the applications which utilize the geo-location of images, such as location-based retrieval [10] or large scale 3D reconstruction [62, 28], are not intended to provide a high-

level understanding of the image content (e.g. parsing the visible objects or recognizing the scene). With the exception of landmark recognition techniques [38, 44, 63] or a few other methods [64, 65] which use the geo-tags in connection with the image content, the potential impact of the image geo-tags on understating the image content is largely unexplored.

The developed Business Recognition framework (Chapter 7) centrally uses the geo-tags for recognizing storefronts in images and provides the user with a high-level understanding of the scene. Generally speaking, recognizing a storefront in an image can be accomplished using two categories of methods: *Non-visual sensor based* and *visual-content based*. The first group of methods rely purely on the data provided by non-visual sensors such as GPS, digital compass, and gyroscop embedded in devices [66, 67]. These approaches are often based on matching the sensor data to a reference set, such as matching the location information received from the GPS-chip of a mobile device to a geo-tagged business directory (e.g. Yelp [68]). Such methods generally require very precise sensor information and accurately tagged reference datasets. Therefore, they typically achieve a limited success as the precision of sensors and reference datasets are currently below the requirements of such methods. The second category of business recognition methods are based on solely processing the image content. Good examples of such approaches are the methods based on recognizing scene text in order to identify the businesses visible in the image [69, 70, 71]. These methods suffer from the complexity of the appearance of businesses in images, in particualr the text on the storefront. For instance, the current state of the art [69, 70, 71, 72] text detection and recognition methods for natural scenes still do not perform well in recognizing business signs.

Our business recognition method is based on a multimodal approach which leverages the sensor data (GPS-tags), image content, business directories, and storefront images saved on the web in a unified framework, and consequently, is capable of dealing with the aforementioned issues.

2.5 Chapter Summary

In this chapter, we provided an overview of the prominent works in the area of geo-spatial analysis of images. First, we reviewed the existing methods for image-matching based image geo-localization. We categorized these methods based on their nominal accuracy and the approach they employ. We also reviewed the existing techniques for leveraging the global context in feature matching and discussed their differences with our multiple-NN feature matching method. Then, we provided an outline of the conventional methods for camera motion estimation, particularly in the field of Robotics. We also overviewed the similarities and differences between the existing techniques for refinement and ranking of textual tags and our GPS-tag refinement method. Finally, we discussed that the majority of the methods which use the geo-tags do not utilize them in order to provide a high-level understanding of the image content. In this context, we overviewed the existing methods which can be used for automatic recognition of storefronts in images and described how they differ from our location-aware solution to this problem. In the next chapter, we describe our novel framework for image geo-localization using Street View imagery based on discovering strong correspondences between local features and geo-spatial pruning.

CHAPTER 3: IMAGE GEO-LOCALIZATION BASED ON LOCAL FEATURE MATCHING AND GEO-SPATIAL CORRESPONDENCE PRUNING

The availability of ground-level imagery empowers adopting an image-matching based approach [14, 29] to image geo-localization which is fundamentally different from the conventional methods like aerial geo-registration [11, 13, 12]. That is, a set of images which strongly match the query are found in the reference dataset, and the geo-location of the query is estimated based on the geo-location of the found matches. In this chapter, we introduce our approach to geo-localization of user-uploaded ground-level images based on Street View imagery.

As the reference dataset, we use a structured dataset of Street View images which provides a dense coverage of 360° spherical views from the public driveways in a large number of countries. Leveraging such a dataset has several advantages, such providing a comprehensive coverage, yet the large amount of data makes devising an efficient method essential. In our approach, we find the reference images which match the query based on establishing a large set of correspondences between the local features extracted from the query image and the local features of the Street View images. We use k-d trees [73] for offline organization of the large number of reference local features and making a timely NN search feasible. In order to deal with the repetitive architectural features in the urban area, the discovered correspondences are fed through a novel pruning method which incorporates the geo-spatial location of the features to identify and remove the incorrect correspondences. This yields a set of highly reliable correspondences which we use for identifying the query's location employing a voting scheme.

3.1 Google Maps Street View Dataset

We propose to use a comprehensive 360° structured image dataset in order to increase the accuracy of the localization task. The images extracted from Google Maps Street View are a very good example of such a dataset. Google Maps Street View is a comprehensive dataset which consists of 360° panoramic views of almost all main streets and roads in a number of countries, with a distance of about 12m between locations.

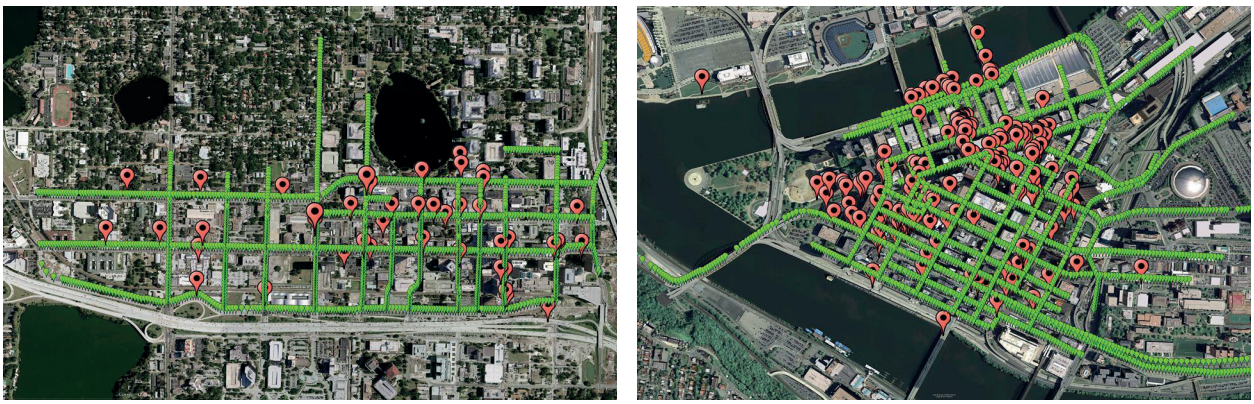

Figure 3.1: We use a dataset of about 100,000 GPS-tagged images downloaded from Google Maps Street View for Pittsburgh, PA (Right) and Orlando, FL (left). The green and red markers are the locations of reference and query images respectively.

Using a dataset with these characteristics allows us to make the localization task very reliable, with respect to feasibility and accuracy; this is primarily due to the comprehensiveness and organization of the dataset. The following are some of the main advantages of using datasets such as Google Maps Street View:

• Query Independency: Since the images in the dataset are uniformly distributed over different locations, regardless of the popularity of a given location or object, the localization task

is independent of the popularity of the objects in the query image and the location.

• Accuracy: As the images in the dataset are spherical 360° views taken about every 12 meters, it is possible to correctly localize an image with a greater degree of accuracy than would be permitted by a sparser dataset comprised of non-spherical images. The achieved accuracy is comparable to - and, in some cases, better than - the accuracy of hand-held GPS devices.

• Epipolar Geometry: The comprehensiveness and uniformity of the dataset makes accurate localization possible without employing methods based on epipolar geometry [22]- methods which are usually computationally expensive and, in many cases, lacking in required robustness. Additionally, the camera's intrinsic parameters for both the query and the dataset images are not required in order to accurately localize the images.

• Paving the Way for Secondary Applications: Using a structured database allows us to derive additional information, without the need for additional in-depth computation. For example, camera orientation can be determined as an immediate result of localization using the Google Maps Street View , without employing methods based on epipolar geometry. Since the dataset consists of 360° views, the orientation of the camera can be easily determined just by finding which part of the 360° view has been matched to the query image - a task that can be completed without the need for any further processing. Localization and orientation determination are tasks that even hand-held GPS devices are not capable of achieving without motion information.

However, the use of the Google Maps Street View dataset introduces some complications as well. The massive number of images can be a problem for fast localization. The need for capturing a large number of images makes using wide lenses and image manipulation (which always add some noise and geometric distortions to the images) unavoidable. Storage limitations make saving very high quality images impossible as well, so a matching technique must be capable of dealing with a distorted, low-quality, large-scale image dataset. The database's uniform distribution over different locations can have some negative effects - while it does make the localization task query-independent, it also limits the number of image matches for each query as well. For example, a

landmark will appear in exactly as many images as a mundane building. This is in direct contrast to other current large scale localization methods like Kalogerakis et al. [37], which can have a large number of image matches for a location in their database - a fact especially true if a location is a landmark; this allows the localization task to still be successful on a single match. The small number of correct matches in our database makes the matching process critical, as if none of the correct matches - which are few in number - are detected, the localization process fails.

We use a dataset of approximately 100,000 GPS-tagged Google Street View images, captured automatically from Google Maps Street View web site from Pittsburgh, PA and Orlando, FL. The distribution of our dataset and query images are shown in Fig. 3.1. The images in this dataset are captured approximately every 12 meters. The database consists of five images per place mark: four side-view images and one image covering the upper hemisphere view. These five images cover the whole 360° panorama. By contrast, Schindler et al.'s [14] dataset has only one side view. The images in their dataset are taken about every 0.7 meters, covering 20km of street-side images, while our dataset covers about 200km of full 360° views.

3.2 Single Image Localization

In order to accurately localize images, we use a method based on a nearest-neighbor tree search, with pruning and smoothing steps added to improve accuracy and eliminate storage and computational complexity issues.

During training, we process the reference dataset by computing the SIFT descriptors [16] for all interest points detected by the SIFT detector [16, 74]. Then, the descriptor vectors (and their corresponding GPS-tags) are organized into a tree using FLANN [73] . As we show later, a well-tuned pruning method allows us to find very reliable descriptors; as such, we generally need to compute at most $\frac{1}{6}$ of the number of interest points that Schindler et al. [14]'s method requires. Fig. 3.2 shows the block diagram of the proposed method for localizing a query image. In the first

step, the SIFT descriptors are computed for SIFT interest points in the same way as we process the dataset during training. Then, in the second step, the nearest-neighbors for each of the query SIFT vectors are found in the tree. Each of the retrieved nearest-neighbors vote for the image that they belong to. The votes can be shown as a plot over the actual map of the area covered by our reference dataset (as shown in third column of Fig. 3.2).

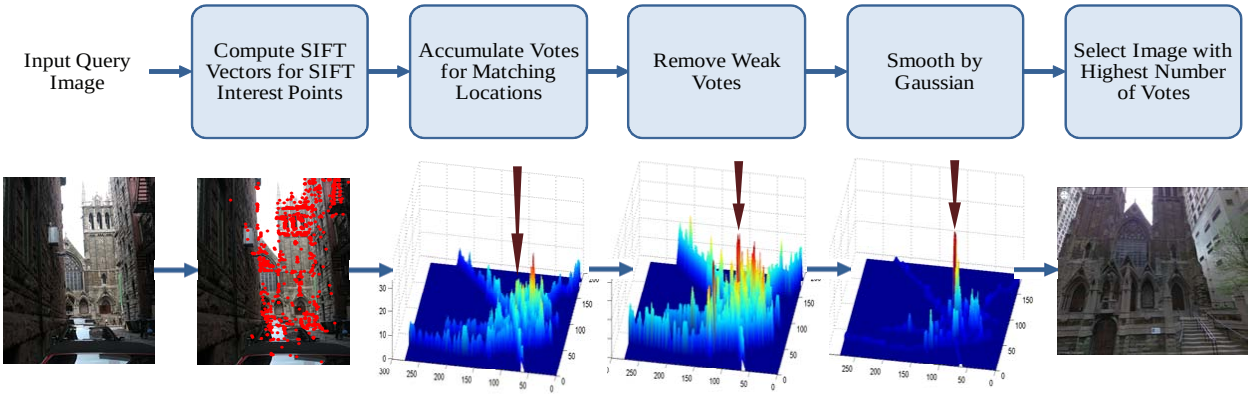

Figure 3.2: Block diagram of localization of a query image. Lower row shows the corresponding results of each step for the image. Note the streets in the vote plots, as the votes are shown over the actual map. The dark arrow points toward the ground truth location. The distance between the ground truth and matched location is 17.8m.

As noisy interest points are commonly detected in an image, a pruning step is essential. Lowe et al. [16] find reliable matches by setting a maximum threshold of 0.8 on the ratio of the distance between the query descriptor and the first and second nearest neighbors. For geo-location tasks in large-scale datasets, the pruning step becomes more important; this is primarily because many of the processed descriptors belong to non-permanent and uninformative objects (ie. vehicles, people, etc), or are detected on the ground plane - both cases where the descriptors become misleading for geo-localization purposes. The massive number of descriptors in the dataset

can add noisy, unauthenticated matches as well. Schindler et al. [14] find the more informative visual words by maximizing an information gain function, a process which requires reference images with significant overlap. Hakeem et al. [22] prune their dataset by setting the maximum SIFT threshold proposed in Lowe et al. [16] to 0.6 in order to keep more reliable matches. We propose using the following function in order to prune the matches:

$$V_{flag}(d_i) = \begin{cases} 1 & \frac{\|d_i - NN(d_i,1)\|}{\|d_i - NN(d_i,Min\{j\})\|} < 0.8 \\ & \forall j \rightarrow |Loc(NN(d_i,1)) - Loc(NN(d_i,j))| > D \\ 0 & otherwise \end{cases}, \tag{3.1}$$

where $V_{flag}(d_i)$ is the flag of the vote corresponding to the query descriptor d_i. If the flag is 0, the descriptor is removed in the pruning step; if the flag is 1, it participates in the voting. $NN(d_i, k)$ is the k_{th} nearest-neighbor of d_i. $Loc(NN(d_i, k))$ is the GPS location of the k_{th} nearest-neighbor to descriptor d_i and $|\ |$ represents the actual distance between the two GPS locations of the nearest neighbor. $\|\ \|$ represents Euclidean norm. At its core, equation 3.1 may appear to be the SIFT ratio [16]; the changes we have made mean that the descriptor in the denominator is dynamically determined, based on actual GPS distance. This is an important difference, as allowing this ratio to be determined dynamically creates a great advantage over the simple ratio between first and second nearest-neighbors used in Lowe et al. [16] and Hakeem et al. [22], in that it allows the localization task to handle repeated urban structures more accurately. The importance of this method becomes clearer by considering the reference images shown in Fig. 1.2. The windows of the skyscraper shown in the 3_{rd} column, 3_{rd} row of the figure are identical, leading to very close nearest-neighbor results for a query descriptor of this window (as shown in bottom left corner image in Fig. 3.3). While the SIFT ratio used in Lowe et al. [16] and Hakeem et al. [22] removes this descriptor in the pruning step, the proposed method retains it, as the location of all of

the very similar nearest neighbors are close to each other. In other words, even though we cannot necessarily determine which of the windows shown in the query image correspond to each of the windows in the skyscraper, they will still be voting for the correct location, as the GPS-tag of all these very similar nearest-neighbors point to one location. To explain it in a less-anecdotal way, equation 3.1 removes a descriptor only if the descriptor in the denominator does not belong to any of the nearby locations of the first nearest-neighbor AND the ratio is greater than 0.8. As can be seen in the 4_{th} column of Fig. 3.2, the votes around the ground truth location are mostly retained, whereas many of the incorrect votes are removed.

Since there is an overlap in the scene between the reference images, some of the objects in a query image may be in several of the reference images. To prevent the votes from being scattered between the overlapping reference images, we smooth the votes based on the order of their locations using this equation:

$$V_{smoothed}(\lambda', \phi') = \sum_{\lambda=-\infty}^{+\infty} \sum_{\phi=-\infty}^{+\infty} e^{-(\frac{\lambda^2+\phi^2}{2\sigma'^2})} V(\lambda' - \lambda, \phi' - \phi) V_{flag}(\lambda' - \lambda, \phi' - \phi) \ , \qquad (3.2)$$

where $V(\lambda, \phi)$ and $V_{flag}(\lambda, \phi)$ are the voting and flags function (respectively), for the GPS location specified by λ and ϕ, and the first coefficient is the 2D Gaussian function with a standard deviation of σ'. As each descriptor is associated with a GPS-tagged image, we can represent the voting function's parameter in terms of λ and ϕ. As can be seen in column 5 of Fig. 3.2, the smoothing step makes the peak which corresponds to the correct location more distinct.

As shown in the block diagram in Fig. 3.2, the location which corresponds to the highest peak is selected as the GPS location of the query image.

3.2.1 *Confidence of Localization*

There are several cases in which a query image may - quite simply - be impossible to localize. For instance, a query might come from an area outside of the region covered by the database; alternatively, the image might be so unclear or noisy that no meaningful geo-location information can be extracted from it. A parameter that can check for (and, consequently, prevent) these kind of positive errors is important. In probability theory, statistical moments have significant applications. The Kurtosis is a measure of whether a unimodal distribution is tall and slim or short and squat [75]. As we are interested in examining the behavior of the voting function in order to have a measure of reliability, we normalize it and consider it as a probability distribution function. Since the Kurtosis of a unimodal distribution can represent the peakedness of a distribution, we propose to use it as a measure of *Confidence of Localization*, since a tall and thin vote distribution with a distinct peak corresponds to a reliable decision for the location; correspondingly, a widely-spread one with a short peak represents a poor and unreliable localization. Our *Confidence of Localization* parameter is thus represented by the following equation:

$$CoL = Kurt(V_{smoothed}) = -3 + \frac{1}{\sigma^4} \sum_{\phi} \sum_{\lambda} [(\lambda - \mu_\lambda)^2 (\phi - \mu_\phi)^2] V_{smoothed}(\lambda, \phi) \ , \tag{3.3}$$

where $V_{smoothed}$ is the vote distribution function (see equation 2). The above equation is the Kurtosis of the 2D vote distribution function, with random variables λ and ϕ, corresponding to the GPS coordinates. μ_λ and μ_ϕ are expected values of λ and ϕ respectively. A high Kurtosis value represents a distribution with a clearer and more defined peak; in turn, this represents a higher confidence value.

However, Kurtosis can be interpreted as the peakedness of *unimodal* distributions only. For the distributions which are not necessarily unimodal, the inverse of Shannon entropy is the proper alternative for quantifying the confidence of localization. That is because a distribution with a high level of uncertainty (e.g. multiple strong peaks or uniformly distributed probabilities) has a

high entropy and also corresponds to an unreliable geo-localization instance. On the contrary, a distribution induced by the votes which are mainly concentrated around one location corresponds to a confident geo-localization, and also, has a low entropy. In Sec. 3.4.2, we empirically compare these two metrics and discuss the potential use of each.

3.3 Image Group Localization

Thus far, we discussed how a single image can be geo-localized by matching it against Street View images. However, there are many images which are incapable of being localized individually due to having a low resolution or lack of cues useful for localization (e.g. distinctive buildings). Many of these images are saved in albums which can act as cues for finding their exact location as images included in one album are typically captured at nearby locations. Therefore, we propose a novel hierarchical approach for localizing an *image group* which utilizes this proximity cue. The only assumption inherent in the proposed method is that all of the images in the group must have been taken within the radial distance R of each other; this radial distance R is a parameter that can be set in the method. In our approach, no information about the chronological history of the images is required.

To localize an image group consisting of images I_1 to I_N, we employ a hierarchical approach consisting of two steps:

• Step 1, Individual Localization of Each Image: In the first step of the approach, all of the images in the group are localized individually, independent from other images. In order to do this, we use the Single Image Localization method described previously in section 3; thus, each one of the single images in the group returns a GPS location.

• Step 2, Search in Limited Subsets: In the second step, N subsets of reference images which are within the distance R of each of the N GPS locations found in step 1 are constructed. Following that, a localization method - similar to the method defined in section 3 - is employed

for localizing the images in the group; however, in this case, the dataset searched is limited to each of the N subsets created by the initial search. We define the CoL value for each of the secondary, sequential search processes done in each of the limited subsets as:

$$CoL_{group}(S) = \sum_{i=1}^{N} \frac{CoL_i}{N} \ , \tag{3.4}$$

where S represents each of the secondary search processes. Once the CoL_{group} value for each of the limited subsets is calculated, the subset that scores the highest value is selected as the rough area of the image group. From there, each query image is assigned the GPS location of the match that was found in that limited subset.

Since this proposed approach to image group localization requires multiple searches in each step, the computational complexity of the method is of particular interest. The number of necessary calculations for localizing a single query image in our method is dependent on the number of detected interest points in the image. If we assume C is a typical number representing the number of required calculations for localizing an image individually, the number of required calculations to localize a group of images using the proposed approach is:

$$Complexity(N, \delta) = C(N + \frac{(N-1)N}{\delta}) \ , \tag{3.5}$$

where N is the number of images in the group and δ is a constant that is determined by the size of the limited subsets used in the step 2 of section 4. δ ranges from 1 to ∞, where 1 means each limited subset is as large as the whole dataset and ∞ means each subset is extremely small. Since the number of required calculations to localize an image individually is C, the number of required calculations to localize N images individually will be $N \times C$, so the percentage increase in computational complexity using the proposed group method vs. the individual localization method

is:

$$Complexity\ \ Increase(N, \delta) = \frac{Complexity(N, \delta) - N \times C}{N \times C} \times 100\ \ , \tag{3.6}$$

i.e.,

$$Complexity\ \ Increase(N, \delta) = \frac{N - 1}{\delta} \times 100\ \ , \tag{3.7}$$

For 4 and 50 - both typical values for N and δ, respectively - the increase in compuational complexity is 6%, garnering a roughly three-fold increase in system accuracy.

3.4 Experimental Results

Our test set consists of 521 query images. These images are all GPS-tagged, user-uploaded images downloaded from online photo-sharing web sites (Flickr, Panoramio, Picasa, etc.) for Pittsburgh, PA and Orlando, FL. Only indoor images, privacy-infringing images and irrelevant images (e.g. an image which only shows a bird in the sky), are manually removed from the test set. In order to ensure reliability of results, all the GPS-tags of the query images are manually checked and refined, as the user-tagged GPS locations are usually very noisy and inaccurate. Fig. 3.3 depicts some of the images.

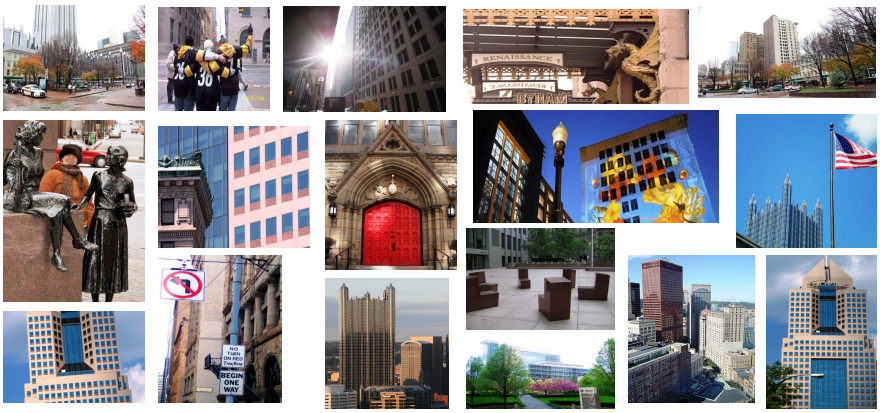

Figure 3.3: Sample query images in our test set.

311 images out of the 521 query images are used as the test set for the single-image localization method; 210 images are organized in 60 groups of 2,3,4 and 5 images with 15 groups for each as the test set for group image localization method.

3.4.1 Single Image Localization Results

Fig. 3.4 shows the results of the localization task for the test set of 311 images. In order to avoid computational issues of indexing the large number of images in a single tree, we construct 5 individual trees spanning the whole dataset. The final nearest-neighbor selected is chosen from among the 5 nearest-neighbor results retrieved across each tree. In these experiments, the queries and reference images of both of the cities are used. In order to make the curves in Fig. 3.4 invariant with respect to differing test sets, we randomly divide the single image localization method's test set into ten smaller test sets; likewise, we divide the group image localization method's test set into 5 smaller test sets. The curves in Fig. 3.4 are the average of the result curves generated for each of the smaller test sets.

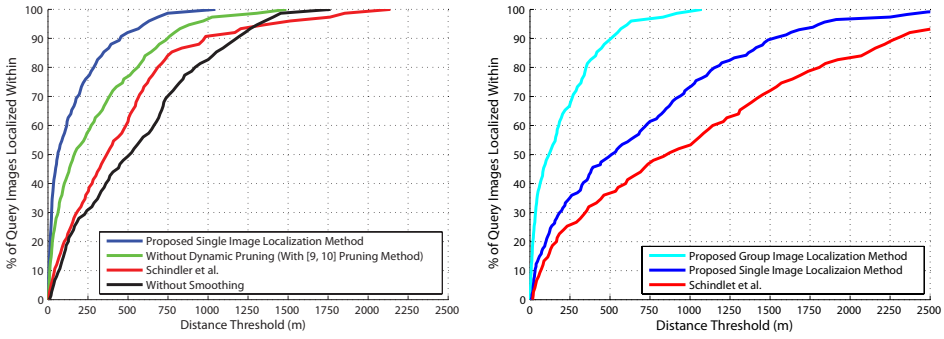

Figure 3.4: The left figure shows the single image localization method results vs. Schindler et al.'s method, along with the curves representing the effect of each step. The right figure shows the localization results using the proposed image group localization method.

As can be seen in Fig. 3.4, all of the steps proposed in Fig. 3.2 improve the accuracy significantly. The smoothing step unifies the votes, leading to a more distinct correct peak, while attenuating the incorrect votes. Dynamic pruning removes the wrong matches, bringing about a more accurate localization task; this enables us to calculate and save fewer SIFT descriptors per image. By comparison, we have (on average) 500 SIFT interest points per image; in Schindler et al. [14], the implementation used about 3000 interest points. As can be seen in Fig. 3.4, our method shows a significant improvement over the bag of visual words method used by Schindler et al. [14]. This is mostly due to the fact that, in the very similar and repeated structures of an urban area, the information lost in the quantization becomes critical. Additionally, the method proposed in Schindler et al. [14] requires reference images with significant overlap to maximize the information gain function, an assumption which can lead to significant issues in large scale localization. As can be seen in Fig. 3.4, about 60% of the test set is localized to within less than 100 meters of the ground truth; by comparison, this number for the method by Schindler et al. [14] is about 22%. However, our method fails when images are extremely cluttered with non-permanent objects (e.g. cars, people) or objects of low informative values (e.g. foliage).

3.4.2 Evaluation of the CoL (Confidence of Localization) parameter

In order to quantitatively examine the performance of the proposed Kurtosis-based CoL parameter, the relationship between the CoL values and the metric errors for the 311 query images is shown by the red curve in Fig. 3.5-left. The 311 queries are grouped into 8 bins based on the CoL values (horizontal axis), and the mean error of the bin members are shown on the vertical axis. As apparent in the figure, a higher CoL value typically corresponds to a lower error, meaning that the localization has been more reliable. Since the value of Kurtosis is not theoretically bounded, we normalize the CoL values to range from 0 to 1, as shown in the plot.

As discussed in Sec. 3.2.1, Kurtosis can be interpreted as a measure of the peakedness of *unimodal* distributions only. Therefore, employing Kurtosis as CoL is meaningful when the query

image shows a distinctive landmark or when the datasets does not include a great deal of repeated architectural features. Such cases normally results in a rather unimodal distribution for which the peakedness can be measured using Kurtosis.

The green curve in Fig. 3.5-left illustrates the results of the above experiment when the inverse of Shannon entropy is employed as the measure of confidence. Even though entropy does not have the unimodality requirement, it suffers from the shortcoming of ignoring the spatial location of the bins in the distribution. Hence, several close peaks in the distribution are treated equally as multiple distant peaks. This is undesirable for our purpose as these two cases clearly correspond to geo-localization tasks with different reliabilities while their entropy value can be identical. Therefore, entropy cannot precisely quantify the confidence of localization when the distribution is unimodal, while it is the proper measure to employ when the distribution is expected to be multimodal, e.g. when the query image does not show a landmark.

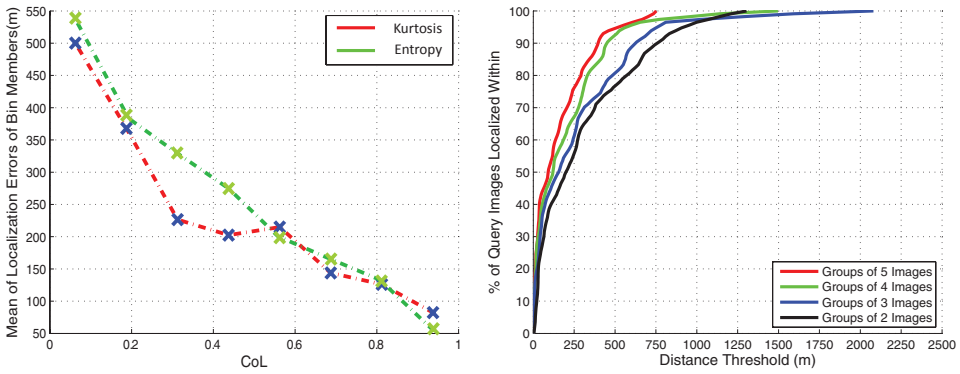

Figure 3.5: The left figure shows the relationship between the CoL values and the metric geo-localization error for the 311 query images. The CoL values are organized in 8 bins; the vertical axis shows the mean error value in meters for each bin. The right figure shows the breakdown of the results from the test set of the group image localization method based on the number of images in each group.

The curves in Fig. 3.5-left show that Kurtosis and entropy perform comparably in terms of the overall quantitative results on our dataset. This is consistent with the characteristics of our test set, as it includes images of both landmarks and visually indistinctive buildings. In general, if the dataset is primarily composed of landmarks and distinctive visual features, the Kurtosis can quantify the amount of peakedness of the mode, and therefore, is the better metric to employ. On the other hand, if the dataset mainly includes indistinctinve buildings and visually common features, the induced distributions are expected to be multimodal, and consequently, entropy is the proper measure to use.

3.4.3 Image Group Localization Results

Fig. 3.6 shows an example of localizing a set of images using the proposed method for geo-locating image groups. The image group has 3 images, which are depicted on the left-hand side of Column (a). As discussed in Section 4, the first step of the proposed method is localization of images individually, resulting in a GPS location for each image. Each query's individual localization is displayed on the map in Column (a). Column (b) shows the result of applying a search within the limited subset created by the initial search in step 1; the other two query images are localized around the initial points found in Column (a). Column (c) shows the voting surfaces for each query in each subset. As can be seen, Subset (2) has the most distinct peaks across all three queries; correspondingly, Subset (2) also has the highest CoL_{group} value and is thus selected as the correct set of matches. Finally, Column (d) shows an inset of the map corresponding to Subset (2) with the matched images represented by blue markers and the ground truth locations for the queries represented by green markers.

As discussed earlier, there are 210 images in our test set for group image localization. Most of the images were selected as they are (individually) very unclear and therefore challenging to localize; this was done in order to show how proximity information can be extremely helpful in localizing images that are incapable of being geo-located individually.

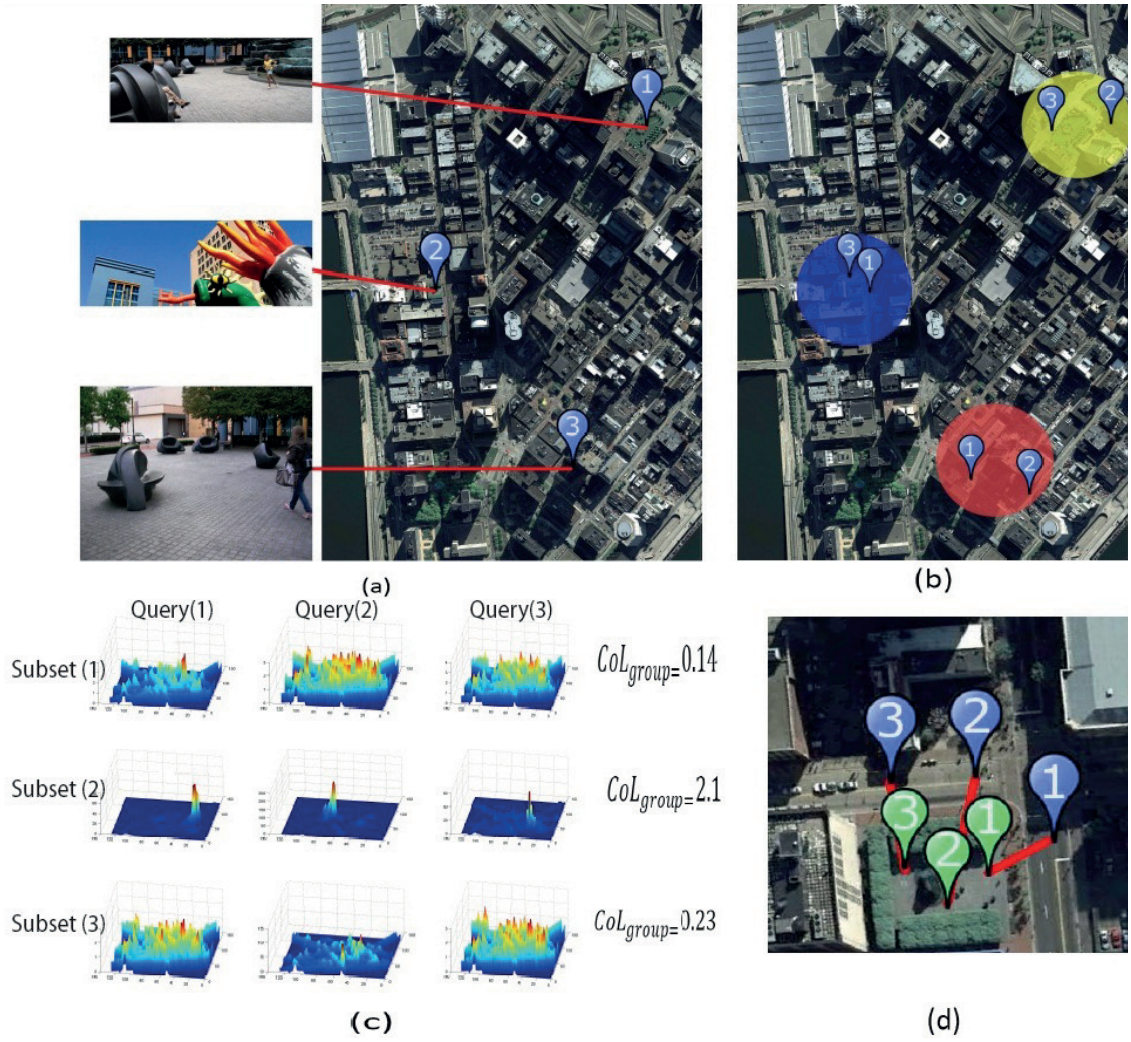

Figure 3.6: An Example of Image Group Localization. (a):Query Images and Single Localization Results (b): Results of Search in Limited Subset. Each colored region is a different limited subset (c): Voting Surfaces and CoL_{group} for each query in each subset. (d): Blue Markers: Matched locations in the specific limited subset. Green markers represent the corresponding ground truth of queries. The red lines connect the ground truth with the respective correct match. The distances between the ground truth and final matched location are 10.2m, 15.7m and 11.4m, for queries 1, 2, and 3 respectively.

We set the parameter R to 300 meters for our tests; this is a conservative assumption. This means that we assume that the images in one group are all taken within 300 meters of each other. The right column of Fig. 3.4, compares the performance of Schindler et al. [14]'s method, our proposed single image localization method, and the group image localization method. As can been seen, the use of proximity information results in a drastic improvement. The right plot in Fig. 6 shows the breakdown of the results of the test set from the group image localization method based on the number of images in the groups. As mentioned earlier, this set consists of groups of 2, 3, 4 and 5 images. As can be seen in Fig. 6, the accuracy of localization for groups with a larger number of images is greater, due to the fact that groups with a larger number of images will search more limited subsets. Consequently the chance of finding the correct location is higher.

3.5 Chapter Summary

In this chapter, we presented a method for finding the exact GPS-location of images. We leveraged a large-scale structured image dataset covering the whole 360° view captured automatically from Google Street View. First, we indexed the SIFT descriptors of the reference images in a tree; said tree is later queried by the SIFT descriptors of a query image in order to find each individual query descriptor's nearest neighbor. We developed a geo-spatial pruning method which employed GPS locations to remove unreliable query descriptors if many similar reference descriptors exist in disparate areas. The surviving descriptors vote for the location of their parent image. The vote distribution function was then smoothed, and the location with the highest number of votes was picked to be the location of the query image. The reliability of the geo-location was represented by a parameter called CoL, which was based on the Kurtosis of the vote distribution. Finally, a novel approach - using the proximity information of images - was proposed in order to localize groups of images; first, each image in the image group was localized individually, followed by the localization of the rest of the images in the group within the neighborhood of the

found location. Later, the location of each image within the rough area with the highest CoL_{group} value was selected as the exact location of each image.

In the next chapter, we argue that local features suffer from an inherent ambiguity as a results of having a limited score. We will demonstrate that leveraging multiple nearest neighbors per query features as well as simultaneous utilization of local and global features can address this shortcoming in the context of geo-localization.

CHAPTER 4: IMAGE GEO-LOCALIZATION BASED ON MULTIPLE-NN FEATURE MATCHING USING GENERALIZED GRAPHS

Local features, e.g. SIFT [16], have been widely used in the Computer Vision systems primarily due to being remarkably view invariant and robustness to partial occlusion. By the same token, the majority of existing location recognition methods are built upon using local features for identifying the similarities between the query image and the reference dataset. However, limiting the scope of a feature to a local patch inherently makes the correspondences established based on such features prone to mismatches. This is mainly due to the reason that small segments of images are often literally indistinguishable when the global context is ignored.

In order to address this issue, we introduce a new framework for geo-locating an image based on a novel multiple nearest neighbor feature matching method using Generalized Minimum Clique Graphs (GMCP). First, we extract local features (e.g., SIFT) from the query image and retrieve a number of nearest neighbors for each query feature from the reference data set. Next, we apply our GMCP-based feature matching to select a single nearest neighbor for each query feature such that all matches are globally consistent. Our approach to feature matching is based on the proposition that the first nearest neighbors are not necessarily the best choices for finding correspondences in image matching. Therefore, the proposed method considers multiple reference nearest neighbors as potential matches and selects the correct ones by enforcing the consistency among their global features (e.g., GIST) using GMCP. GMCP is useful in situations where there are multiple potential solutions for a number of subproblems, as well as a global criterion among the subproblems to satisfy. In the context of our problem, each subproblem is matching a query feature to the reference dataset, the potential solutions are the NNs, and the global criterion is the consistency of global features of the NNs. Therefore, we utilize GMCP in performing our multiple

nearest neighbor feature matching, and a voting scheme on the matched features is employed to identify the strongly matching reference image(s) and estimate the geo-location. In this context, we argue that using a robust distance function for finding the similarity between the global features is essential for the cases where the query matches multiple reference images with dissimilar global features. Towards this end, we propose a robust distance function based on the Gaussian Radial Basis Function (G-RBF).

4.1 Approach

We preprocess the reference dataset (Street View) by computing a set of local features (in our implementation SIFT) from each image. We aggregate these features of all the reference images and organize them in a k-means tree [73]. We refer to the extracted local features, their corresponding reference images, and the built tree as *reference features*, *parent image*s, and *reference tree*, respectively. Additionally, we find a global feature, e.g. color histogram or GPS location, for each reference image.

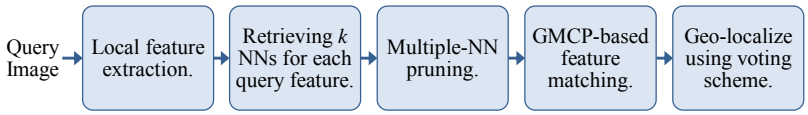

Figure 4.1: Block diagram of the proposed Image Geo-localization Method.

The block diagram of our framework for geo-locating a query image is shown in Fig. 6.1. First, we extract local features from the query image; we refer to them as *query features*. We search the reference tree using the query features and retrieve k nearest neighbors (NN) for each query feature. Next, we apply our multiple-nearest neighbor pruning (Sec. 4.1.1) to coarsely remove the query features which do not have distinctive NNs. In the next step, we employ a robust function

for computing the distance between global features which is particularly essential when multiple reference images with dissimilar global features match the query image (Sec. 4.1.2.2). Unlike the traditional feature matching methods, such as the k nearest neighbor classifier which performs voting using all of the k NNs, we consider the k NNs as potential matches for their query feature and identify the correct one using the GMCP-based feature matching method (Sec. 4.1.2). Lastly, a voting scheme on the matched features is employed to find the reference image which most strongly matches the query. We use the location of the strongest match as an estimation of the location of the query image (Sec. 4.1.3).

Detailed explanation of each step is provided in the rest of this section.

4.1.1 Multiple-Nearest Neighbor Pruning

Let M be the number of local features detected in the query image. Let v_n^j denote the n^{th} NN for the j^{th} query feature where $n \in \mathbb{N} : 1 \leq n \leq k$ and $j \in \mathbb{N} : 1 \leq j \leq M$.

Many of the interest points found in the query image, such as the ones detected on foliage, ground or moving objects, do not convey any useful information for geo-localization purposes. It will be helpful if such features can be coarsely identified and removed prior to performing the feature matching. For this purpose, we utilize the following pruning method which is based on examining how distinctive the first and $(k+1)^{th}$ NN are:

$$\begin{cases} \text{remove } q^i, & \text{if } \frac{\|q^i - \zeta(v_1^i)\|}{\|q^i - \zeta(v_{k+1}{}^i)\|} > 0.8 \\ \\ \text{retain } q^i, & \text{otherwise,} \end{cases} \tag{4.1}$$

where $\zeta(.)$ represents an operator which returns the local feature descriptor of the argument node. q^i is defined as the local descriptor of the i^{th} query feature, and $\|.\|$ represents the distance between

the features. Equation (4.1) states the i^{th} query feature should be pruned if its first and $(k+1)^{th}$ NNs are more than 80% similar. This formulation is consistent with our multiple-NN feature matching scenario as we assume the correct match is among the top k NNs. Therefore, we disregard the top k NNs and compare the first NN to the $(k+1)^{th}$ one. If the first NN is not distinctive, even compared to the $(k+1)^{th}$ NN, the corresponding query feature is pruned since it is expected to be uninformative. The threshold value of 0.8 is empirically found to be optimal for comparing SIFT features by Lowe [16]. Note that the difference between Lowe's criteria [16] and the criteria in equation (4.1) is that we utilize multiple NNs instead of using the first two only, which makes our pruning consistent with our multi-NN formulation.

4.1.2 Feature Matching Using Generalized Minimum Clique Graph

Let L be the number of local features surviving the pruning step. We define the graph $G = (\mathbf{V}, E, \varpi, w)$, where $\mathbf{V}$, E, ϖ and w denote the set of nodes, edges, node costs and edge weights, respectively. The set of nodes, $\mathbf{V}$, is divided into L disjoint clusters. Each query feature point is represented by one cluster, and the nodes therein represent the corresponding k nearest neighbors. C_i, where $i \in \mathbb{N} : 1 \leq i \leq L$, denotes the i^{th} query feature ($\equiv$cluster), and v_m^i denotes the m^{th} candidate ($\equiv$node) for the i^{th} query feature. The edges are defined as $E = \{(v_m^i, v_n^j) | i \neq j\}$ which signifies all the nodes in G are connected as long as they do not belong to the same cluster. We define the node cost, $\varpi : \mathbf{V} \rightarrow \mathbb{R}^+$, as:

$$\varpi(v_m^i) = \|q^i - \zeta(v_m^i)\|. \tag{4.2}$$

The node cost specifies how similar the local features of v_m^i and its corresponding query features are. Edge weight, $w : E \rightarrow \mathbb{R}^+$, is defined as:

$$w(v_m^i, v_n^j) = \|\rho(v_m^i) - \rho(v_n^j)\|, \tag{4.3}$$

where $\rho(.)$ represents an operator which returns the global descriptor of the parent image of the argument node. The edge weight, $w(v_m^i, v_n^j)$, is a measure of similarity between the nodes v_m^i and v_n^j in terms of the global features of their parent images. Low values for an edge weight and its node costs signify a high global consistency between corresponding nodes and vice versa. $\|.\|$ in equations (4.2) and (4.3) denotes the distance between the local and global features of the argument nodes, respectively; however, the type of distances employed in these two equations do not have to be the same.

The task of matching the query features to the reference features requires identifying the correct NN for each one. Therefore, a feasible solution to this problem can be represented by a subgraph of G in which one node (≡NN) is selected from each cluster(≡set of nominated NNs for one query feature). Such a subgraph, $G_s = (\mathbf{V}_s, E_s, \varpi_s, w_s)$, consists of a set of nodes with the general form $\mathbf{V}_s = \{v_a^1, v_b^2, v_c^3, ...\}$ which indicates the a^{th} node from 1^{st} cluster, b^{th} one from 2^{nd} cluster, and so on are selected to be included in $\mathbf{V}_s$. By definition, $E_s = \{E(p,q)|p,q \in \mathbf{V}_s\}$, $\varpi_s = \{\varpi(p)|p \in \mathbf{V}_s\}$, and $w_s = \{w(p,q)|p,q \in \mathbf{V}_s\}$. We use $\mathbf{V}_s$ to denote a feasible solution hereafter since the set of nodes $\mathbf{V}_s$ is essentially enough to form G_s. The cost of the feasible solution $\mathbf{V}_s$ is defined as:

$$C(\mathbf{V}_s) = \frac{1}{2}\sum_{i=1}^{L}\sum_{\substack{j=1,\\ j\neq i}}^{L}\left(\frac{1}{2}\alpha\overbrace{\Big(\varpi(\mathbf{V}_s(i)) + \varpi(\mathbf{V}_s(j))\Big)}^{\text{local features}} + (1-\alpha)\underbrace{w(\mathbf{V}_s(i), \mathbf{V}_s(j))}_{\text{global features}}\right), \tag{4.4}$$

which is the cost of the complete graph induced by the nodes in $\mathbf{V}_s$. $\mathbf{V}_s(i)$ denotes the i^{th} element of $\mathbf{V}_s$, and α is the mixture constant that ranges between 0 and 1 and balances the contribution of local and global features. A larger α corresponds to more contribution from local features in the overall cost and vice versa; $\alpha = 0.5$ corresponds to equal contributions from both features. Equation (4.4) (note the constants) is defined in a way that the number of summed terms corresponding to the nodes and edges are always equal; hence, the balance between the contribution of local and global

features to the cost does not change with L.

Equation (4.4) assigns a cost to a feasible solution utilizing both local and global features; this is done by incorporating the agreement between the global features of the parent images of reference features. Therefore, the potential wrong matches resulting from the limited scope of local features are minimized. By finding the feasible solution with the minimal cost, i.e. $\arg\min_{\mathbf{V}_s} C(\mathbf{V}_s)$, the subset of NNs with the highest agreement is found. In the following subsection, we explain that the definition of Generalized Minimum Clique Graph ideally fits the formulation of our problem and can be used for solving the aforementioned optimization task.

4.1.2.1 Generalized Minimum Clique Problem

Generalized Graphs, also known as Generalized Network Design Problems [19], are a category of graph theory problems which are based on generalizing the standard subgraph problems. The generalization is done by extending the definition of a node to a cluster of nodes. For example, in the standard Traveling Salesman Problem (TSP) the objective is to find the minimal cycle which visits all the nodes exactly once. In the *Generalized* Traveling Salesman Problem, the nodes of the input graph are grouped into disjoint clusters; the objective is to find the minimal cycle which connects all the clusters while exactly one node from each is visited [19].

Similarly, in the Generalized Minimum Clique Problem (GMCP) the vertices of the input graph are grouped into disjoint clusters. As shown in Fig. 4.2, the objective is to find a subset of the nodes that includes exactly one node from each cluster while the cost of the complete graph that the subset forms is minimized [19]. A similar formulation is utilized to solve the Frequency Assignment Problem [76] in telecommunications. It has been utilized for maximizing the number of comparisons between human detections in different video frames in order to perform data association as well [6].

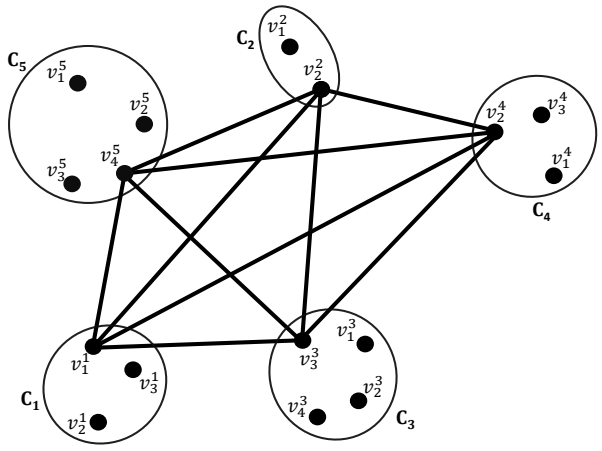

Figure 4.2: An example GMCP. A feasible solution is shown where one node from each cluster is selected. The complete subgraph, G_s, which the selected nodes form is shown using the edges.

The input to GMCP with vertex cost is defined as the graph $G = (\mathbf{V}, E, \varpi, w)$ where $\mathbf{V}$, E, ϖ and w represent the set of nodes, edges, node costs and edge weights. $\mathbf{V}$ is divided into L disjoint clusters, i.e. $\mathbf{C}_i \cap \mathbf{C}_j = \emptyset$ $(1 \leq i \neq j \leq L)$ and $\mathbf{C}_1 \cup \mathbf{C}_2 \cup \ldots \cup \mathbf{C}_f = \mathbf{V}$. A feasible solution of GMCP is denoted by a subgraph $G_s = (\mathbf{V}_s, E_s, \varpi_s, w_s)$, where $\mathbf{V}_s$ and ϖ_s denote a subset of $\mathbf{V}$ which includes only one node from each cluster and its corresponding node costs, respectively. E_s and w_s are the subset of E which V_s induces and the corresponding edge weights. The cost of a feasible solution is defined as the sum of all the edge weights and node costs along the solution subgraph. Note that the subgraph G_s is complete, making any feasible solution of GMCP a clique.

As can be inferred from the formulation of our multiple NN feature matching problem, GMCP can be essentially used for solving the same optimization problem. Therefore, by solving GMCP for the graph G, the optimal solution which has the most agreement in terms of global and local features will be found via:

$$\hat{\mathbf{V}}_s = \arg\min_{\mathbf{V}_s} \; C(\mathbf{V}_s). \tag{4.5}$$

Note that our GMCP-based method differs from basic graphical models in several aspects. Our input graph and feasible solutions are complete as we consider the relationships among all possible pairs of local features. This makes our formulation different from the graphical models which have specific assumptions on the structure of the graph, e.g. being acyclic. Additionally, a graphical model equivalent to our input graph would include a large number of loops as it would have to be complete; the condition under which the inference methods similar to belief propagation [77] converge for graphs with loops is still unknown [78]. On the contrary, we employ a combinatorial approach to solving our optimization problem whose performance does not deteriorate by including loops in the input graph. The details of how to solve GMCP are discussed in Sec. 4.2.

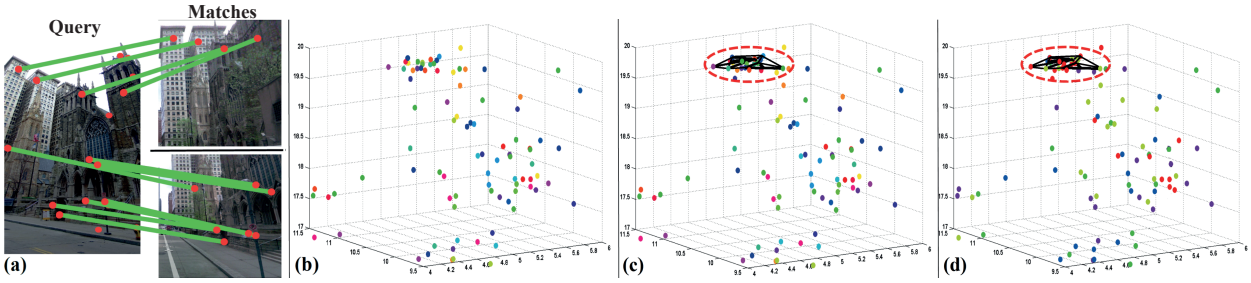

Figure 4.3: Feature matching using GMCP; (a): A query image and matched reference images are shown on the left and right, respectively. Found correspondences are shown by the green lines. (b): All the nodes in $\mathbf{V}$ are shown in 3-dimensional global feature space. Each node represents one NN while the color coding indicates cluster membership. (c): Same as (b) while the black lines indicate GMCP edges. (d): Same as (c) while the color coding shows the rank of the nearest neighbor. red=1^{st}, yellow=2^{nd}, green=3^{rd}, blue=4^{th}, magenta=5^{th}.

Fig. 4.3 demonstrates the process of feature matching using GMCP.[1] (a) shows a query image and two best matching reference images on the left and right, respectively. Discovered

[1]In figures 4.3, 4.4 and 4.6 the nodes which are located exactly on the same spot in the global feature space, i.e. belong to the same reference image, are shown slightly apart in order to demonstrate the density properly.

correspondences by GMCP are shown in green. (b) shows all the nodes in $\mathbf{V}$ in the global feature space. In this example, a 60-dimensional RGB color histogram is used as the global feature, and the dimensionality is reduced to 3 using PCA for illustration purposes. Membership to GMCP clusters, $\mathbf{C}_i$, is color coded meaning each color represents one value of i. (c) shows the subset of nodes included in $\hat{\mathbf{V}}_s$ depicted by the red contour. The black edges are the ones included in the subgraph $\hat{G}_s$. (d) illustrates the same plot as (c) except that the color coding represents the rank of each node when retrieved based on the local features. Red, yellow, green, blue and magenta represent first, second, third, fourth and fifth, respectively. A considerable number of the nodes included in $\hat{\mathbf{V}}_s$ are not marked in red which signifies the nodes with the most consistent global features are not necessarily the first NNs. Also, it is apparent in (c) and (d) that the selected nodes belong to a tight area in the feature space which indicates they share similar global features.

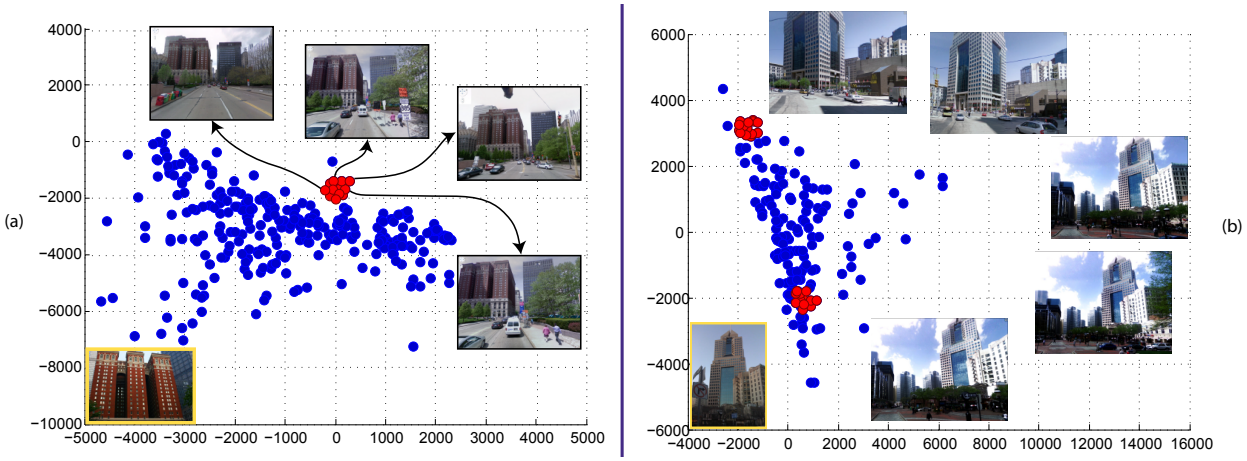

Figure 4.4: (a),(b): All the nodes in $\mathbf{V}$ shown in 2-dimensional global feature space for two sample queries. Outlier and inlier NNs are illustrated in blue and red, respectively. The query images are shown with the yellow border. A subset of the matching reference images are shown linked to their corresponding nodes. (a) A case with one group of matching reference images. (b): A case with two groups of matching reference images with dissimilar global features.

Another example is demonstrated in Fig. 4.4 (a). The utilized global feature is a 60-dimensional RGB color histogram reduced to 2 dimensions using PCA for illustration. Each node in (a) represents one NN included in $\mathbf{V}$ shown in the global feature space. The inlier and outlier NNs are shown in red and blue, respectively.[2] The inlier NNs are the ones which belong to one of the reference images that actually matches the query image, and the outliers are the ones which do not belong to any of the matching reference images. As apparent in the figure, the global features of all the inlier NNs are similar as they are adjacent in the global feature space.

However, we commonly observe cases where the global features of matching reference images are not similar and consequently form disjoint groups in the global feature space. This dissimilarity is mainly due to variations, such as different imaging conditions or diverse camera poses, to which most of the existing global features are not invariant. One case is shown in Fig. 4.4 (b) where the building in the query image is visible from two distinct locations, and the reference dataset includes images taken at both of these locations. The difference in viewpoint between the query and the first and second groups of matching images is less than 30 degrees which causes the local feature to nominate NNs from both groups of images. However, the disagreement in viewpoint and imaging conditions will cause the images in the two different groups to have dissimilar global features. Thus, the inlier NNs are observed in two disjoint groups in the global feature space as shown in Fig. 4.4 (b).

The method explained earlier in this section was based upon the assumption that the global features of all of the inlier NNs should be similar, i.e. they should form one joint group of inliers. In Sec. 4.1.2.2 we argue that the GMCP-based method fails to identify all of the inliers when multiple disjoint groups exist. We address this issue by using a robust distance function for the global features.

[2]The same applies to Fig. 4.6 and Fig. 9.1 of the appendix.

4.1.2.2 *Robustification of the Global Features' Distance Function*

The existence of disjoint groups prevents the GMCP-based method from identifying all of the inliers (refer to the appendix for the formal proof for a relaxed case). That is because a feasible solution which includes inlier nodes from several disjoint groups will have a high cost due to the considerable distance between these groups. We solve this problem by using the following robustified metric for computing the distance between the global features:

$$D(\mathbf{x}, \mathbf{y}) = \sqrt{2 - 2e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}}. \tag{4.6}$$

$\|\mathbf{x} - \mathbf{y}\|$ and $D(\mathbf{x}, \mathbf{y})$ denote the original (e.g. Euclidean) and robustified distance between the two vectors $\mathbf{x}$ and $\mathbf{y}$, respectively.

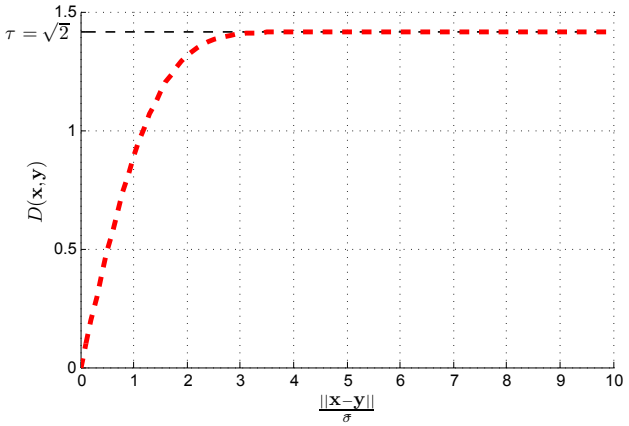

Figure 4.5: The robust distance function D. It has the characteristic of damping the large values and boosting the short ones.

The function in equation (4.6) is plotted in Fig. 4.5. The characteristic of the distance function D is boosting the short distances and damping the large ones. The distances that are significantly larger than σ will be mapped to the constant value τ ($= \sqrt{2}$). In the context of our

problem, it means the distances will contribute to the cost functions equally if they are significantly larger than a certain value which is determined by σ. This trait causes the intra-group distances to matter more than inter-group ones after robustification; this enables the optimization function of equation (4.5) to find the tight groups of global features rather than getting bewildered by the excessive cost the outliers contribute (refer to the Sec. 9.1 of the appendix for the proof for a relaxed case which shows by using a the robust distance function, the inliers from all disjoint groups will be included in the GMCP solution).

Our robustification can be viewed as finding the distances between the global features in a space transformed using a Gaussian Radial Basis Function (G-RBF) kernel. This is because G-RBF kernel is defined as $k_G(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$. Assuming the employed norm is ℓ_2, the distance between two vectors in a projected space transformed using an arbitrary kernel k equals $\sqrt{k(\mathbf{x}, \mathbf{x}) + k(\mathbf{y}, \mathbf{y}) - 2k(\mathbf{x}, \mathbf{y})}$ [79]; plugging G-RBF kernel in this function yields equation (4.6).

In the experiments section, we will empirically show that the distance function defined in equation (4.6) yields the best results compared to ℓ_1 distance, Squared, Linear, and Huber loss functions. However, any function which has a form similar to the curve shown in Fig. 4.5 is expected to give similar robustification effect.

To summarize, we form the input to GMCP using the distance function D for finding the distances between global features; this is done by using the equation (4.6) in the equation (4.3):

$$w_D(v_m^i, v_n^j) = \sqrt{2 - 2e^{-\frac{\|\rho(v_m^i) - \rho(v_n^j)\|^2}{2\sigma^2}}}, \tag{4.7}$$

which provides the edge weights robustified by the metric D. The value of σ is set in a way that the distance between two inlier nodes in one group is unlikely to be significantly larger than σ. Additionally, a distance substantially larger than σ should be likely to involve either an outlier node or two groups of disjoint inliers. Therefore, σ should be set to the expected distance between

the global features of two images of the same scene. [3]. It is a fixed value determined based on the type of the global feature and does not need to be tuned for each query.

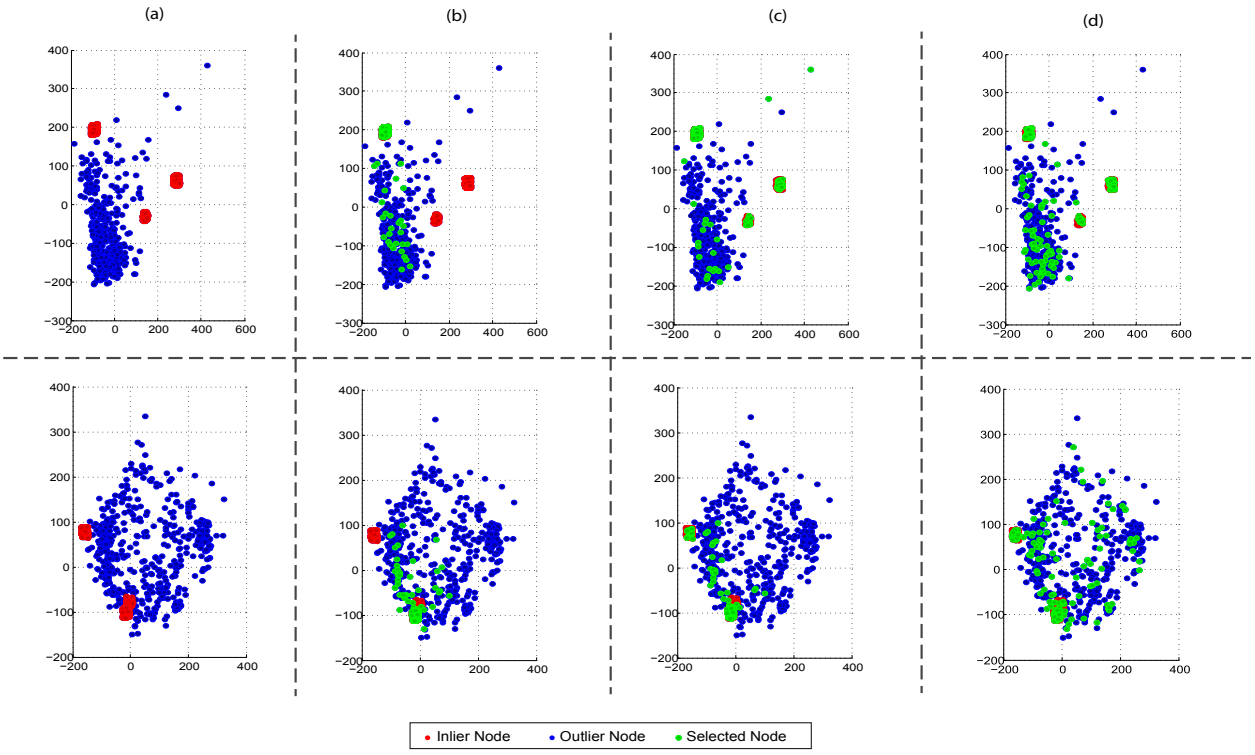

Figure 4.6: The results of robustification. Upper and lower columns show sample cases with three and two disjoint inlier groups. In (a), the outlier and inlier nodes of $\mathbf{V}$ are shown in blue and red, respectively. In (c) and (b), the green nodes show the ones selected by GMCP with and without robustification, respectively. Note that there are some outliers included in $\hat{\mathbf{V}}_s$ which typically correspond to the query features without any inlier NN. (d) shows the selected nodes by GMST. Color histogram was employed as the global feature.

Fig. 4.6 shows the impact of robustification for two sample cases. Upper and lower

[3] i.e. r_1 in Fig. 9.1 of the appendix.

columns show examples with three and two disjoint groups of matching reference images. (a) shows all the nodes in $\mathbf{V}$ where inliers and outliers are shown in red and blue, respectively. Green nodes in (b) represent the selected nodes by GMCP, i.e $\hat{\mathbf{V}}_s$, without robustification (using ℓ_2 distance) which indicates it failed to identify all the inliers from different groups. (c) illustrates the results using D; it signifies that the robustification enables GMCP to include the inliers from all of the disjoint groups.

Generalized Minimum Spanning Tree: One potential way of dealing with the problem of disjoint groups of matching images is leveraging a linkage mechanism in feature matching. Linkage based methods, such as single-linkage clustering [80], are commonly built on the following general definition: *the distance between two groups of entities is defined to be the distance between the two closest elements in the two groups.* Therefore, only one member of each group, and not all of them, are used for computing the similarity. The linkage-based dual of GMCP is GMST (Generalized Minimum Spanning Tree) [81]; the only difference between their definitions is that the cost of the feasible solution $\mathbf{V}_s$ is defined as the cost of the Minimum Spanning Tree found on its nodes rather than the cost of the complete graph it forms: $C_{_{MST}}(\mathbf{V}_s) = MST(\mathbf{V}_s)$. GMST can potentially deal with disjoint groups because of its linkage mechanism: consider the exemplified case in Fig. 4.4 (b). In order to link the two groups in GMST, a single edge between two nodes from the two groups would suffice. Therefore, all of the red nodes of both groups can be included in $\hat{\mathbf{V}}_s$ at the cost of a single long edge which does not add a considerable value to the overall cost of the solution. This is dissimilar to GMCP where all the nodes in the two different groups have to be connected pairwise and consequently would induce an excessive cost. Therefore, GMST is capable of dealing with the issue of disjoint inlier NNs without the need to a robust distance function. Fig. 4.6 (d) shows the selected nodes by GMST which demonstrates that inliers from all groups are included.

However, we use GMST as a baseline in our experiments as we will show that GMCP with the robust distance function outperforms GMST due to an issue known as *chaining phenomenon*

in the linkage-based clustering literature [82, 80]; this phenomenon makes linkage-based methods essentially prone to outliers and noise. In the context of our problem, Chaining Phenomenon occurs when an outlier, which is distant to the majority of inlier nodes, is included in the selected subgraph merely due to being in the proximity of a single inlier node. That way, the outlier will be incorrectly included in the solution since the linkage mechanism of GMST considers only the *shortest* distance between a selected node and the remaining ones, and not all of the distances. Reference [82] provides an in-depth explanation of the chaining phenomenon.

4.1.3 Location Estimation Using the Matched Feature Points

The benefits of using the GMCP-based method for feature matching is twofold: First, it matches the query features to the top few matching reference images; this trait causes the query features which are typically assigned to incorrect reference images using only the 1^{st} NN to be matched to the top few reference images. Second, the algorithm favors to assign the majority of query features to the strongest match among the top few discovered reference images; this is because the distance between the global features of the NNs belonging to the same image is zero, while the distance between the global features of two different matching reference images is non-zero, even though small. These two characteristics cause the strongest matching image found by GMCP to be more accurate compared to the one found by the baselines. Therefore, we estimate the location of the query using a winner-take-all scenario in which the reference image found by GMCP to have the highest number of matched feature points is selected as the strongest match, and its location is identified as an approximation of the location of the query image. If the reference dataset is a dense sampling of the covered area, which is typically the case for our dataset, the location of the strongest match is expected to be within a few tens of meters of the ground truth which is generally acceptable for a city scale localization.

In addition, the matched feature points by GMCP could be used for performing further reasoning about the camera location of the query. For instance, the top few reference images with a

number of matched features beyond a certain threshold can be identified. Then, the camera matrix of the query image, which includes its geo-location, can be computed using the feature correspondences found by GMCP utilizing epipolar geometry-based techniques. In the experiments section, we will show that the simple yet effective winner-take-all scheme yields satisfactory results for geo-localization at a city scale. However, the epipolar geometry-based methods using the feature correspondences found by GMCP are useful for finer localization.

Figure 4.7: **Left**: Forty sample street view images belonging to eight place marks of the reference dataset. **Right**: Sixteen sample user uploaded images from the test set.

4.2 Solving GMCP

GMCP is an $\mathcal{NP}$-hard problem [19]. A few approaches to solving GMCP such as branch-and-cut and multi-greedy heuristics [83, 76] have been explored to date; however, the majority of them are formulated according to particular problems and are suited for small inputs [19]. Our graphs typically include $L \times k = 300$ to 1500 nodes which require an efficient and approximate solver as the problem is $\mathcal{NP}$-hard. Therefore, we employ Local-neighborhood Search to solve the optimization problem of equation (4.5) as it has been shown to work efficiently for large combinatorial problems such as Tabu-search for GMST [84, 85].

Local-neighborhood search methods are based on examining the neighbors of the current

solution in hope of discovering a better one. Two solutions are neighbors of size ε if they are identical except in ε elements. Choosing a small neighborhood size makes the optimization process prone to getting stuck at suboptimal regions. On the other hand, choosing a large neighborhood significantly enlarges the number of neighbors in each iteration, resulting in an increase in the complexity. In order to deal with this issues, we use a different approach in which we change the neighborhood size from 1 to δ repeatedly in each iteration.

Algorithm 1 Local Neighborhood Search GMCP Solver

Initialize the best solution, $\hat{\mathbf{V}}_s$, with a random solution.
while termination conditions not satisfied **do**
  $\mathbf{N}_{size-1} \leftarrow$ size-1 neighbors of $\hat{\mathbf{V}}_s$.
  $\mathbf{\Gamma}_{size-1} \leftarrow \delta$ neighbors in $\mathbf{N}_{size-1}$ with the lowest costs.
  $\mathbf{\Lambda} \leftarrow$ the elements changed in $\mathbf{\Gamma}_{size-1}$.
  $\mathbf{N}_{size-\delta} \leftarrow$ size-1 to size-δ neighbors of $\hat{\mathbf{V}}_s$; only the elements in $\mathbf{\Lambda}$ are allowed to change.
  $\hat{\mathbf{N}} \leftarrow$ the solution with the lowest cost in $(\mathbf{N}_{size-1} \cup \mathbf{N}_{size-\delta})$.
  if (cost of $\hat{\mathbf{V}}_s$) $\geq$ (cost of $\hat{\mathbf{N}}$) **then**
    $\hat{\mathbf{V}}_s \leftarrow \hat{\mathbf{N}}$
  else
    return $\hat{\mathbf{V}}_s$ as the found solution.
  end if
end while
return $\hat{\mathbf{V}}_s$ as the found solution.

The details of our solver are provided in Algorithm 1. First, the solver is initialized with a random solution. We fix the neighborhood size to 1 and identify the δ best solutions. Next, we fix the neighborhood size to δ while we allow only the elements replaced in the top δ neighbors of size-1 to change and find the resulting neighbors. If any of the size-1 to size-δ neighbors induce a cost lower than the best known solution, the best solution is updated. This procedure continues iteratively until the minimum is found or a termination condition (maximum time or maximum number of iterations) is met. Our algorithm allows up to δ elements to change in one iteration, yet

we do not need to examine all of the feasible neighbors of size δ; this can accelerate the process up to δ times where $L \gg k^{\delta-1}$.

In order to investigate the optimality of our solver, we tested it on a set of 1000 GMCP instances with L (ranging from 4 to 15) clusters and k (ranging from 3 to 8) nodes in each cluster. We found the optimal GMCP solution of these instances using exhaustive search and compared them against the solution found by the proposed solver; in 79% of the instances, our approximate solver converged to the optimal answer. Additionally, in the majority of the rest of the cases, the solution found by our solver was less than 30% different from the optimal answer.

The time complexity of the proposed solver is $\mathcal{O}(kL^2 + k^\delta L)$, assuming a fixed number of iterations. This is because there are $(K-1)L$ feasible solutions in the first step of the algorithm (i.e. size-1 neighbors) to verify, while the time complexity of calculating the cost of one clique is[4] $\mathcal{O}(L)$. Hence, the overall complexity of the first step of one iteration becomes $\mathcal{O}((K-1)L^2) = \mathcal{O}(KL^2)$. The second step of one iteration includes k^δ feasible solutions (i.e. size-δ neighbors) to verify which makes the time complexity of the second step $\mathcal{O}(k^\delta L)$. Therefore, the overall time complexity of these cascade steps is $\mathcal{O}(KL^2 + k^\delta L)$.

Therefore, the proposed solver has a *polynomial* time complexity with respect to the number of nodes in one cluster (k) and number of clusters (L) when the other variable is kept constant. Compared to solving GMCP using exhaustive search which would have the time complexity of $\mathcal{O}(k^L L)$, the proposed solver has a significantly better running time since $L \gg \delta$. However, it is not guaranteed to converge to the optimal solution if the space is non-convex.

Employing the proposed optimization method with $\delta = 3$, we can solve a typical GMCP instance of our feature matching problem in less than one second on average, using non-optimized MATLAB code on an octo-core 2.4GHz machine.

[4]the time complexity of summing and subtracting $L-1$ numbers from the so-far-best cost.

4.3 Experimental Results

In this section, we provide the details of our evaluation dataset and present our extensive experimental results for geo-localization and feature matching.

4.3.1 Evaluation Dataset

We evaluated the proposed algorithm using a reference dataset of over 102,000 Google Street View images. The dataset covers downtown and the neighboring areas of Pittsburgh, PA; Orlando, FL and partially Manhattan, NY. Fig. 4.7-left shows forty sample reference images belonging to eight place marks. The place marks are approximately $12m$ apart, and the dataset covers about $204km$ of urban streets. The 360° view of each place mark is broken down into four side views and one top view image. The test set consists of 644 unconstrained user uploaded images downloaded from Flickr, Panoramio and Picasa which are GPS-tagged by users. We manually verified their location and made the necessary adjustments as the user specified GPS-tags are often inaccurate. In our experiments, each query image is matched against the entire reference dataset and not the ground truth city only. Sixteen sample queries are shown in Fig. 4.7-right. The quality of our reference set (made publicly available) surpasses that of the currently available street view datasets [86, 2] in terms of the image resolution.

4.3.2 Analysis of the Proposed Method

In this section, we provide two experiments to quantitatively analyze various aspects of the proposed method and demonstrate its robustness.

Comparison of Different Global Features: Any arbitrary type of global features can be used in the proposed framework. Fig. 4.8-left compares the geo-localization results obtained by four different global features while setting $k = 5$ and employing the winner-take-all voting scheme for location estimation as explained in Sec. 4.1.3. We normalized the local and global feature vec-

tors prior to forming the input to GMCP in order to make sure they produce comparable distances and do not dominate each other. The horizontal and vertical axes shows the error threshold in meters and the percentage of the test set localized within a particular error threshold, respectively. Since the scope of this work is precise localization in a city scale, we focus on error values less than 300 meters in our plots as a higher error typically implies failure.

The blue and red curves show the results of using a 960-dimensional GIST [87] and a 60-dimensional RGB Color Histogram as the global feature, respectively. Each image in our reference dataset is associated with a GPS-tag denoted by a two dimensional vector of latitude and longitude (ϕ, λ). Even though the location is not based on visual information, it can serve as the global feature as it is a holistic tag for the image. The green curve was computed using the (ϕ, λ) location vector as the global feature after conversion to Cartesian coordinates values. The superior performance of the location features is mainly because they provide complimentary information to the visual content of the image as they are non-visual descriptors. We used the location as the global feature in the rest of our experiments.

The cyan curve depicts the results of using the image identity as the global feature; that is, the edge weight between two NNs is zero if they come from the same reference image and 1, otherwise. The fact that the other global features perform better than the image identity, in particular location and color histogram by up to 7.4% and 4.8%, respectively, signifies that the improvement made by our algorithm is not due to simply encouraging the NN matches to be selected from one image or a small set of images; in other words, GMCP is indeed leveraging the relationship between the global features of the NNs to identify the inlier nodes. However, Fig. 4.8-left shows that different global features have different performances in encoding the relationships among NNs, and choosing the appropriate type of global feature is essential.

The value of σ in equations (4.6) and (4.7) was determined empirically using a small validation set of 10 random queries for which there are multiple disjoint groups of inlier NNs. This resulted in the values of 32, 1024 and 256 for color histogram, GIST and location features, respec-

tively. Typically, the bandwidth depends on the type of the feature, the number of dimensions, and the range of values.

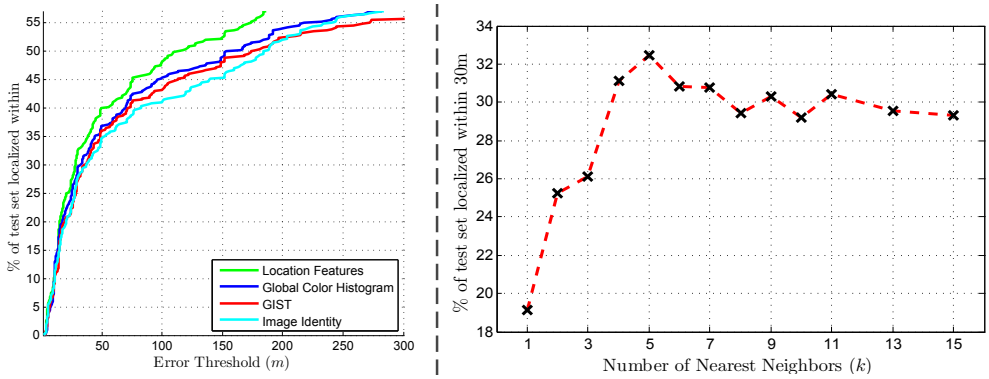

Figure 4.8: **Left:** Comparison of the overall Geo-localization results using different global features. **Right:** Geo-localization accuracy with respect to k.

Number of Considered Nearest Neighbors (k) : The appropriate value for k in the GMCP-based method depends on the amount of repetition and similarity in the features of the reference dataset. An insufficiently large k would lead to considering too few NNs in the matching process and consequently a small chance of discovering the correct one. On the other hand, an excessively large value would result in too many noisy NNs in the input graph and higher complexity of the optimization task. In order to show the impact of the number of considered NNs, we performed an experiment by running the GMCP-based method with different values of k ranging from 2 to 15; Fig. 4.8-right provides the percentage of the test set localized within the arbitrary distance threshold of 30 meters. For all values of k (when $>$1), the overall accuracy is observed to be significantly higher than the baseline, i.e. using the 1^{st} NN only. However, when k becomes too large ($>$5 for our dataset), more features which do not have any inlier among their NNs survive the pruning and consequently, the accuracy slightly decreases. This observation is consistent with the characteristics of our dataset as a building is typically visible in the view of 3 to 5 street view place

marks; therefore, there is a higher chance to find the correct NN among the top five NNs. We set $k = 5$ in the rest of our experiments.

4.3.3 Comparison of the Geo-localization Results

Fig. 4.9 shows the results of evaluating the proposed algorithm along with the baselines in terms of *overall geo-localization results*. The black and light green curves illustrate the performance of GMST and GMCP based feature matching methods, respectively when no robustification is utilized (employing ℓ_2 distance instead), along with the winner-take-all voting scheme for location estimation. The dark green curve illustrates the performance of chance where the query images were randomly matched to the reference images; the curve is generated by calculating the expected value of the percentage of test set localized within a particular error threshold. The poor performance of chance in Fig. 4.9 is due to the fact that the covered area is several square kilometers wide; therefore, it would be unlikely to randomly localize an image within a few hundred meters of its actual location. However, even though many of the query images are localized within few tens of meters of their ground truth using our method, yet some large error values of over 150 meters are observed in the curves of Fig. 4.9. There are two main reasons behind such large distances: First, many of the user uploaded query images were taken at locations where the closest street view place mark is over 150 meters away, e.g. parks and play grounds. For those cases, a large error will be reported in the localization curves even if the algorithm finds the best matching street view image. Second, many of the tall and large buildings in urban area have facades which look identical from different viewpoints. We observed that the algorithm often matches a query image of such facades to the correct building yet not necessarily the correct facade; those cases typically lead to large error values which are due to the confusion caused by the symmetry of buildings. In the next section, we provide the quantitative results which show the majority of images localized within a few hundred meters of ground truth indeed have an overlap in scene with the matched image.

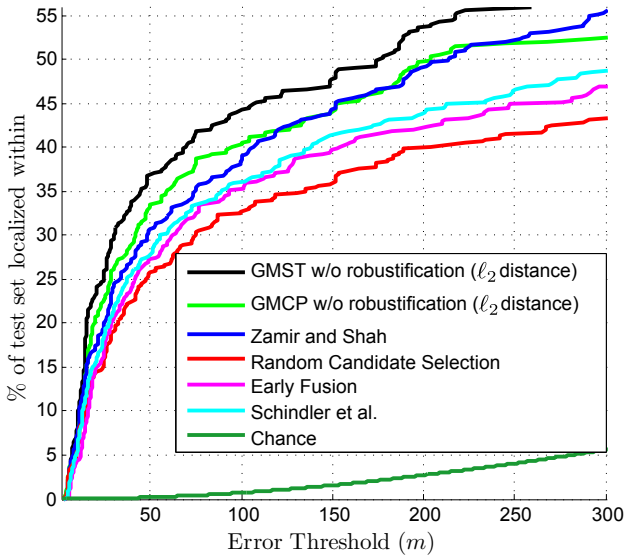

Figure 4.9: Overall geo-localization results using GMST and GMCP (without robustification) along with the baselines. Horizontal and vertical axes show the distance threshold and the percentage of the test set localized within the distance threshold, respectively.

The cyan curve in Fig. 4.9 shows the localization results obtained by Schindler et al.'s method [14] which is based on image matching employing the bag of visual words model. As discussed in Sec. 1.2.1, the issue of excessive quantization loss commonly challenges the localization methods based on bag of visual words model when tested in an urban area. This concern is not applicable to the our method as the matching is performed on raw local features, not a quantized version of them. In this context, particularly the GMCP-based method can be coupled with fast and approximate NN search methods [88, 73] to search through the large number of raw features in a timely manner; this is because our approach does not strictly require the 1^{st} retrieved NN to be the right one. In fact, GMCP is capable of identifying the right NN as long as it appears among the top retrieved NNs, which can partially alleviate the suboptimal performance of the NN search methods.

The red curve represents a baseline in which one of the top k NNs for each feature point is randomly selected as the match. The blue curve depicts the results of the method introduced in the previous chapter which was based on using the first NN only. By increasing the size of the dataset, the pruning methods which are based on the first NN only, e.g. [16] and the one introduced in Sec. 3.2, tend to over-prune the query features. This results in too few query features participating in the voting scheme and consequently less reliable geo-localization which is one of the reasons behind the relatively low accuracy achieved by the method of Chapter 3.

Early Fusion: One way of combining local and global features, typically termed early fusion, is normalizing the features vectors, concatenating them and treating the new vector as one feature. The purple curve in Fig. 4.9 depicts the localization results when feature matching is performed using this method. Using early fusion for feature matching has a number of inherent disadvantages which explain its poor performance. For instance, treating concatenated features as one vector requires the two features to be matched in the same space and using one type of distance, which is undesirable in many cases.

4.3.4 Results of Robustifying the Distance Function

The results illustrated in Fig. 4.9 were obtained without using the robust distance function, which is why the improvement made by the GMCP-based method is not more than 3% over different baselines. The solid and dashed green curves in Fig. 4.10 (a) compare the performance of GMCP with and without the robust distance function D (using ℓ_2 distance instead), which shows the robustification improves the overall localization results by up to 7% as it deals with the issue of disjoint inlier groups.

In Sec. 4.1.2.2 we argued that GMST outperforms GMCP-without-robustification due to its capability of dealing with the disjoint inlier groups; this is consistent with the experimental results of Fig. 4.10 (a). However, as shown in Fig. 4.10 (a), using the robust distance function does not make a sensible difference in the results of GMST since the discussion of Sec. 4.1.2.2 is not

applicable to it. Additionally, Fig. 4.10 (a) shows that the robustified GMCP performs significantly better than GMST. This is due to chaining phenomena which makes GMST more prone to noise and outliers as discussed in Sec. 4.1.2.2. Therefore, we conclude that employing GMCP with the robust distance function D yields the best results in the proposed framework.

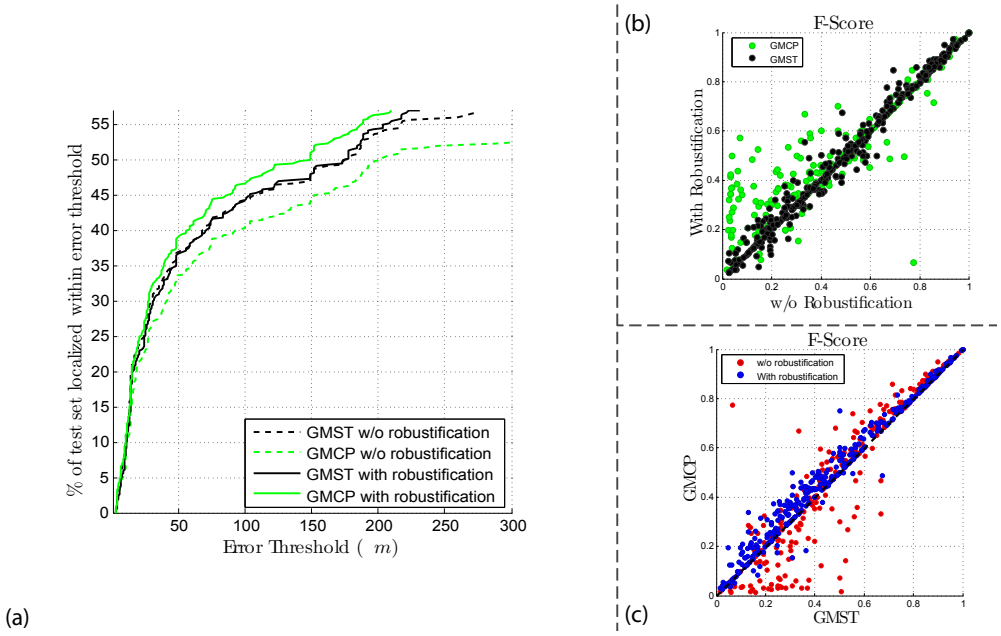

Figure 4.10: The impact of using the robust distance function. (a) depicts overall geo-location results. Note the significant positive effect on GMCP results, and negligible impact on GMST. (b) and (c) show the scatter plots of F-score values. (b) illustrates the effect of robustification; green and black points show that for GMCP and GMST, respectively. (c) compares the performance of GMCP to GMST; that is shown for the two settings of with and without robustification.

In addition, in order to provide further insight into the quality of our results, we manually verified if the found street view images actually match their corresponding query images: We found 94.4% of the query images which were localized within 300 meters of the ground truth by the robustified GMCP (i.e. the solid green curve in Fig. 4.10) had an overlap in scene with the

found reference image while only 5.6% were matched incorrectly and did not show anything in common.

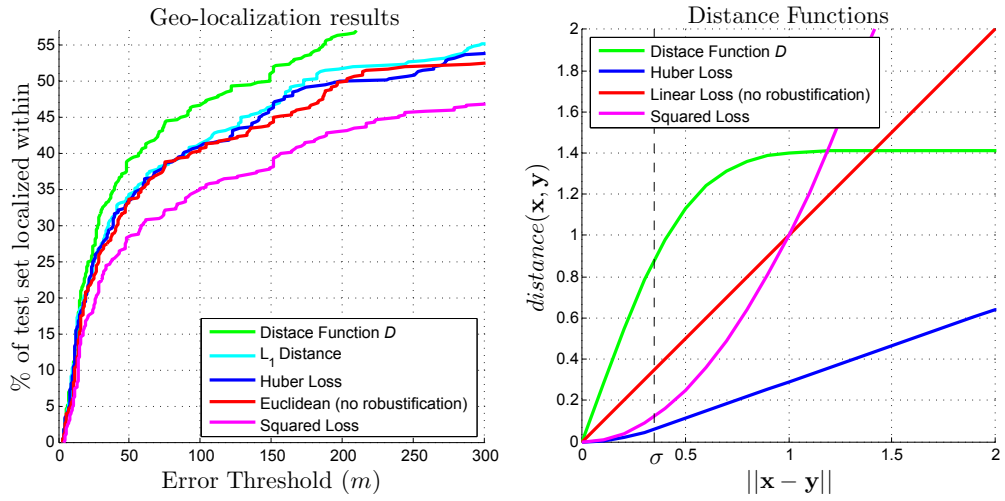

Figure 4.11: Left: Geo-localization results using various distance functions. Right: Illustration of the functions.

Fig. 4.11-(left) shows the geo-localization results obtained using various distance functions; the corresponding functions are illustrated in the right plot. The linear loss (Euclidean) represents the case where ℓ_2 norm was used as the distance (i.e. no robustification). The purple and the dark blue curves illustrate the results of using Squared ($\|\mathbf{x} - \mathbf{y}\|^2$) and Huber-loss [89] functions with tuned σ, respectively. As apparent in Fig. 4.11-(left), robustification using the distance function D yields the best results which justifies our choice. In particular, the fact that Huber-loss performs worse than the distance function D signifies that mapping the large distances to a maximal value, as compared to only damping them with an non-zero slope (which is what Huber-loss does by definition) is essential. Additionally, Squared loss boosts the large distances, and consequently has the worst results, which empirically shows our argument on the necessity of addressing the large distances is valid.

4.3.5 Feature Matching Evaluation

The evaluation of the proposed method and the baselines in terms of the *performance of feature matching* is shown using scatter plots of precision, recall and F-score values (figures 4.10, 4.12). For each query, the set of reference images that have an overlap with it are known. Therefore, we can examine how many of the query features are matched to one of these correct image matches. The precision of feature matching is defined as the number of correctly matched query features divided by the total number of query features after pruning. Recall is defined as the number of correctly matched query features divided by the total number of query features which have a NN belonging to one of the matching reference images among their top k NNs. F-score is a measure which combines both precision and recall and is defined as their harmonic mean.

The green points (where each point represents one query image) in the F-score scatter plot of Fig. 4.10 (b) compare the performance of GMCP before and after robustification. The diagonal dashed line shows the neutral border where the performances of both settings is the same. Therefore, a green node above the dashed line represents a query for which the performance of GMCP is improved after robustification. Similarly, the impact of robustification on GMST is shown using the black points. As apparent in Fig. 4.10 (b), the performance of GMST does not considerably change with and without robustification, while GMCP's performance significantly improves by employing the robust distance function D.

The green nodes in Fig. 4.10 (b) which are positioned close to the neutral line represent the query images which do not match to more than one group of matching reference images, and consequently, the robustification does not make a sensible difference on their corresponding performance. On the other hand, the nodes which are located above the neutral line represent the queries which have disjoint groups of matching reference images.

The scatter plot of Fig. 4.10 (c) compares the performance of GMCP vs. GMST on individual query images: that is shown for the two settings of *with* and *without* robustification in blue

and red, respectively. This plot signifies that GMCP is superior to GMST when the robust distance function is utilized.

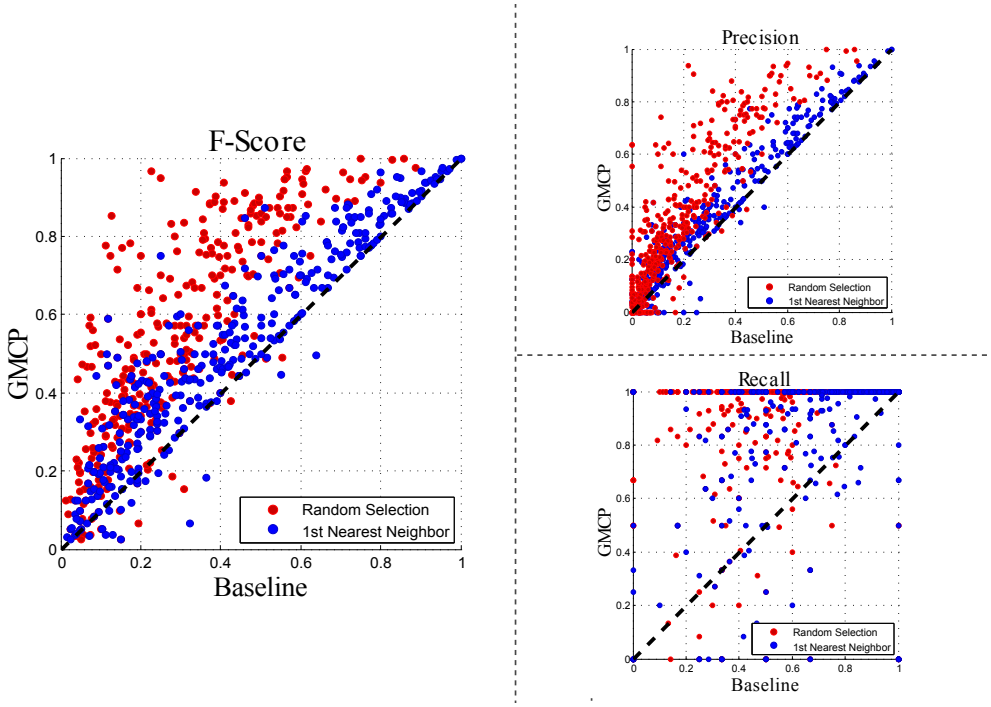

Figure 4.12: Scatter plots of F-score, precision and recall values. Vertical and horizontal axes show the values gained by the GMCP-based method and the baseline, respectively. Each node represents one query image.

Fig. 4.12 shows the scatter plots of precision, recall and F-score values which compare the performance of GMCP (with the robust distance function) vs. the baselines. Blue and red baselines represent represent using the 1^{st} NN and randomly selecting one of the NNs, respectively. As apparent in this figure, the observed improvement in the results of feature matching is attributed to considering more than one NN and the performance of the GMCP-based method in robustly discovering the correct feature correspondences.

4.4 Chapter Summary

In this chapter, we introduced a multiple-NN feature matching method based on Generalized Minimum Clique Problem (GMCP) to address the shortcoming of local features in leveraging the global context. The developed method is capable of incorporating both global and local features simultaneously. We showed that using a robustified function for finding the distances between the global features is essential when the query image matches multiple reference images with dissimilar global features. In this context, we proposed a robust distance function based on the Gaussian Radial Basis Function (G-RBF). Different types of local features can be used for nominating the NNs. Therefore, our method can be adopted to utilize multiple types of local features in order to maximize the amount of leveraged information. Our experiments showed that the GMCP-based feature matching significantly improves the overall accuracy of image geo-localization. In the next chapter, we will show that automatic geo-localization can be extended from images to videos, and the geo-spatial trajectory of the camera can be extracted.

CHAPTER 5: VIDEO GEO-LOCALIZATION AND GEO-SPATIAL TRAJECTORY EXTRACTION

Figure 5.1: Geo-spatial trajectories of thirteen user videos recorded in downtown Pittsburgh.

Thus far, we discussed how the geo-location of *images* can be automatically extracted. Likewise, it is logical and desirable to develop frameworks for estimating the geo-location of *videos*. Such methods are of particular importance as the geo-location of videos are often not preserved at the time of collection (as opposed to images for which the geo-tag is sometimes preserved in the Exif tag). In general, automatic geo-localization of consumer videos is an underdeveloped area of research compared to image geo-localization. This is reflected in practice as there are several online image collections [10, 90] which provide images in a geographically structured manner, while a similar repository for videos has not been developed to date.

In this chapter, we show that the task of geo-localization can be effectively extended from images to videos. We present a novel framework for extracting the geo-spatial trajectory of the camera from a video in a city scale (Fig. 5.1 shows sample extracted trajectories). The devel-

oped method is intended for user-uploaded videos, e.g. YouTube clips, which typically include unwanted defects, such as blurred or uninformative frames, abrupt changes in camera motion, zooming, frequent occlusions, and lack of information of the initial position and pose (e.g. meta-data) where the video was recorded.

Our method is based on a three step process: 1) individual geo-localization of video frames using Street View images to obtain the likelihood of the location (latitude and longitude) given the current observation, 2) Bayesian tracking to estimate the frame location and videos temporal evolution using previous state probabilities and current likelihood, and 3) applying a novel Minimum Spanning Trees based trajectory reconstruction to eliminate trajectory loops or noisy estimations. The details of each step are provided in the following sections.

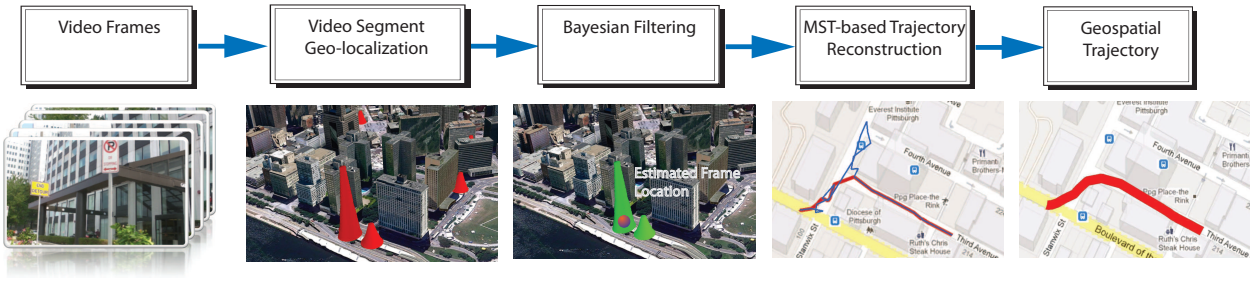

Figure 5.2: Schematic of our method for estimating the geo-spatial trajectory of a camera in a city.

5.1 Video Geo-localization Framework

Figure 5.2 shows the block diagram of the proposed method. Initially, frames are sampled periodically from the video. Each frame of a video is geo-localized according to the procedure described in Chapter 3. The output of the individual frame geo-localization algorithm is a probability map with votes over the most probable locations, as described below. In Chapter 3 [2], the highest peak in the probability map of votes were selected as the GPS location of the query image. Frame

by frame estimation using this technique fails because video sequences typically contain many frames that are not assigned to the correct geo-location. Instead of using the individual estimation of the geo-location, utilizing the aforementioned procedure, we interpret the probability map votes as the 2D likelihood (with random variables of latitude and longitude) given the current frame observation. Thereby, multiple feasible hypotheses are considered for the current frame location, as opposed to a single specific frame position, which was mentioned previously.

With multiple possible locations for each frame, the problem can be understood as a measurement association and single target tracking problem. Therefore, the next step in our method is a Bayesian tracking filter. The Bayesian tracking algorithm enforces the temporal consistency. In an analogy to tracking formulation, we set up a "range gate" where only votes inside the gate region are considered, while detections (votes) outside of the gate are ignored. Data association raises additional difficulties in this problem. Firstly, the geo-localization of individual frames based on visual features is often not accurate. As a consequence, the probability map tends to be very noisy. In fact, it is very common to find probability maps where the highest vote location does not correspond to the real location of the camera in the evaluated frame. Secondly, the size of the gate must be a large region in terms of the local position. The vote maps are associated with GPS-tags that are sampled discretely. Then, the selected gate must be large enough to cover several locations of these geo-referenced tags, which can encompass hundreds of meters. Due to the aforementioned difficulties, the standard data association techniques cannot easily be adapted to obtain precise trajectories. For example, standard nearest neighbor filter will fail because the data is too sparse, which will produce noisy measurements.

Moreover, splitting the track into multiple hypotheses every time that more than one vote in the validation region is detected becomes impractical due to the large number of false alarms. Therefore, the trajectory estimation output from the Bayesian formulation is still noisy, particularly when the images are taken close to street corners where building façades look similar from both sides of intersecting streets, and where distant buildings get into the camera field of view, causing

a false estimation that produces inaccuracies in the trajectory. Therefore, the final step of our approach is a trajectory reconstruction method which will eliminate loops and noisy estimations of the trajectory using our Minimum Spanning Tree (MST) based trajectory reconstruction algorithm. Each step of our method is explained in detail below.

5.1.1 Geo-localization of a Video Segment

We use the method of Chapter 3 as a baseline for single image geo-localization since it produces a vote distribution instead of a single geo-location. As explained in Chapter 3, interest points of the query image and reference images are described using SIFT descriptors [91]. For every SIFT descriptor of the query image, a set of nearest-neighbors is extracted from the reference database using a tree search [92]. Each of the these nearest-neighbors votes for their corresponding geographical position in the database, creating a map of votes for the city. Then, some of the votes are discarded according to the proposed criteria relating the proximity of the query descriptor to the matched descriptor and the geographical proximity of the set of nearest neighbors. The GPS location of the image corresponds to the highest peak found in the obtained map. Also, a *Confidence of Localization (CoL)* parameter, which can be used as a measure of the reliability of the estimation, is derived from the Kurtosis of the map.

The GPS plac-emark locations in the reference dataset are spaced approximately every 12 meters. All the sampled frames from the query video corresponding to the time period where the camera moves 12 meters around a plac-emark, should vote for the same location in the reference dataset. This can be interpreted as a quantization process since we are constraining a continuous set of values (global position) to some discrete set of values (GPS place marks). Processing individual frames will produce a quantization error in the frame position estimation. Therefore, it is more helpful to gather sets of consecutive frames and treat them as a video segment. Hence, a map of votes corresponding to a video segment is achieved by averaging the vote maps of each one of the frames that belong to the segment. Geo-localization of a video segment has also two positive

side effects. The first is the enforcement of the most common vote locations in the segment of frames, which typically correspond to correct geo-locations. The other positive side effect is the attenuation of votes at locations which fewer frames vote for, that typically corresponds to false alarms. Indirectly, geo-localization of video segments facilitates the data association.

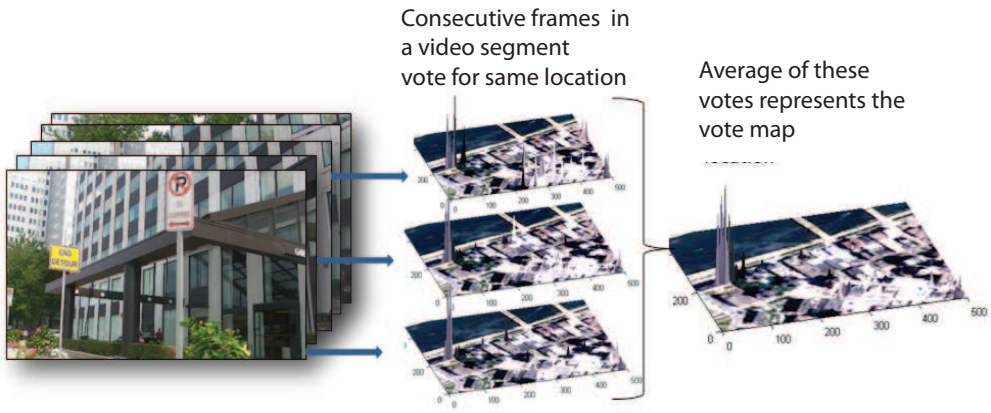

Figure 5.3: The GPS place marks in the reference dataset are located approximately every 12 meters. The vote distribution of frames in a video segment during a period of time where the displacement was shorter than 12 meters are averaged since they are essentially voting for the same place mark.

5.1.2 A Bayesian Formulation

A Bayesian formulation is plausible, if the vote distribution of the video segment is interpreted as the likelihood of the location (latitude and longitude) given the current observation (see Fig. 5.4(a)). A video is constrained in the spatial and temporal domain because consecutive frames correspond to close spatial locations. Consequently, Bayesian tracking is used to estimate the frame localization and its temporal evolution. The objective is to estimate the state x (latitude and longitude) at any sampling time t.

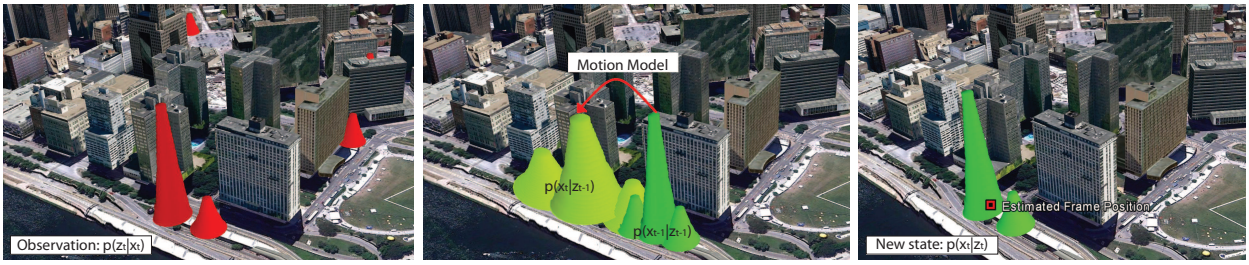

Figure 5.4: Bayesian estimation process. a) The observation is the vote distribution from a video segment. b) The prediction of the state(latitude,longitude) based on the previous state. c) The new state probability function computed using the state prediction and observation.

Let x_t represent the state (latitude and longitude) at the time t, z_t represent the observation at the time t, Z_t represent the history of the observations $z_1, z_2, \ldots, z_t$. We are interested in obtaining the distribution $p(x_t|Z_t)$, which describes the probability of the state x given the previous history of observations. It is evident that the distribution $p(x_t|Z_t)$ can be rearranged as $p(x_t|Z_t) = p(x_t|z_t Z_{t-1})$. Using the Bayes rule, we have:

$$p(x_t|Z_t) = \frac{p(z_t|x_t Z_{t-1})p(x_t|Z_{t-1})}{p(z_t|Z_{t-1})}. \tag{5.1}$$

The term in the denominator is not related to the variable x, it is simply a normalization constant that does not effect the probability distribution. Then, the denominator is replaced by a constant c to obtain:

$$p(x_t|Z_t) = \frac{p(z_t|x_t Z_{t-1})p(x_t|Z_{t-1})}{c}. \tag{5.2}$$

However, the observation process at the frame t is not related to the observation process at

the previous time. Therefore, we can rewrite the previous equation as:

$$p(x_t|Z_t) = \frac{p(z_t|x_t)p(x_t|Z_{t-1})}{c}, \tag{5.3}$$

where $p(z_t|x_t)$ is the observation model or likelihood, and $p(x_t|Z_{t-1})$ is the predictive model of the process. The dynamical model is assumed to be a Markov model, which implies that the current state depends only on the previous state. This is an appropriate assumption when the previously estimated frame localization is correct. The marginalization of the probability distribution representing the predictive model becomes:

$$p(x_t|Z_{t-1}) = \int_{x_{t-1}} p(x_t|x_{t-1})p(x_{t-1}|Z_{t-1}). \tag{5.4}$$

Fig. 5.4 illustrates this process. The term $p(x_t|x_{t-1})$ is the probability of the future state given the current state, which can be derived from the motion model of the camera (constant velocity model); the term $p(x_{t-1}|Z_{t-1})$ is the former probability of the state given the observation, which is available from the previous state estimation. The above equation can be substituted in equation 5.3 to obtain the probability of the state given the current observation. The constant of normalization c can be calculated as:

$$c = \int p(z_t|x_t)p(x_t|Z_{t-1}). \tag{5.5}$$

The normalization constant is also a measure of how often the number of votes estimated in the current state is close to the probability predicted from the previous state (gate). In other words, a value c close to zero could indicate false localization. Therefore, the value of the variable c is used in our algorithm to discard some of the untrustworthy observations. The state estimation in cases where c is close to zero are discarded, and a new probability function is built using the earlier state estimation as the most probable state. In the case that the value of c is close to zero

in several consecutives estimations, a redetection process is performed, the same way the initial geo-location is computed, as described below. In the case where the values of c are not close to zero, the estimation of the state would be given by the expectation:

$$E[x_t|Z_t] = \int x_t p(x_t|Z_t) dx. \tag{5.6}$$

Fig. 5.4(c) shows the result of the new state distribution after taking the product of the frame segment observation (Fig. 5.4(a)) and the state prediction (Fig. 5.4(b)) according to equation 5.3. The state localization for a frame segment at the time t is computed using equation 5.6, and is marked by the red marker in the Fig. 5.4(c).

Discrete version using a constant velocity model.

Our experiments show that the constant velocity motion model performed slightly better than the constant acceleration and random walk/Brownian motion models in our method. A discrete version of the formulation can be implemented by defining the city map as a dense grid. The vote distribution which symbolizes the likelihood of the current state is represented as an array, as is the state given the observations and state prediction. The probability of the future state given the current state $p(x_t|x_{t-1})$ is expected to be a shifted version (according to the constant velocity model) of the previous state distribution with some randomness added. A mathematical expression that fairly characterizes the state prediction given the precedent observations (5.4) is

$$p(x_t|Z_{t-1}) \approx U(x_t - (x_{t-1} + v_t)) * p(x_{t-1}|Z_{t-1}), \tag{5.7}$$

where U represents a uniform distribution centered around the origin, v_t is the velocity (shift) at time t, and the symbol $*$ represents a 2D convolution operator.

Finally, the state estimation for the latitude and longitude is obtained by using the discrete version

of the expectation equation:

$$E[x_t|Z_t] = \int x_t p(x_t|Z_t)dx = \sum_i x_t^i p^i(x_t|Z_t). \tag{5.8}$$

Estimation of the Initial Geo-locations

In order to obtain the initial geo-localization, we consider a group of periodically sampled frames around the first frame of the video. For each one of the sampled frames, its frame geo-localization is estimated as the highest peak of the vote distribution of the frame. It is highly probable that some of these frames are not correctly geo-localized; therefore, we have used two different pruning steps to remove them. The first step is to reduce the number of false geo-localizations using the information provided by the *Confidence of Localization (CoL)*, which was described in Chapter 3. The estimated frame geo-localization is discarded by thresholding the *CoL* value. The second step is to discard the geographically isolated frame localizations by counting the number of frame geo-locations within a prudent radius r of the frame being tested. The frame is discarded if the number of surrounding neighbors is less than the threshold. After applying these two pruning steps, the remaining frames are averaged to obtain an estimation of the initial geo-localization.

5.1.3 Minimum Spanning Tree-based Trajectory Reconstruction

Ideally, employing a sophisticated motion model which is capable of handling abrupt changes in the direction, zooming, tilting, lack of metadata, noisy frame-by-frame localizations, etc. in our Bayesian framework would yield a smooth and appropriate trajectory for a video. However, such motion model which is capable of addressing all aforementioned complications is not developed to date. Typically, any on-line (causal) approach to enforcing temporal consistency which exploits a motion model poses some *inertia* in motion estimation, due to the presumptions the motion mode is based on. Additionally, all the large scale image localization methods [2, 14, 93] which provide the input to the Bayesian Filter, are expected to geo-locate a frame with

an error of a few tens of meters. Although this error value is acceptable for a city scale localization algorithm, it can cause inconsistency in the trajectory that the video segments form, even after applying the Bayesian filter. For instance, the inertia of motion model along with an error value of a few tens of meters in the video segment locations can cause the trajectory to go straight at an intersection for at least a few video segments while the camera has actually made a turn.

An example is depicted by the magenta contour in the Fig. 5.5(a). Video segment locations which slightly deviate from but are still close to the main stream of the trajectory result in another case of inconsistencies caused by the slight inaccuracy of individual frame localization method which Bayesian filter cannot effectively handle (depicted by black contour in Fig. 5.5(a)). These cases, along with other types of complications (e.g. inaccurate yet repeated frame locations, which are due to zooming and focusing on a nearby buildings) cause the extracted geo-spatial trajectory to possess special characteristics which can not be handled effectively by basic trajectory reconstruction or smoothing methods like moving average (MA). Therefore, we propose a trajectory reconstruction method based on Minimum Spanning Trees which can effectively handle these complications.

Minimum Spanning Trees(MST) have been used extensively in a variety of fields ranging from network design [94] to medical image analysis [95]. Ma. et al. [95] use MST in robust image registration. Perlman [94] utilizes MST for the efficient design of computer networks. MST has been used in curve formation as well. I. Lee [96] proposes a curve reconstruction method based on moving least square improved by MST. Figueiredo and Gomes [97] use MST to reconstruct differentiable arcs from dense samples. The reason behind the varied uses of MST is its characteristic ability to find a minimal way of linking some entities. In our case, these entities are the video segment locations acquired from the Bayesian filter. The proposed geo-spatial trajectory reconstruction method using Minimum Spanning Trees is described in Algorithm 1.

Algorithm 2 MST-Based Trajectory Reconstruction

1: Find the M*inimum Spanning Tree* of $G = (\mathbf{N}, \mathbf{E}, \mathbf{W})$

for i where (degree of node i) > 2 **do**

2: Set *Root* to node i.

3: Set *Weight* of each branch connected to the *Root* to the number of nodes on it.

4: Retain the two branches with higher weights and remove others.

end for

5: **return** *Minimum Spanning Tree* with retained nodes.

In Algorithm 1, the nodes, edges and cost of edges of the graph G are represented by $\mathbf{N}, \mathbf{E}$ and $\mathbf{W}$, respectively. Each video segment is represented by one node in $\mathbf{N}$. Each node has the feature vector (x_i, t_i), where x_i is the corresponding video segment's geo-location and t_i is its respective time obtained from Bayesian filter. E includes the edges between all possible pairs of nodes. The cost of each edge is defined as the Euclidean distance between the feature vectors of the nodes that edge connects.

The process of MST-based trajectory reconstruction is illustrated in Fig. 5.5. First, the output locations of the video segments and their respective time (Fig. 5.5 (a)) are acquired from the Bayesian Filter and the graph G is formed. Then, the Minimum Spanning Tree of G is found (Fig. 5.5 (b)). The degree of a node in a MST is defined as the number of edges connected to it. The next step is to identify the nodes with a degree higher than two (orange nodes in Fig. 5.5 (b)). For such nodes, we define the weight of each connected branch as the number of nodes connected through that branch to the root. This is illustrated in Fig. 5.5 (c) for one of the nodes with a degree higher than two. Then, the nodes on the two branches with the highest weights are retained and the rest are removed. When a node with a degree higher than two is observed, it means there is a node which is likely off the mainstream of the trajectory and consequently an additional branch has appeared. Such a node is either geo-spatially or sequentially inconsistent with the rest of the

path. The process of assigning a weight to each branch is intended to identify the branch(es) which contains an outlier and consequently should be removed. The branch which has fewer nodes that are connected to the root is less likely to be on the mainstream, since fewer video segment locations are consistent with its location. Therefore, we retain the two branches with highest weights, which ensures the connectivity of the trajectory, and remove the rest. The final trajectory is shown in Fig. 5.5 (d).

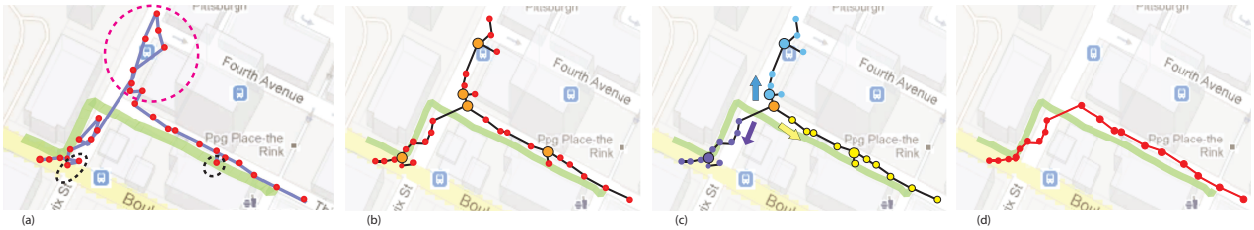

Figure 5.5: Illustration of the different steps of MST based trajectory reconstruction. The green trajectory represents the ground truth. a) Output of the Bayesian filter. b) Minimum Spanning Tree. The nodes with a degree higher than two are shown in orange. c) The branches of a particular node with a degree higher than two (shown in orange) are marked with arrows. Yellow and purple branches are retained and the blue one is removed as it has less weight. d) The final reconstructed trajectory.

Note that the features used to determine the MST, include both time and geo-location information. Therefore, if the camera revisits a previously visited location, the nodes corresponding to the first and second visit will not be mistakenly linked in the MST as their time features are very different even though their geo-spatial locations are close. An alternative algorithm to the one in line 3 of Algorithm 1 performs *breadth first search* with the root set to x_i and retains the nodes of the two *deepest* branches rather than those with the highest weights. However this method would be computationally more expensive than the original algorithm in line 3, yet it performs better if

the branches, including the correct ones, are highly contaminated with outlier nodes.

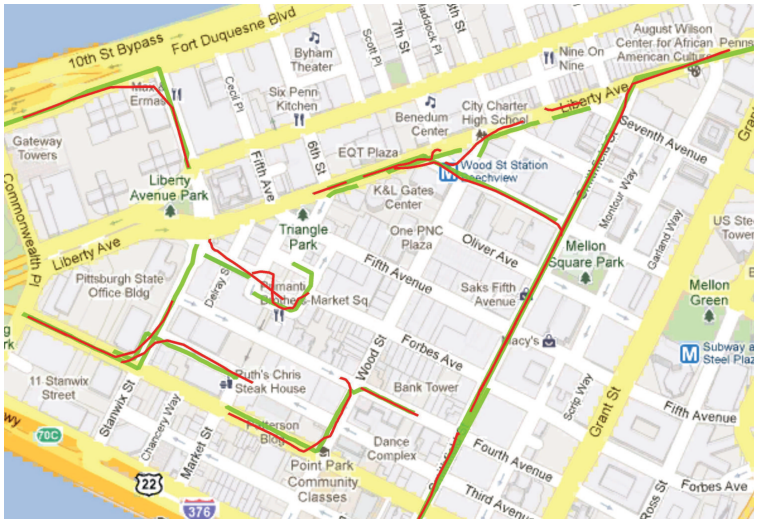

Figure 5.6: A subset of trajectories obtained from videos in downtown Pittsburgh. The green trajectories correspond to the ground truth, while the red ones correspond to our Bayesian framework + MST trajectory reconstruction.

5.2 Experimental Results

We use the Street View dataset presented in Chapter 3 as the reference data. We evaluate our framework using 45 user-shared videos, with the durations ranging from 60 to 120 seconds and total number of 106,200 frames. The query videos were recorded in downtown Pittsburgh, PA and downtown Orlando, FL; they were downloaded from YouTube [98] or captured by different users using a consumer grade video cameras while walking or driving in the city without prior knowledge about the usage of the videos.

Fig. 5.6 shows the trajectories obtained from the query videos recorded in downtown Pittsburgh using our proposed Bayesian filtering and MST based trajectory reconstruction. In the figure, the green lines represent the ground truth trajectories of the camera, while the red ones are

the trajectories produced by our algorithm. These qualitative results corroborate that our algorithm is successful in obtaining the accurate trajectory of a camera in an area as large as a city (Note that the area covered by the dataset is larger than the frame shown in Fig. 5.6). Fig. 5.7 shows examples of two trajectories obtained using Bayesian filtering, and their outputs after performing MST-based curve reconstruction.

Figure 5.7: Two MST based trajectory reconstruction examples. The figures (a) and (c) correspond to the Bayesian filtering of the two examples. Figures (b) and (d) are the trajectories obtained after applying MST based trajectory reconstruction to the trajectories in (a) and (c).

5.2.1 Implementation Details

Each one of the videos is sampled every ten frames to produce a frame rate of approximately 3 frames per second (fps). Ten of these sampled frames are used to form a video segment, since the displacement of an object in a city is typically less than 12 meters in 3 seconds. In the reference dataset, Scale Invariant Feature Transform (SIFT) points are computed for each one of the Google street view images. The SIFT descriptors and their corresponding GPS-tags are indexed in a tree using FLANN [92]. A map of votes for each query frame is calculated by computing SIFT descriptors in the query image, obtaining a list of nearest neighbors to the indexed features for each interest, and using the voting scheme previously described. The initial geo-localization estimation proposed in Sec. 5.1.2 is employed to initialize the algorithm. The value of the *CoL* threshold is

Table 5.1: Comparison of the mean error in meters for a subset of 12 videos from our test set.

Randomly Selected Videos from Pittsburgh and Orlando													
	Avg Error.	Seq. 1	Seq. 2	Seq. 3	Seq. 4	Seq. 5	Seq. 6	Seq. 7	Seq. 8	Seq. 9	Seq. 10	Seq. 11	Seq. 12
Frame by frame	268.6	332	207	198	102.3	161.9	197	143.2	196.1	151.9	276.3	235.0	249.0
Bayesian filtering	10.57	**2.10**	**2.42**	5.60	7.21	12.27	**1.06**	18.01	7.18	1.13	11.03	13.15	16.85
Bayesian + M.A.	10.17	3.20	3.07	6.03	7.13	11.80	1.08	17.31	6.86	**1.00**	11.17	12.96	**15.78**
Bayesian + MST	**9.94**	**2.10**	**2.42**	**5.26**	**5.15**	**11.68**	**1.06**	**10.84**	**5.93**	1.13	**10.80**	**12.26**	16.15

set to 40% and the radius r is set to 40 meters. The uniform function described in equation 5.7 was set to cover an approximate radius of 70 meters.

5.2.2 Quantitative Results

In order to compare our algorithm to FAB-MAP[20, 21], we used the bag of visual words model with the Chow-Liu tree to perform individual frame localization, utilizing the Google street view geo-tagged images as the history of observations. Then, we computed the likelihood of each of the video frames being in any of the possible geo-tagged locations. The average frame-by-frame error of the first step of FAB-MAP algorithm was 441.01 meters. The high error value in the first step prevents the algorithm from forming an appropriate trajectory in the later steps. The large error value is primarily due to the differences in our problem and the one FAB-MAP addresses, which is detecting if a robot is revisiting a previously visited location. The history of frames showing previously visited locations is assumed to be recorded using the same robot, which significantly simplifies the frame localization step compared to our problem which requires matching wild video frames to reference street view images. Note that the mean individual frame localization error of our method is 268.6 meters.

Table 1 shows the results of the experiments for a set of 12 randomly selected videos of the test dataset. The error metric is defined as the mean distance (error) between the estimated

frame geo-localizations and the closest ground truth frame. The results in the table 1 are listed in meters. The first row of the table contains the results obtained using individual frame by frame geo-localization. Mean errors of individual frame by frame geo-localization of these videos range from 66 to 535 meters. These values demonstrate the low performance of frame by frame geo-localization in determining a trajectory. In contrast, the mean errors of the proposed Bayesian filter has an average mean error value of 10.57 meters. The mean errors in most of the videos are lower than 20 meters. The subsequent rows in the table compare the trajectories obtained after applying moving average (MA) smoother to the output of the Bayesian filter versus the trajectories obtained using the proposed MST based trajectory reconstruction applied to the output of the Bayesian filter. The best performances are indicated by bold characters. As can be seen, most of them correspond to the MST-based trajectory reconstruction method.

5.3 Chapter Summary

In this chapter, we showed that the task of geo-localization can be extended from images to videos. We addressed the problem of estimating the geo-spatial trajectory of the camera from "videos in the wild". We developed a solution to this problem based on individual geo-localization of frames, Bayesian filtering, and a MST-based curve reconstruction algorithm. Bayesian filtering enforced the temporal consistency of the video, and the MST-based trajectory reconstruction was intended to handle the near-stochastic motion of the camera in the user video and remove the inconsistencies. The presented method is particularly good for YouTube clips and videos recorded using hand-held devices. In the coming chapter, we will discuss that automatic image or video geo-localization requires an accurately geo-tagged reference dataset. We argue that crowdsourced images can potentially serve as the reference data, but they suffer from significant inaccuracies in their geo-tags. Therefore, we present a novel method for refining the GPS-tags of images to alleviate this issue.

CHAPTER 6: ROBUST REFINEMENT OF GEO-LOCALIZATION USING RANDOM WALKS WITH AN ADAPTIVE DAMPING FACTOR

Thus far, we have assumed reliable geo-tags for the reference imagery are available. The coverage of accurate datasets, such as Street View, is limited (currently less than 25% of the countries in the world) which makes using crowdsourced images unavoidable for particular locations. However, user-uploaded images are well known to suffer from the acute shortcoming of having inaccurate geo-tags. In this chapter, we introduce the first method for refinement of GPS-tags which automatically discovers the subset of corrupted geo-tags and refines them. We employ Random Walks to discover the uncontaminated subset of location estimations and robustify Random Walks with a novel adaptive damping factor that conforms to the level of noise in the input.

We assume a dataset of GPS-tagged images with an unknown subset which includes inaccuracies with unknown statistical properties is given. We automatically discover the contaminated subset and adjust its GPS-tags to the correct locations using the rest of the images in the dataset (i.e no other resource of imagery or data is needed). We accomplish this task by generating a large number of estimations for the location of a particular image in the dataset based on the rest of the images therein. Then, we use Random Walks to identify the reliable estimations based on their pairwise consistency and use them for computing the refined GPS-tag. *Robustness* is the key trait of the proposed method. We show that our approach achieves good characteristics, such as high Breakdown Point or descending Influence Function [26], in this sense.

6.1 Robust Tag Refinement

The block diagram of the proposed method is shown in Fig. 6.1. Given a large dataset of images with contaminated geo-tags, first we perform content-based matching between the query image, which is one of the dataset images, and the rest of the dataset and retrieve a number of

matches. Then, a large number of image triplets comprised of the query and two matches are formed. We perform structure from motion on each triplet to estimate the relative camera locations and convert them to the global GPS coordinate system; each triplet yields one estimation for the location of the query. Since a considerable percentages of these estimations are inaccurate, we perform random walks on a graph defined on the triplet estimations to discover the accurate subset. The final estimation of the query's location is obtained by performing weighted mean of the accurate triplet estimations using the scores acquired from the random walk. The details of each step are provided in the following sections.

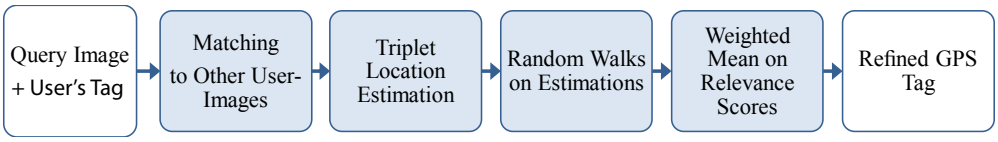

Figure 6.1: The block diagram of the proposed tag refinement method.

6.1.1 Generating Estimations using Triplets

We match the query image, $\mathcal{I}$, against the rest of the images in the dataset and retrieve μ matches $\{m_1, m_2, \ldots, m_\mu\}$. We use bag of words method with a vocabulary size of $50k$ along with the *tf-idf* voting scheme [17] for matching.

Next, $\binom{\mu}{2}$ image triplets using the query image and each possible pair of the retrieved matches are formed. We estimate the relative location of the query image with respect to the two matched images by finding the trifocal tensor and performing structure from motion [99, 100]. This operation is denoted by $\{\boldsymbol{l}_\mathcal{I}, \boldsymbol{l}_i, \boldsymbol{l}_j\} = \mathcal{SFM}(\mathcal{I}, m_i, m_j)$, where, $\boldsymbol{l}_\mathcal{I}$, $\boldsymbol{l}_i$, and $\boldsymbol{l}_j$ are the camera locations of $\mathcal{I}$, m_i and m_j in the image coordinate system, respectively. Note that the locations $\boldsymbol{l}_\mathcal{I}$, $\boldsymbol{l}_i$, and $\boldsymbol{l}_j$, which form a triangle, are typically three dimensional. However, the three vertices of a triangle always fall on some plane, and therefore, the dimensionality of $\boldsymbol{l}_\mathcal{I}$, $\boldsymbol{l}_i$, $\boldsymbol{l}_j$ can be reduced to

two (e.g. using PCA).

We want to have an estimation of the GPS-tag of $\mathcal{I}$ using the triplet. Therefore, the relative locations $\boldsymbol{l}_{\mathcal{I}}$, $\boldsymbol{l}_i$, and $\boldsymbol{l}_j$ should be transformed from the coordinates system returned by SfM (which is usually centered at one of the camera locations) to the global GPS coordinate system[1]. These two coordinate systems are related by a transformation consisting of rotation, translation and scaling:

$$\begin{bmatrix} \boldsymbol{g} \\ 1 \end{bmatrix} = (\boldsymbol{RST}) \begin{bmatrix} \boldsymbol{l} \\ 1 \end{bmatrix}, \tag{6.1}$$

where $\boldsymbol{l}$ is a point in the image coordinate systems and $\boldsymbol{g}$ is its corresponding point in the global GPS coordinate system; $\left[\begin{smallmatrix} g \\ 1 \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} l \\ 1 \end{smallmatrix}\right]$ are homogeneous coordinates of $\boldsymbol{g}$ and $\boldsymbol{l}$, respectively. $\boldsymbol{R}$, $\boldsymbol{S}$ and $\boldsymbol{T}$ denote the three by three rotation, scaling and translation matrices. At least two pairs of $\boldsymbol{g} \leftrightarrow \boldsymbol{l}$ correspondences are needed in order to calculate the $\boldsymbol{RST}$ transformation of equation 6.1. Since the two matches m_i and m_j are GPS-tagged, we use their GPS-tags and $\boldsymbol{l_i}$ and $\boldsymbol{l_j}$ to compute $\boldsymbol{RST}$ of the triplet. This transformation is then used for finding the location of $\mathcal{I}$ in the GPS coordinate system: $\left[\begin{smallmatrix} g_{\mathcal{I}} \\ 1 \end{smallmatrix}\right] = (\boldsymbol{RST}) \left[\begin{smallmatrix} l_{\mathcal{I}} \\ 1 \end{smallmatrix}\right]$. Since we have $\binom{\mu}{2}$ triplets, we will have $\binom{\mu}{2}$ different estimations for the GPS-location of the query using the described method.

6.1.2 Robustification using Random Walk

The estimation of the GPS-location of $\mathcal{I}$ which a triplet yields is accurate only if both of the parent reference images have accurate GPS-tags. Since we assume an unknown subset of the images in the dataset have inaccurate GPS-tags, a considerable number of the estimations are incorrect. However, the correct estimations are expected to show a high consistency with each other whereas the incorrect ones are more or less randomly distributed. Therefore, we use random walks

[1]Note that GPS locations are usually specified by Latitude and Longitude values which are in spherical coordinate system. However, they can be easily converted to a simple Cartesian system called East, North, Up (ENU). Therefore, for the sake of simplicity and without loss of generality, we assume all of the GPS coordinates in this chapter are in the Cartesian ENU system.

for discovering the reliable subset of estimations and assigning a score to each. Random walks have been applied to a wide range of problems, such as document retrieval or web image search [23, 24, 25]. A random walk is a special case of Markov chain with the property of *reversibility* which is essential for making the concept of "walks" on a graph meaningful. Intuitively, random walks diffuse the score of one node to the neighboring ones if they have a high consistency. This can be imagined by assuming a person is to walk from one node of a graph to another and count the number of times each node is visited; the probability of selecting the next node to travel to depends on a predefined consistency between the nodes. Hence, after a large number of walks, the nodes which are more consistent to one another are visited more often and consequently have a higher final relevance score.

We define the graph $G = (\boldsymbol{N}, E)$ where $\boldsymbol{N}$ and E represent the set of node and edges. Each node represents one triplet estimation, i.e. $\boldsymbol{N} = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_\lambda\}$, and there is an edge between each pair of nodes, $E = \{(\boldsymbol{g_i}, \boldsymbol{g_j}), i \neq j\}$. We include the original GPS-tag of $\mathcal{I}$ as an estimation for its GPS-location in $\boldsymbol{N}$ as well. Therefore, the number of nodes is usually equal to the number of estimations plus one[2]: $\lambda = \binom{\mu}{2} + 1$. The robustified probability of transition from node i to j is defined as:

$$p(i,j) = \frac{e^{-\sigma \|\boldsymbol{g}_i - \boldsymbol{g}_j\|_2}}{\sum_{k=1}^{\lambda} e^{-\sigma \|\boldsymbol{g}_i - \boldsymbol{g}_k\|_2}}, \tag{6.2}$$

where $\|.\|_2$ denotes the l_2 norm. The probability of transition from the node i to j is specified according to their GPS-distance; the closer the nodes, the higher likelihood for traversing from one to another. We set the insensitive parameter σ to 0.05 to reduce the transition probability between the nodes which are inconsistent by more than 60 meters to less than 5%. The denominator is a normalization factor which makes the summation of the transition probabilities from one particular node to all of the other ones equal to one.

[2]minus the number of triplets for which SfM failed to estimate $\boldsymbol{l}_\mathcal{I}$.

6.1.2.1 *Incorporating the Geo-density of images*

As discussed earlier, the user-shared images typically show a severely non-uniform geo-distribution; this characteristic can potentially result in a reduction in the accuracy of tag-refinement. To better understand this, consider the case where there exists a popular and unpopular photographed spots in the vicinity of each other. When performing image matching between the query and the dataset, more images from the popular spot are likely to be retrieved as more images from that location exist in the dataset. Consequently, there will be more triplet estimations coming from that spot and the final estimation of the random walk will be leaning towards the location suggested by the images of the popular spot.

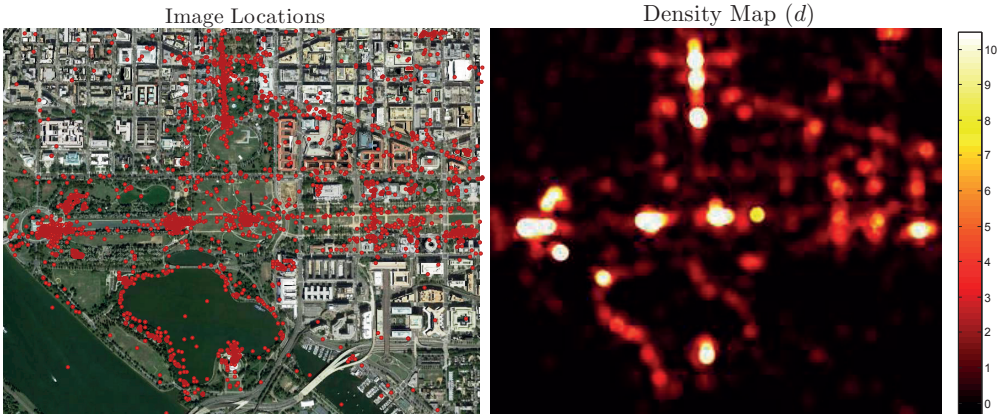

Figure 6.2: Left: the GPS-tags of images in a collection of user-shared images. Right: the corresponding geo-density map d.

In order to reduce the impact of this phenomena, we incorporate the density of the dataset in our random walk formulation. We define the initial score of the n^{th} node in $\boldsymbol{N}$ as:

$$v(n) = \frac{\frac{1}{d_i d_j}}{\sum_a \sum_b \frac{1}{d_a d_b}}, \tag{6.3}$$

where d_i and d_j are the geo-densities of the two reference images which generated the n^{th} triplet estimation. We define the geo-density, d, of an image as the number of other reference images within the radius r of it. The denominator is intended to fulfill the Markov chain requirement of $\|\boldsymbol{v}\|_1 = 1$. The geo-locations of a collection of user images from Washington D.C. are illustrated in Fig. 6.2-left; the corresponding density map (d) is shown on the right.

According to equation 6.3, the higher the densities of the parent images, the lower the initial value of the corresponding estimation. That is because a high density value implies many triplet estimations originated from the corresponding spot will be included in $\boldsymbol{N}$; therefore, their overall impact needs to be reduced to restrain them from dominating the rest of the estimations. To better understand why equation (6.3) helps in realizing this goal, consider the simplified case where there are two spots f and q with d_f and d_q images in their vicinity. The number of triplets formed by taking one image from spot f and another from spot q is $d_f d_q$. Therefore, by defining the initial value of an estimation as the inverse of this number, we constrain the total estimations originated from different spots to have equal values irrespective of the number of references images in their neighborhood.

In our experiments, we set the value of r and the initial score (before normalization) of the estimation corresponding to the initial GPS-tag to 5 meters and 1, respectively.

6.1.2.2 Adaptive Damping Factors

Having the node-to-node transition probabilities and the initial scores, the random walk updates the relevance score of one node at each iteration based on the probability of transmission from other nodes to it. Equation 6.4 is the formula of the basic random walk which performs this operation:

$$x_{(k+1)}(j) = \sum_{i=1}^{\lambda} \overbrace{\alpha}^{①} x_k(i)p(i,j) + \overbrace{(1-\alpha)}^{②} v(j), \tag{6.4}$$

where $x_k(i)$ is the relevance score of the i^{th} node at the k^{th} iteration. The argument of summation (left term) is the part which computes the probability of transition from other nodes to a particular one, and the right one is a damping term. The damping term was added to the random walk in order to enable leveraging the prior knowledge about the relevance of nodes and to ensure *irreducibility* of the transition probabilities matrix which is a convergence condition for the random walk [25, 23]. α is a constant usually set to a value between 0.8 and 1. The summation of the terms ① and ② in equation 6.4 has to be equal to one since the summation of the relevance scores at any iteration has to be equal to one: $\sum_{i=1}^{\lambda} x_k(i) = 1$.

A careful look at equation 6.4 reveals an important characteristic of the basic random walk: the updated relevance scores always include $(1-\alpha)$ of the initial scores. That means $(1-\alpha)$ of the initial score of a node appears in the final relevance score regardless of its consistency with the rest of the nodes. This is undesirable particularly when the set of node could include outliers with arbitrary inaccuracies, as it essentially signifies a fixed portion of the input noise will always appear in the output results; a disadvantage which violates the basic requirements of a robust system. Therefore, we propose to use random walks with a damping factor which adaptively changes according to the consistence of each node to the others. We accomplish this goal by making the damping term of a node a function of its relevance score at each iteration:

$$x_{(k+1)}(j) = \frac{1}{\eta}\Big(\sum_{i=1}^{\lambda} \overbrace{\Big(1-(1-\alpha)x_k(j)\Big)}^{①} x_k(i)p(i,j) + \overbrace{(1-\alpha)x_k(j)}^{②} v(j)\Big). \tag{6.5}$$

Equation 6.5 is equivalent to equation 6.4 with the difference that the damping term (②) is proportional to the relevance score of the node; therefore, the amount of contribution from the initial score of the node depends on its so-far consistency with the other nodes. Hence, an arbitrarily noise in the input can be handled as the input error does not directly propagate in the output. In the context of our problem, we will show (in section 6.2.3) that the random walk with the adaptive

damping factor can handle GPS-location estimations ($\boldsymbol{g}_i$) with arbitrarily large errors while the basic random walk fails to do so by directly passing the input noise to the output.

Similar to equation 6.4, the term ① in equation 6.5 is equal to (1-②). η given below is a normalization constant to make the summation of relevance scores at all iterations equal to one:

$$\eta = \sum_{j=1}^{\lambda} \Big(\sum_{i=1}^{\lambda} \Big(1 - (1-\alpha)x_k(j)\Big)x_k(i)p(i,j) + (1-\alpha)x_k(j)v(j)\Big). \tag{6.6}$$

The matrix form of the random walk with the adaptive damping factor (i.e. equation 6.5) can be derived as:

$$\boldsymbol{x}_{(k+1)} = \frac{1}{\eta}(\boldsymbol{x}_k \boldsymbol{\Gamma} \boldsymbol{P} + \boldsymbol{v}(\boldsymbol{I} - \boldsymbol{\Gamma})), \tag{6.7}$$

where

$$\boldsymbol{\Gamma} = diag(1 - (1-\alpha)\boldsymbol{x}_k). \tag{6.8}$$

$\boldsymbol{x}_{(k)}$ and $\boldsymbol{v}$ are $1 \times \lambda$ dimensional vectors of the relevance scores at the iteration k and their initial scores respectively. $\boldsymbol{P}$ is a $\lambda \times \lambda$ matrix which has the pairwise transition probabilities as defined in equation 6.2. $diag(.)$ is an operator which generates a diagonal matrix where the elements on the main diagonal are the elements of the argument vector and the rest of the elements are set to zero. Also, the simpler matrix form of the normalization constant, η, can be written as $\eta = \|\boldsymbol{x}_k \boldsymbol{\Gamma} \boldsymbol{P} + \boldsymbol{v}(\boldsymbol{I} - \boldsymbol{\Gamma})\|_1$.

Notice the similarity between equation 6.7 and the matrix form of the basic random walk: $\boldsymbol{x}_{k+1} = \alpha \boldsymbol{x}_k \boldsymbol{P} + (1-\alpha)\boldsymbol{v}$; the main difference is that the damping factor matrix $\boldsymbol{\Gamma}$ is adaptively changing at each iteration as compared to being set to a constant.

The relevance scores are iteratively computed until they converge to the final values $\boldsymbol{x}_\pi$, commonly termed as "stationary probability". Therefore, the vector $\boldsymbol{x}_\pi$ includes the final relevance scores of all of the GPS-location estimations.

6.1.2.3 *Final Tag Estimation using Random Walk Scores*

The relevance scores $\boldsymbol{x}_\pi$ acquired from the random walks are the results of diffusing the pairwise consistencies of the GPS-location estimations as well as their initial scores. The estimations which are severely affected by noise are expected to have ≈ 0 scores, and the other estimations gain scores based on their agreement with the other nodes. Hence, we computed the refined GPS-location of the query, $\mathcal{I}$, utilizing a weighted mean using the scores $\boldsymbol{x}_\pi$:

$$\hat{\boldsymbol{g}} = \sum_{i=1}^{\lambda} \boldsymbol{g}_i x_\pi(i), \tag{6.9}$$

where $\hat{\boldsymbol{g}}$ is the refined GPS-location.

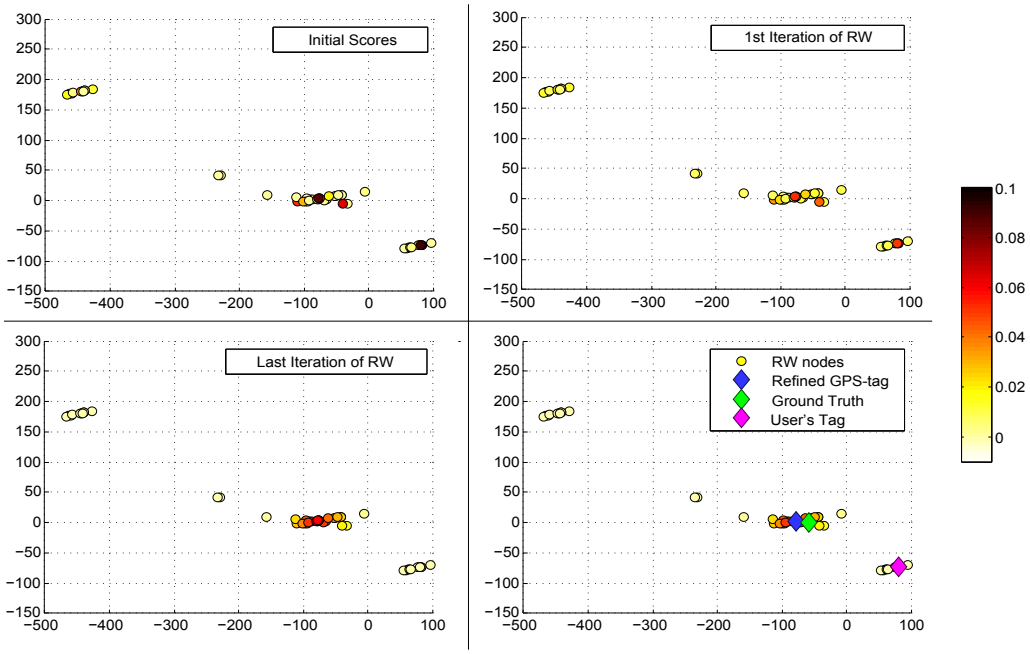

Figure 6.3: The process of the random walk shown for a sample query in the ENU coordinate system. The initial scores based on geo-densities along with the relevance scores after the first and the last iterations, as well as the final estimation are illustrated.

The process of the random walk is illustrated in Fig. 6.3 where the initial scores (based on geo-densities) and the relevance scores after the first and the last iterations are demonstrated. The refined GPS-tag along with the initial tag and the ground truth are shown as well. Notice that the estimations which are far from the correct location are successfully identified by the random walk relevance score.

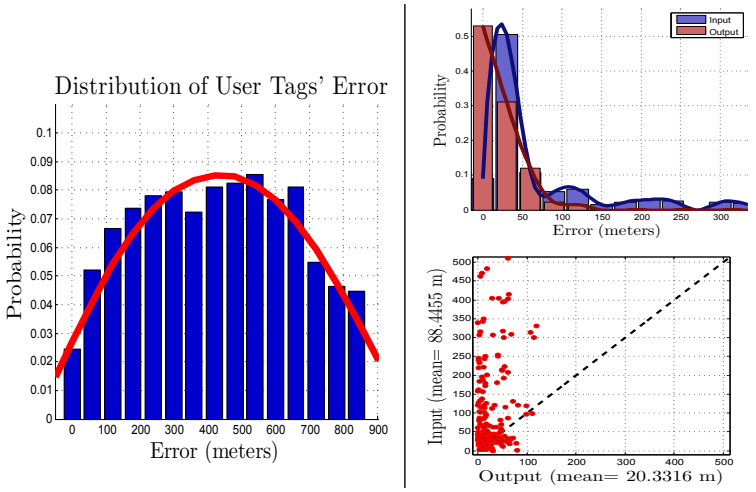

Figure 6.4: Left: the distribution of the error in the user-specified GPS-tags of 8127 images. It shows a near-Gaussian distribution with the mean and standard deviation of 425.6 and 228.0 meters, respectively. Right: the results of tag refinement when no additional contamination is added.

6.2 Experimental Results

We performed our evaluations on a mixed dataset of 18,075 GPS-tagged user-shared images from the cities of San Fransisco, CA; Pittsburgh, PA and Washington, DC. The images were downloaded from Panoramio, Flickr and Picasa and were all captured and GPS-tagged by users.

6.2.1 *Statistical Properties of Error in User Tags*

Even though existence of inaccuracies in the users' GPS-tag was acknowledged in several previous works [62, 2, 15], formal large scale statistics on the amount of noise in such tags has not been reported so far. Therefore, we manually verified the accuracy of the GPS-tags of 8,127 images captured in Pittsburgh. We found that, depending on the resource website, typically about 10.2% to 30% of the user shared images have inaccurate tags (Panoramio showed the least and Picasa had the most percentage of inaccurate tags). By "inaccurate", we mean an image whose GPS-tag has an error more than 15-30 meters which is the nominal accuracy of the commercial GPS devices. The inaccuracy inthe GPS-tags are mainly due to manual tagging, WiFi Positioning System, localization using cell phone network signals or a weak GPS signal. Fig. 6.4-left shows the distribution of the error of the inaccurate GPS-tags in the annotated set. It shows a near Gaussian distribution with the mean and standard deviation of 425.6 and 228.0 meters. We focus on the errors less than $1km$ in Fig. 6.4-left, as the larger values seem to significantly correlate with the layout of the city and consequently fail to generalize.

6.2.2 *Tag Refinement Results*

Figures 6.5 and 6.6 illustrate the results of tag refinement using the proposed method for different amounts of error in the user tags. In order to investigate the performance of our method under various scenarios, we added random Gaussian noise with the mean values of 100, 200, 500, 3,000 meters to 5, 10, 20, 33 and 50 percents of the 18075 images in our dataset; the standard deviation was set to 0.5 of the mean to replicate the user tag errors (see section 6.2.1). Note that these errors are on top of the already existing noise in the user specified tags in our dataset; therefore, the additional contamination determines the lower bound of noise since the exact amount of error in the dataset is unknown as the ground truth location of all of the 18075 images are not known. As the test set, we selected a random subset of 500 images from the dataset and accurately

annotated their ground truth location (with an error < 10 meters). We refined the GPS-tags of the test set images using the rest of the images in the dataset and compared the refined location against the ground truth to find the refinement error. The query images which returned less than 5 matches from the rest of the dataset and the ones for which SfM failed to generate at least 9 estimations were removed from the test as they typically correspond to either isolated images or panoramic/edited ones. We also made sure the query images were among the ones with contaminated GPS-tags to ensure the evaluation is fair and challenging enough.

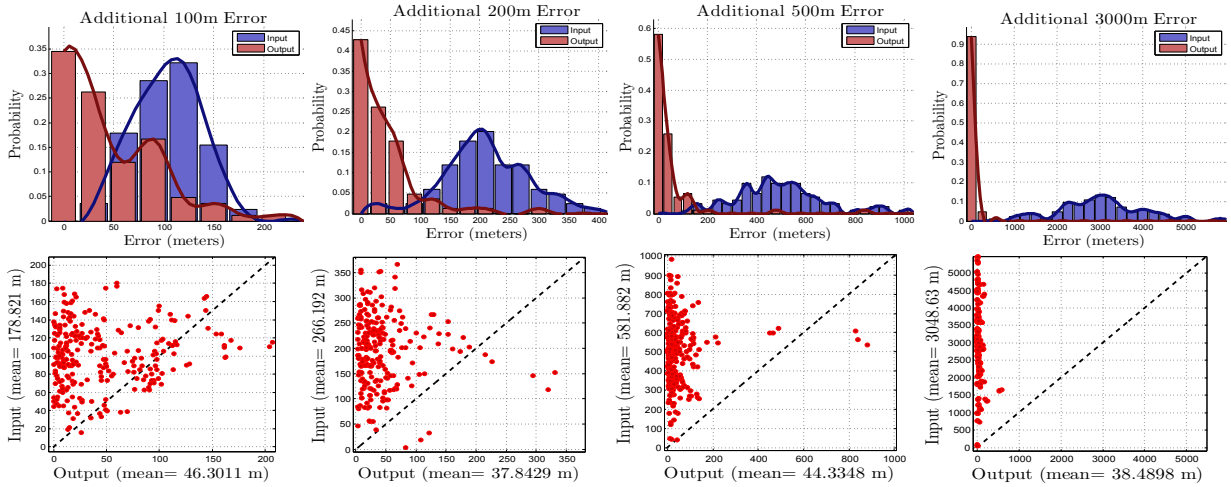

Figure 6.5: The performance of the proposed tag refinement method for various contaminations with the mean values of 100, 200, 500 and 3,000 meters. The distributions and scatter plots are shown on the top and bottom rows respectively. Notice the significant improvement across various amounts of noise in the input.

Fig. 6.5 shows the results of this experiment for the additional contamination percentage of 20% with various error values; the distributions of error in the input and output are shown on the top row, and the scatter plots of the error in which each point represents one query image are illustrated in the bottom. As apparent in both of the distributions and scatter plots, our method significantly

refines the GPS-tags as it has a substantially smaller error than the input tags. Fig. 6.4-right shows the tag refinement results when no additional contamination was added to the dataset; the input error is the inaccuracy of the user tags.

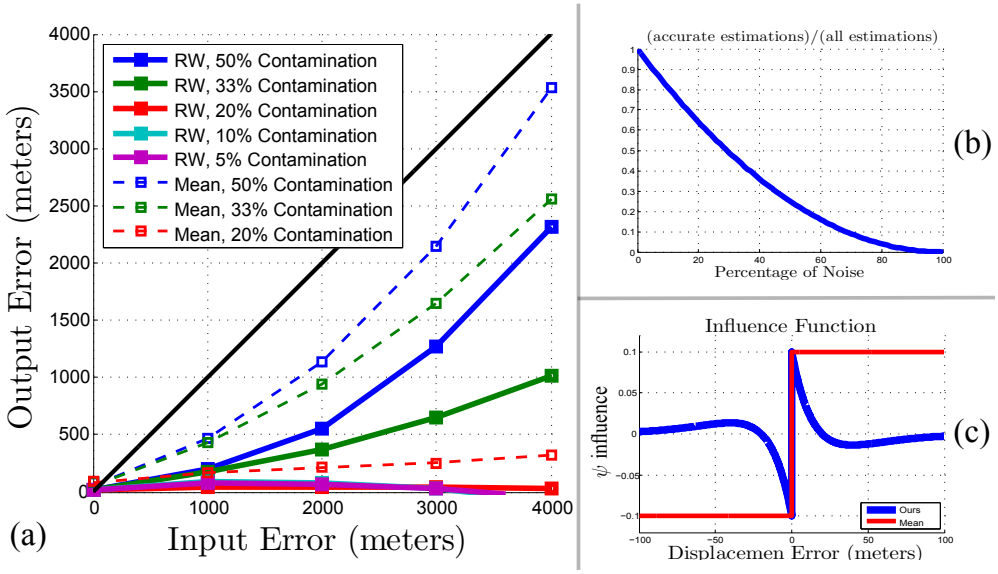

Figure 6.6: (a): The overall performance of the proposed tag refinement method for various values of the mean and percentage of contamination. (b): the ratio of the accurate estimations over the total number of estimations with respect to the percentage of contamination. (c): the Influence Function of our method and the baseline (averaging).

Fig. 6.6 (a) shows the mean of the output error for various values of mean and percentage of contamination in the input tags. Two observations can be made in the Fig. 6.6 (a): first, for the contamination percentages less than 30%, our method nearly eliminates the error regardless of the mean of the contamination in the input; that's why the error curves for the contamination percentages of 5, 10 and 20 are almost flat. However, when the percentage of error increases to beyond 33% and 50%, the output error becomes noticeable, yet it is considerably less than the error in the input. This observation is consistent with the bases of our method as the ratio of the number

of estimations not affected by noise over all of the estimations is $\binom{n-nq}{2}/\binom{n}{2}$ where q and n are the percentage of noisy tags and total number of images in the dataset, respectively. This ratio is illustrated in Fig. 6.6 (b); as apparent in the plot, when the percentage of contamination goes beyond 30% and 50%, the percentage of estimations affected by noise increases to over 50% and 75%, respectively, and therefore, it becomes excessively difficult to discover the right GPS-tag.

Also, Fig. 6.6 (b) justifies why we used images *triplets* for generating the estimations and not quadruplet or quintuplet; the ratio of 6.6 (b) would drop with a sharper slope if more images were used for generating an estimation which is obviously undesirable. Hence, we used an image triplet which is the smallest number of images needed for removing the scale ambiguity and converting the image coordinate system to the global GPS's as discussed in section 6.1.1. The broken lines in 6.6 (a) illustrate the results of using the average of the triplet estimations as the refined GPS location (i.e. bypassing the random walk and using uniform mean). Unlike the random walk results, the output error curves of all contamination percentages are always monotonically increasing, which shows the input error is propagated to the output.

The part of the curves in Fig. 6.6 (a) which corresponds to large errors show the high empirical *Breakdown Point* [26] (defined as the resistance of a method against the proportion of inaccurate observations in the data in robust statistics) of our estimator. Additionally, the *Influence Function*, which is a measure of the dependence of an estimator on the displacement error of one observation [26], of our method has the favorable *descending* shape (see Fig. 6.6 (c)). That means a sample with an arbitrarily large error have a small impact on our final estimation whereas it has an unbounded effect on the results of non-robust methods such as the baseline (mean).

6.2.3 Evaluation of the Adaptive Damping Factor

Fig. 6.7 shows the evaluation of the proposed adaptive damping factor compared to the constant one. The curves on the right illustrate the mean error in the output of the random walk with the constant (i.e. equation 6.4) and adaptive damping factors (i.e. equations 6.5, 6.7). On

the left, the distributions and scatter plots of the error in the input and output for different values of α are shown; the contamination has the mean and percentage of 3000 meters and 20% in this experiment. The value of α determines the contribution of the initial scores in the final relevance scores; the green curve signifies the error of constant damping factor increases with increasing α while the error of adaptive damping factor remains nearly flat. That means the adaptive damping factor successfully prevents the noise in the input from being directly propagated in the output even for the large values of α, while constant damping factor is not suitable in the presence of noise.

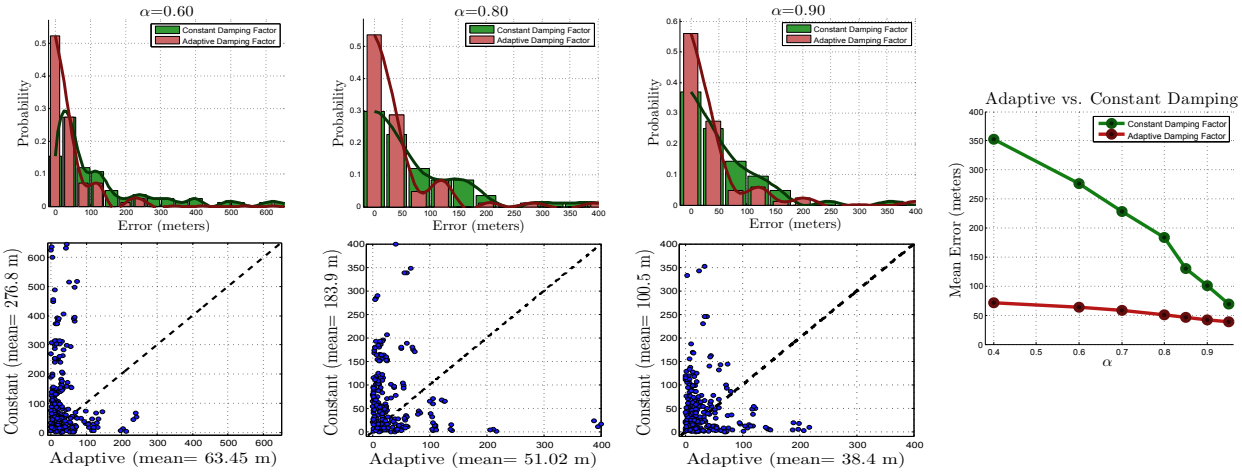

Figure 6.7: Evaluation of the proposed adaptive damping factor. The curves on the right compare the performance of the adaptive damping factor compared to constant damping for different values of α; the corresponding distributions and scatter plots are shown on the left. The mean and percentage of contamination are 3000 meters and 20% respectively.

The table 6.1 provides the performance of utilizing the geo-density (equation 6.3) as the initial score compared to using uniform initial scores for various contamination and α values. As apparent in the table, for almost all values of α and input contaminations, the geo-density yields better output error compared to the uniform score (except for the case of 300 meters where the

performance of both methods are ≤ 1 meter different.); the improvement made by density handling is more noticeable in large errors. The red numbers show the best performance for each value of error. The α values between 0.80 and 0.90 typically yield the best results where lower values work better for lower errors and vice versa; that's because in lower contaminations, the initial scores are more accurate and consequently increasing their influence boosts the performance. Since we can make no prior assumptions about the mean of error, we set the value of α to 0.90 in all of our experiments which works satisfactorily for both small and large errors in all of our experiments.

Table 6.1: Evaluation of the density handling method for various values of α and contamination means. Density and Uniform represent setting the initial scores based on the geo-density or uniform scoring. The bold numbers show the best performance for a particular value of α and contamination means, while the red ones show the best overall performance for a particular contamination mean.

		Input Error (m)							
		100m		300m		3000m		5000m	
		Density	Uniform	Density	Uniform	Density	Uniform	Density	Uniform
α	.95	**35.1**	35.2	35.8	**35.6**	**38.6**	38.8	**39.4**	43.1
	.90	**34.0**	35.2	**34.8**	35.2	**38.4**	42.1	**43.6**	52.1
	.80	**33.8**	34.0	36.0	**35.2**	**51.2**	52.1	**51.6**	73.5
	.60	**34.7**	34.9	37.8	**36.8**	**63.45**	65.1	**67.4**	101
	.40	**36.4**	36.5	38.8	**37.8**	**67.5**	72.4	**81.4**	118

Bear in mind that our query set is a subsample of our large dataset; the improvement made by density handling would be even more significant if the test set had a significantly different distribution from the rest of the dataset.

So far, we generated the estimations, $\boldsymbol{g}_{\cdot}$, using SfM while one could use the GPS-tags of the images matched to $\mathcal{I}$ as the estimations for its location. However, that would imply we assume the dataset is dense enough to the point that there exist similar images in the dataset with camera locations very close to the one of $\mathcal{I}$. Otherwise, performing the tag refinement using the matched images' GPS-tags would achieve a limited success whereas SfM wouldn't have the requirement of having images with near-identical camera locations to $\mathcal{I}$. In order to empirically investigate this, we performed an experiment in which we used the SfM and GPS-tags of the matched images as the estimations $\boldsymbol{g}_{\cdot}$ in our framework. The scatter plot in Fig. 6.8-left illustrates the results for the contamination with the mean value and percentage of 3000 meters and 20%.

6.2.4 Refinement using Image Geo-tags (No SfM)

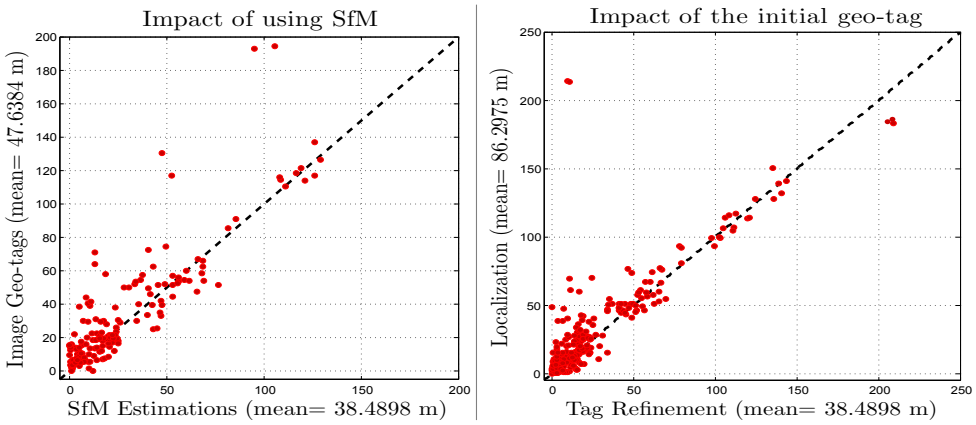

Figure 6.8: Left: The scatter plot showing the effect of using SfM for generating the estimations as compared to directly using the GSP-tags of the matched images as the estimation. Right: The impact of the initial GPS-tag in the overall results (i.e. localization vs. tag-refinement mode).

As expected, using SfM improves the overall accuracy by 9.2 meters. However, bypassing the SfM has some advantages such as lowering the complexity or increasing the number of esti-

mations due to the typical high rate of failure of SfM, which could make it desirable in certain scenarios such as when a fine error in the results is acceptable.

On an 8 core 2.4 GHz machine running MATLAB, our framework, excluding performing SfM on triplets, runs in 0.04 seconds per image. The main reasons behind the high efficiency of our method is the fast convergence characteristic of random walks and forming the graph of estimations in a local manner (i.e. separate for each query). Under realistic circumstances, the user images may not be dense enough to enable forming a graph which expands over the whole dataset [101], unless the area is heavily pictured. Hence, building such large graphs is impractical for many image collections and often unnecessary.

6.2.5 Tag refinement vs. Localization

We used the original GPS-tag of the query image in our framework in order to refine the tag. However, our approach could be viewed as an image geo-localization method *in the presence of noise* if the initial geo-tag wasn't leveraged. The scatter plot of Fig. 6.8-right shows the results of an experiment on the overall impact of the original GPS-tag in the process of estimating the true GPS-tag (i.e. tag-refinement vs. localization mode). The mean and percentage of contamination are 3,000 meters and 20% respectively, while we made sure the initial GPS-tags are *not* contaminated in this experiment as we are investigating their impact. As one would expect, utilizing the initial GPS-tag leads to better results as it is an additional cue to the right location of the query; this additional estimation could become essential particularly for the images for which few matches were retrieved from the dataset or few estimations were generated using SfM.

However, the mean of the output error in localization mode is limited to 82.2 meters while 20% of the dataset images have the mean contamination of 3,000 meters. This confirms our method preserves its robustness trait in the localization mode as well and can be used for geo-localization purpose when the reference dataset includes unknown inaccuracies. Since the majority of current image localization methods [14, 2, 15] do not have an internal mechanism for dealing with noisy

tags in their reference dataset (i.e. the noise in input will directly propagate to the output as those methods typically rely on the GPS-tags of one or few best matching reference images), no direct comparison with existing geo-localization methods would be fair.

6.3 Chapter Summary

In this chapter, we argued that crowdsourced images play a key role in various applications while the suffer from the significant shortcoming of having inaccurate GPS-tags. We developed the first method for refinement of the GPS-tags of crowdsourced images. Given a large dataset of GPS-tagged images with an unknown subset with inaccuracies, we discovered the contaminated subset and adjusted the GPS-tags to the correct locations. This was done by performing image matching, generating location estimations using SfM on image triplets, performing random walks in order to identify the subset with the maximal agreement, and a weighed averaging of the consistent estimations. We developed an adaptive damping factor for random walks and leveraged the geo-density of images in order to minimize the induced bias in the results. The experiments evaluated various aspects of the method and showed it performs consistently robust across different scenarios with superior results. In the next chapter, we discuss how the geo-location of an image can be used for improving the understanding of its content. In this context, we discuss a *location-aware* framework for recognizing storefronts in images.

CHAPTER 7: BUSINESS RECOGNITION USING LOCATION-AWARE IMAGE UNDERSTANDING

In location-aware image understanding, we are interested in improving the image analysis by putting it in the right geo-spatial context. This approach is of particular importance as the majority of cameras and mobile devices are now being equipped with GPS chips. Therefore, developing techniques which can leverage the geo-tags of images for improving the performance of traditional computer vision tasks is of particular interest. In this chapter, we present a location-aware multimodal approach which incorporates business directories, textual information, and web images to identify businesses in a geo-tagged query image.

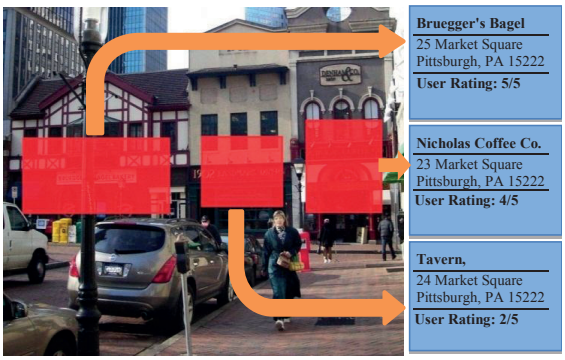

Figure 7.1: A business recognition system can automatically identify businesses in an image and provide the user with additional information such as the addresses of the businesses, ratings and reviews.

Providing smartphone or wearable computer users with extensive information about a particular business of interest in an automatic and convenient fashion is becoming very important. A business recognition system can be used for establishing a link between the massive available visual data from businesses, such as StreetView imagery, and the rich resource of business direc-

tories as well. Such system can also enhance the user experience in surfing maps and the accuracy of business listings and geographical databases.

Our method utilizes multimodal information obtained from both visual content, such as storefront appearance and text, and non-visual information, such as GPS and business directories, in order to achieve accurate results. We show that even though none of the above sources of information result in a desirable outcome when used individually, the proposed method achieves a notable rate of success by combining them.

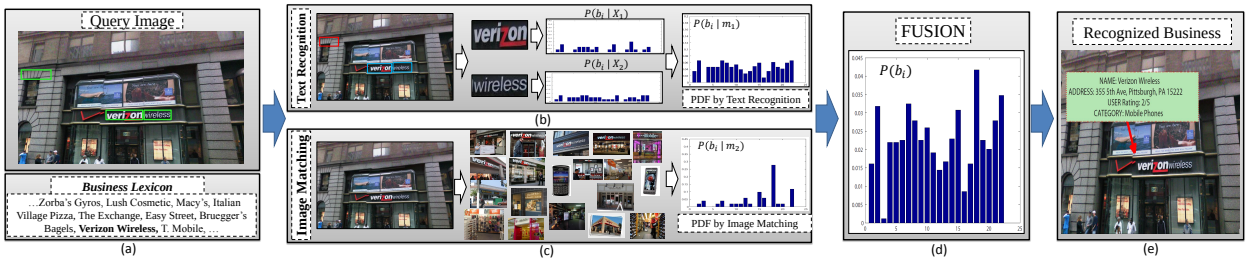

Figure 7.2: The block diagram of our method. (a) shows the query image, detected text and business lexicon. (b) illustrates the process of computing the PDFs of different words and marginalizing them into one PDF. (c) shows the query image, a subset of web images ordered based on how well they match the query, and the resulting PDF from image matching. (d) demonstrates the PDF obtained from the fusion process. (e) shows the business which achieves the highest probability after fusion, as the recognized business.

7.1 Framework Overview

The block diagram of the proposed framework is shown in Fig. 7.2. The images captured using smartphones are usually associated with a coarse location tag. The tag typically comes from the inbuilt GPS-chip, cell tower signal or WPS. We use this approximate location for generating a

list of nearby businesses by querying business directories.

In order to utilize the textual information (subsection 7.1.1), we employ a text detection method to identify the areas of the image which may contain text. Then, we apply a multi-hypotheses text recognition approach assisted by the business lexicon which yields a PDF specifying how well a detected word in the query image matches the nearby businesses. Since the business in the query image may include several words, we combine the PDFs of matching businesses to each word through marginalization in order to have a single PDF which represent the textual information in the whole query image.

In order to leverage the images on the web in business recognition, we use the list of nearby business names as search keywords and collect a set of images from the web for each one. The query image is expected to share some similarity with the web images of the business which is visible in it. Therefore, we match the query image to the collected web images in order to identify the similar ones (subsection 7.1.2). This process yields a PDF which represents how well the web images of each nearby business match the query image. Finally, we combine the two PDFs acquired from text processing and image matching in a probabilistic late fusion step to compute a PDF which utilizes both modalities (subsection 7.1.3).

Generating the Business Lexicon: We use the APIs of Yellow pages and Yelp to automatically retrieve and aggregate the nearby businesses within the distance of 150 meters to the approximate location. Regarding the inaccuracies in the business directories and the coarse location of the query, we set the radius to a large value to make sure the visible businesses in the query are present among the retrieved results. $B = \{b_i | 1 \leq i \leq n_B\}$ represents the set of retrieved businesses where n_B denotes the number of nearby business. A business name may include more than one word, so $W = \{w_{i,j} | 1 \leq i \leq n_B, 1 \leq j \leq n_w(i)\}$ is the set of words in the name of all nearby businesses. $w_{i,j}$ represents the j^{th} word of i^{th} business's name. $n_w(i)$ denotes the number of words the name of i^{th} business includes.

7.1.1 Business Recognition Using Textual Information

Business recognition using textual information is inherently similar to the problem of text recognition in natural scene. However, the goal of business recognition is to establish a relationship between the reference businesses and the text in the query image and not necessarily recognizing it. Such relationship can be probabilistic or fuzzy, while text recognition aims at recognizing the text deterministically. Additionally, scene text recognition does not address other problems specific to business recognition such as combining the information obtained from different query words in order to perform the recognition of a single business. We employ the text processing method described in the rest of this section which is specifically customized for the task of business recognition and addresses the aforementioned issues. Additionally, it makes representing the matching results in a probabilistic manner feasible, as such representation is required in our fusion process.

Multi-hypotheses Character Recognition: We use Stroke Width Transform (SWT) [102] as our text detection method which identifies the regions of the image which might contain a word and each character therein. We use Gabor features for performing text recognition [69] on each character patch. In our training step, we generate 62 synthetic character patches comprised of lower and upper case English alphabet along with single digit numbers using the font `Arial`. Additionally, we compute six variations for each character using four consecutive image dilation and two erosions as we observed that the business signs in natural scenes tend to significantly vary in the width of characters compared the standard fonts. we apply a bank of 108 Gabor filters comprised of $n = 6$ frequencies and $m = 18$ scales to each synthetic character. Each character is then divided into 9 sub patches using a 3 by 3 grid. The Gabor feature of each sub patch is defined as the mean of Gabor features of the pixels therein. Therefore, each character is represented by a 972 dimensional vector which is reduced to 50 dimensional using PCA. The feature vectors of all 62 characters and their erosion-dilation variations form our reference set of character features.

During the test step, the same 108 Gabor filters are applied to a character patch returned by

text detection and the size of the feature vector is reduced to 50 using the mapping matrix found by PCA during the training. Then we use a k-nearest neighbor classifier to find the most similar k reference characters to the query patch. In other words, instead of assigning one character to the query patch, we nominate k characters as the possible matches. We employ this approach as the right character may not necessarily be the first match, while it usually appears among the top few matches. This is shown for the sample query word "**verizon**" in Fig. 7.3-left (b).

We show a feasible permutation of the candidates for a query word by $X = \{\chi_a^1, \chi_b^2, \chi_c^3, \ldots\}$, which means the a^{th} candidate for the first query patch, the b^{th} candidate for the second query patch, and so on are selected. Therefore, each query word possesses a large number of feasible permutations of its character candidates. Eight feasible permutations for a sample query are shown in Fig. 7.3-right (c).

Matching Character Permutations to Business Lexicon: We solve the following optimization problem to identify the best permutation which matches a particular business word in the lexicon:

$$\hat{X}_{i,j} = \underset{X}{\operatorname{argmin}} ||X - w_{i,j}||, \tag{7.1}$$

where $\hat{X}_{i,j}$ represents the permutation which best matches the business word $w_{i,j}$. $||.||$ represents Levenshtein distance between two strings. We solve equation 7.1 once for every word in the business words lexicon W in order to find the best matching permutation to each. This process is illustrated in Fig. 7.3-right for a sample case. (c) shows five sample words in the business lexicon, w, their respective matching permutations, $\hat{X}$, and the edit distance between them.

Bear in mind that the name of one business may include more than one word. Therefore, we solve the following equation in order to find the best matching business word to the query for each nearby business:

$$\zeta(b_i) = \min_j ||\hat{X}_{i,j} - w_{i,j}||, \tag{7.2}$$

where b_i represents the ith businesses among the nearby businesses B. $\zeta(b_i)$ is the Levenshtein

distance between the query word and the best matching word in the name of business b_i. Therefore, ζ can be interpreted as a distance function which represents how well business b_i matches the query word.

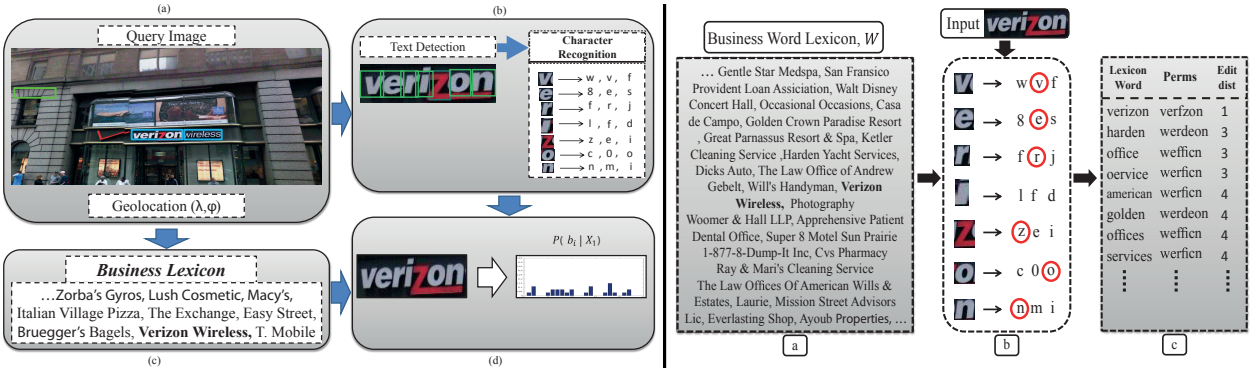

Figure 7.3: **Left:** The process of matching a detected word in the query image to nearby businesses. (a): query image along with the detected text bounding boxes. (b): the nominated candidates for each character of the query word "verizon". (c): the list of nearby businesses. (d): the PDF specifying the probability of the query word X_1 matching each of the nearby businesses. **Right:** Illustration of the process of multi-hypotheses matching (equation 7.1). (b): the query word and nominated candidates for each query patch. The correct candidates are marked with red circles. (c): best matching permutations to each business word and their respective edit distance.

We would like to have a PDF which specifies a probability for each of the nearby businesses matching the query word represented by X. Therefore, the distances function $\zeta(b_i)$ is converted to a PDF using the following equation:

$$p(b_i|X) = \frac{sig(\zeta(b_i))}{\sum_i sig(\zeta(b_i))}, \tag{7.3}$$

where $p(b_i|X)$ is the probability of the business b_i to match the given query word X. sig is the

sigmoid function with the standard form $sig(x) = \frac{1}{1+e^{-\tau x}}$, where τ is a constant which we set to -0.5 in our experiments. Therefore, a large edit distance corresponds to a small probability and vice versa.

Utilizing multiple words for recognizing a Business: The probability distribution function $p(b_i|X)$ acquired from equation 7.3 specifies how well the nearby businesses match one query word. However, the business sign in the query image may include more than one word. Therefore, we need to associate the query words pertaining to one business in order to utilize all of them for recognizing the respective business. Usually the words which belong to one business in the image are spatially close and have similar appearance features. For instance, the words "**verizon**" and "**wireless**" in Fig. 7.2 (a) have similar colors and are located next to each other. Therefore, for each bounding box acquired from the text detector, we form a feature vector by concatenating its RGB color histogram with (x, y) spatial location of its center. Then we perform mean shift clustering on the feature vectors of all the bounding boxes to associate the words which belong to one business. The number of resulting clusters is the number of businesses in the query image, and the elements in each cluster are the bounding boxes associated together. A sample case is shown in Fig. 7.2 (b) where the bounding boxes shown in the same color are associated together.

In order to leverage the associated query words in the business recognition process, we combine the PDFs each one yields through marginalization:

$$p_t(b_i) = \sum_{j=1}^{\alpha} p(b_i|X_j)p(X_j), \tag{7.4}$$

where $p(b_i|X_j)$ is the PDF obtained from equation 7.3 for the query word X_j, and α is the number of associated query words. $p(X_j)$ is the probability of looking at the j^{th} query word for recognizing its respective business. We treat all the query words of one business sign equally by assigning equal chance to all of them: $p(X_j) = 1/\alpha$.

$p_t(b_i)$ in equation 7.4 specifies the probability of each nearby business being visible in the

query image based on the entire textual information in the query. In order to avoid confusing the PDFs obtained using text processing, image matching and fusion, we define $M = \{m_1, m_2\}$ as the set of approaches to business recognition which we employ. m_1 and m_2 represent text recognition and image matching respectively. Therefore, $p(b_i|m_1)$ represents the PDF obtained by employing text recognition which is equal to $p_t(b_i)$ of equation 7.4. Fig. 7.2 (b) illustrates the described process for a sample query.

7.1.2 Business Recognition by Image Matching

Nowadays, for most of the businesses in urban area a number of images which show the storefront can be found on the web. Such images are typically uploaded by customers, business owners, or business directories for both franchise and non-franchise businesses.

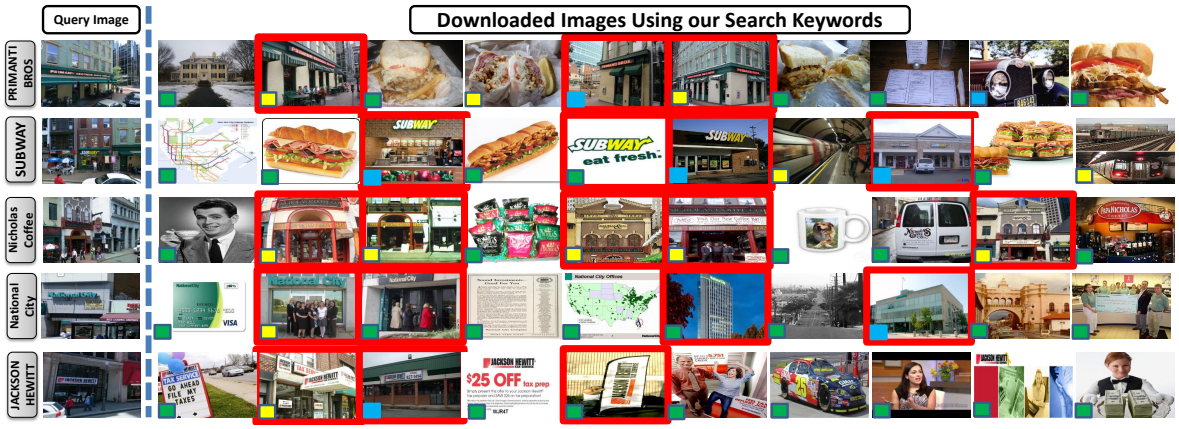

Figure 7.4: Sample web images for five businesses. The red margin marks the positive examples. Green, yellow and blue markers denote the keywords `'business name'`,`'business name+city'` and `'business name+storefront'` respectively.

In order to find the web images which pertain to a particular business, we generate four search keywords for each nearby business as: `'business name'`, `'business name+city'`

and `'business name+storefront'`. We use the keywords to search for images on the web and download the retrieved ones using Ajax-based web image crawling. We save about ten images per keywords which results in forty images for each nearby business. We view the set of downloaded images as a reference dataset that each image therein is associated with a nearby business. Fig. 7.4 shows ten sample web images retrieved for five different businesses.

We employ bag of visual words (BoVW) model for matching the query image to the set of web images. We extract SIFT features from the web images and the query and compute their histogram of visual words using a vocabulary with 2000 words. The vocabulary is pre-computed on a set of 10000 random images. We employe tf-idf weighting scheme which reduces the contribution of less discriminative visual words [17].

We find the most similar web image of a business to the query using:

$$\psi(b_i) = \min_j |h_q - h_{i,j}|, \tag{7.5}$$

where h_q and $|.|$ represent the $BoVW$ histogram of the query, and L_2 distance respectively. $h_{i,j}$ represents the histogram of the j^{th} web image of the i^{th} business. Equation 7.5 identifies the most similar image to the query for each nearby business. Therefore, the distance function $\psi(b_i)$ specifies how well the nearby business, b_i, matches the query based on the web images.

Using a method similar to the equation 7.3 which was intended to convert edit distances to probability values, we convert the image matching distance function $\psi(b_i)$ to a PDF using, $p(b_i|m_2) = \frac{sig(\psi(b_i))}{\sum_i sig(\psi(b_i))}$,where $p(b_i|m_2)$ represents the probability of recognizing the business b_i in the query given the employed approach is image matching.

The procedure of downloading web images and computing their BoVW representation is relatively time consuming. However, since all the businesses in the broad area of interest, e.g. a city, are known, the web images can be downloaded and processed in an offline manner. That way, performing image matching between query and the web images of its nearby businesses can be

done almost instantaneously.

7.1.3 Fusion of image matching and textual info

The purpose of the fusion step is to unify the information obtained from the two methods of text recognition and image matching to perform a more robust business recognition. Theoretically, the law of total probability is utilized for finding the probability of one event when it coincides with a random variable, so we employ it in fusing the PDFs acquired from text recognition and image matching. In our problem, the event is a nearby business, b_i, and the coinciding variable is m_i:

$$p(b_i) = p(b_i|m_1).P(m_1) + p(b_i|m_2).P(m_2) \tag{7.6}$$

where $P(m_1)$ and $P(m_2)$ are the probability of employing text recognition and image matching respectively. We define these two values using a training set of 50 query images. The training set consists of queries for which *only one* of the two methods worked successfully. We define $P(m_1)$ and $P(m_2)$ as:

$$P(m_1) = \frac{n_t}{n_t + n_i}, P(m_2) = \frac{n_i}{n_t + n_i} \tag{7.7}$$

where n_t is the number of images in the training set for which only text recognition successfully recognized the business. Similarly, n_i is the number of images for which only image matching worked successfully. $n_t + n_i$ is total number of images in the training set, i.e. 50.

The fusion process is unlikely to make a notable difference when both or none of the methods correctly recognize the business individually, regardless of the values of $p(b_i|m_1)$ and $p(b_i|m_2)$. However, when only one of the methods identifies the right business, proper values of $p(b_i|m_1)$ and $p(b_i|m_2)$ may results in successful overall recognition at the end. This is the reason our training set includes the query images for which only one of the methods worked. In other words, computing the values of $P(m_1)$ and $P(m_2)$ using the described method maximizes the chance of successful overall recognition for the cases where one of the methods fails.

If the word association method, explained in subsection 7.1.1, finds more than one business in the query, i.e. more than one cluster in mean-shift clustering, the text recognition and fusion process are repeated using the query words of each cluster in order to recognize multiple business. However, in case the best matching business in $p(b_i)$ has a low probability value, typically < 0.10, we disregard it as it most likely corresponds to a false positive from the text detection step.

7.2 Experimental Results

No dataset is currently available for evaluating the proposed framework as visual business recognition has not been studied to date. Therefore, we collected a data set of about 1042 GPS-tagged images comprised of 642 user uploaded photos from Panoramio, Flickr and Picasa and about 400 street view images for the cities of San Francisco, CA and Pittsburgh, PA. We manually filtered the images which do not show a business or have an excessively inaccurate GPS-tag. Each image may include up to four business. In case few businesses were retrieved by querying business directories for a particular query image, we added random businesses to make sure at least 20 businesses and 70 words existed in the lexicon to ensure each test is challenging enough.

Fig. 7.5 shows the results of the proposed business recognition framework for 4 query images which include one business (top rows) and multi businesses (bottom rows). Part (a) shows the query image and detected bounding boxes acquired from text detection. The recognized word for each bounding box is shown as well. (b) shows the PDFs obtained from text recognition, image matching and fusion, along with the best matching web images. Part (c) shows the recognized business in the image. (d) shows the word recognition results of Wang et al. [72] which is assisted by the business lexicon W as well.

The business recognition accuracy is defined as the number of correctly recognized businesses divided by the total number of businesses in the test set. We evaluated the proposed text processing and image matching approaches on the test set individually to examine their business

recognition performance in single modal fashion; this resulted in the accuracy of 69% and 41% for text recognition and image matching respectively. However, when the two methods were combined using the described fusion process, the accuracy increases to 75% which signifies the effectiveness of our multimodal approach.

Figure 7.5: Business recognition results. (a) shows the query image, detected text by text detection and recognized words. (b) the PDFs found by text recognition, image matching and fusion along with the best matching web images for each business. (c) the recognized businesses. (d) word recognition results of Wang et al.

No other framework for visual business recognition has been proposed which we can use as a baseline. However, it would be insightful to compare the performance of our text processing method, which is customized for business recognition, with the state of the art scene text recognition algorithms to see how well it recognizes business words.

Method	Acc.
GNU-Ocrad OCR [9]	4%
Chen et al. [3]	7%
ABBYY OCR [7]	34%
Wang et al. [6]	42%
Proposed Method	**54%**

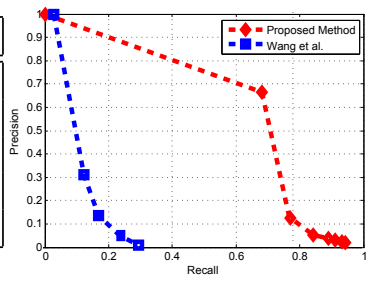

Figure 7.6: Left: Word recognition accuracy of the proposed method and the baselines. Right: Precision-recall curve of our method vs. Wang et al.'s.

The table in Fig. 7.6 compares the performance of our approach to four baselines. The performance measure is the number of correctly recognized words divided by the total number of words in our test images. Wang et al.'s [72] method which employs a pictorial structure for detection and recognition achieves the accuracy of 33% using the code of the authors with best tweaked parameters. Chen et al.'s [69] is another scene text recognition method for recognition of signs. Additionally, we compared our results with two OCR methods [103, 104] which have achieved notable success in document processing. In order to have a fair evaluation of text *recognition* in this experiment, we manually adjusted the text detection results for our methods and the baselines, in case a word is completely missed in detection. For the baseline methods which do not need a separate text detection step [72, 103], we limited their search space to the adjusted text detection results to decrease their false positives. Fig. 7.6 (b) shows the precision-recall curves of our method (red) vs. Wang et al.'s [72] (blue).

Unlike the majority of existing scene text recognition methods [69, 72] which employ a heavy training process, e.g. by using many different fonts and deformations, we used only one font and few deformations in our training. On the other hand, we leverage a more complex test step utilizing our multi-hypotheses character recognition approach. This is one of the reasons behind our superior performance in recognizing business words as they typically show a great deal of deformation and complexity in appearance which can not be effectively learnt in a training step. However, our multi hypotheses test step maximizes the use of business lexicon in order ro alleviate this issue.

Upon availability of an optimized parallel implementation of the framework, business recognition on a query can be done in no more than 3 seconds on average. We observed that the majority of the failure cases of our method are due extreme deformation of characters, lack of a relevant image on the web, and low quality of detected edges in the image.

7.3 Chapter Summary

In this chapter, we showed that using the geo-tag of images can play a major role in understanding the image content. In order to demonstrate this, we developed a multimodal approach for recognizing storefronts in images which centrally uses the image's geo-location. Our framework utilized textual information, web-images and business directories and fused the results of each modality a probabilistic late fusion process. We developed a multi hypotheses character recognition method which is specifically customized for business recognition for processing the text in images. The experiments showed the developed approach can effectively solve this practical challenging task by leveraging the geo-location of the image. In the coming chapter, we summarize the contributions discussed in this thesis and provide a discussion on the directions for the future work.

CHAPTER 8: CONCLUSION AND FUTURE WORK

Visual geo-localization, which is the problem of automatic identification of the location where an image or video was captured, has attracted much interest during the last few years. Since the conventional methods for visual geo-localization were primarily devised for airborne and satellite imagery, they fail to handle the ground-level data due to the dramatic dissimilarity of images and videos in these two viewpoints. In this dissertation, we addressed three questions central to geo-spatial analysis of ground-level imagery: **1) How to geo-localize images and videos captured at unknown locations? 2) How to refine the geo-location of already geo-tagged data? 3) How to utilize the extracted geo-tags?**

In the context of the first question, we argued that the availability of the massive ground-level data empowers image geo-localization to adopt an approach similar to *image matching*. We developed an image-matching based method for image geo-localization which uses Google Street View as the reference data and is capable of identifying the location with an accuracy comparable to hand-held GPS devices. We developed a novel feature correspondence pruning technique which incorporates the geo-spatial location of the Nearest Neighbors of local features in the pruning process to cope with the repetitive architectural features in man-made structures.

Next, we addressed the critical drawbacks of local features, originated from having a limited scope, by developing a *multi-NN* feature matching technique. We showed that the correspondences established based on merely local features often include a considerable number of mismatches, and the NN classifier, which is commonly employed for finding correspondences between local features, frequently fails at ranking the correct NNs as the 1^{st} one. As the remedy, we devised a novel formulation for feature matching which utilizes both local and global features simultaneously and incorporates multiple NN, as opposed to using only the 1^{st} one. In order to identify the correct NN out of the potential matches, we used *Generalized Minimum Clique Problem* (GMCP) which selects the correct NN for each query feature in a way that all of the NNs

matched to one image are similar in terms of their global features. We also showed that robustifying GMCP using Gaussian Radial Basis Function (G-RBF) kernels is crucial when the query image matches multiple reference images with dissimilar global features.

The majority of the existing methods for automatic geo-localization are targeted towards images and introduced a novel approach for geo-localization and geo-spatial trajectory extraction for videos. Our method was composed of three main steps: individual localization of video segments to extract the likelihood of different locations, employing Bayesian filtering to enforce the temporal consistency across different segments of the video, and applying a novel Minimum Spanning Tree (MST) based trajectory reconstruction method, which is free of any parametric-model, to cope with the stochastic motion of the camera and remove the remaining noise in the trajectory.

To address the second question, 'How to refine the geo-location of already geo-tagged data?', we empirically demonstrated that crowdsourced images suffer from the acute shortcoming of having inaccuracies in their geo-tags. We developed the first method for refinement of GPS-tags which automatically discovers the corrupted subset and refines the locations therein. We employed Random Walks to robustly discover the accurate subset from a large number of estimations which we generate for the location of an image. We showed that the Random Walk with the conventional constant damping factor is prone to contaminations in the input and devised an *adaptive damping factor* which conforms to the level of noise and consequently robustifies Random Walks significantly.

In the context of the third question, we argued that geo-tags have been used for rather low-key applications so far. We lifted the function of geo-tags to a higher level and utilized them for *understanding the image content*. In this context, we developed a *location-aware* approach for multimodal recognition of storefronts in images. Our method employs the image's geo-tag for extracting a set of priors composed of the businesses which may be visible in the image from business directories. In order to recognize the storefronts, the extracted priors were matched against the image content using a novel *multi hypothesis* text recognition technique. The multi hypothesis

approach was formulated as an optimization problem in which multi potential candidates for each query character were nominated and the correct one was selected by minimizing the edit distance between the permutations the candidates induced and a lexicon formed from the business priors. In addition, the image content was matched against a set of storefront images retrieved from the web using keywords formed based on the business lexicon. The results of these two modalities were then fused in a probabilistic formulation to identify the business(es) in the image.

8.1 Future Work

A number of potential directions for the future work in the area of geo-spatial analysis of images are provided below. In general, preforming the geo-localization in a cross-view manner and incorporation of semantics in the process of geo-spatial analysis are among the crucial tasks for the future. Developing more geo-localization techniques specifically devised for videos and more location-aware frameworks are of particular interest as well.

• **Cross-vew matching**: the main challenge of visual geo-localization is the availability of proper reference data. For instance, ground-level reference imagery with accurate metadata may not be available for an area of interest. This problem is exacerbated with the high cost of data collection, particularly for systematic efforts for acquiring ground-level imagery such as Street View. However, aerial imagery typically has a dense and broad coverage. Therefore, developing novel techniques that can perform the localization in a cross-view manner is of great importance in this area. The main difficulties of cross-view localization and matching appears to be discovering reliable point correspondences between the different views and modalities. Performing the geo-localization in a semantically meaningful fashion could be a potential remedy for this issue.

• **Incorporation of high-level information and semantics**: in general, semantics and high level information are often ignored by visual geo-localization methods as they mostly focus on establishing low-level correspondences (e.g. SIFT points, image patches, local features) between

the query and the reference data. Developing localization methods which are capable of effective utilization of higher level information and semantics is another important task for the future. Such information can potentially include text, facades, foliage type, building architecture, or the objects visible in the image.

• **Video geo-localization**: as discussed in Chapter 5, video geo-localization is an underdeveloped area of research, compared to images. Therefore, devising techniques which are capable of geo-locating a video, in particular for user-shared data, is among potential tasks for the future. Effective handling of undesired cinematographic effects in user-shared videos, such as abrupt changes or rather stochastic motion, are among the challenges which are yet to be resolved.

• **Development of more *location-aware* frameworks**: exploring new applications in the area of *location-aware* image analysis is another interesting direction for the future work. Such applications are becoming of particular importance as the amount of geo-tagged data on the web is increasing rapidly. Tackling new tasks, e.g. detecting anomalies in the geo-spatial context, or improving traditional Computer Vision problems, e.g. object detection or scene understanding, by leveraging the geo-spatial information are of particular interest in the future.

CHAPTER 9: APPENDIX

In this appendix, we provide a relaxed proof to formally show that employing the robust distance function developed in Sec. 4.1.2.2 can resolve the issue of GMCP with multiple groups of inlier NNs which are disjoint in the global feature space.

Consider the case shown in Fig. 9.1, which is generated using random synthetic data. Without the loss of generality, it represents the cases where the global features of matching reference images form two disjoint groups (e.g. Fig. 4.4 (b)). The larger and smaller groups are called Ω_1 and Ω_2 and include n_1 and n_2 nodes. We name the green (sample inlier) and orange (sample outlier) nodes $\underset{i}{v}$ and $\underset{o}{v}$ respectively.

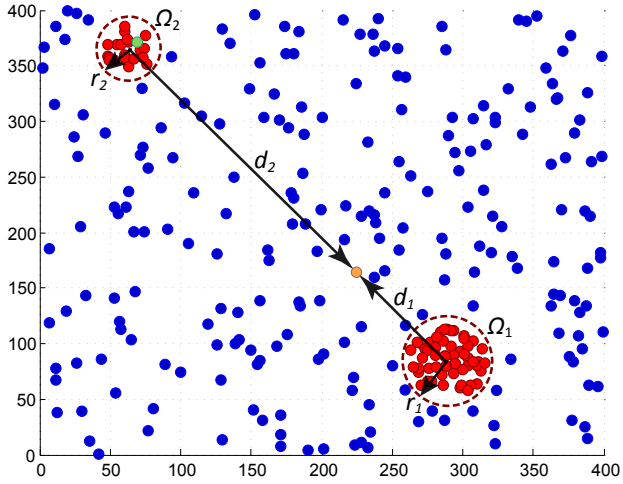

Figure 9.1: Two dimensional global feature space for a general case with two groups of matching reference images. The nodes represent the vertices in $\mathbf{V}$. Red and blue identify inlier and outliers respectively. Green and orange nodes mark one sample inlier and outlier respectively. r_1 and r_2 are the radii of the inlier groups. d_1 and d_2 are distances of the outlier to Ω_1 and Ω_2 respectively.

Assume the ideal case where an optimal GMCP solver is operational, and it is at the last

iteration where it has already included all the red nodes in its solution successfully. Also, suppose $\underset{o}{v}$ and $\underset{i}{v}$ are NNs of one query feature point, with identical local features and dissimilar global features. Hence, the GMCP solver should include one of them in the final solution at the last iteration. Therefore, we define two feasible solutions which are identical except in including $\underset{o}{v}$ or $\underset{i}{v}$:

$$\mathbf{V}_{s_i} = \mathbf{I} + \underset{i}{v}, \;\; \mathbf{V}_{s_o} = \mathbf{I} + \underset{o}{v}, \tag{9.1}$$

where $\mathbf{V}_{s_i}$ and $\mathbf{V}_{s_o}$ are the solutions which include $\underset{i}{v}$ or $\underset{o}{v}$ respectively, and $\mathbf{I}$ represents all the inlier nodes shown in red. From equation (4.4), the cost of $\mathbf{V}_{s_o}$ can be written as:

$$C(\mathbf{V}_{s_o}) = \frac{1}{2}\sum_{m=1}^{L}\sum_{\substack{n=1,\\ n\neq m}}^{L}\frac{1}{2}\Big(\varpi(\mathbf{V}_{s_o}(m)) + \varpi(\mathbf{V}_{s_o}(n))\Big) + \alpha.w(\mathbf{V}_{s_o}(m), \mathbf{V}_{s_o}(n)). \tag{9.2}$$

By substituting $\mathbf{V}_{s_o}$ from equation (9.1), equation (9.2) is expanded to:

$$\begin{aligned} C(\mathbf{V}_{s_o}) = \frac{1}{2} &\sum_{\substack{m=1,\\ \mathbf{V}_{s_o}(m)\neq \underset{o}{v}}}^{L} \sum_{\substack{n=1,\\ n\neq m,\\ \mathbf{V}_{s_o}(n)\neq \underset{o}{v}}}^{L} \Big[\frac{1}{2}\Big(\varpi(\mathbf{V}_{s_o}(m)) + \varpi(\mathbf{V}_{s_o}(n))\Big) + \alpha.w(\mathbf{V}_{s_o}(m), \mathbf{V}_{s_o}(n))\Big] \\ &+ \frac{1}{2}\Big[\sum_{\substack{n=1,\\ \mathbf{V}_{s_o}(n)\neq \underset{o}{v}}}^{L} \frac{1}{2}\Big(\varpi(\underset{o}{v}) + \varpi(\mathbf{V}_{s_o}(n))\Big) + \alpha.w(\underset{o}{v}, \mathbf{V}_{s_o}(n)) \\ &+ \sum_{\substack{m=1,\\ \mathbf{V}_{s_o}(m)\neq \underset{o}{v}}}^{L} \frac{1}{2}\Big(\varpi(\mathbf{V}_{s_o}(m)) + \varpi(\underset{o}{v})\Big) + \alpha.w(\mathbf{V}_{s_o}(m), \underset{o}{v})\Big], \end{aligned} \tag{9.3}$$

where the contributions of $\mathbf{I}$ and $\underset{o}{v}$ to the cost are separated. Edge weights are symmetrical by definition, so the arguments of the two single summations in equation (9.3) are identical thus

reducing to:

$$C(\mathbf{V}_{s_o}) = \frac{1}{2} \sum_{\substack{m=1,\\ \mathbf{v}_{s_o}(m)\neq \underset{o}{v}}}^{L} \sum_{\substack{n=1,\\ n\neq m,\\ \mathbf{v}_{s_o}(n)\neq \underset{o}{v}}}^{L} \Big[\frac{1}{2}\Big(\varpi(\mathbf{V}_{s_o}(m)) + \varpi(\mathbf{V}_{s_o}(n))\Big) + \alpha.w(\mathbf{V}_{s_o}(m), \mathbf{V}_{s_o}(n))\Big]$$

$$+ \sum_{\substack{n=1,\\ \mathbf{v}_{s_o}(n)\neq \underset{o}{v}}}^{L} \Big[\frac{1}{2}\Big(\varpi(\underset{o}{v}) + \varpi(\mathbf{V}_{s_o}(n))\Big) + \alpha.w(\underset{o}{v}, \mathbf{V}_{s_o}(n))\Big]. \tag{9.4}$$

Similar to $\mathbf{V}_{s_o}$, the cost of $\mathbf{V}_{s_i}$ can be derived as:

$$C(\mathbf{V}_{s_i}) = \frac{1}{2} \sum_{\substack{m=1,\\ \mathbf{v}_{s_i}(m)\neq \underset{i}{v}}}^{L} \sum_{\substack{n=1,\\ n\neq m,\\ \mathbf{v}_{s_i}(n)\neq \underset{i}{v}}}^{L} \Big[\frac{1}{2}\Big(\varpi(\mathbf{V}_{s_i}(m)) + \varpi(\mathbf{V}_{s_i}(n))\Big) + \alpha.w(\mathbf{V}_{s_i}(m), \mathbf{V}_{s_i}(n))\Big]$$

$$+ \sum_{\substack{n=1,\\ \mathbf{v}_{s_i}(n)\neq \underset{i}{v}}}^{L} \Big[\frac{1}{2}\Big(\varpi(\underset{i}{v}) + \varpi(\mathbf{V}_{s_i}(n))\Big) + \alpha.w(\underset{i}{v}, \mathbf{V}_{s_i}(n))\Big]. \tag{9.5}$$

The argument of the single summations in equations (9.4) and (9.5) corresponds to the portion of the cost which the nodes $\underset{o}{v}$ or $\underset{i}{v}$ contribute respectively. The argument of the double summations is the portion which all other nodes, i.e. **I**, induce, and therefore are identical in equations (9.4) and (9.5).

GMCP without Robustification: We show that even under the presumed ideal conditions, $\underset{o}{v}$ will still be incorrectly preferred over $\underset{i}{v}$ by GMCP, i.e.:

Proposition 1 *The feasible solution $\boldsymbol{V}_{s_o}$ induces a lower cost compared to $\boldsymbol{V}_{s_i}$ when the robustifi-*

cation is not employed (using ℓ_2 distance):

$$C(\boldsymbol{V}_{s_o}) < C(\boldsymbol{V}_{s_i}). \tag{9.6}$$

By substituting equations (9.4) and (9.5) into inequality (9.6) and canceling equal terms, we obtain:

$$\begin{aligned}
&\sum_{\substack{n=1,\\ \mathbf{v}_{s_o}(n)\neq \underset{o}{v}}}^{L} \Big[\frac{1}{2}\Big(\varpi(\underset{o}{v}) + \varpi(\mathbf{V}_{s_o}(n))\Big) + \alpha.w(\underset{o}{v}, \mathbf{V}_{s_o}(n))\Big] < \\
&\sum_{\substack{n=1,\\ \mathbf{v}_{s_i}(n)\neq \underset{i}{v}}}^{L} \Big[\frac{1}{2}\Big(\varpi(\underset{i}{v}) + \varpi(\mathbf{V}_{s_i}(n))\Big) + \alpha.w(\underset{i}{v}, \mathbf{V}_{s_i}(n))\Big].
\end{aligned} \tag{9.7}$$

By substituting ϖ and w from equations (4.2) and (4.3), inequality (9.7) is expanded to:

$$\begin{aligned}
&\sum_{\substack{n=1,\\ \mathbf{v}_{s_o}(n)\neq \underset{o}{v}}}^{L} \frac{1}{2}(\overbrace{\|Q - \zeta(\underset{o}{v})\| + \ |q^n - \zeta(\mathbf{V}_{s_o}(n))\|}^{①}) + \alpha.\|\rho(\underset{o}{v}) - \rho(\mathbf{V}_{s_o}(n))\| < \\
&\sum_{\substack{n=1,\\ \mathbf{v}_{s_i}(n)\neq \underset{i}{v}}}^{L} \frac{1}{2}(\overbrace{\|Q - \zeta(\underset{i}{v})\| + \|q^n - \zeta(\mathbf{V}_{s_i}(n))\|}^{①}) + \alpha.\|\rho(\underset{i}{v}) - \rho(\mathbf{V}_{s_i}(n))\|.
\end{aligned} \tag{9.8}$$

$\underset{o}{v}$ and $\underset{i}{v}$ are NNs of one query feature point; we call the local descriptor of that particular query feature Q. The terms ① can be canceled since they correspond to the contribution of local features

into the cost while we assumed $\zeta(\underset{o}{v}) = \zeta(\underset{i}{v})$:

$$\sum_{\substack{n=1,\\ \mathbf{v}_{s_o}(n)\neq \underset{o}{v}}}^{L} \|\rho(\underset{o}{v}) - \rho(\mathbf{V}_{s_o}(n))\| < \sum_{\substack{n=1,\\ \mathbf{v}_{s_i}(n)\neq \underset{i}{v}}}^{L} \|\rho(\underset{i}{v}) - \rho(\mathbf{V}_{s_i}(n))\|. \tag{9.9}$$

In order to estimate the value of each side of the inequality (9.9), we make two relaxing assumptions:

Assumption (I): The outlier node, $\underset{o}{v}$, is not located close to the inlier regions. In other words, $d_{1,2} \gg r_{1,2}$ which signifies the distance of outlier node to the inlier regions is significantly larger than their size.

Assumption (II): The outlier node, $\underset{o}{v}$, roughly falls in the space between Ω_1 and Ω_2 in the global feature space. Thus, the distance between the two inlier groups can be approximated as $\approx (d_1 + d_2)$.

Considering the plot of Fig. 9.1 and employing the relaxing assumption *(I)*, the value of $\|\rho(\underset{o}{v}) - \rho(\mathbf{V}_{s_o}(n))\|$ is either $\approx d_1$ or $\approx d_2$ for a n which corresponds to a node in Ω_1 or Ω_2 respectively. Similarly, utilizing the relaxing assumptions *(I,II)*, the value of $\|\rho(\underset{i}{v}) - \rho(\mathbf{V}_{s_i}(n))\|$ is either $\approx (d_1 + d_2)$ or ≈ 0 for a n which corresponds to a node in Ω_1 or Ω_2. Therefore, the inequality (9.9) can be approximated to:

$$n_1 \times d_1 + n_2 \times d_2 < n_1 \times (d_1 + d_2), \tag{9.10}$$

$$n_2 < n_1. \tag{9.11}$$

Inequality (9.11) is true since Ω_1 is larger than Ω_2. Therefore, we proved the proposition inequality (9.6), meaning that the basic formulation of the GMCP-based method without-robustification fails to include all the inliers in $\hat{\mathbf{V}}_s$ even upon optimal solving of GMCP and the presumed ideal conditions. ■

9.1 GMCP with Robustification

We show that $\mathbf{V}_{s_i}$ will be preferred over $\mathbf{V}_{s_o}$ when robustification is utilized, i.e.:

Proposition 2 *Employing the robust distance function D, the feasible solution $\boldsymbol{V}_{s_i}$ induces a lower cost compared to $\boldsymbol{V}_{s_o}$:*

$$C(\boldsymbol{V}_{s_o}) > C(\boldsymbol{V}_{s_i}), \tag{9.12}$$

A similar derivation which resulted in inequality (9.9) from (9.6) can be performed on inequality (9.12) which leads to:

$$\sum_{\substack{n=1,\\ \mathbf{v}_{s_o}(n)\neq \underset{o}{v}}}^{L} \|\rho(\underset{o}{v}) - \rho(\mathbf{V}_{s_o}(n))\| > \sum_{\substack{n=1,\\ \mathbf{v}_{s_i}(n)\neq \underset{i}{v}}}^{L} \|\rho(\underset{i}{v}) - \rho(\mathbf{V}_{s_i}(n))\|. \tag{9.13}$$

Employing approximations *(I,II)*, inequality (9.13) is estimated to:

$$n_1 \times d_1 + n_2 \times d_2 > n_1 \times (d_1 + d_2). \tag{9.14}$$

However, by employing the function D, all distances are mapped according to the function shown in Fig. 4.5. Therefore, if σ is set to the approximate value of the radius of inlier regions, $r_{1,2}$, the distances d_1 and d_2 will be both mapped to $\approx \tau$.[1] This is because $d_{1,2} \gg r_{1,2}$ utilizing the assumption *(I)*. Similarly $(d_1 + d_2)$ will be mapped to $\approx \tau$ reducing the inequality (9.15) to:

$$\begin{aligned} n_1 \times \tau + n_2 \times \tau &> n_1 \times \tau, \\ n_2 &> 0, \end{aligned} \tag{9.15}$$

[1] In practice, the size of the inlier regions are comparable which implies it does not matter if σ is set to r_1 or r_2. However, in case the sizes are not similar, σ should be set to the largest one to make sure the distances between inlier nodes remain undamped while the ones which include an outlier are diminished.

which is true since Ω_2 is a nonempty group, and the proposition is proved.■

The proofs provided in the appendix were based upon the assumptions *(I,II)*. Formal proof for a general case for more than two groups without the relaxing assumptions is cumbersome. However, intuitively, it can be verified that the assumptions do not stop the proof from being generalized: If $\underset{o}{v}$ is not exactly on the line connecting Ω_1 and Ω_2, the inequality (9.12) will still be satisfied as long as $\underset{o}{v}$ does not significantly deviate from the space between Ω_1 and Ω_2 (assumption *(II)*). Additionally, the assumption *(I)* is typically correct as $\underset{o}{v}$ is an outlier, which inherently means it should not be too close to the inlier regions. Hence, in practice the above proof can be extended to more general cases by intuition. Fig. 4.6-left shows employing the robustification yields satisfactory results for a case with three disjoint clusters generated from the real data where the relaxing assumptions are not held.

LIST OF REFERENCES

[1] A. R. Zamir and M. Shah, "Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2014.

[2] A. R. Zamir and M. Shah, "Accurate image localization based on google maps street view," in *European Conference on Computer Vision (ECCV)*, pp. 255–268, Springer, 2010.

[3] G. Vaca, A. R. Zamir, and M. Shah, "City scale geo-spatial trajectory estimation of a moving camera," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[4] A. R. Zamir, S. Ardeshir, and M. Shah, "GPS-Tag refinement using random walks with an adaptive damping factor," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[5] A. R. Zamir, A. Dehghan, and M. Shah, "Visual business recognition - a multimodal approach," in *Proceeding of ACM International Conference on Multimedia (ACM MM)*, 2013.

[6] A. R. Zamir, A. Dehghan, and M. Shah, "GMCP-Tracker: Global multi-object tracking using generalized minimum clique graphs," in *European Conference on Computer Vision (ECCV)*, 2012.

[7] S. Modiri, A. R. Zamir, and M. Shah, "Video classification using semantic concept co-occurrences," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[8] A. R. Zamir, A. Darino, and M. Shah, "Street view challenge: Identification of commercial entities in street view imagery," in *2011 10th International Conference on Machine Learning and Applications (ICMLA)*, vol. 2, pp. 380–383, IEEE, 2011.

[9] "U.s. geological survey, *http://www.usgs.gov/*,"

[10] "Panoramio - photos of the world: http://www.panoramio.com,"

[11] Y. Sheikh, S. Khan, and M. Shah, "Feature-based georegistration of aerial images," *GeoSensor Networks*, vol. 4, 2004.

[12] B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and vision computing*, vol. 21, no. 11, pp. 977–1000, 2003.

[13] R. Kumar, H. Sawhney, J. Asmuth, A. Pope, and S. Hsu, "Registration of video to georeferenced imagery," in *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, vol. 2, pp. 1393–1400 vol.2, Aug 1998.

[14] G. Schindler, M. Brown, , and R. Szeliski, "City-scale location recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[15] J. Hays and A. Efros, "im2gps: estimating geographic information from a single image," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," in *International Journal of Computer Vision*, 2004.

[17] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[18] J. Philbin, O. Chum, , M. Isard, J. Sivic, , and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[19] C. Feremans, M. Labbe, and G. Laporte, "Generalized network design problems," in *European Journal of Operational Research Volume 148, Issue 1*, 2003.

[20] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *International Journal of Robotics Research*, June 2008.

[21] M. Cummins and P. Newman, "Appearance-only slam at large scale with fab-map 2.0," *International Journal of Robotics Research*, November 2010.

[22] A. Hakeem, R. Vezzani, M. Shah, and R. Cucchaira, "Estimating geospatial trajectory of a moving camera," in *International Conference on Pattern Recognition*, 2006.

[23] Y. Jing and S. Baluja, "Visualrank: Applying pagerank to large-scale image search," 2008.

[24] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang, "Tag ranking," in *International Conference on Multimedia*, 2009.

[25] F. Spitzer, *The Theory of Stochastic Processes*. Springer, 2001.

[26] P. J. Huber, *Robust statistics*. Springer, 2011.

[27] "Wi-fi positioning system, *http://www.skyhookwireless.com/howitworks/*,"

[28] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," *ACM Trans. Graph.*, vol. 25, pp. 835–846, July 2006.

[29] J. Knopp, J. Sivic, and T.Pajdla, "Avoding confusing features in place recognition," in *European Conference on Computer Vision*, 2010.

[30] J. Sivic and A. Zisserman, "Video google: a text retrieval approach to object matching in videos," in *International Conference on Computer Vision*, 2003.

[31] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros, "What makes paris look like paris?," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 101, 2012.

[32] T. Sattler, B. Leibe, , and L. Kobbelt, "Fast image-based localization using direct 2d-to-3d matching," in *International Conference on Computer Vision*, 2010.

[33] Y. Li, N. Snavely, and D. Huttenlocher, "Location recognition using prioritized feature matching," in *European Conference on Computer Vision*, 2010.

[34] T. Sattler, B. Leibe, , and L. Kobbelt, "Improving image-based localization by active correspondence search," in *European Conference on Computer Vision*, 2012.

[35] T.-Y. Lin, S. Belongie, and J. Hays, "Cross-view image geolocalization," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2013.

[36] R. Mohedano, A. Cavallaro, and N. Garcia, "Camera localization usingtrajectories and maps," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, pp. 684–697, April 2014.

[37] E. Kalogerakis, O. Vesselova, J. Hays, A. Efros, and A. Hertzmann, "Image sequence geolocation with human travel priors," 2009.

[38] D. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg, "Mapping the world's photos," in *International World Wide Web Conference*, 2009.

[39] W. Zhang and J. Kosecka, "Image based localization in urban environments," in *3DPVT '06: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, pp. 33–40, 2006.

[40] N. Jacobs, S. Satkin, N. Roman, R. Speyer, and R. Pless, "Geolocating static cameras," 2007.

[41] E. Mortensen, H. Deng, and L. Shapiro, "A sift descriptor with global context," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[42] K. Mikolajczyk, A. Zisserman, and C. Schmid, "Shape recognition with edge-based features," in *British Machine Vision Conference*, 2003.

[43] B. Cao, C. Ma, and Z. Liu, "Affine-invariant sift descriptor with global context," in *International Congress on Image and Signal Processing*, 2010.

[44] Q. Hao, R. Cai, Z. Li, L. Zhang, Y. Pang, , and F. Wu, "3d visual phrases for landmark recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[45] Y. Zhang, Z. Jia, and T. Chen, "Image retrieval with geometry-preserving visual phrases," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[46] Y. Avrithis, G. Tolias, and Y. Kalantidis, "Feature map hashing: sub-linear indexing of appearance and global geometry," in *International Conference on Multimedia*, 2010.

[47] T. Sattler, B. Leibe, and L. Kobbelt, "Scramsac: Improving ransac's efficiency with a spatial consistency filter," in *International Conference on Computer Vision*, 2009.

[48] P. Torr and A. Zisserman, "A new robust estimator with application to estimating image geometry," in *IEEE Computer Vision and Image Understanding*, 2000.

[49] H. Wang, D. Mirota, and G. Hager, "A generalized kernel consensus-based robust estimator," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[50] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *CVPR*, 2004.

[51] D. Scaramuzza, "1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints," *IJCV*, vol. 95, no. 1, 2010.

[52] J. Tardif, Y. Pavlidis, and K. Daniilidis, "Monocular visual odometry in urban environments using an omnidirectional camera," in *International Conference on Intelligent Robots and Systems*, 2008.

[53] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *IROS*, 2008.

[54] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real time localization and 3d reconstruction," in *Computer Vision and Pattern Recognition (CVPR)*, 2006.

[55] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (slam): Part i the essential algorithms," *Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[56] J. Civera, O. Grasa, A. Davison, and J. Montiel, "1-point ransac for ekf filtering: Application to real-time structure from motion and visual odometry," *Journal of Field Robotics*, vol. 27, pp. 609–631, 2010.

[57] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *ICCV*, 2003.

[58] G. Klein and D. Murray, "Parallel tracking and mapping for smaller workspaces," in *International Symposium on Mixed and Augmented Reality*, 2007.

[59] A. Hakeem, R. Vezzani, M. Shah, and R. Cucchiara, "Estimating geospatial trajectory of a moving camera," in *ICPR*, 2006.

[60] X. Li, G. M. Snoekand, and M. Worring, "Learning social tag relevance by neighbor voting," in *IEEE Transactions on Multimedia*, 2009.

[61] G. Zhu, S. Yan, and Y. Ma, "Image tag refinement towards low-rank, content-tag prior and error sparsity," in *International Conference on Multimedia*, 2010.

[62] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski, "Building rome in a day," in *International Conference on Computer Vision*, 2009.

[63] Y.-T. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, and H. Neven, "Tour the world: building a web-scale landmark recognition engine," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1085–1092, IEEE, 2009.

[64] K. Yanai, H. Kawakubo, and B. Qiu, "A visual analysis of the relationship between word concepts and geographical locations," in *Proceedings of the ACM International Conference on Image and Video Retrieval*, CIVR '09, (New York, NY, USA), pp. 13:1–13:8, ACM, 2009.

[65] K. Yaegashi and K. Yanai, "Geotagged photo recognition using corresponding aerial photos with multiple kernel learning," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, pp. 3272–3275, IEEE, 2010.

[66] "Google goggles, *www.google.com/mobile/goggles/*,"

[67] "Nokia city lens, *http://betalabs.nokia.com/trials/nokia-city-lens-for-windows-phone*,"

[68] "Yelp, *http://www.yelp.com/*,"

[69] X. Chen, J. Yang, J. Zhang, and A. Waibel, "Automatic detection and recognition of signs from natural scenes," in *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 2004.

[70] X. C. J. Y. J. Z. A. Waibel, "Automatic detection and recognition of signs from natural scenes," in *IEEE Transactions on Image Processing*, 2004.

[71] X. Chen and A. Yuille, "Detecting and reading text in natural scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[72] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1457–1464, 2011.

[73] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm," in *International Conference on Computer Vision Theory and Applications*, 2009.

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

[75] K. P. Balanda and H. L. MacGillivray, "Kurtosis: A critical review," *The American Statistician*, vol. 42, no. 2, pp. 111–119, 1988.

[76] A. Koster, S. V. Hoesel, and A. Kolen, "The partial constraint satisfaction problem: Facets and lifting theorems," in *Operations Research Letters 23*, 1998.

[77] J. Pearl, "Probabilistic reasoning in intelligent systems: Networks of plausble inference," in *Intelligent Systems: Networks of Plausble Inference*, 1988.

[78] Y. Weiss, "Correctness of local probability propagation in graphical models with loop," in *Neural Computation*, 2000.

[79] G. Wu, E. Y. Chang, and N. Panda, "Formulating distance functions via the kernel trick," in *In Conf. on Knowledge Discovery and Data Mining*, 2005.

[80] J. C. Gower and G. J. S. Ross, "Minimum spanning trees and single linkage cluster analysis," in *Journal of the Royal Statistical Society*, 1969.

[81] Y. S. Myung, C. H. Lee, and D. W. Tcha, "On the generalized minimum spanning tree problem," in *Networks*, 1995.

[82] B. S. Everitt, S. Landau, and M. Leese, *Cluster Analysis*. Wiley Publishing, 4th ed., 2009.

[83] E. Althaus, O. Kohlbacher, H. Lenhof, and P. Mller., "A combinatorial approach to protein docking with flexible side chains," in *Journal of Computational Biology*, 2002.

[84] D. Ghosh, "Solving medium to large sized euclidean generalized minimum spanning tree problems," in *WP. No. 2003-08-02, Indian Institute of Management*, 2003.

[85] Z. Wang, C.H, and C. A. Lim, "Tabu search for generalized minimum spanning tree problem," in *Pacific Rim International Conference on Artificial Intelligence*, 2006.

[86] D. Chen, G. Baatz, K. Koeser, S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk, "City-scale landmark identification on mobile devices," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2011.

[87] A. Torralba, "Contextual priming for object detection," in *International Journal of Computer Vision, Vol. 53(2), 169-191*, 2003.

[88] J. Bentley, "Multidimensional binary search trees in database applications," in *IEEE Transactions on Software Engineering*, 1979.

[89] P. Huber, "Robust estimation of a location parameter," in *The Annals of Mathematical Statistics*, 1964.

[90] "Flickr - photo sharing: http://www.flickr.com,"

[91] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, no. 2, 2004.

[92] M. Muja and D. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration.," in *VISAPP'09*.

[93] T. Sattler, B. Leibe, and L. Kobbelt, "Fast image-based localization using direct 2d-to-3d matching," in *ICCV'11*.

[94] R. Perlman, "An algorithm for distributed computation of a spanningtree in an extended lan," in *SIGCOMM*, 1985.

[95] B. Ma, A. Hero, J. Gorman, and O. Michel, "Image registration with minimum spanning tree algorithm," in *International Conference on Image Processing*, 2000.

[96] I.-K. Lee, "Curve reconstruction from unorganized points," *Computer Aided Geometric Design*, vol. 17, pp. 161–177, 2000.

[97] L. H. de Figueiredo and J. de Miranda Gomes, "Computational morphology of curves," *The Visual Computer*, vol. 11, no. 2, pp. 105–112, 1994.

[98] Google, "http://www.youtube.com.".

[99] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Multicore bundle adjustment," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2011.

[100] C. Wu, "Visualsfm: A visual structure from motion system," *http://ccwu.me/vsfm/*, 2011.

[101] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher, "Discrete-continuous optimization for large-scale structure from motion," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 3001–3008, 2011.

[102] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2963–2970, 2010.

[103] "http://www.abbyy.com/,"

[104] "Gnu-orcad ocr, *http://www.gnu.org/s/ocrad/*,"