Air Force Institute of Technology
AFIT Scholar

Theses and Dissertations

9-14-2017

A Location-Aware Middleware Framework for Collaborative Visual Information Discovery and Retrieval

Andrew J.M. Compton

Follow this and additional works at: https://scholar.afit.edu/etd

 Part of the Digital Communications and Networking Commons

Recommended Citation

Compton, Andrew J.M., "A Location-Aware Middleware Framework for Collaborative Visual Information Discovery and Retrieval" (2017). *Theses and Dissertations*. 767.
https://scholar.afit.edu/etd/767

This Dissertation is brought to you for free and open access by AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact richard.mansfield@afit.edu.

A LOCATION-AWARE MIDDLEWARE FRAMEWORK FOR COLLABORATIVE VISUAL INFORMATION DISCOVERY AND RETRIEVAL

DISSERTATION

Andrew J.M. Compton, Major, USAF

AFIT-ENG-DS-17-S-010

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENG-DS-17-S-010

A LOCATION-AWARE MIDDLEWARE FRAMEWORK FOR COLLABORATIVE VISUAL INFORMATION DISCOVERY AND RETRIEVAL

DISSERTATION

Presented to the Faculty

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Andrew J.M. Compton, B.S.C.E., M.S.C.E.

Major, USAF

September 2017

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENG-DS-17-S-010

A LOCATION-AWARE MIDDLEWARE FRAMEWORK FOR COLLABORATIVE VISUAL INFORMATION DISCOVERY AND RETRIEVAL

DISSERTATION

Andrew J.M. Compton, B.S.C.E., M.S.C.E.
Major, USAF

Committee Membership:

LtCol John M. Pecarina, PhD
Chairman

Dr. John F. Raquet
Member

Dr. David R. Jacques
Member

Dr. Adedeji B. Badiru
Dean

AFIT-ENG-DS-17-S-010

Abstract

This work addresses the problem of scalable location-aware distributed indexing to enable the leveraging of collaborative effort for the construction and maintenance of world-scale visual maps and models. These maps and models support numerous activities including navigation, visual localization, persistent surveillance, and hazard or disaster detection. Due to the physical scale of the world, the problem of constructing and maintaining relevant world-scale visual models generally requires significant effort to be spent on visual mapping. The dynamic nature of physical environments also causes visual records to lose relevance, requiring that mapping be re-accomplished periodically in order to account for both natural and man-made environmental changes. While physical scale and environmental dynamism pose the most obvious challenges to solutions for this problem, the underlying computational and data storage infrastructures required are non-trivial. The infrastructure must be capable of providing for the storage, indexing, and retrieval of published images as well as the expiry of images which are not relevant or representative of the physical environment.

A solution to the problem of constructing and maintaining surface level visual world models is presented in the form of a peer-to-peer middleware framework named PeerAppear. The system architecture for PeerAppear is developed including descriptive use-cases, indexing methods, geospatial addressing, and a protocol specification. Computational methods are presented which support the framework by providing for efficient visual search, cell-based geospatial representation for position and viewshed, and a search-based scoring of visual landmarks based on area of visibility. Finally, an evaluation of PeerAppear's indexing performance, scalability, and visual search

is presented using large-scale simulations and small-scale real-world datasets. The results of the evaluation demonstrate that the system provides a highly effective and scalable means for enabling collaborative visual information discovery and retrieval.

Acknowledgements

I would like to thank my research advisor, Lt Col John Pecarina, for his mentorship, guidance and support throughout this research effort. His persistent optimism kept me motivated and his insistence on segmenting the research into publishable units kept the work on track and ensured that the research progressed to completion. I would also like to thank Dr. John Raquet for providing the catalyst and motivational problem upon which this research was based as well as for his mentorship and support while serving as my pro-tem advisor and later as a committee member.

Andrew J.M. Compton

Table of Contents

Page

Abstract iv

List of Figures x

List of Tables xv

I. Introduction 1

II. A Distributed Approach for Collaborative Image-Based World Modeling 7

2.1 Self-Updating World Models 7
2.2 The Need for Collaborative Image-Based World Model Systems 10
2.3 Requirements for Collaborative Image-Based World Model Systems 14
2.4 Research Activities 16
Research Thrust 1: Collaborative Middleware Infrastructure 17
Research Thrust 2: Computational Support for Practical Visual Applications 18
Research Thrust 3: Integration of Middleware for Practical Application-Based Evaluation 19

III. Core Infrastructure Supporting Collaborative World Model Construction and Maintenance 21

3.1 Related Works 25
Geospatial Addressing 25
Architectures for Decentralized Indexing 29
3.2 PeerAppear Middleware Overview 36
Publish and Search Functionality 37
PeerAppear Role-Based Process Interaction 38
System Requirements and Assumptions 41
Middleware Overview 42
3.3 Peer-to-Peer Distributed Geospatial Indexing 45
S2 Geospatial Indexing 45
Networking in an S2-Integrated Kademlia Variant 47
Publish and Search Protocol 51
3.4 Middleware Evaluation 53
PeerAppear Simulation Engine (PASE) 53
Addressing Performance 54

Page

Network Performance 61
3.5 Conclusions 68

IV. Computational Support for Practical Visual Applications Using Collaborative World Models 71

4.1 Related Works 71
4.2 Visual Localization Activities 78
Roles 78
Procedures 79
4.3 An Architecture for Collaborative Image Discovery 80
Extensible Decentralized Geospatial Image Repositories 82
Two-Stage Visual Search with *tf-idf* Weighting 89
Indexing Algorithms and Support 96
4.4 Geospatial Filtering for Position Representation and Search 104
Position Representation 104
Viewshed Representation 107
Path Viewshed Representation 113
S2Trie for Geospatial Filter Representation 114

V. Integration of Middleware Infrastructure and Computational Support for Collaborative Visual Localization 121

5.1 Related Works 121
5.2 Localization using the PeerAppear Visual Search Architecture 126
Theoretical Search Complexity in PeerAppear 126
Evaluation Testbed and Methodology 127
Evaluation Dataset 129
Results 131
5.3 Collaborative Visual Search Using PeerAppear's Distributed Index 135
Integration with the PeerAppear Simulation Engine 137
Evaluation Dataset 138
Results 141

VI. Conclusion 146

6.1 Summary and Conclusions 146
6.2 Potential Applications 149
6.3 Future Work 150
6.4 Thesis Summary 151

Page

Appendix A. Publication and Presentation Summary 153

1.1 CTS 2016 153
1.2 GNSS 2016 154
1.3 CogSIMA 2017 155
1.4 JNC 2017 156
1.5 FGCS Journal 2018 158
1.6 ION Journal 2018 159

Bibliography 160

List of Figures

Figure		Page
1	A self-updating world model concept diagram for navigation.[88]	9
2	A spatial density plot showing the density of Flickr images located within 5km of the Washington Monument.	12
3	The PeerAppear middleware framework enables global availability of indexed data through a globally interconnected network topology and publish-search architecture.	24
4	The quadrilateralized spherical cube projection.[117]	27
5	The Hierarchical Triangular Mesh global decomposition using spherical triangular segmentation.[104]	28
6	Populating the index: The process of summarizing local data into a set of cell addresses to be published to the DGT. Image capture locations (left) covered as cells (middle) to reduce the number of published addresses (right).	37
7	Search: The process of performing a DGT lookup for keys matching a specified search filter. The potential search location (left) is covered with cells (middle) used to narrow potential repositories for search (right).	38
8	A mapping of the globe using Google's S2 Spherical Geometry library with level 3 cells. Center cube face (Africa) shows sequential addressing using Hilbert mapping.	46
9	A depiction of hierarchical S2 cells and corresponding cell prefixes.	46
10	The tree structure of fully populated peer lists of length k. Nodes with the furthest XOR distance are on the left and closest XOR distance are on the right.	49
11	The global mapping of individual peer list coverage areas, each containing up to k peers, shaded by color for a randomly generated node (red dot).	50

Figure Page

12 Representation of great-circle distance in the PeerAppear address space and routing table. 56

13 Spatial locality analysis for peer lists and RPC execution. 58

14 Non-uniform distribution of key storage across address space caused by non-uniform geospatial distribution of nodes. 60

15 Non-uniform distribution of key storage across address space caused by cell level prefix size ($C_L = \{1, 2, 3, 4, 5\}$). 62

16 Effect of numbers of peers on recall and complexity for various alpha values. 65

17 Effect of numbers of peers on recall and complexity for various k values. 66

18 Evaluation of search recall and complexity at different search cell levels for various network sizes. 67

19 Evaluation of search recall and complexity at different search cell levels for various k values. 69

20 Matched visual features and associated perspective transform between images of the Statue of Liberty. 75

21 A convolutional neural network architecture for optical character recognition[114]. 77

22 A collaborative loop closure scenario for monocular visual odometry. 83

23 The local and networked components of clients participating in the PeerAppear network. 83

24 A mapping of the globe using Google's S2 Spherical Geometry library with level 6 cells. 85

25 The structure of layered image data within the PeerAppear database. 88

26 The framework loader XML description of the wifi_tag_svc plugin. 88

27 An example PeerAppear query and associated results. 89

Figure Page

28 Results of image similarity search, top left is query image. 93

29 A pair of visually matched images with RANSAC-validated feature correspondence. 95

30 A histogram depicting the vector representation of the words describing the features of a single image. 97

31 The positions of geo-located images containing a confirmed corresponding element (Washington Monument) for localization. 99

32 The correlation ellipse representing the area from which the iconic structural or geographical element (Washington Monument) can be seen. 100

33 A flowchart depicting the AVSC worker process used to select and score images. 103

34 A position estimate filter comprised of 20 S2 cells. 105

35 A position estimate filter comprised of 4 S2 cells. 106

36 The result of pruning max 25-count search filters using various minimum coverage pruning constraint values. 108

37 A gray scale terrain elevation map of Moraine Airpark in Moraine Ohio. 111

38 A viewshed created by GRASS GIS for a point selected near the west end of the runway at Moraine Airpark. 111

39 The viewshed from Figure 38 represented using a 2D point cluster. 112

40 A viewshed-based search filter at Moraine Airpark created by covering the viewshed point cluster shown in Figure 39 with level 16 S2 cells. 112

41 A path viewshed filter and density plot for a roadway route between Beavercreek Ohio and Wright-Patterson Air Force Base. 114

42 An example *S2Trie* instance containing level 3, 4, and 5 cells on the North American (*100*) cube face. 116

Figure Page

43 The viewshed cover shown in Figure 40 with all possible cell mergings completed. 118

44 The cell counts of the filter from 43 after density-based pruning for a range of minimum density values. 118

45 Search filters for Moraine Airpark which are pruned using the *S2Trie*'s *prune_by_density* method. 119

46 A location inference made by Google's PlaNet network to geolocate for a given image[116]. 125

47 Raspberry Pi-based hardware used to collect image set for experimental evaluation. 129

48 Test hardware mounted to vehicle windshield for collecting evaluation dataset as dashboard camera. 130

49 A map of image collection tracks colored by track. 131

50 Test track and path of associated location inferences for large loop. 133

51 Absolute error for position estimates generated during large loop test execution. 134

52 Test track and path of associated location inferences for small loop. 134

53 Absolute error for position estimates generated during small loop test execution. 135

54 Experiment interface depicting search track including query and result images along with associated weighted BoVW image representations. 136

55 Results of image similarity search, top left is query image. 137

56 A map showing the tracks traversed by each simulated client while capturing images and data used to populate the client's local repository. 140

Figure Page

57 A single frame from the evaluation showing the location of the query image (magenta circle), search filter (blue cells), query image and result image with matched features, and query and result BoVW vectors. 142

58 Recall of individual search queries for the evaluation track and mean recall for the top 1,5,10,20 search results. 144

List of Tables

Table		Page
1	Framework elements used to meet system requirements.	43
2	Remote Procedure Call (RPC) functions used to manage nodes and data within PeerAppear.	52
3	PASE network parameters for locality analysis.	59
4	Environmental and system parameters used to execute experiments within PASE.	63
5	The search space cell counts, coverage and spill factor for radius-based search filters created using specified maximum count cell covers.	107
6	The cell counts, coverage and spill factors for search filters created using 95%, 85% and 75% minimum coverage pruning constraints and specified maximum count cell covers.	109
7	Mean image search precision for complete dataset, small loop track, and large loop track.	131
8	Mean image search recall for complete dataset, small loop track, and large loop track.	132
9	Fulfillment of framework requirements.	148

A LOCATION-AWARE MIDDLEWARE FRAMEWORK FOR
COLLABORATIVE VISUAL INFORMATION DISCOVERY AND RETRIEVAL

I. Introduction

In recent decades our portable electronic devices have gained significant awareness of their environments through the inclusion of an array of new sensors. The prevalence of these sensors has been spurred by miniaturization and massive declines in cost, with many chips now including a miniature array of various sensors in a single inexpensive package. Most modern smart phones include multiple imaging sensors, an accelerometer, gyroscope, magnetometer, barometer, light sensor, microphones, GPS receiver, and various RF transceivers. This leads to unprecedented awareness of the local environment and the sensor's position in it, which can be exploited for visual [123], magnetic [39], and WiFi [101] mapping applications. With significant adoption of these devices occurring within the United States [1] and world-wide [84], the aggregate reach of these sensors and associated mapping capabilities is rapidly approaching global scale, thereby offering the potential for systems supporting the collaborative construction of world-scale maps and models which support various data-driven geospatial applications.

Applications which exhibit or manipulate data on a global scale need highly scalable world models and data collections. To build world-scale visual models, the collection of images from which data is extracted must also be world-scale. The scale, granularity, and fidelity of the data extracted is generally determined by the set of available images and the ability of the system to find or group images related to a specific place or type of visual content. The set of available images and associ-

ated metadata form the basis of a model for the physical elements represented in the images.

The selection of an appropriate addressing scheme is key to the scalable and efficient management of world model data. Researchers generally focus their image database building efforts on small-scale purpose-built collections or on the aggregation of crowd-sourced images from services such as Flickr and Instagram. While small-scale purpose built collections are generally sufficient for the extraction of local visual information, they do not provide for a world-scale capability. Databases built from crowd-sourced images generally do possess the necessary scale, but lack in quality and granularity because images tend to be densely clustered near landmarks and points of interest but sparse elsewhere. In addition, Toblers First Law of Geography states,everything is related to everything else, but near things are more related than distant things.[107]. Previous efforts have respected Toblers Law with local and regional addressing methods, but the area is largely unexplored for fine grained world scale addressing. As such, image databases need to be adapted to be world scalable and support indexing of visual content that makes location integral to the addressing scheme.

The problem of aggregating a world-scale database of imagery is compounded by the dynamic nature of the world. As changes occur, the global database must be updated to remain relevant. The acts of creating and maintaining such a database on a global scale are very costly for a single organization. As an example, in order to build an image-based database for road navigation, Google hires cars to drive and take pictures along roads. For this effort to have complete global coverage in a static environment, it would need to cover 50 million miles of roads, the equivalent of circumnavigating the world 1250 times[3]. This is compounded when the environment is dynamic, where for every mapped road, the same road must be revisited often

enough to detect dynamic events such as new construction or the effects of disasters or accidents. Images and visually extracted information should therefore be considered perishable when their desired use is for inclusion in up-to-date world models. Due to these challenges, the task of collecting and indexing perishable visual data on a world scale is viewed as a problem which is inherently well suited for distributed and collaborative effort.

Thesis Statement.

This dissertation proposes that the formation of an effective self-building and self updating world-scale model comprised of user-contributed and user-annotated visual records can be accomplished in an efficient and scalable manner using a fully decentralized peer-to-peer approach with world scale, location based content addressing.

To support the thesis, this dissertation documents a unified series of research activities to develop a location-aware middleware framework for extensible multi-modal image annotation and collaborative visual information discovery. This framework, named PeerAppear, automates the capture, annotation, indexing and dissemination of visual information in order to facilitate the creation of world scale visual models. This visual information is expressed through various modes of image annotation by pre-existing or custom-built interpreters in the extensible framework. The PeerAppear network achieves scale through Kademlia-inspired indexing and a Bittorrent style peer-to-peer overlay network, which indexes the locations of user's image collections using hierarchical geographic segmentation cell addresses in a distributed geographic table. These cell addresses, derived using Google's S2 spherical geometry library, enable search to be executed in an efficient manner by targeting visual search queries at only those users most likely to produce results for a given geographic area.

Leveraging both the ability to construct localized semantic memories and the

ubiquity of global communications infrastructure, PeerAppear provides for the formation of a collective semantic memory by enabling clients to publish location-based summaries of their local geospatial repositories to a peer-to-peer distributed index. This distributed index, inspired by the Kademlia distributed hash table, uses Google's S2 spherical geospatial addressing scheme to provide hierarchical location-based addressing for published content. Unlike Kademlia's SHA-1 derived addresses, which do not provide meaningful information about the data from which they are derived, the spatial relationships of addresses used by PeerAppear are represented. Additionally, the XOR distance measure, adapted from Kademlia, effectively denotes the tree-based distance of hierarchical S2 cells, and a correlation between XOR distance and great circle distance is demonstrated. The addressing therefore enables search filters, comprised of S2 cell addresses, to be used to identify the clients participating in the network which are most likely to possess data for a given geographic area. Search queries comprised of compact data signatures can then be submitted to clients identified in the previous step for distributed search. By enabling the creation and indexing of searchable decentralized semantic repositories, the PeerAppear framework avoids many of the pitfalls and shortcomings of centralized data aggregation approaches. These include inability to scale, single point of failure, and accumulation of perishable and highly dynamic content.

To demonstrate and evaluate this thesis for practical application, the individual research efforts include case studies and experiments which exercise and analyze the framework's performance for collaborative visual localization. These evaluations include the participation of many simulated clients generated using datasets which are both randomly and manually constructed, storing images captured along intersecting routes in a specified test area. These images are annotated using the bag of visual words content-based image retrieval approach to enable visual similarity searches to

be conducted for images in the PeerAppear network with matching locations and similar content. The result of each evaluation is presented in terms of network performance, search precision, and search recall. Localization accuracy is also evaluated when using the PeerAppear framework to support a rudimentary visual localization application.

This work makes the following contributions:

- It exposes the problem of world model construction and identifies a feasible solution space along with key solution requirements. (Chapter II)
- It proposes a middleware framework for the collaborative construction of world models supported by a novel distributed geographic index using hierarchical cell-based geospatial addressing. (Chapter III)
- It presents computational support methods enabling image ranking, geospatial filtering, and image-based search within the proposed framework. (Chapter IV)
- It demonstrates the framework's use for decentralized, distributed, and collaborative practical visual applications including visual localization and location-based image search using real-world datasets. (Chapter V)

This dissertation is organized as follows. Chapter 2 describes the core problems this work aims to solve and maps out the research activities which develop and evaluate problem solutions. Chapter 3 presents the development of PeerAppear's core infrastructure which is designed to support the collaborative construction and maintenance of world-scale models and data repositories. Chapter 4 further develops concepts relating to the PeerAppear framework which provide computational support to visual world model and image repository application use cases. Chapter 5 ties the core infrastructure, presented in chapter 3, together with computational support

methods, described in chapter 4, in an integration effort resulting in a hardware-in-the-loop discrete event simulation tool which is used to evaluate PeerAppear for real-time and simulation-based performance using both simulated and real-world network participants. Finally, chapter 6 presents conclusions for the overall research and identifies key areas which should be further explored and developed through future work.

II. A Distributed Approach for Collaborative Image-Based World Modeling

In this chapter the problem of constructing systems to enable self-building and self-updating world models is exposed. A model-building framework concept is introduced and motivational applications driving design decisions for the world model framework are explored. Next, requirements that must be accounted for in the framework are enumerated. Finally, 3 primary research thrusts are described through which the framework development, computational support research, and evaluation are undertaken.

2.1 Self-Updating World Models

As humans we rely on an innate ability to sense, process, and store perceived information in order to construct and maintain a mental spatial model of the environment. This model enables us to navigate and function within the environment. Our mental model is also rich with semantic associations, including linkages between locations, imagery, names, colors, smells, and experiences. One of the model's greatest features is the enabling of visual and spatial correlation. By examining imagery captured at a location with which one is familiar, the location and linked information related to the location can be rapidly recalled from the mental model, thereby demonstrating the incredible information retrieval (IR) capability provided for by human cognition. Unfortunately, the data used to construct and maintain these models is restricted to the individual's experience and familiarity with respect to locale and memory, thereby requiring exploration or the absorption of second-hand knowledge in order familiarize oneself with new areas.

In an article advocating for the creation of self-building world models, Raquet describes a concept involving the use of technological constructs which mimic this

innate ability to build and access mental world models, demonstrated by our ability as humans to navigate visually[88]. Raquet's model is shown in Figure 1. Under this model, as an environment is perceived, spatial relationships for visual information are formed giving the model a basis for future inference. The concept is similar to simultaneous localization and mapping (SLAM), a technique which is popular in the world of robotic navigation for autonomous systems. One of Raquet's key insights was that human visual navigation is restricted to the familiarity of an individual with respect to locale and memory. Therefore, once transported to an unfamiliar area, the human can get lost. Raquet suggested that this limitation could be overcome within a technological framework for self-building world models through the creative use and ubiquity of modern communications technology.

In this work, a technological solution which builds upon Raquet's self-building world model concept is developed, extended, implemented, and evaluated. When framing the problem of constructing and maintaining self-building world models, two primary challenges stand out which direct the solution space. The first challenge is the scale of the world. Building a world model requires a global-scale mapping effort. This is not something that can be easily undertaken by a single entity. Even Google's StreetView mapping effort, perhaps the best funded and coordinated visual mapping effort to date, has not achieved complete global roadway coverage in the decade since its inception.

The second challenge is the dynamic nature of the world which constantly reshapes the visual landscape. The dynamism of the Earth comes as a result of both human and natural activity. Roadways, bridges, buildings and other man-made structures are constantly being built, demolished, or altered. People and vehicles are always on the move. The geological structure of the surface of the Earth also changes over time as a result of human activity and climate, seasonal, and natural geological effects.

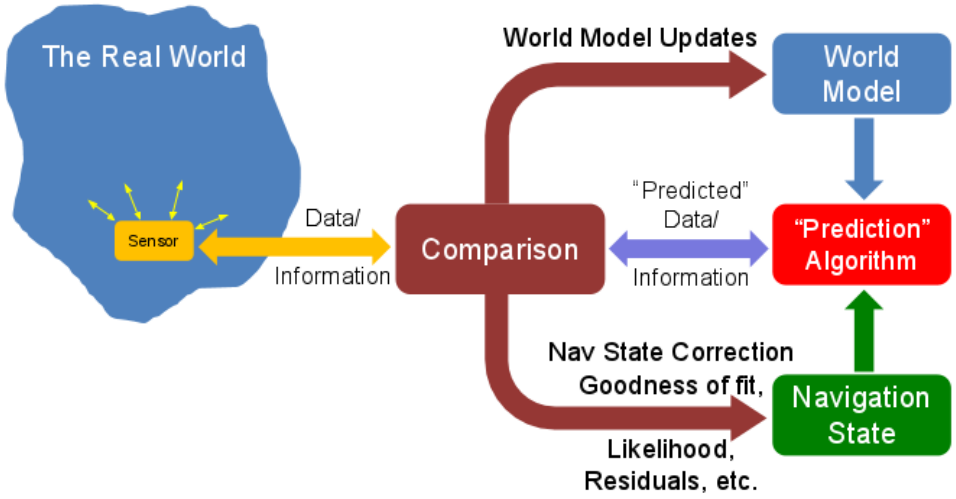


Figure 1. A self-updating world model concept diagram for navigation.[88]

This constant change requires visual mappings to be updated on a regular basis in order to remain representative of the world's current state in order to support real-time vision-based applications.

Due to the scale and dynamism of the world, the problem of world model creation and maintenance imposes coverage and scalability requirements that lead to an inherently distributed solution involving volunteer contributors. While this approach may improve coverage and responsiveness, it introduces the challenges of contributor scalability, quality and security from their reports. A distributed approach can also be centralized or decentralized with implications further affecting these challenges. Centralized solutions offer greater control over security and quality of contribution but can present single-point failure modes and struggle to scale. Decentralized solutions offer less control over user contribution, but tend to be more robust and scale well when properly organized.

In order to promote robustness and scalability of the system, both in terms of geographic scale and number of collaborative participants, this work envisions a peer-to-peer solution which is entirely decentralized. By narrowing the solution space to distributed and decentralized systems, the following key research questions must be

addressed:

- How does the system facilitate storage of world model data?
- How does the system facilitate data discovery and retrieval?
- How does the system facilitate collaborative construction and maintenance of the world model?

2.2 The Need for Collaborative Image-Based World Model Systems

The ability to determine location rapidly and accurately has become a highly sought capability for personal electronic devices and a necessity for autonomous systems. The majority of modern navigation systems rely heavily on the Global Positioning System (GPS) to provide this localization capability. One of GPS's primary shortcomings however is that it requires a clear line of sight to the sky. This limitation effectively prevents GPS from localizing in urban canyons, buildings, caves and underwater. In addition, the reliability of GPS can be negatively affected by both natural and made-made interference, such as space weather phenomena[10] or jamming[42].

The shortcomings of GPS-based localization have driven researchers to pursue a myriad of alternative localization methods which are capable of supplementing and often supplanting GPS. Some alternative localization capabilities make use of nearby signals of opportunity, such as WiFi networks and cell phone towers[13]. These methods however provide only a coarse estimate of position and are prone to large error.

Recent advancements in the field of computer vision (CV) have enabled vision-based solutions which demonstrate that highly accurate localization is possible through the tracking and analysis of visual features[2]. Much of the research in the area of

visual navigation is based on visual odometry, where relative rotation and translation between successive image frames are integrated to determine position[112]. While this approach is able to function without prior mapping, it is susceptible to drift resulting from accumulation of error in rotation and translation estimates.

This work is motivated by a visual localization methodology which operates by making position inferences based on visual search. The concept involves localizing imagery captured at an unknown or uncertain location by finding visual matches within a large repository of localized images. The searchable repository forms the basis for our visual world model with successful visual matches offering drift-free location inference.

Our earliest attempt at building a searchable repository of localized images focused on efforts to crawl and source images from online image sharing sites such as Flickr and Instagram. The area within 5 kilometers of the Washington Monument in Washington D.C. was selected for a test dataset and more than 250,000 localized Flickr images from the test area were downloaded over several days. The resulting dataset presented 2 primary issues. The first was that a significant number of the images contained highly perishable visual content which would not be suitable to support a visual localization capability. The second issue involved the density and spatial uniformity of distribution of images. Image density was greatest near the city's monuments and points of interest with few or no images in much of the search area. A spatial density plot of the images can be seen in Figure 2.

To overcome the challenges of non-uniform density and distribution of images, the idea of enabling a distributed and collaborative visual mapping effort was developed in the form of the PeerAppear middleware framework. The approach involves the passive collection and storage of images by network participants. The locations of images within the participants' repositories are summarized and published to a dis-

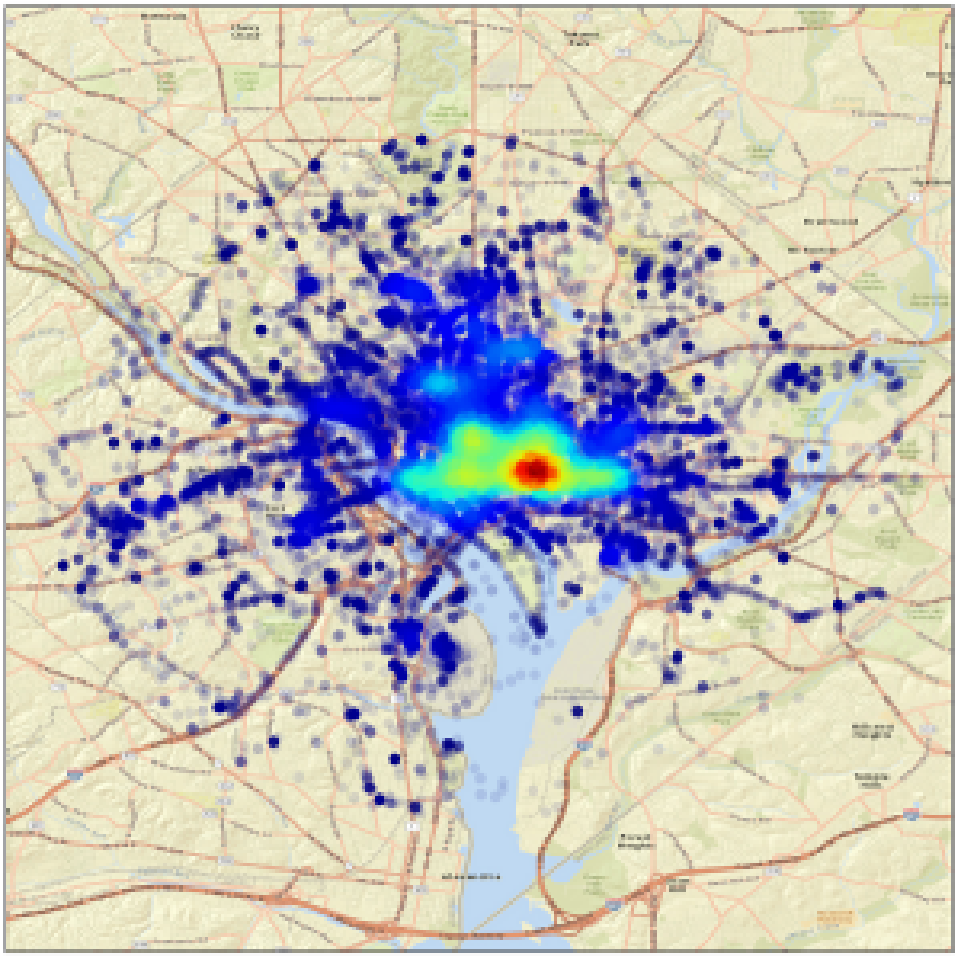


Figure 2. A spatial density plot showing the density of Flickr images located within 5km of the Washington Monument.

tributed index for discovery by other framework participants. This approach enables access to a global visual model which is contructed and maintained through a passive collaborative effort. The scale, density, and frequency of model updates depends entirely on the number and geospatial diversity of network participants.

The usefulness of the PeerAppear concept beyond the initial motivational application of visual localization became rapidly apparent after its conception. Environmental mappings for non-visual measurements can also be recorded and stored within PeerAppear. Examples include magnetic, barometric, and radio frequency mappings. This gave rise to PeerAppear's extensible image annotation aspects whereby captured images are stored along with other sensor data recorded at time of capture.

Alternative applications are also envisioned for the framework. A large decentralized visual mapping effort provides opportunities to construct a crowd-sourced capability similar to Google's StreetView project. However, unlike StreetView's images which are often years out-of-date, PeerAppear could display images for a specified location which were captured the last time the location was traversed. Given significant adoption of PeerAppear and a well-traversed location, this might mean images would only be a few minutes old.

Data from PeerAppear's visual world model could also be used as input for projects like 4D Cities[98]. This project uses a technique known as structure from motion (SFM) to construct 3D models of cityscapes. The construction of these models is organized based on the time the source images were captured, adding a time dimension to the model such that changes of the cityscape over time can be visualized.

A final motivational application envisioned for PeerAppear is distributed surveillance and forensics. Images stored and published by contributing participants in the framework can be retrieved based on time and location filters. This presents the opportunity to find visual records at or near locations where crimes were committed for

which there may not be video or images from traditional fixed surveillance systems.

The past decade has seen significant growth in the number and diversity of decentralized peer-to-peer internet services. These range from IP telephony enabled by services like Skype, file sharing enabled by protocols such as Bittorrent, and digital cryptocurrencies such as Bitcoin. Perhaps the most appealing aspect of these systems is that they are more democratic and not controlled by a single central entity. PeerAppear aims to follow in similar fashion by providing for the creation of artificial relational memory through symmetric peer-to-peer collaboration in a manner which eliminates payment models and single-point failure modes.

2.3 Requirements for Collaborative Image-Based World Model Systems

The problem exposition and application use-cases described above impose key system requirements upon the development and implementation of the PeerAppear framework which must be met through this research effort. These include:

1. Decentralized Operations

The envisioned framework constructs and maintains world-scale maps and models through decentralized and collaborative effort in order to leverage passive sensing and geographic scale of contributors. In order to promote scalability and to eliminate single-point failure modes, all framework operations must function through peer-based transaction.

2. Local Storage of Sensed Geospatial Data

The system is expected to primarily operate on visual data, which is often large, therefore it will be necessary for data to be stored in the local repository of the client which collects the data. By keeping data local it is expected that the availability of data will fluctuate as clients join and leave the network, however the effects of this fluctuation are expected to be mitigated by mass adoption of the system by

large numbers of users. A benefit of local storage is also expected to be resilience of the system to erroneous and malicious contribution by participants because their contribution will not be propagated or aggregated within the network.

3. Distributed Indexing for Location-Based Content

Because data is retained locally by the participants which collect it, the necessity exists for clients to inform their peers about the content which is contained in their repository. This introduces the requirement for a distributed index maintained through peer-to-peer transactions.

4. Uniform Elastic Global Addressing and Filtering

The distributed index must enable the identification of users who have content for a specified locale. This requires an addressing scheme to be used within the index which provides uniform distribution of data across the address space. The addressing must also be elastic, enabling variably granular filtering for flexible search space size and shape.

5. Fully Retrievable Location-Based Search

The system must be capable of finding all users who have reported owning data which was captured in a specified search area. Fully retrievable search has traditionally been a challenging problem for decentralized architectures with high levels of content duplication. It is expected to be of even greater difficulty when all clients have unique data.

6. Efficient Location-Based and Visual Search

The search for localized visual content must be accomplished in a computationally efficient manner to enable the system to operate on low-power mobile systems. This includes both location-based lookups within the distributed index as well as search based on visual similarity.

7. Scalable Operations

All framework operations must demonstrate efficiency when scaled both geographically and for large numbers of users. To quantify this requirement, we will assume that efficient scalability involves growth in computational complexity which is not worse than linear with respect to the increase in geographic size or numbers of nodes or images.

8. Support for Dynamic World

The framework must enable maintenance of the world model to account for dynamism in the physical world. This includes both the expiry of outdated content as well as the addition of new contributions from framework participants.

9. Equitable Distribution of Indexing Effort

The framework is envisioned as a purely decentralized effort with equitable distribution of index storage and communications traffic. This effort is expected to be proportional to the amount of data and number of peers represented with a given area. Because sensed data is retained locally by participants, the number and frequency of search operations is expected to be proportional to the size and scale of the client's local data repository.

2.4 Research Activities

This work is divided into three primary research thrusts which serve to constrain the scope of individual research activities and to encapsulate each activity such that it can be explored and evaluated to the fullest extent possible. Development work for all research activities was completed using open-source tools including Python, OpenCV, Matplotlib, Numpy, and GRASS GIS. Development and evaluation were accomplished on PC workstations and Raspberry Pi single-board computers using the Ubuntu and Raspbian operating systems. All equipment used for this research effort was purchased through an AFIT Faculty Research Council (FRC) grant.

Research Thrust 1: Collaborative Middleware Infrastructure.

The first research thrust, presented in Chapter III, addresses the problem of scalable location-aware distributed indexing to enable the leveraging of collaborative effort for the construction and maintenance of world-scale maps and models. This research thrust begins with the development of PeerAppear, a location-aware framework for peer-to-peer indexing, search and retrieval. Framework development primarily occurs in 2 stages. The first stage targets the development of algorithms and data structures which enable topology constructs for the PeerAppear routing table and distributed index. The second stage targets the development of a discrete event simulator, called the PeerAppear Simulation Engine (PASE), which is used to evaluate the PeerAppear framework. PASE enables simulated clients to be instantiated using data and locations which are randomly generated according to a specified distribution. The operating data and variables for all simulated clients is maintained in a simulation table and clients interact using the framework's primary remote procedure calls (RPCs) of PING, STORE, FIND_NODE, and FIND_VALUE.

Following framework development, the research thrust concludes with an evaluation of the framework which was designed to verify performance, scalability, and enable the tuning of system parameters for the distributed geographic index. The evaluation involves the use of PASE to execute large-scale scripted trials which vary the number of network participants, network parameters, and publish and search cell sizes. All trials are completed multiple times to enable the confidence of the results to be determined. The results of this evaluation demonstrate logarithmic growth of complexity for network operations in response to linear growth in the number of network participants, suggesting excellent scalability for the system. In addition, publish and search for both single-level and mixed-level operations is demonstrated showing excellent recall of published data.

Research Thrust 2: Computational Support for Practical Visual Applications.

The second research thrust, presented in Chapter IV, addresses computational methods which support practical visual applications within the PeerAppear framework. This includes the framework's visual search architecture, localized visual search methodology, geospatial representation and filtering, extensible image annotation, and area of visibility scoring.

The search architecture presented makes use of the Bag of Visual Words (BoVW) search paradigm coupled with term-frequency inverse-document-frequency (*tf-itf*) weighting. This methodology enables effective and efficient visual search to be accomplished using compact image signatures and highly parallelizable computational methods. The BoVW search methodology is used in concert with random sample consensus (RANSAC) to validate image matches using the epipolar constraint.

The framework's localized visual search methodology is presented next. This methodology is comprised by a two-stage process, including location-based search within the PeerAppear distributed index followed by search using BoVW signatures. This search methodology forms the basis for PeerAppear's visual localization capability.

Representation of position and geospatial filters is presented next. Because the visual content captured within an image is most often localized based on the position of the camera, search based purely on location is often not sufficient to find desired content which may have been captured from afar. A solution is presented which makes use of viewshed to find all area from which a location can be seen. The work is then extended to show that viewsheds from points along a path can be integrated to find the viewshed area along a planned route. A method of covering these viewsheds and reducing cell counts in order to form S2 filters for search within PeerAppear is

presented which makes use of a custom trie-based data structure and a density-based pruning approach.

A method for extensibility of the framework using third-party plugins is presented next. Extensibility enables the framework to be used for purposes beyond localized visual search. Examples presented include the annotation of images with WiFi SSIDs and signal strengths in order to map the locations of wireless signals. Another example involves the recording of magnetic field readings in order to construct fine-grained magnetic mappings in a collaborative fashion.

The chapter concludes by presenting early work into the scoring of imagery based on the area from which it is visible. This work makes use of the framework's visual search capability to find the area from which a specified visual content was captured. By scoring visual content based on visibility, the importance of certain images for visual localization can be determined, thereby enabling the possibility of caching of highly-ranked images and the expiry of low-ranked images.

Research Thrust 3: Integration of Middleware for Practical Application-Based Evaluation.

The third and final research thrust, presented in Chapter V, covers the integration of the PeerAppear Simulation Engine (PASE) with prototype hardware in order to conduct real-world evaluations of the framework. This hardware includes a Raspberry Pi single-board computer, monocular camera, and a sensor board.

This integration and evaluation is completed in two phases. The first evaluates visual search and localization of images using PeerAppear's image search capability. Data for this evaluation was collected using prototype hardware on routes which cover Area B of Wright-Patterson Air Force Base. From this dataset two specific routes were selected for evaluation. The first route was a large loop and included primarily

rural-type imagery. The second route was a smaller loop and included urban-type imagery. Results for both routes are reported in terms of search precision and recall as well as localization accuracy.

The second phase of the evaluation involved the integration of PeerAppear's distributed index with the visual search methodology. This evaluation was conducted using a dataset collected in downtown Dayton. The location was chosen in order to find imagery representative of urban canyons, one of the most frequently encountered GPS-denied environments. The dataset includes 2,700 images captured from 13 separate traversals. A final traversal was used to assess image search performance for the framework.

In this chapter the concept of self-building and self-updating world models was presented and motivated by examples of their practical application. Key problems were identified and a series of requirements were proposed for a collaborative image-based world model building framework. The development of this framework was described in 3 research thrusts. The first research thrust, presented in Chapter III, documents research and development of the framework's core indexing infrastructure. The second research thrust, presented in Chapter IV, describes algorithms, data structures, and methods which provide the necessary computational support to support practical visual applications within the framework. Finally, the third research thrust, presented in Chapter V, documents the integration of the core index infrastructure from Chapter III with computational support methods from Chapter IV enabling the evaluation of the full framework for application-based evaluation. In Chapter VI a summary, conclusions, potential applications, and insight for future work effort is presented.

III. Core Infrastructure Supporting Collaborative World Model Construction and Maintenance

Due to the dynamic nature of the world, the problem of constructing and maintaining relevant world-scale models generally requires significant effort to be spent on mapping. In this chapter a decentralized solution is proposed which enables the leveraging of collaborative effort through the implementation of a peer-to-peer middleware framework. The framework automates the indexing of sensed geospatial information captured and stored in the local repositories of participants in order to enable global information retrieval. Scale is achieved through a Kademlia-like overlay network which indexes data based on location by adapting Google's S2 hierarchical geographic segmentation scheme to a globally addressable distributed geographic table. Communications primitives allow search queries to be formed and executed, enabling the discovery of information published in a specified geographic area.

While this effort primarily targets the construction of visual world models, system abstraction allows the processes to be generalized to world model building for all types of sensed information. By extending the visual world model concept to include the storage and indexing of non-visual data, associations and linkages between different types of data can be established, thereby enabling retrieval of all related data for model queries.

The realization of this framework must address three primary challenges.

1) Global Availability: While the aggregate reach of individual sensor platforms may represent near global coverage, the challenge of combining individual local mappings into global-scale world models requires careful consideration. Previous efforts to construct world models collaboratively have principally focused on centralized aggregation, assembly and distribution of model data. While this approach may be suitable for compact sensor data, the bandwidth required for images and other types

of less compact data generally makes the practice infeasible. In addition, centralization poses other challenges such as single-point failure modes and scalability in response to model growth, number of users and bandwidth needs. Our early attempts [23, 22] at building visual world models from aggregators of crowd-sourced imagery also demonstrated that centralized solutions lack sufficient uniform density because manually volunteered images tend to be densely clustered near landmarks and points of interest, but sparse elsewhere.

2) Publish and Search in Dynamic Environments: The problem of aggregating a world-scale database of multimedia sensor data is further compounded by the dynamic nature of the world. Natural and human-driven change constantly reshapes the world, necessitating updates to maps and world models on a regular basis. This change often comes in the form of geologic activity, climate change, weather, disasters, and human activity. As changes occur, the global database must be maintained up-to-date in order to remain relevant. This requires both the inclusion of updated geospatial mappings as well as the expiry of those which are outdated.

3) Scalability: The scale of the world poses a significant challenge to the task of collecting, storing, and indexing globally-sourced localized data. The impact of scale can be addressed in terms of *i) lookup complexity* stemming from quantity of data and number of world-wide contributors and in terms of granularity and global reach of *ii) geospatial addressing*. The system must be resilient and adaptable to growth and variation in both.

Overcoming these challenges leads to a decentralized, distributed and collaborative solution to the problem of creating and maintaining world-scale maps and models for various types of localized data[88]. Meeting this objective requires the realization of a peer-to-peer middleware that utilizes a globally scalable universal addressing scheme for indexing geospatial data.

For these reasons, we present the PeerAppear framework. PeerAppear enables world-scale information discovery through the implementation of a peer-to-peer middleware framework which automates the indexing and sharing of sensed geospatial data stored in a decentralized manner within the local repositories of its contributing users.

PeerAppear achieves *global availability* of data, depicted in Figure 3, by allowing contributors and consumers of sensed geospatial data to discover *(1)* and interact with other global participants *(2)* in order to construct a distributed index supporting publish *(3)* and search *(4)* operations. By enabling a distributed mapping solution, PeerAppear is able to exploit a high degree of parallelism in order to frequently re-map often-traversed areas in order to provide for *publish and search in dynamic environments*. PeerAppear achieves *scalability* in terms of *lookup complexity* through an efficient peer-to-peer distributed indexing scheme which enables contributors to publish location-based summaries of their local content to a searchable distributed geographic index. The framework achieves *scalability* in terms of global reach and granularity through the application of an elastic *geospatial addressing* and filtering scheme based on Google's S2 Geometry Library which enables cell-based hierarchical addressing of the earth. PeerAppear's index constitutes a distributed geographic table (DGT) which enables the collaborative construction and maintenance of world-scale maps and models in an efficient and scalable manner.

This chapter makes the following contributions:

- It proposes a novel decentralized middleware framework supporting discovery and retrieval of geospatial information across a peer-to-peer overlay network, motivated by application-based use cases. (Section 3.2)

- It describes a distributed geographic index which tightly integrates a locality-preserving addressing scheme to enable variably granular geospatial representa-

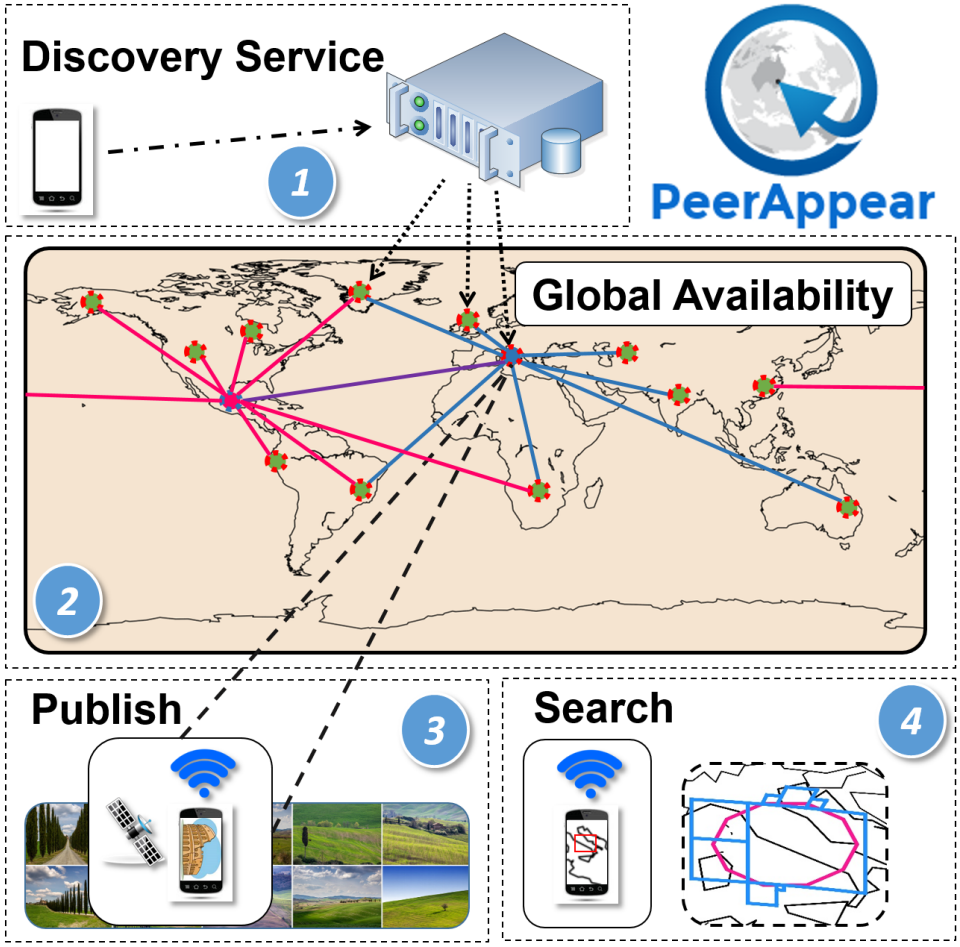


Figure 3. The PeerAppear middleware framework enables global availability of indexed data through a globally interconnected network topology and publish-search architecture.

tions of nodes and data within the framework. (Section 3.3)

- It presents a network topology enabling efficient and scalable logarithmic-time location-based search using novel geospatial filtering techniques and a heavily modified variant of the Kademlia overlay topology. (Section 3.3)

This chapter is organized as follows. Related works is presented in Section 2. Section 3 presents an overview of the PeerAppear framework including a high-level system description and an application-based use case. A low-level description of geospatial addressing, filtering, and network operations is presented in Section 4. An experimental evaluation of the system is presented in Section 5.

3.1 Related Works

Literature examined in this section covers work relevant to the development of algorithms and data structures supporting PeerAppear's distributed index. The areas covered primarily include *Geospatial Addressing* and peer-to-peer *Architectures for Decentralized Indexing*.

Geospatial Addressing.

The nearly ubiquitous availability of localization on portable computing devices such as tablets and smart phones has led to substantial growth in the area of location-aware mobile applications. Many of these applications submit queries to centralized information discovery services in order to find localized information near the user or in a specified area. To enable efficient response to user requests, this data is generally stored using spatial databases which support spatial measurements, functions, and predicates[19]. This section analyzes literature in the area of global geospatial addressing schemes and coordinate projections used to efficiently address geospatial information.

Early work in the area of spatial storage systems identified the need for address schemes which enable the efficient execution of area-based queries for multi-dimension addresses. One of the earliest and most popular of such structures, the R-Tree, was developed by Guttman[45]. The goal of the R-Tree was to enable databases to quickly retrieve items according to their spatial location. The R-Tree is a dynamic height-balanced hierarchical structure with data stored in the structure using leaf nodes. The parents of leaf nodes include coordinates identifying bounding boxes which tightly enclose the boundaries of their child nodes. This organization enables search queries to be executed using a tree traversal such that those nodes which do not intersect the query can be ignored. This has the effect of reducing the number of comparisons required for a given search from the naive $O(N)$ case to $O(logN)$ complexity. Guttman's R-Tree structure has withstood the test of time and is still a popular structure found in many modern spatial storage systems.

While the R-Tree structure worked well for systems using Cartesian coordinates, it was not specifically designed to handle spherical geographic coordinates. In the field of cartography spherical coordinates are often mapped to planar surfaces using rectangular projections. While these projections are useful for visual analysis of some geospatial relationships, the distortions they impose make them unsuitable for the direct computation of spatial operations. One specific mapping, the quadrilaterlized spherical cube, was introduced by Chan and O'Neil[18] and further developed by White et al[117] in order to mitigate the effects of this distortion. This mapping involved dividing the surface of a globe into 6 equal sections and projecting them onto the faces of a cube, as is shown in figure 4. This projection would later play an important role in future geospatial addressing schemes and for use on NASA's Cosmic Background Explorer mission.

One particularly promising approach for geospatial indexing is the Hierarchical

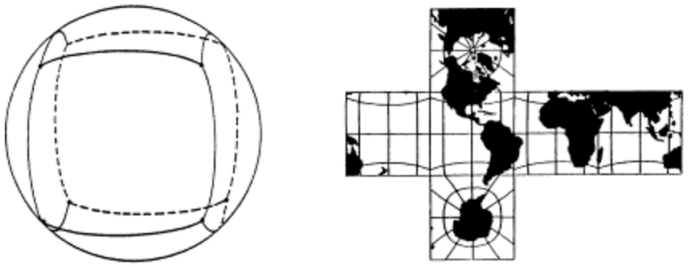


Figure 4. The quadrilateralized spherical cube projection.[117]

Triangular Mesh (HTM)[104] developed by Szalay et al. HTM uses a quad-tree structure which creates a hierarchical mapping of the globe using spherical triangles of similar shape and size. A decomposition of the globe for various subdivision levels can be seen in figure 5. The representation eliminates the distortion generally caused by change in longitude scale as the poles are approached as well as the singularities at the poles themselves. The authors performed an evaluation of performance for the library implemented using both C++ and C#. The system exhibited excellent performance for a data set consisting of a large number of triangular regions. One of the goals of HTM was to create cells of near-equal area for the same cell level. This is an important feature for data indexing because a uniform distribution of data within an address space enables predictable performance for data search and structure traversal. While HTM cells were similar in size, an evaluation conducted by the author showed a worse-case size ratio of 2:1 for larger cells of the same level.

Calabretta and Roukema attempted to solve the problem of mapping to equal area cells through the use of the Hierarchical Equal Area and isoLatitude Pixelisation (HEALPix) projection[16]. Their approach, described as an equal area pixelation, maps regions to hierarchical pixels which vary in size for different latitudes. The authors presented methods for rectangular, triangular, and hexagonal pixel decomposition. While the goal of their work was to define an equal area mapping to enable efficient storage for astronomical databases, HEALPix has found a following and

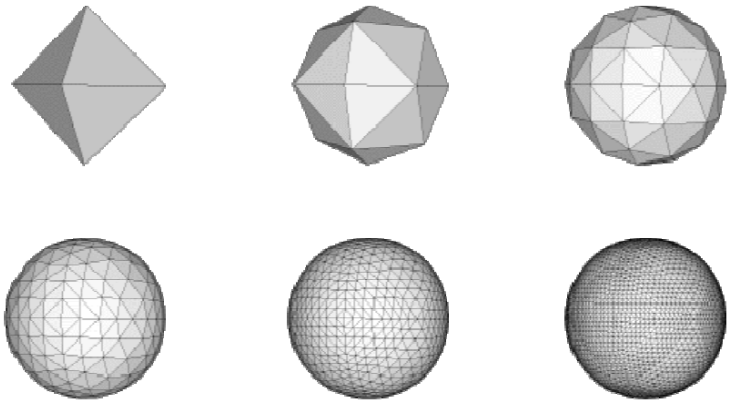


Figure 5. The Hierarchical Triangular Mesh global decomposition using spherical triangular segmentation.[104]

adoption in various mapping communities.

Large database indexing schemes have traditionally relied on indexing using a 1-dimension address to represent data. Geospatial data, which is often addressed using 2 or 3 dimensions, does not easily conform to a single dimension index. While some efforts, such as the R-Tree, have attempted to develop new data structures to accommodate geospatial data, other approaches seek to map the multi-dimension space to a single dimension address. Niemer developed one of the earliest and most common methods, called Geohash, to map latitude and longitude using space-filling curves. These "crinkly curves", as described by Moore[69], enable a hierarchical mapping of 2-dimension space along positions of the curve. A single Geohash can then be used in traditional database systems to index geospatial data. In general, the larger the prefix shared by 2 geohashes is the closer the locations are to each other. This relationship breaks down however near the equator, meridians, and at the poles. In addition, because latitude and longitude hashes are mapped directly to indexes on the space-filling curve, Geohash cells exhibit a wide range of sizes for the same cell level due to the distortion caused by merging longitude lines. Despite these shortcomings, GeoHash has found widespread adoption and is the subject of

continuing research efforts[103, 6].

To mitigate some of Geohash's shortcomings, Eric Veach from Google developed the Google S2 Spherical Geometry library[87]. The S2 library makes use of the quadrilateralized spherical cube method for subdividing the globe using the 6 faces of a cube. Each face is then mapped using a space-filling curve. The curve adopts an indexing scheme that enable hierarchical cells with up to 30 levels to be represented using 64-bit addresses. The hierarchical structure of the addressing scheme enables the relationships between cells to be determined by comparing the length of their common prefix. The mapping of curve indices to latitude and longitude is accomplished using a quadratic function, thereby mitigating the effects of longitude distortion and ensuring that cells of the same level are of a similar size. The S2 library was written in C++ and source code is available[12]. Implementations adapted for Python[32] and Java[41] have also been developed. The ever increasing number of new research efforts[66, 57, 26] in the area of large-scale geospatial database systems demonstrates that this field is currently undergoing rapid growth and active development.

Architectures for Decentralized Indexing.

The challenges associated with world model creation and maintenance leads to the idea of leveraging the efforts of a large network of volunteer contributors to achieve maximal scale and information density. The creation of a collaborative network for mapping is not an entirely new concept. The OpenStreetMap project [46] was founded in 2008 to enable crowd-sourcing geospatial information on a world-scale. The system's largest shortcoming however is the manual effort required for end-users to contribute. CrowdAtlas [113] attempted to solve this shortcoming by automating the process of identifying particularly important GPS traces and uploading them to the OpenStreetMap database. While the two combined efforts have proven to be effec-

tive for creating a geospatial information database for roadways, they are still entirely reliant on GPS and the data they provide does not effectively enable an extensible capability for the collection, extraction, and sharing of sensor data. In addition, the OSM project relies on a centralized back-end to handle the processing of user queries and contributions. This architecture, while common, presents a single point of failure by requiring all OSM network traffic to pass through the OSM servers and centralizing the storage of all OSM data.

Enabling decentralized capabilities on the internet has been the focus of many recent research efforts. One of the earliest mainstream focal points for peer-to-peer activity was the sharing of files which violated copyright laws. Early networks such as Kazaa, Limewire, and Napster relied on centralized peer indexing and discovery services to enable peer-to-peer transfer, while newer services such as Bittorrent have moved towards a decentralized approach through the use of distributed indexes. In more recent years, peer-to-peer overlay networks have found more legitimate uses, enabling scalability and decreasing bandwidth requirements for communication systems. Telephony services such as Skype and online games such as Frontier's Elite Dangerous have demonstrated that the peer-to-peer model can successfully be used for real-time consumer applications. The rise of Bitcoin[73] has demonstrated that peer-to-peer architectures can even be scalable, robust, and secure enough to enable completely decentralized currency transactions on a global scale. Using Bitcoin's blockchain ledger concept, Ethereum[118] has promised to leverage the distributed ledger to maintain global state information for a potentially unlimited number of applications.

This work focuses primarily on concepts relating to peer-to-peer overlay networks and decentralized geospatial information retrieval. Overlay networks are instituted for the purpose of organizing communication between network participants to enable

efficient and scalable systems. Communication between systems on overlay networks takes place over pre-existing network infrastructure, such as the internet. The vast majority of peer-to-peer networks fall into one of two categories, structured and unstructured networks[64]. In structured networks the organization and addressing of peers and data is tightly controlled. This structure enables decentralized indexing capabilities, often referred to as distributed hash tables (DHTs), which can guarantee that a given key will be found, if it exists, and provide provide for provable performance guarantees. Popular examples of structured networks in the literature include CAN[89], Chord[102], Tapestry[122], Pastry[91], Kademlia[68], and Viceroy[67]. The second overlay network category, unstructured, enables clients to connect to peers in a manner that generally does not require them to conform to a specific topology. Communication among peers on unstructured networks is generally limited to flooding or gossip, with each peer having only limited knowledge of local network peers. Because of these factors, unstructured networks are unable to offer performance or localization guarantees for data stored globally by peers in the network. While unstructured overlay networks were once the mainstay of internet file sharing systems, attacks on their centralized indexing services have forced nearly all to move to the use of DHTs in a structured topology.

One of the most promising structured overlay network topologies is the Kademlia DHT[68]. Kademlia organizes clients and data using a shared 160-bit address space. Clients participating in the network select their own address according to a uniform random distribution. Data is indexed into this address space using the SHA-1 hash of the data, which is also 160-bit. Key store and lookup activities within the DHT have proven worse-case complexity of $O(logn)$ due to a bucket system used to store lists of peers for each client. These buckets are structured in a manner that enables a client to maintain more fine-grained knowledge about peers which reside locally

in the address space than those that do not. When locating nodes for key pair storage or retrieval, a client performs a series of recursive lookups to identify the peers with addresses closest to that of the key. Those identified peers can then be sent *STORE* or *FIND_VALUE* commands to facilitate key storage or retrieval. One of Kademlia's most novel features is the *XOR* distance measure used to evaluate the distances between pairs of addresses. This measure enables peer-list buckets to be maintained based on *XOR* distance and solves boundary distance issues in the address space by considering the address space as being circular. Every communication that takes place between peers in the network includes the peer's address, thereby enabling peer buckets to grow passively. The number of peers stored in each bucket and the time between the re-publishing of keys are left as user-defined tunable parameters which can be adjusted based on the expected size and churn of the network. Many file sharing networks which previously used unstructured topologies, such as Gnutella, Kad, BitTorrent, and Overnet, are now using Kademlia-based DHTs for file indexing.

The use of peer-based networking solutions for the task of image storage, search, and retrieval is highly applicable to the research proposed in this work. Early research involving peer-to-peer network support for content-based image retrieval (CBIR) was conducted Muller et al with research primarily focused on unstructured networks. Muller's first efforts involved a system for search and retrieval of generic files through the development of Rumorama[71], a network architecture based on the PlanetP[28] gossip protocol. Rumorama enabled gossip-based dissemintation of 160-bit file summaries to peers within local PlanetP partitions. In the absense of network churn, the system demonstrated a theoretical complexity of $O(logn)$ for lookups. Muller's later work presented Plickr[70], a notional p2p implementation of the Flickr photo sharing service. Using user metrics from Flickr, the performance of the Plickr network was evaluated for both structured and unstructured p2p architectures. While the

author's evaluation of the work has since been made obsolete by recent advancements in network topology, making it of little relevance to the work described herein, the development of Plickr is still significant because it represents one of the first of few peer-based image retrieval systems described in the literature.

Later efforts by Ariyoshi et al presented PIAX[5], a structured peer-to-peer framework for the discovery of geolocated visual content. PIAX includes image search and retrieval functions enabled using a combination of geolocation and image features within search queries. A key insight presented by the paper is the fact that an image's tagged geolocation is not necessarily the location of the content. This insight is key for the viewshed search footprint work which is detailed in section 5.2 of this proposal. The purpose of the PIAX system is to enable a load-balanced CBIR capability for a large collection of georeferenced imagery. The system accomplishes this task by calculating geospatial centroids for the image collection, allowing it to divide the collection into Voronoi cells with near-equal density. Each cell is then assigned to a processing node in the network. These processing nodes are responsible for handling queries that match their geospatial area of responsibility. The image collection used for the network is generated using a web crawler that looks for geolocated images. The paper presents a preliminary experimental evaluation which analyzes the trade-offs made between search radius and computational complexity.

Most early research covering peer-to-peer overlay networks focused on indexing the content of the network using signatures based on binary summaries of a file's content, such as the SHA-1 hash. While this method has proven successful for generic file sharing applications, it does not enable a data exploration capability because content addresses are neither intuitive nor predictable. For user-generated geospatial content the ability to explore and find data based on intuitive and predictable addresses is key. One of the earliest peer-to-peer networks supporting such a capability,

GeoPeer[4], was presented by Araujo and Rodrigues. GeoPeer enabled nodes, identified using their physical location, to self-organize into cell-based topology structures using planar Delaunay triangulation[124]. Global interconnections within the network were facilitated by the maintenance of long-range contacts for each node. The network facilitated the transmission of data and the broadcasting of queries based on location. Therefore, nodes containing content relevant to their node identifier could be explored through the use of location-targeted queries. The authors identified support for location-aware mobile applications and location-aware sensor networks as key goals for their system.

Expanding upon the concepts originated in GeoPeer, Kovacevic et al developed Globase.kom[53] in order to enable efficient peer-to-peer location-based search on a global scale. Globase, short for geographical location based search, enabled peer-to-peer service and content discovery using a super peer topology which was organized based on client locations within rectangular geographic partitions on a Plate Caree projection. Each rectangular partition was managed by a single super peer responsible for handling queries and responses to and from nodes in other geographic partitions. A hierarchical partition structure enabled overloaded super peers to subdivide their partition and assign the newly created partition to an appropriate peer within it. Fault tolerance was achieved through the maintenance of a super peer backup list for each partition. The system was evaluated using a network simulator and metrics recorded included number of hops, operation duration and relative delay penalty. The network's performance was also evaluated for various churn levels, thereby accounting for the effects of one of the primary obstacles faced by peer-to-peer networks. Additional tests evaluated lookup and search performance, communication overhead, and the network's ability to automatically balance loads.

Making slight improvements upon the approach used for Globase, Asaduzzaman

and Bochmann developed GeoP2P[7]. GeoP2P used a similar hierarchical approach for subdividing the topology based on client location. Unlike previous efforts, GeoP2P did not rely on pre-defined geographic partitions. The system's primary goal was to support dynamic hierarchical organization while minimizing network perturbations caused by churn. To achieve that goal, the system managed rectangular partitions dynamically, with splits and merges occurring when minimum or maximum threshold values for partition size were reached. One novel feature offered by GeoP2P was the ability to transmit messages targeting all peers in the same zone as the sender, all peers within a given geographic area, or the closest peer to a specified point.

The approaches taken by GeoPeer, Globase.kom, and GeoP2P were peer-based, but their use of super peers kept them from being entirely decentralized. Building upon the success of the Kademlia DHT, Picone et al introduced GeoKad[79], the first fully decentralized location-based peer-to-peer topology. GeoKad adapted the Kademlia DHT to use 2-dimension spatial coordinates instead of 1-dimension file hash descriptors. In doing so, the authors coined the term distributed geographic table (DGT) which was more appropriate when applied to geospatial indexing applications. While GeoKad adopted most of Kademlia's features, the representation of addresses using spatial coordinates was not compatible with Kademlia's XOR distance measure. Therefore, Kademlia's buckets were re-envisioned as concentric rings called GeoBuckets which store lists of peers at various distances, measured in Euclidean space, from the node. The system was designed for mobile participants. To account for mobility the address assumed by each node is determined based upon the node's current position. The authors present an evaluation of their system which was completed using a discrete event simulation tool. Metrics evaluated in their simulation include message rate, miss ratio, number of peers, and other qualitative factors. The authors present additional papers[15, 83, 81, 82, 78, 80] which build upon GeoKad

and demonstrate its use for vehicular and city-based infrastructure applications.

Following a similar approach to that used by GeoKad, Gross et al created Geodemlia[44] to add search features and make refinements to the addressing scheme. While buckets in both GeoKad and Geodemlia are defined using concentric partitions at a specified radii, Geodemlia further subdivides the partitions based on cardinal directions. The resulting buckets produce better locality for the clients contained within because they share a common bearing from the client. An additional feature implemented by Geodemlia is the reproduction of key-pairs stored locally by neighboring peers. The evaluation of Geodemlia was accomplished using PeerfactSim[72].

One liability shared by Globase.kom, GeoKad and Geodemlia is the requirement to support radius-based search. While the concept of searching for data near a specified center point may seem intuitive to humans, it restricts the shape of search areas and imposes the requirement of perfect locality preservation on the address space. Through the use of an elastic global addressing scheme, we seek to overcome this limitation within the PeerAppear framework.

3.2 PeerAppear Middleware Overview

Many applications, such as simultaneous localization and mapping and structure from motion, require the collection and accumulation of environmental data in order to construct maps and models of the local environment. While these activities are traditionally accomplished independently, there is great potential for improvements in efficiency and awareness through passive collaboration with peers. PeerAppear facilitates this passive collaboration by enabling clients to effectively leverage the knowledge of their peers in order to extend their own environmental awareness while making their own knowledge available to be leveraged by others.

Publish and Search Functionality.

This motivational application involves localizing visual content. An image captured at an unknown location is used as a query to match against localized images within a repository. Successful matches, once validated, offer inference opportunities for the location of the query image.

1) Populating the Index: In order to enable a distributed search capability PeerAppear first requires the construction of an index which facilitates the identification of peers within the network who have images that might be relevant for a specified search area. Figure 6 shows the process of summarizing a local inventory into a publishable list of cell identifiers. The process begins by constructing a list of locations for all localized data within a client's local repository. The red line represents the locations where images in the client's local repository were captured. A cell-base covering, depicted as orange boxes, is then applied in order to summarize the area of the client's data and to reduce the number of addresses which must be published. Finally, each of the cell identifiers is published into PeerAppear's distributed index along with the network address of the owner to enable direct peer-to-peer communication.

2) Search: Figure 7 depicts the process of using PeerAppear to find relevant image matches. This process begins with an image and an approximate location of where it was captured, represented by the magenta circle. A covering of this area is applied in order to construct a list of cell identifiers (blue boxes) for use as a search filter within

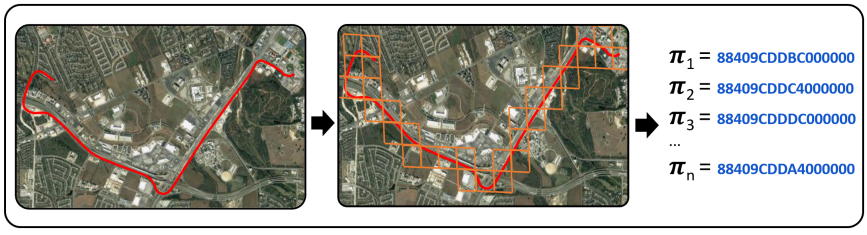


Figure 6. Populating the index: The process of summarizing local data into a set of cell addresses to be published to the DGT. Image capture locations (left) covered as cells (middle) to reduce the number of published addresses (right).

the global index. A lookup for each identifier within the index then produces a list of clients which possess potentially relevant imagery (yellow boxes). Because keys published into the distributed index are paired with the network address of the owner within the underlying network topology, direct communication with the client can be accomplished to exchange both the image query and potential results. In prior work [22, 23] we successfully demonstrated the use of bag of visual words (BoVW) image signatures to facilitate this final step.

PeerAppear Role-Based Process Interaction.

This section documents the processes used by nodes participating within the PeerAppear framework. The specific actors involved in the activities are described as well as the interactions that take place for each specified process.

Actors.

PeerAppear is a decentralized index comprised of symmetric nodes operating within a homogeneous topology. Unlike competitors such as Globase.kom [53], PeerAppear does not use superpeers and all nodes within the network are of equal status. PeerAppear's only centralized component is a peer discovery service.

PeerAppear Discovery Service

The PeerAppear discovery service enables nodes to find active peers within the

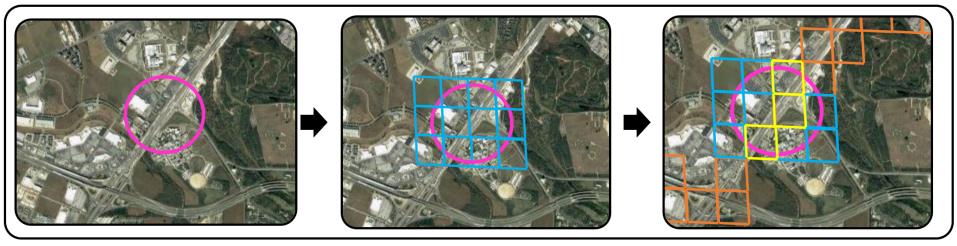


Figure 7. Search: The process of performing a DGT lookup for keys matching a specified search filter. The potential search location (left) is covered with cells (middle) used to narrow potential repositories for search (right).

network. While all operations within the network are decentralized, nodes must first populate their routing tables with other known peers to enable access to the index and to effect index operations. Population of the routing table, described in the *Bootstrap* process below, can be accomplished through contact with a single well-connected node.

If a node is unable to contact other previously encountered nodes, or if the node is starting for the first time, the PeerAppear discovery service provides contact information for recently connected nodes including the client's *IP address*, *port*, and *node address*. When contacting the discovery service the node also provides this information for itself, thereby enabling the discovery service to offer the node's information to other nodes searching for peers.

PeerAppear Node

Nodes within the PeerAppear network are symmetric. Each node carries the same status, precedence of traffic, and interacts using the framework's RPCs which are described in Section 3.3. Nodes are responsible for maintaining a local index inventory for keys sent to them using the *Publish* RPC.

Contribution of data to the network is optional. Adopting terms from Bittorrent, nodes which share local content are referred to as *seeders* while nodes which only consume network content are referred to as *leechers*. Regardless of a nodes status as a seeder or leecher, it must still maintain storage for keys published near its address and respond appropriately for all PeerAppear RPCs.

PeerAppear Index Processes.

The PeerAppear index uses 3 principal processes for participation within the distributed index. These include **Bootstrap**, **Publish Content**, and **Search for Content**. Each is described along with the roles of the other participating actors below.

Bootstrap

When a node first launches it must perform the bootstrapping process in order to populate its routing table and gain access to the index. The process requires knowledge of at least one active and well-connected PeerAppear node. Typically this node would be known from prior contact made during previous sessions.

If no available nodes are known from previous sessions, or if the client is launching for the first time, the PeerAppear discovery service can be used to find other active nodes. The address for the discovery service, PeerAppear's only centralized component, is static and known a priori. When contacting the discovery service the node's own node address, IP address, and port are provided. The discovery service returns a list containing this same information for several recently connected nodes.

Information for known nodes, learned during prior sessions or from the discovery service, is used to initially populate peer lists within the node's routing table. The *FIND_NODE* RPC is then executed using the node's own address. This allows peers in close proximity to the node to be discovered. All discovered peers are added to the node's routing table, thereby enabling connectivity to the global topology.

Publish Content-Based Keys for Local Repository

Seeder nodes must regularly publish keys into the network which represent the locations of content in their local repositories. Publishing a key begins with the execution of the *FIND_NODE* RPC using the address of the key for the RPC. Once the RPC has completely executed, up to k peers closest to the key's address will be found. For each of these peers, the *STORE* RPC is issued instructing the peer to store the specified key. To account for churn in the network and the inclusion and expiry of content, keys must be re-published according to a network-wide interval.

Search for Content Owners

PeerAppear's search process is used to discover owners of localized content within

the network. The search process makes use of the *FIND_VALUE* RPC. Before executing a search the S2 cell address or addresses describing the search area must be identified. Each address is then used as input for the *FIND_VALUE* RPC. Results from each RPC execution include any published key pairs with addresses that have prefixes matching the specified search address. These keys provide contact information (IP address and port) for nodes participating in the network which have reported having content in the specified search area. Additional content-based search can then be facilitated through direct peer communication.

System Requirements and Assumptions.

The primary purpose of the PeerAppear framework is to enable a decentralized information retrieval system allowing clients to identify and retrieve potentially relevant volunteered geospatial data made available by other peers. To achieve this purpose, PeerAppear maintains an index which keeps track of participating peers and summaries of the data that each possesses. This index, called a distributed geographic table (DGT), organizes data based on its geographic location. Unlike centralized indexes or centralized aggregation, the selection of a decentralized approach avoids many common pitfalls such as single point failure modes, bandwidth limitations, and inability to scale with growth in users, traffic, and storage requirements.

The operation of the PeerAppear framework relies on several key assumptions. The first is that all clients are able to communicate directly with all other clients using the underlying network topology, as is the case with internet protocol (IP) enabled communications infrastructure. The second is that all clients are able to localize themselves with some reasonable degree of accuracy. This is most often accomplished using GPS, WiFi SSID lookup, or cell tower triangulation. Finally all clients must be able to sense their local environment and store localized data about

the environment in a database for their own use and to share with others.

While the collaborative nature of the framework presumes network connectivity, execution and storage elements are structured in a manner allowing local mappings to be constructed and used independently in the absence of network connectivity. Each participant maintains their own independent database containing local knowledge of environments which they have mapped. This data is retained locally to the extent their storage capacity allows, and data is not replicated by the network as a means of preventing the propagation of malicious or erroneous contributions. The framework instead relies on mass adoption to achieve a user base significant enough to maintain a mapped presence for most well-traveled areas.

Realizing a system fulfilling the goals and requirements of the framework described above necessitates an efficient location-aware overlay network topology which makes use of an appropriate and compatible geospatial addressing scheme. PeerAppear meets this need through the development and tight integration of a highly modified variant of the widely used Kademlia topology and the Google S2 Geometry Library. In the next two sections we address the overall system requirements, shown in Table 1, which drive the selection of PeerAppear's addressing scheme and network topology as well as the implications of and adaptations needed as a result of their combined integration.

Middleware Overview.

The PeerAppear middleware is comprised of three primary elements including an addressing scheme, network topology, and protocols supporting the publishing and searching of localized data represented within the distributed index.

Table 1. Framework elements used to meet system requirements.

	Global Availability	Dynamic Environments	Scalability	
			Elastic Global Addressing	Lookup Complexity
S2 Addressing	X		X	
Network Topology	X	X		X
Publish and Search Protocol	X	X	X	X

S2 Addressing.

The PeerAppear framework overcomes the challenges of data availability and scalable coverage through the integration of Google's S2 geometry library as a principle means of localizing users and data in both local repositories and globally in the distributed index. Indexing data using multi-dimension location addresses, such as latitude and longitude, has traditionally posed significant challenges for information retrieval systems. Many of these systems, which generally rely on single-dimension keys for indexing, are not well-suited for multi-dimension indexing. Multi-dimension coordinate systems are generally a better fit for spherical addressing, but require pairwise comparisons to determine relevance in response to area or radius-based search queries. Such comparisons must account for spherical distortion, polar anomalies, the equator and meridian. Solutions using grid-based hashing schemes, such as GeoHash [74], have gained some popularity as a means of partitioning localized data in order to simplify these comparisons, but suffer from non-uniform distribution of data across the address space because spherical distortion is still unaccounted for, leaving grid cells to cover greater area at the equator than at the poles. The S2 addressing scheme overcomes these challenges by mapping the Earth using a quadrilateralized spherical cube and further mapping the faces of the cube using quad trees sequenced by a

Hilbert space-filling curve, resulting in a cell-based mapping with near-equal-area cell sizes for each hierarchical level.

Network Topology.

PeerAppear maintains an index mapping the locations of data collections in the network to individual users to enable clients to identify which peers have data that may be relevant for a location-based search. Due to the dynamic nature of the data available on the network at a given time, the aggregation and maintenance of a single centralized index would be challenging and costly. PeerAppear therefore adopts a Kademlia [68] inspired DGT approach for maintaining a distributed index of clients and representations of the locations of the data collections they store. The routing table structure established by the DGT ensures that data within the index is globally available and provoides for efficient lookup complexity in order to support the high operations throughput necessary to enable frequent remapping necessary for dynamic environments.

Publish and Search Protocol.

PeerApper uses a publish and search protocol which adapted from Kademlia in order to support the integration of S2 addressing the the execution of location-based operations. The protocol enables data to be published and queries to be executed using 64-bit S2 addresses for all hierarchical levels. While Kademlia's publish and search protocols are only capable of supporting exact key matches, PeerApper enables mixed-level publish and search in order to allow the discovery of corresponding data based on the hierarchical relationship of query and publish cells.

3.3 Peer-to-Peer Distributed Geospatial Indexing

In this section we describe the technical underpinnings of the PeerAppear framework and present the manner in which it integrates underlying technologies. The includes PeerAppear's geospatial addressing and indexing mechanisms, publish and search protocols, and the implications of their integrated application.

S2 Geospatial Indexing.

PeerAppear uses S2 cell addresses as keys for indexing data stored within the cell. The addressing scheme also enables variably granular representations of data through hierarchical partitioning. These keys can be used to describe the content of a data repository in a compact manner suitable for publishing to the distributed index. They are also used when submitting queries to the index.

Elastic Granularity.

The S2 library enables a one-dimension global addressing scheme by subdividing the Earth using a hierarchical cell-based partitioning scheme. The partitioning begins with a quadrilateralized spherical mapping which divides the surface of the earth into 6 equal area-faces. Each face is further subdivided into near-equal-area cells in a hierarchical fashion using a quad tree. The sequential addressing of each cell is accomplished using a Hilbert space filling curve, as shown in figure 8.

The size of cells in the hierarchical subdivision range from approximately 1 cm^2 for the smallest (level 30) cells to approximately $85,000,000$ km^2 for the largest (level 0) cells covering an entire cube face. All addresses in the S2 library use 64-bit representations, with the hierarchical structure of the quad trees represented in bit order in the address. This enables smaller cells to be grouped and represented by a larger cell address using the largest common prefix of the set of smaller cells, as shown

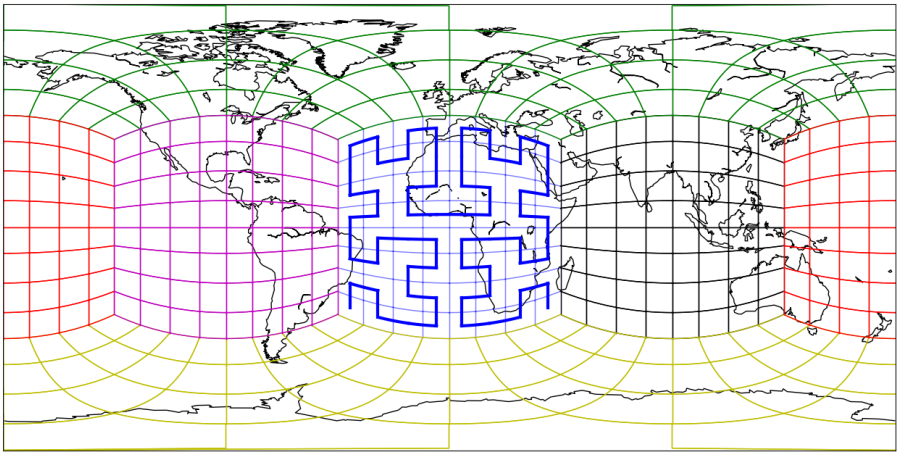


Figure 8. A mapping of the globe using Google's S2 Spherical Geometry library with level 3 cells. Center cube face (Africa) shows sequential addressing using Hilbert mapping.

in figure 9. The S2 address encoding enables PeerAppear to represent the locations of users and user data with varying levels of granularity in order to provide sufficient fidelity to target search queries while still maintaining an abbreviated representation suitable for efficient peer-to-peer indexing.

Because of the varying granularity offered by the S2 cell partitioning, geospatial filters can be constructed which conform closely to most polygons using a heterogeneous set of S2 cells. Figure 7 depicts an S2 cell covering corresponding to a search radius. An item found to have a matching prefix with any member of the search set would be identified as being within the search area. In the figure shown the search

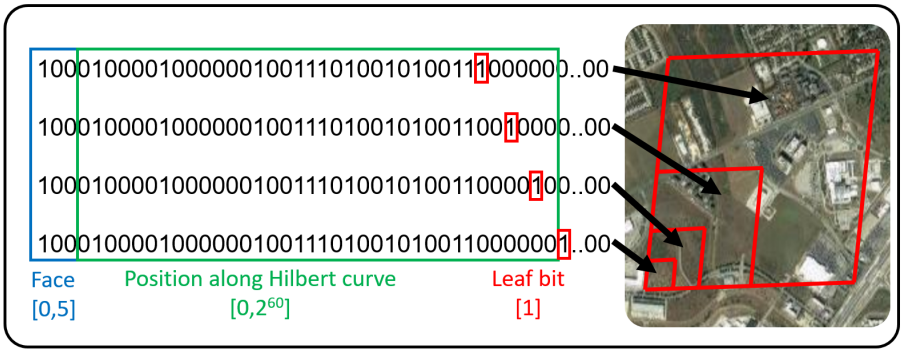


Figure 9. A depiction of hierarchical S2 cells and corresponding cell prefixes.

filter is comprised of 9 S2 cells. By using an increased number of smaller cells the conformity of the filter to the search radius can be improved with a penalty to the efficiency of the search. Conversely, a filter consisting of fewer cells can be formed in order to improve efficiency at the cost of loss of conformity with the specified search area.

Compact Bitwise Radix Tree Inventories.

PeerAppear maintains the cell address inventory for each node in a radix tree data structure. The inventory includes cell addresses for data stored in the node's local repository as well as cell addresses published to the node by peers. This radix structure enables efficient lookup, insertion, and deletion with $O(logn)$ time complexity. Lookup operations are especially efficient for heterogeneous level cell operations because of the inherent parent-child relationships used by the radix tree for cells with matching prefixes.

Networking in an S2-Integrated Kademlia Variant.

PeerAppear's DGT primarily differentiates itself from Kademlia through its use of the S2 library's 64-bit addresses instead of Kademlia's 160-bit addresses. Kademlia's addresses, which are generated using the SHA-1 hash function for files and randomly assigned for users, bear no useful information about the data they represent or the users to which they are assigned. Their primary function is to ensure uniqueness, enabling files shared on Kademlia-powered networks to be uniquely identified and indexed with little chance of collision. This presents challenges however because the 160-bit identifier of a file must be known prior to searching for it on the DHT. Because S2 cells represent known geographic space, the mapping of geographic space to S2 cells enables search to accomplished without prior knowledge of what has been published

to the DGT. This promotes global availability of data within the network.

Global Availability in a Peer Routing Table.

In order to facilitate a contiguous communications capability across the PeerAppear network, each node stores contact information about other peers in lists organized by address. These lists, which are each limited in length to k peers, are structured to ensure that space for knowledge about other peers whose addresses are distributed across the network's address space is reserved. These lists are not uniformly distributed across the address space. Lists are instead organized to ensure that nodes have increasingly greater knowledge of peers near their own assigned 64-bit address, as determined by the XOR distance between the node's own address and those of the known peers, as shown in figure 10. Because each node maintains fine-grained knowledge about peers near its own address, the likelihood of being able to respond to a *FIND_NODE* or *FIND_VALUE* RPC with nodes closer to the target address is greatly increased. This routing structure is also key for enabling logarithmic time complexity for DGT operations.

PeerAppear nodes use an adaptive peer list structure which organizes peers based on their XOR distance from the node, $PEER\_ADDR = NODE\_ID \oplus PEER\_ID$. Initially, only a single peer list is maintained. When the list exceeds k peers it is split into two lists each covering half of the range of the list they replace. This process of splitting lists continues as additional peers are added, however only the list with range including $PEER\_ADDR = 0$ can be split. This rule ensures that nodes track greater numbers of peers near their own *NODE_ID*. A geographic mapping of the adaptive list structure for a fully populated and enumerated peer list is shown in figure 11. For each known peer, a node stores the necessary information required to enable direct communications within the underlying network, including *(PEER_ADDR, [PEER_ID,*

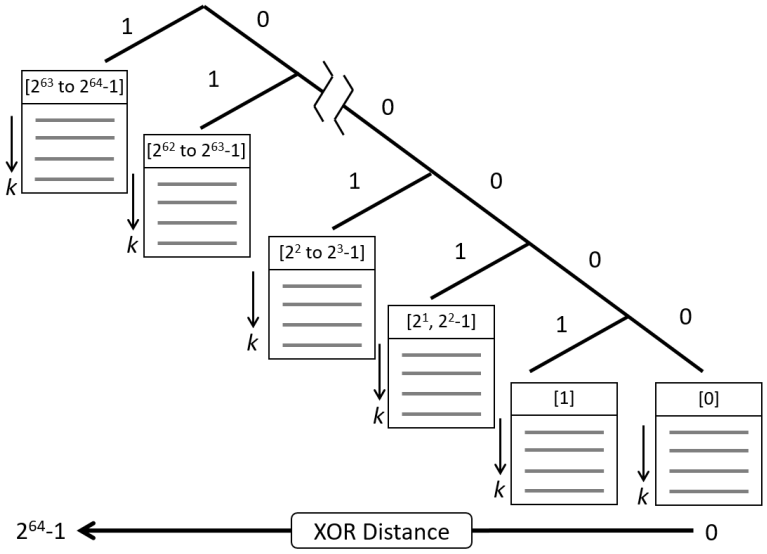


Figure 10. The tree structure of fully populated peer lists of length k. Nodes with the furthest XOR distance are on the left and closest XOR distance are on the right.

PEER_IP_ADDR, PEER_PORT]).

XOR Compatible Geospatial Sharding.

One of Kademlia's key contributions to the area of distributed indexing is the introduction of bitwise XOR as a means of determining distance between pairs of binary addresses. The use of XOR enables nodes to determine how "close" an address is to their own or that of another node, with the XOR distance effectively denoting the magnitude of the difference between two addresses. The XOR measure also provides for unidirectionality. With Kademlia's address space envisioned as a binary tree, selecting nodes based on smallest XOR distance ensures that convergence occurs along the same path, thereby supporting full retrievability. Organization of a node's routing table is also accomplished based on the XOR distance to known peers. The peer "buckets" created by this organization enable sharding for the distributed index.

The XOR distance measure also pairs well with S2 addresses because of their bit-wise representation of hierarchical relationships. This makes XOR especially useful

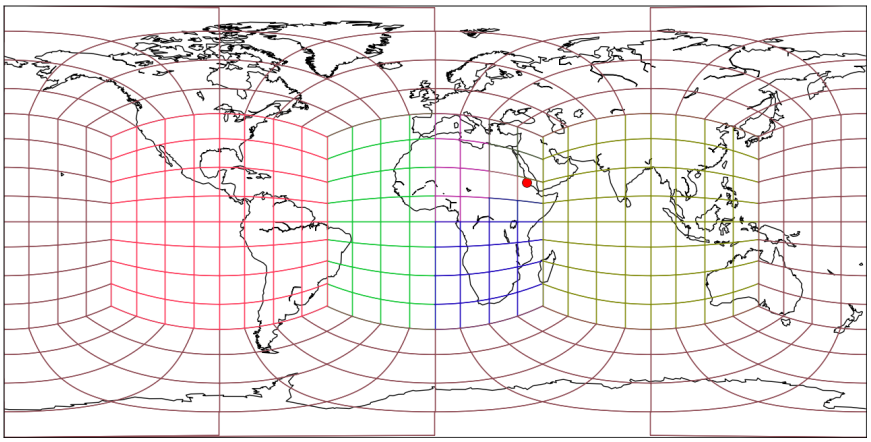


Figure 11. The global mapping of individual peer list coverage areas, each containing up to k peers, shaded by color for a randomly generated node (red dot).

for determining relationships between pairs of S2 cell addresses within PeerAppear because it enables the identification of parent-child relationships within S2's hierarchical structure. The smaller the magnitude of the XOR of 2 cell addresses, the nearer they are to one another in the tree.

As with Kademlia, nodes in the PeerAppear network have addresses which are drawn from the same address space used to identify data. Nodes are responsible for indexing data with addresses that are "close" to their own address. Because addresses for files indexed by Kademlia are based on the file's SHA-1 hash, which generally follow a uniform random distribution, it makes sense that node addresses in Kademlia are generated according to a uniform random distribution. Addresses within PeerAppear are representative of data location, therefore random node address assignment is inappropriate. PeerAppear clients instead assume a 64-bit S2 address corresponding to their location at time of startup rather than a randomly generated one. This benefits both lookup complexity of nearby addresses as well as load balancing for the DGT because areas with the greatest numbers of users also collectively have the largest data repositories, thereby enabling a type of geospatial sharding for the distributed geographic index.

Publish and Search Protocol.

The PeerAppear framework enables DGT access through heavily modified variants of Kademlia's three primary remote procedure calls (RPCs): *FIND_NODE*, *FIND_VALUE*, and *STORE*. These RPCs are further described in Table 2. Each RPC sent within the PeerAppear network includes the *[PEER_ADDR, PEER_IP_ADDR, PEER_PORT]* triplet containing information about the sender. This allows the recipient to add the sender to their peer list if room is available. PeerAppear incorporates these RPCs into its three main operations: *Node Lookup*, *Area Lookup*, and *Publish Inventory*.

The *Node Lookup* operation enables search to be effected in order to find network clients whose identifiers fall closest to a given query address. The lookup operation begins with a node sending the *FIND_NODE* RPC to the α nodes it knows of which are closest to the query address. The operation continues iteratively until the closest k nodes it has discovered to the query address have been contacted without yielding any closer contacts.

The *Area Lookup* operation enables search to be effected in order to find keys which have been published into the network for data matching a query's specified location filter. This operation begins with a list of S2 addresses for the location filter and iteratively builds a list of matching peers using the *FIND_VALUE* RPC, excluding duplicates.

The *Publish Data* operation enables clients to publish key pairs which identify them as owners of data which falls within the bounds of a specified area. For each S2 address from the list which summarizes the node's local data, the *FIND_NODE* RPC is used to build the list of k peers responsible for indexing data at that location followed by the *STORE* RPC to each to instruct the nodes to record the values into their local radix tree inventory.

Table 2. Remote Procedure Call (RPC) functions used to manage nodes and data within PeerAppear.

RPC Name	Description
FIND_NODE	The FIND_NODE RPC returns [PEER_ID, PEER_IP_ADDR, PEER_PORT] for the k nodes from its peer lists which are closest to a specified lookup address.
FIND_VALUE	The FIND_VALUE RPC performs the same function as FIND_NODE, but also returns any items from its index inventory which match the specified lookup address. For an inventory item to be considered a match, it must meet one of three criteria: the addresses must be equal, the lookup address must be a child of an inventory cell address, or the lookup address must be a parent of an inventory cell address. These matching criteria ensure that lookups with addresses specified at any cell level successfully match to corresponding inventory cells which can also be published at any level.
STORE	The STORE RPC instructs a node to insert the [CELL_ADDR, NODE_ADDR, PEER_IP_ADDR, PEER_PORT]} triplet into its index inventory.
PING	The PING RPC is used to determine if peers listed in a node's routing list are still connected to the network.

Because the index is expected to be subjected to high churn rates, the information stored in peer lists and index inventories must be refreshed periodically. Peer lists are maintained using a *ping* for peers a node hasn't communicated with in t_p minutes. Nodes that fail to respond to the *ping* are removed. Inventories are maintained through a perish and re-publish process. Inventory items older than t_{rp} are automatically removed. Nodes must therefore re-publish their repository's cell summary list every t_{rp} minutes. Our system defaults to a t_p of 5 minutes and a t_{rp} of 15 minutes.

When a PeerApear node first starts it bootstraps its peer list by inserting nodes it saved from previous sessions or nodes it learns about through a centralized discovery service. The node then initiates a *Node Lookup* operation using it's own node address as the query address. This process serves both as a means of discovering nodes to insert into it's own list as well as a means of informing other nodes of its existence. Finally, in order to bootstrap its own index inventory, the node sends a request to each of its k closest nodes for a compressed inventory transmittal. These inventories are merged, excluding duplicates, and inserted into its own index radix tree.

3.4 Middleware Evaluation

In this section we present a simulation-based evaluation of the PeerAppear framework. The evaluation is divided into two subsections, with the first demonstrating the performance of the system with respect to its addressing scheme and the second characterizing the performance of the system with respect to its search capability using both single and multi-level cell operations.

PeerAppear Simulation Engine (PASE).

To enable incremental development, parameter tuning and performance evaluation, the PeerAppear framework was implemented in a custom Python-based dis-

crete event simulation environment which we have named the PeerAppear Simulation Engine (PASE). PASE simulates and records all interactions between peers on the simulated framework. Peers and their local repositories can be randomly generated or built from a pre-existing dataset, enabling both large-scale global evaluation and small-scale application-driven evaluation. Each simulated client's local repository, peer inventory, routing table, and local system variables are maintained in a central database. Databases for various network sizes and tunable parameters can be constructed and saved, thereby facilitating evaluations to be executed multiple times for an array of parameter values using the same dataset.

Addressing Performance.

The following section documents our use of PASE to evaluate the performance of the framework's geospatial addressing scheme to preserve locality and the uniformity of data storage and traffic across the framework's address space.

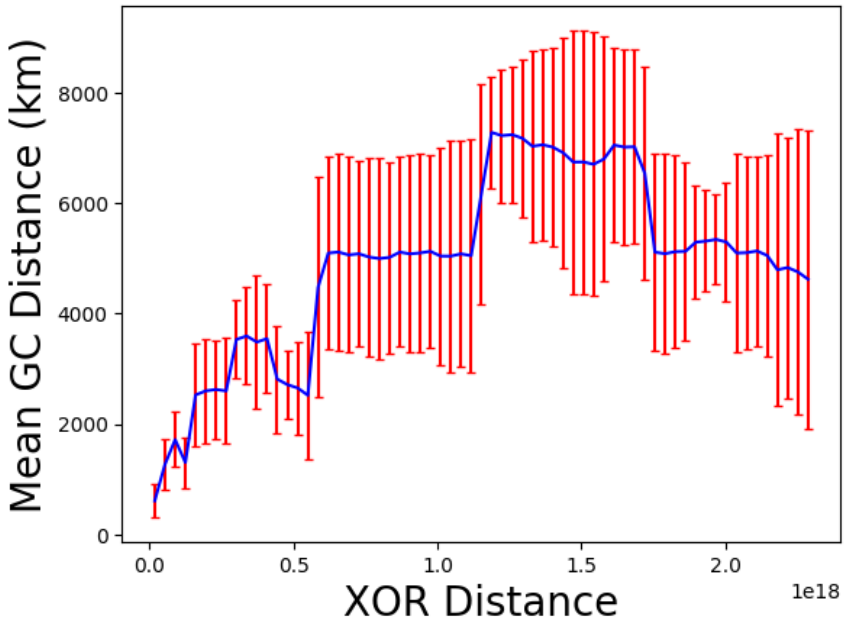
Locality Preservation.

Locality preservation means relative distances between locations in one address space are preserved when mapped into an alternate address space. The mapping of two-dimension spherical coordinates to one-dimension S2 addresses can be considered a type of reversible dimensionally-reducing hash function. A desirable property of hash functions which reduce dimensionality is preservation of locality. For a hash function to be locality preserving, it must preserve relative distances between input values and represent them in their corresponding output values. Because the S2 mapping is reversible, exact relative distances can always be determined by converting S2 addresses back to their corresponding spherical coordinates and calculating great-circle distance. We've already shown S2 can provide global availability and elastic

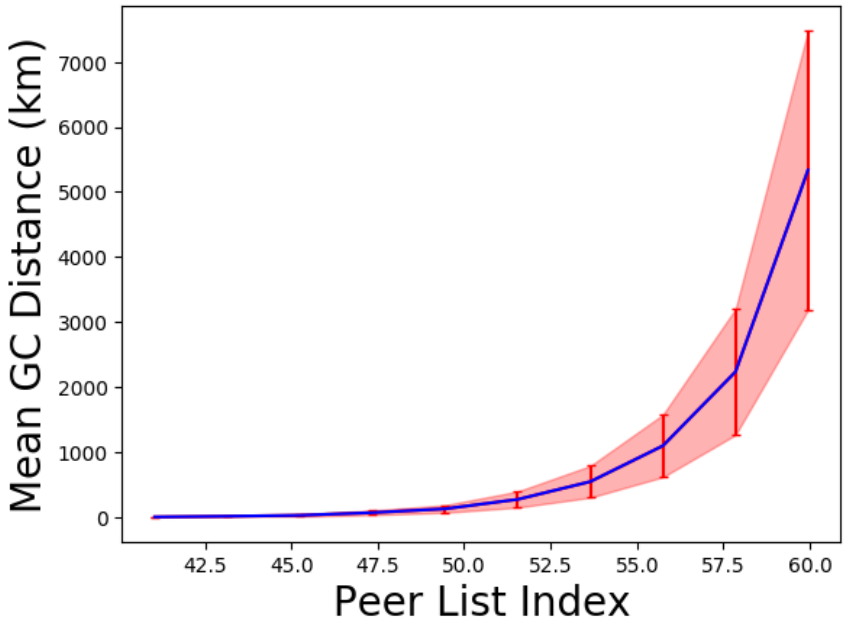
global addressing; however this process is computationally expensive.

To reduce the computational costs associated with calculating distances between cell addresses, PeerAppear establishes a network topology that adopts the XOR distance measure used by Kademlia. The XOR distance measure functions as a computationally low-cost heuristic for great-circle distance, enabling peer routing tables to be organized based on the magnitude of the XOR distance between a node's address and those of known peers. Because PeerAppear does not support proximity search, instead requiring search filters to be specified using S2 cell addresses, locality preservation within the address space is not explicitly required. Locality preservation does however benefit low-latency communications because nodes communicate primarily with peers whose addresses are nearby in the address space.

To evaluate the ability of the XOR heuristic to preserve locality within the PeerAppear address space and routing table, we generated 1,000,000 pairs of S2 addresses from the same level 0 parent cell using a random uniform distribution for analysis. The great-circle distances of the pairs' corresponding spherical coordinates were then plotted against the XOR distance of their S2 addresses. The resulting plot is shown in figure 12a. This plot demonstrates a general trending correlation between XOR and great-circle distance, however portions of the address space do not conform to this trend due the manner in which the Hilbert mapping sequences cell addresses. In figure 12b we can see that these inconsistencies are mitigated by the XOR-based binning employed by the PeerAppear routing table. This graph demonstrates that while the great-circle distances for individual cells from adjacent peer lists within the routing table may not exhibit consistent total ordering when arranged by increasing XOR distance, the mean great-circle distances do demonstrate a consistent monotonic increasing relationship for lists with XOR distances which are further from the source node.



(a) The relationship between XOR and great-circle distance.

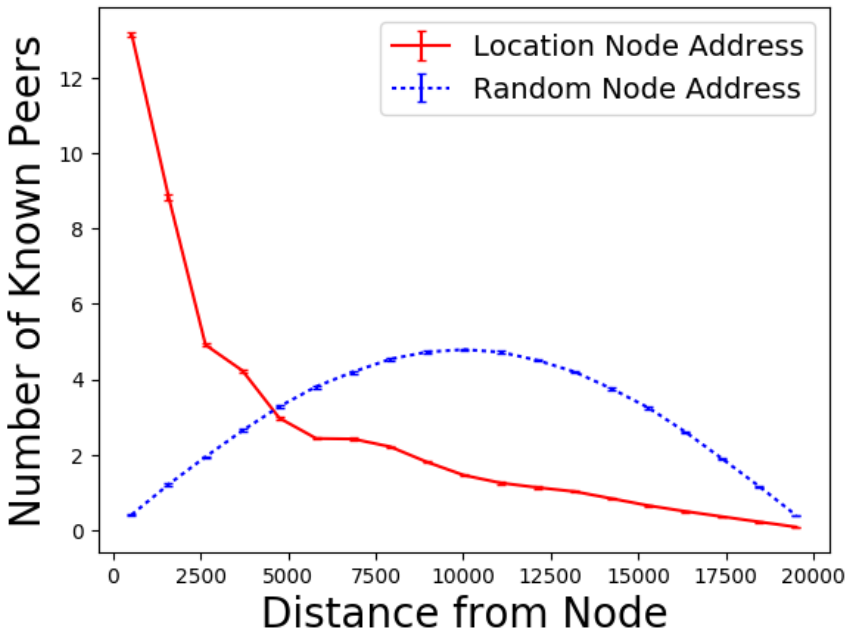


(b) Mean distance of peers by peer list index.

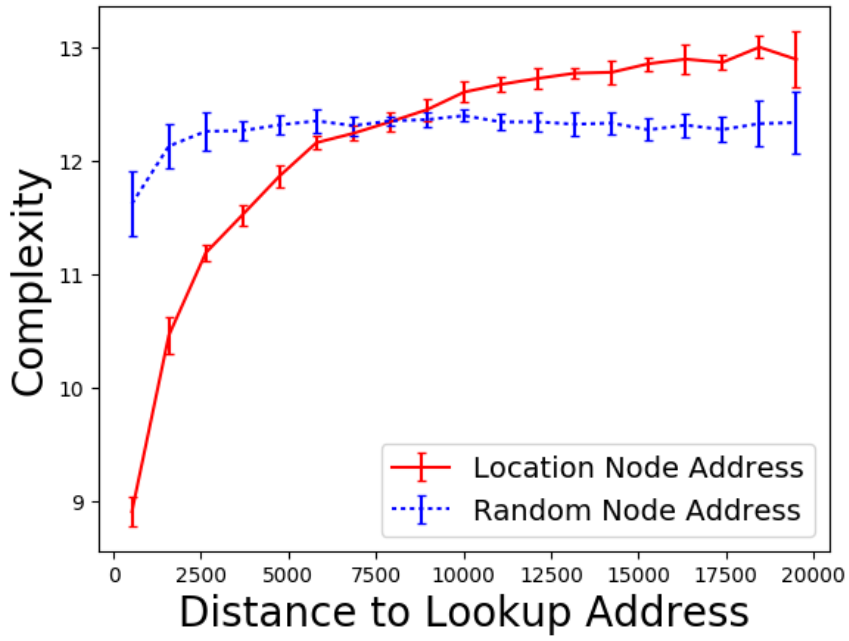
Figure 12. Representation of great-circle distance in the PeerAppear address space and routing table.

Next, we evaluated the effect of XOR's locality preservation on routing table composition and complexity of index lookup operations by effectively disabling locality preservation for comparative analysis. We accomplished this by building a network test set which "incorrectly" uses randomly generated 64-bit node addresses which do not correspond to the node's actual location. A second "correct" network test set was created using 64-bit S2-based node addresses corresponding to the locations of the simulated nodes. The PASE parameters for both network test sets are listed in Table 3.

The mean number of peers contained in the routing tables of both sets of test networks is plotted against the great-circle distance to the peers in Figure 13a. The graph demonstrates the network's ability to build routing lists which include fine-grained knowledge (greater numbers) of peers in local proximity while still maintaining access to more distant nodes. Numbers of randomly addressed peers correlate with the surface area of bands at specific range intervals, increasing up to one quarter of the earth's circumference before decreasing again. The mean lookup complexity for both sets of simulated networks is depicted in figure 13b. The network using correctly assigned S2-based node addresses demonstrated logarithmic lookup complexities which correlate well with great-circle distance to the lookup address. The network using randomly generated node addresses performed worse for node lookups at distances less than 7500 kilometers and only marginally better for further distances. The slight advantage held by random addressing for distant queries is likely due to uniform density of nodes in the routing table. While the network using location-based nod addresses has greater numbers of peers in close proximity, the randomly network with randomly assigned node addresses has uniform knowledge of nodes at all distances and can perform remote lookups for distant addresses by contacting fewer nodes.



(a) Histogram of peer list by distance.



(b) RPC complexity vs distance.

Figure 13. Spatial locality analysis for peer lists and RPC execution.

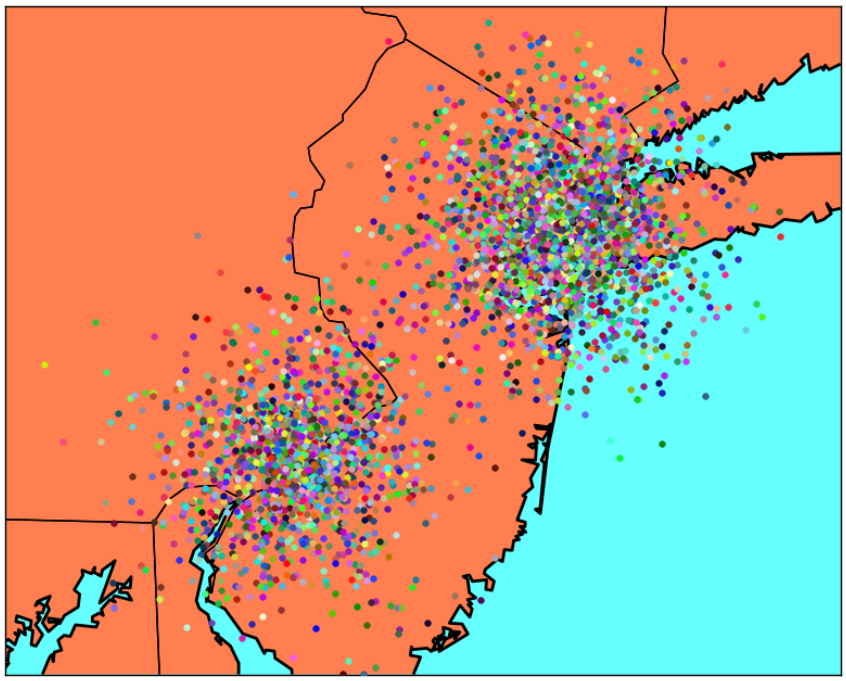
Table 3. PASE network parameters for locality analysis.

Parameter	Value
Environmental Parameters	
Peer Distribution	Global
Size of Area	510.1M km^2
Number of Peers	5,000
Number of Data Items	50,000
System Parameters	
Lookup Concurrency (α)	X
Replication (k)	X
Node Locations	S2 (Uniform Random)
Node Addressing	S2, Random
Data Locations	Gaussian σ=5km

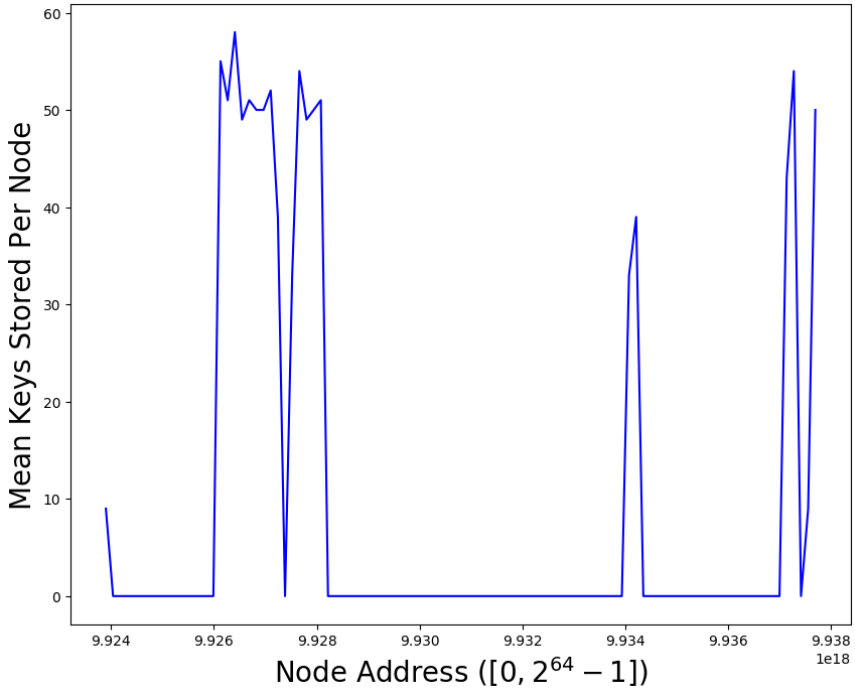
Uniformity of Key Storage.

The use of S2 addressing within the PeerAppear network leads to two types of non-uniformity of communications traffic and storage of published keys which impact publish and search performance. The first occurs because nodes within the network are likely to be geospatially distributed such that the density of nodes is proportional to population density. This leads to high density in urban environments, low density in most less populous and infrequently traveled areas, and extreme sparsity near the poles and water covered areas. An example of this type of non-uniform distribution can be seen in Figure 14. Figure 14a shows a mapping of nodes with locations assigned to New York City and Philadelphia based on a Gaussian mixture distribution and Figure 14b depicts the density of keys stored by nodes within the area's 1-dimension S2 address space. Because nodes assume addresses corresponding to their location at startup, this type of non-uniformity does not negatively impact the network because the number of peers responsible for indexing an area grows as the number of keys published in a given area grows (demonstrated in Section 3.4).

The second type of non-uniform key distribution occurs as a result of publishing and querying cell addresses which do not span the full resolution of the 64-bit address



(a) Map of nodes distributed geographically according to a Gaussian mixture distribution.



(b) Distribution of keys across the address space for a Gaussian mixture distribution.

Figure 14. Non-uniform distribution of key storage across address space caused by non-uniform geospatial distribution of nodes.

space, as is the case with larger (low level) S2 cells. The negative consequences of this type of non-uniform key distribution include increased network traffic and storage requirements levied on clients with addresses closest to the address space subdivisions corresponding to the addresses of larger (shorter prefix) S2 cells. This can be seen in Figure 15, where 50,000 keys were published to 5,000 nodes on the North American cube face at cell levels $C_L = \{5, 4, 3, 2, 1\}$. As the size of the published cell address prefix shrinks (lower cell level) the distribution of keys across PeerAppear's address space forms ever tighter clusters.

The non-uniform distribution of stored network keys has an additional negative consequence which can affect retrievability within the system under certain circumstances. Due to the XOR distance between overlapping cells of different levels, the potential exists for a query to be submitted using an S2 address which is sufficiently distant from a matching address such that the k closest nodes to the query address do not hold the matching cell address in their inventories. To mitigate this possibility for systems employing a heterogeneous mix of cell levels, we propose the use of an adaptive-k system. Under adaptive-k, all results which overlap (child or parent) a specified query address are returned. A minimum of k results, if available, are returned by the *FIND_NODE* and *FIND_VALUE* RPCs. The performance of mixed-level operations and the adaptive-k approach is further evaluated in Section 3.4.

Network Performance.

We conducted a series of simulations using PASE with parameter values specified in Table 4 to evaluate the performance of PeerAppear's DGT operations. These parameters were selected in order to explore the effect that their variability could be expected to have on the performance of non-simulated implementations of the

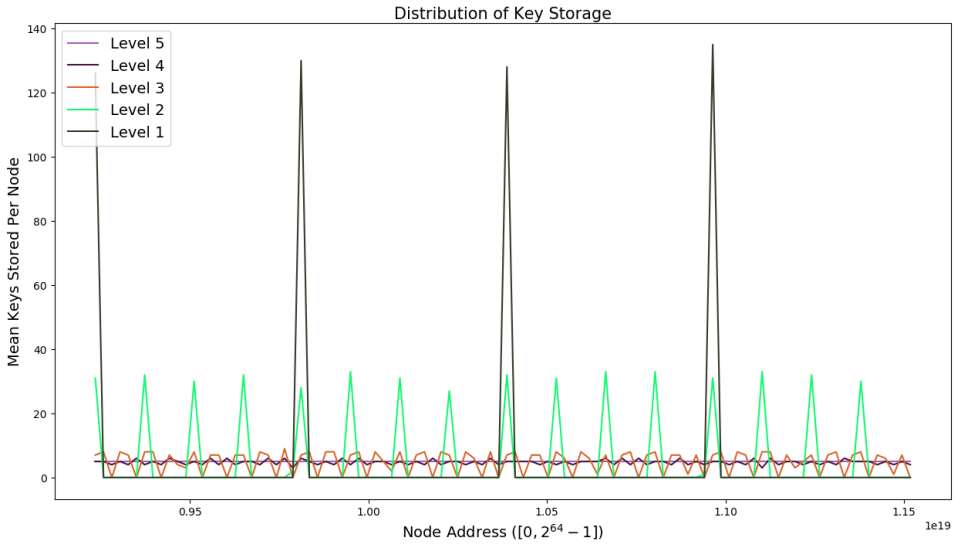


Figure 15. Non-uniform distribution of key storage across address space caused by cell level prefix size ($C_L = \{1, 2, 3, 4, 5\}$).

PeerAppear framework. For evaluations which vary a single network parameter, the remaining parameters are fixed using the underlined value.

The locations of simulated network nodes and the locations of the data they publish to the network are generated according to a Gaussian mixture distribution, resulting in node maps similar to the one depicted in Figure 13a. The nodes are distributed such that 65 percent are located near New York city (μ=(40.7128N, 74.00592W), σ=30km) and 35 percent are located near Philadelphia (μ=(39.9526N, 75.1652W), σ=30km). This distribution was selected to simulate expected real-world conditions with a mix of dense urban and sparse rural node locales. The locations for data owned and published by each node is also Gaussian distributed (μ=(Owner Lat-Lon), σ=5km). All plots depict mean results and 95 percent confidence intervals from 5 trials completed using the same network parameters. Nodes for each trial are loaded and bootstrapped into the network sequentially, with contact information for up to 5 previously connected nodes being provided to new nodes by a simulated discovery service.

Network performance evaluations are divided into two subsections: 3.4) Single

Table 4. Environmental and system parameters used to execute experiments within PASE.

Parameter	Value
Environmental Parameters	
Peer Distribution	NYC and Philadelphia
Size of Area	Approximately $62,500km^2$
Number of Peers (n)	50 - 5,000 - 20,000
Number of Data Items	$n * 10$
System Parameters	
Lookup Concurrency (α)	1 - 3 - 8
Replication (k)	1 - 7 - 8
Node Locations	Gaussian σ=30km
Node Addressing	S2
Data Locations	Gaussian σ=5km

Cell-Level Operations and 3.4) Mixed Cell-Level Operations. The first evaluates the performance of the network for k and α parameter variation in terms of recall and search complexity for various numbers of nodes when keys are published and queried using a single cell level. This evaluation tests the network's ability to perform exact-match search for S2 keys. The second evaluates the performance of the network for various search cell levels in terms of recall and search complexity for various numbers of nodes and k values, including adaptive-k operations. This evaluation tests the network's ability to perform search for network keys using filters comprised of multiple cell levels.

Single Cell-Level Operations.

Single cell-level operations test the network's ability to find publishers of keys where the query address is exactly equal to the published key. This evaluation was accomplished using publish and search cell level $C_L = 16$. Representing an area approximately equal to 20,000 square meters or 1 city block, we consider level 16 cells to be a satisfactory compromise for granularity of data representation. The search recall and complexity of networks with numbers of peers between 50 and 20,000 for

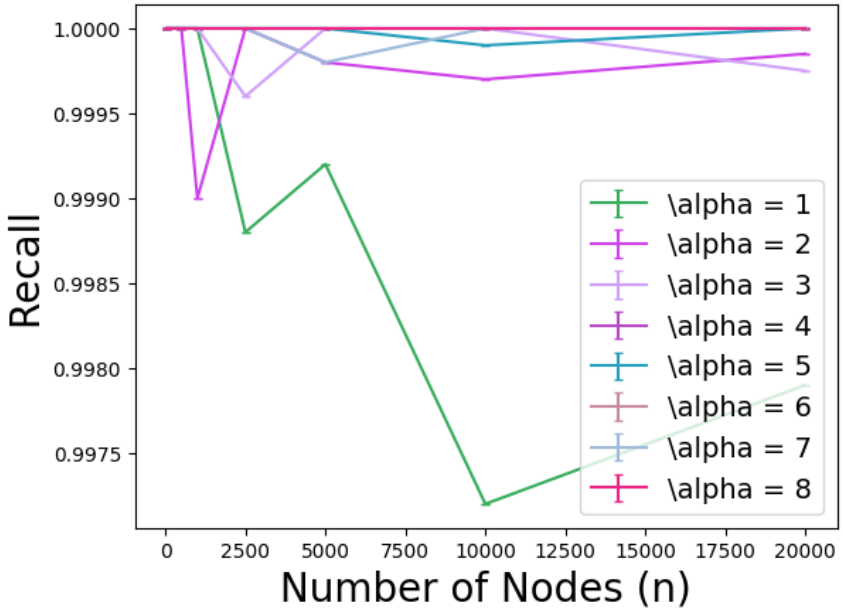
α values between 1 and 8 is shown in Figure 16. Recall results, depicted in Figure 16a show that α values as low as 1 still return near full recall, however values above 1 produced more consistent results. The complexity analysis for the examined range of α values, shown in Figure 16b, showed that increases in α, up to the value of k, created significantly greater search complexities.

Using the same level publish and search cells ($C_L = 16$), the experiment was repeated with a fixed k value of 3 and α values between 1 and 8. The results are depicted in Figure 17. Figure 17a shows that full recall is achieved for all network sizes with k values of 3 and greater. Figure 17b shows marginal increases in complexity as k and the size of the network grow beyond 3. For all values of k network complexities exhibit logarithmic complexity growth as network size increases.

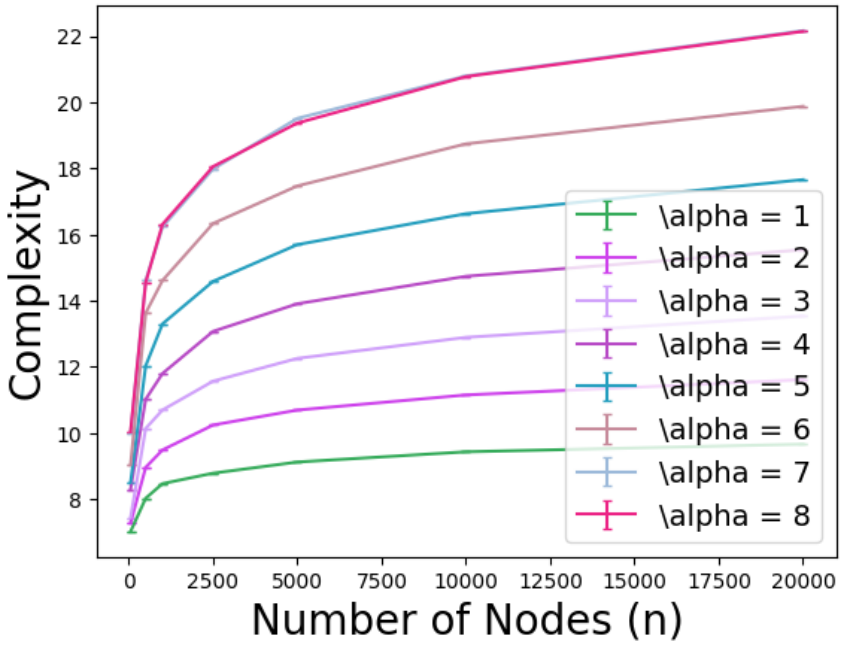
Mixed Cell-Level Operations.

Mixed cell-level operations test the network's ability to find cells which share corresponding areas (share a full prefix) when the publish and search cell levels are not the same. For this evaluation, keys are published at $C_L = 16$ and queries are submitted across the range of $C_L = [5, 19]$. The search recall and complexity of networks with numbers of peers between $n = 50$ and $n = 20,000$ for the specified search levels is shown in Figure 18. The results show search recall for a specified search cell level and network size in Figure 18a and corresponding complexity in terms of number of nodes searched in Figure 18b. Recall results demonstrate marginally higher recall for smaller network sizes when search is performed using lower level cells, however recall for all network sizes converges nearly 1 for cell levels at and beyond level 14.

Using the same cell levels, the experiment was repeated for a fixed network size of $n = 5,000$ and k values ranging between 1 and 7. In addition, the adaptive-k method

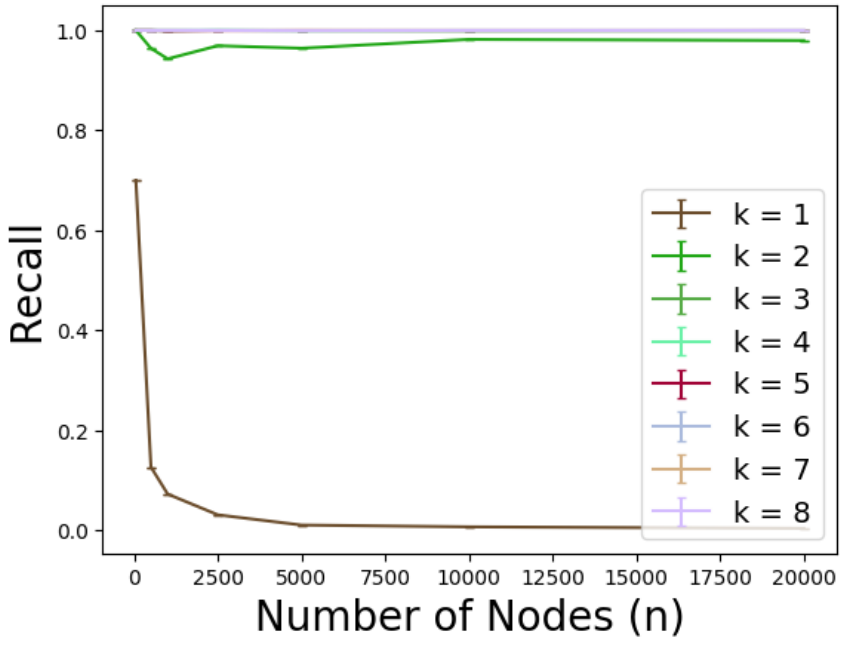


(a) Search Recall

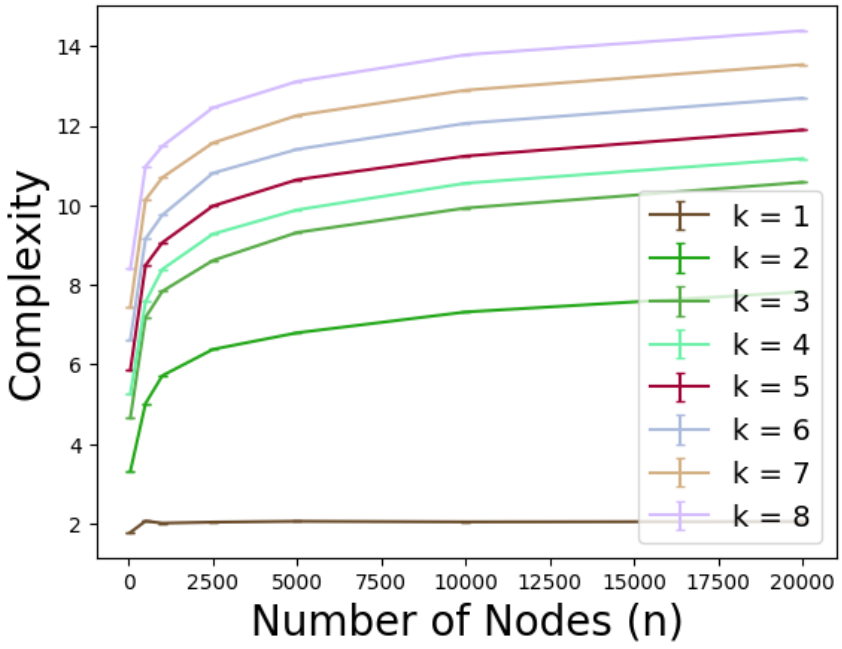


(b) Search Complexity

Figure 16. Effect of numbers of peers on recall and complexity for various alpha values.

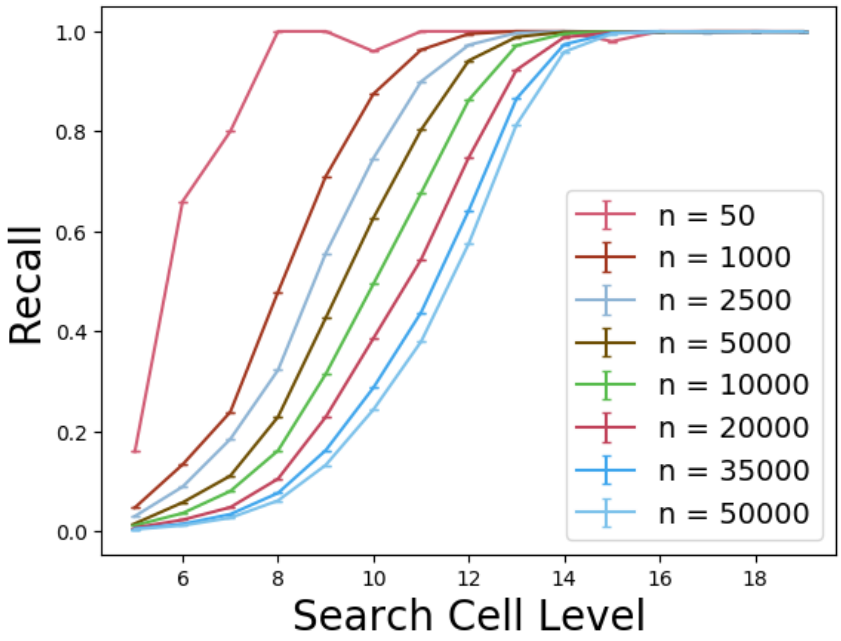


(a) Search Recall

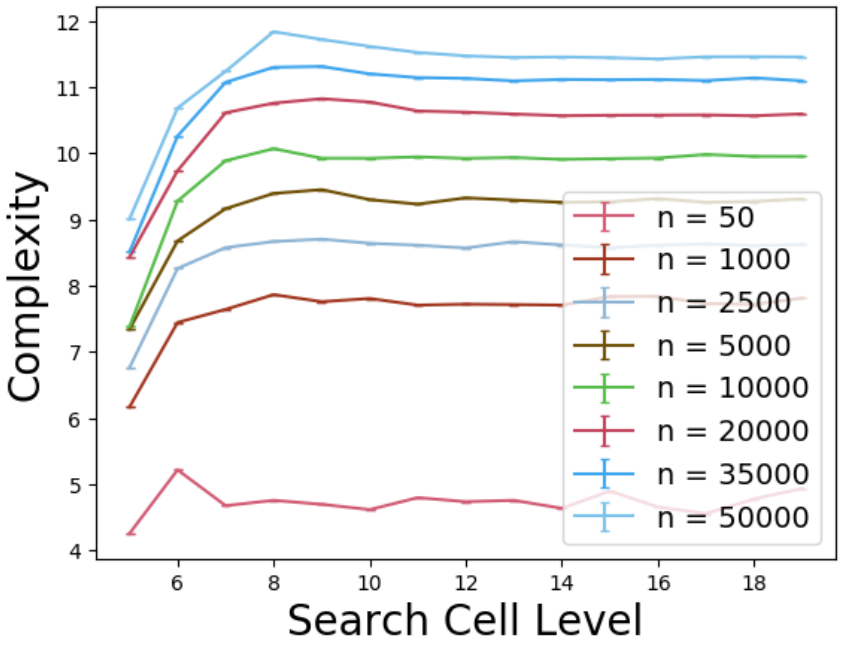


(b) Search Complexity

Figure 17. Effect of numbers of peers on recall and complexity for various k values.



(a) Search Recall



(b) Search Complexity

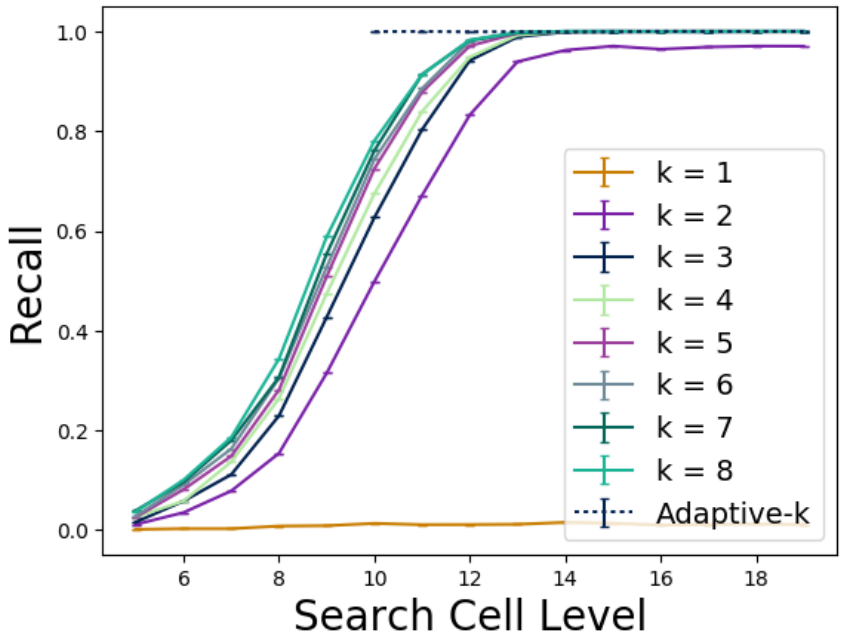
Figure 18. Evaluation of search recall and complexity at different search cell levels for various network sizes.

described in Section 3.4 was also employed using a base-k value of 3. The results of this experiment are shown in Figure 19. Figure 19a demonstrates near-perfect recall for search cell levels at or above $C_L = 13$ using k values of 3 and higher. Below $C_L = 13$ recall rapidly drops with higher-k value networks consistently faring marginally better. Figure 19b shows that complexity for fixed-k operations is a function of k value with little variation across the tested range of search levels. While the adaptive-k search evaluation demonstrated perfect recall, complexity proved nearly intractable as the size of the search space grew to encompass the entire dataset.

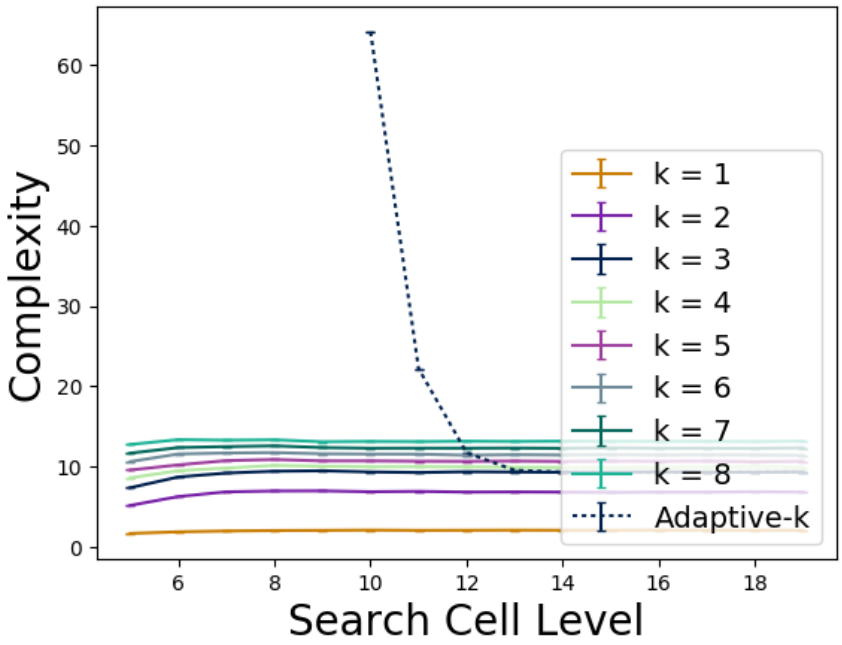
3.5 Conclusions

This chapter describes the core elements of PeerAppear, a middleware framework designed to enable the construction and maintenance of decentralized geospatial data repositories through a peer-to-peer architecture. Use cases for PeerAppear were presented and accompanied by a system integration design to meet use case requirements which included the architecture for the PeerAppear framework's core geospatial addressing and distributed indexing functionality based on the integration of the S2 geometry library and Kademlia peer-to-peer overlay topology. The addressing scheme is demonstrated to exhibit global coverage and elasticity though its hierarchical cell-based partitioning of the earth. The integration of the S2 addressing scheme with the Kademlia P2P overlay also enable global availability of keys published to the distributed index and support for dynamic environments through a topology and associated protocol enabling efficient publish and expiry of user contributions.

Through a series of comprehensive large-scale simulated network evaluations, the suitability of S2 addressing and the overall effectiveness of the framework was established. Testing of the integration of S2 addressing within the distributed index demonstrated that S2 addressing provides sufficient locality preservation in order to



(a) Search Recall



(b) Search Complexity

Figure 19. Evaluation of search recall and complexity at different search cell levels for various k values.

support efficient location-based queries. This analysis also demonstrated that while an undesirable non-uniform distribution of published keys can occur for larger key cell sizes, near-uniform distributions can be achieved through the use of smaller publish cell sizes. The evaluation also analyzed recall and complexity for variation of tunable parameters and network sizes from 50 to 20,000 nodes. Overall, the evaluation demonstrates that the overlay topology pairs well with the S2 addressing scheme, enabling efficient logarithmic time operations and full retrievability of published data for symmetric cell-level operations and high levels of retrievability for non-symmetric cell level operations.

IV. Computational Support for Practical Visual Applications Using Collaborative World Models

In order to meet the needs of our motivational vision-based applications, the framework presented in Chapter III must be extended to support practical image-based and location-based search. In this chapter the framework's architecture for collaborative image discovery is presented. The architecture includes extensible decentralized image repositories, methods for enabling visual search, and algorithms supporting the indexing and ranking of images. The chapter also includes methods for constructing geospatial filters which enable practical location-based search within PeerAppear, including representation of radius-based search filters and algorithms for novel viewshed-based search.

4.1 Related Works

Literature examined in this section covers work related to the computational support methods used for the PeerAppear framework. The works represented describe methods for extracting information from images which broadly fall into the subject area categories of *image classification*, *image annotation*, and *visual search*.

As humans we rely on a natural ability to make semantic associations from visual information[34]. These associations enable humans to identify objects and ideas in images and represent them in a compact form using natural language. The association is bi-directional, allowing objects and ideas to be located visually based on a given semantic concept, thereby enabling a natural visual search capability. Computerized efforts have generally suffered from a semantic gap[8] which prevents the autonomous formation of visual-semantic associations. Recent efforts have attempted to minimize the semantic gap using various forms of deep learning[60] to extract high-level concepts from visual data.

Computerized image search efforts have traditionally been broadly classified into the categories of meta search and content-based image retrieval (CBIR)[60, 92, 120]. Meta search involves identifying images based on associated meta-level tags and annotations which are applied manually or automatically and often include information such as textual item description, time, or location. CBIR methods enable a search and retrieval capability based on the visual content of the image, recorded as a grid of pixel colors or intensities. Recent efforts to automatically annotate images with descriptions of content using computer vision and deep learning have muddled the line between meta search and CBIR.

One of the earlier and most noted papers in the field of CBIR, written by Faloutsos et al, presented Query By Image Content (QBIC)[31]. The system was one of the first to make use of multi-dimension indexing to enable search and retrieval capabilities based on multiple types of image content, such as color, texture, and shape. The work included several novel features. The first was a new method for full cross-term quadratic mapping which enabled Euclidean distance measures to be applied to non-Euclidean feature spaces. The second was a dimensionality reduction method based on the Karhunen Loeve transform which maintained search accuracy while lowering space and computational complexity. Experimental evaluations demonstrated that search using multiple feature spaces could produce highly accurate search results while still being computationally efficient.

The task of automatically applying image annotations is often accomplished using a form of artificial intelligence known as the support vector machine (SVM), which is based on methods pioneered by Fisher[38]. The SVM enables clustering of highly dimensional datasets using labeled training data or in a self-organizing fashion by specifying a pre-defined number of classes or prototypical vectors. These clusters can then be assigned class labels, enabling future observations to be automatically

classified. In one of the earliest of such efforts, Cusano et al used the SVM to enable automatic image annotation for multiple classes of visual content[29]. These classes included sky, skin, vegetation, snow, water, ground, and buildings. The SVM input consisted of color histograms for a fixed number of partially overlapping image subdivisions for each image. Training the SVM was accomplished using a set of manually labeled training images. The results demonstrated that images which contained well-lit content could be partitioned and annotated using the method with fairly accurate results.

Further automatic image annotation efforts conducted by Jeon et al[51] and Feng et al[33] demonstrated that more complex visual-semantic relationships could be learned using statistical relevance models. Both papers present methods using these models which are trained using sets of manually labeled training images. Features for both training and annotation were extracted from rectangular image segments and include metrics based on color and texture. Both papers demonstrated a robust automatic image annotation capability for visual-semantic relationships that were well represented in their respective model's training data.

Building on these early works, many efforts that followed[112, 17, 65, 110, 94] adopted various adaptations of multi-class learning. One of the earliest widely used datasets for benchmarking purposes was the Corel dataset[105]. Evaluations of later work adopted the ImageNet competition dataset for comparison of results[93]. Results in terms of assignment accuracy for feature-based multi-class learning on common benchmark datasets began to plateau as the limits of the methods were fully explored. Alternative methods, such as Li et al's leveraging of CBIR web searches for automatic annotation[59], also encountered difficulty due to the similar constraints faced by image similarity and annotation techniques.

While the use of colar and grayscale intensity histograms often worked well for im-

age classification, the signatures are generally too generic when searching for images of specific objects or places. Many visual search systems achieve better discrimination by incorporating visual point features into the image signature. Visual point features are salient keypoints found in images which are derived using algorithms that identify highly descriptive visual elements of the image and represent them in numerical formats[96]. Two of the most popular feature types commonly used in visual odometry are the Scale Invariant Feature Transform (SIFT)[63] and the Speeded Up Robust Feature (SURF)[11]. Both feature types produce descriptors represented by large vectors (up to 128-dimension) that are invariant to changes in scale and rotation and are highly repeatable. A single image with significant visual variation can produce many SIFT or SURF features while an image with little visual variation may produce few or none. Once features have been extracted from a set of images with related subject matter, they must undergo a matching process to find correlated feature pairs.

Even though good feature transforms are highly repeatable, they rarely generate corresponding features that are a perfect match. Because SIFT feature descriptors are generally represented by 128-dimension vectors they can be be thought of as points located in 128-dimension space. Repeatability ensures that successive occurrences of the feature in alternate images posses similar feature descriptors, thereby locating their points in close proximity[97]. Naive matching methods simply calculate the Euclidean distance between feature descriptors to find potential candidates for matching[52, 14, 97]. An alternative method for matching features was proposed by Lowe using a modified k-d tree algorithm which greatly reduces the number of comparisons required to compute matches[62]. Further research has demonstrated feature matching capabilities in excess of 100 frames-per-second using programmable graphics hardware[25]. An example feature extraction and matching between two

different photographs of the same object from different positions using OpenCV[50] is shown in Figure 20.

Because invalid matches are likely to have occured in the feature matching process, a validation step must occur before attempting to estimate motion between images. The most popular method of eliminating these outliers is known as Random Sample Consensus (RANSAC)[36]. RANSAC is an iterative process which randomly selects data points with which to build a model, then uses that model to evaluate the fit of the remaining data points. The largest set of data points which fit the model are selected as valid matches. Once matched features have been validated, the positions of corresponding features in a pair of images can be used to estimate the motion that occurred between the image pairs[97]. This motion estimate is represented by the essential matrix which describes both rotation and translation. For systems with 6 degrees of freedom, Nister's 5-point algorithm is commonly used to estimate the essential matrix given the locations of a minimum of 5 pairs of matched features[75].

Though important for localizing through homography estimation, feature descriptors are too fine grained of a representation to be useful in large scale search. Thus, visual features have been used as coarse grained terms for visual search and classification generalized within the Bag of Words (BoW) model approach. The BoW model approach is most often used to represent text documents as histograms indicating the

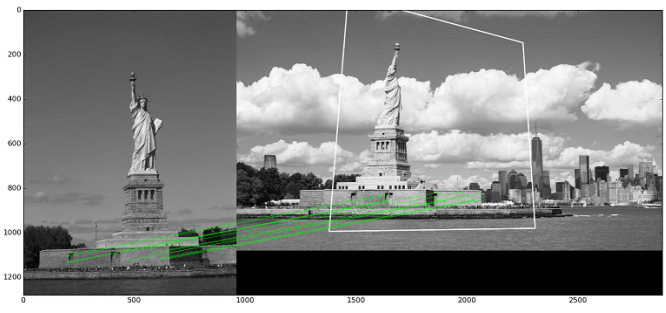


Figure 20. Matched visual features and associated perspective transform between images of the Statue of Liberty.

number of occurrences of each word found in the document[95]. While this representation does not preserve the order of the words, it does enable an entire document to be represented in a compact manner, thereby supporting efficient indexing and search of documents. The same process as adapted to images is known as Bag of Visual Words (BoVW)[119]. This method enables images to be represented by histograms indicating the number of occurrences of visual terms found in an image. While BoW relies on a pre-existing dictionary, as is native to written language, BoVW requires that a dictionary be constructed to represent the visual features within an image.

Before the BoW approach can be applied to images in a dataset, this dictionary or codebook of visual terms must be established. This is necessary because visual feature descriptors are generally quite large. The most common codebook generation approach involves binning the feature descriptor space into Voronoi cells using vector quantization[77]. This is often accomplished in the literature using k-means[27], k-medians[40], or self-organizing map[9] clustering techniques. These clusters form a mapping from the large visual feature descriptor space to a much smaller codeword space. While this process does diminish the differentiation power of individual descriptors, the combination of many codewords from a single image form a signature that is still sufficient for robust image matching[61].

One of the most common applications of the BoVW methodology is topological image categorization and image recognition. These techniques enable objects in images to be recognized due to the presence of their unique feature signatures. Both supervised[27, 106, 30] and unsupervised[58, 86] image categorization has been demonstrated using this approach. While topological classification of images is not the specific goal of this work, the same techniques are applicable when performing visual search on large image datasets as is required for visual location recognition.

While the quantity of image search and annotation research has seen significant

growth in recent years, major breakthroughs and paradigm shifts had been somewhat elusive. Evaluations of various classification models and techniques based on image color, texture, and local features demonstrated capabilities that were sufficient for most tasks, but still prone to frequent errors. The latest techniques[54, 109, 76, 55], building on advances in research from the field of artificial neural networks, make use of convolutional neural networks (CNN) to interpret image content in a manner that is biologically inspired. A typical CNN architecture is shown in Figure 21. Wang et al recently applied CNNs to the task of determining fine-grained image similarity[111]. The method requires that training be accomplished using a triplet set of images from a repository of training images. The triplet includes a reference image, an image that is similar to it, and an image that is not similar. An evaluation conducted using the ImageNet ILSVRC-2012 dataset demonstrated a retrieval accuracy nearly 20 percent greater than methods based on local image features.

Once trained, neural networks offer significant advantages in terms of computational complexity when compared to traditional search and classification methods. While training often requires significant time and computation, recent hardware development efforts to create specialized parallel systems[115, 100, 108] to speed this training have demonstrated results far surpassing those attainable with traditional processors. After training is completed, the neural network achieves constant time and size complexity making it ideal for use in low-power portable systems.

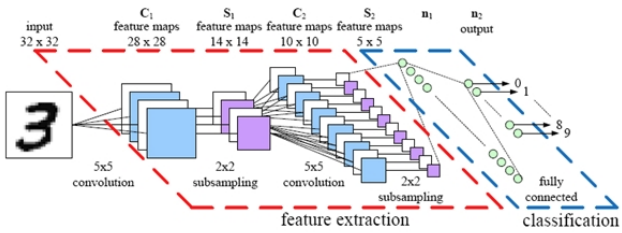


Figure 21. A convolutional neural network architecture for optical character recognition[114].

While the potential for CNN-derived features to be used for image search seems great, they have currently only been successfully demonstrated for use in image classification and pair-wise similarity analysis. Even though the BoVW method has been surpassed by CNNs for classification, it is currently the only search paradigm which conforms to the PeerAppear framework's decentralized architecture. Because BoVW vectors are based on point feature descriptors, images matched through BoVW-based search also offer better chances to be validated using RANSAC and the epipolar constraint. For these reasons, image search methods for the PeerAppear framework are based on the BoVW paradigm.

4.2 Visual Localization Activities

In this work we have identified visual localization as a primary motivational activity for the PeerAppear framework. In this section we identify actors for the activity and describe their roles and procedures used to enable the motivational activity.

Roles.

While actors participating in visual localization activities are expected to be symmetric, the roles they play during individual interactions are not. In this section we describe the roles of the **searcher** and the **content owner**.

Searcher.

The searcher is a node trying to localize an image based on an approximate location where it was captured as well as the uniqueness of the image's visual content.

Content Owner.

The content owner is a node which has published a key pair into the index indicating that the owner is in possession of localized content that lies within the boundaries of the S2 cell described by the key's address.

Procedures.

Procedures used for this activity include the publishing of localized content summaries by content owners and the localization of visual content by searchers.

Publish Visual Content Repository.

The locations of images within the local repository of a content owner must be published to the distributed index to enable searchers to discover the content. This process is depicted in Figure 6. First, a location-based summary is constructed using S2 cells to represent the locations of the images in the repository. This summary does not have to consist of contiguous cells. For example, if a node had visual content from different continents, the cells describing the locations of the content would reflect their geographic disparity. For each S2 cell address comprising the content summary, a key containing the S2 cell's address would be published using the procedure described in Section 3.2.

Localize Visual Content.

Localizing an image requires the BoVW signature for an image as well as an approximate location where it was captured. The approximate location must be specified using one or more S2 cell addresses. A depiction of the lookup process is shown in Figure 7. The image's BoVW signature must also be constructed using a standard visual dictionary (or codebook) which ensures that all visual features are

consistently summarized for all participants.

The localization process begins with a search for owners of localized images within area described by the S2 cells representing the image's approximate capture location. Using the search procedure, described in Section 3.2, each S2 cell in the representation is searched and results are aggregated. The resulting list of content owners can then be used for direct peer-based communication in order to satisfy the visual aspect of the search.

For each content owner identified through location-based search, the BoVW signature of the query image is transmitted through direct peer communication. The content owner determines if it has matching content in its repository (using the procedures from Section 4.3) and responds with the location of the best match, thereby offering opportunities to infer content location by the searcher.

4.3 An Architecture for Collaborative Image Discovery

The architecture of PeerAppear is inspired by Bittorrent[20], a file sharing overlay network which communicates using the internet protocol (IP) on a global scale. Bittorrent's decentralized index, the Mainline DHT, is a Kademlia variant which demonstrates excellent scalability, robustness, and efficiency. Bittorrent nodes maintain a local repository of shared files. For each shared file, the Bittorrent client publishes a key pair to the Mainline DHT which contains information about the content and owner in order to facilitate direct communication and file transfer with other peers in the network. The addressing within the DHT is based on the 160-bit SHA-1 hash of the files being shared. In order to find which nodes in the network have a certain file, the SHA-1 hash of the file must be known. This requires the maintenance of a separate directory or lookup service which enables the matching of file names to hashes.

PeerAppear's architecture seeks to capitalize on the benefits demonstrated through Bittorrent's approach. PeerAppear therefore adopts a similar architecture which includes the maintenance of user-collected images and sensed geospatial data in local repositories. PeerAppear nodes publish key pairs to a distributed index which include information to enable image discovery and to pass information enabling direct peer communication. To support localized image search, PeerAppear uses a 64-bit geospatial addressing scheme based on the S2 library which enables localized search without the need for additional directories or lookup services. The architecture therefore enables collaborative image discovery by allowing clients to both publish and discover image repository content based on location in a globally accessible and addressable index.

To motivate the usefulness of collaborative image discovery we present the following scenario, depicted in Figure 22. Bob and Andy are each wearing mobile GPS and camera-equipped navigation aids which participate in the PeerAppear network. The navigation aids capture images at prescribed intervals and annotate them with other available information, such as the current GPS-derived location. When GPS is not available, an application built on top of the PeerAppear framework processes the images using a monocular visual odometry (VO) algorithm to predict position, which is then added as an additional image annotation. Bob enters the local shopping mall where he makes a large loop while trying to find his wife the perfect birthday present. Shopping malls being what they are, Bob does not find the perfect gift and ends up completing his circuit and leaving. While inside the mall, Bob's personal navigator captured images which were annotated using position estimates from the VO application. Upon completing his circuit, the application was alerted to a visual location match through the PeerAppear framework, thereby allowing loop closure and bundle adjustment for the VO-derived position annotations. Later that evening, Andy en-

tered the mall to buy the latest Tom Clancy game at his favorite game shop. While walking through the mall the VO application on his personal navigator continuously searched for potential loop closure and location recognition opportunities. As he intersected the path previously taken by Bob the search succeeded when a query submitted through PeerAppear was answered by Bob's navigator. This enabled Andy's personal navigator to perform collaborative loop closure and update its own image annotations based on the query results from Bob.

Extensible Decentralized Geospatial Image Repositories.

The structure of the PeerAppear framework is designed to enable clients to function both independently and collaboratively when possible. Participants maintain a file-based collection of imagery captured through the PeerAppear client, with each image file corresponding to an entry in a locally maintained database which includes all image annotations, as derived through extensible components within the framework. To enable location sensitive queries, the framework logically addresses and indexes the data stored in users' collections according to the location of the data. All images must therefore include an annotation which localizes the image.

A chart depicting the primary components of the PeerAppear framework is shown in Figure 23. When joining the network, a client (1) begins by contacting a centralized peer discovery service (2) or by making contact with other network participants discovered during prior sessions. Population of the client's peer list is then enabled through an iterative lookup process (3). Storage of images and sensor data (4) collected by the client is facilitated locally. Image annotations provided by plugins (5) are applied within the extensible layer of a local database used to manage the client's local image repository.

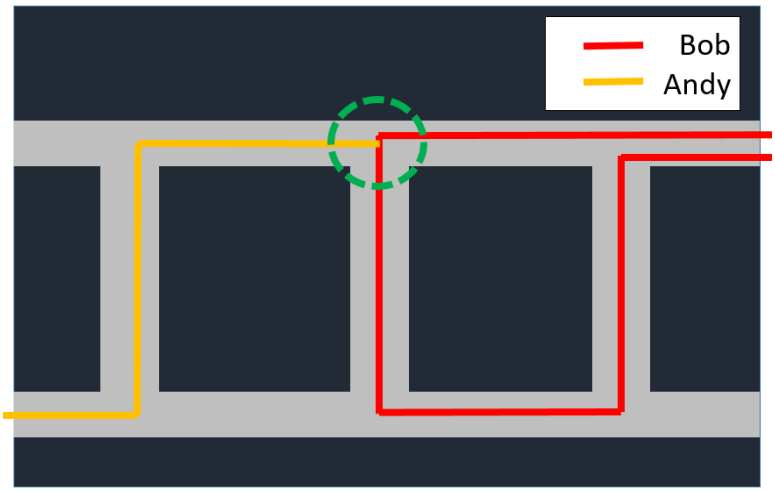


Figure 22. A collaborative loop closure scenario for monocular visual odometry.

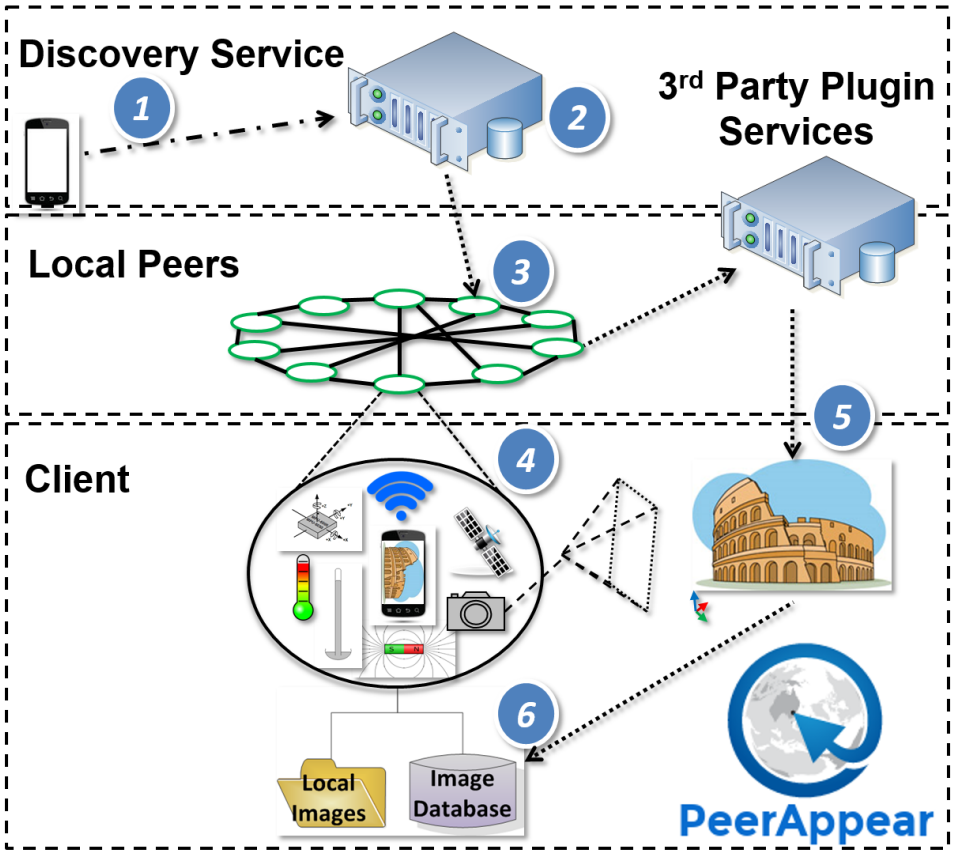


Figure 23. The local and networked components of clients participating in the PeerAppear network.

PeerAppear uses addresses derived through Google's S2 geometry library as a principle means of applying localized annotations for images. The S2 library provides for global-scale addressing through the mapping of the 6 sides of a cube to the earth, as shown in Figure 24. Each face of the cube is subdivided using a quad tree and regions are mapped using a space-filling Hilbert Curve. The smallest addressable units, level 30 leaf cells, each represent an area approximately equal to $1cm^2$. All addresses in the S2 library use 64-bit representations, with the hierarchical structure of the quad trees represented in bit order in the address. This enables smaller cells to be grouped and represented by a larger cell addressed using the largest common prefix of all sub-cells. The S2 address encoding enables PeerAppear to represent the locations of users' image collections with varying levels of granularity in order to provide sufficient fidelity to target search queries while still maintaining an abbreviated representation suitable for peer-to-peer indexing.

An index mapping the locations of annotated image collections in the PeerAppear network to individual users must be maintained to enable clients to identify which peers have images that may be relevant for a location sensitive search. Due to the dynamic nature of the data available on the network at a given time, the aggregation and maintenance of a single index would be challenging and costly. PeerAppear therefore adopts the Kademlia distributed hash table (DHT)[68] approach for maintaining a distributed index of clients and representations of the locations of the image collections they store.

In the same manner that Bittorrent uses the 160-bit address space to map portions of the distributed file index to users, PeerAppear uses the 64-bit cell addresses provided by the S2 library to map the indexing of images to clients participating in the network. Because SHA-1 hashes for files on the Bittorrent network are uniformly distributed, the selection of a uniformly random 160-bit client identifier is appropri-

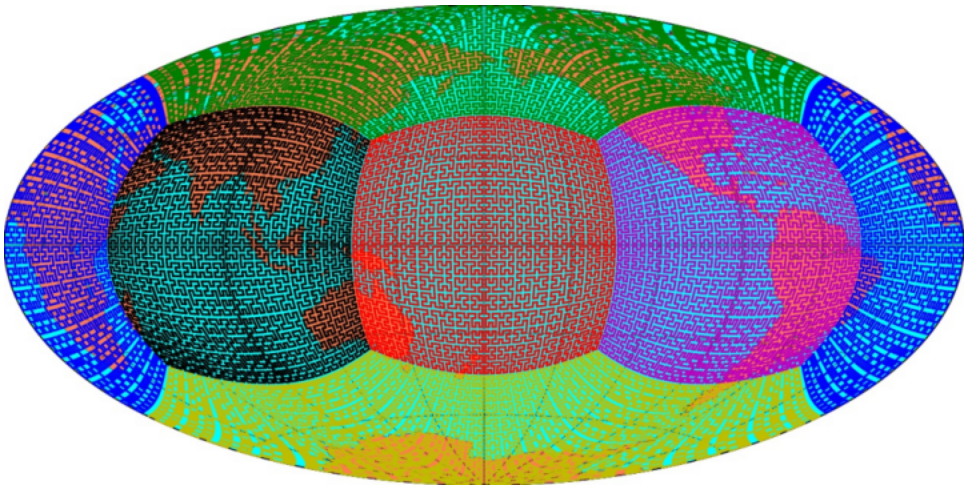


Figure 24. A mapping of the globe using Google's S2 Spherical Geometry library with level 6 cells.

ate. Such is not the case for PeerAppear however, as the S2 addresses used for images are not randomly distributed. PeerAppear clients can therefore use a more appropriate 64-bit identifier, such as their location at time of startup or the average location of the data represented in their own local repository. This has the added benefit of load balancing the DHT because areas with the greatest numbers of users would be expected to also collectively have the largest image repositories.

To enable collaborative visual information discovery, each client publishes a summary of data in its local repository to a distributed geographic table (DGT), inspired by the Kademlia[68] distributed hash table (DHT). In the same way that a DHT enables participants in the Bittorrent network to find files shared by peers, the DGT enables PeerAppear users to find visual information at a given location which is present in the local collections of other network participants. Once a peer is identified as potentially having desired information, based upon the self-published geospatial summary of their data, search queries targeting more fine-grained location and other types of annotations can be sent directly to the owner of the data. Queries are executed by the owner and results are relayed back to the requester. The structure of the PeerAppear framework, shown in figure 3, is designed to enable clients to func-

tion both independently and collaboratively when possible. Participants maintain a file-based collection of imagery captured through the PeerAppear client. Each image file corresponds to an entry in a locally maintained database which includes all image annotations, as derived through extensible components within the framework. To enable location sensitive queries, the framework addresses and indexes the data stored in users' collections according to the location of the data. All images must therefore include an annotation which localizes the image.

When a PeerAppear client launches it begins a peer discovery process to establish a mapping to different parts of the network. This process is facilitated by connecting with a previously known peer or through the use of a centrally hosted peer discovery service. Network lookups can then be performed to find additional users in other parts of the network's address space. Once the list of peers has been established, the process of submitting a search query can begin by first performing a recursive lookup to find the clients with identifiers closest to the location specified in the query. A final lookup is then performed to identify peers that were identified as having images with prefixes that match the search query. The final search query is then formulated and submitted using parameters relevant to a specific type of image annotation, as defined through PeerAppear's extensible multi-modal image annotations.

The PeerAppear framework is designed with extensibility in mind to support many different applications which rely on the extraction of visual information from imagery. Applications such as visual localization and structure from motion use methods which index visual content to enable fast and efficient search for visual similarity. Content-based image retrieval (CBIR) methods generally tag images based on recognizable content to enable object search. Common to these and most other methods is the extract, process, and annotate flow of execution. Visual data is extracted from an image, processed into useful information, and annotated back onto the image. Be-

cause each application relies on different methods for annotating images, and because the methods themselves are constantly in flux as new techniques are discovered, the adoption of an extensible framework makes the most sense for the construction of large-scale collaborative image databases.

Extensibility in the framework is provided through the PeerAppear application program interface (API) which enables third party software and PeerAppear plug-ins to interact with the framework in order to add new images and apply annotations to images within the local collection and to submit queries to the global PeerAppear network. While PeerAppear maintains responsibility for the storage and handling of all annotated information, the structure of the annotations must be specified using PeerAppear's meta-level XML annotation schema. An example annotation schema for the application of WiFi BSSIDs to images is shown in Figure 26. The schema informs PeerAppear of the existence of the *wifi_tag_svc* plugin and specifies the addition of the *BSSID* text vector field for images stored by PeerAppear. It also registers an event listener for PeerAppear's *OnImageAdd* event, thereby allowing it to receive notification when new images are available for annotation. For this example, the *wifi_tag_svc* plugin responds to the event by calling PeerAppear's *ApplyAnnotation* API function specifying the image to be annotated, the *BSSID* annotation field, and a list of WiFi AP BSSIDs detected at the time the image was captured.

Once applied, image annotations can be queried and retrieved from the local collection or remote peers through the PeerAppear API. An example query and associated results are shown in Figure 27. In this example, the query specifies a location code and indicates that results should include the *BSSID* annotation, which was applied previously by the *wifi_tag_svc* plugin.

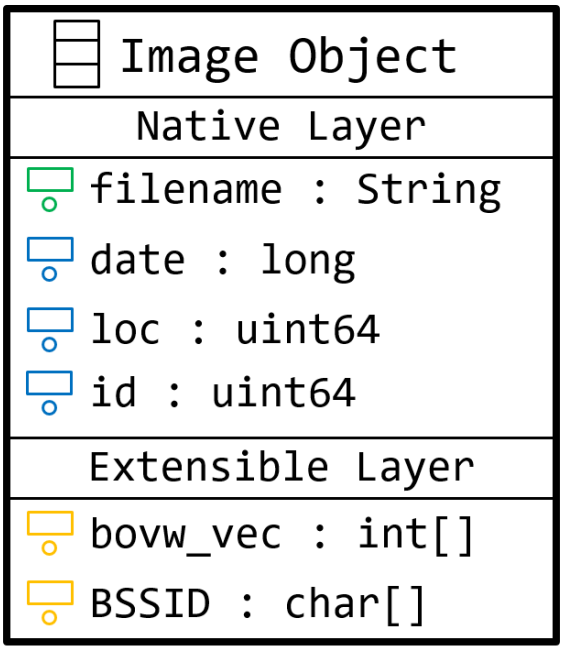


Figure 25. The structure of layered image data within the PeerAppear database.

```
<?xml version="1.0" encoding="UTF-8"?>
<wds version="0.1">
    <extension>
        <name>wifi_tag_svc</name>
        <annotations>
            <name>BSSID</name>
                <type>text[]</type>
        </annotations>
        <events>OnImageAdd</events>
    </extension>
</wds>
```

Figure 26. The framework loader XML description of the wifi_tag_svc plugin.

```
<?xml version="1.0" encoding="UTF-8"?>
<wds version="0.1">
    <query id="ac75fe36">
        <field>loc</field>
            <value>951977d377e723a0</value>
        <select>BSSID</select>
        <select>loc</select>
        <select>date</select>
    </query>
</wds>

<?xml version="1.0" encoding="UTF-8"?>
<wds version="0.1">
    <response id="ac75fe36" count="2">
        <result>
            <field>BSSID</field>
                <value>84e0b2f47f78</value>
            <field>loc</field>
                <value>951977d377e723f9</value>
            <field>date</field>
                <value>1458331612</value>
        </result>
        <result>
            <field>BSSID</field>
                <value>256150e0ded9</value>
            <field>loc</field>
                <value>951977d377e723f4</value>
            <field>date</field>
                <value>1458330375</value>
        </result>
    </response>
</wds>
```

Figure 27. An example PeerAppear query and associated results.

Two-Stage Visual Search with *tf-idf* Weighting.

The search for visually similar images within the PeerAppear framework is accomplished using the bag of visual words (BoVW) model for content-based image retrieval (CBIR). BoVW is based on the bag of words (BoW) model first used to enable keyword and keyphrase search within a large corpus of textual documents. While textual documents and associated queries can be described using a readily available vocabulary, no standard vocabulary for visual elements exits. Therefore, under the BoVW model, highly descriptive and repeatable visual elements in images are identified and quantized to form visual words based a pre-computed visual vocabulary. The content of an image can then be summarized using histogram vectors which indicate the number of occurrences of each visual word found in the image. Our framework makes use of the scale-invariant feature transform (SIFT)[62] to identify and describe

visual features.

The visual vocabulary is constructed by first aggregating the SIFT features extracted from a randomly selected sample of images into a single feature set. The k-means algorithm is applied to this feature set to identify centroids in the high-dimension feature space. The number of centroids specified is a tunable parameter, with higher values providing better discriminative power and lower values providing for better generalization when quantizing visual features. The set of centroids resulting from the application of the k-means algorithm to the feature set represents the visual vocabulary used to quantize image features in order to form compact image descriptors.

$$
\begin{aligned}
I &= \{i_0, i_1, ..., i_n\} && \text{random sample of images} \\
F &= \bigcup_{i \in I} SIFT\_Extract(i) && \text{extract all features} \\
w & && \text{dictionary size} \\
V &= kmeans(F, w) && \text{generate vocabulary} \\
V &= \{v_0, v_1, ..., v_w\} && \text{vocabulary consists of centroids}
\end{aligned}
$$

Images in a repository can effectively be indexed using their BoVW vectors, forming the basis for a search capability for the image repository. These vectors are generated using the vocabulary to vector quantize SIFT features extracted from an image. The number of occurrences of each word in the image is counted and stored in a vector indexed by the ordered list of visual words.

$$
\begin{aligned}
i & \qquad \text{input image} \\
F = SIFT\_Extract(i) & \qquad \text{extract image features} \\
= \{f_0, f_1, ..., f_n\} & \\
D = Vector\_Quantize(F, V) & \qquad \text{quantize features} \\
= \{d_0, d_1, ..., d_n\} & \\
i_{BoVW} = Hist(D) & \qquad \text{generate histogram} \\
= [|v_0|, |v_1|, ..., |v_w|] &
\end{aligned}
$$

The BoVW representations for the collection of local images can then be used to construct a *tf-idf* weighted search matrix, enabling individual words to contribute to search results in a manner that is inversely proportional to the frequency the word occurs in the overall collection. The term frequency (tf) component of the search matrix is constructed by aggregating the individual image vector representations for all images in the global collection as rows of the matrix. The inverse document frequency (idf) component is a vector representing the number of documents in which each term of the visual vocabulary occurs.

Each row of the term frequency matrix (M_{tf}) consists of a single image's term frequencies, or count of the occurrences of term t_j in image I_k (eq 1).

$$M_{tf}(k, j) = |t_j \in I_k| \tag{1}$$

The idf of each visual term is equal to the log of the total number of images divided by the number of images in which the term occurs (eq 2).

$$\overrightarrow{idf}(j) = log\frac{|I|}{1 + |\{i : t \in i\}|} \tag{2}$$

The idf vector, represented as the diagonal of a $|T| \times |T|$ matrix, is multiplied

with M_{tf} to create the weighted search matrix (eq 3).

$$M_{tf-idf} = M_{tf} \times diag(\overrightarrow{idf}) \tag{3}$$

The *tf-idf* weighted search matrix is then row-wise normalized (eq 4).

$$M_{tf-idf} = \frac{M_{tf-idf}}{||M_{tf-idf}||_2} \tag{4}$$

By normalizing the search matrix rows to unit vector length the cosine similarity of a query and all rows of the matrix can be calculated by taking the dot product of the matrix and a normalized query vector Q_{tf} (eq 5).

$$R = M_{tf-idf} \cdot \left(\frac{\overrightarrow{Q}_{tf} \times M_{idf}}{||\overrightarrow{Q}_{tf} \times M_{idf}||_2} \right) \tag{5}$$

The result of the dot product is a vector of cosine similarity measures indicating how similar the query image is to each image represented in the weighted search matrix. These values are then sorted and the rows which produce the highest values are considered as potential candidates for visual location matching. The result of a query generated using an image from our experimental dataset is shown in Figure 28.

While the search method using *tf-idf*-weighted BoVW vectors described above forms the basis for an effective visual search capability, highly-ranked false matches still occur regularly. One of the primary causes of these false matches is the absense of locality for features described by an image's BoVW histogram. Regardless of where in an image the visual words occur, any two images with similar numbers of visual words shared in common will generate higher cosine similarity results when the dot product of their *tf-idf*-weighted BoVW search vectors is evaluated.

Due to the presense of these false matches, a final validation step must occur before making a location inference. This validation, shown in eq 6, is performed using

Figure 28. Results of image similarity search, top left is query image.

epipolar geometry and random sample concensus (RANSAC) to estimate the rotation (R) and translation (t) between matched image pairs. The process takes as input the label and pixel coordinates of visual words which are shared in common between the two images. All possible combinations of visual word matches are enumerated for evaluation by the RANSAC algorithm.

$$\begin{aligned} x, x' = (d_1, d_2)\{\forall D | d_1 \in D, d_2 \in D, d_1 = d_2\} \\ E = findEssentialMat((x, x')) \\ [t] \times R = E \\ x^{'\mathrm{T}}{}_{\mathrm{n}}[t] \times R x_n = 0 \end{aligned} \tag{6}$$

The products of the algorithm include the best fit R and t values along with a bitmask identifying the matches used to compute the values.

Bitmasks indicating that large numbers of matches conform to the selected R and t values are indicative of positive image matches. An example validated match from our experimental dataset is depicted in Figure 29.

The methods decribed form the basis for a visual navigation capability using a database of geolocated imagery. The accuracy, scale, and granularity of this capability is entirely dependent on the accuracy, scale, and granularity of the images within the collection and their associated geospatial coordinates. While a single agent could conceivably populate a local repository with sufficient numbers of georeferenced images to support a visual localization capability in a small area, maintaining an up-to-date repository for larger areas requires collaborative effort from many participants.

Figure 29. A pair of visually matched images with RANSAC-validated feature correspondence.

Indexing Algorithms and Support.

While current state-of-the-art image search and index systems[48, 47] are well-suited for large-scale operations, they are not specifically designed to provide search results supporting a visual location recognition capability, nor are they designed to identify the importance of individual images within a large collection for use in visual location recognition. Ideally, such a system would confine search efforts to those images which have been identified as being useful for visual localization in order to maximize the quality of search results while minimizing search times and the consumption of computational resources. However the automation of indexing, search, and scoring for a large repository of images has not been greatly explored.

To aid the discovery of important images near a landmark, we propose area of visibility scoring supported by a system of offline processes to analyze and score images. Images can then be annotated with this core, enabling more efficient search and filtering capabilities for certain image queries within the PeerAppear framework.

In this section algorithms are presented to solve the problem of efficiently finding location-relevant visually matched images within a large corpus of stored images given a specific query image through the use of the area of visibility scoring model. The area of visibility measure can scope a geographic area of interest for location recognition in a manner consistent with Tobler's Law. To our knowledge, no previous work has exploited BoVW similarity with an area of visibility scoring mechanism.

VLSS Process Algorithms.

This section describes the algorithms and structures comprising each of the 3 VLSS offline processes.

Visual Feature Extraction and Representation Builder (VFERB)

The Visual Feature Extraction and Representation Builder (VFERB) is respon-

sible for extracting visual features from images and building a codebook with which to represent the features from the global collection of images in an abridged manner. This is accomplished for the global collection of images by applying the k-means algorithm on a sample set of images with k being equal to the desired number of visual terms. The prototypes generated by the k-means algorithm are then used to quantize each of the SIFT features found in an image. A vector of length k is then constructed to represent the image where the i^{th} element represents the number of features contained in the element which are assigned to the i^{th} prototype by vector quantization. The process assigns vector representations to images as they are added to the global collection or when the visual vocabulary changes. The visual vocabulary is recomputed when a pre-defined threshold for changes in the global image collection is exceeded. Updates ensure that the codebook remains representative of the global feature space. The vector representation for a single image can be seen in figure 30.

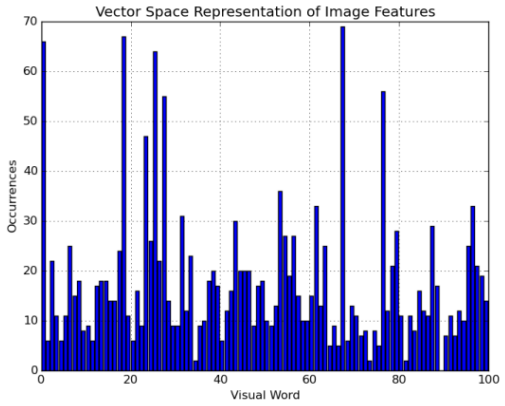


Figure 30. A histogram depicting the vector representation of the words describing the features of a single image.

Visual Search Service (VSS)

The Visual Search Service is responsible for constructing a *tf-idf* weighted search matrix for all images stored in the global image collection. Formal mathematical descriptions of the VSS processes are shown in Section 4.3. The term frequency (tf)

component of the search matrix is constructed by aggregating the individual image vector representations for all images in the global collection as rows of the matrix. The inverse document frequency (idf) component is calculated by the VSS process using map-reduce style processes to find the number of documents in which each term of the visual vocabulary occurs. The *tf-idf* weighted search matrix is then row-wise normalized.

To execute search the dot product between the search matrix and a *tf-idf* weighted search vector is performed. The result of the dot product is a vector of cosine similarity measures indicating how similar the query image is to each image represented in the weighted search matrix. These values are then sorted and the rows which produce the highest values are considered as potential candidates for visual location matching.

Area of Visibility Scoring Crawler (AVSC)

While the *tf-idf* weighted similarity scores generated by the VSS process are sufficient for matching a given image with the most visually relevant matches, they do not necessarily identify the potential usefulness of individual search results for use in location recognition. For example, a visual search using terms derived from a picture of the Washington Monument may return images of models of the monument, images of pictures of the monument, or images of some entirely different structure that just happens to contain features similar to the Washington Monument. The images which match the actual monument are useful for location recognition, while those that do not shouldn't be used to infer a location. In addition, images which contain the largest iconic structures and geographic features are generally those that can be seen from the largest area, and should therefore be designated as "interesting" for location recognition and weighted accordingly. This idea forms the basis of the area of visibility measure for determining an image's importance for location recognition.

In order to identify images which correspond to the actual monument a visual

search must first be performed. This search is likely to yield both monument and non-monument results. Because each image contains the location of the device used to capture the image, the geographic locations of matched features between pairs of images can be estimated using epipolar geometry and a process known as random sample consensus (RANSAC). Through an iterative voting process, the largest collection of images for which the physical geographic location of the monument, within a pre-specified threshold parameter, can be agreed upon will be selected as the set of images containing the true monument. The geographic locations of a notional set of images selected through this process for the Washington Monument is indicated with red triangles in figure 31.

Figure 31. The positions of geo-located images containing a confirmed corresponding element (Washington Monument) for localization.

With the set of results now filtered to contain only images that are likely to be true representations of the monument, the next task is to assign location scores values to the results. Because all of the images in the final validated set were successfully discovered through visual search and met the threshold criteria for the winning physical geographic location for the image subject matter, we will assume that they are all equally important for location recognition. Therefore, we will assign all images in this validated set the same score. The area of visibility scoring system makes the assumption that the importance of an iconic structural or geographic feature for location recognition is related to the size of the area from which it can be seen. By

using area of visibility as a scoring mechanism the images of objects with large areas of visibility will be ranked higher in searches than those with small areas of visibility.

The area of visibility score is determined using a correlation ellipse on the 2-dimension coordinates of the set of validated images with a confidence interval defined as a user-specified parameter. While the specific area of the set of validated images could be determined by constructing a polygon consisting of edge vertex boundaries, doing so would be unlikely to yield an accurate estimate in the presence of sparse numbers of samples. In addition, the confidence value selected serves to diminish the effect of any outliers that should inadvertently make it into the validated set of images, thereby making the system robust to error introduced by mislabeled or misclassified images. A notional correlation ellipse for the visible area of the Washington Monument is shown in figure 32.

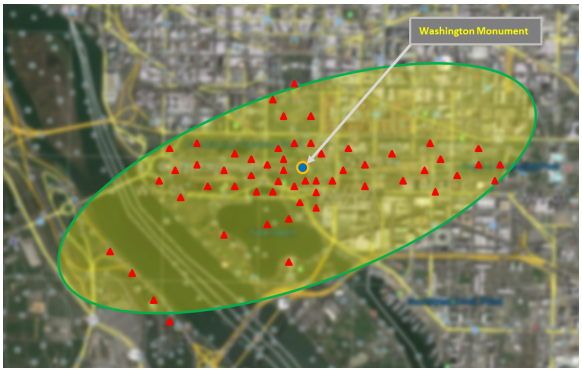


Figure 32. The correlation ellipse representing the area from which the iconic structural or geographical element (Washington Monument) can be seen.

The AVSC process builds upon the visual search capability implemented in the VSS process to include the importance of an image for location recognition within the global collection of images. An instance of the AVSC service includes a single task assignment process and multiple worker processes. While the worker processes could be implemented in an entirely independent parallel manner, the introduction of an assignment process reduces the chances that worker processes will inadvertently

select the same or similar images, thereby preventing duplication of effort. Both of these results support the scalability design goal for the system. In addition, the assignment process is also responsible for submitting batch updates to the database, thereby reducing the number of IO operations necessary.

The task assignment process begins by randomly selecting n un-scored images corresponding to the n available worker processes from the global image collection.

$$S = SelectRandom(I, n, i.lscore = Null) \tag{7}$$

The selected images are then distributed by the task assignment process to each of the worker processes. Once tasked, each worker process begins by submitting their assigned image to the VSS process as an image search query. The search will return a ranked list of visually similar images, including the original image used to perform the search.

$$R = VSS\_Search(s_i) = \{r_0, r_1, ..., r_n\} \tag{8}$$

The process then extracts visual features for the images returned from the visual search.

$$F_i = SIFT\_Extract(r_i) = \{f_{i,0}, f_{i,1}, ..., f_{i,n}\} \tag{9}$$

Using a random sample consensus (RANSAC) approach, the process iteratively determines the mean location of the object represented by different subsets of the

images. This iterative process begins with the selection of two image feature sets at random from the visual search results.

$$S = SelectRandom(F, 2) \tag{10}$$

The features of the two images are matched and validated using the epipolar constraint and the location and orientation meta-data for each image.

$$M = ValidateMatches(S_0, S_1) \tag{11}$$

The mean 2-dimension location for the validated set of matched features is then calculated. This location is an estimate of the position of the object. The remaining images are then evaluated to determine if they contain the object at the estimated location. The number of images which agree upon the object's location are counted as votes indicating that the estimated position is correct. The location that receives the greatest number of votes and the set of images to which it corresponds is selected as the validated set.

$$C = Valid\_Matches(F, L) \tag{12}$$

$$w = max(w, |C|) \tag{13}$$

$$Q = \begin{cases} C, & \text{if } w = |C| \\ Q, & \text{otherwise} \end{cases} \tag{14}$$

The full estimation and voting process is repeated a fixed number of times to find

the largest set of images which correspond to the same location. The 2-dimension coordinates of the resulting validated set of images is then used to compute the area of a covariance ellipse for the set as depicted in figure 32. The ellipse is an estimate of the area from which the object can be seen and represents the importance of the iconic structural or geographic feature for location recognition. This area is assigned to *lscore*, the location score, for each image in the validated set. The scoring updates are returned to the tasking process to be saved to the processed data layer for each image. The worker process flow is captured in the flow chart depicted in figure 33.

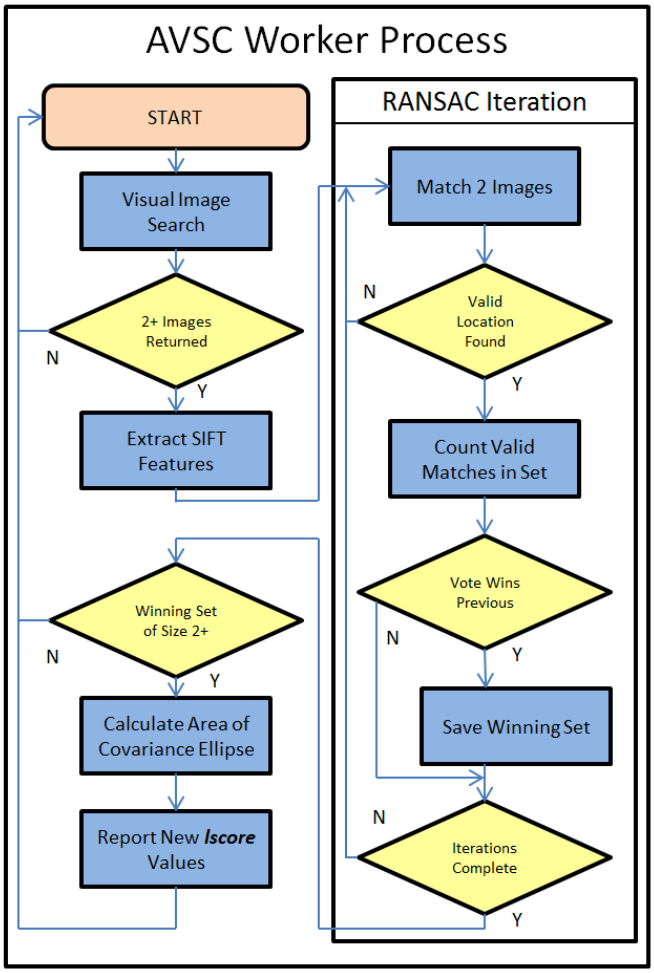


Figure 33. A flowchart depicting the AVSC worker process used to select and score images.

4.4 Geospatial Filtering for Position Representation and Search

Support for cell-based geospatial representation is one primary advantage PeerAppear's distributed index has over its nearest competitors, which only support radius-based search. Using cell-based geospatial representation enables PeerAppear to effect index operations using search filters comprised of a heterogeneous mix of cells representing both continuous and discontinuous search areas of nearly any shape and size. In this section we first demonstrate the ability to construct search filters for circular representations of position, thereby mimicking the capability offered by PeerAppear's competitors. Improving upon the circular representation, we demonstrate the creation of search filters based on the viewshed of a specified search location. Next a method which produces filters defined by the viewshed of points along a path is described and demonstrated. Finally, we present *S2Trie*, a data structure created to efficiently represent and compress complex cell-based search filters.

Position Representation.

The representation of position using radius-based search filters is frequently employed in PeerAppear's visual localization experiments. By adjusting radius, the size of the search area can be extended to account for uncertainty of position estimates or locomotion since the last successful position inference was achieved. Geospatial parameters for a radius-based search filter include its center point, defined by latitude and longitude, and radius, defined in meters. The number of cells used to produce the filter must also be specified.

Allowing the filter to be defined using greater numbers of cells improves search precision at the cost of efficiency, while using lower cell-count filters reduces precision while generally improving efficiency. An example search filter constructed using 11 cells can be seen in Figure 34.

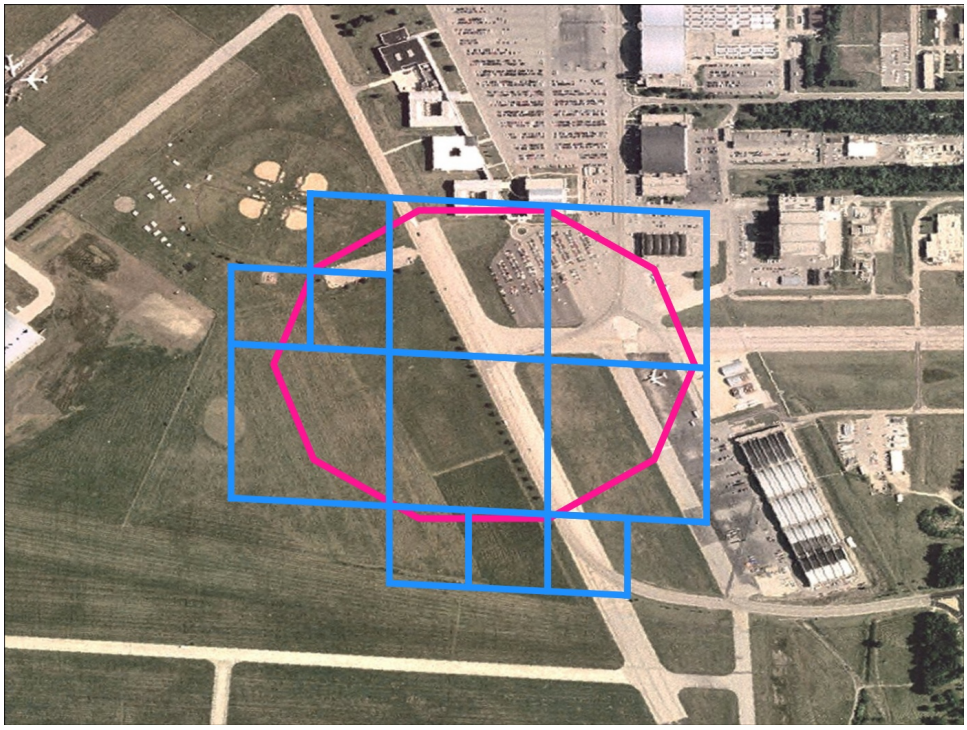


Figure 34. A position estimate filter comprised of 20 S2 cells.

As cell count decreases fewer lookup operations are needed when iteratively executing PeerAppear's *FIND_VALUE* RPC to satisfy the filter. However, the loss of precision imposed by larger sized cells may lead to searches which identify published keys which are not in the search radius. An example search filter constructed using only 4 cells can be seen in Figure 35

The loss of precision which occurs when specifying smaller maximum cell counts is demonstrated in Table 5. For various maximum numbers of cells used to cover the search space, the table shows the the number of cells the cover actually uses, the coverage of the search space, and the spill factor. The coverage, Equation 15, is the percent of the search space covered by the search filter. The spill factor, Equation 16, represents the space covered by the search filter which is not in the originally specified circular search space and is presented as a multiple of the search area size.

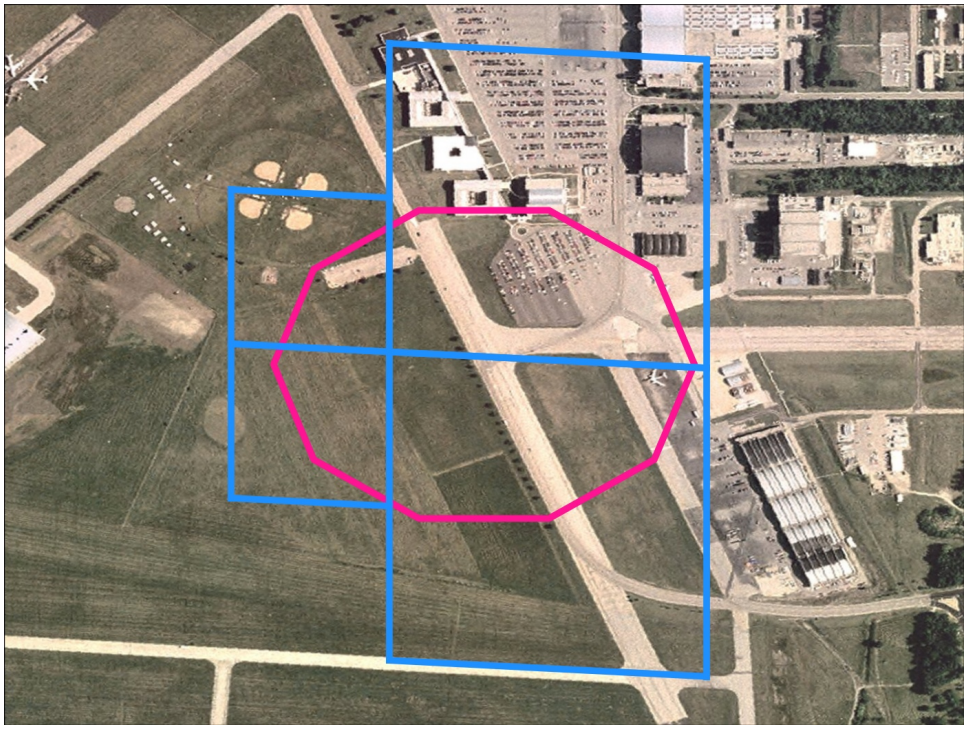


Figure 35. A position estimate filter comprised of 4 S2 cells.

$$cover = \frac{\sum_{i=1}^{n} area(cell_i \cap search\_space)}{area(search\_space)} \tag{15}$$

$$spill = \frac{\sum_{i=1}^{n} area(cell_i) - area(cell_i \cap search\_space)}{area(search\_space)} \tag{16}$$

Large spill factors are undesirable as they indicate area represented by the filter which was not intended to be part of the search space. For the example search filter covers shown above in Figures 34 and 35, the spill factor worsens from 0.388 for the 20 cell cover to 1.440 for the 4 cell cover.

The search filter shown in Figure 34 depicts several search cells which only minimally intersect the specified search space. By applying a pruning approach to our search space cell coverings the number of cells used to define the search filter can be effectively reduced with minimal effect on search coverage. This is accomplished

Table 5. The search space cell counts, coverage and spill factor for radius-based search filters created using specified maximum count cell covers.

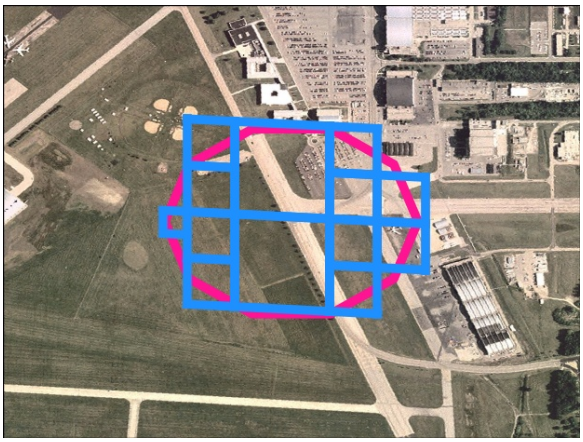
Max Cells	Used Cells	Search Coverage	Spill Factor
1	1	1.000	4009.709
2	2	1.000	30.226
3	3	1.000	15.104
4	4	1.000	1.440
5	5	1.000	0.952
10	10	1.000	0.556
15	15	1.000	0.449
20	20	1.000	0.388
25	22	1.000	0.327

by specifying a minimum coverage pruning constraint. The list of cells which comprise the search filter are sorted based on the percent coverage they contribute to the overall filter and an iterative pruning process is performed starting with the lowest-contributing cell until the minimum coverage pruning constraint is met.

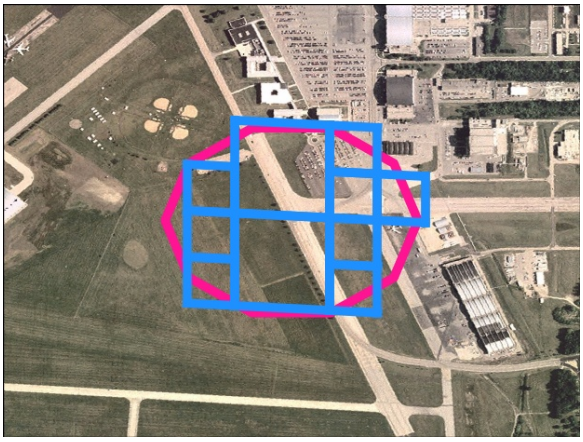
The results of this pruning strategy are shown in Table 6. Cell counts, coverage, and spill factor are listed for pruned versions of the filters previously shown in Table 5 for 95%, 85%, and 75% minimum coverage constraint values. A depiction of pruned search filters for the 25 cell count search cover using 95%, 85%, and 75% minimum coverage constraint values is shown in Figure 36. The results demonstrate effective cell count reduction, reduced spill factor, and coverage which meets the minimum specified constraint.

Viewshed Representation.

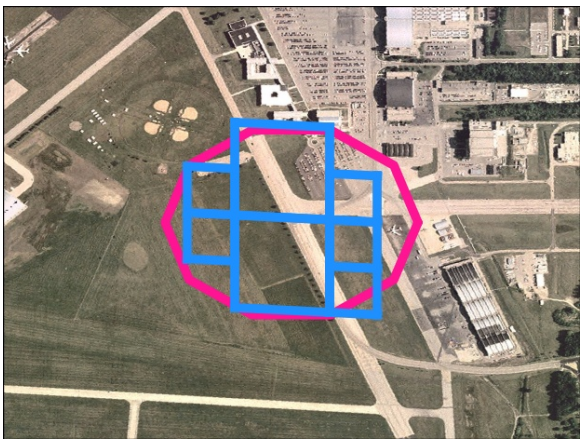
A viewshed is the area that is visible from a specific location[37]. In previous work on visual viewshed estimation[24], the importance of viewshed was identified as playing a key role for efficient visual search. This is because images are georeferenced based on the location of the device capturing the image, not on the locations of the content of the image. Therefore, when searching for images of a specific location,



(a) Search filter with minimum coverage pruning constraint of 95%.



(b) Search filter with minimum coverage pruning constraint of 85%.



(c) Search filter with minimum coverage pruning constraint of 75%.

Figure 36. The result of pruning max 25-count search filters using various minimum coverage pruning constraint values.

Table 6. The cell counts, coverage and spill factors for search filters created using 95%, 85% and 75% minimum coverage pruning constraints and specified maximum count cell covers.

	Minimum Coverage Pruning Constraint								
	95%			85%			75%		
Max	Cells	Cover	Spill	Cells	Cover	Spill	Cells	Cover	Spill
1	1	1.000	4009.709	1	1.000	4009.709	1	1.000	4009.709
2	2	1.000	30.226	2	1.000	30.226	1	0.786	14.830
3	3	1.000	15.104	2	0.910	14.950	1	0.786	14.830
4	4	1.000	1.440	3	0.910	1.286	2	0.786	1.166
5	5	1.000	0.952	4	0.910	0.798	3	0.786	0.678
10	6	0.989	0.475	5	0.899	0.321	4	0.775	0.201
15	8	0.958	0.261	6	0.896	0.201	4	0.775	0.201
20	13	0.962	0.197	10	0.884	0.091	7	0.761	0.032
25	13	0.954	0.159	10	0.884	0.091	7	0.761	0.032

any image located within the viewshed of the search location could potentially be relevant.

The requirement for PeerAppear to support cell-based search filters was driven in large-part by the need to perform viewshed-based search. This type of search is motivated by structure from motion (SFM) and distributed surveillance application use-cases. For example, suppose PeerAppear were to be used to gather images of a monument in order to build a model of it using SFM. If a circular search filter, centered on the monument, were used an arbitrary search radius would have to be specified. Based on the size of this radius as well as nearby structures and landscape topology, the filter could potentially cover significant search area which would be unlikely to contain images of the monument. By instead constructing the search filter based on the viewshed of the monument the precision of the search would increase because the search space would be restricted only to those images captured in the monument's viewable area.

The creation of viewshed-based filters within this research thrust was suported by GRASS GIS[43], an open-source multi-platform geographic information system (GIS)

package. Source terrain elevation data (TED) was obtained from the United States Geological Survey's 3DEP program. After loading this TED into GRASS GIS, the *r.viewshed* plug-in is used to calculate the viewshed given a specified coordinate and altitude above ground level.

A gray scale elevation map of our first test location can be seen in Figure 37. This location was selected because it was familiar to the author and has desirable geographic qualities. Because the airpark is surrounded on three sides by the Great Miami River, a tall dike was constructed to prevent flooding. The dike restricts ground-level visibility, making the area within the dike a perfect test location for search filters based on viewshed.

A point 2 meters above ground near the west end of the runway at Moraine Airpark was selected and the *r.viewshed* plug-in was used to calculate a viewshed. The viewshed, shown in Figure 38, is comprised of numerous high-resolution discontinuous polygons.

This viewshed was then converted into a two-dimension point cluster in order to more easily overlay it with S2 cells. The point cloud is shown in Figure 39.

Covering the two-dimension viewshed point cluster with level 16 S2 cells is then used to create a search filter which is compatible with PeerAppear's geospatial search. This filter is shown in Figure 40.

The search filter shown in Figure 40 is comprised of 235 level 16 S2 cells. Many of the adjacent cells present opportunity for reduction in cell count and filter simplification through the merging of smaller (higher level) cells to create larger (lower level) cells. This process is demonstrated in Section 4.4 where the *S2Trie* data structure is presented and tested.

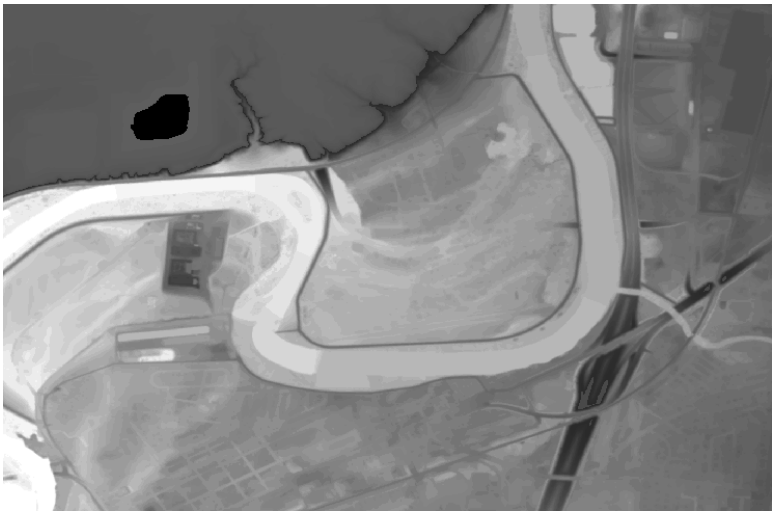


Figure 37. A gray scale terrain elevation map of Moraine Airpark in Moraine Ohio.

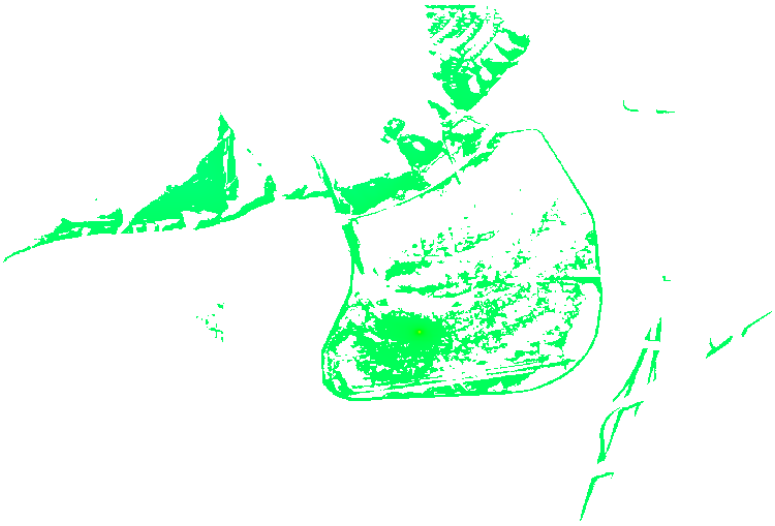


Figure 38. A viewshed created by GRASS GIS for a point selected near the west end of the runway at Moraine Airpark.

Figure 39. The viewshed from Figure 38 represented using a 2D point cluster.

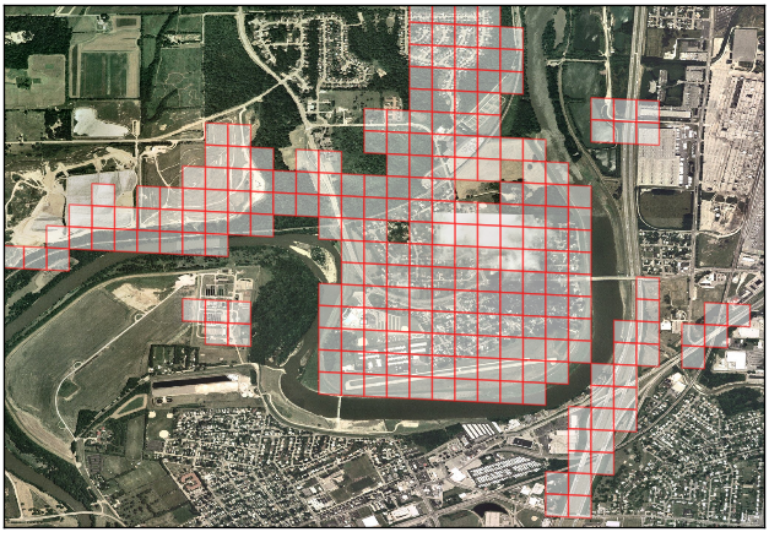


Figure 40. A viewshed-based search filter at Moraine Airpark created by covering the viewshed point cluster shown in Figure 39 with level 16 S2 cells.

Path Viewshed Representation.

Building on the methods presented in Section 4.4, we can further extend viewshed filters from single point representation to path representation. While viewshed filters created using a single geospatial coordinate may be useful for finding images of a stationary object, they are less useful for finding images of non-stationary objects. For example, if someone wanted to search for images of trains along a specified section of track, the images could exist anywhere within the viewshed of any point along the track.

The process of constructing a path viewshed representation begins with the definition of our path, Π, as shown in Equation 17. The path is defined as a list of individual points.

$$Path\Pi = \{\pi_1, \pi_2, \pi_3, \pi_4, ..., \pi_n\} \tag{17}$$

The viewshed of each point is calculated as described previously in Section 4.4. The path viewshed is then created by finding the union of all points' viewshed filters while excluding duplicates, as shown in Equation 18.

$$Viewshed(\Pi) = \bigcup_{i=1}^{n} Viewshed(\pi_i) \tag{18}$$

An example path viewshed filter and corresponding viewshed density plot is depicted in Figure 41. Again, the search filter is comprised of a simple homogeneous cell-based covering using level 16 S2 cells. As in the previous point viewshed example, opportunities for cell count reduction and filter simplification are obvious. In the next section, this process is demonstrated along with a novel density-based pruning strategy which enables the removal of cells which contain minimal viewshed area.

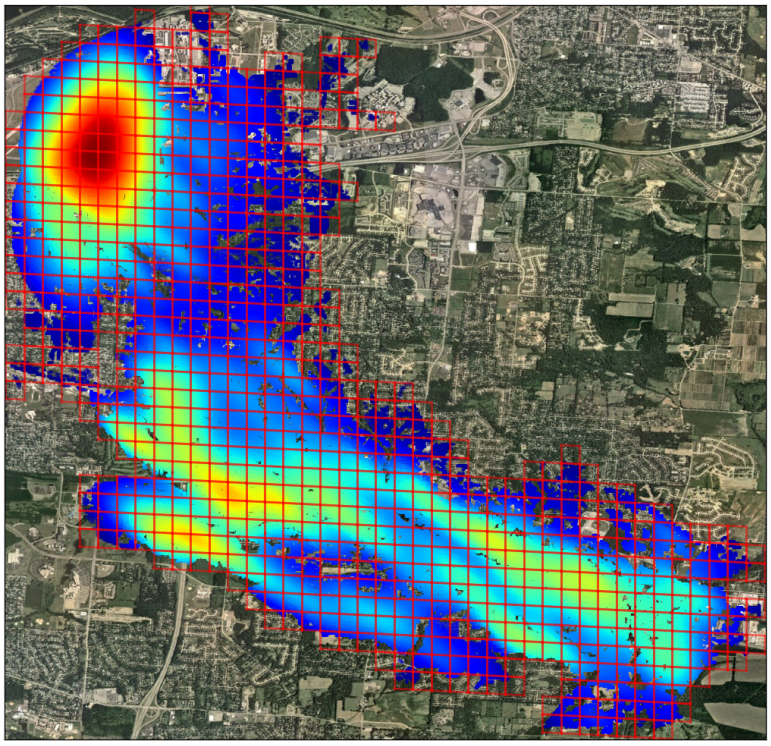


Figure 41. A path viewshed filter and density plot for a roadway route between Beavercreek Ohio and Wright-Patterson Air Force Base.

S2Trie for Geospatial Filter Representation.

In Section 4.4 the creation of search filters based on point viewshed and path viewshed was demonstrated. The examples show that search filters can be comprised of numerous adjacent or non-adjacent cells for both continuous and discontinuous search spaces. For the example filters, a naive covering approach of the search space was used which resulted in filters consisting of many cells of the same cell level. Because S2 cells are subdivided in a hierarchical structure as the cell level increases, groupings of 4 adjacent cells which share the same parent offer opportunities for cell count reduction through merging. PeerAppear uses an iterative lookup process when performing search based on multi-cell search features. This process executes for each cell in a filter causing high cell-count filters to suffer a penalty in terms of communications and index operations complexity. Therefore, reducing a filters cell

count should be considered a high priority.

As part of this geospatial filtering research effort a new data structure named *S2Trie* was created to simplify filter representation and cell count reduction. The *S2Trie* is implemented as a custom class in Python. Much like a radix or prefix trie used to store words or data with bitwise representations, *S2Trie* stores the S2 cells which represent a geospatial filter using the bitwise structure of their binary S2 cell addresses. Unlike other binary prefix tries, *S2Trie* organizes the tree structure based on the groupings of binary bits used by the S2 spherical geometry library. Therefore the first 3 bits of an address, which represent the cube face that the address is on, form the first level of the tree. Subsequent levels are formed for every additional 2 bits of address. This gives the trie 6 possible nodes at the first level, representing the cube faces, and up to 4 children per parent for subsequent levels.

A graph-based representation of the structure of an example *S2Trie* filter is depicted in Figure 42. In this example, a filter consisting of 5 S2 cells is shown with the filter's cells represented by the leaf nodes shaded in green. All of the filter's cells reside on the North American cube face as indicated by the *100* address of the filter's sole level 0 parent. Leaf nodes exist at levels 3, 4, and 5 indicating that the filter is comprised of 3 different sizes of S2 cells.

Filters are constructed with the *S2Trie* class through the iterative addition of S2 addresses using the class' *add_node* method. The method takes the address of the new cell as its sole parameter and builds out a new tree, or adds branches if one already exists, in order to include the new cell. When the leaf node for a newly added cell causes the number of child cells for its immediate parent to increase to 4 the opportunity for cell merging is identified. Merging is accomplished by eliminating child nodes and promoting the parent to a leaf node. The process executes recursively because the promotion of the parent to a leaf node can again present a merging

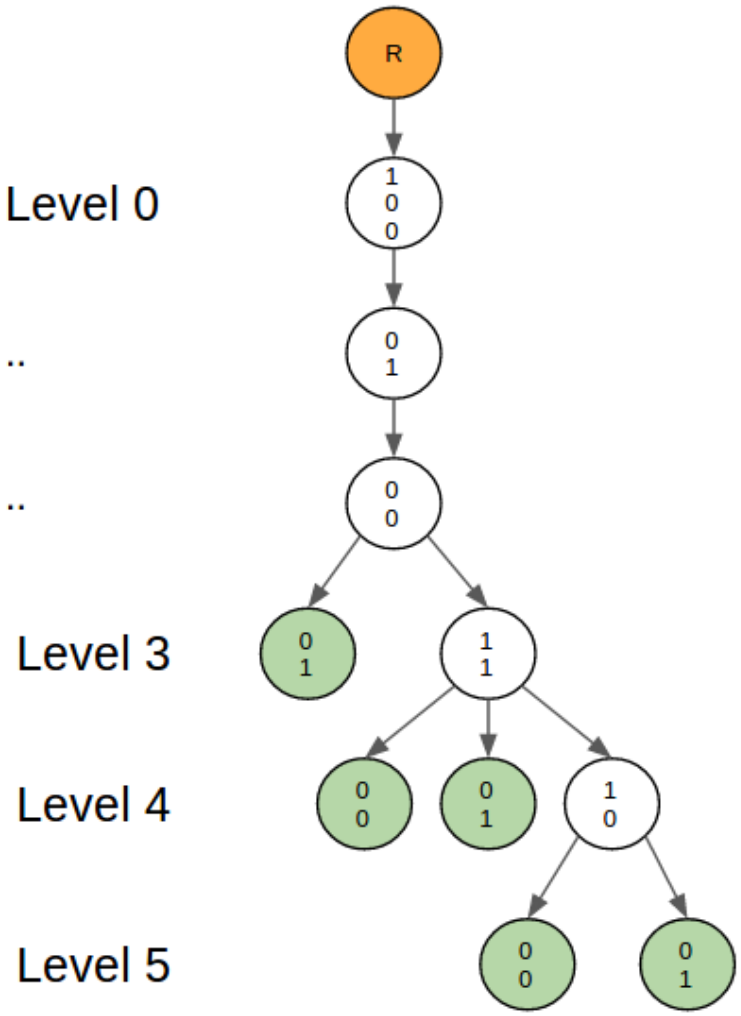


Figure 42. An example *S2Trie* instance containing level 3, 4, and 5 cells on the North American (*100*) cube face.

opportunity if the node already had 3 leaf node siblings.

By using the *S2Trie* class to represent and simplify the filter created previously in Figure 40 we are able to reduce the filter's cell count from 235 to 136. The resulting 136-count viewshed filter is shown in Figure 43. The simplified filter covers the exact same area as unsimplified version with a 42% reduction in number of cells.

In Section 4.4 we demonstrated the pruning of position representation filters based on a minimum coverage pruning constraint. This pruning process was used to remove cells from filters where the cell's coverage only minimally improved the filter's coverage of the desired search space. In a similar manner, we can prune viewshed filters using a minimum viewshed point density constraint to further reduce their cell counts. This process is built in to the *S2Trie* class as the *prune_by_density* method. The method takes the minimum viewshed point density, expressed as points per square meter, as its sole parameter. When executing, the method begins a depth-first recursive enumeration of the trie and prunes all branches where viewshed point densities exceed the minimum density threshold.

A graph showing the resulting cell count of the filter from 43 after density-based pruning for a range of minimum density values is shown in Figure 44. This graph demonstrates the potential for this approach to reduce cell counts to a target number through iterative pruning. In the graph a significant and rapid reduction in cell count can be seen as the minimum density decreases from 5.35 points per square meter, with 24 cells, to 5.2 points per square meter, with 6 cells. A geographical depiction of the 2 resulting filters is shown in Figure 45.

In this section the process of creating geospatial filters for radius and viewshed-based search spaces was described and demonstrated. A process for cell count reduction through a minimum coverage pruning constraint was examined for radius-based filters. Next, a novel *S2Trie* data structure was presented enabling efficient storage

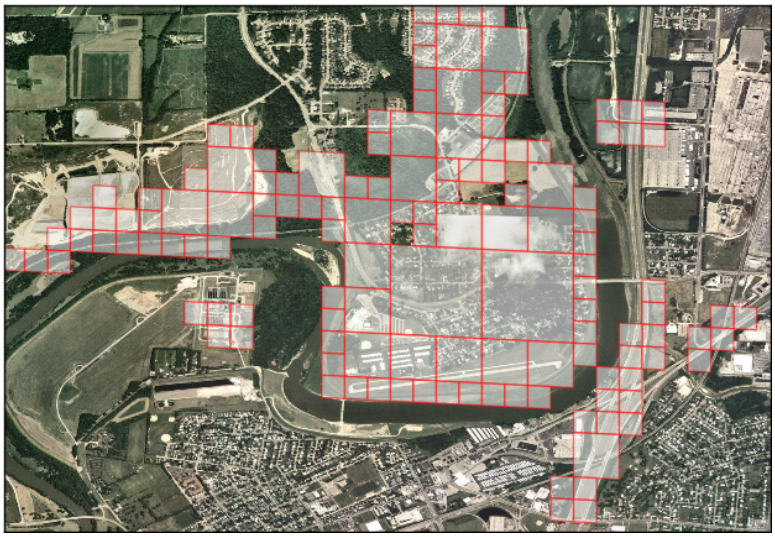


Figure 43. The viewshed cover shown in Figure 40 with all possible cell mergings completed.

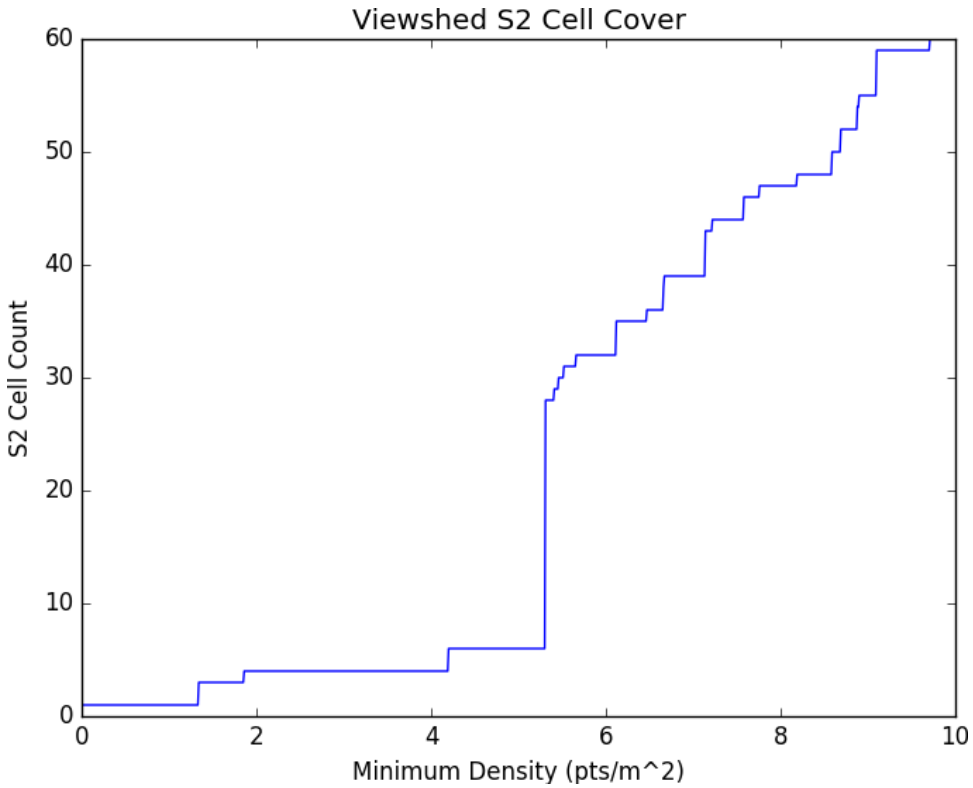


Figure 44. The cell counts of the filter from 43 after density-based pruning for a range of minimum density values.

(a) A search filter pruned to a minimum density of 5.35 points per square meter.

(b) A search filter pruned to a minimum density of 5.2 points per square meter.

Figure 45. Search filters for Moraine Airpark which are pruned using the *S2Trie*'s *prune_by_density* method.

and representation of S2-described search filters. Processes incorporated into the class as methods for hierarchical cell merging and pruning using a minimum density constraint were also presented and results demonstrating their effectiveness were examined for viewshed-based filters. These methods enable the creation of efficient geospatial filters based on search spaces for various shapes and sizes.

V. Integration of Middleware Infrastructure and Computational Support for Collaborative Visual Localization

In Chapter III a novel distributed geographic index was presented which enables participants of the PeerAppear framework to publish and search for content using an addressing scheme based on Google's S2 Spherical Geometry library. In Chapter IV computational support methods for visual search and geospatial filtering were presented. In this chapter the middleware infrastructure and computational support methods are integrated and evaluated for collaborative visual search and localization applications.

5.1 Related Works

Visual localization is perhaps one of the greatest foreseeable applications of the PeerAppear framework. This task involves identifying the location of an image based on visual content and associated annotations. Most previous work has treated this problem as an extension of content-based image retrieval (CBIR), also known as query by image content (QBIC) or content-based visual information retrieval (CBVIR). Under the CBIR model, a location inference can be made for an image by matching it to others in a large repository of georeferenced images based on similarity of extracted features and signature annotations.

One of the earliest visual localization methods to follow this model was proposed by Se et al[99]. Their system effectively demonstrated an indoor localization capability through the mapping and correlation of local visual features. These features, representing salient keypoints within an image, are extracted using algorithms that identify highly descriptive visual elements and represent them in numerical formats[96]. The work made use of an extraction method known as the scale invariant feature transform

(SIFT)[62] which produces 128-dimension descriptors that are invariant to changes in scale and rotation. Using a mobile robotic platform equipped with a trinocular camera system, the system extracted and stored those features which could be matched and localized in 2 of the simultaneously captured images. The resulting feature database then effectively formed the basis for an indoor navigation model, mapping planar feature arrangements to specific locations. Experimental results demonstrated that the system produced only minor offsets after completing several loops in small indoor environments.

While the system proposed by Se et al performed well in small indoor environments, the computational costs imposed by the matching process for larger database sizes made it impractical outdoors and in larger indoor environments. Ledwich and Williams overcame this shortcoming for larger indoor environments through dimensionality reduction for extracted SIFT features[56]. By restricting the rotation of the camera on their robotic platform and eliminating SIFT's rotation invariance, feature dimensionality was reduced creating 32 and 64 dimension features. The 64-dimension features were successfully able to match images at rates close to that of full 128-dimension features, however the 32-dimension features produced unacceptably high numbers of false positive matches. While the use of reduced SIFT features enabled localization in larger environments, it imposed restrictions on camera rotation which are likely to be considered unrealistic for use in outdoor environments.

One of the earliest successful methods for visual localization in outdoor urban environments, proposed by Robertson and Cipolla, determined location through the matching of building facade imagery[90]. Image matching was achieved using line segment features correlated with a random sample consensus (RANSAC)[36] approach. While the system demonstrated effective localization for a small database consisting of 200 building facade images, long inference times again highlighted the method's

computational complexity as an impediment for system scalability.

Making use of SIFT features, instead of building facade imagery, Zhang and Kosecka demonstrated a more generalized outdoor localization capability for urban environments[121]. The approach involved a 3-stage localization process which included the matching of reference views, motion estimation, and triangulation. Experimental evaluation using the ICCV 2015 competition dataset produced localization results that bested the results of the competition's original winner. Inference times of more than 1 minute per image again identified computational complexity as impeding the method's use for real-time localization with large-scale image repositories. Complicating matters, the authors specifically identified the need for denser model view databases in order to increase localization accuracy.

Applying the BoW model to the task of visual localization, Filliat demonstrated an approach enabling visual topological localization in medium-sized indoor environments[35]. As opposed to metric localization, which produces numerical position estimates, Filliat's topological approach was limited to identifying the room in which an image was captured. Location inferences were made using an SVM trained online with operator input used to label training data. Input for the SVM consisted of an image's SIFT BoW, color, and texture histograms. An experimental evaluation demonstrated high localization success rates for a small apartment.

Combining the BoW model approach with pose estimation, Wang et al proposed a 2-stage coarse-to-fine approach for metric visual localization[112]. The system included considerations for Zipf's Law[85], culling terms from its BoW vocabulary that appeared too frequently or not frequently enough. Words that appear too frequently lack discriminative power and are analogous to stop words in textual information retrieval, while those that appear too infrequently inflate the vocabulary for only minor discriminative gains. Using this consolidated vocabulary, SIFT features extracted

from images in a local repository could then be represented using BoW histograms, thereby enabling an efficient and expedient image search matching similar images in the collection to a given source image. The matching of similar images served both as a means of coarse localization and as a means of identifying potential candidates for the fine localization step. By restricting the pool of potential candidate images considered for pose estimation, the computational complexity is significantly reduced. An experimental evaluation was conducted using both indoor and outdoor image collections. Results demonstrated good localization accuracy at significantly reduced computational costs for larger environments.

The need for denser visual databases in order to achieve greater localization accuracy was identified by Zhang and Kosecka[121], however most prior efforts had demonstrated localization techniques using small, sparse, purpose-built image collections. Hays and Efros presented one of the earliest visual localization studies to leverage crowd-sourced georeferenced imagery in an effort to achieve a dense world-scale collection of imagery[49]. Their system, named IM2GPS, included a collection of more than 6 million crowd-sourced georeferenced images. Each image was processed to extract features including tiny images, color and texture histograms, line features, GIST descriptors, and geometric context. The large quantity of features extracted serves to minimize sensory gap[8]. The combination of all of an image's features were formed into a single image signature, enabling mean-shift clusters[21] to be generated and the distances between individual images in signature space to be calculated. An experimental evaluation using a georeferenced image test set demonstrated inference localization accuracies of 2500 kilometers or less for nearly 50 percent of the test set and 750 kilometers or less for nearly 25 percent of the test set. While the system was successful in achieving a world-scale localization capability, the coarseness of inferences limits its potential application.

Looking to improve upon IM2GPS's accuracy and efficiency, Weyand et al developed a method for localizing images using convolutional neural networks (CNNs)[116]. The system, which goes by the name PlaNet, works by training the neural network to map the locations of geotagged images in a large collection to cell-based geographic partitions. These partitions, generated using Google's S2 geographic library[12], enable the creation of partitions sized such that image densities among the global collection of cells are similar as seen in figure 46. The resulting model is able to produce a discrete probability distribution mapping a given query image to the cells in which it was most likely captured. Unlike other methods which rely on local features, PlaNet's approach is robust to articulation and minor changes often found in dynamic natural environments. An experimental evaluation conducted using IM2GPS's set of test images produced significantly higher localization accuracies than IM2GPS.

Both PlaNet and IM2GPS use approaches which require the aggregation of a large corpus of georeferenced images. IM2GPS must maintain access to the collection of image features while PlaNet's collection could potentially be discarded after training the CNN. PlaNet's inference time and space complexity are constant, unlike IM2GPS which grows in time and space complexity linearly with the size of its image collection. While neither the PlaNet or IM2GPS papers discuss model adaptation for dynamic updates, IM2GPS has potentially the clearest path through the addition of images, while PlaNet's CNN would require additional training in the best case and

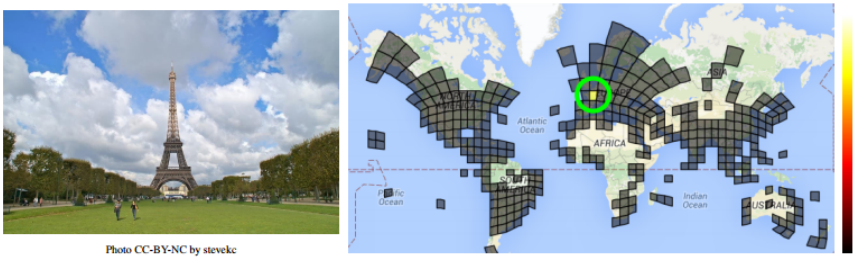


Figure 46. A location inference made by Google's PlaNet network to geolocate for a given image[116].

complete re-training in the worst case. The potential for scalability is apparent for both approaches, with PlaNet's approach besting IM2GPS, however both produce inferences that are sufficient only for very coarse position estimation.

5.2 Localization using the PeerAppear Visual Search Architecture

In this section the PeerAppear framework's theoretical search complexity is explored and an experimental evaluation of the visual search architecture is presented. The evaluation of theoretical complexity demonstrates a reduction in complexity for the framework from intractable for a naive approach to tractable for PeerAppear's approach. The experimental evaluation makes use of a hardware-based data collection device to construct a small-scale multi-track dataset consisting of both urban and rural visual content. The dataset is then used to test the effectiveness of the visual search architecture. Results for this analysis are reported in terms of precision and recall for experimental test tracks. Finally, a visual localization evaluation is presented which demonstrates rudimentary visual localization in both urban and rural environments.

Theoretical Search Complexity in PeerAppear.

PeerAppear achieves efficient and scalable operation through a series of image search and information retrieval optimizations. Starting at the lowest level of operation, the first optimization involves search for image similarity. A naive approach might involve directly comparing all possible pairs of f features extracted from each of n images in a local repository and ranking the results. This operation yields a complexity of $O(f^2(n-1)+n)$. While the BoVW approach for image similarity does require some pre-processing, the complexity for similarity search and ranking of results is reduced to $O(n(2v+1))$ where v is the size of the visual vocabulary.

The BoVW similarity search itself is an optimization used to efficiently narrow potential matches prior to executing the far more computationally expensive RANSAC algorithm. If RANSAC were to be used as a first-stage measure of image similarity, it would require $O(n(n-1))$ evaluations. Extending yet another level deeper, the RANSAC algorithm is an optimization that enables model parameters to be estimated without enumerating all possible combinations of potential feature matches. Assuming the use of a k-point algorithm for approximating the essential matrix and an average of f features per image, exhaustively calculating the optimal model parameters would require $O(\binom{k}{f})$ if features are first matched based on descriptor or visual word and $O(\binom{k}{f^2})$ if all possible combinations of salient points are tried. By approaching the search problem using BoVW to identify likely matches and RANSAC to validate them, the overall search complexity is reduced to $O(n(2v+1))$ plus a constant time operation for RANSAC validation.

PeerAppear's final optimization is achieved through its integration of S2 geospatial addressing and a Kademlia-style distributed index. The S2 library's hierarchical search space representations enable queries to be targeted at only those users who have published matching cell addresses to the DGT, thereby ensuring that search is restricted to repositories most likely to have corresponding images. Querying the DGT to identify those repositories is conducted through a recursive lookup process. Because the index self-organizes into a tree structure, recursive lookups are accomplished in $O(\log n)$ for a network including n participants.

Evaluation Testbed and Methodology.

The purpose of this evaluation is to assess experimentally the performance of the framework's image search capability for rudimentary visual localization using real-world data. The application specifically targeted is visual roadway navigation. This

involves the use of a vehicle's dashboard camera to capture roadway images for visual localization. A real-world image dataset for the evaluation was constructed using a testbed consisting of a Raspberry Pi-based collection platform, shown in Figure 47. The platform includes a Raspberry Pi 2 single board computer, camera module, WiFi module, and an EMLID NAVIO2 GPS and INS expansion board. All components are mounted to an aluminum plate which attaches to the inside of a vehicle windshield with suction cups, as shown in Figure 48.

The dashboard image collector is used to build a set of image repositories, each simulating the collection of localized images stored on a participating PeerAppear client. An evaluation track is also recorded which covers an area overlapping the area covered by the simulated clients. This evaluation track is then used to form visual search queries. To test the precision, Equation 19, and recall, Equation 20, of the BoVW similarity search, an exhaustive evaluation was completed using all images in the overall combined test repository. Each image was submitted as a search query returning cosine distance for their weighted BoVW vector representations. The images were then evaluated using brute-force matching of features and the epipolar constraint. Results with more than 15 matched feature pairs, as validated by the epipolar constraint, were considered to be true matches. The value of 15 matches was selected based on experience working with this dataset, however it is likely that some invalid matched image pairs could produce in excess of 15 validated feature matches.

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} \tag{19}$$

$$recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|} \tag{20}$$

The image in the set of results with the greatest number of validated feature

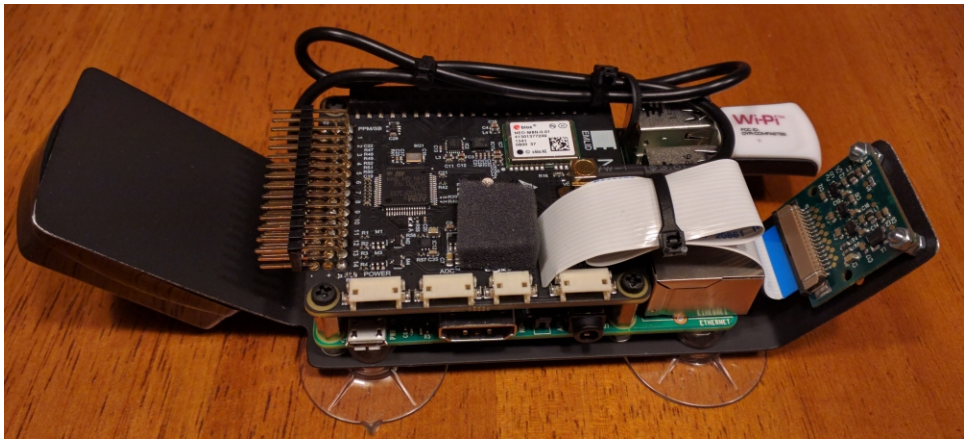


Figure 47. Raspberry Pi-based hardware used to collect image set for experimental evaluation.

matches to the query image is considered to be the best match for localization. The position of the query image is then inferred to be the location of this best match. The localization error is determined as the Euclidean distance between GPS annotations which are recorded for both images.

Evaluation Dataset.

A dataset consisting of 3,730 georeferenced images was captured on roads and parking lots in Area B of Wright-Patterson Air Force Base. Images were captured at 1Hz while driving in an attempt to achieve good coverage for most major roads. The collection is subdivided into individual tracks, shown in various colors in Figure 49. Each colored track represents the individual repository for 1 of 21 total simulated PeerAppear clients.

Two specific test tracks, a large loop around the base and a smaller loop around several buildings, were identified for this evaluation. Each loop was driven with datasets saved specifically for the purposes of conducting this evaluation. To ensure that sufficient numbers of images would be discoverable for each track, both were covered an additional 4 times. All travel was conducted in a counter-clockwise direction

Figure 48. Test hardware mounted to vehicle windshield for collecting evaluation dataset as dashboard camera.

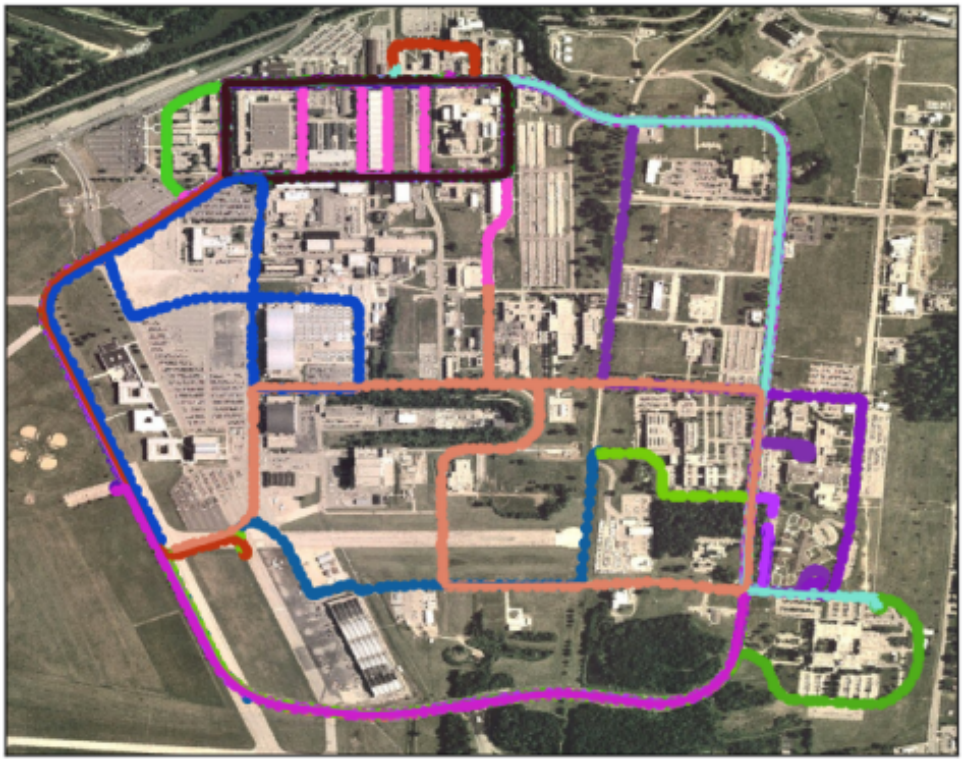


Figure 49. A map of image collection tracks colored by track.

to maximize opportunities for image matching.

Results.

Precision and recall results for both the urban small loop and rural large loop are depicted in Tables 7 and 8. The reported numbers indicate the mean result for each of the images representing the respective test track.

The large loop's true track and and associated position inference track are shown in

Table 7. Mean image search precision for complete dataset, small loop track, and large loop track.

Search Precision			
Top Results	All	Small Loop	Large Loop
10	0.416	0.519	0.413
20	0.303	0.402	0.313
30	0.249	0.349	0.257
40	0.215	0.316	0.225

Table 8. Mean image search recall for complete dataset, small loop track, and large loop track.

	Search Recall		
Top Results	All	Small Loop	Large Loop
10	0.114	0.103	0.095
20	0.165	0.160	0.143
30	0.204	0.208	0.177
40	0.234	0.251	0.207

Figure 50. All matched images were coincident with the true track, despite numerous branching opportunities shown in Figure 49. While the inferences were coincident, they regularly occurred ahead of or behind the true position. The absolute error in meters for the track is shown in Figure 51. Over the course of the track, the image search returned inferences 87 percent of the time and achieved a mean absolute position error of 11.2 meters.

The larger track included a mix of scenery types including areas dominated by dense foliage, sprawling fields, and structures. While a mix of scenery types might be more realistic for real-world use, the second smaller track was intended to test performance in more urban environments. This track's imagery therefore predominantly includes buildings and other man-made structures. The small loop's true track and associated position inference track are shown in Figure 52. Again, the inference track is coincident with the true track and does not branch from it. The error plot for this loop, shown in Figure 53, demonstrates significantly lower error than the large loop. The small loop's 93 percent inference availability and 6.2 meter mean absolute position error also best those of the large loop.

While the experimental evaluations were performed offline, a real-time visualization of the experiment's execution allows search results to be depicted for each inference made along the test tracks. This visualization is depicted in Figure 54.

Considering the large number of images with visually similar content from our

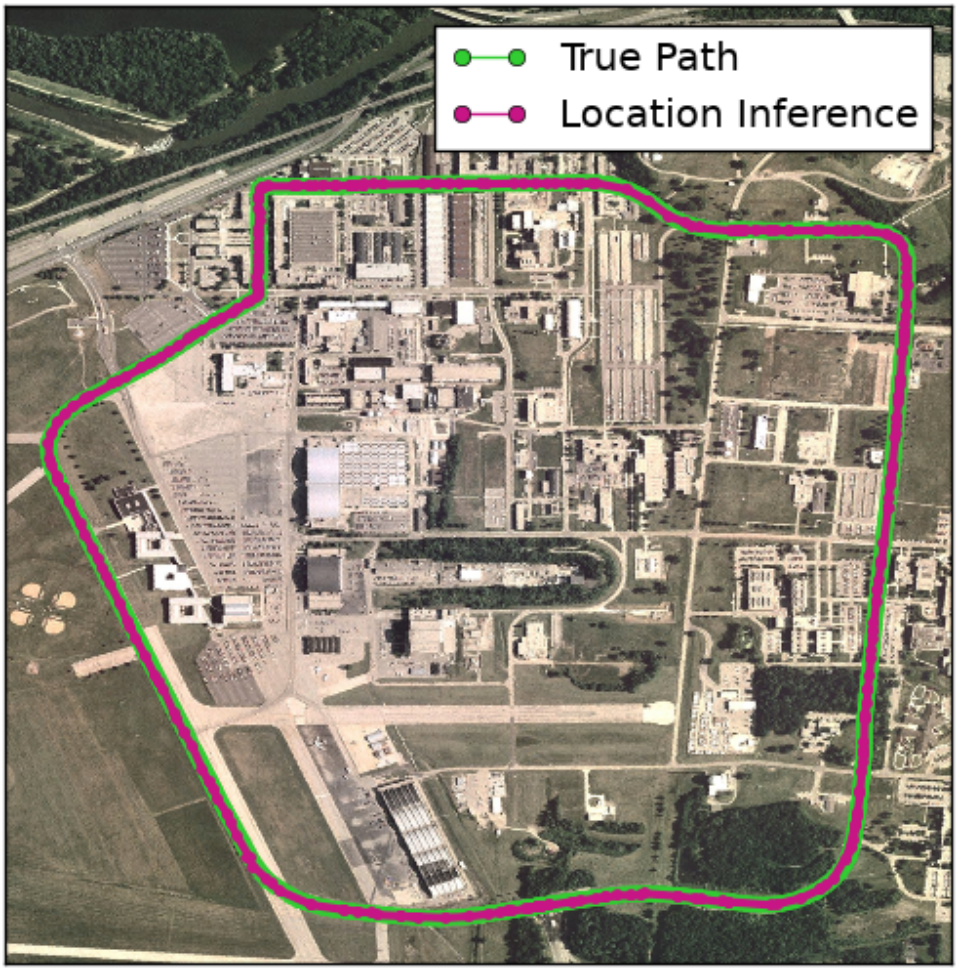


Figure 50. Test track and path of associated location inferences for large loop.

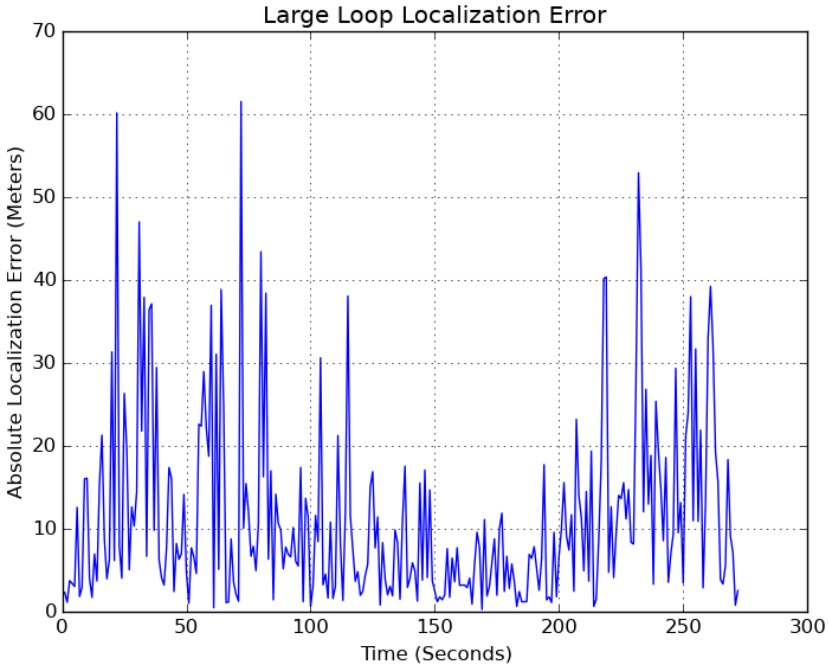


Figure 51. Absolute error for position estimates generated during large loop test execution.

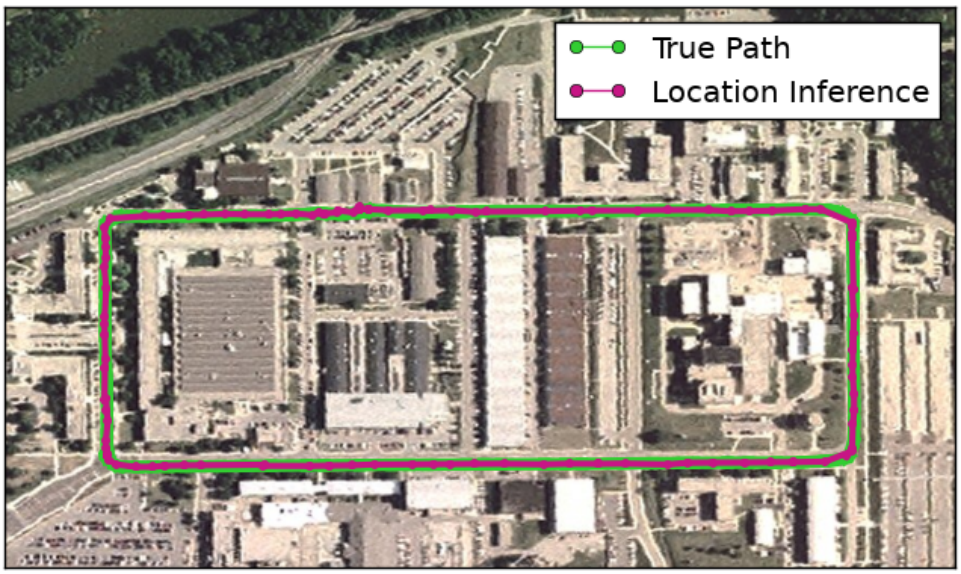


Figure 52. Test track and path of associated location inferences for small loop.

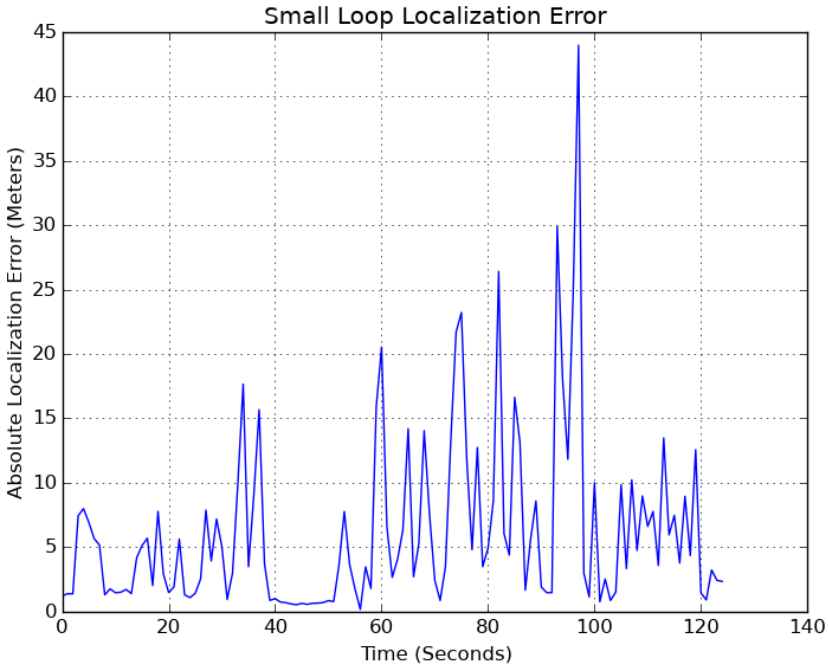


Figure 53. Absolute error for position estimates generated during small loop test execution.

small test area, we consider these results to be quite promising. An example image query from one of the test tracks is shown along with its results in Figure 55. In this example all 5 results returned are positive matches for the original image query.

5.3 Collaborative Visual Search Using PeerAppear's Distributed Index

Previously in this work an evaluation of PeerAppear's network performance (Section 3.4) was presented and later an evaluation of PeerAppear's visual search performance (Section 5.2) was presented. In this section the PeerAppear Simulation Engine (PASE) is integrated with BoVW image search to enable evaluation of the framework as a whole. This evaluation again uses real-world image collections to simulate the local repositories of participating clients.

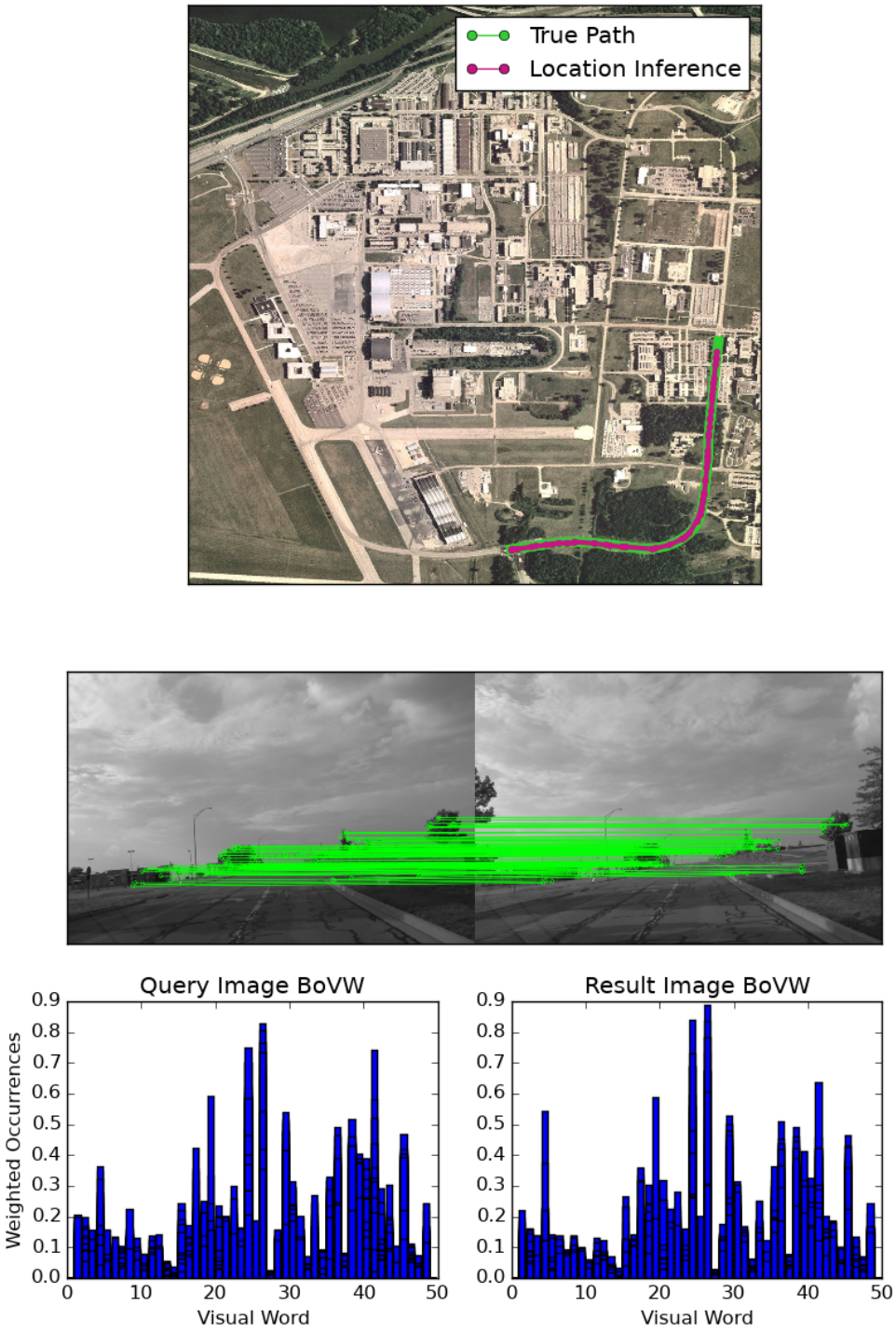


Figure 54. Experiment interface depicting search track including query and result images along with associated weighted BoVW image representations.

Figure 55. Results of image similarity search, top left is query image.

Integration with the PeerAppear Simulation Engine.

The PeerAppear Simulation Engine (PASE) was previously presented in Section 3.4 where it was used to conduct experiments evaluating the performance of framework's distributed index and for tuning network parameters. In the previous experiment PASE constructed simulated datasets consisting of client and data locations. The locations were generated randomly based on Gaussian distributions and pre-set parameters. For this evaluation, PASE was modified such that participating clients in the simulation are loaded from pre-captured and recorded datasets. The objectives of this experiment include a demonstration of real-time localization and the reporting of recall for search.

For each simulated client the dataset includes a repository of roadway images, captured using the device shown in Figure 48, and a database containing location, BoVW signature, and other various annotations for each image. PASE populates the network by sequentially loading each client's database and bootstrapping them into

the network. Per Section 3.3, the network address for each client is the 64-bit level 30 S2 address of their start location. After the bootstrapping process is complete, a covering of the locations where each of the client's images were captured using level 16 S2 cells is created and published.

Once all clients are loaded the evaluation begins. A single track, which was pre-selected and withheld from the full dataset, is used for the search evaluation. The simulated client for this track is bootstrapped into the network, however their local repository is not published. For each image in the evaluation track, a covering of the circular position estimate for the location where the image was captured is created. The cell addresses from this covering are then used to perform location-based lookups of the PeerAppear distributed index.

For each simulated client identified by the index lookup as having published data in the specified search area, an image search is accomplished using the BoVW image signature for the query image to find highly ranked matches in the simulated client's repository. The top matches are validated and the results are recorded. Results for the full trial are compiled and recall is plotted.

Evaluation Dataset.

The results from the experiment described in Section 5.2 showed that the BoVW search process performed better for the small track which consisted mostly of urban-type imagery. By reviewing the in-process experiment visualizations, it is apparent that man-made structures produce SIFT features more reliably and offer better repeatability for descriptor matches. Urban type environments have also been identified as being ideal candidates for GPS alternative navigation because the poor reception caused by urban canyons makes acquisition of GPS position fixes difficult.

For these reasons, the dataset constructed for this evaluation was collected from

the downtown area in Dayton Ohio. As with the previous experiment, data was collected using the dashboard camera platform shown in Section 5.2. Unlike the previous experiment, which did not contain significant dynamism, this dataset was captured in the middle of the day and included significant vehicle and pedestrian traffic. The data was captured over a 4-block by 4-block area. Many of the streets traversed were one-way, benefiting the search process by eliminating the possibility of having images captured in the same location but from different directions.

The full dataset consists of 13 individual tracks and a total of 2,322 images. Images were captured at a nominal rate of 1Hz, however the exact interval varies due to increased processing times on the capture device for images with greater numbers of features. A depiction of the dataset is shown in Figure 56. In the figure each track is represented using a unique color.

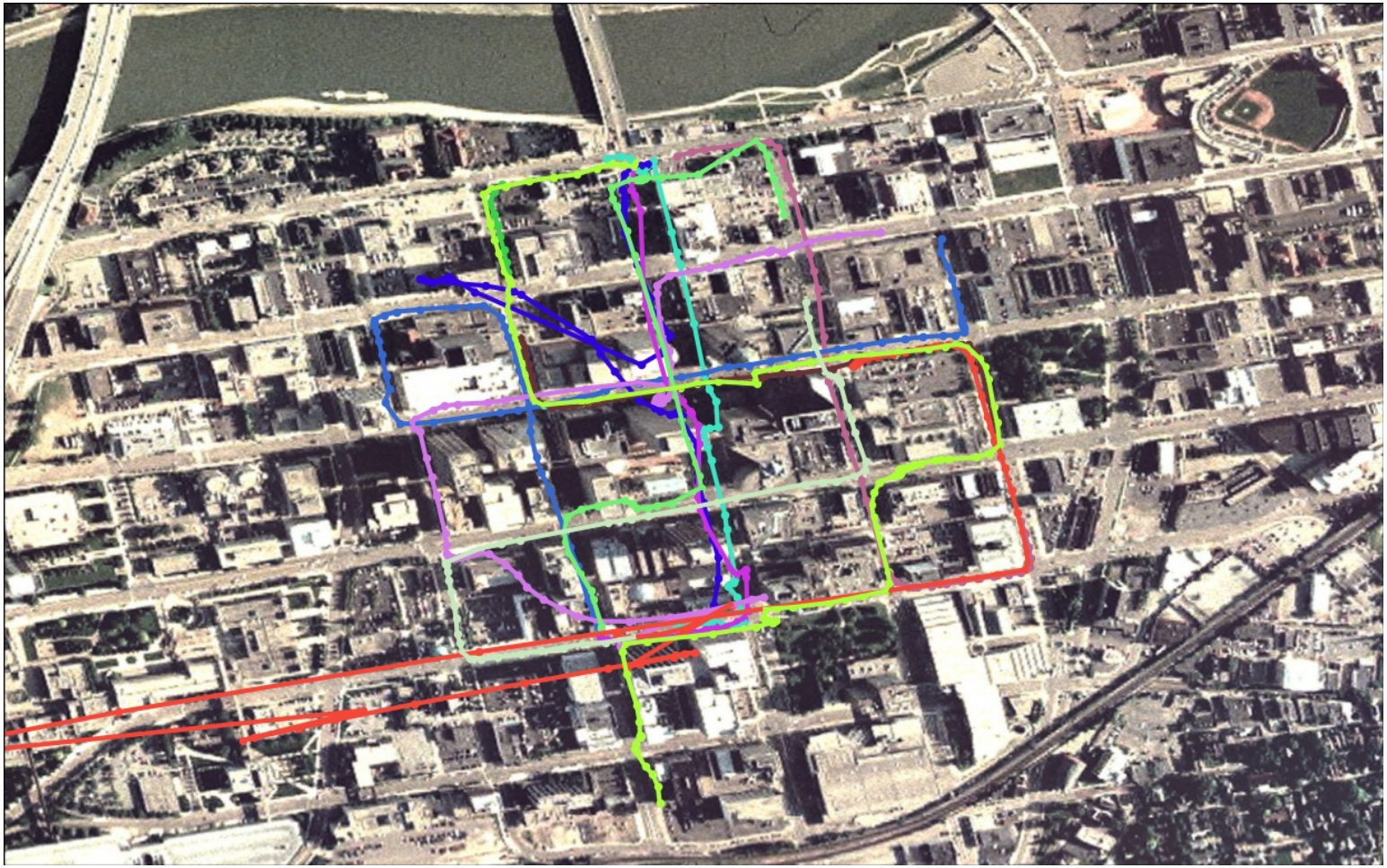


Figure 56. A map showing the tracks traversed by each simulated client while capturing images and data used to populate the client's local repository.

Results.

For each query image in the evaluation a chart was produced showing the location of the query image (magenta circle), the geospatial search filter (blue cells), query image, query image BoVW vector, best result image, and best result image BoVW vector. Feature matches between the query image and result image are depicted using green lines. An example frame from the evaluation is shown in Figure 57.

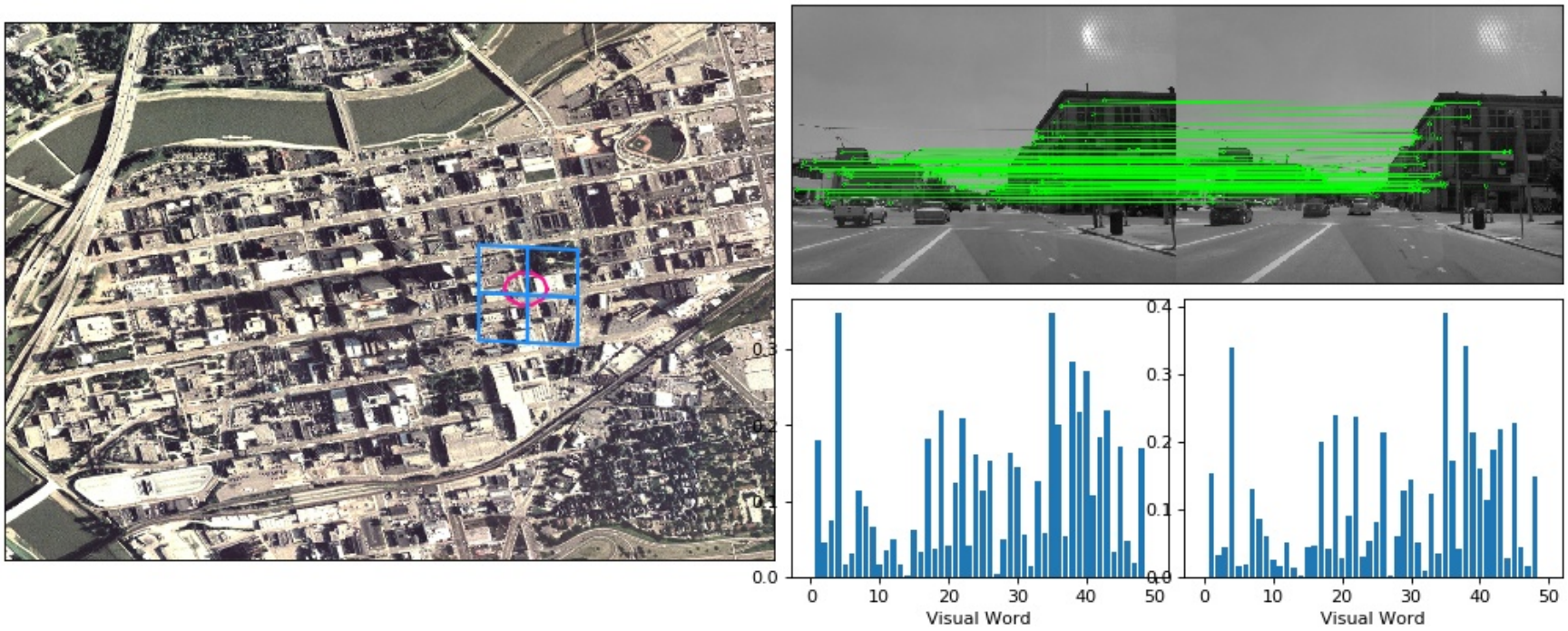


Figure 57. A single frame from the evaluation showing the location of the query image (magenta circle), search filter (blue cells), query image and result image with matched features, and query and result BoVW vectors.

The evaluation track consisted of 55 query images. The result charts for each of the 55 queries was evaluated and matches were manually validated. While each of the 55 queries produced a result that met the requirements for our image match validation using the epipolar constraint, 1 of the matches was found to be incorrect. This match occurred because a honeycomb pattern on the test vehicle's dashboard caused a reflection on the windshield in certain lighting conditions, leading to a matching of features from the reflection. The remaining 54 matches were valid, giving a localization accuracy of 98%.

Recall was calculated for each index lookup operation and for the top 1, 5, 10, and 20 results returned for each query. Both the individual query recall values as well as mean recall are plotted in Figure 58. For the top 1, 5, 10, and 20 a mean recall of 20%, 50%, 70%, and 92% was achieved. Index operations achieved a consistent 100% recall.

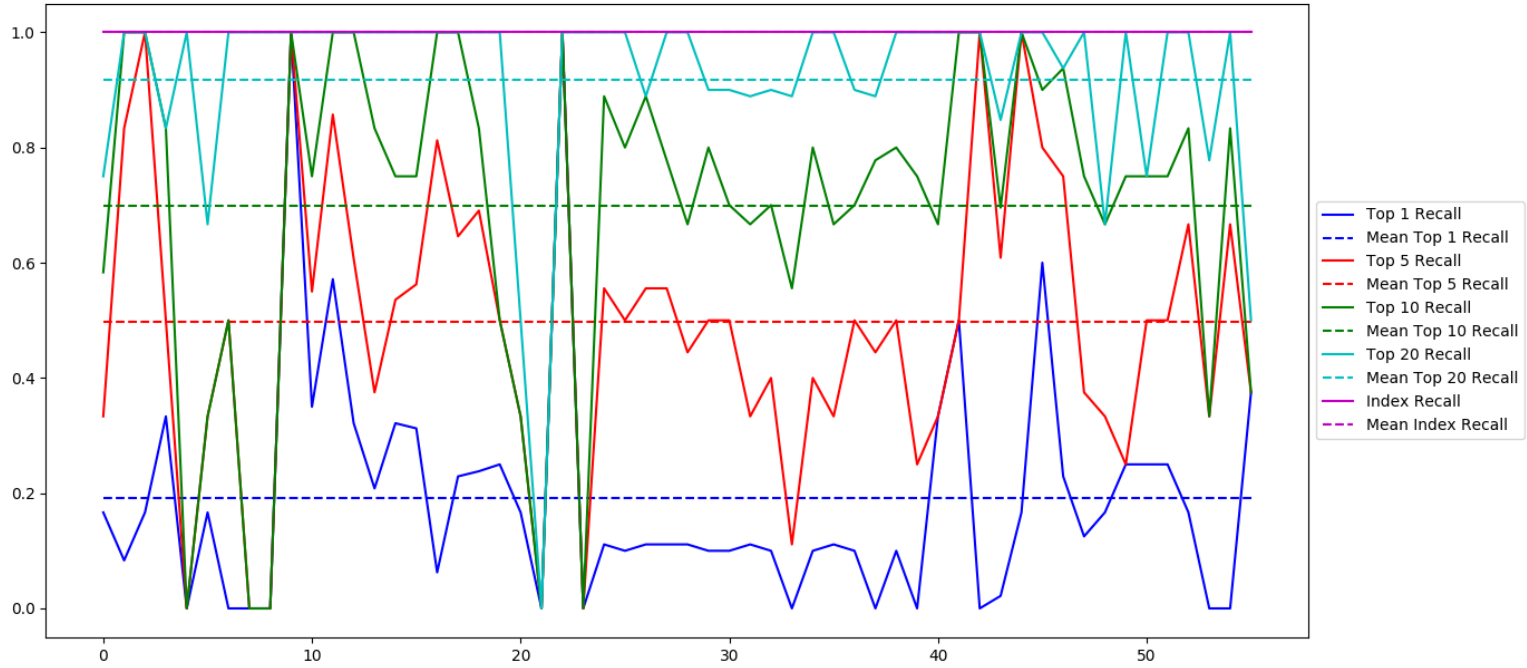


Figure 58. Recall of individual search queries for the evaluation track and mean recall for the top 1,5,10,20 search results.

The results of this evaluation demonstrate that the distributed index works well for our target application. While the number of clients simulated was only 13, we can see in the previous experiment (Section 3.4) that the performance is consistently high as the network scales to large numbers of nodes. With the index achieving a 100% recall, the remaining recall deficiency can be attributed to the BoVW image search methodology. While achieving better recall using visual methods will be an ongoing goal, the improvement in recall from the previous experiment (Table 8) demonstrates the positive effect location-based filtering can have for localized CBIR.

VI. Conclusion

In this chapter the dissertation's final conclusions are presented. Section 6.1 contains an overview of the work which summarizes the contributions of each chapter. The section ends with a statement of conclusion based on the results of the individual research thrusts. In Section 6.2, potential applications of the framework and the implications for its use are discussed. Finally, in Section 6.3 areas of future exploration and ideas for extending the work based on lessons learned are presented.

6.1 Summary and Conclusions

In this work a novel location-aware middleware framework which supports collaborative visual information discovery and retrieval is presented and evaluated. The problem statement and motivational use-cases were introduced in Chapter I. These were further exposed in Chapter II where the problem and motivation are described in greater detail along with a solution approach developed through lessons learned in previous work. The chapter lays out requirements for a solution in the form of a middleware framework and concludes with 3 primary research thrusts which comprise the work's 3 main chapters.

The first research thrust was presented in Chapter III. This thrust covers the framework's core infrastructure, consisting primarily of a distributed geospatial index. An overview and use-cases for the index were presented followed by detailed descriptions of the index's core addressing scheme and distributed network topology. Finally, the infrastructure was evaluated using the PeerAppear Simulation Engine (PASE), a custom-built discrete event simulation tool. The evaluation demonstrates high recall, efficiency and scalability of PeerAppear's network topology and indexing operations. The evaluation also shows that mixed cell level operations can be

accommodated by the framework with high recall.

The second research thrust was presented in Chapter IV. This thrust covers computational support methods which enable the distributed index to support targeted applications. In Section 3.2, the first method was presented which enabled content-based image retrieval (CBIR) using a bag of visual words (BoVW) approach. The approach makes use of text-search paradigms to efficiently characterize image content and enable search. The second method was presented in Section 4.4 and covers the representation of position using S2 cells. A novel filtering scheme was presented which makes use of point and path viewshed to produce visibility filters. Finally, Section 4.3 presented a novel method for scoring images in local repositories based on the area of visibility for content in the image.

The third and final research thrust was presented in Chapter V. This research included 2 application-based evaluations of the framework for collaborative visual localization. The first evaluation, presented in Section 5.2, used a real-world dataset to demonstrate visual localization with only the framework's visual search methods. The second evaluation, presented in Section 5.3, integrated the visual search methods with PASE to demonstrate expected overall system performance when visual search methods were combined with index operations. Again the experiment made use of real-world datasets.

The results of the evaluation in Chapter III demonstrate that PeerAppear's geospatial addressing and indexing methods provide for efficient, scalable, and elastic global operations. The computational methods presented in Chapter IV provide mechanisms for enabling anticipated applications for the framework. Finally, evaluations using real-world data presented in Chapter V demonstrate that the framework meets the needs for the primary motivational application of visual localization. Additionally, the fulfillment of the requirements set forth in Chapter II is summarized in Table 9.

Table 9. Fulfillment of framework requirements.

1	**Decentralized Operations**
The framework's RPCs enable the index of volunteered content to be constructed and maintained through direct peer-to-peer transactions. The architecture's only centralized component is a peer discovery service which aids first-time users find other network participants which can be used to bootstrap into the network.	
2	**Local Storage of Sensed Geospatial Data**
The framework includes provisions for the local storage and processing of various types of geospatial data. Data is not aggregated or replicated by any of PeerAppear's native processes.	
3	**Distributed Indexing for Location-Based Content**
The framework enables content owned by participants to be indexed using a location-based addressing scheme. This enables content to be discovered by peers through location-based search.	
4	**Uniform Elastic Global Addressing and Filtering**
PeerAppear uses an addressing scheme based on the S2 spherical geometry library. The addressing enables elasticity through hierarchical cell-based representations and uniform global coverage through quadrialateralized mapping.	
5	**Fully Retrievable Location-Based Search**
The framework demonstrates efficient near full retrievability in a large-scale simulation-based evaluation for minimum specified system parameters. For the *adaptive-k* method described in Chapter III, full retrievability is achieved with a penalty to search complexity.	
6	**Efficient Location-Based and Visual Search**
In Chapter III the efficiency of the framework's index operations is demonstrated.,Chapter IV presents efficient methods for geospatial filtering and an efficient visual search paradigm which is compatible with the framework's architecture.	
7	**Scalable Operations**
Through the large-scale simulation-based network evaluation presented in Chapter III, the complexity of index operations was shown to grow logarithmically in response to increases in the number of network participants.	
8	**Support for Dynamic World**
The architecture of PeerAppear places no limits on quantity or frequency for,publishing of new content to the distributed index and the expiry of older data occurs through attrition.	
9	**Equitable Distribution of Indexing Effort**
The PeerAppear framework is supported by a symmetric peer-to-peer topology.,There are no super peers. In Chapter III routing table composition is shown to be a function of the XOR distances between peers' addresses which correlates well with great circle distance. Non-uniform distribution of keys is demonstratably minimized by using smaller cell sizes when publishing summaries of repository content.	

These results lead us to conclude that PeerAppear provides an efficient and scalable framework enabling the construction and maintenance of world-scale maps, models, and repositories consisting of sensed geospatial data through a fully decentralized peer-to-peer approach.

6.2 Potential Applications

Like many online services which rely on user-contributed content, the PeerAppear framework will not be useful until it achieves an adoption level that provides enough content to satisfy some minimum capability level for its users. In the near-term, the greatest potential for growing the PeerAppear framework is likely through dashboard camera integration. Initially, the most likely dashcam platform to target would be homemade systems similar to our Raspberry Pi-based evaluation platform, as commercially produced dashboard cameras currently do not have the processing power required.

To attract users to the framework the first priority should be enabling easy and transparent access to the collective data available on the network. This can be accomplished by implementing a map-based browser with search and filtering utility. An early application of such a browser would be to provide a Street View type of capability, albeit with more up-to-date imagery. The search and filtering utility would also enable users to leverage the network for distributed surveillance.

Crawling the network to aggregate images for offline processing also presents opportunities. With sufficient adoption by contributors a network crawler could collect and organize images based on time and location in order to support 4D Cities and similar efforts, thereby enabling the automated construction of physical city models which show changes over time. With sufficient quantities of images physical changes in the environment can be tracked and modeled using a time scale with much greater

resolution.

Finally, the evaluations conducted in this work demonstrate that the framework provides an effective means for enabling the motivational visual localization and collaborative loop closure applications. As vehicles continue to gain environmental awareness through the inclusion of cameras and other sensors, integrating PeerAppear into the vehicle could enable access to visual localization for navigation of human operators and self-driving systems. If camera-equipped wearables, such as Google's now somewhat defunct Glass project, become popular this capability can even be extended for indoor and outdoor pedestrian navigation.

6.3 Future Work

Several activities have been identified as near-term future work in the continuance of this research. The first is the development of a full stand-alone PeerAppear client which operates on commodity development hardware, such as the Raspberry Pi or nVidia Tegra platform. This activity is key to building up a base of users who will contribute data to support future development activities.

The second activity is the development of a PeerAppear network browser. The browser should provide for location search and filtering using a map interface. By enabling access to network data new users will be encouraged to participate and contribute by installing PeerAppear-equipped systems on their vehicles and fixed locations.

The third is large-scale evaluation to enable continued iterative development and tuning of parameters to improve network capabilities and efficiency. This evaluation should make use of actual standalone clients so that the effects of real-world latency and other network effects can be observed. Part of this effort should be an adaption of the PeerAppear RPCs to enable support for operations using multiple addresses,

so that operations based on S2 filters can be accomplished more efficiently.

Finally, the framework's BoVW search capability could be improved upon. While the best image classification approaches now use convolutional neural networks (CNNs), the approach has yet to make its way into the the image search domain. Once better capabilities for visual search using CNN-derived features become available the integration of them into PeerAppear should be prioritized.

6.4 Thesis Summary

The dissertation made the following contributions.

- It exposed the challenges of scale and dynamism for the problem of world model construction and identified a feasible decentralized and distributed solution. (Chapter II)
- It proposed PeerAppear, a middleware framework for the collaborative construction of world models supported by a novel distributed geographic index inspired by the Kademlia DHT and using a hierarchical cell-based geospatial addressing scheme based on the S2 spherical geometry library. (Chapter III)
- It presented computational support methods enabling image ranking through area of visibility scoring, geospatial filtering using S2 cell masks with S2Trie pruning, and image-based search using *tf-idf* weighted BoVW search within the proposed framework. (Chapter IV)
- It demonstrated the framework's successful use for decentralized, distributed, and collaborative practical visual applications including visual localization and location-based image search using simulated clients generated from real-world datasets within the PeerAppear Simulation Engine. (Chapter V)

In this work the PeerAppear middleware framework was presented. PeerAppear enables effective self-building and self updating world-scale models comprised of user-contributed and user-annotated visual records to be created in an efficient and scalable manner. Through a novel distributed geographic index and supportive processing strategy, PeerAppear realizes a system capable of leveraging the aggregate reach of distributed sensors to enable the construction of world scale models which support dynamic environments, are scalable, and globally available.

Appendix A. Publication and Presentation Summary

1.1 CTS 2016

Title: PeerAppear: A Location-Aware Framework for Extensible Image Annotation and Peer-to-Peer Discovery

Authors: A. Compton, J. Pecarina, N. Lesch

Status: Published In Conference Proceedings

Abstract:

This paper addresses the problem of building and maintaining image repositories which form the basis for world-scale visual models. The availability of these models enable capabilities such as visual localization, persistent surveillance, and structure from motion. We approach this problem through the creation of PeerAppear, a location-aware framework for extensible image annotation and peer-to-peer dissemination. Due to the dynamic nature of the world, solutions to this problem generally require significant effort to be spent on mapping. PeerAppear enables a decentralized solution through the implementation of a peer-to-peer middleware framework which automates the indexing and sharing of visual information extracted from images in a user's collection. The PeerAppear network achieves scale through a Bittorrent style overlay network, which indexes the locations of user's image collections using a hierarchical geographic segmentation scheme in a distributed hash table. By enabling a distributed and collaborative solution, network participants are able to provide for frequent re-mapping and a search capability which leverages efficient and effective visual representations. The framework was implemented in Python and evaluated using Raspberry Pi computers for data collection and a workstation computer to simulate network participants. This evaluation showed that the framework was able to provide an effective image search capability and shows promise for large-scale operations.

1.2 GNSS 2016

Title: PeerAppear: A P2P Framework for Collaborative Visual Localization

Authors: A. Compton, J. Pecarina

Status: Published In Conference Proceedings

Abstract:

The ubiquity of GPS receivers in modern devices has given rise to the expectation for always-available real-time precision localization. While this expectation can often be met in rural outdoor environments, it is generally not achievable indoors or when line-of-sight to the horizon is blocked by large urban or geographic structures. The shortcomings of GPS have given rise to a myriad of alternative methods for precision navigation, many of which can directly augment GPS. One of the most promising methods builds upon recent advances in computer vision technologies to determine location visually by matching a captured image taken at an unknown location against a database of localized images. The accuracy and scale of these efforts is often determined by the size, scale, and granularity of the image database. Most research efforts focus their database building efforts on small-scale purpose-built collections or on the aggregation of localized crowd-sourced images from services such as Flickr and Instagram. While small-scale purpose built collections are often sufficient for the experimental validation of localization methods, they do not provide for a scalable localization capability. Databases built from crowd-sourced images generally do possess the necessary scale, but lack in quality and granularity because images tend to be densely clustered near landmarks and points of interest. To overcome the challenges encountered when constructing datasets for visual localization, we present PeerAppear, a middleware framework for the extraction and dissemination of visually derived spatial data. PeerAppear enables collaborative visual information discovery through the implementation of a peer-to-peer middleware framework which automates the in-

dexing and sharing of visual information extracted from images in a user's collection. Evaluations of the framework's theoretical complexity and experimental performance are presented, demonstrating PeerAppear's feasibility for supporting large-scale, decentralized collaborative visual localization.

1.3 CogSIMA 2017

Title: Image Retrieval for Visual Understanding in Dynamic and Sensor Rich Environments

Authors: N. Lesch, A. Compton, J. Pecarina, M. Ippolito, D. Hodson

Status: Published In Conference Proceedings

Abstract:

Vision is vital to decision making, as humans naturally trust their eyes to enhance situation awareness. Yet the modern age has overwhelmed humans with massive amounts of visual information, which is problematic in time sensitive and mission critical situations, such as emergency management and disaster response. More efficient search and retrieval systems address some of these issues, which is why many seek to develop and extend Content Based Image Retrieval (CBIR) techniques to support situational awareness in a more autonomous fashion. However, there is currently no adequate system for CBIR to support situational awareness in dynamic and sensor rich environments. This research proposes an extensible framework for CBIR to support a holistic understanding of the environment through the automated search and retrieval of relevant images and the context of their capture. This constitutes assisted CBIR as embodied in the multi-sensor assisted CBIR system (MSACS). We design the MSACS framework and implement the core CBIR system of MSACS using the state of the art Bag of Visual Words paradigm. The system is evaluated using a dataset of GPS tagged images to show favorable precision and recall of spatially related im-

ages. Applications for localization and search for Wi-Fi access points demonstrate improved situational awareness using the system. Assisted CBIR could enable vision based understanding of an environment to ease the burdens of information overload and increase human confidence in autonomous systems.

1.4 JNC 2017

Title: An Evaluation of the PeerAppear Framework for Large-Scale Collaborative Image-Based Localization

Authors: A. Compton, J. Pecarina

Status: Presented At Conference

Abstract:

Due to the ubiquity of GPS and the prevalence of low-cost receiver modules, the availability of real-time position information has become an expectation on devices ranging from watches and phones to cars, aircraft, and ships. This position information has proven to be extremely valuable for vehicle operators and is essential for most unmanned and autonomous systems. Yet while GPS has demonstrated excellent accuracy and availability in most open outdoor environments, it requires direct line of sight to the sky and is often unable to provide for accurate localization in many environments autonomous systems are expected to operate, including dense urban areas, within tunnels, and indoors. Alternative navigation methods have recently been demonstrated which make use of visual odometry and image-based localization to infer relative and absolute position. Visual odometry offers relatively accurate position inferences but is prone to drift. This drift can be mitigated through the use of efficient image search to identify loop closure and bundle adjustment opportunities. The visual mapping of an environment for this purpose effectively forms a visual memory with artificial semantic linkages to location, thereby enabling landmark-based

visual localization analogous to visual localization performed by humans. Leveraging both the ability to construct localized visual memories and the ubiquity of global communications infrastructure, we developed PeerAppear, a framework for collaborative visual localization. PeerAppear provides for the formation of a collective visual memory by enabling clients to publish location-based summaries of their local image repositories to a peer-to-peer distributed index. This distributed index, inspired by the Kademlia distributed hash table, uses Googles S2 spherical geospatial addressing scheme to provide hierarchical location-based addressing for published content. Unlike Kademlias SHA-1 derived addresses, which do not indicate data relationships, the spatial relationships of addresses used by PeerAppear are represented. The addressing therefore enables search filters, comprised of S2 cell addresses, to be used to identify the clients participating in the network which are most likely to possess imagery for a given geographic area. Visual search queries including compact bag of visual words image signatures can then submitted to clients identified in the previous step for distributed image search in an attempt to provide a location inference based on the image signature. By enabling the creation and indexing of searchable decentralized visual repositories, the PeerAppear framework avoids many of the pitfalls and shortcomings of centralized image aggregation approaches. These include inability to scale, single point of failure, and accumulation of perishable and highly dynamic visual content. The performance of the PeerAppear framework for large-scale collaborative visual localization is addressed through a rigorous experimental evaluation using purpose-built and open source datasets and image repositories. The evaluation framework consists of a simulation engine which enables both simulated and real-world clients to interact, thereby supporting real-time operations for large numbers clients without the physical footprint and overhead of traditional large-scale real-world evaluations. This evaluation is expected to demonstrate the ability of the

network to provide for a scalable, robust, and efficient collaborative visual localization capability. In addition, the performance of the network is presented in terms of operational complexity and overhead for various numbers of clients, network churn rates, and tunable network parameters.

1.5 FGCS Journal 2018

Title: PeerAppear: A Distributed Geospatial Index Supporting Collaborative World Model Construction and Maintenance

Authors: A. Compton, J. Pecarina, A. Lin, K. Hopkinson

Status: With Journal (Under Review)

Abstract:

This paper addresses the problem of scalable location-aware distributed indexing to enable the leveraging of collaborative effort for the construction and maintenance of world-scale maps and models. These maps and models support numerous activities including navigation, visual localization, persistent surveillance, and hazard or disaster detection. We approach a solution through the creation of PeerAppear, a location-aware framework for peer-to-peer indexing, search and retrieval. Due to the dynamic nature of the world, the problem of constructing and maintaining relevant world-scale models generally requires significant effort to be spent on mapping. PeerAppear offers a decentralized solution which enables the leveraging of collaborative effort through the implementation of a peer-to-peer middleware framework which automates the indexing and sharing of sensed geospatial information captured and stored in the local repositories of participants. The PeerAppear network achieves scale through a Kademlia-like overlay network which indexes data based on location by adapting Google's S2 hierarchical geographic segmentation scheme to a globally addressable distributed geographic table. Our communications primitives allow search

queries to be formed and executed, enabling the discovery of information published in a specified geographic area. An evaluation of the framework is presented demonstrating excellent retrievability of published data, logarithmic efficiency and global scalability.

1.6 ION Journal 2018

Title: Leveraging Surface-Level Collaborative Visual Mapping for Navigation

Status: In Draft

Abstract:

Recent advances in computer vision technology have enabled the development of methods for highly accurate visual navigation. These methods have demonstrated the ability to supplement and often supplant GPS-based localization in GPS-denied environments such as urban canyons and inside buildings. While these visual localization techniques have thus far primarily been solitary efforts, requiring a single agent to map and localize as it moves through the environment, the potential exists for collaborative visual mapping efforts which can extend an agent's visual familiarity globally. However, the realization of such a system has yet to be achieved. In this paper, we present PeerAppear, a middleware framework which enables collaborative visual mapping through a location-aware distributed geographic index. Participants in the PeerAppear network construct local visual mappings and publish summaries of their data to the distributed index enabled by PeerAppear's novel network topology. A technical description of the PeerAppear framework is presented followed by an evaluation of the framework using hardware-in-the-loop simulation to assess the framework's suitability for global-scale surface-level visual mapping and navigation.

Bibliography

1. Anderson, M. *Technology Device Ownership: 2015*, [Retrieved 16-February-2017]. http://www.pewinternet.org/2015/10/29/technology-device-ownership-2015/.

2. Angeli, A., and Filliat, D. Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics 24*, 5 (2008), 1027–1037.

3. Anguelov, D., Dulong, C., Filip, D., Frueh, C., Lafon, S., Lyon, R., Ogale, A., Vincent, L., and Weaver, J. Google street view: Capturing the world at street level. *Computer 43*, 6 (2010), 32–38.

4. Araújo, F., and Rodrigues, L. Geopeer: A location-aware peer-to-peer system. In *Network Computing and Applications, 2004.(NCA 2004). Proceedings. Third IEEE International Symposium on* (2004), IEEE, pp. 39–46.

5. Ariyoshi, Y., Kamahara, J., Tanaka, N., Hirayama, K., Nagamatsu, T., and Teranishi, Y. Location-dependent content-based image retrieval system based on a P2P mobile agent framework. *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, March (2013), 72–77.

6. Arnold, T. An entropy maximizing geohash for distributed spatiotemporal database indexing. *arXiv preprint arXiv:1506.05158* (2015).

7. Asaduzzaman, S., and Bochmann, G. v. Geop2p: an adaptive and fault-tolerant peer-to-peer overlay for location based search. In *International Conference on Distributed Computing Systems (ICDCS)* (2009).

8. a.W.M. Smeulders, Worring, M., Santini, S., Gupta, a., and Jain, R. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 12 (2000), 1349–1380.

9. Bakus, J., Hussin, M., and Kamel, M. A som-based document clustering using phrases. In *Neural Information Processing, 2002. ICONIP'02. Proceedings of the 9th International Conference on* (2002), vol. 5, IEEE, pp. 2212–2216.

10. Basu, S., Basu, S., Valladares, C. E., Yeh, H.-C., Su, S.-Y., MacKenzie, E., Sultan, P. J., Aarons, J., Rich, F. J., Doherty, P., Groves, K. M., and Bullett, T. W. Ionospheric effects of major magnetic storms during the International Space Weather Period of September and October 1999: GPS observations, VHF/UHF scintillations, and in situ density structures at middle and equatorial latitudes. *Journal of Geophysical Research 106*, A12 (2001), 30389.

11. Bay, H., Tuytelaars, T., and Gool, L. V. SURF : Speeded Up Robust Features. *European Conference on Computer Vision* (2006), 404–417.

12. Bentley, D. s2-geometry-library a library for spherical math. `https://code.google.com/archive/p/s2-geometry-library/`. Accessed: 2016-05-05.

13. Biswas, J., and Veloso, M. Wifi localization and navigation for autonomous indoor mobile robots. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on* (2010), IEEE, pp. 4379–4384.

14. Botterill, T., Mills, S., and Green, R. Bag-of-Words-driven Single Camera Simultaneous Localisation and Mapping. *Journal of Field Robotics* (2011), 1–28.

15. Brambilla, G., Picone, M., Amoretti, M., and Zanichelli, F. An adaptive peer-to-peer overlay scheme for location-based services. *Proceedings - 2014 IEEE 13th International Symposium on Network Computing and Applications, NCA 2014* (2014), 181–188.

16. Calabretta, M. R., and Roukema, B. F. Mapping on the HEALPix grid. *Monthly Notices of the Royal Astronomical Society 381*, 2 (2007), 865–872.

17. Carneiro, G., Chan, A. B., Moreno, P. J., and Vasconcelos, N. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 3 (2007), 394–410.

18. Chan, F., and O'Neill, E. Feasibility study of a quadrilateralized spherical cube earth data base. Tech. rep., DTIC Document, 1975.

19. Clementini, E., Di Felice, P., and Van Oosterom, P. A small set of formal topological relationships suitable for end-user interaction. In *Advances in Spatial Databases* (1993), Springer, pp. 277–295.

20. Cohen, B. The bittorrent protocol specification. *http://www. bittorrent. org/beps/bep_0003.html* (2008).

21. Comanicu, D., and Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell. 24*, 5 (2002), 603–619.

22. Compton, A., and Pecarina, J. Peerappear: A p2p framework for collaborative visual localization. In *29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016), Proceedings of* (2016), ION, pp. 1080–1090.

23. Compton, A., Pecarina, J., and Lesch, N. Peerappear: A location-aware framework for extensible image annotation and peer-to-peer discovery. In *Collaboration Technologies and Systems (CTS), 2016 International Conference on* (2016), IEEE, pp. 225–232.

24. Compton, A., Pecarina, J., Raquet, J., and Jacques, D. Ranking Volunteered Images with Visual Viewshed Scoring for Landmark Recognition and Retrieval. Unpublished white paper, 2016.

25. Cornelis, N., and Van Gool, L. Fast scale invariant feature detection and matching on programmable graphics hardware. *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on* (2008), 1–8.

26. Cortés, R., Marin, O., Bonnaire, X., Arantes, L., and Sens, P. A scalable architecture for spatio-temporal range queries over big location data. In *Network Computing and Applications (NCA), 2015 IEEE 14th International Symposium on* (2015), IEEE, pp. 159–166.

27. Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. Visual categorization with bags of keypoints. In *Proceedings of the ECCV International Workshop on Statistical Learning in Computer Vision* (2004), pp. 59–74.

28. Cuenca-Acuna, F. M., Peery, C., Martin, R. P., and Nguyen, T. D. PlanetP: using gossiping to build content addressable peer-to-peer information sharing communities. *Proceedings of the IEEE International Symposium on High Performance Distributed Computing 2003-Janua* (2003), 236–246.

29. Cusano, C., Bicocca, M., and Bicocca, V. Image annotation using SVM. *Proceedings of SPIE*, 1 (2003), 330–338.

30. Deselaers, T., Pimenidis, L., and Ney, H. Bag-of-visual-words models for adult image classification and filtering. *Icpr* (2008), 1–4.

31. Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D., and Equitz, W. Efficient and effective Querying by Image Content. *Journal of Intelligent Information Systems 3*, 3-4 (1994), 231–262.

32. Farrell, M. S2-Geometry-Library a library for spherical geometry. `https://github.com/micolous/s2-geometry-library`, 2015.

33. Feng, S., Manmatha, R., and Lavrenko, V. Multiple bernoulli relevance models for image and video annotation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on* (2004), vol. 2, IEEE, pp. II–II.

34. Feng, Y., and Lapata, M. Visual Information in Semantic Representation. *In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, Los Angeles, California, June 2010*, June (2010), 91–99.

35. Filliat, D. A visual bag of words method for interactive qualitative localization and mapping. In *Proceedings - IEEE International Conference on Robotics and Automation* (2007), pp. 3921–3926.

36. Fischler, M. a., and Bolles, R. C. Random Sample Consensus: A Paradigm for Model Fitting with. *Communications of the ACM 24* (1981), 381–395.

37. Fisher, P. F. First experiments in viewshed uncertainty: The accuracy of the viewshed area. *Photogrammetric Engineering & Remote Sensing 57*, 10 (1991), 1321–1327.

38. Fisher, R. A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics 7*, 2 (1936), 179–188.

39. Frassl, M., Angermann, M., Lichtenstern, M., Robertson, P., Julian, B. J., and Doniec, M. Magnetic maps of indoor environments for precise localization of legged and non-legged locomotion. *IEEE International Conference on Intelligent Robots and Systems* (2013), 913–920.

40. Gálvez-López, D., and Tardós, J. D. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics 28*, 5 (2012), 1188–1197.

41. Google. S2-Geometry-Library-Java a java library for spherical geometry. `https://github.com/google/s2-geometry-library-java`, 2011.

42. Grant, A., Williams, P., Ward, N., and Basker, S. GPS Jamming and the Impact on Maritime Navigation. *The Journal of Navigation 62*, 02 (2009), 173–187.

43. GRASS Development Team. *Geographic Resources Analysis Support System (GRASS GIS) Software, Version 7.2*. Open Source Geospatial Foundation, 2017.

44. Gross, C., Stingl, D., Richerzhagen, B., Hemel, A., Steinmetz, R., and Hausheer, D. Geodemlia: A robust peer-to-peer overlay supporting location-based search. *2012 IEEE 12th International Conference on Peer-to-Peer Computing, P2P 2012* (2012), 25–36.

45. Guttman, A. R-trees: A Dynamic Index Structure for Spatial Searching. *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data - SIGMOD '84* (1984), 47–57.

46. Haklay, M., and Weber, P. OpenStreetMap: User-Generated Street Maps. Pervasive Computing. *IEEE Pervasive Computing 7*, 4 (2008), 12–18.

47. Hare, J., and Samangooei, S. Imageterrier: an extensible platform for scalable high-performance image retrieval. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval* (2012).

48. Hare, J., Samangooei, S., and Dupplaw, D. OpenIMAJ and ImageTerrier: Java Libraries and Tools for Scalable Multimedia Analysis and Indexing of Images. In *ACM Multimedia Conference 2011* (2011), pp. 691–694.

49. Hays, J., and Efros, A. A. IM2GPS: Estimating geographic information from a single image. *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 05* (2008).

50. Itseez. Open source computer vision library. `https://github.com/itseez/opencv`, 2015.

51. Jeon, J., Lavrenko, V., and Manmatha, R. Automatic image annotation and retrieval using cross-media relevance models. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR '03)* (2003), 119–126.

52. Ke, Y., and Sukthankar, R. {PCA-SIFT}: A More Distinctive Representation for Local Image Descriptors. *Proc. of IEEE Int'l Conference on Computer Vision and Pattern Recognition (CVPR)* (2004).

53. Kovačević, A., Liebau, N., and Steinmetz, R. Globase.KOM - A P2P overlay for fully retrievable location-based search. *Proceedings - P2P - 7th IEEE International Conference on Peer-to-Peer Computing* (2007), 87–94.

54. Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (2012), pp. 1097–1105.

55. Le, Q. V., Ranzato, M. A., Devin, M., Corrado, G. S., Ng, A. Y., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J., and Ng, A. Y. Building high-level features using large scale unsupervised learning. *International Conference in Machine Learning 28*, 4 (2011), 38115.

56. Ledwich, L., and Williams, S. Reduced SIFT features for image retrieval and indoor localisation. *Australian conference on robotics and automation* (2004).

57. Lee, K., Ganti, R. K., Srivatsa, M., and Liu, L. Efficient spatial query processing for big data. *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL '14* (2014), 469–472.

58. Li, T., Mei, T., Kweon, I. S., and Hua, X. S. Contextual bag-of-words for visual categorization. *IEEE Transactions on Circuits and Systems for Video Technology 21*, 4 (2011), 381–392.

59. Li, X., Li, X., Chen, L., Chen, L., Zhang, L., Zhang, L., Lin, F., Lin, F., Ma, W.-Y., and Ma, W.-Y. Image annotation by large-scale content-based image retrieval. *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia* (2006), 607–610.

60. Liu, Y., Zhang, D., Lu, G., and Ma, W. Y. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition 40*, 1 (2007), 262–282.

61. Liu, Z., Li, H., Member, S., Zhou, W., and Hong, R. Uniting Keypoints : Local Visual Information Fusion for Large-Scale Image Search. *IEEE Transactions on Multimedia 17*, 4 (2015), 538–548.

62. Lowe, D. Object Recognition fromLocal Scale-Invariant Features. *IEEE International Conference on Computer Vision* (1999), 1150–1157.

63. Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110.

64. Lua, E. K., Crowcroft, J., Pias, M., Sharma, R., and Lim, S. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *Ieee Communications Survey and Tutorial 7*, 2 (2004), 72–93.

65. Makadia, A., Pavlovic, V., and Kumar, S. A new baseline for image annotation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 5304 LNCS*, PART 3 (2008), 316–329.

66. Malensek, M., Pallickara, S., and Pallickara, S. Polygon-based query evaluation over geospatial data using distributed hash tables. In *Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing* (2013), IEEE Computer Society, pp. 219–226.

67. Malkhi, D., Naor, M., and Ratajczak, D. Viceroy: A Scalable and Dynamic Emulation of the Butterfly. *Proceedings of the 21st annual ACM symposium on Principles of distributed computing* (2002), 183–192.

68. Maymounkov, P., and Mazieres, D. Kademlia: A peer-to-peer information system based on the xor metric. *First International Workshop on Peer-to-Peer Systems* (2002), 53–65.

69. Moore, E. On Certain Crinkly Curves. *Transactions of the American Mathematical Society 1*, 1 (1900), 72–90.

70. Mueller, W., Boykin, P. O., Sarshar, N., and Roychowdhury, V. P. Comparison of Image Similarity Queries in P2P Systems. *Computer Communications 31*, 2 (2006), 375–386.

71. Müller, W., Eisenhardt, M., and Henrich, A. Scalable summary based retrieval in p2p networks. In *Proceedings of the 14th ACM international conference on Information and knowledge management* (2005), ACM, pp. 586–593.

72. Naicken, S., Basu, A., Livingston, B., and Rodhetbhai, S. A survey of peer-to-peer network simulators. In *Proceedings of The Seventh Annual Postgraduate Symposium, Liverpool, UK* (2006), vol. 2.

73. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. *Consulted* (2008), 1–9.

74. Neimeyer, G. Geohash tips & tricks. `http://geohash.org/site/tips.html`, 2013.

75. Nistér, D. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence 26*, 6 (2004), 756–770.

76. Oquab, M., Bottou, L., Laptev, I., and Sivic, J. Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), 1717–1724.

77. Perronnin, F., and Dance, C. Fisher Kenrels on Visual Vocabularies for Image Categorizaton. *Proc. {CVPR}* (2006).

78. Picone, M., Amoretti, M., Martalò, M., Zanichelli, F., and Ferrari, G. Combining geo-referencing and network coding for distributed large-scale information management. *Concurrency and Computation: Practice and Experience 27*, 13 (2015), 3295–3315.

79. Picone, M., Amoretti, M., and Zanichelli, F. GeoKad: A P2P distributed localization protocol. *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOM Workshops 2010* (2010), 800–803.

80. Picone, M., Amoretti, M., and Zanichelli, F. Evaluating the robustness of the DGT approach for smartphone-based vehicular networks. *Proceedings - Conference on Local Computer Networks, LCN* (2011), 820–826.

81. Picone, M., Amoretti, M., and Zanichelli, F. Proactive neighbor localization based on distributed geographic table. *International Journal of Pervasive Computing and Communications 7*, 3 (2011), 240–263.

82. Picone, M., Amoretti, M., and Zanichelli, F. Proactive neighbor localization based on distributed geographic table. *International Journal of Pervasive Computing and Communications 7*, 3 (2011), 240–263.

83. Picone, M., Amoretti, M., Zanichelli, F., and Conte, G. Location-aware Overlay Scheme for Vehicular Networks. *Researchgate.Net* (2011).

84. Poushter, J. Smartphone ownership and internet usage contributes to climb in emerging economies, February [Retrieved 18-February-2017]. `http://www.pewglobal.org/2016/02/22/`

smartphone-ownership-and-internet-usage-continues-to-climb-\
in-emerging-economies/.

85. Powers, D. M. W. Applications and Explanations of Zipf's Law. In *New Methods in Language Processing and Computational Natural Language Learning* (1998), pp. 151–160.

86. Pratiwi, D. The Use of Self Organizing Map Method and Feature Selection in Image Database Classification System. *CoRR* (2012), 5.

87. Procopiuc, O. Geometry on the Spheregoogle's s2 library. Slide Presentation, 2004. https://docs.google.com/presentation/d/1Hl4KapfAENAOf4gv-pSngKwvS_jwNVHRPZTTDzXXn6Q.

88. Raquet, J. What's Next for Practical Ubiquitous Navigation? *InsideGNSS*, Sep/Oct (2013), 61–69.

89. Ratnasamy, S., Handley, M., Karp, R., and Shenker, S. Application-level multicast using content-addressable networks. *Networked Group Communication* (2001), 14–29.

90. Robertson, D. P., and Cipolla, R. An Image-Based System for Urban Navigation. *Bmvc* (2004), 1–10.

91. Rowstron, A., and Druschel, P. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. *Middleware 2001 2218*, November 2001 (2001), 329–350.

92. Rui, Y., Huang, T. S., and Chang, S.-F. Image Retrieval: Current Techniques, Promising Directions, and Open Issues. *Journal of Visual Communication and Image Representation 10*, 1 (1999), 39–62.

93. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision 115*, 3 (2015), 211–252.

94. Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. Labelme: a database and web-based tool for image annotation. *International journal of computer vision 77*, 1 (2008), 157–173.

95. Salton, G., and McGill, M. J. *Introduction to Modern Information Retrieval*. 1986.

96. Scaramuzza, D., and Fraundorfer, F. Visual Odometry : Part I: The First 30 Years and Fundamentals. *IEEE Robotics & Automation Magazine 18*, 4 (2011), 80–92.

97. Scaramuzza, D., and Fraundorfer, F. Visual Odometry Part II. *IEEE Robotics & Automation Magazine 18*, 4 (2011), 80–92.

98. Schindler, G., and Dellaert, F. 4d cities: analyzing, visualizing, and interacting with historical urban photo collections.

99. Se, S., Lowe, D., and Little, J. Vision-based mobile robot localization and mapping using scale-invariant features. *Proceedings 2001 ICRA IEEE International Conference on Robotics and Automation Cat No01CH37164 2* (2001), 2051–2058.

100. Sierra-Canto, X., Madera-Ramirez, F., and Uc-Cetina, V. Parallel Training of a Back-Propagation Neural Network Using CUDA. *2010 Ninth International Conference on Machine Learning and Applications* (2010), 307–312.

101. Sørensen, L., Jacobsen, L., and Hansen, J. Low cost and flexible UAV deployment of sensors. *Sensors (Switzerland) 17*, 1 (2017), 1–13.

102. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, M. F., Dabek, F., and Balakrishnan, H. Chord: A scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking (TON) 11*, 1 (2003), 17–32.

103. Suwardi, I. S., Dharma, D., Satya, D. P., and Lestari, D. P. Geohash index based spatial data model for corporate. In *Electrical Engineering and Informatics (ICEEI), 2015 International Conference on* (2015), IEEE, pp. 478–483.

104. Szalay, A. S., Gray, J., Fekete, G., Kunszt, P. Z., Kukol, P., and Thakar, A. Indexing the sphere with the hierarchical triangular mesh. *arXiv preprint cs/0701164* (2007).

105. Tang, J., and Lewis, P. H. A study of quality issues for image auto-annotation with the corel dataset. *IEEE Transactions on Circuits and Systems for Video Technology 17*, 3 (2007), 384–389.

106. Tirilly, P., Claveau, V., and Gros, P. Language modeling for bag-of-visual word image categorization. *ACM International Conference on Image and Video Retrieval 2008* (2008).

107. Tobler, A. W. R. A Computer Movie Simulation Urban Growth in Detroit Region. *Economic Geography 46*, 332 (1970), 234–240.

108. Uetz, R., and Behnke, S. Large-scale object recognition with CUDA-accelerated hierarchical neural networks. *Proceedings - 2009 IEEE International*

Conference on Intelligent Computing and Intelligent Systems, ICIS 2009 1, Icis (2009), 536–541.

109. Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 3156–3164.

110. Wang, C., Blei, D., and Fei-Fei, L. Simultaneous image classification and annotation. *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009* (2009), 1903–1910.

111. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., and Wu, Y. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 1386–1393.

112. Wang, J., Zha, H., and Cipolla, R. Coarse-to-fine vision-based localization by indexing scale-invariant features. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 36*, 2 (2006), 413–422.

113. Wang, Y., Liu, X., Wei, H., Forman, G., Chen, C., and Zhu, Y. CrowdAtlas: Self-Updating Maps for Cloud and Personal Use. In *MobiSys 2013* (2013), p. 27.

114. Ward, J., Andreev, S., Heredia, F., Lazar, B., and Manevska, Z. Efficient mapping of the training of convolutional neural networks to a cuda-based cluster. *Eindhoven University of Technology, The Netherlands 12* (2011).

115. Weninger, F., Bergmann, J., and Schuller, B. Introducing CURRENNT: The Munich Open-Source CUDA RecurREnt Neural Network Toolkit. *Journal of Machine Learning Research 16* (2015), 547–551.

116. Weyand, T., Kostrikov, I., and Philbin, J. Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision* (2016), Springer, pp. 37–55.

117. White, R. A., and Stemwedel, S. W. The quadrilateralized spherical cube and quad-tree for all sky data. In *Astronomical Data Analysis Software and Systems I* (1992), vol. 25, p. 379.

118. Wood, G. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper* (2014).

119. Yang, J., Jiang, Y.-G., Hauptmann, A. G., and Ngo, C.-W. Evaluating bag-of-visual-words representations in scene classification. *Proceedings of the international workshop on Workshop on multimedia information retrieval MIR 07 63* (2007), 197.

120. Zhang, D., Islam, M. M., and Lu, G. A review on automatic image annotation techniques. *Pattern Recognition 45*, 1 (2012), 346–362.

121. Zhang, W., and Kosecka, J. Image Based Localization in Urban Environments. *3D Data Processing, Visualization, and Transmission, Third International Symposium on* (2006), 33–40.

122. Zhao, B. Y., Huang, L., Rhea, S. C., Stribling, J., Joseph, a. D., and Kubiatowicz, J. Tapestry: A global-scale overlay for rapid service deployment. *IEEE Journal on Selected Areas in Communications 22*, 1 (2004), 41–53.

123. Zhou, G., Liu, A., Yang, K., Wang, T., and Li, Z. An embedded solution to visual mapping for consumer drones. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2014), pp. 656–661.

124. Ziegler, G. M. *Lectures on polytopes*, vol. 152. Springer Science & Business Media, 2012.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
14-09-2017	Doctoral Dissertation	August 2014 - September 2017

4. TITLE AND SUBTITLE

A Location-Aware Middleware Framework for Collaborative Visual Information Discovery and Retrieval

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Compton, Andrew JM, Maj

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
Wright-Patterson AFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-DS-17-S-010

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Intentionally Left Blank

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution Statement A: Approved for Public Release; Distribution Unlimited.

13. SUPPLEMENTARY NOTES

This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

This work addresses the problem of scalable location-aware distributed indexing to enable the leveraging of collaborative effort for the construction and maintenance of world-scale visual maps and models which could support numerous activities including navigation, visual localization, persistent surveillance, structure from motion, and hazard or disaster detection. Current distributed approaches to mapping and modeling fail to incorporate global geospatial addressing and are limited in their functionality to customize search. Our solution is a peer-to-peer middleware framework based on XOR distance routing which employs a Hilbert Space curve addressing scheme in a novel distributed geographic index. This allows for a universal addressing scheme supporting publish and search in dynamic environments while ensuring global availability of the model and scalability with respect to geographic size and number of users. The framework is evaluated using large-scale network simulations and a search application that supports visual navigation in real-world experiments.

15. SUBJECT TERMS

Distributed geographic index, peer-to-peer, location-based search, overlay network.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			LtCol John M. Pecarina, AFIT/ENG
U	U	U	UU	192	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636 x3638 john.pecarina@us.af.mil