NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A SURVEY OF CLIENT GEOLOCATION USING WI-FI POSITIONING SERVICES

by

Nicholas D. Lange

March 2014

Thesis Co-Advisors: Mark Gondree
Preetha Thulasiraman

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2014	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE
A SURVEY OF CLIENT GEOLOCATION USING WI-FI POSITIONING SERVICES

5. FUNDING NUMBERS

6. AUTHOR(S) Nicholas D. Lange

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

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

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)
Wi-Fi positioning systems (WPS) utilize a location's set of Wi-Fi access point (AP) media access control (MAC) addresses and received signal strength pairs as input to an algorithm that resolves location referencing a database of spatially labeled AP data. WPS are particularly useful in urban canyons where Global Positioning System (GPS) satellite views are often blocked. WPS can provide a quicker result than GPS with more accuracy than Internet Protocol (IP) or cellular geolocation. In this work, we present the design and construction of a corpus of Wi-Fi AP MAC address sets derived from the Wireless Geographic Logging Engine (WiGLE) database and Census Bureau data. We use our corpus of MAC address queries as input to controlled WPS requests. For the resulting WPS responses, we compare the overlap, centroid distance, and provide insight into the services' accuracy and inter-agreement.

14. SUBJECT TERMS Geolocation, RSSI geolocation, location spoofing, location positioning

15. NUMBER OF PAGES
63

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

A SURVEY OF CLIENT GEOLOCATION USING WI-FI POSITIONING SERVICES

Nicholas Lange
Lieutenant, United States Navy
B.S., University of Evansville, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2014

Author: Nicholas Lange

Approved by: Mark Gondree
Thesis Co-Advisor

Preetha Thulasiraman
Thesis Co-Advisor

Peter Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Wi-Fi positioning systems (WPS) utilize a location's set of Wi-Fi access point (AP) media access control (MAC) addresses and received signal strength pairs as input to an algorithm that resolves location referencing a database of spatially labeled AP data. WPS are particularly useful in urban canyons where Global Positioning System (GPS) satellite views are often blocked. WPS can provide a quicker result than GPS with more accuracy than Internet Protocol (IP) or cellular geolocation.

In this work, we present the design and construction of a corpus of Wi-Fi AP MAC address sets derived from the Wireless Geographic Logging Engine (WiGLE) database and Census Bureau data. We use our corpus of MAC address queries as input to controlled WPS requests. For the resulting WPS responses, we compare the overlap, centroid distance, and provide insight into the services' accuracy and inter-agreement.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

A-GPS	assisted global positioning system
AoA	angle of arrival
AP	access point
CI	cell identity
GPS	global positioning system
IP	internet protocol
IPS	indoor positioning systems
MAC	media access control
MLE	maximum-likelihood estimate
OMB	office of management and budget
PoPs	points of presence
RSS	received signal strength
RSSI	received signal strength indicator
SSID	service set identifier
ToA	time of arrival
TDoA	time difference of arrival
WiGLE	Wireless Geographic Logging Engine
WPS	Wi-Fi positioning system

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and appreciation to my advisors, Professor Mark Gondree and Professor Preetha Thulasiraman for their direction and guidance in helping me develop and write my thesis. Without their support and encouragement, I would not have been able to complete this work.

Finally, I would like to thank my wife, April, for her support, patience, and understanding throughout this process.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Wi-Fi positioning systems (WPS) utilize a location's set of Wi-Fi access point (AP) media access control (MAC) addresses and received signal strength pairs as input to an algorithm that resolves location, referencing a database of spatially-labeled AP data. WPS are particularly useful in urban canyons where Global Positioning System (GPS) satellite views are often blocked. WPS can provide a quicker result than GPS, with more accuracy than Internet Protocol (IP) or cellular geolocation. WPS are used in a wide variety of smartphones, web applications, entertainment devices and business tools.

Related work has compared IP-based geolocation services [1] and evaluated different modes of geolocation on single devices [2]. To our knowledge, there has not been a study directly comparing WPS. In this work, we present the design and construction of a corpus of Wi-Fi AP MAC address sets derived from the Wireless Geographic Logging Engine (WiGLE) database and U.S. Census Bureau data. We use our corpus of MAC address queries as input to controlled WPS requests, to investigate the Google, Microsoft and Skyhook WPS services. For the resulting responses, we compare the response precision, failure behavior, and provide insight into the services' accuracy and inter-agreement. We find services to demonstrate notable, unique behaviors Microsoft was found to be most likely to return a failure while Skyhook was least likely to return a failure. All services reported location guesses with precision better than 100 meters for 80 percent of their responses, with best performance in regions with high population density. We find significant differences between services, in both their failure and non-failure behavior. Most failures were shared pair-wise with some other service, but 46.4 percent of non-common failures were unique to some service. Considering service interagreement, we find Google/Microsoft and Microsoft/Skyhook equally likely to agree as disagree while Google/Skyhook are more likely to disagree than agree.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

A Wi-Fi positioning system (WPS) is a service that uses prior observations to determine location from a set of Wi-Fi access points (AP) observed by a client. Media access control (MAC) addresses and received signal strength pairs are the inputs to an algorithm that determines location using a database of spatially labeled AP data. WPS is particularly useful in urban canyons where views of GPS satellites are often blocked [3]. In some scenarios, WPS calculates location faster than GPS and more accurately than IP-based geolocation or cellular-based geolocation [4].

Three general architectures have been proposed for WPS: network based, terminal based and terminal assisted. In network-based WPS, location is determined by the strength of the beacon the mobile device emits, as received by the APs and a central server. Network-based WPS requires each AP to have the capability of routing measurement data to the WPS server; this is also the primary downside to this topology. In terminal-based WPS, the mobile device receives beacons from the APs and determines location from its local database and device-resident logic. The disadvantage to this architecture is the requirement for the mobile device to store the database of past observations. In the terminal-assisted architecture, the mobile device receives AP beacons, forwards its observations to a central server whose database of prior observations is used to infer location [5]. Terminal-assisted WPS architectures are the most common among commercial services. For example, Google, Microsoft, Skyhook and Navizon all employ terminal-assisted architectures. Apple's WPS appears to employ a hybrid of terminal-based and terminal-assisted architectures: client devices receive beacons from APs and send these data to a remote service; the service returns a small, relevant sample from its database to the client; the client determines a final location using this data sample.

All WPS require a calibration phase, where a database is built from signal measurements obtained by some spatially-aware device (i.e., an initial set of labeled data). This is normally accomplished by collecting data for Wi-Fi access points via war driving or using database submissions from GPS-equipped devices. Systems have been

proposed that self map Wi-Fi access points during system operation [6], rather than employ a dedicated calibration phase.

Using measurements in this database, location position can be inferred from any query. Numerous algorithms have been proposed for use in outdoor WPS to infer location: cell identity (CI), trilateration based on time of arrival (ToA), trilateration based on time difference of arrival (TDoA), trilateration based on received signal strength (RSS), triangulation based on angle of arrival (AoA), fingerprinting [5], [3] or signature-based [7], maximum-likelihood estimation (MLE) based on received signal strength (RSS) [8], clustering [9], particle filters [3] and hierarchical Bayesian sensor models [10]. In contrast, indoor positioning systems (IPS) using AP data must employ different techniques for precise indoor positioning [7], [11], [12], [10], [13], [14] to compensate for a variety factors unique to that setting (e.g., signal fading due to building materials and signal echoes from reflection and refraction). The focus of this study is commercial WPS for outdoor geolocation. We note that we have little insight into which algorithms and techniques each service provider employs.

A. WPS SERVICES

Google, Skyhook, Microsoft, Navizon and Apple operate popular commercial geolocation services that determine location, either exclusively or partially-based on queries encoding Wi-Fi signal data. We survey these services briefly in Table 1.

Service	Used by	Technique	Data Source	Accuracy
Skyhook	PlayStation Vita, various mobile apps (MapQuest, Kayak, etc.)	No Data	War driving, user submitted via query	10-20m [5]
Google	Android, Google Maps, Chrome, Firefox [8]	MLE [8]	War driving, user submitted via query [15]	<50m @ 80 percent confidence [8]
Navizon	Business and entertainment applications	Triangulation [16]	User submitted via query or Navizon App [16]	No Data
Microsoft	Windows Phones, Bing, Windows, Internet Explorer	No Data	No Data	No Data
Apple	IOS, OSX, Safari	No Data	No Data	No Data

Table 1. Characteristics of commercial WPS services.

B. RELATED WORK

Shavaitt and Zilberman survey and evaluate IP-based geolocation services [1]. They compare seven IP-based geolocation services using an algorithm to group IP addresses to points of presence (PoPs). They found most services returned consistent results, but the accuracy of these results were occasionally erroneous by thousands of kilometers.

Zandburgen evaluates geolocation provided the iPhone 3G, comparing three different modes of operation: using A-GPS, using Wi-Fi signals, and using cellular positioning. They manually surveyed the behavior at select, known locations. They observed cellular positioning accuracy to be consistent with previous studies, but A-GPS to be much less accurate than standalone GPS and Wi-Fi geolocation to be less accurate than its published specifications [2].

THIS PAGE INTENTIONALLY LEFT BLANK

III. METHODOLOGY

Wi-Fi positioning systems resolve location using MAC addresses and RSSI values derived from beacon frames that are continually broadcast by Wi-Fi APIs [2]. To build a query corpus for WPS, we might have visited a set of test geographic locations to record ground truth (i.e., using a high accuracy GPS device) and then record the output of each WPS at that location. This approach would have been labor-intensive and limited to a relatively small number of non-diverse test locations, due to obvious practical constraints (time and cost). The results of such a survey would be technically infeasible for others to reproduce. Further, due to environmental factors, this procedure may not ensure that queries are stable across trials: a device might observe, and thus query, different MAC and RSSI values at the same location, over short time intervals [17]. Our goal is to make timely, controlled, and repeatable queries, allowing apples-to-apples comparison of WPS service behavior. This motivated us to develop our own query corpus, using assumptions that remove the need for ground truth or field observations.

A. QUERY CORPUS FOR WPS

Our ideal WPS query corpus would contain a large number of longitude and latitude points with some set of wireless access points visible at each particular location. This idealized corpus might be represented by the set of triples {(lat, lon, AP)}, where AP = {MAC, RSSI} is some set of MAC address and RSSI pairs visible at a particular (lat, lon) location. Further, the corpus should distinguish points by a geographic region, to compare the performance of WPS across regions of different population densities (e.g., large metropolitan areas versus small urban areas). We discuss our sampling strategy and process for gathering corpus data, next.

B. CORPUS GENERATION

To generate our query corpus, we require a source of spatially-labeled AP MAC addresses. The WiGLE Project is a community-sourced database of wireless access point data [18]. WiGLE users can upload wireless hotspot data observable to the public, including GPS data, SSID, MAC address and the encryption type used by the AP [19].

WiGLE currently contains over 120 million unique Wi-Fi access points, triangulated using over 2 billion unique observations. Users can query the database by geographic location, using the two lat/lon points defining the region's corners. As the WiGLE database contains observations made by many users over a long period of time, the access point data returned for a region may not reflect the true "view" of a wireless device from any single point in time [18].

Corpus generation occurs for each of three classes of geographic areas defined by the U.S. Census Bureau and U.S. Office of Management and Budget. These classes are: micropolitan, metropolitan, and combined statistical areas. U.S. Census Bureau defines a metropolitan statistical area as a metro area containing a core urban area with a population of 50,000 or more. U.S. Census Bureau defines a micropolitan statistical area as a metro area containing a core urban area with a population between 10,000 and less than 50,000. The U.S. Office of Management and Budget (OMB) defines a combined statistical area based on the socioeconomic ties between adjacent metropolitan and micropolitan areas: if ties between areas pass a certain threshold, they become a component of the combined statistical area [20]. In the United States, as of 2013, there are 11 combined statistical areas containing 99 cities, 577 metropolitan cities, and 564 micropolitan cities [21]. For the purpose of corpus generation, every city is defined by the lat/lon of its city center, as provided by MaxMind [22].

For each of our three geographic classes, we generate an independent corpus of spatially labeled AP data. For each region, the process can be summarized as: (a) city selection, (b) target selection, (c) target AP collection. Unless otherwise noted, all selection is simple random sampling with replacement.

1. City Selection

For metropolitan and micropolitan classes, we randomly select a city from the list of cities in that class, as defined by the 2013 U.S. Census. For the U.S. combined statistical areas class, one of the 11 areas is randomly selected, and then a city in that area is randomly selected.

2. Target Selection

Using the lat/lon of the city-center as a starting point, we generate a target location by traveling a random distance (0–2 km) in a random continuous value direction (0–360°). From this, we define a 100m x 100m square region whose center is this target. The target's region is defined by the lat/lon coordinates at its northeast and southwest corners. According to literature Wi-Fi AP radii commonly range from 30m to 200m with the majority of APs being consumer-grade having a radiation distance on the lower end of the range [8]. Relatively small region dimensions were selected to ensure that access points far from one another were not mixed into a single "view."

3. RSSI Value Selection

As we have no way of knowing the actual RSSI value that would be observed in the center of the query box. The ideal RSSI value for an AP in our corpus could be calculated using data correlating RSSI values and distance (for example, see Figure 1) and by calculating the expected distance from the center of our box. We assume points within the box are composed of random independent x and y coordinates uniformly distributed. The expected distance of a randomly chosen point in a unit square can be calculated as follows:

$$\begin{aligned}\overline{d}_{center} &= \int_0^1 \int_0^1 \sqrt{(x-\frac{1}{2})^2 + (y-\frac{1}{2})^2}\, dxdy \\ &= \frac{1}{6}P \\ &= \frac{1}{6}(\sqrt{2} + \sinh^{-1}1) \\ &= 0.3825978582\end{aligned}$$

Using the unit square expected distance we calculated the expected distance in our 100m x 100m square as 38.26 meters [23]. From Figure 1, we find 82 is the median observed signal strength at 38.26 meters. We chose to submit a RSSI value of 50 for each of the MAC addresses because of a related set of experiments.

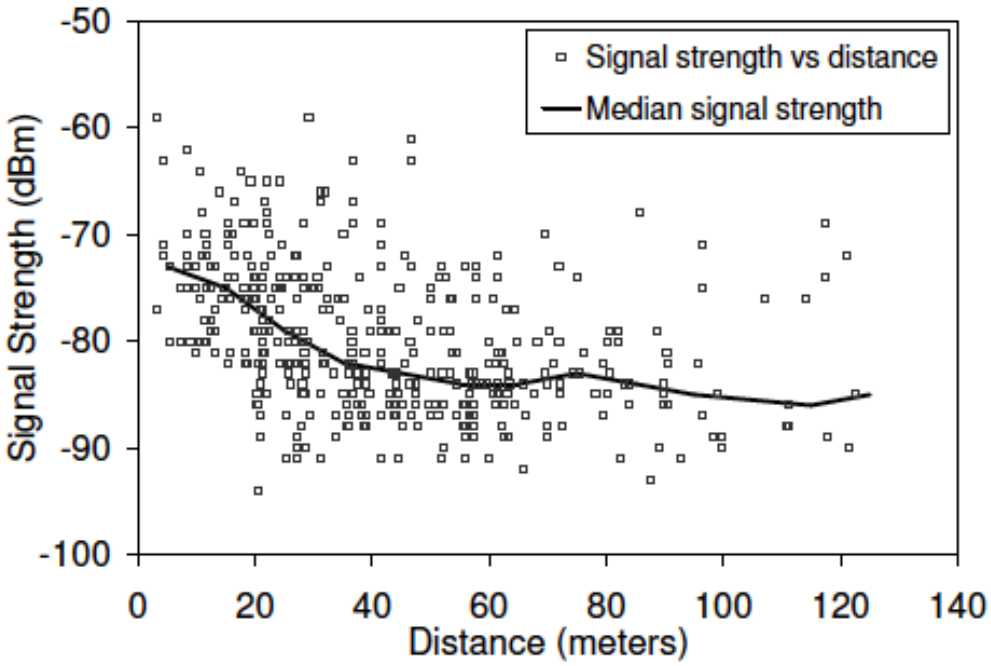


Figure 1. Measured signal strength as a function of distance (from [3]).

4. Target AP Collection

Using the WiGLE database, we gathered access point data associated with the target region. If the database returned two or more MAC addresses for that region, these results were included in the query corpus as an entry. Each corpus entry consists of the lat/lon points defining the 100m x 100m target region ("box"), the lat/lon of the target at the center of this region ("target"), the lat/lon of the city-center originally associated with the target ("origin"), the name and state of the city-center, and the access point MAC addresses associated with the target region ("wireless"). Figure 2 is a sample entry from the query corpus.

```
{   'box': (   [30.147848119691226, -95.4818183792353],
               [30.14874719058295, -95.48077866441854]),
    'origin': {   'city': 'The Woodlands',
                  'city-state': 'The Woodlands, TX',
                  'lat': '30.1577778',
                  'lon': '-95.4891667',
                  'state': 'TX'},
    'target': Point(30.14829765513709, -95.48129852182693, 0.0),
    'RSSI':  '-50'
    'wireless': [   u'00:13:10:1e:ae:02',
                    u'00:40:05:b2:b0:65',
                    u'00:12:17:7a:90:58',
                    u'00:0f:66:57:ac:e8']}]
```

Figure 2. A sample entry in our corpus.

If fewer than two MACs are returned, we discard these results and re-sample, selecting a new city for that geographic class. We continue this process until our query corpus has reached the desired size. Our final query corpus contains 1550 entries for each geographic class, for a total of 4650 target queries. The location of the points in our corpus is depicted in Figure 3 and a summary is given in Table 2.

	Census Data			Corpus		
	Micropolitan	Metropolitan	Combined Statistical	Micropolitan	Metropolitan	Combined Statistical
Queries	N/A	N/A	N/A	1550	1550	1550
Cities Represented	564	577	99	452	477	98
Areas	N/A		11	N/A	N/A	11

Table 2. Summary of corpus queries.

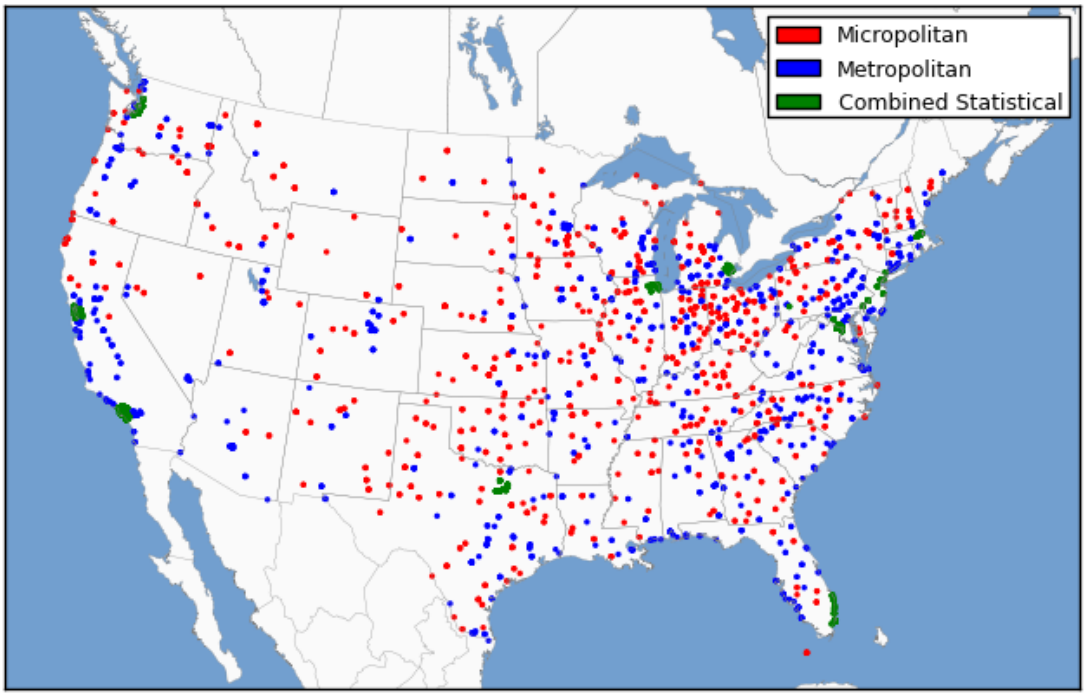


Figure 3. Location of all corpus queries.

C. QUERYING SERVICES

We developed a tool to query each wireless location service using our corpus data. Our tool can submit a query to either of the Google, Skyhook, or Microsoft geolocation services, using the wireless access point and RSSI values from each entry in our corpus. Each geolocation service has some recognizable failure behavior if it is unable to determine the location given the input data. When successful, each service returns a location (lat/lon) and accuracy (in meters). We describe some of the relevant details of this process, next.

1. Skyhook Location Service

During normal operation, Skyhook's WPS uses an installed API to get the Wi-Fi access point data observed by the user's system and submits this information as a query in XML format. To submit custom queries, it is necessary to send a handcrafted XML query via an HTTPS POST request. Others have accomplished this to geo-locate arbitrary wireless routers by submitting a query with a single access point MAC [24, 25, 26]. We

modified these techniques to make multiple MAC queries. Skyhook returns a specific "location not found" message if it is unable to determine a location for a query.

2. Google Location Service

Google's WPS can be queried in a variety of ways, including a handcrafted HTTP request [27]. If the service is unsuccessful in geo-locating based on access point MAC address data, it returns a result based upon IP geo-location. Our tool recognizes when Google returns IP geo-location responses, and discards this result as a failure. Although the service does not explicitly indicate error, any responses based on IP geo-location are recognizable by comparing with a query containing no AP MAC inputs. The service limits each query to include at most 37 MAC addresses. We truncate queries from our corpus when necessary, using up to the first 37 MAC addresses collected from WiGLE.

3. Microsoft Location Service

Microsoft's WPS can be queried using a handcrafted XML request, similar to the Skyhook service [28]. The service will return a "location not found" message if it is unable to determine a location in response to a request.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. ANAYLISIS

We used the tool we developed to query each wireless location service using the corpus data, (see Chapter 3, Section B). This was done during two separate two-week periods at the beginning of December 2013 and at the beginning of February 2014. Our queries were performed against our three target services: Google, Microsoft and Skyhook. We collected a total of 1550 responses from each service per geographic class, with no more than 33 percent of those responses being indicators of failure. We summarize observed failure behavior in section A. In section B we look at a notion of precision using the "accuracy" value returned by the service. We look at accuracy, which we measure as the distance from the service's response to the center of the corpus query box. Finally we look at the level of interagreement between the services. Throughout this chapter we use consistent notation for the relationship between queries and responses, summarized in Figure 4. Where clear, we often abuse notation, writing c instead of c_i and r instead of r_i.

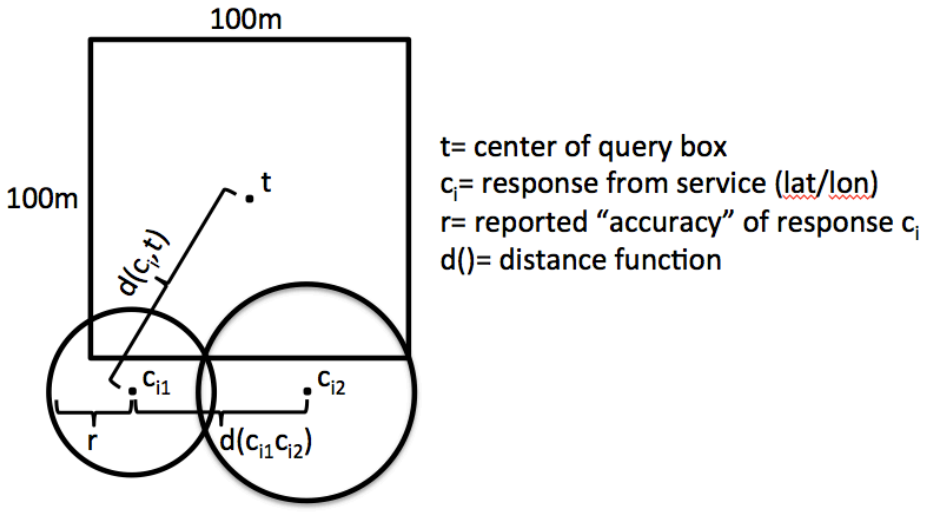


Figure 4. Terms used in analysis.

A. FAILURE ANAYLISIS

When a service is unable to resolve a location given the set of input data, we detect it and mark this as a failure. In Figure 5, we plot the location of all query failures. They are distributed throughout every geographic class and appear to be distributed in proportion to our corpus.

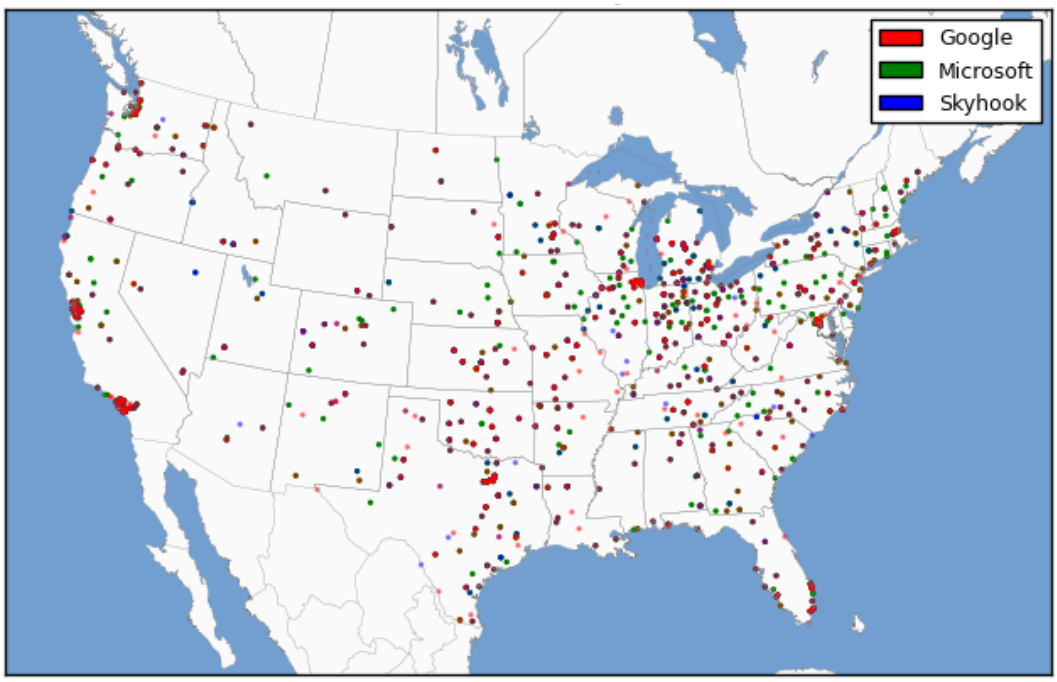


Figure 5. Location of corpus queries yielding WPS failure responses.

We calculated the mean query lengths for each geographic class, separating successful and non-successful queries by service (see Table 3). The mean number of MACs in a query was greater for high-density geographic classes, as expected. When examining the number of MACs in failed queries, we noticed much less variation from class-to-class and a much smaller mean length.

Geographic Class	Service	Mean Number of MACs in Query	Mean Number of MACs in Successful Query	Mean Number of MACs in Failed Query
Micropolitan	Microsoft	9.787	12.36	4.55
	Skyhook		11.49	3.51
	Google		11.38	3.63
Metropolitan	Microsoft	19.854	23.1	7.25
	Skyhook		21.89	3.3
	Google		22.81	4.83
Combined Statistical	Microsoft	30.905	35.56	7.77
	Skyhook		33	2.92
	Google		34.87	4.53

Table 3. Mean query lengths.

In Table 4, we further examine the service failures by number of MAC addresses in the query. We found Microsoft to have a greater number of failures for every geographic class and every query length. Skyhook and Google had nearly equal number of failures in the Micropolitan class. In more densely populated areas (i.e., metropolitan and combined statistical classes), Skyhook returned significantly fewer failures in every case.

Geographic Class	Service	All query lengths	>2 MACs	>3 MACs	>4 MACs	>5 MACs	>6 MACs	>7 MACs	>8 MACs
Micropolitan	Microsoft	512	330	218	147	110	80	61	50
	Skyhook	331	181	99	64	40	24	20	16
	Google	319	173	88	55	38	20	15	12
Metropolitan	Microsoft	318	206	151	114	87	67	55	49
	Skyhook	170	76	44	24	17	13	9	9
	Google	255	148	100	66	49	38	27	25
Combined Statistical	Microsoft	260	178	120	95	75	64	54	50
	Skyhook	108	44	19	10	6	4	4	4
	Google	203	119	75	58	41	31	24	23

Table 4. Failures by region, service and number of MACs in query.

Positioning services require at least two proximate AP MACs in a query to return a position. This behavior is by design, in part, to protect the privacy of Wi-Fi AP owners, preventing the geolocation of arbitrary, individual AP devices. Consequently, queries will fail if the service recognizes less than two MACs in our query as geographically proximate. The fact that data obtained from WiGLE database contains observations made by many users over a long period of time likely contributes to a high number of failures at lower query lengths.

As discussed earlier, the AP data collected from WiGLE for a region may not reflect the true "view" of the wireless environment from any single point in time. To compensate, we removed the 439 failures that were shared amongst all services (see Table 5). We believe the common failures are likely attributable to historic WiGLE data that, when aggregated, fails to reflect an authentic view. Excluding common failures, we continued to observe Microsoft to have a greater number of failures for every geographic class and for every query length. Excluding common errors, 15.5 percent of Microsoft queries resulted in failure, compared to 8.0 percent and 4.0 percent for Google and Skyhook, respectively. Both Skyhook and Microsoft showed fewer failures in areas of higher population density: non-common failure distribution by area (micropolitan, metropolitan, combined statistical areas) is 65.3 percent, 22.4 percent, 12.4 percent for Skyhook and 44.9 percent, 28.6 percent, 26.7 percent for Microsoft. Google's non-common failures, in comparison, were distributed rather evenly between classes (29.3 percent, 36.4 percent, 34.3 percent). Skyhook and Google had nearly equal number (~100) of failures in the micropolitan class; however, this absolute value represents a much larger proportion of failures for Skyhook (failures in the micropolitan class represent 65.3 percent of all non-common failures for Skyhook, vs. 29.3 percent for Google). In Table 6, we examine the unique failures generated by each service. Excluding common failures, 56.4 percent of Microsoft failures were unique to Microsoft alone while only 39 percent and 22 percent were unique to Google and Skyhook, respectively. We observed a significantly fewer total number of unique failures from the Skyhook service (38 across all geographic classes, versus 132 from Google and 367 from Microsoft). In later sections, we consider pair-wise shared failures, as it relates to service interagreement.

Geographic Class	Service	All query lengths	>2 MACs	>3 MACs	>4 MACs	>5 MACs	>6 MACs	>7 MACs	>8 MACs
Micropolitan	Microsoft	292	219	165	116	90	71	54	45
	Skyhook	111	70	46	33	20	15	13	11
	Google	99	62	35	24	18	11	8	7
Metropolitan	Microsoft	186	152	122	100	78	59	49	43
	Skyhook	38	22	15	10	8	5	3	3
	Google	123	94	71	52	40	30	21	19
Combined Statistical	Microsoft	173	143	105	88	70	61	51	47
	Skyhook	21	9	4	3	1	1	1	1
	Google	116	84	60	51	36	28	21	20

Table 5. Failures by region, service and number of MACs in query (excluding common failures).

Service	Geographic Class	Non-common failures	Unique Failures
Google	Micropolitan	99	48
	Metropolitan	123	43
	Combined Statistical	116	41
	Total	**338**	**132**
Microsoft	Micropolitan	292	168
	Metropolitan	186	100
	Combined Statistical	173	99
	Total	**650**	**367**
Skyhook	Micropolitan	111	18
	Metropolitan	38	14
	Combined Statistical	21	6
	Total	**170**	**38**

Table 6. Non-common and unique failures by region and service.

B. PRECISION

In this section, we consider the *precision* of each service. Our working definition of precision is the response "accuracy" reported by the service. This is the radius r of the circle centered at c_i provided in the service's response. Abstractly, we consider a service's response to encode a collection of guesses (possible locations), all of which are contained in the reported circle. The smaller the radius of this circle, the more these guesses tend to agree with one another; this aligns with the traditional notion of precision in repeated trials. Another possible definition of precision is the "closeness" of the circles reported in response to identical queries. Since we control queries very carefully, this definition of precision would be uninteresting to explore: for all our services, responses to the same query are identical (at least over short periods of time).

For the Google service, precision appears quite consistent across all three geographic classes (see Figure 6). Response radii range from 20 m to 405 m, where 80 percent of the radii are ~125 m or less. The most notable feature of Google's service is the dramatic spike in responses with ~35 m radius precision.

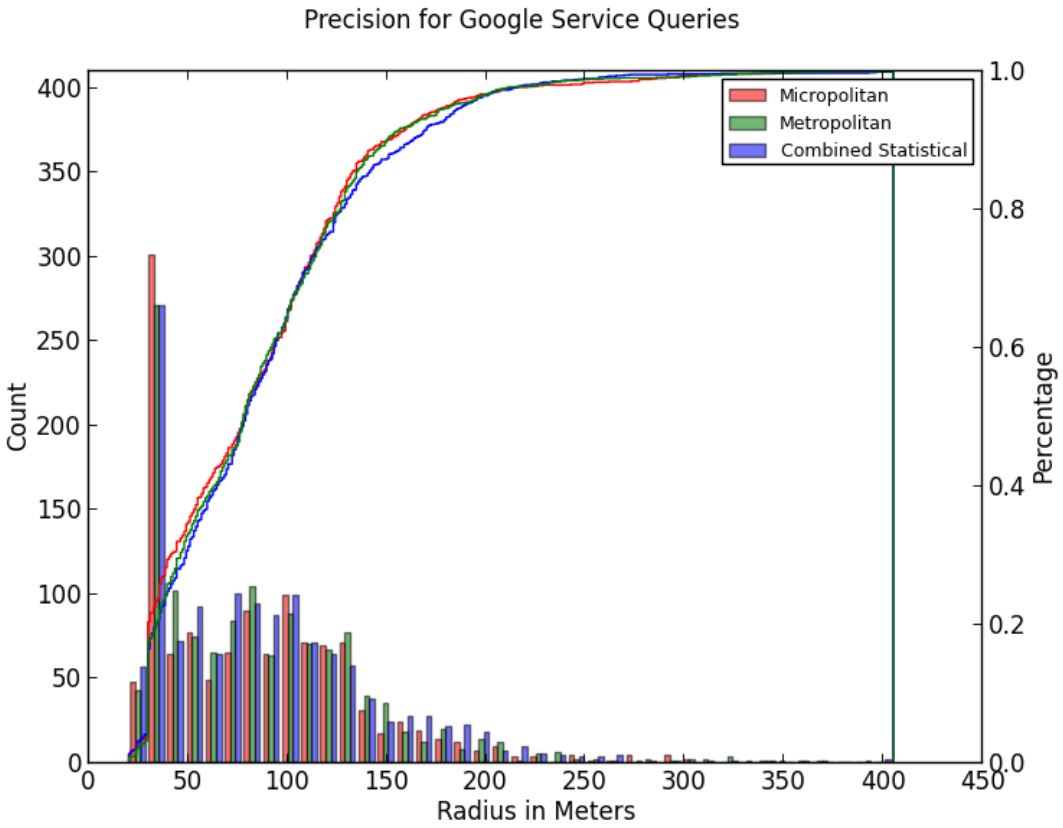


Figure 6. Precision for Google service results.

For the Microsoft service, response radii range from 15 m to 372 m, where 80 percent of the radii are ~100 m or less (see Figure 7). The most notable feature of Microsoft's precision results is the gap in precision values between ~20 m and ~50 m. Microsoft service performed better in more urban areas, as shown by the CDF.

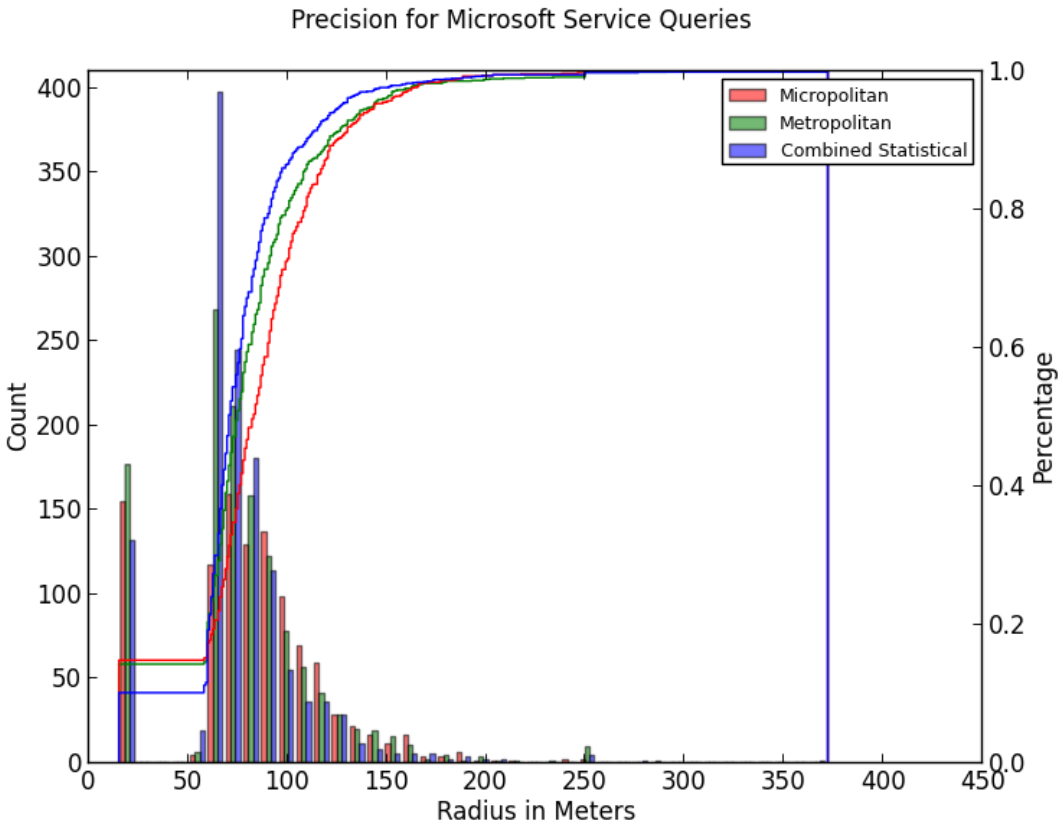


Figure 7. Precision for Microsoft service results.

For the Skyhook service, response radii range from 10 m to 450 m, where 80 percent of the precision values are ~140 m or less (see Figure 8). The most notable feature of Skyhook's precision distribution is the spike of responses with ~150 m and ~200 m radius precision. Skyhook's service performed better in more urban areas: half of all responses for queries in cities of combined statistical areas are 60 m or less in radius.

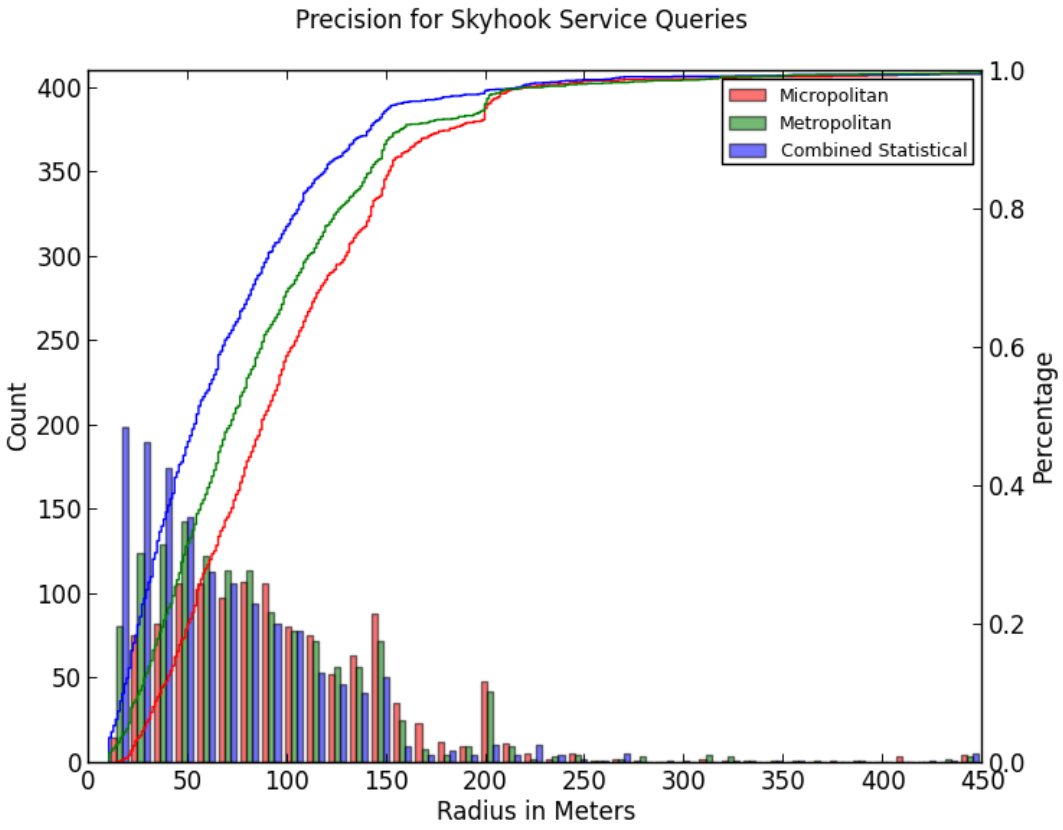


Figure 8. Precision for Skyhook service results.

Comparing Skyhook, Google, and Microsoft, we find Microsoft to have a higher reported precision (smaller radii) than Google, and Google to have higher reported precision than Skyhook. While this may suggest that Microsoft has better performance, one must consider Microsoft's much higher failure rate.

C. ACCURACY

In this section, we consider service *accuracy*, defining this as d(c,t), the distance from the target t to the response's centroid c. Defining accuracy in this way assumes that the target t is a meaningful landmark. The query for target t, however, is derived from user-submitted WiGLE data: it may not reflect an authentic "view" of the APs near t at any one point in time—in particular, these APs may not reflect the view of the target at the time we issued the query to the service. Nonetheless, for each case, we consider the distribution of accuracies by service and region. We consider responses within 400m of the target and those farther than 400m ("outliers") as separate cases, and report on each.

For the Google service, the majority of target accuracies fall between 20–75 m (see Figure 9). Google's service performed significantly better in the combined statistical area class: 80 percent of responses are within ~90 m of the target for micropolitan and metropolitan areas, while 80 percent of the responses are within ~70 m of the target for cities of combined statistical areas.

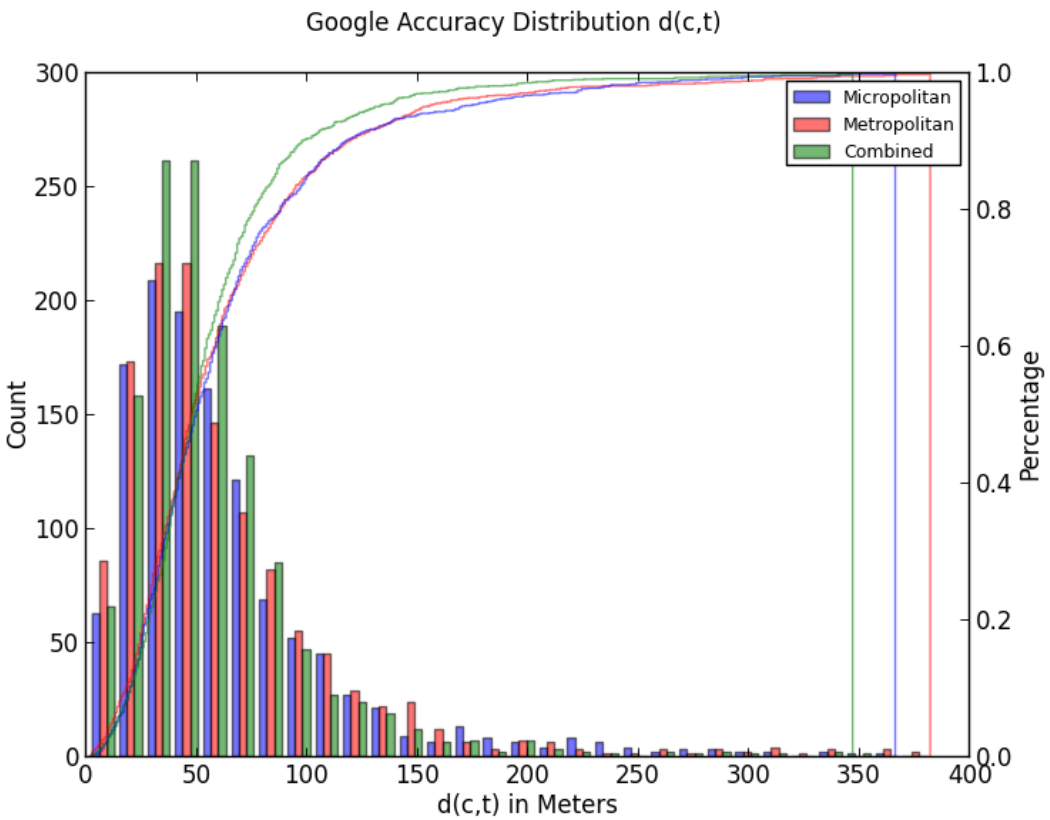


Figure 9. Google service accuracy distribution d(c,t).

For the Microsoft service, the majority of target accuracies fall between 20–75 m (see Figure 10). Microsoft's service achieved greatest accuracy in the combined statistical area class, with slightly poorer accuracy in the metropolitan class: 80 percent of the responses are within ~100 m of the target for micropolitan and metropolitan areas, while 80 percent of the responses are within ~85 m of the target for cities of combined statistical areas. Microsoft's service provided the least accurate results in the micropolitan geographic class.

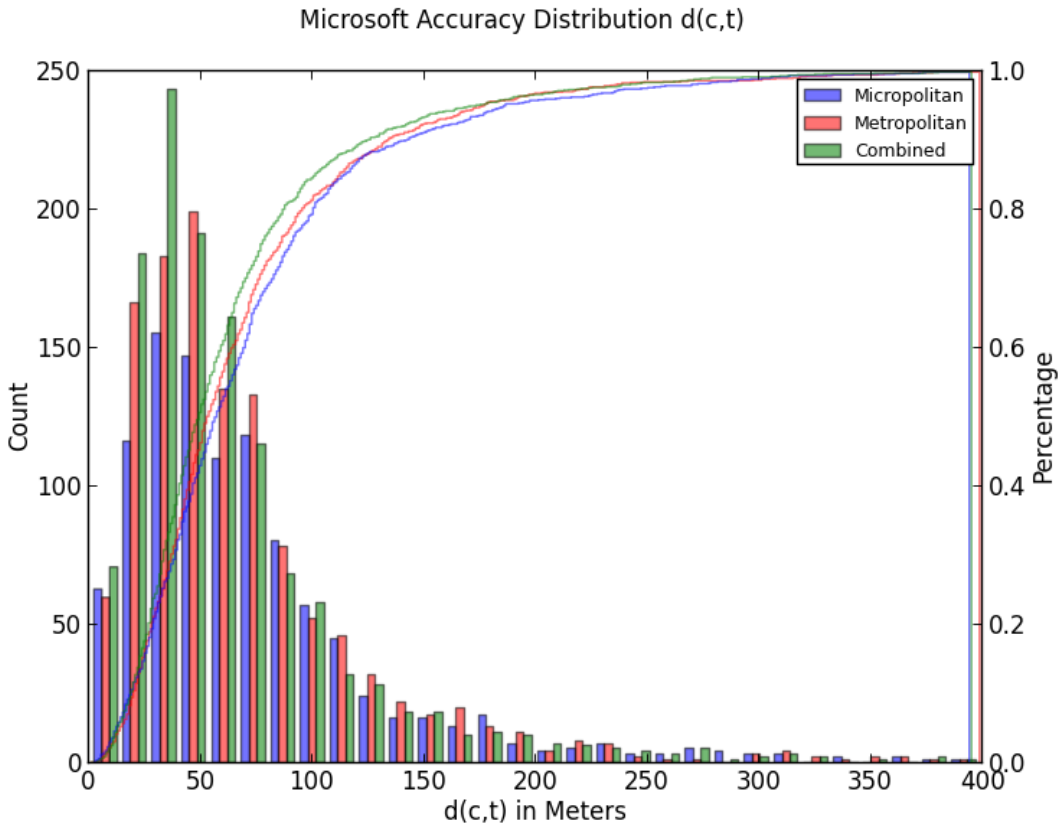


Figure 10. Microsoft service accuracy distribution d(c,t).

For the Skyhook service, the majority of target accuracies fall between 25–75 m (see Figure 11). Skyhook's service achieved greatest accuracy in combined statistical area queries, with slightly poorer accuracy in metropolitan queries and poorest results in the micropolitan geographic class: 80 percent of the responses are within ~100 m of the target for micropolitan areas, 80 percent of responses are within ~90 m of the target for metropolitan areas, and 80 percent of responses are within ~70 m of the target for cities of combined statistical areas.

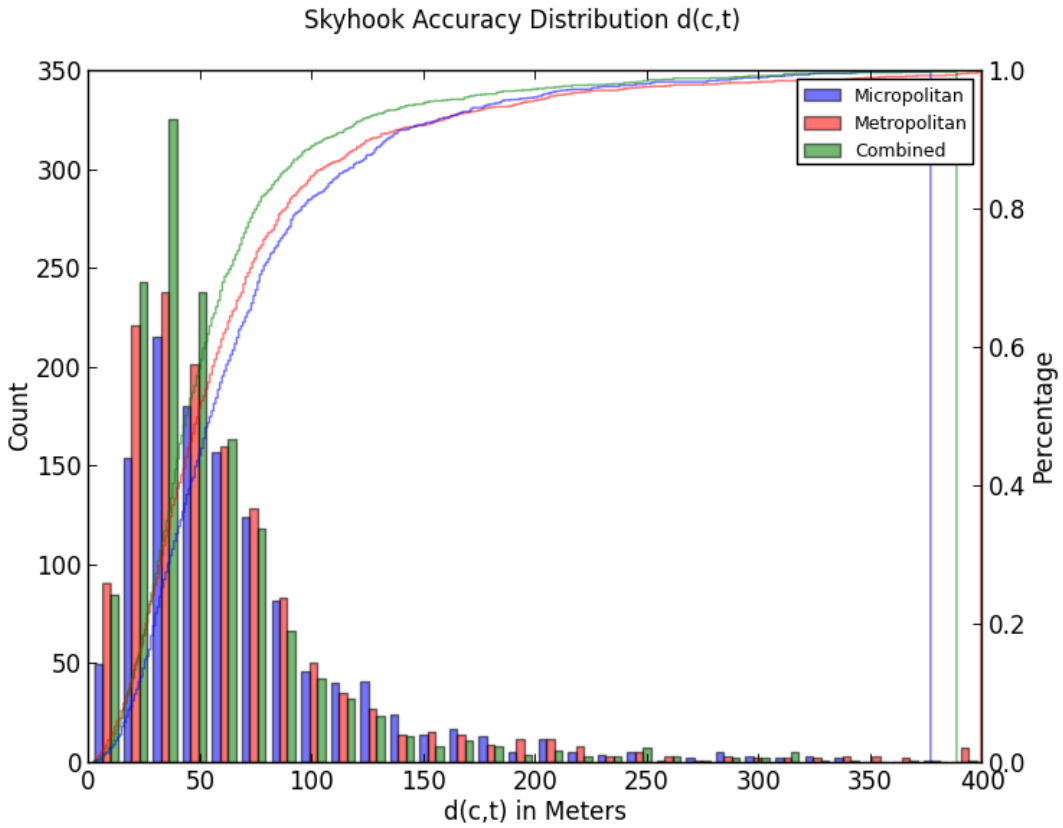


Figure 11. Skyhook service accuracy distribution d(c,t).

Generally, we find all services have highest accuracy for combined statistical areas, followed by metropolitan then micropolitan regions. Next, we consider the relative accuracy of these services per geographic area.

Regardless of service, the majority of responses in the micropolitan class fall within 25–75 m of the target, where 80 percent of responses are within ~100 m of the target (see Figure 12). Google's service achieved best accuracy, measured by both the total number of responses near the target and by the proportion of total responses near the target. Microsoft's service provided the least accurate results in the micropolitan geographic class.

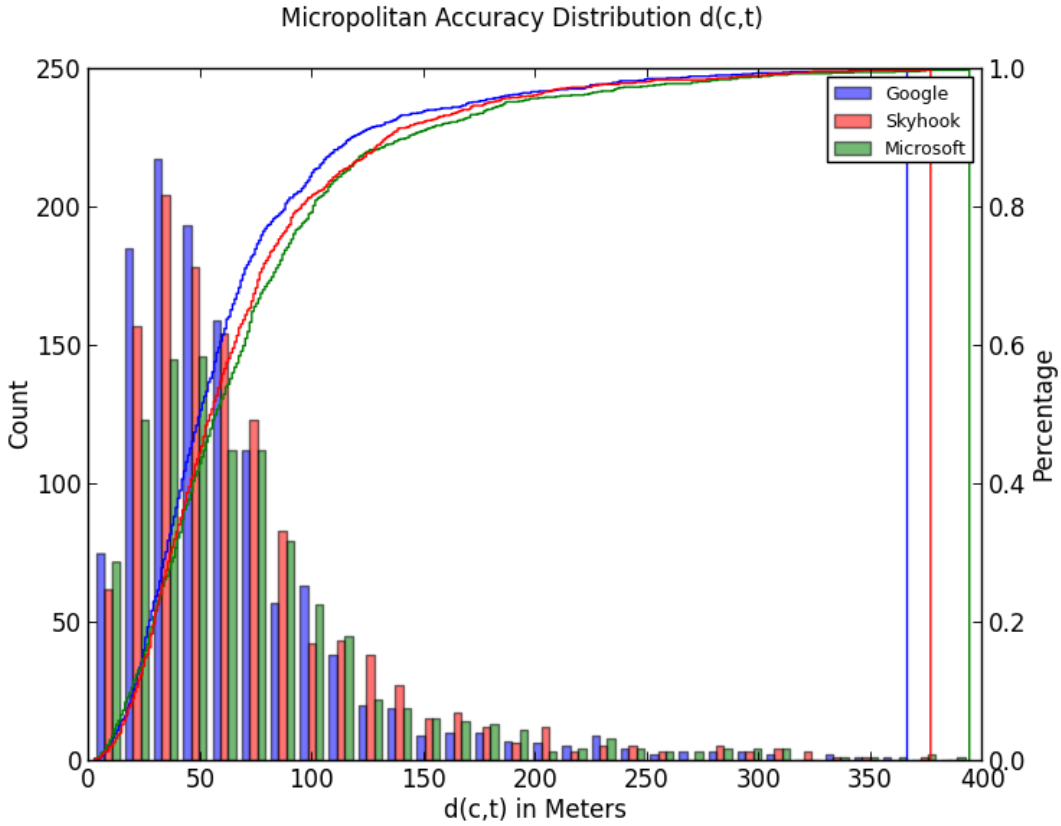


Figure 12. Micropolitan accuracy distribution d(c,t).

Regardless of service, the majority of responses in the metropolitan class fall within 20–75 m of the target, with 80 percent of responses within ~100 m of the target (see Figure 13). By proportion of total responses, we observe Google and Skyhook to share best accuracy in the metropolitan class. By total number of responses within 75 m of the target, we find Skyhook out-performs Google. By most measures, Microsoft provides the least accurate results for the metropolitan class.

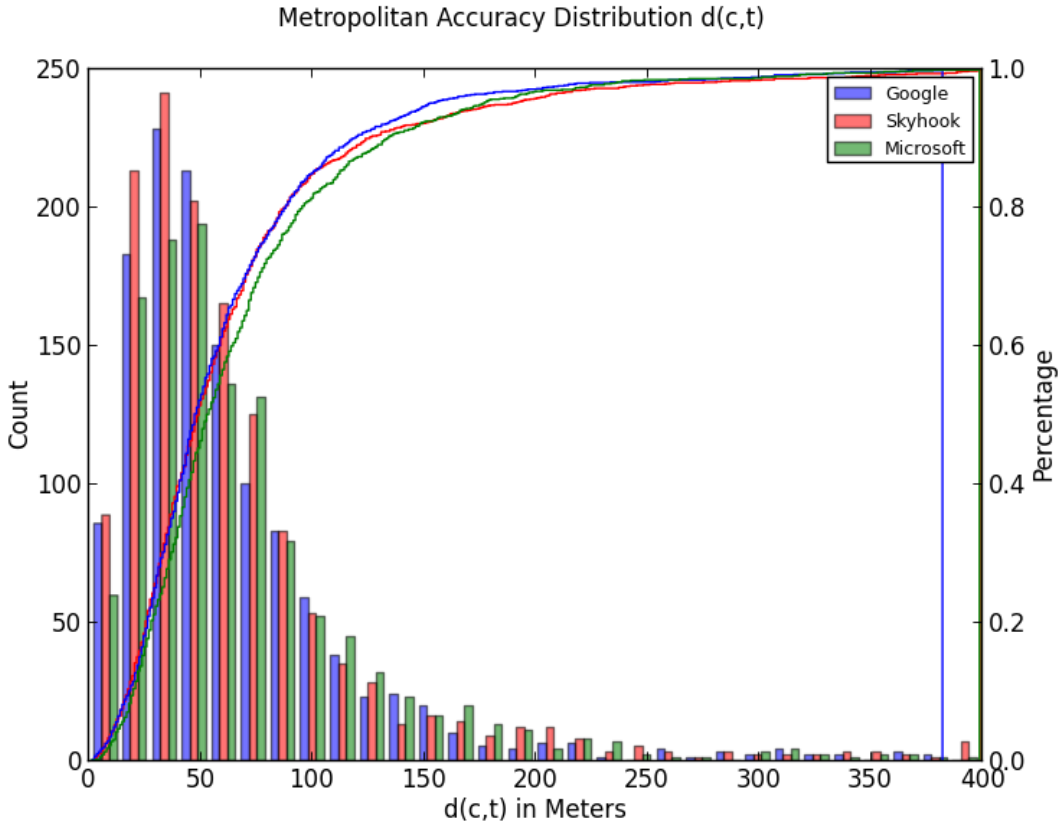


Figure 13. Metropolitan accuracy distribution d(c,t).

Regardless of service, the majority of responses for cities in combined statistical areas fall within 20–75 m of the target, with 80 percent of responses within ~90 m of the target (see Figure 14). For queries in combined statistical areas, we observe Skyhook to have best accuracy, with the most responses within 50 m of the target, and Microsoft to be the least accurate.

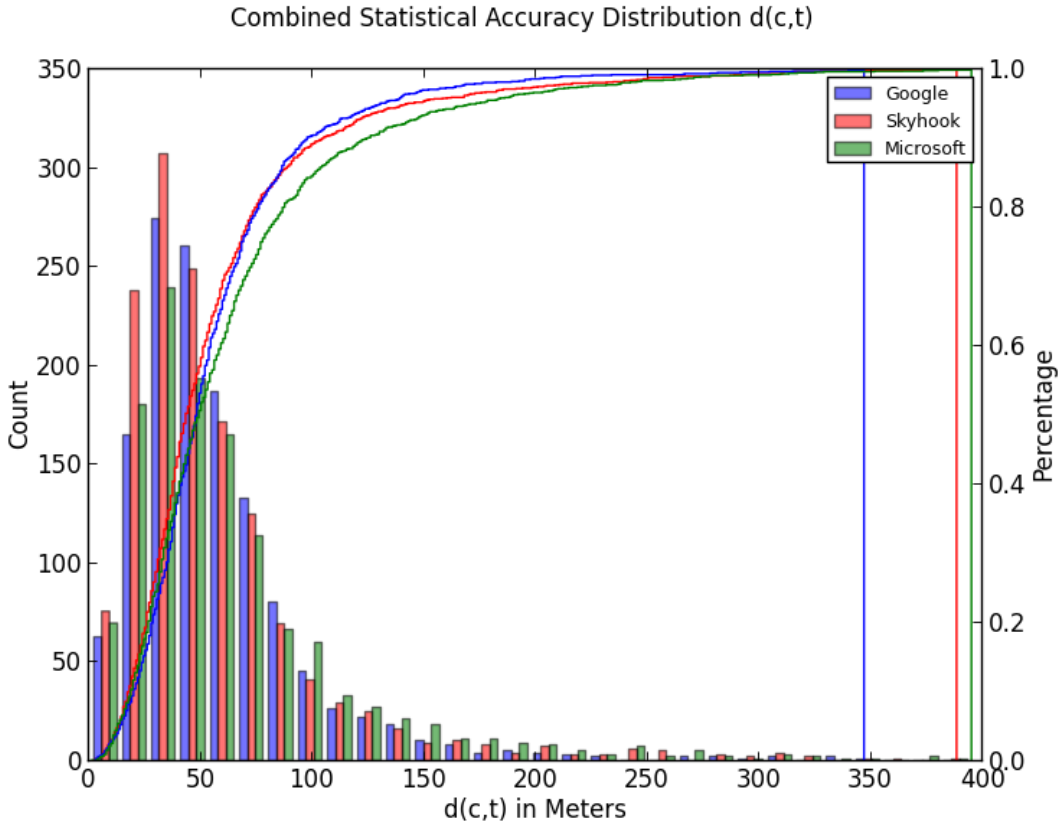


Figure 14. Combined statistical area accuracy distribution d(c,t).

The previous observations (Figures 9–14) ignored "outlier" responses (i.e., those that were farther than 400 m from the target. These outliers account for less than 10 percent of responses; however, we believe they warrant examining in detail. In Figure 15, we plot responses farther than 10,000 m from the target, with details in Table 7. The outliers ranged from 12.7 km to 3,800 km from the target. Most outliers were responses to queries with less than 10 APs. If a household or business moves, relocating their APs, this would likely "confuse" the geolocation service; in this scenario, it is unclear if WiGLE data is out-of-date or if service behavior is out-of-date. Since our corpus is created from temporally-scattered, user-submitted data, any AP relocation may compound this confusion: it is possible for an AP that has moved multiple times to have multiple location entries in the WiGLE database. From a random sample of 75 APs from outlier queries, however, we did not observe any MACs with multiple entries when we queried WiGLE service.

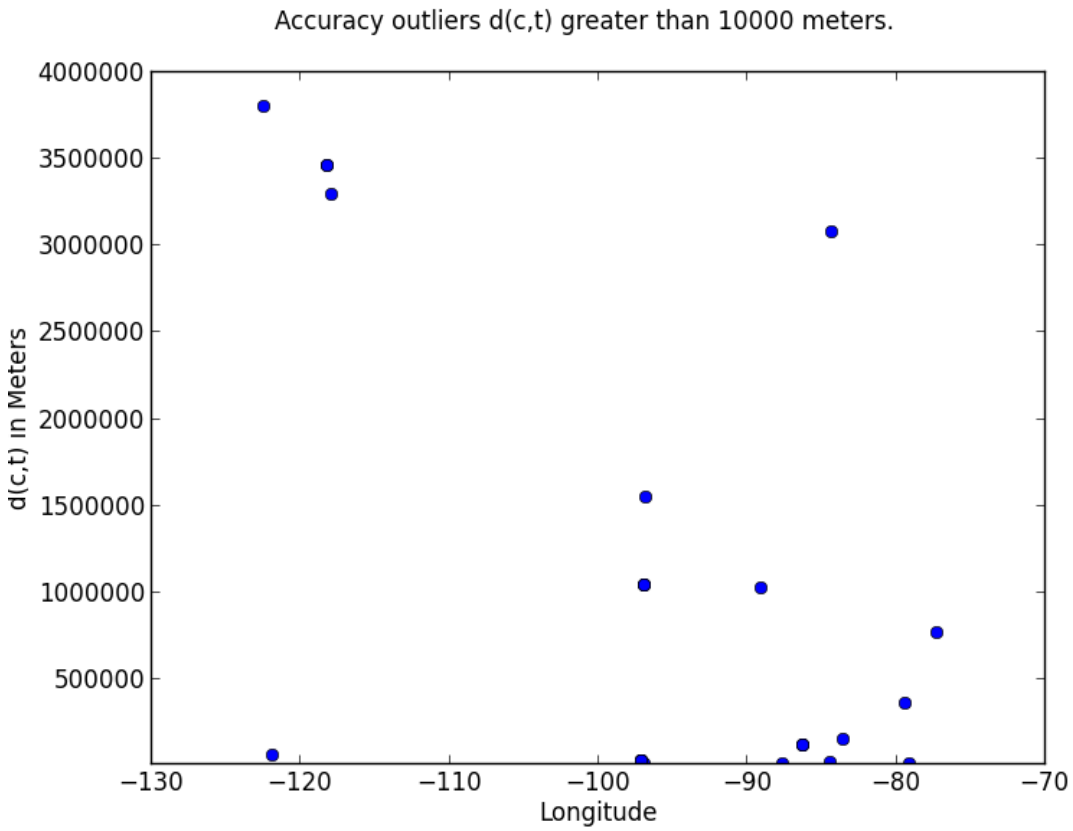


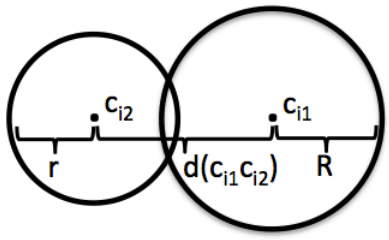
Figure 15. Accuracy "outliers," d(c,t) ≥ 10,000 m.

Geographic Class	Query Length	Google Accuracy	Microsoft Accuracy	Skyhook Accuracy
Micropolitan	3	12703	Failed	46
Micropolitan	10	3075928	240	149
Micropolitan	12	300	Failed	362971
Metropolitan	2	21608	Failed	Failed
Metropolitan	4	116146	116141	116129
Metropolitan	5	1021633	147	Failed
Metropolitan	6	149101	26	162
Metropolitan	100	50	62623	64
Combined Statisical	45	763018	243	246
Combined Statisical	100	3461204	3461247	911
Combined Statisical	9	25983	25986	25983
Combined Statisical	2	3803557	Failed	Failed
Combined Statisical	3	25957	25967	25956
Combined Statisical	7	3293112	54	82
Combined Statisical	8	1043583	1043589	1043562
Combined Statisical	3	Failed	Failed	1546196
Combined Statisical	7	24	Failed	12924
Combined Statisical	9	Failed	13685	Failed

Table 7. Accuracy "outlier," details.

D. INTERAGREEMENT

In this section, we consider service *interagreement* in attempt to measure of the degree to which service behavior agrees with one another. The definition of accuracy used in the previous section was the response's distance from the initial query target, and implicitly assumed the target to be a meaningful landmark. Given our use of user-submitted, geolocated AP data, this was problematic. The intention of measuring interagreement is to relax this, allowing analysis without explicit use of an assumed target location. How to quantify interagreement precisely, however, requires some discussion. Initially, for any two responses, one might consider a metric derived from the intersection of the two responses (see Figure 16). We define the ratio of the intersection to the total area represented by the two responses as *Case-1 Interagreement*. This metric is symmetric and ranges from zero (no intersection) to 0.5 (entirely overlapping areas).

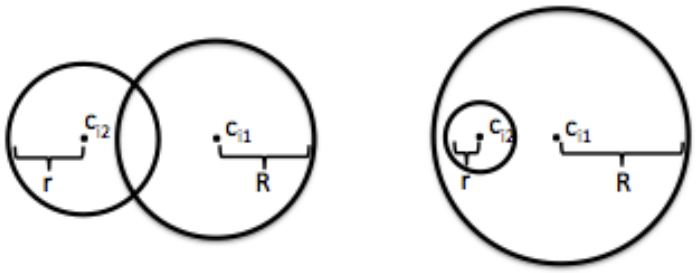


Case 1: if $(d(c_{i1}c_{i2}) + r) > R$ and $d(c_{i1}c_{i2}) < (r+R)$
R=reported "accuracy" of response c_{i1}
r= reported "accuracy" of response c_{i2}
d()= distance function
a()= area function
Interagreement Ratio= $a(c_{i1} \cap c_{i2})/(a(c_{i1}) + a(c_{i2}))$

Figure 16. Case-1 Interagreement metric

There are scenarios where this simplistic metric appears inadequate or misleading. For example, one such scenario is when a circle lays inside another circle: if the inner circle response has high precision (a small radius), the intersection is small and yields a Case-1 interagreement that is equal to the scenario where two responses have a relatively small overlap (see Figure 17). We separate the case of nested circles, analyzing these

using a separate *Case-2 Interagreement* metric (see Figure 18). Case-2 Interagreement is defined by the ratio of the two circle radii, r/R where R is the radius of the outer circle. This is a symmetric metric, ranging between zero and one, with zero indicating an inner radius of zero and one indicating the inner and outer radii are equal.

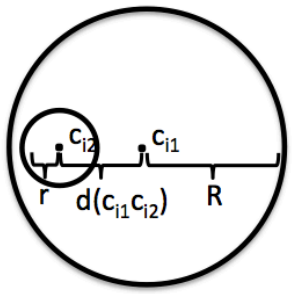


$a(c_{i2})=2$
$a(c_{i1})=3$
$a(c_{i1} \cap c_{i2})=1$
IR = 1/(3+2)=.2

$a(c_{i2})=1$
$a(c_{i1})=4$
$a(c_{i1} \cap c_{i2})=1$
IR=1/(4+1)=.2

a()= area function
Interagreement Ratio (IR)= $a(c_{i1} \cap c_{i2})/(a(c_{i1}) + a(c_{i2}))$

Figure 17. Scenarios motivating multiple interagreement metrics.

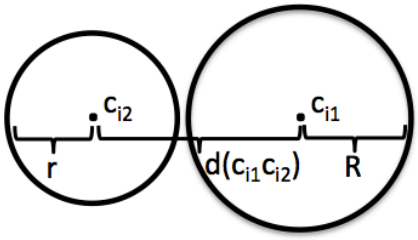


Case 2: if $(d(c_{i1}c_{i2}) + r) <= R$
R=reported "accuracy" of response c_{i1}
r= reported "accuracy" of response c_{i2}
d()= distance function
Interagreement Ratio=R/r

Figure 18. Case-2 Interagreement metric.

Neither Case-1 nor Case-2 metrics characterize the level of disagreement between responses. For example, when the Case-1 Interagreement is zero, one might want a metric

that distinguishes a 50 m disagreement from a 50 km disagreement. The *Case-3 Interagreement* metric is defined as the distance between non-intersecting responses (see Figure 19).



Case 3: if $d(c_{i1}c_{i2}) >= (r+R)$
R=reported "accuracy" of response c_{i1}
r= reported "accuracy" of response c_{i2}
d()= distance function
Distance from agreement= $d(c_{i1}c_{i2})-(r+R)$

Figure 19. Case-3 Interagreement metric.

Finally, we consider service failure scenarios as another type of interagreement. For each pair of services, we consider the number of failures for the individual service and the number of failures shared between the services. We define *Case-4 Interagreement* as a simple 0/1 metric indicating that the failure response is in agreement between the services, and treat non-shared failures as a type of disagreement.

Dividing interagreement into several cases is complex, and becomes a problem for making sense of "the big picture" for interagreement. It was our goal to develop a single metric of interagreement to accomplish this, and considered how to combine these metrics. We decided to give the result of each *pair of services* a value, which we assigned to either an agreement or a disagreement sub-total. Our Case-1 and Case-2 metrics do a good job of characterizing agreement. For Case-1, we double the interagreement ratio (previously ranging 0–0.5) and assign this to agreement, assigning the complement of this to disagreement. For Case-2, we assign the entire value to agreement, and its complement to disagreement. For Case-3, the entire value is assigned to disagreement. For Case-4, if a failure is unique to one service, its value is assigned to disagreement; if it was a shared

failure, then the value was assigned to agreement. We sum the agreement and disagreement values to arrive at agreement and disagreement totals for each service pair. The agreement and disagreement totals will always equal the total number of queries. To arrive at our final summary metric, we normalize each subtotal by the total number of queries. To arrive at an overall average for interagreement between a pair of services, we average the normalized agreement and disagreements across the three geographic classes. We remark that while promising as a first attempt at analysis, this summary statistic should be interpreted with extreme caution.

In Table 8, we summarize the number of occurrences of each case, per service pair and geographic class. Of the 1550 service query pairs per geographic class, we find Case-1 results ranging between 29–43 percent, Case-2 ranging between 26–49 percent, Case-3 ranging between 3–6 percent and Case-4 ranging between 14–37 percent of total queries.

Service Pairs	Geographic Class	Occurances Per Case			
		Case 1	Case 2	Case 3	Case 4
Microsoft/Skyhook	Micropolitan	520	401	89	540
	Metropolitan	638	490	81	341
	Combined Statistical	670	620	86	274
Google/Microsoft	Micropolitan	459	446	74	571
	Metropolitan	587	505	88	370
	Combined Statistical	626	540	75	309
Google/Skyhook	Micropolitan	537	537	56	420
	Metropolitan	549	665	52	284
	Combined Statistical	510	769	55	216

Table 8. Summary of interagreement cases

In Table 9, we summarize details of Case-4 query pairs. We find that while the number of unique failures varies dramatically, the percentage of shared failures remains nearly constant at approximately 50 percent. We will further examine Case-4 as we consider each service pair.

Service Pairs	Geographic Class	Failures		
Microsoft/Skyhook		Unique Microsoft	Unique Skyhook	Shared
	Micropolitan	209	28	303
	Metropolitan	171	23	147
	Combined Statistical	166	14	94
Google/Microsoft		Unique Google	Unique Microsoft	Shared
	Micropolitan	58	252	261
	Metropolitan	52	115	203
	Combined Statistical	49	106	154
Google/Skyhook		Unique Google	Unique Skyhook	Shared
	Micropolitan	89	101	230
	Metropolitan	114	29	141
	Combined Statistical	108	13	95

Table 9. Case-4 details.

In Figure 20, we plot all four metrics (Case-1, Case-2, Case-3, Case-4) for Google/Microsoft service interagreement. In Case-1, 49 percent have less in common than in common (metric is ≤0.25). In Case-2, we observe when service guesses completely overlap, more identify areas that are different in precision (65 percent have r/R ratios < 0.5). In Case-3, we find 56 percent of non-overlapping responses are greater than 50 m away. In Case-4, we observe 49.4 percent of service failures are shared. Proceeding with our summary metric we observe per geographic class, a total agreement (disagreement) of 43.2 percent (56.8 percent) in the micropolitan class, 45.5 percent (54.5 percent) in the metropolitan class, and 45.2 percent (54.8 percent) for the combined statistical areas class. Averaging across classes, we observe 44.6 percent agreement (55.4 percent disagreement) between Google and Microsoft. With no significant and consistent bias to agreement or disagreement we conclude that Google and Microsoft (to some degree) are equally likely to agree or disagree.

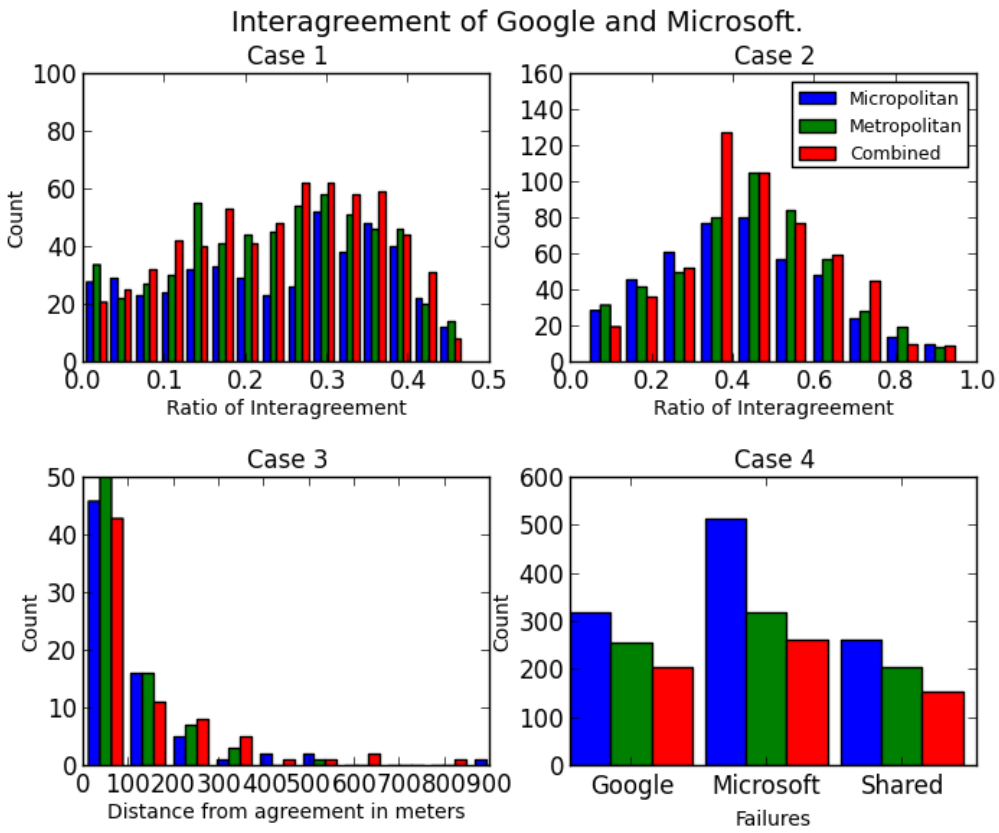


Figure 20. Google/Microsoft service interagreement.

In Figure 21, we plot all four metrics of interagreement between Google and Skyhook. In Case-1, we find 51.5 percent have less in common than in common (metric is ≤0.25). In Case-2, we observe when service guesses completely overlap, more identify areas that are significantly different in precision (72.4 percent have r/R ratio < 0.5). In Case-3, we find 49.6 percent of non-overlapping responses are greater than 50 m away. In Case-4, we observe 50.7 percent of service failures are shared. Proceeding with our summary metric, we observe per geographic class, a total agreement (disagreement) of 45 percent (55 percent) for the micropolitan class, 42.5 percent (57.5 percent) for the metropolitan class, and 38.8 percent (61.2 percent) for the combined statistical area class. Averaging across classes, we observe 42.1 percent agreement (57.9 percent disagreement) between Google and Skyhook. While Case-1, Case-3, and Case-4 indicate equal likelihood to agree or disagree, Case-2 and the summary metric indicate disagreement. From Table 8 we find Case 2 encompasses 42.4 percent of responses in

this service pair. Given the large portion of total responses in Case-2 and the concurrence with the summary metric we conclude that Google and Skyhook are (to some degree) more likely to disagree than agree.

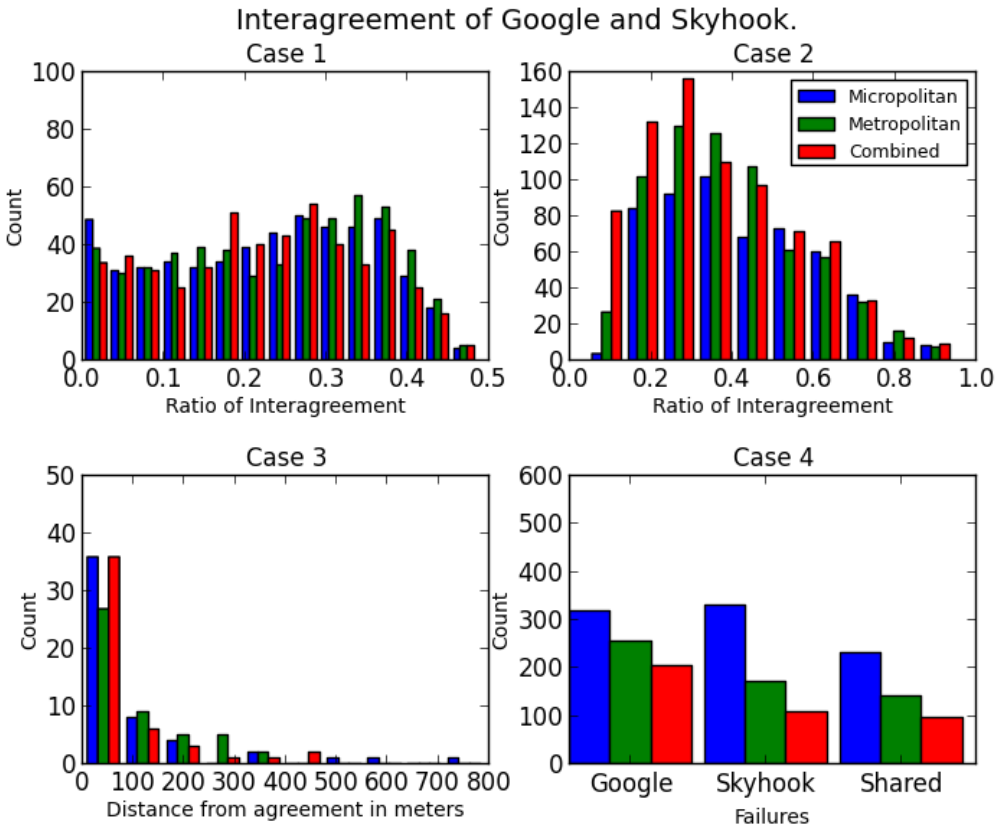


Figure 21. Google/Skyhook service interagreement.

In Figure 22, we plot all four metrics of interagreement between Microsoft and Skyhook. In Case-1, we find 49.9 percent have less in common than in common (metric is ≤0.25). In Case-2, we observe when guesses completely overlap, more identify areas that are significantly different in precision (61.4 percent have r/R ratio < 0.5). In Case-3, we find 52.4 percent of non-overlapping responses are greater than 50 m away. In Case-4, we observe 47 percent of service failures are shared. Proceeding with our summary metric we observe per geographic class, a total agreement (disagreement) of 47.9 percent (52.1 percent) for the micropolitan class, 41.6 percent (58.3 percent) for the metropolitan class, and 38.5 percent (61.5 percent) for the combined statistical area class. Averaging

across the classes, we observe 42.8 percent agreement (57.2 percent disagreement) between Microsoft and Skyhook. With no significant and consistent bias to agreement or disagreement, we conclude that Microsoft and Skyhook (to some degree) are equally likely to agree or disagree.

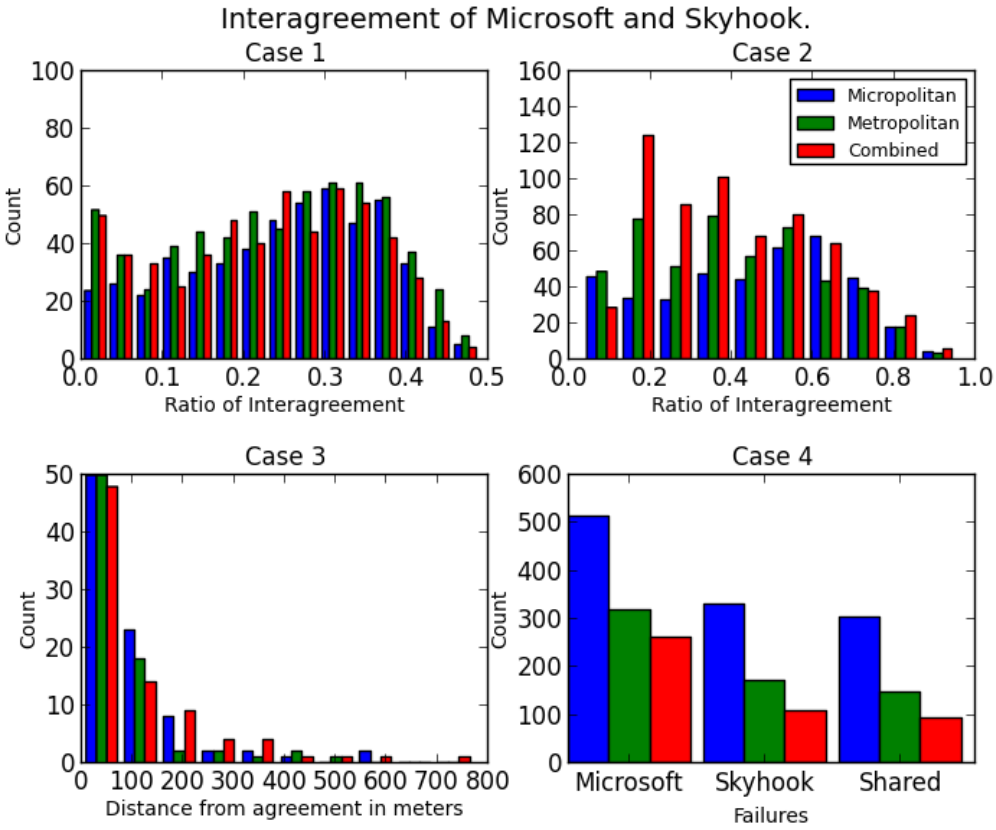


Figure 22. Microsoft/Skyhook service interagreement.

V. CONCLUSION

In this work, we have presented the design and construction of a corpus for testing Wi-Fi Position Systems, using AP MAC addresses derived from the WiGLE database and test cases derived from city classes defined by U.S. Census Bureau data. We employed our query corpus to implement controlled WPS requests to the Google, Microsoft and Skyhook WPS services. In contrast to prior work, our tools are unaffected by environmental conditions or variability associated with native, proprietary service libraries, both of which impact WPS characterization using handheld devices in the field. We propose several metrics expressing "service interagreement," allowing our corpus to characterize service response behavior in the absence of ground truth.

A. FUTURE WORK

Our tests were limited to the Google, Microsoft, and Skyhook WPS services. Future work could expand this survey to include Apple, Navizon and other WPS services. While our corpus allows apples-to-apples comparison between services, the expectation that a useful corpus relate to real-world performance is natural. Comparing results obtained with our corpus and results obtained from a corpus derived from real-world observations ("ground truth") would serve to contextualize our observations.

B. SUMMARY

A significant proportion of our query corpus is relatively uninteresting: 9.4 percent of queries result in failure from all services. In non-failure scenarios, each service gave more than 80 percent of its responses reporting a location guess of no more than 100 meters in radius. As expected, every service demonstrated best performance in cities of densest populations (combined statistical areas). Beyond this, we see significant differences between services, in both their failure and non-failure behavior. Excluding common failures, 4.0 percent of the corpus resulted in failure responses for Microsoft, 8.0 percent for Google, and 16.0 percent for Skyhook. Most failures were shared pair-wise with some other service, but 46.4 percent of non-common failures were unique to some service. On success, the services behaved differently with respect to their reported

precision: Microsoft rarely reported location guesses 20–50 meters in radius, leaving a startling “precision gap.” In comparison, Google results appeared skewed toward guesses with radii in the 20–40 meter range. Skyhook reported better precision in geographic regions with denser populations, while Google’s responses showed similar precision for each geographic region. Considering service interagreement, we find Google/Microsoft and Microsoft/Skyhook equally likely to agree as disagree while Google/Skyhook are more likely to disagree than agree.

LIST OF REFERENCES

[1] Y. Shavitt and N. Zilberman, "A geolocation databases study," *IEEE Journal on Selected Areas in Communications,* vol. 29, no. 10, pp. 2044–2056, Dec. 2011.

[2] P. Zandbergen, "Accuracy of iPhone locations: A comparison of assisted GPS, Wi-Fi and cellular positioning," *Transactions in GIS*, vol. 13, pp. 5–25, Jun. 2009.

[3] C. Yu-Chung, Y. Chawathe, A. LaMarca, and J. Krumm. "Accuracy characterization for metropolitan-scale Wi-Fi localization," in *Proc. of the 3rd international conference on Mobile systems, applications, and services (MobiSys'05)*, Seattle, WA, 2005, pp. 233–245.

[4] C. Post and S. Woodrow, "Location is everything: balancing innovation, convenience, and privacy in location-based technologies," MIT, Boston, MA, Dec. 2008, http://goo.gl/vDYph8

[5] R. Henniges, "Current approaches of Wi-Fi Positioning," TU-Berlin, Berlin, Germany, 2012, http://goo.gl/owQjLi

[6] A. LaMarca, J. Hightower, I. Smith, and S. Consolvo, "Self-mapping in 802.11 location systems," in *Proc. of 7th International Conference on Ubiquitous Computing (UbiComp 2005)*, Tokyo, Japan, 2005, pp. 87–104.

[7] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, Oct. 2000.

[8] T. Chen. (2010). "Location Determination for Mobile Devices, An Overview of Google Location Services" [Online]. Available: http://goo.gl/aNVW3k.

[9] M. Youssef, A. Agrawala, and A. Shankar, "WLAN location determination via clustering and probability distributions," in *Proc. of the First IEEE Pervasive Computing and Communications (PerCom 2003)*, Fort Worth, TX, 2003, pp. 143–150.

[10] J. Letchner, D. Fox, and A. LaMarca, "Large-scale localization from wireless signal strength," in *Proc. of the Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference*, Pittsburg, PA, 2005, pp. 15–20.

[11] P. Barsocchi, S. Lenzi, S. Chessa, and G. Giunta, "A novel approach to indoor RSSI localization by automatic calibration of the wireless propagation model," in *Proc. of the 69th IEEE Vehicular Technology Conference (VTC Spring 2009)*, Barcelona, Spain, 2009, pp. 1–5.

[12] T. Pulkkinen, and P. Nurmi, "AWESOM: Automatic discrete partitioning of indoor spaces for Wi-Fi fingerprinting," in *Proc. of the 10th International Conference on Pervasive Computing*, Newcastle, UK, 2012, pp. 271–288.

[13] A. Haeberlen, E. Flannery, A. Ladd, A. Rudys, D. Wallach, and L. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks," in *Proc. of the 10th annual international conference on Mobile computing and networking (MOBICOM 2004)*, Philadelphia, PA, 2004, pp. 70–84.

[14] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. of Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, Tel Aviv, Israel, 2000, pp. 775–784.

[15] R. Leiteritz. (27 Apr 2010). "Copy of Google's submission today to several national data protection authorities on vehicle-based collection of Wi-Fi data for use in Google location based services." [Online]. Available: http://goo.gl/g01sCd

[16] (2012). "Navizon app—Technical details." [Online]. Available: http://goo.gl/ZUTvoJ

[17] W. Ho, A. Smailagic, D. Siewiorek, and C. Faloutsos, "An adaptive two-phase approach to Wi-Fi location sensing," in *IEEE Conference on Pervasive Computing and Communications Workshops (PerCom Workshops 2006)*, Pisa, Italy, 2006, pp. 452–456.

[18] (Mar. 8, 2014). "WiGLE Frequently Asked Questions." [Online]. Available: https://wigle.net/gps/gps/main/faq/

[19] "WiGLE" (Mar. 8, 2014). *Wikipedia, the Free Encyclopedia*, [Online]. Available: http://en.wikipedia.org/wiki/WiGLE_(WiFi).

[20] Office of Management and Budget, (Feb. 2013), "OMB Bulletin No. 13-01." [Online]. Available: http://www.census.gov/population/metro/data/omb.html

[21] U.S. Census Bureau (Mar. 8, 2014), "Metropolitan and Micropolitan Delineation Files." [Online]. Available: http://goo.gl/stGxBa

[22] MaxMind Incorporated, "Free World Cities Database." [Online]. Available: http://www.maxmind.com/en/worldcities/ [12 July 2013].

[23] E. W. Weisstein, “Square Point Picking,” *MathWorld—A Wolfram Web Resource*, [Online]. Available: http://goo.gl/bgu0y2

[24] coderrr. (Sept. 10, 2008), “Get the physical location of wireless router from its MAC address (BSSID).” [Online]. Available: http://goo.gl/hCe6Py

[25] achillean. (Mar. 8, 2014). “WiFi Position System (source code).” [Online]. Available: http://goo.gl/V0EYD3

[26] I. Mccracken. (2008) “Maclocate (source code).” [Online]. Available: http://goo.gl/QQ2Ce5

[27] StackOverflow. (Mar. 8, 2014). “JSON, CURL and Google’s geolocation.” [Online]. Available: http://goo.gl/jp9j3k

[28] E. Bursztein. (Aug. 4, 2011). “OWADE (source code).” [Online]. Available: http://goo.gl/lKZriA

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California