CGSIL: A Viable Training-Free Wi-Fi Localization

Han N. Dinh, Thong M. Doan
University of Science, Vietnam National University
Ho Chi Minh, Vietnam
jennynguyen.jn293@gmail.com; dmthong@apcs.vn

Nam T. Nguyen
University of Information Technology
Ho Chi Minh, Vietnam
namnguyen@uit.edu.vn

***Abstract* –Localization for indoor environment normally does not use GPS signals since it cannot penetrate through walls and buildings. Instead, many works have focused on using Wi-Fi signals as the mean to locate the position of the mobile devices. However, most of these approaches require a training step to build a Wi-Fi's map for each location. This requirement practically prevents these approaches from being realistic, since the training step is extremely time-consuming (hundreds of labor hours). Recently, ISIL has been proposed as the first Wi-Fi-based technique that is training-free, in which the localization can be done instantly at any location without the need of training and building Wi-Fi map. ISIL collects from the web the related information of all observable access points and infers the current position based on that. As the first search-based Wi-Fi localization, ISIL removes the unacceptable time-consuming training step. However, it still does not provide adequate accuracy due to the lack of exploiting regional correlation of information returned by the search engine. In this paper, we proposed CGSIL, another kind of search-based Wi-Fi localization that provides the accuracy level of nearly twice as much as ISIL by collaborative filtering and clustering geographic information collected from the search engines. Through experiment results, CGSIL proves to be a feasible replacement for future indoor localization due to its high accuracy and reasonable cost.**

Keywords—search engine; clustering; regional relationship.

I. INTRODUCTION

Localization is becoming an essential technique to enable any useful service, such as Google Maps, Facebook and other services [1]. Several localization techniques have been proposed recently using Global Positioning System (GPS) [2], cellular [3]-[5], and Wi-Fi [6]-[14] technologies. GPS-based localization can achieve the accuracy of up to a few meters [2]. However, in GPS, the signals are transferred from the satellites to a device, and thus the signals can be weakened by obstacles. This explains why GPS can only be used for outdoor environment. Approaches using cellular technology [3]-[5] can work for both outdoor and indoor locations (covered by cell towers) but offer low accuracy (several hundred meters). They also require the knowledge of cell towers' map. Recently, many approaches using Wi-Fi (802.11) signals [6]-[14] have been proposed for indoor locations thanks to their high accuracy rate and the increasingly popularity of the 802.11 Access Points (APs).

According to Le et al. [15], Wi-Fi based localization algorithms can be divided into five main categories: range-based, range-free (centroid [6][7]), aggregate and singular, scene matching (fingerprint [8]) and SIL (search-based) [15]. In the first four categories, one common step these algorithms all require is the costly training phase. In this step, some known positions in the network are recorded with their coordinates and associated information. This information map is used to estimate the location in the runtime phase. The biggest challenge of this training step is that it requires a lot of time and physical-labor. Additionally, this step needs to be repeated regularly to adapt to environment changes.

To avoid the costly training phase, Search-based Indoor Localization (SIL), the 5th category, is proposed. The first algorithm in this category is ISIL [15]. ISIL eliminates the need of the costly training step by exploiting nearby observable access points' names at the runtime phase. The algorithm utilizes what the APs' names represent (usually the business) and aggregates the information to predict the device's current position.

However, ISIL does not exploit the geographical relationship between nearby APs; thus it leads to low accuracy when presenting the predicted address to users. Additionally, to increase the accuracy, ISIL presents a list of 16 possible addresses for users to choose from manually. This approach is not user-friendly and prevents automatic localization since it requires explicit user feedback. Another problem is the lack of a ranking strategy for multiple collected addresses on the same street; therefore, ISIL can only return predicted address with up to street name (no street number). In other words, it cannot provide fine-grained result up to street number.

In this paper, we present CGSIL, a Collaborative Geo-clustering Search-based Localization that provides an accuracy level that is two times better than ISIL. In Section II, we will review and categorize the existing Wi-Fi localization algorithms. In Section III, we describe our new approach, CGSIL, and its advantages. In Sections IV and V, we discuss the experiment setup and analyze the experiment results of CGSIL. To prove the practical aspect of CGSIL, we also provide a cost analysis in terms of storage and bandwidth usage in Section V.

II. RELATED WORKS

According to Le et al. [15], Wi-Fi localization techniques can be classified into four categories: range-based, range-free, aggregate and singular, scene matching. All of them require a costly training phase, in which some known positions in the network are recorded with their coordinates and associated information. This information map is then used to estimate the location when the algorithms are in the runtime phase. Recently, ISIL has been introduced as the first training-free localization algorithm [15]. Due to the basic nature of the training-free solution, we classified ISIL to belong to the new category, called SIL. In the next sections, we will first summarize the first four

 ISBN: 978-1-61208-353-7

categories and then have a brief review about SIL, the new category.

A. The first Four Categories (Training-Required Group)

Most of the Wi-Fi localization techniques in the first four categories have two main phases: a training (offline) phase and a deployment (online) phase [15]. The main task of the training phase is to build a map containing known location indicators. These indicators are then used in the deployment phase to estimate the location by retrieving the most appropriately similar location indicators from the pre-built map.

Technically, the training phase could vary depending on the unique property in each category. However, in most cases, this phase requires an extensive amount of time and human labor to accomplish, as the location indicators must be collected at every location. Additionally, this costly training step must be repeated regularly due to the changes of the environment (weather, human, building). Finally, if devices used in the deployment phase are different from the sample devices used in the training phase, the accuracy can degrade remarkably [8]. Re-training for new devices will improve the accuracy but it is time-consuming and impractical for wide-scale deployment due to the variety of mobile devices [15][21][22].

B. SIL (Training-Free Group)

SIL is a Wi-Fi based localization approach that aims to remove the need of the costly training step (ISIL [15] is an example in this group). By analyzing the SSIDs of observable APs collected at a location, SIL will aggregate the information related to the SSID to predict the device's current position. The information related to the SSID can be extracted instantly by querying any search engines [15] or other means. SIL is a simple alternative for indoor localization where GPS signals are not available and when it is nearly impractical to require the training step.

Fig. 1 illustrates the general framework of SIL (used in ISIL [15]). It is composed from three components: Scanning, Geo-information Retrieving and Address Processing. In the Scanning component, the mobile device will scan for information extracted from nearby APs. Next, the Location Geo-information Retrieving component will gather relating information from the Internet and extract a list of potential addresses. Finally, the Address Processing component will rank the addresses and return the correct ones to the users. In this component, we can apply different algorithms with different strategies to process and evaluate the list of potential addresses. One example of such algorithm is ISIL [15].

At first, ISIL is a novelty algorithm due to its training-free properties. ISIL works independently on the type of wireless card of mobile devices, and is not affected by environmental changes. It can work on any Wi-Fi based mobile device that has access to a search engine [15].

Even though ISIL does not require training step, its accuracy is up to street name only, which causes considerable distance error, since some streets can be several kilometers in length. Moreover, ISIL returns result as a list of predicted addresses and requires user to select one manually. In other words, if the size of the returned address list is small (like 1 or 2), the accuracy rate is low (50% to 55%) [15]. If the size is larger, the number of returned addresses could easily confuse the users.

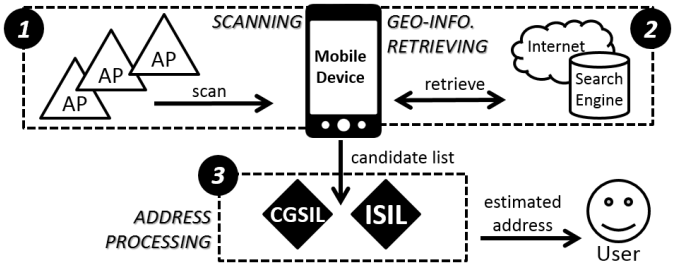

Fig. 1. The General Framework of SIL

To address those constraints of ISIL, we propose CGSIL, a more accurate and finer-grained result than ISIL. Specifically, CGSIL has incorporated Search Engine Optimization property, geographic information and region-based relationship of APs into a comprehensive strategy to predict addresses. Thus, CGSIL only needs to return the result as a single address with the accuracy that is 2 times better than ISIL.

III. OUR APPROACH

In this section, we will give an overview of SIL, ISIL and its weakness. Finally, we propose CGSIL, our new approach.

A. Overview of SIL

SIL relies on the observation that the names of the APs located at a location often contain information relating to that location. For instance, if an AP with the name TokyoDeli is detected, it is a good indicator telling us that our current position is nearby one of the TokyoDeli restaurants. Thus, if SIL can analyze all the SSIDs of observable APs, it can extract the information linking to the user's current position. By aggregating all information returned by the names of all APs, SIL can predict the location of the device. Continuing with the previous example, if we can detect another AP with the name McDonald, it means that the current location must be around McDonald and TokyoDeli restaurants. Thus, if we could find a location that is geographically close to both restaurants, we can use it as the current predicted address.

To do that, SIL needs a database containing the APs' names and their corresponding location. A valuable and always-on database SIL can use is a search engine. As most mobile devices have access to the Internet, querying search engine is totally feasible. The system can feed the AP's name into the search query. The webpages returned from the search engine are parsed to extract all the addresses presented on these webpages. These collected addresses are aggregated and examined to predict the location [15]. The main idea of SIL can be summarized into three phases, corresponding to the three components in Fig. 1.

1) Scanning

In this phase, the deploying device scans nearby APs for their SSIDs. These SSIDs are then pre-processed and split into keyword for querying search engine.

2) Geo-information Retrieving – GR

The keywords extracted from the scanning phase are sent to a search engine. Relevant URLs returned by the search engine are parsed to collect possible location information (addresses).

 ISBN: 978-1-61208-353-7

Since the search engine may return many results (pages), it is impossible to parse them all. Therefore, the top web page results returned by the search engine are selected to parse for location information. The number of selected pages directly affects the breadth of the search space and thus is defined as *breadth*.

The set of webpages returned directly by the search engine is called at *depth* 0. In many cases, it is not sufficient to parse only the webpages at *depth* 0 because the street address of the location may be a few links away. Thus, SIL needs to follow the links appearing on webpages at *depth* 0 to get to subsequent webpages. The successive pages that are one link away from the pages at *depth* 0 are called pages at *depth* 1. We defined *depth* as the number of links away from the pages returned directly by the search engine.

The deeper we crawl for the URLs, the longer it takes for the system to process. The same is for the *breadth*. Therefore, *depth* and *breadth* are the two vital factors we need to analyze to find the optimal values. SIL has shown that *depth* 1 is good enough for the system to return acceptable accuracy [15].

The outcome of this GR phase is a list of potential addresses with high probability to be near to the actual location. This list is defined as the *candidate list*. To point out the correct address from this l*ist*, SIL utilizes the Address Processing component, which is discussed next.

3) Address Processing

From the previous phase, we now have a list of candidate addresses where one of them could be in a close proximity with the actual address. Therefore, the task of this component is to find that address and return it to the users. The performance of SIL greatly depends on the algorithm chosen for this Address Processing component. ISIL is the first algorithm proposed [15]. In the next section, we will describe ISIL in detail.

B. ISIL and its Drawbacks

1) ISIL

Let us define:

- $A = \{ap_1, ap_2, \dots, ap_n\}$: as the set of all access points at one location.
- $extract(x), (x \in A)$: as the function to return all addresses extracted from an access point *x*.

Let *D* be the set of all addresses collected at one location. From the set *A* & function *extract*, we have:

$$D = \bigcup_{i=1}^{n} extract(ap_i), ap_i \in A \quad (1)$$

Finally, we have *S(y)*, the set of all access points belonging to an address (*y*) is constructed by the following function:

$$S(y) = \{x | \, x \in A \wedge y \in extract(x)\}, y \in D \quad (2)$$

According to ISIL algorithm, the authors used two metrics to measure the relevancy of each collected address:

- |*S(y)*|
- The *depth* of the web page where the address appears.

In other words, the ranking of ISIL is based on the following observation:

- If an address is extracted from the search result of more APs, it is more likely to be related to the current location;
- If an address appears in a web page that is further away from *depth* 0, it is less likely to be related to that location.

2) Drawbacks of ISIL

The accuracy of ISIL only works well at street level. The biggest drawback of this is some streets can be very long (10 – 20 km), which negatively affects the accuracy. The second drawback is that two different streets can be in a close geo-proximity, but in ISIL, they will be treated to be unrelated when doing the ranking. Third, ISIL returns the predicted result as a list of possible addresses that requires the user to choose from. This may confuse the user if the list is long. From our experiments, ISIL may return a list of 16 addresses in order to achieve the accuracy level of 80% or more. It is not user-friendly and troublesome to return multiple options for the user to select.

ISIL does not fully exploit the geographic relationship of the APs. In fact, the way the APs in close proximity support each other could be a hint to improve ranking strategy. If an address geographically belongs to the intersected region of more nearby APs, the address is likely to be nearby the current position.

In this paper, we propose CGSIL to address these limitations of ISIL.

C. CGSIL

To address the drawbacks of ISIL, we propose CGSIL, which returns finer-grained and more accurate localization result. This achievement utilized popular technique such as search engine optimization (SEO) [16][19], geographic mutual-relationship, collaborative filtering and cluster analysis [18].

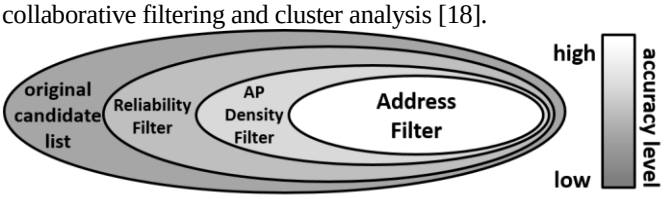

Fig. 2. Venn Diagram of the Filters in CGSIL

Fig. 2 illustrates how we apply different filters in CGSIL to predict the address. The candidate list is narrowed down after each filtering and the accuracy level of the remaining addresses eventually increase.

The detailed process of how CGSIL works can be described in 3 steps: 1) un-related addresses will be filtered out by the Reliability Filter; 2) The AP Density Filter, based on the visibility of surrounding access points, will try to detect a set of addresses having high likelihood to be close to the current location. 3) The Address Filter will rank the addresses and return the top one as the result.

1) Reliability Filter

After the Scanning and GR phases, (Section III.A.2), we now have the *candidate list* composed from addresses extracted from the SSID' search results. However, many addresses from this list are unrelated as they come from irrelevant webpages, such as advertising sites or personal blog-sites. Therefore, CGSIL will use this Reliability Filter to eliminate unwanted addresses.

This filter works by utilizing SEO presentation, embedded inside each web page. SEO is the process of affecting the

 ISBN: 978-1-61208-353-7

visibility of a website or a page in a search engine's results returned to users [17]. The SEO presentation of a web includes: the header text, the footer text, the contents, the codes and the URL itself. Among these SEO attributes, CGSIL will focus on the URLs (the anchor texts) [19] because they often provide more accurate descriptions of Web pages [16]. This observation is utilized in CGSIL to select the pages most relevant to the source SSIDs. Moreover, choosing the URL over other SEO attributes improves performance since processing one line of text is more light-weighted than processing the whole page's content.

If the hyperlink text of one URL does not contain its SSID, the URL and its extracted addresses will be removed from the *candidate list*. After that, the remaining addresses in the list will be sorted in the way that the URL containing more characters from its SSID has higher order than the one containing few characters from SSID; the top addresses in this list are defined as the *F1-candidates*.

However, as displayed in Fig. 3, these F1-candidates ("diamond markers") could be scattered on the geographic map. Hence, our next task is to find a region covering most potentially correct addresses, which is discussed in the AP Density Filter.

2) AP Density Filter

This filter works based on the observation that the area covering addresses from most APs is likely to contain correct estimation for the current location. The idea is illustrated in Fig. 3.

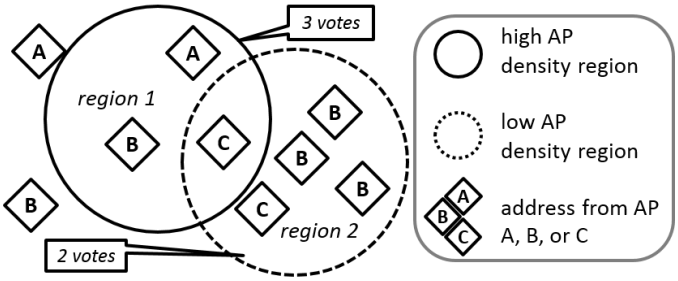

Fig. 3. The Highest Vote Region

Fig. 3 presents a map with 10 scattered addresses and 2 suspected regions that may provide the correct estimations for the current location. For each region, we count the number of votes from the APs. We define a vote from an AP for a region as: the region must contain at least one address extracted from that AP's search results. For example, in Fig. 3, region 1 gets three votes because it contains addresses extracted from three APs ("A", "B" and "C"). Likewise, region 2 gets two votes. This collaborative process chooses the region with the highest votes from all the observable APs. Therefore, the region 1 could be the most likely correct region of the current position.

To find the region of the highest votes, we must have the geographic data associated with each address so that we can perform calculation with the addresses. Such information could be retrieved from any online address database, for example the Google Map, the one we use in our experiment. In addition, we use Google Map API to provide the latitude and longitude coordinates for a given string address. Note that our technique does not depend on any specific map API; for instance, the country's local map API can be selected as an alternative.

Based on the information provided by the Google Map API, we find the region with the highest votes from the APs. If there is one region with the highest vote, we simply return the center of the region as the localization result. Nevertheless, in many cases, there may be multiple regions with the same highest number of votes. These regions could be overlapped or scattered geographically. Thus, we need the Address Filter to estimate the best result to return to the users.

3) Address Filter

The idea of this filter is to find a high-density geographical cluster from multiple same rank regions, discovered from previous step. The center of the result cluster is returned to the users as the localization result.

A cluster is a group of addresses locating at relatively close distance to each other. This problem is classified into typical clustering problem:

- Give a constant d as the maximum Euclid distance between any 2 addresses
- Let A be the set of all *F2-candidate* (all addresses in the *Highest Vote Regions*), C_i be the set of all addresses in one cluster, we have:
$$A = \bigcup_{i=1}^{n} C_i \quad (3)$$
- Define *ed(a, b)* as a function to calculate Euclid distance between 2 addresses a & b, we have:
$$ed(a,b) = \sqrt{(a.x - b.x)^2 + (a.y - b.y)^2} \quad (4)$$
- Finally, we have the condition for any address, called a, to belong to a cluster, called C_i:
$$a \in C_i \leftrightarrow \forall b \in C_i, ed(a,b) \le d(a \in A) \quad (5)$$

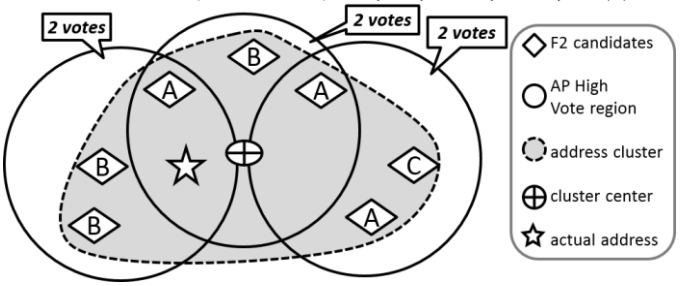

Fig. 4. Clustering and Generating Final Answer for CGSIL

We use condition (5) to distribute all *F2-candidates* into separate clusters. After clustering, the cluster containing addresses from most APs is chosen and its center coordinate is returned as the localization result.

Fig. 4 demonstrates how to cluster the *F2-candidates* to calculate the center of the cluster. Since the 3 circles both cover 2 APs, all 3 circles are considered F2-regions and their addresses are clustered. Satisfying the condition (5), all the addresses are grouped into one cluster (the dark region in Fig. 4). The center of the dark region is returned since the actual address is highly likely to be inside the region.

IV. EXPERIMENT SETUP

In our experiment, we collected data from 4 districts in HCM City which is the same set of districts used in [15]. To increase

 ISBN: 978-1-61208-353-7

the confidence level of the dataset, we collected more than 6,700 locations, which is approximately two times the number of samples collected in [15]. To collect the whole dataset, it took 600 hours of labor.

A. Data Collection

Our Wi-Fi data collection includes around 60 streets in HCM city. On each street, we recorded data at different locations. The collected data includes the AP's name. The exact street number addresses were also recorded for the purpose of evaluating the accuracy of our approach.

The collected data covers District 1 (the city center), 3, 5 and 10. The total street length of our collected data is about 67,500 meters. On each road, we recorded data at different locations, which are 10-15 meters apart from each other. The reason we chose 10-15 meters is that there is not much difference (in terms of observable APs' name) within that distances. The number of locations on a road varies from 40 to 120 depending on its length and availability. At each point, a mobile device continuously scans the Wi-Fi signals for 60 to 90 seconds. On average, there are about 25 APs detected at one location. A group of 150 volunteer students, divided into 60 groups equipped with laptops, participated in the experiment. Each group was responsible for one street. The data set consists of approximately 6,700 locations which take approximately 600 hours of human labor.

B. Accuracy Measurement

To evaluate the accuracy of CGSIL, we recorded actual address at each location (test dataset) to compare with the predicted addresses returned by our algorithm. We defined some terminologies used in presenting results in Section V:

- ***Distance error:*** the Euclid distance between the actual address and predicted address.
- ***Acceptable error range:*** the error range that is acceptable by the users. For example, if the acceptable error range is 500m, that means the users accept the predicted address to be correct if it is within 500m from the actual location.

The accuracy level of CGSIL is calculated as:

$$accuracy = \frac{the\ number\ of\ locations\ yielding\ correct\ address}{the\ number\ of\ all\ locations} \quad (6)$$

V. PERFORMANCE RESULTS

In this section, we will first present the accuracy of CGSIL in comparison with that of ISIL [15]. Next, we analyze how the change in breadth affects the overall accuracy of CGSIL. After that, we will describe Incremental Geo-information Retrieving, used in the Geo-information Retrieving Component of SIL (Section III.A.2), to acquire the information more efficiently. Finally, we study the cost of CGSIL to ensure the feasibility of CGSIL.

A. Accuracy Comparison between CGSIL vs. ISIL

1) Overall Accuracy

Fig. 5 shows the mean localization accuracy of CGSIL and ISIL at District 1 with a variety of acceptable error range.

CGSIL is nearly two times more accurate than ISIL when acceptable range is from 500m or more. This resulted from the collaborative filtering and geographic information clustering implemented in CGSIL. Note that when the acceptable range is 200m, there is not much difference between the two. This is because the Wi-Fi signal normally can cover up to 500m.

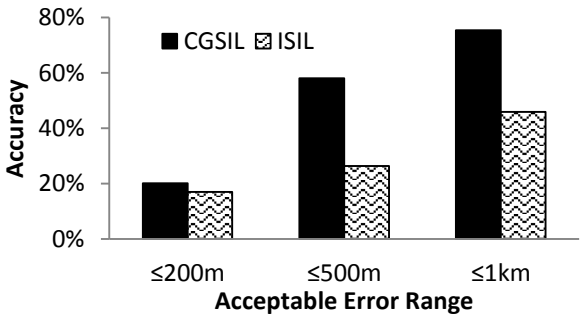

Fig. 5. Accuracy of CGSIL vs. ISIL at District 1 for Variety of Error Range

When the acceptable error range increases from 500m to 1km, the accuracy of CGSIL rises from 58 percent to about 75 percent. For ISIL, to reach this accuracy, it has to return at least 3 candidates for users to choose from.

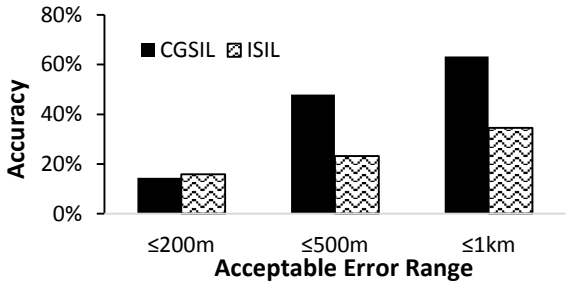

Fig. 6. Accuracy of CGSIL and ISIL at all Districts.

Fig. 6 shows the mean localization accuracy of CGSIL and ISIL for all districts. It has the same pattern as in District 1, but with lower accuracy because it includes non – business districts. This will be discussed more in the next section, V.A.2.

2) Accuracy with respect to Districts

Fig. 7 shows the mean localization accuracy at 4 different districts (acceptable error range is 1 km). The highest accuracy is seen in District 1, which is about 75 percent. The accuracy is lower for District 3, 10 and 5. The accuracy level of these districts is correlated to the business density of the corresponding districts [15]. Crowded business districts tend to yield higher accuracy due to the availability of more APs from nearby business. This is consistent with the finding in [15].

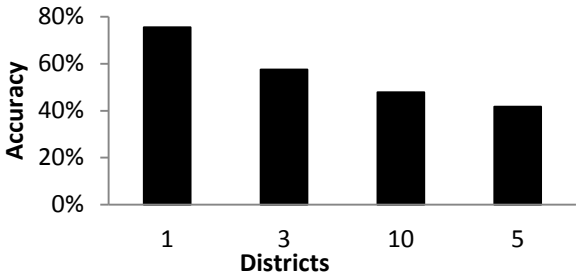

Fig. 7. CGSIL Accuracy in Different Districts (Error Range 1 km)

 ISBN: 978-1-61208-353-7

B. Incremental Geo-information Retrieving – IGR

In this section, we discuss an optimization: IGR. As discussed in III.A.2, whenever mobile device moves to a new location, it must scan the names of all nearby APs and uses those to retrieve the geo-information for localization. This retrieving step requires fetching HTML pages. From our experiment, there are about 25 APs detected at each location on average. Thus, this process may create overhead on bandwidth usage if many APs are detected at each location.

However, adjacent locations are usually covered by many common APs due to the overlap coverage. In other words, when moving from a location to a new one, the mobile device may observe many APs but most of which were previously seen at the old location. From our experiment data, the number of newly detected APs at the new location is only about 2 Aps (out of a total of 25 APs).

Therefore, once moving to a new location, CGSIL only needs to retrieve geo-information for the newly detected APs. This mechanism is called Incremental Geo-information Retrieving (IGR). By doing this, we diminish the bandwidth usage of the device tremendously.

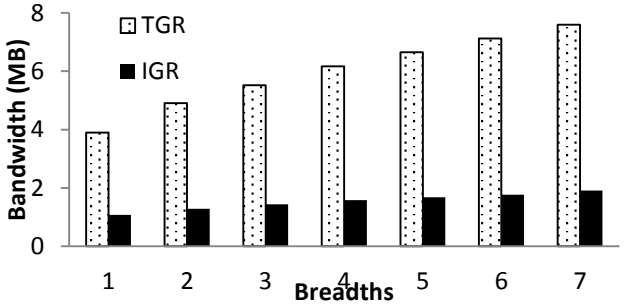

Fig. 8. Bandwidth Usage for Different Level of Breadths.

Fig. 8 shows the bandwidth usage for different breadth levels when using IGR vs. traditional geo-info retrieving (TGR). From the figure, IGR decreased the bandwidth usage by four times comparing to TGR. At breadth 1, IGR used up about 1MB, whereas in TGR, it is about 4MB. Furthermore, when the breadth level increases, the bandwidth usage of IGR rises up slowly from 1MB to 2MB, whereas in TGR, it increases hastily from 4MB to 7.5MB. Note that, the bandwidth usage can even further reduce by using local or cloud storage, which will be discussed in the next section.

C. Cost Analysis of CGSIL

In this section, we will analyze the cost of deploying CGSIL in term of bandwidth cost and storage.

Fig. 9 [20] illustrates the cost mobile users pay per megabyte over the years. The y-axis is in log scale. The x-axis represents the years. In this figure, we see that the cost per megabyte decreases exponentially in prices. With the introduction of 4G, we expect the price will go down in the same trend for 2015 and later.

Thus, if each location requires 2MB of bandwidth to localize (Section V.B), the cost is 0.01 USD/location for 2014. If a user uses CGSIL to localize 100 times/day, the cost that user has to pay for CGSIL is 1 USD/day. However, if the future cost of bandwidth keeps decreasing at the same rate as in the last 4 years, the expected cost of CGSIL can go down to 0.04 USD/day in 2018 and 0.008 USD/day in 2020, which is a negligible quantity. It means that in the next three years, the expected cost for CGSIL is small and affordable for everyone.

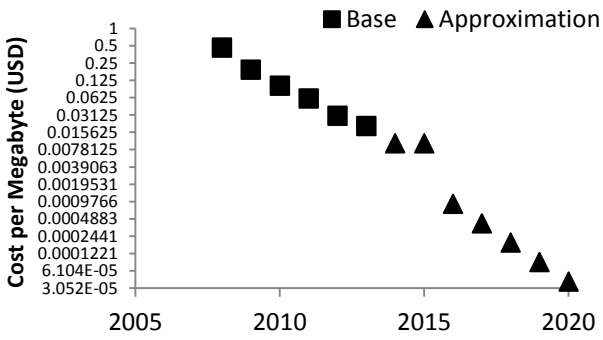

Fig. 9. Cost per Megabyte of Mobile Data worldwide.

s

Moreover, the above estimated cost assumes that the user always moves to the new locations and never go back to any previously visited locations. But, in fact, users are in the habit of moving to the same set of places most of the time: home, office, etc. In that case, if geo-information of visited APs are saved on cache, CGSIL does not need to use bandwidth anymore when users go back to the place they visited before. In other words, after using CGSIL for a few weeks, the users may not need to pay for bandwidth usage or very little.

Note that even though we need to fetch 2MB of HTML files to extract the geo-information, the actual geo-information collected afterward is about 2.5KB in size. Thus, if this geo-information are pushed to the cloud and shared between users, it can be fetched by other user at a rate of 2.5KB/location instead of 2MB/location, which will reduce the bandwidth usage almost 1,000 times (0.1cent/day for 2014).

The storage requirement to implement the geo-information cache at local device is also small. On average, one AP takes about 2.5 KB of storage to save the geo-info on cache. With 100MB cache, the total locations can be cached is about 40,000 locations. Additionally, as the phone storage keep increasing every year, the cache capacity can grow accordingly to hold even more locations if needed.

Fig. 10 illustrates the storage capacity of a common brand phone over time. We see that the capacity jumps double every 2 years. Therefore, a 100 MB of cache on a 64-GB phone takes about 0.015% of its memory, a negligible quantity. With the increment of storage size trend, in the next three years, the expected cache containing geo-info of billion APs is feasible.

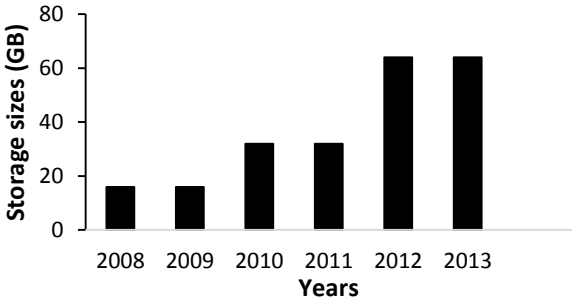

Fig. 10. Phone's Storage Capacity over time

 ISBN: 978-1-61208-353-7

Therefore, we believe that CGSIL is a feasible solution in term of monetary, bandwidth and storage cost.

VI. CONCLUSION

We have proposed CGSIL, a feasible and training-free Wi-Fi localization that is capable of returning higher accurate and finer-grained results. The training-free characteristic of CGSIL makes it more practicable comparing with other Wi-Fi based localization since it can save a lot of money, human-labor and especially time. This is crucial when the localization needs to be implemented in wide-scale with many locations such as city level. Additionally, CGSIL shows a clear advantage over ISIL, the first training-free approach, by offering an accuracy level that is two times better than that of ISIL. This achievement is based on the new ranking strategy, which utilizes the collaborative filtering, SEO properties, and the geographically clustering of location information from observable APs. The cost analysis also showed the feasibility of CGSIL in near future. CGSIL is a good choice when users desire a localization accuracy level of up to 70% with a training-free experience. When the accuracy level of 80% or more is required, other Wi-Fi based approach should be used, yet, with the cost of the expensive training step.

REFERENCES

[1] Nam Nguyen, Leonard Kleinrock, and Peter Reiher, "Debugging Ubiquitous Computing Applications with the Interaction Analyzer" in the International Journal on Advances in Software, IARIA, 2012, vol. 5, no 3-4, pp. 345-357.

[2] Nirupama Bulusu, John Heidemann, and Deborah Estrin, "GPS-less Low Cost Outdoor Localization for Very Small Devices", IEEE Personal Communications, 2000, vol. 7, issue 5, pp. 28-34.

[3] Mike Y. Chen et al., "Practical Metropolitan-Scale Positioning for GSM Phones", in Proc. Int. Conf. on Ubiquitous Computing (Ubicomp), 2006, LNCS 4206, pp. 225-242.

[4] D. Gundlegard and J. M. Karlsson, "Handover location accuracy for travel time estimation in GSM and UMTS", IET Intelligent Transport Systems, 2009, pp. 87-94.

[5] Ian Smith et al., "Place Lab: Device Positioning Using Radio Beacons in the Wild", in Proc. IEEE Conference on Pervasive Computing (Percom), 2005, pp. 116-133.

[6] Yu-Chung Cheng, Yatin Chawathe, Anthony LaMarca, and John Krumm, "Accuracy Characterization for Metropolitan-Scale Wi-Fi Localization", in Proc. ACM Int. Conf. on Mobile Systems, Applications, and Services, 2005, pp. 233-245.

[7] P. Bahl and V. N. Padmanabhan, "Radar: An in-building RF-Based User Location and Tracking System", in Proc. IEEE Conf. of Computer and Communications Societies (Infocom), Apr. 2000, pp. 775-784.

[8] Alex Varshavsky, Denis Pankratov, John Krumm, and Eyal de Lara, "Calibree: Calibration-Free Localization Using Relative Distance Estimations", in Proc. ACM Int. Conf. Pervasive, 2008, pp. 146-161.

[9] Doherty L. Pister and El Ghaoui, "Convex position estimation in wireless sensor net-works", in Proc. IEEE Int. Conf. on Computer Communications (Infocom), 2001, pp. 1655-1663.

[10] Shang Y. Ruml, W. Zhang, and Y. Fromherz, "Localization from mere connectivity", in Proc. ACM Int. Symposium on Mobile Ad-Hoc Networking and Computing (Mobihoc), 2003, pp. 201-212.

[11] Moustafa Youssef, Ashok Agrawala, and A. Udaya Shankar, "WLAN Location Determination via Clustering and Probability Distributions", in Proc. IEEE Int. Conf. on Pervasive Computing and Communications (Percom), 2003, pp. 143-150.

[12] Truc D. Le, Hung M. Le, Nhu T. Q. Nguyen, Dinh Tran, and Nam T. Nguyen, "Convert Wi-Fi Signals for Fingerprint Localization Algorithm", in Proc. IEEE Int. Conf. on Wireless Communication, Networking and Mobile Computing (WiCOM11), Wuhan, China, session 12, 2011, pp. 1–5.

[13] Truc D. Le, Nam T. Nguyen, "A Scalable Wi-Fi Based Localization Approach", in the REV Journal on Electronics and Communications (REV-JEC), 2011, vol. 1, no. 3, pp. 167-174.

[14] Andreas Haeberlen, Eliot Flannery, Andrew M. Ladd, Algis Rudys, Dan S. Wallach, and Lydia E. Kavraki, "Practical Robust Localization over Large-Scale 802.11 Wireless Networks", in Proc. ACM Int. Conf. on Mobile computing and networking (MobiCom), 2004, pp. 70-84.

[15] Truc Le, Thong Doan, Han Dinh, and Nam Nguyen, "Instant Search-based Indoor Localization" in the Proceedings of the 10th Annual IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, Nevada, USA, 2013, pp. 143-148

[16] Sergey Brin and Lawrence Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine", Computer Science Department, Stanford University, Stanford, CA 94305.

[17] Wikipedia Information about Search Engine Optimization, http://en.wikipedia.org/wiki/Search_engine_optimization, 6, 2014.

[18] Cluster Analysis, http://en.wikipedia.org/wiki/Cluster_analysis, 6, 2014

[19] Anchor Text, http://en.wikipedia.org/wiki/Anchor_text , 6, 2014

[20] Mobile data usage trends 20112015., http://www.slideshare.net/KarlPortio , 6, 2014

[21] Liu, Kaikai and Liu, Xinxin and Li, and Xiaolin, "Guoguo: Enabling Fine-grained Indoor Localization via Smartphone", in book "Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services" (MobiSys'13), 2013, Taipei, Taiwan, pp. 235-248.

[22] Priyantha, Nissanka B. and Chakraborty, Anit and Balakrishnan, and Hari, "The Cricket Location-support System", in "Proceedings of the 6th Annual International Conference on Mobile Computing and Networking" (MobiCom'00), 2000, Boston, Massachusetts, USA, pp. 32-43.

 ISBN: 978-1-61208-353-7