Research Article

Effective Schemes for Place Name Annotations with Mobile Crowd

Yao-Chung Fan and Hsueh-Wen Tseng

Department of Computer Science and Engineering, National Chung Hsing University, Taichung 40227, Taiwan

Correspondence should be addressed to Hsueh-Wen Tseng; hwtseng@nchu.edu.tw

Received 29 August 2014; Accepted 1 September 2014

Academic Editor: David Tanar

Copyright © 2015 Y.-C. Fan and H.-W. Tseng. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the popularity of mobile devices, numerous mobile applications have been and will continue to be developed for various interesting usage scenarios. Riding this trend, recent research community envisions a novel information retrieving and information-sharing platform, which views the users with mobile devices, being willing to accept crowdsourcing tasks as *crowd sensors*. With the neat idea, a set of crowd sensors applications have emerged. Among the applications, the geospatial information systems based on crowd sensors show significant potentials beyond traditional ones by providing real-time geospatial information. In the applications, user positioning is of great importance. However, existing positioning techniques have their own disadvantages. In this paper, we study using pervasive Wi-Fi access point as user position indicators. The major challenge for using Wi-Fi access point is that there is no mechanism for mapping observed Wi-Fi signals to human-defined places. To this end, our idea is to employ crowdsourcing model to perform place name annotations by mobile participants to bridge the gap between signals and human-defined places. In this paper, we propose schemes for effectively enabling crowdsourcing-based place name annotation, and conduct real trials with recruited participants to study the effectiveness of the proposed schemes. The experiment results demonstrate the effectiveness of the proposed schemes over existing solutions.

1. Introduction

Over years smart mobile devices have become a ubiquitous medium supporting various forms of retrieving information, sharing information, and interaction. Many studies also suggest significant future growth in the number of smart device users, the devices' hardware features, and the broadband bandwidth availability. With these trends, recent research community envisions a novel information retrieving and information-sharing platform, which views the users with mobile devices, being willing to accept crowdsourcing tasks as *crowd sensors*. The system based crowd sensors support novel information inquiring services beyond traditional geospatial information system by providing information in a real-time manner. Please consider the following scenarios for possible application scenarios.

Remotely Looking Around. Lawrence is at his office at the Administration Hall of National Chung Hsing University. It is 3:00 pm now. He just completes a long meeting and plans

to take exercises to play tennis. However, he is not sure if the tennis courts in the campus are all occupied now and would like to have this information to prevent going for nothing.

Fine-Grained Local Weather Information. Lawrence is at Taoyuan and it is drizzling now. The original plan for this afternoon is to have a family trip to the Taipei zoo at Muzha, Taipei. However, for now, Lawrence is not sure whether to go as planned due to the weather condition. The weather reporting web sites just provide coarse-grained weather information.

In the past, no information inquiring services, for example, Google maps, are able to or cost-effectively address the real-time queries illustrated in the abovementioned scenarios. However, with the ubiquity of smart devices providing location awareness and network connectivity, the mobile crowdsourcing model might be able to address the query in the mentioned scenario. The basic idea is that the users (as sensors) are capable of providing real-time information of

the places where they are. If we can seek the users for help, the real-time geospatial information can be acquired. Please consider the abovementioned scenario. With the crowd sensor idea, when Lawrence inquires about the information, if a user is nearby or at the tennis courts, the user can serve as an information provider to fulfill the query by returning a short text or taking a photo.

The basic working model for crowd sensors is as follows. An inquirer issues a query and provides incentives, for example, a small amount of money, to those who provide the required information, and the users at the inquired location acquire and send the information back to the inquirer through the use of the mobile devices to have the rewards. In the working model, smart devices equipped with location-aware sensors and Internet connection provide users' positions and the ability for interaction between each other, and the users of the smart devices utilize human intelligence to answer queries.

Several systems based on the mentioned-above mobile crowdsourcing model have been developed [1–5] in recent years. The systems all have a similar architecture, which divides the system into two parts, one coordinating server and one app at smart device client. The objective is to allow the system to monitor and interlink participants. The coordinating server provides the three basic operations to (1) monitor locations of users, (2) accept and forward queries for users, and (3) select users to be information providers. And the app is mainly for acquiring and reporting the location information of users and for issuing/receiving queries to/from the server.

In addition, existing real-time geospatial query answering systems all have one common component: a user tracking component that records the current positions of all users participating in the system. This component is important, because, in crowdsourced based geospatial system, the systems rely on the participants near to the query-indicated location to process query. If none of participants are near to the location, the query cannot be handled. All the participants are potential workers to be selected for future query processing. Therefore, tracking users is critical to crowdsourced based geospatial systems.

Existing solutions for tracking users in the systems are mainly based on the GPS system or user check-in information from social media, such as Facebook. However, the two options have their own disadvantages. The user tracking component based on the GPS system [1–3] has two concerns. First, the GPS component is with high power consumption. It is unlikely for users to always keep GPS sensors on. Second, people spend the majority of their time indoors, and the GPS system has the problem of positioning indoor locations, which limits the capability of the existing real-time geospatial query answering systems to be only able to address outdoor places. On the other hand, the system [4, 5] based on user check-in information to places does not have the power consumption concern but suffers from the problem of late position information. The check-in operation depends on users' willing, which may limit the system capability, as it is unlikely that users frequently perform check-in operations.

We are also designing a system called TeleEye system based on the crowd sensor model (please refer to Section 6

for our design). Being aware of the mentioned-above tracking problems, we consider using pervasive Wi-Fi access points, which now well-cover the urban city in our life, for our user-tracking purpose. The basic idea of using Wi-Fi based position is that we can scan the available Wi-Fi access point signals near to a user's current location via smart devices. The basic observation is that Wi-Fi access points are unlikely to be moved, and for the same place we should get the similar set of available Wi-Fi observations. By recording the Wi-Fi observation at a place, we will be able to know users who are near to the place.

Using Wi-Fi observation as position indicators has the following advantages. First, the power consumption of scanning Wi-Fi fingerprints is less expensive with respect to using GPS positioning sensors. Second, the Wi-Fi based approach will not suffer from the indoor locating problem in comparison with the solution using GPS position. Third, the Wi-Fi based approach still provides user positions in a real-time manner without the problem of using check-in information.

While the idea of using Wi-Fi signal is neat, the main challenge for using Wi-Fi observation as position indicators is that there is no mapping between a given observed Wi-Fi observation and a human-defined place. Notice that users will ask queries using human-defined place terms, such as 1005 Classroom in Science Building. However, in using Wi-Fi based position, what we obtain from users' carried smart devices for a place is a set of available Wi-Fi observations, which is meaningless to humans. If we are unable to translate a place given by a inquirer into a corresponding Wi-Fi position, we then are unable to select suitable users as information providers for fulfilling the queries. Therefore, before enabling the idea of using the Wi-Fi based location indicator, we need a mapping between the machine-learned Wi-Fi observation and the human-defined places.

To this problem (mapping between Wi-Fi observation and human-understandable place), our idea is to crowdsourcingly employ user annotation to bridge the gap between machine-learned and human-defined places. More specifically, when a place is considered to be an interesting place, we ask participants to tag a description (place name annotation) to the place. We use the annotation to map a Wi-Fi fingerprint to a human-understandable place.

However, several challenges are needed to be addressed before the crowdsourcing idea is enabled. First, how are we to define an interesting or meaningful place which will be later present to participants for human annotation? Second, as the cost of asking participants for help is expensive, how to rank places according to their significance is therefore another issue. In this paper, we address these challenges with the goal of reducing crowdsourcing cost and improve the usability of the annotated places in our geospatial query answering application.

The contribution of this paper is threefold.

- (i) First, to our best knowledge, we are the first to use the crowdsourcing model for mapping between Wi-Fi fingerprints and human-defined places in the real-time geospatial application based on mobile crowd.

- (ii) Second, we propose a novel framework called FIN-CHA (fingerprint clustering and hub-authority ranking) to effectively define and rank significant places based on collective Wi-Fi fingerprints from mobile crowd.
- (iii) Third, we conduct experiments over real dataset collected by real trials with recruited participants. The experiment results demonstrate the effectiveness of the proposed scheme over existing solutions.

The rest of the paper is structured as follows. Section 2 presents related work, and in Section 3 we introduce a baseline scheme. In Section 4, we introduce our scheme based on analyzing collective Wi-Fi fingerprints to define and rank places. Section 5 provides an experimental comparison for the proposed schemes. Section 6 describes the design of the TeleEye system. Finally, Section 7 concludes this paper and discusses future work to do.

2. Related Work

2.1. Crowdsourcing Applications. In recent years, the crowdsourcing has attracted great attentions. Crowdsourcing describes a distributed problem-solving model, in which small tasks are broadcasted to crowds over the Internet and attempt to motivate the crowds capable of solving the tasks in return for small incentives. From the strategic viewpoint, the crowdsourced tasks can be solved at little cost and in an effective manner. With the neat idea, a set of crowdsourcing platforms have emerged. Two well-known crowdsourcing platforms are Amazons Mechanical Turk (AMT) [6] and CrowdFlower [7], which allows users to post their tasks over the platforms, and the workers capable of addressing the tasks can select their task on the same platform. Furthermore, with the popularity of mobile device, the authors in [8] proposed *mCrowd* which extends the crowdsourcing platforms to mobile ones. In *mCrowd* platform, users can post and work on crowdsourcing tasks using their smart devices. In addition to the outsourcing models, an implicit meaning behind the crowdsourcing model is to utilize human intelligence to address the problem difficult to be tackled by computers, such as image matching, object recognition, or language understanding. Therefore, crowdsourcing has also attracted great attention from the data management research community and has been largely employed in various applications [9–12].

In [9], the authors propose a relational query processing system exploiting crowdsourcing to answer queries that cannot be easily answered by machines. In [10], the idea of utilizing human intelligence to assist the graph research is studied; given a directed acyclic graph with some target node, the authors consider the problem of finding the target node(s) by asking a human questions for assistances. In [11], the authors consider employing mobile crowdsourcing for news reporting. In the research, they mainly conduct two user studies to support the possibility of mobile crowdsourcing process. In [12], the authors study employing crowdsourcing to the relevance evaluation problem for information retrieval. The result shows that the approach based on crowdsourcing is complementary to the traditional information retrieval method and provides a flexible and inexpensive method for large-scale evaluation.

The studies in [1, 4, 13] are most similar to our system [2]. In [13], the authors study the problem of processing geospatial queries by asking the crowd over Twitter, while in [4] the users for answering the queries are selected based on the historical check-in information on social media. However, the two systems all have the problem that the queries might not be able to be addressed in a real-time manner. In [1], the idea of using mobile crowdsourcing for addressing real-time geospatial queries is also proposed. However, it uses the GPS to track users, which comes with high power consumption concern.

2.2. Wi-Fi Based Positioning. The Wi-Fi based positioning system has been studied extensively [14, 15]. The main idea for the systems is to first construct a database that records radio signals of a targeted indoor environment during training phase. Then, during positioning phase, the system is to find Wi-Fi fingerprints in the database most similar to the fingerprint of a position to estimate the position. The main purpose for building such system is to track the moving path of users in an indoor environment. Such Wi-Fi fingerprint based positioning systems all came with expensive training cost. And the techniques provide very fine-grained positions for tracked user (typically within a few meters of accuracy). However, such fine-grained positions are not necessary to our application. In fact, knowing who is at the desired place is sufficient in our application; we need only the granularity of information at the level of knowing which users are at or near to a place. In addition, the studies along this direction need not to address place name annotation problem and therefore are orthogonal to ours.

Several indoor place learning algorithms are proposed [16, 17]. However, the places learning algorithms are all focused on understanding place types, such as a restaurant or a shopping store. The techniques of understanding the types of places are not helpful to our application. In our application, we require the name of a place such that the queries can be sent to right persons.

Among the studies, the research in [18] is most relevant to us. The study also proposes to employ user feedback for semantic place annotations. However, their approach only considers the information observed by individuals to proceed with the annotations. In comparison, our scheme takes into consideration the collective Wi-Fi observations to define important places for annotations, which significantly improve the effectiveness of using crowdsourcing based annotations. In fact, the approach proposed in [18] is analogous to the baseline method (it will be introduced in Section 4) in this paper, and as will be shown in the experiments, most places detected by the baseline method are insignificant ones, which limits the capability of the underlying applications.

3. Baseline Scheme

In this section, we first present the data model addressed in the paper, then introduce a naive scheme that adopts stay duration as a metric to discover places for name annotation, and discuss insights learned from real trails using an app implementing the baseline scheme.

3.1. Data Model. The data model in this paper is as follows. First, the raw data are collected from the mobile device of the participants installed with our app. Our app mainly performs scanning available Wi-Fi signal at a predefined time interval and sends the obtained Wi-Fi observation to a data store server, where raw data from all participants are stored. The raw data is a collection of Wi-Fi fingerprints. A Wi-Fi fingerprint is a set of pairs (BSSID, RSS) plus a participant identification recording all the observed Wi-Fi AP MAC addresses and the corresponding radio strengths by a mobile device. An example for Wi-Fi fingerprint is {(40:4a:03:24:3c:b0, -47), (90:f6:52:bf:18:18, -81), (10:6f:3f:8c:45:20, -85), (IMEI:350077-52-323751-3)} indicating that three Wi-Fi access points were simultaneously observed by a mobile device with IMEI number 350077-52-323751-3.

3.2. Using Stay Duration as a Metric. For mapping Wi-Fi fingerprints and human-defined place, the first task is to define an interesting place according to the raw Wi-Fi observation. In this section, we first present a baseline scheme that uses the duration time as a metric to define an interesting place.

A straightforward approach to define an interesting place is by the duration time (the time stayed at a location). The intuition is that if someone stayed at a place for a substantial amount of time, then the place should be meaningful. Therefore, to verify this idea, we develop an Android app, which runs as a background service on a mobile device and collects Wi-Fi observation at an interval of 15 seconds. In addition, when the user visits a place for the first time and spends more than five minutes at the place, our app shows a dialog that prompts users to give a description on where he is. We use the input description as a name for the place.

With the app, we recruit 40 participants in our department to install the developed app for two weeks. The participants are all students in our departments (20 undergraduate students and 20 graduate students). After the real deployment with the participants, we collected about 80,000 Wi-Fi fingerprints, and a total of 209 places are annotated by the participants.

By investigating the results, we find the following problems for using the baseline scheme. First, we find that many Wi-Fi fingerprints are named by very rough terms, such as “my home” or “Fan’s office,” which is useless to our application. Second, some places are correctly named but only important to a small group of people or just one person. For example, many places are named by the house addresses of the participants, which is again less useful to our application. Third, some important public places are meaningful to most participants but did not get annotated, such as Department Office. This problem is due to the parameter setting of the time threshold to define a place to be named. Most participants are just passed by the office and did not make a longer stay to trigger the pop-up dialog, and therefore the place did not get annotated.

Observing the problems, we consider that they are rooted from that using the baseline scheme only to define interesting places by individual Wi-Fi observations. The approach does not take into consideration collective results from all

participants’ Wi-Fi fingerprints and therefore tend to detect personal places, which is less useful to our application. As a result, we aim to identify and assign significance to a place based on collective Wi-Fi fingerprints from all participants rather than individual fingerprints. In addition, the cost for crowdsourcing is expensive, and therefore the number of casting crowdsourcing annotations should be minimized. We consider the following indicators as the importance of a place: (1) the number of distinct visits and (2) the durations of the visits. In the following section, we introduce a scheme by analyzing the collective Wi-Fi fingerprints from all participants to define and rank a significant place. Our goal is to compute a score for each place such that top- k places can be selected for participant annotations to maximize the effectiveness.

4. FINCHA Framework

The FINCHA framework consists of two steps. The first step is a clustering algorithm which performs clustering analysis over collected Wi-Fi fingerprint datasets from all participants. The goal of this step is to define a potential meaningful place based on collective Wi-Fi fingerprints. The second step is based on the clustering analysis results of the previous step but further ranks the places by analyzing the place visiting patterns. The goal of this part is to compute a ranked list for potential places such that the utilization can be maximized. In Section 4.1, we introduce our clustering algorithm, and, in Section 4.2, the ranking mechanism for places is presented.

4.1. Two-Layered Fingerprint Clustering. The intuition for employing clustering analysis comes from the observation that the Wi-Fi observations of the same place should be similar to each other and should form a cluster structure in a corresponding feature space. Therefore, the basic idea is to form a vector space according to the collective fingerprints from all participants and then to perform clustering analysis over the data space.

As discussed, we consider the number of different visits (the visited times of different users) and the duration of the visits as significance indicators for places. To address the two criteria, our clustering algorithm consists of two layers of clustering analysis. The first layer clustering analysis processes the raw Wi-Fi fingerprints from each individual participant and produces intermediate clustering results. The goal of this layer is to find out the place with a longer stay duration time from individual observations. After that, the aim of the second layer clustering analysis is to identify the place which many different participants visited, as the place should be more meaningful to our application. Thus, the second level clustering analysis takes the results of the first level as inputs and performs another round of clustering analysis to further refine meaningful common places. A final cluster is expected to represent a single, unique meaningful place.

Given a set of fingerprints, the clustering analysis is modeled as follows. Suppose that there are m different Wi-Fi access points observed in the data collection phase. Our

app logs Wi-Fi fingerprints observed at a time interval of 15 seconds. For each log, a fingerprint is recorded, which can be represented by a vector $f = \langle s_1, s_2, \dots, s_m \rangle$, where s_i is the radio signal strength of the i th access point and $s_i = 0$ if the signal of the i th access point cannot be detected. We use the collected dataset for constructing the fingerprint space, over which clustering structure analysis is performed. We adopt the Euclidean distance as distance metric between fingerprints. The formal definition between two fingerprints is as follows. For two Wi-Fi fingerprints $f_i = \langle s_1, s_2, \dots, s_m \rangle$ and $f_j = \langle t_1, t_2, \dots, t_m \rangle$, the distance $\delta(f_i, f_j)$ between f_i and f_j is computed by

$$\delta(f_i, f_j) = \sqrt{\left(\sum_{k=1}^m (s_k - t_k)^2 \right)}. \quad (1)$$

After the distance definition, we then are able to perform clustering structure analysis over the dataset. As our raw data is collected by the mobile participants, the data consist of numerous outliers, which are produced during participants movement, as our app scans available Wi-Fi signals at an interval of 15 seconds. To remove the influence from outliers, we use the DBSCAN [19] clustering algorithm, which is a density based clustering algorithm with the capability of identifying noise. After the clustering analysis, we expect that each cluster represents a meaningful place, which is candidate to be annotated by mobile crowd. We use the centroid of a cluster as a representative for the cluster.

In using the DBSCAN algorithm, there are two parameters required: ϵ (maximum radius of the neighborhood) and MinPts (minimum number of points in an ϵ -neighborhood of a point). We utilize the DBSCAN algorithm in both layers, and we set different clustering parameters for the two clustering analysis layers. This is because the two-layer clustering analysis has different objectives. The first layer analysis processes the raw fingerprints from individuals, where noisy fingerprints exist. The objective of the first layer is therefore to find out a user stop (stay) to define a potential place, and the objective of the second layer is to refine the individual stay to discover common places, which should be meaningful to most participants and are useful to our application. Therefore, in using our two-layered algorithms, there are four parameters needed: the global MinPts, the global ϵ , the local MinPts, and the local ϵ . In the experimental evaluation, we elaborate on the parameter settings.

As will be shown in the experiment section, performing two-layer clustering analysis over collective fingerprints successfully identifies many common places, which were previously missed by the baseline method. However, we would like to point out that such method still discovers many personal places. In using the clustering analysis, we obtain a set of candidate places. However, as mentioned, the cost of place name annotation will be expensive. To maximize the utilization, a rank list for the set of candidate places will be critical in a budget-limited setting.

4.2. Hub and Authority Ranking. To rank a place, the first step is to define the significance of a place. In this study, we assume that a significant place is one that is visited by

many significant participants, and a significant participant is the one who visits many significant locations. Under such assumption, basic significance score definitions for participants and places are as follows. The significance score $s(p)$ of a place p is the sum of significance scores $s(u)$ for each participant u who stayed at p . Likewise, the significance score $s(u)$ of a participant u is the sum of significance score $s(p)$ where u stayed. Formally, we have $s(p) = \sum_{u \rightarrow p} s(u)$ and $s(u) = \sum_{p \leftarrow u} s(p)$. We employ mutual enforcement analysis over the relationship between collective participants and places to compute the enhanced significance scores.

Formally, given m participants and n places identified by the first step, we build an $m \times n$ matrix denoted by A which represents the relationship between participants and places. The entry A_{ij} is 1 if participant i has visited place j and 0 otherwise. Furthermore, our problem can be generalized into the following form. First, we denote the vector of all place scores by $\vec{q}$ and denote the vector of all participant scores by $\vec{p}$. Therefore, we can further write down the following equations:

$$\begin{aligned} \vec{q}_{k+1} &= A^T \vec{p}_k, \\ \vec{p}_{k+1} &= A \vec{q}_k, \end{aligned} \quad (2)$$

where A^T denotes the transpose of the matrix A and k is the number of iterations. The initial scores are set to 1 for all participants and places. We use the power iteration algorithm to compute the vectors $\vec{p}$ and $\vec{q}$. The computed $\vec{q}$ is then employed to rank all the places.

5. Experimental Evaluation

5.1. Experiment Setting. As mentioned in Section 3.2, we develop an Android app running as a background service on a mobile device to collect observed Wi-Fi fingerprints at an interval of 15 seconds. In addition, the app presents a simple question dialog window for prompting participants to give a description on the current place he/she stays at, when the user stays at a place for more than five minutes. The description is then used as the place name annotation for the place (originally in a form of Wi-Fi fingerprint).

We conduct two rounds of experiments. In the first round, we invite 40 participants to install our app in their mobile device for two weeks. However, as mentioned in the previous discussion, the results of using temporal based approach suffer from the problem of ignoring significant common places due to threshold setting reason. Therefore, to verify the benefit of the proposed solution, we conduct another round of experiments by asking 8 participants out of the 40 participants in the first round to install a new version of our app. In the new app, the question dialog window is triggered only when a newly observed fingerprint is a member of a cluster identified in two-layered clustering analysis over the data collected in the first round. The default parameters for our FINCHA clustering algorithm are set to 2, 0.3, 10, and 0.2.

After the two rounds of experiments, two resultant sets for the mapping between machined observed fingerprints and human-defined places are produced. To have a quantity comparison, we ask four participants to score all the places in

TABLE 1: Significance score specification.

Score	Specification
3	Meaningful places to more than 10 persons
2	Meaningful places to 5–10 persons
1	Meaningful places to 1–4 persons
0	Meaningless places

TABLE 2: Method comparison.

Method	Number of places	Average score	Mode score
Baseline	209	1.04	1
FINCHA	32	2.34	3

the resultant sets. The participants score places according to the specification shown in Table 1. For each place, we use the average score from the four participants as the significance metric for the place.

5.2. Significant Place Extraction. In the first round experiment, a total of 80,000 fingerprints are collected, and 209 places are annotated by the participants. In the round of using FINCHA scheme, 32 places are annotated. The comparison for the two resultant sets is summarized in Table 2. As previously discussed, most of the places identified by the baseline method are with low significance scores (with an average score of 1.04). And we see that the places identified by the FINCHA scheme are with high significance scores (with an average score of 2.34). We further show the significance score distribution comparison in Figure 1. We observe that over 85% of the annotated places by the baseline method are insignificant. In comparison, over 80% of the annotated places by the FINCHA scheme are significant. The experimental result suggests the effectiveness of the proposed scheme in discovering significant places.

5.3. Significant Place Ranking. We also verify the effectiveness of ranking the extracted place. As the baseline method is not with the place ranking ability, we compare the following three schemes: (1) ranks-by-stay-time (RBST), (2) ranks-by-visit (RBV), and (3) FINCHA, where the RBST and RBV schemes rank the places extracted in the two rounds by the total stay time at a place and the total number of distinct participants visiting the place, respectively. In Figure 2, we show the cumulative scores of the schemes by enumerating top- k significant places. In the figure, the x -axis is the value of k and the y -axis is the cumulative scores of the top- k places. We can see that our scheme always outperforms the RBST and RBV schemes. We can clearly observe that the RBST scheme shows poor performance compared with the other schemes. This is because RBST scheme ranks the places by the total stay time at a place, and it is reasonable for an individual to spend the majority of their time in personal places, such as home or personal office, which are all low significance score places, as the places are less useful to our real-time geospatial application.

FIGURE 1: Score distribution.

FIGURE 2: Cumulative scores.

In addition, we observe that the RBV scheme is competitive to the FINCHA scheme when the parameter k is small. The reason for such result comes from the fact that the more a place is visited, the more meaningful the place should be and have high chances to be of interest in our application. The RBV scheme ranks by a number of distinct visitors of a place and therefore is capable of discovering the places at the early stage.

6. TeleEye System

In this section, we describe the architecture of our TeleEye application and then detail the operation flow behind the TeleEye.

6.1. User Interface. The prototyped TeleEye system consists of two parts, a coordinating server site and an TeleEye app at smart phone client side. The primary interface of our TeleEye app is shown in Figure 3, and the overall architecture is shown in Figure 4, which will be discussed in the later subsection.

FIGURE 3: TeleEye app interface.

FIGURE 4: TeleEye architecture.

In Figure 3(a), we show the basic interface for the TeleEye app, and, in Figure 3(b), we show the function menu of the app, which allows users to (1) issue a TeleEye query, (2) to check the TeleEye task assigned to her/him, (3) to see the responses to a TeleEye query he/her issued, and (4) to manage the personal profile and the app settings. In Figure 3(c), we show the interface for issuing a TeleEye query, in which a user can specify (1) the location he/she is interested in and (2) the inquired information about the location. In the figure, we use the motivating scenario 1 as an example, in which the TeleEye location is set to the tennis courts at the National Chung Hsing University, Taiwan, and the query is to check if all the tennis courts are occupied. In Figure 3(d), we show the interface for responding a TeleEye query, in which a TeleEye worker can see the task and the task location. The worker can complete the task by simply writing a text or taking a photo for the tennis court. As shown in Figure 3(d), a photo about the tennis court is returned.

6.2. The TeleEye Architecture. This section describes the architecture of the TeleEye application. The TeleEye application

consists of two parts, one coordinating server and one TeleEye app at smart phone client, which allow us to interlink and monitor users. The coordinating server provides the three basic operations to (1) monitor locations of users, (2) accept and forward queries for users, and (3) find suitable users to be information providers. And the TeleEye app mainly responds to (1) reporting the location information of users and (2) issuing queries to the server and receiving results from the server. The relationship between the two major components is shown in Figure 4. In what follows, we outline the modules in the implemented system.

(i) Smart phone client side is as follows.

TeleEye App: the TeleEye app is an Android smart phone app instance, which is responsible for the user interface and the basic utility operations for the TeleEye user, such as sign-up service, profile management, GPS reading acquisition, and background connection with the TeleEye coordinating server.

Power management mechanism: this module responds to optimizing the power usage of the operations in the TeleEye app. One fundamental operation for the smart phone app is to continuously acquire and report GPS reading to the coordinating server. Such operation may be a main drain for power consumption, and therefore a power management mechanism is implemented by suppressing unnecessary communication with the coordinating server.

(ii) Coordinating server side is as follows.

TeleEye coordinating server: this module is responsible for maintaining the TeleEye tracker for all the TeleEye workers and implementing the TeleEye worker selector. The server also includes the basic operations for the TeleEye system. In current implementation, the TeleEye server is hosted in the Amazon Web Service Cloud with an instance of Microsoft Window 2008 Server and MySql database.

Task executor selection mechanism: this module considers the problem to select suitable TeleEye workers to be information providers. When a query is issued to the server, how we pair the given tasks to the potential worker becomes an interesting research problem, especially when more than the expected numbers of workers are eligible for executing the TeleEye tasks. In current implementation, workers are selected by their Euclidean distance to the inquirer location. We plan to investigate other strategies, such as considering Manhattan distance or road network distances, to select workers. We also consider using continuous k -nn query indexing techniques [20, 21] for more efficiently managing the user locations.

Query result assurance mechanism: in this module, we consider the problem when different results are reported for the same task how the query result is organized to the inquirer. In current implementation, we exploit a simple voting mechanism for cleaning the query result. We plan to further investigate this issue in the near future.

6.3. The TeleEye Operation Flow. In this section, we describe the operation flow of the TeleEye application. There are two operation flows in the TeleEye application: the user-tracking process and the user-interlinking process.

User tracking: in running TeleEye application, we require the TeleEye app to continuously acquire the GPS reading of the mobile device and report the reading to the TeleEye coordinating server, and the TeleEye server receives the GPS readings from all users and maintains the location information in a database named *user-location database*. The purpose of the user-tracking mode is to keep tracking of the locations of all the users, as they are potentially information providers to the tasks with locations they are nearby.

User interlinking: on the other hand, the user-interlinking process is invoked when a TeleEye task is issued. When a TeleEye query is issued to the server, the server first converts the indicated location to GPS coordinates and then selects possible TeleEye workers from the user-location database according to their location with respect to the TeleEye location. After that, the server contacts with the selected worker(s) to see if they will take the assigned task. If yes, the task is assigned and the server waits for the result from the workers. Otherwise, the server reselects workers for task execution. Once the results are sent back to the server, the results are sent to the user who issues the TeleEye query.

7. Conclusion

Motivated by the user positioning problem in recent crowdsourcing based geospatial query answering applications, this study proposes to address the positioning problem by leveraging pervasive Wi-Fi signals. However, the problem of using Wi-Fi signals is that there is no mapping between the observed Wi-Fi fingerprints and human-understandable places. In this paper, we propose a novel framework that performs clustering analysis over collective fingerprints from all participants to define significant places and further rank the significance of places based on the participants visiting patterns. The experiment results demonstrate the effectiveness of the proposed schemes over existing solutions.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by the National Science Council, Taiwan, under Grant no. NSC102-2221-E-005-037-MY2 and Grant no. NSC102-2221-E-005-073-.

References

- [1] F. Alt, A. S. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz, "Location-based crowdsourcing: extending crowdsourcing to the real world," in *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordiCHI '10)*, pp. 13–22, October 2010.
- [2] Y.-C. Fan, C. T. Iam, G. H. Syu, and W. H. Lee, "Teleeye: enabling real-time geospatial query answering with mobile crowd," in *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '13)*, pp. 323–324, 2013.
- [3] L. Kazemi and C. Shahabi, "GeoCrowd: enabling query answering with spatial crowdsourcing," in *Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '12)*, pp. 189–198, ACM, November 2012.
- [4] Y. Liu, T. Alexandrova, and T. Nakajima, "Using stranger as sensors: temporal and geo-sensitive question answering via social media," in *Proceedings of the 22nd International Conference on World Wide Web (WWW '13)*, pp. 803–814, 2013, <http://dl.acm.org/citation.cfm?id=2488388.2488458>.

- [5] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*, pp. 851–860, ACM, New York, NY, USA, 2010.
- [6] A. mechanical turk, <http://www.mturk.com>.
- [7] CrowdFlower, <http://www.crowdflower.com/>.
- [8] T. Yan, M. Marzilli, R. Holmes, D. Ganesan, and M. Corner, "mCrowd—a platform for mobile crowdsourcing," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*, pp. 347–348, Berkeley, Calif, USA, November 2009.
- [9] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin, "CrowdDB: answering queries with crowdsourcing," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '11)*, pp. 61–72, ACM, 2011.
- [10] A. Parameswaran, A. das Sarma, H. Garcia-Molina, N. Polyzotis, and J. Widom, "Human-assisted graph search: it's okay to ask questions," in *Proceedings of the VLDB Endowment*, vol. 4, pp. 267–278.
- [11] H. Väättäjä, T. Vainio, E. Sirkkunen, and K. Salo, "Crowdsourced news reporting: supporting news content creation with mobile phones," in *Proceedings of the 13th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '11)*, pp. 435–444, ACM, September 2011.
- [12] O. Alonso, D. E. Rose, and B. Stewart, "Crowdsourcing for relevance evaluation," *SIGIR Forum*, vol. 42, no. 2, pp. 9–15, 2008.
- [13] M. F. Bulut, Y. S. Yilmaz, and M. Demirbas, "Crowdsourcing location-based queries," in *Proceedings of the 9th IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 513–518, March 2011.
- [14] H. Shin, Y. Chon, and H. Cha, "Unsupervised construction of an indoor floor plan using a smartphone," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 6, pp. 889–898, 2012.
- [15] P. Prasithsangaree, P. Krishnamurthy, and P. K. Chrysanthis, "On indoor position location with wireless LANs," in *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '02)*, vol. 2, pp. 720–724, 2002.
- [16] M. Ye, D. Shou, W.-C. Lee, P. Yin, and K. Janowicz, "On the semantic annotation of places in location-based social networks," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*, pp. 520–528, 2011.
- [17] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao, "Automatically characterizing places with opportunistic crowdsensing using smartphones," in *Proceedings of the 14th International Conference on Ubiquitous Computing (UbiComp '12)*, pp. 481–490, Pittsburgh, Pa, USA, September 2012.
- [18] D. H. Kim, K. Han, and D. Estrin, "Employing user feedback for semantic location services," in *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp '11)*, pp. 217–226, 2011.
- [19] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD '96)*, pp. 226–231, 1996.
- [20] G. Zhao, K. Xuan, W. Rahayu et al., "Voronoi-based continuous K nearest neighbor search in mobile navigation," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 6, pp. 2247–2257, 2011.
- [21] G. Zhao, K. Xuan, and D. Taniar, "Path kNN query processing in mobile systems," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 3, pp. 1099–1107, 2013.