Discovering Personally Meaningful Places: An Interactive Clustering Approach

CHANGQING ZHOU, DAN FRANKOWSKI, PAMELA LUDFORD, SHASHI SHEKHAR, and LOREN TERVEEN
University of Minnesota

The discovery of a person's meaningful places involves obtaining the physical locations and their labels for a person's places that matter to his daily life and routines. This problem is driven by the requirements from emerging location-aware applications, which allow a user to pose queries and obtain information in reference to places, for example, "home", "work" or "Northwest Health Club". It is a challenge to map from physical locations to personally meaningful places due to a lack of understanding of what constitutes the real users' personally meaningful places. Previous work has explored algorithms to discover personal places from location data. However, we know of no systematic empirical evaluations of these algorithms, leaving designers of location-aware applications in the dark about their choices.

Our work remedies this situation. We extended a clustering algorithm to discover places. We also defined a set of essential evaluation metrics and an interactive evaluation framework. We then conducted a large-scale experiment that collected real users' location data and personally meaningful places, and illustrated the utility of our evaluation framework. Our results establish a baseline that future work can measure itself against. They also demonstrate that that our algorithm discovers places with reasonable accuracy and outperforms the well-known K-Means clustering algorithm for place discovery. Finally, we provide evidence that shapes more complex than "points" are required to represent the full range of people's everyday places.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*; H.5.m [**Information Interfaces and Presentation (e.g., HCI)**]: Miscellaneous

General Terms: Algorithms, Human Factors

Additional Key Words and Phrases: Ubiquitous computing, location-aware applications, clustering algorithms, place discovery, field studies

ACM Reference Format:

Zhou, C., Frankowski, D., Ludford, P., Shekhar, S., and Terveen, L. 2007. Discovering personally meaningful places: An interactive clustering approach. ACM Trans. Inform. Syst. 25, 3, Article 12 (July 2007), 31 pages. DOI = 10.1145/1247715.1247718 http://doi.acm.org/10.1145/1247715.1247718

Authors' address: Department of Computer Science and Engineering, University of Minnesota, 200 Union Street SE, 4-192, Minneapolis, MN 55414, email: {czhou,dfrankow,ludford, shekhar,terveen}@cs.umn.edu.

1. INTRODUCTION

The past decade has seen the emergence of a promising new technology: *location-aware mobile devices*. These devices can compute their physical locations, in various forms, such as latitude/longitude, location of the nearest cellular tower, or a coordinate on a building floor plan. Figure 1(a) shows a person's GPS location traces in a US metropolitan area.

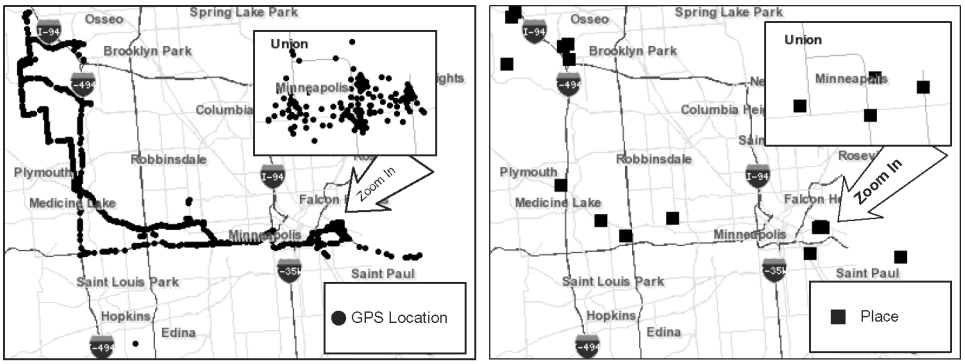
Arising from this technology are a number of *location-aware applications* with great potential [Burrell and Gay 2001; Espinoza et al. 2001; Griswold et al. 2003; Schiller and Voisard 2004]. These applications provide users a rich set of location-based services, including directory services (finding the location of the nearest ATM machine), routing services (providing the directions to the local hospital), and geo-coding (mapping a postal address to latitude/longitude) and reverse geo-coding services.

A critical component in these applications and services is the user's location. Positioning technologies obtain physical coordinations (such as latitude and longitude). Such data are irrelevant to the average user, however, who typically thinks in terms of personal and socially meaningful *places*: "home", "work", "school", "lab", "Mom's house", "Taste of India", "movie theater", "commute route", etc. Figure 1(b) shows a person's places, corresponding to his GPS location traces in a US metropolitan area.

Many location-aware applications have recently begun to incorporate the notion of personally defined places. Weilenmann and Leuchovius [2004] proposed "everyday positioning" to obtain personal places, such as "I'm waiting *where we met last time*" for location-based services. DeDe [Jung et al. 2005], a location-aware mobile messaging application, allows messages to be delivered to places like "Anne's home", "school forest path", "railway station", etc, that are meaningful to the sender and recipients. Place-Its [Sohn et al. 2005], a location-based reminder system, lets users create brief personal reminder messages ("talk to my lab mate") for delivery at a place ("lab") that is defined for a specified group of users.

The key challenge faced by these applications is to understand what constitute a user's personally meaningful places, and establish the mapping from the physical locations, shown in Figure 1(a), to personal places, shown in Figure 1(b). Also known as *personal place acquisition*, this discovery process is an essential requirement for location-aware applications and remains an open research challenge [Hightower 2003]. We detail the general problem of *personal place acquisition* and our problem definition in Section 3.

Although there have been a number of interesting place discovery systems [Marmasse and Schmandt 2000; Ashbrook and Starner 2002, 2003; Jung et al. 2005; Hightower et al. 2005; Liao et al. 2004, 2005a, 2005b], we know of no large scale quantitative empirical studies that evaluate how well these systems work. As an information retrieval problem, discovering places relies not just on the system optimized matching, but also on the user's engagement in the ongoing process [Marchionini 2004]. Thus, both metrics and a procedure must be defined to evaluate how well a place discovery system performs for its users.



(a) A person's GPS location traces (b) A person's places

Fig. 1. A person's GPS location traces and his places.

Further, previous work represents places quite simply as a geographical point or a point plus radius. This simple representation is not expressive enough to represent the complex shapes of certain places, such as "downtown", "campus", "commuting route", "grocery store", etc.

We take on the general problem of *personal place acquisition* here. We extend a clustering algorithm to discover personal places, and validate its effectiveness using an interactive evaluation framework in a large-scale empirical experiment. Our results establish a baseline that future work can measure itself against. They also demonstrate that that our algorithm discovers places with reasonable accuracy and outperforms the well-known K-Means clustering algorithm for place discovery. Finally, we provide evidence that shapes more complex than "points" are required to represent the full range of people's everyday places.

The remainder of the article is organized as follows. Section 2 presents the related work. In Section 3, we detail the problem space, define the problem formally, and state our contributions. We present our density-based clustering algorithm in Section 4 and propose a set of evaluation metrics and an interactive evaluation framework in Section 5. Our large scale empirical experiment is described in Section 6 and results are reported in Section 7. Finally, we discuss the implications of our results for future work in Section 8 and summarize in Section 9.

2. RELATED WORK

Previous work related to the discovery of places can be divided into four groups: exploratory approaches and approaches based on signal fingerprinting, machine learning, and clustering. Details about the exploratory work, fingerprinting, and the early work in machine learning of places are available in the appendix. Here we concentrate on a review of recent machine learning studies as well as clustering approaches to place discovery. As explained in Section 2.3.1, machine learning approaches may be viewed as complimentary to the clustering-based approach that our own work relies on.

2.1 Machine Learning Approaches

Liao et al. [2005b] recently advanced their work by using relational Markov networks (RMN). The algorithms learn both high level human activities and significant places. First, the algorithm spatially groups the raw GPS points 空间 within a distance of 10 meters into segments called nodes. Second, the following features are calculated from historical GPS data and assigned to the node: temporal information such as time of day, day of week and duration of the stay, speed information such as average speed, and information from geographic information databases such as whether it is on a bus route or not, etc. Third, conditional random fields (CRF) are constructed to iteratively learn high level activities on the nodes, namely, driving, walking, riding a bus, work, leisure, visit, drop off/pickup, on/off a bus, in/out of a car, etc. Finally, significant places are extracted from activities (learned from the previous step) based on their frequency of occurrence and other rules, such as "a person usually has only a limited number of different homes and work places".

Experiment results from four subjects showed that the RMN approach achieves an accuracy of 90.6% on discovering significant places. In adition, the RMN approach no longer requires "snapping" GPS data to an existing roadmap, which simplifies data pre-processing procedures. The empirical runtime evaluation showed the algorithm takes about one minute to learn three users' data (10,000 nodes) and one minute to infer the activities and places.

2.2 Clustering Approaches

Researchers have applied clustering algorithms on users' location data to discover spatio-temporal 时空 patterns. These patterns encapsulate 封装 a user's personal places.

2.2.1 *Partitioning Clustering.* Ashbrook and Starner [2002] used a variation of the well-known K-Means clustering algorithm to learn a user's significant locations from location history data.

K-Means is an efficient iterative clustering algorithm. It minimizes an error term which is the sum of the squared distances of each point to its cluster center, a mean vector. In formal notation, the error term to be minimized is

$$E = \sum_{i=1}^{c} \sum_{x \in C_i} d(x, m_i),$$

where m_i is the center of cluster C_i, and $d(x, m_i)$ is the Euclidean distance between a point x and m_i.

The algorithm initially assigns all points to a predefined number of clusters randomly. Then it iterates through each point, finds the cluster center nearest that point, and assigns the point to the cluster that the center belongs to. This iteration is repeated until the error term is deemed small or not decreasing much.

K-Means clustering has several drawbacks for detecting a user's places. First, the number of clusters must be specified before clustering begins. This could be difficult for users since in general they would not know how many places

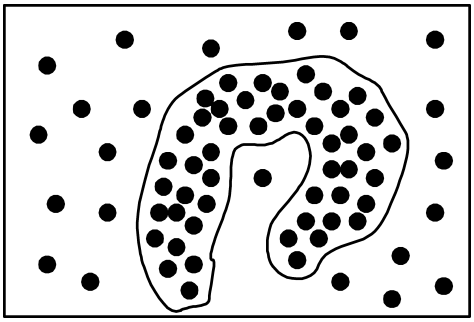


Fig. 2. A density-based approach forms a cluster where point density is high. Note that a density-based cluster can have arbitrary shapes.

they frequent. Second, all points are included in the final clustering results, which makes the results quite sensitive to noise. A single noisy or uninteresting location reading far from other points can pull a cluster center toward it much more than it should, because the squared-distance error term heavily weights distant outliers. Third, the K-Means algorithm is nondeterministic: the final clustering depends on the initial random assignment of points to clusters.

2.2.2 *Time-Based Clustering.* Kang et al. [2004] utilized the access point MAC address (AP) of a WI-FI network to capture location data on a campus. They developed a time-based clustering algorithm to "extract places" taking advantage of the continuity of the WI-FI positioning. A new place is found when the distance of the new locations from the previous place is beyond a threshold d, and when the new locations span a significant time threshold t.

This algorithm is simple and works in a novel incremental way on mobile devices. However, the algorithm does not consider the re-occurrence of readings at the same location. More simply, each time it discovers a place, it is a "different" place. This also makes it difficult to discover places that are visited with high frequency but short dwell time. Finally, this method requires continuous location data collection with very fine intervals, and thus large storage.

2.2.3 *Density-Based Clustering.* The ability of density-based clustering algorithms to manage spatial characteristics makes them good candidates for place discovering from location data. Our approach is density-based and we introduce the basic concepts here.

Density-based clustering algorithms form clusters based on the density of local neighborhoods of points [Ester et al. 1996]. Two parameters define cluster density: Eps, the radius of a circle, and $MinPts$, the minimum number of points within that circle. Density-based clustering also uses a notion of the connectivity of a neighborhood whose points eventually form a cluster. Points that are not in any clusters by the end are deemed noise. Each cluster has a considerably higher density of points than areas outside of the cluster. A sample density-based cluster is shown in Figure 2.

A density-based algorithm overcomes many of the limitations of K-Means.

First, it can discover clusters of arbitrary shape. This is a significant improvement over K-Means, which favors symmetric shaped clusters (circles and spheres). As we discuss later, not all personal places are circular-shaped. For example, while the points of a commuter's location history at a gas station may form a somewhat round shape, the points of a student's location history on a university campus might easily form an irregular shape.

Second, noise, outliers, or just unusual points are less likely to participate in the final clustering results. A user may stop at a gas station he never returns to; he may have expected stops at traffic lights; or his GPS device may show a few points far from any actual location he visited. Such locations are unusual, and perhaps should be ignored. Fittingly, they may generate few enough points to be discarded because they do not meet the density requirement of a density-based algorithm. Since K-Means does not ignore any data, such points will pull cluster centers toward them and thus away from their true locations.

Third, although density-based algorithms require several parameters (*Eps* and *MinPts*) to be specified upfront, these parameters are rather robust for a broad range of discovery applications. Our results (discussed below) provide guidance for setting these parameters for discovery within a metropolitan area. 大都市
K-Means, on the other hand, requires the number of clusters to be specified before running the algorithm, a requirement that generally cannot be met by users in this context.

Finally, our density-based algorithm, described in Section 4, always produces the same clustering given the same input. Anyone who has tried clustering that is sensitive to initial conditions can very much appreciate the taming of a disturbing randomness.

2.3 Motivation of Our Approach

In this section, we discuss the motivation that leads us to extend a density-based clustering approach. We detail how our work differs from and advances prior work.

2.3.1 *Our Approach vs Machine Learning.* Computational complexity analysis shows that clustering algorithms are efficient and scalable. These algorithms are well-known and straightforward to implement. In addition, the primary computational task in a clustering algorithm is the calculation of the distance between GPS readings, and this computation can take advantage of widely available spatial indices for location data.

When machine learning algorithms have been applied to place discovery, they have focused on people's activities at significant places and the transitions between places. Thus, they address a rather higher-level problem than clustering algorithms do. Indeed, the two approaches are complementary: the results of a clustering algorithm, which capture the spatial and temporal semantics of places, can serve as suitable input (activity and place candidates) for machine learning algorithms.

2.3.2 *Our Approach vs K-Means and a Representative Density-Based Algorithm, DBSCAN.* The advantages of DJ-Cluster over K-Means are discussed

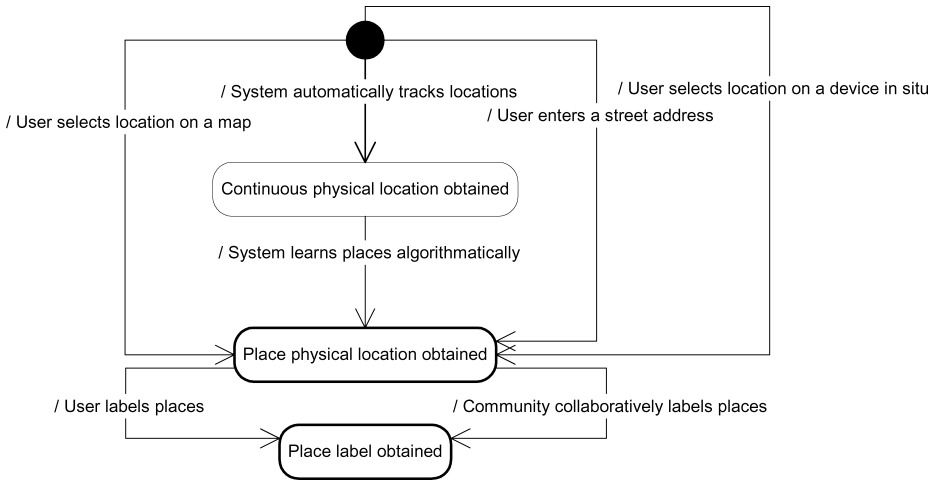


Fig. 3. A state transition diagram for an interactive place acquisition system. The two major states, *place physical location obtained* and *place label obtained*, correspond to two essential tasks in *place acquisition* for location-aware application. This article addresses the first task: discovering place geometries from location data using a semi-automated approach.

and illustrated in Section 7.5. We now discuss the limitations of a representative density-based algorithm, DBSCAN [Ester et al. 1996; Sander et al. 1998].

There is evidence that DBSCAN is very sensitive to the parameters *Eps* and *MinPts*, and does not provide a strategy to efficiently handle datasets that do not fit in memory [Milenova and Campos 2002; Zaïane et al. 2002]. Potentially, the algorithm could generate a large number of points within the cluster density definition (defined by *Eps* and *MinPts*) in the main routine, each of which could be further used to generate its own density-reachable points in the *ExpandCluster* sub-routine. In such cases, it will use a lot of memory and slow down considerably. Our proof-of-concept implementation of DBSCAN in both PL/SQL and C encountered this memory issue with 3000 GPS points.

内存问题

The memory issues in DBSCAN led us to design a new density-based cluster algorithm: DJ-Cluster [Zhou et al. 2004].

3. PERSONAL PLACE ACQUISITION AND PROBLEM DEFINITION

In the above section, we identified the limitations of the prior work in place discovery. We now describe the general problem space in *personal place acquision*, which sets up the necessary context for our problem definition and contribution statement.

3.1 Personal Place Acquisition

As shown in Figure 3, a personal place acquisition system has two essential tasks: *obtaining physical locations* and *obtaining labels for the locations*.

3.1.1 *Obtaining Physical Locations.* So far, we have referred to *physical locations*; place discovery systems, however, represent locations as *location*

geometries, such as a latitude/longitude *point* or a *set* of such points. In the remainder of the article, we use the terms *physical location* and *location geometry* interchangeably.

A location-aware application will save location geometries in each user's personal place database. This database can be hosted on the application server or on a user's mobile device. Just like a personal phone book or calendar, a personal place database can be synchronized between multiple devices, such as PDAs and cell phones.

We identify four approaches that can be used to obtain place location geometries.

—*User selects location from a map*: A user selects a location from an interactive map interface, and geometric information for the location is saved. For example, a user might zoom in to his neighborhood, click the location of his home, causing the place with a latitude/longitude point location geometry to be stored.

—*User selects location on a device in situ*: A user notifies his mobile device to save the "current" location. For example, a user walks into his office building, presses a button on the device, and a point representing the current location is stored.

—*User enters a street address*: A user enters a standard postal address, and the back-end application uses a geocoding service to map the address to a latitude/longitude point.

—*Semi-automated discovery*: A system continuously records a user's location, resulting in a series of points. The system then applies an algorithm to this series of points to discover places for the user. We will further discuss this approach in section 2.1.

3.1.2 *Obtaining Labels.* After location geometries are obtained, the next step is to acquire labels. There are several ways to acquire labels:

—*Users label their own places*. For example, after identifying a place by clicking on an interactive map, a user next types a label for the place.

—*A community collaboratively labels places*. After one user defines a place, including its label, he may choose to add this to a shared database. When another user subsequently wants to add this place to his personal database, he can consult the shared database and reuse this place—both its geometry and label—rather than doing the work of defining it himself.

—*Generating place labels from geo-databases and gazetteers*. Reverse-geocoding services can produce a postal service address from a latitude/longitude point; Naaman et al. [2004] developed algorithms to infer place names ("Stanford", "Butano State Park" and "40Kms S of San Francisco") from geo-databases that contain names and latitude/longitude coordinates of states, cities and parks.

—*Associating place labels from other applications*. Users enter various place names (e.g., *Hyde Park*, *Central Park*) into map search engines (e.g., Multimap and Google Maps) and calender location entries (e.g., the location field

in Yahoo Calendar). These names are meaningful to the user and are perfect candidates for place labeling.

3.2 Problem Definition

The aim of discovering personally meaningful places is to obtain the physical locations and their labels for a person's places that matter to his daily life and routines. The place labeling problem was discussed in Zhou et al. [2005c] and we do not address it further in this article. The input to the place discovery process is a user's historical GPS location data and the output is the location geometries of a set of places. The objective is to discover places that users judge to be accurate. In achieving this objective, we must keep in mind two constraints: the GPS dataset needs to be representative of different groups of location-aware application users, and the data collection must be long enough to capture users' typical places and routines. We focus on a semi-automated approach since we believe this is best suited for a real location-aware application environment.

3.3 Our Contributions

This article significantly extends our previously published work [Zhou et al. 2004, 2005a, 2005b, 2005c]. The main contributions are as follows:

—an evaluation framework for place discovery algorithms that defines a set of metrics and procedures;
—a large-scale empirical study that shows the promise of our DJ-Cluster algorithm and establishes a baseline for future work;
—the identification of several important remaining challenges for place discovery algorithms and a sketch of our approach to the challenges.

4. THE DJ-CLUSTER DISCOVERY ALGORITHM

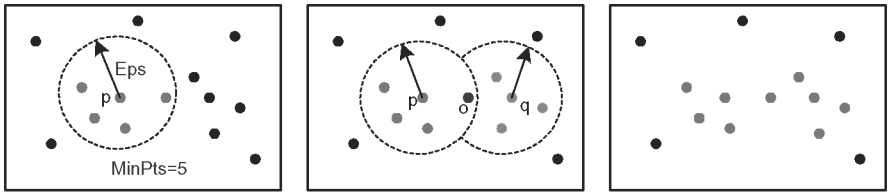
DJ-Cluster is a **D**ensity-and-**J**oin-based algorithm that requires at most a single scan of the data. The basic idea of the algorithm is as follows. For each point, calculate its *neighborhood*: the neighborhood consists of points within distance *Eps*, with the condition that there are at least *MinPts* of such points. If no such neighborhood is found, the point is labeled as noise; otherwise, the points are either created as a new cluster if no neighbor belongs to an existing cluster, or joined with an existing cluster if any neighbor belongs to the existing cluster.

4.1 Definitions and the Algorithm

The following definitions define the *density-based neighborhood of a point* and *density-joinable* relationships.

Definition 1 (*Density-Based Neighborhood of a Point*). The density-based neighborhood N of a point p, denoted by $N(p)$, is defined by

$$N(p) = \{q \in S \mid dist(p, q) \leq Eps\},$$



(a) $Density - N(p)$ (b) $Density - Joinable$ (c) $DJ - Cluster$

Fig. 4. Density-based join concepts. (a) illustrates the density-based neighborhood N of a point p; (b) illustrates that $N(p)$ is density-joinable to $N(q)$; (c) illustrates the final cluster in red(light) color.

where S is the set of all points, q is any point in the sample, *Eps* is the radius of a circle around p that defines the density.

$N(p)$ also needs to satisfy the following condition

$$|N(p)| \geq MinPts,$$

where *MinPts* is the minimum number of points required in that circle.

Definition 2 (*Density-Joinable*). $N(p)$ is density-joinable to $N(q)$, denoted as $J(N(p), N(q))$, with respect to *Eps* and *MinPts*, if there is a point o such that both $N(p)$ and $N(q)$ contain o. A density-joinable relation is illustrated in Figure 4.

The DJ-Cluster algorithm is described in Algorithm 1. It has the following key properties:

—Every point is in exactly one cluster or is ignored as noise;
—There are always at least *MinPts* points in each cluster;
—The algorithm partitions the input into non-hierarchical clusters;
—The clusters are mutually exclusive;

Algorithm 1 DJ-Cluster

```
1: while there is at least one unprocessed point p in sample S do
2:    Compute the density-based neighborhood N(p) wrt Eps and MinPts.
3:    if N(p) is null (p is not in a cluster) then
4:       Label p as noise.
5:    else if N(p) is density-joinable to an existing cluster then
6:       Merge N(p) and all its density-joinable clusters.
7:    else
8:       Create a new cluster C based on N(p).
9:    end if
10: end while
```

The two data structures in the algorithm are: *points* that represent the location geometries, and *collections of points* that represent clusters. When applying DJ-Clustering on a person's GPS traces, it loops through each GPS point and calculates its *density-based neighborhood* $N(p)$ which centers at p with radius

of *Eps*. If the number of points in the neighborhood exceeds *MinPts*, all the points in the neighborhood form a cluster. This cluster is then merged with any existing overlapping clusters. This merging process is critical in aggregating dense GPS readings in places such as "home" and "work". At the end of each loop, the total number of clusters either does not change because no new cluster is found or the new cluster was merged with one existing cluster, or increases because the new cluster is disjoined from other clusters, or reduces because the new cluster merges with two or more existing clusters.

4.2 Determinism of DJ-Cluster

We now present the proof that DJ-Cluster is deterministic.

THEOREM 1. *DJ-Cluster produces one unique clustering.*

Suppose $R(p, q)$ is a relation that is true iff p and q are points in the same cluster. We show that R is an equivalence relation. That is, it is reflexive, symmetric, and transitive.

First, by inspection, R is reflexive (a point is in its own cluster) and symmetric (if p is in q's cluster, then q is in p's cluster).

Suppose we are given $R(p, q)$ and $R(q, s)$. These points must have been processed by the algorithm in some order. Suppose the last point processed was p. We know q is in p's neighborhood because $R(p, q)$, and that q and s are in the same cluster because $R(q, s)$. Thus, p will be merged into the same cluster as q and s, so $R(p, s)$.

Suppose instead that the last point was q. Then both p and s will be in q's neighborhood, and the cluster or clusters with p and s will all be merged, and again $R(p, s)$.

Finally, if the last point processed was s, this case is just like p. This proves that R is transitive.

Since R is reflexive, symmetric, and transitive, it is an equivalence relation, which partitions a space uniquely (up to equivalence classes). Thus, the DJ-Cluster clustering is unique.

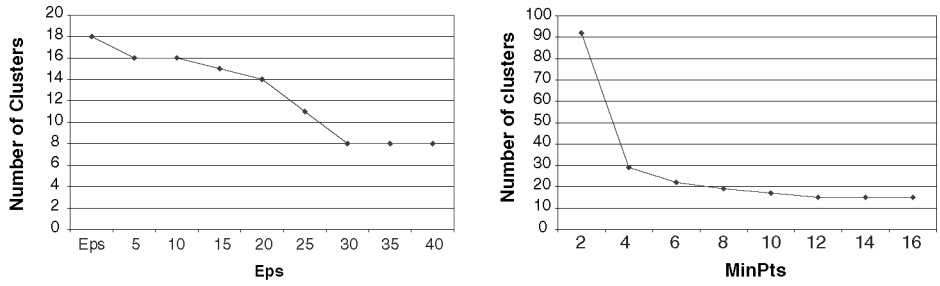
4.3 Computational Complexity

LEMMA 1. *The complexity of DJ-Cluster with an R-tree index is* $O(n \log n)$.

We can analyze the complexity of the algorithm in two steps.

For each point, the first step is computing the point's neighborhood point, which costs $O(n)$ without a spatial index or $O(\log n)$ with an R-tree index. The second step is the join computation of the neighborhood with existing clusters. In average cases, the size of the neighborhood of a point is very small and the number of clusters is also small (in the experiment we described below, the average was just over 15), so the cost can be regarded as constant.

Thus overall, the complexity of the algorithm with an R-tree index is $O(n \log n)$.



(a) Various Eps with fixed $MinPts$= 10 (b) Various $MinPts$ with fixed Eps= 20

Fig. 5. DJ-Clustering: choices of parameter *MinPts* and *eps*.

4.4 Parameter Setting

In this section, we test whether DJ-Cluster suffers the parameter sensitivity problem associated with DBSCAN, and we provide guidance on setting parameters when for using DJ-Cluster on real user data.

We used the first author' GPS data to do this algorithm property analysis. The first author carried a GPS-enabled mobile phone for 3 weeks. The phone was configured to take a GPS reading every minute and send it to a server for storage. This resulted in a total of 3469 GPS readings, amounting to more than 170 per day or over three hours worth of data per day. Concurrently, the author logged all the places he remembered going to during that period, ending up with a total of 18 "baseline" places. We then ran our implementation of DJ-Cluster on the location dataset. (See Zhou et al. [2004] for details.)

Figure 5(a) shows the number of places discovered for different values of *Eps* (in meters). The curve levels out at 8 when *Eps* exceeds 30, which is slightly greater than the accuracy of GPS (20 meters). The number of discovered places fluctuates between 16 and 14 as *Eps* varies from 5 to 20. This makes sense because 5–20 are below the granularity of accuracy for GPS, so the algorithm should not discover a significantly different number of places.

Figure 5(b) shows the number of places discovered for different values of *MinPts*. Given the analysis of the previous paragraph, we fixed *Eps* to be 20 meters. The number of discovered place drops sharply (from 96 to 22) as *MinPts* increases from 2 to 6, drops slightly (from 22 to 18) as *MinPts* increases from 6 to 10, and finally converges to a stable level of 18.

The above analysis show that *Eps* should be set to a value approximately equal to the degree of accuracy of the positioning technology being used, such as 20 meters for GPS data; a wide range of *MinPts* choices lead to the same number of discovered places.

Therefore, we concluded that it is relatively straightforward to choose parameter values for DJ-Cluster and that results are fairly stable for different values.

5. EVALUATION FRAMEWORK

While the results just presented are promising, they are not conclusive. They are based on just one person's data, and one of the authors of this article at

that. To systematically evaluate whether this potential is realized—indeed, to carefully define how any place discovery algorithm should be evaluated—we developed an evaluation framework and set of metrics and used it to perform a large-scale empirical evaluation.

The performance of a place discovery algorithm revolves around these basic questions: (1) *Are the results accurate?* (2) *Are the results useful?* (3) *Are they available in a timely manner?* To address these questions, we describe a set of metrics and explain their utility. We then detail our evaluation framework and procedures for collecting data and computing the metrics.

5.1 Evaluation Metrics

5.1.1 *Accuracy.* The traditional information retrieval metrics *recall* and *precision* are an appropriate starting point for measuring the accuracy of a place discovery algorithm. However, since we are interested in discovering personal places *for a user*, there is no single gold standard. Instead, all users must define their own places, that is, the set of places they find meaningful. Further, since there is no *a priori* corpus of places, we must take into account that people may not be able to think of all the places they care about when producing their baseline. Thus, our metrics must include a way to measure *discovery* of interesting places that were not actually in the baseline set.

Finally, since the object is to obtain physical geometries, we want to evaluate how closely the geometries discovered by the algorithm match the actual geometries of a user's place. For example, if a person's commuting route is a path (a sequence of location points), we want to see whether the system discovers a path. Intuitively, we proposed the following shapes. A "generic" place with multiple locations such as "Target" and "MacDonald's" consists of a *set of points*. Other places such as "campus" or "uptown" are *regions*. And still others such as "my drive to work" or "my favorite bike path" are *paths*.

5.1.2 *Usefulness.* Not all places are equally important to a person. For example, a person is likely to consider his home, place of work, and child's school very important, while a gas station or fast food restaurant he stops at fairly regularly are not important at all. The former set of places are important because they are major organizing foci for one's everyday activities [Genereux et al. 1983; Kramer 1995].

We conjecture that places that are important to a person are likely to be useful when that person uses a location-aware application. For example, location-aware reminder systems like Place-Its [Sohn et al. 2005] or PlaceMail [Ludford et al. 2006] let users associate virtual messages with places. Our conjecture implies that acquisition for these applications should be focused on important places. Further, since we presume that a user must always confirm a discovered place and provide a label, we want to focus the user's attention on those places that really matter. Therefore, we will evaluate whether an algorithm does especially well (or poorly) at discovering places deemed important.

5.1.3 *Timeliness.* Since place acquisition is done in service of a location-aware application, the application becomes more useful as more places are

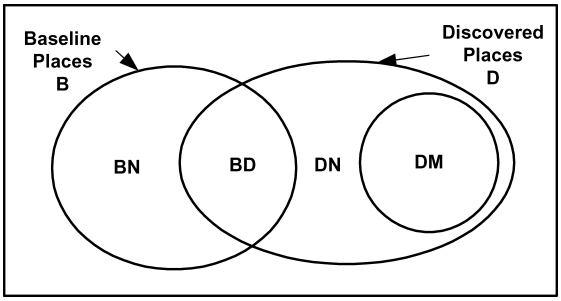


Fig. 6. Sets of baseline and discovered places.

discovered. Thus, we will evaluate how "quickly" an algorithm discovers places.

Note that the timeliness of discovery depends not just on the properties of the algorithm itself, but also on the data collection protocol and the nature of a person's everyday activities. Collecting location data at frequent intervals helps. For example, for a density-based algorithm like DJ-Cluster, it decreases the time required to get the necessary *MinPts* points for a cluster to be formed. However, the structure of a person's everyday activities also matters. For example, a person might consider his church an important place, even if he only goes there once a week. Thus, not just frequent sampling, but sampling for some duration of time may be required.

5.2 Evaluation Framework

Our evaluation framework consists of four major steps: collecting data from users, running the algorithm, obtaining user judgments on the results, and computing performance metrics. We detail each of these steps.

(1) *Collect Data*. For the data collection period, have users carry a location-aware device that takes frequent location readings and stores them in a database—call these the *personal location datasets*. Have users also keep a log of all the places they remember visiting during the collection period—call these the *baseline places*.

(2) *Run the Algorithm*. Run the discovery algorithm on each user's personal location dataset to obtain a set of *discovered places*.

(3) *Obtain User Judgments*. Present each user with his/her set of baseline places and discovered places.
   —Ask the user to try to match the two sets. As shown in Figure 6, some baseline places will be discovered—call these *BD*. Some will not—call these *BN*. Of the remaining discovered places, ask the user which were interesting and meaningful—call these *DM*; which were not meaningful and do not make sense—call these *DN*.
   —Ask the user to describe the geometry of each of the places: as we have discussed, useful geometries are *points*—places consisting of a single geographic location such one's home; *multi-points*—places consisting of multiple points such as all the grocery stores in one's town; *paths*—a sequence

of points such as one's commuting route; and *regions*—a geographical area such as a University campus.
—Ask the user to rate the importance of each place.

(4) *Compute Metrics*.
—*Accuracy*. Compute precision and recall for each user. Precision is the proportion of discovered places that were in the baseline. Recall is the proportion of baseline places that were discovered. To account for discovery of interesting places that weren't in the baseline, compute a "surprise factor", which is the proportion of discovered places that were not in the baseline, but that users judged interesting. Finally, to evaluate accuracy of place geometry, compare the geometry obtained by the system to that provided by the user.
—*Usefulness*. Consider any places with an importance score above a specified threshold (say 4 or higher on a 5 point scale) as important. Compute recall for important places, that is, the proportion of a user's important baseline places that were discovered.
—*Timeliness*. Determine how long it took (i.e., how much location data was required) to discover each place. As we discuss below, different types of algorithms will lead to different ways of doing this computation.

6. EXPERIMENT

In the summer of 2004, we conducted an empirical evaluation of our discovery algorithm using this framework.

Subjects. We thought it plausible that people's daily activities, and thus the types of places they visited, would depend on their life stage. For example, we expected 20-year-old college students, 40 year old working parents, or 60-year-old retirees to frequent different types of places. Therefore, we recruited subjects from across this spectrum.

We ended up with 28 subjects, all from the Minneapolis/St. Paul metropolitan area of the United States. Some subjects lived in the core cities, and some in the surrounding suburbs. They used a variety of travel modes, including walking, biking, public transportation, and personal car. Their ages ranged from the early 20s to late 60s, with an average in the early 30s. Twenty were male, 8 female. Three subjects had preschool children, 4 had school-age children, and 2 had adult children not living with them. Subjects were highly educated, with two thirds having college or advanced degrees. They included 6 college students, 4 engineers, 4 information technology professionals, 4 teachers, a range of other professional and service workers, as well as several retired people.

Data Collection. We equipped subjects with a GPS-enabled Motorola i88s cell phone with Nextel service that ran the Accutracking software (www.accutracking.com). We set the software to take a GPS reading every minute and send it to our server. Subjects carried the phone for about three weeks, and were instructed to keep it with them and turned on at all times. However, they could turn off the phone for privacy reasons whenever they wanted. Subjects logged their baseline places daily in a diary [Hightower 2003; Rieman

Table I. Excerpt from the Interview Table, Showing Data for a Subset of One Subject's Places

Description	Label	Importance	Geometry	Comments
Home	K	5	Dot	
Cub Foods	I,J	5	Dot	Where I work
TCF Bank	I	2	Dot	TCF is inside cub
ME Building		5	Dot	Where I study
Neighbor's house	L	1	Dot	
Bus stop	P,G,H	5	Multi-Dots	I ride bus to work and school
Parking lot	F	3	Multi-Dots	Many places where I catch a ride from friends
Walk to bus stop	M,N,O,P	5	Path	I walk 3-4 blocks
Rosedale mall		1	Region	Where I shop
Anbon's house	A	1	Dot	
Gameworks		1	Dot	
Sally's Bar		1	Dot	

The subject mapped labels we had provided for the discovered places (on the map) to the baseline places in the table. The subject also gave an importance score, the geometry, and comments for each place. A place without a label was not discovered by the algorithm.

1993]. They received a daily reminder (email, instant message or phone call) to record their places; they could email them to the experimenters or record them in a notebook which they would bring at the end of the data collection phase. We met with each subject at the beginning of the study to give them the phone, demonstrate its use, and instruct them in the data collection procedure. After data collection was complete, we conducted a semi-structured interview with each subject.

Running the Algorithm/Interview Preparation. We ran DJ-Cluster on each subject's personal location dataset. For each subject, we printed out an overview map showing all the discovered places, and a set of more detailed maps showing nearby groups of discovered places at a higher level of resolution; for example, at the detailed level, street names were visible, and the city block of each place was apparent. We also printed a table of each subject's baseline places (Table I).

Interviews/Obtaining User Judgments. We conducted a semi-structured interview with each subject organized around the maps and table. We first led subjects through the process of matching their baseline places (in the table) to the discovered places (on the map). This resulted in the various sets defined in the evaluation procedure and let us compute the *Precision*, *Recall*, and *SurpriseFactor* metrics. We also asked subjects to rate the importance of each place on a scale of 1 (least important) to 5 (most important). Finally, we asked subjects to describe the geometry of each place: *Dot*, *Multi-Dots*, *Region*, *Path*, or *Other* (more details in Section 5.1.1). In this experiment, we always presented the results of DJ-Cluster on the maps as dots. We did this because our goal at this stage of our research was simply to understand whether users conceived of their places in more complex geometries (i.e., beyond points) and how often they did so. In other words, our goal was conceptual understanding rather than purely algorithm evaluation.

Table II. Summary of the Personal Location Dataset for All 24 Subjects

	Readings	Data Collect Days	Mean Readings Per Day
Total	152,741	516	7,008
Mean Per Subject	6,364	22	292
Standard Deviation	3,997	3	167

Table III. Baseline and DJ-Cluster Discovered Places for All Subjects

	Total	Mean Per Subject	Standard Deviation
Baseline(B)	681	28.4	13.0
Discovered(D)	369	15.4	8.4
Baseline Discovered(BD)	313	13.0	7.1
Non-baseline Meaningful Discovered(DM)	56	2.3	2.8
Non-baseline Non-meaningful discovered(DN)	0	0	0
Baseline Important(B_{impt})	183	7.6	6.1
Baseline Important Discovered(BD_{impt})	121	5.0	3.7
Non-baseline Important Discovered(DM_{impt})	7	1	0.6

7. RESULTS

In this section, we first detail the data we collected, then present the results for each of our three main metrics.

7.1 Data

All 28 subjects logged their data for three weeks. Three subjects were unable to schedule final interviews, so we discarded their data. Also, the interview with one subject was unsuccessful, so we discarded this data as well. We therefore ended up with data for 24 subjects.

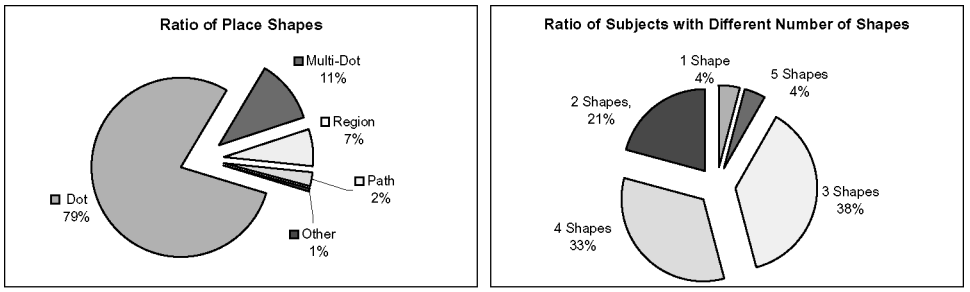
Personal Location Dataset. Table II summarizes the location readings collected from the subjects. This is a large amount of real data—we know of no other studies with samples of this size. Subjects followed the data collection procedure quite faithfully. At one reading per minute, the average number of readings per subject—6,364—represents an average of over 100 hours of data.

Baseline and Discovered Places. As shown in Table III, subjects logged a total of 681 places in their diaries during the experiment, or about 28.4 per subject. The system discovered 313 places from the baseline and 56 nonbaseline places. For each subject, it discovered at least some meaningful places that the subject had not logged.

7.2 Accuracy

We use these numbers to compute our three main accuracy metrics. The results are:

—**Recall** = 313/681 = 46%
—**Precision** = 313/369 = 85%
—**Surprise Factor** = 56/369 = 15%



(a) Ratio of different types of shapes (b) Ratio of subjects with different # of shapes

Fig. 7. Place shapes for all the subjects.

What one would most like to know is: Are these numbers good or bad? Unfortunately, since this is the first empirical evaluation of a place discovery system that we know of, there is no point of comparison. Therefore, we see these numbers as establishing a baseline that other systems can use as a point of comparison. By providing an evaluation framework and metrics and publishing these numbers, we give future researchers in the area a way to answer the question: Is this "better" or just "different"? [Newman 1997].

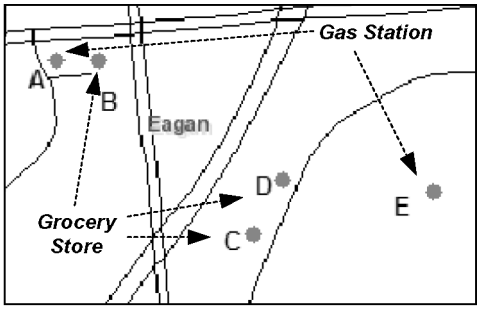
Further, observe that the algorithm discovers personal places with high precision and low recall. Since *Precision* and *SurpriseFactor* sum to 100%, all the places discovered by the algorithm were personally meaningful to subjects. On the other hand, the algorithm failed to discover 54% of subjects' baseline places. If we recall that the system is intended to be an interactive aid to a user, then we can say that what it tells a user will almost always be useful, but that it will make less than half of all useful suggestions.

Why is recall so low? Through asking users about their baseline places that were not discovered, it is our belief that most of these actually were only visited once or twice during the data collection period. For example, one subject's undiscovered places included *City Library, Home Depot, Lifetime Fitness Center, two lunch restaurants, Science Museum and a friend's house*. He reported visiting each of these places just once during the experiment. This would mean that the readings generated for that place would be too few to satisfy the density requirements for the discovery algorithm. We present additional data on this issue in the discussion of Usefulness below.

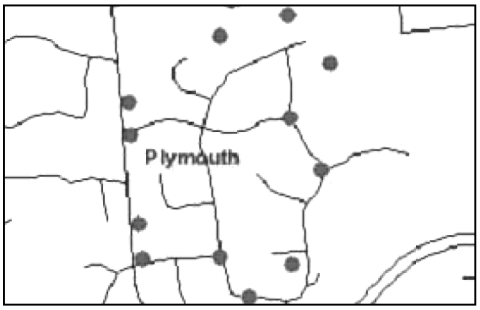
7.2.1 *Shape Correspondence.* Figure 7(a) summarizes the number of places that fell into each place shape. A large majority (79%) of the places were simple point locations or "dots". This may reflect either the nature of the places that people most frequent, and it also may partially reflect that this is the simplest place geometry to understand. However, 21% of the places had more complicated geometries.

Figure 7(b) illustrates the percentage of subjects that reported different number of shapes: 38% of the subjects reported 3 shapes, 33% identified 4 shapes, 21% gave 2 shapes and only 4% reported 1 shape. The large percentage of

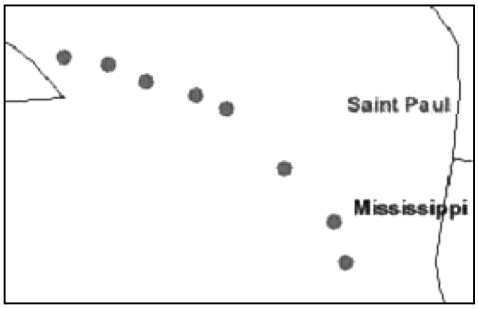
(a) A subject's home(A) and his friend's house(B)



(b) A subject's grocery stores(D,C,B) and gas stations(A,E)



(c) A subject's neighborhood region



(d) A subject's dog walking path

Fig. 8. Example of *Dot*, *Multiple Dots*, *Path*, and *Region* type places.

subjects that reported more than 1 shape suggests that place shapes are well-received concepts.

Figure 8 shows examples of different place types discovered by the algorithm. Presently, it is only we as observers who can tell that Figure 8(b) probably shows two sets of multi-points, that Figure 8(c) probably shows a region, and that Figure 8(d) probably shows a path. DJ-Cluster still treats these simply as distinct places. However, in the Future Work section, we discuss ways to extend DJ-Cluster to discover more complicated geometries.

7.3 Usefulness

We next evaluated how well DJ-Cluster did at discovering those places that users considered important. For the purposes of this analysis, we counted a place as important if a subject gave an importance score of 4 or 5. Here the recall was significantly higher: 66% (121 out of 183) of important baseline places were discovered.

Nonetheless, it still is discouraging that a third of the places that really matter to users were not discovered. To better understand both why recall was higher for important places, but a third of these places still were not discovered, we attempted to identify features that predicted the importance of a place.

Table IV. Attributes of Discovered Places

	Readings	ReadingDays	Visits
Minimum	10	1	1
Maximum	13684	21	41
Mean	362.9	4.3	4.9
StdDev	1447.2	5.3	6.4

Table V. Attributes of Discovered Places Grouped by Different Importance Scores

Importance	Readings	ReadingDays	Visits
5	1468	11	12
4	76	2	3
3	62	3	3
2	41	2	3
1	50	3	3

Since our location datasets consist of timestamped GPS readings, we can compute features for a cluster based on the number of readings, their temporal distribution, or both. We considered a number of such features:

—*Readings*: This simplest of possible factors is still quite plausible since it serves as our best proxy for the amount of time a subject spent at a place. And intuitively, the more time a person spends at a place, the more important that place would likely be to the person. However, there also are several more sophisticated features that take account of the temporal properties of the readings in a cluster.

—*ReadingDays*: Number of unique days on which a reading in the cluster occurred. This lets us distinguish cases such a (a) 3 hours worth of readings in one day at a park that the subject has never been to before and does not intend to go back to again, from (b) 10 minutes worth of readings every weekday near my place of work. Intuitively, it seems that the latter case is more likely to represent an important place.

—*Visits*: A visit represents a contiguous amount of time spent at a place. Intuitively, I might have several visits to my office on a typical day interrupted, say, by going out to lunch and for a coffee break. We compute visits to a place by first sorting all readings for a subject by timestamp, then finding maximal sequences of readings all from the same cluster.

Note that there are additional temporal features that could be computed, such as how many readings(visits) to a place happened on a weekday or weekend, during the daytime or nighttime, and whether the readings/visits followed a regular pattern. However, we did not try out any of these factors.

Table IV summarizes the *Readings*, *ReadingDays*, and *Visits* features for all discovered places, and Table V breaks them down by the importance of places. The raw numbers suggest that there is a difference between the most important places (score = 5) and all other places, but unfortunately, no other clear patterns appear. Indeed, more sophisticated analysis such as building decision trees or regression models do not yield anything interesting.

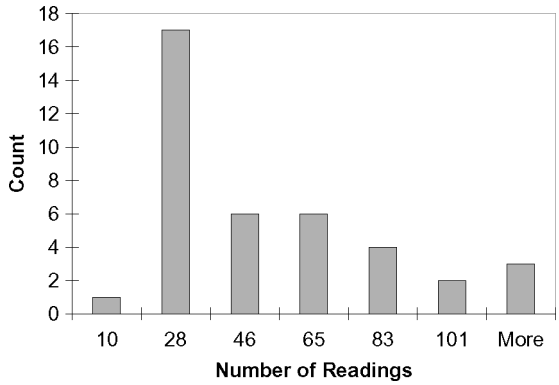


Fig. 9. Histogram of GPS readings for places with important score 5.

Table VI. Number of Readings of Most and Least Important Places for 4 Subjects (Names have been Anonymized)

Subject	Places (most important)	Readings	Places (least important)	Readings
Donna	*Cole (son) and Jenny's house*	4	*Gas station*	76
Matt	*ACM (office)*	1	*Pizza Hut*	310
Nathan	*Church*	0	*Ridgedale Mall*	104
Lisa	*Daycare*	6	*Housing Development*	144

When we looked at the data further, we found two things that accounted for this.

First, the data are not normal. Figure 9 shows the histogram of GPS readings for places with important score 5. The shape displays a skewed right distribution, known as an inverse exponential distribution. We drew similar diagrams for place with other importance scores and they show similar non-normal distribution.

Second, we manually examined the location data for a number of individual places of two types: important places that DJ-Cluster did not discover, and unimportant places with a relatively large number of readings. Table VI presents illustrative data from four subjects. They show starkly that there are going to be places for which our algorithm will do poorly. The problem may be partially due to our data collection process: sometimes when subjects visited an important place, they might have forgotten to bring their GPS phone, or it was turned off. And it also may be partially due to the way people think about places. For example, Lisa considers her children's daycare center a very important place, even though it is her husband who drops off and picks up the kids most of the time. Conversely, even though Donna spent a fair amount of time at a gas station, she simply doesn't consider filling her car with gas important; or, from another perspective, one gas station is just as good as another to her.

7.4 Timeliness

DJ-Cluster is a *batch* algorithm. In other words, it assumes the data it receives as input are all the data to be clustered. It is not *incremental*: there is no simple,

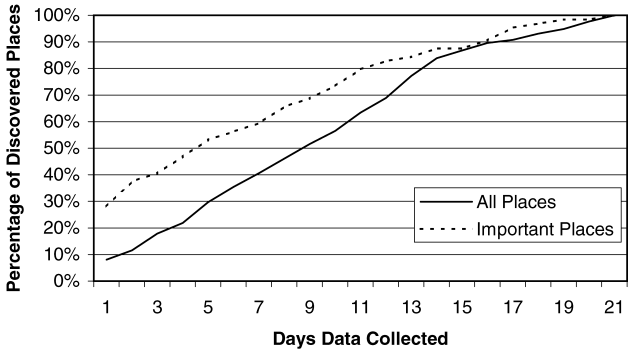


Fig. 10. Timeliness of place discovery: discovering places over time.

efficient way to start it with some clusters and so-far unclustered data and new data. We discuss this issue further below.

Thus, to evaluate timeliness of discovery, we first created a sequence of prefix datasets:

—$Dataset_1$: All the data collected on the first day of the experiment;

—$Dataset_2$: All the data collected on the first two days of the experiment;

—$Dataset_{21}$: All the data collected on the first 21 days of the experiment. This is the entire dataset we collected in the experiment.

We then applied DJ-Cluster to each of these datasets in turn. We compared the results for $Dataset_i$ to those for $Dataset_{21}$; specifically, we computed the proportion of all places and important places discovered after day i.

Figure 10 summarizes the results. It shows that discovery of new places continued throughout the collection phase. This was due in part to subjects' visiting new places throughout the course of this phase, and in part to data accumulating until at least *MinPts* readings for a place had been gathered.

It is worth noting that 30% of important places were discovered on the first day. Since subjects averaged 7.6 important places in total (see Table IV), this mostly amounts to discovering two places that all subjects considered important and for which many readings were collected nearly every day: home and work.

7.5 K-Means Results

One of the motivations to implement our algorithm, DJ-Cluster, is due to the limitations of K-Means, discussed in Section 2.2.3. We now present the performance comparison of DJ-Cluster and K-Means in discovering personal places. We first present an illustrative study with the first author's data, then present the empirical evaluation results with 8 subjects.

7.5.1 *An Illustrative Study with First Author's Data.* DJ-Cluster was run on first author's data with $MinPts = 10$ and $Eps = 10$. Twenty-one places were discovered. We then ran K-Means algorithm on the same dataset with K set to be 21. Figure 11 shows that, at the scale of a city, the places discovered by K-Means are geographically close to the baseline places. However, the

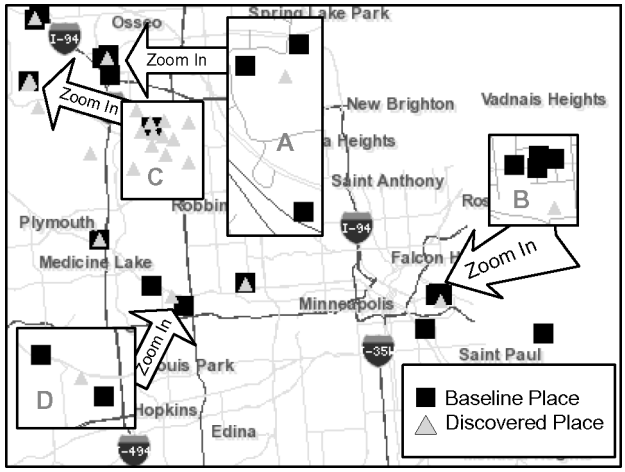


Fig. 11. Discovered places by K-Means. *A*—a cluster is formed from Cub Foods, Community Center, and BestBuy; *B*—a cluster is formed from some campus places, United Noodle, and a tax agency; *C*—many clusters are formed from home; *D*—a cluster is formed from the dentist's and Tea House.

zoomed-in windows in the figure reveal that many discovered places are at least a couple of city blocks away from the baseline places, and the mapping between baseline and discovered places is quite inexact. For example, zoomed-in window B contains four distinct baseline places in the campus area. Yet the K-Means algorithm created only one cluster in that area, and that cluster is 4-5 blocks away from the nearby baseline places. Similarly, no clusters were created for other baseline places such as Dentist, Cub Foods, and United Noodle because there were relatively few location data for these places, and they happened to be far away from the initial randomly selected cluster centers.

On the other hand, zoomed-in window C shows that K-Means created about 10 clusters for just one baseline place, "home". This is because a large proportion (more than half) of all the location data were near "home", so randomly selecting points for the initial cluster centers led to many of these points being selected.

Compared to K-Means, DJ-Cluster shows discovered places closer to each baseline place; in fact, they often actually overlap. See zoomed-in window B of Figure 12 for example. DJ-Cluster also discovered some interesting places that were not in the person's baseline places, such as Chipotle, WalMart, and an ice arena. As we shall argue later, these results provide some evidence that a purely user-driven approach to place acquisition may not be sufficient. Finally, DJ-Cluster minimized the number of spurious places discovered around traffic lights and pedestrian stops.

7.5.2 *Empirical Evaluation with 8 Subjects.* The above case study with the first author's data presented limited data suggesting the superiority of DJ-Cluster over K-Means.

However, to strengthen our case, we carried out a brief followup to our original evaluation study. Essentially, we duplicated the study for 8 of our original

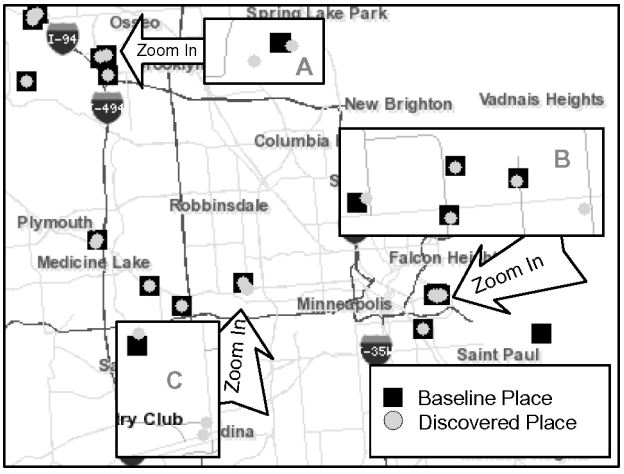


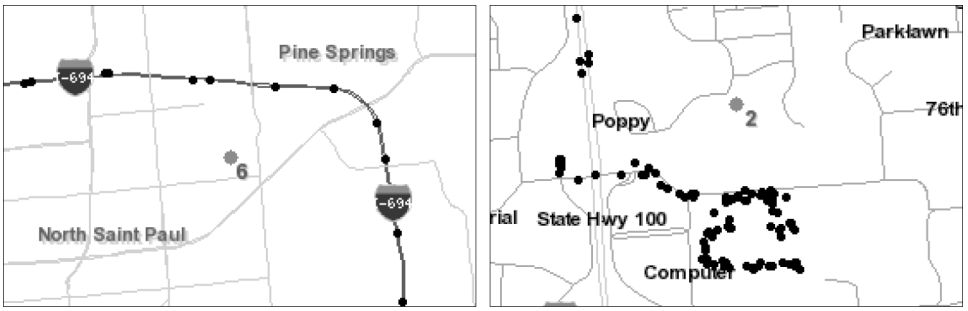
Fig. 12. Discovered places by DJ-Cluster, which correspond much more closely to the baseline places. *A*(Community center)—swimming pool and ice arena; *B*(Campus)—parking I, parking II, EE/CS, coffee shop, and Chipotle; *C*(Work)—work and 2 traffic stops.

Table VII. DJ-Cluster and K-Means: Baseline and Discovered Places for 8 Subjects. ('Mean' stands for 'Mean per subject' and 'STD' stands for 'Standard Deviation')

	K-Means			DJ-Cluster		
	Total	Mean	STD	Total	Mean	STD
Baseline(B)	135	16.9	8.9	135	16.9	8.9
Discovered(D)	135	16.9	8.9	76	9.5	4.0
Baseline Discovered(BD)	23	2.9	1.6	64	8.0	4.3
Non-baseline Meaningful discovered(DM)	0	0	0	12	1.5	2.0
Non-baseline Non-meaningful discovered(DN)	47	5.9	2.8	0	0	0
Recall (BD/B)	$23/135 = 17\%$			$64/135 = 47\%$		
Precision (BD/D)	$23/135 = 17\%$			$64/76 = 84\%$		
SurpriseFactor (DM/D)	$0/135 = 0$			$12/76 = 16\%$		

24 subjects, but used K-Means instead of DJ-Cluster; this was done in February 2006.

We again used the evaluation procedure and metrics described in section 6 to assess the accuracy of K-Means for place discovery. We ran K-Means on each of the 8 subjects' location datasets. Recall that the K-Means algorithm must be provided with the desired number of clusters as a parameter. To enable it to perform as well as possible, we set this parameter for each subject's dataset to the actual number of baseline places for that subject. Then, as for the DJ-Cluster algorithm, we printed out maps showing all the discovered places for each subject and led the subjects through the process of matching their baseline places (Table I) to the discovered places (on the map). Table VII shows the place data and the evaluation metrics.



(a) A cluster(centered at the big dot) is formed from GPS readings(small dots) on the commuting routes

(b) A cluster's center(big dot) is away from its 'real' place(readings around office building) because of the readings on the nearby roads

Fig. 13. Places formed by K-Means due to its insensitivity to the noise data.

The recall of K-Means is about 17%, significantly lower than DJ-Cluster's 47%. While this increases our confidence that DJ-Cluster was more suitable for the place discovery task, we wanted to understand why this was so. Through the interview process and by examining the results in detail, we found three reasons that account for the poor performance of K-Means. First, some clusters are formed from a set of points that are essentially noise, such as readings taken while driving on a highway (Figure 13(a)). Second, some clusters are pulled away from the "real" location by the presence of noise data that are included in the cluster (Figure 13(b)). Finally, multiple clusters may be formed for a single place with many readings; this typically happens for places such as *home* and *work*.

8. DISCUSSION AND FUTURE WORK

Our results show that automated discovery algorithms are a promising way to acquire places but that their current performance leaves considerable room for improvement. We discuss several research directions that have the potential to lead to this improvement. We also discuss where discovery algorithms may be usefully complemented by interactive place acquisition techniques.

8.1 Discovering Complex Shapes

Existing discovery algorithms [Marmasse and Schmandt 2000; Patterson et al. 2003; Liao et al. 2004; Zhou et al. 2004; Kang et al. 2004; Hightower et al. 2005] all are "point-oriented": that is, each cluster represents a single (albeit perhaps large) geographical location or point. Typically, a cluster is represented by the geographical centroid of the points in it. However, our results showed that a significant percentage of subjects' places (21%) had more complex shapes (geometries): multi-points, paths, and regions. Fundamentally, these geometries are *hierarchical places*: they are places consisting of smaller places. For example, a region for a neighborhood will contain multiple places with point geometries for the places in that neighborhood that matter to a person. Similarly,

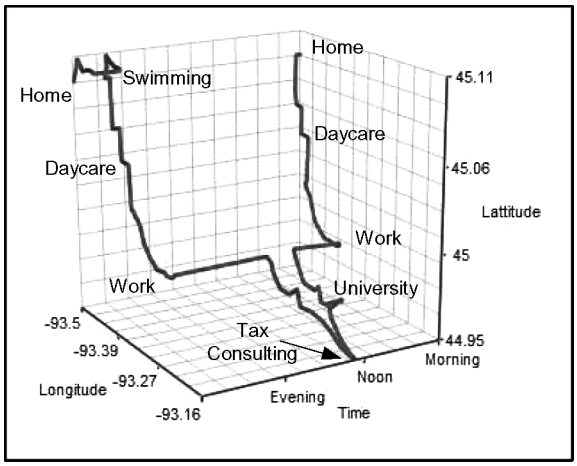


Fig. 14. A 3D visualization places discovered by a spatio-temporal clustering algorithm.

Ashbrook and Starner [2002] introduced "sublocations" concept to represent different place scopes, for example, *physics building* and *campus*. Therefore, algorithms need some way to group places into higher-level places. We are exploring several ways to extend our algorithms to do this.

First, we are looking into spatial and temporal data mining techniques, such as motion pattern detection algorithms [Gudmundsson et al. 2004]. Specifically, for *Paths*, extending existing spatial clustering algorithms to incorporate temporal data processing may be a viable solution.

We developed a spatio-temporal clustering algorithm to extract paths for sparse datasets, for example, a single day's worth of location data. Figure 14 shows a visualization of the results obtained by applying the algorithm to a day's worth of data from the first author's dataset described above. Note that, as with Kang et al. [2004], each visit to a place is treated as a distinct cluster. We don't think this is a desirable property, and we are looking for ways to avoid it.

Second, we are looking into another technique that can discover both paths and regions. We essentially recast the discovery problem as one of hierarchical clustering. For our purposes an agglomerative algorithm is appropriate. Agglomerative algorithms successively cluster results until all the data have been joined into a single cluster. In our case, DJ-Cluster produces the 0th-level set of clusters. We'll need to develop appropriate distance metrics to decide when two clusters are close enough to be joined. We currently are exploring methods based on identifying large gaps in the distances between the (0th level) places for a user.

Finally, neither of these techniques is able to discover multi-points ("all the grocery stores in my town"). We suggest that that these can be acquired interactively.

8.2 Interactive Place Acquisition

The accuracy of DJ-Cluster is superior to that of baseline algorithms such as K-Means. However, one third of even important places were not discovered after

a 3-week data collection phase. As we discussed, this is because users consider some places important even if they don't go there often. Further, the timeliness analysis shows that after 10 days, nearly 50% of all places and about 30% of important places still have not been covered. Therefore, it would be useful to integrate automated discovery with interactive techniques, allowing users to specify for themselves those places they most care about as soon as they begin using a system.

Our overall acquisition framework (see Figure 3) provides for this integration. The specific approach we are exploring currently is to develop interactive map-based interfaces to let users indicate places they're interested in, visualize places suggested by the automated discovery system, and form places with more complicated geometries from simpler places. A well-designed interface should make it easy for people to indicate that a set of dots really all belong to a *Region*, or that a sequence of dots really represents a *Path*.

8.3 Incremental Discovery

As discussed above, our analysis shows that new places were discovered throughout the data collection phase. To be more useful to a subject, we would like our algorithm to run continuously (as new data are available) and report candidate places to the user as they are discovered.

However, since DJ-Cluster was not designed to run continuously, after each run it discards as noise all points that were not added to a cluster according to the density parameters. This could lead to the following situation. Suppose *MinPts* is set to 10. Suppose further that a user's motion patterns lead to less than 10 points being collected each time the user visits a place—at one reading per minute, this is easy to imagine. Then either DJ-Cluster never discovers this place or else it must retain all data. However, the latter strategy eventually leads to unacceptable performance.

Therefore, the problem is to enable efficient incremental discovery. This is a challenging problem. The idea we are pursuing is to store and reuse results from previous computations to reduce the cost of computing new clusters. For example, DJ-Cluster's main cost is calculating the density-based neighborhood for a point; therefore, we need to create techniques that minimize the number of times we have to recompute the neighborhood for a point as new data arrive.

8.4 Discovering Important Places

As discussed in Section 7.3, some places with very few GPS readings are regarded as important places (e.g., *church*, *daycare*) by the user, while some places with relatively large number of readings (e.g., *gas station*, *Pizza Hut*) are thought to be least important.

This is not surprising to us, since DJ-Cluster is a density-based algorithm, which tends to discover places with more GPS readings, or frequent places. Important and infrequent places, shown in Figure 15, however, may not be identified by the current framework. To discover these places using a semi-automated approach, additional information other than frequency may be required. For example, the standard deviation of the time spent at a place may

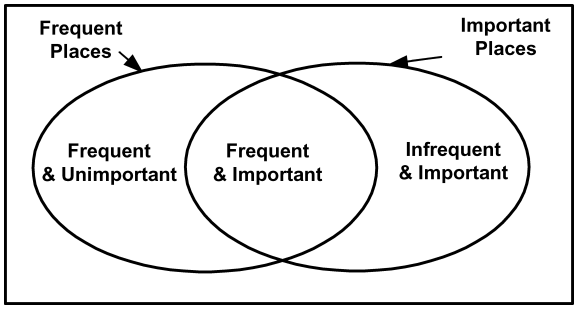


Fig. 15. Sets of frequent and important places.

be used to measure the consistency of a user's routine at the place, thus it may be a useful feature to predicate a place's importance.

9. SUMMARY

Place acquisition—obtaining physical locations and associating them with appropriate place descriptions—is an essential requirement for location-aware applications. This article described a general framework for place acquisition, then explored one approach, semi-automated discovery, in detail. The main contributions were as follows:

—an evaluation framework for place discovery algorithms that defines a set of metrics and procedures;

—a large-scale empirical study that shows the promise of our DJ-Cluster algorithm and establishes a baseline for future work;

—the identification of several important remaining challenges for place discovery algorithms and a sketch of our approach to the challenges.

APPENDIX A. PRIOR WORK

In this appendix, we detail some important prior work in place discovery resarch, namely, the exploratory approach, the fingerprinting approach, and the early work in machine learning.

A.1 An Exploratory Approach

The comMotion system [Marmasse and Schmandt 2000] consists of a device that constantly takes GPS readings. Periodically, the GPS signal is "lost". The loss of the signal is interpreted as a significant cue, namely that a building has been entered. The system maintains a history of readings, and when the signal has been lost within a given radius on three different occasions the agent infers that this location (building) is interesting. When locations are discovered, users are prompted to provide a name; they can do this either immediately or else later while viewing the location on a map. Of course, users also may judge a discovered location to be uninteresting and tell the system to ignore it. Once a location has been discovered and accepted by a user, the user can associate a to-do list with it—a prototypical example is associating a shopping list with a supermarket.

Since the cue for identifying a location is the loss of the GPS signal, only locations such as buildings can be found. Some meaningful places (such as a park or sidewalk cafe) may not cause any GPS signal loss, and thus cannot be discovered. Conversely, in so-called "urban canyons" between tall buildings, GPS signals are often weak and unreliable, which could trigger false discoveries.

A.2 Signal Fingerprinting Approaches

Nowadays, mobile devices live in various networks with ambient wireless signals, for example, WI-FI, GSM and CDMA. The attributes of wireless signals or their statistics can be uniquely identified at different locations with resolutions depending on networking technologies. Thus, they can serve as fingerprints for different locations in a given network, called *fingerprinting positioning*.

Cell-ID and access point MAC address (AP) are the most common fingerprints in the current commodity networks and have been used to represent the location of a mobile device. Trevisani and Vitaletti [2004] carried out an experiment to evaluate cell-ID-based positioning in the U.S. New York area and E.U. Rome area. Their evaluation shows that the quality of a cell-ID is often not appropriate for deployment of even very simple location-based services. The challenges remain in accuracy improvement, such as more accurate cell-ID matching. Laasonen et al. [2004] used a clique-based graph to model the cells and developed more a complex algorithm to improve the accuracy.

Hightower et al. [2005] developed an advanced fingerprinting-based approach, called BeaconPrint. In their study, a fingerprint is represented by a response-rate histogram which is an aggregate statistic based on MAC layer characterist

An advantage of fingerprinting approaches is their ability to collect location data continuously. However, these approaches suffer limitations too. One major drawback is that they do not directly collect physical location; rather, they collect radio signals and infer a "place" that is defined by radio signal statistics. No matter how accurate the inference processes are, what the fingerprinting algorithms obtain are "virtual places" represented by network signals, which need to be mapped to the physical real world to be useful. Second, cells can sometimes be as large as miles in sparsely populated areas, which causes great loss of accuracy of positioning. Third, high density of signals combined with unpredictable radio propagation in urban areas can degrade the performance of fingerprinting approaches, leading to lack of availability, like the GPS "urban canyons".

Recently, Letchner et al. [2005] extended the signal fingerprinting technique using a hierarchical Bayesian model to achieve 2–4 meters positioning precision indoors and 26–40 meters outdoors.

A.3 Earlier Machine Learning Approach

Based on their prior work [Patterson et al. 2003], Liao et al. [2004] introduced a hierarchical Markov model to learn and infer a user's daily movements from GPS data. The model allows one to infer a user's locations, such as home and work; infer a user's mode of transportation, such as foot, car, and bus; and predict when and where he will change mode. This technique is motivated by

applications that alert cognitively impaired users when they depart from their "normal" transportation routines, for example, taking a wrong bus or failing to get off the bus.

In this study, a user's locations are estimated on a street map represented by a graph-structure $S = (V, E)$, where V is the set of vertices, v_i, and E is the set of edges, e_j. Typically, vertices are intersections and the length of edges corresponds to city blocks.

A user's GPS locations are tracked and "snapped"s to a close edge e_j in the graph S. The time spent on each edge e_j is calculated and edges above a certain time threshold are sorted out. Those selected edges are paired and connected to form significant "streets". The end points of "streets" are places where potentially a transportation mode changes, such as "home", "work", "bus stations" and "parking lots".

Although the predictive models generated in Liao's study have superior learning and inferring capabilities, the graph-based location "snapping" approach has its limitations. First, this approach depends on map-generated graphs. Places that are out of the predefined maps therefore cannot be discovered. Second, places have to be part of the "streets" or edges in the graph, while in the real world, places (such as a grocery store, buildings on campus) are not always along a street.

REFERENCES

ASHBROOK, D. AND STARNER, T. 2002. Learning significant locations and predicting user movement with GPS. In *Proceedings of ISWC*. 101–108.

ASHBROOK, D. AND STARNER, T. 2003. Using GPS to learn significant locations and predict movement across multiple users. *Pers. Ubiquitous Comput. 7*, 5, 275–286.

BURRELL, J. AND GAY, G. 2001. E-graffiti: Evaluating real-world use of a context-aware system. *Interact. Comput.*

ESPINOZA, F., PERSSON, P., SANDIN, A., NYSTRM, H., CACCIATORE, E., AND BYLUND, M. 2001. Geonotes: Social and navigational aspects of location-based information systems. In *Proceedings of UbiComp*. 2–17.

ESTER, M., KRIEGEL, H.-P., SANDER, J., AND XU, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*. 226–231.

GENEREUX, R., WARD, L., AND RUSSELL, J. 1983. The behavioral component in the meaning of places. *J. Environ. Psych. 3*, 43–55.

GRISWOLD, W., SHANAHAN, G., BROWN, S., BOYER, R., RATTO, M., SHAPIRO, R., AND TRUONG, T. 2003. Activecampus—Experiments in community-oriented ubiquitous computing. Tech. rep., UC San Diego.

GUDMUNDSSON, J., VAN KREVELD, M., AND SPECKMANN, B. 2004. Efficient detection of motion patterns in spatio-temporal data sets. In *Proceedings of ACMGIS*. 250–257.

HIGHTOWER, J. 2003. From position to place. In *Proceedings of the Workshop on Location-Aware Computing*.

HIGHTOWER, J., CONSOLVO, S., LAMARCA, A., SMITH, I., AND HUGHES, J. 2005. Learning and recognizing the places we go. In *Proceedings of the UbiComp*. 159–176.

JUNG, Y., PERSSON, P., AND BLOM, J. 2005. Dede: Design and evaluation of a context-enhanced mobile messaging system. In *Proceedings of CHI*. ACM Press, New York, 351–360.

KANG, J. H., WELBOURNE, W., STEWART, B., AND BORRIELLO, G. 2004. Extracting places from traces of locations. In *Proceedings of WMASH*. 110–118.

KRAMER, B. 1995. Classification of generic places: Explorations with implications for evaluation. *J. Environ. Psych. 15*, 3–22.

LAASONEN, K., RAENTO, M., AND TOIVONEN, H. 2004. Learning and recognizing the places we go. In *Proceedings of Pervasive Computing*. 287–304.

LETCHNER, J., FOX, D., AND LAMARCA, A. 2005. Large-Scale Localization from Wireless Signal Strength. In *Proceedings of AAAI*. 15–20.

LIAO, L., FOX, D., AND KAUTZ, H. A. 2004. Learning and inferring transportation routines. In *Proceedings of AAAI*. 348–353.

LIAO, L., FOX, D., AND KAUTZ, H. A. 2005b. Location-based activity recognition. In *Proceedings of NIPS*.

LIAO, L., FOX, D., AND KAUTZ, H. A. 2005a. Location-based activity recognition using relational markov networks. In *Proceedings of IJCAI*. 773–778.

LUDFORD, P., FRANKOWSKI, D., REILY, K., WILMS, K., AND TERVEEN, L. 2006. Because I carry my cell-phone anyway: Effective everyday task management. In *Proceedings of CHI 2006*.

MARCHIONINI, G. 2004. From information retrieval to information interaction. In *Proceedings of the 26th European Conference on IR Research*. 1–11.

MARMASSE, N. AND SCHMANDT, C. 2000. Location-aware information delivery with commotion. In *Proceedings of HUC*. 157–171.

MILENOVA, B. L. AND CAMPOS, M. M. 2002. O-cluster: Scalable clustering of large high dimensional data sets. In *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*. IEEE Computer Society, Washington, DC, 290.

NAAMAN, M., SONG, Y. J., PAEPCKE, A., AND GARCIA-MOLINA, H. 2004. Automatic organization for digital photographs with geographic coordinates. In *Proceedings of JCDL*. 53–62.

NEWMAN, W. M. 1997. Better or just different? On the benefits of designing interactive systems in terms of critical parameters. In *DIS '97: Proceedings of the Conference on Designing Interactive systems*. ACM, New York, 239–245.

PATTERSON, D., LIAO, L., FOX, D., AND KAUTZ, H. 2003. Inferring high-level behavior from low-level sensors. In *Proceedings of UbiComp*. 73–89.

RIEMAN, J. 1993. The diary study: A workplace-oriented research tool to guide laboratory efforts. In *Proceedings of CHI*, 321 – 326.

SANDER, J., ESTER, M., KRIEGEL, H.-P., AND XU, X. 1998. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining Knowl. Disc. 2*, 169–194.

SCHILLER, J. H. AND VOISARD, A., Eds. 2004. *Location-Based Services*. Morgan Kaufmann, San Francisco, CA.

SOHN, T., LI, K. A., LEE, G., SMITH, I., SCOTT, J., AND GRISWOLD, W. G. 2005. Place-its: A study of location-based reminders on mobile phones. In *Proceedings of UbiComp*. 232–250.

TREVISANI, E. AND VITALETTI, A. 2004. Cell-ID Location Technique, Limits and Benefits: An Experimental Study. In *WMCSA '04: Proceedings of the 6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'04)*. (Washington, DC). IEEE Computer Society Press, Los Alamitos, CA, 51–60.

WEILENMANN, A. H. AND LEUCHOVIUS, P. 2004. I'm waiting where we met last time: Exploring everyday positioning practices to inform design. In *Proceedings of NordiCHI*. 33–42.

ZAÏANE, O. R., FOSS, A., LEE, C.-H., AND WANG, W. 2002. On data clustering analysis: Scalability, constraints, and validation. In *Proceedings of PAKDD*. 28–39.

ZHOU, C., FRANKOWSKI, D., LUDFORD, P., SHEKHAR, S., AND TERVEEN, L. 2004. Discovering personal gazetteers: An interactive clustering approach. In *Proceedings of ACMGIS*. 266–273.

ZHOU, C., LUDFORD, P., FRANKOWSKI, D., AND TERVEEN, L. 2005a. An experiment in discovering personally meaningful places from location data. In *Proceedings of CHI, Extended Abstract*. 2029–2032.

ZHOU, C., TERVEEN, L., LUDFORD, P., AND FRANKOWSKI, D. 2005b. An experiment in exploring people's concepts of place relating to physical locations. In *Proceedings of INTERACT*. 886–898.

ZHOU, C., TERVEEN, L., LUDFORD, P., AND FRANKOWSKI, D. 2005c. Talking about place: An experiment in how people describe places. In *Proceedings of Pervasive, Short Paper*.

Received December 2005; revised March 2006, July 2006, October 2006; accepted November 2006