

Location Privacy in Geospatial Decision-Making*

Cyrus Shahabi and Ali Khoshgozaran

University of Southern California
Department of Computer Science
Information Laboratory (InfoLab)
Los Angeles, CA, 90089-0781
[shahabi, jafkhosh]@usc.edu

Abstract. Geospatial data-sets are becoming commonplace in many application domains, especially in the area of decision-making. Current state-of-the-art in geospatial systems either lack the ease-of-use and efficiency or sophisticated querying and analysis features needed by these applications. To address these shortcomings, we have been working on a generic and scalable geospatial decision making system dubbed *GeoDec*. In this paper, we first discuss many of the new features of GeoDec, particularly its spatial querying utilities. Next, we argue that in some applications, a user of GeoDec may not want to reveal the location of the query and/or its result set to the GeoDec server to preserve his/her privacy. Hence, for GeoDec to remain applicable in these scenarios, it should be able to evaluate the spatial queries without knowing the locations of the query and/or results. Towards this end, we present our novel space-encoding approach which would enable the GeoDec server to evaluate the spatial queries *blindly*.

1 Introduction

Geospatial information, in the form of traditional maps, have been used for several centuries for decision-making tasks. The oldest map is known to be from 2500 B.C. of a city near Babylon. In the past forty years, the field of Geospatial Information Systems (GIS), with ESRI leading the industry, has been increasing the role of geospatial information in decision-making tasks by allowing their digital manipulation. However, it was not until the last couple of years that the power of digital geospatial information has been brought to mass population through online map services such as Yahoo! maps [Yah] and most recently Google Earth [Goo]. Nowadays, you cannot see a news story without a screen-shot of Google Earth.

While the GIS industry targets the two ends of user population, the expert and the naive, there is a large group of users in the middle who are satisfied with neither the bare navigation/visualization features of Google Earth-like applications, nor the unscalable and non-generic solutions of GIS utilities. The reason for the lack of a middle-ground

* This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC), IIS-0238560 (PECASE), IIS-0324955 (ITR), and unrestricted cash gifts from Google and Microsoft. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

solution is the hard fundamental challenges in building a system that is generic, easy-to-use and scalable while at the same time can perform complex geospatial and temporal analysis efficiently.

Towards this end, in April 2005, we embarked on a multidisciplinary project, called Geospatial Decision-Making (GeoDec) [SCC⁺06]. GeoDec aims to enable geospatial decision-making for users in a variety of geographic application domains, including urban planning, emergency response, military intelligence, simulation and training. Since then, we have developed an end-to-end system that allows navigation through a 3D model of a location (e.g., a city) and enables users to issue queries and retrieve information as they navigate about the area. GeoDec helps the user to be immersed in an information-rich environment which facilitates his/her decision-making. The immersion in GeoDec system is the result of applying relevant techniques developed independently in the fields of databases, artificial intelligence, computer graphics and computer vision to the problem described above. In particular, the system seamlessly integrates satellite imagery, accurate 3D models, textures, video streams, road vector data, maps and point data for a specific geographic location. In addition, users can intuitively interact with the system using a glove-based interface and a large screen to issue a variety of spatial queries.

This paper has two main parts. In the first part, we report on some of our new developments in the GeoDec project. In the second part, we focus on one of the new challenges we face in any geospatial decision-making system including GeoDec: *Location Privacy*. The problem is that some users of geospatial decision-making systems may not want to reveal their locations or the locations of their query results to the system's location server. The challenge is how to provide all the spatial query features of GeoDec to the users without revealing the users locations to the GeoDec server. Here, we define some general metrics to evaluate any location privacy scheme and then discuss our recently proposed privacy model based on space encoding.

2 Geodec: Enabling Geographical Decision-Making

In previous work [SCC⁺06], we developed a multidisciplinary project, called GeoDec, for geospatial decision-making. The goal of GeoDec is to provide the needed information for decision makers in a variety of geographic application domains, including location-based services, urban planning, emergency response, military intelligence, simulation and training. The system, like Google Earth, supports the navigation through a 3D model of a geographical location (e.g., a city). In addition, with GeoDec one can also issue queries and retrieve information as he/she navigates about an area. In particular, the system seamlessly integrates satellite imagery, accurate 3D models, textures and video streams, road vector data, maps, point data, and temporal data for a specific geographic location (see Figure 1).

2.1 Geodec Architecture

Figure 2 depicts the 3-tier architecture that is used in GeoDec. This architecture consists of a data tier, an integration tier, and a presentation tier. The data tier focuses on

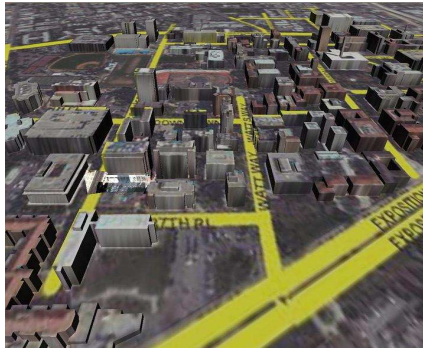


Fig. 1. A Screen-shot of a Geographic Location in GeoDec’s User Interface.

the efficient storing and indexing of geospatial data. The integration tier provides the ability to efficiently query and integrate heterogeneous geospatial sources. And the presentation tier provides a uniform representation so that query results can be visualized in commercial systems (such as Microsoft Virtual Earth [Vir] or Google Earth [Goo]) or in more specialized user interfaces.

I. Query Formulation and Visualization (Negaah): Negaah is a visualization interface for GeoDec that allows a user to navigate the 3D environment in real-time, and submit some customized queries on geospatial data based on a user-defined selection area. The user can selectively query and display different layers of information, and move forward or backwards in time. Negaah also supports formulating more sophisticated spatial queries such as KNN query for points of interest and the line of sight query around the selected query point.

II. Spatio-Temporal Query Middleware (Jooya): All the queries in Negaah are directed to GeoDec’s information mediator/spatio-temporal database component through a middleware layer, Jooya. Jooya offers a universal way of specifying the type of query (e.g., nearest neighbor, range, shortest path, etc.), as well as its parameters and retrieves the results back in a unified format. Depending on the user’s query type, Jooya either sends a query to our Spatio-temporal Database Manager (Darya), or to Prometheus [TAK04], our information mediator (which provides a uniform query interface to a set of web sources that contain information about a geographic location). Examples of queries sent to Darya are queries for infrequently updated or bulky data such as road network data, video metadata, and 3D building models. On the other hand, Jooya uses Prometheus to query highly dynamic data such as live traffic information obtained via a variety of web sources. Jooya returns results in a standard format (specifically, Google Earth’s KML format) and this architecture enables any visualization layer that can support KML to sit on top of Jooya for its integrated query and access needs (Figure 2). Jooya, however, would need to include a customized query-interface blade for each new GUI. This enables Jooya to act as an interface for web based geospatial GUIs such as Google Earth as well.

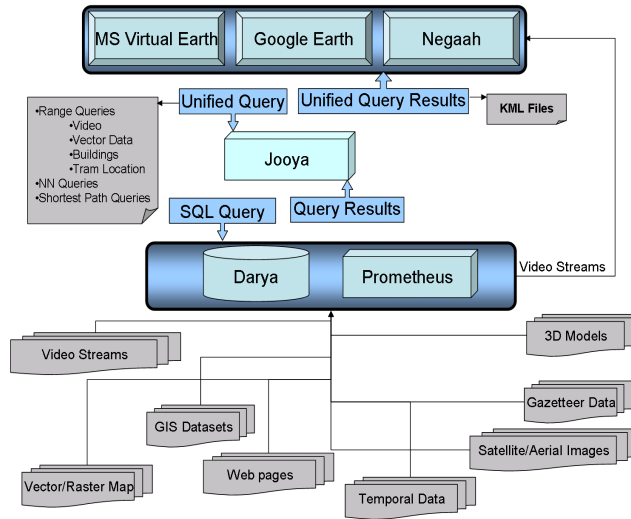


Fig. 2. Geodec Architecture.

III. Spatio-Temporal Database Manager (Darya): As part of GeoDec’s back-end, we have also developed a Spatio-temporal Database Manager (Darya). This module is in charge of managing spatio-temporal data stored in a database, a task that includes data modeling, storage and retrieval (querying). For storage and transmission, our main focus is now on bulky vector data such as road networks. Mainly Darya embeds a multi resolution vector data compression technique which effectively compresses the result of query windows sent to it, taking into account the client’s display resolution. Our vector compression approach enables Darya to efficiently store and transfer bulky data. Furthermore Darya stores several heterogeneous data sources such as satellite/areal imagery, temporal data, GIS gazetteer data, 3D models, video streams, vector/raster maps etc. which allows the query middleware to integrate several data sources and create a compelling and information-rich representation of the geographic area.

One of the key benefits of the above architecture is its scalability. As discussed above, separating these three layers allows Geodec to act as a tool for rapid construction of an information-rich geographic space. The steps to reach that goal include adding the necessary spatio-temporal data to Darya (e.g., satellite imagery, vector data, etc.) and implementing the necessary query-interface blades in Jooya. Once the additions are made, Negaah (or any other proprietary user interface that can communicate with Jooya) will allow users to interactively work with and query the recently added geographic area.

2.2 GeoDec Querying Features

The incorporation of several data sources at the database layer, blended with the integration modules at the middle layer allows Geodec to respond to a wide range of sophisticated queries. In this section we briefly study these queries.

Geodec's query formulation and visualization interface allows users to construct different types of spatial queries for their area of interest. In order to facilitate the process of decision-making Geodec allows users to query Darya for any spatial data relevant to a *specific event* which is of interest to the user. This way, a single generic event query will be translated into several heterogeneous low-level queries by the middle layer and are sent to Darya. Jooya then receives the different pieces of query results returned from the spatio-temporal database layer and compiles the results into different information layers (such as 3D models, moving object trajectories, video feeds, vector data, etc.) for Negaah. When a user notices an event taking place in a geographic area, he uses Geodec to specify different pieces of information he has about the event and what he is expecting to obtain from the system. For instance he uses a bounding box to specify the region of interest while navigating in the 3D model. He can also specify a time duration in which the event has taken place and any other possible piece of information about the event. The user then selects different layers of information each of which corresponding to one (or more) spatial queries. The query results are then provided to the user in different layers to allow him to view/hide each dimension of result returned. We now review several types of spatial datasets and their associated queries to show how an event query is resolved by several spatial queries, using the architecture described in Section 2.1.

3D Building Models: The spatio-temporal database layers store the 3D building information of the geographic area texture mapped with satellite imagery of the area. Each building is associated with a timeframe of its existence. This way we can move back and forth in time and see the changes in an area caused by constructing/demolishing buildings. Once a user specifies the bounding box and timeframe for each event, we can reconstruct the actual 3D space for user's area of interest at the time the event had taken place.

Multi-resolution Vector Data Storage and Retrieval: Darya stores vector data (e.g., point data, spatial extents and road-network data) obtained from Navteq [Nav] for the entire United States. One challenge with storage and access of vector data is their large size. For example, the Santa Monica Blvd. in Los Angeles area, which is only 1.115 miles long, consists of 234 line segments in the Navteq street data set. Since storage is cheap, there is no reason to reduce the size of these data just to save space. However, transferring these bulky data over the network and then rendering it is a slow process. This is especially a waste of resources if the display resolution and zoom level (in case of GeoDec GUIs) is not fine-grained enough to require all the details. Hence, during an offline process, we utilize our multi-resolution vector data compression scheme [KKSS06] to construct different levels of detail for this data so that later, during the query time, Darya can minimize the communication/rendering overhead by dynamically choosing the right level of detail based on user's query so that we only retrieve and transmit what is absolutely needed for the display. The vector data is then

superimposed on top of the areal image for the geographic location and are added to the 3D model of the area.

Moving Object Storage/Retrieval: Another important dataset which is maintained by Darya is the moving object data. For Geodec, we are tracking and storing the location of several moving objects (such as university trams) which are equipped with GPS devices. Therefore when a user specifies a moving object of interest for an event query, we can construct the trajectory of moving object locations for the query timeframe and allow the user to move back and forth in time to see the exact location of the moving object at any given time in past. It also allows users to track the live location of all moving objects in an area of interest as well. The trajectory of the selected moving object will then be added as a separate layer to the query result.

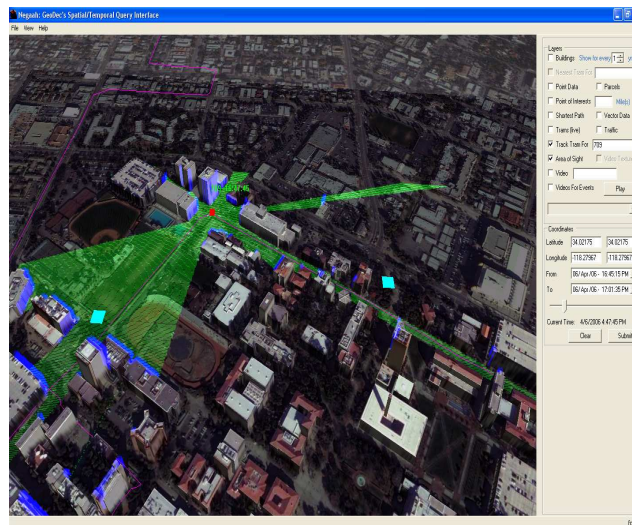


Fig. 3. The Result of a Line-of-Sight Query in Geodec.

Point Data Integration: Darya incorporates gazetteer data (such as building names, parcel information, points of interest, etc.) into its repository to provide users with a wide range of location-based services and spatial queries. Users can use Geodec to issue KNN or range queries associated with each event by specifying a query point in the 3D space. This allows users to view several points of interest deemed related to the event query result.

Video Feeds: Darya also maintains a database of video streams for several surveillance cameras installed in the different locations of an area. Therefore while responding to an event query, Darya queries its video server and retrieves any video surveillance feed associated with the event that might be of interest to the user and sends the video information (e.g., video feed, camera location and coverage area) to the GUI. In order to find the relevant camera feeds, Darya performs a spatio-temporal join between the

event and the video feeds. The available video feeds are then highlighted in the area to be viewed by the user.

Line of Sight: In order to further enhance the capabilities of Geodec in dealing with dynamic data related to an event, Geodec allows users to associate a line of sight query to any moving object associated to an event. This way the line of sight module is blended with the moving object module to allow users to see a color coded 3D view of their area of interest based on whether the points in space are visible from the location of the moving object or not. As the user moves back and forth in time (i.e., the location of the moving object changes), Negaah dynamically updates the coloring of the 3D space (see Figure 3).

As described above, blending different *modalities* of information together, Jooya constructs an immersive result set for the user which is highly interactive. Ultimately, Geodec allows users to save their event queries and their associated data layers for future use.

3 Location Privacy

As discussed in Section 2, one of the main utilities of GeoDec is its rich spatial querying facility. An important requirement for successful evaluation of spatial queries is to know the exact location of the query point(s). With location-based services, for example, the query location is usually the location of the user operating a portable device such as a cell-phone, a PDA, a car navigation system or a laptop. With the advent of inexpensive GPS devices, many of these portable devices can incorporate (and already have incorporated) GPS systems. Hence, the location of the user (or query point) can be accurately identified and reported to a location service provider. In particular, the user of GeoDec can only install its GUI, e.g., Negaah, on his/her PDA or laptop and then use it to issue customized spatial queries based on his/her location. The problem here is that the user would then need to reveal his/her location to other layers of GeoDec, such as Jooya and Darya, which are installed on other servers and maintained by perhaps untrusted entities. This has major privacy implications to the extent that some users would prefer to avoid the potential benefits, not to compromise their location privacy. Therefore, it is essential for Geodec to be able to offer two modes of operations where it can both resolve a user's query with strict location privacy requirements blindly (i.e., without knowing their identity, location or query result), and operate in a normal non-private mode (i.e., in the original 2D space).

Recently, we proposed a fundamentally new approach to evaluate spatial queries, without revealing the original location of the user [KS07]. We term this novel method of performing spatial queries as the "*blind evaluation*" of spatial queries. The main idea behind our approach is to transform the original space and query location to a different space in which the results of the queries remain unchanged. Hence, the location server can perform the queries in the transformed space without acquiring any knowledge on the identities and locations of either the query point(s) or the result set. We formulate our transformation approach into a framework analogous to that of the conventional encryption schemes. That is, we separate the transformation *algorithm* from the transformation *key* for our space transformation approach. Hence, each client, hav-

ing access to both the transformation key and algorithm, can apply the transformation on its location (i.e., encrypting its location) without requiring a trusted third party. The un-trusted location server, however, not having access to the transformation key, cannot acquire the original locations of query points and other points in the space even if the transformation algorithm is known. Meanwhile, by standing on the shoulders of the encryption giants, we benefit from all the techniques developed in the past two decades for encryption-key management, maintenance, distribution and security.

Current encryption schemes [IW06] cannot maintain the space distance property and hence cannot efficiently evaluate spatial queries in the transformed space. On the other hand, the currently proposed location privacy approaches [GG03, GL04, MCA06, Mok06, BWJ05, GL, CBP, BS03] rely on an intermediate third-party. Our approach is a fundamentally new encryption/transformation methodology that maintains the distance properties of space and hence is both efficient and free of the reliance on a third party intermediary.

Let us formally define our hypothesis.

MAIN HYPOTHESIS. There exists a one-way transformation that can encode the 2-D space of static and dynamic objects in which spatial queries can be evaluated privately, accurately and efficiently. \square

Towards this end, here, we first clearly define the terms used in our hypothesis: non-reversible transformation, static and dynamic objects, and spatial queries. Subsequently, we define our privacy, accuracy and efficiency metrics. Next, we briefly discuss our space encoding approach based on Hilbert Curve [Hil91] and its vulnerability issues. In a recent publication [KS07], we conducted some preliminary evaluation of our approach assuming *static* objects and KNN queries. We briefly summarize the main observations in Sec. 3.7. Our ultimate goal is to extend our approach to support any spatial query on both static and *dynamic* objects.

3.1 Definitions of Terms in the Main Hypothesis

A transformation is *one-way* if it can be easily calculated in one direction (i.e., the forward direction) and is computationally impossible to calculate in the other (i.e., backward) direction [Sti02]. The process of transforming the original space with such a one-way mapping can be viewed as *encrypting* the elements of the 2-D space. With this view, in order to make decryption possible and efficient the function has to allow fast computation of its inverse given some extra knowledge, termed *trapdoor* [Sch84]. In practice, many one-way transformations may be reversible even without the knowledge of the trapdoor but the process must be too complex (equivalent to exhaustive try) to make such transformation computationally secure.

A *static* object is defined by a point or a polygon in 2-D space, e.g., with latitude and longitude, and its position does not change over time, e.g., location of a restaurant. A *dynamic* object is the same as a static one except that its position changes over time, e.g., a moving vehicle.

Spatial queries can be divided into two main classes. The first class of spatial queries consists of nearest-neighbor (NN) query and its variations. These queries search for data objects that minimize a distance-based function with reference to one or more query objects (e.g., points). Examples are *K* Nearest Neighbor (KNN) [RKV95, HS99],

Reverse k Nearest Neighbor (R k NN) [KM00, SAA00, TPL04], k Aggregate Nearest Neighbor (k ANN) [PTMH05] and skyline queries [BKS01, PTFS05, SS06]. The second class is the spatial range queries. This includes identifying a range, as a circle with a center point and a radius, a rectangle with a corner point and width and height, or other polygon shapes with a list of points (i.e., vertices).

To evaluate all these spatial queries blindly, we mainly need to hide the location of the point or points in the query and response. Hereafter, without the loss of generality we define our metrics (Secs. 3.2 to 3.4) and discuss our space transformation approach (Secs. 3.5 and 3.6) assuming the K Nearest-Neighbor query (KNN), where we look for all the k closest points to a query point. The same approach and metrics can be used for other types of spatial queries with minor modifications. However, to evaluate our approach given the metrics, we need to study each query type individually. So far, as discussed in Sec. 3.7, we have only studied and evaluated our approach assuming KNN queries. In future, we plan to extend and evaluate our approach for other types of spatial queries. For now, we start by a formal definition of KNN.

Given a set of static objects $S = (o_1, o_2, \dots, o_n)$ in 2-D space, the KNN query with respect to query point q finds a set $S' \subset S$ of K objects where for any object $o' \in S'$ and $o \in S - S'$, $D(o', q) \leq D(o, q)$ where D is the Euclidean distance function. In a typical KNN query scenario, the static objects represent points of interest (POI) and the query points represent user locations.

3.2 Privacy Metrics

We now formally define our privacy metrics with which we evaluate our proposed approach.

Metric 1. u -anonymity: While resolving a KNN query, the user issuing the query should be indistinguishable among the entire set of users. That is, for each query Q , $P(Q) = \frac{1}{M}$ where $P(Q)$ is the probability that query Q is issued by a user u_i and M is the total number of users. Note that satisfying this metric ensures the server does not know which user queried from a point q_i ; however, we also need to ensure that the server does not know which point the query Q is issued from. This requirement is captured in Metric 2.

Metric 2. a -anonymity: While resolving a KNN query, the location of the query point should not be revealed. That is, for each query Q , $P'(Q) = \frac{1}{\text{area}(A)}$, where A is the entire region covering all the objects in S , and $P'(Q)$ is the probability that query Q was issued by a user located at any point inside A .

Note that Metrics 1 and 2 impose much stronger privacy requirements than the commonly used K -anonymity [SS, GL, GG03, KGMP06, BS03, GL04, Mok06, MCA06], in which a user is indistinguishable among K other users or his location is blurred in a cloaked region R . The above metrics for location privacy are free of factors such as K and R . They are in fact identical to an extreme case of setting $R = A$ for spatial cloaking, and an extreme case of setting $K = M$ for K -anonymity.

Metric 3. Result set anonymity: The location of all points of interest in the result set should be kept secret from the location server. More precisely $\dot{P}(o_j) = k/n$ for $j = 1 \dots n$ where $\dot{P}(o_j)$ is the probability that o_j is a member of the result set of size k for query Q and n is the total number of POI's.

Definition 1. Blind evaluation of KNN: We say a KNN query is blindly evaluated if the *u-anonymity*, *a-anonymity* and *result set anonymity* constraints defined above are all satisfied. In blind evaluation of KNN, the identity and location of the query point as well as the result set should not be revealed.

We term our approach *blind evaluation of KNN queries* because it attempts to prevent any leak of information to essentially blind the server from acquiring information about a user’s location. The following example shows how the above properties should be satisfied in a typical KNN query. Suppose a user asks for his 3 closest gas-stations. In this case a *malicious entity* should acquire neither the location of the user (i.e., *a-anonymity*) nor its identity (i.e., *u-anonymity*) nor the actual location or identity of any of the 3 closest gas stations in the response set (i.e., *result set anonymity*) while the user should receive the actual points of interest matching his query.

3.3 Accuracy Metrics

Definition 2. Suppose the actual result of a KNN query, issued by a user located at point Q is $R = (o_1, o_2, \dots, o_K)$, and it is approximated by a transformation T as $R' = (o'_1, o'_2, \dots, o'_K)$. T is KNN-invariant if it yields acceptable values for the following two metrics:

Metric 1: The Resemblance, denoted by α , defined as

$$\alpha = \frac{|R \cap R'|}{|R|} \tag{1}$$

where $|R|$ denotes the size of a set R . In fact α measures what percentage of the points in the actual query result set R are included in the approximated result set R' .

Metric 2: The Displacement, denoted by β , defined as

$$\beta = \frac{1}{K} \left(\sum_{i=1}^K \|Q - o'_i\| - \sum_{i=1}^K \|Q - o_i\| \right) \tag{2}$$

where $\|Q - o_i\|$ is the Euclidean distance between the query point Q and o_i . Therefore β measures how *closely* R is approximated by R' on average. Obviously, since R is the ground truth, $\beta \geq 0$.

Although there is no fixed threshold for acceptable α and β values, depending on the application and the scenario, certain values may or may not be considered satisfactory. In [KS07], we evaluated our approach against these two metrics and showed that it is an accurate enough KNN-invariant transformation.

3.4 Efficiency Metrics

Our main efficiency metrics are the well-known, widely practiced *query response time* and *server throughput*. Due to lack of space and popularity of these metrics, we do not discuss them further. However, in general, a space encoding technique that results in $O(n)$ computation complexity (where n is the total number of points in space) to answer spatial queries is unacceptable. Using hierarchical index structures, e.g., R-Trees, the computation complexity can usually be reduced to logarithmic in non-encrypted spaces. Hence, ideally we would like to achieve such complexity in the encrypted space.

3.5 Space Transformation Using Dual Hilbert Curves (STUDHC)

Introduced in 1890 by an Italian mathematician G. Peano [Sag94], space filling curves belong to a family of curves which pass through all points in space without crossing themselves. The important property of these curves is that they retain the *proximity* and *neighboring* aspects of the data. Consequently, points which lie close to one another in the original space mostly remain close to each other in the transformed space. One of the most popular members of this class is Hilbert curves [Hil91] since several studies show the superior clustering and distance preserving properties of these curves [LK01, Jag90, FR89, MvJFS01].

Similar to [MvJFS01] we define H_d^N for $N \geq 1$ and $d \geq 2$, as the N^{th} order Hilbert curve for a d -dimensional space. H_d^N is therefore a linear ordering which maps a d -dimensional integer space $[0, 2^N - 1]^d$ into an integer set $[0, 2^{Nd} - 1]$ as follows: $H = \ell(P)$ for $H \in [0, 2^{Nd} - 1]$, where P is the coordinate of each point in the d -dimensional space. We call the output of this function its *H-value*. Note that it is possible for two or more points to have the same H-value in a given curve.

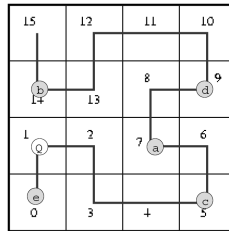


Fig. 4. A H_2^2 Pass of the 2-D Space.

As mentioned above, our motivating application is location privacy and therefore we are particularly interested in 2-D space and thus only deal with 2-D curves ($d = 2$). Therefore $H = \ell(X, Y)$ where X and Y are the coordinates of each point in the 2-D space. Figure 4 illustrates a sample scenario showing how a Hilbert curve can be used to transform a 2-D space into H-values. In this example, points of interest (POI) are traversed by a second order Hilbert curve and are *indexed* based on the order they are visited by the curve (i.e., H in the above formula). Therefore, in our example the points a, b, c, d, e are represented by their H-values 7, 14, 5, 9 and 0, respectively. Depending on the desired resolution, more fine-grained curves can be recursively constructed.

An important property of a Hilbert curve that makes it a very suitable tool for our proposed scheme is that ℓ becomes a one-way function if the curve parameters are not known. These parameters, which collectively form a *key* for this one-way transformation, include the curve's starting point (X_0, Y_0) , curve orientation θ , curve order N and curve scale factor Γ . We term this key, *Space Decryption Key* or *SDK* where $SDK = \{X_0, Y_0, \theta, N, \Gamma\}$.

In [KS07], we showed that two Hilbert curves, termed *dual curve*, where one is a 90 degree rotation of the other one, would actually result in a more accurate encryption of space without compromising its resilience.

Use of Hilbert curves to protect users location information is also suggested by [KGMP06]. However, [KGMP06] bears significant difference to our work in several aspects. First, it uses Hilbert curves to construct the anonymization of spatial region and to make a user k -anonymous. It does not use it as a space encryption algorithm. Basically, it uses $k - 1$ closest H-values around a user to come up with the k -anonymity set. Second, it uses an anonymizer between the users and untrusted location server to blur user locations. This is what we are trying to avoid. Third, it satisfies the k -anonymity and does not achieve a -anonymity and u -anonymity metrics proposed in Sec. 3.3. Finally, it does not transform points-of-interests and users locations into another space in order to preserve location privacy. Instead, it utilizes efficient cloaking techniques to compute spatial queries for a region that includes user location.

3.6 Vulnerability of STUDHC

STUDHC is a One-Way Transformation: A malicious entity, not knowing our transformation key, has to exhaustively check for all combinations of curve parameters to find the right curve by comparing the H-values for all points of interest. As we show in Theorem 1, we make it computationally impossible to reverse the transformation and get back the original points. Even a nominal error in approximating curve parameters will generate a completely different set of H-values.

THEOREM 1. The complexity of a brute-force attack to find the transformation key discussed above is $O(2^{4p})$ where p is the number of bits used to discretize each parameter.

PROOF. Please refer to [KS07] for the proof. \square

Key Management, Maintenance, Distribution and Security:

One advantage of building our model based on encryption schemes is that once we have the concept of the *encryption key*, we can immediately benefit from all the techniques developed and matured in the past two decades in managing keys. For example, one concern might be what happens if the space transformation key is compromised. That is, malicious users can exist in the system trying to subscribe to the service and acquire the key pairs to share them with the location server. Similar to all other encryption keys used widely in mobile devices, our transformation key pair is not accessible by users and is kept in modules in charge of decryption inside their devices' hardware. These devices are all tamper-proof and hence the above scenario will not happen.

Similar arguments can be made for issues in securely distributing, maintaining and updating the keys. We consider all issues related to key management beyond the scope of this paper as they are already being investigated actively by the encryption community and several practical techniques are already adapted by the industry. Note that there is a huge difference between having an *off-line* trusted entity that manages and maintains keys and a trusted *on-line* intermediary that intercepts all communications. While the former is an integral component of many encryption schemes, the latter is a major security flaw. Furthermore, our trusted entity does not need to know user locations/identity simply because there is no need for it to anonymize such data.

Reverse Engineering and Use of Known Landmarks: One of the classic known attacks to unknown transformations is through the use of known landmarks. In the most powerful form of this attack, an attacker subscribes to the service as a client and conspires with the untrusted location server to probe himself with known landmarks. However, note that in our proposed scheme the clients cannot get to know the value of the key they are using to encrypt their locations (e.g., by making their communication devices tamper proof) and thus they do not get to know their encrypted location in the transformed space (in order to share it with the location server). Furthermore, since the clients communicate with the location server through pseudonyms, the server cannot trace back a received query point (such as the one sent from the attacker) to a client to infer its location in the original space. Therefore, no matter how many landmarks are used, this attack will not reveal the key to the attacker.

3.7 Preliminary Evaluation: Privacy, Accuracy and Efficiency of STUDHC

In [KS07], we reported on our preliminary evaluation of STUDHC assuming KNN queries and static objects. In this section, we briefly discuss the main observations.

THEOREM 2. Using an H_2^N Hilbert curve to encode the space satisfies the *a-anonymity*, *u-anonymity* and *result set anonymity* properties defined in Sec. 3.2.

PROOF. Please refer to [KS07] for the proof. \square

In [KS07], we also performed several experiments with real-world datasets to evaluate the effectiveness of our approach. We showed that our proposed technique achieves a very close approximation of performing KNN queries in the original space by generating a result set whose elements on average have less than 0.08 mile displacement to the elements of the actual result set in a 26 mile by 26 mile area containing more than 10000 restaurants. We also showed that a malicious attacker gains almost no useful knowledge about the parameters of our encoding techniques, even when significant knowledge about the key is compromised. Hence, a nominal displacement error in approximating only one of the key parameters, (a meter displacement in a 670 sq-mile area) will result in no useful information to compromise our encryption.

Finally, in [KS07] we showed that the KNNs computational complexity in our scheme is $O(K \times \frac{2^{2N}}{n})$ where N , the curve order, is a small constant. Moreover, since only the K closest points are sent back to the client, the communication complexity becomes $O(K)$. These are much lower than $O(n)$ complexity but we believe the computational complexity can still be improved further.

4 Conclusion and Future Directions

This paper consisted of two main parts. In the first part, we reported on the new extensions and developments of a system that we built in the past two years to enable geospatial decision making, dubbed *GeoDec*. In particular, we focused on several new spatial querying capabilities of *GeoDec* and discussed the importance of these spatial queries in decision making applications. In the second part, we argued that for many of the spatial queries supported by systems such as *GeoDec*, it is critical to preserve the privacy of the locations of both the query point and the result set. Subsequently,

we introduced novel privacy metrics to be met in order for a system to preserve location privacy. We then discussed our space-encoding approach to location privacy and showed that our approach meets the defined metrics. As part of our future plan, we intend to extend our location privacy approach to support dynamic/moving objects as well as other types of spatial queries.

References

- [BKS01] Stephan Börzsönyi, Donald Kossmann, and Konrad Stocker. The Skyline Operator. In *Proceedings of ICDE'01*, pages 421–430, 2001.
- [BS03] Alastair R. Beresford and Frank Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.
- [BWJ05] Claudio Bettini, Xiaoyang Sean Wang, and Sushil Jajodia. Protecting privacy against location-based personal identification. In Willem Jonker and Milan Petkovic, editors, *Secure Data Management*, volume 3674 of *Lecture Notes in Computer Science*, pages 185–199. Springer, 2005.
- [CBP] Reynold Cheng, Elisa Bertino, and Sunil Prabhakar. Preserving user location privacy in mobile data management infrastructures. In *Privacy Enhancing Technology Workshop (PET 2006)*, Cambridge, UK, June 2006.
- [FR89] C. Faloutsos and S. Roseman. Fractals for secondary key retrieval. In *PODS '89: Proceedings of the eighth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 247–252, New York, NY, USA, 1989. ACM Press.
- [GG03] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys*. USENIX, 2003.
- [GL] Bugra Gedik and Ling Liu. A customizable k-anonymity model for protecting location privacy.
- [GL04] Marco Gruteser and Xuan Liu. Protecting privacy in continuous location-tracking applications. *IEEE Security & Privacy*, 2(2):28–34, 2004.
- [Goo] Google earth. <http://earth.google.com>.
- [Hil91] David Hilbert. Über die stetige abbildung einer linie auf ein flächenstück. In *Math. Ann.* 38, pages 459–460, 1891.
- [HS99] Gísli R. Hjaltason and Hanan Samet. Distance Browsing in Spatial Databases. *TODS, ACM Transactions on Database Systems*, 24(2):265–318, 1999.
- [IW06] Piotr Indyk and David P. Woodruff. Polylogarithmic private approximations and efficient matching. In *Theory of Cryptography, Third Theory of Cryptography Conference*, pages 245–264, New York, NY, USA, 2006.
- [Jag90] H. V. Jagadish. Linear clustering of objects with multiple attributes. In *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, pages 332–342, Atlantic City, NJ, 1990. ACM Press.
- [KGMPO6] Panos Kalnis, Gabriel Ghinita, Kyriakos Mouratidis, and Dimitris Papadias. Preserving anonymity in location based services. *A Technical Report TRB6/06, National University of Singapore*, 2006.
- [KKSS06] Ali Khoshgozaran, Ali Khodaei, Mehdi Sharifzadeh, and Cyrus Shahabi. A multi-resolution compression scheme for efficient window queries over road network databases. In *Proc. 1st Workshop on Spatial and Spatio-temporal Data Mining (SSTDM in conjunction with ICDM'06)*, 2006.
- [KM00] Flip Korn and S. Muthukrishnan. Influence sets based on reverse nearest neighbor queries. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 201–212. ACM Press, 2000.

- [KS07] Ali Khoshgozaran and Cyrus Shahabi. Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy. In *Proceedings of the 10th International Symposium on Spatial and Temporal Databases (SSTD'07)*, July 2007.
- [LK01] Jonathan K. Lawder and Peter J. H. King. Querying multi-dimensional data indexed using the hilbert space-filling curve. *SIGMOD Record*, 30(1):19–24, 2001.
- [MCA06] Mohamed F. Mokbel, Chi-Yin Chow, and Walid G. Aref. The new casper: Query processing for location services without compromising privacy. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, pages 763–774, Seoul, Korea, 2006. ACM.
- [Mok06] Mohamed F. Mokbel. Towards privacy-aware location-based database servers. In Roger S. Barga and Xiaofang Zhou, editors, *ICDE Workshops*, page 93. IEEE Computer Society, 2006.
- [MvJFS01] Bongki Moon, H. v. Jagadish, Christos Faloutsos, and Joel H. Saltz. Analysis of the clustering properties of the hilbert space-filling curve. *IEEE Transactions on Knowledge and Data Engineering*, 13(1):124–141, 2001.
- [Nav] Navteq. <http://www.navteq.com>.
- [PTFS05] Dimitris Papadias, Yufei Tao, Greg Fu, and Bernhard Seeger. Progressive Skyline Computation in Database Systems. *ACM Trans. Database Syst.*, 30(1):41–82, 2005.
- [PTMH05] Dimitris Papadias, Yufei Tao, Kyriakos Mouratidis, and Chun Kit Hui. Aggregate Nearest Neighbor Queries in Spatial Databases. *ACM Trans. Database Syst.*, 30(2):529–576, 2005.
- [RKV95] Nick Roussopoulos, Stephen Kelley, and Frédéric Vincent. Nearest Neighbor Queries. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, May 22-25, 1995*, pages 71–79. ACM Press, 1995.
- [SAA00] Ioana Stanoi, Divyakant Agrawal, and Amr El Abbadi. Reverse nearest neighbor queries for dynamic databases. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 44–53, 2000.
- [Sag94] Hans Sagan. *Space-Filling Curves*. Springer-Verlag, 1994.
- [SCC⁺06] Cyrus Shahabi, Yao-Yi Chiang, Kelvin Chung, Kai-Chen Huang, Jaffar Khoshgozaran-Haghighi, Craig Knoblock, Sung Chun Lee, Ulrich Neumann, Ram Nevatia, Arjun Rihan, Snehal Thakkar, and Suya You. Geodec: Enabling geospatial decision making. In *IEEE International Conference on Multimedia & Expo(ICME)*, 2006.
- [Sch84] Manfred Robert Schroeder. *Number Theory in Science and Communication*. Springer-Verlag, 1984.
- [SS] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression.
- [SS06] Mehdi Sharifzadeh and Cyrus Shahabi. The Spatial Skyline Queries. In *Proceedings of the 32nd International Conference on Very Large Data Bases: VLDB'06*, September 2006.
- [Sti02] Douglas R. Stinson. *Cryptography, Theory and Practice*. CHAPMAN & HALL/CRC, 2002.
- [TAK04] Snehal Thakkar, Jos Luis Ambite, and Craig A. Knoblock. A data integration approach to automatically composing and optimizing web services. In *ICAPS Workshop on Planning and Scheduling for Web and Grid Services*, 2004.
- [TPL04] Yufei Tao, Dimitris Papadias, and Xiang Lian. Reverse kNN Search in Arbitrary Dimensionality. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases*, pages 744–755. Morgan Kaufmann, 2004.
- [Vir] Microsoft virtual earth. <http://maps.live.com/>.
- [Yah] Yahoo! maps. <http://maps.yahoo.com/>.