A Cloaking Algorithm based on Spatial Networks for Location Privacy

Po-Yi Li[†] Wen-Chih Peng[†] Tsung-Wei Wang[†] Wei-Shinn Ku[§]
Jianliang Xu [‡] J. A. Hamilton, Jr.[§]

[†]Dept. of Computer Science, National Chiao Tung University, Taiwan
Email: {leeboy, wcpeng}@cs.nctu.edu.tw
[§]Dept. of Computer Science and Software Engineering, Auburn University, Auburn, USA
Email:{weishinn, hamilton}@eng.auburn.edu
[‡]Dept. of Computer Science, Hong Kong Baptist University, Hong Kong
Email: xujl@comp.hkbp.edu.hk

Abstract

Most of research efforts have elaborated on k-anonymity for location privacy. The general architecture for implementing k-anonymity is that there is one trusted server (referred to as location anonymizer) responsible for cloaking at least k users' locations for protecting location privacy. A location anonymizer will generate cloaked regions in which there are at least k users for query processing. Prior works only explore grid shape cloaked regions. However, grid shape cloaked regions result in a considerable amount of query results, thereby increasing the overhead of filtering unwanted query results. In this paper, we propose a cloaking algorithm in which cloaked regions are generated according to the features of spatial networks. By exploring the features of spatial networks, the cloaked regions are very efficient for reducing query results and improving cache utilization of mobile devices. Furthermore, an index structure for spatial networks is built and in light of the proposed index structure, we develop a Spatial-Temporal Connective Cloaking algorithm(abbreviated as STCC). A simulator is implemented and extensive experiments are conducted. Experimental results show that our proposed algorithm outperforms prior cloaking algorithms in terms of the candidate query results and the cache utilization.

1 Introduction

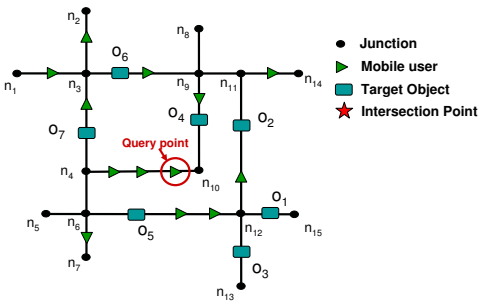
Location-based services (LBSs) have emerged as one of the killer applications for mobile computing and wireless data services. These LBSs are critical to public safety, transportation, emergency response, and disaster management, while providing great market values to companies and industries. Due to the unrestricted mobility of users in the mobile computing environments, users are often interested in acquiring information or services related to their current locations. Consequently, large amount of queries along with user location information are submitted to LBS servers. Examples of such queries include finding the nearest restaurants to a user (k nearest neighbor query) and finding ATMs within 500 meters from a user's current location (range query). While LBSs have shown to be valuable to users' daily life, on the other hand, they also expose extraordinary threats to user privacy. If not well protected, the location information of users may be misused by some untrustworthy service providers or stolen by hackers. Once the location information is exposed, adversaries may utilize them to invade user privacy. Obviously, it is important to protect location privacy.

Recently, the problem of location privacy preserving has growing interests and most research efforts have elaborated on k-anonymity [3, 6, 11]. In k-anonymity, users submit their queries to the LBSs via a trusted server (which is different from the LBS servers). This trusted anonymizer transforms the exact locations of a number of users into a *cloaked spatial area* in accordance with privacy requirements set by users in order to obtain data or services from LBSs. Upon receiving the LBS query with a cloaked region, the LBS server evaluates and returns a result superset (referred to as a candidate query result) containing the query results for all location points in the cloak region. From the candidate query result, mobile users are further to determine the actual query result according to the true location information at the mobile devices. In addition, candidate query results are cached for further data access. Note that when candidate results cached are able to satisfy consecutive queries, users are no longer need to issue queries,

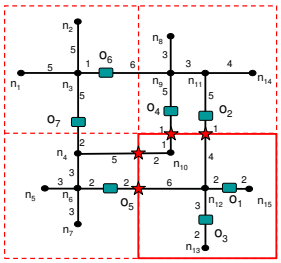
which reduces the location privacy threat of revealing user locations. However, cache in mobile devices has a limited storage size. More candidate query results incur cache replacements. As such, the cache hit probability will be reduced. Consequently, it is important to retrieve candidate query results that are likely to be accessed in the future and will not frequently incur cache replacements due to the limited cache size.

Prior works in [7] proposed a framework for location services in which a free space is divided into a number of grid cells. Then, a cloaked region consists of grid cells in which the total number of users is at least k. Hence, the cloaked spatial area whose shape is rectangle results in the larger candidate size. The problem we study could be better understood by an illustrative example in Figure 1(a), where spatial networks is modeled as a graph with each vertex as a junction and edges between two junctions are roads. Assume that k is set to 4 for k-anonymity and the KNN spatial query is issued. Figure 1(b) shows the cloaked region derived. It can be computed that the size of candidate query results is 5 (i.e., O_1, O_2, O_3, O_4 and O_5). Cloaked spatial areas derived do not take the features of spatial networks and the spatial-temporal moving behavior into consideration. The spatial-temporal moving behavior refers to the feature that the consecutive movements of users are not too far away. In this paper, we argue that the cloaked spatial area should be the cloaked segment set in which there are at least k users along with these road segments. In Figure 1(c), the cloaked region is viewed as a set of road segments (i.e., $(n_4, n_{10}), (n_9, n_{10})$ in this example). It can be verified that the size of candidate query results is 2 (i.e., O_4 and O_7). Hence, the candidate query size is smaller. Furthermore, since the moving behavior of a user has spatial-temporal feature which refers to the feature that the a user is likely to move along the nearby road segments, the cloaked segment set that exploits connective road segments is able to increase the cache hit ratio. This is due to that candidate query result sets are small, which is likely to be stored in limited cache of mobile devices and these candidate query results are very likely to be used in the near feature. Therefore, in this paper, by exploring features of spatial networks and the spatial-temporal moving behaviors for cloaking, cloaked segment sets are able to reduce the size of candidate query results and improve the cache hit ratios, thereby reducing the number of queries. Once the number of queries is reduced, the probability of revealing location information along with LBS queries is thus decreased.

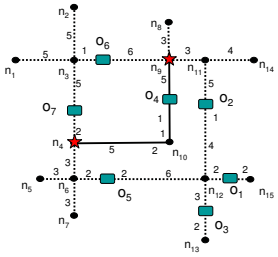
Consequently, in this paper, we propose a spatial network-based cloaking algorithm to derive cloaked segment sets. In traditional cloaking algorithms of k-anonymity, users are required to set their privacy profiles. A privacy profile is basically a pair of (k, A_{min}), where k is the number of users required for k-anonymity and A_{min}



(a) An example of a spatial network.



(b) A grid-based cloaked region



(c) A spatial network-based cloaked region

Figure 1. Examples of cloaked region

is the minimal area required for cloaked regions. Since cloaked regions derived in this paper is the set of road segments, A_{min} is only appropriate for cloaked region in free spaces. Thus, we define a new privacy profile in which the features of spatial networks (i.e., the number of road segments and the total length of road segments) is considered. As such, a user privacy profile is represented as (k, L_{min}) or (k, N_{min}), where k is the minimal number of users in a cloaked region and L_{min} (respectively, N_{min}) is the minimal total length (respectively, the minimal number) of road segments required. According to the user privacy profile, we propose a *Spatial-Temporal Connective Cloaking algorithm*(abbreviated as STCC) to derive cloaked region that contains a set of road segments in which there are at least k users and the number of road segments or the total length of road segments satisfies N_{min} or L_{min}. In order to quickly extract road segments for cloaked region, a hierarchical index structure is developed. Through the index structure built, algorithm STCC is able to efficiently derive cloaked segment set. Extensive experiments are conducted and experimental results show that the cloaked segment sets derived fully capture the spatial-temporal feature of moving behaviors, thereby not only protecting location privacy but also reducing the candidate query size.

We mention in passing that the authors in [13] explored the concept of k-anonymity for data privacy. By exploiting the k-anonymity, the authors in [3] proposed spatial-

temporal cloaking to protect user location privacy. Moreover, the authors in [2] proposed the *CliqueCloak* algorithm to support varied k-anonymous requirement for each user. In their work, the authors construct a clique graph in which some users can share the same cloaked region. However, these researches mainly focus on designing the location anonymizer rather than query processing. Therefore, the authors in [7] proposed a framework that include two main component, location anonymizer and query processor. In particular, location anonymizer will construct a pyramid structure to index different granularity cloaked region. On the other hand, query processor is used to obtain candidate query results according to the cloaked region. However, both of the cloaked region and query processing are in free spaces. Prior works do not take the feature of spatial networks into consideration nor utilize spatial-temporal feature of moving behavior for deriving cloaked region, let alone developing an index structure for spatial network-based location anonymizer. These features differentiate our work from others.

The rest of the paper is organized as follows: Preliminaries are given in Section 2. In Section 3, a spatial network-based algorithm is presented. Section 4 devotes to experimental results. This paper concludes with Section 5.

2 Preliminaries

Figure 2 depicts the system architecture, where there are two components in this system (i.e., a location anonymizer, and privacy-protected query processor). Mobile users are able to set up their location privacy profiles and register to with a location anonymizer. Once a user issues a location dependent query, the location of the user will be sent alone with the LBS query to the location anonymizer. Then, a location anonymizer will blur the true location as a cloaked region and forwards this LBS query with the cloaked region. The *Privacy-Protected Query Processor* is responsible for performing privacy-protected queries. Upon receiving the LBS query with a cloaked region, the privacy-protected query processor evaluates and returns a result superset (referred to as a candidate query result) containing the query results for all location points in the cloak region. From the candidate query result, mobile users are further to determine the actual query result according to the true location information at the mobile devices. As point out early, prior works explore transforms the exact locations of a number of users into a *cloaked spatial area* in accordance with privacy requirements set by users. The privacy requirement is represented as (k, A_{min}), where k is the minimal number of users needed in the cloaked spatial area and A_{min} is the minimal acceptable region size of the cloaked spatial area. Note that in prior works, the cloaked spatial area consists of grid cells and users are moved in the free space.

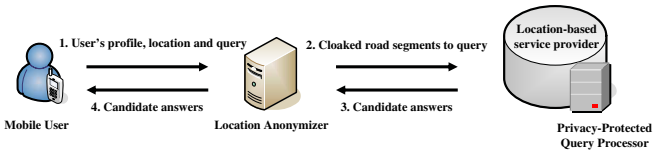


Figure 2. The system architecture

However, in reality, users are restricted to move on predefined roads, which is viewed as a spatial network. Without loss of generality, a spatial network is usually modeled as a graph, G=(V, E), where a vertex denotes a road junction, an edge denotes the road segment between two junctions and the weight of the edge is the length of this road segment. Thus, in our paper, we argue that the cloaking spatial area should be the set of road segments instead of grid cells. Furthermore, since the moving behaviors of users have spatial-temporal locality which refers to the feature that consecutive locations of mobile users are not far away, once the cloaked spatial area is the set of road segments nearby the true location, the cloaked spatial area is able to fully capture the spatial-temporal locality feature in the candidate query result. Thus, mobile users are likely to find out the query results from the cache of mobile devices. As such, the number of queries will be reduced, thereby avoiding location exposure.

Privacy profile: When cloaked spatial area is the set of road segments, a user privacy profile is thus defined as (k, L_{min}) (respectively, (k, N_{min})), where there are at least another $k-1$ users in cloaked spatial area and L_{min} (respectively, N_{min}) indicates the minimum acceptable total length of road segments (respectively, total number of road segments) in the cloaked spatial area. Both L_{min} and N_{min} capture the feature of spatial networks in user privacy profiles. Users are able to select one or both for their location privacy requirement. Note that L_{min} and N_{min} are particularly useful in dense area where even a large k cannot achieve the user's privacy requirements. With a larger L_{min}, it is harder to identify the exact location of a user in road segments. Moreover, a larger N_{min} is used to have more number of road segments in a cloaked spatial area in which the exact road segment that a user is on is harder to determine.

The objective of our work: In this paper, given a spatial network, denoted as G=(V, E), and a user privacy profile (i.e., (k, L_{min}) or (k, N_{min})), we intend to derive a cloaked segment set that satisfies user privacy profiles and the cloaking algorithm should achieve two requirements: 1.) **Accuracy** the cloaked segment set, represented as R, should be a set of road segments in which there are at least k users and those road segments should be as close to the user privacy profile (i.e., (k, L_{min}) or (k, N_{min})) as possible. 2.)

Efficiency: the cloaked algorithm should efficiently derive cloaked segment set based on the user privacy profile due to that a spatial network is usually large-scale and mobile users are dynamically move in a spatial network.

3 The Spatial-Temporal Connective Location Anonymizer

In order to efficiently derive the cloaked segment set, we first develop an index structure for a spatial network. In light of the index structure, we propose a spatial-network based cloaking algorithm.

3.1 Index Structure of a Spatial Network

Similar to the work in [12], given a spatial network, each vertex maintains an adjacency list in which each data node contains the adjacent vertex, the length of the corresponding road segment and the number of users along with this road segment. To efficient derive the cloaked segment set fulfilling the user privacy profile, we proposal a hierarchical structure that decomposes the spatial network into L_h levels and each level contains a various number of blocks consisting of a set of road segments. Clearly, the root of the hierarchical data structure has only one block that covers the whole set of road segments. Since we have two features of a spatial network in a user privacy profile, for each feature, we will build the corresponding index structure. In the following, an index data structure for the length of road segments is described. The index structure for the number of road segments is build in the similar way. To facilitate the presentation of this paper, the jth block in level i is denoted as $B_{i,j}$. Blocks contain a set of pointers to the lower level blocks and the total number of users within the set of road segments in lower levels blocks. For each block in the same level, the total length of road segments in each block should be as close as possible. In other words, the variance of each block in terms of the total length of roads segments is small and thus, it is able to appropriately obtain a set of road segments with their total length of road segments as close to L_{min} as possible.

An index structure is built in a bottom-up manner. The blocks in level 0 are the original road segments in a spatial network given. Then, two adjacent road segments will be put in one block at level 1. As pointed out early, each block at the same level should have the approximate total length of road segments. We will describe the criterion of putting two adjacent road segments into a block later. Once blocks at level 1 are generated, two blocks at level 1 will be formed into a block at level 2. Following the same operation, one could recursively merge two lower level blocks for higher level blocks until one block covers all road segments of a spatial network given.

Given a spatial network, the total length of road segments is denoted as $Total_{length}$ and the total number of road segments is expressed by $Total_{rs}$. The total length of road segments in block $B_{h,i}$ denotes as $Length_{B_{h,i}}$. We intend to let the total length of road segments in each block at the same level as close as possible. Thus, we should first derive the average total length for each block and intend to minimize the variance of each block in terms of total length of road segments. The expected total length for each block at level i is denoted as δ_i. In particular, for level 1, adjacent road segments will be merged to form a block. Hence, the maximal number of blocks in level 1 is determined as $\frac{Total_{rs}}{2}$ if two adjacent road segments are put into one block. Thus, δ_1 is formulated as $\frac{Total_{length}}{\frac{Total_{rs}}{2}}$. Once δ_1 is determined, adjacent road segments are put into one block if their total length is smaller than δ_1. Then, for each time, we consider add one road segments into a block. If one road segment is included and the total length of road segments in the block is larger than δ_1+ϵ, this road segment is removed. ϵ is a acceptable tolerance value when one more road segment is put in the block while the total length is larger than δ_1. For blocks at higher level (e.g., i), we will merge adjacent blocks at lower level (e.g., i-1 in this example). Denote the number of blocks at level $i-1$ is $N\_block_{L_{i-1}}$. We could have the expected total length δ_i as $\frac{total_{length}}{N\_block_{L_{i-1}}}$. Then, adjacent blocks will be merged together if the difference between their total length of road segments and δ_i is within ϵ. Same as in generating blocks in level 1, each time, one adjacent block at lower level is included into a higher level block if the total length of the higher level block is smaller than δ_i. Similar to the principle for deriving blocks at level 1, once the total length in higher level block is larger than δ_1+ϵ after including the newly added road segment. This newly added road segment will be removed. In order to build the index structures, we adopt a bottom-up approach to first derive lower level blocks and iteratively generate higher level block through merging adjacent blocks until the whole set of road segments is covered by one block. An example of a spatial network is shown in Figure **??** and assume that ϵ is set to 2. In the beginning, we should calculate δ_1. Since the total length of road segments in **??** is 86 and the number of road segments is 14, we could have δ_1=$\frac{86}{7} = 12.3$. Consider a road junction n_{10} as an example, where two adjacent road segments are able to put into one block since the difference between their total length and δ_1 is smaller than ϵ (i.e., 14-12.3=1.7 $\leq$ 2). Note that no more adjacent road segments is included since their total length of this block is already larger than δ_1+ϵ. For a road junction n_9, adjacent road segments (n_9, n_3), (n_9, n_{11}) and (n_9, n_6) are in the same block since their total length is smaller than δ_1+ϵ (i.e., 7+3+3=13 $<$ 12.3+2=14.3). In our example in procedure until there exists only one block that covers the whole spa-

tial network. In **??**, $N\_block_{L_1} = 7$. Furthermore, we could have $\delta_2 = \frac{86}{\lceil \frac{7}{2} \rceil} = 21.5$. With δ_2, we could decide whether two adjacent blocks should be put in a higher level blocks or not. For example, consider the adjacent blocks that contain n_{10} and n_9. Since the total length of these two blocks at level 1 is larger than δ_2+2 (i.e., 23.5), these two adjacent blocks cannot form a block at level 2. Hence, only adjacent blocks with their total length is smaller than $\delta_2+\epsilon$ are merged in one higher level block.

Algorithm 1 $Build\_Index_{length}$ Algorithm

Input: A spatial network graph, $G(V, E)$
Output: An index structure, $Index_{length}$, for L_{min}

1: $Total_{length}$= the total length of road segments in $G(V, E)$
2: $Total_{rs}$= the total number of road segments in $G(V, E)$
3: NB_{i-1} is the number of blocks at level i-1
NB_0=$\frac{Total_{length}}{\frac{Total_{rs}}{2}}$
$i = 1$
4: **while** No one block covers E in G=(V, E) **do**
5: **if** $i = 1$ **then**
6: NB_{i-1}=$\frac{Total_{length}}{\frac{Total_{rs}}{2}}$
7: δ_i=$\frac{Total_{length}}{NB_{i-1}}$
8: **if** $i = 1$ **then**
9: **for** each vertex with its adjacent road segments **do**
10: $length$=0
11: $j = 1$;
12: **while** $length$ larger than $\delta_i+\epsilon$ **do**
13: Include one adjacent road segment into $B_{i,j}$
14: $length$+=the length of selected road segment
15: $j++$;
16: $NB_1 = j$
17: **else**
18: **for** each block not marked **do**
19: $length$=the total length of this block;
20: $j = 1$;
21: **while** $length$ larger than $\delta_i+\epsilon$ **do**
22: Include one adjacent block into $B_{i,j}$ and mark this adjacent block
23: $length$+=the length of road segments in the selected block
24: $j++$;
25: $NB_i = j$
26: $i++$

Following the above procedure, we could derive the index structure in Figure 3. Note that the index structure for the number of road segments is built in the same way except that a criteria is set to the number of road segments instead of the total length of road segments.

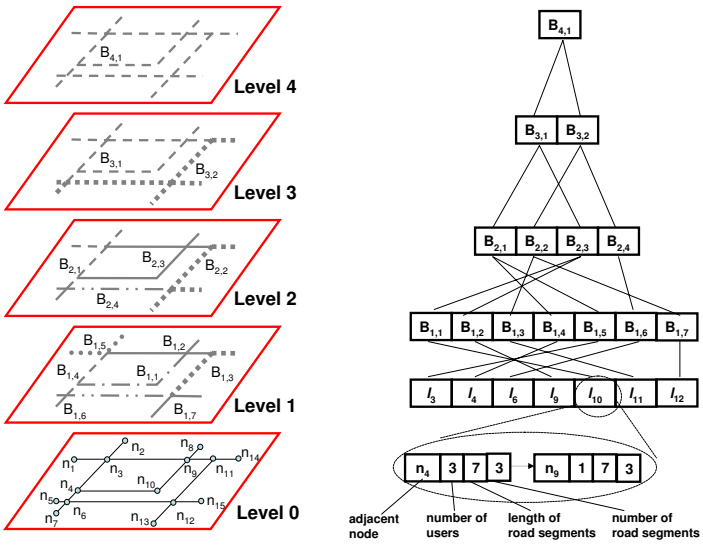


Figure 3. An example of a hierarchical structure

3.2 The Spatial-Temporal Connective Cloaking Algorithm

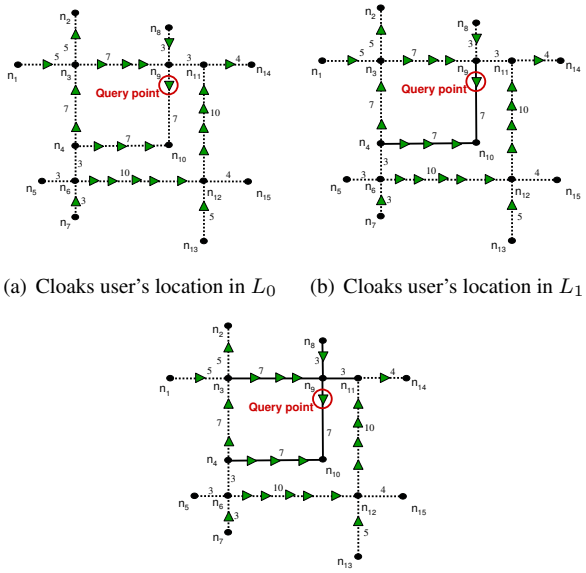
Assume that the user privacy profile is (k, L_{min}) and the location of user is represented as (x, y). When a user issues a query along with his location to the location anonymizer, the location anonymizer will first determine the road segment that the user is currently in. Then, an index structure $Index_{length}$ is used to located which block at level 1 containing this road segment. If the block found already satisfies the user privacy profile (i.e., the total number of users is larger than k and the total length of road segments is larger than L_{min}), the set of road segments in this block is used as the spatial cloaked area. However, if the user privacy profile is not satisfied, the neighboring blocks at the same level is first checked, where two blocks are identified as neighboring blocks if these two blocks have the same higher level block. If the combination of neighboring blocks have at least k users and the sum of their total lengths are larger than L_{min}, the union set of their road segments covered by these blocks is used as a spatial cloked area. On the other hand, if none of the neighboring blocks can be combined with the current block, we will recursively exploit the higher level block and perform the possible combination of higher level neighboring blocks until the user privacy requirement is fulfilled. An example is shown in Figure 4(a), where a user privacy profile is $k = 3$ and $L_{min} = 25$ and the query point is at the road segment (n_9, n_{10}). Then, the corresponding block at level 1 is obtained via index structure $Index_{length}$. Since the corre-

Algorithm 2 *Spatial-Temporal Connective Cloaking* Algorithm

Input: User's profile (k, L_{min}), user location (x, y) and an index structure $Index_{length}$

Output: A set of road segments for cloaking, denoted as CR

1: Find the road segment, (n_i, n_j), that contains (x, y)
2: Find block at level 1, denoted as $B_{x,y}$ includes (n_i, n_j)
3: CQ=a set of road segments in $B_{x,y}$
4: current block=$B_{x,y}$
5: **while** user privacy profile is not satisfied **do**
6: **if** these exist neighboring blocks of current block **then**
7: select one neighboring block of current block
8: include road segments in the selected block in CR
9: **else**
10: Find the parent node of current block in $Index_{length}$
11: select one neighboring block of the parent node of current block
12: include road segments in the selected block in CR
13: current block=parent node



(a) Cloaks user's location in L_0 (b) Cloaks user's location in L_1

(c) Cloaks user's location in L_2

Figure 4. Bottom-up cloak user's location

sponding block only has two road segments (i.e., (n_9, n_{10}) and (n_4, n_{10})) and the total length is smaller than 25 though there are already enough users, the total length in user privacy profile is not obeyed. Thus, we need to seeking neighboring blocks to fulfil the privacy requirement L_{min}. The neighboring block contains three road segments and thus the total length of these two blocks are larger than L_{min}. Consequently, a spatial cloaked area consists of road segments $\{(n_9, n_3), (n_9, n_8), (n_9, n_{11}), (n_{10}, n_4), (n_{10}, n_9)\}$.

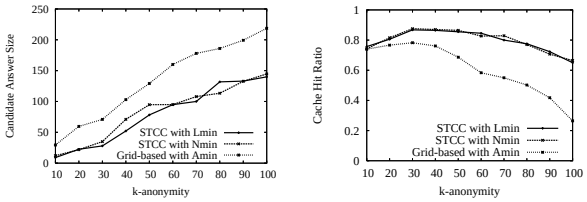
Note that if the user privacy profile is set to (k, N_{min}), we could use the same concept to generate a spatial cloked area. A cloaking algorithm shown above could be slightly modified by checking N_{min} instead of L_{min}. Also, the index structure $Index_{Num\_segment}$ is used as well.

4 Performance Study

In this section, we will evaluate the performance of our proposed algorithm. For the comparison purpose, we also implement traditional grid-size based cloaking algorithm (denoted as Grid-based scheme) in which a spatial network is divided into grids and by exploring the prior work, a pyramid data structure is implemented. When a user issues a query, we will first find out which grid that this user is in and then we extract all road segments within this grid. In all of our experiments, we use the *Network-based Generator of Moving Objects* [1] to generate moving objects. Oldenburg's road map is used in our experiments. The generator will output a set of moving objects that move on the road network of the given map. We set there are 5000 mobile users on the spatial network and they will update their location per time stamp. Next, target objects are randomly distributed on the spatial network. Moreover, we randomly choose one mobile user and the query type is continuous KNN query that persists 30 time stamps in the simulator. The default number of target objects is 3000. Performance metrics are *cache hit ratio* and *candidate answer size* which is the number of objects in the candidate query result. Note that in our simulation model, we set L_{min}=10000, and N_{min}=50. We analysis the road map data and find that when a cloked spatial area satisfied L_{min}, the average number of road segments is 50. Moreover, the average cloaked spatial area is 100*100. Thus, for fair comparison of our proposal algorithm and traditional A_{min}, we set $N_{min} = 50$ and $A_{min} = 100 * 100$. Our cloaked algorithm is abbreviated as STCC and the parameter ϵ used for building up index structures is set to 50. We randomly choose one mobile user as the query point who issues an K-Nearest-Neighbor query (KNN query) which persists 30 time stamps and there are 3000 target objects in this spatial network.

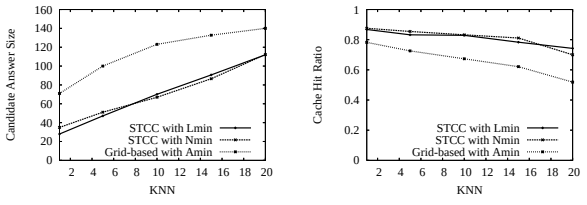
4.1 The Impact of k for k-Anonymity

First, we investigate the impact of k for k-anonymity. Without loss of generality, a user issues 1NN query and the moving speeding set to $\frac{1}{50}$ and the cache size of mobile devices size is 100. Figure 5(a) shows the candidate answer size with various k. As can be seen in Figure 5(a), with a



(a) Candidate answer size (b) Cache hit ratio

Figure 5. Performance comparisons with various k



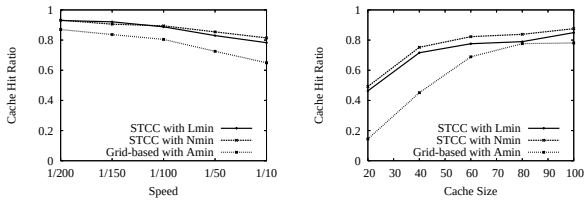
(a) Candidate answer size (b) Cache hit ratio

Figure 6. Performance comparisons with KNN queries

larger k, more users should be included in the cloaked spatial area to meet the privacy requirement. Thus, more road segments are in the cloked spatial area, increasing the candidate answer size. However, the candidate answer size of STCC is smaller than that of Grid-based scheme. This is due to that STCC considers the feature of spatial-temporal locality of moving behaviors for cloaking. By exploring the features of spatial temporal locality of moving behaviors and spatial networks, STCC is able to increase cache hit ratio. Figure 5(b) shows the cache hit ratios with various k. It can be seen that Figure 5(b) that STCC outperforms Grid-based scheme in terms of cache hit ratios. By increasing the k, the cache hit ratios of STCC and traditional cloaked region increase from $k = 10$ to $k = 30$. However, cache hit ratios decrease from $k = 40$ to $k = 100$ since candidate answer size is larger and mobile devices are not able to store a larger number of objects. Even though, the cache hit rate of STCC is significantly larger than that of Grid-based scheme.

4.2 The Impact of KNN Query

Now, we conduct experiments on varying the value of k for KNN queries. The user privacy profile for k-anonymity is set to 30. The default privacy requirements are L_{min}=10000, N_{min}=50 and $A_{min} = 100 * 100$ for fairness. In addition, the cache size is set to 100. The candidate answer size with various K for KNN queries is shown in



(a) Effect of speed (b) Effect of cache size

Figure 7. Performance study with various moving speeds and cache sizes

Figure 6(a). It can be seen that with a larger value of K for KNN queries, the candidate answer size will be increased. Note that the candidate answer size in STCC is still much smaller than that of Grid-based scheme. Since STCC uses a set of road segments for cloaking, those road segments that are near mobile users are form a cloaked spatial area. Furthermore, due to that the cloaked spatial area in STCC is the set of road segments that users are likely to move around, the cache hit ratios of STCC is much higher than that of Grid-scheme, which is shown in Figure 6(b).

4.2.1 The Impact of Moving Speeds and Cache Sizes

Now, the impact of moving speeds is evaluated. The moving parameter are ranged from $\frac{1}{200}$ to $\frac{1}{10}$, where a smaller value of moving parameter means a slower moving speeds. The user privacy profile for k-anonymity is set to 30. The default privacy requirements are L_{min}=10000, N_{min}=50 and $A_{min} = 100 * 100$. Figure 7(a) shows cache hit ratios with various moving speeds. Note that with a faster moving speeds, it is possible that the mobile user is very likely to move out the cloaked spatial area, thereby reducing cache hit ratios. It can be seen in Figure 7(a) that STCC still performs better than Grid-based scheme in terms of cache hit ratios. Clearly, the cache size of mobile devices will also have influence on cache hit ratios. Figure 7(b) is the experimental result by varying the cache size. By increasing the cache size, the cache hit rate increases because more candidate objects are able to store in the cache of mobile devices. In particular, for smaller cache size, which is the common case of mobile devices, the cache hit rate of STCC is better than Grid-based scheme, showing the advantage of exploring spatial-temporal feature of mobile behaviors in STCC.

5 Conclusion

In this paper, we proposed a cloaking algorithm in which cloaked regions are generated according to the features of spatial networks. By exploring the features of spatial net-

works, the cloaked regions are very efficient for reducing query results and improving cache utilization of mobile devices. Explicitly, mobile users can set their privacy profile (k, L_{min}) or (k, N_{min}). Given a user privacy profile, we propose an index structure to efficiently derive cloaked segment set. Note that two hierarchical index structures are able to obtain cloaked segment sets that are very close to the user privacy requirements. Based on index structures, we propose algorithm STCC to quickly blur the true user location as an acceptable cloaked segment set. We experimentally evaluated our proposed algorithm and experimental results shows that the cloaked segment sets derived fully capture the spatial-temporal feature of moving behaviors, thereby not only protecting location privacy but also reducing the candidate query size.

Acknowledgement

W. C. Peng was supported in part by Taiwan MoE ATU Program, and by the National Science Council, Project No. NSC 95-2211-E-009-61-MY3 Taiwan, Republic of China. Jianliang Xu's work was supported in part by the Research Grants Council of Hong Kong under Grant No. HKBU211206. This research has been funded in part by NSF grant DUE 0621307.

References

[1] T. Brinkhoff. A Framework for Generating Network-Based Moving Objects. *GeoInformatica*, 6(2):153–180, 2002.

[2] B. Gedik and L. Liu. "Location Privacy in Mobile Systems: A Personalized Anonymization Model". In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS), Columbus, OH, USA*, 2005.

[3] M. Gruteser and D. Grunwald. "Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking". In *Proceedings of the First ACM/USENIX International Conference on Mobile Systems, Applications, and Services (MobiSys), San Francisco, CA, USA*, 2003.

[4] H. Hu, J. Xu, and D. L. Lee. "A Generic Framework for Monitoring Continuous Spatial Queries over Moving Objects". In *Proceedings of the 2005 ACM International Conference on Management of Data (SIGMOD), Baltimore, Maryland, USA*, 2005.

[5] C. S. Jensen, D. Lin, B. C. Ooi, and R. Zhang. "Effective Density Queries on Continuously Moving Objects". In *Proceedings of the 22nd IEEE International Conference on Data Engineering (ICDE), Atlanta, GA, USA*, 2006.

[6] M. F. Mokbel. "Towards Privacy-Aware Location-Based Database Servers". In *Proceedings of the 22nd IEEE International Conference on Data Engineering (ICDE) Workshops, Atlanta, Georgia, USA*, 2006.

[7] M. F. Mokbel, C.-Y. Chow, and W. 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 (VLDB), Seoul, Korea*, 2006.

[8] M. F. Mokbel, X. Xiong, and W. G. Aref. "SINA: Scalable Incremental Processing of Continuous Queries in Spatio-temporal Databases". In *Proceedings of the 2004 ACM International Conference on Management of Data (SIGMOD), Paris, France*, 2004.

[9] K. Mouratidis, M. Hadjieleftheriou, and D. Papadias. "Conceptual Partitioning: An Efficient Method for Continuous Nearest Neighbor Monitoring". In *Proceedings of the 2005 ACM International Conference on Management of Data (SIGMOD), Baltimore, Maryland, USA*, 2005.

[10] K. Mouratidis, M. L. Yiu, D. Papadias, and N. Mamoulis. "Continuous Nearest Neighbor Monitoring in Road Networks". In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB), Seoul, Korea*, 2006.

[11] B. N. Schilit, J. I. Hong, and M. Gruteser. Wireless Location Privacy Protection. *IEEE Computer*, 36(12):135–137, 2003.

[12] S. Shekhar and D.-R. Liu. CCAM: A Connectivity-Clustered Access Method for Networks and Network Computations. *IEEE Transactions on Knowledge and Data Engineering*, 9(1):102–119, 1997.

[13] L. Sweene. k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[14] T. Xia and D. Zhang. "Continuous Reverse Nearest Neighbor Monitoring". In *Proceedings of the 22nd International Conference on Data Engineering (ICDE), Atlanta, GA, USA*, 2006.

[15] M. L. Yiu, N. Mamoulis, and D. Papadias. Aggregate Nearest Neighbor Queries in Road Networks. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):820–833, 2005.