K-Anonymity in Indoor Spaces through Hierarchical Graphs

Joon-Seok Kim
Department of Computer Engineering
Pusan National University,
Busan 609-735, South Korea
joonseok@pnu.edu

Yangsoo Han
Department of Computer Engineering
Pusan National University,
Busan 609-735, South Korea
aingjoa3@pnu.edu

Ki-Joune Li
Department of Computer Engineering
Pusan National University,
Busan 609-735, South Korea
lik@pnu.edu

ABSTRACT

Due to complex structure of indoor space, the demand on LBS (Location Based Services) in indoor space has been increasing as well as outdoor. Although LBS give convenience for users, they still have problems of exposing personal location and privacy. In order to protect privacy, many researches have been done, among which location K-anonymity is a method by cloaking locations through ASR (Anonymizing Spatial Region) involving *K-1* other users. However there is a limitation of this method to apply in indoor space that it assumes Euclidean Space and indoor space is characterized as non-Euclidean space in most cases unlike outdoor space. In this paper, we propose a new approach to location K-anonymity in indoor space. Our approach is based on the hierarchical structure of indoor space. First, we propose several algorithms to construct hierarchical structures for a given indoor space. Second, we introduce ASR generation algorithms to ensure the location K-anonymity with hierarchical structures. We analyze our methods through experimental analysis.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Application—*spatial databases and GIS*

General Terms

Algorithms, Security

Keywords

Hierarchical Graph, Indoor Space, Cloaking Locations, K-Anonymity

1. INTRODUCTION

Recently technique of positioning and location based services (LBS) with smart phones have been significantly improved. And due to complex structure of indoor spaces such as convention centers, department stores and transfer stations, the demand of LBS in indoor space is growing up as well as outdoor. Users with smart phones can have benefits of LBS in indoor space, for example to find out the shortest path from the entrance to the main conference hall in a building.

However, this may cause a serious problem that locations and identities of the users are exposed to attackers. In the case of server-client architectures of LBS, users should send their locations to the server for a request. For instance, given a query such as "show me the way to the store S in this shopping mall," the current location of the user and his/her identity can be exposed if an attacker has the list of payments of customers at the store. To this end, we need a method to protect the personal location and privacy in indoor space.

Many researches related with securing privacy in LBS have been done [2, 3, 4, 9, 12]. One of the topics focuses on location K-anonymity [5] with cloaking locations. According to this approach, clients send abstract locations, called *anonymizing spatial region* (ASR) that include *K-1* other users at least, instead of specific locations. Unfortunately it is limited to Euclidean space while indoor space is not Euclidean in most cases. As metric properties in indoor space are different to outdoor, the previous researches have limitations to be applied to indoor space without any modification.

For this reason, we propose a new approach for location K-anonymity using hierarchical graphs [6, 13, 14] for indoor space. Our approach is based on the observation that the location in indoor space can be alternatively represented by a symbol as a node on a hierarchical graph, such as `Room-203` and `Lobby`, instead of a point of 3-d coordinate (x, y, z) [10]. In this paper, we introduce new techniques to construct a hierarchical graph and ASR for cloaking locations in indoor space.

The rest of this paper is organized as follows. In next section we survey related work on location K-anonymity and the hierarchical structure in indoor space, and then present our motivation. Section 3 describes the definition of location by hierarchical graph in indoor space. In section 4, we introduce the basic concept and an algorithm for location K-anonymization in indoor space with hierarchical graph and its cost model. In section 5, we suggest methods to generate the efficient hierarchical graphs. In section 6, we analyze the hierarchical graph with experiments. We conclude and discuss future work in section 7.

ACM SIGSPATIAL GIS '12, November 6-9, 2012. Redondo Beach, CA, USA

2. RELATED WORK

In this section, we survey previous work related with location K-anonymity and hierarchical structure in indoor space, and our motivation is presented.

2.1 Location K-Anonymity

The methods for location cloaking are classified into two categories [3]; data-dependent cloaking and space-dependent cloaking.

Center Cloak[9] is an elementary method belonging to the former category for K-anonymity based on distance. In this method, an ASR is determined by containing the *K-1* nearest neighbors of a requester. But it has a weakness in 'center-of-ASR' attack, which has a high probability of inference that the requester may be located in the center of the ASR. In order to overcome this weakness, Nearest Neighbor Cloak (*NN-Cloak*) [9] was proposed as an improvement of *Center Cloak*. In *NN-Cloak*, the ASR is computed by four steps; (1) find *K-1* nearest neighbors of the requester as candidates of the center; (2) arbitrarily choose one of the K users chosen to determine the center of the ASR; (3) retrieve *K-1* nearest neighbors of the center; (4) finally, compute the ASR involve the neighbors and the requester.

Clique Cloak[2] is a region-based approach belonging to the second category to support a personalized K-anonymity model for providing location privacy. In *Clique Cloak*, a dynamic graph is constructed where each user becomes a node of the graph. Each node has a constraint box of which a center point is located on the node. An edge between two nodes is made only if their constraint boxes contain each other node. If a node has more than K neighbors connected by edges, it satisfies K-clique condition to compute the ASR.

Casper[12] and *Interval Cloak*[4], which also belong to the second category, are grid-based solutions to location K-anonymity. In *Casper* and *Interval Cloak*, the main idea is to employ a grid-based pyramid structure that hierarchically decomposes the entire space covered by the anonymizer. The anonymizer retrieves neighbors from the lowest level cell to the highest level cell until satisfying K-anonymity. The difference between *Casper* and *Interval Cloak* is whether or not the same level cells are considered to find neighbors before using the parent node as the ASR. While *Casper* takes neighbor cells into count but *Interval Cloak* does not. *Casper* guarantees smaller area of ASR than *Interval Cloak*. However as is proven in [9], both of *Casper* and *Interval Cloak* are secure only for uniform distributions.

Hilbert Cloak [9] overcomes this defect using Hilbert order. *Hilbert Cloak* generates one-dimensional mappings from two-dimensional locations of users. K-buckets with K users by the mapping are created as K-ASRs and each request is covered by the K-bucket where the requester is located.

2.2 Hierarchical Structure

In this subsection, we survey previous work related with the hierarchical structure in indoor space, which is one of the basic idea of our method [6, 13, 14]. It describes the connectivity graph of indoor space by multiple levels of abstraction.

In [14], the hierarchical graph in indoor space is formally defined and a recursive algorithm is introduced to generate the hierarchical graph from a base graph that consists of cells (e.g. rooms and corridors) and connectivity between cells. In order to construct a hierarchical graph, we firstly select colored nodes in the base graph which satisfy one of the following conditions; (1) the nodes with a connection to the exterior (e.g. entrance halls), (2) the nodes representing a vertical connection (e.g. stairs), and (3) articulation points. Second, connected components of each colored node are retrieved to create the subgraph. Third we determine decision points among colored nodes, such that they connect at least three other colored nodes and divide a sequence of colored nodes into different chains. Then merged graphs in a level are constructed. These steps are to be repeated until the node of graph becomes the root of the hierarchical graph. With the simplification of spaces, hierarchical graphs serve as an efficient analytic tool for various applications of indoor navigation [8, 14]. A detail explanation on hierarchical graph will be given in section 3.

2.3 Motiviation

As mentioned above, the previous solutions for location K-anonymity are based on Euclidean space. And they are not appropriate for indoor space since indoor space is considered as a non-Euclidean constraint space where the ASR is no longer valid.

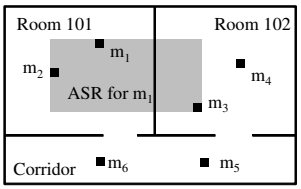


Figure 1: ASR in indoor space, K=3

As shown in an example of figure 1, suppose that $K = 3$ and m_1 is a requester to LBS. Moving objects m_2 and m_3 are selected as neighbors of m_1 for K-anonymity and the ASR is determined as the gray area in this figure. Although m_6 is nearer than m_3 from m_1, it is not included in the ASR. It implies the rectangular ASR cannot properly reflect indoor structure and respect the spatial proximity condition of ASR.

For this reason, we need novel methods for cloaking location, which provide a suitable ASR in indoor space. In this paper, we suggest a new approach to location K-anonymity using the hierarchical graph in indoor space. The goals of our work include

- hierarchy structure construction method for indoor space, and
- ASR definition based on hierarchical structure.

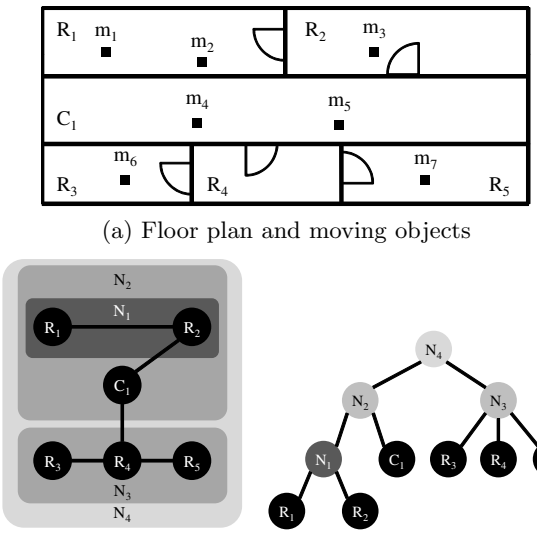
3. HIERARCHICAL STRUCTURE AND LOCATION K-ANONYMITY

Indoor space is a constraint space surrounded by structures like walls, ceilings and flats. This space can be also considered as symbolic space or cellular space that consists of space units (e.g. room, corridor, stair) reflecting the structure of indoor space [1]. Locations in symbolic space are represented by identified symbolic name (e.g. Diamond Hall) instead of 3-d coordinates and the properties of symbolic space are often described by means of connectivity topology. Based on this observation, we represent an indoor space as a

graph $G = (N, E)$, where N is a set of non-overlapping cells and E is a set of connection between cells. Given a graph G, we denote the location of object x as $loc(G, x) = n$ where $n \in N$ is the cell containing x.

Our work is based on the interpretation of indoor space as a hierarchical structure proposed in [14]. At each level of the hierarchical structure, indoor space is represented by a graph and there are hierarchical relations between two nodes in graphs of different levels. This hierarchical structure is called *hierarchical graph* as defined in [14];

Definition 1. Hierarchical Graph
A level$_{l+1}$ hierarchical graph H with regard to a level$_l$ base graph $G = (N, E)$ is defined by a complete partitioning of G into $k \geq 1$ nonempty, connected sets of nodes $\{N_1, ..., N_k\}$. Each set of nodes $N_i \subseteq N$ induces a subgraph $sub_i(G) = (N_i, E_i \subseteq E)$ with $E_i = \{(n_1, n_2) \in E \mid n_1, n_2 \in N_i\}$. Each of these subgraphs, in turn, corresponds to a node in the hierarchical graph H. Edges in H correspond to edges in G between nodes n_i, n_j of two different subgraphs $sub_i(G), sub_j(G)$.



(a) Floor plan and moving objects

(b) Hierarchical graph (c) Hierarchy of nodes

Figure 2: Example of hierarchical graph

Figure 2 illustrates an example of hierarchical graph, where indoor space is represented as graphs of different levels, and a node at i-th level is decomposed into several nodes at $(i + 1)$-th level. The hierarchical graph is used as a basic framework to determine ASR in our work.

4. K-ASR BY HIERARCHICAL GRAPH

In this section, we explain how to compute K-ASR, which contains K objects in the ASR from hierarchical graph. In figure 2, we assume that an indoor space S consists of cells $S = \{R_1, R_2, R_3, R_4, R_5, C_1\}$ and objects $\{m_1, m_2, m_3, m_4, m_5, m_6, m_7\}$ are in S as shown in figure 2(a). Figure 2(b) illustrates a hierarchical graph of the building and figure 2(c) represents a simple hierarchy as a tree structure to help understand. Given that m_1 is a requester with $K = 4$, we first check whether K-anonymity is satisfied in R_1 which contains m_1. The moving objects in R_1 is $\{m_1, m_2\}$ and R_1 is not 4-ASR since the number of objects in R_1 is less than $K = 4$. Then we check the parent node of R_1 in the hierarchical graph, which is N_1 in figure 2(b). The number of objects in N_1 is three and the node N_1 cannot be 4-ASR. In a similar way, we check N_2, which is the parent node of N_1. In this case, the number of objects in N_2 is five, N_2 becomes 4-ASR of m_1.

Let us discuss the cost of K-anonymization using the hierarchical graph. In order to simplify the cost model, we assume that (1) the hierarchical graph is loaded in main memory, (2) the cost for traversing in hierarchical graph can be ignored; (3) the hierarchical graph is balanced. Then the cost is determined by the number of objects within K-ASR, which is directly related with the number of leaf nodes in the hierarchical graph. For this reason, we consider the cost of K-anonymization as the expected number of leaf nodes.

Let the branching factor be b_f, the number of total cells be n, the number of total objects be m, and distribution of objects be uniform. In order to satisfy K-anonymity, it should be $K \leq \frac{m \cdot b_f^l}{n}$. The number of levels of the hierarchical graph is $l(\geq \left\lceil \log_{b_f} \frac{n \cdot K}{m} \right\rceil)$. Assuming that there is no duplication of leaf node cells, then the number of leaf nodes to search for K-ASR in hierarchical graph H is given as the following equation;

$$KAnonymityCost(H, K) = b_f^{\left\lceil \log_{b_f} \frac{n \cdot K}{m} \right\rceil} \quad (1)$$

For instance, given $b_f = 3$, $m = 1000$, $n = 300$ and $K = 25$, then the expected cost of K-anonymity is 9.

$$\begin{aligned} KAnonymityCost(H_{b_f=3}, 25) &= 3^{\lceil \log_3 \frac{300 \cdot 25}{1000} \rceil} \\ &= 9 \end{aligned}$$

The $KAnonymityCost$ can be plotted as a step function for different K. In the above example, the cost is same where $10 < K \leq 30$. The range is mainly determined by branching factor. The number of nodes is b_f^{i+1} where $b_f^i < \frac{n \cdot K}{m} \leq b_f^{i+1}$ (i: integer). For example, the $KAnonymityCost$ with $b_f = 2$ in interval $(1, 2]$, $(2, 4]$ and $(4, 8]$ are 2, 4 and 8 respectively. Figure 3 shows a relation between branching factor and the $KAnonymityCost$ under $b_f = 2, 3$ (x-axis: $K \cdot \frac{n}{m}$, y-axis: the $KAnonymityCost$ of K-ASR). In this figure, we assume that there is one object in each cell (i.e. $m = n = 1$)for the reason of simplicity. The graphs in $b_f = 2$ and $b_f = 3$ are plotted by black lines and gray lines respectively. Except some cases as hashed area in figure 3, $b_f = 2$ gives better performance than $b_f = 3$. From this graph, we draw an important conclusion that small branching factors yield better performance in most cases and it is recommended to build hierarchical graph with small branching factor.

5. BUILDING HIERARCHICAL GRAPH

In this section, we explain how to generate the hierarchical graph for K-ASR.

5.1 Generating Base Graph

In order to generate a hierarchical graph, the base graph of the lowest level is first required. In [14], the authors consider only connectivity to derive a base graph from floor plan. However this may cause some problems of a proper hierarchical graph if the building has wide corridors connected with many rooms. In such as cases, the height of

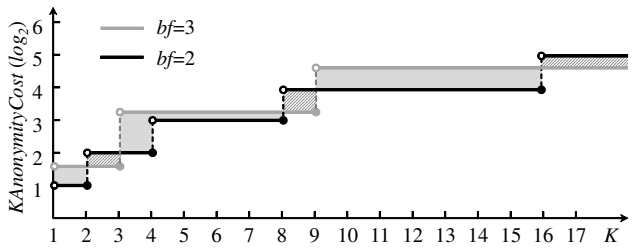
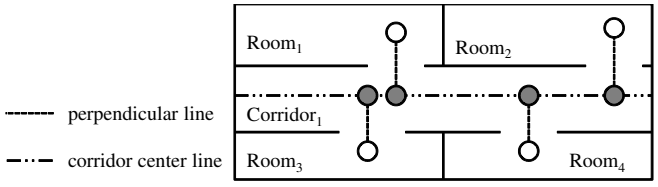


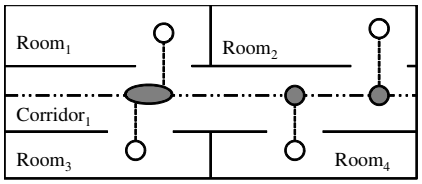
Figure 3: Relation between branching factor, $KAnonymityCost$ of ASR and K

hierarchical graph is small and the hierarchical graph may be unbalanced and areas of ASR may rapidly increase.

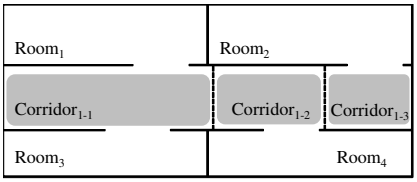
In order to ensure a proper height of hierarchical graph of indoor space, we suggest to divide a complex space into subspaces for generating a base graph as proposed in [11]. Figure 4 illustrate an example of generating base graph by subspacing. First of all, $Corridor_1$ is skeletonized to a center line (Fig. 4(a)). Then new nodes on the line are made, which are intersected with the center line and perpendicular lines from centers of doors. Finally, connected edges between rooms and the nodes are created.



(a) Base graph



(b) Merging adjacent nodes



(c) Subspacing

Figure 4: Example of generating base graph and subspacing

In this paper, we improve this method with an additional condition. If intersected nodes on the center line are too near each other, we merge the nodes into a bigger node to avoid narrow subspaces as shown in figure 4(a). In this figure, the ranges of two doors of $Room_1$ and $Room_3$ overlap and two nodes are therefore merged to the ellipse node like 4(b). Figure 4(c)) shows the final result of base graph generation. Figure 5 shows an example of base graph from a floor plan with by our base map generation method.

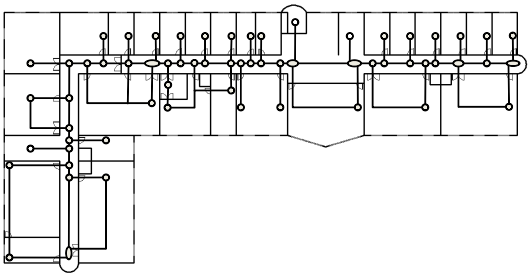


Figure 5: Example of base graph from floor plan

5.2 Algorithm of Generating Graph

Given a base graph, the number of combinations for generating hierarchical graph is exponential. Hence we propose a heuristic to create a hierarchical graph in indoor space in level-by-level way. Generation process of the hierarchical graph is composed of two steps as follows;

1. Select priority nodes to be merged in priority.
2. Select target nodes to be merged with priority nodes.

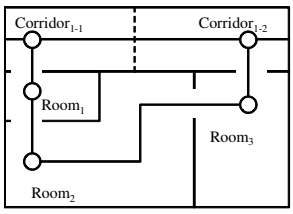
In order to merge nodes of base graph, we apply the control value[1] of space syntax [7] which is a theory for the analysis of spatial configuration. Control value is the measure of relative strength of axial line in pulling the potential from its neighbor cells. If control value of a cell A is greater than 1, the control of A is relatively strong. It means that there are many cells connected with A and probability of passing on A is high during moving from one cell to others. For example, open cells such as corridors or stairs have strong control. On the other hand, if the control of A is less than 1, the cell can be considered as an independent and closed cell. Upon this observation, we select nodes to be merged from base graph.

First we select the cells with weak control as priority nodes since they have weak influence on other nodes. If the control values of two cells are same, then the number of connected nodes and the area are to be considered respectively. Second each priority node is merged with the connected target node. In this paper, we propose three options to select a target node for a given priority node as follows: (1) *Control Value* (2) *Count* (3) *Area*.

Figure 6 shows an example of merging. Figure 6(a) is a base graph constructed from a floor plan and figure 6(b) is the control values of nodes. $Corridor_{1-1}$ and $Corridor_{1-2}$ are connected with other corridors respectively and their control values are greater than 1. In this figure 6(c), three gray nodes are selected as priority nodes. While control values are applied to merge nodes in figure 6, count and area of nodes are used in figure 7. Note that at each step of merging, the control values, counts, and areas are to be recalculated.

In *Count*, the result of merging is like figure 7. The $Corridor_{1-1}$ is the target node of $Room_1$ because $Corridor_{1-1}$ has smaller area than $Room_2$ although the count of $Corridor_{1-1}$ is equal to the count of $Room_2$. $Corridor_{1-2}$

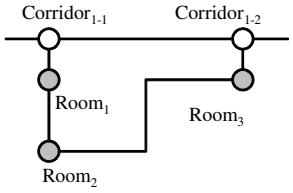
[1] $C(i) = \sum_{k \in A} \frac{1}{C_n(k)}$ (i, K: node, A: a set of nodes connected i node, $C_n(k)$: the number of K connected nodes connected K node)



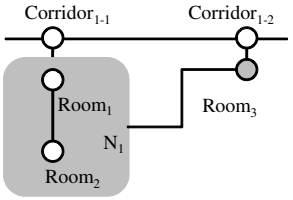
(a) Base graph

ID	Control Value
$Corridor_{1\text{-}1}$	over than 1
$Corridor_{1\text{-}2}$	over than 1
$Room_1$	1/2 + 1/3 = 5/6
$Room_2$	1/2 + 1/2 = 1
$Room_3$	1/2 + 1/3 = 5/6

(b) Control value in (a)



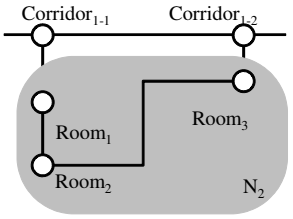
(c) Priority nodes



(d) Merging $Room_1$ and $Room_2$

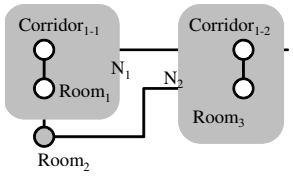
ID	Control Value
$Corridor_{1\text{-}1}$	over than 1
$Corridor_{1\text{-}2}$	over than 1
N_1	1/3 + 1/2 = 5/6
$Room_3$	1/2 + 1/3 = 5/6

(e) Control value in (d)

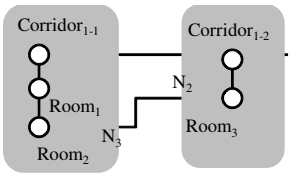


(f) Merging $Room_3$ and N_1

Figure 6: Merging with *ControlValue*



(a) Merging 1



(b) Merging 2

Figure 7: Merging with *Count*, *Area*

is also a target node of $Room_3$. Two priority nodes and two target nodes are merged into new nodes N_1 and N_2 respectively like figure 7(a). The count of N_1 and N_2 are same. In figure 7(b), $Room_2$ and N_1 are merged into N_3. In this example, the result of merging by *Area* is unexpectedly identical to that by *Count* because the counts of candidates of target nodes are equivalent.

Algorithm 1 explains the process of generating hierarchical graph in detail. This is a recursive function where the input parameter is the base graph G and the output is the hierarchical graph G'. Nodes of which control values are equal to or less than 1 are inserted into priory queue by descending order of control values. Other nodes are add into set H (line 3-9).

The target node of each priority node is dequeued from the queue and computed with preference such as *Control Value*, *Count* or *Area* (line 11-12). If the queue of priority nodes contains the target node, it is removed from the queue (line 13-15).

There are two ways of merging a priority node and a target node. If a target node has been merged in a level, the priority node merge with the target node in H. Otherwise a new node as the parent node is created and the priority node and the target node are merged into the parent node. Then this node is inserted into H (line 16-21). Above process is repeated until the priority queue is empty (line 10-22). Finally, if size of H is greater than 1, recall this function with H or return H (line 23-27).

Algorithm 1: GenerateHierarchicalGraph

input : $G = (N, E)$ Base Graph
output: $G' = (N', E')$ Hierarchical Graph of G

```
1  PriorityQueue<Node> priority ← ∅;
2  Graph H ← ∅;
3  foreach node ∈ N do
4  |  if node.ControlValue ≤ 1 then
5  |  |  enqueue node into priority;
6  |  else
7  |  |  add node into H;
8  |  end
9  end
10 while priority ≠ ∅ do
11 |  priorityNode ← an element dequeued from priority;
12 |  targetNode ← Choose(priorityNode);
13 |  if targetNode ∈ priority then
14 |  |  remove targetNode from priority
15 |  end
16 |  if targetNode has been merged in this level then
17 |  |  merge priorityNode with targetNode;
18 |  else
19 |  |  merge targetNode and priorityNode into
   |  |  parentNode;
20 |  |  add parentNode into H;
21 |  end
22 end
23 if H.size > 1 then
24 |  GenerateHierarchicalGraph(H)
25 else
26 |  return H;
27 end
```

5.3 Analysis on Hierarchical Graph

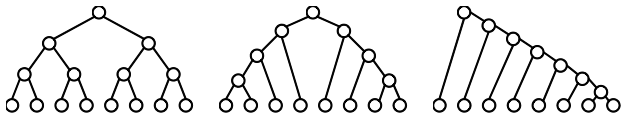
Following factors should be seriously considered in location K-anonymity in indoor space.

- Security against inference attacks
- Performance of query processing

First, it is important to ensure that the identity of the requester cannot be exposed with a probability that is larger than $1/K$, among *K-1* other users. The ASR must be created arbitrarily so that attackers cannot guess the identity even if the attackers know the manner.

Second, the area of the ASR should be minimized for the performance of query processing. Because accuracy of location and performance get lower if the area gets larger. The method of K-anonymity has no practical use if the method has bottleneck although the method is secure.

We employ hierarchical graphs for K-anonymization. Therefore these factors should be reflected in generating the graph. It is important to maintain the balance among nodes of the hierarchical graph during generating the graph. In this paper, we consider a hierarchical graph is balanced if the height from root to each leaf node is identical, and the *KAnonymityCost* and areas of cells are not skewed. Figure 8 illustrates balanced and unbalanced hierarchical graphs. There are hierarchical graphs with equal branching factors, and equal heights, but different average of lengths between leaf node and root (d in figure 8) , areas and counts of nodes.



(a) $d = 3$ (b) $d = 3.25$ (c) $d = 4.735$

Figure 8: different hierarchical graphs

If the graph is unbalanced, the area or the *KAnonymityCost* rapidly increases when expanding the ASR for K-anonymity. In this case, it is possible for attackers to guess identity of requester. For instance, in figure 8(c), suppose that there is an object per leaf node and $K = 2$. If the requester is in the most right leaf node, then the parent node which has two leaf nodes is selected as the ASR. So no one knows who the requester is. However if the requester is in the most left leaf node, then the root node which has eight leaf nodes is selected as the ASR. If the attacker knows $K = 2$ and the structure of the graph, then the identity of the requester is exposed. Consequently, unbalanced hierarchical graphs have high possibility of disclosing location.

Now let us discuss the performance. Table 1 shows result of comparison with the *KAnonymityCost* of three graphs in figure 8 for K-anonymity where $K = 2, ..., 4$ and one object is in each of leaf nodes. It shows that the balanced hierarchical graph (figure 8(a)) has smaller *KAnonymityCost* than unbalanced graphs (figure 8(c)) in most cases.

Table 1: Comparison with the *KAnonymityCost* of balanced and unbalanced graphs

Graph	$K = 2$	$K = 3$	$K = 4$
(a)	16	32	32
(b)	22	26	32
(c)	37	39	42

6. EXPERIMENTAL ANALYSIS

In this section, we show the results of experiments to analyze the proposed location K-anonymization method.

6.1 Experiment Setting

From experiments, we compared the proposed method with three options (*Control Value*, *Count*, *Area*) to choose target node and also observed the following aspects of the proposed methods;

- characteristics of hierarchical graphs
  - comparison of average heights
  - comparison of average branching factors
  - comparison of shapes with visualization
- performance of K-anonymity
  - comparison of areas of leaf nodes
  - comparison of counts of leaf nodes

For our experiments, we prepared a data set of a building in our campus (figure 5). The features of data set are summarized in table 2. We see that the building has cells with various areas as the variance of area of all spaces is about $95m^2$.

Table 2: Features of building dataset

# of floors	7
avg. area of spaces	$30.8m^2$
var. of area of spaces	$95.8m^2$
avg. connectivity	2.4
var. of connectivity	1

We employed two data sets of objects with different distributions. While the objects in data set A are distributed in proportion to area of each cell, equal number of objects are located in each cell in data set B.

Table 3: Features of data sets

	Dataset A	Dataset B
Distribution	Proportional to area	Uniform
Number of object	100 - 1000	100 - 1000

6.2 Characteristics of Hierarchical Graph

Figure 9 shows the comparison between three options of hierarchical graph construction in terms of the average length between leaf node and the root and branching factors. The average branching factors are slightly bigger than 2 for all cases and there is no significant difference between these options. It means that the hierarchical graph gives reasonable performance according to the observation in section 4. On the other hand, the average lengths between leaf nodes and root are around 7.5, the option *Control Value* gives the smallest average length. We discuss the details of the results in the subsections.

Figure 10 shows the result of hierarchical graph construction with the option *Control Value* for each level. In figure 10(a), level-0 graph is the base graph from floor plan. From figure 10(a) and (b), we find that independent nodes such as room within each floor, are firstly merged through level-0 and level-1 as we expected. From level-2, merging between different floors takes place via edges between different floors as shown in figure 10(c)-(h). Finally the graph is reduced to a node, which is the root node of the hierarchical graph. When we apply option *Count* or *Area*, slight different generations of hierarchical graphs are performed from level-2, as depicted in figure 11 and 12, respectively. Comparisons between different options are given in subsection 6.3.

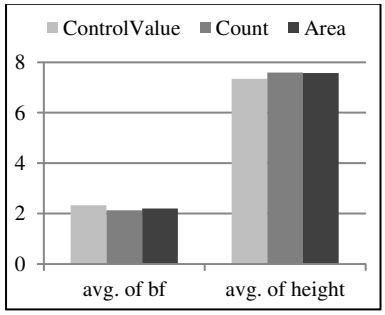
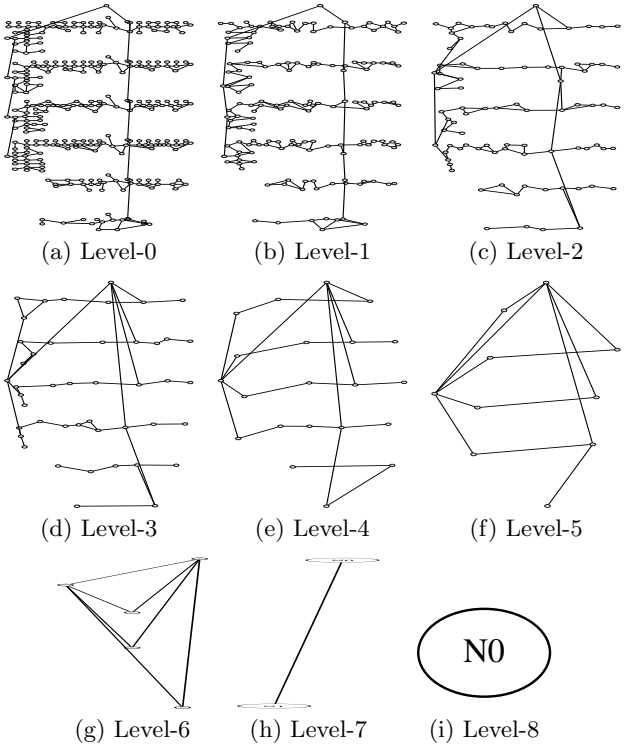


Figure 9: Comparison of average b_f and depth

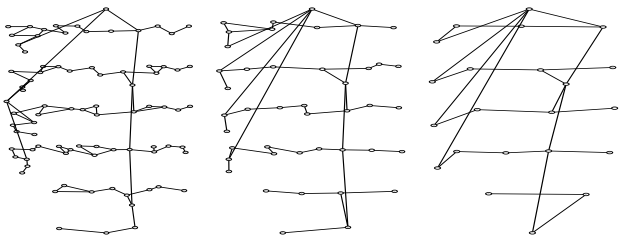


(a) Level-0 (b) Level-1 (c) Level-2

(d) Level-3 (e) Level-4 (f) Level-5

(g) Level-6 (h) Level-7 (i) Level-8

Figure 10: Hierarchical graph with *ControlValue*

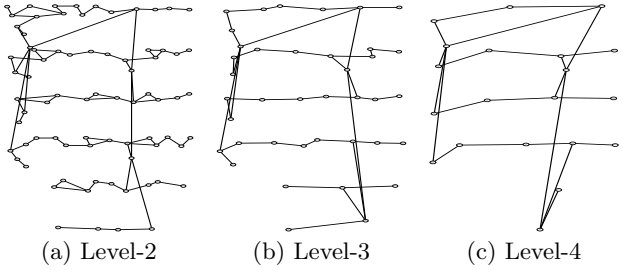


(a) Level-2 (b) Level-3 (c) Level-4

Figure 11: Hierarchical graph with *Count*

6.3 Evaluation of K-Anonymity

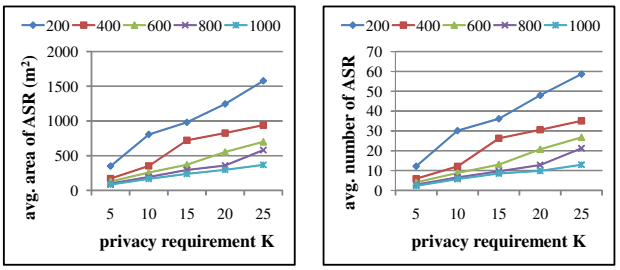
Figure 13 shows the average number of cells and areas of computed K-ASR by option *ControlValue* for varying K with data set B. The area and number of cells increase in almost constant rates as K increases. And as the density



(a) Level-2 (b) Level-3 (c) Level-4

Figure 12: Hierarchical graph with *Area*

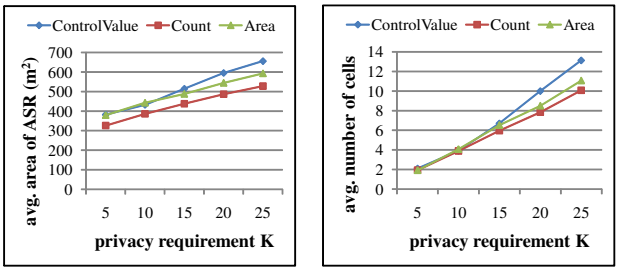
of objects increase, the area and number of cell of computed ASR become small. For example, when $b_f = 2.32$, n($leaf\ nodes$) = 302, $K = 25$, $m = 1000$, the estimated cost for location K-anonymity by equation 1 is $KAnonymityCost(H, 25) = 2.32^{\lceil \log_{2.32} \frac{302 \cdot 25}{1000} \rceil} \simeq 12$. We see that this estimated value is close to the value in figure 13(b).



(a) average area of spaces (b) average number of spaces

Figure 13: Evaluation of K-anonymity with *ControlValue* for varying K and # of moving objects

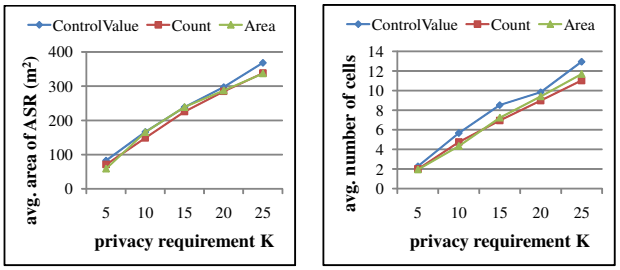
In figure 14 and 15, we compare three options *Control Value*, *Count*, and *Area* with data set A and B. Figure 14 shows that the option *Count* yields better performance than option *ControlValue* and *Area* with data set A, while no significant difference is found between three options in figure 15 for data set B. It implies that the distribution of objects influences on performance of the proposed method in a certain level.



(a) average area of spaces (b) average number of spaces

Figure 14: Comparison between three options with data set A

From figure 14 and 15, we conclude that the option *Count* is the best way to select target node in algorithm 1 of the proposed method. The option *Count* considers overall balance of the graph so that it can reduce branching factor.



(a) average area of spaces (b) average number of spaces

Figure 15: Comparison between three options with data set B

7. CONCLUSION

Previous methods for cloak location for privacy are not adequate since they assume Euclidean space while indoor space is often considered as a symbolic or cellular space where location is identified by symbolic code. In this paper, we proposed a new method to cloak location in indoor space using the hierarchical graph. The hierarchical graph has two advantages. First, it is an appropriate representation model for symbolic or cellular location in indoor space. A bounding rectangle only has been considered as ASR in previous work based on Euclidean distance. Second, the graph is an efficient data structure retrieving objects for K-anonymity.

In order to construct proper hierarchical graph for K-anonymity, we introduced the concept of control value from the space syntax theory. Since the computation of the optimal merging requires an exponential computation, a heuristic method is proposed in this paper. With other measures, area and count of nodes, it plays an important role in merging neighbor cell to produce K-ASR from hierarchical graph. We also analyzed the cost model of K-anonymity. From experiments, we observed the performance of the proposed method from several viewpoints. In particular, we compared three options of hierarchical graph construction and found that option *Count* gives the best performance.

Our future work includes the improvement of the proposed method. We expect that the proposed method could be improved by exploring semantics of indoor space, for example, difference between long corridor and large hall. And we will also consider the distribution of objects for subdivision of big or long cell and better merging method. If many users are located in a certain indoor space, then range of computed ASR may be too narrow. In this case, the space may contain semantically meaningful location which can be attacked with background knowledge on indoor semantic information. Other aspects of location privacy issues such as location l-diversity [15] in indoor space should be considered as well as outdoor in the future.

8. ACKNOWLEDGMENTS

This research was supported in part by Brain Korean 21 Project and a grant(11 High-tech Urban G11) from High-tech Urban Development Program funded by Ministry of Land, Transport and Maritime Affairs of Korean government.

9. REFERENCES

[1] C. Becker and F. Durr. On location models for ubiquitous computing. *Personal and Ubiquitous Computing*, 9(1):20–31, 2005.

[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)*, pages 620–629, 2005.

[3] A. Gkoulalas-Divanis, P. Kalnis, and V. S. Verykios. Providing k-anonymity in location based services. *SIGKDD Explorations*, 12(1):3–10, 2010.

[4] M. Gruteser and D.Grunwald. Anonymous usage od location-based services through spatial and temporal cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, pages 31–42, 2003.

[5] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services(MOBISYS)*, pages 31–42, 2003.

[6] B. Hagedorn, M. Trapp, T. Glander, and J. Dollner. Towards an indoor level-of-detail model for route visualization. In *Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware*, pages 692–697, 2009.

[7] B. Hiller. The social logic of space. 1984.

[8] H. Hu and D. Lee. Semantic location modeling for location navigation in mobile environment. In *Proceedings of the 5th IEEE International Conference on Mobile Data Management*, pages 52–61, 2004.

[9] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location-based identity inference in anonymous spatial queries. In *Proceedings of the IEEE Transactions on Knowledge and Data Engineering*, volume 19, pages 1719–1733, 2007.

[10] K. Li. A new notion of space. In *Proceedings of the 8th International Symposium on Web and Wireles Geographical Information Systems*, pages 1–3, 2008.

[11] M.Meigiers, S.Zlatanova, and N.Pfeifer. 3d geoinformation indoors: Structuring for evacuation. In *Proceedings of Next Generation 3D City Models, Bonn, Germany*, pages 11–16, 2005.

[12] 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)*, pages 763–774, 2006.

[13] K. Richter, S. Winter, and U. Ruetschi. Constructing hierarchical representations of indoor spaces. In *Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware*, pages 686–691, 2009.

[14] E. Stoel, K. Schoder, and H. J. Ohlbach. Applying hierarchical graphs to pedestrian indoor navigation. In *Proceedings of the 16th ACM SIGSpatial International Conference on Advances in Geographic Information Systems*, page 1, 2008.

[15] M. Xue, P. Kalnis, and H. Pung. Location diversity: Enhanced privacy protection in location based services. In *Proceedings of the 4th International Symposium on Location and Context Awareness*, pages 70–87, 2009.