# In Search of Indoor Dense Regions: An Approach Using Indoor Positioning Data

(Extended Abstract)

Huan Li† Hua Lu‡ Lidan Shou† Gang Chen† Ke Chen†
† Department of Computer Science, Zhejiang University, China
‡Department of Computer Science, Aalborg University, Denmark
{lihuancs, should, cg, chenk}@zju.edu.cn, luhua@cs.aau.dk

## I. Background and Motivation

People spend significant parts of lives in indoor spaces like office buildings and transportation facilities. When many people are inside an indoor space, it is useful and important to measure the densities and find the dense regions in scenarios like flow management and security control. Considering a large airport, it is helpful for the authority to effectively determine which regions are currently most crowded so that actions, e.g., to open more fast tracks, can be taken timely at the right place to help passengers in need.

Assuming GPS-like data, density query techniques in free space [2] or road network [5] are unsuitable for indoor density due to the unique challenges in indoor settings. First, indoor spaces feature distinct entities like rooms, walls, and doors, which altogether form the indoor topology that enables and constrains object movements. Second, indoor positioning systems offer lower sampling frequency and report discrete locations, thus leaving considerable uncertainty at a particular time. The issue becomes even more challenging to deal with in the context of complex indoor topology. Specialized in indoor settings, a naive approach is to install counting sensors at the doors of each *indoor partition*, a basic topological unit (e.g., a room) connected to other units via doors. However, this method requires extra hardware investment and rigorous sensor deployment that falls short in a dynamic indoor environment where indoor topology may change from time to time. Motivated by the limitations of existing approaches, we propose a low-cost, data-driven approach that harvests needed values from online indoor positioning data [4]. Our approach demands no specific hardware, it is not bothered by indoor topology variations, and it works well for user-defined indoor regions. The key contributions of our study are as follows.

- We design an indoor density definition amenable to indoor object location uncertainty, and formulate the problem of finding top-$k$ indoor dense regions.
- We analyze the indoor location uncertainty, derive bounds of indoor densities, and introduce distance decaying effect into indoor density computing.
- By making use of the uncertainty analysis outcomes, we design efficient algorithms to search for the current top-$k$ indoor dense regions.
- We conduct extensive experimental studies to evaluate our proposals on synthetic and real datasets.

## II. Problem Formulation

In our setting, only the *latest* object positioning information $(o, loc, t)$ is maintained in an *online indoor positioning table* (OIPT), meaning that the object $o$ was last seen at location $loc$ at time $t$. Given an indoor location $loc$ reported at a past time $t_l$, its *indoor uncertainty region* $UR_I(loc, t_c, t_l)$ describes the indoor portions where the object can reach at the current time $t_c$ under the maximum speed constraint $V_{max}$. Introducing the *distance decaying* effect [4] that reflects the locality of object movement, we define *object presence* that stipulates how an object should be counted for an indoor region $r$ at time $t_c$.

**Definition 1 (Distance Decaying Object Presence):** Given an indoor region $r$, an object $o$'s indoor uncertainty region $UR_I(loc)$ with the distance decaying function $\Gamma$, $o$'s presence in $r$ is $\phi_r^\Gamma(o) = \frac{\int_{l \in (UR_I(loc) \cap r)} \Gamma(dist_I(loc,l))\mathrm{dl}}{\int_{l \in UR_I(loc)} \Gamma(dist_I(loc,l))\mathrm{dl}}$.

Accordingly, we define *density* for an indoor region and formulate the top-$k$ indoor dense region search problem.

**Definition 2 (Density):** Given a set $O$ of indoor moving objects, an indoor region $r$'s density is $\tau_O(r) = \frac{\sum_{o \in O} \phi_r^\Gamma(o)}{Area(r)}$.

**Problem 1 (Top-$k$ Indoor Dense Region Search):** Given a set $O$ of indoor objects, whose latest positioning records are captured in the OIPT, and a query set $Q$ of indoor regions, the top-$k$ indoor dense region search returns $k$ densest indoor regions in a $k$-subset $Q_k \subseteq Q$ such that $\forall r \in Q_k, \forall r' \in Q \setminus Q_k, \tau_O(r) \geq \tau_O(r')$.

Our problem setting allows users to customize semantic-dependent query regions according to their practical needs.

## III. Summary of Our Approach

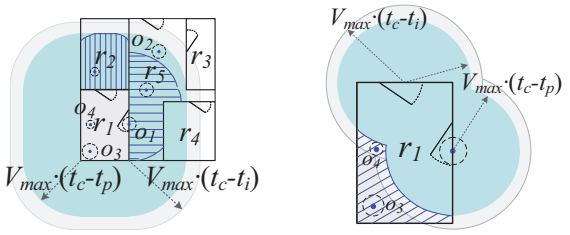
When the top-$k$ search is triggered at $t_c$, the object location in OIPT is already out-of-date due to the discrete nature of indoor positioning, making it complex to compute the precise object presences and indoor densities. To this end, we propose the indoor buffer and core regions to derive bounds of indoor densities, in order to make the density computation concentrate on the relevant objects only. Subsequently, we design the top-$k$ search algorithms based on the derived density bounds.

### A. Bounds of Indoor Density

Due to the object location uncertainty, a region $r$ where there was no object at a past time $t_p$ can contain objects at time $t_c$. However, such objects cannot come from locations too far away due to the speed constraint $V_{max}$. Instead, they can only

come from a buffer region that contains $r$, i.e., a *$\delta$-Minkowski region* [1] where $\delta = V_{max} \cdot (t_c - t_p)$. Considering the indoor topology, we further define $r$'s *indoor buffer region* $\Theta_I^{\triangleright}(r)$ as the intersection of $r$'s buffer region and the indoor parts from where one can reach $r$ within time interval $[t_p, t_c]$. Oppositely, we define $r$'s *indoor core region* $\Theta_I^{\triangleleft}(r)$ as a reduced region of $r$ from where one cannot leave $r$ within $[t_p, t_c]$.

The shaded parts in Fig. 1 depict the precise indoor buffer and core regions of a room $r_1$. In [4], we propose an iterative algorithm to determine an *arbitrary* shape region $r$'s $\Theta_I^{\triangleright}(r)$ and $\Theta_I^{\triangleleft}(r)$ by utilizing the shortest indoor path to $r$'s sides.




(a) Indoor Buffer Region (b) Indoor Core Region

Fig. 1. Precise Indoor Buffer Region and Indoor Core Region

Further, we use function *COUNT*$(r)$ to obtain the number of objects whose last reported location is contained by region $r$, Lemma 1 gives the upper and lower bounds that involve moving objects within $r$'s indoor buffer and core regions.

**Lemma 1:** $\frac{COUNT(\Theta_I^{\triangleleft}(r))}{Area(r)} \leq \tau_O(r) \leq \frac{COUNT(\Theta_I^{\triangleright}(r))}{Area(r)}$.

Lemma 2 loosens the above density bounds by using a longer temporal interval that ends with the current time $t_c$.

**Lemma 2:** For two past timestamps $t_i$ and $t_p$, if $t_i \leq t_p$ we have $\frac{COUNT(\Theta_I^{\triangleleft}(r,t_c,t_i))}{Area(r)} \leq \frac{COUNT(\Theta_I^{\triangleleft}(r,t_c,t_p))}{Area(r)} \leq \tau_O(r) \leq \frac{COUNT(\Theta_I^{\triangleright}(r,t_c,t_p))}{Area(r)} \leq \frac{COUNT(\Theta_I^{\triangleright}(r,t_c,t_i))}{Area(r)}$.

### *B. Algorithms for Top-k Indoor Dense Region Search*

Our overall framework uses a max-heap to control the processing order of query regions, using a pruning rule enabled by Lemmas 1 and 2. Specifically, it uses the oldest timestamp in OIPT to derive $\Theta_I^{\triangleright}(r)$ and $\Theta_I^{\triangleleft}(r)$ for each $r$ (c.f. Lemma 2), and overestimates (underestimates) $r$'s density by counting all objects whose last reported location is in $\Theta_I^{\triangleright}(r)$ ($\Theta_I^{\triangleleft}(r)$). After that, only the regions whose upper bound density is no less than the current $k$-th highest lower bound density should be further processed. Subsequently, the framework calls a search algorithm to find the top-$k$ results. Our full paper [4] details two versions of search algorithms. Both versions use a max-heap to give priority to the regions with higher overestimated density values. Compared to our one-pass search, our improved search makes use of a tighter upper bound $\tau_O(r) \leq \frac{OverCount(r)}{Area(r)} \leq \frac{COUNT(\Theta_I^{\triangleright}(r))}{Area(r)}$ where function *OverCount*$(r)$ obtains the number of objects whose uncertainty region just overlaps with the region $r$.

## IV. Experimental Results

We verify the efficiency and effectiveness of our approach, with a synthetic dataset generated by Vita [3] and a real dataset collected in a university building.

**Efficiency studies.** We compare five alternatives to our two search algorithms in a default setting, the results are reported in Table I. DC directly counts the objects contained by $r$ and thus has the lowest time and memory cost. However, its effectiveness is very poor, as to be shown in the below. Other nested-loop variants (NL*) sum up the object presences to $r$'s density based on Definition 2. Among them, NLRegion (NLObject)'s outer-loop is oriented to the regions (objects); NLwgbr and NLwibr use the general buffer region and indoor buffer region to reduce the search space, respectively. Our proposed methods significantly outperform all of them. The pruning ratio indicates that the bounds of indoor density in our methods are very effective in pruning objects. Also, TopkIDRsImprd has a higher pruning capability than TopkIDRs1Pass, as it can early prune some query regions whose final density is already lower than other regions' tighter overestimated bounds.

TABLE I
Efficiency Comparison on Synthetic Data (Default Setting)

| Algorithms | Running time (ms.) | Pruning ratio | Memory cost (MB.) |
|---|---|---|---|
| TopKIDRs1Pass | 399.7 | 81.56% | 147.8 |
| TopKIDRsImprvd | 365.7 | **85.02%** | 156.1 |
| DC | **68.5** | - | **2.2** |
| NLRegion | 148386.2 | 0 | 342.5 |
| NLObject | 2248.1 | 0 | 321.3 |
| NLwibr | 1082.2 | 60.74% | 68.6 |
| NLwgbr | 1597.7 | 34.85% | 51.2 |

**Effectiveness studies.** We compare our uncertainty model based method (UM) to DC in terms of Kendall coefficient $\tau$ and recall. The effect of changing $|Q|$ and the query time interval $\Delta t$ on the search effectiveness is shown in Fig. 2. Significantly, UM outperforms DC in all tests. UM's both measures decrease when more query regions are involved. Still, its $\tau$ is close to 0.7 when using all query regions. Besides, an object's uncertainty region becomes larger when $\Delta t$ increases, which reduces the accuracy of indoor densities computed. Nevertheless, UM's recall is still higher than 0.78, showing that our method is very effective in finding correct results even the OIPT contains relatively old location reports.

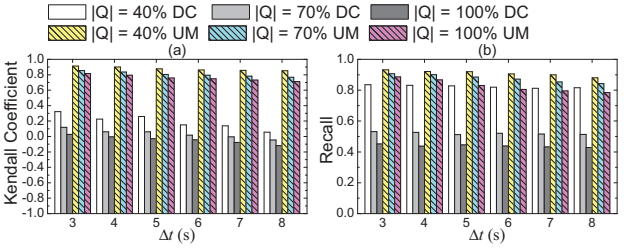



Fig. 2. Effectiveness vs. $\Delta t$ and $|Q|$ on University Data

## References


[1] C. Böhm. A cost model for query processing in high dimensional data spaces. *TODS*, 25(2):129–178, 2000.

[2] C. S. Jensen, D. Lin, B. C. Ooi, and R. Zhang. Effective density queries on continuously-moving objects. In *ICDE*, page 71, 2006.

[3] H. Li, et al. Vita: A versatile toolkit for generating indoor mobility data for real-world buildings. *PVLDB*, 9(13):1453–1456, 2016.

[4] H. Li, H. Lu, L. Shou, G. Chen, and K. Chen. In Search of Indoor Dense Regions: An Approach Using Indoor Positioning Data. *IEEE Trans. Knowl. Data Eng.*, 30(8): 1481–1495, 2018.

[5] M. L. Yiu and N. Mamoulis. Clustering objects on a spatial network. In *SIGMOD*, pp. 443–454, 2004.