# Geospatial Streams Publish with Differential Privacy

Yiwen Nie(✉), Liusheng Huang, Zongfeng Li, Shaowei Wang, Zhenhua Zhao, Wei Yang, and Xiaorong Lu

University of Science and Technology of China, Hefei, China
{nyw2016,lzf01,wangsw,hzq,ldayy}@mail.ustc.edu.cn,
{lshuang,qubit}@ustc.edu.cn

**Abstract.** Continuous releasing geospatial data is benefiting numerous areas, such as information push service, traffic scheduling and task assignment in crowdsourcing, etc. This kind of data is generated by people using positioning service in daily life, from which much sensitive information can be derived. Differential privacy is a strong theoretical and practical tool to provide protection; it has already been used on streams composing by datasets with fixed attributes. However, there is limited work on geospatial stream releasing with dynamic *scopes* for the requirement of accurate query. In this paper, aiming at achieving privacy protection of real-time geospatial synopsis with high utility, we introduce a method, called *Realtime Geospatial Publish* (*RGP*), which adopts differential privacy to geospatial stream with a new structure *k-memo*. We prove the privacy and utility of *RGP* theoretically and show the improvement of utility by experimental comparison with existing approaches on real datasets.

**Keywords:** Differential privacy · Geospatial partition · Streams · Location

## 1 Introduction

Personal data have been increasingly collected and analyzed. With the development of mobile device positioning technologies, such as GPS, WiFi or cellular network based positioning, there are plenty of ways to pinpoint individual's location. The collection of location data is useful for analysis, so as for providing customized services for mobile users. For instance, advertisers can benefit from these geospatial datasets, by understanding the market deeply and making more profits with less cost through delivering ads to target group at specific places; crowdsourcing server may schedule the tasks to executors with better acceptance, for most of the users choose tasks based on the distance from current location to target; police can be arranged in time in case of emergency relying on people flow.

Though geospatial data analysis facilitates enormous applications in daily life, the privacy breach issue should not be ignored when these data are directly


S. Wang and A. Zhou (Eds.): CollaborateCom 2016, LNICST 201, pp. 152–164, 2017.
DOI: 10.1007/978-3-319-59288-6_14

released. Location stream, which is an important component to geospatial datasets, is severely vulnerable to privacy breach due to the time correlation between neighboring timestamps. [15] shows that approximate 6 locations with timestamps would be able to uniquely identify a trajectory and further locate the individual, though the sensitive information, such as name, gender and address, has been deleted. The malicious third-party can further abstract users′ daily schedule, activity range and social relations by analyzing trajectories. Hence, releasing geospatial datasets privately has received more and more attention.

Some of the previous works focusing on static geospatial datasets indeed guarantee location privacy and utility for query, but only reflect empirical information from the past and cannot catch up with the trend in time. Some other proposed several definitions and privacy preserving methods for dynamics streams, but their topologies of locations are under fix structure, which, to some extent, ensure the total utility in one-way publish, but ignore the accuracy of query based on publishes in user-interaction model. In that case, how to overcome the limitations of two aspects above is the key to make geospatial synopsis practical without compromise of privacy. To this end, in this paper, we propose a method to release geospatial streams privately and accurately with adaptive region structure in interactive model.

Our contributions are as follow:

(i) We observe a newly common scenario of geospatial data releasing, which is realtime publish in interactive model, and show some limitations of existing works in this scenario.
(ii) We propose a new method, *Realtime Geospatial Publish* (*RGP*), to mitigate limitation of previous works in dynamic geospatial data publish by combining the adaptive region partition with the strategy of privacy budget allocation. What's more, a new data structure *k-memo* is used to optimize privacy strategy.
(iii) Theoretically, we prove the process of *RGP* over streams can satisfy differential privacy requirement. Then we further analyze the utility of algorithm and the effect *k-memo* has on the accuracy of result.
(iv) Through the experiments on real world datasets, we demonstrate utility improvements of *RGP* with comparison to existing methods. We also experimentally study the effects of the size of *k-memo* on the utility of synopsis.

## 2 Background

In this section we present some necessary background knowledge for our method. Differential privacy provides rigorous information-theoretical guarantee for data privacy, which conceals the small change of the original datasets in the output.

**Definition 1 (Neighboring datasets).** *Let $D$ and $D'$ be the two neighboring datasets, if $D'$ is obtained from $D$ by adding or deleting one tuple.*

Based on the neighboring datasets, *differential privacy* is defined as follow,

**Definition 2 ($\epsilon$-differential privacy [6]).** *Let $D$ and $D'$ be the two neighboring datasets. $\mathcal{A}$ is a randomized mechanism over these datasets and o be the possible output of $\mathcal{A}$. $\mathcal{A}$ is said to satisfy $\epsilon$-differential privacy, where $\epsilon \geqslant 0$, if $Pr[\mathcal{A}(D) = o] \leqslant e^{\epsilon} Pr[\mathcal{A}(D') = o]$.*

According to the definition, for two neighboring datasets, the multiplicative difference between the two probabilities of final outputs should not be more than $e^{\epsilon}$ or less than $e^{-\epsilon}$. Hence, parameter $\epsilon$ (namely, privacy budget) is important to control the leakage of privacy. Besides $\epsilon$, the design of randomized mechanism which achieves $\epsilon$-differential privacy also has much impact on the utility of data.

**Definition 3 (Global sensitivity).** *The global sensitivity of a query $q : D \rightarrow R^d$, denoted as $\Delta q$, is defined as the largest $L1$ norm of the difference between the answers of querying two neighboring datasets $D$, $D'$, written as $max_{D,D':\|D-D'\|=1} \|q(D) - q(D')\|_1$.*

Through global sensitivity, randomized mechanisms can be adaptive to various queries and provide appropriate noisy outputs to achieve *differential privacy.* There are mainly two mechanisms, *Laplace Mechanism* for numerics, and *Exponential Mechanism* for non-numerics.

**Theorem 1 (Laplace Mechanism [7,8]).** *A query $q : D \rightarrow R^d$ with the global sensitivity $\Delta q$. A randomized mechanism $\mathcal{A}$ outputs $o = q(D) + < Lap(\frac{\Delta q}{\epsilon}) >^d$, by adding noise derived from the Laplace distribution with scale $\lambda = \frac{\Delta q}{\epsilon}$ to $q(D)$, where $\epsilon$ is the privacy budget for $\mathcal{A}$.*

**Theorem 2 (Exponential Mechanism (EM) [13]).** *A utility function $u(D, o)$ measures the quality of an output $o$, given the dataset is $D$. A randomized mechanism $\mathcal{A}$ satisfying $\epsilon$-differential privacy, outputs $o$ with probability proportional to $exp(-\frac{\epsilon u(D,o)}{2\Delta u})$, where $\Delta u$ refers to the sensitivity of $u$.*

In practice, there are two rules for exploiting these mechanisms sequentially or respectively.

**Theorem 3 (Composition Theorem [14]).** *Let $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}_3, ..., \mathcal{M}_n$ be a series of mechanisms, where $\mathcal{M}_i$ provides $\epsilon_i$-differential privacy. The $\mathcal{M}$ is said to satisfy $(\sum_{i=1}^{n} \epsilon_i)$-differential privacy, if it executes $\mathcal{M}_1(D), \mathcal{M}_2(D), \mathcal{M}_3(D)...$, $\mathcal{M}_n(D)$ with randomness independently and outputs a vector of these mechanisms.*

**Theorem 4 (Parallel Theorem [14]).** *If $db_i$ is one of disjoint subsets of original dataset $D$, and $\mathcal{M}_i$ is a set of mechanism which provides $\epsilon_i$-differential privacy, applying on $db_i$. Then the overall $\mathcal{M}_i$ assures $max(\epsilon_i)$-differential privacy for $D$.*

# 3 Problem Model and Proposed Method

## 3.1 Model Description

**Publish Model.** The publish model considered here is a two-way interactive model. One of the interacting parties is users or any third-parties, malicious or not, who query on the sanitized geospatial synopses; the other is the published sanitized synopses provider. Some trusty location data collectors can be the provider, e.g. *Cellular Service Provider* (*CSP*). Every user of mobile phone has ratified an accord that allows *CSP* to access their locations. As a consequence, *CSP*s are capable to integrate the coordinates from various moving objects and publish the processed synopses. The query type discussed in this paper is the count of moving object in any area of any size.

**Data Model.** The type of dataset we consider is the coordinate data of moving objects collected periodically. The original collected dataset exploits user identification as data collecting unit, without consistent counting structure. By mapping user location into relevant subdomain, attributes of the transferred dataset $D_i$ are a set of non-overlapped subdomains over the whole region; every tuple records the location of an user at time $i$. The released geospatial synopses are the count of each subdomain.

However, the releases will bring potential privacy threaten, if they are not handled carefully. Through the continuously observation of synopses and relevant background knowledge, adversaries can guess a certain user's trace with high probability. This attack mode is *time correlation attack* (*TCA*).

In Fig. 1, we give a concrete example, in which user $u$ is the target. Suppose that points can only move among adjacent grids. The synopsis is collected from 11 p.m to 12 p.m, which indicates that most of people have already slept and few users hang around. Attacker *Alice* has known that $u$ is in the region $c_{31}$ at time $i$. From time $i$ to $i+1$, only the neighboring grids $c_{22}$ and $c_{31}$ have changed, and the trajectory of $u$ is $c_{31} \rightarrow c_{22}$. In time interval $[i-1, i]$, one user leaves from $c_{11}$ and $c_{21}$; one enters to $c_{12}$ and $c_{31}$ respectively. Based on the movement rules, user $u$ only can be in grid $c_{21}$ at $i-1$. Therefore, *Alice* infers that the trace of $u$ is $c_{21} \rightarrow c_{31} \rightarrow c_{22}$ with certainty.

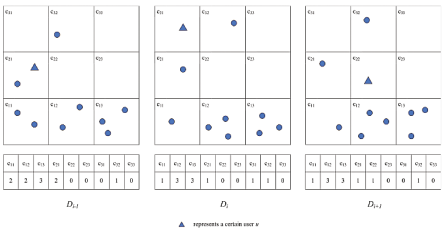


**Fig. 1.** *TCA*

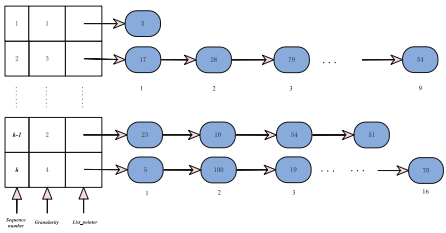


**Fig. 2.** *k-memo*

**Privacy Model.** Motivated by the privacy threaten mentioned above, we synthesize sliding window strategy [12] with adaptive domain partition [16] to defend against *TCA*s. Here are some relevant concepts which need to be redeclared on streams.

**Definition 4 (stream prefix).** *Let stream* F *be a series of sequential geospatial datasets* $F = (D_1, D_2, D_3, ...)$ *and* $F[i] = D_i$. *The stream prefix at time* $t$ *is* $F_t = (D_1, D_2, ..., D_t)$.

In analogy to the *neighboring datasets* for static datasets, the similar concept over streams is defined below.

**Definition 5 ($\omega$-neighboring [12]).** *Two prefixes* $F_t$ *and* $F'_t$ *are said to be* $\omega$*-neighboring, if one of these conditions is satisfied: (1) there is only one timestamp* $i \leqslant t$, $F_t[i] \neq F'_t[i]$ *or (2) there are two timestamps* $i$, $i'$, *with* $i < i'$ *and* $i' - i + 1 \leqslant \omega$, $F_t[i] \neq F'_t[i]$ *and* $F_t[i'] \neq F'_t[i']$.

The definition of differential privacy based on $\omega$-neighboring merges the privacy gap between *user-level* [9] (hiding any single user over finite streams) and *event-level* [3,4,12] (hiding any single event over infinite streams).

**Definition 6 ($(\omega, \epsilon)$-differential privacy [12]).** *A randomized mechanism* $\mathcal{A}$ *takes a stream prefix as input, and can be decomposed into sub-mechanisms* $\mathcal{A}_1, ..., \mathcal{A}_m$, *with each* $\mathcal{A}_i$ *providing independent* $\epsilon_i$*-differential privacy. The* $\mathcal{O}$ *is defined as the set of all possible outputs of* $\mathcal{A}$, *such that* $\mathcal{A}_i(F_t[i]) = o_i$ *and* $o_i \in \mathcal{O}$. $\mathcal{A}$ *is said to satisfy* $(\omega, \epsilon)$*-differential privacy, if for arbitrary* $t$ *and* $\omega$*-neighboring stream prefix* $F_t$, $F'_t$, *formula* $Pr[\mathcal{A}(F_t) \in \mathcal{O}] \leqslant e^{\epsilon} Pr[\mathcal{A}(F'_t) \in \mathcal{O}]$ *is held, with* $\forall t, \sum_{i=t-\omega+1}^{t} \epsilon_i \leqslant \epsilon$.

**Main Idea.** The granularity of $1_{st}$-level partition is calculated by $\sqrt{\frac{N\epsilon}{c}}$, where $N$ is initialized total count of moving objects. It is fixed during subsequent processing. Then the $1_{st}$-level grid is split into $2_{nd}$-level with the same formula, and keeps a *k-memo*, as shown in Fig. 2, to save previous results.

According to Theorem 4, the privacy budget is independent among grids of the same level. $\epsilon$ is total budget that every $1_{st}$-level grid has over stream, and is split into two parts. One is for $1_{st}$-level counts, denoted as $\epsilon'$, uniformly distributed inside the $\omega$-sliding window; the other $\epsilon^*$, is for the count of $2_{nd}$-level grids.

$\epsilon^*$ is also divided into two sections with ratio $\frac{1}{4}$. The first uniformly allocated in window is used to protect *dissimilarity estimation*. The second part $\frac{3\epsilon_*}{4}$ works for disturbing the $2_{nd}$-level count with the distribution strategy *BD* [12].

Figure 3 shows a snapshot of the process of *RGP*. The size of sliding window $\omega$ and *k-memo* is set to 5 and 3 respectively; the total budget for the $2_{nd}$-level count is $E = \frac{3\epsilon_*}{4}$. The table above shows the consumption of privacy budget over time. When $t = 5$, the *k-memo* of grid *ld*, shown in blue, has already recorded 2 different versions of synopses. The count of *ld* decreases obviously, resulting in a new partition. *RGP* uses half of remaining budget ($\frac{(E - \frac{E}{4})}{2} = \frac{3E}{8}$) to protect

*ld*'s count privacy, and inserts it to *k-memo*. At time $t = 7$, based on the result of *dissimilarity estimation*, *k-memo[2]* can be the substitute without consuming budget. Next, the distribution of points inside *ld* is totally different and cannot be replaced, but the *k-memo* is full. Therefore, *RGP* removes the oldest record and insert the new version.

### 3.2 Algorithm

**Initialization for Geospatial Data.** This algorithm mainly works for setting parameters and fixing the basic $1_{st}$-level grid structure. Line 1–3 is parameter initialization. $E_0$ will be explained in detail in the next algorithm. The calculation of the granularity of $1_{st}$-level $m_1$ (line 4) uses all the possible budget can be used to assure the accuracy of structure and lower down the uniform error, including budget $\epsilon^{\#}$ for $1_{st}$-level count, half of residue $\frac{3\epsilon^*}{8}$ for $2_{nd}$-level count and $\frac{\epsilon^*}{4\omega}$ for protecting dissimilarity.

**Algorithm 1.** Initialization

**Require:** $D_0$, $\epsilon$
**Ensure:** $Publish_i$.
1: $c_1 = 10$, $\epsilon' = \frac{\epsilon}{4}$
2: $\epsilon^* = 1 - \epsilon'$
3: $\epsilon^{\#} = \frac{\epsilon'}{\omega}$, $\epsilon_0 = \epsilon^{\#} + (\frac{3}{8} + \frac{1}{4\omega})\epsilon^*$
4: $m_1 = max(10, \frac{1}{4}\sqrt{\frac{|D_0|\epsilon_0}{c_1}})$
5: $E_0[1...m_1^2] = \frac{3\epsilon^*}{4}$
6: Partition the region into $m_1 \times m_1$, recorded in $Grid\_1$
7: $Publish_0 = RGP(D_0, Grid\_1, kmemo, m_1, E_0)$

**Real-time releasing.** Algorithm 2 shows the process of *RGP*. It publishes every $1_{st}$-level grid independently. $E_i[t]$ (line 2) records the available privacy budget for $2_{nd}$*-level count calculation* of $1_{st}$-level grid $t$ at time $i$, called *budget residue*. The renewed granularity *temp* (line 4) is grounded on the newest noisy count $N'$ calculated from actual point number in $t$ with fixed budget $\epsilon^{\#}$ (line 3). Then, if the number of POI in grid $t$ has non-neglectable change (line 5), leading to a new granularity which cannot be found in $k$ different publishes before, *RGP* exploits *UA* (line 6) to update the structure inside $t$ and recompute the $2_{nd}$-level. Otherwise, *k-memo* is valid (line 13–30). For every $1_{st}$-level grid, the average dissimilarity between actual $count_i^t$ and versions having the same $2_{nd}$-level division recorded in *k-memo* is calculated by *mean absolute error* (*MAE*) (line 16). $Kmemo_t[h].syn$ shows the noisy counts of $2_{nd}$-level, and $Kmemo_t[h].gra$ is the corresponding granularity.

**Algorithm 2.** Real-time Geospatial Partition ($RGP$)

**Require:** $D_i$, $region$, $Kmemo$, $m_1$, $E_i$
**Ensure:** $E_{i+1}$, $Kmemo$, $Publish_i$.

***/*Basicmodule*/***
1: **for** $t = 1$ to $m_1^2$ **do**
2: $\lambda' = \frac{1}{\frac{E_i[t]}{2} + \frac{\epsilon^*}{4\omega}}$
3: $N'$ = the count of grid $t$ with noise $Lap(\frac{1}{\epsilon^\#})$
***/*Submodule*/***
4: $temp = \sqrt{\frac{N' \frac{1}{\lambda'}}{\sqrt{2}}}$
***/*Futile k-memo*/***
5: **if** ($\nexists h \in [1, k]$, s.t. $temp = Kmemo_t[h].gra$) **then**
6: $Publish_i[t] = UA\ (N',\ \frac{1}{\lambda'},\ region[t], D_i)$
7: $E_{i+1}[t] = \frac{E_i[t]}{2} + Expire_{i-\omega+1}[t]$
8: **if** $Kmemo_t$ is full **then**
9: Delete the record having the oldest timestamps.
10: **end if**
11: Insert $Publish_i[t]$ into $Kmemo_t$
***/* Valid k-memo*/***
12: **else**
13: $\lambda = \frac{2}{E_i[t]}$
14: Calculate the real number of point $count_i^t$ in grid $t$ with granularity $temp$
15: **for** (all $h \in [1, k], s.t. temp = Kmemo_t[h].gra$) **do**
16: $var = \{\frac{1}{temp^2} \sum_{j=1}^{temp^2} |Kmemo_t[h].syn[j] - count_i^t[j]|\}$
17: Choose the $var$ by $EM(-var\frac{\epsilon^* temp^2}{32\omega})$, and memorize $h$
18: $dis = var + Lap(\frac{16\omega}{3\epsilon^* temp^2})$
19: **end for**
20: **if** $dis > (\lambda + \frac{16\omega}{3\epsilon^* temp^2})$ **then**
21: $Publish_i[t] = count_i^t + Lap(\lambda)$
22: $E_{i+1}[t] = \frac{E_i[t]}{2} + Expire_{i-\omega+1}[t]$
23: **if** $Kmemo[t]$ is full **then**
24: Delete the record with the oldest timestamps.
25: **end if**
26: Insert $Publish_i[t]$ into $Kmemo[t]$
27: **else**
28: $Publish_i[t] = Kmemo_t[h].syn$
29: $E_{i+1}[t] = E_i[t] + Expire_{i-\omega+1}[t]$
30: **end if**
31: **end if**
32: **end for**

$RGP$ uses $MAE$ as a score function for $EM$ to choose substitute for current synopsis (line 17) with a quarter of the budget; the rest is for Laplace Mechanism to protect the chosen $var$ (line 18). Relying on Definitions 1 and 3, it is easy to infer that the sensitivity of count query $count_i^t$ is $\Delta count_i^t = 1$. In that case, the sensitivity of $MAE$ is $\Delta var = \frac{1}{temp^2}$.

Next, if the variation of point distribution in grid $t$ is so obvious (line 20) that the accuracy of the estimation by a previous publish is worse than generating new version with noise, *RGP* releases a new one (line 21). In the two situations mentioned above, there is a new release needed to be saved in *k-memo* (line 11, 26). If the size of *k-memo* is larger than $k$, it discards the oldest version in *k-memo* (line 8–10, 23–25). The *budget residue* for next timestamp is the sum of current remain and the budget $Expire_{i-\omega+1}[t]$ used at $i-\omega+1$ which will be outside of $\omega$-windows (line 7, 22). When the estimation in *k-memo* is acceptable, *RGP* replaces the current one (line 28), so as to save the budget (line 29). As there is no new version generated, the *k-memo* remains.

**Uniform Partition.** Algorithm 3 generates the new partition for $1_{st}$-level grids. The value of parameter $c_2$ is optimized, which has been proved in [17].

**Algorithm 3.** The *UA* Partition

**Require:** $N$, $\epsilon$, region, $D_i$
**Ensure:** $Publish_i$.
1: $c_2 = \sqrt{2}$
2: $m_2 = \sqrt{\frac{N\epsilon}{c_2}}$
3: Partition the region into $m_2 \times m_2$
4: **for** $t = 1$ to $m_2^2$ **do**
5: $Grid\_2_i[t]$ record the region of grid $t$
6: $Num\_2_i[t]$ record the count of points in grid $t$
7: $Publish_i[t] = Num2\_i[t] + Lap(\frac{1}{\epsilon})$
8: **end for**

### 3.3 Utility Analysis

The error of *RGP* comes from three parts — *uniform error* due to the partition, *noise error* and *replacement error* which is from *k-memo*. *Uniform error* and *noise error* are treated as an integral to be analyzed. Assume that the region is partitioned into $b \times b$ grids, and the query covers $r$ portion of the whole area. Apparently, the uniform error is coming from the grids intersected with query edge, which is $\sqrt{r}b$; the noise error is proportional to the number of grids the query contains which is $\frac{\sqrt{2r}b}{\epsilon}$. Total error for a query is $\sqrt{r}b + \frac{\sqrt{2r}b}{\epsilon}$. When $b = \sqrt{\frac{n\epsilon}{\sqrt{2}}}$, the sum is minimized, where $\sqrt{2}$ can be replaced by other constants to accommodate different datasets. The more detailed proof can be seen in [16].

The distribution strategy is the source of *replacement error*. When $k = 1$, the *replacement error* per timestamp is $4\frac{2^m-1}{m\epsilon} + \frac{16\omega}{3\epsilon^* temp^2}$, if $m$ new publications occur in a window [12]. With the increase of the size of *k-memo*, $m$ decreases, as well as the *replacement error*, for there is more chance to find a appropriate substitute in *k-memo*. The lower bound is $\frac{4}{\epsilon} + \frac{16\omega}{3\epsilon^* temp^2}$. In Fig. 4, we show the limitation of *RGP*, by randomly selecting several timestamps to set the used budget to zero imitating replacement. $\lambda$ fluctuating with time, reflects the usage of budget at every timestamp; more importantly, it leads to the non-stable quality of chosen substitute.

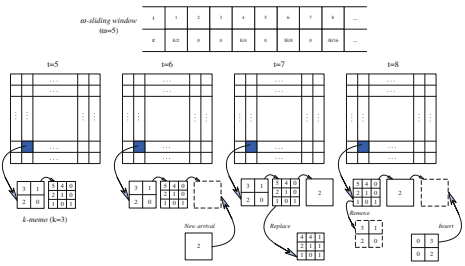



**Fig. 3.** A snapshot of *RGP*

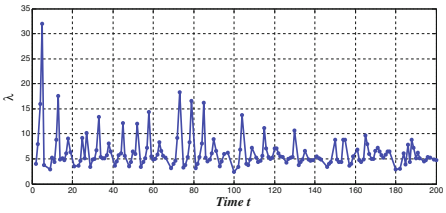



**Fig. 4.** The change of $\lambda$ ($\omega = 10$)

### 3.4 Privacy Analysis

The process of *RGP* is on two levels, $1_{st}$-level as *Basicmodule* and $2_{nd}$-level as *Submodule*. Depending on privacy budget distribution, *RGP* in *Submodule* can be split into *dissimilarity estimation* and $2_{nd}$*-level count calculation*; it can also be divided into *futile k-memo* and *valid k-memo*. We prove that *valid k-memo* satisfies $(\omega, \epsilon^*)$-differential privacy at first.

**Lemma 1.** $2_{nd}$*-level count calculation satisfies* $(\omega\text{-}\frac{3\epsilon^*}{4})$*-differential privacy.*

*Proof.* According to the Algorithm 1, $\frac{3\epsilon^*}{4}$ is total privacy budget over $\omega$-window on $2_{nd}$*-level count calculation*, for which half of the *privacy residue* $E_i[t]$ within window until current timestamp $i$ is allowed to be used.

Therefore, no matter what the timestamp is, the budget consumption in window is no more than $\frac{3\epsilon^*}{4}$, written as $\sum_{j=i-\omega+1}^{i} \epsilon_j \leqslant \frac{3\epsilon^*}{4}$. Based on Definition 6, we have the conclusion that $2_{nd}$*-level count calculation* satisfies $(\omega\text{-}\frac{3\epsilon^*}{4})$-differential privacy.

**Lemma 2.** *dissimilarity estimation satisfies* $(\omega\text{-}\frac{\epsilon^*}{4})$*-differential privacy.*

*Proof.* When current granularity of $1_{st}$-level grid $t$ can be found in *k-memo*, $-var\frac{\epsilon^* temp^2}{32\omega}$ is taken as an argument for *EM* to choose estimation result privately. Since the sensitivity of *MAE* is $\Delta var = \frac{1}{temp^2}$, according to Definition 3, substitute selection is $\epsilon_1^*$-differentially private at one timestamp, where $\epsilon_1^* = -\frac{2(\Delta var)\epsilon^* temp^2}{32\omega} = \frac{\epsilon^*}{16\omega}$. The noise abstract from Laplace distribution for dissimilarity protecting follows the scale of $\lambda_{dis} = \frac{16\omega}{3\epsilon^* temp^2}$, so the dissimilarity protection holds $\epsilon_2^*$-differential privacy, where $\epsilon_2^* = \frac{\Delta var}{\lambda_{dis}} = \frac{3\epsilon^*}{16\omega}$. In terms of Theorem 3 and Definition 6, *dissimilarity estimation* meets the requirement of $(\omega\text{-}\frac{\epsilon^*}{4})$-differential privacy.

**Theorem 5.** *vaild k-memo satisfies* $(\omega\text{-}\epsilon^*)$*-differential privacy.*

*Proof.* From Lemmas 1 and 2, we deduce that $(\omega\text{-}\epsilon^*)$-differential privacy is hold for *vaild k-memo*.

**Theorem 6.** *Submodule satisfies* $(\omega\text{-}\epsilon^*)$*-differential privacy.*

*Proof.* In *futile k-memo*, the granularity is new to *k-memo*, but the budget usage is similar to *valid k-memo* with replacement. Additionally, these two parts are non-overlapped over stream. In that case, *Submodule* integrating *futile k-memo* with *valid k-memo* satisfies ($\omega$-$\epsilon^*$)-differential privacy.

**Theorem 7.** *RGP satisfies* ($\omega$-$\epsilon$)*-differential privacy.*

*Proof.* For grid $t$, the newest noisy count $N'$ is necessary at every timestamp for which the budget $\epsilon^{\#} = \frac{\epsilon^{'}}{\omega}$ is needed, to make a judgement for internal structure. The process on $1_{st}$-level *Basicmodule* is ($\omega$-$\sum_{j=i-\omega+1}^{i} \epsilon^{\#}$)-differentially private. With Theorems 3 and 6, *RGP* satisfies ($\omega$-$\epsilon$)-differential privacy for whole geographical region, where $\epsilon = \epsilon^* + \epsilon^{'}$.

### 3.5 Efficiency Analysis

The efficiency of *RGP* is analyzed from two aspects, time complexity and space complexity.

**Time Complexity.** *RGP* only scan the data twice to form the final sanitized synopsis. The first scan is for determining the structure inside every $1_{st}$-level grid; the second is to compute the private count of the $2_{nd}$-level grid. In that case, the time consumption of *RGP* is proportional to the number of moving objects $N$, denoted as $O(N)$.

**Space Complexity.** The space occupation of $1_{st}$-level is related to the number of grids which is computed by $\frac{N\epsilon_1}{16c_1}$. $\epsilon_1$ represents the privacy budget used in $1_{st}$-level partition. Every $1_{st}$-level grid needs extra memory to store *k-memo*. Assuming the average size of the version recorded in *k-memo* is $s$. The average count of $1_{st}$-level grid is $\frac{16c_1}{\epsilon_1}$. According to the same dividing formula, we figure out $s = \frac{16c_1\epsilon_2}{c_2\epsilon_1}$. In that case, the total space consumption from the structure *k-memo* is $\frac{kN\epsilon_2}{c_2}$. The space complexity of *RGP* is $O(N)$.

## 4 Performance Evaluation

### 4.1 Experiment Settings

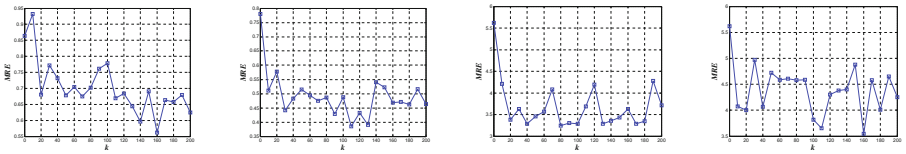
Two datasets we used here for experiments are T-drive and Rome. T-drive (*T-BJ*) records the *GPS* trajectories of almost 10 thousand taxies in Beijing [19] over a week within the range of ($39.6°N$, $116.1°E$) and ($40.1°N$, $116.612°E$). The trajectory dataset of Rome (*T-R*) [1] contains about 3 hundred taxies in the bounding box ($41.6°N$, $12.1°E$) and ($42.1°N$, $12.6°E$). We reorder these trajectories by time and set the sampling interval to 10 min. In order to make comprehensive contrast with fixed structure, the control groups used here are of three different division granularity $d_1 = 0.005°$, $d_1 = 0.01°$ and $d_3 = 0.05°$ with the same strategy of distribution *BD*. The query are of 4 different sizes, $q_1 = 0.005° \times 0.01°$, $q_2 = 0.01° \times 0.02°$, $q_3 = 0.02° \times 0.04°$ and $q_4 = 0.04° \times 0.08°$. For each size, we randomly choose query location and test 200 times. The experimental result is measured by Mean Relative Error (*MRE*) of different query sizes.

### 4.2 Experiment Result

**Effect of** *k-memo***.** The total privacy $\epsilon$ is set to 1 in this experiment. As shown in Fig. 5, the relative error is decreasing with the incremental size of *k-memo* as a whole; when $k$ has covered the period of data, the query error would be stable. There is a drop-off in the range $[0, 20]$, which means for most $1_{st}$-level grids, the irreplaceable $2_{nd}$-level count is more than 20. In Fig. 5(a) and (b), we see that *MRE* approximately stable when $k$ reaches 110 and 140 respectively for *T-BJ* and *T-R*, inferring that the period of location records in datasets are possibly 18 hours and 23 hours which conforms to people daily routine.

**Performance Comparison.** The performance of fixed uniform shows various trends with the granularity. Under fine partition, the total error goes up following the incremental query size, shown by blue and green lines in Fig. 6; yet goes down under coarse partition, like red lines. We believe that with the increased query size, the main error for answer is changed from uniform error to noise error, because the ratio between area and perimeter is increased as query range enlarges, lowering the uniform error percentage. For the large size, more grids are contained on refined resolution than on coarse one, causing that more noise is added declining the utility. On the contrary, noise is limited for the small size due to only few covered grids, but uniform error is much larger on coarse-grained domain. These two reason lead to the trend in Fig. 6. On the whole, the *MRE* of *RGP* remains low and outperforms fixed structure methods, no matter what granularity the partition is.

From the two performance analysis and comparison, we conclude that *RGP* outperforms methods with the fixed partition without *k-memo* over streams.

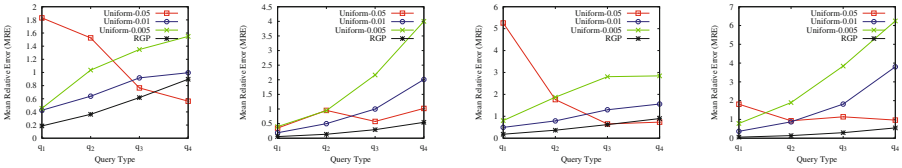


(a) *T-BJ* ($\omega = 10$) (b) *T-R* ($\omega = 10$) (c) *T-BJ* ($\omega = 50$) (d) *T-R* ($\omega = 50$)

**Fig. 5.** *MRE* vs $k$.

## 5 Related Work

Numerous work study releasing static geospatial data with region partition to improve publish utility without privacy compromise. Most of them used *differential privacy* [6] as theoretical support and corresponding mechanisms [7,8,11,13] to balance the utility and privacy. Cormode used non-uniform noise with post-processing [5] to generate spatial decomposition satisfying privacy requirement.




(a) *T-BJ* ($\epsilon = 1$) (b) *T-R* ($\epsilon = 1$) (c) *T-BJ* ($\epsilon = \frac{1}{2}$) (d) *T-R* ($\epsilon = \frac{1}{2}$)

**Fig. 6.** Performance Comparison.

Qardaji proposed an adaptive uniform partition on space [16] to improve the accuracy of query aiming at static geospatial dataset.

Some researchers work on differentially private releasing over infinite streams (*event-level* privacy) [3,4,12] and finite streams [9](*user-level* privacy). Fan [10] proposed a filtering and sampling-based method *FAST* for real-time publish. Kellaris [12] proposed $\omega$-*event privacy*, to merge the gap between these two levels over streams. Further, Andrés proposed a generalized differential privacy, *geo-indistinguishablity* [2], for location based systems. Xiao optimized the sensitivity set of differential privacy [18] to deal with the temporal correlations.

## 6 Conclusion

In this paper, we propose an approach *RGP* with new structure *k-memo* for geospatial streams release. This approach avoids individual privacy breach on real-time geospatial data under the *time correlation attack*. Meanwhile,*RGP* makes these synopses available on user-interaction model with improved accuracy for count query, which is favourable for various practical scenarios. In theoretical aspect, we prove privacy property of *RGP*, and illustrate its improvement of utility. Also, from a practical standpoint, we study the effectiveness of *RGP* with *k-memo* and show its significant error reduction achieved by *RGP* through experimental comparison with existing methods on real stream datasets. For the future work, we will investigate more sophisticated schemes to allocate privacy budget more appropriately over stream and try to answer broader types of query privately without much utility loss.

## References

1. Amici, R., Bonola, M., Bracciale, L., Rabuffi, A., Loreti, P., Bianchi, G.: Performance assessment of an epidemic protocol in vanet using real traces. Procedia Comput. Sci. **40**, 92–99 (2014)
2. Andrés, M.E., Bordenabe, N.E., Chatzikokolakis, K., Palamidessi, C.: Geo-indistinguishability: Differential privacy for location-based systems. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, pp. 901–914. ACM (2013)

3. Bolot, J., Fawaz, N., Muthukrishnan, S., Nikolov, A., Taft, N.: Private decayed predicate sums on streams. In: ICDT, pp. 284–295. ACM (2013)
4. Chan, T.-H.H., Li, M., Shi, E., Xu, W.: Differentially private continual monitoring of heavy hitters from distributed streams. In: Fischer-Hübner, S., Wright, M. (eds.) PETS 2012. LNCS, vol. 7384, pp. 140–159. Springer, Heidelberg (2012). doi:10.1007/978-3-642-31680-7_8
5. Cormode, G., Procopiuc, C., Srivastava, D., Shen, E., Yu, T.: Differentially private spatial decompositions. In: ICDE, pp. 20–31. IEEE (2012)
6. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). doi:10.1007/11787006_1
7. Dwork, C.: A firm foundation for private data analysis. Commun. ACM **54**(1), 86–95 (2011)
8. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). doi:10.1007/11681878_14
9. Fan, L., Xiong, L.: Real-time aggregate monitoring with differential privacy. In: CIKM, pp. 2169–2173. ACM (2012)
10. Fan, L., Xiong, L., Sunderam, V.: Fast: differentially private real-time aggregate monitor with filtering and adaptive sampling. In: SIGMOD, pp. 1065–1068. ACM (2013)
11. Inan, A., Kantarcioglu, M., Ghinita, G., Bertino, E.: Private record matching using differential privacy. In: EDBT, pp. 123–134. ACM (2010)
12. Kellaris, G., Papadopoulos, S., Xiao, X., Papadias, D.: Differentially private event sequences over infinite streams. VLDB **7**(12), 1155–1166 (2014)
13. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: FOCS, pp. 94–103. IEEE (2007)
14. McSherry, F.D.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: SIGMOD, pp. 19–30. ACM (2009)
15. de Montjoye, Y.A., Hidalgo, C.A., Verleysen, M., Blondel, V.D.: Unique in the crowd: the privacy bounds of human mobility. Scientific reports 3 (2013)
16. Qardaji, W., Yang, W., Li, N.: Differentially private grids for geospatial data. In: ICDE, pp. 757–768. IEEE (2013)
17. To, H., Ghinita, G., Shahabi, C.: A framework for protecting worker location privacy in spatial crowdsourcing. VLDB **7**(10), 919–930 (2014)
18. Xiao, Y., Xiong, L.: Protecting locations with differential privacy under temporal correlations. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1298–1309. ACM (2015)
19. Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., Huang, Y.: T-drive: driving directions based on taxi trajectories. In: SIGSPATIAL, pp. 99–108. ACM (2010)