Efficient Selection of Geospatial Data on Maps for Interactive and Visualized Exploration

Tao Guo, Kaiyu Feng
tguo001,kfeng002@e.ntu.edu.sg
SCSE, Nanyang Technological University, Singapore

Gao Cong
gaocong@ntu.edu.sg
SCSE, Nanyang Technological University, Singapore

Zhifeng Bao
zhifeng.bao@rmit.edu.au
RMIT University, Australia

ABSTRACT

With the proliferation of mobile devices, large collections of geospatial data are becoming available, such as geo-tagged photos. Map rendering systems play an important role in presenting such large geospatial datasets to end users. We propose that such systems should support the following desirable features: representativeness, visibility constraint, zooming consistency, and panning consistency. The first two constraints are fundamental challenges to a map exploration system, which aims to efficiently select a small set of representative objects from the current region of user's interest, and any two selected objects should not be too close to each other for users to distinguish in the limited space of a screen. We formalize it as the Spatial Object Selection (sos) problem, prove that it is an NP-hard problem, and develop a novel approximation algorithm with performance guarantees. To further support *interactive* exploration of geospatial data on maps, we propose the Interactive sos (isos) problem, in which we enrich the sos problem with the zooming consistency and panning consistency constraints. The objective of isos is to provide seamless experience for end-users to interactively explore the data by navigating the map. We extend our algorithm for the sos problem to solve the isos problem, and propose a new strategy based on pre-fetching to significantly enhance the efficiency. Finally we have conducted extensive experiments to show the efficiency and scalability of our approach.

KEYWORDS

Geospatial Data Visualization; Map Exploration; Sampling

ACM Reference format:
Tao Guo, Kaiyu Feng, Gao Cong, and Zhifeng Bao. 2018. Efficient Selection of Geospatial Data on Maps for Interactive and Visualized Exploration. In *Proceedings of 2018 International Conference on Management of Data, Houston, TX, USA, June 10–15, 2018 (SIGMOD),* 16 pages.
https://doi.org/10.1145/3183713.3183738

1 INTRODUCTION

Large collections of geospatial data are becoming increasingly available. Examples of such data include points of interest data from online Maps and social media websites (e.g., Yelp and FourSquare), geo-tagged photos in social photo sharing websites (e.g., Flickr and Instagram), geo-tagged micro-blogs (e.g., Twitter), geo-tagged news, etc. Such data are featured with both geospatial content and other content, such as texts and photos. It is useful to provide support for end-users to perform visualized exploration on such geospatial data on maps. We illustrate the desired features of such systems in the following example.

Example 1.1. Given a collection of points of interest (POIs), an end-user would like to browse a small number (denoted by k) of representative POIs for an area on an online map using her pad. Ideally, the set of selected POIs can well represent the POIs of the area (i.e., Representative Constraint), and they should not be too close to each other so that they will not overlap with each other on the map (i.e., Visibility Constraint) when shown on the pad screen. Figure 1 demonstrates Example 1.1, where a small number of representative POIs are shown to users (Figure 1(b)). The user may be interested in some POIs, and he may click one to check more information such as descriptions and user's voting score. Moreover, the "hidden" POIs that are represented by the user-viewing POI are also highlighted (Figure 1(c)) as related recommendations for users to explore.

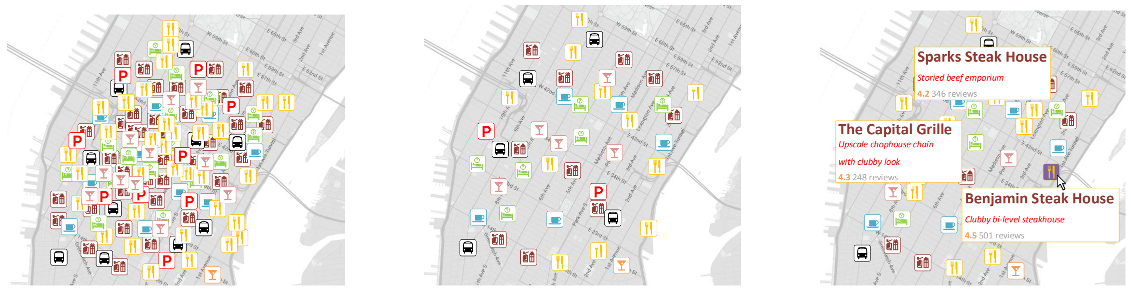
The example informally introduces the first goal of this paper.

Goal 1: Efficient Spatial Object Selection (sos)
Given the current region of user's interest, how to efficiently select a set S of k objects (among all the spatial objects falling in this region), so that it meets (i) Visibility Constraint — the distance between any two objects in S is larger than a given distance threshold θ, and (ii) Representative Constraint — the aggregated similarity between S and the whole spatial objects in that region is maximized. We define a general function (in Section 3.1) which can cope with various types of data resources and similarity metrics to meet the needs in different scenarios, as described at the beginning of introduction.

By reviewing the literature in the areas of cartographic selection and spatial sampling (see Table 1), we find that there have been some studies [31, 38, 39] taking the visibility constraint into account. However, none of them considers representativeness except for a study done by Drosou et al. [16], in which it is assumed that the representativeness of a spatial object is based on its spatial distance to other objects, and its proposed solution is built based on the assumption.

However, for a truly general solution, the representativeness/similarity definition should be application dependent. For instance, in Example 1.1, we could consider both the distance of two POIs and the semantic similarity of the two POIs. Furthermore, if additional information, such as the popularity of POI and the importance of tweets, is available, it should be taken into account when selecting the set of representative objects.

SIGMOD, June 10–15, 2018, Houston, TX, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-4703-7/18/06…$15.00
https://doi.org/10.1145/3183713.3183738



(a) Without any selection, it is messy to view all the POIs.

(b) After objects selection, only few representative POIs are shown to users.

(c) The user can click one for more detailed information, and other "hidden" similar POIs will also be highlighted.

Figure 1: Spatial Object Selection Example

To this end, we propose the sos problem which for the first time takes both visibility constraint and representative constraint into consideration in object selection, and we accept any similarity metric in defining visibility constraint and representative constraint.

Goal 2: Interactive Spatial Object Selection (isos)
The sos problem deals with the object selection problem at the current region of user's interest. To support the objective of exploring a geospatial dataset, we should provide the interactive exploration function so that users can interact with the map by performing map navigation operations, including zoom-in, zoom-out and panning. As users navigate the map, the proposed visualized exploration system needs to maintain consistency when selecting a new set of representative objects for the new map region, as illustrated in the following example.

Example 1.2. Continue with Example 1.1. The user now would like to further explore the POI dataset by zooming-in on the map as illustrated in Figure 2(a)). There are 9 POIs in the regions, denoted by $o_1, \ldots, o_9$. The black nodes are visible to the user. Those spatial objects that are visible in the current granularity and fall in the new map region should also be visible in the new map region at a finer granularity. As a specific example, object o_5 should be visible after zooming in. In addition, objects o_3 and o_4 become visible after zooming in. Similarly, when zooming out, the set of spatial objects that are visible in the coarser granularity should be a subset of the spatial objects that are visible in a finer granularity as illustrated in Figure 2(b). When the user pans the map from the current window, denoted by r_1, to a nearby region r_2 that overlaps with r_1, the objects appearing in the overlapping area before the movement should also appear in the region after the movement. As shown in Figure 2(c), objects o_1 and o_5 are still visible after the panning.

The consistency constraint with respect to zoom-in and zoom-out operations is referred to as *zooming consistency*, and the consistency constraint with respect to panning operation is referred to as *panning consistency*. Based on the sos problem, the isos problem additionally considers the zooming consistency and panning consistency to provide the seamless experience for end-users to interactively navigate a map to explore the data.

In order to achieve the aforementioned two goals, we make the following contributions.

First, we formally introduce four desired features that an interactive visualized exploration system on a map should meet, i.e., representative constraint, visibility constraint, zooming consistency constraint and panning consistency constraint. To the best of our knowledge, this is the first work that is able to meet all the four features to achieve an interactive visualization map exploration system.

Second, we propose the problem of Spatial Object Selection (sos) considering representative constraint and visibility constraint, and prove that it is an NP-hard problem. The intractability result motivates us to design approximation algorithms for the sos problem. We propose a greedy algorithm with performance guarantees and it has an approximation ratio of 1/8.

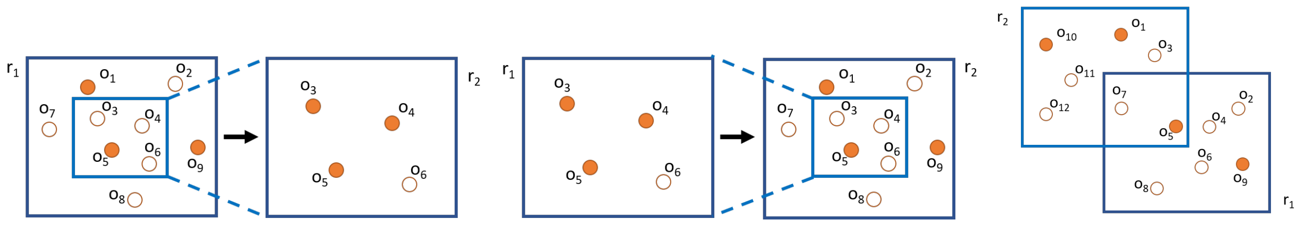
Third, we propose the problem of Interactive Spatial Object Selection (isos) by enforcing the zooming consistency constraint and panning consistency constraint on top of the sos problem to support interactive exploration of geospatial data on maps. We extend the proposed greedy algorithm to address the isos problem. We further propose a new approach to improve the efficiency of our algorithm in finding the new set of representative objects to support interactive exploration in isos, by up to 2 orders of magnitude (Section 5).

Fourth, to improve the efficiency of sos and isos on large datasets, we propose to use sampling strategy (Section 6). We prove that it has a theoretical guarantee, and the experiments show that it only needs less than 2% of the whole data (which is of size up to 100 million) to achieve a very small error bound (Section 7.3.2).

Finally, we have conducted extensive experiments over three real-life datasets to evaluate the performance of our proposed algorithms (Section 7). The experimental result shows that our greedy algorithm is efficient. Our pre-fetching technique further improves the efficiency of our greedy algorithm for isos by almost 2 orders of magnitude, and it ensures a very high response speed (usually 0.1 second) for new object selection to cater for zoom-in, zoom-out and panning operations.

2 RELATED WORK

Existing Map Services Although visualization of geographical data, such as POIs, is an important part of online map systems such as Google Maps and MapQuest, the main focus is to help users find the result they want, assuming that users know their search intention and formulate such intention as a query. When working with queries, existing map services retrieve a subset of spatial objects based on the user's (keyword) query, and a record is evaluated as "good" if it matches user's intention. Without the userâĂŹs query, Google Maps chooses objects to be shown on map according to their weight by default, i.e., those objects that can maximize the



(a) Zooming in (b) Zooming out (c) Panning

Figure 2: Interactive Operations on Map

Table 1: Comparison of closely related work on different aspects

Methods	Zooming Consistency	Movement Consistency	Visibility Constraint	Representativeness	Similarity Metrics	Online
Our Work	Yes	Yes	Yes	Yes	General	Yes
Sarma et al. [14]	Yes	No	No	No	N.A.	No
Konstantin et al. [31]	Yes	No	Yes	No	N.A.	No
DisC [16]	No	No	Yes	Yes	Distance	Yes
Nutanong et al [38]	No	Yes	Yes	No	N.A.	Yes
Mahdian et al. [33]	No	No	No	No	N.A.	Yes
Peng et al. [39]	Yes	No	Yes	No	N.A.	Yes

total weights are selected [14]. Our work is different from existing map services, because we focus on how to explore a large collection of geospatial data based on representative samples, when users have no prior knowledge on the data or do not know exactly what to search at that moment. Our system can also work with existing map services as below: on top of all relevant results retrieved by the map search engine for a user query, our method is employed to choose the representative ones among all results which are overwhelming if displaying them all. We believe that this is the first paper to study the geospatial object selection problem that takes into account the representativeness of objects to support interactive visualized exploration. Table 1 summarizes the difference between our work and closely related work in terms of the aforementioned consistencies, constraints, similarity metrics adopted and whether online computation is supported.

Cartographic Generalization and Selection Our work is related to cartographic generalization [19, 40, 45, 50], which is a classic research topic. One of the challenges in cartographic generalization is sifting through all of the data available and deciding what to place on the map at a given zoom level. We next review the recent studies addressing the challenge.

Sarma et al. study the map thinning problem [14], which is to determine appropriate samples of data to be shown on each predefined geographical region and zoom level on maps. Konstantin et al. [31] extend the work [14] by proposing a concept similar to the visibility constraint defined in our work. Our work differs from the work [14, 31] in the following two aspects. First, Pre-defined Granularity & Region Cell vs. Arbitrary Granularity & Region. The approaches [14, 31] are precomputation-based, and perform an *offline* computation of selecting objects for all cells of different zoom levels. However, in real-world a user's region of interest may not match well with any of the pre-defined cells, and thus the selected objects for those pre-defined cells may not provide a good solution to our SOS problem. Moreover, the pre-selected objects will not work if users specify some filtering condition, e.g., names should contain "restaurant." Second, our work considers the representative constraint while the previous ones do not.

Nutanong et al. [38] define the problem of sampling large geographic dataset falling within a region of user interest for display, and they propose to utilize SQL statements to achieve several features like visibility constraint and object distinctiveness (representing importance of object). However, it does not explicitly address the zooming consistency and panning consistency constraints. Mahdian et al. [33] propose a POI selection problem, which aims to highlight a subset of POIs with the maximum utility instead of showing all to users. The utility of each POI is a user-defined function, and it takes into account the POIâĂŹs relevance and quality. This is similar to the weight in our problem definition. The problems in the two studies are different from our problem as our work considers the representativeness of geospatial objects and the visibility constraints. Furthermore, we explicitly address the consistency constraint to support visualized exploration as users navigate a map.

The map labeling problem [8] is to select a subset from a geographic dataset to maximize the sum of their importance without any overlap between the textual labels of selected objects. Peng et al. [39] develop a filter-and-refine algorithm for the problem. The map labeling problem is different from our work in that it does not consider the representativeness of objects.

Interactive Data Exploration Our work is related to studies on interactive exploration of geospatial data. The existing work supporting Interactive Data Exploration [28] is query-based, where users interactively issue multiple queries until the user is satisfied with the result. However, these approaches have different focus from our work that aims to select representative geospatial objects. For example, ScalaR [6] is a system that dynamically performs resolution reduction by inserting aggregation, sampling or filtering operations to reduce the size of the result when the expected result of a DBMS query is too large to be effectively rendered.

Spatial Sampling Our work is related to sampling techniques [10, 32]. Recently Wang et al. [49] utilize map APIs to sample POIs, and a disk-resident spatial sampling problem is studied in [48]. The objective of both studies is to achieve accurate statistical aggregations such as calculating the average price of all sale transactions in

NYC based on a small sample of data, and thus each object should be sampled with equal probability (without bias). The challenge is how to take random samples uniformly distributed while the whole data distribution is unknown. However, our problem and objective are different from uniform spatial sampling. We aim to choose representative objects utilizing the similarities between objects.

Spatial sampling is also a classic problem in statistics. Apart from uniform spatial sampling, other sampling methods include stratified sampling [36], cluster sampling [37], multistage sampling [9], two-phase sampling [27], and sequential sampling [42]. These methods assume the samples are independent and identically distributed (i.i.d.), and thus they are inapplicable to our problem. Sampling for spatial autocorrelation (or spatial dependence) [34, 35] measures how a set of objects tend to be clustered together in space. Sampling for spatial heterogeneity [12, 22, 23] evaluates the uneven distribution of various densities of each species within an area, which is typically used in environmental sciences. In these two problems, samples are assumed to be i.i.d., and the distribution of entire population and the objects similarities are not utilized during the sampling; while in our problem both properties need to be considered.

Query Result Diversification Query result diversity has been extensively studied to improve the user's experience (e.g., [3, 16, 20, 41]). Our work is related to DisC [16], which aims to select a subset of diversified spatial objects that can represent the whole dataset. However, the algorithm for solving the DisC problem is dependent on the distance metric, and is not applicable to a general definition of representativeness based on any similarity function. Furthermore, we consider the consistency constraint to support interactive exploration, which is not considered in DisC.

Approximate Query Processing Approximate Query Processing (AQP) techniques [21] are proposed to achieve low latency for interactive data exploration. AQP systems like AQUA [1], BlinkDB [2] and DICE [30] are based on offline sampling that requires extensive preprocessing cost, and particular techniques are developed depending on a certain type of AQP needed in a particular context. Systems that perform online aggregation [25] (e.g. CONTROL [24], DBO [29], HOP [11], FluoDB [51]) allow users to get iteratively refined approximate answers until the users terminate the query execution. AQP studies on spatial data [4, 7, 13] focus on ad-hoc queries like spatial join, kNN and top-k problems. These AQP techniques are inapplicable to our problem for two reasons: first, like the Spatial Sampling studies, the AQP systems return aggregations as the result, which ignores the relationship between objects; second, since we aim to select a set of objects as the result, users will be annoyed if the result is iteratively updated during the exploration.

3 PROBLEM STATEMENT

We first introduce two concepts and the Spatial Object Selection (SOS) problem, which is to support end-users to explore a geospatial dataset on the limited space of a map. Then we introduce the Interactive Spatial Object Selection (ISOS) problem.

3.1 Representative and Visibility Constraints

A geospatial object o is represented by a triple $o = \langle \lambda, \omega, \mathcal{A} \rangle$, where $o.\lambda$ is the location where o is posted, $o.\omega$ is the weight (normalized in $[0, 1]$), which can be either computed from some attributes to represent the popularity or importance of the object, or simply be assigned with a unit weight, and $o.\mathcal{A}$ is a set of attributes of the object. For example, a geo-tagged tweet is a geospatial object. The textual content of the tweet is an attribute of the object. A geo-tagged image is also a geospatial object. The image content is viewed as an attribute of the geospatial object. In this paper, we consider a collection of geospatial objects, denoted by O.

It is overwhelming and annoying to display all geospatial objects to end users in a window of a map. It remains a challenge on how to select a small set S of geospatial objects to represent the collection O of geospatial objects, where $S \subseteq O$.

Representative Constraint Usually, a larger S leads to a more representative set. However, visualizing too many geospatial objects in a window or screen of a map is overwhelming for users to find out truly useful information. For example in Google Maps, around 500 geospatial objects are displayed to users in a single window [14]. Therefore, we aim to enforce a *representative constraint*: choose k objects to represent the set of geospatial objects falling within the region of user's interest, where different users can have different specifications of k. We assume w.l.g. that k is much smaller than the set of objects falling within the region of interest.

We compute the representative score of an object by its similarity with other objects. We denote the similarity between two objects o_i, o_j by a function $Sim(o_i, o_j)$ that is computed from the attributes $o_i.\mathcal{A}$ and $o_j.\mathcal{A}$, and then normalized in $[0, 1]$. In this paper, we leave $Sim(.,.)$ as a general function. We believe that a general function is important for us to cope with various types of resources to meet the needs in different scenarios. For example, we may use textual similarity and geospatial distance to measure the similarity of two geo-tagged tweets.

With a given similarity function $Sim(o_i, o_j)$ between objects, we define the similarity between an object o and a set S of objects by:

$$Sim(o, S) = \max_{o' \in S} Sim(o, o') \tag{1}$$

Equation 1 measures how well an object o can be represented by set S. Intuitively, each object o is represented by the object in S that is most similar to o. Our proposed solution can also be extended to handle other aggregation metrics, such as *sum* or *avg*. For the ease of presentation, we only discuss *max* in this paper.

We next define the representative score of S, namely $Score(S)$, by the similarity between S and O, which incorporates the weight of each object o and is evaluated by

$$Score(S) = Sim(O, S) = \frac{1}{|O|} \sum_{o \in O} o.\omega \times Sim(o, S) \tag{2}$$

Here we combine the weight of o and the similarity between S and o, which can be viewed as the utility of object o. Equation 2 aims to maximize the total utilities of all the objects, where similar definition is also used in [33].

Visibility Constraint. Similar to the previous work on query result diversification and cartographic selection (e.g. [14, 16, 38]), we also enforce that any two selected objects should not be too close to each other, so that users can distinguish them on the map.

3.2 Advantages of the Representative Score

Maximizing the representativeness of all the objects. To describe how well a subset S can represent the whole collection of geospatial objects O, an intuitive idea is to define how well S represents each single object o in O (Equation 1). Since the representativeness of S on an object o_1 should be irrelevant to that of S on another object o_2, the representative score of S w.r.t. O is simply an aggregation of the representative score over each object in O (Equation 2). Then our objective is to find S that can maximize the $Sim(O, S)$.

Maximizing the utility of each object. Instead of treating each object equally, we combine the weight of each object o and the representativeness of S on it as the utility of o, such that the important objects are more likely to be represented. Intuitively, if an object o can be represented by other objects in S, the similarity between o and the objects in S should be high. In particular, an object should always represent itself. That explains why we have the weight associated with objects in S in Equation 2.

Supporting various kinds of geospatial objects. Since we use the attributes $o.\mathcal{A}$ of the object to measure the similarity, different types of geospatial objects can be supported by our method. Another salient feature is that for a specific type of objects, we can vary the definition of the similarity function for different applications without ad-hoc algorithms.

Extending map exploration. The efficient exploration feature introduced in Figure 1(c) can be directly supported by the definition of Representative Constraint. Each object o that is not shown in the map ($o \in O - S$) is represented by a selected object o' ($o' \in S$), $Sim(o, S) = Sim(o, o')$. It indicates that o' is the most similar object to o among all objects shown in the window, and if we view o' for details o can be displayed as an extension of map exploration.

A user study in Section 7.2 demonstrates that the representative score is consistent with the users' satisfaction.

3.3 Spatial Object Selection (SOS) Problem

We are now ready to define the SOS problem.

Definition 3.1. **Spatial Object Selection (SOS) Problem** Given a set of geospatial objects $O = \{o_1, \ldots, o_n\}$ in a region of interest, a distance threshold θ, and an integer k, the SOS problem aims to select a subset of k objects $S \subseteq O$ such that

(1) $dist(o_i, o_j) \geq \theta$ for any $o_i, o_j \in S$, and
(2) $Sim(O, S)$ is maximized.

The first condition guarantees that the users can easily distinguish two close geospatial objects on the map. The second condition ensures that the selected geospatial objects can well represent all geospatial objects in the region. Note that if users want to specify filtering condition on the set of objects O, e.g., objects should contain keyword "president election," existing database querying engines can be employed to perform the filtering.

THEOREM 3.2. *The SOS Problem is NP-hard.*

PROOF. We prove the theorem by a reduction from the decision version of the Minimum Dominating Set problem, which is known to be NP-hard. The Minimum Dominating Set (MDS) problem aims to find the minimum number of nodes such that each node in the graph is either selected or a neighbor of the selected nodes. The decision problem is to decide whether there is a solution of no more than k nodes.

Consider a set of geospatial objects, each of which has the same weight. We assume that for each pair of objects o_u and o_v we have $Sim(o_u, o_v) \in \{0, 1\}$. The distance threshold θ is set small enough such that any pair of objects fulfills the Visibility Constraint. For any instance of the Minimum Dominating Set decision (MDSd) problem, we can build an SOS problem to solve it as follows. Given a graph $G(V, E)$, we map each node u_i to an object o_i. If there is an edge between nodes i and j, we set $sim(o_i, o_j)$ to be 1, otherwise $sim(o_i, o_j) = 0$.

(1) Given an MDSd problem, if nodes T are the result, we let $S = \{o_i | u_i \in T\}$, and obviously S is the result of the SOS problem.
(2) Given the result S of the SOS problem, if we have $|O| = \sum_{o \in O} Sim(o, S)$, for each object o we know that $Sim(o, S) = 1$. According to Equation 1, we have either $o \in S$ (where $Sim(o, o) = 1$) or o is represented by S (where $Sim_{o' \in S}(o, o') = 1$). Let $T = \{u_i | o_i \in S\}$ be a set of nodes, and T is the result of the MDSd problem, since for each node u we have either $u \in T$ or $(u, v) \in E \& v \in T$ (u is dominated by T).

Therefore, the SOS problem is NP-hard. □

3.4 Interactive Exploration on Map

When users navigate the map to explore the geospatial data, they can perform three different navigation operations: (1) zooming in, (2) zooming out, and (3) panning.

Zooming in (out). By zooming in (out), the map is displayed with a finer (coarser) granularity and more (fewer) details in the region are visible to users (while the center of the map remains unchanged).

Panning. Users can move the displayed region to a new place with the same granularity.

With regard to the three operations, in order to provide a seamless experience for the end-user, the map needs to fulfill the *zoom consistency* and the *movement consistency* constraints.

Zooming Consistency Constraint: for any object o appearing at any coarse granularity, it should also appear in all finer granularities of regions containing the location of that object as users zoom the map.

Panning Consistency Constraint: at a certain granularity, for any object o appearing at a region, it should also appear in all other regions which contain the location of this object as user pans the map.

Next we present how the two consistency constraints affect the SOS problem as users navigate the map. We consider the three examples shown in Figure 2(a), 2(b), and 2(c), respectively. Each point in the map is a POI. The set of POIs that are visible to users are marked with red color. Assume that three objects should be selected to display to users in the region of interest.

Example 3.3 (Zoom in). Consider the example in Figure 2(a). Before the zoom-in operation, region r_1 is displayed to the user. There are nine objects in the map region ($o_1, \ldots, o_9$), in which o_1, o_5 and o_9 are visible to users. After the zoom-in operation, region r_2 is displayed to the user. We need to select three objects from r_2 to display. However, since object o_5 is visible to the user before zooming in, it should still be visible. Thus, we need to select two objects from all objects in r_2 excluding o_5 to display.

Example 3.4 (Zoom out). Consider the example in Figure 2(b). Before the zoom-out operation, region r_1 is displayed to the user. There are four objects in the map region ($o_3, \ldots o_6$), in which o_4, o_5 and o_6 are visible. After the zoom-out operation, region r_2 is displayed to the user with a coarser granularity. We need to select three objects from r_2 to display. Note that o_6 cannot be selected, since it is not visible in the finer granularity. This means only the black nodes in r_1 can be selected in the new map region. Thus, in the new map region, we need to select three objects from all objects in $r_2 \setminus r_1$ and the black nodes in r_1.

Example 3.5 (Panning). Consider the example in Figure 2(c). Before the panning operation, region r_1 is displayed to the user. There are 7 objects in the map region, in which o_5 and o_9 are visible. After the panning operation, region r_2 is displayed to the user. We need to select three objects from r_2 to display. Note that object o_7 cannot be selected, since it is not visible before panning. Moreover, object o_5 should be selected because it is visible before panning. Thus, we need to select two objects from the region $r_2 \setminus r_1$ to display.

From the three examples, we observe that in the new map region (1) some objects must be included into the representative set, and (2) some objects must be excluded from the representative set. Specifically, in the new map region, there is a set D of geospatial objects that must be selected because of the consistency constraints. In these three examples, the set D are $\{o_5\}$, $\{\}$ and $\{o_5\}$, respectively. There is also a set G of candidate objects only from which we can select objects into the representative set. In the above examples, the set G are $\{o_3, o_4, o_6\}$, $\{o_1, o_2, o_3, o_4, o_5, o_7, o_8, o_9\}$ and $\{o_1, o_3, o_{10}, o_{11}, o_{12}\}$, respectively.

3.5 Interactive Spatial Object Selection (ISOS) Problem

We proceed to introduce the *interactive spatial object selection* (ISOS) problem, which enforces the zooming and panning consistency constraints on top of the SOS problem to support interactive exploration of geospatial data.

Definition 3.6. **Interactive Spatial Object Selection (ISOS) Problem** Given a set of geospatial objects $O = \{o_1, \ldots, o_n\}$ in a region, a distance threshold θ, and an integer k, let $G \subseteq O$ be the set of candidate geospatial objects, $D \subseteq O$ be the set of geospatial objects that should remain visible to users after users perform any of the three navigation operations according to the zooming consistency and panning consistency. The ISOS problem aims to select a subset $S \subseteq G$, $|S \cup D| = k$, such that

(1) $dist(o_i, o_j) \geq \theta$ for any $o_i, o_j \in S \cup D$, and
(2) $Sim(O, S \cup D)$ is maximized.

4 PROPOSED ALGORITHM FOR SOS

In this section we present the proposed greedy algorithm for finding an approximate solution to the SOS problem, and we prove that the proposed algorithm has an approximation ratio of 1/8.

4.1 A Greedy Algorithm

The intractability result motivates us to develop a greedy algorithm for the SOS problem. To select a set S of geospatial objects from the set O of objects in a greedy manner, we need to take both the representative constraint and the visibility constraint into consideration. To choose representative objects, the geospatial object that we greedily select should have a high marginal similarity increase. To impose the visibility constraint, the distance between the new selection and any previously selected object should be no less than the given threshold. Specifically, in the proposed greedy algorithm, we select the representative set of objects iteratively. In each iteration, we select the geospatial object with the maximum marginal similarity increase. Then we remove the remaining geospatial objects that do not satisfy the visibility constraint, i.e., its distance to the newly-selected geospatial object is less than the given threshold. The algorithm terminates when k geospatial objects are selected.

One main challenge is how to efficiently find the object with the maximum marginal similarity increase in each iteration. One naive idea is to compute the marginal increase for each geospatial object. However, this is prohibitively expensive. To address this challenge, we utilize a "lazy-forward" strategy based on the following lemma.

LEMMA 4.1. *(Submodularity.) Let S and T be two sets of geospatial objects, and $S \subseteq T$. Let v be a newly inserted object. We have $Sim(O, S \cup \{v\}) - Sim(O, S) \geq Sim(O, T \cup \{v\}) - Sim(O, T)$.*

PROOF. See Appendix A. □

Since we select one object in each iteration, Lemma A.1 shows that the marginal similarity increase of an object (corresponding to v in Lemma A.1) in current iteration (corresponding to T) is not larger than the marginal similarity increase in the previous iteration (corresponding to S). Therefore, based on this lemma, to avoid massive recomputation we utilize the "lazy forward" strategy, which works as follows: For each geospatial object o, we construct a tuple $t = \langle o, \Delta(o), Iter\rangle$, where $\Delta(o) = Sim(O, S \cup \{o\}) - Sim(O, S)$ is the marginal similarity increase caused by adding object o to the current representative set S, and $Iter$ records the iteration in which $\Delta(o)$ is computed. We use a max-heap to maintain the tuples for all spatial objects according to $\Delta(o)$. In the i-th iteration, we check the top tuple t in the heap. If $\Delta(o)$ of tuple t is not computed in this iteration, i.e., $t.Iter \neq i$, the value $\Delta(o)$ is not up-to-date. Note that $\Delta(o)$ can serve as an upper bound for its real marginal increase according to Lemma A.1. Therefore, we need to recompute its marginal increase and push it back to the heap. We keep checking the top tuple until the marginal increase of the top tuple is computed in this iteration ($t.Iter = i$)—The real marginal increase of the geospatial object of the top tuple is higher than the upper bound marginal increase of the other geospatial objects. Therefore, the corresponding geospatial object is picked into the representative set.

Note that in the "lazy forward" strategy, we recompute the marginal increase only for those objects appearing as the top tuple in the heap, rather than for all geospatial objects. For those objects whose marginal increases are computed in previous iterations, their values become outdated, but they can still serve as upper bound values for the marginal increase in the current round. This guarantees the correctness of the "lazy forward" strategy.

The details of the algorithm are shown in Algorithm 1. It takes as input a set O of geospatial objects, the size of the representative set k, and a distance threshold θ. Its output is the set of selected objects

S, and it is initialized as an empty set (line 1). First, the algorithm initializes the max-heap with tuples (lines 2–3). Specifically, for each geospatial object o, it pushes a tuple $\langle o, Sim(O, \{o\}), 0\rangle$ into the heap, where $Sim(O, \{o\})$ is the marginal similarity increase of adding o to S (S is empty set now). Then, the algorithm selects the representative set iteratively (lines 4–12). In each iteration, the algorithm keeps checking the top tuple from the heap (line 6). If the marginal increase of the top tuple t is not computed in this iteration, we need to recompute its marginal increase (line 7) and update $t.Iter$ (line 8) by the current iteration number. We then push t back to the heap. When the marginal increase of the top tuple is computed in current iteration, we select the corresponding object $t.o$ into the representative set (line 10). In addition, for any geospatial object o' whose distance to o is smaller than the threshold, we remove the corresponding tuple from the heap due to the visibility constraint (lines 11–12). An example is demonstrated in Appendix D.

Algorithm 1: Greedy

```
input  : A set of geospatial objects $O$, size $k$, distance threshold $\theta$
output : A subset of objects $S$
1  $S \leftarrow \emptyset$;
2  foreach object $o$ in $O$ do
3      Push $\langle o, Sim(O, \{o\}), 0\rangle$ into heap $H$;
4  while $|S| < k$ and $H$ is not empty do
5      $t \leftarrow$ pop the top tuple from heap $H$;
6      while $t.Iter \neq |S|$ do
7          $t.\Delta(o) \leftarrow Sim(O, S \cup \{o\}) - Sim(O, S)$;
8          $t.Iter = |S|$;
9          Push $t$ to heap $H$;
10         $t \leftarrow$pop the top tuple from heap $H$;
       // select $o$ as result
11     $S \leftarrow S \cup \{t.o\}$;
12     foreach object $o'$ s.t. $dist(o', o) \leq \theta$ do
13         remove $o'$ from $H$;
14 return $S$;
```

Complexity It takes $O(n)$ time to recompute the marginal similarity increase for a geospatial object, where n is the number of geospatial objects in O. Let n_c be the number of geospatial objects whose marginal increases are recomputed in the k iterations. The time complexity of the greedy algorithm is $O(n_c \cdot n)$. In practice, n_c is much smaller than n.

4.2 Approximation Ratio Analysis

We proceed to analyze the approximation ratio of the proposed greedy algorithm. We first present two lemmas that will be leveraged to prove the approximation ratio.

Lemma 4.2. *Let S and T be two sets of geospatial objects, and $S \subseteq T$. We have*

$$Sim(O, S) \leq Sim(O, T) \tag{3}$$

Proof. Since $S \subseteq T$, we have $max_{o' \in S} Sim(o, o') \leq max_{o' \in T} Sim(o, o')$. Therefore, we have $Sim(O, S) \leq Sim(O, T)$. □

Lemma 4.3. *Let S be a set of geospatial objects that satisfy the visibility constraint. Let o be a geospatial object and $o \notin S$. At most 7 objects in S conflict with o.*

Proof. See Appendix B.

With the aforementioned lemmas, we are ready to prove the approximation ratio of the greedy algorithm.

Theorem 4.4. *The Greedy algorithm has an approximation ratio of* $1/8$.

Proof. See Appendix C.

5 PROPOSED ALGORITHM FOR ISOS

In this section, we first present the greedy algorithm for the isos problem, and then propose a pre-fetching strategy to accelerate the processing for the isos problem.

5.1 A Greedy Algorithm

The isos problem extends the sos problem in two aspects: (1) The isos problem selects objects from the candidate set G, while the sos problem selects from all objects in the region of interest. (2) The selected objects and the pre-determined set of objects D together form the representative set in the isos problem, while the representative set in the sos problem consists of the selected objects only. We can slightly extend the greedy algorithm to solve the isos problem.

The main idea of the greedy algorithm is still the same. In each iteration, we select the geospatial object with the maximum marginal similarity increase. To make the greedy algorithm work for the isos problem, we make the following changes: (1) Since the representative set S must contain all objects in the pre-determined set D, we initialize the set S with all geospatial objects in D, i.e., $S \leftarrow D$ in line 1 in Algorithm 1. (2) Because we can only select geospatial objects from the candidate set G, we initialize the heap H by only using the objects in G. In line 2 of Algorithm 1, we compute the marginal increase for each object $o \in G$.

With the two modifications, the greedy algorithm can be used to find a set of representative geospatial objects for the isos problem.

5.2 Pre-fetching Strategy

We proceed to present the new idea of using pre-fetching to speed-up the greedy algorithm. When an end-user explores the geospatial data on the map, she may (1) check the content in a displayed map region, (2) perform a navigation operation, like zoom-in, zoom-out, and panning, and then (3) wait for the visualized exploration system rendering the set of representative geospatial objects on the map. User may repeat such kind of exploration several times until she is satisfied with the results. To reduce the waiting time of the user and provide the user with a seamless browsing experience, we propose to pre-fetch and pre-compute some useful information while the user is still in step 1. In our work, we focus on how to utilize the pre-fetched data to accelerate the greedy algorithm for the isos problem. We are not solving the problem of predicting the user's next region of interest, which is addressed by Leilani et al. [5]. In fact, this work is complementary to our work, and can be employed to predict what region of data to pre-fetch.

The main challenges in designing a pre-fetching strategy include (1) what kind of information should we pre-fetch and pre-compute, and (2) how the information can be used to accelerate the selection of representative set.

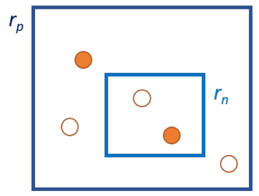


Figure 3: Zoom-in

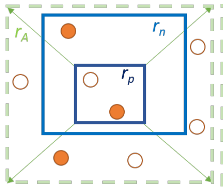


Figure 4: Zoom-out

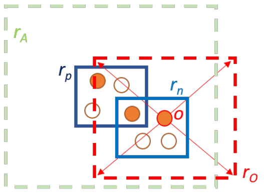


Figure 5: Panning

To address the aforementioned challenges, we first analyze the bottleneck of the greedy algorithm. Because the "lazy forward" strategy can greatly speed up the subsequent computation of marginal similarity increase, the bottleneck of Algorithm 1 is the initialization of the heap. In the initialization, we need to compute the marginal increase of representative score for each object in G, and push the tuple for each object into the heap. This initialization takes $O(n \cdot |G|)$ time, where n is the number of objects in the map region, and $|G|$ is the number of objects in the candidate set. A natural idea is that if we can estimate an upper bound for the marginal increase for each object, we can use the "lazy forward" strategy in the first iteration in the greedy selection.

Next, we present a pre-fetching strategy to estimate the upper bound of the marginal increase for each object in initialization of the heap, in the context of the three navigation operations, respectively.

5.2.1 Pre-fetching for Zoom-in. Recall that after the zoom-in operation, the new region of interest on the map is inside the old region of interest. Figure 3 shows a zoom-in example, where r_p is the old region of interest, and r_n is a possible new region of interest after the zoom-in operation.

Let O_p be the set of objects in r_p, and O_n be the set of objects in the new region r_n of interest. Let D be the set of pre-determined geospatial objects in r_n according to the zooming consistency constraint. To accelerate the greedy algorithm, we estimate the marginal increase of representative score for each object o in $O_n \setminus D$, i.e., $Sim(O_n, D \cup \{o\}) - Sim(O_n, D)$ by the following lemma.

Lemma 5.1. $Sim(O_n, D \cup \{o\}) - Sim(O_n, D)$
$\leq \sum_{o' \in O_p} Sim(o, o')$.

Proof. According to Lemma A.1, we have $Sim(O_n, D \cup \{o\}) - Sim(O_n, D) \leq Sim(O_n, \{o\})$. Since $O_n \subseteq O_p$, we have $\sum_{o' \in O_n} Sim(o', o) \leq \sum_{o' \in O_p} Sim(o', o)$. Therefore, $\sum_{o' \in O_p} Sim(o, o')$ is the upper bound of the marginal increase for object o. □

According to Lemma 5.1, for any new region of interest r_n as a result of zoom-in from r_p, we pre-compute the upper bound of marginal increase of representative score for every object in O_p. Then, once the zoom-in operation is performed, we can find the geospatial objects in the new region of interest and obtain their upper bounds in $O(1)$ time.

5.2.2 Pre-fetching for Zoom-out. Recall that after the zoom-out operation, the new region of interest on the map must contain the old region of interest. Figure 4 shows a zoom-out example, where the region r_p is the old region of interest and region r_n is one possible map region after the zoom-out operation. Since r_n must contain region r_p, let r_A be the union of all possible new regions of interest.

Let O_n be the set of objects in the new region of interest, and O_A be the set of objects in the union of all possible new regions of interest. Let G be the candidate set of geospatial objects in the new region of interest according to the zooming consistency constraint. To accelerate the greedy algorithm, we estimate the marginal increase of representative score for each object $o \in G$, i.e., $Sim(O_n, \{o\})$ in the new map region as follows.

Lemma 5.2. $Sim(O_n, \{o\}) < \sum_{o' \in O_A} Sim(o, o')$

Proof. According to Lemma A.1, we have $Sim(O_n, \{o\}) < Sim(O_A, \{o\})$. According to the definition of the representative score, we have $Sim(O_A, \{o\}) = \sum_{o' \in O_A} Sim(o, o')$. □

According to Lemma 5.2, we can pre-compute the upper bound of the marginal increase for each object o in the union of all possible new regions of interest. Then, once the zoom-out operation is performed, we can find the objects in the new map region and obtain their upper bounds of the marginal increase in $O(1)$ time.

5.2.3 Pre-fetching for Panning. Recall that after the panning operation, the new region of interest has the same size as the old region of interest on the map. Figure 5 shows a panning example, where r_p is the old region of interest before panning, and r_n is one possible new region of interest after panning. Since r_n must overlap with r_p, the union of all possible new regions of interest is r_A, as shown in Figure 5.

Let O_n be the set of objects in a new region of interest, and O_A be the set of objects in the union of all possible new regions of interest. Let G and D be the set of candidate objects and the set of pre-determined objects according to the panning consistency constraint, respectively. We estimate the upper bound of the marginal increase of representative score for each object $o \in G$, i.e., $Sim(O_n, D \cup \{o\}) - Sim(O_n, D)$ as follows.

Lemma 5.3. *For an object o, we draw a square r_o centered at o with a width twice of the old region of interest, as shown in Figure 5. Let O_r be the set of geospatial objects in the overlapped region $r_A \cap r_o$. We have $Sim(O_n, D \cup \{o\}) - Sim(O_n, D) \leq \sum_{o' \in O_r} Sim(o, o')$*

Proof. The new region of interest is inside the overlapped region $r_A \cap r_o$, thus $O_n \subseteq O_r$. The proof follows the proof for Lemma 5.2 □

By Lemma 5.3, we can pre-compute the upper bound of marginal increase of representative score for every object o in O_A. Once the panning operation is performed, we can find the geospatial objects in the new region and obtain their upper bounds in $O(1)$ time.

6 A SAMPLING EXTENSION

We propose a sampling-based method to improve the efficiency of our algorithm, which is especially useful when the size of geospatial objects $|O|$ is large. We prove that with a very high probability this method returns a solution with a very small error ϵ compared to the optimal one. We introduce the sampling extension for the sos problem for simplicity, and similar conclusion can be easily extended for the isos problem.

6.1 A Sampling Method

When the number of candidates in O is large, it is time-consuming to obtain a result for sos problem even with our proposed Greedy algorithm, because computing the similarities or testing the Visibility Constraint alone will cost $O(n^2)$ time in the worst case. To tackle this problem, our idea is to sample a small set of objects O', such that the characteristics of O' are similar to those of O. Ideally, if we apply our Greedy Selection algorithm to objects O', the selection result can represent O as well, while satisfying the Visibility Constraint. We denote this algorithm by SaSS and it is shown in Algorithm 2. The challenge is how to determine a proper size of O'. We will address this in the next subsection, and we will show that with a high probability the result of SaSS provides theoretical guarantees on the error bound of the representative score.

Algorithm 2: Sampling for Spatial Object Selection(SaSS)

```
input : A set of geospatial objects $O$, size $k$, distance threshold $\theta$,
        confidence $\delta$, error tolerance $\epsilon$
output: A subset of objects $S$
1 $m = \left\lceil \frac{1}{\frac{2\epsilon^2}{\ln\frac{2}{\delta}} + \frac{1}{|O|}} \right\rceil$;
2 Draw $m$ samples $O'$ from $O$ randomly;
3 $S \leftarrow$Greedy($O'$, $k$, $\theta$) ;                // invoking Algorithm 1
4 return $S$;
```

Note that simply sampling k objects as the result of the sos problem is not desirable due to the following reasons. The random sampling does not take into account representativeness and visibility constraints. Even if the visibility constraint can be easily enforced in the random sampling method as we do in experiments (Section 7), the sampling results are not representative as shown in our results. We show in Section 7 that such a random selection strategy results in a poor representative score.

6.2 Determining Sampling Size

Before the proof of theorem, we first introduce some useful concentration inequalities that will be utilized for the proof.

Lemma 6.1. *(Hoeffding's Inequality) [26] Given a set of n random variables $\mathcal{X} = x_1, x_2, \cdots, x_n$ in $[0,1]$ with a mean μ, for any $\epsilon > 0$, we have*

$$\mathbb{P}\left[|\frac{1}{n}\sum_{i=1}^{n} x_i - \mu| \geq \epsilon \right] \leq 2exp(-2\epsilon^2 n). \tag{4}$$

Lemma 6.1 provides a general bound for infinite population, i.e., the number of variables n. In most specific cases, for a finite population there exists a tighter bound:

Lemma 6.2. *(Serfling's Inequality) [44] Given a set of n random variables $\mathcal{X} = x_1, x_2, \cdots, x_n$ in $[0,1]$ with a mean μ, for any $\epsilon > 0$ and $1 \leq k < n$, we have*

$$\mathbb{P}\left[\max_{k\leq m\leq n-1} |\frac{1}{m}\sum_{i=1}^{m} x_i - \mu| \geq \epsilon \right] \leq 2exp(-\frac{2k\epsilon^2}{1-\frac{k-1}{n}}). \tag{5}$$

Note that both Lemmas 6.1 and 6.2 can provide a confidence δ. When the error bound ϵ is large, the confidences are close. In other cases, Equation 5 provides a smaller size for sampling. For simplicity, we use Equation 4 for the proof, and similar results can be derived by replacing it with Equation 5.

Note that the Greedy method in Algorithm 2 can be replaced by any other methods for solving sos problem and we have the following Theorem:

Theorem 6.3. *With probability at least $1-\delta$, for any approach $\mathcal{F}$ used to solve the sos problem our SaSS returns an $(1-\epsilon)$-approximate solution w.r.t. the solution computed by $\mathcal{F}$.*

Proof. The sos problem aims to maximize $\frac{1}{|O|}\sum_{o\in O} o.\omega \times Sim(o,S)$. Since $o.\omega \in [0,1]$ and $Sim(o,S) \in [0,1]$, we can take $o.\omega \times Sim(o,S)$ as a random variable in $[0,1]$ w.r.t. object o. We use OPT to denote the optimal representative score $Sim(O,S)$ for a given set of objects O. According to the law of large numbers [18], when O is large, we can simply assume that $OPT = \frac{1}{|O|}\sum_{o\in O} o.\omega \times Sim(o,S) = \mu$. Let $x_i = o_i.\omega \times Sim(o_i,S)$, and we take Equation 2 into Equation 4: $\mathbb{P}[|Sim(O',S) - OPT| \geq \epsilon] \leq \delta = 2exp(-2\epsilon^2|O'|)$. So we have: $\mathbb{P}[|Sim(O',S) - OPT| < \epsilon] \geq 1-\delta$, and

$$|O'| = min\begin{cases} \lceil \frac{1}{2\epsilon^2}\ln\frac{2}{\delta} \rceil \\ |O| \end{cases} \tag{6}$$

□

Similarly, we can obtain the size of objects to be sampled using Eq 5, and we have:

$$|O'| = \left\lceil \frac{1}{\frac{2\epsilon^2}{\ln\frac{2}{\delta}} + \frac{1}{|O|}} \right\rceil, \tag{7}$$

when $|O| \to \infty$ we can see that it is identical to Eq 6.

7 EXPERIMENTS

We conduct extensive experimental evaluation on the efficiency and effectiveness of our solutions to the Spatial Object Selection (sos) problem and the Interactive Spatial Object Selection (isos) problem. We first introduce the experimental setup, and then report the efficiency of our algorithms and the representative scores, which reflect the effectiveness of the proposed solutions.

7.1 Experimental Setup

Datasets. We conduct the experiments on two types of real-life geospatial datasets:

1) **Twitter** is a geo-tagged tweet dataset crawled using Twitter API[1]. We use tweets posted by users in United Kingdom (UK) and United States (US), denoted by **UK** and **US**, which contain up to 2 million and 200 million geo-tagged tweets, respectively. Unless specified otherwise we extract 1 million tweets of **UK** and 100 million tweets of **US** for all experiments except for scalability evaluation (in Sec. 7.3.4).

2) **POI** is a Point-of-Interest dataset crawled using Foursquare API[2]. Each POI is associated with textual description. It contains 322,006 POIs in Singapore (SG).

For each geospatial object, we randomly set the weight ω in $[0,1]$.

Algorithms. For the sos problem, as mentioned in Sections 1 and 2, this is the first work that takes both representative constraint and visibility constraint into consideration in k-size object selection and accepts a general definition of the representative score between selected objects and the whole objects (within a particular region). Therefore, there exists no previous work for direct comparison.

[1]https://dev.twitter.com/rest/public
[2]https://developer.foursquare.com/

Table 2: Parameters

Parameter	Range
Query Region Size ($*10^{-2}$)	2^{-2}, 2^{-1}, $\mathbf{2^0}$, 2^1, 2^2
Number of Selected Objects k	60, 80, **100**, 120, 140
Distance threshold θ ($*10^{-3}$)	1, 2, **3**, 4, 5
Zoom-in Size R_{in}/R (by length)	2^{-3}, $2^{-2.5}$, 2^{-2}, $2^{-1.5}$, $\mathbf{2^{-1}}$
Zoom-out Size R_{out}/R (by length)	$\mathbf{2^1}$, $2^{1.5}$, 2^2, $2^{2.5}$, 2^3
Upscale of data size for scalability test	**1**, 1.25, 1.5, 1.75, 2
Relative Error Bound ϵ ($*10^{-2}$)	3, 4, **5**, 6,7
Confidence Error δ	0.08, 0.09, **0.1**, 0.11, 0.12

We consider a baseline method Random to compare with our Greedy method (proposed in Sec. 4.1). Random is a uniform random selection strategy used in [48, 49], and it is implemented in main memory. To meet the visibility constraint, we repeatedly pick a random object o if adding o into the current result does not break the visibility constraint. Once there are k objects, they are returned as result. We also compare with four other baselines, which are introduced in Section 7.2. MAXMIN, MAXSUM [17] and DisC [16] are methods for selecting the most diverse objects, and K-means is used for clustering. Note that the results of these four methods may not fulfill the visibility constraint.

Performance Measurement. For all the methods, we use R-tree as the spatial index for region queries, which returns all the objects inside a given query region. We evaluate the performance of all methods by their runtime, and we report the runtime after the object fetching is finished. Specifically, for SASS, we also study how many objects are sampled, i.e., $|O'|/|O|$. Given a sampling selection result S, we can compute its representative score on the whole dataset O. We report the score differences, i.e., $|Score(O, S) - Score(O', S)|$. This result is to show how well the result on the sampled objects can represent the whole dataset. Each experiment is repeated 50 times, and the average performance is reported.

Similarity Metric. Note that our approach is able to deal with any similarity metric depending on the richness of data in particular application scenarios. For the Twitter and POI datasets, each object is associated with some keywords. We measure the similarity of two objects with Cosine Similarity of the keyword vectors.

Query Generation. For the SOS problem, we randomly pick an object from the dataset and generate a square-shape query region R centered at this object. For the ISOS problem, each interactive operation will result in a new region that the user may further explore. For the Zoom-in operation, we randomly locate a new square-shape query region R_{in} that is completely inside the previous region R. For the Zoom-out operation, we randomly locate a new square-shape query region R_{out} that completely covers the previous region R.

Parameters. Table 2 shows the detailed settings of all parameters, where the default one is highlighted in bold. By default, we set the query region R as 0.01 of the size of the whole dataset, which usually represents a suburb. It is a reasonable setting as most users explore a suburb at a time. In average, there are about 0.5% objects of the whole data in a query region of size 0.01, and up to about 15% objects in some dense regions. In each query region, we select k=100 objects by default out of all the objects. For the visibility constraint, we set a distance threshold θ as 0.003 of the size of the query region by length. We set the relative error bound ϵ as 0.05 and the confidence error δ as 0.1 for SASS. We set R_{in} as half of that of R by length, and set R_{out} as two times of R by length. For the panning operation, we randomly locate the new query region R_{pan} that intersects with R, and we assume they are of the same size. We study the impact of overlap ratio of R_{pan} and R on the runtime in Section 7.4.1.

Setup. All algorithms are implemented in C++ complied with GCC 4.8.2 and run on Linux with a 2.66GHz CPU and 64GB RAM.

Table 3: User Study Result for SOS

Method	Greedy	Random	MaxMin	MaxSum	DisC	K-means
RP Score	0.95	0.89	0.86	0.56	0.78	0.87
User Vote	4.9	3.6	1.6	1.0	2.1	3.0

7.2 A User Study on the Representative Score

To demonstrate the usefulness of the representative score function, we compare with five other selection algorithms, which represent the best baselines we can come up with, and ask 15 students to rate the representative quality. We use Euclidean distance as the similarity metric, which is straightforward to be judged by users, and we assume each object has the same weight. Note that by using Euclidean distance as the similarity metric, our objective function reduces to a statistics criterion called Weighted Mean of the Shortest Distances (WMSD) [15, 43, 47], which was introduced by Van Groenigen et al. [46]. Figure 6 shows the selection results on the Twitter dataset in UK, where each method selects 30 objects out of 500. Figure 6(a) shows the original distribution of all the objects, and the selection result by our proposed Greedy is shown in Figure 6(b). Figure 6(c) shows a set of randomly selected objects. Figures 6(d) and 6(e) show the result of k-DIVERSITY problem[17] denoted by *MAXMIN* and *MAXSUM*, and their objective functions are defined as $f_{MIN}(S) = \max_{\substack{S\subseteq O\\|S|=k}} \min_{\substack{o_i,o_j\in S\\o_i\neq o_j}}(1-Sim(o_i,o_j))$, and $f_{SUM}(S) = \max_{\substack{S\subseteq O\\|S|=k}} \sum_{\substack{o_i,o_j\in S\\o_i\neq o_j.}}(1-Sim(o_i,o_j))$. Figure 6(f) shows the result of DisC [16] method. Since it does not specify the number of selected objects k, we tune the parameter radius r carefully until the size of output is close to k. Figure 6(g) shows the result of k-means clustering, where for each cluster we select the object which is the closest to the cluster centroid. Note that *MAXMIN*, *MAXSUM*, DisC and k-means may not fulfill the visibility constraint, and thus in this user study we focus on the representative constraint while ignoring the visibility constraint. Each student is asked to compare the 500 tweets and the selected tweets by each method in term of tweets' content coverage, and gives a score between 1 and 5 for the representative quality, i.e., how well the selected objects represent the 500 tweets, where 1 is the worst and 5 is the best.

We find that the results of *MaxMin* and DisC methods are evenly distributed in space, and the information of original data distribution is lost. The reason is that these two methods aim to cover all the objects by circles with radius r, and each object of the original data (blue pin) is close to the selected data (red pin) for at most r distance.

The average rating result is shown in Table 3. We can see our method outperforms other methods. We also compute the representative score using Equation 2 for the result of each method. We find that the users' rating is consistent with the Representative Score

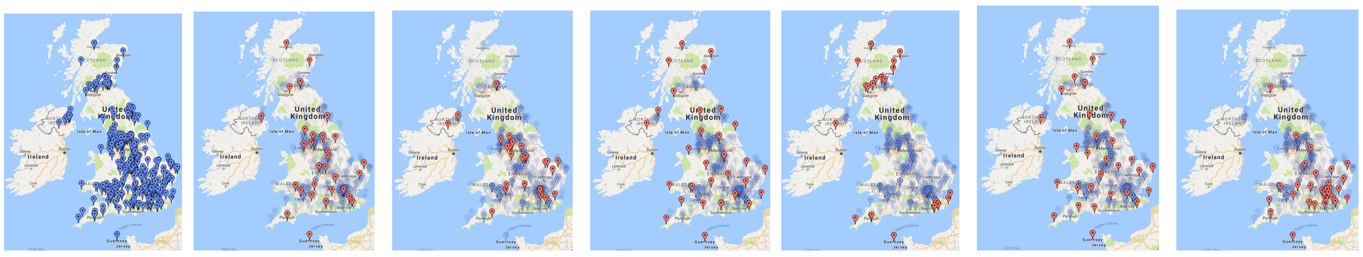

(a) All the objects (b) Our method (c) Random (d) MaxMin (e) MaxSum (f) DisC (g) K-means

Figure 6: Selection Result by Different Methods

Table 4: User Study Result for ISOS

Method	Greedy	Random	MaxMin	MaxSum	DisC	K-means
Z-in	0.93	0.85	0.72	0.46	0.81	0.83
User Vote	4.8	3.5	1.4	1.0	2.2	3.2
Z-out	0.91	0.83	0.76	0.53	0.78	0.81
User Vote	4.7	3.2	1.5	1.2	1.9	2.5
Panning	0.94	0.87	0.78	0.51	0.81	0.80
User Vote	4.8	3.7	1.6	1.1	2.5	3.1

computed by Equation 2, which indicates the general usefulness of our quantification on the representative constraint.

We further study the users' experience on the interactive exploration, i.e., the users' satisfaction after zooming in, zooming out and panning operations. We still compare the algorithms for selecting 30 objects out of 500. However, to support zooming out and panning navigations we shrink the window size by half (compared to the previous study). The students are asked to evaluate the representative quality after each of the three operations. The rating result is shown in Table 4. We observe that for the ISOS problem, the user's rating is also consistent with the Representative Score.

7.3 Evaluation of Solutions to SOS Problem

7.3.1 Comparing Different Methods. In this experiment, we use default settings for all parameters to compare the performance of our Greedy and SASS algorithms with the baseline methods in solving the problem.

Figures 7 and 8 show the runtime and representative score on two smaller datasets, UK and POI. Note that the left y-axis of Figure 8 uses logarithmic scale. We notice that the runtime of our proposed algorithm Greedy is nearly the same with that of Random and is about 2/3 of that of K-means, while Greedy achieves a much better representative score than all the other baselines. We also observe that SASS is a bit faster than Greedy, and much faster than all the other baselines, since it works on a small set of sampled objects, while the score is close to Greedy and better than the others.

The results of K-means, DisC, MAXMIN and MAXSUM do not fulfill the visibility constraint, and SASS is the sampling extension of Greedy and is not significantly better than Greedy on small datasets. Therefore for the rest experiments on the two small datasets, we only show the results of Greedy and Random on UK and POI. We evaluate SASS on US, the larger dataset, on which other methods including Greedy are very slow.

7.3.2 Varying Sampling Parameters. This experiment is to study how the sampling parameters δ and ϵ affect the performance. The experiment is conducted on the large dataset US, and the results are shown in Figures 9 and 10. We only show the results for SASS and Random since all the other methods are much more slower,

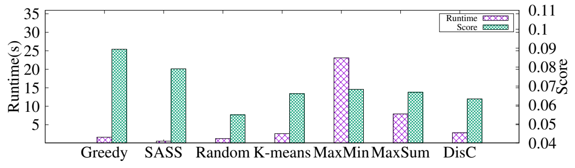


Figure 7: Comparing Different Methods on UK

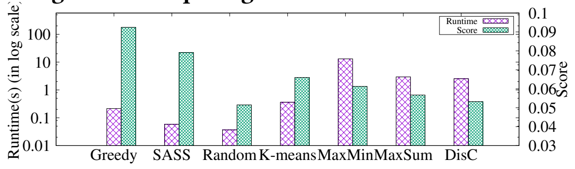
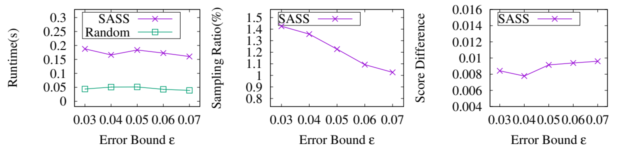


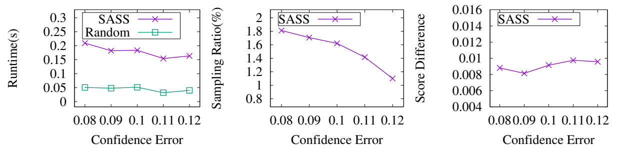
Figure 8: Comparing Different Methods on POI

among which the fastest algorithm, Greedy, is slower than SASS by an order of magnitude. We observe that for both errors the ratio of sampled objects increases when the errors increase, and it is shown from Figure 9(b) that at most 2% of objects suffice to approximately solve the SOS problem. We also observe from Figures 9(a) and 10(a) that the runtime decreases, because the number of candidate objects decreases. Last, the relative score differences are always less than 0.01 when we vary θ or ϵ, which indicates that using sampled data only slightly lose the representativeness.



(a) Runtime (b) Sampling Ratio (c) Score Error

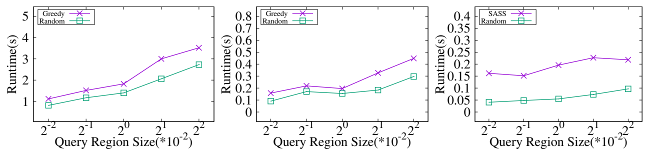
Figure 9: Varying ϵ on US



(a) Runtime (b) Sampling Ratio (c) Score Error

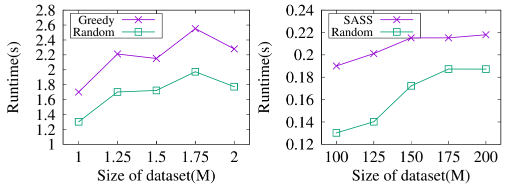
Figure 10: Varying δ on US

7.3.3 Varying the Query Region Size. This experiment is to study how the query region size affects the performance of the proposed approach. The default region 2^0 indicates a region of 1% size of a country (US, UK) by length, which is roughly a city. From the result shown in Figure 11 we find the runtime of the proposed approach increases almost linearly when the query region size increases. This is because for a larger query region size, more objects are considered as the input of our algorithms, leading to more runtime. We do not show the runtime of Greedy on US as it is 3 orders of magnitude slower than SaSS. It is because SASS is based on sampled objects, while Greedy works on all objects.



(a) UK (b) POI (c) US

Figure 11: Varying query region size



(a) UK (b) US

Figure 12: Varying scalability on Twitter

7.3.4 Scalability Test. Last, we study the scalability of our proposed algorithms by varying the data size. We vary UK dataset from 1 million to 2 million and US from 100 million to 200 million. It is observed from Figure 12(a) that the runtime of Greedy generally increases w.r.t. the size of the dataset. Since the tweets are from one country, when the number of objects increases, the density of the objects increases in general. Since we fix the query region size, a large dataset usually induces more objects as input and thus the runtime increases. However, we observe from Figure 12(b) that the runtime of SASS only changes slightly as we vary the number of objects. It is because SASS is based on a certain number of sampled objects. When parameters δ and ϵ are fixed, varying the size of dataset from $100M$ to $200M$ does not change the number of sampled objects significantly.

Other experiments of SOS problem are reported in Appendix E.

7.4 Evaluation of solutions to ISOS Problem

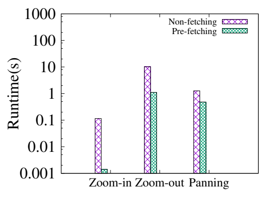
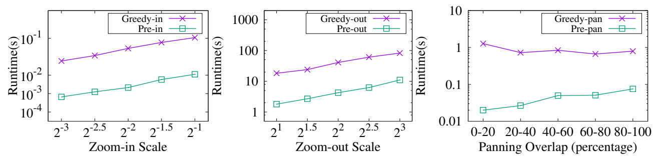


Figure 13: Pre-fetching vs. non-fetching on UK

This experiment is to study the performance of our solution to solve the Interactive Spatial Object Selection (ISOS) problem. In particular, we need to consider the response time for three interactive operations: Zoom-in, Zoom-out and Panning. We denote our greedy algorithms for each of the three operations by Greedy-in, Greedy-out and Greedy-pan, and they are compared with the algorithms using pre-fetching (proposed in Sec. 5), which are denoted by Pre-in, Pre-out and Pre-pan respectively.

First of all, we record the runtime for Pre-in, Pre-out and Pre-pan using default settings for all parameters, and the result is shown in Figure 13. We find that the pre-fetching technique improves the efficiency of Greedy-in, Greedy-out and Greedy-pan by almost 2, 1 and 1 order of magnitude respectively, which is very significant.

In the remaining experiments, we deploy the sampling enhanced greedy algorithm, namely SASS on the large dataset US for the three operations, and they are denoted by SASS-in, SASS-out and SASS-pan.



(a) Zoom-in (b) Zoom-out (c) Panning

Figure 14: Varying zooming scale and panning overlap on UK

7.4.1 Zooming Scale and Panning Overlap Study. Figure 14(a) presents the performance result w.r.t. a varying zoom-in scale from 2^{-3} to 2^{-1}. We find that the runtime of Greedy-in scales linearly while Pre-in scales sub-linearly; also the pre-fetching method helps improving the performance of greedy method by almost 2 orders of magnitude. The runtime is less than 10ms for all cases, providing a seamless user experience in interactive exploration. A similar observation is made for the zoom-out case (Figure 14(b)). Last, we study how the overlap rate between two regions before and after the panning operation affects the runtime of our method. As shown in Figure 14(c), we have the following observations: (1) when the overlap rate is small, the pre-fetching method can improve the performance by 2 orders of magnitude; (2) when the rate increases to [80%-100%], the significance of pre-fetching reduces accordingly.

Other experiments of ISOS problem are reported in Appendix F.

8 CONCLUSIONS

We have developed an interactive visualized exploration system for geospatial data, which took representativeness, visibility, zooming consistency, and panning consistency into consideration. We first proposed the SOS problem to select top-k representative objects from the current region of interest, and any two selected objects should not be too close to each other for users to distinguish in the limited space of a screen. We proved that it is an NP-hard problem, and developed an approximation algorithm with performance guarantees. To provide a seamless experience for users to interactively explore the data when navigating the map, we formally defined the ISOS problem, and proposed a pre-fetching solution on top of our approximation algorithm to improve the efficiency by almost 2 orders of magnitude. We further enhanced our efficiency for large datasets by a sampling technique with theoretical guarantee. Experiments demonstrated the efficiency and scalability of our approach.

ACKNOWLEDGMENTS

This work was supported by the MOE Tier-2 grant MOE2016-T2-1-137, MOE Tier-1 grant RG31/17 and a grant from Microsoft. This work was also supported by the ARC DP180102050, ARC DP170102726, NSFC 61728204, and NSFC 91646204. Zhifeng Bao is a recipient of Google Faculty Research Award.

REFERENCES

[1] S. Acharya, P. B. Gibbons, V. Poosala, and S. Ramaswamy. The aqua approximate query answering system. In *ACM Sigmod Record*, volume 28, pages 574–576. ACM, 1999.

[2] S. Agarwal, B. Mozafari, A. Panda, H. Milner, S. Madden, and I. Stoica. Blinkdb: queries with bounded errors and bounded response times on very large data. In *Proceedings of the 8th ACM European Conference on Computer Systems*, pages 29–42. ACM, 2013.

[3] A. Angel and N. Koudas. Efficient diversity-aware search. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 781–792. ACM, 2011.

[4] L. G. Azevedo, G. Zimbrão, and J. M. De Souza. Approximate query processing in spatial databases using raster signatures. In *Advances in Geoinformatics*, pages 69–86. Springer, 2007.

[5] L. Battle, R. Chang, and M. Stonebraker. Dynamic prefetching of data tiles for interactive visualization. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 1363–1375, 2016.

[6] L. Battle, M. Stonebraker, and R. Chang. Dynamic reduction of query result sets for interactive visualizaton. In *Big Data, 2013 IEEE International Conference on*, pages 1–8. IEEE, 2013.

[7] A. Belussi, B. Catania, and S. Migliorini. Approximate queries for spatial data., 2013.

[8] J. Christensen, J. Marks, and S. Shieber. An empirical study of algorithms for point-feature label placement. *ACM Trans. Graph.*, 14(3):203–232, July 1995.

[9] W. G. Cochran. Relative accuracy of systematic and stratified random samples for a certain class of populations. *The Annals of Mathematical Statistics*, pages 164–177, 1946.

[10] W. G. Cochran. *Sampling techniques*. John Wiley & Sons, 2007.

[11] T. Condie, N. Conway, P. Alvaro, J. M. Hellerstein, K. Elmeleegy, and R. Sears. Mapreduce online. In *Nsdi*, volume 10, page 20, 2010.

[12] F. Csillag, M. Kertész, and Á. Kummert. Sampling and mapping of heterogeneous surfaces: multi-resolution tiling adjusted to spatial variability. *International Journal of Geographical Information Systems*, 10(7):851–875, 1996.

[13] A. Das, J. Gehrke, and M. Riedewald. Approximation techniques for spatial data. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 695–706. ACM, 2004.

[14] A. Das Sarma, H. Lee, H. Gonzalez, J. Madhavan, and A. Halevy. Efficient spatial sampling of large geographical tables. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 193–204. ACM, 2012.

[15] E. M. Delmelle. Spatial sampling. In *Handbook of Regional Science*, pages 1385–1399. Springer, 2014.

[16] M. Drosou and E. Pitoura. Disc diversity: result diversification based on dissimilarity and coverage. *Proceedings of the VLDB Endowment*, 6(1):13–24, 2012.

[17] M. Drosou and E. Pitoura. Diverse set selection over dynamic data. *IEEE Transactions on Knowledge and Data Engineering*, 26(5):1102–1116, 2014.

[18] W. Feller. *An introduction to probability theory and its applications*, volume 2. John Wiley & Sons, 2008.

[19] A. U. Frank and S. Timpf. Multiple representations for cartographic objects in a multi-scale treeâĂŤan intelligent graphical zoom. *Computers & Graphics*, 18(6):823–829, 1994.

[20] P. Fraternali, D. Martinenghi, and M. Tagliasacchi. Top-k bounded diversification. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 421–432. ACM, 2012.

[21] A. Galakatos, A. Crotty, E. Zgraggen, C. Binnig, and T. Kraska. Revisiting reuse for approximate query processing. *Proceedings of the VLDB Endowment*, 10(10):1142–1153, 2017.

[22] M. F. Goodchild and R. P. Haining. Gis and spatial data analysis: Converging perspectives. *Papers in Regional Science*, 83(1):363–385, 2004.

[23] J. L. Green and J. B. Plotkin. A statistical theory for sampling species abundances. *Ecology letters*, 10(11):1037–1045, 2007.

[24] J. M. Hellerstein, R. Avnur, A. Chou, C. Hidber, C. Olston, V. Raman, T. Roth, and P. J. Haas. Interactive data analysis: The control project. *Computer*, 32(8):51–59, 1999.

[25] J. M. Hellerstein, P. J. Haas, and H. J. Wang. Online aggregation. In *ACM SIGMOD Record*, volume 26, pages 171–182. ACM, 1997.

[26] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[27] D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685, 1952.

[28] S. Idreos, O. Papaemmanouil, and S. Chaudhuri. Overview of data exploration techniques. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 277–281. ACM, 2015.

[29] C. Jermaine, S. Arumugam, A. Pol, and A. Dobra. Scalable approximate query processing with the dbo engine. *ACM Transactions on Database Systems (TODS)*, 33(4):23, 2008.

[30] N. Kamat, P. Jayachandran, K. Tunga, and A. Nandi. Distributed and interactive cube exploration. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 472–483. IEEE, 2014.

[31] P. K. Kefaloukos, M. Vaz Salles, and M. Zachariasen. Declarative cartography: In-database map generalization of geospatial datasets. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 1024–1035. IEEE, 2014.

[32] A. Kim, E. Blais, A. Parameswaran, P. Indyk, S. Madden, and R. Rubinfeld. Rapid sampling for visualizations with ordering guarantees. *Proceedings of the VLDB Endowment*, 8(5):521–532, 2015.

[33] M. Mahdian, O. Schrijvers, and S. Vassilvitskii. Algorithmic cartography: Placing points of interest and ads on maps. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 755–764. ACM, 2015.

[34] B. Matérn. *Spatial variation*, volume 36. Springer Science & Business Media, 2013.

[35] A. Milne. The centric systematic area-sample treated as a random sample. *Biometrics*, 15(2):270–297, 1959.

[36] J. Neyman. On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society*, 97(4):558–625, 1934.

[37] J. Neyman. Contribution to the theory of sampling human populations. *Journal of the American Statistical Association*, 33(201):101–116, 1938.

[38] S. Nutanong, M. D. Adelfio, and H. Samet. Multiresolution select-distinct queries on large geographic point sets. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 159–168. ACM, 2012.

[39] S. Peng, H. Samet, and M. D. Adelfio. Viewing streaming spatially-referenced data at interactive rates. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 409–412. ACM, 2014.

[40] E. Puppo and G. Dettori. Towards a formal model for multiresolution spatial maps. In *International Symposium on Spatial Databases*, pages 152–169. Springer, 1995.

[41] L. Qin, J. X. Yu, and L. Chang. Diversifying top-k results. *Proceedings of the VLDB Endowment*, 5(11):1124–1135, 2012.

[42] R. Royall. An old approach to finite population sampling theory. *Journal of the American Statistical Association*, 63(324):1269–1279, 1968.

[43] E. Scudiero, R. Deiana, P. Teatini, G. Cassiani, and F. Morari. Constrained optimization of spatial sampling in salt contaminated coastal farmland using emi and continuous simulated annealing. *Procedia Environmental Sciences*, 7:234–239, 2011.

[44] R. J. Serfling. Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, pages 39–48, 1974.

[45] K. S. Shea and R. B. McMaster. Cartographic generalization in a digital environment: When and how to generalize. In *Proceedings of AutoCarto*, volume 9, pages 56–67, 1989.

[46] J. W. Van Groenigen. The influence of variogram parameters on optimal sampling schemes for mapping by kriging. *Geoderma*, 97(3):223–236, 2000.

[47] J.-F. Wang, A. Stein, B.-B. Gao, and Y. Ge. A review of spatial sampling. *Spatial Statistics*, 2:1–14, 2012.

[48] L. Wang, R. Christensen, F. Li, and K. Yi. Spatial online sampling and aggregation. *Proceedings of the VLDB Endowment*, 9(3):84–95, 2015.

[49] P. Wang, W. He, and X. Liu. An efficient sampling method for characterizing points of interests on maps. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 1012–1023. IEEE, 2014.

[50] J. M. Ware, C. B. Jones, and N. Thomas. Automated map generalization with multiple operators: a simulated annealing approach. *International Journal of Geographical Information Science*, 17(8):743–769, 2003.

[51] K. Zeng, S. Agarwal, A. Dave, M. Armbrust, and I. Stoica. G-ola: Generalized on-line aggregation for interactive analysis on big data. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 913–918. ACM, 2015.

A PROOF OF LEMMA 4.1

Lemma A.1. *(Submodularity.) Let S and T be two sets of geospatial objects, and $S \subseteq T$. Let v be a newly inserted object. We have $Sim(O, S \cup \{v\}) - Sim(O, S) \geq Sim(O, T \cup \{v\}) - Sim(O, T)$.*

Proof. We first prove that

$$Sim(o, S \cup \{v\}) - Sim(o, S) \geq Sim(o, T \cup \{v\}) - Sim(o, T).$$

We consider the following cases:

Case 1: $Sim(v, o) \leq Sim(o, S)$, $Sim(v, o) \leq Sim(o, T)$. Then $Sim(o, S \cup \{v\}) = Sim(o, S)$ and $Sim(o, T \cup \{v\}) = Sim(o, T)$.

Case 2: $Sim(v, o) > Sim(o, S)$, $Sim(v, o) \leq Sim(o, T)$. Then $Sim(o, T \cup \{v\}) - Sim(o, T) = 0$ and $Sim(o, S \cup \{v\}) - Sim(o, S) > 0$.

Case 3: $Sim(v, o) > Sim(o, S)$, $Sim(v, o) > Sim(o, T)$. In this case, $Sim(o, S \cup \{v\}) = Sim(o, v)$ and $Sim(o, T \cup \{v\}) = Sim(o, v)$. Since $Sim(O, S) \leq Sim(O, T)$, we have $Sim(o, v) - Sim(O, S) \geq Sim(o, v) - Sim(O, T)$.

Since $Sim(O, S) = \frac{1}{|O|} \sum_{o \in O} o.\omega \times Sim(o, S)$, the lemma is proved. □

B PROOF OF LEMMA 4.4

Lemma B.1. *Let S be a set of geospatial objects that satisfy the visibility constraint. Let o be a geospatial object and $o \notin S$. At most 7 objects in S conflict with o.*

Proof. If an object $v \in S$ conflicts with o, then v must be inside the circle with a radius θ centered at o. Since S satisfies the visibility constraint, any two objects in S do not conflict with each other, i.e., $dist(v_1, v_2) > \theta$ for $v_1, v_2 \in S$. Thus, the circle with a radius of θ centered at o can cover at most 7 objects from S, which is illustrated in Figure 15. Note that o is at the position of o_1. □

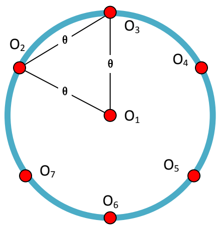


Figure 15: Conflict Objects Example

C PROOF OF THEOREM 4.5

Theorem C.1. *The Greedy algorithm has an approximation ratio of* 1/8.

Proof. Let $S^* = [v_1^*, \cdots, v_k^*]$ be the optimal set of k objects with maximum score. Let $S = [v_1, \cdots, v_k]$ be the set of k selected by Greedy. We use $\Delta(v|S) = Sim(O, S \cup \{v\}) - Sim(O, S))$ to denote the increase of the score brought by adding v into S. According to Lemma 4.2 and Lemma A.1, we have

$$\begin{aligned} Sim(O, S^*) &\leq Sim(O, S^* \cup S) \\ &= Sim(O, S) + \sum_{j=1}^{k} \Delta(v_j^* | S \cup \{v_1^*, \cdots, v_{j-1}^*\}) \\ &\leq Sim(O, S) + \sum_{v^* \in S^*} \Delta(v^* | S) \end{aligned} \tag{8}$$

Let v^* be an object in S^* and θ be the distance threshold. For an object o, we refer to the circle with a radius of θ centered at o as the circle. We consider the following cases:

Case 1: Object v^* does not fall into the circle of any object in S, i.e., $dist(v^*, v) > \theta$ for any $v \in S$.

When selecting the k-th object in S, v^* also satisfies the Visibility Constraint. Since S is greedily selected, w have

$$Sim(O, S) - Sim(O, S_{k-1}) \geq \Delta(v^* | S_{k-1}),$$

where $S_{k-1} = [v_1, \cdots, v_{k-1}]$.

According to Lemma 4.2, we have $\Delta(v^* | S_{k-1}) \geq \Delta(v^* | S)$. Then we can derive $Sim(O, S) - Sim(O, S_{k-1}) \geq \Delta(v^* | S)$.

Case 2: Object v^* does not fall into the circles of objects $v_1, \cdots, v_i$, but falls into the circle of v_{i+1}, i.e., $dist(v^*, v_j) > \theta$ for $j \in [1, i]$, and $dist(v^*, v_{i+1}) \leq \theta$.

When selecting the object v_{i+1}, v^* satisfies the visibility constraint. Since S_{i+1} is greedily selected, we have

$$Sim(O, S_{i+1}) - Sim(O, S_i) \geq \Delta(v^* | S_i)$$

According to Lemma 4.2, we have $\Delta(v^* | S_{k-1}) \geq \Delta(v^* | S)$. Then we can derive $Sim(O, S_{i+1}) - Sim(O, S_i) \geq \Delta(v^* | S)$.

From the two cases, we can see that for each object $v^* \in S^*$, we can find an $v_i \in S$ such that v^* and v_i are conflicted and $Sim(O, S_{i+1}) - Sim(O, S_i) \geq \Delta(v^* | S)$. Since S^* satisfies the visibility constraint, according to Lemma B.1, we have

$$\begin{aligned} \Delta(v^* | S_i) &\leq 7(Sim(O, S_1) - Sim(O, S_0) + \cdots \\ &\quad + Sim(O, S) - Sim(O, S_{k-1})) \\ &= 7Sim(O, S) \end{aligned}$$

From Equation 8, we have

$$Sim(O, S^*) \leq Sim(O, S) + 7Sim(O, S) = 8Sim(O, S) \tag{9}$$

Thus, the Greedy has an approximation ratio of 1/8. □

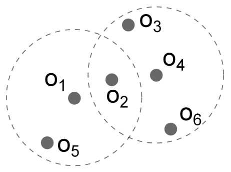
D EXAMPLE OF GREEDY ALGORITHM

Example D.1. Consider the example shown in Figure 16. The collection of geospatial data comprise six geospatial objects o_1, o_2, o_3, o_4, o_5 and o_6. The similarity between any two objects is given in the table on the right side. We assume that every object has a weight of 1. We want to select a set of two geospatial objects to be visualized.

The algorithm first initializes the heap by computing the marginal increase of each object. For example, the marginal similarity increase of adding object o_1 to $S = \emptyset$ is

$$\Delta(o_1, S) = (1 + 0.9 + 0.2 + 0.5 + 0 + 0) - 0 = 2.6$$

The status of the heap after the initialization is shown in Figure 17(a).



	o_1	o_2	o_3	o_4	o_5	o_6
o_1	1	0.9	0.2	0.5	0	0
o_2	0.9	1	0.2	0.1	0	0
o_3	0.2	0.2	1	0.9	0	0
o_4	0.5	0.1	0.9	1	0	0
o_5	0	0	0	0	1	0
o_6	0	0	0	0	0	1

Similarity Score

Figure 16: Example for the greedy algorithm.

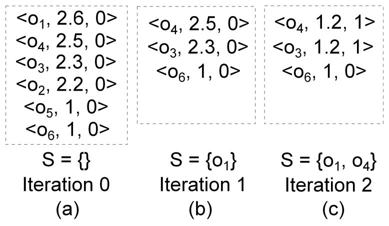


Figure 17: Status of the max-heap.

In the first iteration, the top tuple in the heap is $t = \langle o_1, 2.6, 0\rangle$. since $t.Iter = 0$, the marginal increase of o_1 is computed in this iteration. Therefore we select o_1 into the representative set S. Moreover, we need to remove all objects that conflict with o_1 from the heap due to the visibility constraint. We use a dashed circle to denote the region in which the objects conflict with o_1. Since $dist(o_1, o_2) < \theta$, $dist(o_1, o_5) < \theta$, o_2 and o_5 are removed from the heap. There are only three tuples left in the heap. The status of the heap is shown in Figure 17(b).

In the second iteration, the top tuple in heap is $t = \langle o_4, 2.5, 0\rangle$, and its marginal increase is not updated in this iteration. Therefore we need to recompute the marginal increase as follows:

$$\begin{aligned}&\Delta(o_3, S)\\ &= (1 + 0.9 + 1 + 0.9 + 0 + 0) - (1 + 0.9 + 0.2 + 0.5 + 0 + 0)\\ &= 1.2\end{aligned} \tag{10}$$

Similarly, we next recompute the marginal increase of o_4 as

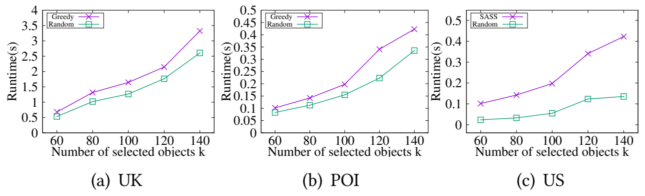
$$\begin{aligned}&\Delta(o_4, S)\\ &= (1 + 0.9 + 1 + 0.9 + 0 + 0) - (1 + 0.9 + 0.2 + 0.5 + 0 + 0)\\ &= 1.2\end{aligned} \tag{11}$$

In this iteration, we have recomputed the marginal increase for two objects, and the status of the heap is shown in Figure 17(c). Note that since the current upper bound for object o_6 is smaller than that of the other two objects, its marginal increase will not be recomputed. With the "lazy forward" strategy, we prune one recomputation in this iteration. Then o_4 is selected into the heap since it has the maximum marginal increase. Now we have successfully selected two objects into the representative set.

E ADDITIONAL EVALUATION OF SOLUTIONS TO SOS PROBLEM

E.1 Varying the Number of Selected Objects k

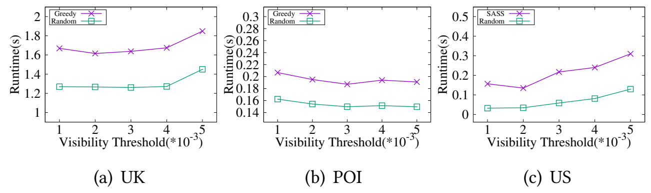
This experiment is to study how the number of selected objects, i.e. k, affects the performance. It can be observed form Figure 18 that, when k increases the runtime of all algorithms increases. It is because more rounds of iteration are needed for a large k.



(a) UK (b) POI (c) US

Figure 18: Varying k

E.2 Varying the Distance Threshold



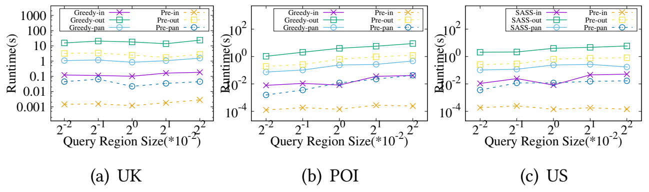
(a) UK (b) POI (c) US

Figure 19: Varying distance threshold θ

Figure 19 shows the effect of distance thresholds θ on the performance. We observe that the runtime of both algorithms stays stable regardless of the choices of distance threshold.

F ADDITIONAL EVALUATION OF SOLUTIONS TO ISOS PROBLEM

F.1 Varying the Query Region Size



(a) UK (b) POI (c) US

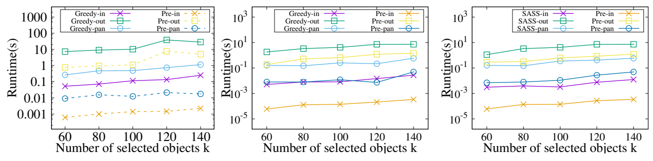
Figure 20: Varying query region size

As we can see from Figure 20, each method's performance keeps stable when the region size increases, and our pre-fetching method significantly reduces the runtime by 3, 1 and 2 orders of magnitude for Zoom-in, Zoom-out and Panning operations, respectively.

F.2 Varying the Number of Selected Objects k

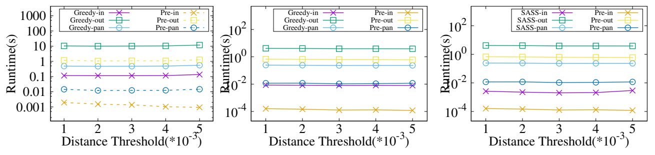
Figure 21 shows that the runtime for each operator increases when k increases. Also, pre-fetching techniques help improve the performance up to 2 orders of magnitude.

F.3 Varying the Visibility Constraint



(a) UK (b) POI (c) US

Figure 21: Varying k

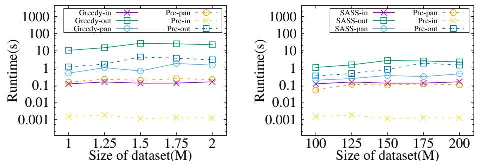


(a) UK (b) POI (c) US

Figure 22: Varying distance threshold θ

Figure 22 shows similar trends with its counterpart in addressing the sos problem.

F.4 Scalability Test



(a) UK (b) US

Figure 23: Varying scalability

Figure 23 shows the runtime while we vary the size of datasets, and we observe similar trends as those in the sos problem.