POIsam: a System for Efficient Selection of Large-scale Geospatial Data on Maps

Tao Guo[†] Mingzhao Li[‡] Peishan Li[‡] Zhifeng Bao[‡] Gao Cong[†]
tguo001@e.ntu.edu.sg,mingzhao.li@rmit.edu.au,peishanl1@student.unimelb.edu.au
zhifeng.bao@rmit.edu.au,gaocong@ntu.edu.sg
[†] SCSE, Nanyang Technological University, Singapore [‡]RMIT University, Australia

ABSTRACT

In this demonstration we present POIsam, a visualization system supporting the following desirable features: representativeness, visibility constraint, zooming consistency, and panning consistency. The first two constraints aim 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. One unique feature of POISam is that any similarity metrics can be plugged into POISam to meet the user's specific needs in different scenarios. The latter two consistencies are fundamental challenges to efficiently update the selection result w.r.t. user's zoom in, zoom out and panning operations when they interact with the map. POISam drops a common assumption from all previous work, i.e. the zoom levels and region cells are pre-defined and indexed, and objects are selected from such region cells at a particular zoom level rather than from user's current region of interest (which in most cases do not correspond to the pre-defined cells). It results in extra challenge as we need to do object selection via online computation. To our best knowledge, this is the first system that is able to meet all the four features to achieve an interactive visualization map exploration system.

ACM Reference format:
Tao Guo[†] Mingzhao Li[‡] Peishan Li[‡] Zhifeng Bao[‡] Gao Cong[†] . 2018. POIsam: a System for Efficient Selection of Large-scale Geospatial Data on Maps. In *Proceedings of 2018 International Conference on Management of Data, Houston, TX, USA, June 10–15, 2018 (SIGMOD'18),* 4 pages.
https://doi.org/10.1145/3183713.3193549

1 INTRODUCTION

Large collections of geospatial data are becoming increasingly available, such as geo-tagged micro-blogs (e.g., Twitter), urban data like real estate properties, etc. Such data are featured with both geospatial content and other content, such as attributes, texts and photos. It is useful to provide support for end-users to perform visualized exploration on such geospatial data on maps.

In this demonstration, we aim to achieve two goals. The first goal is to achieve an efficient spatial object selection, namely the SOS query: 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 illustrate the above two features in the following example.

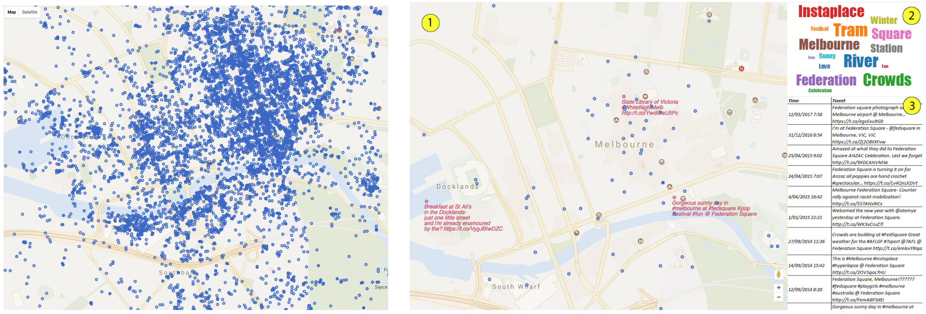
Example 1.1. Given a collection of points of interest (POIs), an end-user would like to browse a small number (denoted by k) of representative POIs for an area on an online map. Ideally, the set of selected POIs can well represent the POIs of the area (i.e., Representativeness Constraint), and they should 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 screen. Figure 1 demonstrates Example 1.1, where a small number of representative POIs are shown to user (Figure 1(b)). Note that without selection, it looks like Figure 1(a). The user may be interested in some POIs, and he may click one to check the detailed information. Moreover, the "hidden" tweets that are represented by the selected tweet are also listed and summarized by a word cloud for users to further explore.

By reviewing the literature in the areas of cartographic selection and spatial sampling, we find that there have been some studies [5, 7, 8] taking the visibility constraint into account. However, none of them considers representativeness except for a study done by Drosou et al. [3], 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 this assumption.

The POIsam system distinguishes from the literature in two-folds. First, it supports various types of data resources, and the users can define their personalized similarity metric to meet different needs. For example in Figure 3(b), when exploring properties there are different attributes associated with each house, and users can choose among them to specify their own preference. Second, POIsam takes the importance of different objects as a factor, such that the results shown on map are highly related to the user's preference. For example, when exploring properties users can choose options like "Distance to nearest shopping center" or "Land size" such that different needs can be met. User can also specify the number of shown results and the minimum overlapping threshold θ such that the objects can be displayed in the best experience.

Our second goal is to further extend the SOS query to achieve an interactive spatial object selection, namely the ISOS query. In particular, the POIsam system is able to answer the SOS query in an efficient manner, in response to user's common map navigation operations, which include zoom-in, zoom-out and panning. As a result, we define the zooming consistency constraint and panning consistency contraint when selecting a new set of representative objects for the new map region.

SIGMOD'18, 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/10.1145/3183713.3193549



Time	Tweet
12/03/2017 7:58	Federation square photograph at Melbourne airport @ Melbourne... https://t.co/egcExultG0
31/12/2016 8:54	I'm at Federation Square - @fedsquare in Melbourne, VIC, VIC https://t.co/Zj2OBtXFvw
25/04/2015 9:02	Amazed at what they did to Federation Square ANZAC Celebration. Lest we forget http://t.co/9XDCANVM3e
24/04/2015 7:07	Federation Square is turning it on for Anzac all poppies are hand crochet #spectacular... https://t.co/Lv4QnLKDVt
4/04/2015 16:42	Melbourne Federation Square- Counter rally against racist mobilization! http://t.co/S57ANsRiCs
1/01/2015 22:21	Welcomed the new year with @atemye yesterday at Federation Square. http://t.co/WK3vCruZ7l
27/09/2014 11:36	Crowds are building at #FedSquare Great weather for the #AFLGF #7sport @7AFL @ Federation Square http://t.co/emlsvYRlqa
14/09/2014 15:42	This is #Melbourne #Instaplace #hyperlapse @ Federation Square http://t.co/2OVSqoc7HU
12/09/2014 8:20	Federation Square, Melbourne!?????? #fedsquare #playgirls #melbourne #australia @ Federation Square http://t.co/FemA8P3dEI
	Gorgeous sunny day in #melbourne at

(a) Wihtout any selection, it is messy to view all the POIs

(b) Only a few representative POIs are shown to the user on the map view (1). After the user clicks on a tweet near *Fedration Square*, a word cloud view (2) summaries all the similar tweets and a detailed tweet view (3) is shown to the user.

Figure 1: Spatial Object Selection Examples (Twitter data)

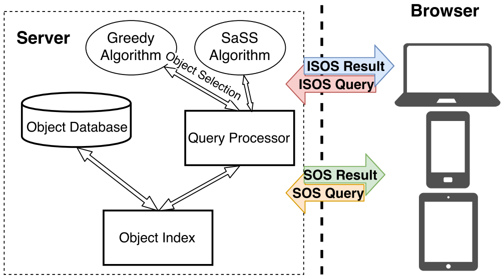


Figure 2: System Architecture

The ISOS query drops a common assumption from all previous work, i.e., the zoom levels and region cells are pre-defined and indexed, and objects are selected from such cells at a particular zoom level rather than from user's current region of interest (which in most cases do not correspond to the pre-defined cells). It poses a challenge in object selection via online computation.

Section 2 formally defines the SOS and ISOS query, whose scenarios are demonstrated in Section 4, where two realworld datasets from Twitter and Australia Real Estate are used. The problem of solving SOS/ISOS queries can be shown to be NP-hard. Hence, we devise two approximation algorithms with provable performance guarantees. The detailed algorithms to support these two goals can be found in our recent research work [4].

2 DATA MODEL AND QUERIES.

Geospatial object. 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. In this demonstration, we consider a large collection of geospatial objects, denoted by O.

2.1 Spatial Object Selection (SOS) Query

Representativeness Constraint. 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 $[0, 1]$. We leave $Sim(.,.)$ as a general function to cope with various types of resources to meet different user and application needs.

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

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

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

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 1 aims to maximize the total utilities of all the selected objects.

Visibility Constraint. Similar to the previous work on query result diversification and cartographic selection (e.g., [2, 3, 7]), 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.

Definition 2.1. **Spatial Object Selection (SOS) Query.** 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 and show a subset of k objects $S \subseteq O$ on map 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.

2.2 Interactive Spatial Object Selection (ISOS) Query

In order to provide a seamless experience for user's exploration on the map, i.e., zooming in, zooming out and panning, it needs to fulfill the *zoom consistency* and the *movement consistency* constraints.

Zooming Consistency Constraint: By zooming in (out), the map is displayed with a finer (coarser) granularity and more (fewer) details in the region are visible to users. 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: Users can move the displayed region to a new place with the same 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.

Definition 2.2. **Interactive Spatial Object Selection (ISOS) Query.** 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 efficiently 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.

3 POISAM PROTOTYPE

The system architecture is shown in Figure 2, consisting of a geospatial data repository, an indexing module, a query processor module and a browser module.

3.1 Indexing Module

When the user explores the map utilizing the browser, the query processor sends a *spatial range query* based on the current window to the object index, which is formalized by a region of rectangle. The indexing module organizes the object database using an R*-tree[1], which can efficiently solve the *spatial range query*. The objects that are in the query range are then sent back to the query processor for the subsequent selection.

3.2 Query Processor Module

Given the query sent from the browser module, the query processor finds a region of objects according to the user's current map window. The processor invokes the selection algorithms [4] to reduce the number of objects, and finally the results are sent back to users and displayed on the map using Google Maps API.

POIsam aims to find the best subset of a given size to represent the whole data collection. Two algorithms are involved in our POIsam prototype:

Greedy Algorithm. We select the representative set S of objects from the whole set O of objects iteratively. In each iteration, we select the geospatial object with the maximum marginal similarity increase by using a heap. 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. To address this challenge, we propose a "lazy-forward" strategy, which recomputes 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. Let n_c be the number of geospatial objects whose marginal increases are re-computed 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.

SaSS Algorithm 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 in this approach 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 prove that with a probability of at least $1-\delta$, our SASS returns a $(1-\epsilon)$-approximate solution if we sample $|O'|$ objects out of O, where $|O'|$ can be bounded by $min(\lceil \frac{1}{2\epsilon^2} \ln \frac{2}{\delta} \rceil, |O|)$.

The details of the two algorithms are introduced in [4].

3.3 Browser Module

The browser module provides interfaces for users to locate query windows and view the selected results. It provides interactions with the map through Google Maps API. Queries are sent from the client browser to the server by standard HTTP post operations.

When issuing an SOS query, users choose a window of map at some specific level as their region of interest, and a number k that indicates the number of objects in the returned result. The query is then sent to the server for processing. When issuing an ISOS query, the procedure is similar while the browser also sends the objects shown in current map to the query processor. After the query is processed, a set of spatial objects are returned and displayed on the map. Users can click the objects to browse the detailed information, and meanwhile the map shows relevant objects that are hidden before to extend user's exploration.

4 DEMONSTRATION AND SCENARIOS

In this section, we will demonstrate POIsam based on two real-world datasets. The first one is Australian's real estate data [6] which contains 1.42 million records of sold properties in Australia. Each property has 36 descriptive attributes (such as price, distance from the property to the nearest shopping center). The second dataset is a geo-tagged tweet data crawled using Twitter API, which comprises 1 million records. Each tweet contains coordinates and the text information posted by the users and other attributes (such as timestamp and user's id).

In our demonstration, (1) we will illustrate the SOS query with the multidimensional real estate data and show how our SOS query considers both the representativeness constraint and the visibility constraint; (2) we will illustrate the ISOS query with the twitter data, and show how new tweets are generated after user interactions such as zooming and panning.

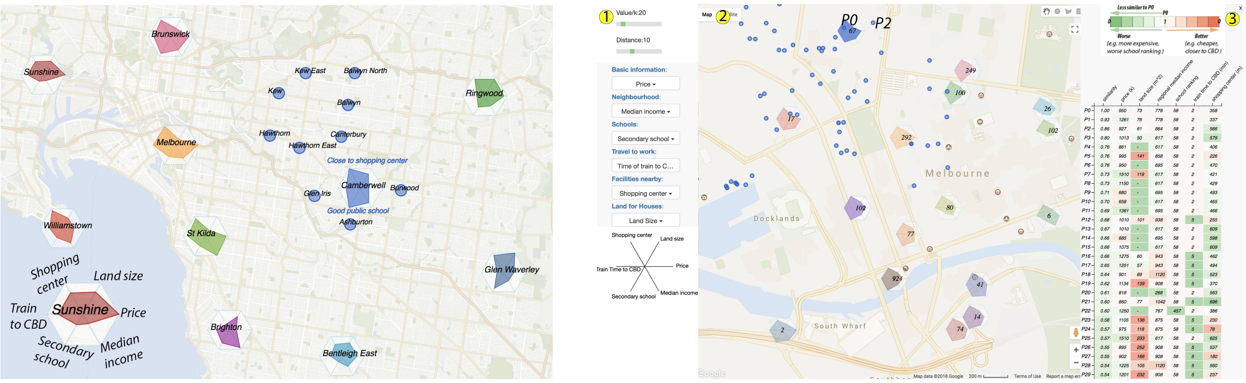
4.1 Demonstration I: the real estate data

The system is initialized with four parameters: a default map window (i.e., maximum and minimum latitudes and longitudes), an integer k, a distance threshold θ, and a user-defined attribute set A (i.e., those factors of properties that the user concerns about). The map window is directly affected by users' zooming and panning operations. All the other parameters can be modified by users from the selection panel (Figure 3(b)-1).

Specifically, our first demonstration is based on the scenario that a user, *John*, is trying to find a property to live in at Melbourne.

SOS Query 1: Exploring suburbs with the map view. *John* first selects the suburb view, as he wants to explore the local real estate market at suburb level, and understand how real estate properties vary from each other across locations. *John* selects those attributes that he is concerned about from the selection panel, including price, public schools, distance to the train station, etc. As shown in Figure 3(a), 10 (the value of k) representative suburbs are shown to *John*, visualized with a radar chart on top of the Google Maps. Each axis of the radar chart corresponds to a selected attribute. As shown in Figure 3(a), *Camberwell* has high-ranked public secondary schools, is close to shopping centers, and is with reasonable distance to the CBD comparing to other suburbs. When John clicks on *Camberwell*, those suburbs (e.g. *Hawthorn* and *Kew*) that are similar to and represented by *Camberwell* will be visualized as circles on top of the map, which are rendered with the same colour as *Camberwell*.

SOS Query 2: Exploring properties with the map view and the detailed information view. John then selects the property view



Value/k:20

Distance:10

Basic information: Price

Neighbourhood: Median income

Schools: Secondary school

Travel to work: Time of train to C...

Facilities nearby: Shopping center

Land for Houses: Land Size

	similarity	price (k)	land size (m^2)	regional median income	school ranking	train time to CBD (min)	shopping center (m)
P0	1.00	950	73	778	58	2	358
P1	0.93	1261	76	778	58	2	337
P2	0.86	927	61	864	58	2	566
P3	0.80	1013	50	617	58	2	579
P4	0.76	861	-	617	58	2	406
P5	0.76	995	141	858	58	2	226
P6	0.76	950	-	695	58	2	470
P7	0.73	1510	119	617	58	2	421
P8	0.73	1150	-	617	58	2	429
P9	0.71	680	-	695	58	2	493
P10	0.70	658	-	617	58	2	465
P11	0.69	1361	-	695	58	2	466
P12	0.68	1010	101	938	58	5	255
P13	0.67	1010	-	617	58	2	609
P14	0.66	685	-	695	58	2	598
P15	0.66	1075	-	617	58	2	609
P16	0.66	1275	60	943	58	5	462
P17	0.65	1251	57	943	58	5	494
P18	0.64	901	89	1120	58	5	523
P19	0.62	1134	139	908	58	5	370
P20	0.61	818	-	268	58	2	563
P21	0.60	860	77	1042	58	5	696
P22	0.60	1250	-	767	457	2	386
P23	0.58	1105	136	875	58	5	230
P24	0.57	975	118	875	58	5	78
P25	0.57	1510	233	617	58	2	625
P26	0.55	895	252	908	58	5	537
P27	0.55	902	166	908	58	5	180
P28	0.54	1225	105	1120	58	5	560
P29	0.54	1201	232	908	58	5	237

(a) Exploring suburbs with the map view

(b) Exploring individual properties: (1) selectio panel; (2) map view; (3) detailed information view

Figure 3: Demonstration examples based on the real estate data

to explore individual real estate properties. Similar to that in the suburb level, k representative properties in the current map window are displayed as k radar charts on top of the map (Figure 3(b)). When John clicks on a property (p_0) that interests him, all the properties ($P = \{P_i | i \in [1, k]\}$) that are similar to and represented by p_0 will be displayed as circles on the map, rendered with the same colour as p_0. Simultaneously, a linked view is presented on the screen to visualize the detailed information of P. As shown in Figure 3(b)-3, we visualize P as a 2D heatmap. Each row corresponds to a property (p_i), sorted based on the similarity score ($Sim(p_i, p_0)$) between p_i and p_0. Each column represents a user-defined attribute (a_j), which can be changed by the user from the selection panel (Figure 3(b)-1). Particularly, the first row represents p_0 itself; and the first column visualizes the similarity score $Sim(p_i, p_0)$. The colour of the cell ($p_i.a_j$) is designed to reflect the similarity of property p_i and property p_0 on the attribute a_j: p_0 lies in the middle of the colour scale (shown at the top of Figure 3(b)-3); to both ends of the colour scale, p_i is less similar to p_0, with the right side of scale meaning p_i is better than p_0 (e.g. cheaper, better school rank, etc.) and the left side meaning the opposite. For example, p_2 is highly similar to p_0; the differences between them are that p_2 has a larger distance to shopping centres and a slightly cheaper price.

4.2 Demonstration II: the twitter data

We also demonstrate POIsam based on the twitter data. While the real estate data is high-dimensional, the main content of each tweet is textual. We measure the similarity of two tweets with Cosine Similarity of the keyword vectors. As shown in Figure 1(b), our demonstration system for the twitter dataset includes a Google Maps view, a word cloud view and a detailed tweet view. In the Google Maps view, each representative tweet is mapped as a circle based on its geographic locations, and the content of some selected tweets is shown together with the location (Figure 1(b)-1). When the user clicks on a representative tweet (t_0), the detailed textual information of those tweets (T) that are similar to t_0 will be summarized in the word cloud view (Figure 1(b)-2), and listed in the detailed tweet view (Figure 1(b)-3).

In our demonstration, we will ask the audiences to interact with the map (zooming, panning, etc.), and show them how POIsam responds to their interactions. To differentiate the new representative tweets (after user interactions) with the old representative tweets, we display the new tweets in a different colour.

ISOS Query 1: Zooming. Users zooming in/out on the map will triage the ISOS query. After users zoom in to a finer granularity, those tweets that are from the previous map window and are still in the current map window will be remained; while new representative tweets will be also added. As a result, tweets that are representative by one tweet might be split to several groups and represented by different new tweets. The screenshot is omitted due to space limit.

ISOS Query 2: Panning. POIsam keeps the consistency as users pan on the map. As a result, those representative tweets from the previous map window and fit in the new map window will be kept; and representative tweets in the new region will be calculated.

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] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The r*-tree: an efficient and robust access method for points and rectangles. In *Acm Sigmod Record*, volume 19, pages 322–331. ACM, 1990.

[2] A. Das Sarma, H. Lee, H. Gonzalez, J. Madhavan, and A. Halevy. Efficient spatial sampling of large geographical tables. In *ACM SIGMOD 2012*, pages 193–204.

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

[4] T. Guo, K. Feng, G. Cong, and Z. Bao. Efficient selection of geospatial data on maps for interactive and visualized exploration. In *ACM SIGMOD*, 2018.

[5] P. K. Kefaloukos, M. Vaz Salles, and M. Zachariasen. Declarative cartography: In-database map generalization of geospatial datasets. In *IEEE ICDE 2014*, pages 1024–1035.

[6] M. Li, Z. Bao, F. Choudhury, and T. Sellis. Supporting large-scale geographical visualization in a multi-granularity way. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 767–770. ACM, 2018.

[7] S. Nutanong, M. D. Adelfio, and H. Samet. Multiresolution select-distinct queries on large geographic point sets. In *ACM SIGSPATIAL 2012*, pages 159–168.

[8] S. Peng, H. Samet, and M. D. Adelfio. Viewing streaming spatially-referenced data at interactive rates. In *ACM SIGSPATIAL*, pages 409–412, 2014.