GeoGuide: An Interactive Guidance Approach for Spatial Data

Plácido A. Souza Neto*, Behrooz Omidvar-Tehrani†, Felipe M. Freire Pontes‡, Francisco Bento‡
*‡*Federal Institute of Rio Grande do Norte - IFRN, Brazil*
**placido.neto@ifrn.edu.br,* ‡*{freire.pontes, bento.francisco}@academico.ifrn.edu.br*
†*The Ohio State University, USA*
†*omidvar-tehrani.1@osu.edu*

***Abstract*—Spatial data is becoming increasingly available in various domains such as urban management and social science. Discovering patterns and trends in this data provides improved insights for planning and decision making in several applications such smart city and disaster management. However, exploratory analysis of such data is a challenge due to its huge size of spatial data. It is often unclear for the analyst *what to see next* during an analysis process, i.e., lack of guidance. To tackle this challenge, we develop GEOGUIDE, an interactive guidance approach for spatial *urban* data. GEOGUIDE captures the feedback of analysts and exploits it to highlight potentially interesting analysis options.**

1. Introduction

Nowadays, there exists huge amounts of spatial data in various fields of science, such as agriculture, transportation and social science. Analysis of such data is interesting as it is grounded on reality: each record represents a specific geographical location. Moreover, understanding patterns and trends provides insights leading to improved user planning and decision making. Some instance applications of spatial data are smart city management, disaster management and autonomous transport [**?**], [**?**].

Spatial data analysis is often performed in *exploratory context*: the analyst does not have a precise query in mind and she explores data in iterative steps in order to find potentially interesting results. Traditionally, an exploratory analysis scenario on spatial data is described as follows: the analyst visualizes a subset of data using a query in an off-the-shelf product (e.g., Tableau[1], Exhibit[2], Spotfire[3]). The result will be illustrated on a geographical map. Then she investigates on different parts of the visualization by zooming in/out and panning the map in order to discover patterns and trends of interest. The analyst may iterate on this process several times by issuing different queries and focusing on different aspects of data.

To overcome the challenge of value in exploratory analysis, visualization environments offer a complete tool-set to manipulate data (filter, aggregate, etc.). In practice, this duplicates the problem: the analyst is left alone in a huge space of spatial data and tools. The principled challenge for the analyst is *"what to see next"* in the exploratory context. A *guidance* mechanism is then necessary to point out potential future directions of analysis.

The following example illustrates the challenge in practice.

Example 1. Liam is planning a short trip to Paris. He decides to rent a home-stay from Airbnb website[4]. He is open to any type of lodging and he wants to explore different options (i.e., exploratory analysis). He queries all available locations in Paris with a fair price. His query results in 3000 locations. As he has no other preferences, an exhaustive investigation needs scanning each location independently which is nearly infeasible. In case he wants to focus on a smaller set of options, it is not clear which subset he needs to look at. While he is looking at primary locations in the list, he shows interest in having "balcony" as amenity and being close to Eiffel tower. An ideal system can capture this feedback in order to short-list a small subset of remaining locations that Liam should consider as high priority.

In order to contribute to overcome this challenge, we propose GEOGUIDE, an interactive framework to highlight a subset of geographical points based on analyst feedback. Although GEOGUIDE operates on points, its functionality can be easily extended to regions using point-clustering methods. The highlighted set facilitates the decision-making process by providing guidance on what the analyst should potentially concentrate on. The set of highlights is deliberated over high quality. We consider two quality metrics in GEOGUIDE: *relevance* and *diversity*. First, each highlighted point should be relevant to historical choices of the analyst. Second, highlights should be geographically diverse to unveil different aspects of analysis. Both quality metrics are interdependent to compute the set of highlights.

Despite literature contains several instances of feedback exploitation to guide the analyst in further analysis steps (e.g., [**?**]), the common used approach is the top-k processing methodology in order to prune the search space based on the explicit feedback and recommend a small subset of

1. *http://www.tableau.com*
2. *http://www.simile-widgets.org/exhibit/*
3. *http://spotfire.tibco.com*
4. *http://www.airbnb.com*

interesting results of size k. A clear distinction and contribution of GEOGUIDE is that it doesn't aim for pruning, but leveraging the actual data with potential interesting results that the analyst may miss due to the huge volume of spatial data. While in top-k processing algorithms, analyst choices are limited to k, GEOGUIDE has a freedom of choice where highlights get seamlessly updated with new analyst choices.

The literature in spatial data analysis also has a focus on *efficiency* of exploratory iterations: *"how can analysts navigate in spatial data fluidly?"* The common approach is to design pre-computed indexes which enable efficient retrieval of spatial data (e.g., [**?**]). However, there has been fewer attention to the *value* of spatial data. Despite the huge progress on efficiency front, an analyst may easily get lost in the plethora of geographical points because i. she doesn't know what to investigate next in an exploratory context and ii. she may get distracted and miss interesting points by visual clutter caused by huge point overlaps. In other words, although iteration transitions (between one analysis step to the other) can be performed efficiently, the decision which forms a transition, remains unclear.

There exist still few instances of information-highlighting methods in the literature [**?**], [**?**], [**?**], [**?**]. All these methods are *objective* and do not apply to the context of spatial guidance where user feedback is involved. In terms of recommendation, few approaches focus on spatial dimension [**?**], [**?**] while the context and result diversification are missing.

To better demonstrate our approach, we summarize the following sections: in Section 2 we present GEOGUIDE and its associated concepts. Section 3 presents the possible scenarios to run GEOGUIDE . Section 4 shows some initial experiments and Section 5 presents some conclusions and future directions.

2. GEOGUIDE Approach: Highlighting Spatial Data

Given a dataset with a set of spatio-temporal information points, our system is able to process and generates highlighted informations base on analyst preferences and behaviour. Our proposed framework is able to highlight different information based on specific data attributes, highlighting, for instance, each points by size or color intensity. Using GEOGUIDE framework the analyst can also define a subset of points to be highlighted over the dataset by a simple filtering action. The functionalities of GEOGUIDE are an inspiration from both recommendation [**?**] and visual highlighting [**?**], [**?**] methodologies. GEOGUIDE is a layer on top of a raw visualization to guide analysts in large-scale spatial data analysis. Figure 1 illustrates the main components of GEOGUIDE architecture.

The following example describes an application of GEOGUIDE in business domain.

Example 2. Tiffany is a data scientist and is tasked to design a "chain marketing" strategy for a Peking Duck product (a Chinese duck dish). She decides to exploit Yelp data[5] (i.e., restaurant check-ins) to find out the advertisement chain. She performs her analysis in GEOGUIDE. In the first step, she shows interest towards New York region, where the headquarters of the company is located and the product has already gained success. The system will then provide few highlights in diverse regions: San Fransisco, Washington DC and Marlton, NJ. All regions seem interesting to Tiffany as they exhibit similar eating profile with New York, hence potentials for chaining the advertisement. She decides to pick Marlton due to its proximity so that she can reduce transportation costs. The system will then provide other highlights based on her updated feedback. She can then make the city-to-city chain marketing strategy in iterative steps using highlights.

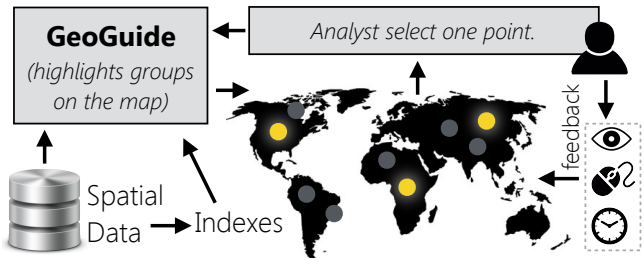

Figure 1. GEOGUIDE Framework

In our approach, we consider a spatial database $\mathcal{D}$ consisting $\langle \mathcal{P}, \mathcal{A} \rangle$ where $\mathcal{P}$ is the set of geographical points and $\mathcal{A}$ is the set of point attributes. For each $p \in \mathcal{P}$, we consider a tuple $\langle lat, lon, alt \rangle$ which denotes p's geographical coordinates (latitude, longitude and altitude respectively). The set $\mathcal{A}_p$ contains attribute-values for p over the schema of $\mathcal{A}$. For instance, on a restaurant check-ins dataset, $\mathcal{A}_p = \langle$`female`, `young`, `chinese-food`$\rangle$ on the schema $\mathcal{A} = \langle$`gender`, `age`, `type`$\rangle$ denotes that p is associated to a young female who likes chinese dish. The set $\mathcal{A}$ is domain-dependent and defines the semantics of a spatial dataset.

2.1. Relevance and Diversity

At each step of the analysis, GEOGUIDE highlights few points based on the feedback content $\mathcal{F}$. The highlighting decision is made based on two quality metrics, i.e., relevance and diversity.

Relevance. Highlights should be in the same line with analyst feedback (captured either by gaze, mouse cursor or session time). Note that we consider *contextual-based* relevance and not *distance-based* relevance. The reason originates from our data observation. For instance in a taxi dataset, consider a ride in New York for a young male customer for an itinerary of 10 kilometers and $3 tip. In contrary to thousands of kilometers of geographical distance, the ride is very relevant to another one in San Fransisco for

5. *https://www.yelp.com/*

a middle-age male customer for an itinerary of 8 kilometers and $2.5 tip. The relevance between a point p and the feedback vector $\mathcal{F}$ is defined as follows.

$$relevance(p, \mathcal{F}) = average_{a \in \mathcal{A}_p \cap \mathcal{F}}(sim(p, \mathcal{F}, a)) \quad (1)$$

The similarity function $sim()$ can be any function such as Jaccard and Cosine. Each attribute can have its own similarity function (as string and integer attributes are compared differently.) Then $sim()$ works as an overriding-function which provides encapsulated similarity computations for any type of attribute.

Diversity. Highlighted points should also represent distinct regions so that the analyst can observe different aspects of data and decide based on the big picture. Given a set of points $s = \{p_1, p_2 \ldots\}$, we define *diversity* as follows.

$$diversity(s) = average_{\{p,p'\} \subseteq s | p \neq p'} distance(p, p') \quad (2)$$

The function $distance(p, p')$ operates on geographical coordinates of p and p' and can be considered as any distance function of Chebyshev distance family such as Eucledian. However, as distance computations are done in *spherical space* using latitude, longitude and altitude, it is au-naturel to employ Haversine distance shown in Equation 3.

$$\begin{aligned} &distance(p, p') \\ &= [acos(cos(p_{lat}).cos(p'_{lat}).cos(p_{lon}).cos(p'_{lon}) \\ &\quad + cos(p_{lat}).sin(p'_{lat}).cos(p_{lon}).sin(p'_{lon}) \\ &\quad + sin(p_{lat}).sin(p'_{lat}))] \times earth\_radius \end{aligned} \quad (3)$$

GEOGUIDE employs a best-effort greedy approach to efficiently compute highlighted points. We consider an offline step followed by the online execution of GEOGUIDE. In order to speed up computing relevance in online execution, we pre-compute an inverted index for each single geographical point in $\mathcal{P}$ in the offline step (as is commonly done in Web search). Each index $\mathcal{L}_p$ for the point p keeps all other points in $\mathcal{P}$ in decreasing order of their relevance with p.

During online execution, GEOGUIDE admits as input a point $p \in \mathcal{P}$ (the user explicit choice) and returns the set of highlights $\mathcal{H} \subset \mathcal{P}$. GEOGUIDE makes sequential accesses to $\mathcal{L}_p$ to greedily maximize diversity. Points in $\mathcal{L}_p$ get a weight using $\mathcal{F}$. Points with a larger weight (i.e., closer to the analyst feedback) have a higher chance to be in $\mathcal{H}$. To speed up comparisons with $\mathcal{F}$ vector, we exploit bit-wise comparisons. We convert both $\mathcal{F}$ and point p to boolean representations and compute relevance (Equation 1) using bit-wise operators.

GEOGUIDE does not sacrifice efficiency in price of value. We consider a *time limit* parameter which determines when the algorithm should stop seeking maximized diversity. Scanning inverted indexes guarantees the relevance even if time limit is chosen to be very restrictive. Our observations with several datasets show that we achieve the diversity of more than 0.9 with time limit set to $200ms$.

Algorithm 1: HIGHLIGHTER Algorithm

```
   Input: p ∈ P, σ, k, tlimit
1  S_p ← get_top_k(L^p)
2  p_next ← get_next(L^p)
3  while (tlimit not
     exceeded ∧ relevance(p, p_next) ≥ σ) do
4      for p_current ∈ S_p do
5          if diversity_improved(S_p, p_next, p_current)
             then
6              S_p ← replace(S_p, p_next, p_current)
7              break
8          end
9      end
10     p_next ← get_next(L^p)
11 end
12 return S_p
```

2.2. Algorithm

We propose a solution for GEOGUIDE by inspiring from both recommendation [**?**] and visual highlighting [**?**], [**?**] methodologies. GEOGUIDE requires an efficient algorithm for dynamically analyzing and comparing geographical points. We propose GEOGUIDE as a solution for the generic guidance problem in spatiotemporal data (Figure 1). Although GEOGUIDE operates on points, its functionality can be easily extended to regions using point-clustering methods such as k-means.

GEOGUIDE operates in two steps: PREPARATION and HIGHLIGHTER. In order to speed up computing relevance in online execution, we pre-compute an inverted index for each single geographical point in $\mathcal{P}$ in the offline PREPARATION step (as is commonly done in Web search). Each index $\mathcal{L}_p$ for the point p stores all other points in $\mathcal{P}$ in decreasing order of their relevance with p. Thanks to the parameter σ, we only partially materialize the indexes.

Algorithm 1 illustrates the online execution step of GEOGUIDE so called HIGHLIGHTER. The algorithm is a single greedy procedure that solves the GEOGUIDE problem. HIGHLIGHTER is called at each interactive step of GEOGUIDE (as in Figure 1). The algorithm admits as input a point $p \in \mathcal{P}$ and returns the best k points denoted $\mathcal{S}_p$.

HIGHLIGHTER begins by retrieving the most relevant points to p by simply retrieving the k highest ranking points in $\mathcal{L}_p$ (line 1). Function $get\_next(\mathcal{L}_p)$ (Line 2) returns the next point p_{next} in $\mathcal{L}_p$ in sequential order. Lines 3 to 11 iterate over the inverted indexes to determine if other points should be considered to increase diversity while staying within the time limit and not violating the relevance threshold with the selected point. Since points in $\mathcal{L}_g$ are sorted on decreasing relevance with p, the algorithm can safely stop as soon as the relevance condition is violated (or if the time limit is exceeded).

The algorithm then looks for a candidate point $p_{current} \in \mathcal{S}_p$ to replace in order to increase diversity. The boolean function $diversity\_improved()$ (line 5) checks if

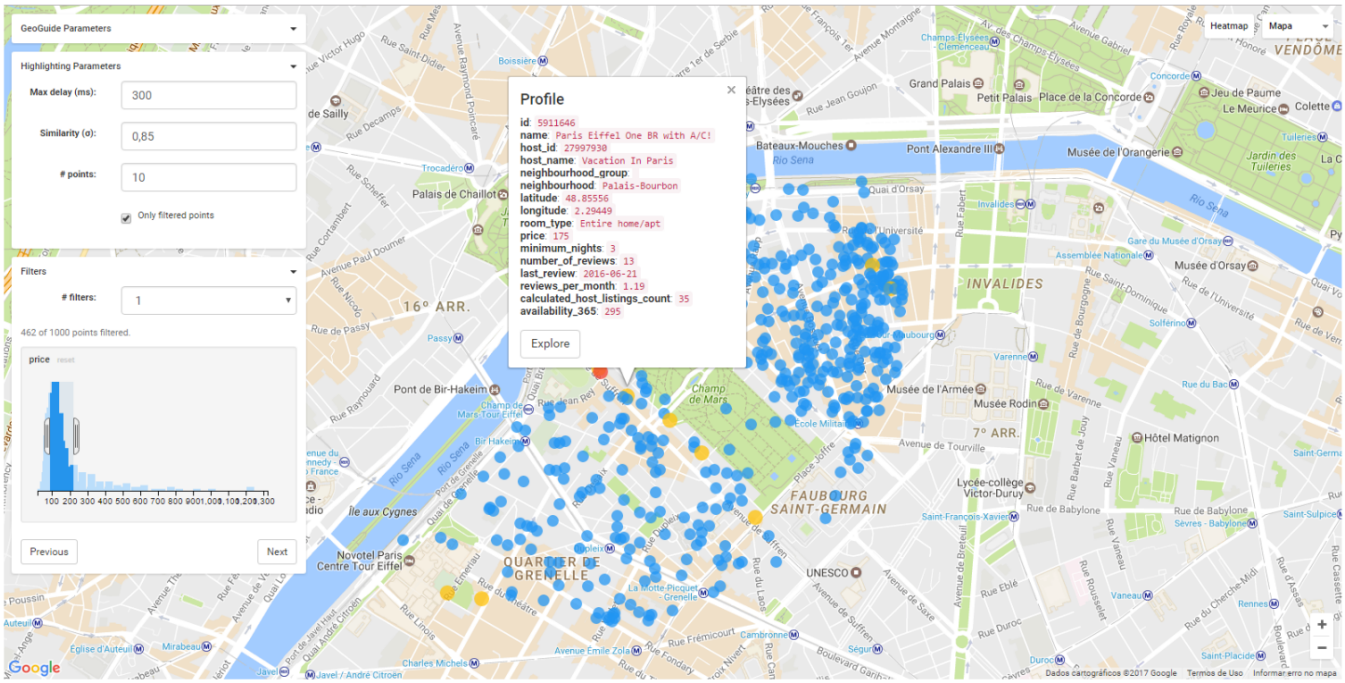

Figure 2. GEOGUIDE Image on Airbnb Dataset - Paris City

by replacing $p_{current}$ by p_{next} in $\mathcal{S}_p$, the overall diversity of the new $\mathcal{S}_p$ increases.

2.3. Tracking User's Preferences

The feedback vector $\mathcal{F}$ on the schema $\mathcal{A}$ is initialized by zero. The vector gets updated by $\mathcal{A}_p$ whenever the analyst shows interest in a geographical point p. Feedback vector is always kept normalized, i.e., $\Sigma_{v \in \mathcal{F}}(v) = 1.0$. Unlike the literature which mainly focuses on explicit feedback (where the analyst should clearly reflect her likes and dislikes), we investigate on implicit feedback. This enables the system to capture *what the analyst may miss* instead of what the analyst has clearly investigated before. We consider different ways to capture implicit feedback.

- **Gaze Tracking.** During spatial data analysis, it is often the case that analysts look at some regions of interest but forget to provide an explicit feedback. For instance in Example 1, while Liam is focusing on home-stays close to the Eiffel tower, he also looks at farther locations with easy train access. However, he never clicks on those points. We call this latent signal, *gaze*. It shown in [**?**] that gaze has a strong correlation with "user attention". The signal can be captured by tracking eye movements aka saccades [**?**]. We employ IXLABS gaze tracking[6] as it only needs a simple web-cam to capture the gaze signal.
- **Cursor Tracking.** To address privacy issues of web-cam exploitation for gaze tracking, we consider an alternative option of tracking the mouse cursor. It is shown in [**?**] that mouse gestures have a strong correlation with "user engagement". Intuitively, a point receives a positive feedback if the cursor moves around it frequently.
- **Session Time.** In most spatial datasets, there is a profile page dedicated to each point. Examples are restaurant pages in Yelp and lodging pages in Airbnb. We consider the amount of time that the analyst spends in a page as an implicit feedback. For instance, if the analyst spends few minutes in a page for an Indian cuisine restaurant, this counts as positive feedback for this type of restaurants.

3. Application Scenarios

Our application scenarios will describe how GEOGUIDE can be used. First, we present two different scenarios and its datasets. For each scenario, we demonstrate a general view of GEOGUIDE . Then, we describe the sequence of actions to see the highlighted points generated by the environment, and its properties. After, we will present how to align different filter types in order to improve the results. Finally, we will present some results from explicit user's feedbacks, e.g., by choosing different parameters to be highlighted in therms of size and colors.

Scenario 1. On Airbnb dataset, we demonstrate how GEOGUIDE can contribute to approach a lodging of interest based on analyst's feedback. We consider the concrete case of finding a cheap lodging solution with a balcony near Eiffel tower. We will observe how feedback converges the exploration towards the goal very quickly. The analyst, after uploading the possible hosting locations available on Airbnb, she will see some possible places of her interest (in this case, the Eiffel tower) by zooming in on the map. She can filter by price range of her interest, in this case between

6. *http://www.xlabsgaze.com/*

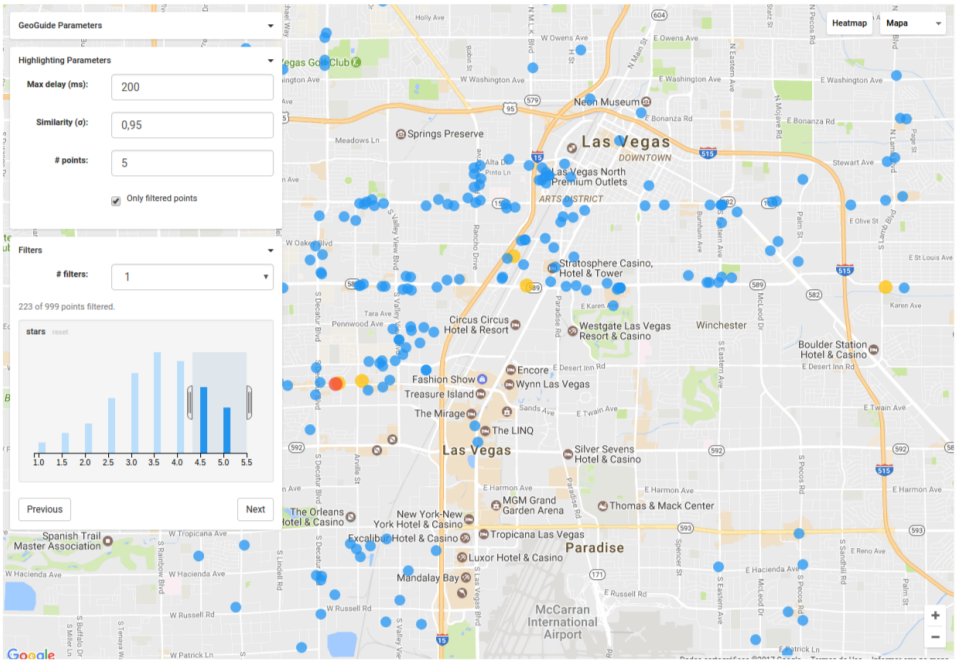

Figure 3. GEOGUIDE Image on Yelp Dataset - Las Vegas City

$100 - $200 dollars. Then, she can select a host (point in the map) of her interest. Let's consider the place *Chambre luxe 100M Tour Eiffel* (red point in Figure 2 latitude 48.85639, longitude 2.29311), with price of $ 150 dollars. This place has 1.79 "reviews per month" and availability of 362 over 365. She wants to know what other places are similars to this. The analyst sets the environment parameters (*Highlighting Parameters*) with *300 ms* of processing, the minimum similarity of *0.85* σ and 10 for the most silimar accommodations from its chosen point.

The result presented in Figure 2 shows the hosts with price in the same range defined in the filter. From this, the analyst can find, between other highlighted point, the place/point "Paris Eiffel One BR" with price of $ 175 dollars per night , 1.19 reviews per month and availability of 295 over 365. This place is a little further from the Eiffel Tower, but still on the same avenue, with similar features and an affordable range based on analyst's first choice.

Scenario 2. On Yelp dataset, e demonstrate how GEOGUIDE can contribute to reach an early consensus on a restaurant. The attendee will observe that his/her preferences will be immediately captured and reflected in future highlights.

When the analyst decides to explore restaurants similar to Ronald's Donuts in Las Vegas with 4.5 stars, 538 ratings and 1090 check-ins, GeoGuide uses the time the participant observed each restaurant (session time), the selected filters and the profile of the restaurants to be explored in the analysis, which combines implicit and explicit behaviors to suggest new restaurants for observation. After run the environment, GeoGuide highlights the points considered relevant to the participant, that is close to the most analyzed areas and the restaurant that was most observed, with an average evaluation between 4.5 and 5. The points are highlighted in yellow, while the exploded spot is left in red as in Figure 3. The analyst will then evaluate the options suggested by the GeoGuide to decide whether to choose one of these restaurants or whether to continue exploring.

4. Experiments

To validate our design choices in GEOGUIDE (quality dimensions and feedback capturing), we design a user study with 24 participants (students in Computer Science). We define a task for each participant and ask him/her to fulfill the task using GEOGUIDE and TABLEAU (as the most advanced off-the-shelf visualization product). Then we measure the number of steps to reach the goal. We define two tasks, *T1: finding a point in a requested location* (e.g., find a home-stay in the Central Park area, New York), and *T2: finding a point with a requested profile* (e.g., find a cheap home-stay with balcony in Paris.) Participants may begin their navigation from three different starting points: *I1: close to the goal*, *I2: far from the goal*, and *I3: random*.

In TABLEAU, participants employ filtering and querying tools to reach their goals. In GEOGUIDE, participants benefit from relevant and diverse highlights and feedback capturing using cursor tracking. Figure 4 illustrates the results of this study. We report results for separate sub-populations: the left figure illustrates the results for novice participants (who don't know the location, be it Paris or New York) and the right figure illustrates expert's results.

We observe that in general, it takes in average 10.7 steps to reach a defined goal in GEOGUIDE, i.e., 33 steps less than TABLEAU. This shows that the highlighting component equipped with the feedback mechanism helps analysts

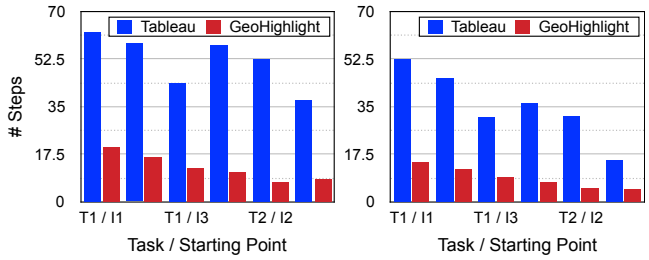

Figure 4. User Study

discover their spatial data and reach to the goal. Level of expertise improves the analysis length in average by 4 steps. Interestingly, starting points do not have a huge influence. It is potentially due to the diversity component which provides distinct options. We also observe that *T2* is an easier task than *T1*. This is potentially due to similarity component where the analyst can request options similar to what she has already seen and greedily moves to match profiles.

GEOGUIDE [7] environment[8] is implemented in Python (as the computation engine), JavaScript D3 (as the visualization engine) and a set of Python libraries. We use and provide some spatial datasets[9] to be run in the framework : Yelp dataset[10] of restaurant check-ins with 229,907 geographical points, Airbnb dataset[11] for short-term lodging with 4,200,000 points, New York taxi dataset[12] with 173,179,759 points and citibike New York dataset[13] with different sizes of data.

5. Conclusions

We addressed the problem of guidance and introduced GEOGUIDE, the first efficient interactive highlighting approach in spatial data. We formulated our problem in form of a constrained optimization and proposed HIGHLIGHTER, a greedy algorithm to highlight k-best points for a given point of interest within a time limit. We introduced the discussion of genericness of our approach by materializing few examples from restaurant and apartment rental datasets. We also showed the efficiency and usability of our framework in form of performance experiments and user study. There are several directions of improvement for this work. Specifically, we want to consider an analyst profile vector which is built during interactive steps and will be exploited to return more analyst-tailored results.

Some future extensions include the integration of generic query approach based on [**?**], and we also are going to consider an analyst profile vector which is built during interactive steps and will be exploited to return more analyst-tailored results.

7. GEOGUIDE can be played accessing its environment on http://geoguide.herokuapp.com/.

8. A short GEOGUIDE demo can be seen in https://www.youtube.com/watch?v=MTTccStmd0E&feature=youtu.be

9. https://github.com/placidoneto/Data-Visualization/experiments/inputs

10. https://www.yelp.com/dataset_challenge

11. http://insideairbnb.com/get-the-data.html

12. https://data.cityofnewyork.us/Transportation/Taxi/mch6-rqy4/data

13. https://s3.amazonaws.com/tripdata/index.html