On Designing a GeoViz-Aware Database System - Challenges and Opportunities

[1]Mohamed Sarwat and [2]Arnab Nandi

[1] Arizona State University, 699 S. Mill Ave, Tempe AZ
[1]msarwat@asu.edu
[2] Ohio State University, 2015 Neil Ave, Columbus OH
[2]arnab@cse.osu.edu

1 Motivation and Challenges

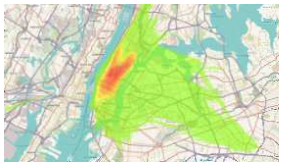

Fig. 1. NYC taxi trip heatmap

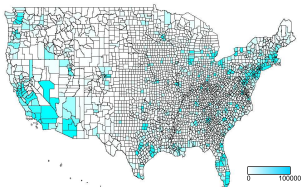

Fig. 2. Tweets Choropleth map

The human's ability to perceive, consume, and interact with the data is in-fact limited. One key observation is worth considering – that Geospatial data is typically consumed as aggregate visualizations. e.g., Heatmap, Choropleth map (Figure 2), Cartogram. For instance, Figure 1 shows a heatmap of the drop-off locations of 1.1 billion NYC Taxi trips. Also, interactions are performed in a manner constrained by the user map interface. Thus, leveraging this observation allows us to aspire towards large data sizes, while keeping the user-facing outputs constant. GeoVisual analytics, *abbr. GeoViz*, is the science of analytical reasoning assisted by interactive GeoVisual map interfaces. While there exists decades of research in spatial data management, enabling practitioners in non-Computer Science fields to perform highly interactive GeoViz over large-scale spatial data remains challenging. Off-the-shelf visualization and data exploration tools focus on business domains, while existing GIS products focus on data management and exploration. We recognize a number of challenges in analysis of spatiotemporal data: *(1) Scalability:* The massive-scale of spatial data hinders visualizing it using traditional GIS tools. Many map services such as MapBox and GIS tools (e.g., QGIS) allow users to visualize a fairly small amount of spatial data. The problem becomes more challenging when the GIS tool has to load, render, and visualize terabytes of geospatial data. *(2) Interactivity:* Currently, an expert can easily produce a one-shot GeoViz analysis in spatial data, producing a static visualization. However, sifting through different attributes to interactively inspect a map view of the data is a challenging task, due to the latency involved in regenerating the visualization. Since interactivity impacts the ability to derive insights at the speed of thought, it is important to reduce roundtrip latency at all points of the stack.

2 Vision: A GeoViz-Aware Database System

The straightforward approach to interactively visualizing spatial data completely decouples the GIS (Geographic Information System) application and the spatial database system such as PostGIS. In this approach, the GIS tool runs at the client side and the DBMS runs at the server side; each performs its task independently from the other. When the user performs a spatial data visualization task, this approach first loads the spatial data that lie within the visualization window from the database into a format understood by the GIS tool. The visualization tool, in turn, visualizes the retrieved spatial data on the map. Analysis of spatial data often begins without a specific intent as an agnostic search, and blends through browsing into concrete user intent and precise querying. An expert interacts with the data by constructing consecutive queries using insights gained from the query session. Such interactions are enabled by interactive dashboards, which provide instantaneous response, helping the user quickly discover insights such as trends and patterns. However, the straightforward approach issues a new spatial range query to the database system for each user interaction with the map. The user will not tolerate delays introduced by the underlying spatial database system to execute a new spatial query for every single change in the viewport. Instead, the user needs to visualize useful information quickly and interactively change her visualization (e.g., zoom in/out and pan) if necessary.

Expressing GeoViz in Database Systems: We envision a new approach, namely GeoVizDB, that injects interactive GeoViz map exploration awareness inside the spatial database system. The user can issue a GeoViz query to the system. Once the initial query is exhibited, the expert interacts with data. She can continue studying the preliminary GeoViz query result, generate new visualizations, or replace the default one with the analysis and measure of his choice, which will utilize the already generated materialized view. Note that the initial query is independent of the user's intent, and hence can be stored as a materialized view in the database. Also, note that any change to the underlying spatial data or the visualization window will result in the materialized view being updated, and resulting queries and visualizations being regenerated. Following [10], GeoVizDB needs to recognize following categories of interactions: *(a) Navigational Interactions:* They consist of actions such as zoom in/out or panning on the map. Navigational interactions can be formulated as *brushings* (a brush on any visual component immediately updates other components) on the visual components. Brushing is an *incremental query*, and results in a union or deletion of one of the current spatial filters and the subsetted data. As a result, an incremental query does not need to be fully materialized – only differential data subset (and result in the case of algebraic and distributive measures [9]), needs to be retrieved. A brushing action is composed of three components: *action* (union, subtraction), *affected boundary* (lower, upper bounds), and *attribute*. *(b) Informational Interactions:* These actions provide more details for a selected point. An example of this kind of interaction is the *"what's here"* operation in Google Maps. *(c) Fusion Interactions:* Such interactions enable merging current data with external sources, which is necessary to discover the extrinsic causali-

ties of current spatial observations. Note that a pair of spatial data points can be joined using their latitude, longitude, and time. An expert often goes through a sequence of navigational interactions in an *exploratory context* and then switches to *investigation context* by using informational interactions to request complementary details, or by using fusion interactions to explain observations.

Prefetching and caching spatial data: Spatial data is likely to be accessed again in the near future due to the temporal locality principle, hence the system needs to cache results of recently executed queries in memory. If a subsequent request is a subset of one of the previous queries (e.g. zoom), the database systems stays untouched – results can be returned to the analyst almost immediately from the cache. Assume the user visualizes spatial objects within a specific rectangular range R and then decided to slightly expand, shrink, or move the original viewport R to $R^{'}$. If the system could predict $R^{'}$, it might speculatively pre-fetch the answer to $R^{'}$ so that the user gets the answer to $R^{'}$ very fast when needed. To achieve that, the system needs to employ a smart speculation algorithm that is able to predict what kind of interactions the user might issue in her GeoViz session. This task is challenging for the following reasons: (1) The spatial query variations might be endless and hence speculatively computing the answer to all possible variations leads to huge system overhead. (2) Even if the number of speculative queries is reasonable, the user might wind up not using any of the speculatively calculated answers and hence the amount of work spent on speculation and data pre-fetching would be a waste. With cache being a limited resource, the system needs to provide principled caching strategies to improve the cache-hit rate. Three have been several research efforts on predicting and preloading possible upcoming data chunks. ATLAS [4] and ImMens [8] employ simple user movement prediction approaches such as Momentum and Hotspot. Existing work [3] leverages Markov chain to further improve the prediction accuracy. Researchers [6] in vehicle navigation community also use Markov chain to predict traffic trajectory which is similar to user movement. DICE [5] considers a cube traversal-based model to shrink the prefetching space while ForeCache [1] argues that the prediction model should consider not only user movement but also the data chunk features (e.g. color histogram). However, these systems do not provide native support for general spatial objects, e.g., points, polygons. Also, these systems are crafted for a custom visualization tool and cannot be easily plugged into generic GIS tools. GeoVizDB must, on the other hand, support GeoViz-aware spatial data caching/pre-fetching as a generic middleware between the GIS application and the spatial database system.

Sampling spatial data: Achieving real-time performance for GeoViz applications is quite challenging even when employing high performance computing and modern hardware infrastructure. For instance, the NYC heatmap in Figure 1 requires the retrieval of billions of spatial objects from the database, which may take so long to run. The problem is further amplified when more spatial objects need to be loaded in response to the user's interactions with the GeoViz map. Given that a heatmap (same for other GeoViz) represent an aggregate view of the data, there is room for trading interactive performance for accuracy. To

achieve that, data sampling techniques may scale the GeoViz process by getting rid of overly-detailed spatial objects. Random sampling and stratified sampling are two widely-used simple approaches when people want to only pick the most representative objects. Nano Cube[7] and Hashed Cube[11] maintain compressed aggregates of the spatial data to scale the GeoViz process. RS-Tree [13] augments the R-tree data structure to retrieve just a sample of the spatial data that lie within the query range. ScalaR [2] and VAS[12] store precomputed multiple resolution aggregates of the data using a database system to achieve interactive performance. Even though spatial data sampling/compression techniques allow users to visualize the spatial sample using de-facto GIS applications. This is due to the fact that each spatial query returns compact version of the spatial data that the GIS tool is able to efficiently visualize. Nonetheless, such sampling/compression techniques face the following challenges: (a) fail to provide high quality GeoViz map images for the user, (b) are not tailored to handle geospatial map visualizations and hence may highly compromise the GeoViz accuracy, (c) cannot easily support the streaming nature of geospatial data.

References

1. L. Battle, R. Chang, and M. Stonebraker. Dynamic prefetching of data tiles for interactive visualization. In *SIGMOD*. ACM, 2016.
2. L. Battle, M. Stonebraker, and R. Chang. Dynamic reduction of query result sets for interactive visualizaton. In *Big Data*, pages 1–8. IEEE, 2013.
3. U. Cetintemel, M. Cherniack, J. DeBrabant, Y. Diao, K. Dimitriadou, A. Kalinin, O. Papaemmanouil, and S. B. Zdonik. Query steering for interactive data exploration. In *CIDR*, 2013.
4. S.-M. Chan, L. Xiao, J. Gerth, and P. Hanrahan. Maintaining interactivity while exploring massive time series. In *Symposium on Visual Analytics Science and Technology*, pages 59–66. IEEE, 2008.
5. N. Kamat, P. Jayachandran, K. Tunga, and A. Nandi. Distributed Interactive Cube Exploration. *ICDE*, 2014.
6. J. Krumm. A markov model for driver turn prediction. In *SAE*, 2008.
7. L. Lins, J. T. Klosowski, and C. Scheidegger. Nanocubes for real-time exploration of spatiotemporal datasets. *TVCG*, 19(12):2456–2465, 2013.
8. Z. Liu, B. Jiang, and J. Heer. immens: Real-time visual querying of big data. In *Computer Graphics Forum*, volume 32, pages 421–430. Wiley Online Library, 2013.
9. A. Nandi, C. Yu, P. Bohannon, and R. Ramakrishnan. Data cube materialization and mining over mapreduce. *IEEE TKDE*, 24(10):1747–1759, 2012.
10. B. Omidvar-Tehrani, S. Amer-Yahia, and A. Termier. Interactive user group analysis. In *CIKM*, pages 403–412. ACM, 2015.
11. C. Pahins, S. Stephens, C. Scheidegger, and J. Comba. Hashedcubes: Simple, low memory, real-time visual exploration of big data. *TVCG*, 2016.
12. Y. Park, M. Cafarella, and B. Mozafari. Visualization-aware sampling for very large databases. In *ICDE*. IEEE, 2016.
13. L. Wang, R. Christensen, F. Li, and K. Yi. Spatial online sampling and aggregation. In *VLDB*, volume 9, pages 84–95. VLDB Endowment, 2015.