Visualizing Accessibility With Choropleth Maps

David Li
dli7319@umd.edu
University of Maryland
College Park, Maryland, USA

Hanan Samet
hjs@umd.edu
University of Maryland
College Park, Maryland, USA

Amitabh Varshney
varshney@umd.edu
University of Maryland
College Park, Maryland, USA

ABSTRACT

We present a system to visualize accessibility to various destinations from essential institutions such as schools and hospitals to common attractions such as beaches. Our visualization system supports real-time computations of driving distances by leveraging the path coherent pairs (PCP) decomposition which allows for fast computation between thousands of points of interest. Our system allows users to import and switch between several datasets without any precomputation of road distances between specific entries in the dataset. We present a case study that demonstrates our visualization system generating Choropleth maps of accessibility to various destinations in the San Francisco Bay Area which could be used to guide tourism and event planning decisions.

CCS CONCEPTS

• **Human-centered computing → Geographic visualization**.

KEYWORDS

choropleth, maps, road distance, accessibility

ACM Reference Format:
David Li, Hanan Samet, and Amitabh Varshney. 2021. Visualizing Accessibility With Choropleth Maps. In *5th ACM SIGSPATIAL International Workshop on Location-Based Recommendations, Geosocial Networks, and Geoadvertising (LocalRec'21), November 2–5, 2021, Beijing, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3486183.3492801

1 INTRODUCTION

When planning events such as conferences, weddings, or vacations, the goal is to pick the optimal destination for their trip. Typically, the optimal destination will have ready access to both amenities such as beaches and restaurants as well as essential institutions such as hospitals. While picking a sub-optimal location could merely lead to reduced participant satisfaction, in the worst case, the wrong choice could be catastrophic. For instance, a 2005 study by Buchmueller *et al.* [5] found that just an additional one-mile increase in distance to hospitals leads to a 6.45% increase in heart-attack deaths and an 11% increase in deaths from unintentional injuries each year.

One method of easily visualizing average distances is through Choropleth maps. By dividing a map into different regions, each represented by a solid color, Choropleth maps allow people to quickly compare statistics across different counties, states, or even countries. However, as data in Choropleth maps are aggregated within each region, building Choropleth maps may pose a challenge for complicated queries such as average road distance.

In this paper, we present an interactive Choropleth visualization system for measuring accessibility via road distances. By leveraging existing distance oracle technology, our visualization system enables users to quickly build Choropleth maps from a set of dynamic queries and quickly determine the best and worse areas for accessibility.

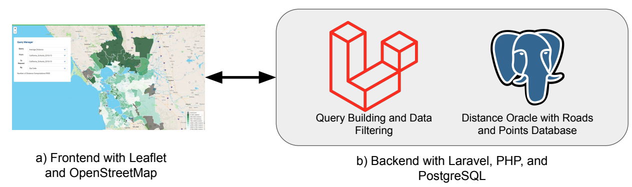


Figure 1: A high-level overview of our Choropleth visualization system.

2 BACKGROUND AND RELATED WORKS

Our work builds upon the existing work on road distance queries and map visualizations but does not deal with other contexts such as distance-based image decompositions (e.g., [11, 12]) or similarity retrieval (e.g., [7]). In this section, we provide a brief overview of existing road-distance computation techniques and highlight a few recent map-based visualization works.

2.1 Road-distance Computations

Road distance computations are a fundamental problem for many computer science applications. The naive way to compute road distances is to perform a shortest-path search through a graph with weighted edges using traditional algorithms such as Dijkstra's or the Bellman-Ford algorithms. However, these algorithms, while popular, are computationally heavy when performing several distance queries over the same graph.

Sankaranarayanan and Samet [14–18] develop distance oracles and distance join algorithms for spatial networks. Their algorithm works by preprocessing the graph of a road network into path-coherent pairs (PCP), an idea based on well-separated pairs. With path-coherent pairs, vertices are placed into different sets. Pairs of sets, e.g. A and B, are placed into tuples (A, B, Ψ, λ), containing the approximate distance λ and intermediate vertex Ψ making use of hierarchical spatial data structures (e.g., [13]). These tuples can then be used to compute distance queries and shortest paths in linear time.

LocalRec'21, November 2–5, 2021, Beijing, China

ACM ISBN 978-1-4503-9100-9/21/11.
https://doi.org/10.1145/3486183.3492801

2.2 Map Visualizations

Peng *et al.* [10] extend the distance oracles implementation to run on Spark which allows queries to be distributed across multiple machines. Doing this allows them to compute a geographical heat map for the average drive distances from home to work across California. Building such a heat map requires 13.6 M distance queries which can be done within several seconds on 5 machines.

Recently, Zhang *et al.* [19] analyze 668 COVID-19 visualizations which include choropleth maps, bubble maps, and proportional symbol maps which are all different methods for visualizing COVID cases and deaths across the United States. They note that Choropleth maps are among the most popular geospatial visualizations but visualizations of COVID cases may need to be normalized by area or regional populations.

3 SYSTEM

We first present a high-level overview of our system. Then we describe the interface of our system which allows users to interactively build and manipulate the Choropleth map.

3.1 Overview

Our system consists of a frontend and backend. Users interact with the frontend which streams an interactive map and builds a Choropleth visualization. Meanwhile, the back end receives queries from the frontend based on user inputs, computes all necessary road distance computations, and returns the result. An overview of our system is shown in Figure 1.

3.1.1 *Frontend.*

For our frontend, we use Leaflet[1] which streams an interactive map from Mapbox[2]. Users can access our system using any modern web browser and quickly build Choropleth visualizations in real-time. Our Choropleth visualizations are subdivided based on ZIP codes. When the frontend sends queries to the backend, only distance computations within the visible ZIP codes are requested in the query.

3.1.2 *Backend.*

For our backend, we use Roads Inside Any Database [8, 9] that computes road distances using a path coherent pairs decomposition and a distance oracle. This is powered by a single PostgreSQL[3] database which performs all of the distance computations necessary for each query. A PHP application built on Laravel processes HTTP requests from the front-end into SQL queries.

To perform average distance computations, our backend executes a query similar to the following:

```
SELECT t1.zip_code, avg(t1.dist) FROM
  (SELECT DISTINCT ON (id1)
    x.id AS id1,
    y.id AS id2,
    dist(x.lat, x.lon, y.lat, y.lon)
      AS dist,
    x.zip_code
   FROM restaurants x, parking y
   WHERE x.id != y.id
   AND x.lat <= y.lat + 2.0 / 110
   AND x.lat >= y.lat - 2.0 / 110
   AND x.lon <= y.lon + 2.0 / 110
   AND x.lon >= y.lon - 2.0 / 110
   AND x.zip_code IN (?, ?, ?, ?, ?)
   ORDER BY x.id, dist) t1
  GROUP BY t1.zip_code
```

The above query first performs a subquery that computes the road distance between every restaurant and nearby parking lot. The subquery then sorts by the distance for each restaurant and takes the smallest distance for each restaurant. From the subquery output, the main query then averages the distance for each restaurant within the same ZIP code and returns the result by ZIP code. When working with different datasets, *restaurant* and *parking* are replaced with appropriate subqueries to other points of interest.

[1]Leaflet: http://leafletjs.com/
[2]Mapbox: https://www.mapbox.com/
[3]PostgreSQL: https://www.postgresql.org/

3.2 Interface

Our visualization system supports two types of controls for viewing as well as a data management interface. Standard map controls allow users to move around the 2D map and zoom in and out while the query selector allows the user to determine what the live Choropleth map shows.

3.2.1 *Map Controls.*

Our system supports standard map gestures built into Leaflet. The user can move around the map by simply dragging it with the mouse cursor or the finger on a touch screen. As the user interacts with the map, different regions appear and disappear, prompting the system to automatically send a query for the new regions.

3.2.2 *Query Selector.*

Our system features a query selector, shown in Figure 2, which allows the user to interactively build queries and control what is being shown in the Choropleth map. Each option in the query manager is powered by a drop-down menu of predefined options. After the query has completed execution, the query selector displays the number of distance computations used to generate the visualization.

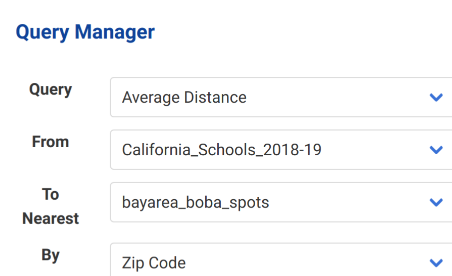
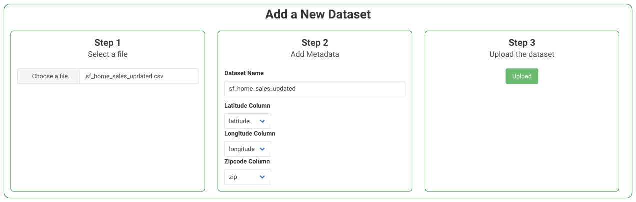


Figure 2: The query selector allows the user to dynamically change what is being visualized using dropdown selections.

3.2.3 *Inspector Tooltip.*

To view the details of any region, users can hover their mouse cursor over the region on the map. Upon doing so, a tooltip appears which displays details of the region such as the city, state, ZIP-code, and query value information.



(a) Adding A New Dataset

Existing Datasets

Dataset Name	Count	Actions
sf_home_sales_updated	34091	Delete
California_Schools_2018-19	10070	Delete
Public Beaches	1637	Delete
bayarea_boba_spots	603	Delete
Park_and_Ride_Lots	325	Delete
Facilities_with_Chemical_Hazards_-_All	1798	Delete
San_Francisco_Department_of_Public_Health_Flu_Shot_Locations_2013	40	Delete
OSHPD_Healthcare_Facilities	6738	Delete
Dataset Name	Count	Actions

(b) View Existing Datasets

Figure 3: The dataset management interface allows users to add new datasets or remove existing datasets.

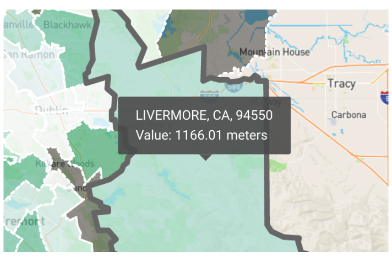


Figure 4: The inspector tooltip shows details and query values associated with each region on the Choropleth map.

3.2.4 Data Management. To allow users to quickly import new data, our system provides a management interface that allows users to upload CSV files containing new points of interest. The frontend parses the file and allows the user to select columns representing the latitude, longitude, and ZIP code. For ease of use, columns named "latitude", "longitude", or "zip" are automatically selected if available. After naming the new dataset and selecting the relevant columns, the user can upload the dataset to be saved into the server. Upon completion, the dataset appears under the existing datasets table and can be used to build queries. Our dataset management interface is shown in Figure 3.

4 BAY AREA ACCESSIBILITY

To demonstrate the types of Choropleth visualizations which can be produced by our system in real-time, we perform an initial case study on accessibility in the Bay Area. For all distance computations, we use Roads Inside Any Database [8, 10] which provides a dataset of roads as well as a prebuilt oracle for the Bay Area of California.

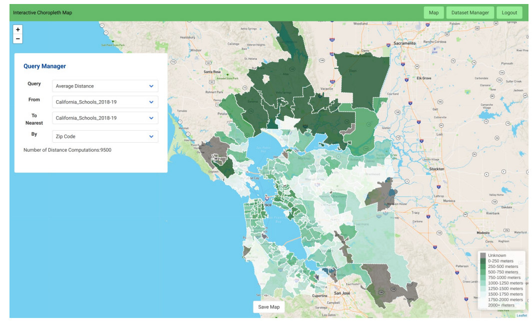


Figure 5: Visualization of the average distance between schools.

4.1 Distance between Schools

We first visualize the driving distance between public schools within California. For this, we use the "California Schools 2018-19" dataset [2] by the California Department of Education. The dataset consists of 10,070 K-12 public schools throughout all of California. We visualize distances around the San Francisco Bay Area organized by ZIP code in Figure 5.

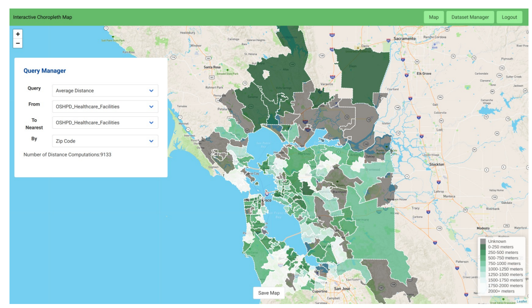


Figure 6: Visualization of the average distance between healthcare facilities.

4.2 Distance between Healthcare Facilities

Our second visualization presents the distance between healthcare facilities in Figure 6. The distance between healthcare facilities may be of interest during events such as natural disasters or pandemics where patients may need to be transferred between hospitals for specialized treatment or logistical reasons. For this visualization, we use the "OSHPD Healthcare Facilities" dataset from the California Office of Statewide Health Planning and Development [1].

4.3 Distance from Schools to Chemical Hazards

Another factor when determining an ideal location for raising a family is the proximity to chemical hazards. Therefore, we visualize the average distance from schools to the nearest facility with chemical hazards in Figure 7. We use the "Facilities with Chemical Hazards" dataset by the California Department of Toxic Substances Control [4]. This dataset consists of 469 facilities around the San Francisco Area that store chemicals with fire, reactivity, or pressure hazards.

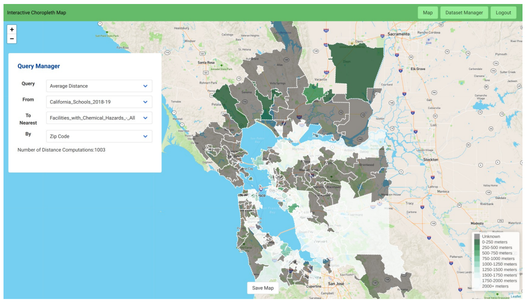


Figure 7: Visualization of the average distance from schools to the nearest chemical hazards.

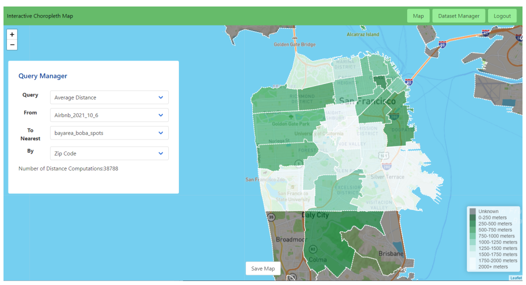


Figure 8: Visualization of the average distance from Airbnb to the nearest boba shops.

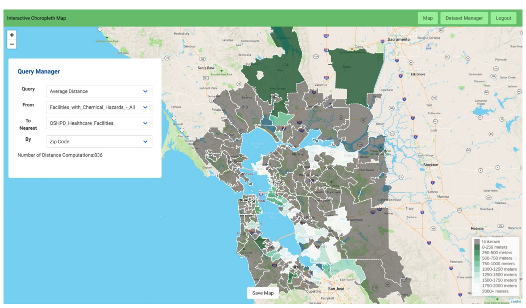


Figure 9: Visualization of the average distance from facilities with chemical hazards to the nearest healthcare facilities.

4.4 Distance from Airbnb Listings to Boba Shops

Next, we visualize the distance from Airbnb listings to boba shops in Figure 8 which could provide insights for those looking to advertise their listings. We use Airbnb listings around San Francisco from October 6, 2021 [3] which contains 6558 listings and "Boba Shops in the Bay Area" [6] by Vincy C. which contains 603 boba shops.

4.5 Distance from Chemical Hazards to Healthcare Facilities

Finally, we visualize the average distance from chemical hazards to the nearest healthcare facilities in Figure 9. This type of visualization may potentially help visualize more hazardous ZIP codes. For example, city planners may be interested in introducing additional healthcare facilities near areas of chemical hazards to minimize the ambulance response time.

5 CONCLUSION

We present a system that produces Choropleth maps to visualize accessibility based on road distances. Our system leverages the Path Coherent Pairs decomposition to support real-time computations of average road distances between points of interest. Our system enables users to explore the average distance to the nearest locations such as beaches and boba shops. An initial case study is performed on locations in the Bay Area to demonstrate the visualizations produced by our system.

We envision our system being used by event planners, tourism agencies, and travellers to guide event location and travel decisions based on the accessibility to hotels, restaurants, and popular attractions. Furthermore, our system could also allow urban planners, public officials, and transportation researchers to more quickly evaluate accessibility to remedy issues of unequal access.

ACKNOWLEDGMENTS

This work was sponsored in part by the NSF under Grants IIS-18-16889, IIS-20-41415, and IIS-21-14451.

REFERENCES

[1] 2018. OSHPD Healthcare Facilities. https://gis.data.ca.gov/datasets/CHHSAgency::oshpd-healthcare-facilities
[2] 2020. California Schools 2018-19. https://gis.data.ca.gov/datasets/CDEGIS::california-schools-2018-19
[3] 2021. Detailed listings data for San Francisco, 06 Oct., 2021. http://insideairbnb.com/get-the-data.html
[4] 2021. Facilities with chemical hazards - All. https://gis.data.ca.gov/datasets/DTSC::facilities-with-chemical-hazards-all
[5] T. C. Buchmueller, M. Jacobson, and C. Wold. 2006. How far to the hospital?: The effect of hospital closures on access to care. *Journal of Health Economics* 25, 4 (2006), 740–761.
[6] Vincy C. 2018. Boba shops in the Bay Area. https://www.kaggle.com/vnxiclaire/bobabayarea
[7] G. R. Hjaltason and H. Samet. 2000. *Incremental similarity search in multimedia databases*. Computer Science Technical Report TR-4199. University of Maryland, College Park, MD.
[8] S. Peng. 2016. Try this example! https://roadsindb.com/what-is-our-technology/demo-time/
[9] S. Peng and H. Samet. 2016. CDO: Extremely high-throughput road distance computations on city road networks. In *GIS '16* (Burlingame, California). Article 84, 4 pages.
[10] S. Peng, J. Sankaranarayanan, and H. Samet. 2016. SPDO: High-throughput road distance computations on Spark using distance oracles. In *IEEE ICDE 2016*. 1239–1250.
[11] H. Samet. 1983. A Quadtree Medial Axis Transform. *CACM* 26, 9 (Sept. 1983), 680–693. Also see CORRIGENDUM, *CACM*, 27(2):151, Feb. 1984.
[12] H. Samet. 1985. Reconstruction of quadtrees from quadtree medial axis transforms. *Computer Vision, Graphics, and Image Processing* 29, 3 (Mar 1985), 311–328.
[13] H. Samet. 1990. Hierarchical spatial data structures. In *Design and Implementation of Large Spatial Databases*. Springer, Berlin, Heidelberg, 191–212.
[14] J. Sankaranarayanan, H. Alborzi, and H. Samet. 2006. Distance Join Queries on Spatial Networks. In *GIS '06* (Arlington, Virginia). 211–218.
[15] J. Sankaranarayanan and H. Samet. 2009. Distance oracles for spatial networks. In *IEEE ICDE 2009*. 652–663.
[16] J. Sankaranarayanan and H. Samet. 2010. Query processing using distance oracles for spatial networks. *IEEE TKDE* 22, 8 (2010), 1158–1175.
[17] J. Sankaranarayanan and H. Samet. 2010. Roads belong in databases. IEEE Data Engineering Bulletin 33, 2 (2010), 4–11.
[18] J. Sankaranarayanan, H. Samet, and H. Alborzi. 2009. Path oracles for spatial networks. *Proc. VLDB Endow.* 2, 1 (Aug. 2009), 1210–1221.
[19] Y. Zhang, Y. Sun, L. Padilla, S. Barua, E. Bertini, and A. G. Parker. 2021. Mapping the landscape of COVID-19 crisis visualizations. In *CHI 2021* (Yokohama, Japan) *(CHI '21)*. Article 608, 23 pages.