Motorch: A Mobile Trajectory Data Management System

Chen Wu[1], Sheng Wang[1], Zhiyong Peng[1,2]
[1]School of Computer Science, Wuhan University
[2]Big Data Institute, Wuhan University
chenwu@whu.edu.cn,swangcs@whu.edu.cn,peng@whu.edu.cn

ABSTRACT

In this paper, we would like to demonstrate a personal trajectory management system called Motorch, which allows users to manage their trajectories and enables risk analytics based on a lightweight similarity measure called LCTS. At the back end, a web crawler collects the desensitized COVID-19 cases information from data sources (news, social media, etc.) and pushes them to Elasticsearch for storage after data cleaning. At the front end, Motorch implements a set of operations including data collection, data preprocessing, indexing, storage, and visualization in a mobile application. Motorch aims to help individuals manipulate their data and evaluate personal risk without uploading data to a server.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**.

KEYWORDS

spatio-temporal data mining, interactive exploration, contact risk

ACM Reference Format:
Chen Wu, Sheng Wang, Zhiyong Peng. 2022. Motorch: A Mobile Trajectory Data Management System. In *The 30th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '22), November 1–4, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3557915.3560949

1 INTRODUCTION

With the rapid development of mobile internet and locate technologies, human trajectory data are collected by various devices like cellular phones or GPS. Mobility analysis has helped prevent the outbreaks of COVID-19. One direct application is contact tracing [4–6, 8] which aims to detect whether people have come into direct contact with patients. However, existing solutions require users to upload their trajectory data and complete analysis on the server-side, which is unacceptable for users due to privacy considerations. This motivates us to develop a convenient and trustworthy management system for personal trajectory data.

This paper demonstrates a trajectory management system, Motorch, which consists of two modules, a mobile application for personal trajectory data management and a server for COVID-19 cases (short as cases) collection and retrieval. In the server side, we employ a web crawler to extract real-time spatio-temporal trajectories of cases reported by health departments. In the mobile application, users can record daily trajectory data with a GPS module and store them in a local database. To support efficient risk analysis over personal trajectories, we propose a new measure called *longest companion time similarity* (LCTS) to evaluate the risk based on a simplified trajectory representation.

Compared with existing trajectory search engines running on the server-side such as SALON [7], Torch [11], and T4 [12], the advantage of Motorch is that it achieves personal trajectory data management, including collecting, processing, and storage in mobile devices, and reduces the storage burden on the server-side. Moreover, users can request cases from the server by spatio-temporal conditions and evaluate personal risk between cases and local trajectory data. In this way, users can manipulate their data without uploading it to a server for similarity analysis, which is convenient and trustworthy in daily life.

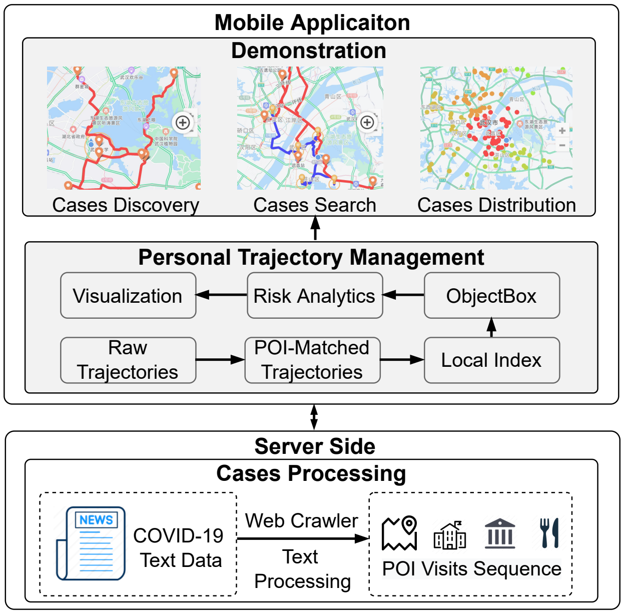


Figure 1: System overview of Motorch

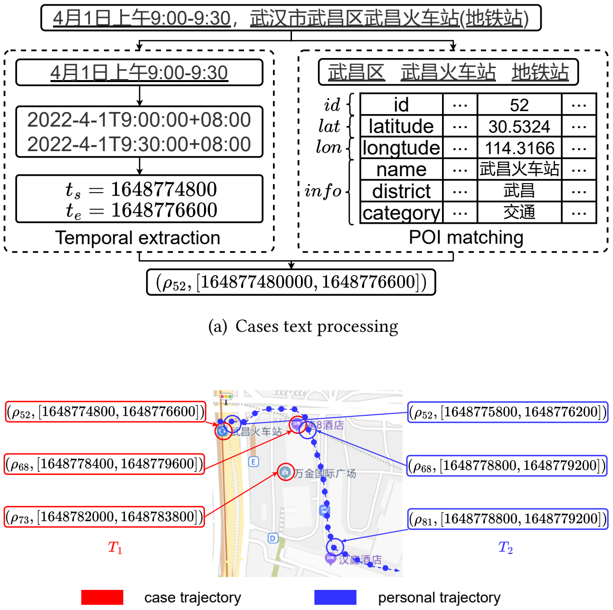
2 SYSTEM OVERVIEW

As shown in Figure 1, Motorch includes a server for cases processing and a mobile application for personal trajectory management. In the server, we utilize a web crawler to collect COVID-19 text data from news websites and extract visits over POIs (Point of Interests). In the application, raw trajectories are collected by a GPS module and transformed into POI-matched trajectories (defined in Section 3.1). The application builds a local index for personal

SIGSPATIAL '22, November 1–4, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9529-8/22/11.
https://doi.org/10.1145/3557915.3560949

trajectory data retrieval and stores these data in an object-oriented database ObjectBox [1]. Motorch supports interactive query over cases by setting various spatio-temporal conditions and efficient risk analytics based on historical trajectory data. In summary, Motorch is efficient and reliable to fulfill the requirements of personal trajectory management during pandemics.



(a) Cases text processing

$(\rho_{52}, [1648774800, 1648776600])$
$(\rho_{68}, [1648778400, 1648779600])$
$(\rho_{73}, [1648782000, 1648783800])$
T_1
武昌火车站
8酒店
万金国际广场
汉庭酒店
E
D
$(\rho_{52}, [1648775800, 1648776200])$
$(\rho_{68}, [1648778800, 1648779200])$
$(\rho_{81}, [1648778800, 1648779200])$
T_2
🟥 case trajectory
🟦 personal trajectory

(b) Contact risk evaluation

Figure 2: Examples of text processing and contact risk evaluation

3 BACK-END TECHNIQUES

This section first describes how to model the data, including the cases collected by the web crawler and personal data held by mobile devices. Then, we introduce more details about data cleaning, indexing, and storage at the server and mobile application.

3.1 Data Modeling

Definition 1 (GPS Point). A GPS point $p = (p.lat, p.lon, p.t)$ includes latitude $p.lat$, longitude $p.lon$ and timestamp $p.t$.

Definition 2 (Point of Interests (POI)). A POI is a location point $\rho = (id, lat, lon, info)$, where id is a unique identifier, lat and lon represent the latitude and longitude, and $info$ contains extra information including name, district and category.

Definition 3 (Visit). A visit is a tuple $\pi = (\rho, \tau)$ which represents an indoor stay in POI ρ during the time interval $\tau = [t_s, t_e]$.

Definition 4 (Raw Trajectory). A raw trajectory T consists of a set of points $\{p_1, p_2, \cdots, p_n\}$ in ascending order by time.

Definition 5 (POI-Matched Trajectory). A POI-matched trajectory $\bar{T} = \{\pi_1, \pi_2, \cdots, \pi_n\}$ is a set of visits in ascending order by time.

Definition 6 (LCTS). The Longest Companion Time Similarity is defined as:

$$\bar{M}(\bar{T}_1, \bar{T}_2) = \sum_{\rho \in P_{\bar{T}_1} \cap P_{\bar{T}_2}} |l^{\rho}_{\bar{T}_1} \cap l^{\rho}_{\bar{T}_2}| \tag{1}$$

where $P_{\bar{T}_1}$ is a set of POIs visited in $\bar{T}_1$, $l^{\rho}_{\bar{T}_1}$ is an interval list of visits at ρ in $\bar{T}_1$, and $|l^{\rho}_{\bar{T}_1} \cap l^{\rho}_{\bar{T}_2}|$ represents the overlapped time of two interval lists in POI ρ.

In Motorch, we utilize LCTS to accumulate common indoor stay time as the contact risk. Meanwhile, POI-matched trajectory is a lightweight representation for data modeling and storage.

3.2 Cases Management

Text Processing. In Motorch, we create a crawler program to collect COVID-19 information from Weibo[1] in Wuhan, where the health department reports daily COVID-19 cases, including temporal information and spatial locations. Figure 2(a) represents an example of text processing for a reported case. Firstly, we split the raw text into temporal and spatial parts. The temporal text will be completed in the format of China Standard Time and transformed into timestamps. For the spatial text, we further extract location name, district and category to match it with the POI dataset. Finally, the two parts are combined together as a visit to the matched POI.
Storage and Cases Query. We employ Elasticsearch [2] as the back-end storage engine for processed cases since it supports efficient spatio-temporal query, keyword query with low query latency due to the efficient index mechanism.

3.3 Personal Data Management

Trajectory Pre-processing. In Motorch, we develop a mobile application to collect raw trajectories and transform them into POI-matched trajectories, similar to stay point detection [9, 10]. As shown in Figure 3, we implement a two-phase processing algorithm. Firstly, the raw trajectories are clustered into stay points using a clustering-based algorithm [9]. Then, we mark POIs around these stay points within a distance threshold r (default 50m by our experience) as visited and assign associated time intervals.
Storage. We employ ObjectBox [1] as local storage engine in our mobile application, where ObjectBox is a lightweight, object-oriented NoSQL database for edge devices and well-suitable for complex data modeling. As shown in Figure 4, we design our data modeling as classes in ObjectBox and maintain their reference relations such as "one-to-one" or "one-to-many". To support efficient retrieval over personal trajectories, we build a grid index for spatial query and an interval tree index[2] for temporal query.
Contact Risk Evaluation. Figure 2(b) presents an example of contact risk evaluation between a case trajectory T_1 and a personal trajectory T_2. We have that $P_{\bar{T}_1} = \{\rho_{52}, \rho_{68}, \rho_{73}\}$ and $P_{\bar{T}_2} = \{\rho_{52}, \rho_{68}, \rho_{81}\}$. The common visited POIs are $P_{\bar{T}_1} \cap P_{\bar{T}_2} = \{\rho_{52}, \rho_{68}\}$. Then we calculate overlapped time intervals on these two POIs, that is, 800 seconds as the similarity. In a real scenario, Motorch will request relevant cases from the server side by a coarse-grained spatial condition for the sake of privacy, e.g., a rectangle region not an exact POI location, and evaluate contact risk locally.

[1] http://m.weibo.cn/
[2] https://github.com/lodborg/interval-tree

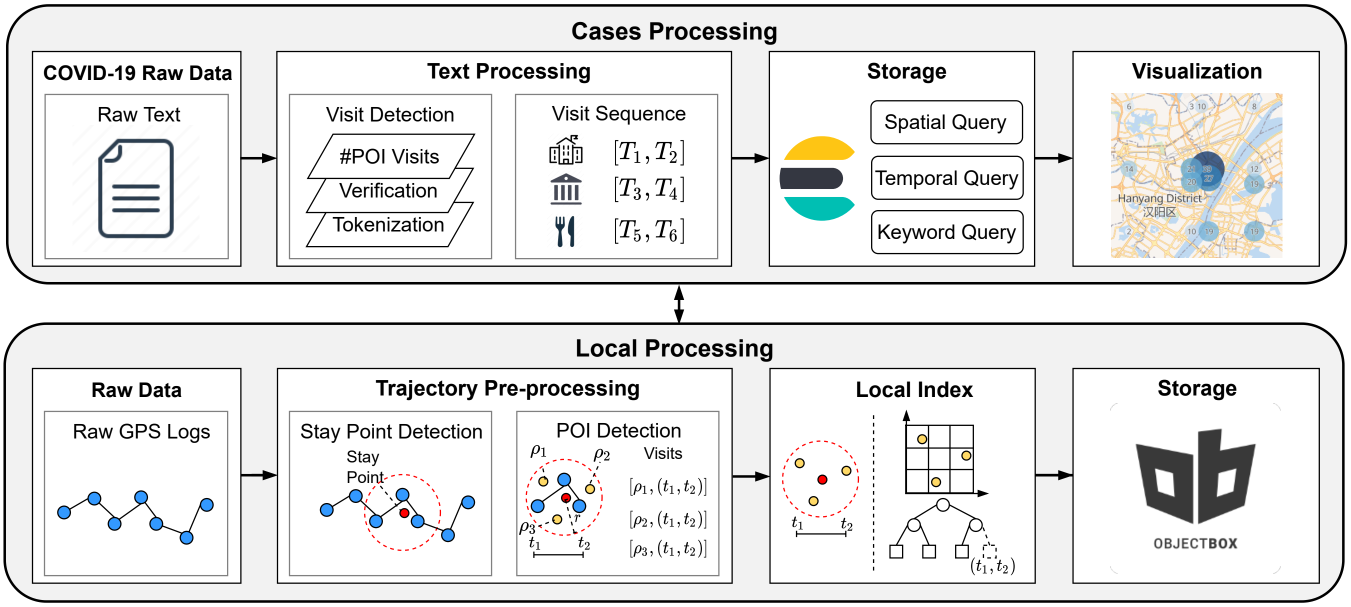


Figure 3: Trajectory processing in Motorch

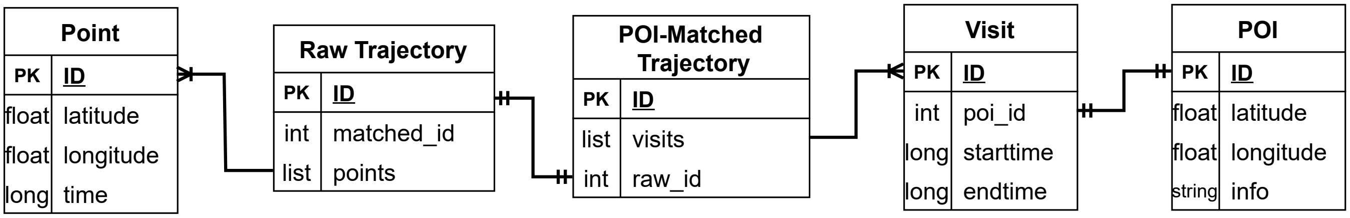


Figure 4: Trajectory modeling in ObjectBox

4 DEMONSTRATION

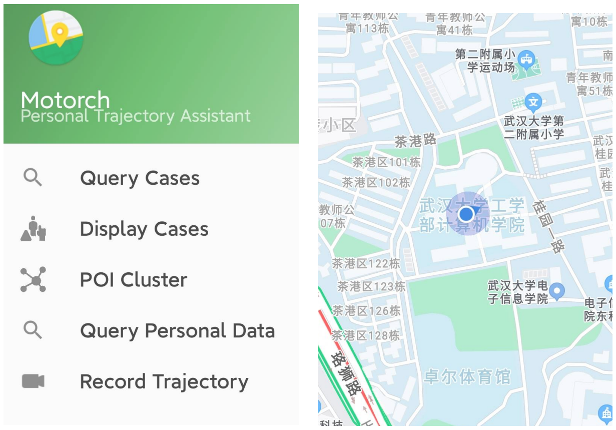
We will demonstrate Motorch based on real-world cases in Wuhan collected in April 2022. More than 200,000 POIs in Wuhan are stored in Motorch client and server. Motorch supports spatio-temporal search, keyword search, and similarity search. Motorch has been deployed in a server and an Android mobile phone (Redmi Note 7). Motorch is efficient with an average latency of less than 1s and our program can support 100,000 devices in simulated experiments.

4.1 Overview of User Interface

We present user interfaces of the client in Figure 5. There is a map view based on a location service called Amap [3] in the initial startup for trajectory visualization. When users slide from the left side of the screen, a side menu navigates users to other views for different functionalities. Below we will describe three scenarios to further elaborate on how Motorch works when users want to explore personal trajectories and cases.

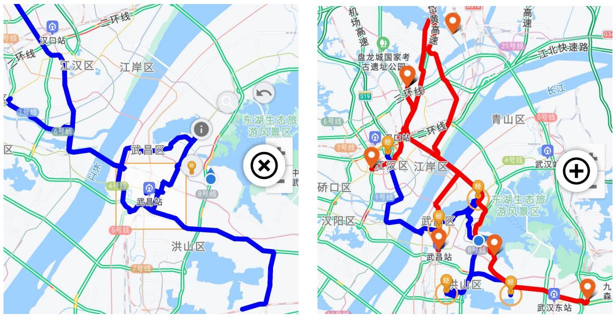
4.2 Scenario 1: Cases Discovery

To search personal historical trajectories, users can form a query by drawing a rectangle on the map view. Besides the spatial ranges, users also can query personal trajectories by location names or temporal conditions in the search view. Figure 6(a) shows an exemplar spatial query over personal trajectories in blue lines.



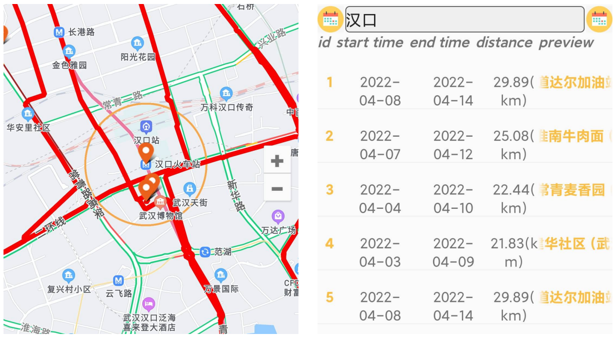
(a) Side menu (b) Map view

Figure 5: Main user interface of Motorch client



(a) Personal trajectory query (b) Top-k query over cases

Figure 6: Example of cases discovery



汉口

id	start time	end time	distance	preview
1	2022-04-08	2022-04-14	29.89(km)	达尔加油
2	2022-04-07	2022-04-12	25.08(km)	南牛肉面
3	2022-04-04	2022-04-10	22.44(km)	青麦香园
4	2022-04-03	2022-04-09	21.83(km)	华社区（武
5	2022-04-08	2022-04-14	29.89(km)	达尔加油

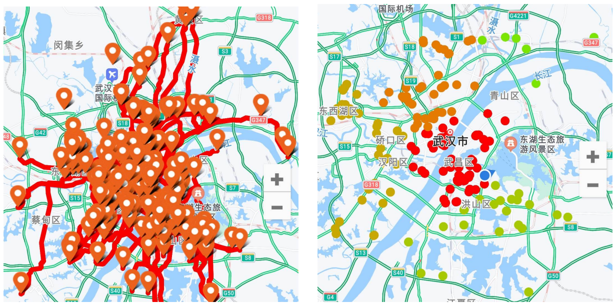
(a) Spatial search over cases (b) Keyword search over cases

Figure 7: Example of cases search

Once there are cases in the city, users may want to know potential contact between personal historical trajectories and cases. Motorch enables users to search related cases using LCTS. In Figure 6(b), users choose a personal trajectory and set the value of k as three, Motorch presents the three most related cases in red lines and adds markers to visited POIs. Users can further click these markers and observe more time details in the below bar, and the orange circles indicate the contact locations.

4.3 Scenarios 2: Cases Search

For the sake of safety, people tend to avoid visiting the locations where cases happen. It is necessary for users to search cases around target destinations and make traveling plans during a pandemic. Motorch enables users to search cases by location names or spatio-temporal conditions. In Figure 7, users plan to travel by train, so they need to know the cases around the railway stations. They click a station and specify the distance as 600m, then Motorch displays related cases in red lines and reveals the visited POIs in the orange circles. Users can search the cases by temporal conditions and location names in the search view. Motorch will display the details of cases, including the time, lengths, and visited POIs. Users have to change their traveling plans due to these cases.



(a) Cases movement (b) K-means clustering over cases

Figure 8: Example of cases spatial distribution

4.4 Scenarios 3: Cases Distribution

There may be numerous cases when the pandemic deteriorates. Governments may take the stay-at-home order to restrict the movement of people to control the spread of COVID-19. When newly detected cases decrease to zero, the locked region can be unblocked. In this case, users may want to know the distribution of the cases, especially around their living regions. Note that cases are a set of visits over POIs which are represented as spatial points, and Motorch supports k-means clustering over cases. In Figure 8(a), Motorch displays all the cases with detailed movement information. As shown in Figure 8(b), users choose k as five, and Motorch displays the results in different colors, where the brightness of colors indicates the cases number in the clustered regions.

ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China (No. U1811263).

REFERENCES

[1] 2017. Fast lightweight Java Database for storing and syncing objects in Mobile IoT. https://github.com/objectbox/objectbox-java
[2] 2019. Free and Open, Distributed, RESTful Search Engine. https://github.com/elastic/elasticsearch
[3] 2021. Amap is a leading domestic technology LBS service provider. https://lbs.amap.com/.
[4] Mohammed Eunus Ali, Shadman Saqib Eusuf, and Kazi Ashik Islam. 2020. An Efficient Index for Contact Tracing Query in a Large Spatio-Temporal Database. (2020). http://arxiv.org/abs/2006.12812
[5] Edward Buckland, Egemen Tanin, Nicholas Geard, Cameron Zachreson, Hairuo Xie, and Hanan Samet. 2021. Managing Trajectories and Interactions During a Pandemic. *SIGSPATIAL*, 423–426.
[6] Pingfu Chao, Dan He, Lei Li, Mengxuan Zhang, and Xiaofang Zhou. 2021. Efficient Trajectory Contact Query Processing. *DASFAA*, 658–666.
[7] Yue Hu, Sijie Ruan, Yuting Ni, Huajun He, Jie Bao, Ruiyuan Li, and Yu Zheng. 2021. SALON: A Universal Stay Point-Based Location AnalysisPlatform. *SIGSPATIAL*, 407–410.
[8] Fumiyuki Kato, Yang Cao, and Mastoshi Yoshikawa. 2020. PCT-TEE: Trajectory-based Private Contact Tracing System with Trusted Execution Environment. *ACM Transactions on Spatial Algorithms and Systems* 8, 2 (2020), 1–35.
[9] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma. 2008. Mining User Similarity Based on Location History. *SIGSPATIAL*, 1–10.
[10] Rafael Pérez-Torres, César Torres-Huitzil, and Hiram Galeana-Zapién. 2016. Full on-device stay points detection in smartphones for location-based mobile applications. *Sensors* 16, 10 (2016).
[11] Sheng Wang, Zhifeng Bao, J. Shane Culpepper, Zizhe Xie, Qizhi Liu, and Xiaolin Qin. 2018. Torch: A search engine for trajectory data. *SIGIR*, 535–544.
[12] Sheng Wang, Yunzhuang Shen, Zhifeng Bao, and Xiaolin Qin. 2019. Intelligent traffic analytics: From monitoring to controlling. *WSDM*, 778–781.