A Stream Data Service Framework for Real-Time Vehicle Companion Discovery

Zhongmei Zhang[1(✉)] and Shuai Zhang[2]

[1] School of Management Engineering, Shandong Jianzhu University, Jinan 250101, Shandong, China
zhangzhongmei20@sdjzu.edu.cn

[2] Jinan Branch of China Unicom Software Research Institute, Jinan 250100, Shandong, China
zhangs837@chinaunicom.cn

Abstract. Because of the large-scale and complexity nature of vehicle trajectory data, existing methods have struggled to guarantee the efficiency and effectiveness of vehicle companion discovery in real-time. This paper proposes a stream data service framework to real -time discover vehicle companion. It relaxes the spatial and temporal constraints of vehicle companion definition to find more potential companion vehicles. And to ensure the performance, we make use of flexible service collaboration to process data generated by related monitoring sites selectively, and employ a stream partition strategy to realize service collaboration in parallel. Experiments conducted with real and simulated data demonstrate that our method can identify a greater number of potential vehicle companion sets while requiring less transmission and processing resources.

Keywords: Stream Data Service · Trajectory Data · Vehicle Companion · Service Collaboration

1 Introduction

Recently, with the rapid development of GPS, Mobile Smart Devices, RFID and other technologies, a vast amount of trajectory data with space-time attributes have been produced, capable of recording the movement of vehicles, people, animals and natural phenomena in social networks, traffic management, scientific research and military surveillance and other applications. Researchers expect to find various companions in moving groups [1–4] based on processing trajectory data. The study of vehicle companion can discover the movement patterns of vehicles that travel together within specific time range, which can provide good solutions for accident investigation [5], group tracking [6], and carpooling recommendation [7].

Currently, the companion vehicle discovery problem is mainly based on two kinds of vehicle companion's definition: (1) time point-based companion [8–14], which is applied to Global Positioning System (GPS) trajectory data. As GPS equipment may be only installed in limited equipment, and may not satisfy specific mining requirements.

A. Zaslavsky et al. (Eds.): MobiQuitous 2023, LNICST 593, pp. 281–296, 2024.
https://doi.org/10.1007/978-3-031-63989-0_14

(2) spatial point-based companion [7, 15–17], which is applied to Automatic Number Plate Recognition (ANPR) data. Contrasted with GPS data, ANPR data offers the capability for continuous monitoring across significantly larger areas, potentially spanning entire cities. To summarize, existing methods are limited to adjusting either temporal or spatial constraints for vehicle companion, potentially overlooking companion vehicles that fall within broader constraints and thereby impacting the overall completeness of identified companions. Besides, existing work primarily focusses on static data sets and rely on centralized methodologies, which results an increase in communication and computational costs. Hence, it is quite important to provide a high-quality and low-cost method to discover vehicle companion in real-time based on amount of trajectory data stream.

Due to the reusability and interoperability of service, many researchers [18–20] have integrated Service-Oriented Architecture (SOA) technologies into Internet of Thing systems to decouple low-level streaming data from upper-layer applications. The servitization of trajectory data can help developers discretely process trajectory data through flexible management of software components. In our previous studies [21, 22], we introduced a proactive data service as a form of service abstraction, enabling flexible service collaboration in the context of dynamic stream data.

For discovering more complete companion vehicles, we proposed a new vehicle companion definition in our work [23]. In this paper, we aim to propose a stream data service framework based on ANPR data to process real-time trajectory data more flexibly and efficiently. We provide a service collaboration-based method to find vehicle companion, and examine a distributed strategy to realize service collaboration in parallel. We firstly encapsulate trajectory data produced by each monitoring site into fine-grained services, then build more powerful services based on dynamic service collaboration, and generate the vehicle companion sets in gradually and discrete way. In summary, this paper has followed contributions:

- we redefine vehicle companion by loosening the spatial constraints based on existing spatial point-based companion, resolve the issue of vehicle redundancy, and can find more potential companion vehicles.
- instead of accessing and processing all trajectory stream data, our proposed service collaboration-based method dynamically accesses and processes relative trajectory data based on the vehicles' real-time status. And it can ensure efficiency due to largely reducing of the data processing volume.
- to further increase the efficiency of our method, the service framework employs a stream partition strategy and run the service collaboration in a distributed processing environment. It can divide monitoring sites into partitions and realize vehicle companion discovery in parallel.

The rest of the paper is structured as follows: Sect. 2 introduces related work. Section 3 presents a motivational scenario and definitions of the problem. Section 4 introduces our method for vehicle companion discovery and outlines our parallel framework. Section 5 presents the results of our experiments. Lastly, in Sect. 6, we conclude and discuss avenues for future research.

2 Related Work

2.1 Stream Data Service

With the fast development of stream data researches, researchers have explored integrating SOA [18–20] technology into stream data applications, leading to two kinds of approaches based on the core concept of service models.

One kind of approach center on treating "resource" as the core concept [24–27], where sensors and their generated streams are viewed as shareable resources with the external world. Many studies [26, 27] have aimed at leveraging lightweight Web service protocols to enable the servitization of sensor data, effectively transitioning IoT into the WoT (Web of Things). These works realize sharing and interoperability of sensor stream data by using open Web standards. The other approach [7, 28–30] takes "function" as the core concept, focusing on providing a service model centered around stream data processing. These studies encapsulate stream data processing functions into services, allowing applications to readily utilize general or complex stream data processing capabilities through service invocation.

Overall, existing works in stream data servitization have proven effective in enhancing the sharing and reusability of stream data across diverse user scenarios. The encapsulation of stream data processing capabilities streamlines the development of complex application systems by enabling efficient service composition and collaboration. Moreover, adapting service composition processes to dynamic operational environments or data requirements has been addressed [31–33], aiming to alleviate user burdens and actively manage complex and dynamic data processing demands.

2.2 Companion Patterns Discovery

Many works were proposed obvious definitions and method to explore companion pattern. There are some typical companion pattern definitions such as Flock [8, 9], Convoy [10, 11], Swarm [12, 13], Platoon [14, 15], etc. The Flock [8, 9] definition was among the earliest proposed companion pattern definitions, specifying that companion sets must appear simultaneously within a circular geographic space across continues time points. Convoy [10, 11] extended the spatial constraints to include density-reachable geographic areas, building upon the foundation established by Flock. Swarm and Group [12, 13] relaxed Convoy's time constraint, enabling companion sets to be simultaneously present in density-reachable geographic spaces at multiple discrete time points. Platoon [14, 15] further refined the time constraints by specifying thresholds for the number of time points during which companion groups co-occurred, as well as the number of consecutive time points during which companion sets co-occurred. Platoon required the companion sets to co-occur in density-reachable geographic spaces during enough continuously time points. These works were focused on companion patterns at specific time points, and were more suitable for stream generated by floating sensors, such as GPS data.

In addition to the above definitions, many studies [7, 16, 17] were focused on data produced by fixed sensors, and regarded moving objects that occur successively or simultaneously at the same spatial point within a given time range as site companion. Based

on the site companion, moving objects occurred in multiple site companion relations are considered as a companion set.

In recent years, many work tried to focus on the efficiently of discovering co-occurrence patterns from large-scale of trajectory data. Zhu et al. [15] introduced the PlatoonFinder algorithm for real-time mining the Platoon companion patterns from automatic number plate recognition data streams, which adopted pseudo-projection technologies for effectively coping with the speed and scale of data streams. Xiao et al. [17] integrated streaming data with dynamic graph calculations to identify vehicle companion in real-time scenarios. Zhang et al. [34] introduced a distributed framework for mining companion patterns from data streams, aiming to enhance the discovery capability and performance. Currently, most of existing research is centralized, requiring substantial processing of stream data.

The ANPR data focused on this paper was generated by fixed traffic camera sits. Therefore, we aim to propose a stream data service framework in the basic of improved site companion, and we try to encapsulate traffic stream data into proactive data services, and realize dynamic service collaboration to efficiently find vehicle companion sets in parallel.

3 Scenario and Problem Definition

We firstly introduce a typical scenario about vehicle companion discovery.

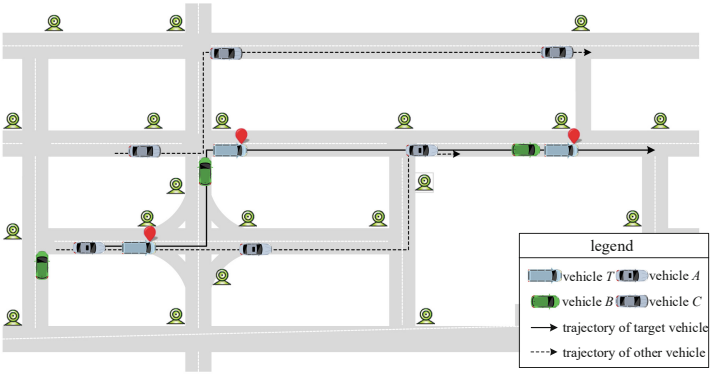


Fig. 1. Examples of vehicle companions based on ANPR data.

Crimes special targeting vehicles such as bus, taxis, armored car, and vehicles transporting dangerous or important goods have become increasingly prevalent. By promptly detecting suspicious tracking vehicles and alerting the tracked vehicle, occurrences of such crimes can be effectively prevented when a specific vehicle passes through a traffic camera. Figure 1 illustrates some examples of vehicle companions. The discovery of the vehicle companions for target vehicles in real time serves as an effective way to identify suspicious tracking vehicles.

In existing works, vehicles that pass the same site in given time range with target vehicle (vehicle T) several times can be identified as its companion vehicles. Figure 1 shows that vehicle B and target vehicle T passed by the same sites within a close time range, indicating identical trajectories. While in practice, tracking vehicles often opt not to follow the exact same trajectory as the target vehicle to avoid easy detection. For example, although vehicle A and vehicle C in Fig. 1 had different trajectories with target vehicle T, the monitoring sites they passed through are adjacent to the sites that vehicle T passed through, making them potential suspicious tracking vehicles. Therefore, solely relying on identical sites as the criterion for identifying vehicle companions may not fully uncover all companion vehicles.

For avoiding missing some potential companion vehicles, we redefine vehicle companion pattern by loosening the spatial constraint of spatial point-based companion pattern. We firstly give a definition about area companion based on site companion [7], which indicates vehicles passed through exactly the same monitoring site or spatial point within given time range.

Definition 1. Area Companion. An areas companion set a_k can be represented as $a_k = \{<v, c, t>\}$, is as set of triples including vehicle v, the monitoring site c passed by v, and the time point t that vehicle v passed monitor site c. And for any two triples $<v_i, c_i, t_i>$ and $<v_j, c_j, t_j>$ in a_k, exists $|t_i - t_j| <= \delta_t$, where δ_t is a given time range threshold, and c_i and c_j have adjacent relationship.

Figure 2 show some example of site companion and area companion. In Fig. 2, five monitoring sites are shown as the vertical axis, and it can be seen that there is a site companion set $\{v_1, v_2, v_3\}$ at monitoring site A. While area companion refers to vehicles passing nearby monitoring sites at similar time points. Based on Definition 1, if monitoring site A and B in Fig. 2 have adjacent relations, vehicles $\{v_5, v_7\}$ and $\{v_1, v_2, v_3\}$ will also have companion relation.

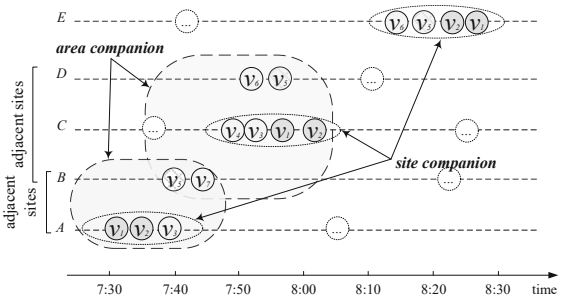


Fig. 2. Examples of Site Companion, Area Companion and Vehicle Companion

It is important to note that because of the loosen of temporal and spatial constraints in area companion, redundancy of vehicles and monitoring sites may occur across different sets of area companions. For example, v_5 appears twice with different sites in one area companion as shown in Fig. 2.

Vehicles that occurred in the same area companion several times can be regard as vehicle companions.

Definition 2. Vehicle Companion. An vehicle companion set means a set $vc = \{v\}$, for two vehicles v_i and v_j in it, their companion degree $cd(v_i, v_j) <= \delta_{com}$, where

- $cd(v_i, v_j) = \sum p(c_{ik}, c_{jk})/max(n_i, n_j)$, in which n_i and n_j are the numbers of monitoring sites passed by vehicle v_i and v_j, and $p(c_{ik}, c_{jk})$ is the adjacent degree of the monitoring sites when they appear in the k-th area companion.
- $\delta_{com} \in (0,1]$ is a given threshold.

For a given vehicle, we can obtain more complete companion vehicle set based on above definitions. Taking Fig. 2 as an example, assuming that monitoring sites $\{A, B\}$ have respectively adjacent relationships with sites $\{C, D\}$, and giving δ_{com} as 0.5, for target vehicle v_1, sets $\{v_2, v_3, v_5, v_7\}$, $\{v_2, v_3, v_4, v_5, v_6\}$ and $\{v_2, v_5, v_6\}$ are all area companion sets with v_1. And the vehicles have vehicle companion relationship with v_1 are in set $\{v_2, v_3, v_5, v_6\}$.

Traditional methods predominantly employed a centralized processing way to identify vehicle companions. These methods involved gathering vast amounts of stream data into a single data center, followed by analysis and processing rely on various discovery methods and companion definitions, and required significant transmission and processing resources. Therefore, this paper aims to address the challenge of efficiently processing stream data while ensuring the effectiveness and performance of vehicle companion discovery.

4 Method

Based on the definition of vehicle companion introduced above, we firstly introduce our service collaboration method to flexibly and efficiently find companion vehicle for given vehicle. Then we introduce our parallel strategy in the service framework which can further ensure the performance of our method.

4.1 Vehicle Companion Discovery Based on Service Collaboration

Figure 3 illustrates the schematic of our service collaboration-based method. This paper creates three types of stream data services, i.e., Site Companion Discovery (SCD) services that corresponding to the monitoring sites, several Area Companion Discovery (ACD) services, and one Vehicle Companion Discovery (VCD) service.

To be specific, SCD service is mainly in charge of monitoring of target vehicle and generate site companion vehicles in real-time. ACD service is in charge of receiving site companion results from SCD services and discovering area companion results. And VCD service receives trajectory information from SCD services to obtain the completed trajectory, and receives area companion information from corresponding ACD services to generate final vehicle companion sets.

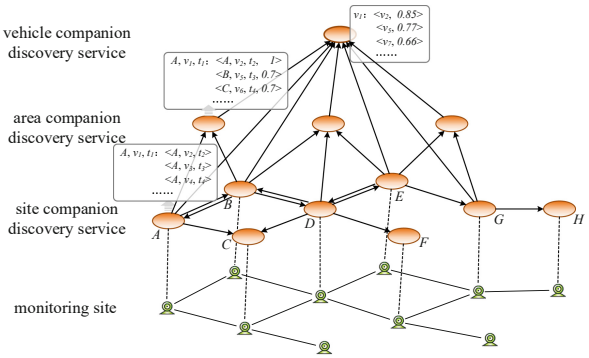


Fig. 3. Overview of vehicle companion discovery method.

When an SCD service monitors that given vehicle passed the service's corresponding site, it will target a series of actions, i.e., the site companion discovery function of this service, starting of SCD services corresponding to nearby monitoring sites, and starting one ACD service. The detail execution process of each SCD service is shown in Algorithm 1.

Algorithm 1. execution process of SCD service.
Input:
given vehicle v_i;
time threshold δ_t;
spatial threshold δ_s.
Output:
vehicle v_i discovery event e_i.

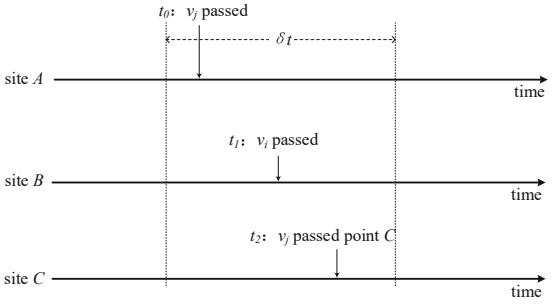
1. start procedure *VehicleMonitor*(v_i);
2. if discover e_i
3. start procedure *SiteCompanionMonitor*(e_i, δ_t);
4. get neighbor ACD services based on δ_s;
5. send e_i , δ_t, δ_s to neighbor SCD services;
6. send e_i to ACD service;
7. send e_i to VCD service;
8. stop procedure *VehicleMonitor*(v_i);
9. end if

For reducing the volume of trajectory data to process, at the initial stage, it needs all SCD services to monitor target vehicle v_i only (line 1). Once one service discovers vehicle v_i is passed by, an event e_i which contain vehicle's id, site's id and passed time is generated (line 2), and the area companion discovery process *SiteCompanionMonitor* will be started (line 3). Each service will contain a buffer to obtain vehicles passed corresponding site in certain time range according δ_t and e_i, and the *SiteCompanionMonitor* will stop automatically at appointing time. Nearby SCD services will be targeted to continue monitor given vehicle and generate other site companions (line 5), and the number of SCD services is determined by spatial threshold (line 4). One ACD service and VCD

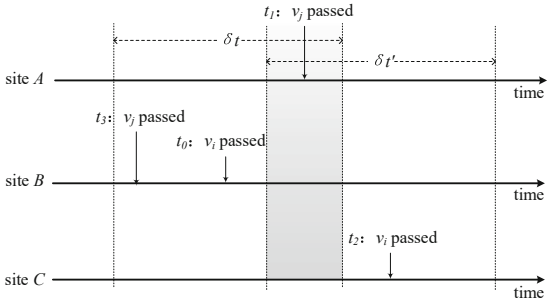
service are also targeted to further generated area companion and vehicle companion (line 6, 7). And the monitoring of vehicle v_i is no need (line 8).

To be mentioned, VCD service also in charge of stop all the monitoring procedure of irrelevant SCD services at the initial stage once one SCD services find target vehicle's first appearance. Except the procedure shown in Algorithm 1, each SCD service also executes two other procedures, i.e., monitoring events from VCD service that target vehicle is appeared, and keeping monitoring events from other SCD service to target the procedure of Algorithm 1.

One ACD service will be called once target vehicle passes one monitoring site, and it will collect all site companions from corresponding SCD services to generate area companion. As mentioned above, there may be redundancy in the area companion sets. Figure 4 shows some example of two types of area companion redundancy.



(a) redundancy in the same area companion



(b) redundancy in different area companions

Fig. 4. Examples of area companion redundancy.

As shown in Fig. 4(a), the first kind of redundancy is redundancy in the same area companion. In Fig. 4(a), monitoring sites A and C are adjacent sites of monitoring site B. When vehicle v_i passes through monitoring site B at t_1, all vehicles passing through monitoring site A and C within given time range threshold δt have area companion relationship with vehicle v_i, such as $\{<v_i, B, t_1>, <v_j, A, t_0>\}$. Since vehicle v_i also passed through site C within δt, there is also a companion relationship as $\{<v_i, B, t_1>, <v_j, C, t_2>\}$, which resulting repeating vehicle v_j in the same area companion.

Figure 4(b) shows one example of redundancy in different area companions. In Fig. 4(b), monitoring site A has respectively adjacent relationships with monitoring site B and set C. When vehicle v_i passed monitoring site B and site C respectively in time t_0 and t_2, there may be an interleaved time range between monitoring sites A and C as the dash area in Fig. 4(b). Vehicles passed by certain sites within this interleaved time range can both have area companion relationships with $<v_i, B, t_0>$ and $<v_i, C, t_2>$. For example, $<v_j, A, t_1>$ is both in area companion $\{<v_i, B, t_0>, <v_j, A, t_1>\}$ and $\{<v_i, C, t_2>, <v_j, A, t_1>\}$. To be mentioned, it is also happened that one vehicle is both belong to these two kinds of redundancy. For example, vehicle v_i in Fig. 4(b) is also shown in area companion $\{<v_i, B, t_0>, <v_j, A, t_2>, <v_j, A, t_3>\}$ twice.

The redundancy vehicles need to be removed for avoiding affection to vehicle companion discovery. However, the removing of vehicles may also affect the final result. Taking Fig. 4(b) as an example, if we remove $<v_j, A, t_3>$ from area companion $\{<v_j, A, t_2>, <v_j, A, t_2>, <v_j, A, t_3>\}$, there is still exists redundancy in area companion $\{<v_i, B, t_0>, <v_j, A, t_1>\}$ and $\{<v_i, C, t_2>, <v_j, A, t_1>\}$. And if we further remove $<v_j, A, t_1>$ from $<v_i, B, t_0>, <v_j, A, t_1>\}$. It results that the co-occurrence number of vehicle A and B is reduced to one from two. Hence, instead of removing redundancy of the same companion in ACD services, we keep all the redundancy and process them together in VCD service. To avoid above problem, if there are vehicles both belong to two kinds of redundancy, we firstly remove ones that in different area companions.

In summary, our method leverages the dynamic collaboration of SCD and ACD services to extract trajectory and site companion sets for a given vehicle by processing only relevant stream data, and send the information to the VCD service. Based on Definition 2, the VCD service can eliminates redundancy in area companionship, generates vehicle companion sets in real-time by receiving trajectory and area companion information, and continuously provides them as output.

4.2 Parallel Framework

Figure 5 shows our framework that can create proactive data service, and support dynamically service collaboration for satisfying users' requirements.

To shield users from the details about the data stream source and offer various representations, our framework adopts the Representational State Transfer (REST) architectural style. Since traditional Web services, with their "request-and-response" pattern, may not effectively facilitate developers in retrieving data streams. We utilize a Web push technology, i.e., Web Socket, and define declarative rules to realize stream data service based on our proposed proactive data service abstract [20, 21]. With Web Socket, stream data service can proactively transmit events containing data records to users over the web. By utilizing predefined declarative rules, users can dynamically get events that satisfy their specific requirements.

For ensuring the performance of our service and service composition, we utilize could environment and design a partitioning strategy to process the stream data service in parallel. We employ a straightforward approach of dividing the monitoring sites into several partitions and processing the corresponding services in parallel. However, several considerations must be taken into account when dividing the sites into partitions. Firstly, to prevent data skew among worker nodes, it is essential to evenly distribute the data

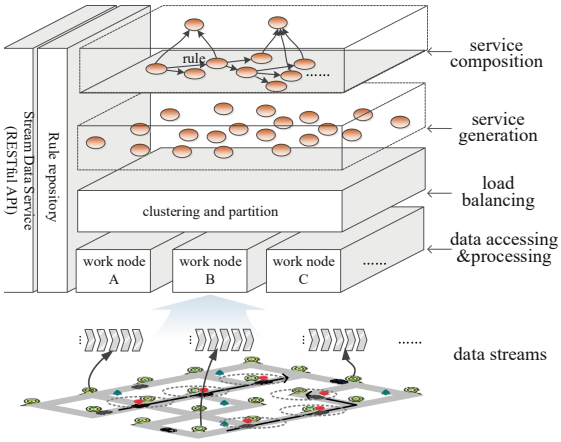


Fig. 5. The architecture of proposed framework.

volume across each partition. Secondly, to minimize communication overhead between workers, neighboring monitoring sites should ideally be grouped within the same set. For solving the above issues, this paper proposes a stream partition strategy and realized it based on the improvement of the Number Partition Problem (NPP) [35].

Definition 3. Stream Partition Strategy. Given a data stream S produced by a set of monitoring sites, the stream partition strategy sp can be regarded as a partition function $sp(S) = \{S_1, S_2 \ldots S_k\}$, s.t.

(1) $S = \bigcup_{i=1\ldots k} S_i$;
(2) $S_i \cap S_j = \emptyset, i, j = 1 \ldots k \cap i \neq j$;
(3) For any data stream $s \in S_i$, $\exists s' \in S_i$ and s' is the data generated by neighboring site which directly connect with the site generated s;
(4) min $\left(\sum_{i,j=1\ldots k}^{k} \left(|S_i| - \left|S_j\right|\right)\right)$, where $|S_i|$ means the number of data streams in set S_i.

It should be noted that condition (3) and (4) may not always be satisfied simultaneously. In this paper, we prioritize satisfying condition (3), which can ensure that there are no isolated sites within one partition set, and then we satisfy condition (4) as possible to ensure the balance of different partitions. The relationship of SCD services can be represented by monitoring sites' relations. Firstly, we traverse the graph breadth-first, and then partition the services into k partitions.

5 Experiments

We will respectively evaluate the efficiency and effectiveness of our discovery method by comparing with PlatoonFinder [15], which was realized based on spatial point companion and adopted centralized processing method.

5.1 Experiment Setup

1. **Datasets**
   The datasets used in our experiment is the real ANPR data from municipal traffic administration bureau of three large cities in China. In each data record, the data includes information about the vehicle, the monitoring site, the timestamp, and other additional data. The detail of data sets is show in Table 1. In the experiment, we convert real ANPR data into streaming data based on the generation time of each record, enabling real-time discovery of vehicle companions.

Table 1. Detail of data sets

datasets	Number of monitoring sites	Number of vehicles	Size
City 1	1794	2.3 million	25.7 GB
City 2	1040	1.7 million	18.9 GB
City 3	985	1.6 million	17.4 GB

2. **Experiment environments**
   Our method is implemented in a cluster consisting of 5 nodes, each running on virtual machines with CentOS release 6.4 and java 1.80. Table 2 shows the detailed configuration information of our cluster.

Table 2. Configuration of the Cluster

IP of node	CPU	Memory
10.61.6.153	Intel Xeon E312xx (Sandy Bridge)	6G
10.61.6.154	Intel Xeon E312xx (Sandy Bridge)	6G
10.61.6.155	Intel Xeon E312xx (Sandy Bridge)	3G
10.61.6.156	Intel Xeon E312xx (Sandy Bridge)	3G
10.61.6.157	Intel Xeon E312xx (Sandy Bridge)	3G

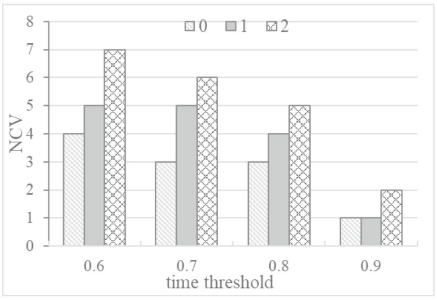
3. **Experiment criteria**
   We used (1) the Number of discovered Companion Vehicles (NCV) to verify the effectiveness; (2) Data Volume Processed per second (DVP) and (3) Average Response Time (ART) to verify the efficiency.

   The DVP in this paper means per second's received data volume from the stream data source side to the service side. And the ART is the average time difference between data sending time from data source side and the completion time of the data processing.

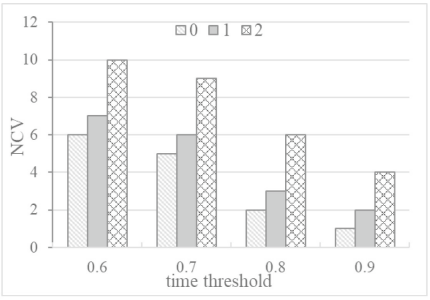
5.2 Effectiveness

Firstly, we conduct the discovery of vehicle companions using the same historical datasets, and compare the real-time discovered companion vehicles of our method by adjusting various thresholds. We randomly select 50 target vehicles from three data sets respectively, and set the time range threshold as 5 min, spatial threshold as 0, 1, 2, and the companion degree threshold as 0.6, 0.7, 0.8, 0.9. Figure 6 shows average NCV for three data sets.

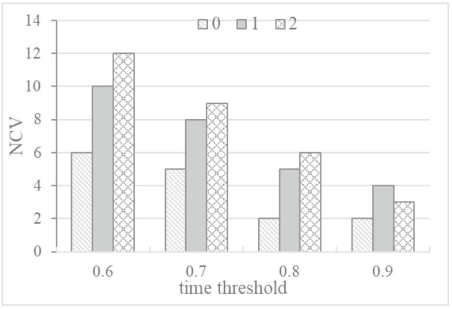
In fact, the results of our method shows that we can discover accurately the same companion vehicles with the results based on the same history trajectory data sets, which indicate that our real-time method can discover vehicle companion accurately. To simplify the figure, we only show one set of results in Fig. 6. It can also be observed that as the spatial threshold increases, more companion vehicles can be identified.



(a) city 1



(b) city 2



(c) city 3

Fig. 6. The average NCV for three data sets with different threshold.

Then, we compare our method with PlatoonFinder. We randomly selected 10 vehicles as target vehicles to respectively discover their vehicle companion sets. We respectively set the spatial threshold of our method as 2, the minimum companion degree threshold as 0.6, the time range threshold as 5 min, and record the average companion vehicles' numbers at given vehicles' 4, 6, 8, 10th passed monitoring sites to show continuously effectiveness of different methods.

Figure 7 illustrates the comparison of the detected companion vehicles' number of different methods. It is evident that our method consistently identifies more potential

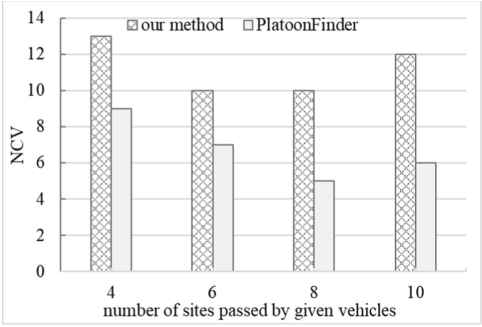


Fig. 7. Comparison of the companion vehicles' number.

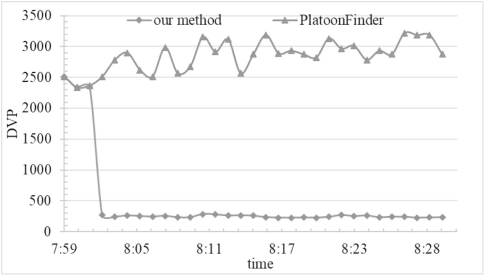
companion vehicles given the same vehicle movements. This is attributed to our method's relaxation of the spatial constraints for vehicle companionship, allowing for the discovery of vehicles passing through adjacent areas in given time range threshold. In essence, PlatoonFinder is equivalent to our method with a time threshold of 0.

5.3 Performance

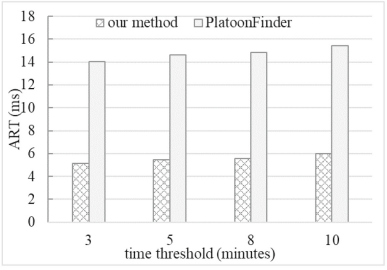
For evaluating the performance, we firstly compare our method with PlatoonFinder method. We set the minimum companion degree threshold as 0.6, spatial threshold as 2, the time range threshold as 8 min, and record the DVP of different methods continuously for half an hour. We also set time threshold as 3, 5, 8, 10 min, and calculate the ART of different methods.

Because our method is also supported running in parallel, then we evaluate the expandability of our method by adjusting degrees of parallelism. We simulated trajectory stream data with higher frequency as 1000 Hz, set the parallelism degree as 1, 3, and 5, and respectively calculate the ART.

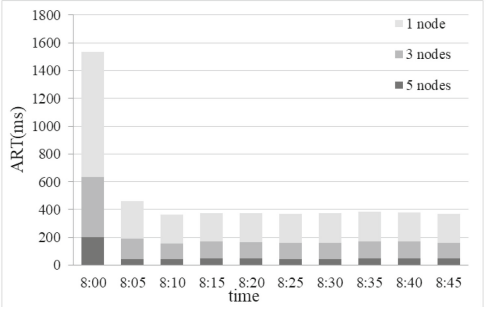
Figure 8 shows the results of the efficiency of our service collaboration-based method and PlatoonFinder method, and the efficiency of our method with different degree of parallelism. As shown in Fig. 8(a), the Data Volume Processed per second of PlatoonFinder method consistently remained high. Conversely, the DVP of our method initially peaked but then sharply declined after a certain period, stabilizing at a significantly lower level. This discrepancy arises from the centralized processing approach, which necessitates receiving ANPR data generated by all sites throughout the entire execution period. In contrast, our method only requires receiving all ANPR data at the beginning, resulting in improved efficiency over time. Once the target vehicle is discovered, it only needs to receive ANPR data from relevant sites, thereby avoiding the reception and processing of large amounts of irrelevant data. Likewise, Fig. 8(b) shows that our method's ART was also greatly lower than PlatoonFinder method. And Fig. 8(c) shows that with the degree of parallelism increasing, the ART in our method is reduced. Especially at the beginning of vehicle companion discovery, when we need to process all the ANPR data, the difference becomes even more pronounced. This highlights the effectiveness of service composition parallelism in enhancing performance, even when dealing with substantial amounts of data streams.



(a) Different methods' DVP



(b) Different methods' ART



(c) The ART of differernt digrees of parallelism

Fig. 8. Comparison of efficiency between different methods.

6 Conclusion

We propose a stream data service framework to discover vehicle companion efficiently and effectively. Specifically, we redefined the vehicle companion pattern based on relaxed temporal and spatial constraints, and deal with vehicle redundancy in area companions. The definition of vehicle companion helps more potential companion vehicles' discovery. And we encapsulate each trajectory data generated by monitoring site into corresponding stream data services, and enhance value density of trajectory data initially. These services are designed to collaborate with each other dynamically based on the vehicles' real-time status and the monitoring sites' adjacent relationships. Through this collaboration, higher-level services are constructed to ultimately identify the companion vehicles. Dynamic service collaboration ensures efficiency of our method because of the reducing of unnecessary communication and processing costs. And a stream partition strategy that divides services into partitions is also proposed to further ensure the performance of our method. The results of experiment based on real datasets demonstrate that our discovery method can identify more potential companion vehicles while requiring fewer processing and communication resources.

For future work, we intend to study more interesting and extensive correlations and patterns in trajectory streams, and study how to better utilize stream data service in intelligence transportation area.

Acknowledgment. This work was supported by Youth Foundation of Shandong Natural Science Foundation (No. ZR2021QF099) and Doctor Scientific Research Foundation of Shandong Jianzhu University (No. X21007Z).

References

1. Gao, Q., Zhang, F.L., Wang, R.J., Zhou, F.: Trajectory big data: a review of key technologies in data processing. J. Softw. **28**(4), 959–992 (2017)
2. Kong, X., Li, M., Zhao, G., et al.: COOC: visual exploration of co-occurrence mobility patterns in urban scenarios. IEEE Trans. Comput. Soc. Syst. **6**(3), 403–413 (2019)
3. Jia, J., Ying, Hu., Zhao, B., Ji, G., Liu, R.: Discovering collective converging groups of large scale moving objects in road networks. In: Jensen, C.S., et al. (eds.) DASFAA 2021. LNCS, vol. 12682, pp. 307–324. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-73197-7_21
4. van Mulken, M., Speckmann, B., Verbeek, K.: Density approximation for moving groups. In: Morin, P., Suri, S. (eds.) WADS 2023. LNCS, vol. 14079, pp. 675–688. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-38906-1_45
5. Ning, Z., Xia, F., Ullah, N., Kong, X.J., Hu, X.P.: Vehicular social networks: enabling smart mobility. IEEE Commun. Mag. **55**(5), 16–55 (2017)
6. Zheng, Y., Xie, X., Ma, W.Y.: GeoLife: a collaborative social networking service among user, location and trajectory. Bull. Tech. Committee Data Eng. **33**(2), 32–39 (2010)
7. Han, Y.B., Wang, G.L., Yu, J., et al.: A service-based approach to traffic sensor data integration and analysis to support community-wide green commute in China. IEEE Trans. Intell. Transp. Syst. **17**(9), 2648–2657 (2016)
8. Vieira, M.R., Bakalov, P., Tsotras, V.J.: On-line discovery of flock patterns in spatio-temporal data. In: Proceedings of the 17th ACM International Symposium on Advances in Geographic Information Systems (ACM SIGSPATIAL), pp. 286–295. Association for Computing Machinery, New York (2009)
9. Zaleshina, M., Zaleshin, A.: Flock patterns when pigeons fly over terrain with different properties. In: ICPRAM, pp. 334–341 (2019)
10. Jeung, H., Shen, H.T., Zhou, X.F.: Convoy queries in spatio-temporal databases. In: Proceedings of the IEEE International Conference on Data Engineering (ICDE), Washington, pp. 1457–1459. IEEE Computer Society (2008)
11. Yan, S., Wu, B., Shang, L., Wang, Y., Lyu, J.: A convoy discovering algorithm for passengers in the cruise based on UWB positioning. In: 2021 6th International Conference on Transportation Information and Safety (ICTIS), pp.392–397 (2021)
12. Li, Z.H., Ding, B.L., Han, J.W., Kays, R.: Swarm: mining relaxed temporal moving object clusters. Proc. VLDB Endow. **3**(1), 723–734 (2010)
13. Wang, X., Zhang, Y., Wang, L., et al.: Task decision-making for UAV swarms based on robustness evaluation. In: 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), pp. 242–248 (2019)
14. Li, Y.X., Bailey, J., Kulik, L.: Efficient mining of platoon patterns in trajectory data-bases. Data Knowl. Eng. **100**, 167–187 (2015)
15. Zhu, M.L., Liu, C., Wang, X.B., Han, Y.B.: Approach to discover companion pattern based on ANPR data stream. Ruan Jian Xue Bao/J. Softw. **28**(6), 1498–1515 (2017)
16. Zhuofeng, Z., Shuai, L., Yanbo, H.: Similar trajectory query method based on massive vehicle license plate recognition data. J. Tsinghua Univ. (Sci. Technol.) **57**(2), 220–224 (2017)
17. Xiao, Y., He, X., Yang, C., Liu, H., Liu, Y.: Dynamic graph computing: a method of finding companion vehicles from traffic streaming data. Inf. Sci. **591**, 128–141 (2022)

18. Showail, A., Tahir, R., Zaffar, M., et al.: An internet of secure and private things: a service-oriented architecture. Comput. Secu. **120**, 102776 (2021)
19. Mishra, S., Sarkar, A.: Service-oriented architecture for Internet of Things: a semantic approach. J. King Saud Univ. Comput. Inf. Sci. **34**(10), 8765–8776 (2021)
20. Huang, B., Zhang, B., Sheng, Q.Z., Lam, K.-Y.: A multi-task learning approach for predicting intentions using smart home IoT services. In: Troya, J., Medjahed, B., Piattini, M., Yao, L., Fernández, P., Ruiz-Cortés, A. (eds.) ICSOC 2022. LNCS, vol. 13740, pp. 413–421. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20984-0_29
21. Zhang, Z., Liu, C., Su, S., et al.: SDaaS: a method for encapsulating sensor stream data as services. China J. Comput. **40**(2), 445–463 (2017)
22. Han, Y., Liu, C., Su, S.: A decentralized and service-based approach to proactively correlating stream data. In: S2 International Conference on Internet of Things, pp. 93–100 (2016)
23. Zhang, Z., Hu, Q., Hou, G., Zhang, S.: A real-time discovery method for vehicle companion via service collaboration. Int. J. Web Inf. Syst. **19**(5/6), 263–279 (2023)
24. Ali, Z.H., Ali, H.A., Badawy, M.M.: A new proposed the Internet of Things (IoT) virtualization framework based on sensor-as-a-service concept. Wireless Pers. Commun. **97**(1), 1419–1443 (2017)
25. Silva, B.N., Khan, M., Han, K.: Integration of Big Data analytics embedded smart city architecture with RESTful web of things for efficient service provision and energy management. Future Gener. Comput. Syst. **107**, 975–987 (2018)
26. Belhadi, A., Djenouri, Y., Srivastava, G., Lin, J.C.: Fast and accurate framework for ontology matching in web of things. ACM Trans. Asian Low Resour. Lang. Inf. Process. **22**(5), 147:1–147:19 (2023)
27. Ahrabian, A., Kolozali, S., Enshaeifar, S., et al.: Stream data analysis as a web service: a case study using IoT sensor data. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, New Orleans, United States, pp. 6000–6004. IEEE (2017)
28. Aguilar, J., Sanchez, M., Cordero, J., et al.: Learning analytics tasks as services in smart classrooms. Univ. Access Inf. Soc. **17**(4), 693–709 (2019)
29. Malik, M., Abdallah, S., Alaraj, M.: Data mining and predictive analytics applications for the delivery of healthcare services: a systematic literature review. Ann. Oper. Res. **270**(1–2), 287–312 (2018)
30. Lu, Y., Misra, A., Wu, H.: Smartphone sensing meets transport data: a collaborative framework for transportation service analytics. IEEE Trans. Mob. Comput. **17**(4), 945–960 (2018)
31. Zatout, S., et al.: A model-driven approach for the verification of an adaptive service composition. Int. J. Web Eng. Technol. **15**(1), 18–26 (2021)
32. Zhang, Z.M., Yang, Z.G., Ali, S., Asshad, M.: A dynamic declarative composition scheme for stream data services. Mob. Inf. Syst. **1–8**, 2021 (2021)
33. Wang, Y., Wang, S., Yang, B., et al.: An effective adaptive adjustment method for service composition exception handling in cloud manufacturing. J. Intell. Manuf. **33**, 735–751 (2022)
34. Zhang, J., Liu, S., Yang, Q., Zhou, Y.: DMFUCP: a distributed mining framework for universal companion patterns on large-scale trajectory data. J. Comput. Res. Dev. **59**(3), 647–660 (2021)
35. Mertens, S.: The easiest hard problem: number partitioning. Comput. Complex. Stat. Phys. **125**, 125–139 (2003)