Towards spatiotemporal integration of bus transit with data-driven approaches

Júlio C. Borges [Universidade Tecnológica Federal do Paraná | julio.2018@alunos.utfpr.edu.br]
Altieris M. Peixoto [Universidade Tecnológica Federal do Paraná | altieris.marcelino@gmail.com]
Thiago H. Silva [Universidade Tecnológica Federal do Paraná | thiagoh@utfpr.edu.br]
Anelise Munaretto [Universidade Tecnológica Federal do Paraná | anelise@utfpr.edu.br]
Ricardo Lüders [Universidade Tecnológica Federal do Paraná | luders@utfpr.edu.br]

Abstract
This study aims to propose an approach for spatiotemporal integration of bus transit, which enables users to change bus lines by paying a single fare. This could increase bus transit efficiency and, consequently, help to make this mode of transit more attractive. Usually, this strategy is allowed for a few hours in a non-restricted area; thus, certain walking distance areas behave like “virtual terminals.” For that, two data-driven algorithms are proposed in this work. First, a new algorithm for detecting itineraries based on bus GPS data and the bus stop location. The proposed algorithm’s results show that 90% of the database detected valid itineraries by excluding invalid markings and adding times at missing bus stops through temporal interpolation. Second, this study proposes a bus stop clustering algorithm to define suitable areas for these virtual terminals where it would be possible to make bus transfers outside the physical terminals. Using real-world origin-destination trips, the bus network, including clusters, can reduce traveled distances by up to 50%, making twice as many connections on average.

Keywords: bus transit network, spatiotemporal integration, data-driven model, urban computing, smart city

1 Introduction

The collective public transit service has been little expanded and modernized over the last few decades, compared to private individual transit, which currently consumes the largest available urban road space. Motta et al. [2013] show historical conflicts and contradictions at the origin of the problems affecting collective public transit systems throughout Brazil.

Aiming to revert this, attract new users, and improve the quality of the service, several companies that manage collective public transit and researchers in this area are using vehicle GPS data more frequently by providing an efficient and accurate way to track the position of vehicles at a given time [Bona et al., 2016; Curzel et al., 2019; Santin et al., 2020; Gubert et al., 2023]. In this way, public transit can be investigated from different perspectives, examining the system’s dynamic behavior, measuring the efficiency of services, identifying movement patterns, integration capacity, peak hours, etc.

In analyzing the operation of a public transit network, it is generally necessary to identify the instants of time in which the bus passes at the stops, whether or not the bus stops. In general, this identification is done by a map-matching algorithm crossing the geolocation information of the bus with the location of the bus stop. A suitable method for this task was employed in the work of Martins et al. [2022]. The method presents a solution for detecting bus stops, even on two-way roads, correcting GPS inaccuracies, and identifying the exact time of passage at a bus stop.

However, the practical application of working with real GPS data can pose significant challenges due to intermittent communication failures, leading to data gaps that may adversely impact analysis outcomes if not appropriately managed. Another problem is the need to cross-reference scheduled timetables with GPS information, as outdated timetables generate inconsistencies, such as the cases reported by Martins et al. [2022]. Thus, researchers are often forced to look for periods in which there is consistency in logs and tables. Nevertheless, this limitation confines the analysis to specific time intervals and conditions during which the data is optimal.

The itinerary detection problem, as defined in this article, goes further, as it consists of matching the GPS logs of the movement of a bus with its respective itinerary and schedule, which is defined in the bus operation planning. This problem is affected by communication failures in sending GPS data, making it impossible to detect the bus itinerary correctly. An approach for detecting itineraries was employed in the work of Peixoto et al. [2020] using the schedule table for a bus line. In this case, both the starting point departure time and the scheduled arrival time at the endpoint must be known. In addition, logs of bus movements are needed. Although this algorithm identifies most itineraries, several logs from bus monitoring are discarded, as they do not appear associated with any itinerary. Unlike the previous work, the algorithm proposed in this article does not use the scheduled bus line schedule.

In our previous work [Borges et al., 2023], an algorithm was introduced capable of detecting the itinerary of a bus in operation using data from its geolocation without the need to
use information from the scheduled timetables of each bus line. Furthermore, the proposed algorithm adds missing data due to communication failures by interpolating known time values.

In this context, this article aims to develop a spatiotemporal integration of bus transit using data-driven strategies. There is no single definition for public transit service integration as it depends on the context and the transit modes examined. This work focuses on the integration of the public bus transit, which can be understood as a strategy to increase the network connectivity, offering users greater route possibilities without the need to create new bus stops [Wang et al., 2021].

The main contributions of this paper are as follows:

• Bus itinerary detection algorithm for building itineraries with spatiotemporal information based on GPS bus monitoring;
• Analysis of the impact of the interpolation of GPS data on the different types of lines of the transit system;
• Full database assessment using real GPS bus data and temporal evaluation of bus service;
• Bus stop clustering algorithm for grouping nearby stops where connections can be made between bus lines with a single fare;
• Evaluation of the bus stop clustering algorithm and the bus service synchronization;
• Evaluation of the spatiotemporal integration impact on bus trips in terms of distance traveled and number of transfers between bus lines using real-world origin-destination trips.

The article’s structure is as follows: In Section 2, an overview of related works is presented. Section 3 provides a detailed explanation of the proposed algorithms. The outcomes of these algorithms are discussed in Section 4, followed by the conclusion in Section 5.

2 Related Works

2.1 Public transit Mobility Understanding

Detailed information about public transit systems, including bus stops, stations, terminals, routes, and timetables, are widely deployed in standard data formats, such as the General Transit Feed Specification (GTFS). In recent decades, new technologies have improved tools for monitoring and verifying schedules and other performance metrics, such as Automatic Vehicle Location (AVL) and other automated data collection to traffic [Wilson et al., 2009]. In particular, GPS equipment allows a wide range of applications with the potential to improve service and efficiency. The studies of Sridivi et al. [2017], Hakeem et al. [2022], and Desai et al. [2022] exemplify recent applications in which onboard equipment collects GPS trajectories of buses and centralizes them in a server. These data can be used in various transit system management applications.

A literature review on the bus trajectory data application can be seen in War et al. [2022]. Aspects such as data sources and methods of Big Data and IoT in mass public transit are described in Welch and Widita [2019]. Further discussion of using bus GPS trajectory data is provided in Singla and Bhatia [2015].

Concerning the Integrated transit Network (ITN), several studies such as Bona et al. [2016]; Curzel et al. [2019], Santin et al. [2020]; Rosa et al. [2020]; Gubert et al. [2023], used open public transit data, such as vehicle GPS trajectory data, timetables, and itineraries, in the creation of models that allowed the expansion of the understanding of the characteristics and behaviors of the transit network, to promote the improvement of the efficiency of the service. Other studies on urban mobility, transit networks, and computational models that use similar data are performed by Rodrigues et al. [2017]; Wehmuth et al. [2018]; Maduako et al. [2019]; Sadeghian et al. [2021]; Li and Rong [2022]. These models provide many efficiency measures for transit services, helping to identify opportunities for important improvements such as cost reduction.

2.2 Public Transit Data Quality

Another important issue is the data quality. Cleaning the raw data must reduce inconsistencies for the models to work properly. For example, Martins et al. [2022] pointed out several issues present in real GPS data, and therefore, they developed a model of map matching. This model can be used for: i) vehicle stop detection between nearby bus stops on a two-way street, in which one point serves both directions (“going” and “returning”) from the bus line; ii) GPS inaccuracies; and iii) the vehicle’s exact time of passage at a bus stop. The problem of detecting bus stops was also addressed in the work of Peixoto et al. [2020], where the vehicle’s itinerary was also employed using data from timetables. Other authors also faced the challenge of detecting compatibility between the trajectory of buses and their respective itineraries, such as Yin et al. [2014]; Queiroz et al. [2019]; Chawuthai et al. [2023]. In these works, the main objective of detection is to identify whether or not a bus GPS trajectory is in accordance with the planned itinerary to signal any inconsistency. In the work of Gallotti and Barthelemy [2015], inconsistencies in vehicle stop times were corrected using a temporal interpolation method, but no measure of interpolation error was presented.

The algorithm proposed in the present article fills some gaps, identifying and correcting inconsistencies in GPS trajectories according to the itinerary. Furthermore, a method to measure the interpolation error is presented. Therefore, the proposed approach treats the itinerary detection problem and the temporal interpolation by reducing the inconsistencies in the raw data and providing a reliable database for new applications.

2.3 Public Transit Integration

Coordinating several transit routes, schedules, and vehicles while looking to attend to passenger preferences is a complex challenge [Arriagada et al., 2022]. Although bus services have previously defined timetables, current trips often breach schedules due to uncertainties such as congestion,
traffic lights, poor weather conditions, boarding and alighting times, and maintenance interruptions [Li et al., 2021; Kumar and Khani, 2023]. Public transit users often avoid long waits, extensive walking distances, or route changes. However, transfers are seen as a strategy to deal with overcrowding in vehicles and bus stations [Pan et al., 2023]. In addition, transfers are necessary to integrate routes without building new stops [Wang et al., 2021] while reducing the total operational cost [Liu et al., 2023]. The distance required to walk to a bus stop and the time spent are important aspects for building more accessible public transit systems [Zochowska et al., 2022]. In this direction, Noichan and Dewancker [2018] employed spatial analysis to assess route connectivity, examining the distances between transfer points as a crucial factor. A better synchronization between timetables and vehicle schedules may reduce the wait time [Prathyusha et al., 2021; Kumar and Khani, 2023].

In general, works that present approaches for integrating transit networks focus on route optimization problems. They propose to combine different routes in a transit network to complete a trip. These approaches usually employ optimization methods such as mathematical programming, heuristics, or simulation. In this case, transfer points are previously known or identified by static data such as vehicles’ scheduled departure and arrival times provided in the timetables. However, known transfer points or timetable approaches may not be the best option for bus networks because of the uncertainties in travel times that are introduced by several of the previously mentioned factors. Therefore, many works employ AVL data from online bus travel monitoring systems because it offers a more accurate way of evaluating transit networks [Li et al., 2021; Kumar and Khani, 2023].

A summary of related works for public transit integration is presented in Table 1. It presents relevant approaches for bus or multimodal networks using timetables or AVL data and considering temporal, spatial, or spatiotemporal dimensions of the integration. The results are generated and reported using data-driven, optimization, or simulation methodologies. “Virtual Terminal” refers to the ability to identify bus line integration regions as described below.

Our approach of identifying transfer points between bus lines relates to AVL-based (online GPS information of vehicles) and data-driven approaches. However, it is different from previous works because it identifies suitable transfer points using both spatial (bus stop location) and temporal (online frequency of buses) information. Regions of interest are identified using a measure of maximum walking distance and “synchronization” between bus lines at a particular bus stop. They can be used as “virtual terminals” for bus line integration. Moreover, it is general enough to be adapted to other GTFS and AVL data systems and can identify opportunities to integrate bus lines based on historical data.

3 Spatiotemporal Integration of Bus Transit

A spatiotemporal integration of bus transit is a strategy that allows users to change bus lines by paying a single fare. Usually, this strategy is allowed for a few hours in a non-restricted area (terminals are restricted areas, for instance). Certain walking distance areas function similarly to “virtual terminals”. Two goals must be considered: i) detection of bus itineraries and ii) bus stop clustering to define suitable areas for these “virtual terminals.” Both objectives are achieved by the two data-driven algorithms proposed in this section.

The central issue is to develop an itinerary detection algorithm independent of the bus schedule table. Initially, it is necessary to differentiate the concepts of “static network” and “dynamic network.” These concepts were also used in the work of Peixoto et al. [2020]. A static network represents the topology of the transit network, that is, the sequencing of bus stops on a specific line covering all itineraries offered by the service, as considered in Bona et al. [2016].

Since the static network describes the topology of the bus line and its respective itineraries without including timetables, the dynamic network is formed as a given vehicle reaches the points provided for in its service itinerary. The proposed itinerary detection algorithm is composed of 3 steps:

- **step 1**: mark the time a bus passes at bus stops (map matching algorithm).
- **step 2**: sequence bus stops according to these time marks (temporal sequencing).
- **step 3**: associate a temporal sequence of bus stops to a known itinerary, interpolating and removing marks if necessary (the proposed algorithm).

A map-matching algorithm is used at step 1. The algorithm used in this work is based on Martins et al. [2022] algorithm. It calculates the Haversine distance from each vehicle position \(log\) as used in Panigrahi [2014], Lawhead [2015] to all bus line stops and assigns the \(log\) to the closest stop. This way, it is possible to mark the time of passage of a vehicle to each bus line stop. **Step 2** sorts time marks in ascending order to obtain a temporal sequencing of points. **Step 3** is accomplished by the algorithm proposed below.

The itinerary detection algorithm aims to associate a \(log\) of events captured by the movement of a specific bus to the sequence of points of their respective bus line registered in the table LinePoints. Thus, it is possible to associate the instant time of passage of the bus at all points on the line. This is illustrated in Figure 1, where a \(log\) of events \(log = (l_1, t_1, l_2, t_2, l_3, t_3, l_4, t_4, l_5, t_5)\) will be associated with an itinerary \(iti = (p_1, p_2, p_3, p_4, p_5)\), where \(t_i\) is the moment when the bus passes at coordinate \(l_i\) and \(p_i\) is a point on the bus itinerary.

For example, according to Figure 1, there is no record of a bus passing at point \(p_3\), and there is a record of a bus passing at a position \(l_4\) that does not correspond to any registered point on the line.

The map matching algorithm associates the locations \(l_i\) to the respective bus stops by evaluating a measure of spatial proximity between \(l_i\) and a point on the itinerary. In the case of Figure 1, the result of map matching is the mapping \(map = ((p_1, l_1), (p_2, l_2), (p_3, l_3), (l_4, l_4), (l_5, l_5))\), and no point is associated with location \(l_4\) and there is no record of the passage through point \(p_3\).
Figure 1. Example of an itinerary $iti = (p_1, p_2, p_3, p_4, p_5)$ in full line and a log $log = ((l_1, t_1), (l_2, t_2), (l_3, t_3), (l_4, t_4), (l_5, t_5))$ in dashed line.

Then, the proposed algorithm detects the itinerary with temporal information $det = ((p_1, t_1), (p_2, t_2), (p_3, t_3), (p_4, t_4), (p_5, t_5))$, associating the moment of passage to each point of the bus line. In this case, the time instant $t_k = t_2 + (t_3 - t_2)/2$ is estimated by averaging the times $t_2$ and $t_3$ of points adjacent to $p_3$.

The above result for itinerary detection $det = \{(p_i, t_i)\}$ with temporal information can be generalized to (1) for $iti$ of dimension $n$ and $log$ of dimension $m$.

$$det = \{(p_i, t_i) : (p_i, t_i) \in map; i = 1 \text{ to } n; j = 1 \text{ to } m\}$$

(1)

where $t_i = t_{i-1} + ((k \cdot w) - t_k)/w$ for $k < i < (k + w)$ and $t_k = t_{k-1} + (w - 1) \text{ bus stops that were not mapped by mapping between points } (p_k, t_k)$ and $(p_{k+w}, t_{k+w}) \in map$. The above procedure is summarized in Algorithm 1, whose main variables are described in Table 2.

### Algorithm 1 Itinerary detection

**Input:** $iti = \{p_i\}, 1 \leq i \leq n; log = ((l_j, t_j)), 1 \leq j \leq m$ // ordered by points and time, respectively

**Output:** $det = ((p_i, t_i)), 1 \leq i \leq n$

1. $map \leftarrow \{\}$
2. $time \leftarrow -1$
3. for each $p_i \in iti$ do
4.   $found \leftarrow False$
5.   for each $(l_j, t_j) \in log$ do
6.     if $(p_i, t_i) > time$ then
7.       $map \leftarrow map \cup \{(p_i, t_i)\}$
8.     end if
9.   end for
10. end for
11. if $(found = False)$ then
12.   $det \leftarrow det \cup \{(p_i, t_i)\}$
13. end if
14. end for
15. $det \leftarrow \{\}$
16. for each $(l_j, t_j) \in map$ do
17.   if $(l_j, t_j) \neq None$ then
18.     $det \leftarrow det \cup \{(p_i, t_i)\}$
19.   else
20.     computes $u, \Delta t = (t_{k+w} - t_k)$ and $t_k = t_{k-1} + \Delta t/w$
21. end if
22. end for
23. end if
24. end for

The proposed algorithm has as requirements the static network and the dynamic network. That is, it is necessary to provide as input both the structure (or topology) of the transit network, according to the file PontosLinha, as well as the $log$s of GPS of buses from the file Veículos. The advantage over the method proposed by Peixoto et al. [2020] is that there is no need for a table of scheduled bus lines (files TabelaLinhas and TabelaVeículos).

Given a set of markings with known times $(p_i, t_i)$ and estimated times $(\hat{p}_i, \hat{t}_i)$, temporal interpolation introduces an estimation error given by $err = |\hat{t}_i - t_i|$. This work takes known values from the original data set to obtain the estima-
tion error evaluation results.

The second algorithm deals with bus stop clustering that behaves like virtual terminals (not constrained to a particular area), where users can change between bus lines with a single fare. They are defined when nearby bus stops are clustered within a 600 meter radius, which is considered a walking distance suitable for changing bus lines [Peixoto et al., 2020].

<table>
<thead>
<tr>
<th>Table 3. List of variables for Algorithm 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sets</strong></td>
</tr>
<tr>
<td>candidates</td>
</tr>
<tr>
<td>Bus stops ( { c_{i} } ) candidate to cluster centroids sorted in ascending order by the average number of buses</td>
</tr>
<tr>
<td>bus_stops</td>
</tr>
<tr>
<td>All stops ( { b_{i} } ) of the bus network</td>
</tr>
<tr>
<td>cluster</td>
</tr>
<tr>
<td>Bus stops of a cluster</td>
</tr>
<tr>
<td>clusters</td>
</tr>
<tr>
<td>Set of clusters</td>
</tr>
<tr>
<td><strong>Indexes</strong></td>
</tr>
<tr>
<td>( i ) Bus stop in a list of centroid candidates, ( 1 \leq i \leq n )</td>
</tr>
<tr>
<td>( j ) Bus stop in a list of all bus stops, ( 1 \leq j \leq m )</td>
</tr>
</tbody>
</table>

Centroids and clusters are computed according to Algorithm 2, whose main variables are described in Table 3. It starts with a descending-order list of candidate centroids ordered by the average number of buses. For each centroid in the head of the list (line 3), all neighbor bus stops within a given distance from the centroid are included in the cluster (line 7). The new cluster is then added to the set of clusters (line 10), and all bus stops of the cluster are removed from the candidates list. It means that clustered bus stops are no longer candidates for another cluster. The algorithm ends when the list of candidates is empty (line 12).

**Algorithm 2** Bus stop clustering

```
input: candidates = \{ c_{i} \}, 1 \leq i \leq n; // ordered list of n centroid candidates
bus_stops = \{ b_{i} \}, 1 \leq j \leq m // list of bus stops
output: clusters = \{ \{ cluster \}_{j} \}, 1 \leq j \leq l // set of l clusters
1: clusters ← \{ \}
2: while candidates ≠ \{ \} do
3: centroid ← head(candidates) // first element of the ordered list
4: cluster ← \{ centroid \}
5: for each \( b_{i} \in \text{bus_stops} \) do
6:   if haversine(centroid, \( b_{j} \)) ≤ 600 then // Haversine distance
7:     cluster ← cluster ∪ \{ \( b_{j} \) \} // Haversine distance
8: end if
9: end for
10: clusters ← clusters ∪ \{ \{ cluster \} \}
11: candidates ← candidates \−\ \{ candidates ∩ cluster \} // remove the clustered bus stops from the candidates list
12: end while
```

4 Results and Discussions

The evaluation of our proposal is accomplished using the real bus GPS data. Data description, itinerary detection, and evaluation of interpolation errors are presented in Sections 4.1 to 4.3. Results for spatiotemporal integration of bus transit are presented in Sections 4.4 to 4.6.

4.1 Public Transit Data

The C3SL repository\(^1\) is recognized as the main source of data for academic applications on public transit in Curitiba. It has been used in several studies.

Geolocated data from bus monitoring are needed to deal with the map matching problem and static information from the transit network and bus schedule. The Open Data Portal of Curitiba City Hall provides a daily updated database containing data on public transit in Curitiba available via WebService with relevant information such as GTSF Files, Lines, Points, Itineraries, Position of Vehicles, and Tables of Schedules. Data is transferred through files in JSON format through an API. The API data dictionary can be found in technical documentation [URBS, 2022b].

According to the operational data published in URBS [2022a], ITN has a fleet of 1,226 vehicles (disregarding the reserve buses) that serve 250 lines, 22 terminals, and 329 tube stations and make, on average, 1,365,615 trips per day useful. These vehicles periodically send their location according to URBS [2022b], which is stored in a daily log to be consulted via the API. Because the native service does not offer requests by date, C3SL provides JSON files containing a daily and complete history of ITN operations updated on day 1. An extensive exploratory analysis of these data is given in Vila et al. [2016].

The following data files of C3SL are used in the experiments:

- **Linhas**: contains code, name, service category, color, and other attributes of all ITN bus lines.
- **PontosLinhas**: stores name, code, type, latitude, and longitude of all ITN bus stops, and describes the correct sequence of stops according to bus line itineraries.
- **Vehicles**: contains the coordinate history of vehicles in operation. The GPS position of a vehicle with date and time is sampled every 20 seconds on average.
- **TabelaLinhas**: stores bus line timetables at stops; most stops do not have timetable information.
- **TabelaVeículos**: stores schedule times of bus itinerary segments.

4.2 Case Study - Bus Line 829

The bus line 829 “Universidade Positivo” (“Alimentador”) was chosen for a case study. It is circular, i.e., the same start and end stops, with single-direction trips. In addition, it has few stops that make visualization and interpretation of the data easier. Table 4 shows the scheduled itinerary of bus line 829, containing bus stop names and sequences.

<table>
<thead>
<tr>
<th>Table 4. Scheduled itinerary of bus line 829 with bus stops</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bus stop</strong></td>
</tr>
<tr>
<td>Terminal Campo Comprido</td>
</tr>
<tr>
<td>R. Angelo Nebosne, 75</td>
</tr>
<tr>
<td>R. Prof. Pedro Viciato Parigot de Souza, 4716</td>
</tr>
<tr>
<td>R. Prof. Pedro Viciato Parigot de Souza, 5136</td>
</tr>
<tr>
<td>R. Casemiro Augusto Rodacki, 233</td>
</tr>
<tr>
<td>R. Carlos Müller, 331</td>
</tr>
<tr>
<td>R. Carlos Müller, 871</td>
</tr>
<tr>
<td>R. Eduardo Sprada, 5273</td>
</tr>
<tr>
<td>R. Dep. Heitor Alencar Furtado, 5181</td>
</tr>
<tr>
<td>R. Dep. Heitor Alencar Furtado, 4900</td>
</tr>
<tr>
<td>Terminal Campo Comprido</td>
</tr>
</tbody>
</table>

The bus route starts at stop 1 in “Terminal Campo Comprido”, reaches intermediate stops 2 to 10, and returns to the starting stop 1, as shown in Figure 2.
The procedure marks the exact time when bus BA020 passes through intermediate points 2 to 10, and returns to the starting point.

The scenario was built using real logs from 07/11/2022 of bus BA020. A round trip occurs between 06:04 to 06:32, during which there is no loss of GPS data. Therefore, this scenario is a suitable case to verify the application of the proposed algorithm and evaluate interpolation errors, simulating communication failures. Some logs are then deleted within specific time intervals. In this case, the map-matching algorithm does not detect the vehicle passing at some points, and the proposed algorithm can recover the information using interpolation. Points 3, 5, and 8 of bus line 829 were chosen to be removed from the original data set. The case study parameters are shown in Table 5.

Table 5. Case study parameters for line 829.

<table>
<thead>
<tr>
<th>Line</th>
<th>829 - Universidade Positiva</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus</td>
<td>BA020</td>
</tr>
<tr>
<td>Date</td>
<td>7/11/2022</td>
</tr>
<tr>
<td>Return time</td>
<td>06:04 to 06:32</td>
</tr>
<tr>
<td>Failure Interval</td>
<td>06:17 to 06:19 at point 5</td>
</tr>
<tr>
<td></td>
<td>06:26 to 06:28 at point 8</td>
</tr>
</tbody>
</table>

Table 6 presents the result of steps 1, 2 and part of step 3. The procedure marks the exact time when bus BA020 passes at stops (map matching), creates a temporal sequencing (step 2), locates the itinerary, and assigns a sequence number to each log (part of step 3).

Table 6. Results of applying steps 1, 2, and part of step 3 to the case study.

<table>
<thead>
<tr>
<th>Bus stop</th>
<th>Time</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terminal Campo Comprido</td>
<td>06:04:51</td>
<td>1</td>
</tr>
<tr>
<td>R. Dep. Heitor Alencar Furtado, 4900</td>
<td>06:14:08</td>
<td>10</td>
</tr>
<tr>
<td>R. Angelo Nobiske, 75</td>
<td>06:14:36</td>
<td>2</td>
</tr>
<tr>
<td>R. Prof. Pedro Vizarioi Parigot de Souza, 4716</td>
<td>06:15:35</td>
<td>3</td>
</tr>
<tr>
<td>R. Prof. Pedro Vizarioi Parigot de Souza, 5136</td>
<td>06:16:34</td>
<td>4</td>
</tr>
<tr>
<td>R. Casemiro Augusto Rodacki, 233</td>
<td>06:18:07</td>
<td>5</td>
</tr>
<tr>
<td>R. Carlos Muller, 331</td>
<td>06:19:30</td>
<td>6</td>
</tr>
<tr>
<td>R. Carlos Muller, 871</td>
<td>06:20:06</td>
<td>7</td>
</tr>
<tr>
<td>R. Eduardo Sprada, 527</td>
<td>06:23:48</td>
<td>8</td>
</tr>
<tr>
<td>R. Dep. Heitor Alencar Furtado, 5181</td>
<td>06:28:30</td>
<td>9</td>
</tr>
<tr>
<td>R. Dep. Heitor Alencar Furtado, 4900</td>
<td>06:29:06</td>
<td>10</td>
</tr>
<tr>
<td>Terminal Campo Comprido</td>
<td>06:31:41</td>
<td>1</td>
</tr>
</tbody>
</table>

However, Table 6 contains some inconsistencies. For example, the stop “R. Dep. Heitor Alencar Furtado, 4900” marked at 06:14:08 is incorrect because it is at the end of bus itinerary. A close examination reveals that the bus route between points 1 and 2 passes very close to point 10, as illustrated in Figure 3. In this case, the map-matching algorithm generates a markup error. This algorithm is thus insufficient to handle logs properly. The marking error is identified by combining the result of map-matching with the proposed Algorithm 1.

The proposed algorithm identifies gaps in the sequence after completing step 3. Due to the communication failure simulation, it identifies points 3, 5, and 8 were added to the itinerary due to temporal interpolation. This itinerary corresponds to the complete sequence of bus stops registered for line 829 with temporal information.

Table 7. Final result of applying Algorithm 1 to the case study.

<table>
<thead>
<tr>
<th>Bus stop</th>
<th>Time</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terminal Campo Comprido</td>
<td>06:04:51</td>
<td>1</td>
</tr>
<tr>
<td>R. Angela Nobiske, 75</td>
<td>06:14:36</td>
<td>2</td>
</tr>
<tr>
<td>R. Prof. Pedro Vizarioi Parigot de Souza, 4716</td>
<td>06:15:35</td>
<td>3</td>
</tr>
<tr>
<td>R. Prof. Pedro Vizarioi Parigot de Souza, 5136</td>
<td>06:16:34</td>
<td>4</td>
</tr>
<tr>
<td>R. Casemiro Augusto Rodacki, 233</td>
<td>06:18:07</td>
<td>5</td>
</tr>
<tr>
<td>R. Carlos Muller, 331</td>
<td>06:19:30</td>
<td>6</td>
</tr>
<tr>
<td>R. Carlos Muller, 871</td>
<td>06:20:06</td>
<td>7</td>
</tr>
<tr>
<td>R. Eduardo Sprada, 527</td>
<td>06:23:48</td>
<td>8</td>
</tr>
<tr>
<td>R. Dep. Heitor Alencar Furtado, 5181</td>
<td>06:28:30</td>
<td>9</td>
</tr>
<tr>
<td>R. Dep. Heitor Alencar Furtado, 4900</td>
<td>06:29:06</td>
<td>10</td>
</tr>
<tr>
<td>Terminal Campo Comprido</td>
<td>06:31:41</td>
<td>1</td>
</tr>
</tbody>
</table>

In evaluating the interpolation error, data from the movement of buses on line 829 during a whole day from 06:04 to 23:19 were used. Known points were randomly taken from the original data set, simulating communication failures. The estimation error \( err_i = |t_i - \hat{t}_i| \) was calculated from the known real values \( (p_i, t_i) \) of the points removed and the estimated values \( (\hat{p}_i, \hat{t}_i) \).

Error measures are computed as a function of the number of consecutive bus stops missing. In this experiment, error measures are calculated in seconds for 1 to 7 consecutive missing points, or \( w \in \{2, 3, \ldots, 8\} \). For each case, 100 samples without replacement were used to generate the result in Figure 4. It is observed that the interpolation error increases with the number of missing points. For most cases, the error ranges from less than 1 min to approximately 2 min (125 seconds).

This result suggests that the uncertainty introduced by interpolation is acceptable. A delay or advance of 2 minutes can be considered tolerable in an urban bus transit system. However, a closer look is needed to understand better which lines are most affected by the interpolation error at which times of day.

### 4.3 Database Assessment

The proposed algorithm was applied to logs on 07/11/2022 to evaluate the ability to detect itineraries using the entire database. The result is compared with the algorithm of...
Towards spatiotemporal integration of bus transit with a data-driven approach

Towards spatiotemporal integration of bus transit with a data-driven approach

Peixoto et al. [2020], which uses bus schedule tables. Table 8 shows the total number of assignments each algorithm makes to a valid itinerary by type of bus line. It is observed that the proposed algorithm provides a global increase from 68.83% to 99.33% in itinerary traceability gain. The new algorithm presents a result 44.31% better than Peixoto et al. [2020]. Except for lines of MADRUGUEIRO that should be further investigated, all lines of other types benefit. This result increases valid data in the database, preventing data from being discarded due to not being associated with any itinerary.

Table 8. Comparison between the number of tags assigned to a valid itinerary according to Peixoto et al. [2020] and the proposed algorithm. The percentage values are relative to the number of matches made by the map-matching algorithm.

<table>
<thead>
<tr>
<th>Line type</th>
<th>[Peixoto et al., 2020]</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>tags</td>
<td>%</td>
<td>tags</td>
</tr>
<tr>
<td>ALIMENTADOR</td>
<td>160,750</td>
<td>225,434</td>
</tr>
<tr>
<td>CONVENCIONAL</td>
<td>52,338</td>
<td>84,310</td>
</tr>
<tr>
<td>EXPRESSO</td>
<td>21,986</td>
<td>35,206</td>
</tr>
<tr>
<td>JARDINEIRA</td>
<td>234</td>
<td>499</td>
</tr>
<tr>
<td>LIGEIRÃO</td>
<td>2,988</td>
<td>4,677</td>
</tr>
<tr>
<td>LINHA DIRETA</td>
<td>8,421</td>
<td>11,879</td>
</tr>
<tr>
<td>MADRUGUEIRO</td>
<td>5,659</td>
<td>5,455</td>
</tr>
<tr>
<td>TRONCAL</td>
<td>26,136</td>
<td>34,447</td>
</tr>
<tr>
<td>TOTAL</td>
<td>278,512</td>
<td>401,907</td>
</tr>
</tbody>
</table>

Table 9. Distribution of the tags of Table 8 that had two error types: i) out of order and ii) missing bus stops for the proposed algorithm.

<table>
<thead>
<tr>
<th>Line type</th>
<th>Tags with errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>ii</td>
</tr>
<tr>
<td>ALIMENTADOR</td>
<td>15,764</td>
</tr>
<tr>
<td>CONVENCIONAL</td>
<td>2,432</td>
</tr>
<tr>
<td>EXPRESSO</td>
<td>487</td>
</tr>
<tr>
<td>JARDINEIRA</td>
<td>0</td>
</tr>
<tr>
<td>LIGEIRÃO</td>
<td>83</td>
</tr>
<tr>
<td>LINHA DIRETA</td>
<td>126</td>
</tr>
<tr>
<td>MADRUGUEIRO</td>
<td>1,470</td>
</tr>
<tr>
<td>TRONCAL</td>
<td>478</td>
</tr>
<tr>
<td>TOTAL</td>
<td>20,840</td>
</tr>
</tbody>
</table>

Although the tags of Table 8 are assigned to valid itineraries at the end of the proposed algorithm, some of them had errors in the sequence of stops or missing bus stops. Table 9 shows the distribution of tags that had two error types: i) out of order and ii) missing bus stops for the proposed algorithm. The percentage of 13.38% of tags is practically all corrected after the adjustment of sequence and missing points performed at the end of the proposed algorithm.

The interpolation error by line type is evaluated similarly to Section 4.2 but using all bus lines in the database that take only complete paths with real values. Figure 5 shows the interpolation error in seconds by line type. Bus lines ALIMENTADOR, CONVENCIONAL, EXPRESSO, MADRUGUEIRO, and TRONCAL have an error between 0 and 1 min approximately. On the other hand, JARDINEIRA, LIGEIRÃO, and LINHA DIRETA are more susceptible to interpolation errors. The big errors for JARDINEIRA and LINHA DIRETA might be due to the long distances between bus stops of these lines, whose time interpolation can be significantly affected by road traffic conditions.

4.4 Temporal Assessment of Bus Service

A bus service is provided according to the timetables of bus lines. Therefore, users can estimate the time interval between consecutive buses of a given bus line. However, from a user’s perspective, at a single bus stop, several buses from different bus lines interact to provide the bus service to that stop. In this case, how often do buses serve a particular bus stop (eventually by different bus lines)? This section provides a temporal assessment of bus services by identifying well-served urban areas with many buses and bus lines. It considers the database from 07/11/2022 to 07/15/2022. The GPS bus trajectories are tracked according to the method described in Section 3.

The bus service is evaluated in a time window of 10 minutes, representing an expected waiting time for most users. The number of buses that pass a stop is counted considering consecutive time windows of 10 minutes. This way, buses within a time window of 10 minutes are counted, a shift of 1 minute is then given to the window, allowing buses to be counted within the next 10 minutes, and so on. For each bus stop, a time series represents the number of buses observed in a 10-minute time interval in each minute of the day.

Figure 6 shows the average number of buses observed at a bus stop within a time window of 10 min shifted from 5:00 to 23:00 and aggregated into three categories of bus stops (terminal, street stop, and tube station). All bus stops inside a...
terminal are considered a single stop for capturing the number of buses available at a given time.

**Figure 6.** Average number of buses observed at a stop of terminals, street stops, and tube stations in a 10-minute moving window from 5:00 to 23:00.

**Figure 6** shows a peak from 6:00 to 8:00, with a maximum of around 7:00, and from 16:00 to 19:00, with a maximum of around 17:00. This behavior is relevant for terminals and tube stations with minor effects on street stops. Moreover, terminals present up to 4 times more buses than tube stations at peak hours, as expected, because they are hubs integrating different bus lines. Terminals and tube stations play an important role in Curitiba’s transit system because they offer users the possibility of transferring with a single fare.

The time series of **Figure 6** are then aggregated in a day, obtaining the average number of buses at a stop for terminals, street stops, and tube stations as shown in the boxplots of **Figure 7**. Terminals have the highest average number of buses, as expected. Street stops and tube stations have fewer buses but stand out as outliers, ranging from 2 to 13 buses on average. This means that street bus stops can also act as hubs if some integration between bus lines could be provided.

**Figure 7.** Boxplots showing the average number of buses observed at a stop of terminals, street stops, and tube stations in a 10-minute moving window.

The outliers of **Figure 7** represent an opportunity to improve the bus service because they have bus stops with a high frequency of buses. If the stops are close enough to each other, a hub can be built to allow connections between the respective bus lines. For instance, if temporal integration (with payment of a single fare) is allowed in certain regions of interest, new links between bus lines can be made in the network, eventually shortening distances and trip times. The regions of interest contain stops with a high concentration of buses, as shown in **Figure 8**.

**Figure 8.** Heat map of regions with a high density of stops and frequency of buses obtained from outliers of street stops and tube stations.

It is a heatmap obtained from outliers of street stops and tube stations (terminals are far from each other and usually do not have the potential for integration). The map highlights red regions with a high density of stops and frequency of buses. The most dense regions partially follow the North/South transit corridor. It means that buses running in this corridor have great potential to improve bus line connections in areas other than terminals. Based on this result, we aim to build bus stop clusters to behave like virtual terminals.

### 4.5 Virtual Terminal Evaluation

The results of Algorithm 2 are shown in **Figure 9**. It shows centroids of 104 clusters computed with 27 bus stops, each serving 15 bus lines on average and covering 2,472 bus stops. **Figure 10** shows the average number of buses observed at a cluster in a 10-minute moving window from 5:00 to 23:00. Peak hours occur in the morning between 06:00 and 8:00 and between 16:00 and 19:00 in the afternoon. More than 100 buses, on average, are observed between 6:00 and 7:00.

**Figure 10.** Average number of buses observed at a cluster in a 10-minute moving window from 5:00 to 23:00.

There is a correlation between the average number of buses and the number of bus lines in a cluster according to **Figure 11**. It shows a Pearson’s correlation coefficient of 0.78 with p-value < 0.001. In other words, not only do many buses attend a cluster, but also many bus lines.

**Figure 11.** Pearson’s correlation coefficient of the average number of buses and the number of bus lines in a cluster.

However, it is necessary to show that some “synchronization” exists between buses passing at cluster stops during the day. It is accomplished by evaluating the correlation between the bus time series of two stops of the same cluster.
Pearson’s correlation between the time series of two bus stops is computed for all pairs of stops in a cluster. For instance, the correlation matrix between bus stops of the cluster with centroid 170121 is shown in Figure 12. A matrix is shown for each period of the day: i) morning from 6:00 to 9:00; ii) midday from 11:00 to 14:00. According to Figure 12a, there are pairs of bus stops whose correlation achieves 0.75, which means that buses can meet each other more often in the morning. A similar behavior is observed in the evening from 17:00 to 20:00, as shown in Figure 12c. However, this behavior is not observed midday, according to Figure 12b. Some pairs of bus stops with a strong correlation in the morning now show a weak correlation in the midday.

It can be seen that the average correlation is more relevant in the morning and evening. It means that buses passing in a cluster are better “synchronized” in the morning or evening on average, i.e., they can meet each other approximately simultaneously considering a time window of 10 min. It is then expected that users at bus stops of the same cluster can change between bus lines within 10 min on average.

When the time window increases, the correlation between the time series tends to increase; in other words, if the passenger is willing to wait longer, they will be more likely to make a bus line transition within the cluster. However, this is not valid for all periods of the day. Figure 14 shows the results obtained using time windows of \{10, 15, 20, 25, 30, 35, 40, 45\} minutes.
Towards spatiotemporal integration of bus transit with a data-driven approach Borges et al. 2024

Figure 12. Correlation matrix between bus stops in the cluster with centroid 170121 and all its neighbors for morning, midday, and evening.

Figure 14. Observed correlation between time series for \{10, 15, 20, 25, 30, 35, 40, 45\}-minute moving window.

The result suggests an improvement in “synchronization” during the morning and evening, but this does not happen at midday. The explanation is that during this period, as it is not a peak demand, transit companies remove several buses from the streets, negatively impacting the correlation between the time series.

4.6 Impact of Spatiotemporal Integration on Bus Transit

Given an origin and a destination in the bus transit network, the impact of spatiotemporal integration is measured by the distance traveled and the number of transfers made in a trip with and without clusters of bus stops. Each cluster can be seen as a virtual terminal where transfers are made between bus lines with a single fare. The question is how these additional possible connections between bus lines benefit a trip. Networks with and without clusters of bus stops are shown in Figure 15.

The results presented in this section are based on an origin-destination (OD) survey by IPPUC in Curitiba [IPPUC, 2017]. Given an OD pair, the closest bus stops from the origin and destination are identified using a search distance of 600 m. Then, a short-path algorithm computes a feasible bus trip in the network and obtains its distance traveled and the number of transfers between bus lines. When a transfer is made in a cluster, an additional walking distance computed between the bus stops is considered. Because several trips can be found for a single OD pair, Yen’s algorithm [Yen, 1971] computes K-shortest paths with \(K = 30\). In other words, the number of shortest paths is limited to the top 30 alternatives. The results for distance traveled, and the number of transfers are shown in Fig. 16, with and without clusters.

According to Figure 16a, the average distance of 22.4 km traveled in the original network (without clusters) is longer than the average distance of 12.3 km using clusters. The opposite is observed concerning the number of transfers as shown in Figure 16b. The average value is two transfers without clusters, while the average number of transfers is four with clusters. Therefore, the results suggest that trip distances with bus clusters decrease by almost half at the expense of twice the number of transfers on average.

5 Conclusion

This work proposed data-driven approaches for detecting bus itineraries from GPS data and integrating bus transit in space and time. This spatiotemporal integration allows passengers...
to switch bus lines with a single fare by defining “virtual terminals” in specific walking distance areas where transfers can occur during a limited timeframe.

The first algorithm for itinerary detection outcomes valid itineraries in most cases – improving other proposals in the state-of-the-art. The results show an increase from 68.83% to 99.33% in itinerary traceability gain when compared with a method that uses bus timetables. This result increases valid data in the database, preventing them from being discarded due to not being associated with any itinerary.

The second algorithm for bus stop clustering groups bus stops in walking distance areas for establishing “virtual terminals” where bus transfers can occur outside traditional physical terminals. An analysis using real-world origin-destination trips in Curitiba revealed that our approach could potentially reduce travel distances significantly. The average distance of 22.4 km traveled in the transit network without clusters is reduced to 12.3 km with clusters. However, it increases the number of transfers by two on average.

The results are limited regarding time estimated at bus stops because road traffic conditions should not affect them significantly when bus stops are located at short distances from each other. Another important limitation is using the correlation between bus time series to measure transfer times. A strong correlation means that buses are more likely to meet each other at bus stops of the same cluster.

Our contribution can enhance the efficiency of bus transit and even attract more people to public transit. Several future works could be done in this direction. For instance, it may be interesting to consider arrival times for evaluating and selecting routes with better synchronization and also allowing travel time to be computed.

Declarations

Acknowledgements

The authors thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), Project Smart City Concepts in Curitiba, IPPUC, and Curitiba City Hall. JB thanks HUB de Inteligência Artificial e Arquiteturas Cognitivas - Brasil (H.IAAC), and Serviço Nacional de Aprendizagem Industrial - Brasil (SENAI).
Funding
JB is funded by the National Council for Scientific and Technological Development - CNPq (grant 142519/2020-0). This research was supported in part by the project SocialNet (grant 2023/00148-0 from Sao Paulo Research Foundation - FAPESP), CNPq (grants 441444/2023-7 and 314603/2023-9).

Authors’ Contributions
JB and AP performed the experiments. JB, TS, AM, and RL helped in the conceptualization of the study and writing of the manuscript. JB is the main contributor and writer of this manuscript. All authors read and approved the final manuscript.

Competing interests
The authors declare they do not have competing interests.

Availability of data and materials
The datasets generated and/or analyzed during the current study are available in https://github.com/jcnborges/busanalysis.git.

References


