Representing multimodal routes in an intelligent traffic system

Tuukka Tolvanen

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo April 23, 2018

Thesis supervisor:

Prof. Heikki Saikkonen

Thesis advisor:

D.Sc. (Tech.) Seppo Törmä

Aalto University
School of Science

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author: Tuukka Tolvanen

Title: Representing multimodal routes in an intelligent traffic system

Date: April 23, 2018 Language: English Number of pages: 6+46

Department of Computer Science

Professorship: Software Systems

Supervisor: Prof. Heikki Saikkonen

Advisor: D.Sc. (Tech.) Seppo Törmä

Automatically detecting individuals' door-to-door multimodal trips has important applications in an intelligent transport system. These include assisting users in multimodal navigation, optimizing the transit network, and more.
Smartphones and other mobile devices today carry a multitude of radios and sensors, including those suitable for detecting location via e.g. Wi-Fi access point mapping or satellite navigation systems, and for detecting motion activity including modes of transport using the accelerometer. Combining these sources with open data from public transport operators, such as static timetables and mass transit vehicle location time series, it is possible to also detect use of mass transit by the smartphone user.
In this thesis project a representation for multimodal routes was developed, suitable for analysis of mobility patterns. The modeling includes prerequisite identification of stops and trips, trip origin and destination, mode of transport and use of mass transit in trip legs, and recognizing the user's regular destinations and routes.
The discovered mobility patterns can further be combined with data from other sources to produce relevant notifications of exceptions in traffic conditions, such as traffic jams, accidents, or public transport disruptions.

Keywords: route representation, destination detection, transportation mode detection, mobility patterns, route recognition

AALTO-YLIOPISTO
PERUSTIETEIDEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Tuukka Tolvanen

Työn nimi: Monikulkutapaisten reittien kuvaus älyliikennejärjestelmässä

Päivämäärä: 23. huhtikuuta 2018 Kieli: Englanti Sivumäärä: 6+46

Tietotekniikan laitos

Professuuri: Ohjelmistojärjestelmät

Työn valvoja: Prof. Heikki Saikkonen

Työn ohjaaja: TkT Seppo Törmä

Ihmisten monikulkutapaisten reittien automaattisella havaitsemisella ovelta ovelle on olennaisia sovelluksia älykkäässä liikennejärjestelmässä. Näihin lukeutuvat mm. dynaaminen opastus monikulkutapaisella reitillä, ja tietojen mahdollistama liikennejärjestelmän optimointi.
Nykyiset älypuhelimet ja muut mobiililaitteet sisältävät moninaisia antureita ja radiolaitteistoa, joita voidaan käyttää laitteen paikannukseen kartoitettujen Wi-Fi -tukiasemien tai satelliittipaikannuksen avulla, sekä liikeaktiviteetin tunnistukseen kiihtyvyysanturin avulla. Kun näitä tietoja yhdistetään julkisen liikenteen palveluntarjoajien tuottamaan avoimeen dataan kuten joukkoliikennevälineiden ajantasaiseen paikannustietoon sekä aikatauluihin, voidaan myös tunnistaa puhelimen käyttäjän joukkoliikennematkoja.
Tässä diplomityöprojektissa kehitettiin monikulkutapaisten reittien kuvaamiseen malli, jota voidaan käyttää liikkumistapojen analyysiin. Mallinnukseen sisältyy edellytyksinä pysähdysten ja matkojen havaitseminen, matkojen alku- ja loppupaikkojen kokoaminen, liikkumismuodon ja joukkoliikennematkojen tunnistaminen, sekä käyttäjän toistuvien päämäärien ja reittien jäsentäminen.
Liikkumistapamallin tietoja muihin tietolähteisiin yhdistämällä voidaan myös tarjota käyttäjälle relevantteja ilmoituksia poikkeustilanteista liikenteessä, kuten merkittävistä ruuhkista, onnettomuuksista, tai joukkoliikennehäiriöistä.

Avainsanat: reittikuvaus, päämäärän havaitseminen, kulkumuodon tunnistus, liikkumiskaavat, reitin tunnistus

Preface

I would like to thank my thesis supervisor Professor Heikki Saikkonen and instructor D.Sc. (Tech.) Seppo Törmä, for their patience and prodding during this at times glacial writing project. The structural and methodological reviews of D.Sc. (Tech.) Kimmo Karhu were also invaluable.

Also thanks are due to the other collaborators within and around the Aalto TrafficSense project for their friendly encouragement and feedback, and refreshing side quests for science. In particular, to D.Sc. (Tech.) Mikko Rinne who was responsible for the mobile aspect of the implementation, and others who contributed to the software before and during my project.

This work was supported by the Foundation for Aalto University Science and Technology, and the TrafficSense project funded by the Aalto Energy Efficiency Programme of Aalto University.

Otaniemi, April 23, 2018

Tuukka Tolvanen

Contents

Terms and Abbreviations

API	Application Programming Interface
DSR	Design Science Research
GeoJSON	a format, based on JSON, for encoding a variety of geographic data structures
GLONASS	Globalnaya navigatsionnaya sputnikovaya sistema, a GNSS
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HRT/HSL	Helsinki Region Transport / Helsingin seudun liikenne -kuntayhtymä
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IT	Information Technology
ITS	Intelligent Transport/Traffic System
JSON	JavaScript Object Notation
Wi-Fi	WLAN technology based on IEEE 802.11 standards
WLAN	Wireless Local Area Networking

1 Introduction

The smartphones of today have come far from the cumbersome talk-only mobile phones of decades past, and their pager siblings. Subsequent development has produced integration of features from text messaging, to clock and calendar functions, note taking and basic games. Now, smartphones have become always present, always connected internet-enabled multifunction devices, with large touchscreens, pocket quality cameras and full-featured operating systems enabling marketplaces carrying vast collections of third-party applications.

The first global navigation satellite systems (GNSS), the United States' Global Positioning System (*GPS*[1]) and the Russian *GLONASS*[2], were developed for military purposes [4]. Today, these systems, along with the European Space Agency's *Galileo*[3], and other regionally operational systems, have been made generally available to enable a variety of public and commercial applications. Receivers in mobile phones, activity wristbands, and vehicles help users to navigate by placing them on a map, to track exercise activities, and allow transport and delivery operators to track vehicle fleets.

In addition to location services, smartphones typically include a host of other sensors, such as accelerometer, ambient light and proximity sensors. The accelerometer in particular allows for recognizing activities performed by a user carrying the device, including modes of transport.

The European Union directive 2010/40/EU [1] defines Intelligent Transport Systems (ITS) as "advanced applications which without embodying intelligence as such aim to provide innovative services relating to different modes of transport and traffic management and enable various users to be better informed and make safer, more coordinated and 'smarter' use of transport networks".

In a road traffic system, cameras can be placed at specific locations to measure traffic levels and identify congestion. Location tracking, on the other

[1] http://www.gps.gov/

[2] https://www.glonass-iac.ru/en/

[3] https://www.gsc-europa.eu/

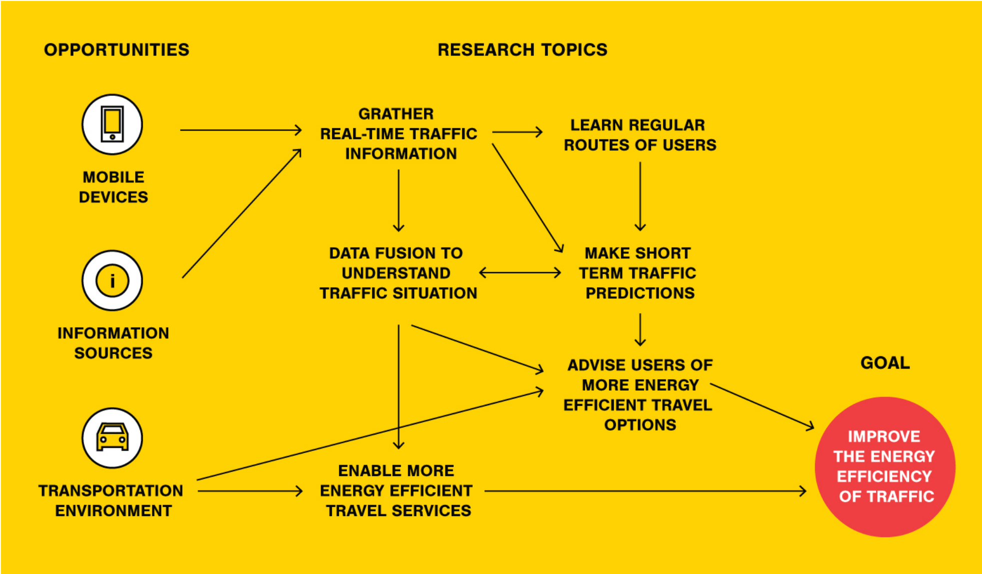


Figure 1: TrafficSense research project context and goals. [3]

hand, can be used for crowdsourcing of traffic congestion status anywhere with a sufficient number of contributing terminals, by comparing movement speed to typical or nominal speeds along the route.

Many public transport operators around the world provide publicly-accessible mass transit data, such as static timetables and real-time vehicle locations.

Going forward, the proliferation of smartphones, and developments in mobile sensing technology built into them, enables detailed analysis of phone users' multimodal mobility patterns. With new traffic-related information services and transportation solutions, the combination can enable improvements in e.g. the efficiency of transportation systems.

The TrafficSense research project aims (cf. Figure 1) at improving the energy efficiency of traffic with a mobile crowdsensing service that learns users' regular door-to-door routes — composed e.g. of walking, biking, driving a car, or riding a bus — and can detect in real-time the likely routes users are following. Previous experiments on mobile crowdsensing in traffic have focused on private cars and route choices of individuals; the novelty of TrafficSense is in its focus on energy efficiency of the whole traffic system and the coverage of multimodal, door-to-door routes. [3]

The objective of this thesis project is to develop representations for multimodal routes, suitable, in the context of TrafficSense and similar systems, for analysis of mobility patterns. This includes prerequisite identification of stops and trips, trip origin and destination, mode of transport in trip legs, and recognizing the user's regular destinations and routes. The discovered mobility patterns enable combining data from other sources to notify the user of exceptions in traffic conditions, e.g. traffic jams, construction work, mass transit outages and exceptions. There is also potential to improve energy efficiency by suggesting mass transit routes or ride-sharing corresponding to the user's predicted destinations.

This document covers the prerequisite background, detailed technical environment, as well as implementation and evaluation of a model of stops, trips, regular destinations, routes and their transport modes, generated from a user's location and activity trace collected using a mobile device. The methodology and structure of the of the document is described in the following chapter.

2 Method

As this work concerns feature development, it can be described in terms of the Design Science Research (DSR) framework laid out by Hevner et al. [13]:

> Information Systems are purposefully designed human-machine artifacts that significantly impact people, organizations, and society. Two paradigms characterize research in this discipline: behavioral (or natural) science and design science. Whereas the behavioral science paradigm seeks to discover and verify laws or principles that explain or predict human or social behavior, the design science paradigm seeks to extend the boundaries of human and social capabilities by creating new and innovative artifacts.
>
> Research in IT that uses a behavioral (natural) science paradigm is fundamentally reactive. Its goal is to identify and codify emergent properties and laws governing human and organizational behavior as it affects and is affected by existing information technologies. Research in IT that uses a design science paradigm is fundamentally proactive. Its goal is to create innovative artifacts that extend human and social capabilities and aim to achieve desired outcomes. These artifacts often define the object of study in behavioral IT research.

This framework for understanding, executing, and evaluating design science research in the Information Systems field is illustrated in Figure 2.

The environment defines the problem space and context, composed of the business and technology aspects. The knowledge base provides the materials for constructing and evaluating artifacts. Here, the business environment described in the introduction is further related to the knowledge base background and foundations in the following chapter. The specific technological environment of the TrafficSense system is described in Chapter 4.

The constructed design artifact implementation of data models, prerequisite and other improvements to the system are presented in Chapter 5.

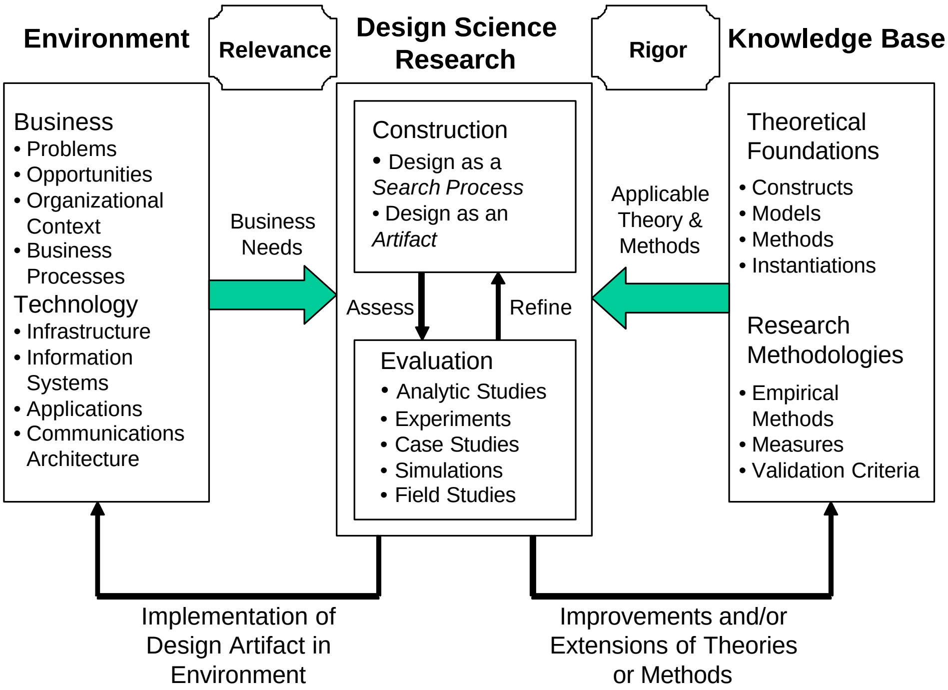


Figure 2: Design Science Research Framework, Hevner et al. [13, fig. 2, p. 5]

Hevner et al. [13, p. 15, table 1] also catalog methods for evaluation of design artifacts under the Design Science Research framework.

On the observational front, the implementation in this work is subject to a field experiment with a total of 93 distinct users contributing some data in the span of 39 days. Also a smaller-scale one-day targeted data gathering and evaluation was done.

Analytical evaluation discusses the fit into the technical context — the resulting model of stops, trips, regular destinations, routes and their modes, generated from a user's location and activity trace, is assessed to evaluate the suitability of the model. Under dynamic qualities the performance of accessing processed data used for display to the user by the client software, is evaluated, as well as the efficiency of storage.

These evaluations are described in Chapter 6, followed by conclusions and further work in Chapter 7.

3 Background

A representation of a multimodal route should allow effective comprehension of different transport modes within trips. For regular routes, it is of interest to compare trips by their origins, destinations, modes of transport, and paths taken. The representation should also allow efficient visualization of the discovered trips, routes and regular destinations. Some of the use cases for the multimodal route model are outlined in Figure 3. The background concepts within the problem subdomains that arise from these requirements are introduced in this chapter.

Geolocation

To begin to analyze an individual's mobility patterns, it is necessary to capture their location as a time series, hereinafter called the *location trace*. This could take the form of a manually recorded journal, but such recording is quite tedious, as can be resolving geographic locations from manual input. Automatic data collection on the other hand can produce more consistent and fine-grained data. As mentioned in the introduction, there exist a variety of methods of determining mobile phone location.

Zhao [28] foresaw the technologies of locating a mobile phone having a significant impact on automotive telematics and modern public transit systems, once these technologies are mature enough to be deployed. A particular driver was the ability to locate the caller to an emergency number. Radio-based technology typically uses cellular base stations, satellites, or devices emitting radio signals to the mobile receiver to determine the position of its user.

The coverage of satellite-derived location can be intermittent especially around large obstacles, or underground. Mobile phone location can be also determined to a lesser accuracy, but with also less power demand, by the cell tower in use, and nearby Wi-Fi access points.

Of the typical mobile phone geolocation methods, GPS and other satellite

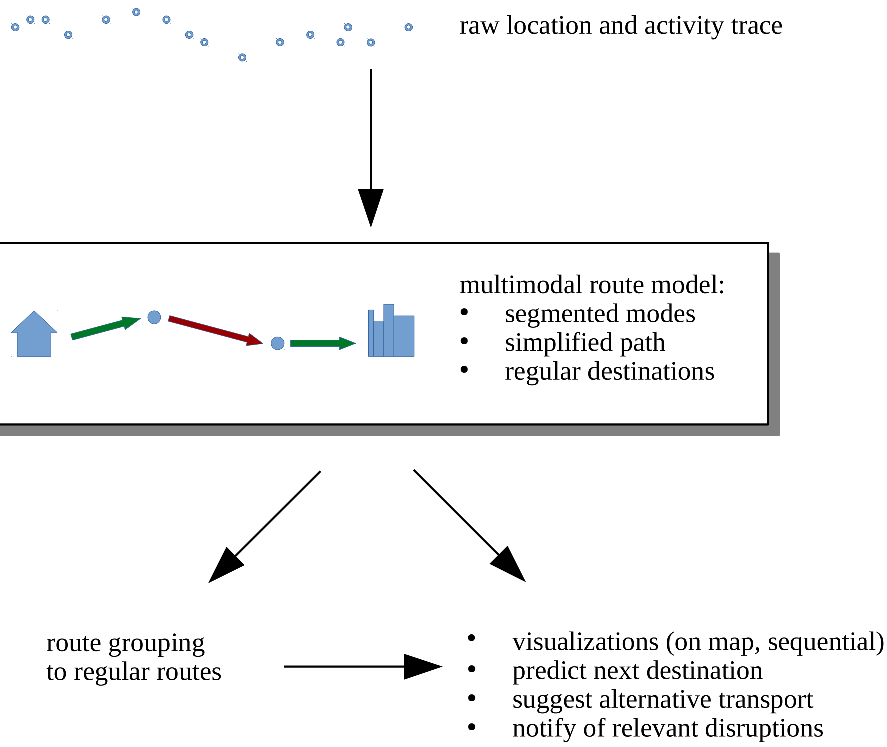


Figure 3: Use cases for multimodal route model.

signal based systems can be more accurate at their best, but Park et al. [22] found location reports based on mapped Wi-Fi base stations more accurate in an urban canyon environment, which is relevant for observing transit patterns in an urban context.

Trip and route features

To describe and reason about an individual's mobility patterns, it is necessary to establish definitions for the various features in the model. Here, a *stop* is a pause in movement more significant than e.g. traffic lights. Letchner et al. [19] used a five minute threshold for detecting a stop in journey segmentation. The duration is not a completely accurate proxy for significance however, as e.g. stopping for only a few seconds to drop off a passenger will be missed by the criterion.

A *trip leg* is defined as contiguous movement such that no above described stops are detected within it, and using the same mode of transport throughout. A *trip* is a sequence of such legs between significant stops, possibly containing also short stops for e.g. mass transit transfers.

Strathman, James G and Dueker, Kenneth J [25] define a *trip chain* from as travel involving multiple purposes to single or multiple destinations and begins and ends at home, or a similar origin — for example, a commute consisting of a trip to pre-school, followed by a trip to work.

A trip chain at its most basic level includes a stop on the way to another destination. McGuckin and Nakamoto [21] describe an operational definition by the U.S. Federal Highway Administration of a trip chain as a sequence of trips bounded by stops of 30 minutes or less. A stop of 31 minutes or more defines the terminus of a chain of trips, and that chain of trips is considered a *tour*.

For the purposes of this work, a *route* is established by the path and modes taken in a trip chain instance from the *origin* to the *destination*, and as such can be repeated, or routes with alternate path or modes taken between a given pair of origin and destination.

Activity recognition

While many aspects of an individual's mobility patterns could be derived from a location trace alone, determining the mode of transport is facilitated by recognizing the physical activity being performed.

Activity recoginition takes wearable or mobile phone sensor readings as input and recognizes a user's motion activity. It is the core building block in many applications, ranging from health and fitness monitoring, personal biometric signature, urban computing, assistive technology, and elder-care, to indoor localization and navigation. [26]

The most widely used inertial sensors for activity recognition are accelerometers and gyroscopes. An accelerometer consists of a mass suspended by a spring and placed in a housing, and the displacement of the mass is measured as the difference of acceleration. A gyroscope sensor measures angular velocity by using the tendency of vibration in the same plane as an object vibrates. [6]

Activity recognition systems based on accelerometer data employ statistical feature extraction and, in most of the cases, either time- or frequency-domain features [18]. For example, Ilomäki [16] explored recognizing transportation modes by multivariate clustering of accelerometer data.

Detection of mass transit use

An important problem in creating efficient public transport systems is obtaining data about the set of trips that passengers make. Kostakos et al. [17] described a wireless system using a Bluetooth scanner to wirelessly detect and record end-to-end passenger journeys.

Ekholm [10] explored the feasibility of recognizing tram trips by matching location traces from mobile phones and trams. This is one of the approaches taken in the TrafficSense system.

Recognizing regular routes and destinations

To enable navigation suggestions and comparison or prediction of user routes, it is necessary to recognize regular destinations visited and routes taken.

González et al. [11] found that human trajectories show a high degree of temporal and spatial regularity, each individual being characterized by a

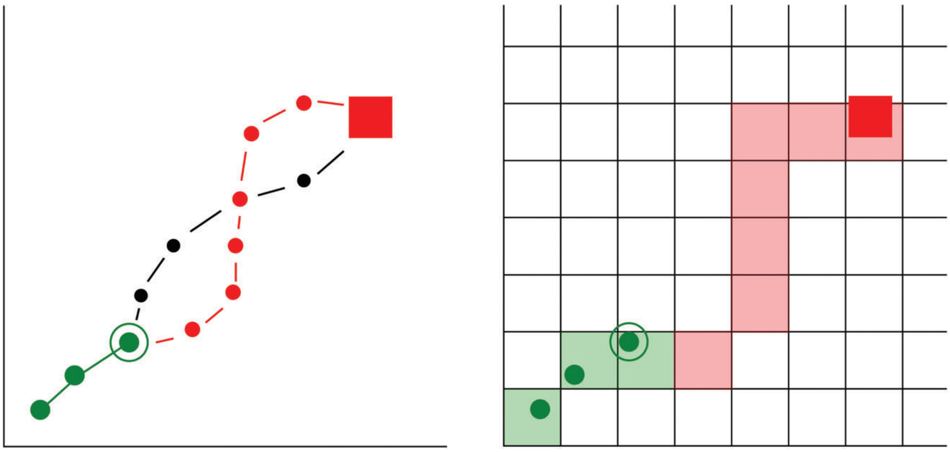


Figure 4: Generalized path representation options, network of waypoints (left) vs grid (right). [24, fig. 2, p. 4]

time-independent characteristic travel distance and a significant probability to return to a few highly frequented locations.

Liao et al. [20] show an approach for how to extract and label a person's activities and significant places from traces of GPS data. For instance, while walking, driving a car, or riding a bus are not associated with significant places, working or getting on or off the bus indicate a significant place.

In order to generalize traces to discover regular routes taken by users, it is practical to have a representation of the path that is coarser than a raw location trace as recorded by a mobile device. Various approaches to this end are possible, such as coordinate quantization into a grid, or a road/transit network based waypoint and link model. These are illustrated in Figure 4. [24].

The simplified representations of traces facilitate applications like predicting the next route or destination. For example, Letchner et al. [19] used traces snapped to a road network graph to produce such individualized route prediction based on a driver's preferred routes.

4 Environment: The TrafficSense System

The TrafficSense system consists of a mobile client for data collection and display to the user, as well as server software implementing storage, analysis, and web views. An overview of the system components and the data flow between them is shown in Figure 5.

A time series of the raw location and recognized activity are captured by the TrafficSense mobile client software using, respectively, the Fused Location Provider and Activity Recognition APIs provided by *Google Play Services*[4] on Android.

The captured data points are buffered, then sent to the server in batches, and stored in a relational database. The server then periodically post-processes the collected traces, finding contiguous vehicle transit mode legs to identify mass transit lines taken. The result is a filtered trace of the detected contiguous spans. These processes are further described in this chapter.

The server side software consists of the *apiserver*, *siteserver*, *devserver*, and *scheduler* components. These are implemented in Python, using the PostgreSQL relational database system with the PostGIS extension for spatial and geographic objects.

The apiserver implements an HTTP API for use by the TrafficSense mobile client. It handles device registration and authorization using Google Sign-In, and allows the client to upload collected activity and location data. The client can also fetch processed data for display to the user, e.g. the path taken on a map, or a graphic evaluating the energy efficiency of the user's travels.

The siteserver implements the service web site, which also provides a map view and the energy efficiency visualization. The devserver provides additional views and reports for development purposes.

The scheduler runs periodical tasks, such as fetching data from various sources — mass transit fleet locations, road and mass transit disruption bulletins,

[4] https://developers.google.com/android/guides/overview

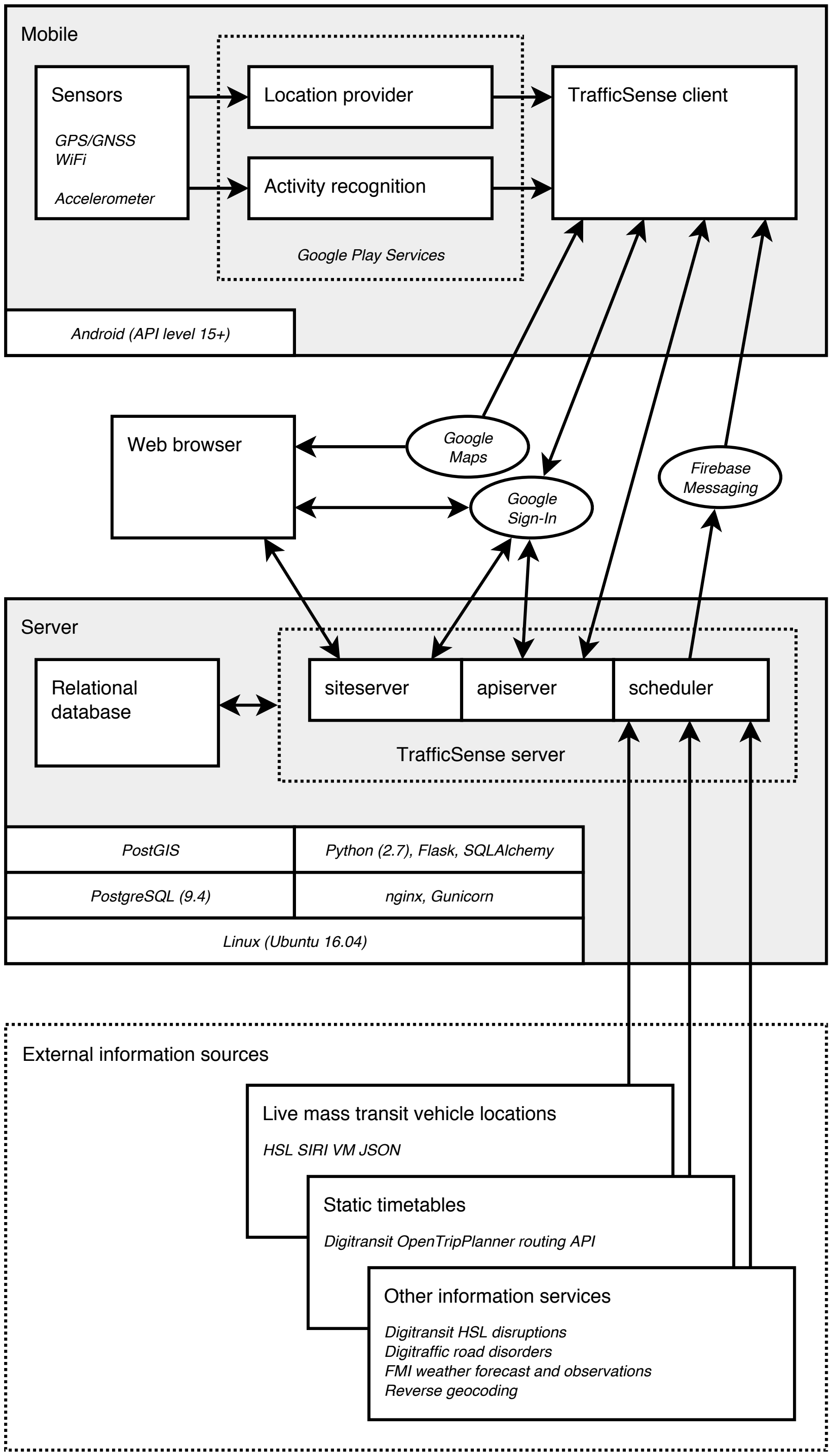


Figure 5: Overview of system components and data flow.

weather forecasts and reports — and batch processing of incoming data uploaded by the client instances. The scheduler may also push notifications of disruptions relevant to the user's mobility history to the client application when appropriate.

4.1 Client data collection

The TrafficSense mobile client captures a location and activity trace. A trace record includes a timestamp, an identifier for the device, the latitude and longitude, an accuracy radius in metres, and the top three activities identified as most likely, with a confidence indication for each activity. The client process is described in greater detail in Rinne et al. [23].

Activity recognition

The first part of inferring the user's mode of transport is done by using the accelerometer of the mobile device to recognize the present activity. This is achieved in the TrafficSense client using the *Activity Recognition API*[5] provided by Google on Android.

The possible activities returned by the Activity Recognition API are IN_VEHICLE, ON_BICYCLE, ON_FOOT, RUNNING, STILL, TILTING, UNKNOWN, and WALKING [12]. Several activities may be returned, along with a confidence rating ranging from 0 to 100 for each. The three activities with the highest confidence are stored.

Location tracing

The user's location is captured by the TrafficSense client using the *Fused Location Provider API*[6] of Google Play Services. When the detected activity is not STILL, the location is sampled at a 10 second interval with "high accuracy" requested. If the detected activity becomes, and remains, STILL, the client enters SLEEP state, where the location requests are dropped to "no power" priority.

In smoothly moving rail transport, the activity recognition may report STILL. This is why the location is sampled at "high accuracy" for an additional 40 seconds after transition to STILL. If the location changes for a distance greater than the reported accuracy of the location fix, the timer is restarted, and "high accuracy" maintained.

The Fused Location Provider API uses several underlying providers, such as Wi-Fi, cell tower positioning and GPS, as available [5]. This means the accuracy of the resulting trace can vary from a few metres to kilometres.

A client-side filter, illustrated in Figure 6, removes location updates where the stated accuracy radius of the point, given by the location provider, is greater

[5] https://developers.google.com/android/reference/com/google/android/gms/location/ActivityRecognitionApi

[6] https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderApi

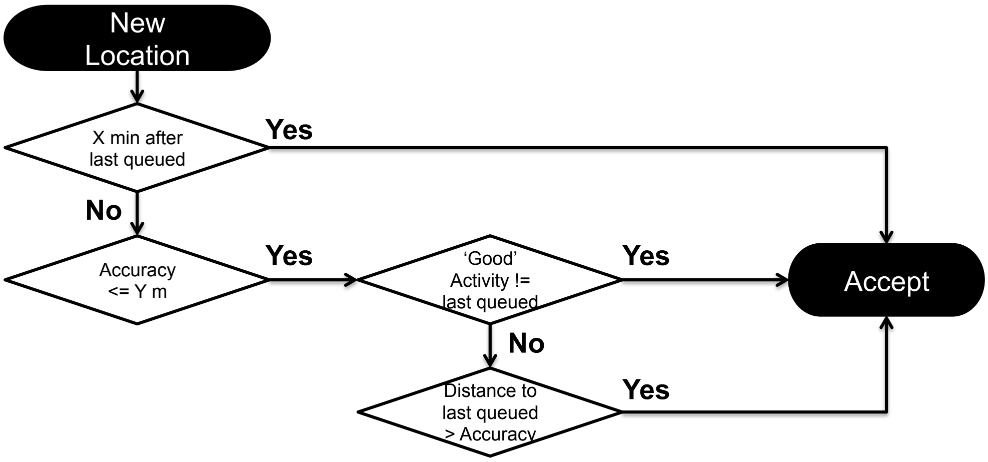


Figure 6: Client side filtering algorigthm for incoming position fixes. [23, fig. 1, p. 82]

than the distance from the prior accepted point. This way points that by their stated accuracy do not establish that the client terminal has definitely moved, are discarded. Also points whose reported accuracy is poorer than a threshold, set at 50 metres by default, are discarded.

4.2 Server processing

The data points received from the client by the apiserver are stored in a relational database. The scheduler component then periodically analyzes the collected traces to split contiguous transit mode legs, and identify mass transit lines taken.

To recognize mass transit usage, the live mass transit vehicle locations and static timetables, in conjunction with the user location traces, are used to automatically recognize mass transit trips taken by users. Contiguous spans, after post-process filtering, of IN_VEHICLE activity, are matched against the vehicle position traces and timetable data.

The result is a filtered trace of the detected contiguous spans, as shown in Figure 7.

The description of mass transit detection methods here closely follows that in Rinne et al. [23], which goes into more depth especially with regard to the static timetable matching algorithm.

Contiguous activity splitting

The activity recognition captured in the received trace is noisy, and often includes spurious transitions. The initial implementation addresses this by selecting the activty with highest reported confidence at each sample point, then requiring six consecutive samples — typically one minute at the ten second sampling interval — of the same activity to change activity states in the

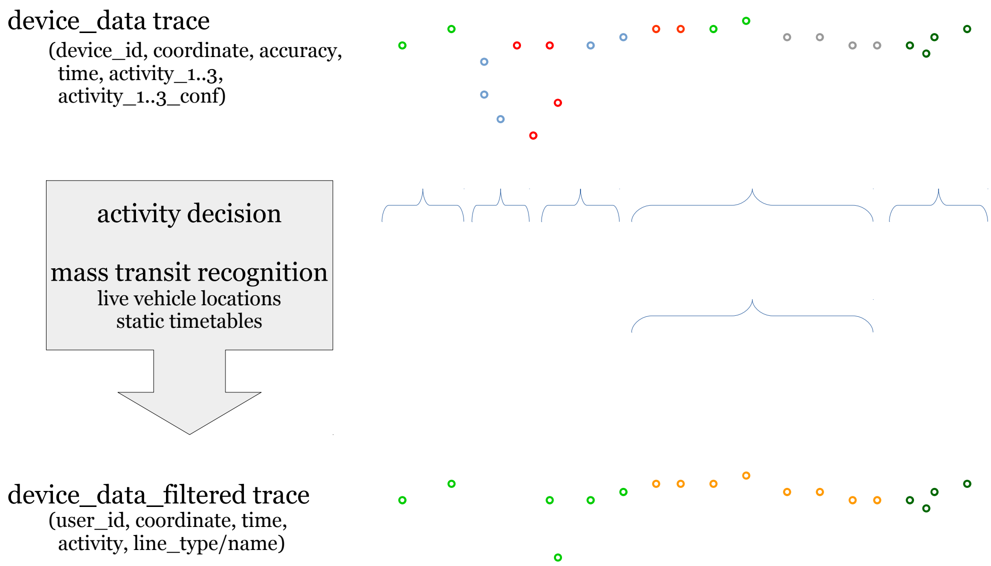


Figure 7: Filtering process for received device data.

resulting filtered trace, splitting any indeterminate region between consecutive stable activities in half. In addition, points more than five minutes apart in the trace, either due to the terminal being stationary or lacking location reception, causes a segment break. Segments where no stable activity is found, are omitted.

Mass transit detection using live vehicle locations

The latest locations of part of the mass transit vehicle fleet are made available online by the local transport provider[7]. This data is sampled at 30-second intervals, interpreted from JSON, and stored in the database. The vehicle locations can then be compared with the user locations collected by the Android app.

The collected location records contain a timestamp, the latitude and longitude, a unique vehicle reference, the name of the transit line, and a line type of BUS, FERRY, SUBWAY, TRAIN, or TRAM.

Potential issues in matching include:

- Missing vehicle or user data points
- Inaccurate location points
- Clock differences
- Distance between user and vehicle location sensor in longer vehicles
- Distance between location samples at higher speeds

[7] https://www.digitransit.fi/en/developers/services-and-apis/4-realtime-api/vehicle-positions/

- False matches to other mass transit vehicles
- False positives where car trip occurs near mass transit vehicles
- Intermittently changing line name label on some vehicles in the live data

To counteract the location accuracy and vehicle length issues, a distance limit of 100 metres was used for collecting vehicle matches. A greater limit may cause more false positive matches to appear.

The vehicle location points for comparison are collected in a ±60-second window around the timestamp of each user point sample. This allows for some clock difference, and sampling time difference.

For a vehicle to be accepted as a possible match, its must be within the 100 metre distance limit for a minimum of 75% of the user location samples.

In the initial implementation, four sample points from the user's vehicle leg were considered, and the vehicle with the most matches wins. In case of a draw, the vehicle with the least total distance between matched user and vehicle points wins.

Mass transit detection using static timetables

For finding mass transit trips in IN_VEHICLE trip legs, the first and last point of the contiguous activity are used for the time and place of the origin and destination, respectively. They are used in a query to the journey planner interface of the local transport provider[8], and the resulting plans matched against the user location trace.

Potential issues in matching include:

- Missing or inaccurate user location points
- Inaccuracy in activity determination and filtered transition points
- Clock differences
- False positives where car trip occurs near mass transit line

The device data can often have a fair amount of inaccuracy in location and activity detection. For this reason the filtered transition point starting the vehicular trip leg can often appear between stops during the actual mass transit leg. Adjacent stops are assumed to be no more than one kilometre apart. To account for inaccuracy in time and place of the origin, the earliest permissible start time of the trip is adjusted back by 6.2 minutes, allowing 500 metres of walking at a speed of 1.3 m/s. Correspondingly, to relax both the origin and destination location constraints, the query to the journey planner specifies a maximum of total 1000 metres of walking allowed in each resulting plan.

[8] https://www.digitransit.fi/en/developers/services-and-apis/1-routing-api/x-service-architecture/

From the three plans requested from the planner, the best match, if good enough, is chosen.

Only plans containing a single vehicular leg are considered for matching. Quick transfers between vehicles may not be detected as separate trips by the activity detection and filtering determination, in which case a corresponding plan cannot be found.

Plans that have a duration more than 3 minutes shorter than the user vehicle leg, are discarded to avoid false positive matches, so assuming that the transit vehicle must travel closely according to schedule.

Plans that have a duration more than 18 minutes longer than the user vehicle leg, are also discarded. This includes the walk times for dealing with location inaccuracy, discussed above, and the time assumed for a mass transit vehicle to travel between adjacent stops.

Plans where the duration of the included vehicular leg differs by more than 5.6 minutes from the user recorded vehicular leg, are discarded, based on assumed time of travel of the vehicle between adjacent sparse stops.

Plans where the start of the included vehicular leg differs from that of the recorded leg by more than 5.8 minutes, are likewise discarded.

The returned plans include a location point sequence of the planned trip. This is also matched against the recorded user trace. For comparison, the recorded user trace is sampled at no less than 100 metre intervals, and leading and trailing points may be ignored based on assumed inaccuracy of origin and destination. A minimum of 70% of sample points must match a plan point within 100 metres, and no more than four consecutive sample points may fall outside 100 metres of the plan points, for the plan to qualify.

Of the qualifying plans, the one with the closest start time to that of the recorded leg wins.

5 Implementation

This chapter covers enhancements and features developed as part of the work.

A more compact model for trips was developed, identifying stops and trip legs of different transport modes as distinct entries. The stops were clustered into regular destinations and trips grouped into regular routes. Some prerequisite and related improvements were also implemented.

The location trace reported from devices sometimes includes incorrect location points, where the reported location is far outside the reported accuracy radius from the real location. This apparent movement would cause problems to stop detection. Also trip legs would be affected by inflated distance, and incorrect geometry making visualizations and route grouping less reliable. False location filtering was implemented to elide these points.

Another raw data quality handling improvement was made to the determination of contiguous activity, particularily for cases where the real activity is reported intermittently and with a weak confidence level.

The performance of mass transit detection from live position data was improved, as well as its accuracy especially in case of vehicles moving at relatively high speeds.

These enhancements are described in more detail in this chapter, followed by the route model features and applications.

5.1 Conceptual model and schema

A more compact model for trips was developed. This model records stops and trip legs having different transport modes, as discussed in the background (Chapter 3), as distinct entries. An overview of the updated process is shown in Figure 8.

The resulting relational database schema is shown in Figure 9. Here, both stop segments and movement segments with their detected contiguous activity, found in the location and activity trace, are identified in the *legs* table. The *modes* table records opinions on the mode of transport for each movement leg,

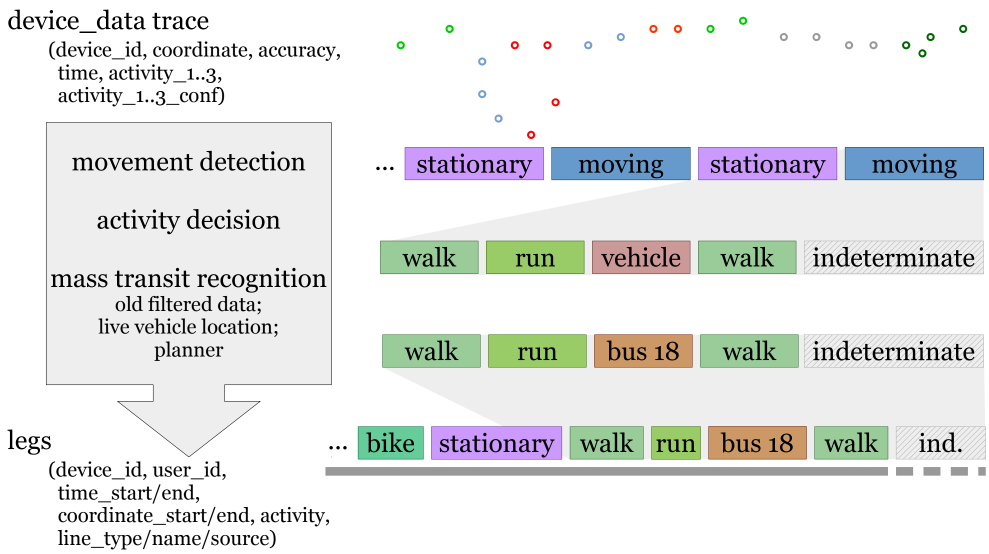


Figure 8: Updated process.

as claimed by different sources: the mass transit matching using *live* vehicle positions, the mass transit matching using static timetables via the journey *planner*, or input from the *user* themself via the web or mobile app.

When a user has multiple devices collecting data concurrently, the legs are first processed per device. To create a single leg sequence for the user, the more frequently changing legs are selected, with the assumption that a device that is, for example, left at home, will not produce activity transitions on its own. This is achieved by sorting the legs from multiple devices in ascending order by their ending timestamp, and selecting those legs that do not overlap in time with previously selected legs. The device legs selected to be part of the user leg sequence have *user_id* set, while for unselected legs it remains null.

The user's stop locations, along with the start and end coordinates of each movement leg, are clustered in the *leg_ends* table, enabling queries to the user's regular destinations and transfer locations. The user-specific leg_ends are further clustered into *places* that are common to all users, and automatically labeled using reverse geocoding.

The trip chains between longer stops, referenced as the *origin* and *destination*, are collected into the *trips* table. As the origin and destination reference longer stops (cf. Section 5.3), and only leg instances that are not longer stops reference a trip, the relationship is not circular on the instance level. A condensed location trace, snapped to *waypoints* at road crossing clusters derived from OpenStreetMap data, is recorded in *leg_waypoints*. The waypoint trace records the timestamp from the first raw trace point in a run of consecutive points snapping to a given waypoint. A coarser path representation such as the waypoint trace, along with leg transport mode information, can then be used to dynamically cluster the trips into the regular route options used between a given origin and destination.

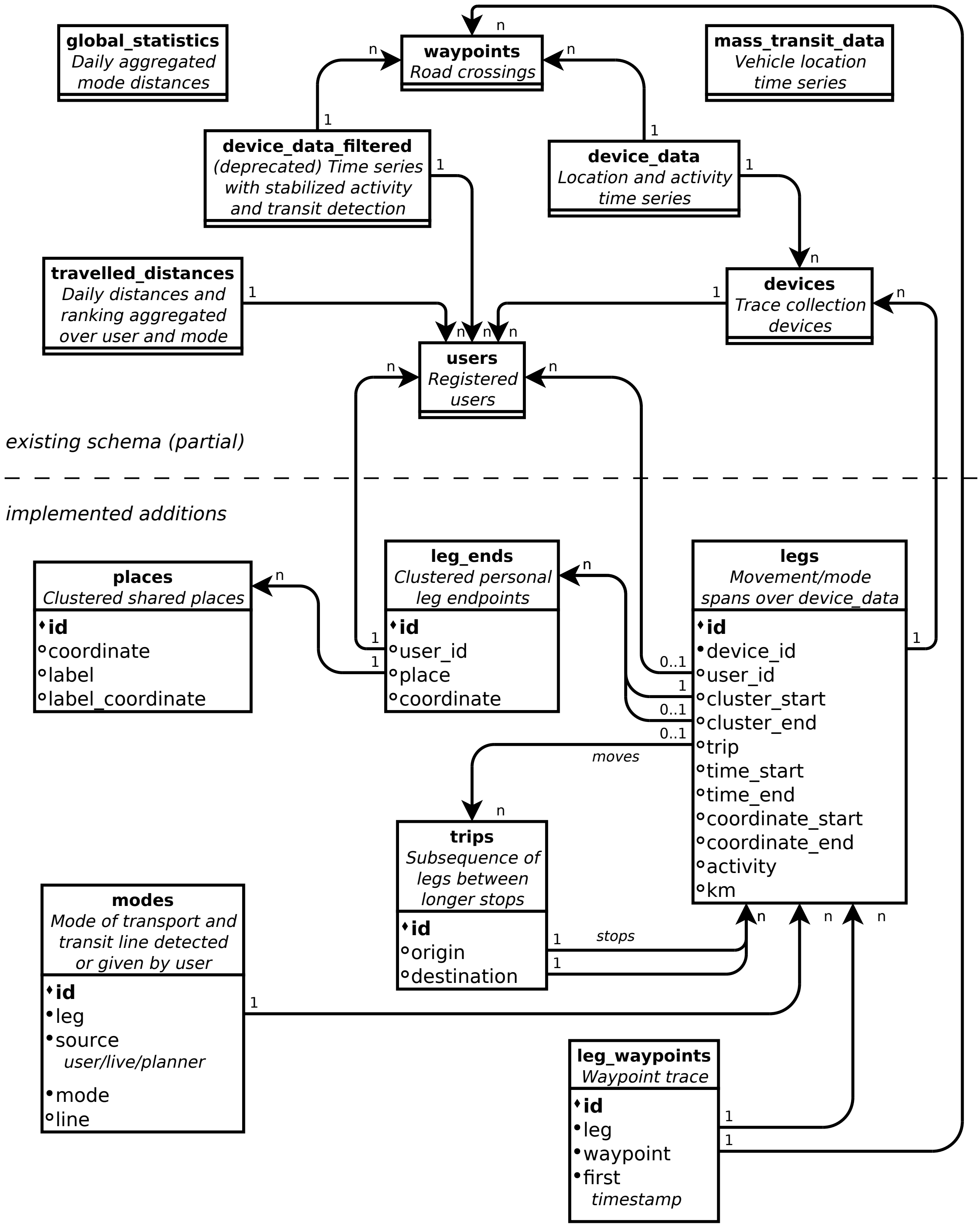


Figure 9: Database schema.

The following sections describe the stop detection, mode detection, clustering, and related improvements in processing order.

5.2 False location filtering

Due to redundancy, noise and outliers from inaccuracy and errors in location source data, some filtering of the data is required.

In some circumstances the captured location trace included data points where the location is incorrect by a much greater distance than indicated by the stated accuracy of the data point. The context of their appearance suggests they may be caused by the location provider finding the location by a mobile Wi-Fi access point that has since moved, but the cause was not investigated further.

Figure 10 shows one such trace from clockwise trips from Kamppi, Helsinki (at right) to Otaniemi, Espoo (at left) and back, with two intermittent false locations. The true roadway is shown in grey outline.

Chen et al. [7, p. 4] describe five trace cleanup filters based on duplication, speed, acceleration, total distance, and angle. Some of these methods were considered or trialled but found not entirely satisfactory for the particular pathology of this case. This was due to difficulty finding parameters specific enough to the false location points, such that as much of the real trace can be preserved, while getting the false points removed in the most cases. While such filters can identify many of the suspicious leaps in the trace, a more specific approach was needed to identify the true from the false side of the trace, especially in a short window over the trace. A filter using a four-point window was implemented based on the observed characteristics of the false location points.

The intermittent false location points were found to be clustered very close together, well within one metre from each other, except for one case where the shift was nearly eight metres. The true side of the trace typically involves more movement, or greater spatial noise even when the collecting terminal is stationary.

To simplify further processing, the trace is preprocessed to group subsequences of points no more than ten metres from the preceding accepted point, leaving the first point in such a subsequence as a representative of the group. After filtering, point groups whose representative is not filtered out, are reintroduced.

The approach then uses a triangle inequality factor criterion, such that it considers a point p_i in the sequence suspect, if the distance between its neighbors p_{i-1} and p_{i+1} through the point exceeds twice the direct distance between the neighbors:

$$\frac{\|\mathbf{p_i} - \mathbf{p_{i-1}}\| + \|\mathbf{p_{i+1}} - \mathbf{p_i}\|}{\|\mathbf{p_{i+1}} - \mathbf{p_{i-1}}\|} > 2$$

A solitary suspicious point, not followed by another suspicious point, is

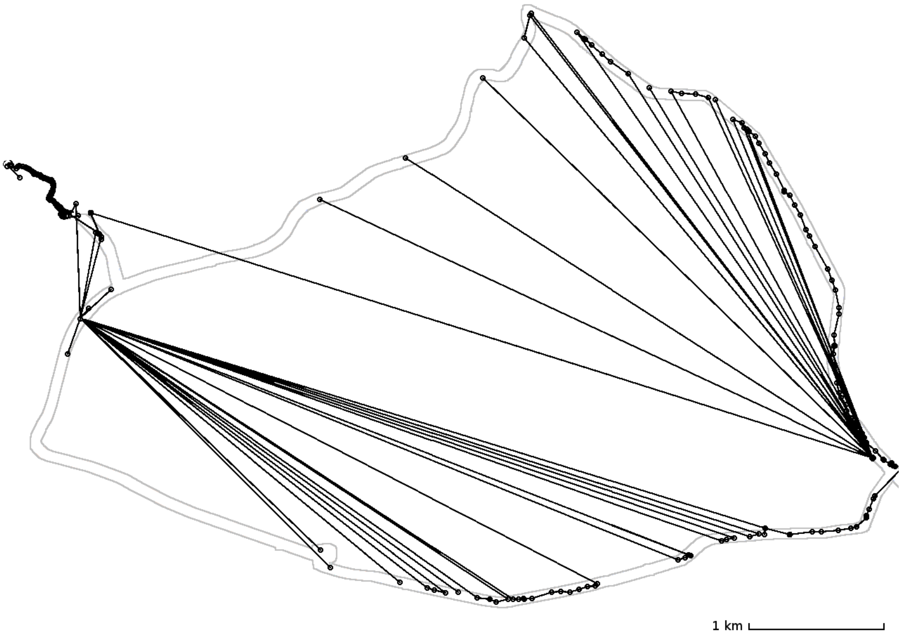


Figure 10: Unfiltered location trace with two intermittent false locations

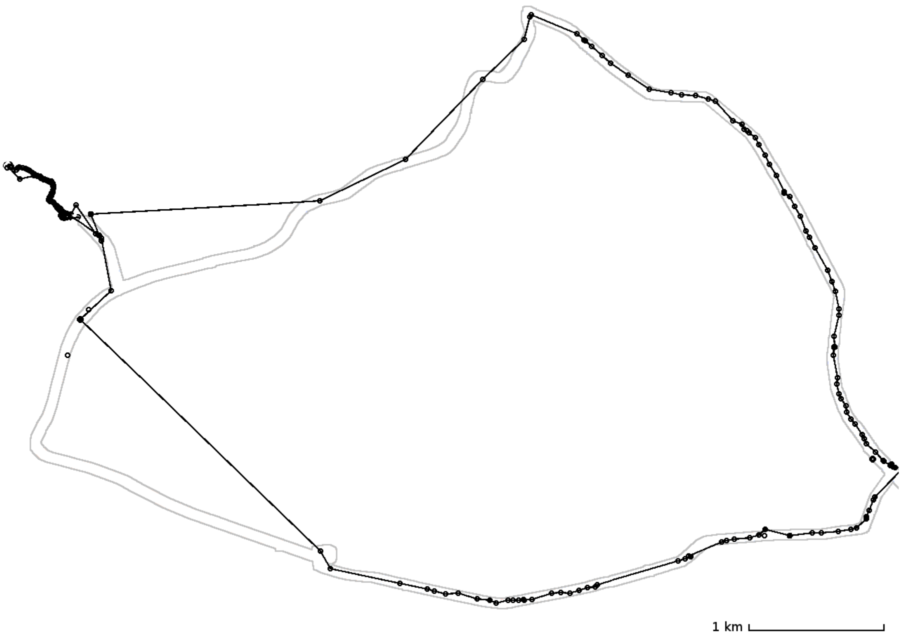


Figure 11: Trace of Figure 10 after filtering.

filtered out. If a suspect point is followed by another suspect point, the lesser movement and noise on the false side of the trace causes the direct distance between the two neighbors of a point to typically be smaller, resulting in a greater triangle inequality factor for the true side, so the point with the lesser factor is filtered out.

Figure 11 shows the example trace after filtering. Two good location points are seen detached from the path on the left, but otherwise the specificity is good.

5.3 Stop detection

Stops are detected in the location trace where the time spent traversing a given distance exceeds a threshold. An example is illustrated in Figure 12.

The distance parameter establishes a granularity for stops and for destinations derived from the stops, as consecutive stops occurring within up to twice that distance could be interpreted as one stop. In consideration of the transit focus and routing applications, the distance parameter was selected with the intent to, in most cases, uniquely identify subsequent transit stops along any line. While the HRT planning principles for the Helsinki region mass transit recommend that the distance between stops not be less than 300 metres within the street network [15], intervals of less than 200 metres do occur in the most dense areas within the region.

Also the typical accuracy of the location trace points establishes a lower bound for a useble distance parameter. To prevent inaccurate location points from breaking up a stop, such points are discarded that claim an accuracy of worse than half the distance parameter.

With these factors in mind, the distance parameter was set at 100 metres, corresponding with the accuracy threshold for accepted location trace points set at 50 metres.

In this implementation, the stop detection time parameter was set to five minutes, same as was used for journey segmentation by Letchner et al. [19]. In informal exploration using collected data, this threshold appeared to effectively eliminate non-destinations like traffic lights, while preserving more purposeful stops as much as possible.

As another consideration to parameter selection, the chosen distance and time parameters together in conjunction with the stop detection approach used establish a lower bound on velocity of 0.33 m/s, or 20 metres per minute, that can establish a movement leg distinct from a stop.

Further experimental analysis derived from the TrafficSense data regarding the time and radius parameter selection is found in Stegmann et al. [24].

Entry and exit refinement

Given a stop where the location trace enters along a line, stays put for at least the duration parameter, and exits again linearly, the stop entry and exit

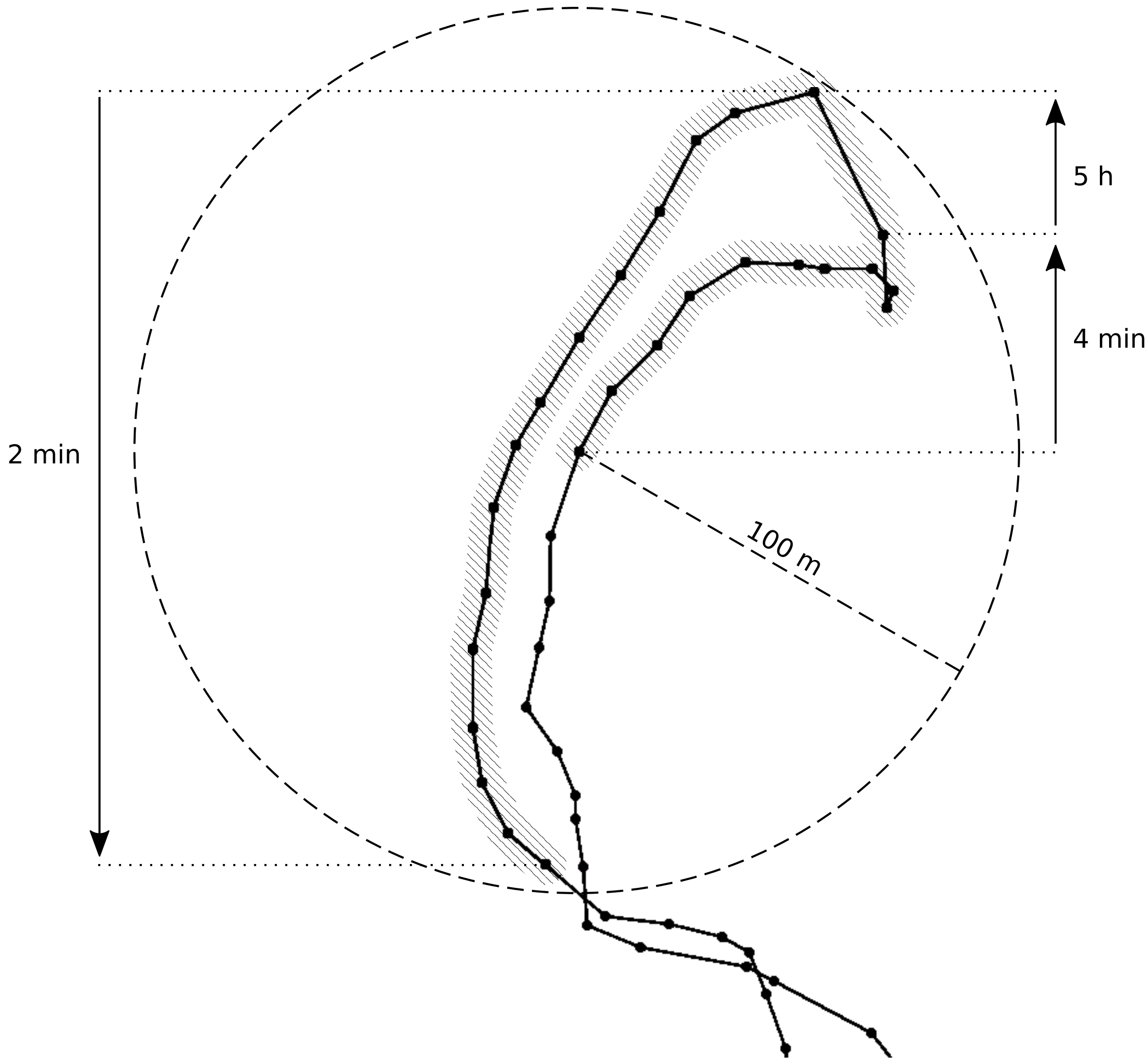


Figure 12: Stop detection in an example trace. Arrows indicate the direction and duration of travel in the corresponding portion of the trace. The shaded portion indicates the detected stop condition where the trace stays within 100m of a base point in excess of the threshold interval. The refined stop consists of the points at the ends of the 5h arrow.

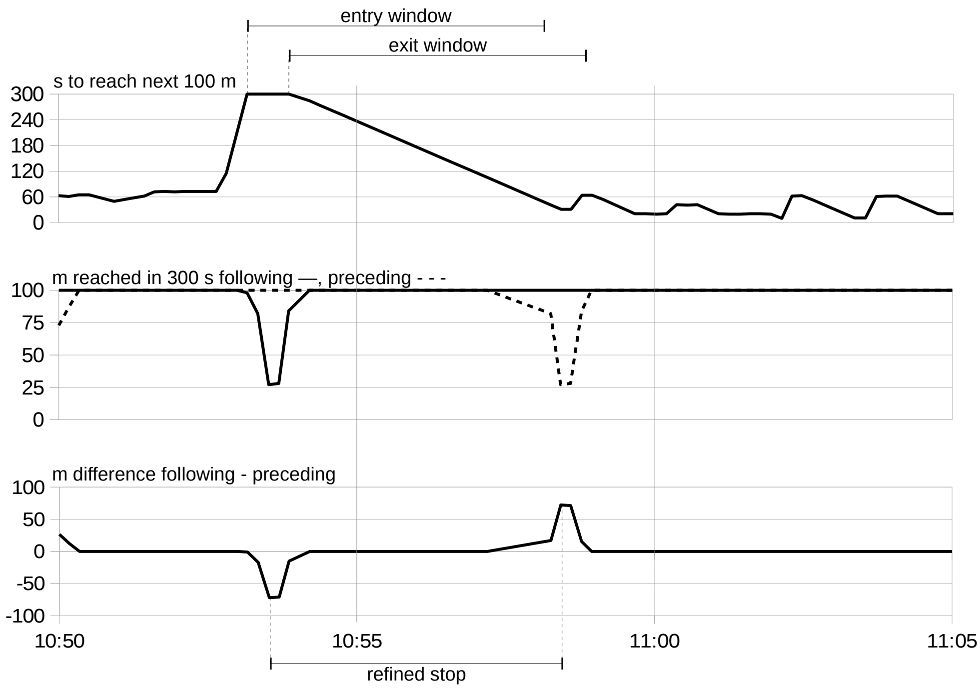


Figure 13: Example of stop entry/exit refinement for a short stop waiting for a bus. In the top graph, the stop condition is found when the time spent to reach the next 100 metres exceeds five minutes (capped to five minutes in graph). The stop entry point is sought within the five minutes after the stop condition begins, and the exit point within the five minutes after the stop condition ends. The middle graph shows the distance reached in the following and preceding five minutes, capped to 100 metres. Finally, the refined entry and exit point is found at the minimum and maximum difference of the capped distances within the entry and exit windows, respectively.

points would both be a whole distance parameter away from the actual stop location. With a distance parameter chosen for suitable granularity of stops, the inaccuracy in entry and exit time and location poses a problem for mass transit detection and other analysis.

This gives rise to the need to refine the stop entry and exit points. The refinement can be accomplished by running a two-pane window with the parameters exchanged, that is, the duration criterion used as window size and the distance travelled in the window panes as the measure. The points where the difference in distance travelled between the panes is the greatest, are used for entry and exit points.

To avoid window edge effects from changes in approach speed due to corners and traffic lights further away from the actual stop, the refinenement distance measure is also clamped to the detection distance criterion.

The stop entry and exit refinement process is illustrated in Figure 13.

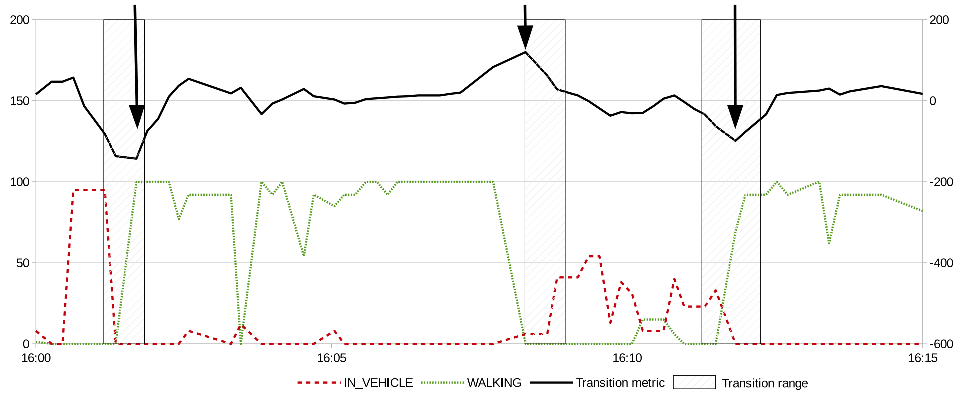


Figure 14: Activity transitions in an example trip. Collected IN_VEHICLE and WALKING confidence values (left scale), $\pm 60s$ WALKING to IN_VEHICLE transition metric (right scale). Confidences for other activities are omitted. Arrows indicate selected next activity start point at the extremum metric within each transition range.

5.4 Activity selection

As part of the work, the following improvements to activity recognition were implemented.

While the activity collection itself makes no use of location, the stop detection with entry and exit refinement allows the activity selection to work on only the movement segments of the trace, which reduce spurious activity bleed from stationary activities into the movement segments.

The goals of the system do not have much use for very short mode legs of less than a minute or so in duration. Also, the previously implemented activity stabilizer already had a six point stability requirement, corresponding to a minimum of one minute at the $10s$ collection rate. To help the stabilizer stabilize over noisy data, a summing window over the activities recorded in data points within $\pm 60s$ of a point was implemented. This way, any longer than one-minute stretch of an activity collected at full confidence will appear in the trace for stabilization, while activities with lower duration or confidence may be elided.

In particular, it was found that during trip legs in rail vehicles, the corresponding IN_VEHICLE activity appeared intermittently and with low confidence in the raw activity trace. This is presumably because such vehicles travel relatively smoothly, and exhibit less of the kinds of vibration patterns typical of road motor vehicles. The window summing approach helps uncover these lower confidence activities.

Due to the low confidence of IN_VEHICLE activity in vehicles that move relatively smoothly, the plain summing window approach has a tendency, in a transition between a more confidently received activity, e.g. WALKING and an adjacent less confident one, to shift the transition into the less confident activity up to almost the entire window length. Another approach using a split window was implemented, with a front window and a back window both extending $60s$ from the focus point. Transition windows can be found where the dominant activity in the back and front windows differ, and a better transition point selection within the transition window can be achieved by using the differences in confidence sums between these windows. An example is illustrated in Figure 14.

After brief evaluations of mass transit detection performance and inspection of traces, this final modification was not taken into use, as the resulting improvements did not appear worth the added complexity.

5.5 Mass transit detection

As part of the work, the following improvements to mass transit detection were implemented.

As discussed in the environment decription (Chapter 4), the maximum distance for matching between user and vehicle location trace points is set at 100 metres, and the maximum time difference at 60 seconds. With the sample rate of thirty seconds in the collected vehicle positions, a vehicle traveling at 80 km/h would produce samples every 667 metres, much in excess of the abovementioned distance limit. This caused false negatives with the user location samples falling in between the vehicle points in such a way that they are not matched to trace points from the vehicle the user travelled on, sufficiently for the 75% requirement, for vehicles moving faster than 2×100 m / 30 s / 0.75 = 32 km/h. Increasing the distance limit to several hundred metres would be computationally less efficient and more prone to false matches.

To correct the issue without increasing the distance limit, for each vehicle the portion of its position sequence contemporaneous to the user location sample is processed into linestrings, and user point distances calculated against those line geometries. This matching approach is illustrated in Figure 15.

To make the updated comparison more efficient, it was implemented directly in SQL using PostGIS extensions, replacing the prior Python implementation. To maintain similar performance in terms of execution speed, the number of user location samples used for matching a trip leg was restricted to a maximum of 40, increased from the prior 4 points.

The match scoring was also made less granular, such that each match within the permitted 100 metres' distance accumulates a score of $100 - d$ when the distance is d metres. The vehicle with the highest score wins.

The line type and name are set according to the matched vehicle. On some lines, a vehicle can intentionally change line name when passing a certain stop. Also, some vehicles in the transit data source would erroneously change line name intermittently to false values. In case of the vehicle having multiple line names in the matches, the most frequently occurring name is used.

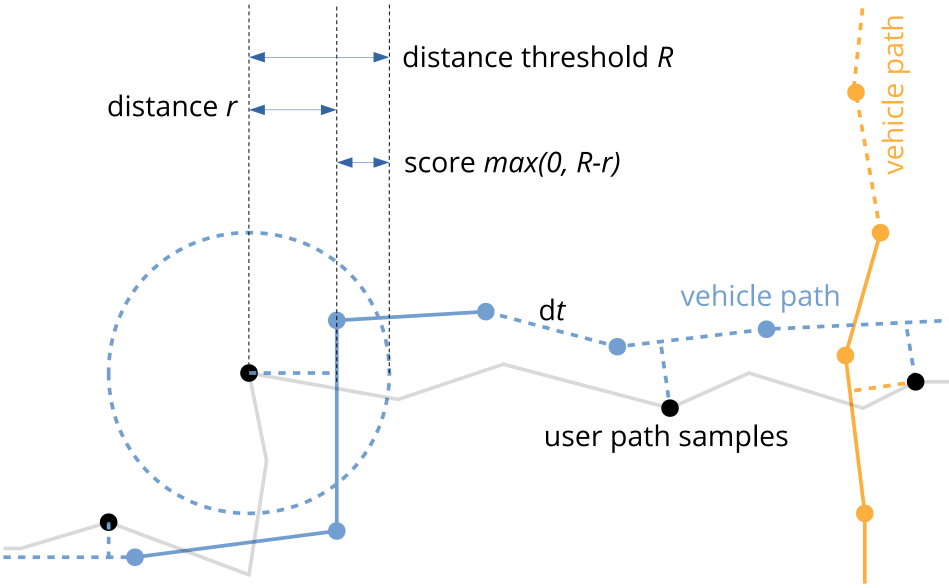


Figure 15: Mass transit matching using live vehicle positions. The user location trace point under inspection is shown circled, the linestrings formed for comparison with it for each vehicle from trace points contemporaneus with the user point are shown solid, with the non-contemporaneous portion of the paths as a dotted line. A match is found where the distance to the vehicle linestring is within the matching radius of the user point.

5.6 Regular destinations

As mentioned in the model and schema description in Section 5.1, in order to identify the regular destinations visited by a user, and the transfer points frequented on the way, the stop locations and trip leg endpoints are clustered.

In order to preserve the identity of existing clusters as new data is collected, the clustering is achieved by a simple online algorithm. A newly added stop or leg endpoint forms a one-point cluster. The two clusters closest to each other are merged, and the location of the merged cluster updated to the mean of the points collected therein, until there are no cluster pairs within a given distance limit parameter. This stage is depicted in Figure 16.

The distance limit selected for use in clustering is 200 metres, or twice the stop detection threshold, to avoid clustering consecutive stops along a linear path together.

A second level of clustering combines per-user stop and leg endpoint clusters in the same manner into shared place clusters. This stage is depicted in Figure 17. The shared places are labeled, combining two of primarily the “street” and falling back on “name” properties, taken from the first ten results of a reverse geocoding query.

While shared places could be clustered directly from all users’ stops and leg endpoints, two levels of clustering is used in order to avoid splitting user

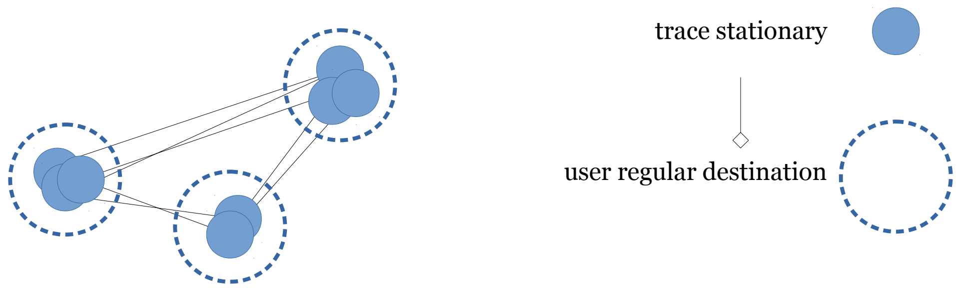


Figure 16: Clustering of stops and leg endpoints into leg_ends.

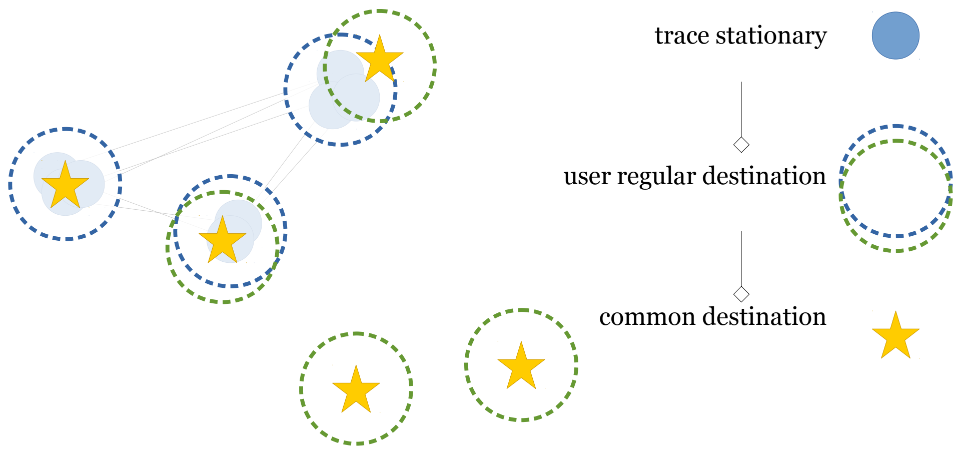


Figure 17: Clustering of user leg_ends into shared places.

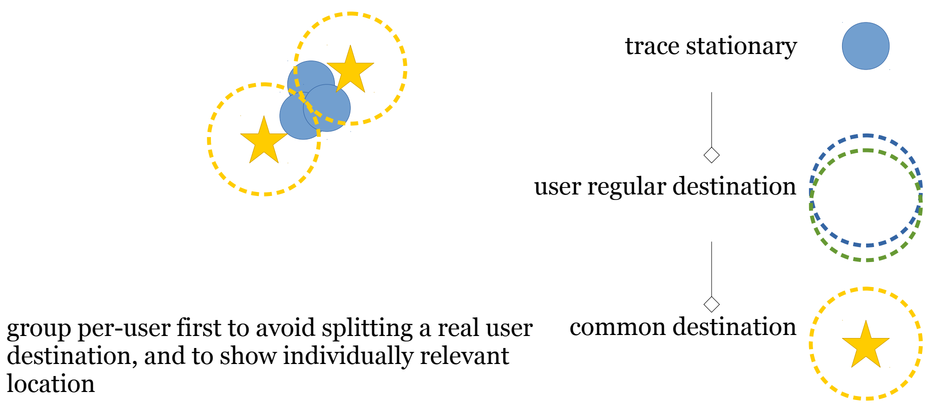


Figure 18: Two level clustering.

destinations. In a dense urban environment, and given enough users, places significant to some user occur roughly everywhere. In particular, a place significant to a user could occur between two single-level clusters. Without two-level clustering or another solution, stops at such locations could be split, finding affinity in different clusters depending on variation in location measurement. This would lead to irregularity also in the origin-destination assignment of routes. This issue is illustrated in Figure 18.

Also, in using the regular destinations, to for instance assist the user in navigation to the likely next destinations, displaying a location based on cross-user data could result in location information leak across users. Maintaining per-user locations makes the placement of such map markers more relevant to each user, and doesn't reveal information about nearby users inadvertently.

5.7 Trip visualization

Enabled by the abovementioned trip leg generation, stop and leg endpoint clustering and place labeling, a user trip log view was developed. The trip log features visualizations as shown in Figure 19, describing for each trip leg its start and end times and places, duration, distance travelled, and transport mode.

For path display to the client, presented as GeoJSON through the API, a heap based line simplification was implemented as described by e.g. Eberly [9, sec. 3, "Fast algorithm"], a local processing algorithm for redundant point weeding using the perpendicular distance criterion, as illustrated by Weibel [27, p. 117-118].

The trip log on the web provided by the siteserver, and the path display in the mobile app, also allow for user correction or confirmation of transport mode detection.

5.8 Route grouping

Grouping trips into regular routes taken by a user between given origin-destination pairs can be used to summarize the user's mobility patterns, and to compare route options using data from several trip samples. Potentially data across users relating to similar routes could be used in suggestions for faster or more energy-efficient route options.

For purposes of route grouping, trips are separated at stops longer than 30 minutes in duration, corresponding to the threshold presented for the trip

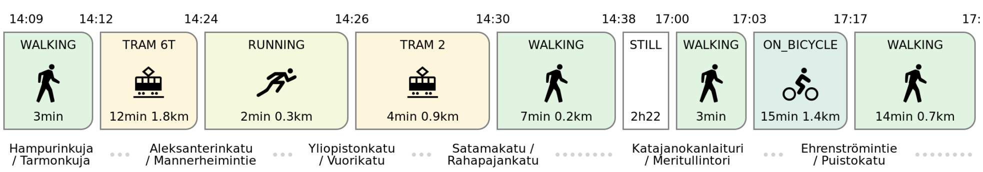


Figure 19: Portion of a recognized multimodal trip sequence.

chain delimiter in the definition of a trip chain described by McGuckin and Nakamoto [21].

In order to group similar routes from trips taken by a user between an origin-destination pair, a similarity criterion between trips or trip clusters is needed. A similarity criterion in turn requires a generalized trip representation practical for calculating such a comparison criterion.

A set of mode-location points, with a the location generalized by snapping onto fewer, coarser coordinates, annotated with the mode of transport at that location, is used for the generalized comparable trip respresentation. This allows the comparison of routes for purposes of clustering using fuzzy sets. Such a fuzzy set consisting a single trip contains simply the set of coarse mode-location points visited by that trip. When trips are grouped in a fuzzy set, the weight of each mode-location point in the set is the proportion of its occurrence within the trips to the total number of trips in the group.

The comparison of similarity of trip clusters represented as fuzzy sets is done by division of intersection size to union size, generalized such that the union sums over the maximum probability of each mode-location point, and the intersection over the minimum. The caveats of various generalizations of fuzzy set connectives are discussed in depth by Dubois and Prade [8].

One alternative explored for the coarse location were waypoints set at way crossings generated from OpenStreetMap data, snapping the sampled geolocation coordinates to the nearest way within reasonable distance, and then to the nearest crossing on that way. Another alternative is rounding coordinates to a grid of appropriate size.

One downside of using a grid approach is that longer trips end up with very large point sets, making the comparison operations slower. Waypoints at crossings have lower density along higer-speed, more exclusive roadways, reducing the number of points along those parts of the path. Also, waypoints at crossings to a fair extent model the routing decisions made along the path. A grid approach weights the importance of any distance along the route the same way, whereas the choices at crossings should be more relevant to the travel time outcome.

In practice, route grouping using crossings was found less effective in cases where the route choice was between a more dense city route and a sparse highway route — the low number of points generated in the highway route segment makes it carry relatively little weight in route comparisons relative to other differences that may occur in denser segments of the trips. One possibility to counteract this would be to grant a higher weight to sparse crossings as compared to dense crossings.

Another concern arises in selecting the coarse location properties due to the location sampling interval. If the grid or waypoint locations are denser than the sampled device location, some coarse points in between will be left out of the trip properties. Unless the skipped-over grid cells or waypoints are filled in by lines, with different time alignment on the path on different trips leaving out different coarse locations, there appears a false difference

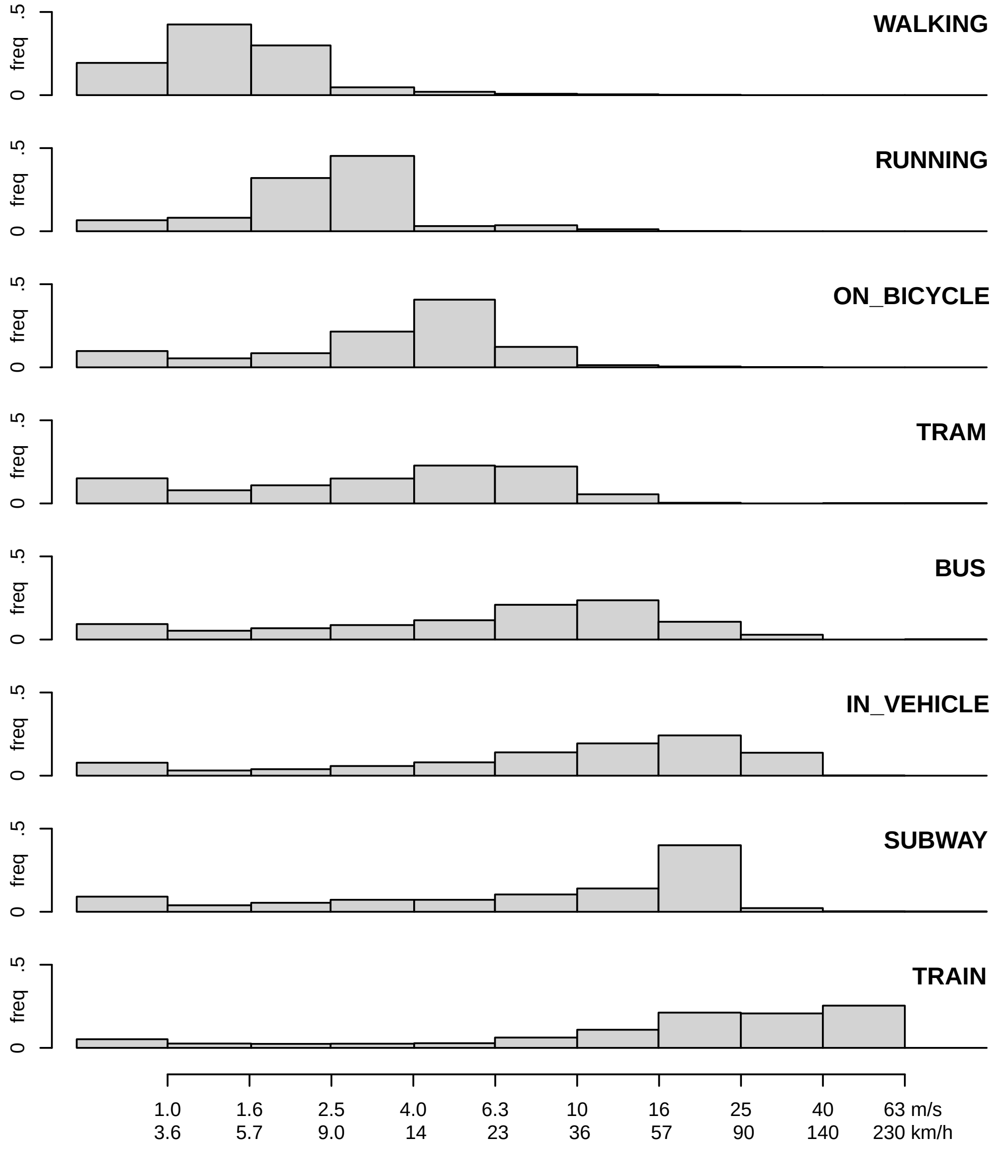


Figure 20: Histogram of movement speed by transport mode (detected and partially user corrected, Feb 8th – Mar 8th 2017, 82 distinct users). IN_VEHICLE excludes other vehicular modes shown.

between paths that are identical in reality. This sets a minimum for the grid or waypoint location granularity d_{coarse} based on expected maximum speed v_{max} and sampling interval t_{sample}:

$$d_{coarse} \geq v_{max} t_{sample}$$

In the data gathered, the speed rarely (1.3%) exceeds 30 m/s (108 km/h), so with the $10s$ position sampling interval used by the client, the resulting granularity limit is

$$d_{coarse} \geq 30m/s \cdot 10s = 300m$$

The chosen granularity of 0.002° latitude, 0.004° longitude, or approximately 222×222 metres square at 60°latitude, yields a speed limit of

$$v_{max} = \frac{222m}{10s} = 22.2m/s \approx 80km/h$$

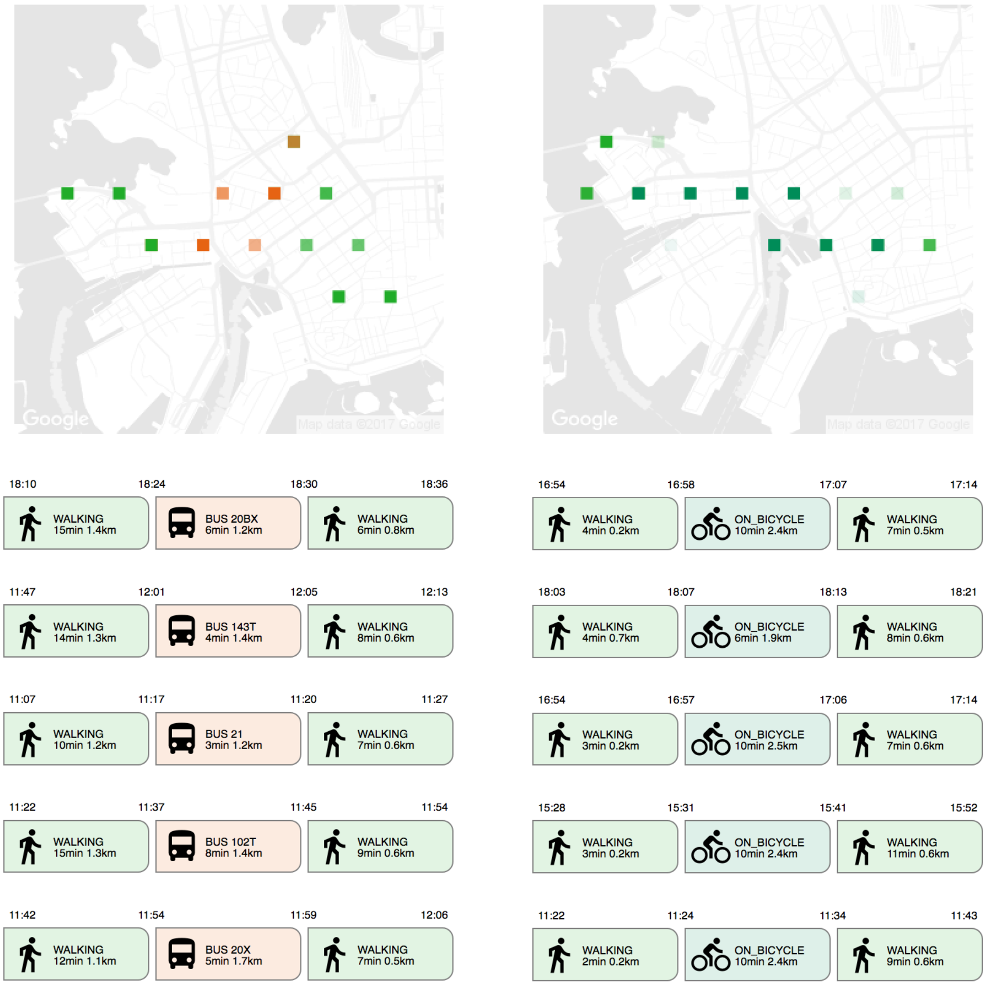
which is exceeded in 8.5% of the data point pairs. For reference, histograms of speed by transport mode in a sample of the collected data are shown in Figure 20.

The density of crossings typically relates inversely to the speed limit of a road, so this issue could be mitigated by the waypoint crossings option.

The longitudinal grid size is adjusted by the cosine of the snapped latitude to maintain an approximately square cell aspect. The drawback is that adjacent cell rows are not aligned, but for route comparisons that is not a requirement.

Trip grouping is done in similar manner to leg end and shared place grouping, using pairwise clustering, with a distance threshold applying in this case to the union/intersection fuzzy sums similarity metric. Each trip is initialized as a route point set, each point being coarse coordinates and transport mode at that point. The nearest routes by similarity are then merged until a threshold minimum similarity distance is reached.

To illustrate the clustering results, Figure 21 shows an example selection of trips with the same origin and destination, from two distinct groups.



(a) Route group with bus use. (b) Route group with bicycle use.

Figure 21: Route comparison example for two main route groups arising from an origin-destination pair. The visualization of the group's mode-location fuzzy set overlaid on the map is shown at top, followed by a selection of trip instances from the group.

6 Evaluation

This chapter presents evaluations performed on the route model features and improvements made within the scope of this work.

Analytical evaluations discuss the fit into the technical context — the resulting model of stops, trips, regular destinations, routes and their modes, generated from a user's location and activity trace, is assessed to evaluate the suitability of the model. Under dynamic qualities, the performance of accessing processed data used for display to the user by the client software, is evaluated, as well as the efficiency of storage.

Public study and surveys

During a wider campaign, the system was used in a field experiment with a total of 93 users in a period of January 10th through March 17th 2017. 981,291 data points were collected in this time period, forming 13,999 active trip legs, consisting of 1,208 bus, 201 tram, 113 train, 4,087 car and 1,020 bicycle trips plus walking and running. Users also submitted 546 corrections to trip legs.

Two questionnaire links were presented to users through the mobile app messaging during the campaign. The first survey concerned mostly mobility habits and demographic questions, while the second survey was focused more on the application and feature usage and preferences. 57 users responded to the first, and 49 to the second survey.

The favourite features identified by the users in the second survey were the display of previous trips on the map facilitated by the route model and path simplification, the personal energy certificate visualizing the user's mobility energy efficiency, and the client-side convenience feature incorporating Google's real-time traffic information map overlay in the app.

The most interesting features for development, picked from options given by the team, were notifications of exceptional traffic jams on predicted route, automatic re-planning in case of a public transport problem, and better focus of disruption notifications.

In a freeform question “What’s the most important thing for TrafficSense to address in the future?” the most common suggestions were related to quality of transport mode and mass transit detection mentioned in 9 out of 29 entries, followed by usability and interface related comments with 5 entries, and dynamic rerouting based on disruptions in 3 entries.

Mass transit detection enhancements

The improved implementation of live vehicle location matching was evaluated against the prior implementation. In the older implementation, four user trace sample points were used, and matched against vehicle location points in the surrounding time window. With the vehicle location sampling interval of 30 seconds, and the 100 metre distance criterion, this would be expected to cause otherwise optimal user trace point samples to potentially fall outside the matching radius once the vehicle speed exceeds $2 \times 100\text{m}/30\text{s} \approx 6.67\text{m/s} \approx 24\text{km/h}$.

For trams, in Helsinki having an average speed of 14.7 km/h in 2013–2014 [14], with dense stops, the new and old methods should produce similar results. More differences can be expected on the subway, and on bus lines with highway portions, where speeds are higher and stops more sparse.

A smaller-scale one-day targeted data gathering and evaluation was performed by the research group, with a focus on the subway train travel mode that presented challenges due in part to location fix sparseness in underground conditions. The eight participants — seven using public transportation and one reference participant driving a private car — recorded trips using the TrafficSense software during a seven-hour window, also manually journaling a total of 103 vehicle trips. Within that experiment, the live mass transit matching enhancements described in Section 5.5 resulted in an increase in mass transit trip legs successfully matched using live vehicle location data from 20 to 29, with eight of the added matches being subway legs, and one a bus. Overall, of the 97 journaled public transport trips, 47, or 48%, resulted in detection of the correct public transport mode and line. This dataset and experimental results are described in detail in Rinne et al. [23].

Evaluation of detected trips against journal

A 31-day window of data collected from the author, and processed using the auxiliary mass transit data — live vehicle locations and static timetables — from the same period, was used to evaluate the effectiveness of the data collection and the correctness of its interpretation in the trip model. A corresponding trip journal, collected using a web form at each stop and travel mode transition, was used as the reference. Evaluation was done with respect to correctness of detected mode and mass transit, missing or extraneous legs, and accuracy of transfer times.

This data set consists of 547 manually recorded transitions between stops and trip legs, and 347 automatically detected stops and trip legs without user

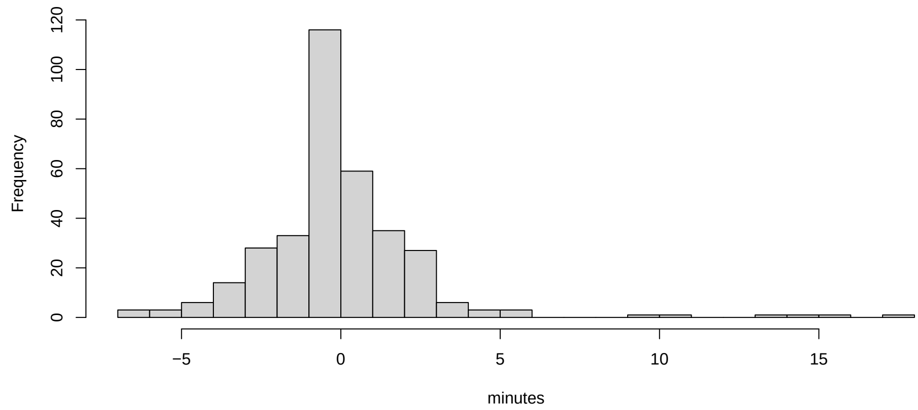


Figure 22: Lateness of detected versus manually recorded transitions.

	NONE	BICYCLE	BUS	RUN	STILL	TRAM	VEHICLE	WALK	ANY	ALL
NONE	-	-	-	1	1	-	-	3	5	5
BICYCLE	-	**4**	-	-	-	-	-	-	4	4
BUS	28	-	**43**	-	-	-	11	1	55	83
STILL	83	-	1	-	**107**	1	3	1	113	196
TRAM	23	-	-	-	-	**13**	7	1	21	44
WALK	71	-	-	1	1	-	-	**147**	149	220
ANY	205	4	44	1	108	14	21	150	342	547
ALL	205	4	44	2	109	14	21	153	347	552

Table 1: Confusion matrix for mode detection in a 31-day sample. Manually journaled true modes on rows, detected modes in columns.

	sensitivity incl. missing	sensitivity excl. missing	specificity excl. missing
BICYCLE	$4/4 \approx 1.000$	$4/4 \approx 1.000$	$338/338 \approx 1.000$
BUS	$43/83 \approx 0.518$	$43/55 \approx 0.782$	$286/287 \approx 0.997$
STILL	$107/196 \approx 0.546$	$107/113 \approx 0.947$	$228/229 \approx 0.996$
TRAM	$13/44 \approx 0.295$	$13/21 \approx 0.619$	$320/321 \approx 0.997$
WALK	$147/220 \approx 0.668$	$147/149 \approx 0.987$	$190/193 \approx 0.984$
OVERALL	$314/547 \approx 0.574$	$314/342 \approx 0.918$	$1362/1368 \approx 0.996$

Table 2: Sensitivity and specificity per mode in the 31-day sample.

corrections. The manually entered transitions have an automatically recorded timestamp, and a user-supplied estimate of form entry lateness in minutes. As such, the expected accuracy of the time differences is limited to one minute at best.

An excerpt of the data is presented in Table 3. Notably, there are no trip legs taken by car in this 31-day sample, so false positive mass transit matches for such vehicle legs cannot occur.

Of the 547 manually recorded transitions, 205 were not present in automatic detection. However, while the automatic detection was tuned to not find stops of less than five minutes in duration, the manual bookkeeping had no such limits. Restricting to the 251 manually recorded activity spans longer than five minutes, 65, or 26%, were not present in automatic detection.

Of 342 corresponding transitions, 175, or 51%, were within one minute of each other in manual bookkeping and automatic detection. 244, or 71% were within two minutes, and 298, or 87%, within three minutes. The full distribution is shown in Figure 22. For context, 273, or 80%, of these transitions were manually logged within one minute.

The stop and activity detection within the 342 time-correspondent legs was correct in 328 cases and incorrect in 14 cases. The correct mode including mass transit type was found in 310 cases.

The confusion matrix of transit mode detection is shown in Table 1. The sensitivity and specificity of detection per transit mode is shown in Table 2.

Recognition of tram trip legs was less reliable than buses, partly due to being less reliably detected as IN_VEHICLE in activity recognition. This could be improved by using speed determined from location data to augment the interpretation of the activity data. When using such augmentation, in particular STILL activity while moving would strongly imply being in a smoothly moving vehicle.

Route grouping

In inspection of regular routes grouping results, it is found that for frequently travelled origin-destination pairs, the most common routes are clearly identified according to path taken and major transport modes. Taking the author's most travelled origin-destination pair as an example, with a total of 136 trips recorded, 30 trips match the most common route (via Kamppi to Otaniemi, through northern Lauttasaari), and 15 the second most common variation (via Kamppi to Otaniemi, through central Lauttasaari).

Irregularity in the trace recording, depending on the phone's ability to get location fixes, results in a long tail of smaller and individual route groups. For the same example origin-destination pair, 55, or 40% of the trips are uniquely intermittent in the clustering. In other words, with the devices and location capture configurations used, getting a complete record of mobility could not be achieved, but finding the most common patterns is possible.

In longer routes, trips with mode variation in a shorter trip leg may get

Table 3: Manual trip log excerpt aligned with detected trip legs. The columns contain, respectively:

- time — at journal stop or transit mode start transition,
- mode/line — stop or transit mode and line name,
- place — location where transition occurred,
- late — user-estimated lateness of form submission in minutes,
- time — at detected stop or transit mode start transition,
- km — distance travelled along location trace in kilometres,
- activity — detected activity, or stop found from location trace,
- live — mass transit line detected using live vehicle locations,
- timetable — mass transit line detected using static timetable,
- place — reverse geocoded label for place cluster at transition.

The raw journaled place names have been edited for consistency, and some reverse geocoded place labels masked.

manually recorded transitions				leg start transitions detected from trace					
time	mode/line	place	late	time	km	activity	live	timetable	place
				Sunday					
15:45:44	Walk	Home	2	15:42:02	1.5	WALKING			[Home / stop A]
16:06:42	Stop	Pool	2	16:04:02		STILL			Kauppatori / Market Square
19:40:04	Bicycle	Pool	0	19:41:07	2.2	ON_BICYCLE			Kauppatori / Market Square
19:52:34	Stop	Café	0	19:53:31		STILL			Ehrenströmintie / Itäinen Puistotie
15:58:55	Bicycle	Café	0	19:59:03	1.5	ON_BICYCLE			Ehrenströmintie / Itäinen Puistotie
20:09:34	Walk	Home stop A	1	20:05:56	0.2	WALKING			[Home / stop A]
20:09:55	Stop	Home	1	20:10:25		STILL			[Home / stop A]
				Monday					
12:00:24	Walk	Home	0	12:00:06	0.4	WALKING			[Home / stop A]
12:05:33	Bus 18	Home stop B	0	12:05:03	0.7	IN_VEHICLE			[Home stop B]
12:08:18	Walk	Hub stop	0	12:08:38	0.1	WALKING			Urho Kekkosen katu / Kamppi
12:12:15	Stop	Hub	0						
12:16:47	Bus 103T	Hub	0	12:29:53	4.2	IN_VEHICLE		BUS 103T	Luoteisväylä / Katajaharjuntie
12:37:26	Walk	Office stop	0	12:37:20	0.6	WALKING			Metallimiehenkuja / Otaniementie
12:45:10	Stop	Office	0	12:46:18		STILL			Konemiehentie / Tietotie
21:43:35	Walk	Office	0	21:41:03	0.7	WALKING			Konemiehentie / Tietotie
21:54:44	Stop	Office stop	2						
22:04:05	Bus 102T	Office stop	2	22:05:01	7.2	IN_VEHICLE		BUS 102T	Karhusaarentie / Keilaniementie
22:20:28	Stop	Hub	0	22:20:03		STILL			Urho Kekkosen katu / Kamppi
22:25:38	Bus 18	Hub	0	22:25:30	1.2	IN_VEHICLE	BUS 18		Urho Kekkosen katu / Kamppi
22:29:47	Walk	Home stop A	0	22:29:38	0.1	WALKING			[Home / stop A]
22:32:40	Stop	Home	0	22:31:33		STILL			[Home / stop A]
				Tuesday					
11:10:35	Walk	Home	0	11:16:06	0.8	WALKING			[Home stop B]
11:26:36	Stop	Hub	0						
11:35:23	Bus 102	Hub	0	11:38:15	7.8	IN_VEHICLE			Porkkalankatu / Itämerenkatu
11:52:51	Walk	Office stop	2	11:51:01	0.6	WALKING			Metallimiehenkuja / Otaniementie
11:59:41	Stop	Office	0	11:57:42		STILL			Konemiehentie / Tietotie
15:30:56	Walk	Office	0	15:31:36	0.6	WALKING			Konemiehentie / Tietotie
15:39:53	Bus 102	Office stop	1	15:38:47	8.2	IN_VEHICLE		BUS 102	Metallimiehenkuja / Otaniementie
15:55:02	Walk	Transfer A	0						
15:55:44	Stop	Transfer B	0	15:55:07		STILL			Hietalahdenkatu / Ruoholahdenkatu
16:01:58	Tram 9	Transfer B	0	16:01:10	1.1	IN_VEHICLE			Hietalahdenkatu / Ruoholahdenkatu
16:08:07	Walk	Gym stop	0	16:07:33	0.2	WALKING			Tyynenmerenkatu / Hietasaarenkuja
16:11:03	Stop	Gym	0	16:10:30		STILL			Tyynenmerenkatu / Hietasaarenkuja
19:10:44	Walk	Gym	0	19:08:15	0.3	WALKING			Tyynenmerenkatu / Hietasaarenkuja
19:13:39	Stop	Gym stop	0						
19:14:39	Tram 9	Gym stop	0	19:13:19	1.7		TRAM 9	TRAM 9	Tyynenmerenkatu / Hietasaarenkuja
19:20:48	Walk	Hub stop	0	19:20:02	0.4	WALKING			Urho Kekkosen katu / Kamppi
19:24:02	Stop	Hub shop	0						
19:42:59	Walk	Hub shop	0						
19:46:01	Stop	Hub	0	19:46:35		STILL			Urho Kekkosen katu / Kamppi
19:58:50	Bus 18	Hub	0	19:55:20	1.3	IN_VEHICLE			Urho Kekkosen katu / Kamppi
20:08:42	Walk	Home stop A	0	20:08:16	0.2	WALKING			[Home / stop A]
20:10:25	Stop	Home	0	20:11:29		STILL			[Home / stop A]

Table 3: Manual trip log excerpt with detected trip legs (continued)

manually recorded transitions				leg start transitions detected from trace					
time	mode/line	place	late	time	km	activity	live	timetable	place
				Wednesday					
08:55:57	Walk	Home	0	08:55:31	1.2	WALKING			[Home / stop A]
08:58:19	Stop	Home stop A	1	09:14:42	6.0	IN_VEHICLE			Hietalahdenkatu / Ruoholahdenkatu
09:01:14	Bus 14	Home stop A	0						
09:05:24	Walk	Hub stop	0						
09:09:33	Stop	Hub	0						
09:12:20	Bus 102	Hub	0						
09:29:00	Walk	Office stop	2	09:23:21	1.3	WALKING			Miestentie / Karhusaarentie
09:34:16	Stop	Office	0	09:36:55		STILL			Konemiehentie / Tietotie
21:32:30	Walk	Office	0	21:33:28	0.6	WALKING			Konemiehentie / Tietotie
21:40:38	Stop	Office stop	0						
21:44:45	Bus 103T	Office stop	1	21:43:51	8.3	IN_VEHICLE		BUS 103T	Metallimiehenkuja / Otaniementie
22:01:01	Walk	Hub	0	22:00:23	0.3	RUNNING			Malminkatu / Lapinlahdenkatu
22:02:57	Stop	Hub	0						
22:03:45	Tram 2	Hub	0	22:03:45	3.4	TRAM 2		TRAM 2	Urho Kekkosen katu / Kamppi
22:22:36	Walk	Home stop A	0	22:22:09	0.4	WALKING			[Home / stop A]
22:24:57	Stop	Home	0	22:27:03		STILL			[Home / stop A]
				Thursday					
12:29:34	Walk	Home	0	12:39:19	0.4	WALKING			[Home stop C/D]
12:31:56	Stop	Home stop A	0						
12:35:46	Bus 18	Home stop A	0						
12:40:26	Walk	Hub stop	0						
12:44:33	Stop	Hub	0						
12:47:40	Bus 102	Hub	0	12:52:05	5.8	IN_VEHICLE			Salmisaarenkatu / Sulhasenkuja
13:01:46	Walk	Office stop	0	12:58:43	1.3	WALKING			Miestentie / Karhusaarentie
13:10:29	Stop	Office	0	13:10:52		STILL			Konemiehentie / Tietotie
15:33:48	Walk	Office	0						
15:41:32	Stop	Office stop	0						
15:44:23	Bus 103	Office stop	2	15:45:16	7.2	IN_VEHICLE		BUS 103	Miestentie / Karhusaarentie
15:55:47	Walk	Transfer A	0						
15:57:25	Stop	Transfer B	0	15:56:16		STILL			Hietalahdenkatu / Ruoholahdenkatu
16:01:19	Tram 9	Transfer B	0	16:01:16	1.4	IN_VEHICLE			Hietalahdenkatu / Ruoholahdenkatu
16:06:10	Walk	Gym stop	0	16:05:49	0.2	WALKING			Tyynenmerenkatu / Hietasaarenkuja
16:09:14	Stop	Gym	0	16:11:34		STILL			Tyynenmerenkatu / Hietasaarenkuja
18:51:10	Walk	Gym	1	18:50:01	0.2	WALKING			Tyynenmerenkatu / Hietasaarenkuja
18:52:38	Stop	Gym stop	0						
18:55:17	Tram 9	Gym stop	0	18:53:00	1.9	IN_VEHICLE			Tyynenmerenkatu / Hietasaarenkuja
19:03:18	Walk	Hub stop	0						
19:06:27	Stop	Hub shop	0						
19:19:00	Walk	Hub shop	1						
19:21:33	Stop	Hub	0						
19:23:45	Bus 14	Hub	0	19:19:15	1.5	IN_VEHICLE		BUS 14	Urho Kekkosen katu / Kamppi
19:29:47	Walk	Home stop A	0	19:29:43	0.1	WALKING			[Home / stop A]
19:31:51	Stop	Home	0	19:31:18		STILL			[Home / stop A]
				Friday					
11:27:59	Walk	Home	0						
11:30:46	Stop	Home stop A	1	11:29:06	0.4	IN_VEHICLE			[Home / stop A]
11:33:32	Tram 3	Home stop A	0						
11:36:45	Walk	Home stop D	0	11:34:47	0.3	WALKING			[Home stop C/D]
11:39:04	Bus 18	Home stop C	0	11:38:37	0.3	IN_VEHICLE	BUS 18		[Home stop C/D]
11:40:44	Walk	Hub stop	0	11:40:44	0.2	WALKING			Urho Kekkosen katu / Kamppi
11:45:50	Bus 102	Hub	0	11:48:29	6.8	IN_VEHICLE		BUS 102	Porkkalankatu / Itämerentori
11:58:23	Walk	Office stop	0	11:58:43	0.6	WALKING			Metallimiehenkuja / Otaniementie
12:05:37	Stop	Office	0	12:08:27		STILL			Konemiehentie / Tietotie
21:50:08	Walk	Office	1	21:46:39	0.7	WALKING			Konemiehentie / Tietotie
21:56:40	Stop	Office stop	0	21:57:29		STILL			Metallimiehenkuja / Otaniementie
22:04:11	Bus 102T	Office stop	2	22:02:23	8.4	IN_VEHICLE		BUS 102T	Metallimiehenkuja / Otaniementie
22:19:51	Walk	Hub	0	22:21:25		STILL			Urho Kekkosen katu / Kamppi
22:22:05	Stop	Hub	0	22:24:42	0.3	WALKING			Urho Kekkosen katu / Kamppi
22:27:02	Bus 18	Hub	1	22:27:20	1.1	IN_VEHICLE	BUS 18		Malminkatu / Lapinlahdenkatu
22:32:04	Walk	Home stop A	1	22:31:30	0.1	WALKING			[Home / stop A]
22:33:37	Stop	Home	0	22:35:40		STILL			[Home / stop A]
				Saturday					
11:04:24	Walk	Home	0	11:04:27	0.5	WALKING			[Home / stop A]
11:11:59	Stop	Home stop D	0	11:10:38		STILL			[Home stop C/D]
11:15:36	Tram 6	Home stop D	0	11:15:45	0.8	IN_VEHICLE		TRAM 6	[Home stop C/D]
11:19:28	Walk	kalevankat	0	11:18:31	0.3	WALKING			Abrahaminkatu / Lönnrotinkatu
11:21:11	Stop	Gym stop	0						
11:22:15	Tram 9	Gym stop	0	11:20:47	1.0	IN_VEHICLE			Ruoholahdenranta / Eerikinkatu
11:25:32	Walk	Gym stop	0	11:25:21	0.3	WALKING			Tyynenmerenkatu / Hietasaarenkuja
11:28:34	Stop	Gym	0	11:32:15		STILL			Tyynenmerenkatu / Hietasaarenkuja
13:38:00	Walk	Gym	0						
13:41:36	Stop	Gym-shop	1						
14:00:08	Tram 6T	Gym stop	0	14:00:30	1.6	IN_VEHICLE		TRAM 6T	Tyynenmerenkatu / Hietasaarenkuja
14:09:24	Walk	Home stop D	0	14:08:25	1.6	WALKING			[Home stop C/D]
14:24:55	Stop	Pool	0	14:27:51		STILL			Kauppatori / Market Square
				16:01:58	0.2	WALKING			Kauppatori / Market Square
16:06:15	Bicycle	Pool	0	16:06:23	3.2	ON_BICYCLE			Kauppatori / Market Square
16:21:37	Walk	Home stop E	0	16:20:57	0.4	WALKING			[Home / stop A]
16:27:45	Stop	Home	0	16:29:26		STILL			[Home / stop A]

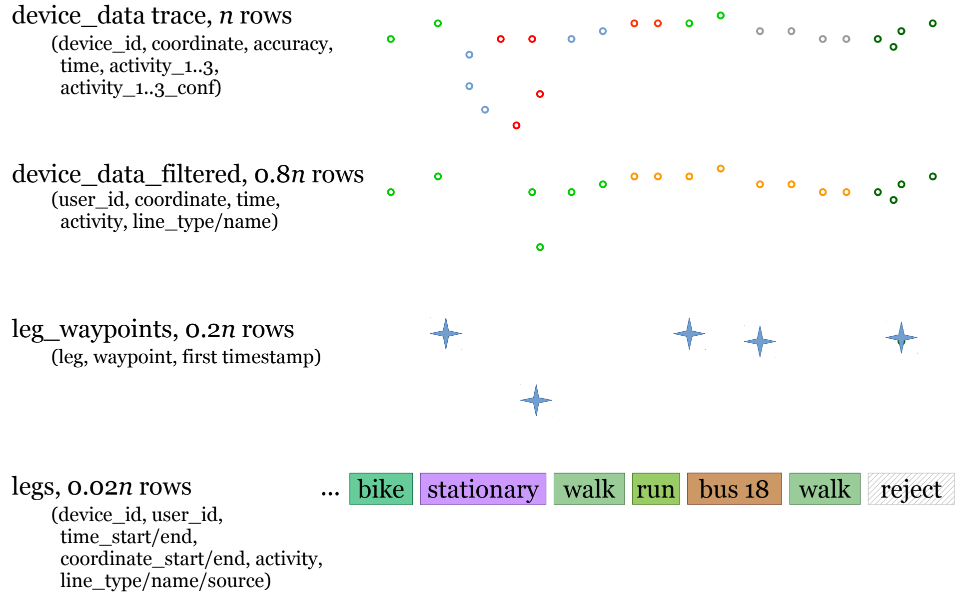


Figure 23: Relative compactness of updated model.

grouped together, depending on the fuzzy set similarity clustering threshold. For example, as might be expected, with a similarity parameter of 0.75, in a ten-kilometre route, variation between a one-kilometre walk or bus leg is not distinguished in the grouping. Higher similarity requirement results in more fragmentary grouping. The location-mode point set comparison effectively weights by distance, but comparing the leg structure directly using transfer location clusters could produce more relevant results in the case of mass transit trips. On the other hand, such an approach would not generalize well to e.g. car and bike trips that, unlike mass transit lines, can take variable paths.

Access and storage performance

The trip leg data in the updated model is significantly more compact than the raw or filtered trace, as illustrated in Figure 23, requiring 50 times fewer rows of a similar width. Also the road crossings waypoint trace is about five times more compact, and can be more practical for applications working with a road network representation.

The updated mass transit detection was tuned, using the number of matched points parameter, to perform in the same time as the prior implementation.

7 Discussion and conclusions

Automatically detecting individuals' door-to-door multimodal trips has important applications in an intelligent transport system. The main objective of this thesis project was to develop representations for multimodal routes, suitable, in the context of TrafficSense and similar systems, for analysis of mobility patterns.

Previous experiments on mobile traffic crowdsensing and personalized routing, such as Letchner et al. [19] have focused primarily on private cars and route choices of individuals; the novelty of TrafficSense is in its focus on energy efficiency of the whole traffic system. The software system with the additions presented brings together use of activity recognition for transport mode detection, previous approaches of mass transit use detection from location traces explored in Ekholm [10], and trip chain partitioning, to cover multimodal, door-to-door routes.

To that end, features were developed such as discriminating of stops and trip legs from a location trace, identifying trip origin and destination and the mode of transport in trip legs, and recognizing the user's regular destinations and routes — along with a concrete model for storing these route features. Improvements were also made to the existing system's location trace and activity recognition postprocessing as well as mass transit line detection.

The system was found functional for discovering and describing regular routes and destinations, as well as providing trip logs, energy efficiency analysis, and other user-facing features. While the most frequently occurring destinations and routes become apparent, challenges with location trace data continuity, and reliability of mass transit matching, leave the continuous picture of user mobility less complete.

The related infrastructure makes progress however — more recently, the expansion of live location data availability to more of the public transport vehicle fleet in the greater Helsinki region makes more lines available for matching in this manner. Also, anecdotal evidence during the project suggests that data quality from newer mobiles can surpass that of older ones when it comes to trace reliability. These developments suggest that some improvement is possible already with the current approaches.

Other options for mass transit detection could be looked into, besides matching location traces. For instance, installing Bluetooth beacons onto public transport stops and vehicles [2] could allow mobile devices to use them to detect public transport trips.

When matching mass transit using statit timetables, only plans containing a single vehicular leg are considered for matching in the present implementation. Quick transfers between vehicles may not be detected as separate trips by the activity detection and filtering determination, so allowing plans with transfers, and splitting the leg accordingly, could produce improved detection.

The regular destination recognition is effective in finding common places where the user spends some time. For places visited for short durations, the distinction between true intentional activity locations (e.g. a drop-off), and transfer stopovers merely necessary for transit, is not so easily deduced from the data. More informative analysis than a plain half-hour time threshold may be available.

The accuracy of route recognition is suitable for finding regular mobility patterns at the origin-destination level. Within origin-destination groups, identifying regular routes based on reoccurrences of the route taken and the transit modes used is also possible. However, due to the intermittency common in the collected data, many individual trip instances to fall outside such groups. A comparison method that generalizes better over intermittent data could be sought.

Depending on use case, instances with more missing data could also be discarded, if only complete traces are of interest.

In general, activity recognition is a low power feature and as such shouldn't depend on the higher power demand of fine location tracing. In this application, however, as location data is required anyway, it could be used to inform the activity recognition process, to better identify moving vehicles that exhibit less motor and road vibrations.

Mobility data gathering and analysis remains a developing area, with wide interest from transport related agencies and businesses, and municipalities. The results can be applied for modeling and prediction in the traffic system. Opportunities for optimization are enabled as well, either through design or resourcing, or by influencing individual route choices by providing relevant and current information. Also individuals interested in their own mobility habits can make use of mobility analysis applications, whether for the sake of curiosity, fitness, time tracking, or optimization.

References

[1] Directive 2010/40/EU of the European Parliament and of the Council of 7 July 2010 on the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport. URL http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2010:207:0001:0013:EN:PDF.

[2] Toiminta- ja taloussuunnitelma 2017–2019 HSLH 25.10.2016, October 2016. URL https://www.hsl.fi/sites/default/files/hsl_-_toiminta-_ja_taloussuunnitelma_2017-2019.pdf.

[3] Aalto Energy Efficiency Research Programme (AEF) website. TrafficSense – Energy Efficient Traffic with Crowdsensing (2013–2017), August 2015. URL http://aef.aalto.fi/en/research/trafficsense/.

[4] R. M. Alkan, H. Karaman, and M. Sahin. GPS, GALILEO and GLONASS satellite navigation systems & GPS modernization. In *Proceedings of 2nd International Conference on Recent Advances in Space Technologies, 2005. RAST 2005.*, pages 390–394, June 2005. doi:10.1109/RAST.2005.1512598.

[5] Android Developers website. Changing Location Settings, September 2016. URL https://developer.android.com/training/location/change-location-settings.html. Accessed 2016-09-07.

[6] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga. Activity Recognition Using Inertial Sensing for Healthcare, Wellbeing and Sports Applications: A Survey. In *23th International Conference on Architecture of Computing Systems 2010*, pages 1–10, February 2010. URL http://ieeexplore.ieee.org/abstract/document/5759000/.

[7] Ling Chen, Mingqi Lv, Qian Ye, Gencai Chen, and John Woodward. A personal route prediction system based on trajectory data mining. *Information Sciences*, 181(7):1264–1284, April 2011. ISSN 0020-0255. doi:10.1016/j.ins.2010.11.035. URL http://www.sciencedirect.com/science/article/pii/S0020025510005888.

[8] D. Dubois and H. Prade. A review of fuzzy set aggregation connectives. *Information Sciences*, 36(1):85–121, July 1985. ISSN 0020-0255. doi:10.1016/0020-0255(85)90027-1. URL http://www.sciencedirect.com/science/article/pii/0020025585900271.

[9] David Eberly. Polyline Reduction, 2001. URL https://www.geometrictools.com/Documentation/PolylineReduction.pdf.

[10] Arttu Ekholm. *Liikennevälineen automaattinen tunnistaminen.* PhD thesis, 2015. URL http://tampub.uta.fi/handle/10024/97595.

[11] Marta C. González, César A. Hidalgo, and Albert-László Barabási. Understanding individual human mobility patterns. *Nature*, 453(7196): 779–782, June 2008. ISSN 0028-0836. doi:10.1038/nature06958. URL http://www.nature.com/nature/journal/v453/n7196/abs/nature06958.html.

[12] Google APIs for Android website. DetectedActivity, August 2016. URL https://developers.google.com/android/reference/com/google/android/gms/location/DetectedActivity. Accessed 2016-09-06.

[13] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design Science Research in Information Systems. *MISQ*, 28(1):75–105, 2004.

[14] HSL Helsingin seudun liikenne. Raitioliikenteen linjastosuunnitelma, 2015. URL https://www.hsl.fi/sites/default/files/uploads/13_2015_raitioliikenteen_linjastosuunnitelma_netti.pdf.

[15] HSL Helsingin seudun liikenne. Joukkoliikenteen suunnitteluohje HSL-liikenteessä, 2016. URL https://www.hsl.fi/sites/default/files/uploads/joukkoliikenteen_suunnitteluohje_hsl-liikenteessa_2016.pdf.

[16] Juuso Ilomäki. Recognizing Transportation Modes by Multivariate Clustering of Accelerometer Data. June 2016. URL https://aaltodoc.aalto.fi/handle/123456789/21068.

[17] V. Kostakos, T. Camacho, and C. Mantero. Wireless detection of end-to-end passenger trips on public transport buses. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 1795–1800, September 2010. doi:10.1109/ITSC.2010.5625062.

[18] O. D. Lara and M. A. Labrador. A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Communications Surveys Tutorials*, 15(3): 1192–1209, 2013. ISSN 1553-877X. doi:10.1109/SURV.2012.110112.00192.

[19] Julia Letchner, John Krumm, and Eric Horvitz. Trip Router with Individualized Preferences (TRIP): Incorporating Personalization into Route Planning. In *Proceedings of the 18th Conference on Innovative Applications of Artificial Intelligence - Volume 2*, IAAI'06, pages 1795–1800, Boston, Massachusetts, 2006. AAAI Press. ISBN 978-1-57735-281-5. URL http://dl.acm.org/citation.cfm?id=1597122.1597139.

[20] Lin Liao, Dieter Fox, and Henry Kautz. Location-based activity recognition. In *Advances in Neural Information Processing Systems*, pages 787–794, 2006. URL https://papers.nips.cc/paper/2911-location-based-activity-recognition.pdf.

[21] Nancy McGuckin and Yukiko Nakamoto. Trips, Chains and tours-using an operational definition. In *National Household Travel Survey Conference*,

2004. URL http://onlinepubs.trb.org/onlinepubs/archive/conferences/nhts/McGuckin.pdf.

[22] M. H. Park, H. C. Kim, S. J. Lee, and K. S. Bae. Performance evaluation of Android location service at the urban canyon. In *16th International Conference on Advanced Communication Technology*, pages 662–665, February 2014. doi:10.1109/ICACT.2014.6779045.

[23] Mikko Rinne, Mehrdad Bagheri, Tuukka Tolvanen, and Jaakko Hollmén. Automatic Recognition of Public Transport Trips from Mobile Device Sensor Data and Transport Infrastructure Information. In *Personal Analytics and Privacy. An Individual and Collective Perspective*, Lecture Notes in Computer Science, pages 76–97. Springer, Cham, September 2017. ISBN 978-3-319-71969-6 978-3-319-71970-2. doi:10.1007/978-3-319-71970-2_8. URL https://link.springer.com/chapter/10.1007/978-3-319-71970-2_8.

[24] Roelant A. Stegmann, Indrė Žliobaitė, Tuukka Tolvanen, Jaakko Hollmén, and Jesse Read. A survey of evaluation methods for personal route and destination prediction from mobility traces. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, November 2017. ISSN 1942-4795. doi:10.1002/widm.1237. URL http://onlinelibrary.wiley.com/doi/10.1002/widm.1237/abstract.

[25] Strathman, James G and Dueker, Kenneth J. Understanding Trip Chaining. 1990. URL http://nhts.ornl.gov/1990/doc/attributes.pdf.

[26] X. Su, H. Tong, and P. Ji. Activity recognition with smartphone sensors. *Tsinghua Science and Technology*, 19(3):235–249, June 2014. doi:10.1109/TST.2014.6838194.

[27] Robert Weibel. Generalization of spatial data: Principles and selected algorithms. In *Algorithmic Foundations of Geographic Information Systems*, Lecture Notes in Computer Science, pages 99–152. Springer, Berlin, Heidelberg, 1997. ISBN 978-3-540-63818-6 978-3-540-69653-7. doi:10.1007/3-540-63818-0_5. URL https://link.springer.com/chapter/10.1007/3-540-63818-0_5.

[28] Yilin Zhao. Mobile phone location determination and its impact on intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 1(1):55–64, March 2000. ISSN 1524-9050. doi:10.1109/6979.869021.