GPS Data Management with Applications in Collective Transport[*]

Christian S. Jensen Dalia Tiešytė

Department of Computer Science, Aalborg University, Denmark

Abstract

Recent technological advances in geo-positioning, mobile communications, and computing hardware are combining to offer new opportunities for improved collective transport services. While the routes are getting more and more congested, it is important to attract people to use public transport, which should be more attractive compared to the private cars. This paper presents the objectives of the newly-started project that covers the development of methods for GPS data management with applications in collective transport. The proposed methods are expected to increase the quality of the collective transport services and improve collective transport management systems, such as AVL (automated vehicle location) systems. The project focuses on bus travel time prediction and communication in-between the buses and the server.

1 Introduction

Recent technological advances in geo-positioning, mobile communications, and computing hardware are combining to offer new opportunities for improved collective transport services. Several places in Denmark, including in Aalborg and the surrounding region, the public bus services now provide on-the-fly information to their users, including the current locations of buses and the predicted arrival times at bus stops. The buses use the Global Positioning System (GPS) for positioning and use wireless communications such as GPRS for communicating their positions to a central server.

This project's objective is to develop efficient software techniques that utilize GPS data for establishing a better correspondence between the status of the buses, most importantly their current positions, and the information about this status as known by the surrounding infrastructure, e.g., the central server and the on-line electronic displays at bus stops. In particular, focus is be on more accurate prediction of the arrival times of buses at bus stops, on reduction of the communication needed between the buses and the infrastructure, and on improved scheduling of buses.

Real-time prediction of bus arrival times have been studied in literature for a couple of decades. Currently, automated vehicle location (AVL) systems has become an active research area within ITS. Bus arrival time prediction (ATP) is an important part of AVL systems. The main methods in use are Kalman filter (KF), e.g., [1, 4, 11], and artificial neural networks (ANNs), e.g., [2, 5, 6, 7, 10]. Some researchers use simple mathematic algorithms, e.g. prediction according to deviation from the schedule [8]. KF prediction is mainly based on real-time data, although historical data can be used as well as an external influence parameter. KF works well when a short-term prediction is needed, but deteriorates when prediction is needed for more than one time step. ANNs are based on historical data. Their learning are computationally expensive, therefore, the ANNs cannot be update on real time.

It is a central hypothesis of the project that it is often possible to achieve better ATP at bus stops when using historical and current GPS data, in comparison to what is offered by currently used techniques. This project extends the existing works by developing techniques for the modeling of time and space varying bus speeds based on GPS data, and the project also employs these methods in comprehensive position-prediction algorithms.

---

[*] Contact details: Christian S. Jensen, Department of Computer Science, Aalborg University, Fredrik Bajers Vej 7E, DK-9220 Aalborg Øst, Denmark, e-mail: csj@cs.aau.dk, phone +45 96358900, fax: +45 98159889.

Further, the ATP pursued in this project is faced with the challenge of ensuring low-cost communication, which is done by sending updates from the buses only when needed to preserve accuracy guarantees, rather than at pre-defined time intervals.

The envisioned techniques will use historical GPS data from the buses' traversals of their routes for modeling the speeds of the buses along their routes for different days and different times during the day. Current GPS data will then be used in conjunction with the resulting model. The new techniques will be compared with the existing techniques used by our enterprise collaborators. In the project, the techniques that aim to reduce the communication between buses and the central server will be developed as well. A test infrastructure will be developed that allows to empirically evaluate and compare techniques.

The techniques developed in this project are expected to be applicable in other areas of intelligent transport systems than for buses. Envisioned applications include the use of taxis for transportation of physicians and medical patients, children in rural areas, and the elderly. Increasingly efficient transport contributes to European Union's strategy to make transportation safer and more environmentally friendly.

In addition to a group of researches in Aalborg University, a number of Danish and international ITS-related companies are being involved in the project. **Nordjyllands Trafikselskab (NT)** [9] provide public bus transport in Northern Jutland. They supply GPS data for the project and evaluate the project results from the point of view of the user. **TNC Connect A/S Denmark (TNC)** [13] are suppliers of computing, positioning, and communication infrastructures for buses, including the NT buses. In addition, TNC are working more generally on the development of intelligent transport systems for collective buses. Their AVL (Automated Vehicle Location) systems are aimed at providing high-quality bus services. **Center for Intelligent Transport Systems (CITS)** [3] is a research center that focuses on the development of intelligent transport systems. The center interconnects the relevant research projects at Aalborg University with national and international companies to achieve synergy and supply knowledge. CITS are offering their networking support to the project.

The rest of the paper is outlined as follows. Section 2 discusses the ATP algorithms. Section **??** discusses the methods for reduction of communication cost between the buses and the central server. Section 4 summarizes the paper.

2 ATP and Communication System

We will develop methods that predict the bus arrival and departure times by deriving travel time patterns from historical data, and fitting these patterns to the real-time data. A number of factors have to be considered. Some of the algorithms give long-term predictions, while the others are suitable only for short-term predictions. Specific algorithms can be derived for rural and urban areas. Different algorithms may have different timing and accuracy requirements. A short-term prediction is expected to be more accurate, while the long-term prediction is often allowed to have a bigger error interval. Sometimes it may be required that not the arrival time, but the expected interval of arrival time $[t_{min}, t_{max}]$ should be provided. The algorithms will be developed that dynamically choose the most proper algorithm for a specific situation.

Patterns of GPS data The goal of pattern derivation is to distinguish patterns in GPS data and correlation with other data (e.g. time). The derived patterns can then be used in ATP algorithms as a summary of historical data.

Datamining methods can be applied for pattern recognition. Datamining helps to extract previously unknown, comprehensive information from large databases. The datamining process is as follows: (1) selection of data, (2) preprocessing of data, (3) transformation of preprocessed data, (4) pattern recognition from transformed data, (5) and knowledge extraction from the patterns [12]. The classification algorithms are divided into (1) template matching that measures the similarity between two entities of the same type, (2) statistical approach that calculates the probabilities of the entity to be in certain classes, (3) syntactic or structural matching where rules help to construct the sentences that represent the patterns, (4) machine learning algorithms that learn from a training set, where the class value is known, and (5) ANNs that are non-algorithmic methods that can learn complex non-linear relationships between the attributes.

We define two kinds of patterns. A speed-position pattern is defined as a sequence of speeds $SSeq =$

$[s_1, ..., s_n]$, defined for each road segment Seg_i, $i = 1, ..., n$, where n is the number of segments in the route. The speed 0 may only be defined for a vertice Ver_i, $i = 0, ..., n$, where n is number of stops in the route, and Ver_0 is the starting point of the route. For each vertex i, a stopping time st_i is defined. A speed-time pattern is defined as a sequence of speeds for each time segment that may be either segment travel time $t(Seg_i)$ or vertice waiting time $t(Ver_i)$. Example graphs of patterns Figure 1.

The data for pattern recognition may be grouped by already known factors, such as time of the day, day of the week, weather conditions, etc. Also, new unknown dependency patterns may be derived. We are concerned with both deriving specific patterns for the transportation data and developing methods for the derivation of the patterns.

ATP System Architecture The structure of the system is given in Figure 2. The server has a *Tracker* module and a *Predictor* module. The *Tracker* module tracks and predicts current positions of the buses. When an update is generated, the tracker updates its state. Tracking is discussed more in Section **??**.

The *Predictor* module predicts arrival times for all buses of the system. The *Predictor* obtains the current positions of the buses from the *Tracker*. It also has an access to the database of external data. The patterns of historical data, external data, and the current positions of buses are used to make predictions. The server sends updates to the bus stop terminals with expected arrival times. The buses are equipped with GPS receivers. The buses accumulate their position data for the purpose of further processing. They may also store some additional data for tracking improvement.

ATP Algorithms An ATP consists of pre-processing of historical data, when the patterns of GPS data, according to external factors, are derived. The on-line algorithm chooses the best-fitting pattern according to known conditions (e.g., calendar). The pattern is then adjusted on-line as the bus proceeds on its route. The adjustments are made according to the waiting times at the traffic lights. In this model, the communication between the server and buses is not considered, and it is assumed that all GPS available GPS data is known on the server.

The external data is the data that can influence the speed of a bus. The following data may be considered: calendar—the time of the day, the day of the week, special days of the year, meteorological data—the weather conditions that influence driving conditions on the road, traffic congestion data, automatic passenger count (APC) data.

The basic steps of the pre-processing algorithm are as follows: (1) Select the categories of the patterns, according to which the patterns will be derived. We have chosen the following criteria for categorization: time of the day and day of the week. More criteria will be derived in the project. (2) Group the data according to time of the day and day of the week. The number of groups have to be determined according to the differences in data. Heuristic algorithm is used to determine the number of groups. (3) In a single group, derive patterns of speed in correlation with position on the road.

The pattern has to be chosen in the initialization step of the on-line algorithm. The initialization of the on-line algorithm is as follows:

1. Receive n initial GPS positions on the route RP_i, $1 = 1, ..., n$.
2. Apply the map-matching algorithm to these positions and obtain n map-matched positions MP_i, $1 = 1, ..., n$ (map-matching is a process of locating a bus on a segment or vertice of its route given its GPS position).
3. Choose possible models that fit the external data and/or the initial sequence of positions. The

The basic steps of the on-line algorithm after the initialization phase are as follows:

1. Receive GPS position, RP_i.
2. Map-match the RP_i using the map-matching algorithm, and obtain a position on the road MP_i.
3. Compare how the position MP_i fits to the chosen pattern.
4. Adjust the pattern so, that it fits to the current situation.
5. Merge the pattern with the schedule assuming the fact that the bus cannot leave a bus stop before the scheduled time.

6. Calculate the predicted arrival times to all bus stops in the route according to the available information.

The bus arrival time at stop $stop_i$, when a bus has arrived at stop $stop_k$, is calculated from a speed-position and speed-time patterns by an equation:

$$t^k_{\text{arr}} = t^i_{\text{arr}} + \sum_{j=i+1}^{k-1} (s_j/dist(seg_j) + st_j) + s_k/dist(seg_k) \tag{1}$$

A simple pattern fitting algorithm, when the pattern is already chosen, works as follows: Equation 1 is applied when the bus arrives to bus stop $stop_i$. However, this algorithm is highly dependent on the historical data (the pattern). More dynamics has to be introduced into the model. There are a few ways, that will be considered in the continuation of this work:

- Use a function that increases/reduces the travel time of the up-coming links according to the travel time of already traveled links. The differences between the travel time, the schedule time, and the pattern time are considered.
- Use a function that takes into consideration the travel time of one or few previous buses on the same route, that is, for a bus $bus_{i,k}$ consider l buses $bus_{i,k-j}$, $j = 1, ..., l$.
- Combine both above listed methods.

KF algorithms are widely used for link travel time prediction and bus travel time prediction, as discussed in Section 1. We will consider the use of KF algorithms together with pattern fitting.

3 Communication System

The communication cost has to be reduced by reducing the number of updates to the server and still preserving the required accuracy guarantees for bus locations. The bus sends an update to the server only when the prediction deviates from the actual movement of the bus by a chosen threshold. Two methods will be analyzed in the project: (1) update only when the actual position of the bus deviates from the predicted position of the bus on a server; (2) update only when the arrival time to the nearest bus stop predicted by the bus deviates from the arrival time predicted by the server by a chosen threshold. The second policy may require less updates, but it can only be applied, if the knowledge about the current bus position is not needed.

The class diagram of communication system is shown in Figure 4. The class *Bus* represents a bus on the route. An object is created for each bus. The *Bus* is aware of its *Route* that is represented as a sequence of vertices of class *Vertice* (that can be either a *Stop* or a *Crossing*) and segments of class *Segment* (road segments). These objects have certain properties, e.g. geometrical information. A *Bus* class contains an object *BusTracker* that tracks the location of the bus and updates the tracker class *CentralTracker* that is used by a *CentralServer* by sending a message *UpdateServer*. The server subsequently has to update each *Display* that is physically allocated in bus stops and informs the passengers about expected arrival times.

4 Summary

The paper has introduced to the research directions of the discussed project. Namely, the algorithms for ATP and communication between the buses and the central server will be developed. The full version of the paper will provide substantially more details on these algorithms and the expected benefits to the transportation companies and their clients.

References

[1] F. Cathey and D. Dailey. A prescription for transit arrival/departure prediction using automatic vehicle location data. *Transportation Research Part C*, pages 241–264, 2003.

[2] S. I.-J. Chien, Y. Ding, and C. Wei. Dynamic bus arrival time prediction with artificial neural networks. *Transp. Engrg.*, 128:429–438, September/October 2002.

[3] Center for Intelligente Transportsystemer (CITS). *http://www.cits.aau.dk/.*

[4] D. Dailey, S. Maclean, F. Cathey, and Z. Wall. Transit vehicle arrival prediction: An algorithm and a large scale implementation. *Transportation Research Record, Transportation Network Modeling 1771*, pages 46–51, 2001.

[5] J. R. Hee and L. R. Rilett. Bus arrival time prediction using artificial neural network model. In *Proceedings of the 7 th International IEEE Conference in Intelligent Transportation Systems*, October 2004.

[6] R. H. Jeong. *The prediction of bus arrival time using automatic vehicle location systems data*. Doctoral dissertation, Texas A&M University, December 2004.

[7] R. Kalaputapu and M. J. Demetsky. Modeling schedule deviations of buses using automatic vehicle location data and artificial neural networks. *Transportation Research Record 1497*, pages 44–52, 1995.

[8] W.-H. Lin and J. Zeng. An experimental study on real time bus arrival time prediction with GPS data. *Journal of Transportation Research Board*, pages 101–109, 1999.

[9] Nordjyllands Trafikselskab (NT). *http://www.nordjyllandstrafikselskab.dk/.*

[10] T. Park, S. Lee, and Y.-J. Moon. Real time estimation of bus arrival time under mobile environment. In *Computational Science and Its Applications - ICCSA 2004, International Conference*, volume 3043 of *Lecture Notes in Computer Science*, pages 1088–1096, 2004.

[11] A. Shalaby and A. Farhan. Prediction model of bus arrival and departure times using avl and apc data. *Journal of Public Transportation*, 7, 2004.

[12] B. Son, H. J. Kim, C.-H. Shin, and S.-K. Lee. Bus arrival time prediction method for its application. In *Knowledge-Based Intelligent Information and Engineering Systems, 8th International Conference*, volume 3215 of *Lecture Notes in Computer Science*, pages 88–94. Springer, 2004.

[13] TNC Connect A/S. *http://www.telenorconnect.com/.*

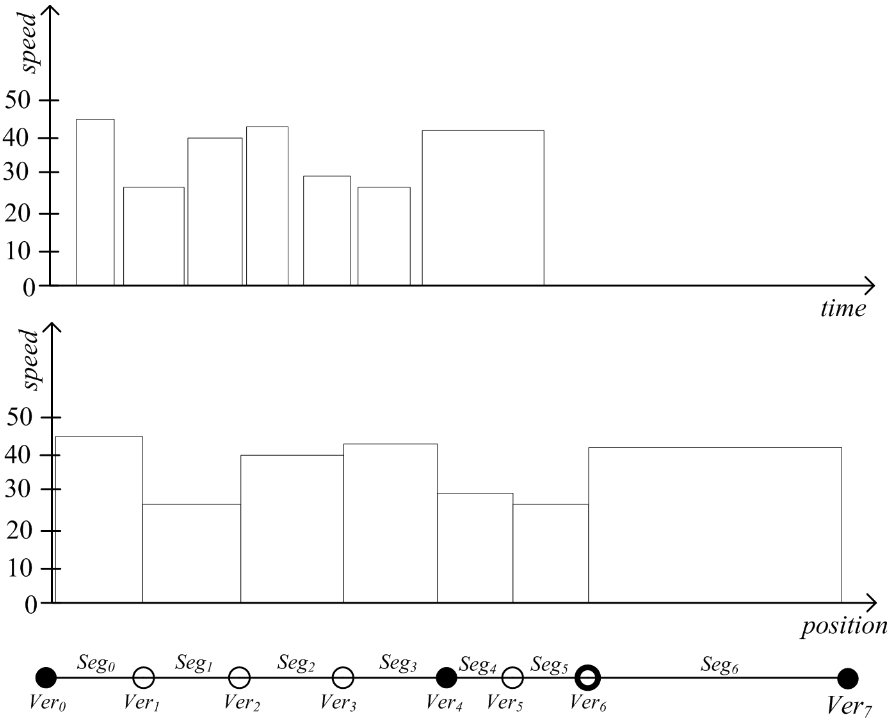

Figure 1: The speed patterns

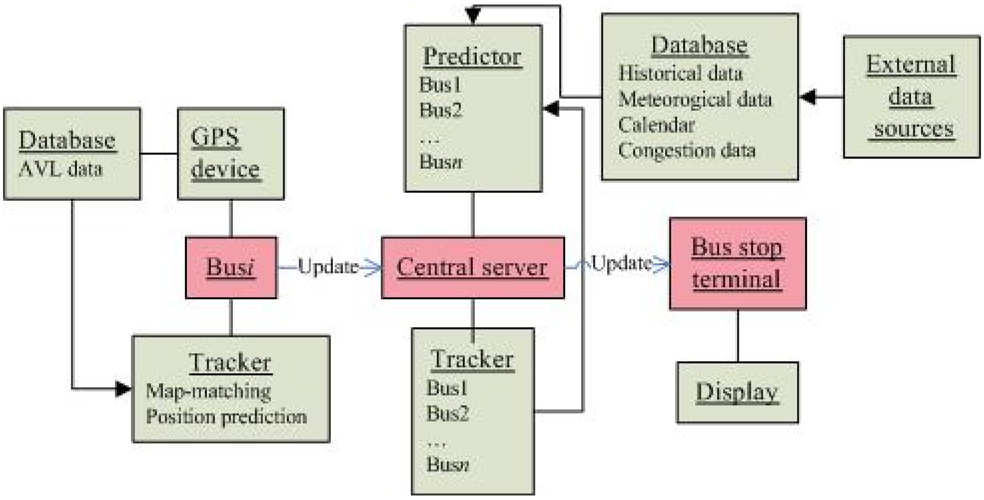

Figure 2: The system architecture

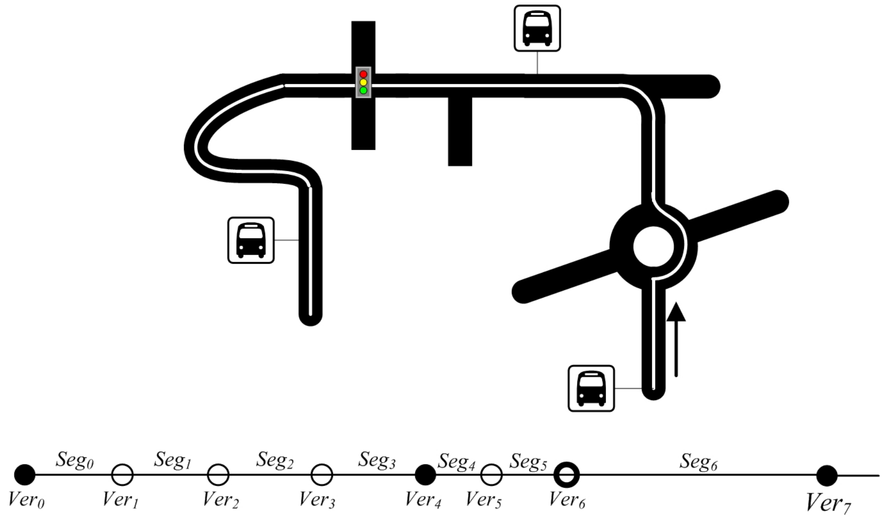

Figure 3: The road representation

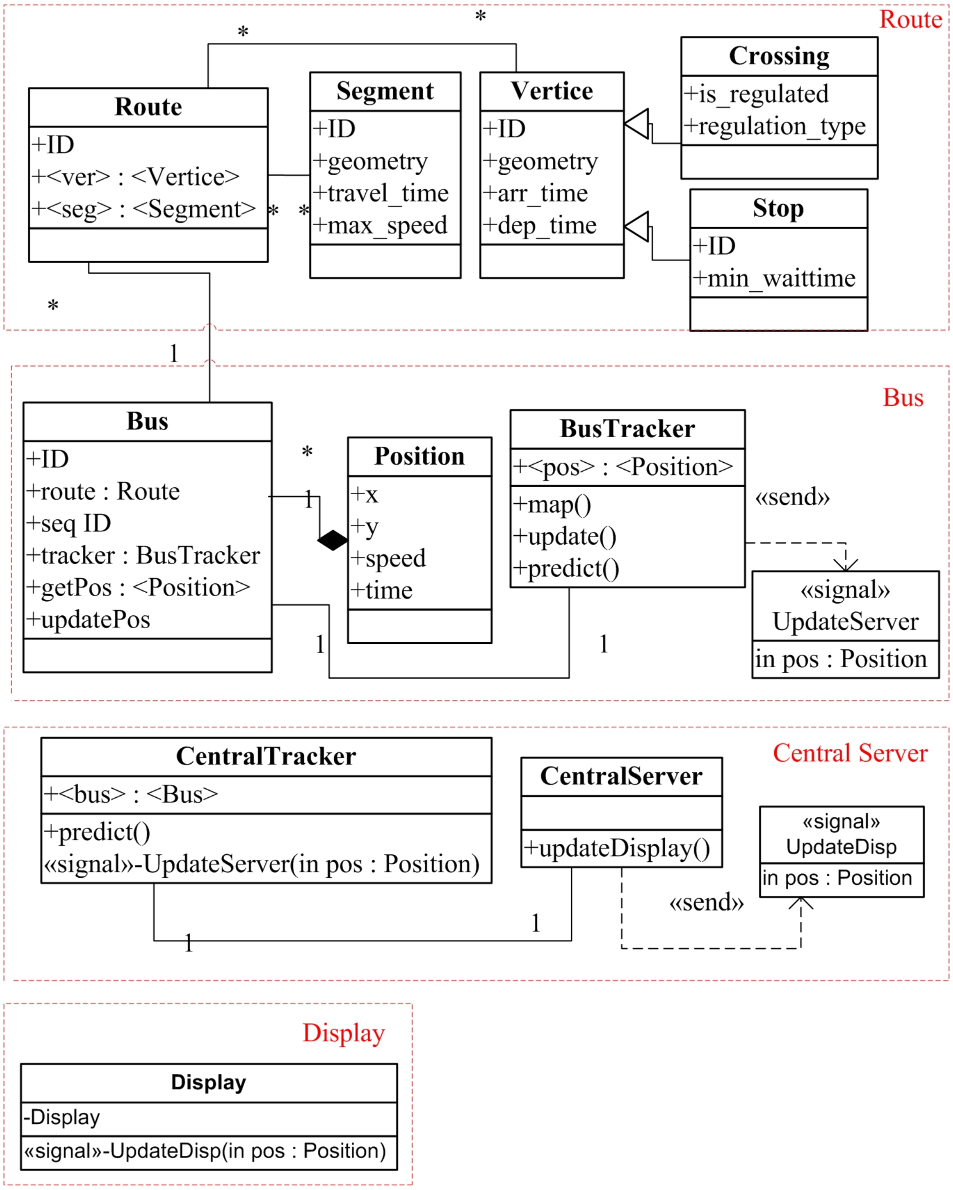

Figure 4: The UML diagram of communication system