Proceedings of the 11th International IEEE
Conference on Intelligent Transportation Systems
Beijing, China, October 12-15, 2008

An Estimation-based Automatic Vehicle Location System for Public Transport Vehicles

Tino Morenz and René Meier, *Member, IEEE*

Lero @ TCD, Department of Computer Science, Trinity College Dublin, Ireland
morenzt@tcd.ie, rene.meier@cs.tcd.ie

Abstract—**Public transport vehicles often share a road network with other road users making their journeys susceptive to changing road conditions and especially to congestion. Travelers using such public transport increasingly depend on real-time information to plan their journeys. While such information can be provided by Automatic Vehicle Location (AVL) systems, AVLs depend heavily on large-scale deployment of designated sensory equipment, which may prevent their pervasive adoption. This paper presents a system for estimating vehicle location based on information generated by data sources typically integrated within existing ITS platforms. This enables location estimation for public transport vehicles without the need for deploying a designated sensor infrastructure in each vehicle, thereby reducing deployment and maintenance cost significantly. A prototypical vehicle location estimation system has been realized as part of and using data provided by the iTransIT ITS framework. Initial evaluation results show that such a system is feasible in a distributed manner and that estimated results are within 20% compared to empirical data.**

I. INTRODUCTION

With the price of transportation rising people start to rediscover the advantages of public transport. Public transport can provide fast, reliable, and 'eco'-friendly transportation at reasonable cost. However, when public transport vehicles, such as buses, are sharing the road network with other road users, their travel speed and reliability can decrease significantly due to varying traffic conditions and congestion.

To ease the impact of varying travel times, public transport providers are inclined to provide up-to-date journey information to travelers using Real-Time Passenger Information (RTPI) systems. Such systems may provide information on the current location of vehicles, estimated travel times of vehicles and passenger waiting times at stops. Travelers can then use this up-to-date information to adjust their travel plans accordingly and ultimately to plan their journey more effectively.

Information on vehicle positions is at the heart of RTPI systems and used to calculate travel times and to present useful information to travelers. Data on vehicle location is traditionally gathered using an Automatic Vehicle Location (AVL) system that tracks each individual vehicle and collects the location information in a central depository. Although the cost of radio location technologies, such as GPS, has significantly decreased over the last decade, the cost of deploying and maintaining an AVL system is often immense. Public transport fleets of several hundred vehicles are not uncommon and designated location technology needs be deployed in every vehicle.

This paper presents iTranSIM, an estimation-based approach to obtaining information on vehicle location that does not depend on the deployment of designated sensory equipment in each vehicle. Our approach uses statistical information derived from existing Intelligent Transportation Systems (ITSs) rather than depending on sensors and radio communication equipment installed in vehicles. Contextual information describing the environment of public transport vehicles as well as journey specific information is used to estimate vehicle locations. Information on the environment may include the road network and prevailing congestions levels while journey information may include journey start times and routes. We argue that this information is typically available from existing ITS infrastructures and hence, we propose a system that extends ITS architectures thereby leveraging existing data sources and the information sharing capabilities inherent to these architectures.

A prototypical version of the vehicle location estimation system has been realized as part of the iTransIT ITS framework [1]. Relevant contextual transport information from a prototypical realization of iTransIT is used to simulate traffic in a real-time, online simulation. From that simulation we can extract location data for public transport vehicles, which can then be used as part of a RTPI system providing information to travelers. The evaluation shows that such a system is feasible in that it can be realized as part of an ITS framework using existing ITS data and in that it can support metropolitan scale transportation systems. Furthermore, initial evaluation results show that estimated results are within a 20% margin of empirical measurements. Hence, we expect that given the necessary contextual data is available, the system can be realized and deployed on a large scale at a low cost compared to traditional approaches.

The remainder of this paper is structured as follows: Section II gives an overview of existing AVL technologies. Section III introduces the iTransIT ITS framework for integrating individual transportation systems and related user services. Section IV describes the vehicle location estimation system and its integration into the iTransIT ITS

Manuscript received June 15, 2008.

framework. Section V presents our evaluation of this work. Finally, section VI concludes this paper by summarizing our work.

II. Automatic Vehicle Location

In 1977, Riter and McCoy [2] presented an overview of automatic vehicle location. At the time, the main purpose of automatic vehicle location was to help law enforcement agencies to dispatch their vehicles rather than passenger information. They identified three major categories of approaches and these approaches are still applied in today's AVL systems. These include:

- Dead-Reckoning
- Proximity based
- Radiolocation

Unlike the vehicle location estimation system described in this paper, the approaches called out above all depend heavily on designated equipment deployed in each vehicle. The individual approaches are introduced and discussed below. Using statistical vehicle location estimation based on existing ITS data as proposed in this paper is, to the best knowledge of the authors, a novel approach that has not been proposed thus far.

A. Dead-Reckoning

The Dead-Reckoning approach is based on recording the direction a vehicle travels and the distance covered. This information is then used to follow the path of the vehicle in order to deduce its location. The main problem with Dead-Reckoning is that errors caused by external influences and internal imprecision accumulate over time because any position calculation depends on previous calculations.

Dead-Reckoning is mainly in use for tracking track bound vehicles and is often combined with other AVL systems as a backup. For example, Zaho et. al [3] describe a system where GPS data is enhanced with Dead-Reckoning data to compensate decreased reception of GPS signals in build up areas.

B. Proximity based AVL

Proximity-based approaches depend on signal receivers strategically placed at locations along the routes of the tracked vehicles. Vehicles are equipped with senders transmitting a beacon that identifies the vehicle with the receiver. An advantage of proximity-based approaches is that the actual position of the vehicle is captured at the receiver rather than in the vehicle eliminating the need for transferring the location information from the vehicle using wireless communication. However, in order to gain a sufficiently fine grained resolution of vehicle locations the number of deployed receivers needs to be significant. Also, the number of receivers depends on the area and routes used by the vehicles, which may increase their number further, especially, when used in large urban environments.

Although mainly used for track bound vehicles, new sensor technologies, such as camera-based license plate recognition systems, have recently increased interest in proximity-based AVL systems for freely moving vehicles [4].

C. Radiolocation

The Radiolocation-based approach uses triangulation, where a sensor receives radio signals from different transmitting stations and uses the delay of the different signals to calculate its position. Currently, the NAVSTAR Global Positioning System (GPS) is the main source of signals for Radiolocation. Satellite-based Radiolocation systems have replaced many of the other AVL systems since the GPS signals can be used free of charge. For example, the RTPI system Q-Time [5] uses GPS as the main source of vehicle location.

III. iTransIT Framework

The iTranSIM vehicle location estimation system has been designed based on the iTransIT ITS framework. As illustrated in Figure 1, the iTransIT ITS architecture structures legacy systems, iTransIT systems, and context-aware, end-user applications into three tiers. These tiers define the relationships between systems and applications, and provide a scalable approach for integrating systems, in that individual components can be added to a specific tier without direct consequences to the components in the remaining tiers.

A. Tier Architecture

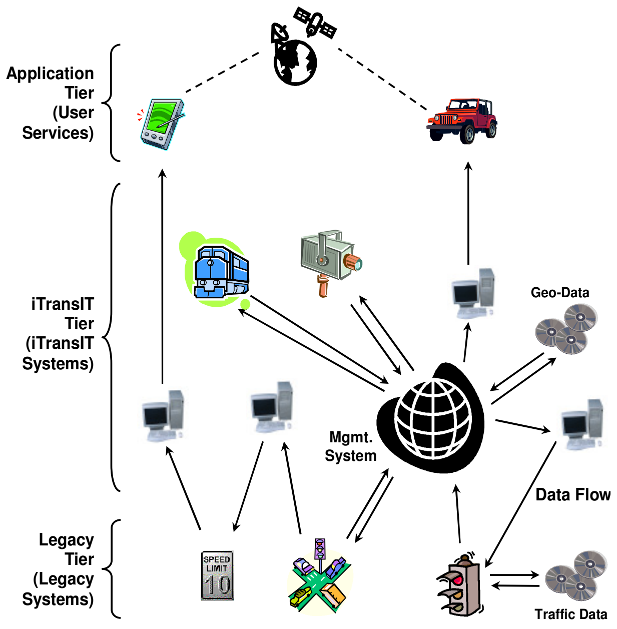
Application Tier (User Services)
iTransIT Tier (iTransIT Systems)
Legacy Tier (Legacy Systems)
Geo-Data
Mgmt. System
Data Flow
SPEED LIMIT 10
Traffic Data

Figure 1. The iTransIT Framework

The iTransIT system architecture supports the integration of legacy systems, and supports future systems that conform to the overall architecture and data-layer.

The purpose of the iTransIT tier is to integrate transportation systems that model spatial information and implement the Spatial Application Programming Interface (Spatial API) [6]. Therefore, this tier comprises of a federation of transportation systems that implement the spatial data layer. The data layer is distributed across these iTransIT systems, with each system implementing the subset of the overall layer that is relevant to its operation. iTransIT systems maintain their individual information, which is often gathered by sensors or provided to actuators, by populating the relevant part of the spatial data layer. However, some of the information maintained in an iTransIT system specific part of the data layer may actually be provided by underlying legacy systems. Most significantly, traffic information captured in this tier is maintained with its primary-context, and persistently stored data is geo-coded typically by systems exploiting a database with spatial extensions. The iTranSIM vehicle location estimation system exists in the iTransIT layer and implements its part of the spatial data layer, namely, the vehicle locations. It implements the Spatial API to enable access to vehicle location data and queries information from other relevant iTransIT Systems.

The application tier includes pervasive value added services that provide context-aware user access to traffic information. These services use the distributed data layer and associated context to access information potentially provided by multiple systems. They could include a wide range of interactive (Internet-based) and embedded control services, ranging from the monitoring of live and historical traffic information to the display of waiting times at bus stops or tram stops.

B. Common Spatial Data Layer

The spatial data layer, common to all iTransIT systems, is comprised of a set of potentially distributed sub-layers and represents the central component of these systems. Individual iTransIT systems implement one or more of these sub-layers (or parts of sub-layers) and maintain the static, dynamic, live, or historical traffic data available in that sub-layer. For example, a system might implement a sub-layer describing the current weather conditions, while another sub-layer capturing intersection-based traffic volumes might be maintained by a different system. This allows the iTranSIM system to query all necessary data using a common mechanism, the Spatial API. Furthermore, other iTransIT systems can use the same common interface to access and retrieve the data generated by iTranSIM.

IV. The iTranSIM Location Estimation System

The iTranSIM vehicle location estimation system uses the data sharing capabilities of the iTransIT architecture to access contextual information from a range of legacy ITS systems. The information derived from these systems is then used to estimate the location of public transport vehicles and the resulting data is provided as part of the overall spatial data layer.

A. Architecture

To estimate the vehicle locations we will fetch the existing contextual data from the ITS architecture and use it to create an online traffic simulation. Traditionally, traffic simulations are run in batch mode, i.e. a dataset is provided to the simulation, the simulation is run and the results are analyzed once the simulation has finished. An online simulation on the other hand runs synchronized with the real world, i.e. in real time. Information such as congestion levels are fed into the simulation constantly to recreate traffic patterns that match the real world at any point in time. By simulating public transport vehicles in the online simulation we are able to gather location information for those vehicles within the simulation environment. With a sufficiently accurate online simulation the measured locations are expected to correspond to the vehicle locations in the real world. The data can then be used by other services, for example to provide real time passenger information to travelers.

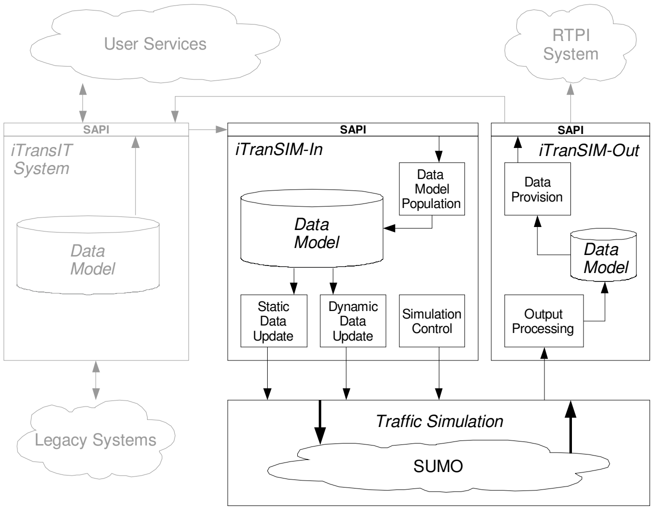

Figure 2. The iTranSIM architecture

Figure 2 illustrates the architecture of the iTranSIM vehicle location estimation system. It consists of three components, two are located in the iTransIT system tier and one resides in the legacy system tier. The iTranSIM-In component provides the input required by the traffic simulation by gathering relevant information from other iTransIT systems using the Spatial API. This data is then used to model traffic using the Simulation of Urban Mobility (SUMO) traffic simulator [7]. The output of the traffic simulation is processed by the iTranSIM-Out component. This component retrieves the vehicle location information estimated in the traffic simulator and makes it available to other iTransIT systems and services through its realization of the Spatial API.

B. Data Model

To create an accurate online simulation it is vital to create a precise model of a vehicle's journey. The model is then populated with information from other ITS components, and used in the traffic simulator. To build this model we have identified and modeled three components. First, the environment, i.e. the road network, second, the traffic including all vehicles and their behavior and finally, the journey of public transport vehicles with their specific routes and timings.

Modeling the road network accurately is essential because this part of the model defines the possible flows of active vehicles in the simulation and therefore has a significant impact on the journey times of individual vehicles. The two main components of the road network modeled for this approach are junctions and links. Junctions have the following properties:

- Physical location (geo coordinates)
- Type (traffic lights, priority based or roundabout)
- Traffic Light sequences or priorities for incoming links

Links connect the junctions and consist of a start and an end junction and a number of lanes. Lanes have the following properties:

- Speed limit
- Exclusions (e.g. no HGVs)
- Exclusivity (e.g. bus lanes)
- Condition (e.g. slower than usual due to snow)
- Induction loops and their positions
- Induction loop counts
- Bus stops and their positions
- Legal turning maneuvers on exiting the lane
- Turning rations on exiting the lane

To model the traffic we used a microscopic car-following model. In this model every vehicle is modeled as a single entity and its behavior is influenced by the vehicle in front of it. A minimum distance between two vehicles is to be kept at all times depending on the velocity of the leading car. Additionally, the reaction time needed by a driver to respond to actions taken by the leading vehicle is taken into account.

To enable the car-following model we have added vehicles with following properties to our data model:

- Acceleration
- Deceleration
- Maximum speed
- Driver imprecision

The above properties determine how the vehicles pass through the simulation, their routes are determined by the turning ratios stored for each lane, i.e., once they reach the end of a lane they will turn to the next lane with the given percentage. These ratios can be determined from traffic count data for the adjacent lanes which eliminates the need for detailed journey information of individual vehicles.

The journey of a public transport vehicle, such as a bus, is basically the same as a journey of an ordinary vehicle in the microscopic simulation, but with a few notable exceptions. The routes of a bus trip are predetermined, so are the starting times. Using this information, we have modeled a public transport journey with the following properties:

- Route (list of lanes and bus stops served)
- Starting time
- Average passenger numbers per stop

With the given information we are able to represent the journey of a public transport vehicle within its environment which in turn can be used to create the online simulation.

C. ITranSIM-In

To create an accurate online simulation it is necessary to provide all the required data in a timely manner to the simulation. The iTranSIM-In component collects and stores the data to populate the data model and provides it to the traffic simulator. It also manages the simulation to guarantee the synchronization between the incoming data and the simulation. To gather the necessary information other ITS components are queried using the iTransIT Spatial API. The data collected by the iTranSIM-In component can be divided into two categories: static and dynamic data. Static data is data that is not expected to change regularly and includes road layouts, bus routes and timetables. Dynamic data typically changes frequently and includes traffic counts provided by induction loops.

After retrieving the data from other iTransIT components it is stored in the data model by the Data Model Population component. The data is then processed so it can be used as input data for the simulation. The SUMO traffic simulator requires a number of configuration files to run. These are generated from the static data and provisioned through the Static Data Update component. The Dynamic Data Update component updates the traffic simulator at runtime with the data needed to adopt the simulation to changes in the real world.

The last element is the Simulation Control. It ensures that the simulation stays 'in sync' with the incoming data and handles any critical errors that may occur during the simulation.

D. Online simulation

The traffic simulator creates a model of the environment and the traffic based on the data provided by the iTranSIM-In component. For our experiments we have chosen the open source traffic simulator SUMO and integrated it into iTranSIM as a legacy system. SUMO uses a microscopic simulation model, which allows us to track individual vehicles and extract location data for them.

By constantly updating the simulation with live traffic data we create a reflection of the traffic situation in the real world. This approach requires more resources than a purely statistical approach based on historical data but it will adapt to changes in the traffic situation immediately which will yield more accurate results in extreme situations such as very long journey times due to unusually high congestion.

To create the online representation of the current traffic situation it is necessary to adjust the number of vehicles in the simulation according to the induction loop counts provided through the Dynamic Data Update. Traffic can be adjusted by adapting either the flow or the density. We based our design on flow calibration which, despite being the simpler approach, has been proven to generate realistic online simulations, such as the simulation of the city of Duisburg, Germany, presented by Wahle et. al [8]. The flow describes the number of vehicles that pass a point in a specific amount of time. To regulate the flow the number of vehicles that pass over an induction loop in the simulation has to be adapted according to the readings of the corresponding induction loops in the real world. Granular induction loop data, consisting of individual numbers for different vehicles classes, can be used to adapt the number of vehicles individually for each vehicle class.

In addition to the general traffic we model each individual public transport vehicle based on its route and scheduled departure time. Thus the location information for each of these vehicles can be extracted at every simulation step and is then processed by the iTranSIM-Out component which makes it accessible to other systems.

E. *Data provision*

The iTranSIM-Out component implements the Spatial API to make the vehicle location data estimated by the simulation available to end-user services in the application tier as well as to other iTransIT systems. The vehicle location information generated by the simulation is stored in a spatial data layer. A record for each vehicle is captured that contains vehicle identification, vehicle location and a timestamp indicating the last update to the record. We envisage that a variety of end-user services, possibly as part of an RTPI system, accesses the estimated vehicle location information to deliver it to travelers and controllers. Likely applications might include real time monitoring of the vehicles, using location data to organize and to schedule connection services, for example, to request one vehicle to wait for another vehicle so passengers can change over, or to display estimated waiting times at stops.

V. Evaluation

A prototype of the iTranSIM vehicle location estimation systems has been realized as an extension of an existing prototype of the iTransIT ITS architecture. The iTransIT system prototype integrates various contextual information derived from a range of ITS currently deployed in Dublin City in the Republic of Ireland. The captured information is real data from legacy systems that has been integrated using emulations of the real legacy system interfaces. The notable exception to using data from the real systems is the data describing the detector loop counts. Although this data is available, its integration into the iTransIT system prototype has not been completed in time for this evaluation. Instead, a batch of empirical loop counts for the part of the road network modeled for this evaluation has been used which allowed us to make observations for average traffic conditions. Furthermore, the prototype was specifically designed so that once the live data becomes available experiments can be run using that data without any further alterations.

All iTranSIM components, as well as the iTransIT system were hosted on a laptop computer with a single core, 2.5 GHz, Pentium 4 processor and 512 MB RAM running Windows XP Professional.

For our experiments we have modeled a section of the Dublin road network, consisting of 100 junctions and 146 links with a total of 288 lanes, 36 of which are bus lanes. Figure 3 shows a high level overview of the section which stretches along one of Dublin's major bus corridors.

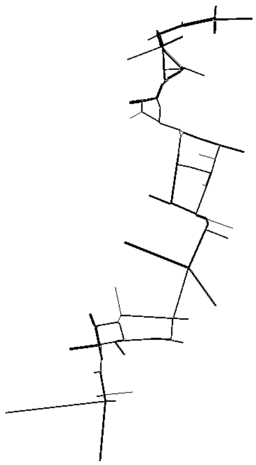

Figure 3. An overview of the modeled road network

A. *Traffic Simulation*

For the evaluation of the online simulation we have used empirical loop count data that was supplied via the Spatial API. The data was taken at a day with average traffic levels, for 20 induction loops spread across the road network that was modeled for our experiment. We used this data to simulate the traffic over the course of one day and in this environment simulated all journeys of a Route 15 bus in Dublin city center, starting with the first bus at 6:55am up until the last bus at 10:50pm adding up to a total of 43 bus journeys over the day. During the same time a total of 80000 vehicles were simulated, up to 750 being active simultaneously.

Figure 4 shows the travel times for individual buses within the simulation. The average vehicle density is split into 3 groups, high, medium and low, which are also presented in the chart (indicated by congestion levels 1-3).

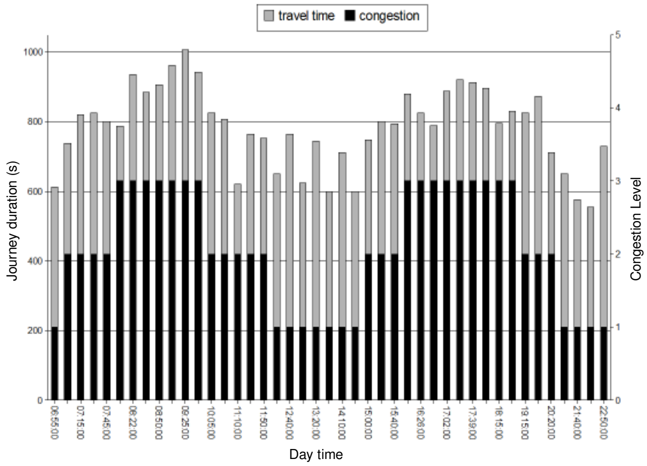

Figure 4. Bus travel times in the simulation

The graph shows that the travel time for buses is longer with heavier traffic. Also, a comparison to empirically collected bus journey times illustrates that overall the simulated journey times deviate less than 20% from the average journey times measured on these bus trips, as shown in Table 1. The journey times were measured over three weeks and the median of those measurements was taken to find an average journey time corresponding to the empirical traffic data we have used. Once live induction loop count data becomes available in iTransIT we intend to repeat these experiments to further validate the accuracy of our approach.

Table 1. Difference simulated and measured travel times

Start Time	**Simulated Journey Time [s]**	**Measured Journey Time [s]**	**Deviation**
09:25	1006	928	**-8.41%**
15:40	794	768	**-3.39%**
18:15	795	991	**19.78%**
22:10	555	689	**19.45%**

B. *Performance and Scalability*

Performance and scalability are vital for an estimation-based system because ultimately, the estimated vehicle locations need to be made available in real-time for large vehicle fleets to be of use to passengers. This implies that a simulation step has to be completed before the next step is due. In the SUMO simulator we used for our experiments the length of a simulation step is one second, implying that all data input and output as well as all processing has to happen within one second. The length of a step could however be extended to account for larger networks where more processing capacity is needed to complete a single simulation step. We have identified three possible bottlenecks that might impact on the performance of our system. Two of those are related to the communication between the components, the third is the processing time needed for the simulation.

The first bottleneck is the data retrieval from iTransIT systems using the Spatial API. The retrieval of the induction loop data has to happen continuously to keep the traffic flow calibration up-to-date. With the current implementation the data has to be requested for each induction loop individually, which leads to a considerable overhead even for our small scenario. The issue was flagged during our evaluation and a revised version of the Spatial API is planned. For evaluation purposes we also used a JDBC interface to access the spatial database directly. The retrieval of 4000 induction loop counts, the amount necessary to cover Dublin's inner city completely, was achieved in approximately 500ms. This shows that once performance improvements are implemented in the Spatial API the data retrieval time will meet required deadline with an adequate buffer.

The second potential bottleneck is the data communication to and from the traffic simulation. In our prototype the data updates were fed into the simulator in binary format using a TCP socket, the output from the SUMO simulator was also over a TCP socket, but in XML format. The messages necessary to transmit 4000 vehicle counts to the simulator added up to a total of approximately 120kB which can be transferred in well under 100ms. On the output side experiments were conducted with 1000 vehicles reporting their position at the same time, which equals approximately the number of buses in the bus fleet in Dublin. Transfer and parsing of this data took about 500ms. From the results we see that the data transfer to and from the simulation can be achieved in the required timeframe while the components can be distributed over the network.

The computation of the simulation is considered another possible bottleneck as it is very CPU intensive. The main influence on the time needed for the computation of the next step is the number of active vehicles in the simulation. The time can be approximated to be a linear function of the number of active vehicles, because for every vehicle the calculation of its new position is independent of the other calculations. In a simulation of Dublin's inner city, which represents approximately 200 miles of road network with up to 20000 active vehicles, the time required to calculate a single step was about 700ms.

Although a simulation step length of one second was used for our simulation the impact on the accuracy is expected to be minimal should the simulation step be extended to a few seconds. This would widen the timeframe required to meet the real time requirements which in turn would allow to perform the simulation with fewer resources.

Taking into account the above measurements and the potential for distributing the different components we conclude that, with the current state of hardware, it should be feasible to realise an estimation-based vehicle location system on a metropolitan scale.

VI. Conclusion

This paper presented a system for estimating the location of public transport vehicles based on contextual information commonly available in existing ITS infrastructure thereby eliminating the need to deploy and maintain sensory equipment in every tracked vehicle. We have shown how such a system can be realized as part of the ITS infrastructure of Dublin. Our initial evaluation shows the feasibility of such an approach based on a prototypical implementation that extends the iTransIT ITS framework.

We envisage our work leading towards an advanced public transport system providing real-time passenger information to travelers. We are planning to further evaluate our approach by increasing the scale of the public transport network and with a special focus on the accuracy of the estimated vehicle locations.

Acknowledgment

The work described in this paper was partly supported by Lero - the Irish Software Engineering Research Centre. We would also like to thank Dublin City Council for providing information on Dublin’s transport infrastructure, which made this work possible.

References

[1] R. Meier, A. Harrington, and V. Cahill, "A Framework for Integrating Existing and Novel Intelligent Transportation Systems," in *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems (IEEE ITSC´05).* Vienna, Austria: IEEE Computer Society, 2005, pp. 650-655.

[2] Riter, S., McCoy, J. "Automatic vehicle location - An overview" *in IEEE Transactions on Vehicular Technology*. Vol. 26, no. 1, pp. 7-11., February 1977

[3] Zhao, L., Ochieng, W.Y., Quddus, M.A., Noland, R.B., "An Extended Kalman Filter Algorithm for Integrating GPS and Low Cost Dead Reckoning System Data for Vehicle Performance and Emissions Monitoring" in *Journal of Navigation* (2003), 56, pp. 257-275, Cambridge University Press, Cambridge, UK, May 2003

[4] Shyang-Lih, C., Li-Shien, C., Yun-Chung, C., Sei-Wan, C., "Automatic license plate recognition" , *in IEEE Transactions on Intelligent Transportation Systems*, Volume: 5, Issue: 1, pp. 42-53, March 2004

[5] Caulfield, B., O'Mahony, M. M., "The provision of on street passenger information via real time passenger information; A case study of Dublin", in *Proceedings of the IEEE Conference on Road Transport Information & Control 2004* (RTIC2004), pp. 1-10, London, UK, 20 - 22 April 2004

[6] R. Meier, A. Harrington, T. Termin, and V. Cahill, "A Spatial Programming Model for Real Global Smart Space Applications," in *Proceedings of the 6th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS 06),* LNCS 4025. Bologna, Italy: Springer-Verlag, 2006, pp. 16-31.

[7] Krajzewicz, D., Hertkorn, G., Rössel, C., Wagner, P., "SUMO (Simulation of Urban MObility); An open-source traffic simulation" in *~Al-Akaidi, A. [Ed.]: Proceedings of the 4th Middle East Symposium on Simulation and Modelling* (MESM2002), SCS European Publishing House, pp. 183-187, Sharjah / United Arab Emirates, Sept. 2002

[8] Wahle, J., Neubert, L., Esser, J., Schreckenberg, M., "A cellular automaton traffic flow model for online simulation of traffic", in *Parallel Computing*, Volume 27, Number 5, April 2001 , pp. 719-735