Demo: Tracking Transit with EasyTracker

Tomas Gerlich tgerli2@uic.edu
James Biagioni jbiagi1@uic.edu
Timothy Merrifield tmerri4@uic.edu
Jakob Eriksson* jakob@uic.edu

Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607, USA

Abstract

EasyTracker is an automated system that assists small public or private transit agencies in deploying bus tracking and arrival time prediction. This demo will showcase how data from GPS sensors embedded in smartphones can be automatically processed in order to accurately estimate routes, bus stop locations, schedules, and make annotated maps with real-time bus tracking and arrival time predictions. We will also demonstrate a website portal which transit agencies can use to further interact with their bus transit systems.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Types of Systems—*Logistics*

General Terms

Algorithms, Design, Experimentation

Keywords

Public Transit, Transportation, Bus, Real-Time Tracking, Smartphone, GPS

1 Overview

Creating a new bus tracking, route classification and arrival time prediction system can be a costly, elaborate, and lengthy process. We offer a system design suitable for small public or private transit agencies in need of bus tracking systems that are ideal for college or corporate campuses, commuter shuttles traveling to transit hubs, or school buses.

*This material is based upon work supported by the National Science Foundation under Grants CNS-1017877 and DGE-0549489, as well as the Transportation Research Board of the National Academies contract Transit IDEA #T-65.

SenSys'11, November 1–4, 2011, Seattle, WA, USA.
ACM 978-1-4503-0718-5/11/11

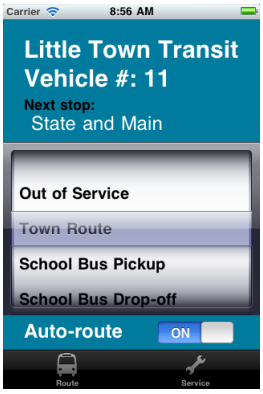

Figure 1. Envisioned smartphone/driver interface.

EasyTracker [1] provides a solution for automating many of the steps in creating a transit tracking and mapping system and it provides an opportunity to interact with the system on a continuous basis.

2 System Description

The transit agency places a smartphone running our application into each bus. Every second, the application transfers the GPS location of the bus to the back-end server which is then able to track in real-time each bus in the system and display its location on a map in a web interface. Figure 1 shows the envisioned driver interface of the smartphone application.

In order to estimate routes, bus stops and schedules, the devices are left in the buses (without any further driver input or interaction with the device) while the driver follows her usual route, stops at bus stops in order to pick-up or drop-off passengers, and maintains a schedule or a headway interval. Once a sufficient amount of data is collected on the back-end server, the system automatically processes the unlabeled GPS traces in order to create accurate routes, bus stops, and schedules for each route in the transit system. The algorithms can produce accurate routes despite the fact that each bus is free to follow different routes on different days while the driver does not provide information about which route she follows. The system also automatically determines the

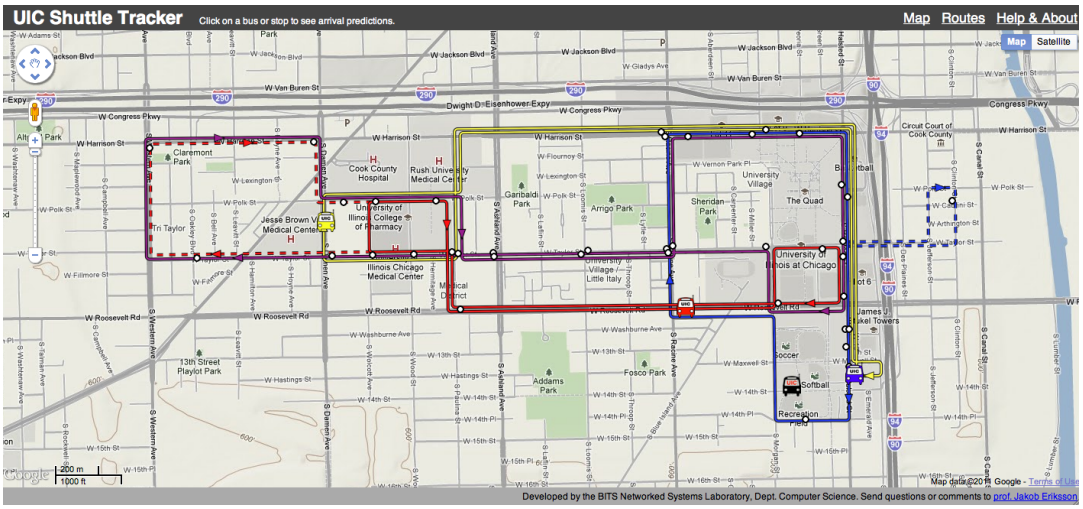

Figure 2. Screenshot of prototype end-user facing web interface.

locations of bus stops, which is a difficult task as the driver typically stops at various other places in addition to regular bus stops (e.g. at intersections, traffic signals, due to congestion, or at staging or maintenance stops). Furthermore, the system produces bus schedules despite the fact that the daily volatility of traffic conditions and traffic congestion typically prevents the driver from maintaining accurate arrival times at each bus stop. Finally, an annotated map of the transit system is automatically created that provides the relevant information about routes and bus stops to the transit rider. Figure 2 is a screenshot of our current end-user interface prototype displaying the route shapes and bus stop locations along with the real-time locations of the buses.

In addition to real-time tracking, after the set of routes and bus stop locations are estimated, the server also performs real-time route matching and classification. Each bus is classified as following a certain route from the set of possible routes, and this information is displayed on the web interface by color-coding each bus icon as in Figure 2. Furthermore, the server generates bus arrival time predictions for each bus stop and shows the estimated arrival times on the web interface. This real-time information enables bus riders to use the transit system more efficiently, reduce wait time, and increase overall safety and convenience. The utility of a real-time tracking and bus arrival time prediction system is quantified in [2].

All of the information about the transit system is accessible via the web interface where the transit agency administrators can log-in. The transit agency has the option to manually adjust the estimated routes, add or delete missing or erroneous bus stop locations, label and color-code each route or bus stop, and add further information about the transit system. In addition, the web interface also displays the transit system performance statistics.

3 The Demonstration

We will demonstrate the functionality of each of the building blocks of EasyTracker.

- **Smartphone Front-end.** The demonstration will showcase our application installed on a smartphone which continuously transfers its GPS location to a back-end server. This application is used for real-time vehicle tracking and classification as well as for GPS trace collection for routes, bus stops, and schedule extraction.
- **Routes, Stops, and Schedule Extraction.** The output and accuracy of our route, bus stop, and schedule extraction algorithms from the collected GPS traces will be demonstrated. In addition, an automatically drawn map of the transit system will be created.
- **Web Interface.** We will show a website portal where transit agencies can log-in and see the output of the extraction algorithms overlaid on a map along with an interface that can be used to correct any erroneous output of the algorithms (e.g. missing bus stops). The interface will also allow labeling of the routes and bus stops and provide further information.
- **Real-time Tracking.** In this part we will demonstrate the functionality of the real-time tracking using the smartphones placed in buses. We will display the movement of buses on a map along with the route matching and classification output. We will also generate and display bus arrival time predictions for each bus stop.
- **Transit System Statistics.** The web interface will also display the performance statistics of the transportation system.

4 References

[1] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson. Easytracker: Automatic transit tracking, mapping, and arrival time prediction using smartphones. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, SenSys '11, 2011. (To appear).

[2] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson. Cooperative transit tracking using smart-phones. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 85–98, 2010.