A SYSTEMATIC APPROACH FOR IMPROVING PREDICTED ARRIVAL TIME USING HISTORICAL DATA IN ABSENCE OF SCHEDULE RELIABILITY

Allen Worthington Rawls

A Thesis Submitted to the
University of North Carolina Wilmington in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

Department of Computer Science
Department of Information Systems & Operations Management

University of North Carolina Wilmington

2008

Approved by

Advisory Committee

Dr. Curry Guinn	Dr. Ling He
Mr. Albert Eby (Wave Transit)	Dr. Ron Vetter (Chair)

Accepted by

Robert D. Roer

Digitally signed by Robert D. Roer
DN: cn=Robert D. Roer, o=UNCW, ou=Dean of the Graduate School & Research, email=roer@uncw.edu, c=US
Date: 2008.05.14 16:52:16 -04'00'

Dean, Graduate School

TABLE OF CONTENTS

ABSTRACT

Public transit operations are susceptible to change, both in traffic flow and other conditions that could affect operations such as bridge openings, road floods, and torrential downpours. Traditionally, riders at waiting stops are not informed of the transit vehicles' status along the route. Although it is normally not needed for daily transit operations, live location information is particularly useful in cases when vehicles are running behind schedule.

This thesis introduces a method for gathering and analyzing historical location and telemetry data of public transit vehicles to better determine estimated arrival time for a vehicle on a closed-loop public transit pattern. The research creates a system for sending real-time locations of transit vehicles to riders through a wide array of mediums including web pages, computer programs, graphical information displays in public locations, mobile phone applications, mobile text messaging, and internet feeds. The system incorporates a weighted estimated arrival time for one route in the city, the University of North Carolina Wilmington campus loop shuttle route, which serves as a working demonstration of these concepts. The approach shows improvement over an arrival time estimate using only average speed

ACKNOWLEDGEMENTS

This work would not have been possible without the support and encouragement of my advisor, Ron Vetter, who always helps me develop new ideas into working projects. To my other committee members, Curry Guinn and Ling He, thank you for your extended support with my research and I am gratified that you accepted me as your advisee.

I gratefully acknowledge the financial support of the University of North Carolina Wilmington's Division of Business Affairs, with associate vice chancellor Sharon Boyd, and Wave Transit, with director Albert Eby, for taking my research ideas and deploying it in a real world setting. Wave Transit contributed the data on which this thesis is based.

I would like to extend thanks to my peers and friends; you were willing to put aside your own research to chat or toss around ideas to assist in any way you could. You provided support and encouragement to make it through the program and helped me retain my sanity. To the faculty of the University, your open doors enabled a relationship, for which I am grateful.

I wish to thank my entire extended family for supporting me in everything I do. Attending college the farthest from home, I appreciate the willingness of my family to let me explore and find my own purpose.

I am very grateful to my parents, Jeff and Molly Rawls, who provided a loving environment and an impetus to succeed. You have remained faithful that I will reach the level of accomplishment that I have struggled to achieve. Your endless belief in my academic abilities has encouraged me to both make it this far and to strive for more.

Thank you to everyone else who played a role in this endeavor.

DEDICATION

This thesis is dedicated to my teachers and mentors, past and present. I cherish your guidance and continued support that remains well beyond the time I was your student. I am delighted to dedicate this to you, in reflection of how you have made a difference in my life. For planting the seed and always believing, I thank you.

Celebrate the temporary, but follow your bliss; adapt or die.

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

In the area of public transportation, there is a lot of emphasis on rider satisfaction. Riders want to know when a vehicle will depart a given location so they can be sure to catch the vehicle with the minimum wait time. Having a system that runs on time can be hard to achieve, but ridership will increase if the system is known to be reliable. Even if the vehicle is off schedule, informing riders how the vehicle location differs from the published schedule would be useful so that commuters can adjust their plans accordingly. Therefore, the ultimate goal is to create a notification system that is aware of the current transit conditions.

The unreliability of public transportation has encouraged many researchers to search for a solution to determine the real-time location of vehicles. This can be accomplished with location sensing devices such as satellite tracking or vehicle proximity sensors. Installing location sensing devices on vehicles serves two purposes: the first benefits the transit operator as a method of quality assurance and asset tracking, while the second benefits the public by providing a notification system for riders. When either system is in place, there is a direct gain to the transit operator.

Once the vehicles are equipped with automatic location equipment, the next step is to provide the information to the public with vehicle locations, estimated arrival times, and estimated wait times. Since 2000, several companies in the United States have begun using this technology to realize the benefits of vehicle tracking from both perspectives. While the majority of tracking companies offer public information services as opposed to

asset tracking, it is important to think about how reliable the system would be without the live tracking, that is, when solely reliant on the published schedule. Even with live reporting, the reliability of transit operations still tends to fall back on the accuracy of the live information, as justification for a late or early arrival, citing that the riders had this information and therefore can make the necessary adjustments. This is a downside for users because live tracking and reporting is designed to improve rider satisfaction.

One example of live tracking is the subway system in Munich, Germany that uses sensors along the track to locate trains in tunnels between stations. This information is displayed to commuters via signs over the tracks at each station. At any time, commuters can see the minutes to next train. However, one drawback of this system is that users cannot view transit delays before arriving at the station. With subway trains, it is not as big of an issue since trains arrive every few minutes. In the station where this was observed, multiple trains came on the same platform, so it was important to know which train was coming next, and that information displayed by an 'arriving in station' status.

The market for transit information is still growing even beyond live vehicle tracking. Google Transit is a service offered by Google that overlays maps and directions with alternative public transportation information. It currently covers eighteen cities in the United States in addition to regional rail, ferry, and domestic air in Japan (A screenshot demonstrating Google Transit for Portland, Oregon is presented in Figure 1.1). Unlike other transit applications, the difference agencies submit routes, schedules, and fares to Google Transit, allowing users to create custom trips that can be ranked by cost, arrival time, or departure time, and even span multiple municipalities and transit companies.

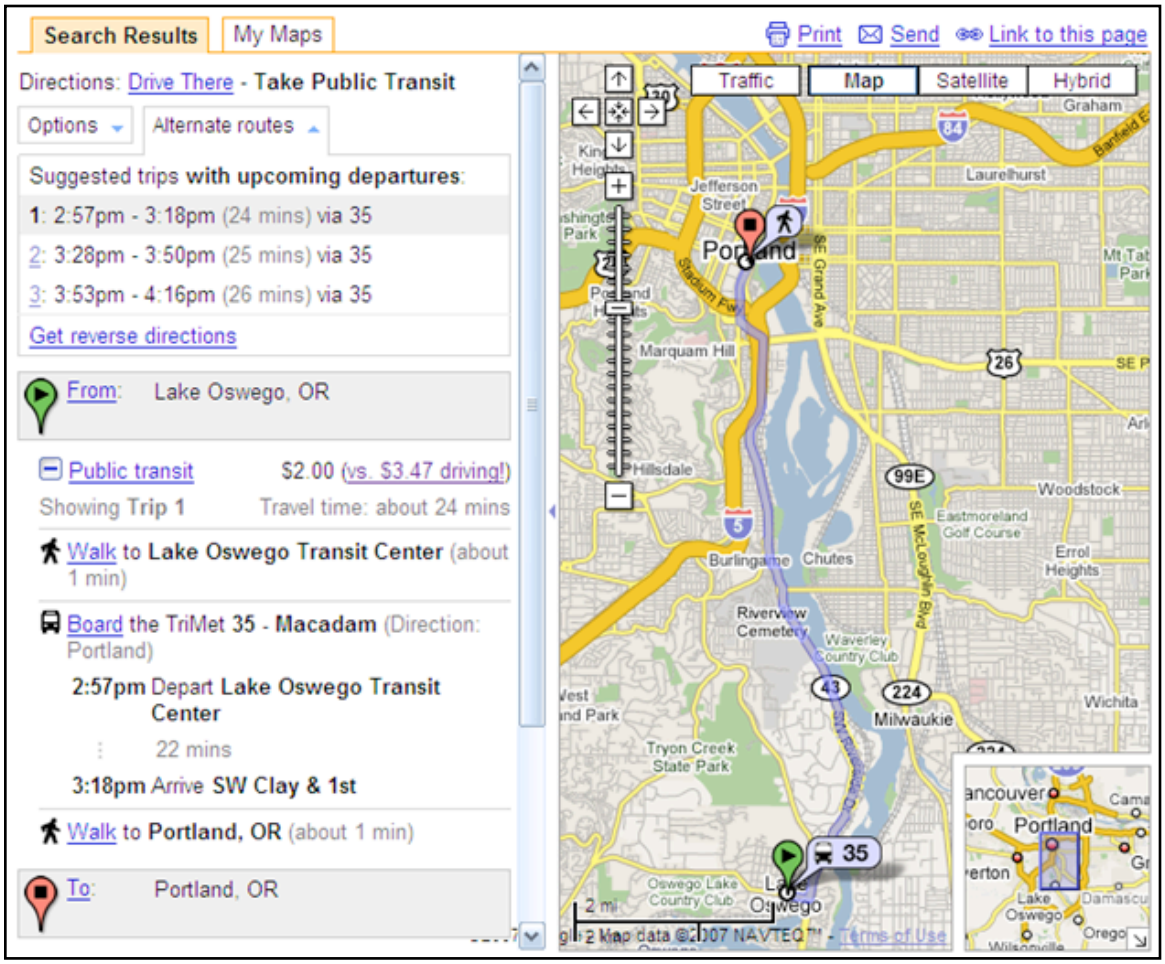


Figure 1.1: Google Transit Sample Screenshot of Portland, Oregon.

In the fall of 2006, the local transit agency in Wilmington, North Carolina, Wave Transit, installed a system for vehicle asset tracking. When approached during the early stages of this project about offering tracking information to the public, they expressed strong interest. The plan included using data from Wave Transit's location server to display bus information to the public through a collection of online dynamic mediums described in later chapters. Estimated arrival time (ETA) was something not available natively on the location server but would be useful to provide on the interfaces. When considering methods for calculating ETA, there are additional factors besides speed limit, average speed, and distance to consider. For the route itself, the wait time at all

preceding stops and traffic along the route all play a major role in arrival time. However, there are no sensors in place to capture traffic flow for all of the roads where the bus travels. These additional factors, among others, must be considered when predicting an accurate estimated arrival time.

One such route that displayed large discrepancies was the campus loop shuttle, running in a continuous loop around the campus of the University of North Carolina Wilmington. It was an ideal route for conducting experiments because it was nearby and the environment could be both predicted and observed throughout the day. While observing this route, multiple factors had a direct impact on the transit time. Being an on-campus only route allowed for a controlled setting to experiment with these differing factors.

1.1 Thesis Statement

In transportation applications, predictions of estimated time of arrival (ETA) are calculated using the average speed of the vehicle, the distance to the location, and the scheduled arrival time. The vehicle observed in this experiment, which continually loops on a twenty-minute schedule, rarely uses all twenty minutes to traverse the trip. Instead of waiting for the schedule to catch up during the route or when returning to the start, it may simply continue. With this precondition, the vehicle location and its consistency with the schedule will fluctuate during the day. For a route that often disregards the schedule, vehicle arrival times can be improved from average speed by analyzing historical data such as time of day, day of week, weather conditions, and transit operator.

1.2 Thesis Overview

This thesis addresses the problem of calculating an accurate estimate of the arrival time of a vehicle where the arrival time fluctuates due to various factors. Typically, a schedule would help keep the vehicle on time, but in most instances on this loop route, the schedule is ignored. This thesis presents a solution using historical data as a means for predicting when the vehicle will be in different locations under certain conditions. As an extension of this work, several sub-systems were developed using the same transit data to deliver real-time transit information to riders by various means. The thesis is organized as follows:

Chapter 2 presents a system overview, outlining the various technologies and data sources used for transit vehicle tracking. It also discusses how these components differ from those in the system used in this experiment. Finally, the chapter introduces some terminology to help the reader better understand some general terms and transportation concepts used in this thesis.

Chapter 3 presents and contrasts related research in this field, including vehicle communication systems, prediction methods, and information delivery options to get the resulting information to the users.

Chapter 4 introduces the transit vehicle predicted arrival time problem at hand, detailing the layout of the route used for this experiment and additional justification behind the selection of this route. It then discusses the steps needed to create a working data set for analysis. Next, the chapter overviews the data mining process and a series of small tools for conducting a complete and statistically relevant data analysis with the aim

to improve predicted arrival time. It then presents methods for delivering the information to users and the methodologies behind which ones were chosen and why.

Chapter 5 takes the experiments and procedures that were outlined in Chapter 4 and presents the primary findings and results. It also discusses which methods had the most effect on arrival time. The chapter then proposes an approach for predicting the arrival time.

Finally, Chapter 6 contains a summary of the work, including some directions for possible future research.

Chapter 2

System Overview

In order to better understand the concepts used in this thesis, terminology as related to vehicle location and transportation systems is presented here. This chapter lays out the technologies and details of generalized tracking systems and integration procedures for the experiment and then presents the specific methods used in this research.

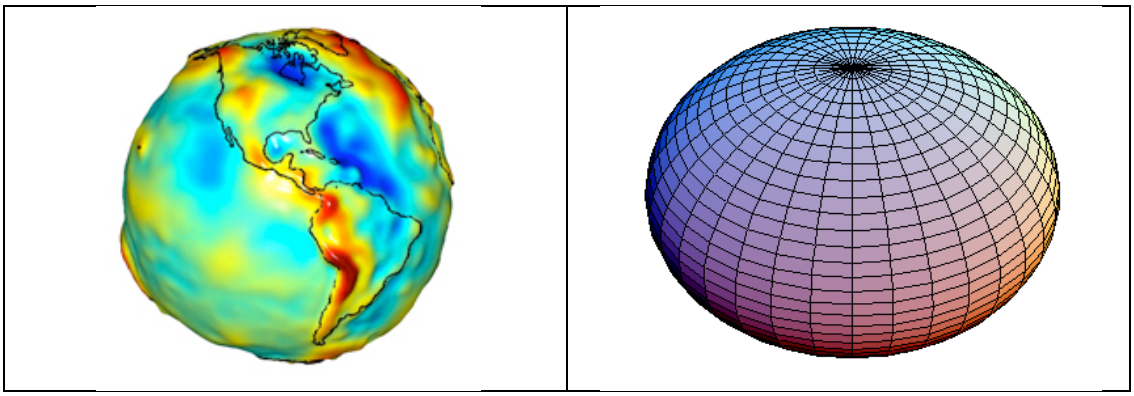
2.1 Terminology

This section reviews terminology often used in transit applications.

- **Automatic Vehicle Location (AVL)** : a system for locating and tracking a fleet of vehicles, typically used in conjunction with a backend system for archiving status reports and alerting the company operator to vehicle problems.
- **Geiod** : a model of the Earth that has an equipotential surface that is the average elevation of Earth's surface over a certain area. Models are compounded using different sources such as average elevation from core and average gravitational force. A three dimensional model of a geiod is shown in Figure 2.1.
- **Reference Ellipsoid** : a model of the Earth similar to an oblate spheroid with two different axes, used as a more mathematically simplistic Earth model than a geiod (Figure 2.2).
- **Scheduled Stop** : in general transportation terms, a bus stop location where a

vehicle will stop that appears published with departure time and a location name paired with geodetic coordinates, also known as a toponym.

- **Flag Stop** : a bus stop that is indicated by a waiting patron or verbally requested by a passenger along the route. Flag stops might not be marked along the route or may be identified by a solitary route-numbered sign without a bus shelter; in absence of a sign, a landmark or driveway is often sufficient.



Figures 2.1 and 2.2: Geiod (left) and Reference Ellipsoid (right) Earth Models

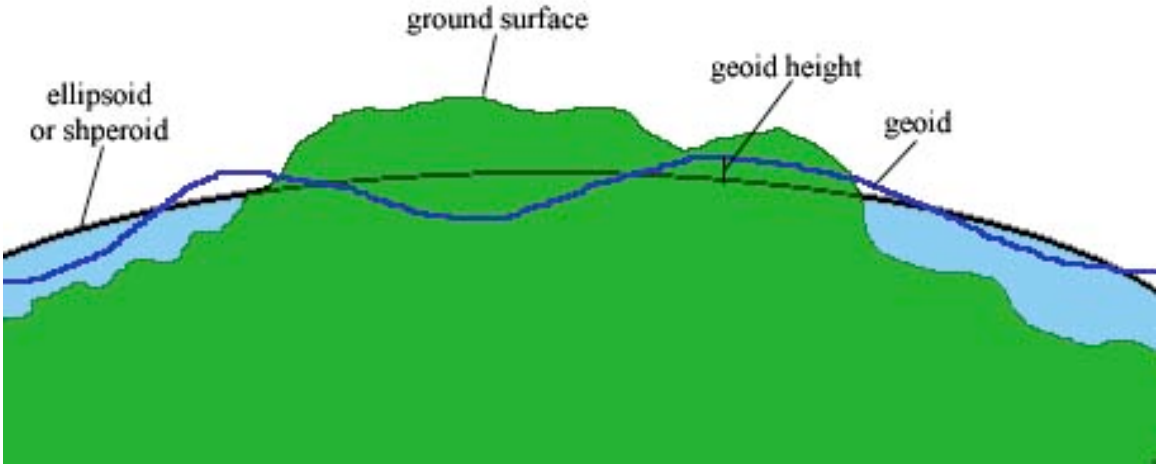


Figure 2.3: Contrasting Lines of Geiod vs Reference Ellipsoid vs Actual Earth Surface

Figure 2.3 shows these two models of the Earth together in one diagram. The green solid area is the actual terrain outline as it curves with the Earth's surface. Like the green, the light blue demonstrates water. There are also two lines, blue and black, which represent the two Earth models, geoid and ellipsoid respectively.

2.2 Automatic Vehicle Location Model

A generic Automated Vehicle Location (AVL) system relays information from transportation vehicles to a central data center. The general AVL specification neither defines the means required for this transaction of information nor the means at which the vehicle location is determined. For this experiment, each vehicle is equipped with Global Positioning System (GPS) devices that report locations every thirty seconds over a cellular network to a virtual transit operator in the transportation company data center.

Figure 2.4 outlines this target system, beginning with the busses gathering their location information and reporting it through the cellular network over a General Packet Radio Service (GPRS) data connection to the AVL server. The hardware for gathering and reporting location information was purchased from Digital Recorders, which also authored the database used by the transit agency. When riders request information, a query is sent from the local web server to the AVL server. Upon every user inquiry, an additional query is sent to the AVL server for the specific information requested by the user. The response from the AVL server is parsed to fit the medium on which the request was sent, whether it be a cell phone, text message, or desktop computer. Every thirty seconds, each vehicle send new location updates so the incoming user inquiries always return the most recent data.

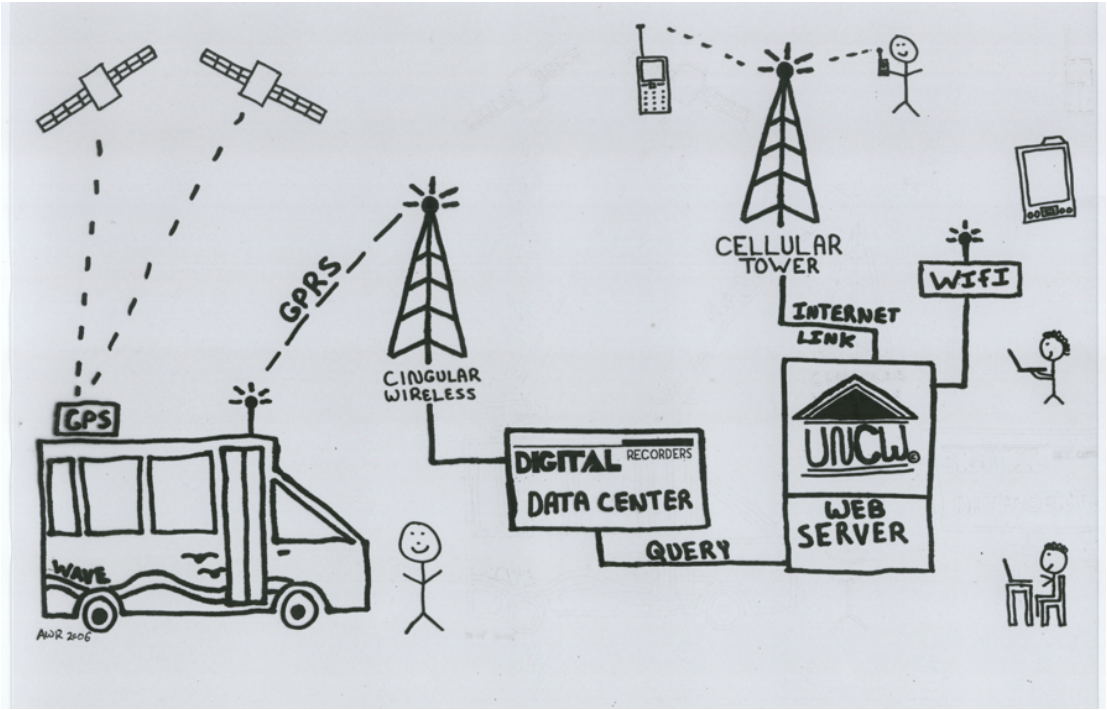


Figure 2.4: Overall (AVL and Client-Side) System Diagram

Global positioning technology via satellite is not new. GPS became available for consumers and commercial uses in recent years due to the federal disabling of selective availability in 2000 under President Bill Clinton. Selective Availability introduces random errors of up to three hundred feet such that only coded military devices receive an accurate location fix. GPS works by calculating the delay when pinging multiple satellites containing precise clocks with assigned orbits and altitudes. By calculating the ping delay from multiple satellites, devices can determine their locations on the Earth's surface. One of the newer technologies involving GPS details collecting various sources from around the Internet that are tagged with GPS data and plotting them on a map. Before Google Maps and Google Earth, custom-mapping solutions needed to grab the mapping or satellite images manually from the National Geological Survey website. One such application, a pre-Google Earth plotting application that uses a flat map for plotting data, is outlined in [5]. They created an application that could be fed a series of data

points to be plotted on a map or even used to plot a route with connecting lines. A few years later, Yahoo now provides a similar web service that returns a map image with a plotted data point with a given distance radius to determine the zoom level of the map.

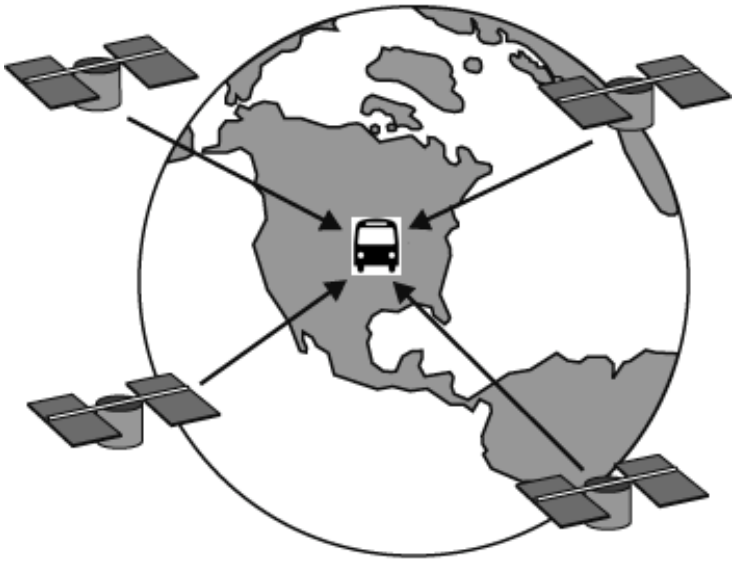


Figure 2.5: Global Positioning System

Since the Earth is not spherical like it is often perceived, but rather ellipsoidal, a reference ellipsoid model of the Earth is used. Although a geoid is a more exact model of the Earth, the calculative complexity is beyond the scope of this thesis. In addition, the primary location for this study is relatively flat, near sea level, and covers a small geographic area of less than four square miles. However, sea level is not the mean altitude of the Earth from the center, which is why the geoid model is significant.

2.3 Predictor System Model

The predictor system developed by this research is piggy backed with a system for displaying the information reported in the AVL database to users that visit the transit

operator's website. To display this information to the users, AVL reports are combined with descriptive data about the route and vehicle operator. When users request information on the client website, the last location update is presented to the user on a web page as human-readable information about the location. Not only for web pages, the web filter parses the information into a generic form that can be translated to many different delivery mediums. When new updates are reported, the previous update is archived in the AVL database for future analysis and the web page is then refreshed with the updated information. The specifics of these transactions are detailed in Chapter 4.

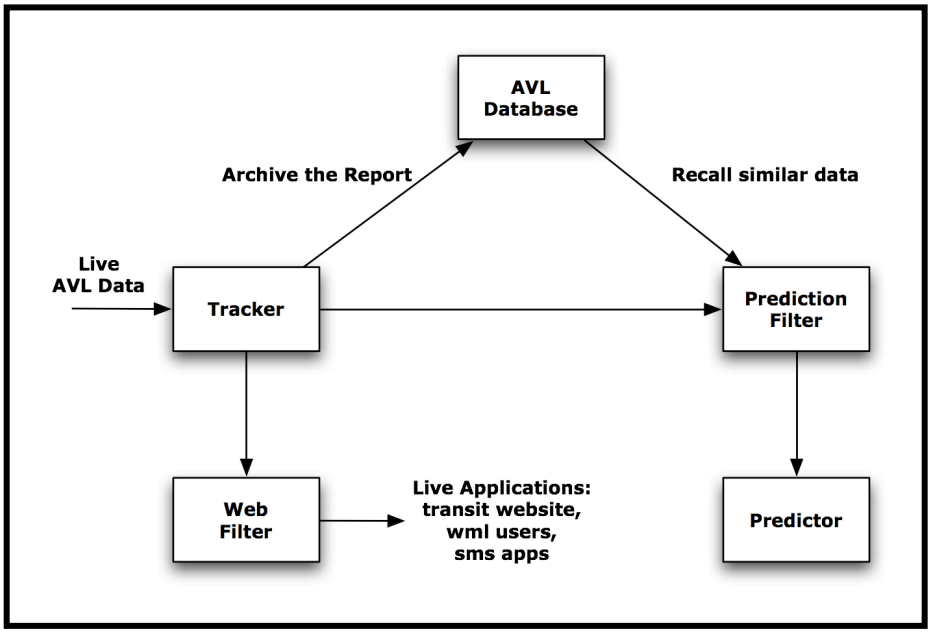


Figure 2.6: System Overview Workflow

The live update then proceeds onward to be compared to historical data and to make a prediction. In reference to the system overview workflow in Figure 2.6, the Prediction Filter searches for data similar to the current report to determine important properties of the current transit report. It also attempts to determine the distance into the

route, previous stops, and future stops to give the Predictor a starting point on which to base its prediction.

The Predictor will use historical data from transit vehicles gathered over some period of time, including factors such as time of day, day of week, weather conditions, and bus driver to determine what has the greatest effect on arrival time. Given a future trip with these known factors, the system will predict an estimated trip model on which the vehicle should follow.

However, if these extra factors are not enough to predict the route with a certain level of confidence, the first few points of the pattern could be considered to gain a baseline for the median and average speeds as well as an estimate of the percentage of time the vehicle is stopped for those conditions, if they are needed. This approach delays the predictions, waiting for points to accumulate on the current pattern before the algorithm can execute. In the event there is still not enough data to make a prediction, factors will be eliminated until an appropriate estimation can be made or a fixed average speed value will be used that best represents a typical route traversal.

2.4 Campus Loop Route

The campus loop route contains a total of eleven stops and is identified to riders as the loop shuttle or the purple route. The other university-associated routes are red, red express, green, green express, yellow, and blue. The express routes run back and forth to off-campus park-and-ride lots while the non-express routes loop through nearby neighborhoods. Students who live within one mile of campus are not permitted to park on campus, so these routes provide transportation to and from campus. In this instance,

the loop route is unique in that it stays on campus for the entire route, not needing to cross major roads or deal with traffic outside the campus. The total distance of the route is just less than 20,000 feet. A campus map representing this route is shown in Figure 2.7, followed by a list of the stops along the route, beginning with the first stop in the lower left hand corner of the map figure and going clockwise around campus.

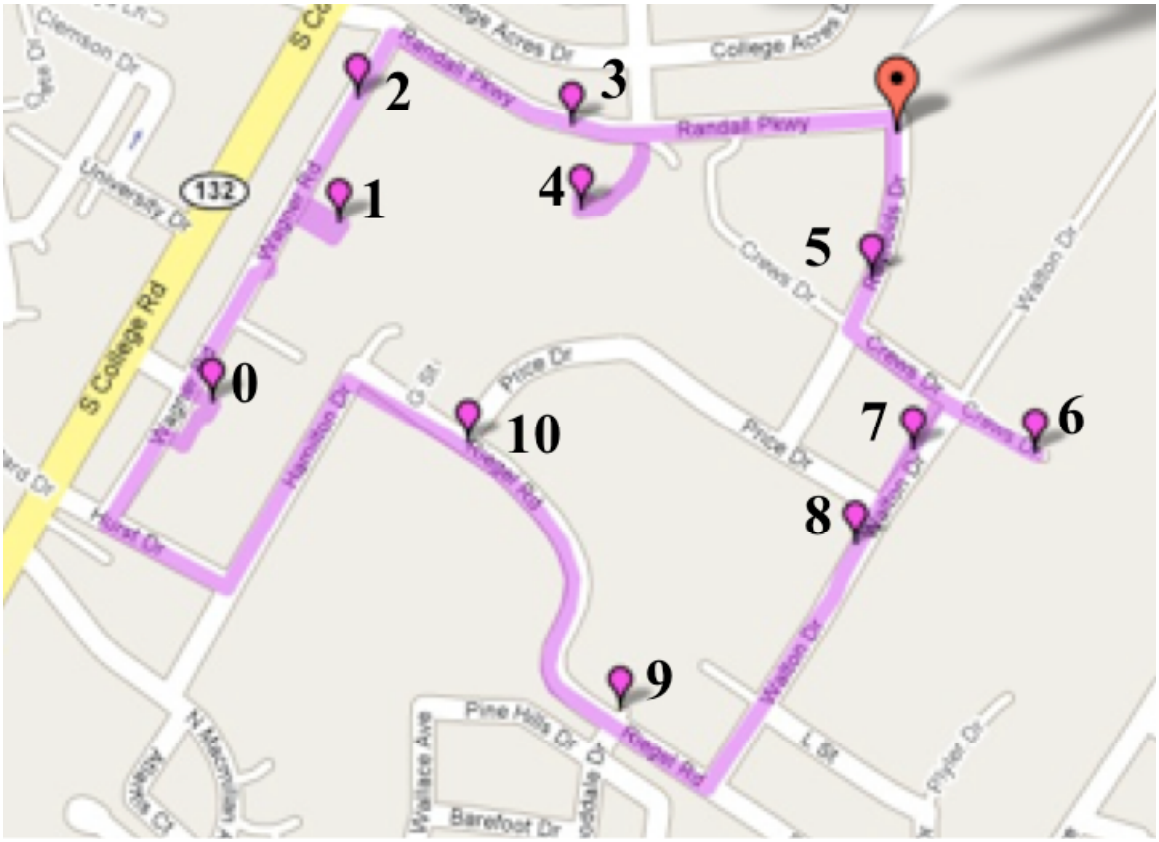


Figure 2.7: Campus Map with Highlighted Loop Shuttle Route

0. Trask Coliseum
1. Parking Lot B
2. Kenan Auditorium
3. Bear Hall
4. Morton / Leutze Halls
5. School of Education
6. Seahawk Village
7. Walton Drive
8. Student Recreation Center
9. Graham-Hewlett Halls
10. Reigel Road / Price Drive

For the campus loop route, several scheduled stops serve as flag stops, such that if no one is waiting and in the absence of a verbal or pull-chain request, the vehicle will not stop. The cause behind this is the lack of pull-offs for buses along campus roads. With

no place to pull off, stopping for an extended time to wait for riders would cause traffic buildup behind the vehicle. Not knowing if or for how long the vehicle will stop at each of these flag stops limits what the system can predict about future vehicle locations, since stops are often passed without hesitation. An allowance for stop time is built into the schedule; as a result the vehicle often arrives early. At the end of the route, the vehicle may or may not stop to let the schedule catch up. For a prediction late in the day, it may be difficult to tell whether the vehicle is early, late, or on schedule.

Chapter 3

Related Work

The technology that allows transit vehicles to gather and report bus locations to a database server and make a schedule-based prediction is a necessary component of this research. First, it incorporates the software for reading the most recent database input and discerning where the vehicle is along the route. Second, it attempts to make predictions using the information fed in by the vehicle's most recent report. Last, it takes the resulting prediction information and displays it to riders through various innovative means. This chapter overviews a sampling of the related work in these areas and compares these methods to the problems of this thesis.

3.1 Vehicle Location Systems

Determining the next stop from previous stops, calculating an estimated arrival time (ETA), and parallel structure displaying information to users are all ways vehicle location software contributes to the complete AVL system. There are several patents and other reports of how to best handle the software side of AVL technology as well as outlining a complete AVL system in hardware. This section describes some of these approaches.

One of the better-known companies deploying AVL technology is NextBus, based in Emeryville, California. NextBus strives to deliver real-time information to riders by displaying wait times at key stops, announcing the next stop on vehicles, and

publishing information to their website about vehicle location and arrival time. They have published a number of patents for both their hardware on the vehicles and the software that runs the backend. The solution patented by [7] details aspects of the entire system from gathering location information on the vehicle to displaying it to users at requested intervals. They illustrate the means by which GPS input is transferred over a cellular network to the host machine. The host machine then files the report based on aspects of the previous time point, transit trip, block of transit trips for the day. In addition, this is extended in [13] by further defining what information can be displayed to the users on head signs and through other output devices.

The primary difference between this research and the NextBus technology is the motivation behind installing an AVL system. The goal of NextBus is to provide information to riders in hopes of reducing wait times and thus making the public transit system more desirable as a means of transportation. The goal for this project, however, focuses on reporting schedule deviations to the transit operator. This purpose for installing AVL equipment was to reduce the fines incurred when vehicles arrive late. It turns out that having big brother looking over the shoulder of the driver has a positive effect on the punctuality of transit vehicles! At the beginning of this thesis project, there was no system in place for notifying riders; it has been developed as part of this research.

Once an AVL system is in place, archived data reports provide reference to how the system can be improved from a management and performance standpoint. Such a method is introduced in [16] by identifying key performance measures from collected AVL data as a cost-effective method for service analysis. The traditional method is to observe transit operations first-hand with transit auditors at random intervals over a week.

When using archived AVL data, all past locations and schedule deviations are recorded in the database, so the archived data has a higher value since it incorporates all historical points, not just when the auditor was on the vehicle. Historic AVL data play a key part in this research, but are not intended to improve the transportation system through route and urban planning. Instead, the data are used as the basis for predictions on how much deviation to expect from what the transportation company advertises.

3.2 Prediction Methods

Knowing when a vehicle will reach a destination is a common problem, even ourtside the scope of public transportation. With such a common problem, there are several useable solutions. This section overviews some of the different prediction methods discovered in research for this thesis, but it is not a complete or comprehensive list.

3.2.1 Kalman Filtering

A common method used in transit predictions is Kalman Filtering, a recursive algorithmic solution to the discrete data linear filtering problem. Furthermore, it is a recursive filter for estimating a dynamic system state given a set of noisy parameters. It supports estimations of past, present, and future states, even if the exact model of the system is unknown. For an example of Kalman Filtering applied to transit technology, see [14].

3.2.2 Artificial Neural Networks

Another extension of using historical data to make predictions is through the use of artificial neural networks (ANNs). It is accomplished by feeding the ANN with historical data of past vehicle journeys with the current journey to determine an arrival time. Using ANNs promotes that an algorithm can learn and adapt to inputs to make better decisions with new data. One such implementation of an ANN in public transit is detailed by [3] which presents that arrival time can be improved with ANNs when using the microscopic simulation model CORSIM over ones without integration of the adaptive algorithm.

3.2.3 Hybrid Methods

The two methods mentioned above are accurate methods for determining an estimate arrival time given varying degrees of data. However, in certain cases, the best approach might be a hybrid of more than one usable subcomponent. The related research in this area is detailed below.

Along the same lines of this project, a sample tracking system is introduced in [18] with several means for creating a prediction. For their experiment, they are working with an industrial partner and have control over all aspects of the AVL system. In addition, the partner is striving to maximize results while minimizing the cost. These are two key parameters that are outside the scope of this thesis; the methods for data collection and the interval at which it is collected can be neither controlled nor specified in this project. While the paper mentions the option to draw on historical vehicle

journeys, it does not try to identify key aspects of those historical journeys that might weight a decision.

3.3 Information Delivery

The last step when generating a prediction is to convey the information to the user. This can be accomplished in several ways including notification along the route, verbal alarms on the vehicles, interactive web-based information, on-demand mobile info delivery and scheduled notifications. While some of these methods are standard for transit operations, a few are innovative means for delivering real-time updates. This section overviews some of the approaches and compares them to the methods deployed in this research, focusing on the emerging technology of mobile devices.

3.3.1 Mobile Web Pages

The most common new technology for relaying information to users on-the-go are mobile devices. These are commonly small hand-held computers with data or text messaging capabilities, which can be personal digital assistants (PDAs) or cellular telephones. Text messaging is a lightweight short messaging service similar to email. Devices with data capabilities often include a lightweight mobile browser that accepts a mobile web format called WML, or wireless markup language. As explained in [8], mobile users can utilize the WML functionality of their devices to navigate a simplified version of the MyBus.org website that is intended for access by a PC browser. They introduce the limitations of mobile devices including input options, limited display size, and text-based information delivery and how it affects the user design.

Since the above was published in 2001, the capabilities on the average mobile device have evolved to include more feature-rich content and the format rules governing WML content have been relaxed to be more like HTML. For this research, three web portals were created in parallel including a bare-bones WML page, a hybrid rich-content web page designed for newer cellular phones, and a HTML page for PC browsers. Unlike in the above research, the user need not navigate to a different page to receive the mobile content, but rather the web server determines the capabilities of the device and sends out the appropriate content.

3.3.2 Mobile Text Messaging

Browsing web pages on mobile devices requires either an active data connection with a cellular provider or a nearby WiFi access point. For the users targeted in this research, only about 1 out of 7 people ever used the mobile browser on their cellular device. The more common information medium used was text messaging, which is very popular with the university students. As demonstrated in [4], text messaging is also an innovative means for delivering transit information. Their research focused on increasing ridership from choice riders, those who had other means of transportation, and those who used public transportation as their only means of commuting. Information signs were placed at each stop identifying how to interact with the text messaging system. Users would send a code identifying the route and stop to a designated number, which would respond with an estimated arrival time of the next two vehicles.

The projects described above highlight different methods for delivering transportation content to mobile users. When it comes to mobile phones, the two primary

areas of content delivery are covered, mobile web pages and text messaging. However, text messaging can be further divided into auto-responding services and push services. The above example is an auto-responding system while scheduling notification is an example of a push service. For information delivery, this thesis attempts to address all possibilities, hoping the users will select the method that best meets their needs. Text messages can also be scheduled to alert users that a bus is near a stop at a user-defined day and time or a set number of minutes from a stop. This is likely the preferred method for university students as it can serve as an alarm to begin walking to the bus stop.

Chapter 4

Closed-Loop Vehicle Location Predictor

In Chapter 2, AVL technology was introduced and different related research was presented as background and supporting information for Chapter 3. The notions behind which route was selected for this experiment were also mentioned in Section 2.4, but this chapter will go into more detail about why the loop route is ideal for this research, how the data was collected and used, and the backend tools created for both prediction analysis and user information interfaces.

4.1 Overview

This research assumes three major components are in place:

1. There is a transit vehicle that travels along a predicted route.
2. There is an automatic vehicle location (AVL) system in place, where the vehicle periodically reports its progress along the route to a transit database.
3. The transit database contains several traversals of the route for analysis and algorithm computation.

In addition to these general requirements, aspects of the transit pattern chosen for this study are further detailed by the following characteristics.

- The pattern is a closed loop such that the stop following the last stop in the pattern is also the first stop of the subsequent pattern. This is opposed to a pattern where

the vehicle backtracks over previous time points in reverse order, more commonly known as an inbound/outbound set.

- Although the entire pattern is in a loop, only the first half of the pattern is used in this experiment. This was chosen because the stop that is half way through the route is a common termination point for riders, giving access to a large percentage of campus housing. It is also a stop that does not disrupt traffic flow, providing space in a looped driveway for the vehicle to stop and wait if needed. Additionally, good data was gathered for this half of the route, which would make finding usable segments easier.
- The segment used contains six stops, and is scheduled to take ten minutes to traverse. The first, fourth, and sixth stop of the pattern are stops that do not inhibit traffic flow while stops two, three, and five are along a road with no pull-off space. Reference Figure 2.7 for graphical representation of these stops.

4.2 Live Data Logging

Since the AVL system used for this project was created as an auditing tool for the transit operator, it did not log vehicle telemetry data for every report if there was not an event to cause alarm. Therefore, in order to capture this information, a listener needed to be created to grab each update as it reported. The AVL system did have a table containing the last update for each vehicle with specified route. Since it was known that updates came in every thirty seconds, the listener was tuned to grab the latest information, sleep for thirty seconds minus execution time, then request the next location. Initially, only a small subset of data was gathered in each update, but as the project

progressed, more information was determined relevant, so it was integrated into the captured set. For each logging upgrade, a patch was written to re-generate the additional information if it could be from the previous versions. The log iterations are outlined below by version.

Version:

1. The initial logging application captured latitude, longitude, heading, instantaneous speed, next stop (according to the AVL system), and report timestamp.
2. The major change was the addition of average speed estimation. While logging in version one, work was done to represent the route in a digital form, detailed later in this chapter. Having a representation of the route that could be understood by the computer enabled it to know where the vehicle was in the route and make blind predictions based on average speed and distance to next stop. When reading log entries from version two, the latitude and longitude itself is not easily human understandable. However, stating that the vehicle is twenty-eight seconds away from Stop 3 is a better indication of where the vehicle is located in the route.
3. The AVL system from which this information is gathered was relatively new to the transit agency when they granted this project access to the data. With that in mind, it was commonplace for minimal downtime and small upgrades. One such upgrade was the addition of vehicle operator to the live reports. With this upgrade, the log also now incorporated vehicle operator. On an operational standpoint, the log did not handle AVL system downtime well, so additional failsafe lines were included in the logging system to handle an unresponsive AVL server and sleep until it came online again.

4. Along with small improvements of the logging application for each new version, the fourth version incorporated a weather listener. In addition to gathering location and telemetry data, the log now fetched weather information on the hour, including temperature, humidity, wind, and sky conditions from the local international airport a few miles away from campus. The county airport located about five miles north served as a reliable data source for weather updates. They also made available a historical data archive, so past AVL entries could be matched with weather information from previous versions.
5. The targeted AVL system now keeps its own logs of all pertinent details for location, operator, and telemetry data for vehicles running on all routes. Therefore, a logging listener is no longer needed to grab the information as the AVL server receives every update twice per minute. The logging application was modified to grab all the information in the AVL server's log from the last local log entry to the end of the current day. This information is then supplemented with archived weather data and is added to the local logging database.
6. There are no functional changes in this logging version but the version flag is an easy way to distinguish data sets. Using this flag, all data gathered using version five and previous will be used for generating and fine-tuning the prediction algorithm. All data from version six and higher is used for testing.

Over 110,000 log points make up the data in logging versions one through five. A typical day with no service interruptions produces about thirteen hundred points. The log contains enough data that given any coordinate pair, there exists at least one other coordinate pair that is either identical or within one foot. When all data points are plotted

on a map, the average distance between points is less than one foot for the four-mile route. A sample shot all log points plotted on a map is shown in Figure 4.1 on a road where the bus only travels from right to left. This figure demonstrates just how many points were collected for a segment of road and how the points vary in location in relation to the road, which is common in commercial-grade global positioning devices where some error is expected.

Figure 4.1: Log Points on Map Showing Abundance of Points Gathered.

4.3 Vehicle Tracking

This section overviews components developed for ease in understanding the routes from a computer perspective. It details the underlying steps taken to enable the computer to make accurate predictions of the route and components that provide content delivery to users. These are the support structures that enable the system to operate smoothly while supporting the prediction algorithm.

4.3.1 Handling Geodetic Data

In all applications, two decimal numbers, a latitude and longitude, are the minimalistic way to describe location data. A coordinate class serves as a container for these geodetic coordinates, basic telemetry data, and a textual description. The distance formula, explained later, took coordinate objects as parameters, which would otherwise be four arguments instead of two. The text can be anything that the programmer wants to bundle with the object. In addition to the distance formula, two useful functions are contained in this class. The first function translates decimal heading degrees into cardinal directions. For example, the AVL database returns that the vehicle is heading 229 degrees at 14 mph. Parsed, the value displayed in the user application reads, SW at 14MPH, which is useful when displaying information to the public. The second is a function to test if a direction is within ±45 degrees of another, thus heading approximately the same direction with an allowance for some variation. For determining which plot point relates to which live point, the approximate vehicle direction excludes directions going the opposite way, thus on the other side of the road.

4.3.2 Reverse Geocoding

Geocoding is a method of determining geodetic coordinates for streets and street addresses. Reverse geocoding is simply the opposite: assigning street addresses or toponyms to geodetic coordinates. Often in a reverse geocode, additional, more generalized information is presented such as city, state, zip code, telephone area code, etc. The US Census Bureau maintains the Topologically Integrated Geographic Encoding and Referencing (TIGER) system, containing entries for nearly every street and

highway in the United States. The campus of UNCW is included in this database but recent E911 road name changes and new road cuts are either erroneous or not included. Using plot locations from transit vehicle trips, known stop locations, known landmarks, survey monuments, and a CAD map of campus, polygons with geodetic tags were created outlining roads, parking lots, and 4-way intersections. For ease of creation, Google Earth was used to create the polygons that were exported through XML to a custom tool created to resolve geodetic reports to roads, intersections, and parking lot identifiers. Now, not only are the latitude, longitude and heading of a report known, but also the paved location on campus for the particular report. This additional information aided in human recognition of the vehicle location.

4.3.3 Digitizing the Transit Pattern

Knowing on which paved part of campus the vehicle is located aids in determining location, but now the vehicle's location must be defined further, by placing it at a certain location in the transit pattern to determine the next stop and distance to that stop. This is achieved in a similar method to the reverse geocoder using polygons to represent the area where a vehicle can be along the route. However, experiments show that for cataloging the pattern and determining distance, using evenly spaced points along the road with a directional heading made distance calculations easier. This is especially important when multiple vehicles use the same route since the GPS devices may vary slightly in the reported locations. If polygons were used for individual sides of the road, even with direction, a report that puts a vehicle over the centerline might be attributed to

a faulty report and not be recorded. Using a series of plots reduces this and other small rare inconsistencies.

Over the four-mile route, road plots were gathered about every 25 feet, producing a vector list of about 820 points that make up the route data structure. The list is then further defined by which locations lead up to which stops and which locations are on which roads, as defined by the reverse geocoder tool. With the plot locations every twenty-five feet, an incoming report is matched to the closest plot location using a distance calculation, described below. Since the road does not always go one direction, the algorithm is unable to determine if the vehicle reported is in front of or behind the nearest point in the road plots. To determine the relative location of a report to the plot points, a method would need to take into account the direction of the road at that location and the cardinal direction that the report is in relation to the plot point. The cardinal direction is computed by subtracting the two sets of latitude and longitude to make a triangle, and then compute the angle relative to north. However, this method is trumped by using plots that are twenty-five feet apart, minimizing the need to know where the report is in relation to the road plot; a variation of twenty-five feet, or about one bus length, is negligible.

4.3.4 Calculating Distance

As mentioned earlier, a reference ellipsoid of the Earth is the chosen means for mathematically representing the surface of the Earth for all distance calculations. If, however, a spherical representation of the Earth were used, the Haversine Distance formula would serve as an approximate means for calculating distance on the surface.

The Haversine Distance formula presented in Figure 4.2 is similar to the Great Circle calculation, finding the shortest distance between two points on a sphere, also known as arc distance, but applied to geodetic coordinates. Since the units of interest are feet, the Earth's radius needs to be represented in feet as well [15].

```
Given two latitude/longitude coordinate pairs, 1 and 2:
Let R = radius of the Earth (in feet)
```

$$dLat = lat2 - lat1$$

$$dLon = lon2 - lon1$$

$$A = \sin\left(\frac{dLat}{2}\right)^2 + \cos(lat1) * \cos(lat2) * \sin\left(\frac{dLon}{2}\right)^2$$

$$Dist = 2 * \arctan\left(\frac{\sqrt{A}}{\sqrt{1-A}}\right) * [Radius]$$

```
Dist is the distance between the two points in feet.
```

Figure 4.2: Haversine Geodetic Distance Formula

The alternative to the Haversine formula is to use the Vincenty Distance formula, presented in Figure 4.3. The Haversine formula is known to have errors of at least 0.3% with exact coordinates. The Vincenty formula utilizes the reference ellipsoid and therefore is a more precise measurement algorithm on for locations on the Earth's surface since it uses both the semi-major and semi-minor axes unlike Haversine, which has only one radial measurement, assuming the Earth is spherical. For arc distances on the surface on an ellipsoid, specifically a geodesic calculation on the Earth's surface, the Vincenty formula is the most accurate known to date. Thaddeus Vincenty (27. October 1920 - 6. March 2002) derived the formula published in 1975 that became essential for the development of GPS. At the cost of a complicated formula, this distance method is well known for its extreme accuracy of half a millimeter (about 0.02 inches) [19].

s is the distance (in the same units as a & b)
$\alpha 1$ is the initial bearing, or forward azimuth $\alpha 2$ is the final bearing (in direction p1→p2)

a, b = major & minor semiaxes of the ellipsoid f = flattening (a−b)/a
$\varphi 1$, $\varphi 2$ = geodetic latitude L = difference in longitude
$U1 = \mathrm{atan}((1-f).\tan\varphi 1)$ (U is 'reduced latitude') $U2 = \mathrm{atan}((1-f).\tan\varphi 2)$
$\lambda = L$, $\lambda' = 2\pi$

while $\mathrm{abs}(\lambda-\lambda') > 10\text{-}12$ { (i.e. 0.06mm)

$$\sin\sigma = \sqrt{[\ (\cos U2.\sin\lambda)^2 + (\cos U1.\sin U2 - \sin U1.\cos U2.\cos\lambda)^2\ ]} \quad (14)$$

$$\cos\sigma = \sin U1.\sin U2 + \cos U1.\cos U2.\cos\lambda \quad (15)$$

$$\sigma = \mathrm{atan2}(\sin\sigma, \cos\sigma) \quad (16)$$

$$\sin\alpha = \cos U1.\cos U2.\sin\lambda / \sin\sigma \quad (17)$$

$\cos^2\alpha = 1 - \sin^2\alpha$ (trig identity; §6)

$$\cos 2\sigma m = \cos\sigma - 2.\sin U1.\sin U2/\cos^2\alpha \quad (18)$$

$$C = f/16.\cos^2\alpha.[4+f.(4-3.\cos^2\alpha)] \quad (10)$$

$$\lambda' = \lambda$$

$$\lambda = L + (1-C).f.\sin\alpha.\{\sigma+C.\sin\sigma.[\cos 2\sigma m+C.\cos\sigma.(-1+2.\cos^2 2\sigma m)]\} \quad (11)$$

}

$$u^2 = \cos^2\alpha.(a^2-b^2)/b^2$$

$$A = 1+u^2/16384.\{4096+u^2.[-768+u^2.(320-175.u^2)]\} \quad (3)$$

$$B = u^2/1024.\{256+u^2.[-128+u^2.(74-47.u^2)]\} \quad (4)$$

$$\Delta\sigma = B.\sin\sigma.\{\cos 2\sigma m+B/4.[\cos\sigma.(-1+2.\cos^2 2\sigma m) - B/6.\cos 2\sigma m.(-3+4.\sin^2\sigma).(-3+4.\cos^2 2\sigma m)]\} \quad (6)$$

$$s = b.A.(\sigma-\Delta\sigma) \quad (19)$$

$$\alpha 1 = \mathrm{atan2}(\cos U2.\sin\lambda, \cos U1.\sin U2 - \sin U1.\cos U2.\cos\lambda) \quad (20)$$

$$\alpha 2 = \mathrm{atan2}(\cos U1.\sin\lambda, -\sin U1.\cos U2 + \cos U1.\sin U2.\cos\lambda) \quad (21)$$

Vincenty observes that line (18) becomes indeterminate over equatorial lines (since $\cos^2\alpha \to 0$); in this case, set $\cos 2\sigma m$ to 0 and the result is computed correctly. Also, the formula may have no solution between two nearly antipodal points.

Figure 4.3 Vincenty Distance Formula [19]

While it may be overkill for this application, it is the best geodetic distance formula and will ensure that distances are as accurate as possible, since distance plays a major role in ETA. It will also develop a basis system if the research extends to a larger scope. The most widely accepted reference ellipsoid model of the Earth is WGS-84, but others are available that offer a better fit to the local geiod. This application uses WGS-84, but the local geiod adjusted model is NAD83, which is functionally equivalent to WGS-84. Visit the National Geodetic Survey website (http://www.ngs.noaa.gov) for additional information. The AVL equipment used resolves geodetic coordinates to six decimal places; at the latitude of Wilmington, North Carolina, that translates to an accuracy of about four inches.

4.3.5 Average Speed Calculation

Once the distance to next stop can be calculated, average speed ETA becomes possible given an average speed value. Although this value is called average speed, it may not truly be the average speed, such that the value is the sum of the speeds divided by N. It is rather a hybrid value comprised of the median and the average speed when the vehicle is moving. Since the vehicle is stopped a large majority of the time, the median is a more accurate assessment of speed, but is still influenced by the average within one standard deviation. Figures 4.4 and 4.5 below demonstrate this data spread. At first, the value was a pure guess to the average speed. Once the data points were logged, statistics like mean and median yielded a better speed value, so the calculation was adjusted. In addition, another result of having an average speed value was to predict the vehicle location five minutes in the future. These future point estimations were calculated using

this average speed, which was fine-tuned to reduce the error to 1000 feet or less. This future prediction method later proved less practical, because there is no technology in place to determine at what time a vehicle is at what location. That would require location sensors. Alternatively, the technology is in place to know the location based on the time, so it became imperative to predict a time given a location rather than a location given a time, since that could be verified using the database. This tool later incorporated the virtual map and distance formula, so given a location along the route, it would determine an ETA in seconds from the last report to a given stop, which might not be the next one.

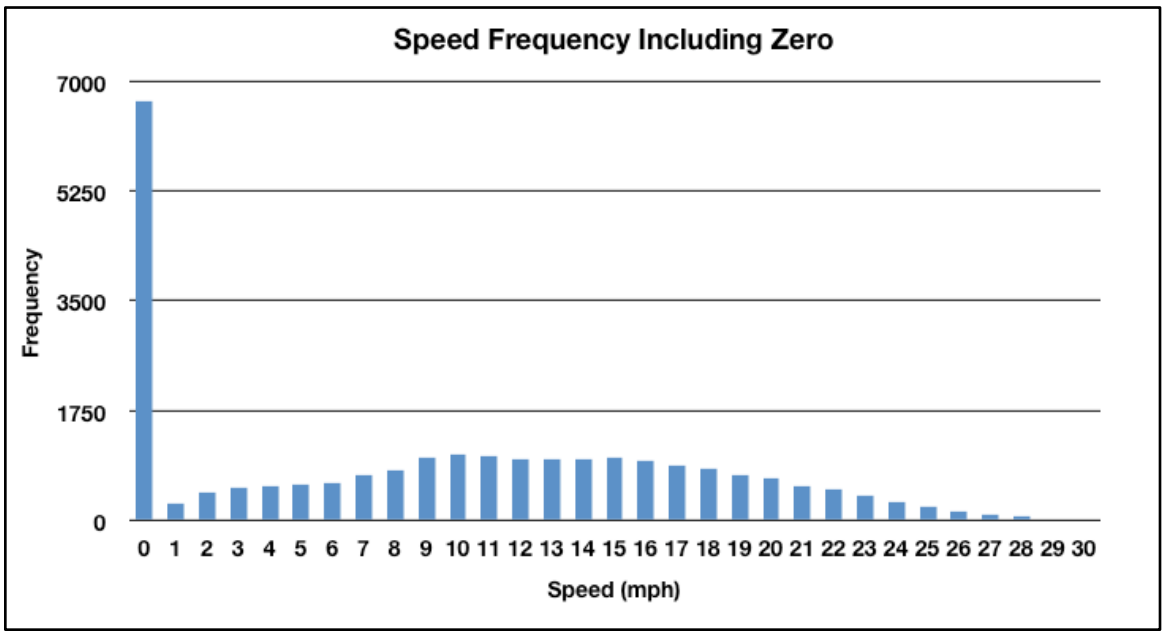


Figure 4.4: Occurrences of Speed Including Zero

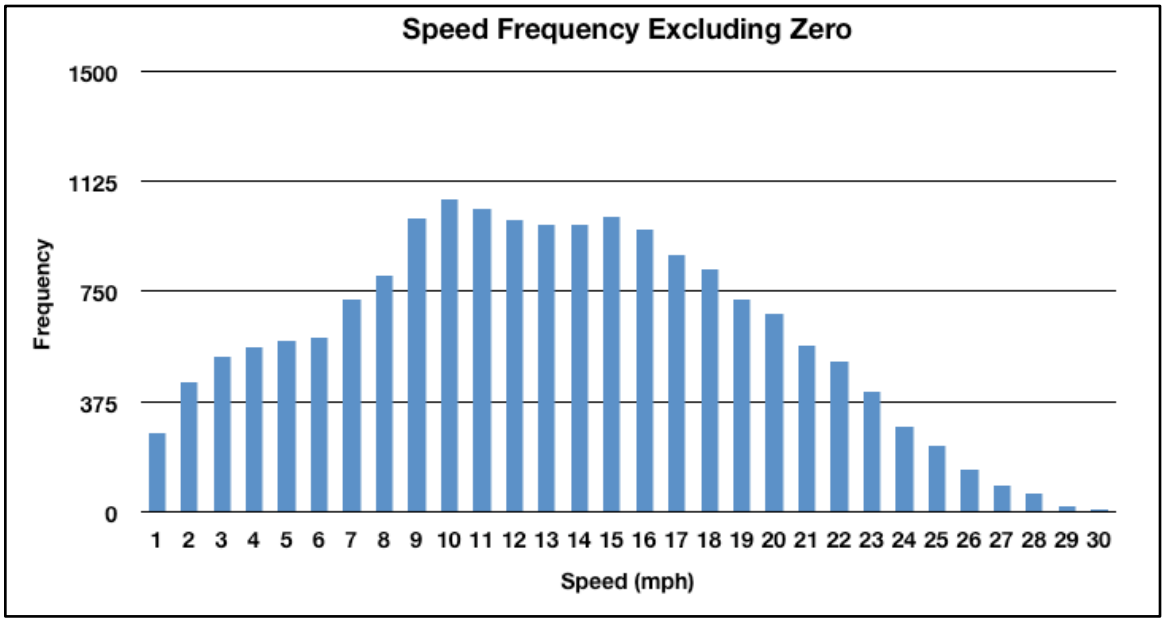


Figure 4.5: Occurrences of Speed Excluding Zero

4.4 Prediction

Now that there are components in place to locate the vehicle in the pattern, the problem of where it will be can be addressed. This is accomplished through several steps of trial and error on finding the acceptable solution. However, it may be the case where one solution only works during certain conditions. This section presents a few tools written to assist in the prediction methods, and the concepts targeted to make the predictions. Analysis on the predictions will follow in more detail in chapter 5.

4.4.1 Filter Usable Segments

Not all the plots gathered in the logging phases will be used. As stated earlier, the project is only concerned with the first half of the route, which is referred as a half route. Targeted segments include the time between the first and last stop of the half route.

The locations of the first and last stops are determined by either:

- Defining a bounding box polygon such that all applicable points must lie inside the box. This method is identical to how the reverse geocoder determines if a point is on a specific road.
- Defining a point such that all applicable points must be at no greater than a maximum defined distance from the reference point.

For determining which points mark the beginning and end of a half route, the first method mentioned is used. Both locations have a stop region, outlined by analyzing the log points to determine where the vehicle stops 95 percent of the time for each stop location. When the criteria are known for the location, it is important to note that for the vehicle to be making a stop, the speed needs to be close to zero. However, since updates come in at timed intervals, it is possible to have a valid half route except for the speed of the vehicle around the stop region. Therefore, applicable speeds are relaxed to within five miles per hour, which would suggest that the vehicle is either slowing down to make a stop or speeding up while departing the stop. The algorithm also checks that a limited time elapsed between start and finish of the half route. Sample pseudo code for finding these usable segments is demonstrated in Figure 4.6.

```
/* The following finds the last valid data point then finds the first valid data point
        Can also be reversed if the application required
        Stoppage into the route was the first constraint
        Stoppage earlier in the same segment was the second constraint  */
dataset = historicalLog.getAllEntries()
while ( dataset.hasMore() )  {
    capture logid, latitude, longitude, datetime, heading, speed from dataset
    if region1.contains( dataset[n].latitude , dataset[n].longitude )  AND speed <= 5 {
        // speed <= 5 indicates was or will be stopped
        // valid ending point
        newlogID = dataset[n].logid - ( some_interval )
        dataset2 = historicalLog.getAllEntries ( logID <= x <= newLogID )
        while ( dataset2.hasMore() ) {
            capture logid, latitude, longitude, datetime, heading, speed from dataset2
            if region2.contains ( dataset2[n].latitude , dataset2[n].longitude )
                AND dataset2[n].speed <= 5 {
                print dataset2[n].logID , dataset[n].logID is valid interval
                print time = dataset[n].datetime - dataset2[n].datetime
            }
        }
    }
}
```

Figure 4.6: Pseudo Code for Finding Qualified Segments

Not only should certain segments be targeted, but also segments that meet specific criterion. These segments might include not only the location in question, but the surrounding points that lead up to and follow the location in question. For instance, in one application it is important to find the route segments when a driver waits three minutes at a specified stop. A route segment filter tool loops through the logging database, locating times that this criterion is met, recording the entire route segment that contains the desired instance. Then the segment is contrasted as a whole against other valid route segments to draw similarities and determine what would cause the long wait. The statistics gathered by this filter can be sent through another filter that prepares the data for creating a time verses distance graph. The distances are determined using the distance function and a virtual map of plot points, both defined previously.

The filter is designed to allow for different types of parameters to be used to limit the output. In addition, multiple levels of constraints can be defined. For example, one experiment needed to find all instances where it was a valid half route, which occurred on

a Wednesday, and the vehicle waited until after the 57th minute but before the 3rd minute of the subsequent hour to depart the first stop, indicating the driver was attempting to sync with the schedule. This functionality is possible with the segment filter. Another application of the filter and the output created is described in the next section.

4.4.2 Sample Set

For one such application of the filter tool, six segments were chosen randomly from the log between February 8th and May 9th 2007 where the vehicle is logged with a near-zero speed at Seahawk Village (Stop 6) and a previous near-zero speed at Trask (Stop 0), taking the first instance at Stop 0 of a speed of near zero. Often times the vehicle waited at Stop 0 before beginning the route. It is important to gather all the points while it is sitting idle, as they will help determine an average wait time to be added on to the actual half route traversal time. With simple statistics of each segment, it was immediately apparent that there was large variation in the transit time for what should be a ten-minute half route.

As an expansion of this analysis, using the filter described above, detailed graphs have been created for applicable segments that fulfill the near-zero speed requirement. The driver graphs are one per driver that has ever driven the route, listing segments by day of week and time of day. The second set contains graphs grouped by time of day for every half hour that the bus is in service; this is further defined as every segment that passes the speed requirements and the half hour increment occurs when the vehicle is between Stops 0 and 6. For instance, if the vehicle leaves Stop 0 at 1031 and arrives at Stop 6 at 1042, the segment is not considered. However, a segment with a 0849

departure and 0901 arrival would be valid for the 0900 graph. A sample graph is shown in Figure 4.7. These graphs are used for analysis, discussed more in Chapter 5.

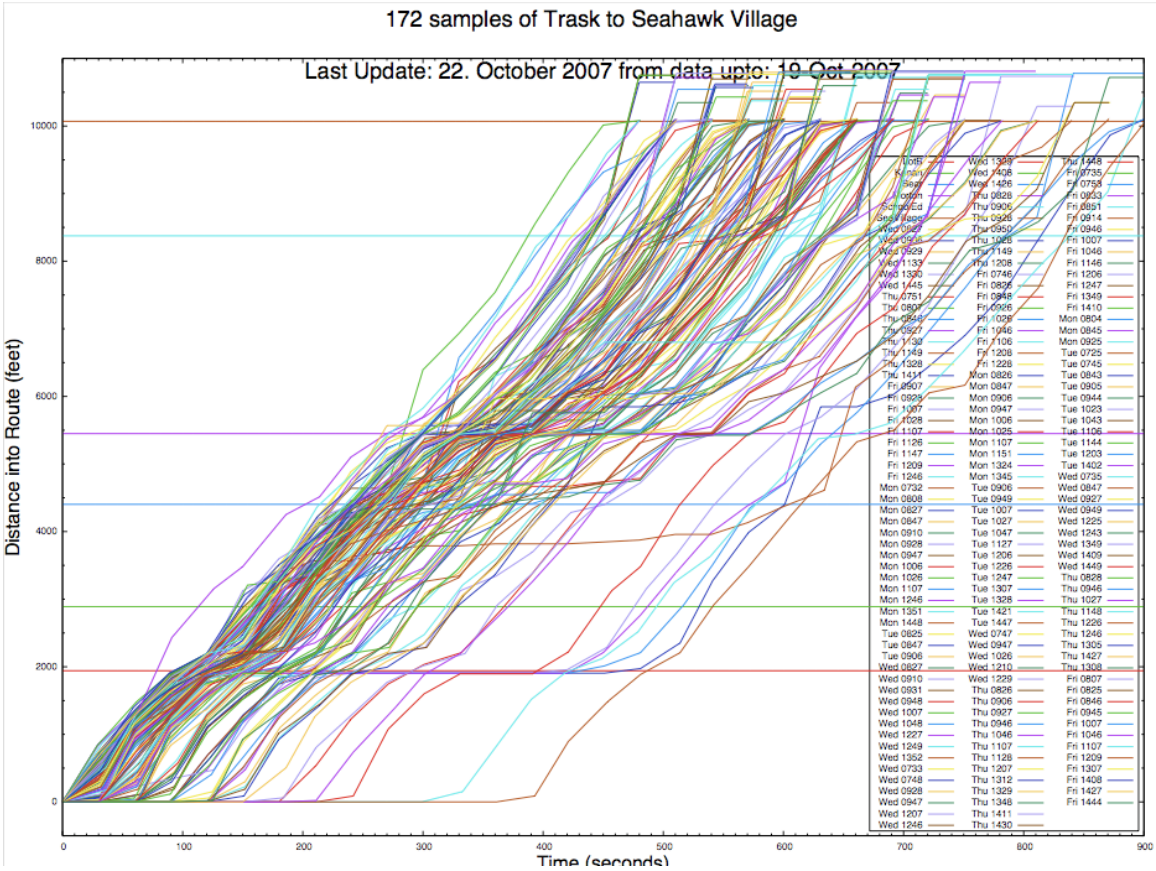


Figure 4.7: Sample Output Graph from Segment Filter

In addition to the graphs, each segment is analyzed statistically. This yields a segment average, median, and standard deviation speed; ETA estimate using this average speed and median speed with percent error from actual; and the actual transit time. Since all these segments are from Stop 0 to Stop 6, the scheduled time is constant and known (ten minutes). To demonstrate, Figure 4.8 shows the sample statistical output one instance on Aug 27, 2007 for the driver whose graph appears in Figure 4.7.

```
Log Points: 84975 -> 84994
Beginning: Monday @ 09:10:23 -> 09:19:54
Weather: Partly Cloudy 78.1 F
Driver: [HIDDEN]
Average Speed: 941.6
Median Speed: 1056
StdDev Speed: 612.28
Time Est w/ Avg Speed: 641.68s (10.69m) [09:21:04], 12.38% Err
Time Est w/ Med Speed: 572.16s (9.54m) [09:19:55], 0.2% Err
Actual Transit Time: 571s (9.52m)
```

Figure 4.8: Sample Segment Filter Statistical Output

4.5 Access to Real-Time Transit Information

This section discusses the different methods for displaying transportation vehicle information to users. These do not include notification along the route or inside the vehicles. Internal vehicle alerts are handled by the native AVL hardware on the vehicles. Dynamic signage is beyond the scope of this research, but is being considered as an external addition to the AVL system by the transit operator. The types of delivery discussed in this section include both mobile device and personal computer delivery and several forms of each. All of the technologies discussed require an Internet or cellular data connection to retrieve status updates.

4.5.1 Interactive Websites

To display the vehicle location and information to users for use on HTML websites, there was a need for a thin web display. A simple webpage is generated given a vehicle number or route name with basic vehicle telemetry data such as direction, speed, and next stop. In addition to this information, the geodetic location is sent to a Yahoo

web service to fetch a portable network graphics (PNG) map image of the requested location, supplemented with an arrow indicating the direction of motion. Yahoo provides this map imaging service, which eliminates the need for a custom solution. This map and information make up the static map page, a sample of which is shown in Figure 4.9. It is called the static map page because the image is static, unlike overlaying the information on a Google Map, as described in the next section.

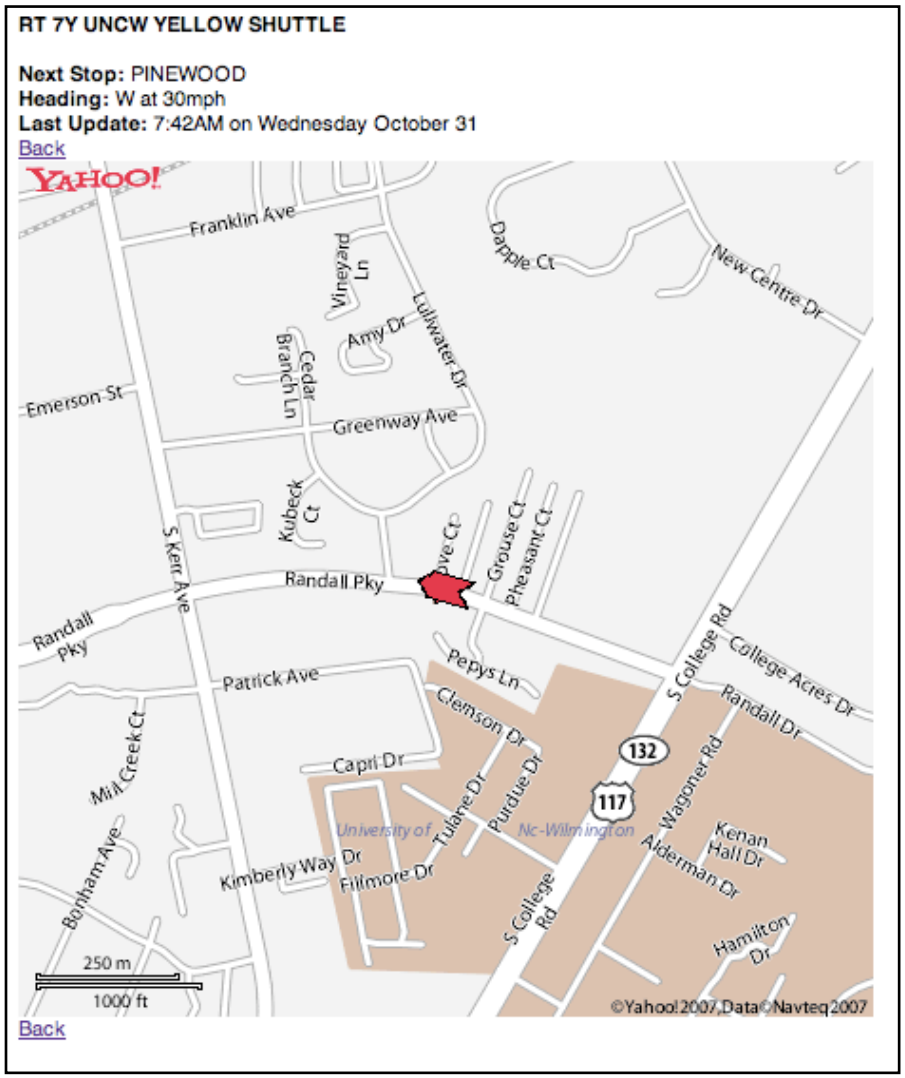


Figure 4.9: Example of Static Maps Web Site with Bus Location and Directional Icon

On August 1, 2007, a version of these static maps was deployed for the public on Wave Transit's website (www.wavetransit.com) shown in Figure 4.10. This was the first time any of the project's components gained active production status, which was a true test for the application and supporting classes. Despite minor humps in the first few months, the system has been running with minimal downtime since deployment. The original deployed version contained static report maps and a link to the Google Earth script, detailed in the next section.

Figure 4.10: Wave Transit's Live Tracking Website at the Time of Deployment

4.5.2 Third Party Application Integration

For users wanting a more enhanced experience, dynamically generated Keyhole Markup Language (KML) files can be downloaded into Google Earth as demonstrated in Figure 4.11. This application plots all buses currently active in Google Earth and is set to update at select intervals. Google Earth already contains both road maps and satellite images, so location recognition is easier for the user and not dependent on the programmer to include. In addition to the bus locations, other live information can be added to this map such as weather, live traffic, and area web camera feeds.

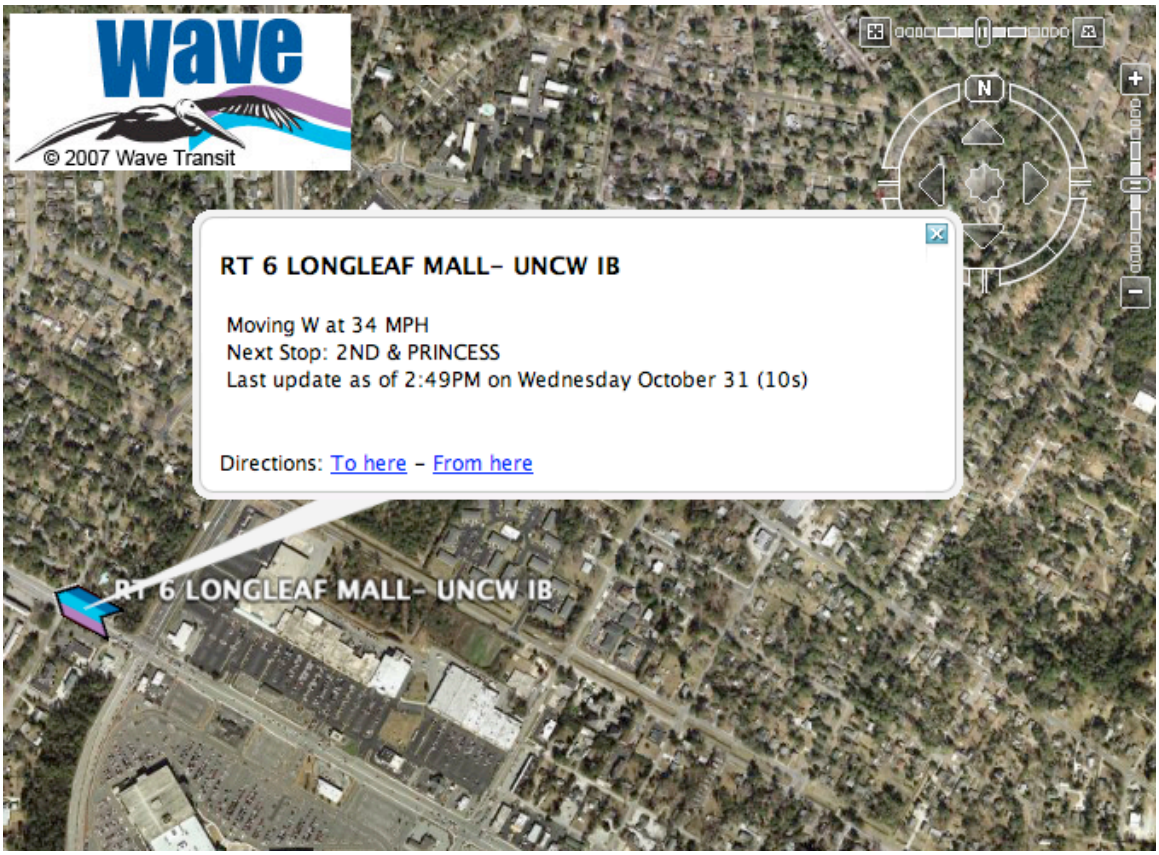


Figure 4.11: Google Earth Sample Screenshot

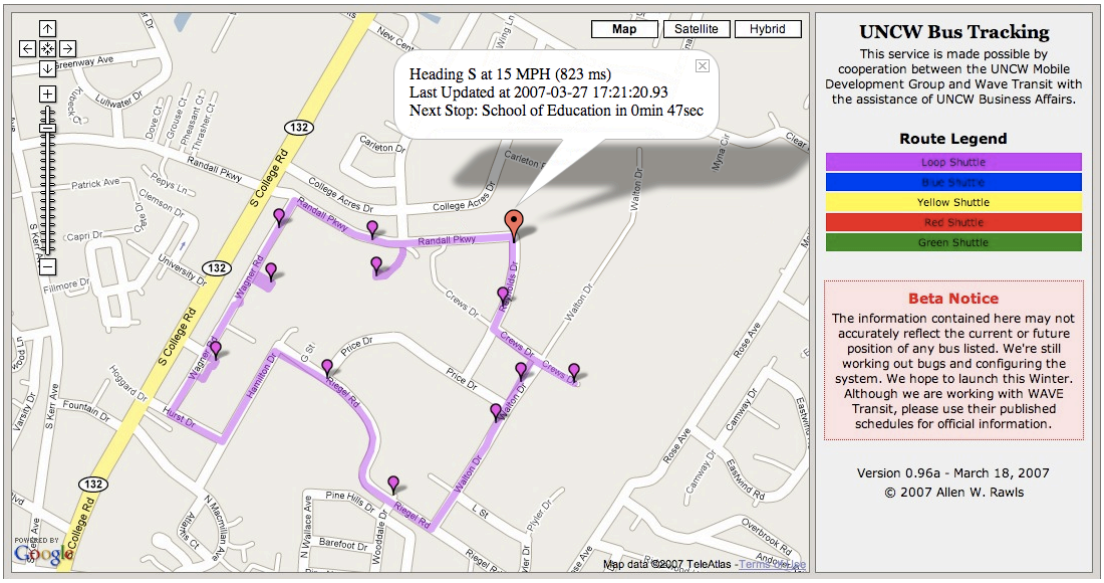


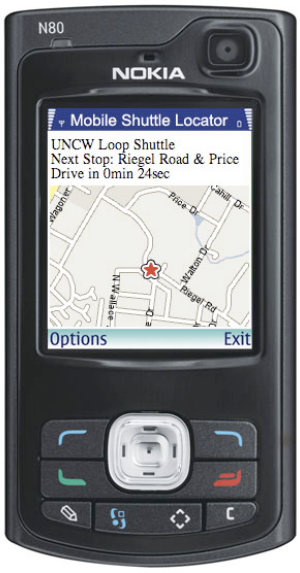
Figure 4.12: Google Maps Interface Prototype Sample Screenshot

Not only is Google Earth a useful display means for the resulting KML files, but they can also be displayed in Google Maps, either by integrating the Google Maps API into a website, or by sending the KML URL to the Google Maps site as an overlay. The ETA shown in Figure 4.12 is computed using the distance to next stop and the average speed value defined in the above section.

In addition to these two solutions from Google, they also provide a third application, targeting transit companies. Mentioned earlier, Google Transit allows transit companies to upload data about their routes for display in Google Maps. The service is provided at no charge to the transit agency and the data is submitted to Google via an archive of text-based data files. These files include the routes, stops, fares, and system schedule, all information that this project is able to provide. Through an agreement with the transit operator, another extension of this project is to provide Google Transit with the necessary information to include Wilmington, North Carolina's transit agency routes and schedules with their other big city and national agencies.

4.5.3 Mobile Websites

The same design for creating the static map webpage is applied to mobile devices, only substituting WML for the HTML as described above. A smaller sized image is grabbed for display on the mobile device. The website can detect the incoming mobile device's capabilities through the device's web user agent, therefore sending out a page that can be understood by the device, whether it be old WML, more enhanced WML, or even HTML. Figure 4.13 shows a sample of what the old WML would look like in the format it was originally intended for: an older monochrome limited line phone. Alternatively, Figure 4.14 shows a screenshot of the interface developed for today's cell phones providing the user with textual data in addition to a map graphic.



Figures 4.13 and 4.14: Original WML Sample (left) and Rich-WML Sample (right)

4.5.4 Mobile Text Messaging

For the most up-to-date information in vehicles, text messages to cellular devices was an innovative idea. For the students on the campus routes, pulling out a cell phone to

send a text message is almost becoming second nature. It seemed practical to take this technology that already exists in the campus community and apply it to making bus location inquiries. With that in mind, a text messaging responder added an additional front end to bus tracking so users could request the status of a bus from their cell phone. To interact with the system, a user sends a message with a pre-defined bus identifier to a defined phone number. On the server side, a web server receives the bus identifier and returns a XML response containing relative information about any vehicles on that route such as road location and next stop. From the web server, a text messaging response is created that responds to the sending telephone number. During the design phase of this component, there was not an XML schema for delivering bus status information. Therefore, a schema was created to define the XML format, which is located in Appendix A. Mobile Education LLC, a company in Wilmington, North Carolina currently uses this schema to provide text messaging in Wilmington and will use the schema when requesting transit information from other municipalities.

4.5.5 Experimental Technology

In light of text messaging as an emerging technology and the partial obsoleteness of call-in systems, creating a natural language processor for transportation information was though to be a useful addition to the existing means of information delivery. Such a program was created to handle questions that people might ask about the bus system and live locations. The application attempted to match keywords in questions to known facts or topics, returning a best-guess answer. For instance, the system knows all vehicles and details about each one, including whether or not it is active and its location, next stop,

previous stop, and bus driver. A large disadvantage of the system was that the data on stop locations was too specific. If a user asked, "When will the loop bus be at the library?" the system did not have an entry for the closest bus stop to the library, so it would not know what to return. If the system were to be deployed, more data would need to be gathered to build a database of toponyms that mapped to known stop locations along a route, which would also include nearby points of interest.

4.5.6 Conclusion

While this research project attempts to create similar functionality to the systems overviewed in Chapter 3, the information delivered is designed to be informative, giving at minimum the current location of the vehicle. The information delivery system was deployed for the entire transit fleet while the research focus on arrival time applies to one route only. For other routes, they are covered with the basic AVL functionality sent to the user with information such as last report, location, and next stop. User familiarity with the routes was high, so stating the current location of the vehicle would be valuable information to give a rider waiting at a stop.

4.6 Summary

This chapter presented extended reasoning for the route chosen to analyze and described a series of tools to ease both the data collection and the resulting analysis. While none of the tools follow a set standard, custom solutions were devised for each, and at least one production system was deployed using a subset of these tools. The other information delivery items described in this section serve as proof of concept

demonstrations and not all were intended for deployment. Nonetheless, this section presented a system for collecting and filtering AVL data from which the analysis can be performed in the next chapter.

Chapter 5

Results and Findings

Using the data collected by the local transit logs, filtered for suitable half routes described previously, it can now be analyzed to determine which factors have the maximum effect on transit time of vehicle on the campus loop shuttle route. This chapter presents those findings and discusses the importance of the calculated values and how they play a part in an estimated arrival time calculation.

5.1 Two Week Sample Set

After the first two weeks of data collection, a sample set of data was collected for preliminary analysis. At this point, the project had been recently granted access to the transportation database, so bugs were frequent. However, the data collected shed light on future logging versions and became the motivation for continuing to log transit data. The logging application was not robust enough to handle reporting problems or an unresponsive server, so it often crashed and was not restarted until the following day. Over the two-week period, 14,000 points were collected, which were later categorized in logging version one.

The collected data was analyzed primarily by presenting it in tables and graphs, thereby highlighting different aspects of the data. Figure 5.1 shows the distribution of number of points per day-of-week collected. If there were a large percentage of points collected on the same day in the data set, it would be easier to attribute the transit time

fluctuations to the day of week. However, since there is a fairly even distribution over this data, the day of week factor is negligible. Similar to the speed count graphs presented earlier, Figures 5.2 and 5.3 show the speed distributions by speed in miles per hour and colorizes for the day of week.

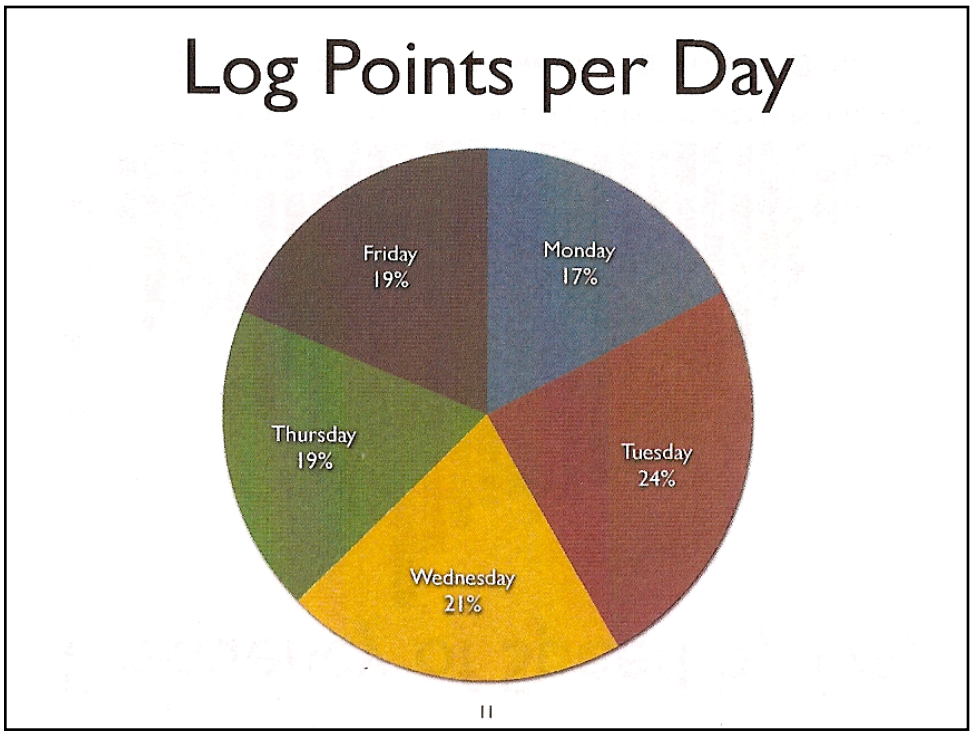
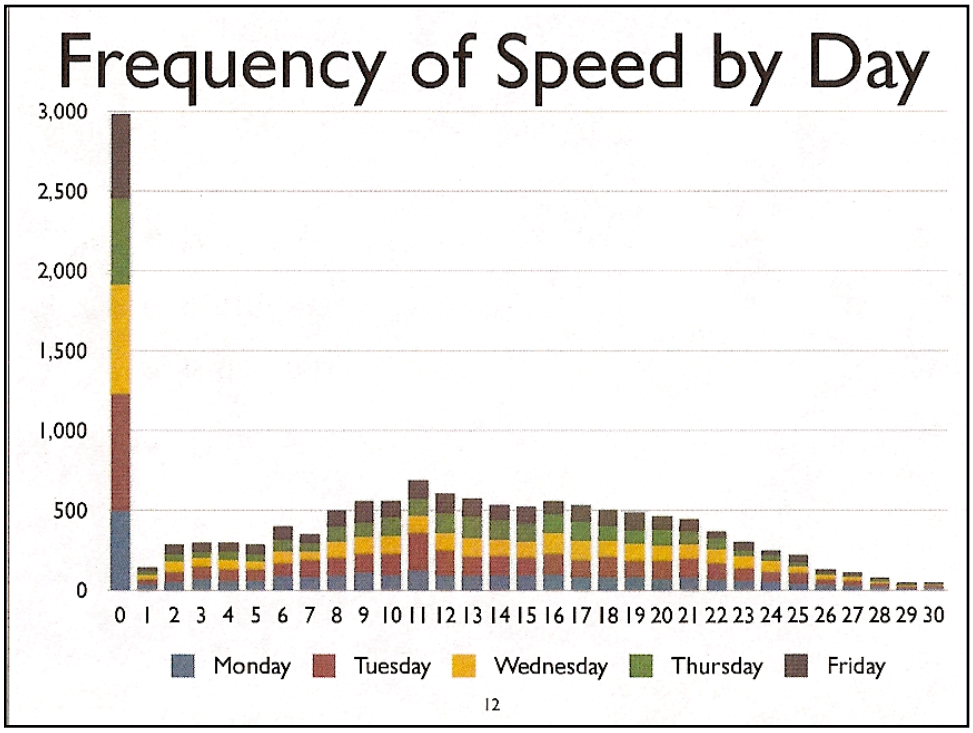
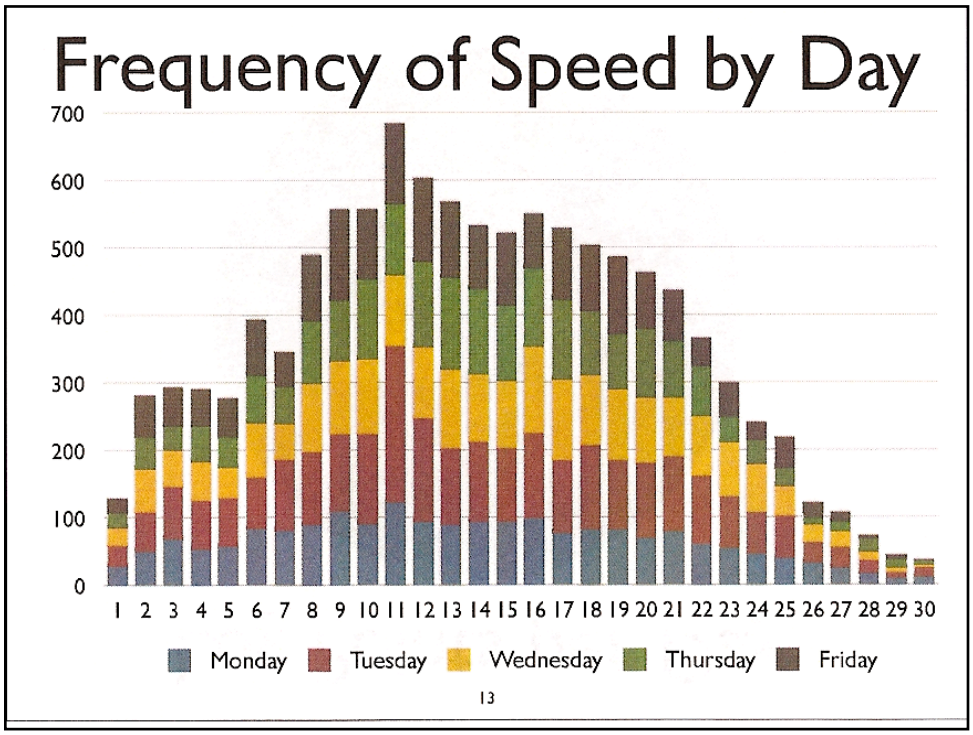


Figure 5.1: Distribution of Log Points per Day in a Two-Week Sample





Figures 5.2 and 5.3: Distribution of Reported Instantaneous Speed in a Two-Week Sample Set, Reports of Zero Speed (top) and excluding Reports of Zero Speed (bottom).

Given the small size of the data set used here, the data can be shown in its entirety on a table, which makes it easy to draw direct conclusions. Table 5.1 gives the details that are known for each logging day. One key thing to point out is the variability of standard deviation speed on days that it was reported to be precipitating. All precipitating days collected around 1300 points, so the speed fluctuations cannot be attributed to the number of data points collected. Having this lower value for a standard deviation speed suggests that weather plays an important role in vehicle speed and therefore ETA. Having a lower speed standard deviation combined with a higher average speed, it suggests that the vehicle made more stops, stopped faster, and gained speed faster, as if the vehicle was struggling to stay on schedule. Without having a full day of accurate data, it was difficult to prove if the vehicle really was off schedule. Nonetheless, it added an additional parameter to consider in future data analysis: weather. If it is believed that bad weather promotes additional riders, ridership is also a parameter worth testing. Unfortunately, ridership data proved to be problematic due to a lack of a solution for gaining ridership data on such a granular level as per day per route per stop from the transit operator since it was not a component that was tracked through the AVL system.

The combined conclusion of this data analysis suggests average speed is not enough to make an accurate prediction due to the influence of additional factors, but could serve as an accepted solution as a fallback. The determining factors included time of day, day of week, ridership, traffic, and weather. Even for this small sample, historical data mining would be more useful than average speed. Noting that on a previous rainy day the average speed would be slightly higher and the standard deviation speed would be slightly lower, and the rest of the data falls in line with the initial assumption.

Ext. Details by Log Day

Day	Low	Median	Average	St. Dev	High	Points	Weather
Tue, 14 Nov	0	15.5	10.678	9.257	31	205	74
Wed, 15 Nov	0	15	10.9103	8.0752	30	1316	73
Thu, 16 Nov	0	15.5	11.1583	7.582	31	1308	75
Fri, 17 Nov	0	16.5	10.3523	8.1031	33	1300	61
Mon, 20 Nov	0	17.5	10.6799	7.759	35	1131	57
Tue, 21 Nov	0	15.5	11.1016	7.8139	31	1319	52
Tue, 28 Nov	0	17	10.3183	8.4818	34	578	71
Wed, 29 Nov	0	16.5	10.4764	8.555	33	296	75
Thu, 30 Nov	0	16.5	10.9296	8.2233	33	1307	78
Fri, 1 Dec	0	15	11.0557	7.6188	30	1310	78
Mon, 2 Dec	0	17.5	11.1678	8.7022	35	1299	53
Tue, 3 Dec	0	16.5	11.3092	8.5881	33	1313	56
Wed, 4 Dec	0	16.5	11.0336	8.6994	33	1310	60

16

Table 5.1: Extended Details including Weather Conditions for Two-Week Sample Set.

5.2 Case Study

Before pursuing an independent solution, a sample set of the data collected by the project was run through the algorithm defined by [2] in an attempt to replicate their results. However, their method for computing a predicted arrival time was not specified in sufficient detail to proceed. They used coefficients and tolerances for predetermined models to influence their predictions, such as values for whether it was snowing or not. The coefficient was mentioned without providing the value or how it contributed to the calculation. Because the details were not described in sufficient detail, this project pursued a custom solution. This new solution incorporated what was already known and discovered from reviewing the work described in [2].

5.3 Collective Data Analysis

After migrating away from an existing solution, it seemed useful to create a table of the sample data segments, comparing different transit times such as published ETA, estimated ETA using average speed, and the actual arrival time, along with the known particulars of the trip that might influence prediction. For the average speed calculation, 1056 feet per minute was used, since it was the established average speed for this half route. This average speed value will be used in all future calculations unless otherwise noted and was the established value when modifying the logging versions that reduced the predicted future distance error. Table 5.2 shows the six sample segments to be analyzed, which is only a snapshot of the database.

LogID	Time (min)	Day	Time	Avg Speed	Med Speed	Schedule ETA	Est. ETA (Avg/Med)	Actual Arrival
20802	9.5	R	1630-1640	783.6 ft/min	1056 ft/min	1637	1638 / 1642	1639
30914	9.5	T	1230-1239	646.8 ft/min	1232 ft/min	1237	1245 / 1238	1239
45455	13	W	1250-1303	778.9 ft/min	1056 ft/min	1257	1303 / 1259	1303
64550	9	W	1310-1319	949.8 ft/min	1188 ft/min	1317	1320 / 1318	1319
76756	8	T	1431-1439	1127.5 ft/min	1144 ft/min	1437	1440 / 1440	1438
77776	10	W	1152-1202	716.5 ft/min	1232 ft/min	1157	1206 / 1200	1152

Table 5.2: Six Samples Gathered at Random Intervals Over Three Months

As outlined, a half route can take anywhere from eight to thirteen minutes to complete, each with a variant average and median speed for that segment. The data samples were chosen to span different days and driven by different drivers. This indicated that each driver needed his/her own classification system to further define the

likely transit times for that driver. With that in mind, each driver's segments were charted such that all plots for the individual AVL reports along the segment were indicated, connected by a line to identify that individual segment, yielding a line relating time to location along the route. However, Figure 5.4 shows one such graph for one of the more active drivers.

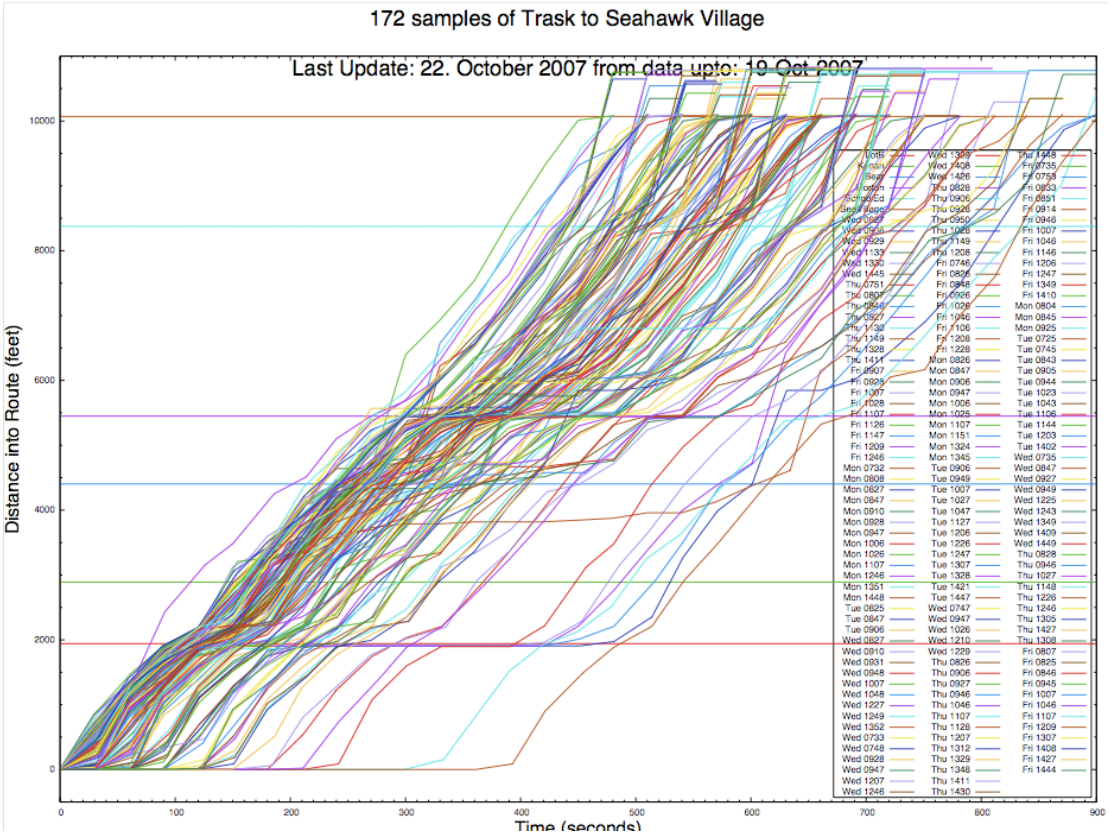


Figure 5.4: Graph of One Driver's Segments, Labeled by Time and Day of Week. (The Horizontal Lines Indicate Stop Locations along the Route.)

Referencing the graph, there is a clear lower bound on the minimal time it takes this driver to get from Stop 0 to Stop 6. It is also evident that the slopes are similar. The one thing that sets the arrival time off is the amount of time spent waiting at Stop 0 before beginning the route. This is explained by either running ahead and taking time to let the schedule catch up, high bus activity, or the changing of drivers. There are three distinct

regions where the slope remains the same and the rest of the route is time dependent on the departure time from that stop. Those would be between stops 0 to 1, 1 to 4, and 4 to 6. This is justified because stops 0, 1, 4, and 6 are located off the main road, in a parking lot or side driveway, respectively. Alternatively, when a stop is located on a road, the driver is less likely to sit and wait for passengers, especially if it delays traffic.

Noting the three main regions as mentioned from the graph above, the analysis may yield an accurate piece-wise function to describe the route. This function may be further split by the physical factors from historical data that might influence the transit segment. These include analysis on time of day, day of week, weather, driver, and some combination of these factors.

Given the advantage these graphs provided for identifying outliers and as a way to better visualize the variance of certain factors along the route, additional graphs were created for other factors that might influence the arrival time of vehicles. A graph set was created for each day of week, each hour and half hour of the day, each outside temperature range, and each varying weather conditions.

To minimize the complexity of this project, it was only desirable to know when the vehicle will reach Stop 6. Making this prediction as early as possible was still a top priority. With that in mind, the details along the route of stoppage only made a difference if it directly affected the end result time. Therefore, minimal stoppages became less relevant, but could still be used to make a better prediction if one of the other methods failed.

Here is an outline of possible approaches for making an estimated arrival time calculation:

1. Using overall fixed average speed value
2. Using overall fixed median speed value
3. Reference the graphs by:
    a. Driver
    b. Hour / Half Hour
    c. Temperature / Weather
    d. Day of Week
4. Tiered approach using weighted values
    a. Driver
    b. Hour / Half Hour
    c. Temperature / Weather
    d. Day of Week
    e. Combination of any of the above parameters

5.4 Finding the Best Criteria

In order to test each possibility as a viable solution, the key criteria needed to be extracted and tested for accuracy. The test set of data generated with the half route segment filter contained 1052 usable segments. To be an applicable segment, the vehicle must be within the stop region for Stop 0 having a speed less than or equal to five miles per hour. Within twenty minutes of the first approved data point, the vehicle must be within the stop region for Stop 6, having a speed less than or equal to five miles per hour.

Limiting the transit trip to twenty minutes removed any reporting anomalies from appearing in the sample set. Assuming a sequential progression of points along the route from the first to second criteria, this half route was considered applicable. For generating these half routes, the data set was limited to exclude the first 29,000 data points. During the semester, the campus route was changed due to construction; the old Stop 0 was now located in a closed parking lot, so a new Stop 0 location was created. With this change, the distance calculation and virtual campus map needed to be recomputed to reflect the change. Since this would yield incorrect predictions in the data set, those plots were excluded.

5.4.1 Prediction by Weather

The first parameter that showed the most potential in the two-week sample set was weather condition. Table 5.3 shows the percentage of each weather condition for the sample segments gathered. The condition of Clear skies occurred 61.62% of the time for 1050 usable data segments. This number of segments is two less than the total number of usable samples because two results had unknown weather, a condition where the airport METAR was likely under repair. Being right on the Atlantic Ocean, a full day of clear skies is rare. Therefore, to constitute a nice day, any amount of clouds except completely overcast could yield a nice day, or at least it does not inhibit transit operations in a negative way. It would be difficult to measure how the percentage of cloud cover influences riders to take the shuttle or walk across campus. In that case, segments categorized as nice days make up 88.95% of all usable segments.

The condition indicating falling precipitation of any kind only occurred 3.3% of the time. The other conditions included haze or fog, which also would have little effect on rider decisions. Adding further granularity, grouping the weather conditions with the temperature to give the type of day it would produce, such as nice day, rainy day, or muggy day, the dataset still limits the conclusions. With such little data to draw on for the key weather conditions, the aspect of using weather to make predictions had to be discarded. This is primarily due to the number of bad segments that had to be discarded for the ten months of data collected. However, over all plot points gathered, the distribution is about the same, with a slightly higher percentage of rainy days. Given more usable segments, considering weather conditions could be a viable solution.

Weather Condition	Count	% Total
Clear	647	61.62
Scattered Clouds	112	10.67
Partly Cloudy	88	8.38
Mostly Cloudy	87	8.28
Overcast	70	6.67
Light Rain	27	2.57
Haze	10	0.95
Rain	4	0.38
Heavy Rain	2	0.19
Thunderstorms and Rain	2	0.19
Fog	1	0.09

Table 5.3: Frequencies of Weather Conditions for 1050 Usable Segments

5.4.2 Prediction by Driver

With weather out of the equation, driver seemed the next likely factor to influence the transit time of a trip. For the campus shuttles, more than just driving the bus makes for a good driver and the data reflects the driver behaviors. It was observed that some

students sit up front just for this reason. However, getting back to what is clear directly in the data, each driver has an average speed that is unique only to that driver, since it's the average speed of that driver's trips only. The only downfall when using driver is for substitute drivers. In Table 5.4, several of the lesser percentage drivers were likely substitute drivers because their percentages were low even when compared against all segments, usable or not.

Driver	Count	% Total
A	477	45.34
B	114	10.84
C	67	6.37
D	55	5.23
E	55	5.23
F	41	3.90
G	30	2.85
H	29	2.76
I	23	2.19
J	17	1.62
K	16	1.52

Driver	Count	% Total
L	16	1.52
M	15	1.43
N	15	1.43
O	15	1.43
P	14	1.33
Q	13	1.24
R	12	1.14
S	8	0.76
T	7	0.67
U	7	0.67
V	6	0.57

Table 5.4: Frequencies of Drivers for 1052 Usable Segments

There are two problems with using driver alone as a means for determining estimated transit time. The first is accounting for an unknown driver, in the instance of a new substitute driver or a new employee. The database does not provide any additional information besides the driver identifier, so there cannot be two separate categories for unknown drivers, and it is very likely that a substitute driver will act very different compared to a new employee altogether. The second problem with using driver alone is that while the driver might behave the same every day, in public transit, traffic is predicted to be the key determiner. Although a seasoned driver may react the same to

varying traffic conditions, detailing when those traffic conditions occur is still a something to consider.

5.4.3 Prediction by Day and Hour

The transportation network used in this experiment is located in a medium sized city. Furthermore, the specific route for this thesis is a small campus shuttle bus route, which navigates along campus roads that wind their way through campus connecting the school's buildings and parking lots. Unlike other areas of town, the campus does not have traffic flow sensors or even traffic cameras, so determining the traffic flow became a challenge. However, a solution was devised based on class times on certain days. Try to drive on campus at 1:55PM on a Wednesday and you will likely be sitting more than driving due to the class change. However, wait ten minutes and apart from the few people getting out of class late, the campus roads will be relatively empty.

This method provides a base for traffic flow measurement. Although the affect of traffic flow on two different days might not be easily contrasted, it is a valid solution for comparing one Thursday at 3PM to another Thursday at 3PM, which is what the historical lookup comparison will do. To ensure this method is not too granular like weather, the dataset is divided by day and time-of-day in Table 5.5 to show there are at least five other entries for each day and time-of-day combination. Although there are some times toward the end of the day with smaller counts, these are discarded because the route changes after 5PM, so only hours before 5PM will be used for this project.

Hour	Monday	Tuesday	Wednesday	Thursday	Friday
7:00 AM	11	16	9	10	12
8:00 AM	24	24	20	21	25
9:00 AM	22	26	30	21	22
10:00 AM	26	24	27	20	23
11:00 AM	19	21	24	23	28
12:00 PM	26	20	23	27	23
1:00 PM	23	26	24	25	23
2:00 PM	15	19	18	16	28
3:00 PM	20	20	23	20	12
4:00 PM	15	8	10	9	10
5:00 PM	7	5	9	5	6
6:00 PM	3	2	1	1	2

Table 5.5: Number of Records of Day and Hour for 1052 Usable Segments

Hour could be further subdivided by half hour. However, there is not enough data in the primary dataset for the addition of half hour to provide an accurate solution for the majority of the time. Nonetheless, it could be used in instances when segment detail lookups against the historical data do yield enough data points to justify a prediction with these extremely granular criteria.

5.4.4 Combination of Prediction Methods

As mentioned in previous sections, using driver alone is probably not the best solution, but might be useful if there are a lot of records containing the given driver for pulling historical data analysis. Additionally, only using hour and day does provide a metric for traffic flow, it also is not telling the whole story. If enough data points are obtained, the more optimal solution would be to pull from a combination of all three factors, driver, day of week, and hour of day.

$$\text{Mean Squared Error [Dataset of n records]} = \frac{1}{n}\sum_{j=1}^{n}\left(actualTime - predictedTime\right)^2$$

Figure 5.5: Mean Squared Error (MSE) Formula

In order to determine which of these factors is the best and rank the others accordingly, there must be a way to gauge their effectiveness. This is done with a mean squared error calculation shown in Figure 5.5. The mean squared error is a method to quantify the amount by which an estimator differs from the true value of the quantity being estimated. In this instance, the calculation takes the actual transit time for a particular segment and compares it to the transit time as predicted by a lookup of identical instances of those criteria to make a prediction. If there is no base data on which to make a prediction, the estimated transit time is zero, thus returning a large value for that squared error.

The result shown in Table 5.6 calculates the mean squared error on the primary data set, called segments in the table, containing 1052 segments as compared to itself. The two sections of the table vary slightly because they use the start time of the segment instead of the end time for determining the hour of day. This was used to see if using a different segment reference times made a difference. A value for average transit time is computed by all segments matching the criteria and compared to the actual segment transit time. This is similar to the median calculation, except the median is used instead of the average. Next, Table 5.7 shows the same algorithm run on the testing dataset, identified by segments_new in the table, containing 148 segments compared to itself. Finally, Table 5.8 shows the same process, as it would be in an actual prediction calculation, comparing the test data to the large segment database to make predictions and compare to the actual transit times in the testing data.

segments, segments, endtime

Criteria	Avg Time	Med Time
Speed	20956	467473
Driver	9132	9680
Day	10054	10309
Hour	9755	10312
Hour&Day	9340	10318
Driver&Day	8790	9718
Driver&Hour	7272	9681
Driver&Day&Hour	5830	9787

segments, segments, starttime

Criteria	Avg Time	Med Time
Speed	20956	467473
Driver	9132	9680
Day	10054	10309
Hour	9776	10312
Hour&Day	9328	10307
Driver&Day	8790	9718
Driver&Hour	7322	9693
Driver&Day&Hour	6082	9797

segments_new, segments_new, endtime

Criteria	Avg Time	Med Time
Speed	18911	468580
Driver	7081	7774
Day	6693	7650
Hour	6084	7526
Hour&Day	4020	7701
Driver&Day	6045	7650
Driver&Hour	5124	7484
Driver&Day&Hour	3462	7701

segments_new, segments_new, starttime

Criteria	Avg Time	Med Time
Speed	18911	468580
Driver	7081	7774
Day	6693	7650
Hour	6554	7545
Hour&Day	3835	7760
Driver&Day	6045	7650
Driver&Hour	4878	7545
Driver&Day&Hour	3125	7760

Table 5.7: MSE Results on Testing Dataset Compared to Itself.

segments_new, segments, endtime

Criteria	Avg Time	Med Time
Speed	18911	468580
Driver	* 62863	9758
Day	7713	7776
Hour	7435	7694
Hour&Day	8458	7562
Driver&Day	* 63022	8562
Driver&Hour	* 80882	9702
Driver&Day&Hour	* 87270	8367

segments_new, segments, starttime

Criteria	Avg Time	Med Time
Speed	18911	468580
Driver	* 62863	9758
Day	7713	7776
Hour	7234	7658
Hour&Day	8267	7498
Driver&Day	* 63022	8562
Driver&Hour	* 80509	9747
Driver&Day&Hour	* 87080	8670

Table 5.8: MSE Results on Testing Dataset Compared to the Primary Dataset.

In Table 5.8, the stars signify a separate denotation for the higher values for criteria containing driver because the testing dataset contained an unknown driver for which the primary dataset had no sample segments, so an estimate could not be made using the historical data on those factors alone. Considering the other tables, although they vary slightly in which criteria show improvement, these values can be used to test in what order the step-down system will transgress until an acceptable number of segments are identified and an estimate can be made.

For an additional metric, each of the 148 segments in the testing set is compared against the dataset for each of the predicted factor combinations that might have the greatest affect on transit time. For example, one segment contains driver C on Wednesday in the four o'clock hour. For the driver percent error, all transit times of driver C in the primary dataset are averaged to produce an estimated transit time. This value is compared to the actual transit time, which yields a percent error. The process is repeated for the driver and day combination and also for subdividing further into driver, day, hour, and half hour. This entire process is then repeated over each testing segment. The chart in Figure 5.6 shows the occurrence of each percent error range by each testing criteria in an attempt to justify that average speed produces a wide range of percent errors while other combined criteria produce more instances of a smaller percent error.

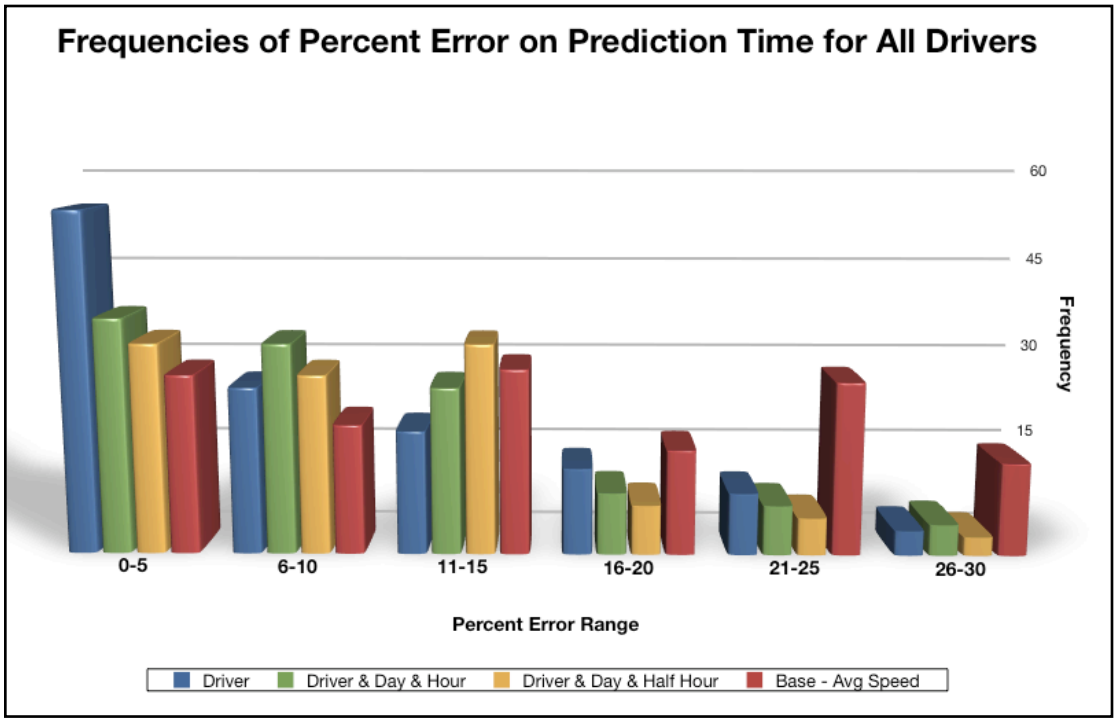


Figure 5.6: Frequencies of Percent Error on Predictions

5.5 Tiered Prediction Algorithm

While it is important to consider the most granular breakdown of criteria, there might not be enough usable data points to make a valid prediction. Therefore, there must be an implemented fallback method to still use valid criterion to make a guess. However, if there is still not enough data to use any criteria for prediction, the average speed value can be used as a last attempt. This next section ties together all the different methods and presents a solution based on the test data for determining a best-guess transit time and therefore an estimate arrival time.

For this hybrid solution, the following are tested until one returns enough samples of identical criteria. The average of the actual transit times for the segments that meet the criteria is used as a transit time estimation. In addition to this fall back structure,

different values are used for the minimum identical segments needed to produce a result. The fallback structure is as follows with the minimum matching segments needed:

1. Check on Driver, Day, Hour, and Half Hour - 14 matches required.
2. Check on Driver, Day, Hour - 14 matches required.
3. Check on Driver and Hour - 14 matches required.
4. Check on Driver - 14 matches required.
5. Check on Hour and Day - 5 matches required.
6. Use fixed average speed value over the route distance.

Using this method, the testing dataset is compared to the primary dataset using the start time for determining time-of-day. Table 5.9 shows several segments as they are estimated, giving the actual transit time verses the estimated transit time and how far off from the actual it was. With thirty seconds updates, an estimate within thirty seconds is considered to be the most accurate. However, the error percentages indicate the error from the predicted transit time to the estimated arrival time. So given these errors, an error of plus or minus three percent is acceptable to be zero error since more granularity cannot be observed in the data.

In addition to this table, the mean squared error calculation as demonstrated in Figure 5.5 and shown through data in Tables 5.6, 5.7, and 5.8 was performed on this output. It was only calculated for these parameters because they proved to be the most useful. The resulting value was 7644. Looking back at Tables 5.6, 5.7, and 5.8, this value is close the values in the tables, especially Table 5.8 where the dataset was not compared to itself.

Predicted	Actual	Error / Actual vs Predicted
654	720	9% Error / 1.1m After Predicted
617	690	11% Error / 1.2m After Predicted
617	631	2% Error / 14s After Predicted
684	537	27% Error / 2.5m Before Predicted
701	721	3% Error / 20s After Predicted
663	661	0.0% Error / 2s Before Predicted
658	750	12% Error / 1.5m After Predicted

Table 5.9: Sample Output from the Tiered Prediction Algorithm

5.6 Summary

The system returns an estimated transit time successfully when provided aspects of the current transit conditions. While it is not a superior algorithm, it still shows improvement over average speed for making an arrival estimate. Considering a few problems encountered during the project, a few improvements are outlined in the following paragraphs.

The test set used in this thesis was from the months of November and December, while the data set contained data from April through October. This was done so data collection for the test set could continue while building the algorithm using the primary dataset. However, a better approach would be to stop gathering data altogether and create one main collection of data, then extract 12% or so to be used as test data while the remaining 88% would be used for the algorithm creation. It turns out that the vehicles began better following the schedule towards the end of 2007, so the Primary dataset for which the algorithm is based contains highly erroneous data points. So when comparing

the late 2007 data to the early 2007 data, the system wants to predict a transit time that will vary from the actual time because of the erroneous data used to make that prediction.

In continuation from the previous discussion, invoking a data aging process on the primary data set that is used when making predictions would be desired. That way, as the transit company made improvements to help the vehicles stay on schedule, the old data is no longer referenced. The recommended time for this aging process is one year. Given more data, criteria could be further divided by month if enough segments are captured.

Chapter 6

Summary and Conclusion

This last chapter summarizes the contributions of this thesis and briefly discusses future work and additional development in this research area. It also addresses the problems encountered with advice to others pursuing similar projects.

6.1 Contribution

This thesis attempted to compute an accurate transit time for one route, the campus loop shuttle. In doing so, a series of tools was created to assist in various sub-processes of creating a working system. The campus was mapped, the transit system was put online to offer real-time updates through a wide range of display mediums, information was sent to Google Transit for world-wide recognition, and a reverse geocoder was devised for paved areas on campus. These tools function independently from other aspects of the project.

In the area of estimated arrival time, a hybrid approach was devised given the particular circumstances of the campus loop shuttle. The usefulness of the algorithm is likely limited to these campus conditions. This data could not be applied directly to other transit routes, but the methods to gather and interpret the data could be replicated in other environments, given a similar AVL system.

6.2 Future Work

Calculating transit time for a highly variant environment is not an easy task. Even with all the training data gathered, the devised algorithm was still unable to compute a good estimate. An evolving system is likely to have issues, which were evident during this project. Although the logging application ran continuously for eleven months, there were not eleven months of functional data for which to draw usable segments. In addition to the lack of logged data, knowing information about ridership and knowing how to deal with unknown drivers would be useful. However, these are areas that can be explored in further research.

6.3 Project Perception

In reflection of this thesis work, there are things that worked well and things that would be done differently should the experiment be replicated.

1. Having the foresight to begin logging the data when access to the AVL output was first granted saved a lot of time in the long run. Even collecting for a year, there were only 1052 usable segments. Otherwise the project would be delayed by a semester in order to gather more data.
2. Creating a failsafe and delayed logging application to handle system hiccups and sleep for the delay between report updates from the AVL server. This failsafe proved to be very useful when the system would stop responding altogether. In this case, that one thread would die and a new one would be created with a mandatory sleep period for two hours. Then that thread would try again, pinging every thirty seconds.

6.4 Conclusion

This thesis introduced a method for gathering and analyzing historical transportation location and telemetry data output from an AVL server to better determine estimated arrival time for a vehicle on a closed-loop public transit pattern. The research produced tools to assist in both transit and mapping operations associated with this thesis. Other tools include methods for relaying the AVL data to riders through a wide array of mediums including web pages, computer programs, graphical information displays in public locations, mobile phone applications, mobile text messaging, and internet feeds. The applications presented as part of this thesis serve as a working demonstration of the concepts presented herein.

References

[1] Brown, J., Vetter, R., Rawls, A., Shipman, B., "A Proposal for the Investigation into New Mobile Technologies and Applications for UNCW's Business Affairs Division." 4. September 2006.

[2] Cathey, F.W. and Dailey, D.J.. A Prescription for Transit Arrival/Departure Prediction using Automatic Vehicle Location Data. Transportation Research Part C, Vol 11C, No. 3-4, pp 241--264, June-Aug, 2003.

[3] Chien, Steven I-Jy and Ding, Yuqing and Wei, Chienhung. Dynamic Bus Arrival Time Prediction with Artificial Neural Networks. Journal of Transportation Engineering, Vol 128, No 5, pp 429-438, 2002.

[4] Collins, C., Grude, A., Scholl, M., and Thompson, R. 2007. txt bus: wait time information on demand. In CHI '07 Extended Abstracts on Human Factors in Computing Systems (San Jose, CA, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, 2049-2054.

[5] Crist, Dave, Martin, Andrew, Carland, Tristan. "Track Star: Web-based, Dynamic, GPS Point Tracking." Software Development Manual. 2005.

[6] Dailey, D.J. and Wall, Z.R. and Maclean, S.D. and Cathey, F.W.. An Algorithm and Inplementation to Predict the Arrival of Transit Vehicles. Intelligent Transportation Systems, 2000.

[7] Jones, M. K., "Base Station System and Method for Monitoring Travel of Mobile Vehicles and Communicating Notification Messages," U.S. Patent 6,317,060, November 13, 2001.

[8] Maclean, S.D. and Dailey, D.J.. Real-time Bus Information on Mobile Devices. Intelligent Transportation Systems, pp 988-993, 2001.

[9] Rawls, A., Vetter, R., Brown, J., "Real-Time Shuttle Bus Tracker." Poster Presentation at the Interact Technology Showcase, UNCW Information Technology Systems Division, Wilmington, NC. March 29, 2007.

[10] Rawls, A., Vetter, R., Guinn, C., "A Proposal for AVL Enhancements for the Cape Fear Public Transit Authority." July 2, 2007.

[11] Rawls, A., Vetter, R., "A Proposal for Continuation of UNCW's Bus Tracking Initiative." July 2, 2007.

[12] Rawls, A., Shipman, B., "Shuttle Bus Location Demonstration and New Mobile Technologies." Presentation to the UNCW Board of Trustees for Business Affairs, Wilmington, NC. October 26, 2006.

[13] Schmeir et al, “Public Transit Vehicle Arrival Information System,” U.S. Patent 6,374,176, April 16, 2002.

[14] Shalaby, Amer and Farhan, Ali. Prediction Model of Bus Arrival and Departure Times Using AVL and APC Data. Journal of Public Transportation, Vol 7, No. 4, 2004.

[15] Sinnott, Roger W. “Virtues of the Haversine.” Sky and Telescope Magazine August 1984: 159.

[16] Shan, Jingtao and Zhao, Fang. AVL Data Mining for Transit Service Performance Information. Proceedings of the 2006 Applications of Advanced Technology in Transportation Conference, Chicago, IL, August 13-16, 2006, pp 719-724.

[17] Sinnott, R. W. "Virtues of the Haversine," Sky and Telescope Magazine, Vol. 68, No. 2, August 1984, pp.159

[18] Tiesyte, D., Jensen, C., Challenges in the Tracking and Prediction of Scheduled-Vehicle Journeys. Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07), Los Alamitos, CA, USA, Vol 0, pp 407-412, 2007.

[19] Vincenty, T. “Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations.” Survey Review, Vol. XXII No. 176, Directorate of Overseas Surveys, Tolworth, Surrey, UK., April 1976.

[20] “Where’s the Bus, New System Will Tell You,” Wilmington Star News, Wilmington, NC, Page B1, December 6, 2006.

Appendix A

XML Schema for Short Message Service Transmissions of Bus Information.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<!--
            Document created on 13. August 2007
            By Allen W. Rawls
            XML schema definition for UNCW Bus Tracking
            XML->SMS Interface
            Validation URL:
http://www.validome.org/grammar/validate/?lang=en&grammarTyp=SCHEMA&url
=http://www.csc.uncw.edu/bustracking/images/sms.xsd
 -->

<!--  Definition of Simple Types -->
<xs:simpleType name="cardinalType">
      <xs:restriction base="xs:string">
            <xs:pattern
value="N|NNE|NE|ENE|E|ESE|SE|SSE|S|SSW|SW|WSW|W|WNW|NW|NNW"/>
      </xs:restriction>
</xs:simpleType>

<xs:simpleType name="degreeType">
      <xs:restriction base="xs:integer">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="360"/>
      </xs:restriction>
</xs:simpleType>

<xs:simpleType name="speedIntType">
      <xs:restriction base="xs:integer">
            <xs:minInclusive value="0"/>
      </xs:restriction>
</xs:simpleType>

<xs:simpleType name="longtimeType">
      <xs:restriction base="xs:long">
            <xs:minInclusive value="0"/>
      </xs:restriction>
</xs:simpleType>


<!--  Definition of Complex Types -->

<xs:complexType name="responseType">
      <xs:sequence>
            <xs:element name="route" type="routeType" minOccurs="0"
maxOccurs="1" />
      </xs:sequence>
</xs:complexType>

<xs:complexType name="routeType">
```

```
      <xs:sequence>
            <xs:element name="vehicle" type="vehicleType" minOccurs="0"
maxOccurs="10" />
      </xs:sequence>
      <xs:attribute name="name" type="xs:string" />
      <xs:attribute name="wtanumber" type="xs:integer" />
</xs:complexType>

<xs:complexType name="vehicleType">
      <xs:sequence>
            <xs:element name="position" type="positionType" />
            <xs:element name="nextstop" type="nextstopType" />
            <xs:element name="lastupdate" type="lastupdateType" />
      </xs:sequence>
      <xs:attribute name="id" type="xs:integer"/>
</xs:complexType>

<xs:complexType name="lastupdateType">
      <xs:attribute name="text" type="xs:string"/>
      <xs:attribute name="longtime" type="longtimeType"/>
      <xs:attribute name="age_s" type="xs:integer"/>
</xs:complexType>

<xs:complexType name="nextstopType">
      <xs:attribute name="name" type="xs:string"/>
      <xs:attribute name="distance_ft" type="xs:integer" default="-1"
/>
      <xs:attribute name="distance_sec" type="xs:integer" default="-1"
/>
</xs:complexType>

<xs:complexType name="positionType">
      <xs:sequence>
            <xs:element name="latitude" type="xs:decimal" />
            <xs:element name="longitude" type="xs:decimal" />
            <xs:element name="heading" type="headingType" />
            <xs:element name="speed" type="speedType" />
            <xs:element name="street" type="xs:string" />
      </xs:sequence>
</xs:complexType>

<xs:complexType name="speedType">
      <xs:attribute name="mph" type="speedIntType"/>
</xs:complexType>

<xs:complexType name="headingType">
      <xs:attribute name="cardinal" type="cardinalType"/>
      <xs:attribute name="degrees" type="degreeType"/>
</xs:complexType>

<!-- SCHEMA ROOT  -->

<xs:element name="response" type="responseType" />

</xs:schema>
```