Improving Traveler Information and Collecting Behavior Data with Smartphones

By

Jerald Jariyasunant

A dissertation submitted in partial satisfaction of the requirements for the degree of

Doctor of Philosophy

in

Engineering - Civil Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Raja Sengupta, Chair

Professor Joan Walker

Professor Shachar Kariv

Spring 2012

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

Abstract

Improving Traveler Information and Collecting Behavior Data with Smartphones by

Jerald Jariyasunant

Doctor of Philosophy

in Engineering – Civil Engineering

University of California, Berkeley

Professor Raja Sengupta, Chair

The recent growth of smartphones along with cheap, scalable cloud computing infrastructure has allowed for a plethora of new applications to be built. In transportation, two main efforts have been greatly impacted by this, delivering better traveler information to users, and collecting travel behavior information for researchers. This thesis describes 4 major efforts along these two themes; the development of a real-time transit trip planner, the evaluation of the value of real-time data, the development of a smartphone based automated travel diary system, and the design and evaluation of a behavior change experiment using the travel diary system. The first half of the thesis describes the technical development of the real-time transit trip planner along with experiments showing the positive impacts to travelers. Prior to this work, transit trip planners primarily used schedule data to route people through a transit network. This work is the first solution to the K-shortest paths problem to use real-time transit arrival data retrieved from a third party API. The algorithm then was implemented in an application called BayTripper, which serves over 1,000 users in the San Francisco Bay Area. The second half of the thesis describes the technical development of the automated travel diary system, which consists of battery efficient smartphone applications, server infrastructure to process data with trip determination algorithms, and web tools used to evaluate the accuracy of the system. The contribution to the literature is a catalogue of problems and related algorithmic solutions to building an end-to-end, battery efficient, automated travel diary. A behavior change experiment was designed and run using the automated travel diary system, which showed the potential for changes in users' awareness of their travel behavior, intentions to change, and short-term behavior change. This experiment represents a large scale test of the automated travel diary system, as well as a demonstration of using behavior change techniques, feedback and comparison, to promote sustainable travel behavior.

Contents

List of Figures

List of Tables

Acknowledgements

This thesis was made possible by a lot of people who have helped me over my lifetime. I would like to thank my mom and dad, my advisors, Raja Sengupta and Joan Walker, my fellow colleagues, Branko Kerkez, Dan Work, Eric Mai, Andre Carrel, Venky Ekambaram, DJ Gaker, Justin Martinez, all the undergraduates who helped me with my work, Adam Bemowski, John Gunnison, Stasa Zivojnovic, Michael Nole, Thomas Wong and Tracy Stallard, my coworkers, David Palmer, Thejo Kote, Ram Jayaraman, Swaroop CH, Ljuba Mijlkovic, and Jerome Tave. I would also like to thank Nextbus for providing me with data and the University of California Transportation Center and USDOT SafeTrip21 Networked Traveler project for funding the research.

1 Thesis Introduction

1.1 RECOGNIZING OPPORTUNITIES IN TECHNOLOGY

Development of technology is often the result of recognizing opportunities - when creative ideas become technically feasible to solve problems in the world. Examples include the birth of search engines used to index the world's information on the internet, or in-car navigation units which displaced maps once GPS technology became public. My work over the past few years, summarized in this thesis, is the result of four trends which started around 2007, which opened up opportunities at the intersection of mobile technology and transportation. The technology that I've developed stands on the foundation laid by others who worked in these fields.

- Proliferation of Smartphones. Smartphone usage has been growing at a rapid rate since 2007. Smartphones offer more advanced computing ability, with features such as touch screens, cameras, GPS, and accelerometers, which have been used by human computer interaction designers to implement persuasive technology.

- Open Transportation Data. Since 2007, hundreds of transit agencies have released their schedule and route configuration data in a popular format called GTFS. Many of those agencies have also made available real-time positions and of buses and released open Application Programming Interfaces (APIs). Mapping data, thanks to Open Street Map and the USGS has allowed for developers to enhance transportation datasets to deliver innovative routing applications for ordinary citizens.

- Persuasive Technology. Persuasive technology is broadly defined as technology that is designed to change attitudes or behaviors of the users through persuasion and social influence. This technology focuses on the design, research, and analysis of interactive computing products created for the purpose of changing people's behaviors. Although behavior change methods have been used by psychologists for years, only in the past few years have these techniques been implemented on computing devices, delivering information in an automated manner.

- The Quantified Self. The concept of the Quantified Self describes applications which enable the process of recording behavior, processing the data collected, and feeding it back to the individual or group so that they can better understand the patterns of their activity and eventually adapt their behavior more intelligently than they would without these augmentations. As smartphone usage has increased, the launch of applications on smartphones has increased to track and study many features of people's daily lives (i.e. fitness, mood, spending habits).

The first trend is the most significant - smartphones and mobile technology have lead to far reaching advancements in many fields. Specifically in the field of transportation, mobile phones have lead to applications which help people get directions on the go, retrieve real-time

transit information, check traffic conditions, hail taxis, and tools that help planners learn about travel patterns and route choices, amongst other things. Some of the aforementioned applications have been enabled as a result of the second trend, the release of transportation data to the public, and a number of mapping related tools developed by both private industry and open-source communities.

The first two chapters of my thesis take advantage of the first two trends. At the time I developed a real-time transit trip planning application, NextBus, had just installed GPS devices on public buses and released the position information and estimated arrivals on the web. Google had just developed a format by which to encode transit routes and schedules called GTFS, and about 10 agencies had packaged their transit data into GTFS and released it to the public. The "app economy" had barely started, as Apple had just released their SDK (Software Development Kit), Android did not exist, and mobile Java development for phones existed for a few games. The work that was put together thanks to open transit data and smartphone app development was a real-time transit trip planning application that used the positions of buses to determine the quickest route between origin and destination. Chapter 2 describes the algorithm and evaluation of the system, while Chapter 3 describes the value of real-time data in trip planning. This work represents the design and implementation of the first transit trip planner to use real-time data instead of schedule data to route people in a transit network. Prior to this work, there did not exist an algorithm to fuse real-time information from a third party API with transit trip planning, and more importantly, there did not an actual application that users could run to route themselves with real-time data. The application that was developed, called BayTripper, was an implementation of the real-time routing algorithm and serves over 1,000 active users in the San Francisco Bay Area daily.

1.1.1 Influencing Mode Choice

The second half of thesis is about using these trends and technology to influence mode choice behavior. Mode choice behavior refers to understanding the way people use transport, where they go, what mode they take, and the reasons why they make these decisions. There is a large body of work dealing the with the modeling of transportation behavior and also attempting to influence travel behavior. It was recognized that there now exists an opportunity now to advance that body of research even further because of advances in technology over the past few years. The latter two major technological trends have given rise to advances in psychology and human computer interaction which has lead to a growth in academic research involving new methods of behavior change.

The first step to changing behavior is understanding the person's current travel behavior - which has been done in the past with manual pen and paper travel surveys. These travel surveys have started to take advantage of new advances in technology, namely GPS, current solutions are not ideal due to the burden of carrying around additional devices or still having to log information. With smartphones being equipped with multiple sensors, the manual work of recording travel diaries has been enhanced with GPS traces. Chapter 4 describes my work in automated travel diaries on smartphones, and the challenges of working with the limited battery life of smartphones. The result of that work is a system that generates travel diaries, which can be used for experiments to influence mode choice behavior. The

contribution to the literature is an end-to-end, battery efficient, automated travel diary system using smartphones, and a catalogue of the challenges of using smartphones as the primary data collection tool with solutions to these problems.

Smartphones, persuasive technology and apps around the Quantified Self movement have already been recognized by industry and the academic community, leading to new research in the fields of health and fitness, as well as environmental conservation. Those trends have advanced the fields of behavioral psychology by using technology; the aim of my research is to advance the body of knowledge around influencing transportation mode choice behavior. Chapter 5 describes the first step to achieving this; the development of a website called "The Quantified Traveler" which uses the automated travel diary system to provide feedback to users about their travel habits in a easy to understand graphic interface. Results on changes to users' awareness of their travel behavior, intentions to change behavior and actual measured behavior change show that there is more potential for behavior change experiments.

2 Mobile Transit Trip Planning with Real–Time Data

2.1 INTRODUCTION

Mobile phones equipped with GPS and Internet access are promising platforms upon which future transportation information will be shared and collected. With these devices, real-time monitoring of the transportation system will become not only feasible, but ubiquitous. Numerous emerging traffic monitoring applications use vehicle probe data collected from on-board GPS devices to reconstruct the state of traffic (for example, velocity or density maps). This information is used to predict travel time of vehicles in the transportation network. Some emerging examples include Mobile Millennium [1, 2], *CarTel* [3], *JamBayes* [4], *TrafficSense* [5], and systems for surface street estimation [6].

The Internet has expanded these capabilities significantly, with various mapping providers (i.e. Google and Navteq) merging static and real-time traffic data to help drivers make the most of their commute. Yet, relatively little attention has been paid to public transportation and real-time data applications. Services such as 511.org and Google Transit allow users to plan public transit trips by generating routes based on static scheduling data. A multi-modal trip planning system was developed by Rehrl et al. [7] to address the increased complexity and lack of information required by travelers. Their platform includes directions for public transit based on published schedules, as well as transfer directions between different transportation modes. An open-source multi-modal trip planner, Graphserver, has also been developed by the online transit developer community to address the same issues [8]. However, buses and trains do not always run on time. Some transit agencies, TriMet in Portland, Chicago Transit Authority, Bay Area Rapid Transit, and King County in Seattle have responded by releasing real-time bus arrival feeds online, allowing application developers to use this data in new, novel ways.

BayTripper is a real-time public transit trip planning system accessible on mobile devices designed to use this information. It combines a user's geographic location with real-time transit information provided by transit agencies to determine the fastest route to a desired destination. It fuses real-time data feeds with the existing technology of schedule-based transit trip planners (TTPs) currently available online. To the best of our knowledge, the research is the first instantiation and evaluation of a real-time TTP which uses the predicted bus arrival feeds to route users on mobile devices.

This article is organized as follows. In Section 2.2 the challenges and previous work with real-time information in transit. In Section 2.3, we describe the system architecture of BayTripper, which has been deployed to serve transit networks in two metropolitan areas. We describe the dynamic K-shortest paths algorithm with predicted link travel times in Section 2.4. Using data from hundreds of trips planned in San Francisco, we evaluate the performance of BayTripper in comparison with a schedule-based TTP in Section 2.5. From analysis of the data, we describe areas of research to improve the accuracy and optimality of the system.

2.2 NEED FOR REAL-TIME INFORMATION IN TRANSIT

Annual system reliability reports are published by most transit agencies to address the issue

of schedule adherence. Measured at terminals and intermediate points, system reliability of a public transit network is defined as the percentage of vehicles that run on time according to schedule (up to four minutes late and one minute early) [9]. Schedule adherence in the San Francisco area is estimated to be 70% [9]. For the past two years, official *Metropolitan Transportation Authority* (MTA) numbers for New York, show a system reliability of 80% for subways and 66% for buses. In such cases, real-time information may be used to reduce time wasted waiting for delayed transit vehicles, and it can also enable users to take faster alternative routes. Hickman and Wilson [10] analyzed the benefits of real-time information, limiting the assessment to look at travel time improvements, and concluding that real-time information systems may only lead to marginal travel time benefits from improved path decisions. Their conclusion was supported by the development and subsequent evaluation of a dynamic path choice problem, in which a mathematical optimization reflects the decision of a transit rider to board a transit vehicle based on the availability of real-time information. Mishalani [11] et al. developed an evaluation method for the value of real-time information and noted that the value of information to passengers is affected by the type of available information as well as operations characteristics. The research lead to the development of a piece-wise linear function to model passenger utility, taking into account waiting times at stops as well as projected waiting times given by an arrival time engine. Surveys conducted by Caulfield et al. [12] suggest that users are generally unhappy about the on-time performance of public transit vehicles, and that the bulk of those surveyed would use real-time information if it were available.

The need for real-time data stems from the unreliable nature of bus schedules, which a number of researchers have explored. [13, 14]. One significant problem is the phenomenon of bus bunching, in which multiple buses on the same line arrive at a stop concurrently, followed by no subsequent buses for a significant amount of time [15, 16]. This can lead to travel delays, as some passengers experience longer wait times than predicted by the published schedules. Various control schemes have been proposed to combat this issue, the most popular of which allots slack time in the routes in a procedure known as holding [17]. By building some wait time into the schedule, a bus can wait at a stop if it is on time or ahead of schedule, or skip the waiting process if behind schedule.

The present chapter does not attempt solve the bus bunching issue, but rather aims to enable commuters to make informed decisions based on the inherent variability in the system, and to quantify the discrepancy of travel time as predicted by published schedules and real-time data. This work provides a tool which enables a commuter to use real-time information to find an optimal route in the transit system. BayTripper leverages bus arrival estimation engines developed by NextBus, TriMet, and other agencies which provide a real-time feed for bus arrival predictions.

Previous studies have focused on transit vehicle arrival prediction, which convert real-time bus locations to predictions for arrivals at downstream transit stops. However, none of the studies have consider applying the research into a transit trip planning tool. Previous work by Jula et al. [18] used historical and real-time data to show that travel times can be estimated confidently along arcs of dynamic stochastic networks. Ignoring correlation between adjacent arcs on a network, the authors employed a technique based on a predictor-corrector form of the Kalman filter, in which historical data were used to predict travel time, and real-time measurements were used to correct the predictions at each time instant.

Shalaby and Farhan [19]used two Kalman filter algorithms for the prediction of running times and dwell times alternately in an integrated framework. Separating the bus dwell time prediction from bus running time prediction in this modeling framework captures the effects of lateness or earliness of bus arrivals at stops on the bus dwell time at those stops, and hence the bus departure from such stops. Other models were developed by Abdelfattah [20] and Chien [21] which used a large number of parameters such as live traffic volumes, speeds, and passenger counts. In the previous two papers, a very comprehensive models were developed, although many of those parameters featured in these models may be difficult to measure in real-time and consolidate for a user-application for transit trip planning. The chapter does not propose new estimation algorithms; it uses the information as provided from the online real-time data feeds.

2.3 SYSTEM ARCHITECTURE

This section presents the architecture for BayTripper, a real–time transit trip-planning system developed for mobile devices which incorporates data from a variety of sources. Unlike traditional schedule-based services, BayTripper integrates a user's origin obtained by GPS destination select on a map with real-time transit vehicle arrival time estimates, allowing transit riders to plan trips from their current location. The prototype system relies on three components, described in the following sections and illustrated in Figure 1:

1. Clients: Travelers using location aware mobile devices (such as GPS or cell tower based location technology).

2. Server: A routing engine which determines the K-shortest paths between an origin and destination

3. Third Party static and dynamic information providers:

- A transit vehicle arrival prediction system generating estimates from GPS equipped mass transit vehicles

- A set of static schedules from which to build the transit network graph.

2.3.1 Client Side: Mobile device implementation

BayTripper was developed on two client-side platforms: the iPhone (programmed in Objective C), and JavaScript enabled web-browsers. The two implementations feature a user-interface overlaid on top of a map, which eases the use of mass transit without requiring significant knowledge about a particular geographic area. Users select their origin and destination, either by using the mobile device's on-board GPS, tower based triangulation, or by manual entry into the device. The origin and destination points are geo-coded into latitude and longitude points and submitted as a query to the server.

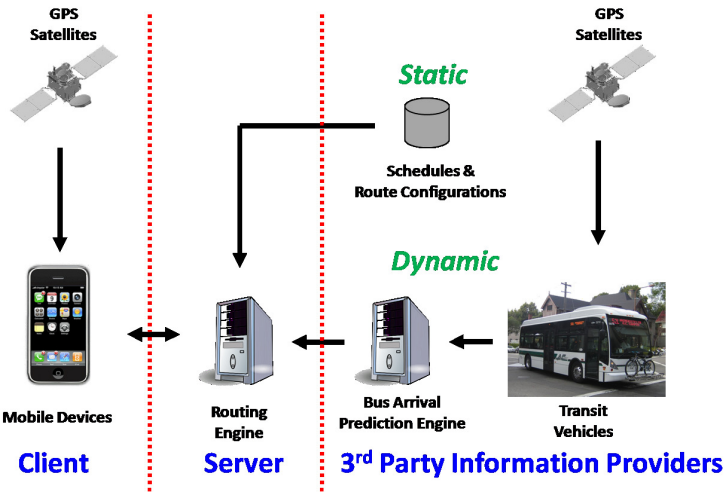

Figure 1: Architecture and system implementation of BayTripper.

Upon receiving the above request, the server creates an XML response which contains information for the five fastest routes. XML provides a simple, open, and extensible format to encode pertinent information about a user's trip, including such as walking directions, location of stops and transfers, and the duration of the trip. Although the application has been developed on two platforms, the same XML feed can be generated in response to a request from any mobile device or phone, thus making the underlying technology portable to multiple platforms.

2.3.2 Server Side: Routing Engine

The web server contains a routing engine written in Java and deployed as a web application using Java Server Pages. The routing engine determines the optimal routes by performing a database look-up on a set of feasible routes, which are generated from static schedules and route configuration information. The routing engine then communicates with external servers to obtain feeds for real-time bus arrival predictions. The construction of the optimal routes is described in detail in Section 2.4. The routing engine was implemented on a Linux version 2.6 server, with a QuadCore 64-bit processor, 1024 MB RAM, 400GB of storage, a MySQL database infrastructure, and a Jetty 6 web application server. Tests of the application server were conducted in New York City during the ITS World Congress on November 14-20th as part of the Federal Safe Trip 21 program. Over five days, approximately 1000 transit riders used the iPhone application to plan trips around Manhattan and the five Burroughs with requests processed within five seconds. Additional load testing and benchmarking of server performance revealed that 72% of that process time is due to communication with the 3rd party real-time data feed. An additional experiment of the first prototype of the system was performed and filmed in Berkeley, CA, which is available online at YouTube [22].

Figure 2: iPhone and web-browser implementation. Markers on screen indicate transfer points, and clicking on the markers or swiping the bar at the bottom of the screen reveals more information, such as the name/location of the stop and waiting times. The arrows on the top right of the touch screen can be used to toggle between different routes, while an information panel displays the total travel time for a selected route.

2.3.3 Third Party Information Providers

2.3.4 Static Information

BayTripper's routing engine requires a transit agency's static schedules and route configuration. With the growth in popularity of Google Transit, many transit agencies offer this data in the format specified by Google called the *Google Transit Feed Specification* (GTFS) [23]. The type of data needed to by the routing engine is a subset of the Google Transit Feed. Within the GTFS format, the data required for each bus stop in the network includes: names of routes which serve the bus stop, latitude, longitude, the scheduled times each bus arrives at the stop, and an identifier - typically the name of the closest intersection to give directions to transit riders. Other transit agencies store the static information in a proprietary format which is parsed into the GTFS format to create the routing engine.

The first implementation of BayTripper system was tested in three agencies in the San Francisco Bay Area, the *San Francisco Municipal Transportation Agency* (SFMTA), *AC Transit, and Bay Area Rapid Transit* (BART). For SFMTA and BART, the data was provided in the GTFS format. The *Metropolitan Transportation Commission* (MTC) provided the data for the AC Transit network, which serves Oakland, Berkeley, and other cities east of San Francisco Bay. BayTripper was also implemented for TriMet, which serves the Portland, Oregon area and publishes its static data using GTFS.

2.3.5 Dynamic Information

While static information allows for the transit network graph to be constructed, real-time information is required to update the wait and travel times between links. This real-time information is provided in the form of bus arrival predictions from external servers, which

aggregate bus data from locations of GPS equipped transit vehicles and generate predictions.

For SFMTA, roughly 1,000 buses equipped with GPS units operate 87 different routes, broadcasting their position at approximately one minute intervals to NextBus, a private company, which performs bus arrival predictions and provides the real-time estimates via an XML feed. NextBus also provides real-time data for a subset of routes run by AC Transit. Delays due to factors such as traffic are calculated by NextBus and incorporated into the bus arrival predictions that are used by the BayTripper routing engine. Some transit agencies internally operate their own real-time data feeds which are made available to the public, such as TriMet.

BayTripper is able to serve any transit agency that provides both static schedule information along with an interface to real-time bus arrival information.

2.4 ROUTING ALGORITHM

This section describes the construction of the routing engine running inside the web server described in Section 2.3.2. The routing is solved using K-shortest path techniques specific to this problem, outlined below. The algorithm determines k-shortest paths instead of a single shortest path in order for the user to decide between a set of optimal routes, possibly using personal preferences not captured by the algorithm. The set of optimal routes is constructed in two steps. First, we construct a graph representing all possible ways to go from any point in the network to any other point. The construction of the graph requires static schedules and the resulting graph is stored in a database. Next, the dynamic updates of the graphs are constructed using real-time information in the form of updates to the static graph. The routing operations on the graph utilize the user inputs (origin and destination), obtained from geo-positioning. This method was chosen to minimize computation time when responding to a real-time query while shifting the computation time to the process of pre-calculating the feasible routes in the graph with static information. In this implementation, the dynamic updates are received from a third party source - thus, the problems of finding K-shortest paths and estimation of bus arrivals are decoupled. The bus arrival estimates which update the graph are received from online feeds. They do not take uncertainty into consideration; this is a limitation of the real-time data feeds available online, which only produce a single estimate for an arrival time.

2.4.1 Static network flow framework

Using a technique similar to time-expanded graphs [24], we construct the transit network as a directed graph. In this graph, the shortest path represents the minimum time to reach a target from a given starting location. For this, we introduce the set of time-indexed vertices $\mathcal{V} \times \mathcal{T}$. A vertex $(v, t) \in \mathcal{V} \times \mathcal{T}$ corresponds to a physical location v at a given time t. Edges can thus be defined to model motion between the different vertices. Three types of edges can be constructed:

- *Waiting*. The action of waiting at vertex v from time t to time t' is encoded by an edge $e_{(v,t),(v,t')}$. This mode occurs when someone is waiting for the bus at a bus stop.

- *Walking*. The action of walking from vertex v to vertex v' starting at time t can be encoded by an edge $e_{(v,t),(v',t+d(v,v')/w)}$ where $d(v,v')$ is the distance between v and v' and w is the *walk speed*. It is assumed that the model works in integer increments, i.e. that $d(v,v')/w \in \mathbb{N}$ for all v, v'.

- *Riding*. The action of taking a transit vehicle from vertex v to vertex v' starting at time t can be encoded by an edge $e_{(v,t),(v,t+d(v,v')/r_{v,v',t})}$ where $r_{v,v',t}$ is the average *ride speed* between v and v' at time t (note that $r_{v,v',t}$ depends on t since transit buses' travel times are contingent on traffic conditions).

We introduce the cost of an edge $e_{(v,t),(v',t')}$ as $c_{(v,t),(v',t')} = t' - t$, which is the time to travel from v to v' using the edge $e_{(v,t),(v',t')}$, if this edge is allowed. We first construct a static network problem encoding published schedules as follows. Let us introduce $\mathcal{T}$ the set of times considered for the scheduling problem.

- *Waiting subgraph.* It is always possible to wait everywhere: $\forall v \in \mathcal{V}$, $\forall t < |\mathcal{T}| - 1$, assign $c_{(v,t),(v',t+1)} = 1$.

- *Walking subgraph.* For every pair (v,v') connected by a physical road walkable in time $d(v,v')/w$, assign $c_{(v,t),(v',t+d(v,v')/w)} = d(v,v')/w$ for all $t < |\mathcal{T}| - d(v,v')/w$. This encodes that it takes $d(v,v')/w$ time units to walk from v to v' and thus that this option is open any time until $|\mathcal{T}| - d(v,v')/w$ (after that time, the walk exceeds the duration of the considered period).

- *Riding subgraph.* For every pair $((v,t),\ (v', t + d(v,v')/\ r_{v,v',t}))$ of the graph connected by a transit option at time t, assign $c_{(v,t),(v',t+d(v,v')/r_{v,v',t})} = d(v,v')/r_{v,v',t}$.

The *published* transit graph, which encodes all possible options of a pedestrian starting at any vertex v of the physical graph is the union of the waiting subgraph, the walking subgraph and the riding subgraph. A path from a given origin $o \in \mathcal{V}$ at time t, (o,t) to a destination $d \in \mathcal{V}$ at time t', (d,t') on this graph corresponds to a feasible way to travel from o to d in $t' - t$ time units, and is given as a sequence of vertices $\{(o,t), \cdots, (d,t')\}$. Given an origin node $o \in \mathcal{V}$ at time t and a destination node $d \in \mathcal{V}$, the smallest t' such that a path $(o,t) \to (d,t')$ exists defines the fastest route $t' - t$ to go from o to d at time t. Assuming all static schedules are known in advance, riding edges are constructed using $d(v,v')/r_{v,v',t}$ as the cost for these edges. The construction of the shortest path for the static (published) schedule can be computed using standard dynamic programming tools to answer questions such as the shortest path problem [25], the all points shortest path problem [26, 27], and the k shortest paths problem [28]. The set of edges constructed from the static (published) schedule is denoted $\mathcal{E}_s$.

2.4.2 Dynamic network flow problem

As time t is marched in $\mathcal{T}$, transit schedules are updated based on knowledge of delays from online data feeds which run an estimation engine, predicting the arrival time of the transit vehicles.

When the knowledge of a delay appears at time t in the system, the corresponding edge of the transit vehicle in question must be updated. If the vehicle which left stop $v \in \mathcal{V}$ at $\theta \leq t$, scheduled to arrive at $v' \in \mathcal{V}$ at time $\theta' = \theta + d(v, v')/r_{v,v',\theta}$, incurs a delay t_d (known from a transit monitoring system), the edge $e_{(v,\theta),(v',\theta')}$ is removed and replaced by $e_{(v,\theta),(v',\theta'+t_d)}$ (the length of the corresponding edge is thus extended by t_d). Similarly, adjacent edges $e_{(v',\theta'),(v'',\theta'')}$ which represent the same vehicle scheduled to leave v' at θ' to a third vertex (v'', θ''), are removed and replaced by $e_{(v',\theta'+t_d),(v'',\theta''+t_d)}$ (the corresponding edge is shifted later in time).

The full dynamic graph is represented similarly to the static graph, using a union of a waiting, walking and the updated riding subgraphs. The walking and waiting subgraphs are identical to the static network; it is always possible to wait everywhere, and walking is always allowed on roads which can be physically walked. The static schedule graph is thus updated for all $\mathcal{T}$, leading to an online (time varying) set of edges denoted by $\mathcal{E}_t$, where $\mathcal{E}_t$ is revealed at time t.

2.4.3 Online shortest paths implementation

A variety of methods have been proposed for computing the solution to shortest path problems on a dynamic graph. A brute force implementation involves recomputing the shortest paths after every update, using static algorithms such as the all pairs shortest path algorithms of [26, 27], for all $t \in \mathcal{T}$. Alternatively, one can dynamically maintain the all points shortest paths on the dynamic graph as the edges are added and removed [29, 30, 31]. The efficiency gain is made by recomputing (or updating) the solution of the all pairs shortest path problem using the previous solution.

Another implementation technique leverages an offline pre-computation of feasible paths on the graph [32]. Since the feasible paths, denoted by the set $\mathcal{P}$ are precomputed, the shortest path $p_{(o,t),(d,t')}$ in the set of feasible paths $\mathcal{P}_{(o,t),(d,t')} \subseteq \mathcal{P}$ for an origin destination pair $(o, t), (d, t')$ can be computed by updating the costs on the feasible paths at the time the user query is generated. Thus, the update step is completed on all paths for a given origin destination pair, then the fastest path is returned. This implementation has two advantages for the system presented in this article. First, the main computation expense is paid once upfront. Second, when the number of updates to the graph is large relative to the number of queries, the algorithm is likely to be more efficient than algorithms which place the computational burden on the update step.

In the current system, a pre-computation method is selected. Using the static transit graph, a list of feasible paths from all pairs of vertices is computed. When a transit vehicle is delayed, the specific feasible path on the static graph becomes infeasible (because the $e_{(v,t)(v',t')}$ edge is removed), but a new set of feasible paths can be constructed using the new edge $e_{(v,t+t_d)(v',t'+t_d)}$. We wish to maintain a list of feasible paths on a dynamic graph by modifying the feasible paths on the static graph when updated edge data becomes available.

The static feasible paths are updated with dynamic information as follows. When an edge from the static graph is updated, it is replaced by a new edge. If the new edge begins at a later time t_d, and $t_d < |\mathcal{T}|$, then it is feasible to wait for the later edge. Similarly, at the destination end of the edge, if the destination vertex appears in the static feasible path, then the new dynamic path remains feasible, and the path is successfully updated. If the destination vertex is not in the static feasible path, the new dynamic path is infeasible.

Since the look up cost of the shortest paths in the pre-compute algorithm is linear in the number of feasible paths in the dynamic network, we implement two techniques to decrease the number of feasible paths which are stored after the pre-computation step. First, we remove all paths which exceed a maximal number of transfers. Second, for a fixed origin destination pair, we remove all paths whose fastest historical time is longer than another path's slowest historical time, since no series of updates will likely ever lead to the dominated path becoming the shortest path. These steps reduce the size of the database while still maintaining a list of feasible paths.

2.4.4 Routing engine summary

To implement the routing engine, we build the published graph from the transit schedules. Then, we compute all of the all pairs feasible paths on the static graph, and store them by origin destination pair. Then, for each origin destination pair, paths which have more than three transfers, or paths which under no edge update scenarios could ever be fastest are removed. The remaining edges are stored permanently. At each time $t \in \mathcal{T}$, the new edge information is obtained. The first time a given origin destination is queried at t, the set of static feasible paths and their costs are updated to find dynamic feasible paths and their associated costs. The shortest k paths are cached for other queries at t for the same origin destination pair, then returned to the user.

2.5 SYSTEM EVALUATION

In this section we present an evaluation of the BayTripper system. In order to quantify the effects of real-time data in trip planning, a schedule-based transit trip planner (TTP) was developed using the same routing algorithm as the real-time trip TTP. The two TTPs differ in two aspects, the sources of data for the wait times at bus stops, and the travel time between bus stops. The schedule-based TTP uses a static database of bus schedules, while the real-time TTP obtains estimated wait times from a real-time bus prediction engine. The real-time TTP takes a snapshot of the actual state of the system at the time the user makes a request, and uses historical data to predict the evolution of the network, and the user's trip. By comparing two nearly identical TTPs, we can examine the scenarios in which real-time data more accurately estimates travel times and more frequently predicts the optimal route.

This section is organized as follows. First, we present a description of an experimental evaluation of the system using the transit system in the San Francisco Bay Area. Next, we characterize the accuracy of the transit vehicle arrival estimates provided by NextBus, on which the real-time TTP relies. For each TTP, the accuracy of the estimated trip travel time along a fixed path is assessed by comparing the estimated travel time given by the TTP at the start of the trip to the observed travel time to complete the trip. Because of the errors in the estimated trip travel times, the TTPs do not always select the optimal (shortest

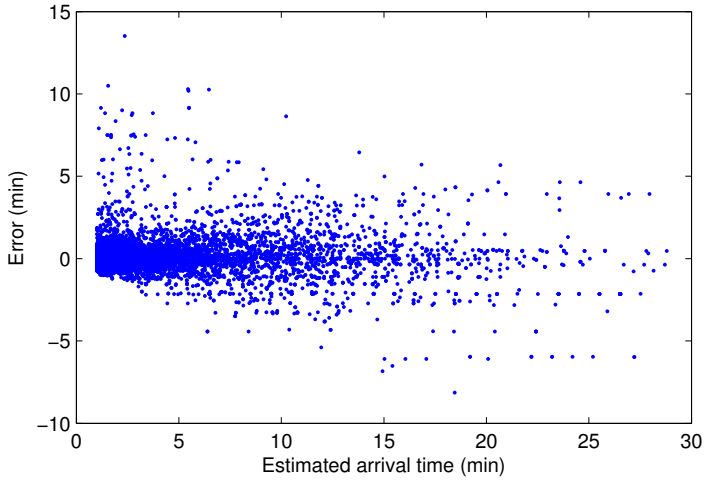

Figure 3: Estimated arrival time error. The error in minutes in the estimated arrival time for 23,349 estimates, as a function of the estimated arrival time. X-axis: estimated arrival time in minutes. Y-axis: error in minutes. Points of the negative half of the Y-axis correspond to bus arriving earlier than predicted, while points on the positive half correspond to the bus arriving later than predicted.

travel time) route. The frequency with which the optimal route is selected, and the degree of sub-optimality of each TTP when the optimal route is not returned is also assessed.

2.5.1 Experimental setting

Over six-hundred trips were planned using the schedule-based TTP and the real-time TTP using real–time data from San Francisco Municipal Transit Authority buses and light rail trains running 87 routes across the city. The trip origin and destination pairs were generated using a spatially uniform random distribution over the city. The trips were planned over the period of July 23 - 28, 2009 at various times throughout the day.

Because each transit vehicle is tracked using the on-board GPS receivers, the actual travel times along each path between the origin-destination (O-D) pairs can be determined after the trip is completed. These travel times serve as a ground truth upon which the transit vehicle arrival predictor (NextBus), the schedule-based TTP, and the real-time TTP are benchmarked. These GPS tracks also allow for the actual optimal (fastest) route to be determined between the O-D pairs.

2.5.2 Accuracy of the estimated transit vehicle arrival engine

The performance of the real-time TTP is dependent on the estimation of the transit vehicle arrival times at its upcoming stops, which is provided by NextBus. NextBus generates predictions by using historical averages of travel times between segments, calculated at different times of day. Fig. 3 shows the error of the arrival time estimate for 23,349 queries to NextBus. Each time NextBus is queried, an an arrival time is estimated, and the error is

calculated after the bus arrives at that stop.

Estimated arrival times between 0 and 10 minutes have a median error of zero, and the interquartile range (the middle 50% of estimates) tend to fall within a minute of the true travel time. However, a small number of outliers, specifically for small arrival times, can potentially cause significant challenges if the information is used to determine if a transfer between buses can be made. As the estimated arrival time increases, variance in error also increases. This variance effects the results of real-time TTP in two ways: determining the travel time of the bus to the transfer point, and determining the wait time at the transfer point for the next leg of the trip. In the set of six-hundred trips planned, there existed over four-hundred trips which required transfers over 10 minutes past the time the trip request was made.

2.5.3 Accuracy of the estimated trip travel time

The accuracy of of the travel time estimate along the paths suggested by schedule-based and real-time TTPs are compared to the ground truth travel time along the same paths. Fig 4 characterizes the amount of uncertainty in the estimated travel times, divided by the number of cases in which the planned trip required zero transfers, one transfer, or two transfers. Evaluated over all the cases, real-time data increases accuracy for travel time prediction, most noticeable in trips requiring transfers. The Wilcoxon signed-rank test was used to test for statistically significant difference between the two trip planners . The Wilcoxon signed-rank test evaluates the null hypothesis that two related samples have the same distribution. The data compared were the percentage error of the travel times returned by the schedule-based and real-time TTP, each verified against the actual travel times. The median error of the schedule-based TTP is 14.9% vs 11.7% for the real-time TTP. Statistical significance was defined as $z = 4.08$, $P < .00001$ two-tailed, thus the null hypothesis is rejected. It is concluded that the use of real-time data reduces travel time prediction error. However, the small difference between the errors show that the improvement in travel time accuracy is marginal.

2.5.4 Route selection optimality

The main functions of the TTP are to provide an expected travel time from origin to destination as well as suggest a set of routes based on the state of the transit network at the time of the user's request. The routing algorithm calculates the K-shortest paths and returns the five fastest routes to the user. This allows the user to make a decision based on the total trip time, as well as personal preferences such as walking distances and number of transfers and other variables which the application does not factor in. In this analysis done in this section, only one route predicted by each TTP - the expected optimal route determined by the shortest travel time, is evaluated. The predicted route by both TTPs is compared to the ground truth - the actual optimal route, obtained by tracking the position of multiple buses over the duration of the trips.

Table 1 summarizes the route selection optimality of the TTPs. The percentage of cases in which the two TTPs accurately predicted the actual optimal route was determined. These cases are split into three rows, when both TTPs predicted the optimal route, and when

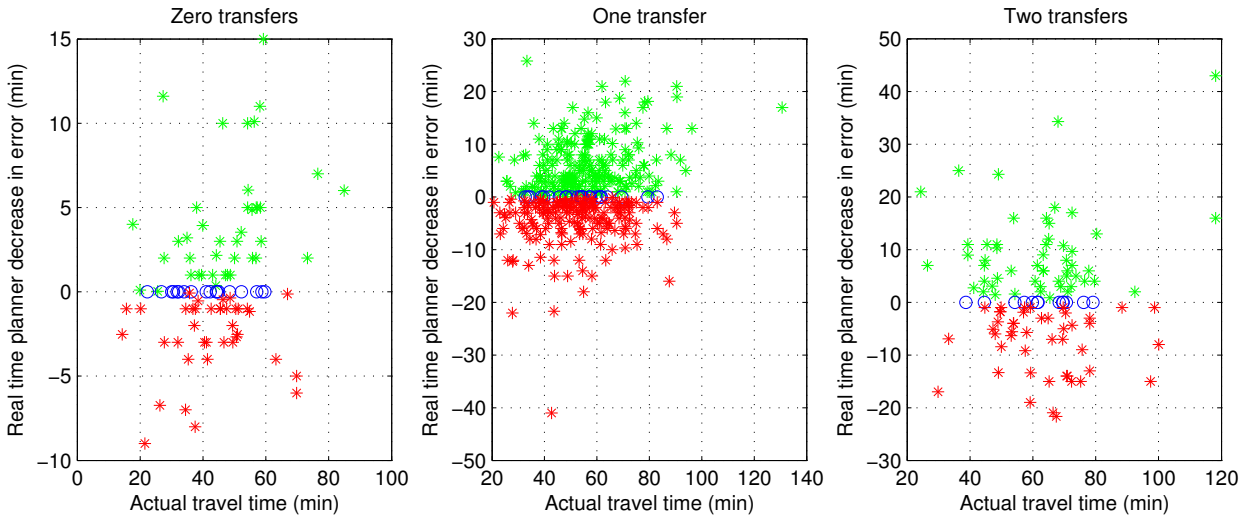

Figure 4: Effect of travel time accuracy when using real-time data. Positive values on the Y-Axis refer to cases in which real-time TTP more accurately predicted total travel time. Trips with zero transfers: 40 (41.6%) more accurate, 38 (39.6%) less accurate 18 (18.8%) same prediction. Trips with one transfer: 232 (51.3%) more accurate, 190 (42.0%) less accurate, 30 (6.6%) same prediction. Trips with two transfers: 58 (48.8%) more accurate, 47 (39.5%) less accurate, 14 (11.8%) same prediction.

only either one of the TTPs made the correct route prediction. If neither TTP predicted the actual optimal route, the actual travel times of the predicted routes were compared. The next three rows show the percentage of cases in which one of the sub-optimal routes resulted in a shorter actual travel time, or if the same sub-optimal route was predicted by both TTPs. The results are split into four scenarios: long trips (trips longer than 30 minutes), short trips (trips shorter or equal to 30 minutes), trips in which a transfer was involved, and trips which required no transfers. In most cases, the real-time TTP outperformed the schedule-based TTP by a small margin.

	Long	Short	With transfers	No transfers
Both predicted optimal	27.3	44.4	20.5	75.0
Only real-time predicted optimal	20.3	16.7	23.8	0.0
Only schedule-based predicted optimal	14.1	11.0	15.6	4.1
Neither predicted optimal: real-time faster	12.5	11.2	13.9	4.2
Neither predicted optimal: schedule-based faster	10.2	5.6	11.4	0.0
Neither predicted optimal: same prediction	15.6	11.1	14.8	16.7

Table 1: Optimality of fastest route returned by the schedule-based TTP and the real-time TTP with respect to the observed fastest route. Values represent percentage of cases.

2.5.5 Implications for further research

The study shows that the inclusion of real-time information in the BayTripper system provides two benefits over schedule-based TTPs: more accurate predictions for total trip times, and a larger number of cases in which the optimal route was suggested. However, these benefits were found to be marginal, leaving room for improvement in three areas: removal or detection of outliers as shown in Figure 3, improved long-term estimation of buses arriving in over 15 minutes, and incorporation of uncertainty into bus arrival estimates.

Currently, BayTripper is dependent on third party data sources to perform bus arrival estimation, which is based on time-of-day historical averages of segment trip times. As described in Section 2.2, newer estimation algorithms exist which take into consideration factors such as real-time traffic, but have not been tested in the context of a trip planning application. As shown in Figure 3, NextBus accurately predicts bus arrivals when the expected arrival time is under 15 minutes, with the exception of outliers in 1-5% of cases. However, the incorporation of real-time data into a trip planner requires estimation of bus arrivals at transfer points, which often occur greater than 30 minutes from the time the trip is initially planned. In these problems, the current estimation engine is less accurate in predicting bus arrivals, leading to missed transfers and large discrepancies in estimated trip times and optimality of suggested routes. Further studies will be done to incorporate more advanced bus arrival estimation techniques into the BayTripper application, while characterizing uncertainty into the predictions to compensate for potential missed transfers.

In addition, the experiment examined the ability of the TTP to determine the optimal route based on information only at a single time instant prior to the trip. Table 1showed that the real-time TTP performed better in shorter trips. Larger errors in estimation for the total time of longer trips, and trips which include transfers, lead to greater number inaccuracies in the suggestion of optimal routes. The conclusion from these results is that accuracy drops with respect to the amount of time into the future the TTP must predict the state of the system. A solution to this problem is to allow for the algorithm to re-calculate the expected optimal route after the trip has been planned, allowing a user to receive updated information about their route as the state of the transit system evolves. As more recent information becomes available, users will receive new estimates for their route travel time. It is also possible that the prediction for the optimal route may change after the user has reached a transfer point, or even while riding the bus. Further evaluation of the system includes re-calculating the optimal route at different stages of a trip, and examining the accuracy of the total travel time at these stages.

2.6 CONCLUSION

This chapter described an implementation of a transit trip planning system, which includes an application developed for mobile phones, a routing engine running a K-shortest paths algorithm, and interfaces to bus arrival prediction engines provided by third parties. It was discovered through over six-hundred experiments in San Francisco that the real-time data provided marginal improvements to current schedule-based trip planners. The study showed that there is potential for much improvement in transit trip planning by using real-time data in conjunction with more advanced estimation techniques which focus on solving long term

estimates for bus arrivals, and characterizing uncertainty into the predictions. The growth of geo-positioning systems in transportation and the availability of real-time data feeds from transit agencies and other transportation modes is promising, as more opportunities for research are opened in the field of real-time transit, and multi-modal navigation.

3 Algorithm for finding optimal paths in a public transit network with real-time data

3.1 INTRODUCTION

The focus of the chapter is the introduction of a new algorithm which overcomes the problem of computing shortest paths in a transit network which pulls real-time data from a third-party Application Programming Interface (API). Over a web-based or smartphone interface, a user enters a geo-coded origin and destination (O-D), and the algorithm must respond by returning K-shortest paths based on the real-time state of the transit network. Thus, our principal performance measure is the time elapsing between input of the O-D and output of the path, which we call the path computation time. Web usage studies show this time should be less than 7 seconds [33, 34].

Paths cannot be wholly pre-computed since the aim is to return results based on real-time state, which needs to be acquired at run-time. We call the time required to get this information the real-time state acquisition time. Industry Service Level Agreements (SLAs) create constraints on how this real-time state can be acquired. The industry appears to be evolving towards an open model where the public agencies are making real-time bus data available on the web, allowing third party developers to use this data to provide transit information via web-based services or native applications on smartphones or over the web. The real-time data needs to be consumed through web-based APIs that limit the amount of data per request to the transit agency web server and the number of requests per second. For example, NextBus, a provider of real-time data in 61 cities, limits real-time data to 300 stops per request.

Data analyzed for this chapter showed that in most transit networks, running a common K-shortest path algorithm after obtaining predictions for all transit stops in a network is not feasible. For example, the Washington DC transit network has 28,627 stops, which takes an average of 75 seconds to acquire the real-time data for the entire network. Moreover, the 75 seconds does not account for the request rate constraint, 1 request every 10 seconds, imposed by the data provider[35]. It takes 96 requests to get all the Washington DC data, which adds at least another 9600 seconds to the time required to acquire the data for the entire network. This time is long enough for at least some portion of the data to become obsolete by the time the data for the entire network has been acquired. The transit routing algorithm in this chapter needs to acquire its real-time data and compute with the K-shortest paths in less than 7 seconds. Thus, we need an algorithm able to route each trip by acquiring the real-time data for only some small part of the total network.

Fortunately part of the literature is helpful. The obvious approach might be to model the transit network as a graph, albeit time expanded [36, 37, 38, 39] or time dependent [40, 41, 42, 43], and run one of the widely used shortest path algorithms on it [25, 44, 45]. There has been research on speed up techniques [46, 47, 48] to Dijkstra's algorithm as well as heuristic algorithms developed [49, 50, 51, 52]. The literature also covers the K-shortest path [53, 54, 55, 56] and reasonable (multi-objective) path computation [57] problem. This literature helps with schedule based transit routing, i.e, when the edge costs of the graph are single numbers known a priori. The purpose of the discussion in this paragraph is to show

the real-time transit routing problem is not easily reduced to this case. These algorithms require that all link costs (travel time) in the network are known at all times. However, this time required to acquire the edge costs for the network graph from the transit agency web servers is the major bottleneck due to the software architecture of implemented transit information systems today. Thus, the run-time of traditional K-shortest path algorithms in this problem is extremely slow.

There is also literature on routing in transit networks formulated as shortest path problems in stochastic and dynamic networks [58, 59, 60, 61, 62]. This literature helps consider the reliability of travel time as well or other measures of robustness. These methods could be blended to improve the algorithm in this chapter to optimize higher moments of travel time. In this chapter we do not leverage this literature.

Feder [63] addresses the routing problem when edge costs are approximately known but can be made more precise at run-time at a cost, a parallel to the problem of determining the wait and travel times at bus stops by accessing a real-time API. Likewise, Pallottino developed an algorithm for transit graphs where the edge costs are known, but subject to small changes (i.e. updates in real-time information) [64]. Though this chapter is about a problem similar to the ones in these papers, we use a different solution technique, one leveraging off-line pre-computation in a manner similar to Bast, Sanders, and Schultes[65, 66], for routing in road networks.

Through 770,000 simulations covering transit networks in 77 US cities, we find this approach is able to keep the real-time data acquisition time under 1.5 seconds and the total path computation time less than 3 seconds for 99% of the O-D requests.

The structure of the chapter is as follows. Sections 3.1.1 and 3.1.2 provide further information on open transit data and the routing methods of Bast et al, respectively. Section 3.2 describes our algorithm. Section 3.3 is the evaluation. We evaluate the algorithm by doing 10,000 simulations of trip requests for each transit network in 77 cities. Worst-case bounds on path computation time, real-time data acquisition time, off-line pre-computation time, and run-time memory requirements, are based on all 770,000 simulations. We measure the time required to service it by actually executing the request that communicates over the Internet in real-time to the transit agency data server for the city relevant to the O-D. Since execution times are machine dependent measures, we also provide histograms showing machine independent measures determining the complexity of our algorithm. The route computation time depends on the size of a database of paths produced by pre-computation plus the real-time data acquisition time. The real-time data acquisition time depends on the number of bus stops per transit trip request. We present histograms on these measures for Washington DC. The DC transit network turns out to be the most complex of the 77 cities, including Los Angeles, Chicago, and San Francisco. Section 3.4 concludes the chapter.

3.1.1 Recent Trends of Open Data

The development of a real-time transit routing algorithm has been spurred by two trends in public transportation technology: open data, and real-time tracking of buses.

Starting in 2007, a growing number of transit agencies have packaged their route configuration and schedule data into a format called the Google Transit Feed Specification (GTFS)[23] in order to have trips planned by Google Transit. The benefit of this was the in-

tegration of public transit into online trip-planners, but more importantly, the release of the GTFS data by transit agencies has allowed for a plethora of innovative apps being created by third party and independent developers. For example, an open-source multi-modal trip planner, Graphserver, was developed and has been embraced by the online transit developer community. Last year, our research group introduced a system for real-time transit trip planning on mobile phones called BayTripper. The potential travel time benefits for incorporating real-time data into travel time predictions were evaluated[67, 68]. As of July 2010, BayTripper is the only real-time transit trip planning application released and available to the public.

The second trend is the use of GPS to track buses, not only for transit agencies to monitor their fleet, but to open the data up for users to code with. Pioneered by TriMet in Portland Oregon, transit agencies have begun to not only release route configuration and schedule data, but also real-time feeds of the location and expected arrival time of their transit vehicles. Transit agencies throughout the United States have slowly been opening data to the public. As of July 2010, there exist 787 transit agencies in the United States, 107 of which provide open data and a small fraction those agencies provide open real-time data. This number has been growing, as has research involving the value of real time data[69].

3.1.2 Relation to Transit Node Routing in Road Networks

Although there has been an enormous body of work on routing in public transit networks, the algorithm we developed takes cues from the work of Bast, Sanders, and Schultes, who developed an innovative algorithm for routing in road networks. Their algorithm, called Transit Node routing is currently the fastest static routing technique available [65, 66]. The main observation of theirs was something intuitively used by humans: When you drive to somewhere far away, you will leave your current location via one of only a few access routes to a relatively small set of transit nodes interconnected by a sparse network relevant for long-distance travel. Likewise, in public transit networks, when you travel to a destination that is not served by a bus/train route near your origin, you will look at a set of potential transfer stations to change buses. Transit Node Routing precomputes distance tables for important (transit) nodes and all relevant connections between the remaining nodes and the transit nodes. As a result, the difficult problem of finding shortest paths on extremely large road networks is "almost" reduced to about 100 table lookups. This lookup table only stores data for a certain subset of points to speed up the performance of the algorithm, which is a major innovation. The same strategy is used in the real-time transit routing algorithm presented in Section 3.2. We calculate a set of feasible paths for only a subset of nodes in the transit network and store the results in a lookup table.

3.2 ALGORITHM IMPLEMENTATION

As described in Section 3.1.2, the algorithm takes cues from Transit Node Routing. The Transit Node Routing algorithm has a pre-computation step to store distances between important nodes. The nontrivial concept is defining 'important' nodes such that optimal routes are found. In our routing algorithm, the 'important' nodes to perform the pre-computation are defined by the following statements about routing in transit network:

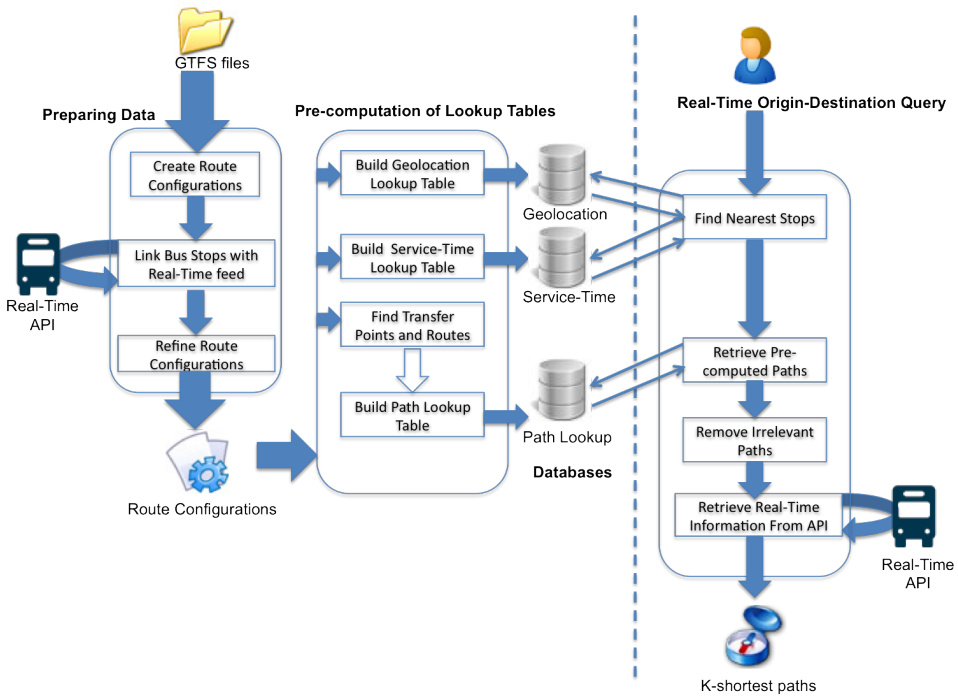

Figure 5: Flowchart of the algorithm described in Sections 3.2.1, 3.2.5, and 3.2.10.

1. If the origin and destination are not served by the same bus route, humans intuitively plan trips by finding transfer points to connect between bus routes.

2. The set of feasible paths from any bus stop along Route X to any bus stop along Route Y is a subset of all feasible paths from the origin station of Route X to the terminus station of Route Y.

Statement 2 defines the important nodes that are precomputed; it is only necessary from the origin node of each bus line to the terminus of every other bus line to create a lookup table the size of the number of total bus routes squared. Statement 1 defines the process by which the pre-computation is done: storing transfer points to build a sequence of locations of where to transfer to reach the terminus of the bus line. In this chapter, a feasible path is defined as any path that can be traversed by taking a series of buses between an O-D with a maximum of four transfers. The constraint of four transfers puts a cap on the number of paths needed to be precomputed, and prior research has shown that the accuracy of real-time data deteriorates significantly when predicting arrival times over 20 minutes into the future, well over the time a fourth transfer would be made [68]. Any O-D query that requires more than four transfers cannot be computed by the algorithm.

The routing algorithm is broken up into three steps: preparing the schedule data received from third-party providers, building a lookup table of feasible paths from a subset of O-D pairs, and dynamically performing the real-time calculation of the K-shortest paths using the lookup table. These steps are shown in a flowchart in Figure 5.

Figure 6: Example transit network structure built from open transit data with no link travel times defined.

3.2.1 Preparing Data

Software was built to convert schedule data to build a graph of the network with no link travel time data as shown in Figure 6. Link travel time data is only known at real-time by accessing an API over the internet. Although the amount of data that is needed to be transferred to deliver all real-time predictions for all bus stops in an entire network is not large, today's API's are not designed to do this. Until the groups that perform the real-time arrival predictions create a standard for this, this chapter presents a solution that is implementable immediately to perform routing with real-time data.

3.2.2 Convert Open Transit Data To Route Configurations

Transit data is read into the system from two different sources, Google, and NextBus.com. Google has established a standard for encoding transit data, GTFS, which can be converted into route configuration files that store the following relevant information in the structure specified in Figure 7, the route name, direction name, stop / intersection name, latitude, longitude, stop sequence, and agency name.

A route configuration is categorized by a route, direction, and agency name. Each configuration specifies a set of stop names with corresponding latitude and longitude points in the sequence which the bus traverses the route. Configurations with the same route, direction, and agency name are grouped together.

3.2.3 Link Bus Stop To Real-Time Feed

GTFS does not have a specification for a real-time data format, thus, each stop must be

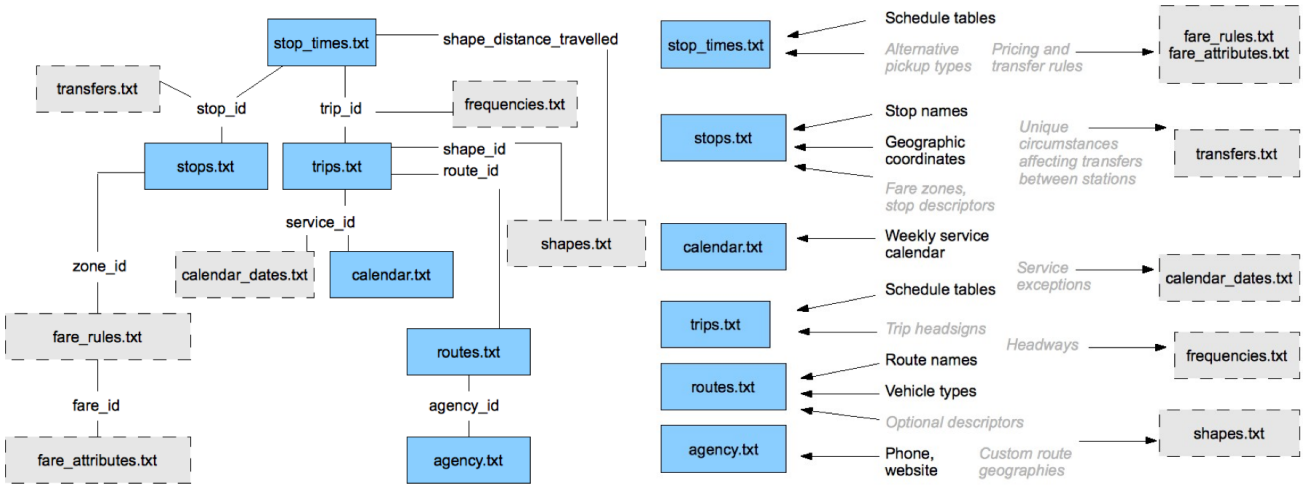

Figure 7: GTFS file structure. Optional Files in dashed boxes, Required files in solid boxes (a) Files and linking columns (b) Properties of files needed to be extracted

linked to a real-time URL by an unique stopcode. This assignment is done by matching the route configurations with the real-time feeds provided by Nextbus.com and Portland TriMet. Real time predictions are accessed through HTTP by specifying the agency name, bus route, direction, and stopcode.

Not all transit agencies that have released open data in the GTFS format are served by NextBus.com, however any future real-time feeds provided by those agencies can be integrated into the algorithm.

3.2.4 Refine Route Configurations

Most train routes only run in two opposing directions, however there are numerous cases in which bus routes have different route configurations but run under the same route name. The transit routes are broken up into segments as described in Figure 8 such that each unique segment defines a route configuration that all buses follow. If the set of stations in a configuration that a bus follows is a subset of another configuration of stops, the two are considered the same. Whether or not a bus stops at every station is determined at run-time.

3.2.5 Pre-computation of Lookup Tables

The pre-computation step generates three tables, a path lookup table, a bus service-time table, and a geolocation table, which are defined in the following sections.

3.2.6 Find Transfer Points and Routes

The entire set of stops is sorted by either latitude or longitude, and stops within a reasonable distance are stored as a potential transfer points between the two routes. In the implementation, the reasonable transfer distance is arbitrarily chosen at half a mile. Previous research has shown that transit riders' preference for walking between transfers varies quite a bit, and

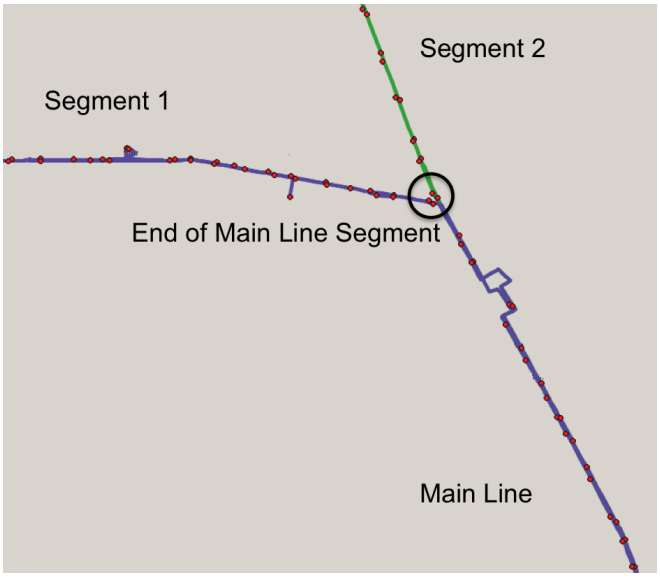

Figure 8: Example of a bus route with two termini. This particular route is broken up into three segments.

is dependent on many variables, including land use, open space, and topographical features [70]. In a future implementation, this transfer distance can be varied to account for these variables.

Bus routes are often designed to share common stations, such as in a trunk-line configuration. In these route configurations, there exist a large number of potential transfer points between lines, which increases the size of the lookup table linearly. To address this scenario, all transfer points between the two lines are removed except for the optimal point. This optimal point is defined by the station with the lowest probability of missing a transfer as specified by the GTFS schedule.

3.2.7 Build Path Lookup Table

A set of feasible paths is built from the origin stop of every bus route to the terminus of every bus route in each direction and stored in a path lookup table. The set of bus routes is iterated through to create a list of reachable routes in one to four transfers. For each O-D pair, the transfer points corresponding to each of the list of reachable routes are stored in the path lookup table. Paths that require more than four transfers will not be found, and thus not stored in the database.

The size of this database grows polynomially with the number of bus routes in the transit network, as shown in Figure 11. The potentially infinite size of this database is limited in two ways: by setting the number of possible transfers at four transfers, and excluding paths that take two transfers more than any existing feasible paths between an O-D. The second limitation measure is based on an analysis of GTFS schedule data, which shows that a direct route between two points is always faster than a route with two additional connections with

no waiting time at transfer points.

3.2.8 Build Service-Time Lookup Table

The lookup table includes paths from all bus routes in the transit agency's network. However, not all bus routes are in service at the same time: there are often nighttime or rush hour only routes being run. The final step in the pre-computation of routes builds a "service-time" table of hours in which each bus route is in service. In real-time, the bus service table used to remove any paths which include a bus route not in service.

3.2.9 Build Geolocation Lookup Table

All bus stops from the route configuration files are aggregated and a geolocation table is created to store the set of all transit stops in the entire network, indexed by their latitude and longitude.

3.2.10 Real-Time Queries

The following section describes the algorithm when it responds to real-time queries from users. A query is a request for the shortest path between an O-D, specified by a latitude and longitude point. The process is very simple: a set of paths is retrieved from the lookup table, the travel times are accessed from the real-time API, and the total travel times for each path are calculated and sorted. Thus, in real time, the computational complexity of this algorithm is linear in the number of paths retrieved from the lookup table.

3.2.11 Find Nearest Stops

Transit routes in a half-mile walking radius of both the user's origin and desired destination are looked up from the geolocation table. For each route, the stop nearest to the user is saved, and all other stops are not considered. If no stops are within this distance, the search radius is increased until a bus route is found.

3.2.12 Retrieve precomputed paths

Precomputed paths are retrieved for each O-D pair of bus routes retrieved from the geolocation table. When the query is made in real-time the set of transfer points is sent to the real time API to retrieve bus arrival predictions at each node. Every time a query is made by a user, new real-time arrival predictions are sent to the real-time API, retrieving the latest data for each user request. Thus, the algorithm instantly handles changes in bus arrival predictions: even in major situations such as a bus breaking down, or multiple buses bunching together, such that the wait time at any bus stop is greater than the specified headway.

3.2.13 Remove irrelevant paths

As described in Section 2.2, the precomputed lookup table contains the superset of all feasible paths from the origin station of every bus route to the terminus station of every bus route. Therefore, not all paths retrieved from the path lookup table are physically possible to be

made and are thus ignored. In addition, paths which include a bus route not in service are removed.

3.2.14 Retrieve Real-Time Information From API

The set of feasible paths contains information for all bus stops to board, get off, and transfer. These stops are all bundled into a HTTP request to the real-time API. At this point, the algorithm is dependent on the speed at which the third-party server can process the bus arrival predictions and internet traffic between the third party server and the localhost.

The total time of each path is calculated, all paths are sorted and returned to the user.

3.3 PERFORMANCE AND RESULTS

The real-time routing algorithm was tested with transit data from 77 different cities. The data was obtained from GTFS Data Exchange [71], an online aggregator of GTFS files, NextBus [72], and Portland TriMet [73]. For each transit agency's network 10,000 simulated queries for real-time paths were made. The 10,000 different O-D points were randomly drawn from the set of bus stops served by each agency. As described in Section 3.2.11, the K-shortest-paths algorithm computes paths to all bus stops within a half-mile walking distance of the random O-D. In the implementation of the iPhone application, only the five shortest paths are displayed for each request in order to provide users with a variety of options to reach their destination.

The route computation time depends on the size of a database of paths produced by pre-computation plus the real-time data acquisition time. The real-time data acquisition time depends on the number of bus stops per transit trip request. The tradeoff of using the real-time routing algorithm is the time to precompute the lookup table and size of the table, versus the reduced time to respond to queries in real-time. To evaluate this trade off, we collected data for four metrics:

1. Number of paths. We show that the algorithm searches through a set of paths that is tractable at runtime, and even in the worse case scenario it responds to queries in a timely manner.

2. Number of requests sent to real-time API. We show that our algorithm makes a small number of queries to the real-time API for all networks analyzed.

3. Number of total paths stored in precomputed lookup table. We show that the size of the lookup table is small enough to be loaded into memory to run on a server on even a home computer.

4. Time to generate precompute lookup table. We show that the time to compute the lookup table is small: under 1 minute for most networks, with a worst-case of 100 minutes.

3.3.1 Implementation Details

The simulated queries were run on an Amazon EC2 server located in Northern California. Amazon EC2 is a cloud computing service that allows users to rent virtual machines called "instances". The algorithm is running on the smallest instance available, which provides 1.7 GB memory, 1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit), 160 GB instance storage, on a 32-bit platform at a cost of 4 cents per hour. In addition to the tests run for the purpose of this chapter, the server is used daily 350 times daily by 600 active users as of July 2010 via the BayTripper iPhone application. The EC2 server serves requests from the iPhone application and the web for real-time routing on the BART and MUNI systems in the San Francisco Bay Area.

3.3.2 Real-Time Algorithm Performance

In the following section we present histograms of metrics 1 and 2: the number of paths and requests to the real-time arrival API. We also focus on discussing two worst-case scenarios found in all 770,000 simulated queries, the O-D pair that requires computation of travel times amongst the largest number of paths, and another scenario that requires the largest number of hits to the real-time API. In both the worst-case scenarios the algorithm computes the K-shortest paths problem in less than 3 seconds, while taking 1 second to query the real-time API in 99% of cases. The speed of the algorithm is affected by the number of precomputed paths retrieved from the lookup table. The larger number of paths, the more estimates for wait and travel times are needed to be retrieved from the real-time API, the bottleneck in the system.

Both of the worst-case scenarios were found in Washington D.C., the agency with the largest number of bus stops and routes served that we tested. The worst-case scenario for number of feasible paths retrieved from the precomputed lookup table is 10,127. The worst-case scenario for number of total hits to the real-time API was 127. Figure 9 shows a histogram of the number paths computed in real-time, generated from 10,000 simulated queries. Washington D.C. is presented because to demonstrate that even in the worst-case the algorithm computes in under 3 seconds.

The performance of the system is a function of the real-time API and constraints placed on accessing the API. In current real-time APIs the number of stations able to be queried in a HTTP request is limited to 300 bus stops per request, and 10 stops per bus route. Additionally, consecutive requests cannot be made more often than every 10 seconds. Thus, as the number of hits required to the API increases, the response time grows approximately as a step function. The worst-case scenario for number of queries to the real-time API was 127, in Washington D.C. Figure 10 shows a histogram of the number of hits to the real-time API for the simulated queries in Washington D.C. The number of bus stops in the worst case scenario, 127, can be made in a single request and fits well within the limitation of the real-time API.

The real-time algorithm makes a new request for bus arrival estimates to the real-time API every time a user request is made, ensure the most up to date estimates are used in the algorithm. As a comparison, the total number of stops in the network is 28,627; the time

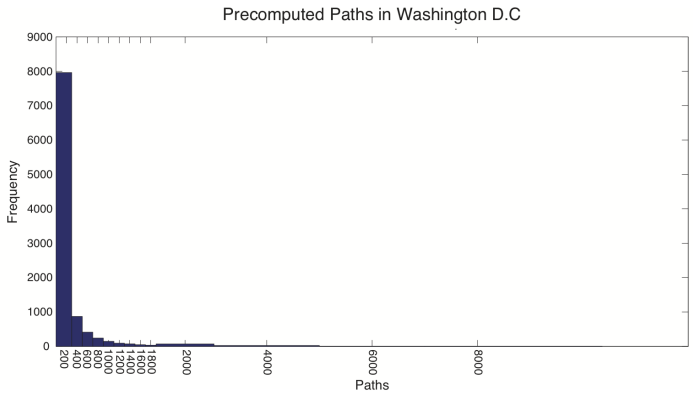

Figure 9: Histogram of number of paths searched Washington D.C.. 99% Percentile: 1900 paths. 80% Percentile: 304 paths. 50% Percentile: 224 paths. X-Axis: Number of precomputed paths which are computed in real-time. Y-Axis: Frequency of occurrences

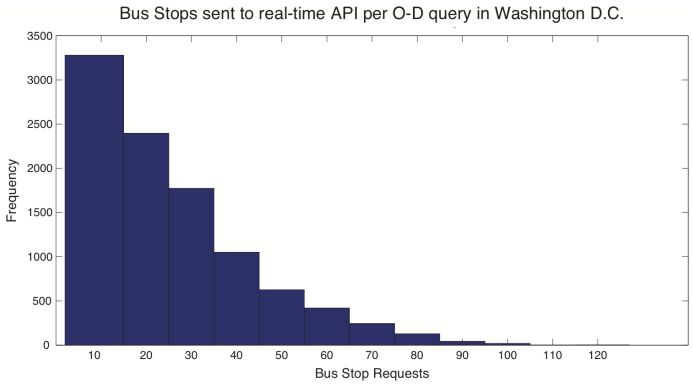

Figure 10: Histogram of number of bus stops sent to real-time API per O-D query in Washington D.C. 99% Percentile: 82 hits. 80% Percentile: 40 hits. 50% Percentile: 26 hits. X-Axis: Number of hits to real-time API. Y-Axis: Frequency of occurrences

to query this is 75 seconds. The alternative of querying every route in the system for this network and running Dijkstra's algorithm on the transit graph is infeasible, as bus positions update more frequently than every 75 seconds.

3.3.3 Pre-computation Performance

In the prior section we demonstrate that the real-time performance of the algorithm returns results in 3 seconds in the worst-case, with 1 second being spent with the real-time API query, versus 75 seconds to query the entire network. This performance comes at a cost of having to precompute a lookup table.

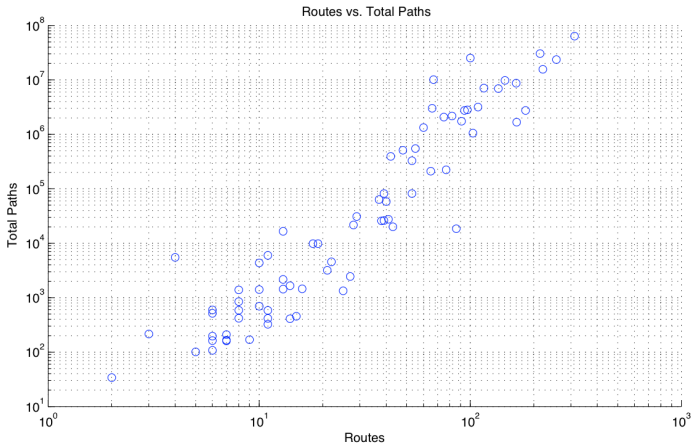

Figure 11: Scatterplot of lookup table size for all transit agencies. Y-Axis: Size of path lookup-table generated for transit agency X-Axis: Number of routes run by transit agency

3.3.4 Memory Usage

Each path takes on average 20 bytes to store. The size of the path lookup table that has to be loaded to memory ranges from .664KB to 1.2GB for the largest transit network we analyzed. The geolocation and bus service tables range from 8KB to 2MB.

As shown in Figure 11, the size of the lookup table increases polynomially as the number of routes served by a transit agency grows.

3.3.5 Pre-computation Time

The largest region we analyzed, Washington D.C. took 99 minutes to precompute 63,308,845 paths, while 70% of all other cities took under 1 minute to complete. Every time a transit agency makes a route configuration change, the lookup table must be re-computed.

3.3.6 Sensitivity Analysis

The number of paths searched through at runtime is dependent on the distance a person is willing to walk to his origin, and from the last bus stop to his destination. Although the previous simulations were made with a fixed half-mile distance, the distance can be varied per the user's request. Figure 12 shows a histogram of the number of bus stops sent to the real-time API for three different walking distances, .5 miles, .75 miles, and 1 mile. With a 1 mile walking radius, the number of hits to the real-time API increases, but never exceeds 140 requests. The response time from the server for 140 requests still is 1 second on average, thus there is a negligible increase in the time to compute the K-shortest-paths with this walking distance.

Additionally, the number of paths searched at runtime is dependent on an even greater

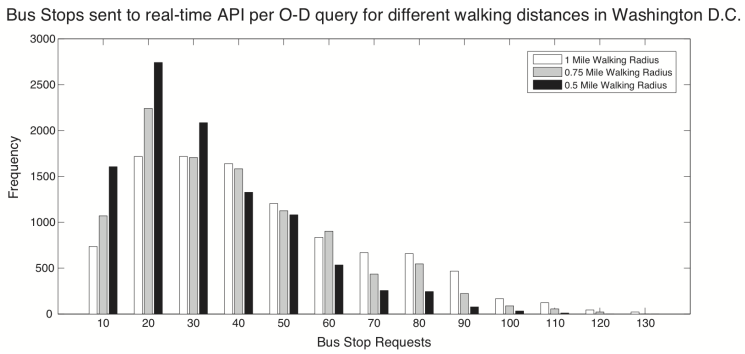

Figure 12: Histogram of number of bus stops sent to real-time API per O-D query in Washington D.C. for three different walking distances X-Axis: Number of hits to real-time API. Y-Axis: Frequency of occurrences

variety of specific features of transit agencies and metropolitan regions, such as number of bus stops served, number of transfer stations, as well as geographic locations and spatial distribution of these stops. These factors affect the following performance metrics of the algorithm in ways which have not been fully explored, but are important to consider for future implementations of the algorithm in larger-scale metropolitan transit networks than Washington D.C.

3.4 CONCLUSION

This chapter presented a new algorithm for calculating K-shortest paths in a transit network with real-time data. We describe that the modeling of this particular shortest path problem is a result of the state of technology used by transit agencies to disseminate information today. It was shown that pre-computation of a lookup table for paths between certain transit stops significantly reduces the run-time speed of responding to user requests. Transit data in 77 cities was precomputed into lookup tables, and 10,000 simulated user requests for each transit agency were made, showing that in the worse case scenario, the algorithm computes K-shortest paths in 3 seconds. Based on these results, additional testing is proposed to investigate the performance of the algorithm in larger regional transportation areas which include multiple transit agencies and intercity networks.

4 Overcoming battery life problems of smartphones when creating automated travel diaries

4.1 ADVANCEMENTS IN AUTOMATED TRIP DIARIES WITH SMARTPHONES AND GPS

Collecting data on the way people travel has been a longstanding research problem; the information is incredibly valuable, can be applied in a variety of industries and research fields, but obtaining a rich dataset is not simple. In the transportation community, work on travel survey innovation has been researched for years, and today behavior modification programs such as travel feedback programs is a growing research topic.

Travel surveys and data collection are essential for travel behavioral studies, which are used to understand individual travel behavior, collecting socio-economic and demographic data, household and vehicle data, alongside travel diaries listing all trips the individual made in a day. The data is used to estimate transportation planning models, which forecast traffic, land use patterns, changes to the transportation infrastructure, and policies. The problem with the current state of practice is that household travel surveys are expensive and do not capture people's lifestyles[74]. These traditional data collection methods with paper surveys are expensive and require survey subjects to constantly record data, disrupting their normal behavior. Therefore, these surveys are typically only carried out with small sample sizes, at intervals of $10 - 15$ years, covering $1 - 2$ days of travel. With these sparse amounts of data, models are derived for 20 to 25 year forecasts of both transportation and land use conditions, making strong assumptions to cover for the lack of long-term data (e.g., that work location follows residential location choice). As an example, the San Francisco Bay Area travel survey (BATS) was last conducted in 2000 and only covers two days per study participant in 15,000 households and is used for long-range transportation planning in the entire Bay Area. Researchers have recognized that this style of data collection: running 48-hour surveys with a large sample population does not provide as much value as travel diaries over longer survey periods but with small sample sizes[75]. Because of these difficulties of running surveys, there is a limited amount of data available for travel behavior modeling. It has been suggested for some time that addressing the data collection pain-point is the role that such technology can play [76, 77] .

For decades, researchers have been using GPS technology to supplement travel diaries, and have shown the benefits of using GPS location data to create travel surveys as a result of the increased accuracy. Many early studies focused on using independent GPS devices [78, 79] and validating conventional diary surveys with these devices. These studies started with participants in the range of 100 persons and have expanded to as many as 1500 with the French National Travel Survey and 2200 households in Halifax [80].

Starting in 2007, researchers began looking at smartphones as a potential tool to perform travel survey studies. There are many advantages of using a smartphone versus a separate GPS device. First, the smartphone is a ubiquitous device used by over 200 million Americans. Of the 450 million mobile phones used everyday, 46.8% are smartphones [81]. This number is expected to grow to 70% by 2015. Thus, running a large-scale travel survey is much more feasible and affordable for researchers and local governments. Another advantage is

the distribution of travel surveys - instead of GPS devices that have to be sent to people, installing an application on a smartphone is quick and cheap, and more importantly it is a concept that people are already accustomed with.

Unfortunately, smartphones cannot collect location data in the same way wearable GPS devices collect data because they were not designed for that purpose. Smartphones cannot constantly collect location data due to the power usage of location based services and the limited battery capacity of phones; a battery will be drained in 5 hours if the phone's GPS sensor continuously collects location points[82]. Between running apps, surfing the internet, writing emails, and talking on the phone, there is a limited power budget that a smartphone can use for the purpose of collecting location data for travel studies. On top of that, people are very concerned about the battery life of their phones, are aware of the power consumption of their apps, and actively manage applications running on their phone to minimize battery drain [83]. People have regular charging patterns, and it is the opinion of the authors that any system for creating automated travel diaries must do so in a battery efficient manner that does not change the user's behavior and interaction with their smartphone.

There is a solution: the battery usage of phones can be managed by sampling various sensors on the phone very infrequently. By doing this, it is possible to collect a small amount of location points without draining the user's battery, and still recreate trips using as little data as possible from various sensors: GPS, WiFi radio, and accelerometer. With sparse data, determining trip attributes, even as simple as the start time and location of trips become more complex.

Thus, the challenges of building a travel behavior data collection system is thus two-fold. First, developing an application which gathers location data in a battery efficient manner, and second, developing algorithms to translate sparse location data into accurate trips. Fortunately, there exists a large community of researchers focused on solving the location tracking problem in a battery efficient manner. In fact, many of these researchers had already nearly solved the problem of collecting location data that can be applied for the purpose of generating trip diaries.

Using the ideas developed by this community, a smartphone app was developed to track a user's movements and locations. The smartphone app sends data to a server which runs a trip determination algorithm that outputs the following details of a person's trips taken daily: trip start/end times, start/end locations, route taken, and travel mode used. The trip data is displayed on a web site or can be downloaded to analyze and run travel models on.

In theory, solving this automated travel diary problem is not complex - if phones behave as expected and provide sparse, yet accurate data, and if users never shut off their phones or turn off location services, this research simply combines the work of the battery research community with the research done to determine travel mode on smartphones. However, in practice this is not the case. The number of external variables: types of phones, geographic locations, topographies, user behaviors, users' phone settings, and many more, require a robust solution to the problem.

To learn about these externalities, a lengthy evaluation of the system was conducted, with 125 users providing data on 21 different phones in 20 different states featuring many types of topographies - amounting to a total of over 120,000 user-hours of data recorded. The data provided insight into the issues of various phones, the effect of different types of geographies on the accuracy of data, and smartphone habits of users which affected the

quality of the location data. In addition, the quality of the trip determination system was evaluated: a set of metrics is proposed as the standard to evaluate any automated travel diary system, and a detailed evaluation of 1850 trips made over 3 months was conducted, consisting of a thorough daily review of trips made.

The chapter is outlined as follows: Section 4.1.1 gives a history of travel studies using GPS devices. Recent work in the field of travel mode determination on smartphones is described in Section 4.1.2. The vast field of research of battery usage in smartphones is discussed in Section 4.2. Section 4.3 describes the architecture of the travel diary system, and the algorithms which process the location data into trips made are described in Section 4.4. The evaluation of the system is described in Section 4.5 and Section 4.6 outlines the research areas needed to take the project further.

4.1.1 Trip Determination with GPS devices

Starting in the late 1990s, researchers began incorporating GPS devices into travel diary studies[84]. Transportation researchers recognized that pen and paper travel surveys were not providing accurate enough data in an efficient manner[85], and attempted to use new technologies to improve the state of tracking individual travel behavior.

The first GPS enhanced travel diary studies started with tracking driving movements only. A proof of concept in-vehicle GPS logger was first demonstrated by Wagner [86, 87] in 100 household vehicles in Lexington, KY. In this study, any many subsequent studies, a personal digital assistant (PDA) was used as the data recorder. Respondents were asked to enter such information as the identity of the driver of the vehicle, the purpose of the trip, and the identity of other persons accompanying the driver. Further studies were conducted with subjects in the range of 100-200 participants [88, 84]. The GPS devices showed many flaws in the traditional methods of collecting travel data: the standard trip-based CATI survey conducted in the US under-reports travel by about 20–25% [89], short trips lasting 10 minutes or less are likely to be unreported [90].

Portable GPS devices were developed to capture a person's activity through out the day, not just while driving. Initial devices were bulky, heavy, had issues with battery life [91], but future iterations lead to smaller, lighter, and battery efficient devices. These devices captured all trips, even non-driving trips, and as a result significant research has been made to develop algorithms that automate the identification of trip details, including trip purpose and travel mode [92, 76, 93, 94, 95, 96] . Studies with wearable devices ranged from small studies of 100-200 people to much larger studies of 2,000 people of varying time periods [91, 97].

Many studies required participants to enter information into PDAs or write down their trips in addition to carrying around GPS devices [98, 99, 100]. Participants experienced fatigue in recording their travel activity, thus some studies were performed with passive devices: users were required to simply carry the device with no data entry needed. The data from the passive device was collected, processed, and only in some studies the user was asked to respond to a prompted recall survey [101, 102, 103, 104, 79, 89] . In these GPS-based prompted recall studies, the data was processed then presented back to the respondents for confirmation and/or correction of derived trip details and the completion of other traditional elements that cannot be derived from GPS data, such as number of travel companions and

destination activities. [105]

4.1.2 Trip Determination with Mobile phones

4.1.3 Prior to smartphones

Before the widespread popularity of smartphones, researchers explored using mobile phones as a data collection tool, using GSM technology (Wermuth et al 2003). An electronic questionnaire on mobile phone was also used to record the trip purpose, travel mode, and start and end time of the trip. Locations, which were derived from phone communication towers rather than GPS, were successfully used for estimating travel patterns in a region [106]. Building on these studies, more technology solutions were explored, include Bluetooth, WiFi, RFID, and smart-cards as a replacement, or enhancement to location gathered by GPS [107].

In Japan, PHS (personal handyphone systems), which work by gathering locations from base stations, similar to how location via WiFi beacons work today, became popular for doing geo-location in cities [108, 109]. More than 20 case studies using PHS technology with participants of up to 100-300 were conducted in Japan since 2003 [110, 111, 112, 113].

One problem with these studies is that there have been varying levels of validation performed to date on GPS processing software, with a general lack of ground truth datasets available to perform this critical task[84]. Although mobile phone technologies seem to be an obvious method for collecting travel data, only a handful of studies have verified the accuracy of automated processing algorithms on GPS data from mobile phones.

4.1.4 The smartphone era

An increasing number of researchers have worked on the problem of inferring modes of transportation using smartphones, both in real-time[114, 115] as well as via post-processing[116, 117]. This research attempts to differentiate a person walking, biking, driving, taking a bus, or some subset of these five modes. Common to all previous work is the extraction of significant attributes from the raw sensor data, using a training set of data, and building a model with a classification algorithm. One of the earliest systems predicted not only mode of transportation, but destination, and route taken in real-time with a hierarchical Bayesian network[115]. Greater classification accuracy was reached when post-processing GPS data was enhanced with mapping data[116]. However, compared with the previous works, the highest levels of accuracy were demonstrated by combining GPS and accelerometer data to determine mode of transportation in real-time[114].

Determining the transportation mode cannot be done with the GPS sensor alone - data about the underlying transportation network is required to differentiate driving with transit trips. Stenneth proposed and validated a method which separated walking, cycling, driving, and public transit (Stenneth 2001). One problem that has plagued both standalone GPS devices and smartphones is problems getting GPS signals in certain regions (near high rises). By augmenting GPS-based positioning with network-based localization to the impact of missing GPS signals is reduced and transportation modes can determined in a more robust manner [118].

However, creating a solution for automated travel diaries using smartphones is not only about determining the mode of transportation a person takes. It is also necessary to determine the time of the trip, route, and trip purpose among other things. A system called CTrack was developed, which recreated trajectories from GSM data only, due to the battery constraints of keeping GPS running constantly - a major issue in the use of smartphones as data collection tools which is discussed in great detail in Section 4.2. [119].

The reported accuracy of these algorithms is very high - over 90% in nearly every case - which shows that data collection for automated travel diaries can be done. However, these algorithms have not been used in long-term studies to analyze two things:

1. Battery consumption of the application on smartphones
2. Accuracy of determining characteristics trips made, which include more than just travel mode: start/end times, start/end points, route, and trip purpose among a large dataset.

4.2 CONCERNS ABOUT BATTERY CONSUMPTION OF SMARTPHONES

When collecting data to generate a person's travel history, location-based services on smartphones are required to be turned on. The use of these location-based services is a major limitation: constant tracking with GPS drains smartphones batteries very rapidly, allowing only for a few hours of usage, and making users very unhappy. The issue of battery life is of major concern: the smartphone is not solely a travel diary tool and for the system to work the travel diary application cannot be a burden to the user. Our philosophy when designing the system was: If one is to develop a system that uses smartphones as a tool for automated travel diaries, the system cannot change the regular habits of the person using his smartphone.

4.2.1 Users care about battery life

Battery consumption on devices used for travel surveys is a major issue. Researchers in the travel behavior community have also been concerned with the battery life of standalone GPS devices. Stopher did a test for average battery life of various GPS devices, finding that the average was 22:46 hours [120]. Battery consumption on smartphones is an even more serious issue. Previous research has shown that existing battery interfaces present limited information, and, as a consequence, users develop inaccurate mental models about how the battery discharges and how the remaining battery percentage shown in the interface correlates to application usage [121].

Rahmati[122]coined the term Human-Battery Interaction (HBI) to describe mobile phone users' interaction with their cell phones to manage the battery available. According to a survey they conducted, 80% of users take measures to increase their battery lifetime, and maximizing battery life will continue to be a key concern for users due to the major usability issues involved in this task. Further studies analyzed user charging characteristics and how users consume battery in their devices [123, 122].

According to the study done by Ferriera, users avoided lower battery levels, with the daily average of the lowest battery percentage values being 30%. In addition, the majority of the charging instances occur for a very small period of time (up to thirty minutes) or between one to two hours, which is the average required time to recharge completely a battery. The charging habits were found to be erratic by time of day, but maintained the average battery percentage greater than 30%.

The result of these studies show that it is important for the travel behavior application on the smartphone to not have a noticeable effect on the battery life of the phone and be able to last through out the day or longer.

4.2.2 Prior research on everyday location monitoring

Outside of the transportation community, many researchers and businesses are also interested in monitoring users' mobility during daily life. Understanding human mobility provides useful information in a variety of fields [117, 124, 125], and as a result there has been significant research in identifying a person's places of interest using smartphones, often called "everyday location monitoring". This research in this field has direct applications to the travel survey community.

For the everyday location monitoring researchers, the major problem that must be solved is minimizing energy consumption of the smartphone while providing accurate location readings. Continuous measurement of a user's position is possible by keeping the GPS on, however, smarter solutions exist to balance the need to obtain GPS data with extending the life of the smartphone battery. First, instead of simply using GPS, Wireless Positioning Systems (WPS) using cell towers and Wi-Fi access points (AP) are technologies used to provide a user's raw coordinates [126, 127]. Also, rather than simply sampling the GPS, or the other location sensors at a constant rate, new algorithms have a variety of strategies to minimize the amount of time the GPS samples location. In nearly all studies, a movement detector is used, either by using the accelerometer[128, 129, 130, 131] or a combination of sensing WiFi and GSM networks[132]. An an adaptive scheduling policy is also used to adjust sampling rates for the various sensors, depending on the objective of the algorithm: to minimize location error[133, 128], minimize energy consumption[134, 130], or maximize place detection. In a few studies, a mobility predictor is also used to estimate times when a person is likely to move, and adapt the sampling rate based on these predictions[133, 134, 132]. These studies have been used to collect significant locations, or points of interest which a person travels to, and similar innovations are used in our design the system to not only detect significant locations, but also the mode of transportation used by individuals. This requires higher sampling rates and levels of location accuracy while a person is in motion.

This prior research shows the problem of tracking and monitoring a person's meaningful places (points of interest) using the regularity of individual mobility pattern has been solved. Thorough evaluations of the energy consumption of mobile phones and the various sensors on the phone have been done, and innovative algorithms have optimized the performance of the tracking users' locations. For this research to be used in an automated travel diary system, this work has to be extended in two ways.

1. The algorithms focus on "where people go". It is need to know "where people go and

how and when people got there". Evaluations on the battery consumption of apps as well as the accuracy of predicted trips is necessary.

2. The travel survey community does regular evaluations with hundreds of people and knows how to correctly sample from household populations. It is not likely that these households will all carry the same types of phones, and use their phones in the same way.

Thus, it is necessary to perform tests on a variety of phones, in a variety of locales, and with various types of users with different phone usage characteristics (turning on and off GPS, WiFi, charging habits, etc.)

4.3 OVERALL ARCHITECTURE

Before detailing the way the trip determination algorithm works, this section describes the big picture: the overall architecture and data flow of the trip diary system. The design of the data collection system is shown in Figure 13. It consists of three components: the tracking application on smartphones, the server architecture to handle incoming location data and handle data requests, and the analytics software to transform the raw data into trips made and meaningful statistics and information about those trips. The applications running on the participants' phones collect raw sensor data which is uploaded to cloud-based server and stored in a database. A periodic job reads the raw data, processes it to infer trip origin and destination location and times and determines route and travel mode(biking, walking, driving or taking transit). This also makes use of information from third party sources such as public transit data which are stored on the server. Trips are further augmented with data such as addresses/neighborhoods of trips made, distance traveled, time spent traveling, CO2 emitted, calories expended, travel costs, and other data that can be layered once trip times and locations are determined.

4.3.1 Mobile application

The mobile application runs in the background on all smartphones running the Android operating system. The goal of the application is to minimize the amount of energy used to collect location and movement data while collecting enough data to allow the algorithms on the server to recreate trips made. In this section, we describe the algorithm implemented on Android phones.

We implemented an algorithm with ideas already implemented by previous groups, as described in Section 4.2.2. The application used similar techniques to reduce battery consumption: a movement detector using WiFi and accelerometer, an adaptive sensor selection during movement adjusting the duty cycles of GPS, WiFi, and accelerometer, and a mobility predictor based on a person's familiar locations at various times of day. We refer the reader to Chon[82] as an example for more detail on the mobility learner, mobility predictor, and adaptive duty cycling strategies used by our application. The only difference between this application and the ones described in Section 4.2.2, is the adjustment of the duty cycling of the GPS sensor while a user is in motion. When a person is detected as moving faster than a

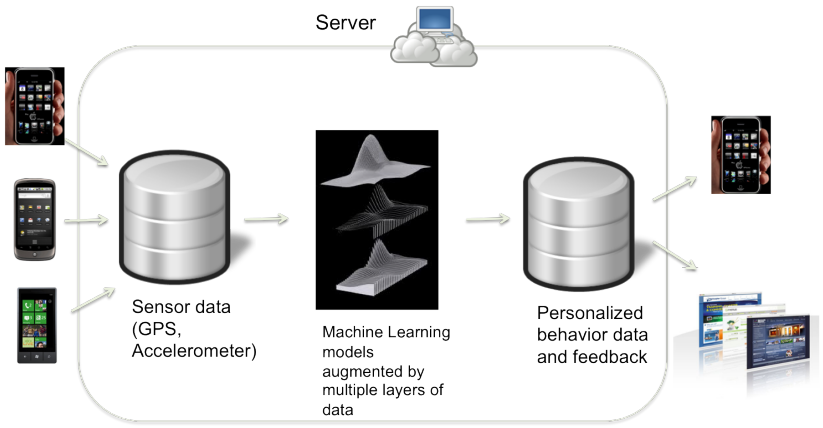

Figure 13: System Architecture Diagram. The components of the system consist of mobile phones, trip determination algorithms running on a server, and web tools to view and correct trips.

walking speed, GPS is triggered more frequently, until the person begins to move at walking speed or slower. Thus, the energy consumption of the application can be broken down into two numbers representing the energy consumption while one is in motion and while one is in stasis. Based on readings from 125 users with 21 different types of phones (described in Section 4.5.1) on average, the amount of power consumed while a person is in motion is 8.7mA while the amount of power consumed while a person is in stasis is 0.7mA. These values are directly queried from the Android SDK. On average, the application accounts for 3% of total battery consumption at any given time. With the average mobile phone battery capacity being 1500mAH, the energy consumption of the application allows for 33 hours of energy consumption, assuming 2 hours of traveling each day and average smartphone usage (internet use, texting, talking on phone, using apps). This is significantly longer than many applications used in practice, simply collecting GPS data every second, or even every 5 seconds, which will drain a user's battery in 5 hours[82].

Due to the requirement to minimize battery consumption, the number of location points generated by the application is very low. The map in Figure 14 shows the number of points received when all sensors are turned on compared to the number of points received using our mobile application. The large rings around the points represent the horizontal accuracy of the location points - red represents GPS points, and green represents network points (retrieved from either WiFi or cell tower positioning). In addition, accelerometer points are only requested once per minute while the user is in motion, and mac addresses of nearby WiFi access points are recorded only when a person transitions from stasis to motion, as detected by the accelerometer. These pieces of data from each sensor is sent to the server with a timestamp, which does the work of sorting through these independent readings and generating trips.

Upon launch of the application, the user is asked two survey questions: the number of days per month they bike, and take transit. The results of the survey are used in the mode determination portion of determining trips made, described in Section 4.4.6. Afterwards,

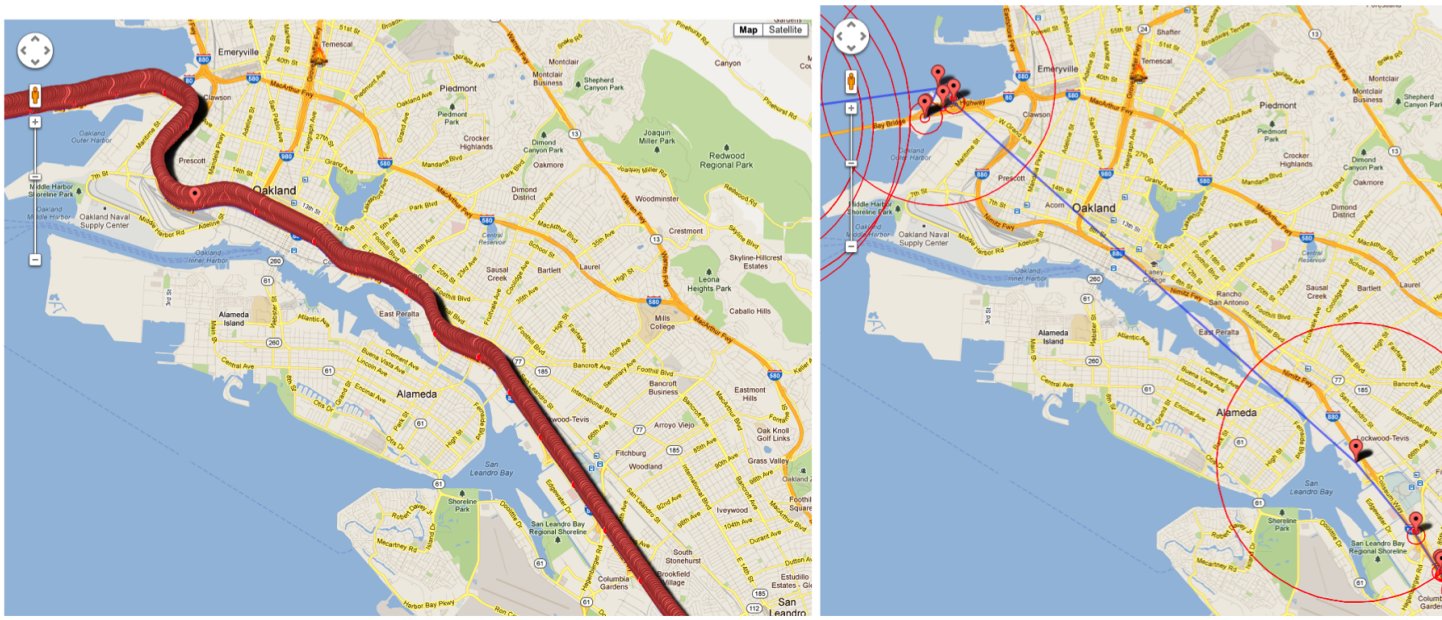

Figure 14: 1Hz data vs. Sparse data. The figure on the left shows the amount of location data from the phone with the GPS sensor gathering data at 1 Hz. This exemplifies the type of data a standalone GPS logger would receive. The figure on the right shows the amount of data generated by our mobile application. The red circles represent the horizontal accuracy values, which captures the uncertainty of the location.

the application runs in the background. If the application is killed or the phone restarts, the application is brought back up after a short delay. This allows for the phone to record data at all times to determine trips.

4.3.2 Server

The server is the hub of data where all trip processing occurs. Sensor data from the phone is uploaded once every three hours and split into various databases depending on the type of data: accelerometer readings, wifi access points, battery levels, locations (latitude and longitude) based on cell tower / wifi positioning, locations based on GPS, whether or not various location providers are on, and other data points. All the pieces of data play a different role in re-creating trips made by a user: the algorithm is described in greater detail in Section 4.4. The server contains databases of third party data used to process trips (i.e. transit, weather data). The server is run on a medium sized instance on Amazon Elastic Compute Cloud (EC2), with API access from the phone handled by NGINX storing data in a Postgres and Mongo database.

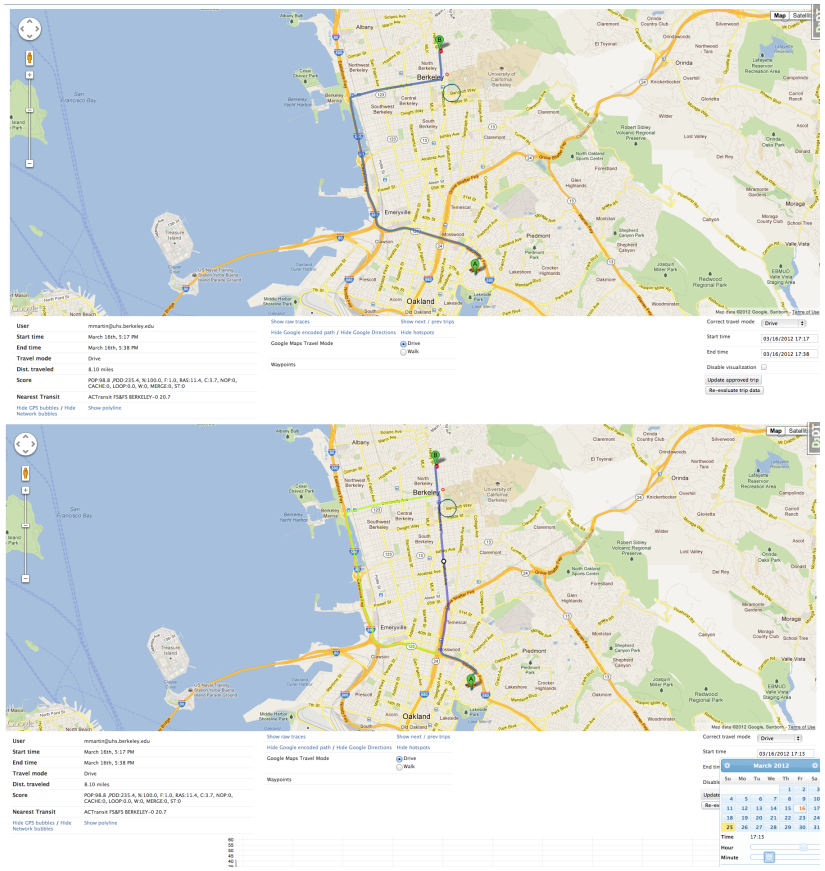

Figure 15: An example of the tool used to correct a trip's route and start time. The figure on top shows the trip data that is generated by the trip determination algorithm. The figure on the bottom shows the trip after the route and start time have been corrected.

4.3.3 Website - Online Mapping Tools

A web interface was written to provide a suite of tools to handle research subjects' data. The interface allows a researcher to process trips with one-click, and also allows the researcher to visually inspect the set of trips generated for any user with mapping tools. In addition to simply inspecting the trips for correctness, the researcher can correct the trip, by selecting different start/end points, changing the start/end time, adjusting the route taken, or transportation mode.

These tools are used in the evaluation of the algorithm, described in Section 4.5. Using the tools, testers validated the accuracy of their trips by viewing their data and having the researcher correct the computer generated trips; the users' inputs generated the ground truth which was used to evaluate the algorithms, as shown in Figure 15.

4.4 AN ALGORITHM FOR TRIP DETERMINATION WITH SPARSE DATA

The goal of the trip determination algorithm is to take the sparse data from the phone's sensors and recreate trips made. For the purpose of this chapter, the definition of a trip is a period of motion surrounded by two periods of stasis greater than 3 minutes. During this period of motion, a person must have traveled a minimum of 250m from an origin to destination able to be mapped to points along the street network. As described in Section 4.3.1, and shown in Figure 14 data from the location sensors is not frequently generated, however, the much bigger issue is that there exists a wide variety of phones that have a variety of behaviors which can only be learned through real world testing. A list of some of the problems encountered during testing of the application are described in Section 4.5.1. These discoveries from field testing the application have contributed to the various processes which the data flows through. A flowchart of the trip determination algorithm is shown in Figure 16, and each block in the flowchart is described in the following sections.

4.4.1 Filtering noisy data

All location points generated by the phone are returned with a latitude, longitude, horizontal accuracy (the error range of the latitude longitude pair), velocity, altitude, and the source of the location point (from GPS or Wifi/Cell towers).

In theory, the true location of the phone is contained within the centroid around the latitude, longitude geopoint with value of the horizontal accuracy as the radius of the circle. However, this is not true; once in a while completely inaccurate GPS points are created, and inaccurate network points occur extremely frequently. In a test of 777,670 data points gathered from the data set of 125 users (described in Section 4.5.1), under 1% of GPS points lied outside the circle defined by the latitude, longitude and horizontal accuracy, while 51% of Network points, with the location sourced from either cell towers, or WiFi beacons lied outside the circle.

Despite the fact that network points are often inaccurate, they are used as part of the mobility detection algorithm, to ensure that the GPS sensor is not used very frequently. Thus, faulty points are simply filtered out before attempting to build trips. Although much more rare, inaccurate GPS points are generated from time to time. Like the inaccuracies with the network points, the reasons for this are unknown, but the points are filtered out as well.

Location points are filtered under the following conditions:

1) If two points are generated with the same timestamp, the more accurate point is taken.

2) If the speed required to travel the distance between the edge of the location centroids is greater than 105 miles an hour between consecutive points, but the first and last point is less than 105 miles per hour, the in-between points are removed. The speed of 105 mph was chosen from empirical evidence of the same observations used to calculate the location accuracy values descried in this section.

3) It has been observed that once in a while the GPS sensor can enter a "disrupted mode" in which it can temporarily return faulty points outside the horizontal accuracy centroid for

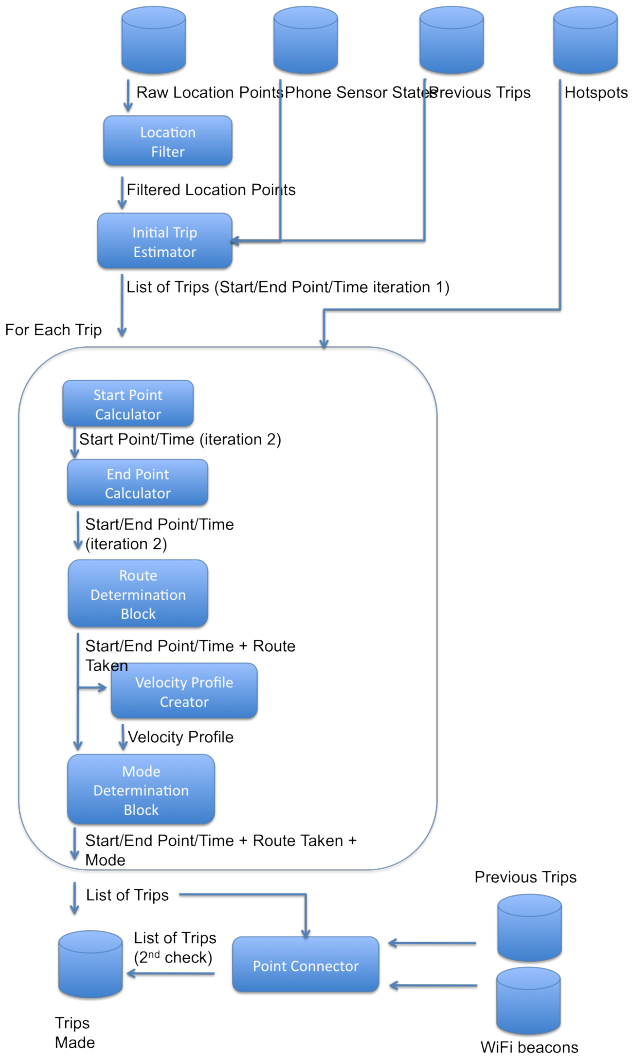

Figure 16: Overall algorithm flowchart. Each block and data source is described in Sections 4.4.1 - 4.4.9

up to 2 minutes at a time (based on empirical evidence). Any points created during the "disrupted mode" are removed.

4) Faulty networks points often are returned repeatedly from the network listener - if a WiFi/Cell based location point is tagged as faulty even once, all future instances of the exact same location point are removed.

The filtering algorithm with the preceding rules is run forwards and backwards in time.

4.4.2 Hotspots

Before any trips are calculated, the raw location points are used to identify locations where a person spends a large amount of time (i.e. home, work, gym) which we refer to as "hotspots". In Section 4.3.1, we referred to a number of papers which minimized battery life to identify

points of interest. We follow the same logic here, and further describe the methods for identifying trips made in the following sections - the contribution of this chapter on top of the research from Section 4.2.2.

Figure 17 shows the algorithm flowchart for hotspot generation. The first step, the location filter, is used such that all non-GPS points and GPS points with accuracy greater than 66 meters are not used to determine the hotspot centers. In the process of marking these points, each GPS point with a horizontal accuracy of under 66 meters is assigned a duration for which it can be proven that the person was at the latitude, longitude specified by that GPS point.

The duration of each point is set as follows. For each GPS point, each subsequent location points are analyzed to see if it can be proven that the person has moved. First, if the areas covered by the centroids (specified by latitude, longitude, and horizontal accuracy) of the location points do not intersect, then the duration of the GPS point is set as the difference in time between the two points. In addition, WiFi beacon scans can prove a person has moved. If the intersection of access points seen by two subsequent WiFi beacon scans is null, then it the duration is set as the difference in time from the second WiFi beacon scan and the time the GPS point was captured. Figure 18 illustrates a situation in which location points are used to set the duration which the hotspot is valid. In the figure, the duration of the points in the hotspot cluster is calculated from the time of the earliest location point seen in the cluster until the time of the earliest location points seen in the other cluster. The points in the other cluster may be as a result of movement, or faulty readings from the phone, but because neither of those can be proven from the raw data itself, it is assumed that the person has moved.

The output of the location filter is a set of latitude-longitude-duration tuples, which is run through a weighted clustering algorithm[135]. Any hotspots within 250m of each other are merged, with the center of the hotspot set as the hotspot with points holding the longest duration values. Any hotspots with a total duration value of less than 2 hours is eliminated - this is an arbitrary value used to capture a person's total points of interest. For example, if a person spends two hours at a library, or goes to a fast food restaurant for 20 minutes at least 6 times, both will be considered hotspots. However, a location where a person repeatedly drops off or picks up another passenger probably would not be captured by this algorithm.

Finally, each hotspot is annotated with the set of all WiFi beacon seen within a 250m radius of the point.

4.4.3 Determining Start/End Location of trips

Determining trips requires input from a variety of databases, raw locations, phone sensor states, prior trips made, and hotspots. The data from the various databases are fused, as the algorithm makes a series of guesses to determine the features of each trip made.

The first guess for the start and end location of trips initiated by the state of all the sensors on the phone: this is defined as the sampling rate of the accelerometer, WiFi beacon sensor, GPS, and Network Listeners. When the phone detects a person is traveling at a speed faster than a walking speed, the duty cycle of GPS is reduced - the time first instance

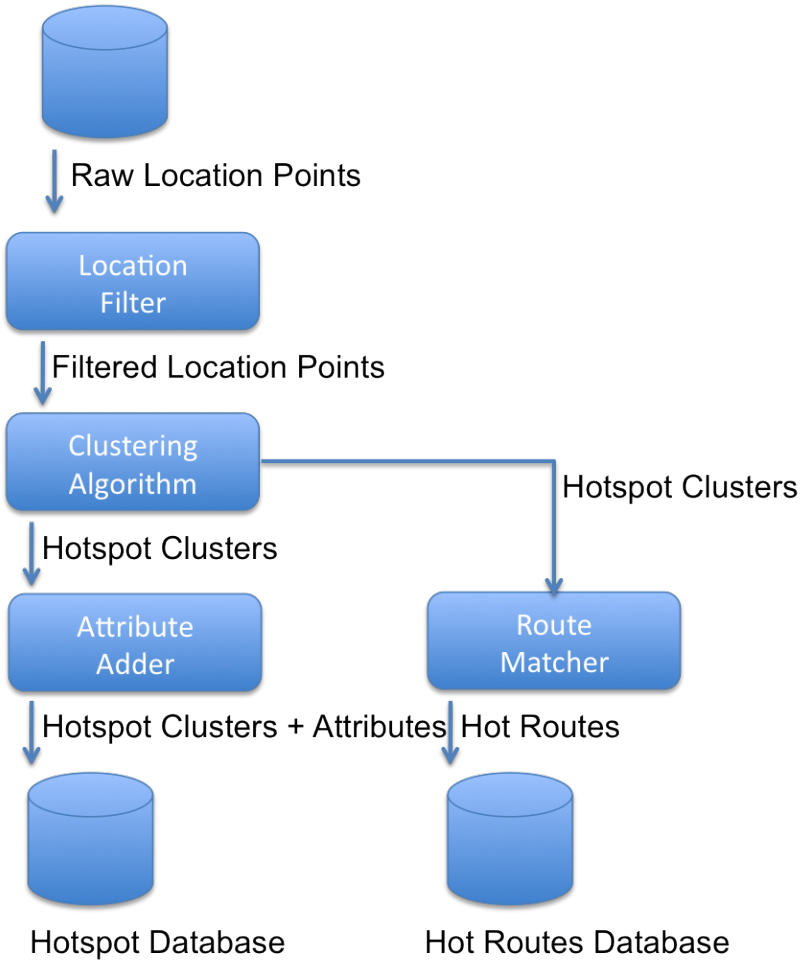

Figure 17: Hotspot algorithm flowchart

of the change in duty cycle is the initial guess of the trip start. The nearest location point to that timestamp is defined as the first guess of the trip start location. Likewise, the time in which the duty cycle of the GPS sensor increases back to the stasis mode is the initial guess of the trip end. The nearest location point to that timestamp is defined as the first guess of the trip end location. Based solely on the raw location points and the state of the phone sensors, a set of initial trip guesses are made.

For each initial trip guess, the trips are refined to a second estimate of the actual trip start/end time and location. The second guess of the start location is determined by iterating through the filtered location points backwards in time - after it can be proven that the smartphone has not been moving, as described by Section 4.4.2and Figure 18, the set of location points during the timeframe in which the phone is in stasis is returned. Each point in this set of points has a corresponding horizontal accuracy; these points are sorted by horizontal accuracy. If the most accurate point in the set has a horizontal accuracy below 66m and is generated by the GPS sensor, it is assigned as the starting point of the trip. Otherwise, if there exists hotspots that are contained within the shape created by the set of

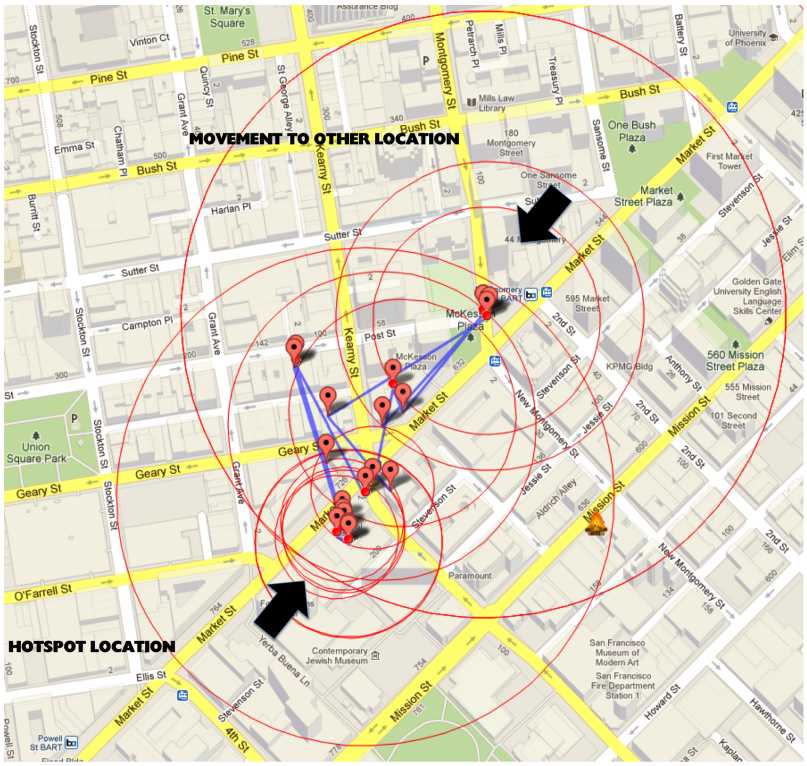

Figure 18: Example of location points at a hotspot center and at another location which signifies movement away from the hotspot cluster.

potential starting points, those hotspots are set as the potential starting points of the trip. The hotspot which most closely matches the characteristics of the start time of the trip is assigned as the starting point of the trip - for example, if a trip starts at 7:00AM, the hotspot which corresponds with the location where the person usually spends the hours of 6PM - 7AM daily (most likely one's home) is favored over the location where the person spends the hours 3PM - 4PM on Saturdays. If no hotspots are contained within the shape created by the set of potential starting points, the raw location point with the smallest horizontal accuracy is used - trips which fall into this scenario are rare, and are described in greater detail in the evaluation of the accuracy of hotspots in Section 4.5.5.

The initial guess for the start time of the trip is determined a combination of the accelerometer, Wifi, and location sensors. The accelerometer runs on a duty cycle of approximately 1 minute; when the accelerometer detects movement, the phone scans for WiFi beacons. If the intersection of the set of beacons seen at subsequent WiFi beacon scans is null, then it is assumed that movement occurred during the time between the two scans.

Determining the end location and time of scans is easier than determining the start location of trips. To determine the start time, many of the sensors have long duty cycles in a "low-power" mode. On the other hand, while the person is in motion and arrives at their destination, the sensors have shorter duty cycles. The end location of a trip is determined by iterating through the filtered locations forwards in time from a location prior to the initial guess of the end location. After it can be proven that the smartphone has not been moving,

same steps are taken as determining the start point of trips - sorting the potential endpoints and using the hotspots database when necessary.

Determining the end time of a trip is based on a combination of the location sensor and accelerometer. If the accelerometer detects a prolonged period of no motion, the trip end time is defined as the time of the first reading of the no motion period. However, while the accelerometer can provide a clear signal when the phone is still, not all trips end with the phone being still - for example, after one has arrived at an office building, the person must walk up the stairs to his final destination. If a prolonged period of stasis is not detected, the location sensor is used as the method of determining the trip end. The same set of location points used to determine the end location - proving that a person has stopped moving - is used by using the time of the first point in the set of points as the end time. In the event that accurate location points are not returned from the phone, WiFi scans are used, If a single common access point is seen between two consecutive scans, the end time of the trip is set as the time of the first of the two WiFi scans.

4.4.4 Determining Route taken

Once the trip's start and end locations are determined, the route taken between the points is generated. The route determination algorithm uses these start points, the raw location points, and the Google Directions API[136]. The flowchart of the Route Determination block is shown in Figure 19.

While it is possible to have built our own routing engine and mapping database, the routing algorithms were outsourced to the Google Directions to save time and take advantage of the expertise of the Google team. The input to the web request are, the starting and ending points, a set of waypoints, a choice of directions by mode (driving, bicycling, walking), and whether or not alternate routes should be searched. The output of the API is a set of shortest paths (by time) between the origin, destination, and that run through the waypoints. It should be noted that there are a couple of minor disadvantages of using Google Directions. First, the number of calls to the API is limited to 2,500 requests per day, with a limit of 100,000 requests for paying customers. Second, the number of waypoints that can be sent to the API is limited to 8, and for paying customers the limit is 23. This means that any trip which uses extremely suboptimal routes(by travel time) such as looping around blocks or taking a long series of residential streets instead of the highway, may not have its routes calculated accurately.

Before sending a request to the Google Directions API, a set of possible waypoints is generated. The purpose of generating a set of waypoints that does not include a raw points timestampped between the start time and end time of the trip is to minimize the chances of generating an incorrect route. The Google Directions API assumes the origin, destination points and waypoints are exact, while the phone returns points with horizontal accuracy readings that are not centered at the exact latitude longitude point. Location points greater than 100m accuracy potentially encapsulate multiple streets, and even GPS points with a 5m horizontal accuracy often are centered on the wrong side of the street, which affects the route returned from Google. Getting the route correct is important not just for the sake of being correct, but the route is also necessary for the map matching of transit routes when determining travel mode, as described in Section 4.4.6.

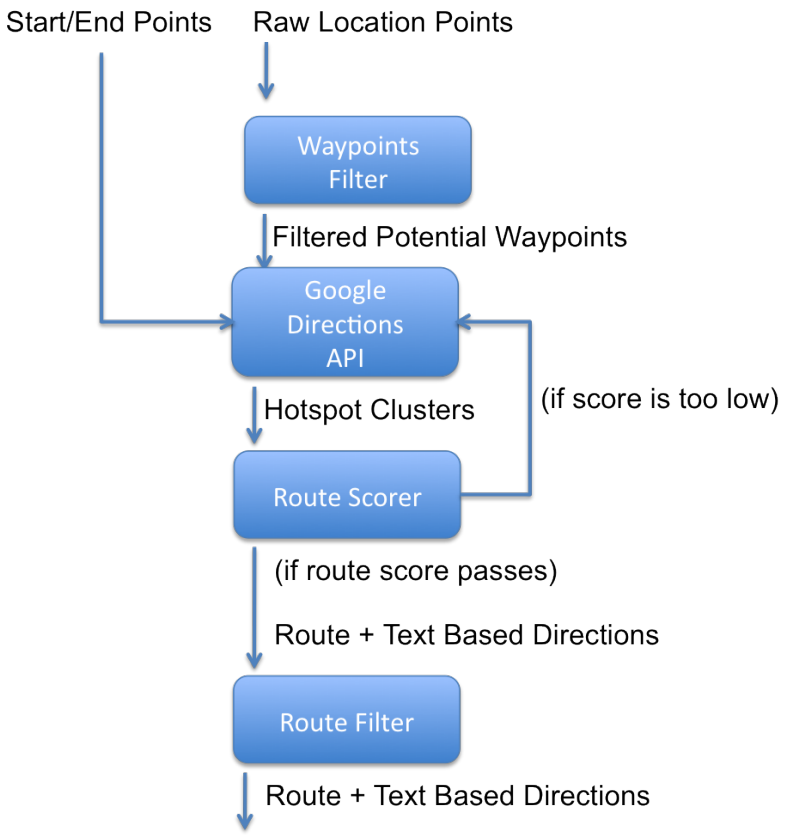

Figure 19: Routing Algorithm flowchart

To minimize these routing errors, the most accurate GPS point, measured by smallest horizontal accuracy radius, for each minute is kept in the set of waypoints, all other points are filtered out. Before waypoints are considered, only the origin and destination points are sent to Google. One to three routes are returned, and if none of the routes pass through all of the raw points, the path is sent to google again with a waypoint at the most accurate point along the section of the route missed by the returned routes from Google. This process is repeated until the route matches or the waypoint limit is exceeded.

4.4.5 Loop problems

Even with the measures taken to minimize routing errors, there exists cases in which using any combination raw location points as waypoints does not allow for the correct path to be returned from the Google Directions API. In these situations, the incorrect sections of the route have to be removed.

The first step involves detecting all loops in the route - this includes loops that are actually valid sections in-between loops which should be eliminated.

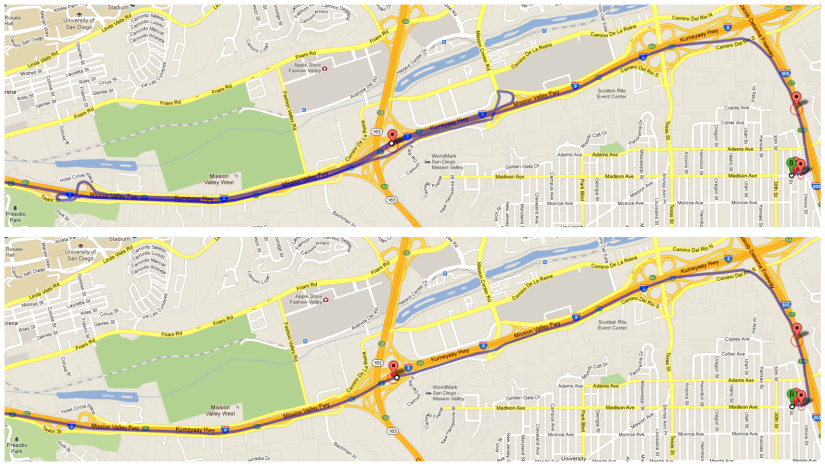

Figure 20: Top: Example of an inaccurate location point causing a loop in the route Bottom: Corrected route with loop removed

Loops are then eliminated if the speed required to traverse the loop(based on the raw location points) exceeds the speed limit by 1.5x.

Figure 20 shows an example of a loop removed when meeting the preceding conditions, and a loop that is not removed. Because the number of GPS points are sparse, the velocities at any given time along the route are generated by a velocity profile generator, which snaps raw locations to the route returned from Google and generates a range of potential velocities for the duration of the trip.

4.4.6 Determining Mode taken

The last piece of data needed to generate a trip is annotating the mode of transportation to the start/end time and location and route taken. Determining travel mode is done in two steps: a classifier to differentiate between motorized modes, bicycling, and walking, and a map matching algorithm to differentiate between public transit and driving. The travel mode is defined as the main mode a person during a trip (defined in Section 4.4). While a trip may consist of many segments, such as walking to a car in a parking lot, then driving home, only the main mode, here, driving, is classified.

4.4.7 Classifier

The classifier is built with the same techniques used in prior research to determine mode of transportation as described in Section 4.1.4. In that research, techniques were developed to perform real-time mode determination, which identifies the transportation mode being used at each unit time interval (e.g. every second). The purpose of this classifier is to identify the main mode used during a trip, but the same steps are followed: collecting a set of training data, identifying a large set of features, performing feature selection, and picking a classification algorithm. The only difference this research and prior work is the set of

features selected; instead of using features which can only be observed in real-time, features that are defined over the course of a trip are used, such as the median speed between the trip start and end times.

A set of training data was collected daily across various modes of transportation over the course of 1 month for 6 individuals. A set of features was identified, which included:

1. Speed
2. Trip duration
3. Trip length (distance covered)
4. Expected Driving duration (obtained via Google Directions API)
5. Expected Biking duration (obtained via Google Directions API)
6. Expected Walking duration (obtained via Google Directions API)

Although the accelerometer is a useful sensor to determine transportation mode, features from the accelerometer are not used for the following reasons. Velocity alone can differentiate walking from biking/driving modes. Biking and driving cannot be successfully distinguished because similar accelerometer signatures when the phone is placed in a backpack, or a location that does not allow the phone to capture the biking motion. Previous work has shown that Decision Tree-based classification is simple and effective for classifying the sensory features into transportation modes [114, 116]. In this work we used the Random Forest classifier, which is also tree-based, based on our preliminary experiments and comparisons with Decision Trees. A Random Forest classifier consists of a number of decision trees, each performing classification and voting for the predicted class. Overall output is the class which has the highest number of votes. Each decision tree is built using samples randomly selected from the training set. While growing the tree, a fixed-size subset of features is randomly selected at each node. Random Forests are known to be efficient on large datasets, estimating missing data and handling unbalanced datasets. In addition, they evaluate the importance of variables for classification[137]. In this work, we built a Random Forest with 10 decision trees and 6 attributes.

The classifier predicts between walking, motorized, or biking. An additional classifier was built to predict only between walking and motorized. Based on the results of the survey questions asked in the mobile application about the frequency of biking (Section 4.3.1), the appropriate classifier is used. For example, a user who responds that he never bikes will have his trips processed through the classifier that distinguishes between walking and motorized only.

4.4.8 Map Matcher

Before the entire trip generation algorithm is run, a database of transit route configurations is built using open transportation data. In 2006, Google established a standard for encoding transit data called GTFS, which contains data that can construct route configurations. Files that store the following relevant information in the structure specified in

Figure 3, the route name, direction name, stop / intersection name, latitude, longitude, stop sequence, and agency name. A route configuration is categorized by a route, direction, and agency name. Each configuration specifies a set of stop names with corresponding latitude and longitude points in the sequence which the bus or train traverses the route. Configurations with the same route, direction, and agency name are grouped together. The data also specifies whether or not the path taken by the route occurs on the road network or on dedicated lanes, Currently, GTFS-Data-Exchange[138] is the premier repository of all open transit data stored in the Google Transit Feed Specification format. GTFS files are pulled from GTFS-Data-Exchange and converted into route configurations which are stored in a database. At the time of writing, the database stores data from 339 different transit agencies.

At the time of processing, trips with a route are sent to the Map Matching algorithm to determine the percentage of the route that matches with any bus or train lines which crosses the route taken. The algorithm for performing the map matching is a variation of the Bentley-Ottmann algorithm[139], which finds intersections in set of straight lines. The bus route, and the actual route taken are defined as a series of latitude and longitude points, which when connected form the set of straight lines which the algorithm iterates through. The bus or train line with the highest percentage is returned, and any trip matching greater than 90% is defined as a trip taken as a transit trip.

4.4.9 Catching all trips made

The preceding Sections 4.4.2 - 4.4.6 describe the flow of data through the algorithm when the phone properly adjusts its' duty cycles to capture location data while a person is in motion. However, many trips are made while the phone does not function properly. There are many reasons for this; as discovered in the evaluation in Section 4.5, users often turn off their phones, turn off GPS or other location services, and most commonly, the phone does not respond properly to requests to return readings from the accelerometer, GPS, or Network listener. To account for these situations when trips are made, but zero, or very few sensor readings are received, data flows through a separate module.

This module, called the Point Connector in Figure 16, evaluates all sensor data during the "non-trip" times when the person is expected to be in stasis. The same clustering algorithm as the one described in Section 4.4.2 is used to identify locations where a person potentially could have traveled to. Like the hotspot algorithm, the locations must have a horizontal accuracy reading of 100m or less, due to the number of inaccurate network point readings returned by the phone. In the situation that no point in a cluster has an accuracy of under 100m, sensor readings from WiFi access points are queried against the master WiFi access point database to provide a more accurate position.

The result of the clustering algorithm is a set of centroids defined by latitude, longitude, and horizontal accuracy, and the corresponding time a person was assumed to be at that centroid. The centroids are sorted by time and routes are generated by sending the origin and destination with no waypoints to the Google Directions API. The end time of the trip is set as the timestamp of the earliest reading at the centroid, and the start time of the trip is set as the same timestamp minus the duration of the trip as returned by the Google Directions API. The mode of transportation is determined in two steps. If the same trip has been made in the past, the transportation mode most commonly used for that trip is set as

the transportation mode. If not, the transportation mode most commonly used for a trip of that distance is set as the transportation mode. Trips which are generated by this method mainly occur during short trips under 2km and are less accurate in terms of start/end time, route taken, and transportation mode.

4.5 EVALUATION

Evaluations were performed on the system to accomplish two goals. The first goal was to get the application running on as many different types of phones, living in different locations, with different user behaviors as possible. The second goal of the evaluation is define a set of metrics and to test accuracy of trip determination system with these set of metrics. In this evaluation, separate from the first evaluation, a set of users manually checked the accuracy of all trips generated by the system and made corrections to set the ground truth for trips made. This evaluation consisted of a three-month process in which six users discussed their trips daily with the researcher and used web-based tools to correct their logged trips. This allowed us to evaluate the errors in the start/end time, start/end location, route taken, and travel mode taken for these trips made and determine error statistics, instances in which the system missed trips, or incorrectly created trips when no trip was actually made. These six error metrics are described in the following sections and are proposed as the measures for which all automated travel diary systems should be evaluated upon.

4.5.1 Phone Tests

This system is meant to be used by researchers to gather data from the field. In the field, there exists a large number of uncontrollable variables: smartphones are made differently, by different manufacturers, users live in different locations which don't produce data in a uniform way across all topographies and locations, and user's phone behaviors are very different. To run a large scale trip diary study, the system must be robust enough to handle all different types of scenarios and corner cases which do not show up in a limited study.

The application was tested with 125 different users on a number of phones in a variety of states and cities with different topologies and population densities. The phones tested include: HTC Desire, HTC Hero, HTC Droid Incredible, HTC Incredible S, HTC Evo 4G, HTC Inspire 4G, HTC myTouch, HTC Wildfire, HTC Thunderbolt, LG LS670 Optimus S, LG P970 Optimus, Google Nexus One, Motorola Atrix 2, Motorola Droid, Motorola Droid 2, Motorola Droid X, Motorola Droid X2, Motorola Droid Pro, Samsung Galaxy S II, Samsung Galaxy Nexus, and Google Nexus S. The data was collected in a variety of cities with different topologies and population densities, in the states of Washington, Oregon, California, Texas, Utah, Illinois, Missouri, Alabama, Georgia, Florida, South Carolina, North Carolina, Virginia, Ohio, Maryland, New Jersey, Massachusetts, New York, New Hampshire, and Ontario, Canada. This data was collected over the course of 9 months for a total of 6,440 days of logged data across all testers. The 125 testers used the application for different periods of time, from as short as 3 days to as long as 9 months, starting in July 2011. 30 of the 125 testers ran the application for more than 3 months.

This evaluation allowed allowed us to learn about the problems encountered from the field, which is divided into phone problems, and user issues. Because of these problems, the

phone does not always detect when a person is in motion, and trips are not always perfectly detected.

4.5.2 Phone issues

Across all phones, it was very quickly learned that location readings from non-GPS sources return inaccurate readings very frequently: this means that the true position of the person lies outside the circle defined by the latitude, longitude, and radius being the horizontal accuracy returned from the phone's location readings. The number of inaccurate location readings was as high as 80% for some users while being as low as 5% for other users. The reason for the inaccuracies was not investigated, but the filtering algorithms described in Section 4.4.1 were adapted as a result of the large number of inaccurate network readings.

While most phones were found to be able to access the accelerometer in the background, several models of phones are unable to do so, mostly from the Motorola line of phones (at the time of testing running Android 2.3 or lower). As a result, WiFi beacons were necessary to act solely as the motion detector instead of both the accelerometer and WiFi beacon sensor.

The accuracy of location readings were also highly affected by geographic topology. It has been widely reported that GPS does not work well in areas with large skyscrapers, however this study showed location readings in areas with skyscrapers were very accurate, possibly due to both the density of WiFi access points used in positioning. The areas which provided the most inaccurate readings were suburban and rural areas when only location was attempted to be gathered from non-GPS sources. In these scenarios, network points were inaccurate up to 10 miles for one particular user, and were about 1 mile off the actual location in general. Most troubling is that multiple inaccurate points were often returned in sequence, making it appear as if the phone could be moving. In these instances, the accelerometer, and WiFi access point scans allowed for the algorithms to ignore the result of the inaccurate network points.

While inaccurate network points are common, inaccurate GPS readings are very uncommon. However, errors in GPS readings may lead to faulty generation of trips made. It has been observed that GPS readings of high accuracy often drifts anywhere from 20m to 50m while the phone is in stasis. In addition, readings have been observed to be as far as 1500km away from the true location of the phone. While faulty trips are generated, many of these trips do not follow the road network - leading the system to flag the trip as a potentially inaccurate trip.

Finally, it is expected that when the app makes a request for a reading from the GPS, accelerometer or any sensor, the appropriate process will execute to provide the data to the app. However, this is not always the case: 82% of the time when a location point is requested, a point is actually returned from the operating system to the app. The app tries again if a location is not obtained, however, there have been instances when a person has traveled for up to 32 kilometers with the app requesting GPS the entire time but with no location points generated. For the accelerometer: this problem occurs on average 95% of the time. This behavior is often hard to predict and reduces the accuracy of the route taken on a trip.

4.5.3 User behavior issues

In the Android operating system, a user can choose whether or not to turn on GPS sensors, Network location sensors, and data connection. To perform optimally, the application requires that these three settings are turned on - however during the tests, we did not force the user to do so. A small notice was shown to users that the settings should be, but did not require users to follow the given instructions. This allowed us to observe users' phone behaviors and their affect on the trip diary application running properly.

Not all users turned on GPS services on their phone. It is widely known that turning off GPS increases the battery life of phones, and 4% of the testers followed this advice $> 99\%$ of the time. An additional 2% of users frequently switched GPS settings on and off, while the rest constantly kept GPS running. The system still works when GPS services are turned off, however, the start/end locations trips can only be determined to the nearest kilometer for most trips.

Another problem encountered is users turn off their phones, allow the phone to run out of battery, or the phone reboots (for undetermined reasons), although these cases happen infrequently. In these situations, which occurred .11% of the total recorded time, no data could be collected. In cases when this occurred between trips, the algorithm in Section 4.4.9could be implemented, but in cases when round-trips were made, the phone is unable to detect trips.

4.5.4 Trip evaluation

After the 125 person test was conducted, a second evaluation was run to evaluate the accuracy of the algorithm for detecting trips made. This evaluation consisted of 6 people over 3 months, which generated a total of 1850 trips. Due to the extensive work required to have people manually verify their trips made, a decision was made to run a long test with few people, rather than a short test with a large amount of people. The reason for the smaller number of people in this test compared to the prior test is the ability to gather a dataset with large longitudinal depth to capture people's lifestyles, hotspots, and travel behaviors. However, the 3 month period is the longest study of its kind for an evaluation of a travel diary system using smartphones. To collect the ground truth for trips made, each of these trips were discussed verbally with the researcher to gather the most accurate readings and manually verified using the tools described in Section 4.3.3 . This one-on-one discussion made possible the detailed metrics by pinpointing locations of trip start/end locations, hotspots, and routes taken on a map to gather error rates measured to the meter accuracy.

4.5.5 Hotspot evaluation

The first evaluation of the system is the evaluation of the accuracy of the hotspots identified by the system. Validating the accuracy of determining hotspots is important because most trips made start and/or end at a hotspot. Across all users, of the trips made, 48% of trips occurred between two hotspots, 32% of trips occurred starting or ending with a hotspot, and 20% of trips did not include a hotspot at all. The use of hotspots is important because accurate location data at the start/end point is not always present for all trips.

The most common hotspots, such as one's home and work can be identified in the first

Mode	Trips Logged and Verified
Biking	154
Driving	998
Transit	349
Walking	349

Table 2: Mode Split for trip determination accuracy evaluation.

day of using the system while other hotspots such as a friend's house or the supermarket slowly enter the system as they are visited.

4.5.6 Method for verifying accuracy of the data

Here, we propose a definition of a trip, and the 6 metrics used to verify the accuracy of the trip determination algorithms and overall system:

1. Start Time
2. End Time
3. Start Location
4. End Location
5. Path Match Accuracy
6. Transportation Mode

Of the 1850 trips recorded, the start/end location, was verified for all trips, while the mode and route taken was verified for only 609 trips, which occurred during the last month of the evaluation. The mode split of the trips made is described in Table 2. The start/end time of trips were verified in a different manner, described in Section 4.5.10. The reason why the start/end time, path, and transportation mode were not validated for all trips was because of the fatigue of manually correcting trips daily by the testers.

4.5.7 Errors in Origin and Destination location

Errors in location are measured by the true start/end location of the trip as validated by the person making the trip compared to the start/end location automatically generated by the system. The true start/end location was recorded using the online web tools described in Section 4.3.3 which used a draggable google maps marker to set the location. The average error for a hotspot location was 25m with a standard deviation of 101m. Across all users, a total of 182 hotspots were detected. This error understandably increases to 197m with a standard deviation of 599m for origin/destination locations which were not hotspots.

There results show that errors in hotspot trips is significantly lower than trips that do not contain any hotspots. Across the 3 month evaluation, 90% of hotspots were discovered

by the 3rd week for 4 out of 6 of the users. Two users moved their residences during this time, thus their hotspots changed significantly before and after the move. These results are promising for long-term studies, but a wider evaluation with more users is required to validate this.

4.5.8 Routing Errors

To measure errors in the route generated by the system, the system was evaluated with three measures: meters traveled in which predicted route matches true route, meters traveled in which predicted route does not match any part of the true route, meters of the true route which is not covered by the predicted route. These three measures were translated into metrics called :

1. Route Match : meters traveled in which predicted route matches true route / true route meters traveled, expressed as a percentage

2. Route Error 1: meters traveled in which predicted route does not match any part of the true route/ true route meters traveled, expressed as a percentage

3. Route Error 2: meters of the true route which is not covered by the predicted route / true route meters traveled, expressed as a percentage

Figure 21 shows an example of the three different types of errors. In the evaluation, only driving, transit and biking trips were evaluated. Many walks occur in parks, on campus, or areas where there is no mapping data for all paths.

Table 3 shows the breakdown of errors across these three metrics for the six individual users. The errors are separated by user to show the larger errors in the system for users who use underground transit systems. Users 1 and 2 primarily travel by underground subway, which does not run on a path able to be mapped by the Google Directions API, which only returns results for driving, bicycling, and walking directions. On the other hand, users 3 and 4 use the underground subway once in a while, and users 5 and 6 are primarily auto users.

The second half of Table 3 shows the routing errors after user input. Each day, after the trips were generated, users corrected their trip details: start location and end location, and for one month users corrected their start location, end location, route taken, and transportation mode taken. To generate data for the second table, the trip determination algorithm was re-run daily, taking these corrected routes, called "hot routes", as generated by the hotspot algorithm described in Section 4.4.2. These "hot routes" were used only for trips between two hotspots, and significantly increased the route match percentage. The evaluation shows that with a set of user input, there is potential for designing a feedback system to greatly increase the accuracy of the overall trip determination system.

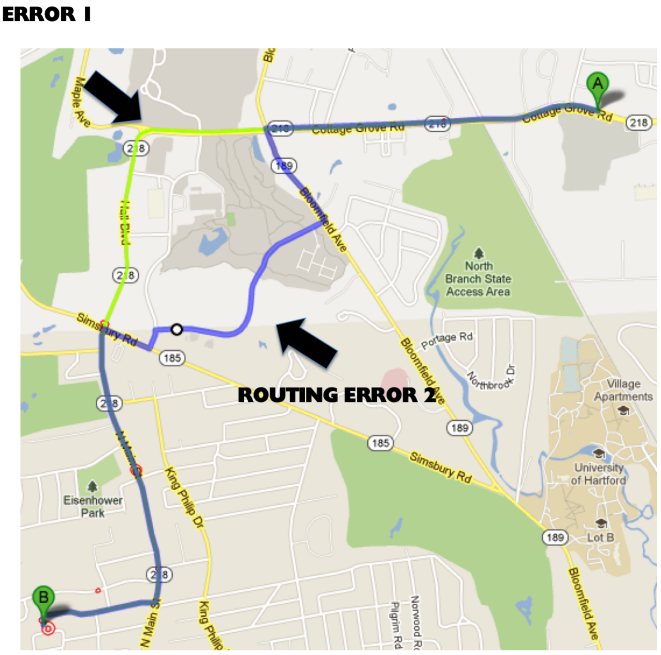

Figure 21: The two types of routing errors are shown on a map. The green line represents the ground truth, and the purple line represents the predicted route

4.5.9 Travel Mode Errors

The errors in the travel mode determined by the system are described by the confusion matrix in Table 4.

The errors show that much work is needed to improve the system. Reasons for the errors include the following:

1. One of the users is a very fast biker in the city, and the set of features used in the classifier did not distinguish between motorized and biking very well.

2. Because some transit trips are made underground, location points had very large horizontal accuracy values, causing the route to be calculated incorrectly

3. Errors in the route predicted by the system propagated to the travel mode determination due to the map matching of incorrect routes.

These errors show that the initial predictions for trips made needs room for improvement. However, immediate and significant improvement can be made with some user input. As described in 4.5.8, users checked and corrected their trips made, which were cached as "hotroutes". Any future trip made between the same origin and destination used the cached hotroute and corresponding transportation mode if the raw location points matched up with the stored route. On average, a total of 49 hotroutes per user were stored, with an average of 1.56 hotroutes per origin-destination pair. Considering these values, it is plausible to develop interfaces to prompt users of an automated travel diary system to check their trips made

user	Route Match	Route Error 1	Route Error 2
1	45.3	41.4	29.8
2	40.5	44.9	34.6
3	59.3	30.8	11.8
4	85.9	5.5	11.6
5	94.1	4.2	4.2
6	90.1	9.5	5.7
user	Route Match	Route Error 1	Route Error 2
1	79.3	9.2	13.3
2	83.5	6.2	13.3
3	87.8	8.9	7.0
4	90.6	9.1	5.7
5	95.4	3.0	3.6
6	93.4	5.0	4.0

Table 3: Routing Errors. The table above represents the error metrics when no user corrections are used for six different users. The table below represents the error metrics when user corrections are used.

without creating a large burden for the user. Future work will investigate a system with more methods for user input to increase the accuracy of the trip determination algorithm.

4.5.10 Errors in Trip start time and end time

During the manual correction phase, the testers did not correct the trip start or end time, due to the amount of work required to accurately log the start and end time for trips made. However, the error range in the start and end time of trips can be calculated based on the accelerometer and Wifi readings during the trips made.

In the ideal scenario, movement is detected by the accelerometer, which leads to a scanning of WiFi beacons to determine if a significant location change has occurred. It is then known that a trip has started within the 2 minute window in which these sensor events occurred. Of the 1850 trips evaluated, 48% of trips fell into this category.

In some cases it takes two to three cycles for the Wifi scan to identify a significant location change. The reason for this is if the intersection of the set of beacons seen at subsequent WiFi access point scans is null, then it is assumed that movement occurred during the time between the two scans. However, the range of WiFi access points vary in different environments, anywhere from 50m for 802.11b routers to 200m for 802.11g routers. A person can walk to start a trip and still be in range of an access point in 2 minutes. Of the 1850 trips evaluated, 36% of trips were detected within 4-6 minutes of their true start time.

In the last 16% of cases, a variety of scenarios lead to the system being able to pinpoint the start time of the trip to under 6 minutes. This includes a variety of reasons described

Predicted\True	Bike	Drive	Transit	Walk
Bike	66.1	30.8	1.8	1.3
Drive	19.2	53.1	19.2	8.5
Transit	5.0	9.4	85.0	0.6
Walk	2.5	8.6	0.5	88.4
Predicted\True	Bike	Drive	Transit	Walk
Bike	81.9	15.2	0.6	2.3
Drive	10.4	80.4	9.7	0.5
Transit	4.0	5.8	90.1	0.1
Walk	2.5	8.0	0.5	89.0

Table 4: Transportation Mode Errors. The table above represents the error metrics when no user corrections are used for six different users. the table below represents the error metrics when user corrections are used.

in Section 4.5.1, such as user behaviors such as a phone turned off, phone behaviors such as the sensors unable to obtain readings, or the nature of the trip, which are often short trips of under 1 kilometer. In these scenarios, the algorithm described in Section 4.4.9 is used to determine the start/end time of trips.

Calculating the end times of a trip is a much simpler process. While a person is moving during a trip, the duty cycles for location acquisition are much smaller, and thus can be used to determine a transition to the end of a trip. 40% of trip end times are accurate within 1 min. 44% of trip end times are accurate within 2 minutes. The reason for the increase is the sampling period of the GPS sensor is increased in certain scenarios, such as when a person is traveling greater than a particular speed, or traveling along a hotroute. The last 16% of trips the system is unable to detect the end time within 2 minutes as the problems described above in detecting the start time of a trip cascade into the end time prediction problem.

4.5.11 Falsely detected trips and missed trips

Due to the issues described in Section 4.5.1, it is often the case that trips are falsely detected, or trips are completely missed. A total of 33 trips were not identified by the system (missed trips), and 39 were identified as trips when no movement occurred (false trips). The 33 missed trips occurred were all 2 kilometers or less. Falsely detected trips occur as a result of inaccurate location points which slip through the filter described in Section 4.4.1. Certain sequences of inaccurate location points make it appear that a trip is occurring, which fools the system into generating trips. Missed trips are a result of two problems. As described in Sections 4.4.9 and 4.5.10, not all movements are captured, leading to a situation where location points exist only at the origin and destination of trips. In situations where the points at the destination have a high horizontal accuracy, it is difficult to determine if the point is a faulty reading from WiFi/cell tower positioning, or if the person actually made the trip. Thus, there are thresholds set, by horizontal accuracy and number of points seen, for which any location points under the threshold will set a valid destination. Increasing this threshold will increase the number of faulty trips made, therefore, there is a balance between missing, and falsely identifying trips.

4.6 FUTURE WORK

The research presented in this chapter represents a first step in introducing a complete, end-to-end, and battery efficient automated travel diary system which operates on smartphones, and describing the number of difficulties of using smartphones rather than GPS loggers, as the main data collection tool. Smartphones are ubiquitous and are equipped with the technology needed to be used as a data collection device in travel diary systems, but there are a number of problems that must be overcome before they can be used in a wide-scale automated travel diary system. In this chapter, large scale tests have allowed us to catalogue a large list of problems with smartphones and solutions found from real devices in the field for over 9 months. These early results from the system are promising, and more work has to be done evaluating the trip metrics defined in this chapter and analyzing how small amounts of human input can increase the accuracy of these metrics. Identifying the start and end locations of trips have shown to be very accurate, while improvements are needed to more accurately identify the start/end time of trips, route, and mode taken. First, the research showed that the start/end time of 16% of trips could not be identified within a 6 minute window. The problem here lies with the mobile application, whose parameters for detecting motion with the accelerometer need to be better tweaked. Accurately identifying the route taken is a larger research problem. With sparse location data, the amount of improvement on the current algorithm is limited. However, fusing accelerometer, and magnetometer data with location data can improve on accurately identifying routes by better identifying turns and changes in direction. , The mode determination portion of the algorithm can also be improved by adding features which come from the accelerometer and magnetometer. However, all these improvements come at a cost of reducing the battery life of phones - and the tradeoff between improving the metrics for trip determination and increasing battery usage is a tricky balancing act.

Another area for improvement is expanding upon the 6 person test to obtain more data for the accuracy of the trip determination algorithm. While validating the ground truth for trips for an extended period of time is an arduous task, better web tools can be built to help the researcher and users to make this process easier to expand the number of users who can validate trips.

Aside from these goals for future research, there are a few research directions which were not tackled in this automated travel diary system. Travel diaries always record contextual information about a trip, for example, the number of passengers in a car, or the purpose of a trip. There is work to be done to derive these pieces of data from the smartphone, either actively by prompting users, passive detection, or a combination of both. For this system to be of use to researchers building travel behavior models, trip attributes like these must be annotated. An area to explore is the use of third party APIs such as Yelp, Foursquare, or Google Places to obtain trip purpose information and design a system which prompts the user for feedback on their trip purpose.

5 The Quantified Traveler: Changing transport behavior with personalized travel data feedback

5.1 BEHAVIOR CHANGE OPPORTUNITY IN TRANSPORTATION

There exists a large body of work dealing the with the modeling of transportation behavior and also attempting to influence travel behavior, and we believe there is a new opportunity to advance the body of research even further because of recent advances in technology over the past few years. In this chapter, we present a system and experiment which are designed to collect statistics on study participants' travel footprint (emissions, calories burned, time and costs) and to feed back that information in a personalized, informative way. The goal is to explore the possibility of influencing people's awareness, attitudes and potentially behavior, and to encourage them to engage in more sustainable transportation behavior. This study is informed by previous research in the area of behavior change that has shown some success; in particular, personalized feedback programs that have been able to achieve certain changes in behavior. However, given recent developments in mobile technology and behavioral economics, we believe it is time to revisit this issue. The four major technological trends described in Section 1 have given rise to advances in psychology and human computer interaction which has lead to a growth in academic research involving new methods of behavior change: persuasive technology, the proliferation of smartphones, the rise of self-tracking and the Quantified Self movement, and open, accessible transportation data.

The first three trends have already lead to new research in the fields of health and fitness, as well as environmental conservation. This chapter describes how we have taken advantage of these four trends to develop and evaluate a system which influences users' awareness about their travel behavior, intentions to take sustainable modes of transportation, and actual behavior. We developed an automated travel diary system which used smartphones to collect location and other sensor data and sent data to a server which processed the raw data into trips and attributes of each trip, including travel mode, travel time, cost, and CO_2 emissions among other statistics. The technical details of the infrastructure and algorithms are discussed in Section 4. Using this smartphone travel diary system, we ran a three-week experiment with 135 participants, in which they were presented a webpage which gave users feedback on their personal travel statistics and comparisons to various peer groups. Section 5.2 describes related work in behavior change and persuasive technology in non-transportation fields, which influenced the design of our feedback system. Section 5.3 describes the smartphone travel diary system in more detail, only recently made possible due to advancements in smartphone technology and accessible transportation data. Our experiment, called the Quantified Traveler, along with the evaluation of the experiment and changes in users' awareness, attitudes, intentions, and behaviors are outlined in Section 5.4. We recognize that this work is just the first step in a long process to influence travel mode choice, and ideas for future work are presented in Section 5.5.

5.2 LESSONS FROM PRIOR BEHAVIOR CHANGE WORK

In this section, we review prior work which allowed us to develop and design the system, website, and surveys. The first step was to learn about the successes of prior behavior change studies in transportation, which have been ran for the past 20 years. These studies were successful, but had many difficulties, with room for improvement thanks to modern smartphone technology. The design of the system and implementation of the behavior change techniques was also influenced by other studies non-transportation persuasive technology experiments, while the design of the surveys were influenced by models of behavior change from the psychology literature. Based on this prior work, our contribution to the literature is the design and evaluation of an updated version of a Travel Feedback Program using smartphones and web technology.

5.2.1 Behavior Change in Transportation

There is a large body of work in Travel Feedback Programs, which aim to influence mode choice behavior with information and psychological factors. There are various styles of Travel Feedback Programs, but common to all programs are: users receive feedforward information (i.e. directions for using alternative modes) as well as feedback information (i.e. amount of CO2 emitted) which is gathered from program participants filling out travel diaries. There have previously been several small-scale travel behavior feedback programs conducted by researchers in Japan [140]; in those experiments, the feedback was based on paper-and-pen surveys, and participants were often given feedback during face-to-face contact with a "travel coach". It could be shown that through travel feedback programs, measurable and lasting shifts away from automobile use and toward more sustainable modes of transportation could be achieved. Individualized marketing has also succeeded in changing travel behavior towards more pro-environmental modes of transportation [141].

Although Travel Feedback Programs have been shown to be successful in inducing a mode shift away from auto usage, the implementation of these programs is not scalable. Manually entering information in travel diaries is a time consuming process. Using smartphones for data collection provides an opportunity to partially or fully automate many of the tasks involved in the travel feedback programs, and do things that was previously not technically feasible. There is an opportunity to leverage the lessons learned from HCI researchers in the design of automated tools on the web and smartphones to motivate behavior change, starting with the integration of open transportation data into persuasive technology programs.

5.2.2 The Quantified Self : Self tracking to change behavior

The concept of the Quantified Self describes applications which enable the process of recording behavior, processing the data collected, and feeding it back to the individual or group so that they can better understand the patterns of their activity and eventually adapt their behavior more intelligently than they would without these augmentations. Recently, the increasing abundance of low-cost sensing devices(including smartphones), coupled with the use of social networks, mobile devices and web-based applications for many different aspects of daily life (e.g., banking) has led to an abundance of detailed data becoming available to end-users. This has given rise to many companies which have incorporated self-tracking

and behavior change into their products: Zeo - tracks sleep patterns, Fitbit - fitness levels, RunKeeper - jogs and runs, CureTogether - reactions to various medication, Mint - personal finance, RescueTime - time usage and productivity.

The Quantified Self website and regular meeting groups across the country have become active forums where people exchange ideas, experiences and findings about themselves. While there have been significant advances in self-tracking applications for health and fitness, there has been a relative lack of work on quantifying one's travel behavior.

5.2.3 Self-tracking potential in transportation

The maturity of GPS tracking technology and the surge in self-tracking interest present a new and powerful opportunity to collect traveler data by combining these two areas. In our system, smartphone technology is used to collect data through continuous, unobtrusive sensing with minimal effort required from the traveler. With large amounts of individual travel behavior data, the research community can also model behavior and manage demand by getting a better understanding of how and why people travel.

There are also considerable benefits to introducing self-tracking in transportation: Many of the costs are not paid when the trip is made, but are hidden in infrequently paid items such as car insurance fees or the price of a season parking pass. Emissions are not routinely measured and quantified, and since travel is an induced activity that is conducted for the purpose of another desired activity (e.g., shopping), many travelers may not be making a conscious, informed decision about how much time they want to spend traveling, or how many calories they want to burn when traveling. However, transportation decisions have a large impact on people's lives. For instance, the average Californian household spends around 15% of total income on transportation [142, 143], and the average Californian spends 26.5 minutes per day commuting; this adds up to more than 110 hours per year - almost as much as a typical worker's yearly vacation[144]. If people were to track their own travel patterns and attempted to reduce travel time or costs, for example, this could also benefit society as a whole as it may catalyze a reduction in vehicle miles traveled or a shift to more sustainable modes of transportation. Raising awareness of negative impacts of transportation on the environment and public health can be considered a policy tool in its own right to reduce the overall footprint of the transportation sector. The introduction of automated self-tracking devices presents a new, powerful method to present every traveler with personalized information on their specific contributions.

5.2.4 Recent Examples of Technology Designed for Behavior Change

Many researchers recognized the potential of implementing many of the behavior change techniques into computer technologies, including providing personalized feedback on actions measured with smartphones. Two very active fields in which Human Computer Interaction researchers have built and evaluated applications are in Health/Fitness and Energy Conservation/Eco-Feedback Technology. Studies using these applications have shown that feedback is a powerful behavior change technique in health and fitness applications [145, **?**], and eco-feedback technologies [146].

Health and fitness researchers have tried using goal-setting and feedback to design appli-

cations that help people maintain healthy lifestyles. One of the most notable applications was Ubifit[147], which automatically detected the physical activity levels of a user wearing a custom device, and also provided feedback to users. One of the notable features of Ubifit was the simplicity of the feedback: the person's cell phone background changed depending on the amount of physical activity, such that a user could understand their data at a glance. Ubifit is one example of numerous applications which have evaluated the effects of Goal-Setting and Feedback in monitoring fitness[148, 149, 150]. There are also numerous examples of applications of eco-feedback technology, which have been designed to change behavior[151, 152, 146] and successfully shown that feedback has an effect to conserve energy. While transportation plays a role in potentially reducing one's environmental impact on the earth, only one behavioral HCI study has been conducted to influence transportation behaviors[153] and showed a high potential for behavior change to sustainable modes of transportation.

5.2.5 Understanding Behavior Models to Measure Aspects of Change

While the goal of a system may be to design behavior, it is just as important to measure the factors which contribute to behavior change.

theory of Planned Behavior showed that behavior was influenced by attitudes, normative beliefs[154], a person's level of self-efficacy[155] and a person's past experiences, social persuasion and emotional states[156]. Later models based on the TPB expanded the influences on one's behavior to habits, environmental constraints, knowledge and skills to perform behaviors, and moral obligations[157, 158, 159]. Thus, our system was designed to surveys were designed to capture these influences.

Other behavior change models such as the TransTheoretical Model, which used people's attitudes and levels of self-efficacy to classify people in to different stages of change[160]. This model is notable because it recognizes that behavior change techniques have different effects on people in different stages of change. While education and information may be the most effective technique in influencing someone at the very early stages of change, goal-setting may be a better technique for one attempting to maintain a changed behavior.

While acting "green" has been a popular movement for a while, in transportation most people are not actively attempting to change their modes of transportation or distance traveled. Therefore, the feedback website was designed as a educational and information page, and an attempt was made to measure a change in education by surveying if one's "awareness" of their emissions, calories burned, and other factors changed.

5.3 THE QUANTIFIED TRAVELER SYSTEM

We built an automated travel feedback system based on the work of prior researchers and new technology using smartphones. The system consists of many parts, from smartphones to collect data efficiently without draining the battery, server infrastructure to receive data and handle large loads of data requests, algorithms to process data into trips and attributes, and a carefully designed website which presented a large amount of data to users in a concise graphical manner. The first three parts are briefly described in this section, with further described in Section 4 while the website is described here.

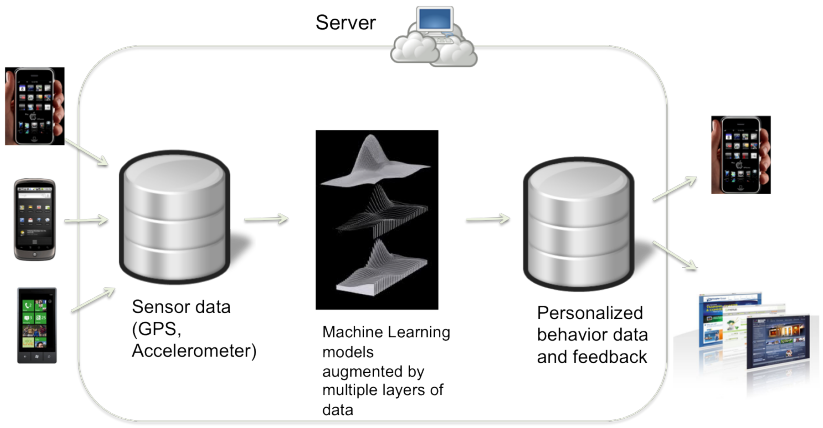

Figure 22: System Architecture Diagram. The components of the system consist of mobile phones, trip determination algorithms running on a server, and web tools to view and correct trips.

5.3.1 Architecture and Data-flow

The design of the data collection and feedback system is shown in Figure 22. It consists of three components: the tracking application on smartphones, the server architecture to handle incoming location data and handle data requests, and the analytics software to transform the raw data into trips made and meaningful statistics and information about those trips. The applications running on the participants' phones collect raw sensor data and upload it to a cloud-based server which saves it to the database. A nightly job reads the raw data, processes them to infer trip origin, destination, start time, end time, route and travel mode. Trips are further augmented with data such as addresses/neighborhoods of trips made, distance traveled, time spent traveling, CO_2 emitted, calories expended and travel costs. The methodology for computing the last three of these items is detailed in Table 5.

As described by Section 4, the trip determination system consists of a smartphone application that runs on Android phones and iPhones, in a battery efficient manner. A person who travels 2 hours a day experiences on average 33 hours of typical usage (including texting, talking o the phone, browsing the web, and using apps). The application runs in the phones' background and periodically uploads location data to a server which runs a trip determination algorithms to generate trips and statistics to be shown on the website.

5.3.2 Website Design

Users of the data collection system are given access to a website on which they can view their individual travel behavior data in different ways. Specifically, the website consists of four pages, as shown in Figures 23 - 26. After logging onto the website, participants are

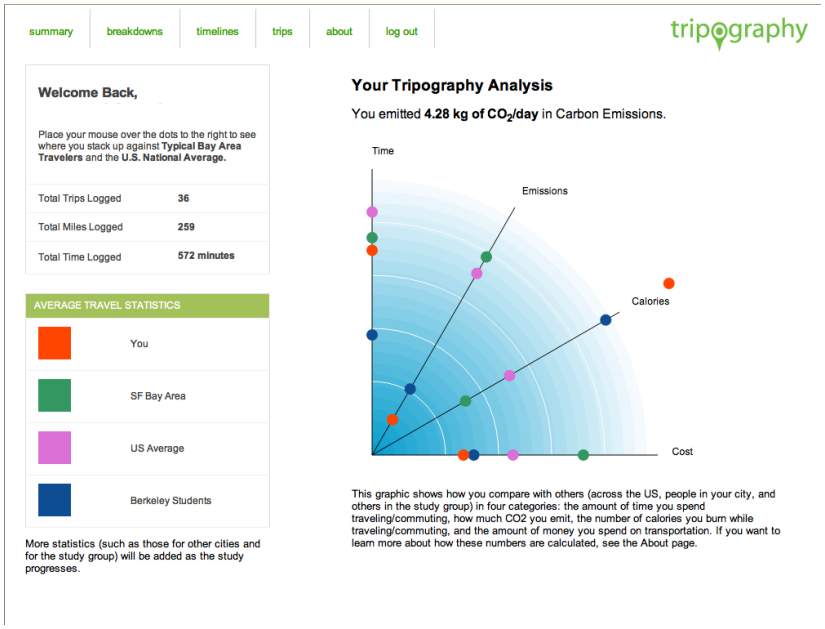

Figure 23: The summary page, which shows a person's travel stats and comparisons with peer groups.

first presented with a "summary"page, which presents an overview of their aggregate travel data and a brief explanation. The main metrics which are shown to the user are emissions, calories burned, cost, and travel time, for their own data and a set of groups they can compare themselves to, the average America, the average resident of the San Francisco Bay Area, and the average of other people in the study. In this page, our goal was to deliver a summary of one's travel history such that a person could quickly glance at the page and immediately understand their data and trends. From there, users can access three pages: One that explains the scoring methodology, one with the detailed daily history of scores and one with the trip history (termed "Tripography"). The "Breakdown" and "Timelines" page provide more detail from the summary page and present the information in a different manner. The "Breakdowns" page focuses on describing a person by his mode split, while the "Timelines" page shows how a person's travel statistics change over various periods of time.

The "Trips" page shows a history of one's trips on a map by calendar day, and also allows users to correct the travel mode of any incorrectly predicted trips. Changes entered into the system also update the travel statistics shown on the various feedback pages to give the most accurate view of the user's travel behavior data.

5.4 EVALUATION

To evaluate the effectiveness of the system, we designed and conducted an experiment in the San Francisco Bay area with 135 participants. The participants were recruited from the subject pool of the UC Berkeley XLab (the "Experimental Social Science Laboratory"),

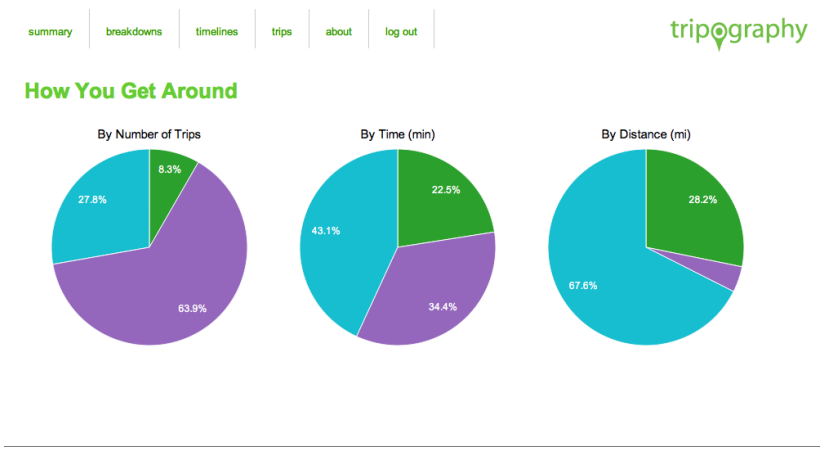

Figure 24: The breakdowns page, which shows mode split by trips made and distance traveled.

which is run by the Haas School of Business. This is a computer laboratory for conducting human-subject experiments. The lab maintains a subject pool of over 2500 members, all of whom are UC Berkeley affiliates and most are undergraduate students. Xlab administration handles the recruiting and requires that researchers provide subjects with participation fees of around $15/hour.

For recruitment, we reached out to both students and staff at UC Berkeley. In total, 111 students and 24 members of staff participated in the experiment, of which 37 were male and 98 female. All participants owned smartphones; 82 were iPhones and 53 Android phones. Of the 135 participants, 121 completed the two surveys that were given (see below). These surveys showed that 106 (94%) were between 17 and 29 years old. Of those 113 participants, 44 had access to an automobile, 17 regularly biked, 101 regularly used transit and 103 had a transit pass.

This three-week experiment ran from March 18 to April 7, 2012. We utilized the infrastructure described in Section 5.3 for collecting traveler data via smartphones, and the goals of the experiment were twofold: first, it was intended to demonstrate how the Quantified Self movement can be leveraged by the travel demand modeling community for data collection, with a positive outcome for both sides. Second, as stated above, the experiment was designed to investigate whether a data collection effort that includes elements of traveler feedback and a direct engagement of subjects via a website can lead to a change in behavior toward more sustainable modes of transportation. If that proved to be possible, it would provide evidence of the viability of automated, web-based traveler feedback on a large scale for use as a policy tool to promote more sustainable transportation.

We chose to focus on studying the effect of feedback and peer influences on (1) attitudes towards sustainable travel and (2) awareness of the impacts of one's own travel behavior. The latter comprises both awareness of absolute values (e.g., amount of emissions) and an awareness of where the person stands compared to average Americans, San Francisco Bay Area residents and to their peer group, which in this case was the group of survey participants from the XLab. In addition, we were interested in using this experiment as a learning

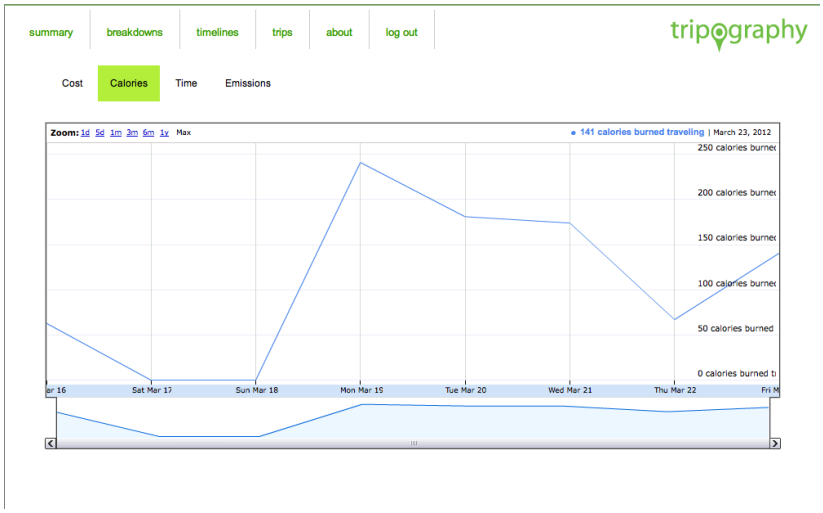

Figure 25: The timelines page, which shows a person's change in travel time/emissions/calories/cost over time.

experience for the long-term research goal of using technology to persuade individuals to use more sustainable transportation modes.

5.4.1 Experimental Design

At the beginning of the study, the participants were asked to fill out a survey about their awareness of transportation impacts and their attitudes toward sustainable modes of transportation. This was followed by a three-week period in which the users were tracked. During the first week, participants received no feedback information until the seventh day, when they were sent a link to a website on which they could view their trip history and a set of personalized statistics (explained in section 5.3.2) related to their travel patterns. Following this, the students went on a spring break, in which trips were recorded, but the data shown in users' "summary" page did not reflect the trips during spring break. Users then received another email reminder to log into the site and view their data. On average, the participants logged in 4.1 times during the final week of the experiment. At the end of the three weeks, participants were asked to fill out a survey that contained the same questions as the first one, but with an added section asking them for feedback on the website. The following sections describe the components of the experiment in detail.

5.4.2 Recorded Activity

During the three-week study we recorded a total of 8607 trips and 115,169 miles of travel across 6 different modes: walking, biking, taking the bus, taking the train, and driving. Plane trips are included in the amount of miles logged but ignored in the calculation of trip statistics. The breakdown of the usage of modes is shown in Figure 27. As described in Section 5.3.2, study participants corrected their trips on the website if the system incorrectly predicted their travel mode. Out of the 8607 trips, 13.5% of trips were corrected.

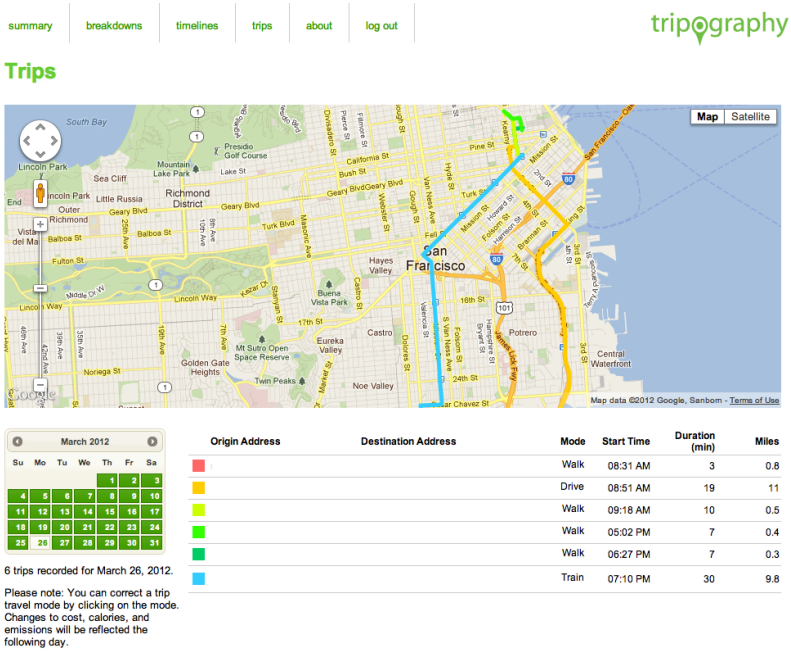

Figure 26: The trips page, which shows all trips for a calendar day on a map. The addresses are blanked out in this figure.

5.4.3 Measuring Attitudes

The field study provided valuable information with respect to both of its goals. The surveys showed that there was a potential for increasing participants' awareness of their transportation carbon footprint via the personalized statistics, trends, and trips shown on the website.

5.4.4 Survey Questions and Baseline Results

The design of the survey was based on measuring factors contributing to behavior change as identified by the Integrated Behavior Model: Experiential attitudes, Instrumental attitudes, Injunctive Norms, Descriptive Norms, Perceived Control, Self-Efficacy, and Moral Obligation for public transit, biking, driving, and walking attitudes on a 7-point Likert scale. These questions were based on common travel surveys used by the transportation community[164], and categorized into the various components of the behavior model.

The survey consisted of 55 statements which participants agreed or disagreed with on a seven-point Likert Scale. 11 of these questions were asked about the participant's level of awareness of their own travel behavior. This included questions about their knowledge of the amount of CO2 emitted by their daily transportation habits, the number of calories burned

Mode	CO_2	Calories	Costs
Walking	0	Used a calories calculator which adjusts calories burned by walking speed [161], assuming a 150lb person.	0
Biking	0	Same as above.	0
Driving	Used a CO_2 calculator for driving [162].	0	58.6 cents / mile [163].
Train	Averaged to 39g/mile [162].	0	Appropriate costs for taking BART or Caltrain as specified by the respective transit agencies.
Bus	Averaged to 25g/mile [162].	0	Same as above.

Table 5: Methodology for calculating trip footprint

by traveling, their amount of time spent traveling, and the amount of money they spent on transportation. Based on ideas from theory of planned behavior, 3 questions were asked about participants' willingness to set goals to change travel behavior. Sample questions for these various categories of questions are listed in table 2. The baseline (pre-experiment) survey showed that participants responded positively to questions asking about their environmental sentiments (M = 4.73, SD = 1.73; on a scale from 1 to 7 where 7 corresponds to the strongest pro-environmental attitudes), but on average, they slightly disagreed with the statement that they engaged in more sustainable travel behavior than the average person at UC Berkeley (M = 3.22, SD = 1.40). Furthermore, participants responded positively to statements about health, exercise and the possibility of burning calories while traveling (M = 5.09, SD = 1.45). The results of the survey also showed that the participants were not very aware of how "green" they actually were (M = 4.41, SD = 1.69; 1 corresponding to completely unaware, 7 corresponding to very aware). In particular, the participants didn't know the amount of CO_2 they emitted, the amount of CO_2 emitted by the average person in San Francisco, nor the magnitude of the impact of their emissions. This showed that there was an opportunity for education on the environmental effects of using different transportation modes, and that this was a group that was generally motivated to engage in sustainable behavior.

Due to the heterogeneous nature of the participant pool (students and staff), we further decomposed the group by their self-reported planned mode use during the months following the experiment. Specifically, we created the following three groups, based on their planned auto use:

- People who were planning to use a car once a month or less (including never): Non-

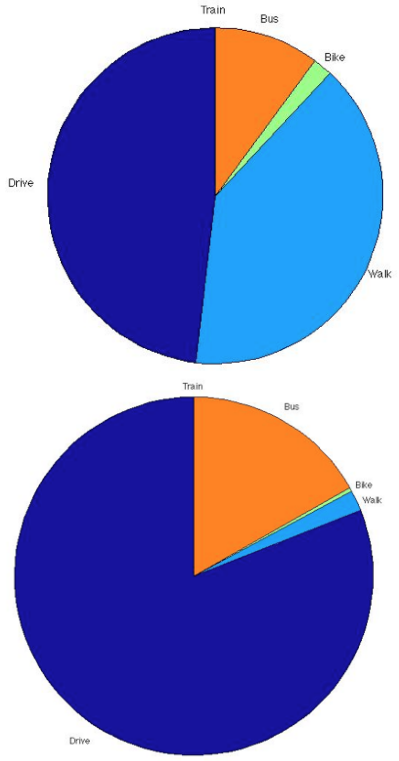

Figure 27: Mode Split of all trips recorded in 3-week period. Top: Mode Split measured by number of trips made. Bottom: Mode Split measured by distance traveled.

drivers (55 observations)

- People planning to use a car between once a month and once a week: Multimodal travelers (29 observations)
- People planning to use a car more than once a week: Drivers (29 observations)

We found that in terms of their attitudes toward environmental and health issues, there was no significant difference between the three groups.

A last set of questions was related to people's intention to change their transportation behavior in the future. On average, the study participants slightly disagreed with statements suggesting that they were going to change ($M = 3.83$, $SD = 1.44$), and there was no noticeable difference between the three mode use groups.

5.4.5 An increase in awareness, changes in intention, but not in pro-sustainability attitudes

We were able to measure statistically significant changes in participants' awareness of environmental, health, financial and time impacts of travel as well as their attitudes towards sustainable travel behavior. The questions in the survey were divided into 6 categories:

Category	Sample Question
Awareness	I know how much CO_2 I emit from my daily transportation.
Self-Efficacy	I can get exercise when traveling.
Perceived Norms	My friends actually engage in sustainable transportation behavior (carpooling/biking/walking/taking public transit)
Setting Goals	I would consider setting a goal to reduce my carbon footprint.
Attitudes on Sustainable Behavior	I value the benefits to society when I take sustainable modes of transportation.

Table 6: Sample questions given to participants at the beginning and end of the study

awareness, perceived norms, goal-setting, attitudes towards sustainable behavior and towards health benefits of sustainable transportation modes. A paired t-test was run to compare the pre- and post-experiment results for each individual question. In addition, the questions were grouped together to create composite scores for the five categories. A Hotelling's T-Squared test for two multivariate independent samples with unequal covariance matrices was carried out to compare composite scores for the five categories between the pre- and post-experiment survey. The different questions in each of the categories correspond to different variables that are correlated and the T-squared test is used to see if there is a significant difference in these categories.

Example statements which showed a statistically significant ($p < 0.001$) increase in awareness of the amount of CO_2 emitted were: "I know how much CO_2 I emit from transportation" and "I know how much CO_2 the average person in my city emits from transportation". While all awareness questions (environmental, health, financial and time) showed an improvement of awareness over the study period, the change was the strongest with respect to the environmental footprint. On the other hand, even though we detected a positive shift in some of the attitude questions related to sustainable travel behavior, the overall shift in this category was statistically insignificant. An example statement (with a p-value of 0.19) was: "We should raise the price of gasoline to reduce congestion and air pollution." There was also no significant difference between the three mode use groups.

Finally, a set of questions was asked regarding people's intention to change their travel behavior in the future towards using more sustainable transportation modes. A example statement was: "I am certain that I am going to change my transportation behavior for the benefit of the environment". Overall, the average answer for this group of questions was "neither agree nor disagree", and it did not change significantly during the experiment ($p = 0.46$). However, in a related question, people were asked to indicate whether they were planning to increase or decrease their use of certain modes in the future. It was interesting to find that even though people did not state explicitly they had an intention to change, the planned mode use of drivers revealed significant shifts. Specifically, after the experiment, drivers reported that they were planning to walk more and drive less compared to what they had answered before the experiment. The exact question was "Over the next few months,

		Pre-survey		Post-survey			
Participant type	Mode	M	SD	M	SD	t stat	p-val
Drivers	Driving	4.76	1.4	3.86	1.55	5.31	0.02
	Walking	3.86	0.87	4.79	1.56	7.8	0.01
Non-drivers	Driving	3.8	1.35	3.96	1.36	0.4	0.53
	Walking	4.45	1.03	4.75	1.43	1.49	0.22
Multimodal	Driving	4.31	0.96	4.34	1.07	0.02	0.89
	Walking	4.68	1.13	4.48	1.15	0.47	0.49

Table 7: Answers to question on future mode use

and compared to what you do now, how often do you intend to use the following modes of transportation for commuting/traveling?", and the possible answers, on a scale from 1 to 7, ranged from "much less than now" to "much more than now". Answers are shown in table 7. Planned use of other modes (carsharing, biking, transit) did not change. Multimodal participants and non-drivers on the other hand did not state mode use plans after the experiment that were significantly different from those before the experiment.

While this does not mean that people were averse to change, it appears that they were unsure about how they might change in the near future, and the feedback provided during the survey did not have a statistically significant influence on that.

5.4.6 Classification of participants

We suspected that participants viewing their stats on the website would be influenced by whether they were above or below the average of their peer group on the four dimensions plotted (emissions, calories, money and time), and by how far from the mean they were. This led us to conduct an analysis of how the participants in our study compared to the mean of their group. Given 16 possible combinations (above or below the mean on each of the four dimensions), the study participants can technically be divided into 16 groups. However, we found that in our population, only 7 groups were represented, and of those, only 4 were larger than 5 people. This is shown in table 8; a "+" means that the participant was above the mean in that category, whereas a "-" is below the mean.

Which group a person falls into is related to that person's travel-related lifestyle, which we shall call mobility style. Groups 1 is worst on all four dimensions and likely represents habitual drivers. People in group 2 are quite similar to group 1, except that they burn more calories. It is likely that these are high-mileage travelers. The opposite is group 3, which is consistently below the average on all four dimensions and presumably includes low-mileage travelers, which is plausible since many participants live near campus and only travel minimally. Groups 4 and 5 might be multimodal travelers who combine transit with walking and biking, while group 6 is too small to meaningfully interpret. Finally, group 7, which is the largest group, is best on all four dimensions. This group is likely characterized by very green travel behavior, predominantly walking and biking.

Group	Observations	Time	Calories burned	Emissions	Costs
1	15	+	-	+	+
2	7	+	+	+	+
3	26	-	-	-	-
4	5	+	-	-	-
5	5	+	+	-	-
6	2	-	-	+	+
7	30	-	+	-	-

Table 8: Mobility styles among the study participants; "+" = above mean, "-" = below mean.

	First week		Last week		
Mode share	M	SD	M	SD	p-val
Drive	0.63	0.35	0.55	0.36	0.08
Walk	0.19	0.26	0.27	0.29	0.03
Bike	0.03	0.10	0.03	0.08	0.99
Bus	0.14	0.23	0.14	0.21	0.83
Train	0.01	0.01	0.01	0.04	0.15

Table 9: Mode split by distance traveled, showing the significant shift from driving to walking

5.4.7 Measured behavior change

Interestingly, our analyses of participants' travel behavior showed a shift in mode usage. Table 9 shows the mode split by distance traveled for the first and last survey weeks. Most importantly, we observed a significant decrease in driving and a significant increase in walking. In other words, a number of trips were shifted in the direction of lower emissions, higher calories, and lower costs. Regarding the shares of other modes, there was no significant change in the use of bikes and buses, and there was a slight, though not significant, increase in train rides (not visible in the table due to rounding).

In order to gain a better understanding of the potential influences of the feedback provided on the website, we built several linear regression models. The individuals' distance from the mean, averaged over all four dimensions, were regressed on either the participants' changes in time, emissions, costs and calories burned between the first and the last survey week or on the log-odds ratio of walking as the dependent variables. Every group described in section 5.4.6 was represented by an explanatory variable. Additional variables were binary variables indicating whether somebody was female and a student, as well as the number of logins to the website by that person. The goal of the regression model was to understand whether the different groups reacted differently to the presentation of the feedback, and which ones were most likely to shift their behavior. Unfortunately, the regression analysis was complicated by the strong correlation between the four feedback dimensions, and even though we had observed a significant shift from driving to walking at the aggregate level, we were not able to find statistically significant covariates explaining what types of people and travelers were

more likely to shift.

5.4.8 Feedback about the website

To evaluate the website, the post-experiment survey contained an additional set of questions regarding the participants' evaluation of the website. They were asked to login to the website and respond to the following four statements on a 7-point Likert scale about the website.

1. I enjoyed taking a look at my dashboard/statistics/trip history page and getting a summary of my travel (M = 5.37, SD = 1.2)

2. In the future, this web page is something I would consider using. (M = 5.87, SD = 1.01)

3. If I were to set a goal to change my travel behavior (be greener, reduce cost, travel less), I consider this web page helpful. (M = 5.12, SD = 1.28)

4. This web page was easy to use. (M = 5.3, SD = 1.13)

Overall there was positive feedback on the website and the subjects liked the presentation of their trip data. In particular, they reported that they enjoyed seeing their transportation data and enjoyed seeing how they compared with their peers. Users were also asked which webpage they liked the most, and there was a strong preference for the summary and trips page: Summary: 38.46% Breakdown: 15.38% Timeline:6.84% Trips: 39.32%. On average many of them reported that they would consider using the webpage in the future and felt that it would help setting a goal to change their travel behavior.

5.5 CONCLUSIONS AND FUTURE WORK

This chapter showed the reasoning behind the designs of an automated smartphone travel diary system and the subsequent experiment showed how feedback on one's travel history can affect one's awareness of their impact on the environment, and for some segments of the population, intentions to change behavior, and actual behavior change. In addition, we were able to show that an automated travel diary system using smartphones along with a web interface to view trips could successfully collect data from 135 participants and process location data into trips, which was used to both influence behaviors, intentions, and awareness and build a travel behavior model.

The next step is expanding the study from 3 weeks to a longer duration. A number of other variables may have contributed to the measured changes in behavior, such as the participants' mindfulness of being tracked and taking part in a study, or even a novelty effect of using a new application. The measured change in behavior is a step in the right direction, and the real test of a behavior change system is prolonged and maintained changes in travel mode usage over a much longer study period.

The experiment we ran used two behavior change strategies of feedback and comparison. This was an initial attempt into the concept of using technology to influence users' transportation behavior, and there exists a large number of other behavior change techniques which have been successfully applied in other fields that could have been experimentally

incorporated in our system. Behavior change techniques which have been recognized and tested include: Information, Goal-Setting, Comparison, Incentives, and Feedback [165]. All of these techniques have been tested in a variety of fields and largely confirmed to have positive effects for behavior change. The next step involves analyzing successful experiments which have used one of these techniques and molding it to our problem of nudging people towards sustainable modes of transportation.

6 Thesis Conclusion

6.1 IDENTIFYING PROBLEMS AND USING TECHNOLOGY TO SOLVE THEM

The work presented in this thesis covers two related but different areas, improving transit information delivered to users, and tracking and influencing people's travel behaviors. They both follow a similar formula:

1. Identify a pain point in the transportation world(of users of the transportation system and researchers)
2. Propose a solution using smartphone technology and behavior change techniques
3. Build the system and describe the steps on how to do so
4. Evaluate the system to see if it achieved the goal of solving the problem identified in the first place.

The first problem that I identified was that people were not using the public transportation system as effectively as they could. There was a new wave of GPS devices installed on buses, and access to the information was made available to the public, but this information was being used to only inform users when the next bus was coming. There was potential to solve the main goal: get users to their destination in the fastest way possible with the current state(position of buses) of the transportation system, without them having to know any information about the network.

The proposed solution was a new routing algorithm that routed users in a network according to the realtime positions of buses, not based on the schedule, like all other trip planners discussed in the literature up until this point. The routing algorithm was made into part of an iPhone application called BayTripper, which served over 1,000 active users. The routing algorithm also proved not just an innovative algorithm, but it also saved time as well. Studies showed that the value of real-time data in the trip routing application was immense: it gave users more accurate predictions of their expected travel times, and suggested faster alternative routes to use. Another contribution was made by solving this problem in a real-world network, and showing how the algorithm could scale to other larger networks as proved by a large scale test of 10,000 simulated trips in 77 different real world networks, including large systems as Washington D.C., New York City, and Los Angeles.

The second problem I identified focused solving a problem for both users and researchers. Getting people to "be green" has been a ongoing goal of both researchers and people with eco-friendly ideals. Changing behavior is a long and complex process, and before starting to influence behavior, researchers needed to understand how users currently behave. Thus, there was a sub-problem identified. Researchers needed a more efficient way to collect data on how people travel, and the current state of the art (using either pen and paper surveys or GPS loggers to collect travel behavior data) was both expensive and not scalable to long-term studies.

The proposed solution to this sub-problem was the creation of an automated travel diary system which used smartphones as the tool to collect data. While smartphones were an

obvious solution to the traveler data collection problem due to their widespread use and ease of distributing applications, there were a number of problems to overcome, which previously had not been discussed in the travel diary literature: collecting location data with the constraints of a limited battery life, processing data into trips with sparse, and often inaccurate data, and high variations in data accuracy and collection with different models of phones, geographic locations, and user behaviors. These were all discovered and solved with a large scale evaluation across 20 states and 125+ users. A smaller scale evaluation proved that it was possible to accurately generate trips from smartphone data. This contribution allows future researchers to easily collect data from large sets of users, and over long periods of time, using smartphones. The collected data can be used for a variety of purposes, such as building models, or influencing behavior, the main problem I aimed to solve.

Once this data collection sub-problem was solved, I took a crack at the larger behavior change problem. Using the smartphone based travel diary system built, I proposed a feedback system in which users received personalized feedback about how they traveled. The design of the website and survey questions were informed by prior research in psychology and related behavior change studies in health and fitness, and eco-feedback. The designed webpage allowed users to view their data over time and along with various peer groups, average Americans, average residents of the San Francisco Bay Area, and their peers taking part in my experiment.

The system was shown to be effective: users increased their awareness about the way travel and their impact on the environment, a subset of the group, primary drivers, showed their intention to drive less and walk more, and the data showed that they actually changed their behavior for a short period, Although there is a lot more work that needs to go into showing a sustained, long-term behavior change. A short term success is a step in the right direction, but there are a number of proposed solutions to take my research further.

6.1.1 SUGGESTIONS FOR FURTHER RESEARCH

The baseline technology for conducting traveler behavior experiments has been built, but there are some algorithmic improvements that need to be made before running large scale behavior change experiments. The experiment in Section 5 was a successful experiment and test of the automated travel diary system, but the participants were still asked to view and correct their trips daily. While only 13.5% of trips were corrected, having users check their trips instead of just their feedback pages is not ideal. The most promising research area is combining small amounts of user input about their trips to boost the accuracy of the trip determination system. As shown in section 4, user inputs were able to boost the accuracy of the route and travel mode metrics significantly. The development of an interface as well as algorithms that fuse user feedback with the trip determination algorithms is the most promising technique to improve upon the system.

While there are areas in algorithm development and technology that can be addressed, there are also many potential experiments that can be run. There exists a broad spectrum of techniques to influence traveler behavior: Information, Goal-Setting, Comparison, Incentives, Feedback, and Prompts. These techniques can be used in different ways and different applications. In the first half of my research I focused on improving Information to transit riders. For this group, information as well as feedback about their travel along different

routes is a potential research area. A technique that I am most excited about is prompts. Prompts are timely messages of either information or feedback which influences a person at the time of decision. For example, a person walking out the door to work typically does not consider his mode of transportation, but if prompted about alternatives at the moment he leaves the door, may be persuaded to changes depending on transportation conditions. This technology is already made possible by my prior work, and a system designed to deliver prompts would be an interesting step, However, it has been studied that prompts work best on those people who are motivated to change and have the ability to change. There are other groups of people who lack the motivation or ability to change. For these groups, I propose attempting to use incentives along with different methods of feedback and information to understand what it may take for a person to change travel mode, and change behavior. Understanding a person's decision making process and utility of various modes is critical to changing behavior.

With a strong technology infrastructure that I have built, it is possible to conduct these further experiments for long-term studies. The work presented in this thesis has only laid the groundwork to learning more about travel behavior - the innovative experiments are still to come.

References

[1] D. B. Work, O.-P. Tossavainen, S. Blandin, A. M. Bayen, T. Iwuchukwu, and K. Tracton, "An ensemble kalman filtering approach to highway traffic estimation using gps enabled mobile devices," *47th IEEE Conference on Decision and Control, Cancun, Mexico*, 2008.

[2] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. B. Work, J. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson, "Virtual trip lines for distributed privacy-preserving traffic monitoring," in *MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services*, Breckenridge, CO, June 17-18 2008, pp. 15–28.

[3] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: using a mobile sensor network for road surface monitoring," in *MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services*, Breckenridge, CO, June 17-18 2008, pp. 29–39.

[4] E. Horvitz, J. Apacible, R. Sarin, and L. Liao, "Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service," *In Twenty-First Conference on Uncertainty in Artificial Intelligence, (UAI-05), Edinburgh*, 2005.

[5] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Trafficsense: Rich monitoring of road and traffic conditions using mobile smartphones," *Technical Report MSR-TR-2008-59*, 2008.

[6] J. Yoon, B. Noble, and M. Liu, "Surface street traffic estimation," in *6th International Conference on Mobile Systems, Applications, and Services*, San Juan, Puerto Rico, June 11-14 2007, pp. 220–232.

[7] K. Rehrl, S. Bruntsch, and H. Mentz, "Assisting multimodal travelers: Design and prototypical implementation of a personal travel companion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 31–42, 2007.

[8] Graphserver, "The open-source multi-modal trip planner." *http://graphserver.sourceforge.net/.*

[9] "Service standards report," San Francisco Municipal Transportation Authority, Tech. Rep. Q3 FY08, July 2008.

[10] M. Hickman and N. Wilson, "Passenger travel time and path choice implications of real-time transit information," *Trans. Research C*, vol. 3, no. 4, pp. 211–226, 1995.

[11] R. Mishalani, S. Lee, and M. McCord, "Evaluating real-time bus arrival evaluating real-time bus arrival information systems," *Transportation Research Record*, vol. 1731, no. 00-0739, pp. 81–87, 2000.

[12] B. C. M. O'Mahony, "An examination of the public transport information requirements of users," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 21–30, 2007.

[13] M. D. Rossetti and T. Turitto, "Comparing static and dynamic threshhold based control strategies," *Trans. Research A 32A, 607-620*, 1998.

[14] G. F. Newell and R. B. Potts, "Maintaining a bus schedule," *Proc. Second Conference Australian Road Research Board*, pp. 388–39, 1964.

[15] M. Dessouky, R. Hall, L. Zhang, and A. Singh, "Real-time control of buses for schedule coordination at a terminal," *Transportation Research Part A: Policy and Practice*, vol. 37, no. 2, pp. 145–164, February 2003.

[16] C. Daganzo, "How to improve bus service," *Institute of Institute of Transportation Studies, Working Paper UCB-ITS-VWP-2008-6, University of California, Berkeley, CA*, 2008.

[17] J. Xu, N. Wilson, and D. Bernstein, "The holding problem with real-time information available," *Trans. Sci. 35, 1-18*, 2001.

[18] I. P. H. Jula, M. Dessouky, "Real-time estimation of travel times along the arcs and arrival times at the nodes of dynamic stochastic networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 97–110, 2008.

[19] Shalaby and Farhan, "Prediction model of bus arrival and departure times using avl and apc data," *Journal of Public Transportation*, vol. Vol. 7, no. No. 1, pp. 41–61, 2004.

[20] Abdelfattah and Khan, "Models for predicting bus delays," *Transportation Research Record*, vol. 1623, no. 0361-1981, pp. 8–15, 1998.

[21] D. Chien and Wei, "Dynamic bus arrival time prediction with artificial neural networks," *Journal of Transportation Engineering*, pp. 430–438, September/October 2002.

[22] YouTube, "Real-time transit application," *http://www.youtube.com/watch?v=fSeTuisLw6c*.

[23] Google, "Google transit feed secification," *http://code.google.com/*.

[24] R. Ahuja, T. Magnanti, and J. Orlin, *Network flows: theory, algorithms, and applications*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1993.

[25] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[26] D. B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *Journal of the ACM*, vol. 24, no. 1, pp. 1–13, 1977.

[27] R. W. Floyd, "Algorithm 97 (shortest path)," *Communications of the ACM*, vol. 5, no. 6, p. 345, June 1962.

[28] D. Eppstein, "Finding the k shortest paths," in *Proc. the Annual Symposium on Foundations of Computer Science (FOCS)*, 1994, pp. 154–165.

[29] C. Demetrescu and G. F. Italiano, "A new approach to dynamic all pairs shortest paths," *Journal of the ACM*, vol. 51, no. 6, pp. 968–992, 2004.

[30] V. King, "Fully dynamic algorithms for maintaining all-pairs shortest paths and transitive closure in digraphs," in *In Proc. 40th IEEE Symposium on Foundations of Computer Science (FOCS)*, 1999, pp. 81–91.

[31] G. Ramalingam and T. Reps, "An incremental algorithm for a generalization of the shortest-path problem," *Journal of Algorithms*, vol. 21, no. 2, pp. 267–305, 1996.

[32] J. Miller and E. Horowitz, "Algorithms for real-time gathering and analysis of continuous-flow traffic data," in *Proc. IEEE Intelligent Transportation Systems Conference ITSC '06*, 2006, pp. 1454–1459.

[33] R. Jain, T. Raleigh, C. Graff, and M. Bereschinsky, "Mobile internet access and qos guarantees using mobile ip and rsvp with location registers," *IEEE Int. Conf. Commun.*, vol. 3, pp. 1690–1695, 1998.

[34] T. Erl, Ed., *Service-oriented architecture (SOA): concepts, technology, and design*. Prentice Hall, 2005.

[35] N. T. Conditions, *http://www.sfmta.com/cms/asite/nextmunidata.htm*.

[36] M. Muller-Hannemann and K. Weihe, "Pareto shortest paths is often feasible in practice." *In Proc. 5th Workshop on Algorithm Engineering (WAE 2001)*, vol. 2141, pp. 185–198, 2001.

[37] S. Pallottino and M. G. Scutella, *Equilibrium and advanced transportation modelling, chapter 11.* Kluwer Academic Publishers, 1998.

[38] F.Schulz, D.Wagner, and K.Weihe, "Dijkstra's algorithm on-line: An empirical case study from public railroad transport," *ACM Journal of Experimental Algorithmics*, vol. 5(12), 2000.

[39] F. Schulz, D. Wagner, and C. Zaroliagis, "Using multi- level graphs for timetable information in railway systems," *In Proceedings 4th Workshop on Algorithm Engineering and Experiments*, pp. 43–59, 2001.

[40] G. Brodal and R. Jacob, "Time-dependent networks as models to achieve fast exact timetable queries," *In Proc. 3rd Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS 2003)Electronic Notes in Theoretical Computer Science*, vol. 92, issue 1, Elsevier, 2003.

[41] K. Nachtigal, "Time depending shortest-path problems with applications to railway networks," *European Journal of Operations Research*, vol. 83, pp. 154–166, 1995.

[42] A. Orda and R. Rom, "Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length," *Journal of the ACM*, vol. 37(3), 1990.

[43] ——, "Minimum weight paths in time-dependent networks," *Networks*, vol. 21, 1991.

[44] R. Bellman, "On a routing problems," *Quarterly of Applied Mathematics*, vol. 16(1), pp. 87–90, 1958.

[45] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, pp. 100–107, 1968.

[46] E. Pyrga, F. Schulz, D. Wagner, and C. Zaroliagis, "Efficient models for timetable information in public transportation systems," *ACM Journal of Experimental Algorithmics*, vol. 2.4, 2007.

[47] R. Bauer, D. Delling, and D. Wagner, "Experimental study on speed-up techniques for timetable information systems," *Networks*, no. 0.1002/net.20382, 2009.

[48] D. Delling, T. Pajor, and D. Wagner, "Engineering time- expanded graphs for faster timetable information." *Lecture Notes in Computer Science*, vol. 5868, pp. 182–206, 2009.

[49] G. Desaulniers and D. Villeaneuve, "The shortest path problem with time windows and linear waiting costs," *Transportation Science*, vol. 34, pp. 312–319, 2000.

[50] R. Dial, "Transit pathfinder algorithms," *Highway Research Record*, vol. 205, pp. 67–85, 1967.

[51] J. Granat and F. Guerriero, "The interactive analysis of the multicriteria shortest path problem by the reference point method," *European Journal of Operational Research*, vol. 151 (1), pp. 103–118, 2003.

[52] T.-N. Chuang and J.-Y. Kung, "The fuzzy shortest path length and the corresponding shortest path in a network." *Computers and Operations Research*, vol. 32(6), pp. 1409–1428, 2005.

[53] N. Van der Zijpp and C. Fiorenzo, "Path enumeration by finding the constrained k-shortest paths," *Transportation Research Part B*, vol. 39 (6), pp. 545–563, 2005.

[54] W. Xu, S. He, R. Song, and S. S. Chaudhry, "Finding the k shortest paths in a schedule-based transit network," *Computers & Operations Research*, 2010.

[55] Y.-L. Chen and H.-H. Yang, "Finding the first k shortest paths in a time-window network." *Computers and Operations Research*, vol. 31 (4), pp. 499–513, 2004.

[56] L. Zhao, Y. Xiong, and H. Sun, *The K Shortest Transit Paths Choosing Algorithm in Stochastic Transit Network*, R. Sets and K. Technology, Eds. Springer, 2008.

[57] M. Tan, C. Tong, S. Wong, and J. min Xu, "An algorithm for finding reasonable paths in transit networks," *Journal of Advanced Transportation*, vol. 41, No. 3, pp. pp. 285–305, 2010.

[58] R. W. Hall, "The fastest path through a network with random time-dependent travel times," *Transportation Science*, vol. 20(3), pp. 182–188, 1986.

[59] D. E. Kaufman and R. L. Smith, "Fastest paths in timedependent networks for intelligent vehicle-highway systems applications," *IVHS Journal*, vol. 1(1), pp. 1–11, 1993.

[60] M. P. Wellman, K. Larson, M. Ford, and P. R. Wurman, "Path planning under time-dependent uncertainty," in *Eleventh Conference on Uncertainty in Artificial Intelligence*, 1995, pp. 532–539.

[61] O. E. Karasan, M. C. Pinar, and H. Yaman, "The robust shortest path problem with interval data," manuscript, August 2001.

[62] P. Zielinski, "The computational complexity of the relative robust shortest path problem with interval data," *European Journal of Operational Research*, vol. 158, pp. 570–576, 2004.

[63] T. Feder, R. Motwani, L. O'Callaghan, C. Olston, and R. Panigrahy, "Computing shortest paths with uncertainty," *Journal of Algorithms*, vol. 62, pp. 1–18, 2007.

[64] S. Pallottino and M. Scutell, "A new algorithm for reoptimizing shortest paths when the arc costs change," *Operations Research Letters*, vol. 31 (2), pp. 149–160, 2003.

[65] H. Bast, S. Funke, P. Sanders, and D. Schultes, "Fast routing in road networks with transit nodes," *Science*, vol. 316(5824):566, 2007.

[66] D. Schultes, "Route planning in road networks," Ph.D. dissertation, Universitat Karlsruhe (TH), 2008.

[67] J. Jariyasunant and E. Mai, "Baytripper," *www.baytripper.org*. [Online]. Available: www.baytripper.org

[68] J. Jariyasunant, D. B. Work, B. Kerkez, R. Sengupta, S. Glaser, and A. Bayen, "Mobile transit trip planning with real-time data," in *Proceedings of the 2010 Transportation Research Board Annual Meeting*, 2010.

[69] B. Ferris, K. Watkins, and A. Borning, "Onebusaway: Results from providing real-time arrival information for public transit." in *Proceedings of CHI*, 2010.

[70] Z. Guo and N. H. M. Wilson, "Assessment of the transfer penalty for transit trips geographic information system-based disaggregate modeling approach," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1872, pp. 10–18, 2004.

[71] GTFS-DataExchange, *www.gtfs-data-exchange.com*.

[72] NextBus, *www.nextbus.com*.

[73] TriMet, *developer.trimet.org*.

[74] E. D., H. Timmermans, and L. V. Veghel, "Effects of data collection methods in travel and activity research," Prepared for European Institude of Retailing and Services Studies, Tech. Rep., 1996.

[75] P. R. Stopher, K. Kockelman, S. P. Greaves, and E. Clifford, "Reducing burden and sample sizes in multiday household travel surveys," *Transportation Research Record*, vol. 2064, pp. 12–18, 2008.

[76] J. Wolf, R. Guensler, and W. Bachman, "Elimination of the travel diary: An experiment to derive trip purpose from gps travel data." in *Transportation Research Record No. 1768, Transportation Research Board*, Washington, DC, USA, 2001, pp. 125–134.

[77] P. R. Stopher, N. Swann, and C. FitzGerald, "Using an odometer and a gps panel to evaluate travel behaviour change programs," in *11th TRB National Planning Applications Conference*, Daytona Beach, FL, USA, May 2007.

[78] L. M. T. M. . A. C. Wolf, J., *Transport survey quality and innovation.* Oxford: Pergamon Press, 2003, ch. Trip rate analysis in GPS-enhanced personal travel surveys., pp. 483–498.

[79] J. Wolf, *Travel survey methods: Quality and future directions.* Oxford: Elsevier, 2006, ch. Applications of new technologies in travel surveys, pp. 531–544.

[80] S. Bricka, "Non-response challenges in gps-based surveys," in *8th International Conference on Travel Survey Methods*, Annecy, France, May 2008.

[81] D. N. P. P. i. t. U. Nielsen, Smartphones Account for Half of all Mobile Phones, "http://www.nielsen.com/us/en.html," Tech. Rep., 2012.

[82] Y. Chon, E. Talipov, H. Shin, and H. Cha, "Mobility prediction-based smartphone energy optimization for everyday location monitoring," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '11. New York, NY, USA: ACM, 2011, pp. 82–95. [Online]. Available: http://doi.acm.org/10.1145/2070942.2070952

[83] D. Ferreira, A. Dey, and V. Kostakos, "Understanding human-smartphone concerns: A study of battery life," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, K. Lyons, J. Hightower, and E. Huang, Eds. Springer Berlin / Heidelberg, 2011, vol. 6696, pp. 19–33.

[84] J. Wolf, "Applications of new technologies in travel surveys," in *7th International Conference on Travel Survey Methods, Costa Rica*, 2004.

[85] P. R. Stopher and S. P. Greaves, "Household travel surveys: Where are we going?" *Transportation Research Part A: Policy and Practice*, vol. 41, no. 5, pp. 367–381, 2007, bridging Research and Practice: A Synthesis of Best Practices in Travel Demand Modeling. [Online]. Available: http://www.sciencedirect.com/science/article/B6VG7-4M6SBB1-2/2/65f88300cd48e4189ca4bd661243ba5d

[86] D. P. Wagner, E. Murakami, and M. Guindon, "Using gps for measuring household travel in private vehicles," in *6th TRB Conference on the Application of Transportation Planning Methods*, Dearborn, Michigan, USA, 1997.

[87] L. a. t. d. c. t. g. p. s. f. p. t. s. F. R. t. F. F. H. A. Battelle Transportation Division, "http://fhwa.dot.gov/ohim/trb/reports.htm," Tech. Rep., 1997.

[88] E. Murakami and D. P. Wagner, "Can using global positioning system (gps) inprove trip reporting?" *Transportation Research C*, pp. 7, 149–165, 1999.

[89] P. Stopher, C. FitzGerald, and M. Xu, "Assessing the accuracy of the sydney household travel survey with gps," *Transportation*, vol. 34, pp. 723–741, 2007, 10.1007/s11116-007-9126-8. [Online]. Available: http://dx.doi.org/10.1007/s11116-007-9126-8

[90] J. Zmud and J. Wolf, "Identifying the correlates of trip misreporting–results from the california statewide household travel survey gps study," in *10th IATBR*, Lucerne, 2003.

[91] G. Draijer, N. Kalfs, and J. Perdok, "Global positioning system as data collection method for travel research," *Transportation Research Record*, vol. 1719, pp. 147–153, 2000.

[92] J. Wolf, "Using gps data loggers to replace travel diaries in the collection of travel data," Ph.D. dissertation, Georgia Institute of Technology, School of Civil and Environmental Engineering, Atlanta, Georgia, USA, 2000.

[93] R. de Jong and W. Mensonides, "Wearable gps device as a data collection method for travel research, its working paper 03-02," Institute of Transport Studies, Sydney, Tech. Rep., 2003.

[94] J. Wolf, S. SchÃ¶nfelder, U. Samaga, and K. Axhausen, "80 weeks of gps-traces: Approaches to enriching trip information," in *83rd Annual Meeting of the Transportation Research Board*, Washington, D.C., USA, 2004.

[95] E. Chung and A. Shalaby, "Development of a trip reconstruction tool to identify traveled links and used modes for gps-based personal travel surveys," in *83rd Annual Meeting of the Transportation Research Board*, Washington, D.C., USA, January 2004.

[96] E. Chung, T. Ye, and A. Shalaby, "An integrated gps-gis system for personal travel surveys," in *GEOIDE Annual Conference*, Quebec, May 2004.

[97] S. Greaves, "A panel approach to evaluating voluntary travel behavior change programs–south australia pilot survey," in *Transportation Research Board 85th Annual Meeting*, 2006.

[98] S. T. Doherty, N. Noel, M.-L. Gosselin, C. Sirois, and M. Ueno, "Moving beyond observed outcomes: integrating global positioning systems and interactive computer-based travel behavior surveys." *Transportation Research Circular*, pp. 449–466, 2001.

[99] E. Hato, T. Mitani, and S. Itsubo, "Development of moals (mobile activity loggers supported by gps-phones) for travel behavior analysis," in *Transportation Research Board 85th Annual Meeting*, 2006.

[100] C. Williams, J. Auld, A. K. Mohammadian, and P. C. Nelson, "An automated gps-based prompted recall survey with learning algorithms," *Transportation Letters: The International Journal of Transportation Research*, vol. 1, pp. 59–79, 2009.

[101] P. R. Stopher, P. Bullock, and F. Horst, "Exploring the use of passive gps devices to measure travel," in *Proceedings of the 7th International Conference on Applications of Advanced Technologies to Transportation*, S. N. K. C. P. Wang, S. Medanat and G. Spring, Eds. Reston, VA: ASCE, 2002, pp. 959–967.

[102] C. A. Stopher, P. and P. Bullock, "Gps surveys and the internet," in *27th Australasian Transport Research Forum*, Adelaide, September 2004.

[103] M. E. H. Lee-Gosselin, S. T. Doherty, and D. Papinski, "Internet-based prompted recall diary with automated gps activity-trip detection: System design," in *TRB 85th Annual Meeting Compendium of Papers CD-ROM, Transportation Research Board*, Washington, D.C., USA, 2006.

[104] S. Y. A. Tsui and A. S. Shalaby, "Enhanced system for link and mode identification for personal travel surveys based on global positioning systems. transportation research record, 1972," in *Transportation Research Board*, Washington, D. C., USA, 2006, pp. 38–45.

[105] P. Bachu, R. Dudala, and S. Kothuri, "Prompted recall in global positioning survey: Proof of concept study," *Transportation Research Record*, vol. 1768, pp. 106–113, 2001.

[106] S. Krygsman, J. Nel, and T. de Jong, "The use of cellphone technology in activity and travel data collection in developing countries." in *18th International Conference on Transport Survey Methods*, Annecy, France, 2008.

[107] P. R. Stopher, *Transport survey methods: Keeping up with a changing world*. Bingley, UK: Emerald, 2009, ch. Collecting and processing data from mobile technologies.

[108] Y. Asakura, E. Hato, Y. Nishibe, T. Daito, J. Tanabe, and H. Koshima, "Monitoring travel behaviour using phs based location positioning service system," in *6th ITS World Congress*, Toronto, Canada, 1999.

[109] Y. Asakura and E. Hato, "Tracking survey for individual travel behaviour using mobile communication instruments," *Transportation Research C*, vol. 12(3/4), pp. 273–291, 2004.

[110] K. Sugino, S. Yano, E. Hato, and Y. Asakura, "Empirical analysis of sightseeing behaviour using probe person survey data," *Proceedings of Infrastructure Planning*, vol. 32, 2005.

[111] S. Itsubo and E. Hato, "Effectiveness of household travel survey using gps-equipped cell phones and web diary: comparative study with paper-based travel survey," in *Transportation Research Board 85th Annual Meeting*, 2006.

[112] H. Yatsumoto, T. Kitazawa, S. Nakagawa, A. Okamoto, and Y. Asakura, "Analysis of route choice behavior under flexible toll system of urban expressway based on probe person trip survey," in *33rd Meeting of Infrastructure Planning*, 2006.

[113] T. Mitani, "Study on the applicability of travel information provision system supported by probe person survey," Ph.D. dissertation, Ehime University, 2005.

[114] S. Reddy, M. Mun, J. A. Burke, D. Estrin, M. Hansen, and M. B. Srinivastava, "Using mobile phones to determine transportation modes," 2008.

[115] L. Liao, D. J. Patterson, D. Fox, and H. Kautz, "Learning and inferring transportation routines," *Artificial Intelligence*, vol. 171, no. 5-6, pp. 311 – 331, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/B6TYF-4N49VP9-1/2/8f05b8caf7327ceb8762ab5e1b95efc9

[116] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma, "Understanding transportation modes based on gps data for web applications," *ACM Trans. Web*, vol. 4, pp. 1:1–1:36, January 2010. [Online]. Available: http://doi.acm.org/10.1145/1658373.1658374

[117] P. Gonzalez., J. Weinstein, S. Barbeau, M. Labrador, P. Winters, N. Georggi, and R. Perez, "Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks," *IET Intelligent Transport Systems*, vol. Vol. 4, Iss. 1, pp. 37–49, 2010.

[118] X. Y., D. Low, T. Bandara, P. Pathak, H. B. Lim, D. Goyal, J. Santos, C. Cottrill, F. Pereira, C. Zegras, and M. Ben-Akiva, "Transportation activity analysis using smartphones," Massachusetts Institute of Technology, Cambridge, MA, USA, Tech. Rep., 2011.

[119] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, low-energy trajectory mapping for mobile devices," in *Proceedings of the 8th USENIX conference on Networked systems design and implementation*, ser. NSDI'11. Berkeley, CA, USA: USENIX Association, 2011, pp. 20–20. [Online]. Available: http://dl.acm.org/citation.cfm?id=1972457.1972485

[120] P. Stopher and N. Speisser, "Evaluation of gps device properties for a possible use in future household travel surveys," The University of Sydney, Sydney, Australia, Tech. Rep. Working Paper ITLS-WP-11-08, April 2011.

[121] K. N. Truong, J. A. Kientz, T. Sohn, A. Rosenzweig, A. Fonville, and T. Smith, "The design and evaluation of a task-centered battery interface," in *Proceedings of the 12th ACM international conference on Ubiquitous computing*, ser. Ubicomp '10. New York, NY, USA: ACM, 2010, pp. 341–350. [Online]. Available: http://doi.acm.org/10.1145/1864349.1864400

[122] A. Rahmati, A. Qian, and L. Zhong, "Understanding human-battery interaction on mobile phones," in *Proceedings of the 9th international conference on Human computer interaction with mobile devices and services*, ser. MobileHCI '07. New York, NY, USA: ACM, 2007, pp. 265–272. [Online]. Available: http://doi.acm.org/10.1145/1377999.1378017

[123] E. Oliver, "The challenges in large-scale smartphone user studies," in *Proceedings of the 2nd ACM International Workshop on Hot Topics in Planet-scale Measurement*, ser. HotPlanet '10. New York, NY, USA: ACM, 2010, pp. 5:1–5:5. [Online]. Available: http://doi.acm.org/10.1145/1834616.1834623

[124] C. Song, Z. Qu, N. Blumm, and A.-L. Barabsi, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010. [Online]. Available: http://www.sciencemag.org/content/327/5968/1018.abstract

[125] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating next-cell predictors with extensive wi-fi mobility data," *IEEE Trans. Mobile Comput.*, vol. 5(12), pp. 1633–1649, December 2006.

[126] F. Alizadeh-Shabdiz and E. J. Morgan, "System and method for estimating positioning error within a wlan-based positioning system," USA Patent 2008/0 108 371 A1, 2008.

[127] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, "Place lab: Device positioning using radio beacons in the wild," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, H. Gellersen, R. Want, and A. Schmidt, Eds. Springer Berlin / Heidelberg, 2005, vol. 3468, pp. 301–306.

[128] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The jigsaw continuous sensing engine for mobile phone applications," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '10. New York, NY, USA: ACM, 2010, pp. 71–84. [Online]. Available: http://doi.acm.org/10.1145/1869983.1869992

[129] Z. Zhuang, K.-H. Kim, and J. P. Singh, "Improving energy efficiency of location sensing on smartphones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 315–330. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814464

[130] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive gps-based positioning for smartphones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 299–314. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814463

[131] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava, "Sensloc: sensing everyday places and paths using less energy," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '10. New York, NY, USA: ACM, 2010, pp. 43–56. [Online]. Available: http://doi.acm.org/10.1145/1869983.1869989

[132] Y. Ma, R. Hankins, and D. Racz, "iloc: a framework for incremental location-state acquisition and prediction based on mobile sensors," in *Proceeding of the 18th ACM conference on Information and knowledge management*, ser. CIKM '09. New York, NY, USA: ACM, 2009, pp. 1367–1376. [Online]. Available: http://doi.acm.org/10.1145/1645953.1646126

[133] I. Constandache, S. Gaonkar, M. Sayler, R. Choudhury, and L. Cox, "Enloc: Energy-efficient localization for mobile phones," in *INFOCOM 2009, IEEE*, 2009, pp. 2716 –2720.

[134] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, "Energy-accuracy trade-off for continuous mobile device location," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 285–298. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814462

[135] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999. [Online]. Available: http://doi.acm.org/10.1145/331499.331504

[136] "Google directions api, http://code.google.com/apis/maps/documentation/directions/," Tech. Rep., 2012.

[137] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001, 10.1023/A:1010933404324. [Online]. Available: http://dx.doi.org/10.1023/A:1010933404324

[138] GTFS-Data-Exchange, "http://www.gtfs-data-exchange.com/," Tech. Rep.

[139] J. Bentley and T. Ottmann, "Algorithms for reporting and counting geometric intersections," *IEEE Transactions on Computers*, vol. 28, pp. 643–647, 1979.

[140] S. Fujii and A. Taniguchi, "Reducing family car-use by providing travel advice or requesting behavioral plans: An experimental analysis of travel feedback programs," *Transportation Research Part D: Transport and Environment*, vol. 10, no. 5, pp. 385 – 393, 2005.

[141] "How much do californias low-income households spend on transportation?" *Public Policy Institute of California*, no. 91, July 2004.

[142] P. Haas, C. Makarewicz, A. Benedict, and S. Bernstein, "Estimating transportation costs by characteristics of neighborhood and household," in *Transportation Research Record 2077*, 2009, pp. 62–70.

[143] *How much do California's Low-Income Households spend on Transportation?*, 2004.

[144] *"Americans Spend Over 100 Hours Commuting every Year", Census and Statistics*, 2005.

[145] S. Consolvo, D. W. McDonald, T. T., M. Y. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, I. Smith, and J. A. Landay, "Activity sensing in the wild: a field trial of ubifit garden," in *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ser. CHI 08. New York, NY: ACM, 2008, pp. 1797 – 1806.

[146] K. Kappel and T. Grechenig, "show-me: Water consumption at a glance to promote water conservation in the shower," in *Proceedings of the 4th International Conference on Persuasive Technology*, ser. Persuasive 09. New York, NY, USA: ACM, 2009, pp. 26 1 – 26 6.

[147] S. Consolvo, P. Klasnja, D. W. McDonald, and J. A. Landay, "Goal-setting considerations for persuasive technologies that encourage physical activity," *Persuasive Technology Conference*, 2009.

[148] C. Obermair, W. Reitberger, A. Meschtscherjakov, M. Lankes, and M. Tscheligi, "perframes: Persuasive picture frames for proper posture," *Persuasive*, 2008.

[149] J. Lin, L. Mamykina, S. Lindtner, G. Delajoux, and H. Strub, "Fish n steps: Encouraging physical activity with an interactive computing game," *Proceedings of UbiComp*, pp. 261–78, 2006.

[150] P. Klasnja, S. Consolvo, D. W. McDonald, and J. A. Landay, "Using mobile personal sensing technologies to support health behavior change in everyday life: Lessons learned," *American Medical Informatics Association (AMIA) Symposium Proceedings*, 2009.

[151] D. Holstius, J. Kembel, A. Hurst, P. Wan, and J. Forlizzi, "Infotropism: living and robotic plants as interactive displays," *Proc. DIS 04*, pp. 215–221, 2004.

[152] J. Mankoff, D. Matthews, S. R. Fussell, and M. Johnson, "Leveraging social networks to motivate individuals to reduce their ecological footprints," *Proc. HICSS 07, 2007*, pp. 87–96.

[153] J. Froehlich, D. T., P. Klasnja, J. Mankoff, S. Consolvo, B. Harrison, and J. A. Landay, "Ubigreen: investigating a mobile tool for tracking and supporting green transportation habits," in *Proceedings of the 27th international conference on Human factors in computing systems*, ser. CHI 09. New York, NY: ACM, 2009, pp. 1043 – 1052.

[154] M. Fishbein and I. Ajzen, "Belief, attitude, intention, and behavior: An introduction to theory and research," *Reading, MA: Addison-Wesley*, 1975.

[155] A. Bandura, "Self-efficacy: Toward a unifying theory of behavioral change," *Psychological Review*, vol. 84(2), 1977.

[156] I. Ajzen, "The theory of planned behavior," *Organizational Behavior and Human. Decision Processes*, vol. 50, pp. 179–211, 1991.

[157] P. Stern, "Toward a coherent theory of environmentally significant behavior," *Journal of Social Issues*, vol. 56, pp. 407–424, 2000.

[158] C. Carver and M. Scheier, *On the Self-Regulation of Behavior*, Cambridge, Ed. Cambridge University Press, 1998.

[159] P. Gollwitzer, "Goal achievement: the role of intentions," *European Review of Social Psychology*, vol. 4, pp. 141–185, 1993.

[160] J. Prochaska and C. DiClemente, "Toward a comprehensive model of change," 1986.

[161] Table of calories burned per hours. Wisconsin Department of Health Services. Last retrieved 08/01/2011. [Online]. Available: http://www.dhs.wisconsin.gov/health/physicalactivity/ToolCalcs.htm

[162] M. Chester, "Life-cycle environmental inventory of passenger transportation in the united states," PhD Thesis, University of California, Berkeley, 2008.

[163] Your drivig costs. American Automobile Association. Last retrieved 08/01/2011. [Online]. Available: http://www.aaaexchange.com/

[164] T. Schwanen and P. L. Mokhtarian, "Attitudes toward travel and land use and choice of residential neighborhood type: Evidence from the san francisco bay area," *Housing Policy Debate*, 2007.

[165] e. a. Geller, E.S., "A conceptual framework for developing and evaluating behavior change interventions for injury control," *Health Education Research*, vol. 5, pp. 125–137, 1990.