APPLICATION OF SMARTPHONE FOR INTERSECTION PERFORMANCE MEASUREMENT

A Thesis
Presented to
The Graduate Faculty of The University of Akron

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

Baoji Wang
December, 2011
APPLICATION OF SMARTPHONE FOR INTERSECTION PERFORMANCE MEASUREMENT

Baoji Wang

Thesis

Approved: _________________________________

Advisor
Dr. Ping Yi

Accepteed: _________________________________

Department Chair
Dr. Wieslaw K. Binienda

Committee Member
Dr. Anil Patnaik

Dean of the College
Dr. George K. Haritos

Committee Member
Dr. William H. Schneider IV

Dean of the Graduate School
Dr. George R. Newkome

Date

ii
ABSTRACT

Traffic engineers and system analysts rely on timely and accurate data in decision making that impacts the safety and efficiency of the transportation system. However, since data collection for such a purpose has never been an easy task at intersections, many people often relies on simulation to evaluate engineering plans. This research presents the development and testing of an innovative method to collection turning movement and vehicle delay information data at an intersection. It uses the smartphone technology for movement identification and object tracking. The algorithms are explained in detail and preliminary tests have been conducted. The results of this study demonstrate the feasibility of the proposed method and its future promise.
ACKNOWLEDGEMENTS

First of all I would like to extend my thanks to my supervisor Dr. Ping Yi. He has been very helpful and has guided me through the project’s twist and turns.

Acknowledgements are also extended to my committee members, Dr. William H. Schneider, Dr. Anil Patnaik for reviewing my work and helpful recommendations.

I would also like to thank the commit member’s advice. I would also like to thank Dr. Chun Shao for his helpful ground work and suggestions from improvement.

Finally I would like to thank everyone in the Transportation Lab of University of Akron for their support and for making my time at UA enjoyable.
TABLE OF CONTENTS

LIST OF TABLES ................................................................. vii
LIST OF FIGURES ........................................................................................................ viii

I. INTRODUCTION ....................................................................................................... 1
   1.1 BACKGROUND ............................................................................................... 2
   1.2 THESIS OUTLINE ....................................................................................... 5

II. LITERATURE REVIEW ............................................................................................. 6
   2.1 TURNING MOVEMENT AT SINGLE INTERSECTION .................................. 7
   2.2 DELAY MEASUREMENT ............................................................................ 8
   2.3 TRAFFIC COUNT AND DEVICE ............................................................... 10
      2.3.1 Traffic Count ...................................................................................... 10
      2.3.2 Traffic Count Device .......................................................................... 11
   2.4 MULTI-TOUCH TECHNOLOGIES .............................................................. 13
   2.5 DEVELOPMENT OPERATION SYSTEM .................................................. 14

III. METHODOLOGY ..................................................................................................... 16
**LIST OF TABLES**

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1. Traffic Count Form</td>
<td>11</td>
</tr>
<tr>
<td>2.2. Existing Mobile Development Operation System</td>
<td>15</td>
</tr>
<tr>
<td>3.1 Classification of Turning Movements by Its Approaches</td>
<td>18</td>
</tr>
<tr>
<td>3.2 The XML Data</td>
<td>19</td>
</tr>
<tr>
<td>6.1 Field test result of the application</td>
<td>35</td>
</tr>
<tr>
<td>6.2 Intersection VTM identification Tests</td>
<td>38</td>
</tr>
<tr>
<td>6.3 Vehicle Delay Measurement Tests</td>
<td>39</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1. Different Types of Vehicle Delay</td>
<td>10</td>
</tr>
<tr>
<td>2.2. TDC Ultra Hand-held Traffic Data Collector</td>
<td>13</td>
</tr>
<tr>
<td>2.3 MULTI-TOUCH Technology In IPHONE</td>
<td>14</td>
</tr>
<tr>
<td>3.1 Flow-chart for Turning Movement Identification</td>
<td>18</td>
</tr>
<tr>
<td>3.2 Directions of Finger Movements</td>
<td>21</td>
</tr>
<tr>
<td>3.3 Calculation of Angle for Direction</td>
<td>22</td>
</tr>
<tr>
<td>3.4 Example of Gesture Recognition</td>
<td>24</td>
</tr>
<tr>
<td>4.1 Approach Travel Time Measurements</td>
<td>27</td>
</tr>
<tr>
<td>4.2 Example of Travel Time Estimation through Average Time Stamp</td>
<td>29</td>
</tr>
<tr>
<td>5.1 Recognizing Vehicle Turning Movements from Gestures</td>
<td>31</td>
</tr>
<tr>
<td>5.2 Vehicle Turning Movement Modules</td>
<td>32</td>
</tr>
<tr>
<td>5.3 Intersection Delay Measurement Module</td>
<td>34</td>
</tr>
</tbody>
</table>
CHAPTER I
INTRODUCTION

Conventional methods of traffic data collection have always been labor-intensive and, time-consuming, and often tedious. Manually transcribing traffic data and entering into a spreadsheet is not only ineffective but also error-prone. Automatic traffic data collection by detectors and other traffic sensors requires installation of expensive hardware equipment costly to maintain, yet some critical information, such as vehicle turning movements and intersection delay, is hard to obtain due to a lack of vehicle tracking capability of by the detectors/sensors and the limitations of data processing algorithms. Some companies have contributed to improving the experience of manual data collection by developing hand-held devices that are easy to use; however, the ability of such devices to collect different types of traffic and system performance data is very limited.

Recently, the development of touch screen technology, especially multi-touch technology after it was introduced, has led to many innovative applications in computer operation and cell phone use. Since the introduction of iPhone in 2007, multi-touch technology has been gradually accepted by the public and is becoming the industry standard portable devices.
This technology can be applied in traffic data collection with versatile functions and at no additional cost. The objective of this paper is to conduct a feasibility study that develops and demonstrates an innovative method of collecting and analyzing intersection data using a multi-touch enabled smartphone.

1.1 BACKGROUND

Turning Movements Information for vehicles at signalized intersection is important for many traffic operations including but not limited to adaptive signal control, dynamic traffic assignment and traffic demand estimation. The traffic information of road network is very useful for navigation system or intelligent transportation system. Existing works deploy sensors along roads to collect traffic information. Due to the expensive cost of each sensor for monitoring traffic information, only freeway/highway roads are deployed sensors. Thus, a limited amount of traffic information is available for users.

Traditionally, on the local roadway, traffic data collection has always been a tedious, labor-intensive and time-consuming work. Manually transcribing traffic data and typing into a spreadsheet later is not only an ineffective but also an error-prone method. Automatic traffic data collection by detectors and sensors requires installation of expensive facilities and costly maintain service, and some critical data, such as vehicle turning movements, is hard to obtain due to the accuracy of detectors and limitations of algorithms. Some companies have contributed to improving the experience of manual data collection and developed hand-held device to assist traffic engineers. However, the complicated operation panel and small display screen increase
the difficulty to manipulate this kind of devices and their functions have been limited by the built-in data analysis algorithm and the memory size. In addition, current collection methods are focused on traffic data at single intersection instead of the whole traffic network. To solve the above problems, multi-touch technology has revealed an innovative approach to collect and analyze traffic data. Since the introduction of iPhone from Apple Inc. in 2007, multi-touch technology has been accepted by the public and is becoming the standard for not only smartphones but also other portable devices. This technology can be applied in traffic data collection with less cost, labor and error, and in the meantime, better experience.

Most of current data collection methods are focused on single intersection and the interested data includes vehicle turning movements and vehicle delay measurement. We will discuss the limitations in the existing data collection methods in the following part.

The most widely used method to measure approach delay is proposed in Highway Capacity Manual 2000 (5). In this method, the number of vehicles in queue will be counted at the end of each time interval during the study, and the final result needs to be adjusted with several factors to address the impact from vehicle acceleration/deceleration and overestimation from engineers. Besides the error from counting the vehicles, the selection of the factors is also debatable. In addition, this method still needs lots of labors which in generally equals to the number of lanes.

Many researchers have introduced other methods to measure vehicle delay or travel time on approaches. However, many models are not applicable in the field because of many assumptions have to be fulfilled. For instance, input-output model requires the assumption of FIFO, no lane changing and 100% accuracy of detection. Models using photography technology are very time consuming and costly. For vehicle identification technique, its costly equipments have prevented it to be used widely.
Although hand-held devices for traffic data collection can help traffic engineers to reduce the burden on data collection, it still has some limitations as follows.

1. **Control panel with too many buttons**

   No matter TDC Ultra from JAMAR Technologies or HI-TRAC from Road and traffic technology (6), they are built with many buttons on it to accomplish different jobs. The more buttons usually represents more functions this device will support. However, more buttons on the control panel will increase the difficulty to manipulate this device. Operator has to be very careful to avoid hit the wrong button during the data collection, and this will affect the accuracy and efficiency of data collection.

2. **Small display screen and not friendly user interface**

   Most of the hand-held devices on the market are using monochrome LCD screen without touch function. User interface, such as navigating the menus, is not friendly compared to other portable devices such as cell phones or media players that increase the difficulty for engineer to learn and use.

3. **Limited memory size**

   Some models of hand-held devices have very limited memory size, counted in KB, to save traffic data. This will influence the resolution of collected data and limit their applications. Unfortunately, the memory is usually built inside the device and cannot be replaced or upgraded easily.

4. **Bigger and heavier compared to smartphones or other portable devices**

   Though they are hand-held devices, according to today’s technologies, it is outdated by its “giant” size and weight compared with other portable devices which can be put in the pocket.

5. **Limited voice assistance**
Most of the hand-held devices can only make simple beep sound and this can barely help traffic data collection. However, voice assistance is very important in the field to help data collector identify the error and correct it at the first time. For example, in turning movement study, the name of turning movement can be pronounced right after the input, so the operator can confirm the input while keeping the eye on the roadway.

1.2 THESIS OUTLINE

A literature review, which examines the histories and limitations of the traditional traffic data collection equipment in recording vehicle turning movements and the vehicle delay estimation. Chapter III illustrates the methodology of mobile device for intersection performance measurement and discusses the detail of its algorithm. The results of laboratory and field experiments along with the analysis are presented in Chapter IV. After that, the application of in vehicle delay estimation is demonstrated in Chapter V.
CHAPTER II
LITERATURE REVIEW

Traffic engineers and system analysts rely on timely and accurate data in decision-making that impacts the safety and efficiency of the transportation system. However, data collection for such a purpose has never been an easy job. Traditionally, engineers had to record traffic data in the field with pen and paper, and then manually enter the data into the computer system for analysis. Furthermore, in many cases the time stamp information of the data is desired, thus the data collectors’ engineers have to do time synchronization, watch the roadway and monitor the clock at the same time. This process is labor intensive and error-prone. For example, today, to collect vehicle turning movement information at a typical four-leg intersection, many state DOTs and metropolitan planning organizations (MPOs) today are still using people to account the turning vehicles in the field, and at least two persons must be assigned to each busy intersection in order to reduce human errors due to fatigue and distraction.

Many efforts have contributed to improving the efficiency and accuracy in data collection. Benekohal et al (1) proposed to use the input-output model to measure vehicle delay with an image processing technique; other studies on data collection include those using vehicle identification techniques (2), ground-based time-lapse photography (3), and aerial time-lapse photography (4), etc. However, many of those efforts still tend to be time consuming and costly.
Hand-held devices have been introduced to traffic data collection since 1970s. They are designed to help reduce the manual work by recording traffic data and saving them automatically in electronic format. For example, TDC Ultra Hand-held Traffic Data Collector from JAMAR Technologies Inc. is one of the most widely used traffic data collectors device in the world today. Using the push buttons to account traffic, the saved data can be downloaded to a personal computer with proper communication interface and application software. Other hand-held devices are also available on the market, such as Tracker (5) from International Road Dynamics Inc., all built in with similar functions.

2.1 TURNING MOVEMENT AT SINGLE INTERSECTION

Vehicle turning movement is one of the most important information in traffic operation. Even with the help of hand-held device, it is hard for one engineer to record all turning movements for a single intersection at one time. To prove the feasibility of collecting intersection turning movements by one engineer, a program will be developed for smartphone or other portable device with multi-touch function. The finger movements will be tracked in the program and interpreted into vehicle turning movements. The data will be automatically saved in the memory with time stamp information, and it can also be exported easily to personal computers for further analysis.
2.2 DELAY MEASUREMENT

Delay time study of traffic network is usually broken down into links and nodes. With network decomposition, we can apply specific model to estimate delay at link and node separately. In addition, it is more reliable and robust since error in collected data will only affect certain link/node rather than the whole network. As shown in Figure 2.1, vehicle delay includes approach delay at the link and intersection delay at the node, which in total called control delay. Intersection delay is discussed in a separate paper submitted by the authors along with vehicle turning movement identification. In the scope of this paper, we will focus on the approach delay estimation, which is also part of real-time traffic network performance evaluation system.

According to HCM 2000, control delay is defined as the additional travel time experienced by a vehicle affected by intersection control. As shown in Figure 2.1, control delay can be separated into different sections such as deceleration delay, stop delay, acceleration delay, approach delay and intersection delay. Their definitions are listed as follows:

As shown in Figure 2.1, approach delay is defined as the difference between Free Flow Travel Time (FFTT) and measured travel time from input detector (upstream) to output detector (downstream). Vehicle’s trajectory is shown in solid green line with horizontal axis as the time and vertical axis as the position. From this figure, we can find that no matter how complex the vehicle moves between the input and output detector, the measured travel time is only determined by the time stamps of vehicle passing these two detectors. Since FFTT can be
calculated by the free flow speed and the distance between input and output detector, approach delay can be estimated from the measured travel time.

Vehicle travel time estimation has been studied for decades. Many methods and techniques have been introduced and implemented to track individual vehicle’s travel time or estimate average travel time in a short time interval. Although previous researches have achieved various degrees of success, there is no practical model for reliable real-time vehicle travel time measurement. Among these approaches, some methods tend to be labor intensive, time consuming and expensive, such as vehicle re-identification and photographic techniques. Some traditional method like probe vehicles is limited by its sample size and not suitable for real-time data collection. Regression model is another solution; however it relies on huge data surveillance and has limitation on locality. Other methods either need extra devices or manual error check has to be involved, which makes them not suitable for real-time travel time estimation.

In this paper, we will introduce a method called Gap Recognition Approach to Delay Estimation (GRADE) to estimate average travel time at each link based on existed facilities or require minor changes. Reasonable and accurate estimation will be made from imperfect field data with complex geographical configurations. Preliminary experiments have shown GRADE is a practical and useful method with error percentage in 10% and at most 15%.
2.3 TRAFFIC COUNT AND DEVICE

2.3.1 Traffic Count

A traffic count is a count of vehicles or/and pedestrians along a particular road, either done electronically or by people counting by the side of the road. Traffic counts can be used by local councils to identify which routes are used most, and to either improve that road or provide an alternative if there is an excessive amount of traffic. Also, some geography fieldwork involves a traffic count. They are useful for comparing two or more roads, and also can be used alongside other methods to find out where the CBD of a settlement is located.
The following is an example of a traffic count, showing the type of vehicle and the data collected at a particular place in each direction. As you can see, this data has been compiled into numbers from each direction and a total count. The original table (which was used out on the road) used data recorded in tally form.

Table 2.1. Traffic Count Form

<table>
<thead>
<tr>
<th>Vehicle Type</th>
<th>Direction X</th>
<th>Direction Y</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car</td>
<td>48</td>
<td>47</td>
<td>95</td>
</tr>
<tr>
<td>Lorry</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Van</td>
<td>5</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>Bus</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Bike</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Pedestrian</td>
<td>3</td>
<td>4</td>
<td>7</td>
</tr>
</tbody>
</table>

2.3.2 Traffic Count Device

A traffic counter is a device, often electronic in nature, used to count, classify, and/or, measure the speed of vehicular traffic passing along a given roadway. The device is usually deployed in near proximity to the roadway and uses an intrusive medium, such as pneumatic road tubes laid across the roadway, piezo-electric sensors embedded in the roadway, inductive loops cut into the roadway, or a combination of these to detect the passing vehicles. Recently, in the interest of worker safety and ease of installation, non-intrusive technologies have been developed. These
devices generally use some sort of transmitted energy such as radar waves or infrared beams to detect vehicles passing over the roadway.

In order to improve the accuracy and efficiency of traffic data collection, many companies and research organizations have contributed to it and advancements in both algorithms and equipment have been made. Input-Output model was introduced by Benekohal et al (6) in 1993 to measure vehicle delay with an image processing technique. Other studies include those using vehicle identification technique, ground-based time-lapse photography, and aerial time-lapse photography. However, many of these procedures still tend to be labor intensive, time consuming and costly.

Hand-held devices have been introduced to traffic data collection since 1970s. They are designed to help reducing the labors involved in collection and save traffic data automatically. As shown in Figure 2.2, TDC Ultra Hand-held Traffic Data Collector form JAMAR Technologies Inc. is one of the most powerful traffic data collectors in the world today. It can help engineer to collect traffic data from turning movements to vehicle delay at intersections. Collected data will be automatically saved in the build-in memory of this unit and can be downloaded and open in a personal computer with proper software and communication interface.
2.4 MULTI-TOUCH TECHNOLOGIES

Multi-touch is an enhancement to the touch screen technology which allows user operate the device by applying multiple fingers or stylus onto the display screen simultaneously. The concept of multi-touch was first introduced in 1982 when researchers in the University of Toronto built a system with frosted-glass and a camera behind it. Three years later, a tablet was built with multi-touch technology in the same university based on capacitance instead of camera. Almost at the same time, Bob Boie built the first multi-touch screen, not tablet, in Bell Labs by using a transparent capacitive array of touch sensors overlaid on a CRT (7). Breakthrough occurred in 2007 when Microsoft released its Microsoft Surface (8) and Apple unveiled iPhone (9). The application of multi-touch technology in iPhone started a revolution in smartphones market. In just two years, iPhone grabbed 23% share of smartphones market in North America.
Despite being tied to a single carrier in the US (10). More and more cell phone manufacturers start building smartphones with multi-touch function such as HTC, Palm and Google. Besides smartphones, multi-touch technology is also used in other portable devices such as portable media player, for example Microsoft Zune HD and Apple iPod Touch. The increasing market of multi-touch devices has provided a platform to implement innovative traffic data collection system.

![FIGURE 2.3 MULTI-TOUCH TECHNOLOGY IN IPHONE](image)

2.5 DEVELOPMENT OPERATION SYSTEM

As seen in Table 2, there are several development operation systems for mobile development, in this project we selected the Android Operation System. Unlike most of the competitors, Android is built upon an open-source platform, and most of the Android code is released under the open source Apache License. Android applications are written in the Java programming language, which is a powerful, mature and extensively adopted language in the global development
community. Android’s Java is not the same as J2ME, however, most of the existing J2ME applications can be ported to Android with small modifications.

The Android market is Google's open source initiative to provide applications for Android based smart phones. Apple strives to ensure applications meet its defined standards of quality, utility and appropriateness whereas Google follows a very simple procedure. Google only sees to that the applications meet the terms the developers agree to when they sign up for the Android Market (11). Applications are not evaluated on any other grounds, so developers will have more freedom to create the type of application or content they like (12). Android market lets developers directly list their applications, which speeds up the launch of the apps. Sales of Google Android phones in the United States are rising so rapidly, that the devices have outsold Apple handsets for the first time on record (13).

Table 2.2 Existing Mobile Development Systems

<table>
<thead>
<tr>
<th>System</th>
<th>Company</th>
<th>Development Language</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>Google</td>
<td>Java</td>
<td>Android mobile operating system is built on top of Linux kernel version 2.6. The Linux kernel provides the interfaces to access low level hardware control functionalities.</td>
</tr>
<tr>
<td>IOS</td>
<td>Apple</td>
<td>Object C</td>
<td>The IOS is a derivative of the Darwin open source POSIX compliant computer operating system developed by Apple Inc.</td>
</tr>
<tr>
<td>Platform</td>
<td>Company</td>
<td>Development Kit</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------</td>
<td>-----------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Symbian OS</td>
<td>Nokia</td>
<td>NA</td>
<td>The user application level does not permit the usage of multi-threading and context switching</td>
</tr>
<tr>
<td>Windows Mobile</td>
<td>Microsoft</td>
<td>Microsoft .NET</td>
<td>This platform is based on Windows CE (WinCE). WinCE is a compact OS specifically designed, for devices that have minimal storage. Windows CE is a distinct operating system and kernel rather than a trimmed down version of desktop Windows.</td>
</tr>
<tr>
<td>Blackberry</td>
<td>Canadian company Research in Motion (RIM)</td>
<td>JAVA</td>
<td>BlackBerry provides associate Java software development kit in order to encourage programmers to develop Blackberry Applications</td>
</tr>
</tbody>
</table>
CHAPTER III
METHODOLOGY

3.1 TURNING MOVEMENT AT SINGLE INTERSECTION

Vehicle turning movement (VTM) information is vitally important for efficient intersection operation. To demonstrate the feasibility of using smartphone to collect traffic data, an algorithm for vehicle turning movement identification has been developed in this research on an open source smartphone platform, named Android. Android was introduced by Google in 2008 and supported by abundant third-party applications due to its open source feature. Our algorithm is implemented on the Android platform using Java as the programming language. The algorithm is based on pattern recognition over input gestures to identify VTM.

In a typical four-leg intersection, there are in total twelve possible turning movements assuming U-turn is not allowed. A good classification method for VTM identification is by using directions, including the one the vehicle comes from and the one it is going to. As shown in Table 1, all the twelve turning movements can be easily categorized into different groups.

In order to identify turning movement for each gesture, we should find out the direction of the gesture at the beginning part and the ending part. After that, we need to check against Table 3.1 to obtain the corresponding turning movement as the output. This process is explained in a flowchart shown in Figure 3.1 and the implementation steps below.
Table 3.1 Classification of Turning Movements by Its Approaches

<table>
<thead>
<tr>
<th>Coming From</th>
<th>Going to</th>
<th>West Bound</th>
<th>North Bound</th>
<th>East Bound</th>
<th>South Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>East Bound</td>
<td>EL*</td>
<td>ET</td>
<td>ER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South Bound</td>
<td>SR</td>
<td>SL</td>
<td>ST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>West Bound</td>
<td>WT</td>
<td>WR</td>
<td>WL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>North Bound</td>
<td>NL</td>
<td>NT</td>
<td>NR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* The first letter represents the direction (E for East, W for West, S for South, and N for North) and the second letter represents the turning action (R for Right Turn, L for Left Turn and T for Through).

Figure 3.1 Flow-chart for Turning Movement Identification
STEP 1: GET ALL THE POINTS IN GESTURE

There is a coordinate of each point. The coordinate is a relative position of the screen of the smartphone.

The direction of gesture at the beginning, or the start part, will represent the direction of the coming approach, while the direction at the end, or the ending part, can indicate the direction for the leaving approach. In order to find the directions, all the points in the gesture should be obtained and sorted in the manner of time sequence recorded on the touch screen.

A serious of coordinates of the gesture point was recorded by the phone when the user starts to draw the turning movement on the smartphone (see below).

Table 3.2 The XML Data

<table>
<thead>
<tr>
<th>POINT X</th>
<th>Y</th>
<th>TIME</th>
<th>PRESSURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>214.77684</td>
<td>153.27478</td>
<td>64009218</td>
<td>0.11764706</td>
</tr>
<tr>
<td>215.65707</td>
<td>153.79657</td>
<td>64009234</td>
<td>0.13333334</td>
</tr>
<tr>
<td>217.04028</td>
<td>154.1879</td>
<td>64009242</td>
<td>0.14901961</td>
</tr>
<tr>
<td>218.42351</td>
<td>154.70969</td>
<td>64009256</td>
<td>0.16078432</td>
</tr>
<tr>
<td>220.05823</td>
<td>154.84015</td>
<td>64009266</td>
<td>0.16862746</td>
</tr>
<tr>
<td>222.19594</td>
<td>154.57925</td>
<td>64009276</td>
<td>0.1764706</td>
</tr>
<tr>
<td>224.4594</td>
<td>154.05746</td>
<td>64009287</td>
<td>0.18039216</td>
</tr>
<tr>
<td>227.47733</td>
<td>153.40521</td>
<td>64009299</td>
<td>0.18039216</td>
</tr>
<tr>
<td>230.99828</td>
<td>152.23123</td>
<td>64009312</td>
<td>0.18039216</td>
</tr>
</tbody>
</table>
There are several points of each turning movement line segment. The coordinate is a relative position of the screen of the smartphone. Every coordinate of the point is stored in the random of the smartphone.

**STEP 2: IDENTIFY DIRECTION AT THE BEGINNING**

In order to identify the direction of the beginning part of gesture, imagine in Figure 3.2 (a four-leg intersection) we set up a coordinate using the first point (start point) in the gesture as the origin, the angle between the origin and any point in the gesture can be calculated.

![Figure 3.2 Directions of Finger Movements](image)

For direction identification, we set up the boundaries for each direction represented by the dashed lines using east as the reference direction (angle zero). For north bound, the angle varies
from 45° to 135°; for west bound, it is from 135° to 225°; for south bound, it is from 225° to 315°, so for east bound it can be either from 315° to 360° or from 0° to 45°.

Since there are many points in each gesture, we should define proper criteria to select another point to calculate the direction. This point should not be too close to the start point, or it may be erroneous because of the shakes by the finger; on the other hand, the point should not be too far away from the start point as the gesture may have turned the direction. As shown in Figure 3.3, the point that is closest to the boundary (green circle) will be selected to calculate the angle $\theta$ with the following formula:

$$\theta = \tan^{-1}\left(\frac{y_2 - y_1}{x_2 - x_1}\right)$$  \hfill (1)

where $(x_1, y_1)$ represents the coordinates of start point and $(x_2, y_2)$ is the coordinates of selected point.

Figure 3.3 Calculation of Angle for Direction
The direction can then be found by comparing the angle $\theta$ and the boundaries for all four directions.

**STEP 3: IDENTIFY DIRECTION AT THE END**

The vehicle does not only move to NB, SB, EB and WB, but also make left and right turns on the road. Therefore it is not able to identify the direction of the movement if we only check the beginning part of the coordinates.

In order to check the turning movement for the vehicle we also need to check the end part of the coordinates.

The algorithm in this step shares similar procedure to that in Step 2. However, we should use the coordinates of end point as $(x_2, y_2)$ and the selected point as $(x_1, y_1)$ according to the time sequence of the points. Using Equation 1 listed above, we can find the $\theta$ and hence the direction at the ending part of the gesture.

**STEP 4: SEARCH TURNING MOVEMENT TABLE AND RETURN MATCHED RESULT**

As shown in Figure 3.4, when a gesture is received, all the points in it have been plotted as red dots. The directions of the beginning and the ending parts have been determined by the previous steps, which are east bound and south bound, respectively. Then, in Table 1 the algorithm automatically find the matched turning movement –east bound right turn, as the output.
Figure 3.4 Example of Gesture Recognition
CHAPTER IV
DELAY ESTIMATION

Vehicle delay is another important performance indicator at intersections. Vehicle delay can be calculated by actual travel time subtracts free flow travel time going through the intersection. The free flow travel time is determined by the travel length and the free flow speed. Once the intersection is selected and the reference point chosen, the free flow travel time is determined. The travel time can be separated into intersection travel time and approach travel time.

4.1 INTERSECTION TRAVEL TIME AND DELAY

Intersection travel time is defined as the time a vehicle spent inside the intersection. When we are collecting VTM data, the travel time inside the intersection can be obtained simultaneously. We require that the operator touches the screen when the vehicle enter the intersection, and holds the finger until the vehicle leaves the intersection, the duration of the gesture will be the actual travel time of the vehicle inside the intersection. Since the position of each point in gesture will be recorded along with its time stamp, the duration of gesture is the time difference between its start point and end point. The average delay for each turning movement can be obtained by:
\[ \overline{DT}_t = \frac{\sum_{i=1}^{N} itti_t}{N} - ifftt_t \]

where,

\( \overline{DT}_t \) is the average delay of turning movement t,

itti is the intersection travel time for vehicle i,

N is the number of vehicles in this turning movement,

ifftt is the intersection free flow travel time.

### 4.2 APPROACH TRAVEL TIME

Approach travel time is defined as the time a vehicle spent in an intersection approach between the reference point (located upstream) and the stop line. The reference point usually is selected to allow enough distance from the stop line to avoid blockage from queue spillback, as shown in Figure 4.1. Time stamp \( t_r \) is recorded when a vehicle passes the reference point. On the other hand, the system records the time stamp \( t_s \) when a vehicle leaves the stop line (as VTM is identified). The vehicle approach travel time \( att \) can be calculated as:

\[ att = t_s - t_r \] (3)
4.3 AVERAGE VEHICLE DELAY

Since vehicles change speed and lane positions, the First In First Out rule when we measure travel time will not apply. Therefore, we programmed our algorithm to calculate average vehicle delay instead of individual delay. In this way, we can eliminate the errors caused by incorrect pairing of the vehicles in time stamp measurement. Assume N vehicles have been recorded; the following formula is used in calculation:

$$\overline{att}_d = \frac{\sum_{i=1}^{N} t_{r,i} - \sum_{i=1}^{N} t_{s,i}}{N}$$  \hspace{1cm} (4)$$

where,

$att_d$ is the average approach travel time at direction $d$,

t$_{r,i}$ is the time stamp of vehicle $i$ passes the reference point,

t$_{s,i}$ is the time stamp of vehicle $i$ leaves the stop line.
As mentioned above, using Equation 4 does not have to require matching the right sequence of the time stamps. One vehicle can have different index numbers at the reference point and the stop line and the measured data will compensate themselves. In other words, the result will be the same as long as we include the same N vehicles in the calculation.

Thus, the average approach delay can be calculated by:

$$\overline{DA_d} = \overline{att_d} - \overline{afftt_d}$$

(5)

where,

$\overline{DA_d}$ is the average approach delay at direction d,

afftd is the approach free flow travel time at direction d.

Summarizing Equations 2 and 5, the average vehicle delay can be calculated by:

$$\overline{D_t} = \overline{DA_d} + \overline{DI_t}$$

(6)

where,

$\overline{D_t}$ is the average delay for turning movement t,

$\overline{DI_t}$ is the average intersection delay for turning movement t.

To further explain above, average time stamp is introduced to estimate average travel time. As shown in Figure 4, there are seven vehicles involved in the travel time study and the travel time of each individual vehicle $v$ can be calculated by

$$TT_v = O_v - I_v$$

(7)

Where
$TT_v$ is the approach travel time of vehicle $v$;

$O_v$ is the time stamp of vehicle $v$ at output detector, which is the off time of output detection;

$I_v$ is the time stamp of vehicle $v$ at input detector, which is the off time of input detection.

Figure 4.2 Example of Travel Time Estimation through Average Time Stamp

Suppose there are in total $N$ vehicles involved in the studied time interval, it will be $N$ input time stamps and $N$ output time stamps if no error exists. The average travel time $\overline{TT}$ can be calculated by:

$$\overline{TT} = \frac{\sum_{v=1}^{N} TT_v}{N} = \frac{\sum_{v=1}^{N} (O_v - I_v)}{N} = \frac{\sum_{v=1}^{N} O_v}{N} - \frac{\sum_{v=1}^{N} I_v}{N}$$

\[(8)\]
We name $\bar{O} = \frac{\sum_{i=1}^{N} O_i}{N}$ as the average time stamp for output and $\bar{I} = \frac{\sum_{i=1}^{N} I_i}{N}$ for input. The average travel time can also be written as

$$\bar{TT} = \bar{O} - \bar{I}$$ (9)

The advantage of estimating travel time from $\bar{O}$ and $\bar{I}$ is its independence to the order of detections (time stamps). As long as the input and output detections are collected from the same group of vehicles, the result will not be affected by lane changing or vehicle passing. On the other hand, if there is any error, the average process can also reduce the impact from the error data.
CHAPTER V
SYSTEM AND MODULE

The Nexus Android smartphone runs the Turning Movement collection application. It has a simple user interface with three activity screens. The first is the main activity and can be seen in figure 1. It is possible to press the “Start Turning Movement” for starting to record the turning movement, to press “LOAD” for review the history data and press “ANALYSIS” for delay analysis, and for exiting the application just press the button “QUIT”.

The Android platform comes with a menu button, so that users can press this button to save, set values of speed limit and record ground truth of the turning movement or quit the application.

Figure 5.1 Recognizing Vehicle Turning Movements from Gestures
5.1 VEHICLE TURNING MOVEMENTS MODULE

Vehicle Turning Movements Module is based on pattern recognition over input gestures to identify Vehicle Turning Movements.

Users can simply draw the direction by finger on the screen of the phone. The system will record the timestamp of the vehicle entering the intersection. The time when vehicle left of the vehicle will be recorded automatically when the user finishes drawing the movement and their finger left touching the screen.

By applying the methodology of identifying the turning movement, system will not only show the track of the turning movement but also show the direction of the movement and add the data, which shows on the top and bottom part of the screen.

During the data collection, traffic engineer can press the back button, which comes with the android platform to delete the redundant count or add the missed count.

![Figure 5.2 Vehicle Turning Movement Module](image_url)
5.2 HISTORY REVIEW MODULE

The turning movement data, date and time of the experiment can be saved in the memory or SD card. I choose the xml format to save it.

When press the “Load” button, it will display all the history data saved in the system. After choosing the data, the system will display the total number of vehicles for 12 different directions. And also users can check every movement by finger from the beginning to the end.

5.3 INTERSECTION DELAY MEASUREMENT MODULE

Intersection Delay is calculated by actual travel time minus free flow travel time. In order to obtain actual travel time, the system records the time when a vehicle passes a preset reference point and tracks when the vehicle leaves the intersection. Free flow travel time can be calculated by inputting travel length and free flow speed.

By clicking the “ANALYSIS” button. An info box will ask you to set the free flow speed and street length. The calculated result will display as the figure showed above.

The first column of the data is the travel time for four directions and second one is the delay.
Figure 5.3 Intersection Delay Measurement Module

NB: 16.189 Delay: 16.078
SB: 17.323 Delay: 17.212
EB: 11.386 Delay: 11.275
WB: 14.621 Delay: 14.510
CHAPTER VI
TESTING AND VALIDATION

To test the efficiency and error of this android application, we did the field test in the South Alington St and Fifth Ave Akron OH for 2 hours. The result shows below.

Table 6.1 Field test result of the application

<table>
<thead>
<tr>
<th></th>
<th>S. Arlington St.</th>
<th>Fifth Ave.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SB</td>
<td>NB</td>
</tr>
<tr>
<td><strong>Direction</strong></td>
<td>R</td>
<td>L</td>
</tr>
<tr>
<td><strong>Ground Truth</strong></td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td><strong>App</strong></td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>**Diff (M-A)</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td><strong>% error</strong></td>
<td>12.5</td>
<td>0.529101</td>
</tr>
</tbody>
</table>

The total error of the application from 11:25 to 11:50 is 2.231237323%.
The total error of the application from 11:50 to 12:15 is 0.795228628%.

<table>
<thead>
<tr>
<th></th>
<th>S. Arlington St.</th>
<th>Fifth Ave.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SB</td>
<td>NB</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>L</td>
</tr>
<tr>
<td>Manual</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>App</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>Diff (M-A)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>% error</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The total error of the application from 12:17 to 12:42 is 0.980392157%.

| S. Arlington St. |          | Fifth Ave. |
|                 | SB       | NB         | EB         | WB         |
|                 | R    | L   | T  | R    | L   | T  | R    | L   | T  | R    | L   | T  |
| Manual  | 19   | 3   | 180| 8    | 3   | 243| 5    | 13  | 5  | 16   | 8   | 7  |
| App     | 18   | 3   | 179| 7    | 3   | 244| 5    | 13  | 5  | 15   | 8   | 7  |
| Diff (M-A) | -1 | 0   | -1 | -1   | 0   | 1  | 0    | 0   | 0  | -1   | 0   | 0  |
| % error | 5.263158| 0   | 0.555556| 12.5 | 0   | 0.411523| 0   | 0   | 6.25| 0   | 0  |
The total error of the application from 12:44 to 13:09 is 0.576923077%.

The algorithms for turning movement identification and vehicle delay measurement have been tested by the research team. The Java program code is installed in a Nexus One smartphone supporting multi-touch applications, this work involves two testing scenarios, and one is at a four-leg intersection used to test the turning movement function and the other at an intersection approach to measure vehicle delay.

Specifically, the intersection in scenario one has two through lanes in the North and South directions plus a left turn lane at each approach; the cross street is a minor road where the all the vehicle movements in the East and West directions are made from a single lane. The operator holding the smartphone collected the VTM data at one corner of the intersection for ten trials of 10 minute each in moderate traffic flow. Table 6.1 shows the results of the tests as compared with the ground truth collected by separate people at the same time periods. Only a portion of the results is included in the table to save space.
Table 6.2 Intersection VTM identification Tests

<table>
<thead>
<tr>
<th>Trial No.</th>
<th>North Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LT</td>
</tr>
<tr>
<td>1</td>
<td>-2</td>
</tr>
<tr>
<td>2</td>
<td>-3</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>-2</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>-4</td>
</tr>
<tr>
<td>8</td>
<td>-6</td>
</tr>
<tr>
<td>9</td>
<td>-2</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

One two-lane intersection approach is used in scenario two, where the reference point is selected 200 feet upstream of the intersection stop line. Similarly, the operator used the smartphone between the reference point and the stop line to measure vehicle delay in ten trials, each lasting 5 minutes.

It can be observed in Table 6.2 that the trials yielded consistent results, with errors ranging from -6% to 1% with an average weighted value of -1.5%. We noticed that the negative errors are generally larger than the positive errors; further checking on the details (recorded by
smartphone) indicates that the problem is caused by missing detection due to failure of the program to recognize the gestures. The positive errors indicate the mistakes by the operator. In Error! Not a valid bookmark self-reference., the difference between the measured vehicle delay and the (video recorded) ground truth is in the range of -4% to 6% with an average weighted value of 3%. We found that frequently there is a delay by the operator to input the gesture to track the turning movements, which may help explain the positive errors.

Table 6.3 Vehicle Delay Measurement Tests

<table>
<thead>
<tr>
<th>Trial No.</th>
<th>South Bound</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LT</td>
<td>TH</td>
<td>RT</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
</tr>
<tr>
<td>6</td>
<td>-2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>-4</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
CHAPTER VII

CONCLUSION

It is very difficult to collect turning movements and vehicle delay data to measure intersection performance even with the help of advanced detection systems today. Because of the complexities in collecting those data and lack of data processing algorithms, many transportation agencies at state and local levels still rely on data collectors who use hand-held devices to obtain still only part of the needed data due to the limitations of the devices. In this research, we presented the development and testing of an innovative method to collect intersection performance data, by using a smartphone in which advanced data collection algorithms are installed. The advantages of using the smartphone for this purpose are reviewed, and tests have been conducted to evaluate the effectiveness of the algorithms that are built on multi-touch technology. Preliminary testing results look very promising. The system is stable and the errors are generally small and can be reduced through refinements to the system.

This research shows the success of a feasibility study. Further evaluation (through statistical testing) and enhancements to the algorithms will help improve the system to be more accurate and reliable.
APPENDIX

LIST OF ABBREVIATIONS

EB: Eastbound
ET: Estimated Travel Time
ITT: Intersection Travel Time
GT: Ground truth
MT: Modified Travel Time
SB: Southbound
SR: Southbound Right Turn
SL: Southbound Left Turn
ST: Southbound Through
TM: Turing Movements
NB: Northbound
WB: Westbound
OD: Original Destination
BIBLIOGRAPHY


11. AT&T Developer Program- Developing Applications for Android. s.l. : AT&T .


13. Android handset sales grew 886 percent year on year in Q2 2010. [Online]