NCHRP Project 3-79a
Arterial Performance Measures

Final Report
Volume II

Vehicle Detector Signal Processing for Travel Time Estimation

Prepared for:
National Cooperative Highway Research Program
Transportation Research Board
National Research Council

Prepared by:

Joseph M. Ernst, James V. Krogmeier, and Darcy M. Bullock

Purdue University

August 9, 2010

Contents

Acknowledgments

This report contains the results of work made possible by funding from the Federal Highway Administration under National Cooperative Highway Research Program (NCHRP) project 3-79A.

The authors thank those who assisted in the progress of this research. They appreciate the helpful comments of the NCHRP panel and TRB paper reviewers. They also appreciate the support from George Coffey and all of Global Traffic Technologies, LLC. for their assistance in the development of the data collection infrastructure. The authors would also like to thank Purdue graduate student Mandoye Ndoye for his assistance with algorithm development and data collection and undergraduate students Dennis Lee, and Dhruv Lamba for their assistance with data collection and with the ground truthing and data reduction.

Abstract

Modern traffic management systems require accurate vehicle detection, speed estimates, and link travel times for traveler information, incident detection, ramp metering, traffic signal timing, and planning. Travel time is a performance metric of particular interest because it is well understood by passengers travelling throughout the network. Average travel times are currently reported, but the variance in travel time (i.e., travel time reliability) is also of great interest.

This report provides a signature matching algorithm for travel time estimation. It first reviews the literature for travel time estimation methods and sensors. This report then proposes a generalized framework for signature matching to be used with any sensor that collects sufficiently detailed signatures from a passing vehicle. This approach is based upon a stochastic communication theory based approach for estimating the travel time between any two standard speed trap vehicle detector pairs. The database developed to store the collected data is described in detail to facilitate the design of future signature matching travel time estimation projects. The resulting database had over 7000 records. Those signatures are viewable at the following URL: http://civl1122db02.ecn.purdue.edu/nchrp

The signature matching travel time estimation algorithm was evaluated on approximately 7,000 vehicle's signatures collected at two locations on 12 different days. Although in practice, one would like to estimate travel time over segments of 1-2 miles, the algorithm was developed using a sensor spacing of 100-400 feet to provide an efficient and cost-effective method for developing ground truth data.

The signature matching algorithm is shown to work well to estimate general characteristics of a travel time distribution, with match rates on the order of 50%. Several pages of histograms are provided to graphically compare the estimated travel time histograms with ground truthed data.

1 Introduction

Modern traffic management systems require accurate vehicle detection, speed estimates, and link travel times for traveler information, incident detection, ramp metering, traffic signal timing, and planning [1][2]. While there are many traffic management performance measures, travel time is a metric of particular interest, because it is well understood by all passengers travelling throughout the network. While average travel time is an important metric, the variance in travel time (i.e., travel time reliability) is also evaluated in many studies. To study these performance measures, the travel times must be estimated.

The travel time estimation method described here uses currently installed roadway sensors (e.g., inductive loops and microloops) to generate travel time estimates. The method is motivated by a model of the electromagnetic data that can be collected from these sensors. This model and the travel time estimation algorithm is presented. An implementation of the travel time estimation algorithm is described in detail and the results are analyzed to show that reasonable travel time histograms can be generated. This travel time estimation algorithm is intended for deployment at any pair of networked speed trap data collection stations.

1.1 Existing Work

Travel times can be estimated either by directly measuring the travel time of probe vehicles or by using point measurements (e.g., speed, occupancy, vehicle counts, etc.) to infer the travel time. Point measurements are convenient because they are already automatically collected at many locations [3][4][5][6]. The quality of these inferred travel time estimates depends on how well the point measurements reflect the traffic for the entire path of the vehicle. Inferred travel times can be very effective where the sensor stations are dense [7][8].

Travel times can also be measured directly by recording when a particular vehicle enters and leaves the road segment of interest [9]. An early method of capturing travel time data was to write down license plates and times at two locations and then to pair the data sets to generate a list of travel times. This license plate travel time estimation method has become more viable with Automatic License Plate Recognition (ALPR) [10] [11] [12]. Other methods for collecting measured travel times is to track cell phones [13] [14] [15], Bluetooth devices [10] [14] [16] [17] [18] [19], video processing [20] [21], or toll tags [7]. Also, much more detailed information about a vehicle's trajectory can be found by collecting GPS measurements from an instrumented vehicle [23] [24] [25] [26].

One disadvantage of the measured travel time methods is the lack of privacy. Some states have begun to pass legislation against collecting data from electronic devices. One example is House Bill 1031 in Washington State. Additionally, the methods may require expensive equipment (i.e., ALPR devices), or may have a low percentage of vehicle travel times captured if relying on capturing data from specific electronics.

Signature matching is another alternative for travel time estimation [27] [28]. This is similar to the measured travel time methods except that the data collected from each vehicle is not a unique identification like a license plate. Instead, it uses data collected from common roadway sensors.

This data serves as a fingerprint for each vehicle; however, the fingerprints of vehicles are not necessarily unique, because the data collected from two vehicles of the same make and model are likely to be indistinguishable. The data collected from the sensors is divided into segments where each segment corresponds to the data collected from a vehicle. These segments, which serve as the vehicle's fingerprint, are referred to as the vehicle's signature. The signatures from the upstream sensors are compared to the signatures from the downstream sensors to decide which upstream and downstream signatures are likely to have been from the same vehicle. Since the data collected does not require onboard electronics (i.e., Bluetooth devices, or toll tags), in principle it can attempt to match every vehicle. For the same reason, it does not have the certainty of correctly matching vehicles that the other methods do. However, signature matching does not have privacy concerns since the fingerprint is not guaranteed to be unique.

The effectiveness of the signature matching algorithm is dependent on its ability to correctly identify which signature from the upstream data collection site is from the same vehicle as a signature from the downstream data collection site. A number is associated with each possible upstream and downstream pair that indicates the similarity (or difference) between the two signatures. This similarity metric, along with some constraints about traffic flow, are used to pair upstream and downstream signatures corresponding to the same vehicle.

A travel time is then extracted from each pair by subtracting the time that the signature was collected at the upstream data collection site from the time that the signature was collected from the downstream data collection site.

Early work in matching upstream and downstream data sets was not done for each vehicle, but instead the matching was performed on time averaged counts data. This means that the number of vehicles that arrived in each time window at an upstream and downstream location is recorded. The patterns of clustered vehicles (i.e., platoons) are matched from the upstream data collection site to the downstream data collection site. The method used to match these platoons is the cross-correlation of this averaged counts data. This then yields an aggregate travel time histogram [29].

Most traffic sensors are connected to a detector card. This detector card controls the sensor, processes the sensor's data and presents useful data primarily to the traffic controller, but the data is also used by other devices like data loggers. The simplest data that can be collected from a standard detector card for matching purposes is the vehicle's length [30][31][32]. This can be collected with information that is available from the binary presence indication from the detector card. This data is readily available from any detector card since the detector card must send the vehicle detection data to the controller.

The Sensys sensor [33] [34] has been used by several studies to collect more detailed information about each vehicle for use in signature matching [35] [36] [37]. These sensors are a magneto-resistive sensor used to measure the disturbance of the Earth's magnetic field caused by a passing vehicle. The Sensys sensors are wireless, which allows for easier installation, but also requires battery power. These sensors do not transmit the raw form of the magnetic measurements, but instead usually report only the presence or absence of a vehicle. They can also be switched to another mode where they transmit a vector of local extrema of the measured

magnetic field. These local extrema are used as the vehicle's signature. Sometimes multiple sensors used in each lane to enhance the data set as in [37] where seven sensors were used per lane. The Sensys sensors were also used for signature matching in [36] where five sensors were used in each lane. The similarities of the upstream and downstream vehicle's signatures are then used in a shortest path algorithm to decide which upstream vehicle signature matches each downstream vehicle signature.

Inductive loops and microloop sensors are also used for signature matching for travel time estimation [28] and also for the dual problem of speed estimation [41]. These studies use the cross-correlation function between upstream and downstream signatures to estimate the similarity of two signatures and the delay between them. A similar method has been used with microphones for speed estimation [43]. Signature matching methods using these sensors that are based on the cross-correlation similarity metric are described in Chapter 2.2.3 and Chapter 3.2.2.

1.2 Vehicle Detection Sensors

A variety of technologies are in use for vehicle detection including pneumatic tubes [44], magnetometers (e.g., microloops) [45], mutual inductive coupling sensors (i.e., inductive loops), video cameras [45], and microwave radar sensors [46]. Each technology has its own strengths and weaknesses [47].

This document focuses on data collected from inductive loop and microloop detectors. Inductive loops detect the presence of a vehicle when mutual inductance between the loop and the vehicle lowers the inductance of the loop. Microloops measure the perturbation of the Earth's magnetic field due to the vehicle. This measured field is transduced to an inductance that mimics that of an inductive loop. Both the inductive loop and the microloop are then seen as a variable inductor to the rest of the detector circuit. This variable inductance is an inductive element in an LC tank, which sets the frequency of a sinusoidal oscillator.

When no vehicle is present, the frequency of this oscillator is at its resting frequency. When a vehicle passes over the sensor, it causes the sensor to change its inductance and therefore causes a change in frequency. The difference between the period of the oscillation of the instantaneous frequency and the resting frequency is measured by counting the oscillations of a 32 MHz oscillator. This integer value is then streamed from the detector card at a rate of approximately 100 samples per second.

Because of significant parasitic inductances associated with long lead-in cabling from the sensor under the road to the detector card in the roadside cabinet and parasitic capacitances associated with coupling to the Earth, it is impossible to precisely set the resting frequency [47]. Furthermore the resting frequency varies with time and environmental parameters. It is therefore necessary for the detector card to estimate the resting period of the oscillator.

In addition to this streaming data, the detector card also produces a discrete output of either 0 Volts or 24 Volts indicating whether the detector card is currently detecting a vehicle. Throughout this document the streaming data due to a vehicle is called the vehicle's signature and the discrete output is the detector card's “call” function.

Since an LC oscillator's frequency is inversely proportional to the square root of the inductance (i.e., $f = K/\sqrt{L}$ [48], it can be shown that for small perturbations $\Delta L = L_{ref} - L$ in the sensor's inductance relative to the reference inductance, the relative change in frequency is proportional to the relative change in inductance. Furthermore, the relative change in period count is approximately proportional to the relative change in frequency or inductance assuming that 32 MHz is significantly larger than the micro-loop oscillation frequency (in fact, it is more than 100 times larger for the sensor used to make our measurements). Therefore,

$$\frac{\Delta N}{N_{ref}} \cong -\frac{\Delta f}{f_{ref}} = \frac{1}{2}\frac{\Delta L}{L_{ref}} \quad (1)$$

1.3 Approach

The signature matching algorithms discussed in other works rely mostly on feature recognition and other heuristic measures of similarity. The approach taken here is to model the matching problem as a communication system and to then find the maximum-likelihood estimator of travel time. This method will be applied first to a single pair of sensors that are arranged in a speed trap configuration as shown in Figure 1.1. This model and associated travel time estimation algorithm is developed in Chapter 2.

Figure 1.1: Sensors in a speed trap configuration with the lead sensor and lag sensor 14 ft to 22 ft apart

The Traffic Detector Handbook [49] recommends that the lead and lag sensors should be placed approximately 16 ft center to center (but they often range from 14 ft to 22 ft in practice). This configuration is intended to provide a speed estimate of each vehicle. This speed estimate has a one to one relationship to the vehicle's travel time between the sensors through Equation (2).

$$tt = \frac{d}{v} \quad (2)$$

where tt is the travel time, d is the distance between the sensors, and v is the magnitude of the velocity of the vehicle.

This travel time is then used to produce a segment travel time estimate between two pairs of speed traps as shown in Figure 1.2. The first speed trap that the vehicle crosses is referred to as the upstream speed trap and the second is referred to as the downstream speed trap. The travel times are found by matching vehicles between the two speed traps as shown in Figure 1.3. This algorithm is developed in Chapter 2 and Chapter 3.

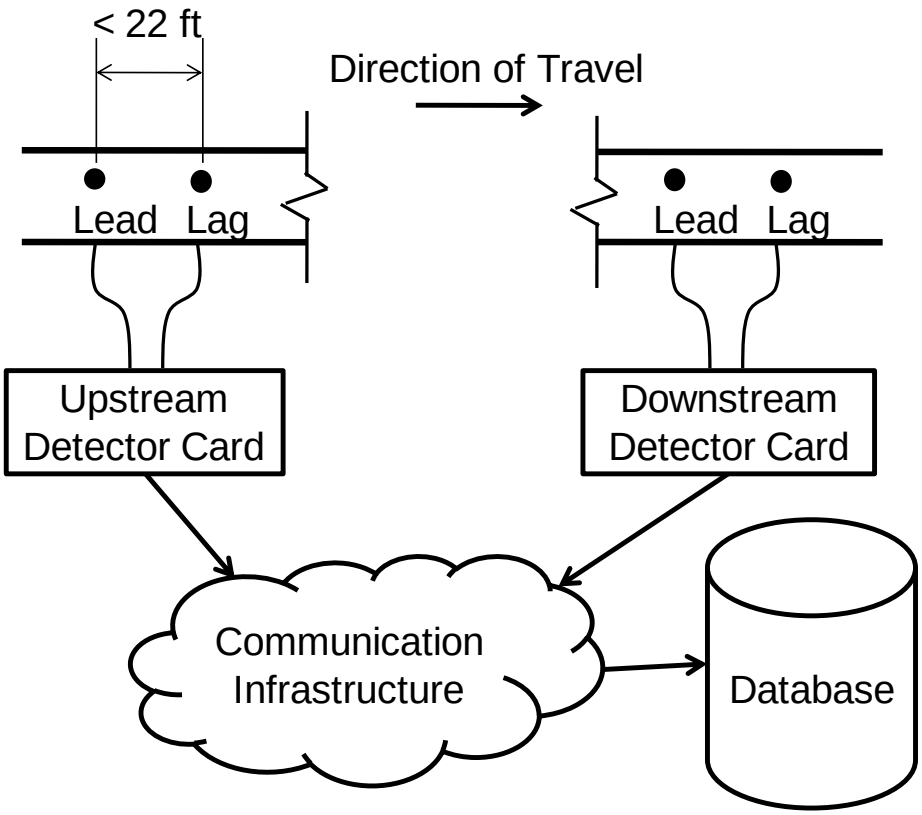


Figure 1.2: Generic upstream and downstream speed trap pair configuration

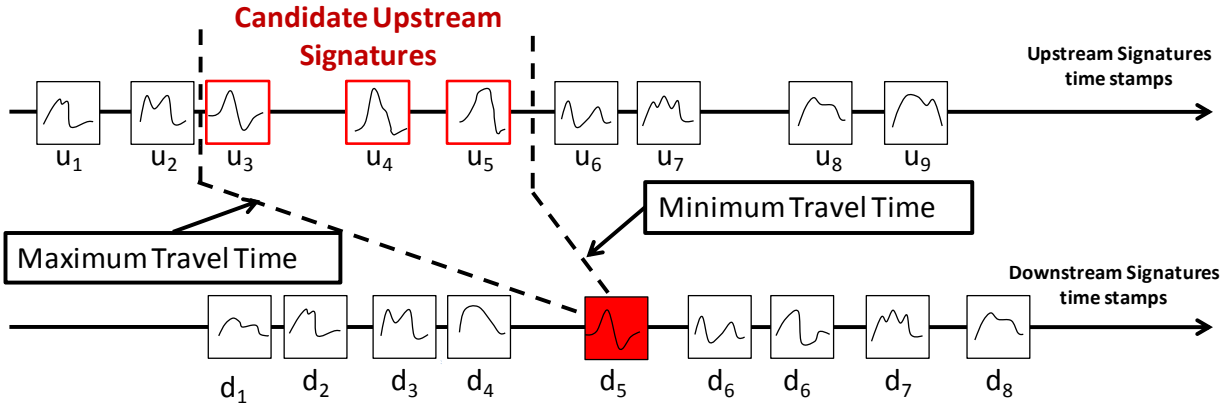


Figure 1.3: Upstream and downstream signature matching.

1.4 Organization of Paper

Chapter 2 develops the travel time estimation algorithm for two closely spaced sensors by solving the dual problem of speed estimation. Chapter 3 extends this algorithm to find the travel time with two pairs of sensors with a longer distance between them. Chapter 4 describes the data collection sites, infrastructure, and procedure. Chapter 6 analyzes the results of the travel time estimation algorithm. Chapter 7 discusses the predicted performance of the travel time estimation algorithm across multiple intersections. Chapter 8 summarizes the work thus far and discusses future work to improve travel time estimation and signature matching probabilities.

2 Travel Time Estimation Between Lead and Lag Sensors

This chapter will develop a travel time estimation algorithm between a pair of closely spaced sensors. First, a model for the signatures and its assumptions are described. Second, the maximum-likelihood travel time estimate is derived for the model presented.

2.1 Model Description

When a vehicle travels over any sensor, there is effectively a signal, $s[n]$, transmitted with information about the vehicle (i.e., the vehicle's signature) to the sensor. The signature received by the sensor is assumed to be this transmitted signature embedded in noise such that

$$r_i[n] = s[n] + w_i[n] \qquad i \in \{1,2\} \tag{3}$$

where $r_1[n]$ is the signature from the lead sensor, $r_2[n]$ is the signature from the lag sensor, $w_i[n]$ is the associated noise. In this report, the discrete index n should be understood such that, $t = n\Delta T$, where t is the continuous time (in seconds) and ΔT is the sampling period (in seconds). For this model, the noise is assumed to be independent identically distributed Additive White Gaussian Noise (AWGN) with zero mean and variance, σ^2.

Several assumptions are used in deriving the maximum-likelihood delay estimate. First, the sampling rate is assumed to be above the Nyquist frequency which makes this delay estimation problem equivalent to the continuous time problem. This is justified since the signature should be limited to a frequency range less than 22 Hz. Assuming that the aperture of a microloop detector is approximately 2 ft, a vehicle travelling over the microloop at 30 mph would have an aperture in time of about 45 milliseconds. This can be modeled as a simple averaging of the vehicle's effect. If this averaging is uniform, the frequency response would be a sinc function, $\frac{\sin(\pi t)}{\pi t}$, with its first null at about 22 Hz. This may be different for various sensors and data collection sites. This approximate estimate of the maximum frequency can be found for any sensor from the following equation:

$$F_N = \frac{1}{T_N} = \left(\frac{v \text{ mph}}{S_A \text{ feet}}\right)\left(\frac{5280 \text{ feet}}{1 \text{ mile}}\right)\left(\frac{1 \text{ hour}}{3600 \text{ seconds}}\right) \cong 1.47\frac{v}{S_A} Hz \tag{4}$$

where S_A is the length of the sensor's aperture in feet and v is the velocity of the vehicle in mph.

The next two assumptions rely on the close proximity of the sensors in the speed trap configuration. Due to this close proximity it is assumed that the vehicle maintains a constant speed while traversing both sensors and that each vehicle that travels over the lead sensor proceeds directly to travel over the lag sensor.

Estimating the delay between two vehicle signatures is then equivalent to the well understood problem of estimating the delay between a transmitted signal and a noisy received version of the transmitted signal [50].

Let $r_i(t)$ be the signature from sensor i at time t, where $t = 0$ when the vehicle is first detected at sensor i. Even though the magnetic field at the sensor is a function of the vehicle's relative position, the data can be equivalently indexed in time due to the constant velocity assumption.

Also, $s_2[n + n_0] = s_1[n]$, where n_0 is the travel time between the two sensors. We can then write $r_2[n + n_0] = r_1[n] + w[n + n_0]$, where $w[n]$ is $w_2[n] - w_1[n - n_0]$. If $r_1[n]$ is then interpreted as the transmitted signal and $r_2[n]$ is the received signal embedded in AWGN with variance, $2\sigma^2$, then this is the standard delay estimation problem except that the noise has twice the variance. It is well established that the maximum-likelihood estimator for this delay problem is the delay that maximizes the cross-correlation of the two signals, $r_1[n]$ and $r_2[n]$[50].

2.2 Algorithm Development

To apply the results of this model, the portion of the data stream corresponding to each vehicle must be extracted and prepared for the cross-correlation step. This algorithm includes the following steps:

- Low-Pass Filter
- Segmentation
- Cross-Correlation
- Quality Filter

These steps are described in the following subsections.

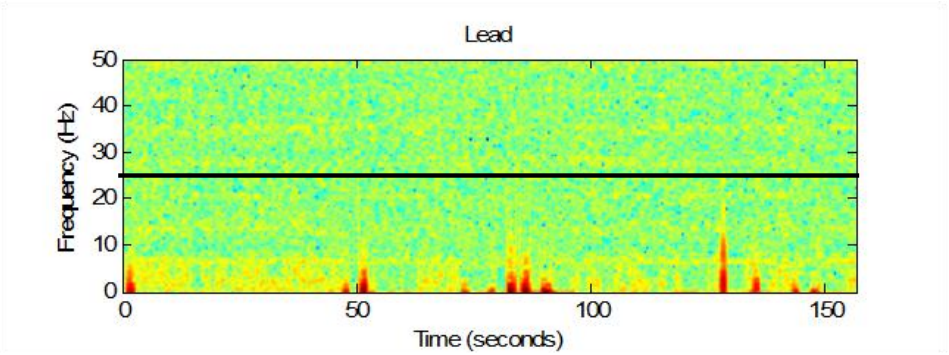
2.2.1 Low-Pass Filter

The data stream from each sensor is first passed through a low pass filter. This removes noise from the system in frequency ranges where vehicle signatures do not exist. As was described in the model, this includes all frequencies above 22 Hz. To allow for errors in the model, 25 Hz is used as a safe cutoff frequency.

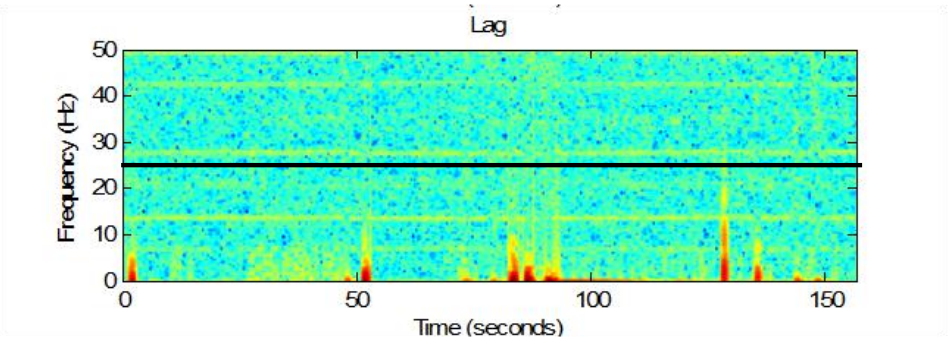
Figure 2.1a and Figure 2.1b show a spectrogram of data collected at one pair of sensors. Time is plotted on the x-axis and frequency is plotted on the y-axis. Each dark region of the spectrogram corresponds to the frequencies in the signature generated by a passing vehicle at that time. The frequencies influenced by the vehicle are almost always less than 25 Hz, which is consistent with the model of the sensor's averaging described above. Therefore, the low-pass filter is designed to only pass the frequencies less than 25 Hz. The frequency response of this filter is shown in Figure 2.1c. The magnitude of the frequency response is a unitless scaling factor that is applied to the components at each frequency ω. In signal processing it is given the name $\left|H\left(e^{jw}\right)\right|$. The filter chosen is an equiripple Parks-McClellan low pass filter. This graph shows that portions of the signal corresponding to frequencies lower than 25 Hz will be kept (multiplied by a number close to one) and the portions of the signal corresponding to frequencies greater than 25 Hz will be removed (multiplied by a number close to zero). The extent to which the multipliers deviate from zero and one is characterized by the frequency response plot

If this algorithm were to be used on a highway installation where speeds are on the order of 70 mph, a higher frequency might need to be used. Even in the cases where there might be energy

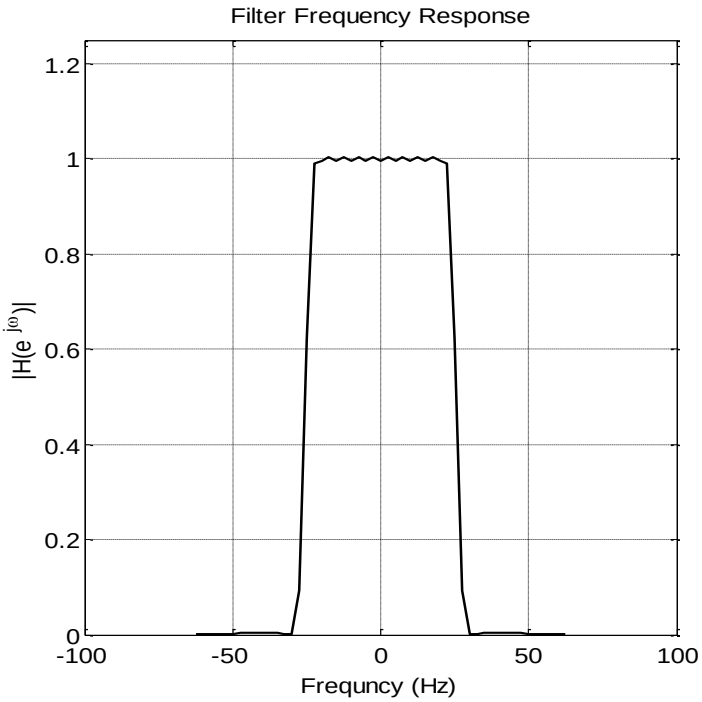
in the higher frequencies, this will not greatly affect the correlation coefficient because of the natural low pass filter characteristic of the cross-correlation function. For example, Figure 2.2 shows an example of two vehicles. Their spectrograms in Figure 2.2c and Figure 2.2d show that their signatures have energy up to approximately 7.5 Hz. A filter with a 5 Hz upper bound is used and the results are shown. The vehicle's signatures before and after the filter are shown in Figure 2.2a and Figure 2.2b. The correlation coefficient from the lead to lag signature for the first vehicle changes from 0.9713 to 0.9742 and the correlation coefficient between the lead and lag signatures for the second vehicle changes from 0.9976 to 0.9987. These numbers change only slightly because the correlation coefficient metric is only slightly affected by the low pass filtering. Also, notice that both numbers increased because the differences caused by noise are removed. Also, the correlation coefficient calculated from the lead signature of vehicle one and vehicle two only changes slightly from 0.8539 to 0.8560. Even though part of the signal was removed by the low pass filter, the correlation coefficient remains a reliable metric for determining whether or not signatures originate from the same vehicle.



a) Spectrogram from Lead Sensor

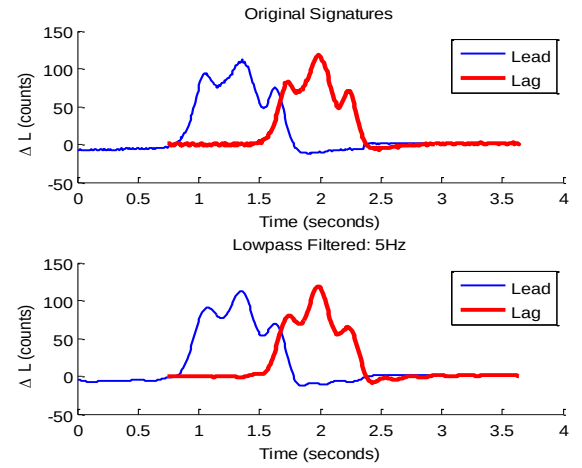


b) Spectrogram from Lag Sensor

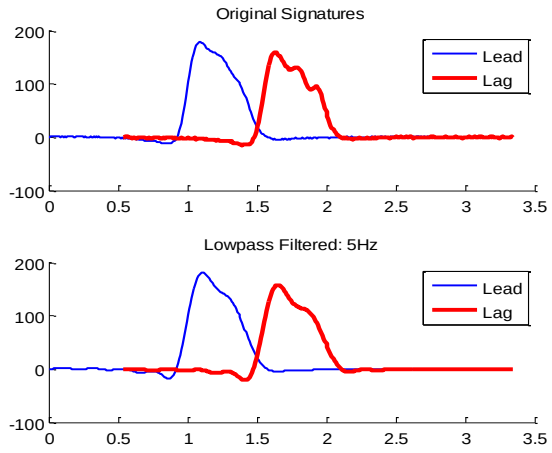


c) Frequency Response

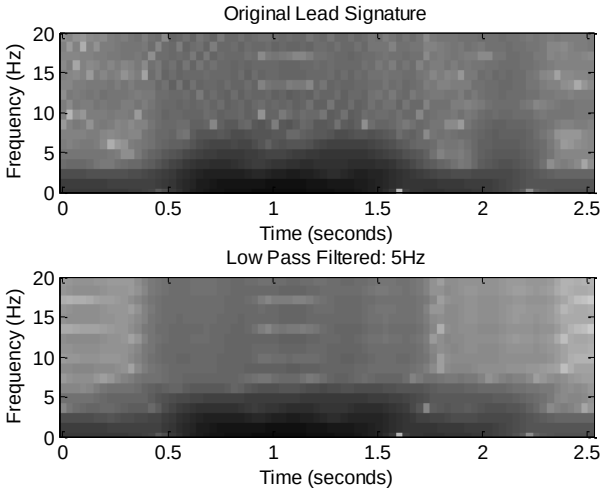
Figure 2.1: Design of Low-Pass Filter



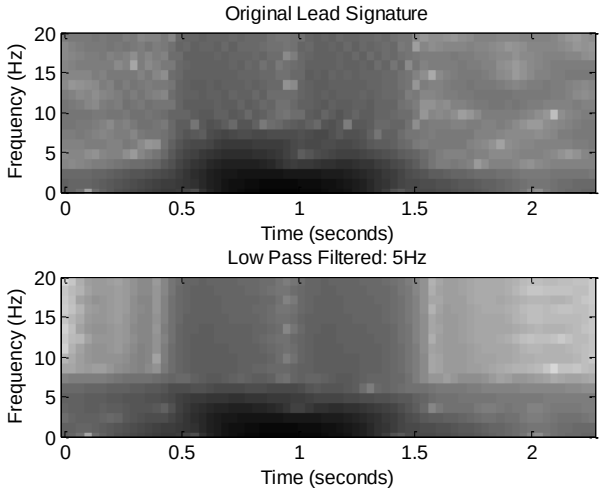
a) Vehicle 1 signatures



b) Vehicle 2 signatures



c)Vehicle 1 Spectrogram



d) Vehicle 2 Spectrogram

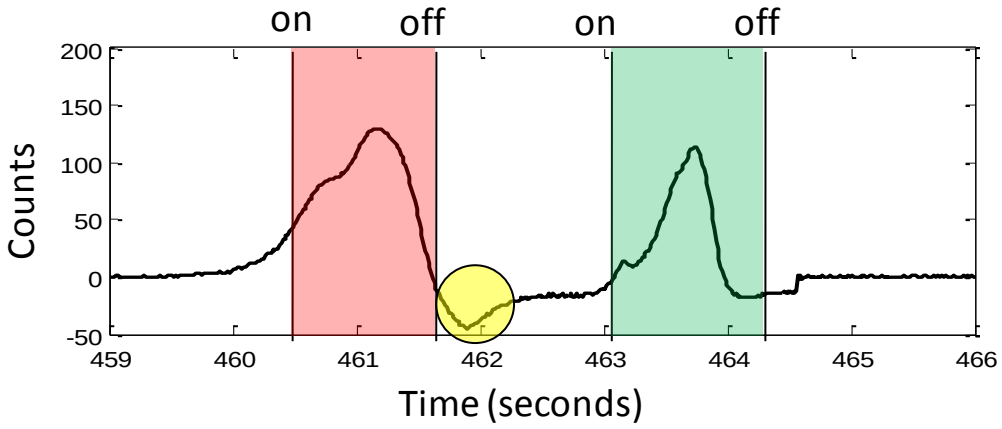
Figure 2.2 Effect of low pass filtering.

2.2.2 Segmentation

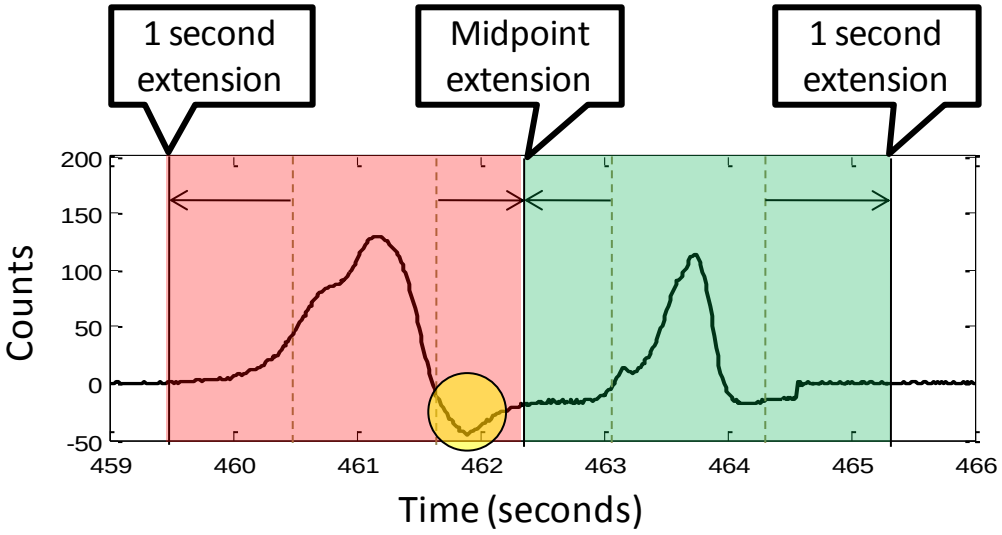
Segmentation is the process of breaking a data stream into finite length individual signatures. Figure 2.3 shows a data stream from a microloop sensor corresponding to two vehicles. The solid vertical lines in Figure 2.3a show the presence indication from the detector card. For example, the detector card detects the second vehicle from about 463.1 seconds to about 464.3 seconds. The detector card's internal detection algorithm is used because these vehicle detection algorithms are known to be very reliable for counting vehicles. Therefore it is unlikely that many vehicles are undetected or treated as multiple vehicles.

These detections alone are not appropriate for signature matching, because they tend to throw away useful portions of the signature that fall outside the detected region. One example is the circled portion of the first vehicle. To avoid losing this information, the segments are extended

by adding one second to the beginning and the end of each detected region. The segments should not however include any of the adjacent vehicle's signatures. Therefore, if the gap between detections is less than 2 seconds, the segments are extended only to the midpoint of the gap. These extended segmentations are shown in Figure 2.3b. Examples of the full one second extension occur before the first vehicle and after the second vehicle. The midpoint segmentation has been chosen between the vehicles. The dip at the end of the first vehicle's signature is included by the extended segmentation.



a) Detector Card Segmentation



b) Extended Segmentation

Figure 2.3: Justification for extension of signatures from detector card

2.2.3 Cross-correlation

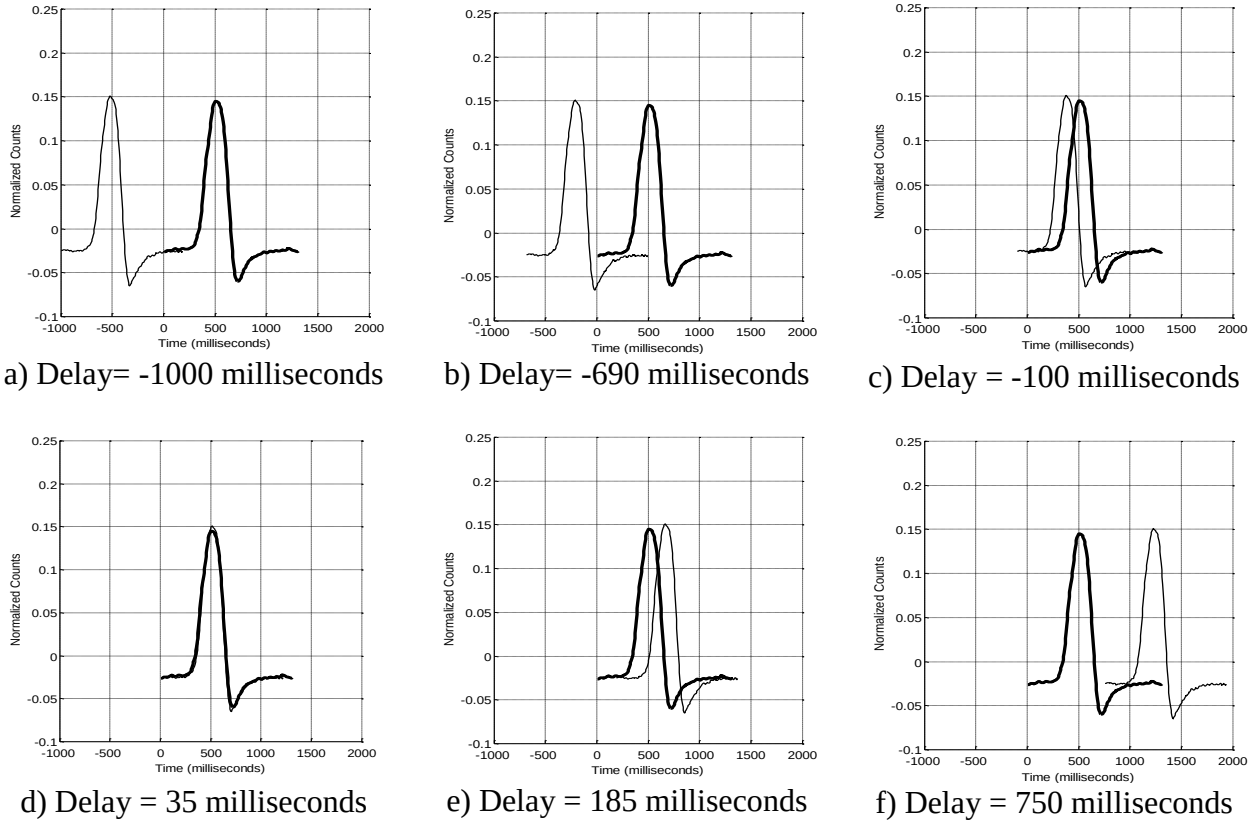
Figure 2.4 and Figure 2.5 and show the process of generating the cross-correlation function of an lead and lag normalized signature. Figure 2.4a-f show the lag signature plotted with a thick line and the corresponding delay applied to the lead signature plotted with a thin line. In each of the subfigures, the lead signature has been delayed or shifted in time to correspond to a potential travel time. Figure 2.4 a, b, and f show examples where the lead and lag signature are not aligned. Since they are not aligned, the cross-correlation function at points a, b, and f in Figure 2.5 is not yet maximized. Figure 2.4c and e show that the correlation function starts to rise as the signals are close. Figure 2.4d shows that the correlation function is almost one when the correct delay is applied. The maximum value of the cross-correlation function is called the *correlation-coefficient*. The more similar two signatures are, the higher the correlation coefficient will be. The correlation-coefficient has a range of minus one to one where it can only be equal to one

when the signatures being compared are identical or scaled versions of each other. Since both of the signatures in this example are from the same vehicle, the correlation coefficient is close to one. Correlation functions of signatures from different vehicles tend to have a lower correlation coefficients. This makes the correlation coefficient an appropriate similarity metric.

Another way to interpret the correlation coefficient is as a generalized dot product. The correlation coefficient between two signals is defined as:

$$\rho = \frac{x \cdot y}{\|x\|\|y\|}$$

The cross-correlation function allows the x and y vectors to shift relative to each other before the dot product is taken. The maximum dot product is taken as the correlation coefficient and the shift corresponding to the maximum dot product as the travel time estimate of the vehicle.



a) Delay= -1000 milliseconds b) Delay= -690 milliseconds c) Delay = -100 milliseconds

d) Delay = 35 milliseconds e) Delay = 185 milliseconds f) Delay = 750 milliseconds

Figure 2.4: Six Example Time Shifts Used to Calculate Cross-correlation function.

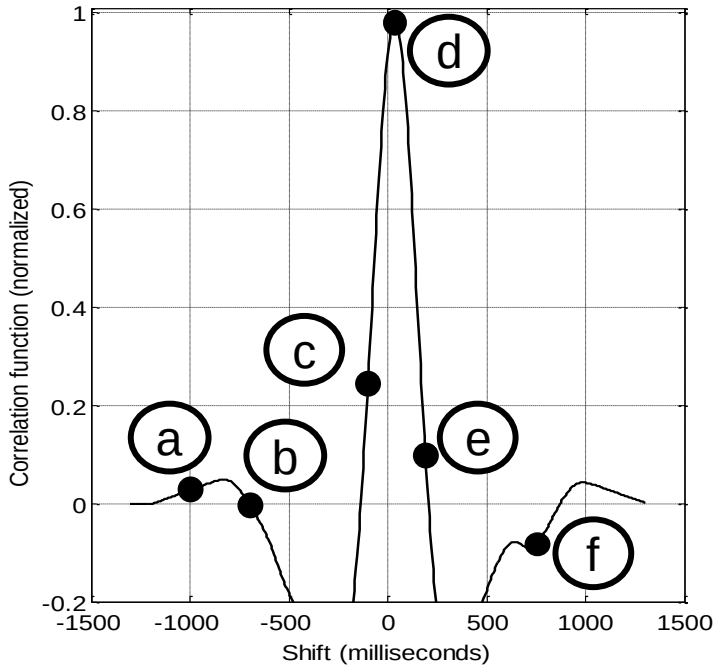
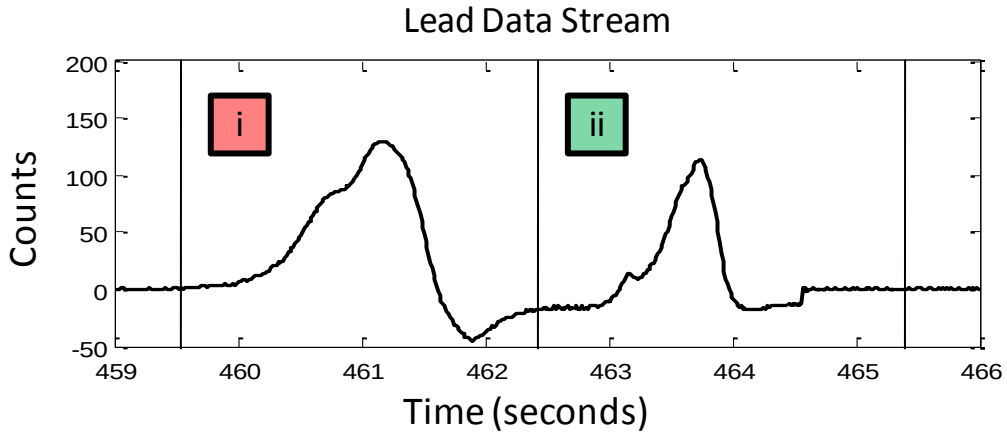


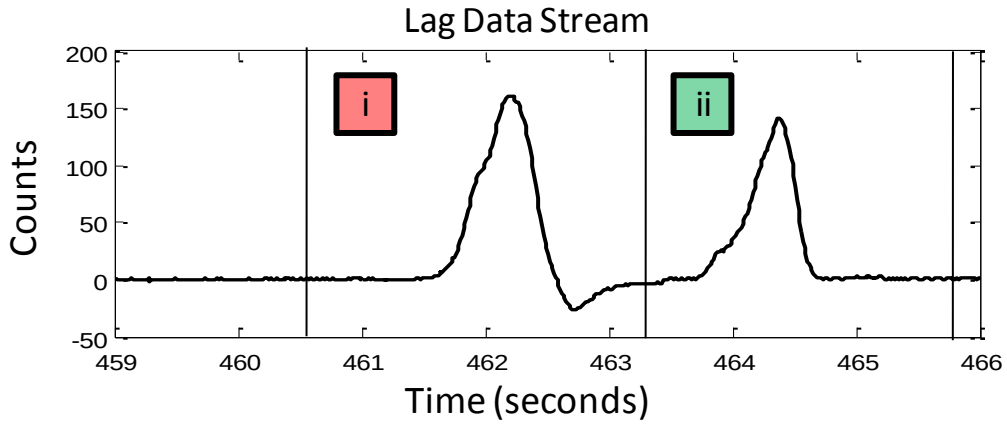
Figure 2.5: Correlation Function calculated from lead and lag signatures in Figure 2.4.

To calculate the cross-correlation function, the lead and lag signatures must first be paired. Figure 2.6 demonstrates the Lead/Lag pairing step. This pairing is not complicated due to the close proximity of the lead and lag sensors. Each lead signature is simply paired with the next lag signature. The normalized cross-correlation function for vehicle (i) and vehicle (ii) are shown in Figure 2.7a and Figure 2.7b respectively. The delay on the x-axis that maximizes the function is x in Equation (5).

$$\text{speed} = \left(\frac{\text{distance}}{\text{time}}\right)\left(\frac{14 \text{ feet}}{x \text{ seconds}}\right)\left(\frac{1 \text{ mile}}{5280 \text{ feet}}\right)\left(\frac{3600 \text{ seconds}}{1 \text{ hour}}\right) = \frac{9.545}{x} \text{ mph} \quad (5)$$

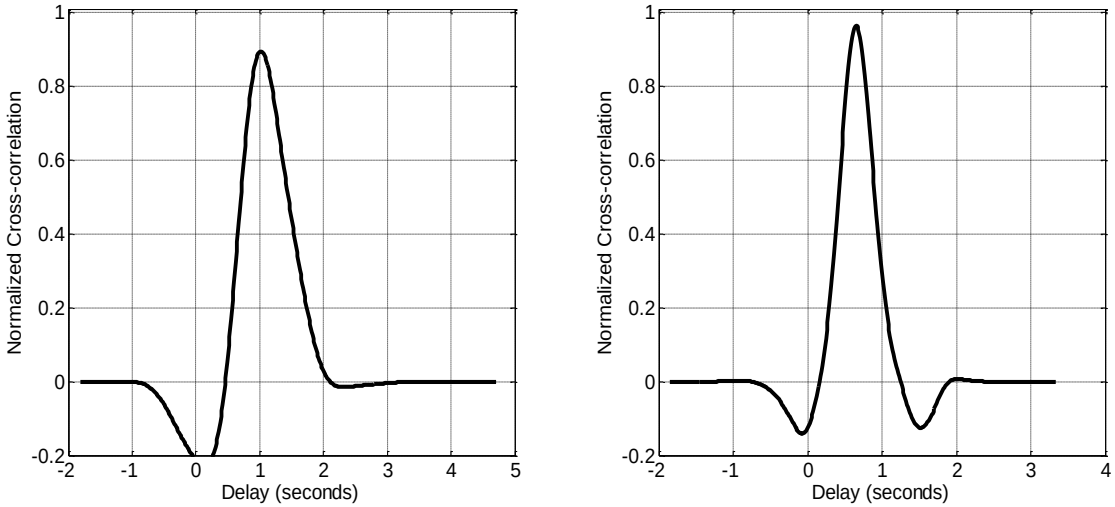


a) Lead Data Stream



b) Lag Data Stream

Figure 2.6: Lead/Lag Pairing for Speed Estimation



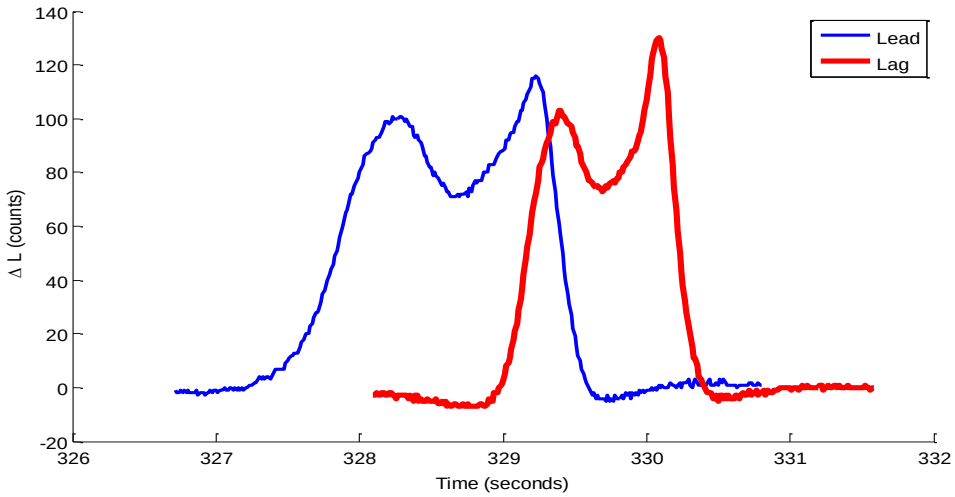
a) Vehicle i b) Vehicle ii

Figure 2.7: Cross-correlation function used to find the speed of a vehicle over a speed trap by finding the delay corresponding to the maximum of the cross-correlation function

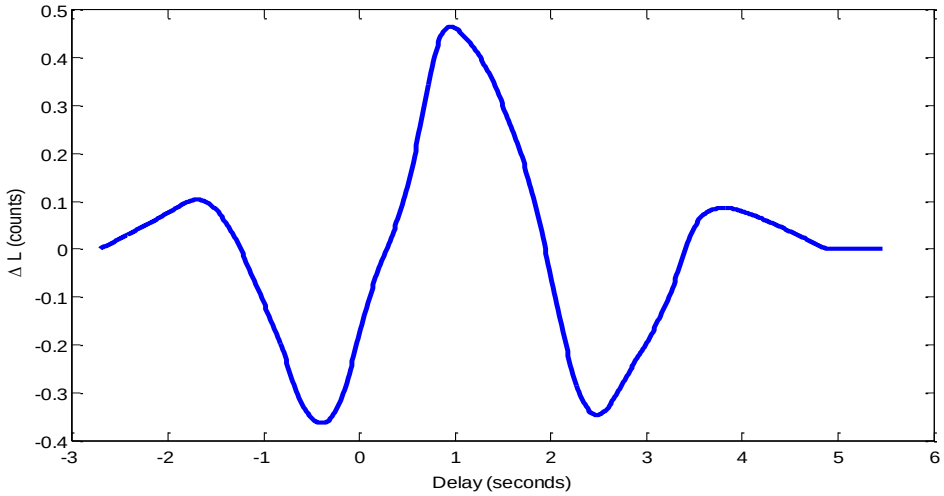
2.2.4 Quality Filter

The quality of the signatures collected from a speed trap can be evaluated by using the cross-correlation function between the lead and lag signatures from a vehicle. If the vehicle is travelling a constant speed, the lead and lag signatures tend to match well. When the vehicle is accelerating, the lead signature and lag signature become distorted. This distortion means that the signature is not a valid representation of the vehicle's signature.

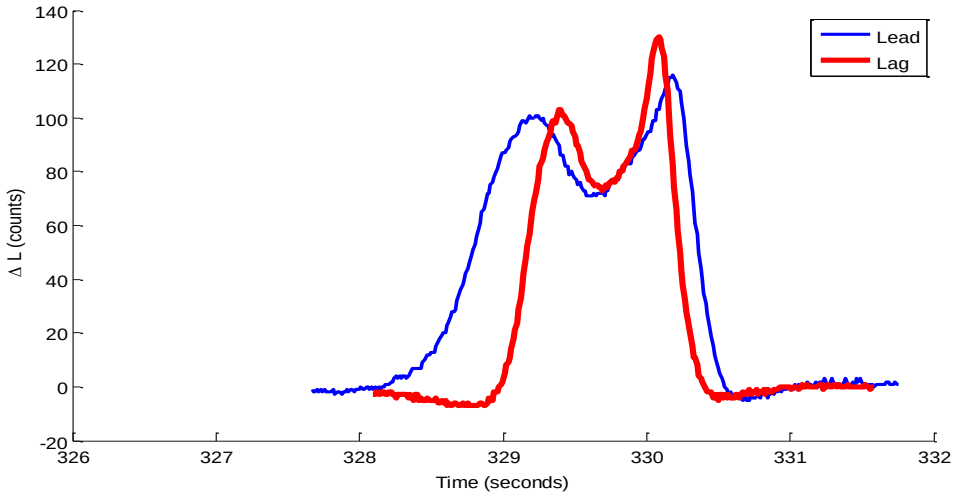
An example pair of signatures that are distorted due to acceleration/deceleration are shown in Figure 2a. The lead signature is much wider than the lag signature because the vehicle travelled faster over the lag signature than it was over the lead signature. The corresponding normalized correlation function is shown in Figure 2b. The correlation coefficient for this pair is only 0.465. The shift calculated by the cross correlation function is 951 milliseconds. This shift is applied to the lead signature in Figure 2c. The shifted lead signature begins before the lag signature and ends after the lag signature. No shift can better align the signatures. Lead and lag signature pairs with low correlation coefficients should not be used without acceleration/deceleration compensation.



a) Lead and Lag signature pair with acceleration/deceleration distortion.



b) Normalized cross correlation function for lead and lag signature.



c) Lead signature shifted to optimal delay for comparison with lag signature.

Figure 2.8: The adverse effect of acceleration/deceleration distortion

3 Travel Time Estimation Between Two Pairs of Sensors

3.1 Model Description

The model used for each of the sensors in this case is the same as that described in Section 2.1. The difference is that the upstream pair of sensors are a long distance from the downstream pair of sensors. The assumption that the vehicles travel over the second site immediately after the first is then no longer valid. This requires a more complicated algorithm to pair the signatures. Also, the vehicle's speed may change between the pairs of sensors. This algorithm has therefore been designed to normalize the signatures for the speed variability before attempting to match the vehicles.

3.2 Algorithm Development

The travel time estimation algorithm is actually the algorithm from the previous chapter applied multiple times in the process of producing one segment travel time estimate. A block diagram of the matching algorithm is shown in Figure 3.1. This first step in the algorithm is called the Normalized Signature Generator, which produces a vehicle signature which has been normalized in both time and energy. This step normalizes the signatures from both the upstream and downstream data collection stations. The similarity processor calculates the similarity of each downstream vehicle with each upstream vehicle in its feasibility window. Each downstream signature is paired with an upstream signature that most likely was generated by the same vehicle. These travel time pairs are then statistically processed to generate reliable travel time histograms. These steps are now described in detail.

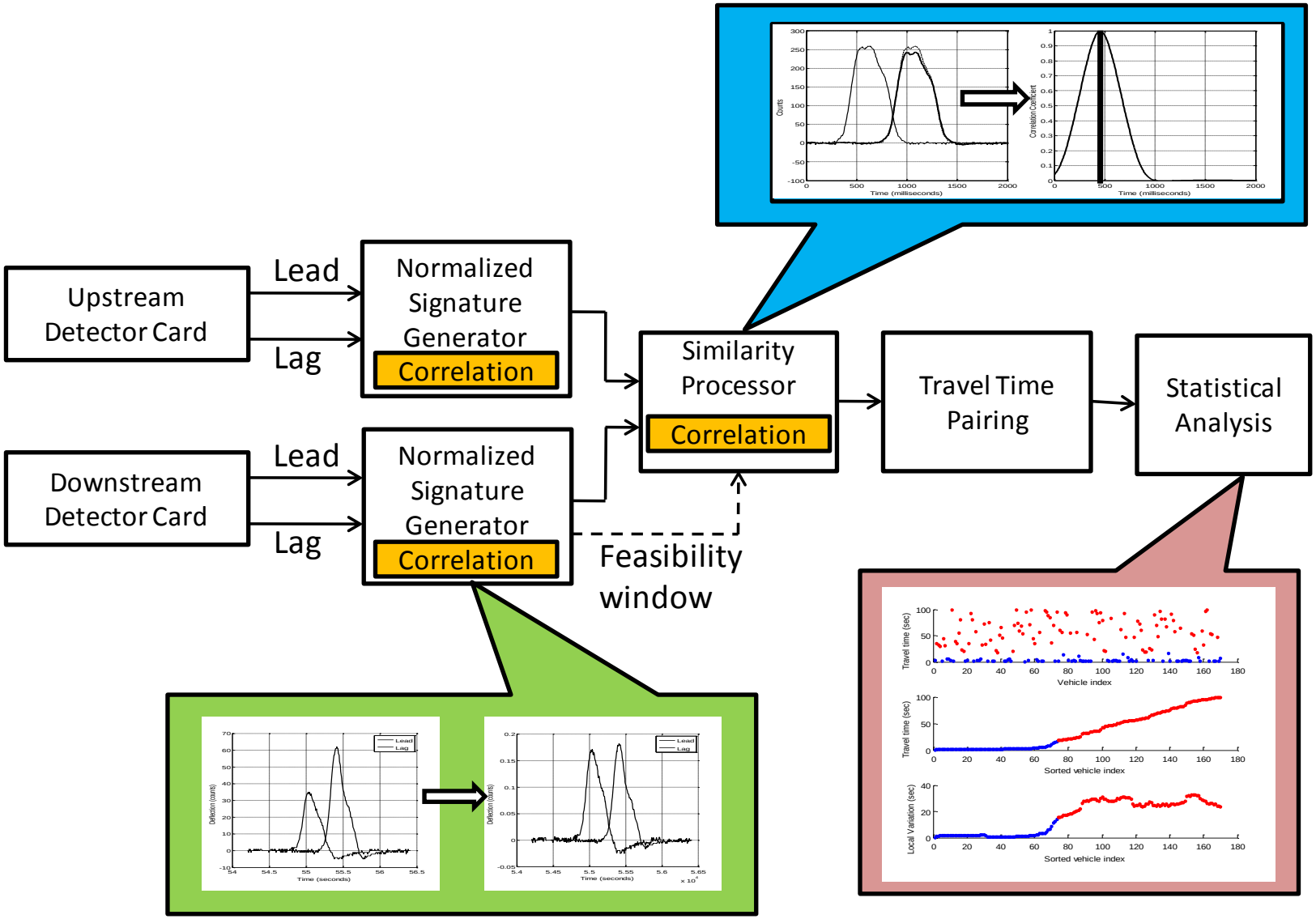


Figure 3.1: Block diagram for matching with two pairs of sensors.

3.2.1 Normalized Signature Generator

In reference to Figure 3.1, the Normalized Signature Generator produces appropriately normalized signatures from the lead and lag data streams for use by the similarity processor. There are four sub blocks of the Normalized Signature Generator as shown in Figure 3.2.

- Low-pass filter
- Segmentation
- Lead/Lag Pairing
- Normalization

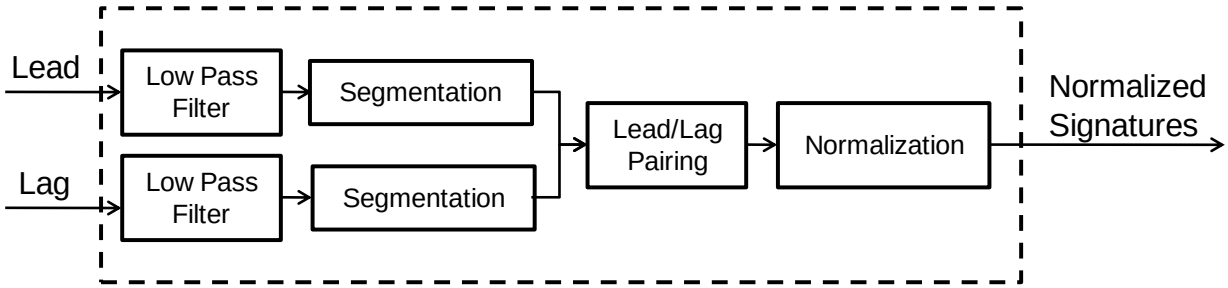
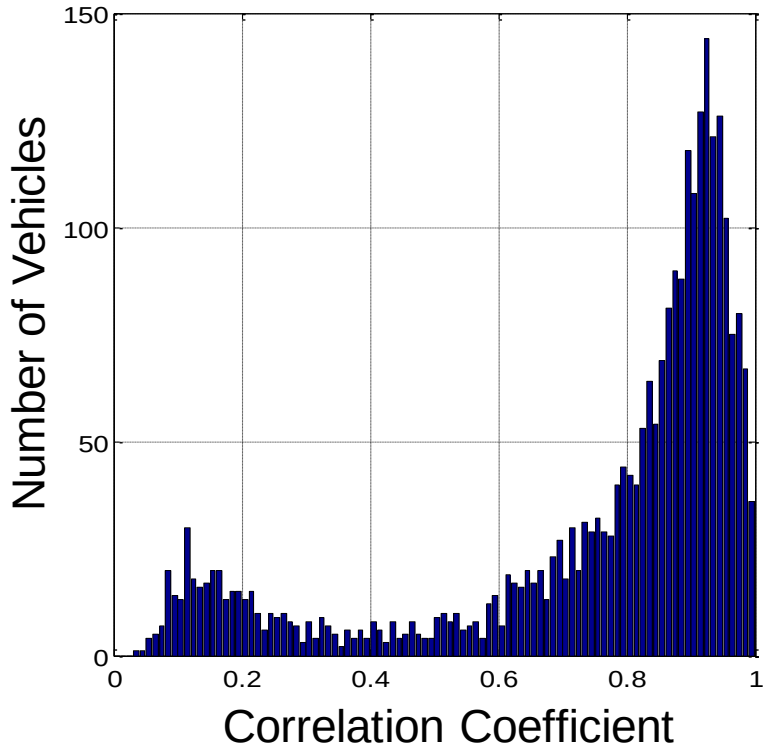
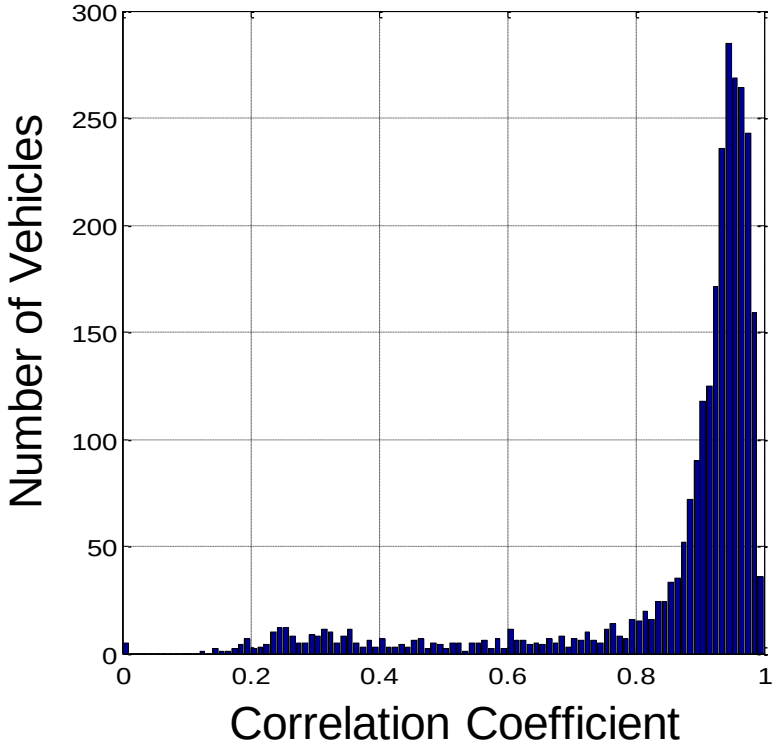


Figure 3.2: Block diagram for the normalized signal generator.

The first three steps are the same as the travel time estimation for a single speed trap as described in Section 2.2. This travel time estimate is then used to normalize the lead and lag signatures. The last step is normalization of the signatures for both speed and energy. The signatures are normalized to a speed of 30 mph with a sampling rate of 5 milliseconds by resampling the data points. This means that each data point in the normalized signature corresponds to approximately 0.22 feet of the vehicle.



a) Correlation coefficient without speed normalization



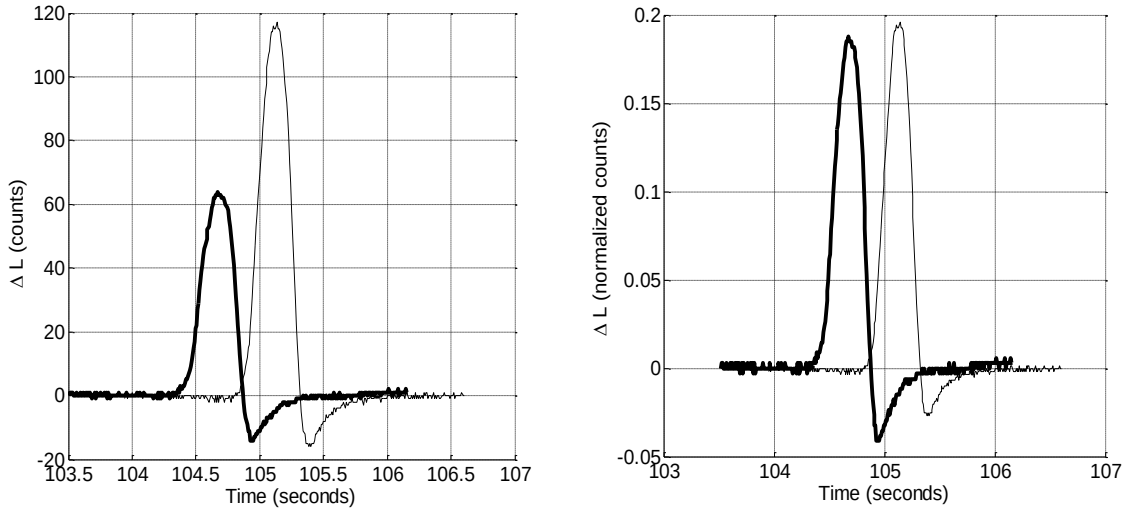
b) Correlation coefficient with speed normalization

Figure 3.3: Effect of speed normalization on correlation coefficients.

Speed normalization is important for increasing the probability of matching an upstream signature with the correct downstream signature. Speed normalization increases the correlation-coefficient for signatures belonging to the same vehicle. This increase in similarity between a correct pairing of upstream and downstream signatures gives a higher probability that the correct pair will be chosen by the matching algorithm. Figure 3.3 shows a comparison of the correlation coefficients of upstream to downstream correct matches with and without speed normalization. The histogram in Figure 3.3a (no speed normalization) has a large number of correlation coefficients less than 0.2 and many more between 0.8 and 0.9 with only 36.8% above 0.9. After the speed normalization, 60.1% of the correlation coefficients are above 0.9 as seen in Figure 3.3b. After the speed normalization, the signatures are then energy normalized so that if a speed normalized signature is x, then the speed and energy normalized signature is:

$$x_{norm}[n] = \frac{x[n]}{\sum_k x^2[k]} \tag{6}$$

This normalization is appropriate because the amplitude of the signature can vary for reasons that are not due to the vehicle. The most common reasons are variability in sensor installation depth and orientation [42]. Figure 3.4a shows the lead and lag signatures from a speed trap before this amplitude normalization. One method of amplitude normalization is to set the maximum value of the signature to one and the minimum value to zero. This method is not robust because only two points determine the scaling factor applied to the signature. Instead the signal's energy is normalized to one. In signal processing, the sum of the squared values of all points in a signal is defined as the signal's energy. Figure 3.4b shows the signatures after energy normalization.



a) Before Energy Normalization b) After Energy Normalization

Figure 3.4: Removing differences in sensitivity by energy normalization.

3.2.2 Similarity Processor

As shown in Figure 3.1, the similarity processor uses the output of the normalized signal generator. The normalized signature pairs are used to identify which upstream and downstream signature pairs are likely to belong to the same vehicle. Each downstream signature is compared to a list of upstream signatures. This list is comprised of all vehicles that would correspond to a travel time within a reasonable range. This travel time range is usually based on the distance between the sensors and possible speeds of vehicles travelling between them. This time window is called the feasibility window. The example in Figure 3.5 shows that the upstream signatures u_3 through u_5 are in the feasibility window of downstream signature d_5. The correlation coefficient between each of the feasible upstream signatures and the downstream signature is computed and the pair with the highest correlation coefficient is chosen as shown in Figure 3.6. The time stamps from the upstream and downstream pairs are subtracted to give feasible travel times that are tagged with their correlation coefficient to designate the likelihood of a given match and associated travel time.

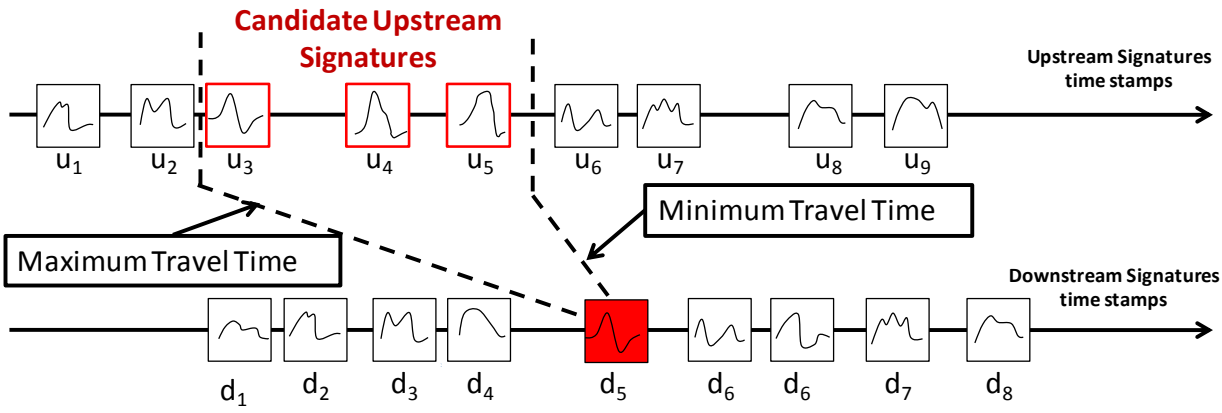


Figure 3.5: Downstream signature and candidate upstream signatures.

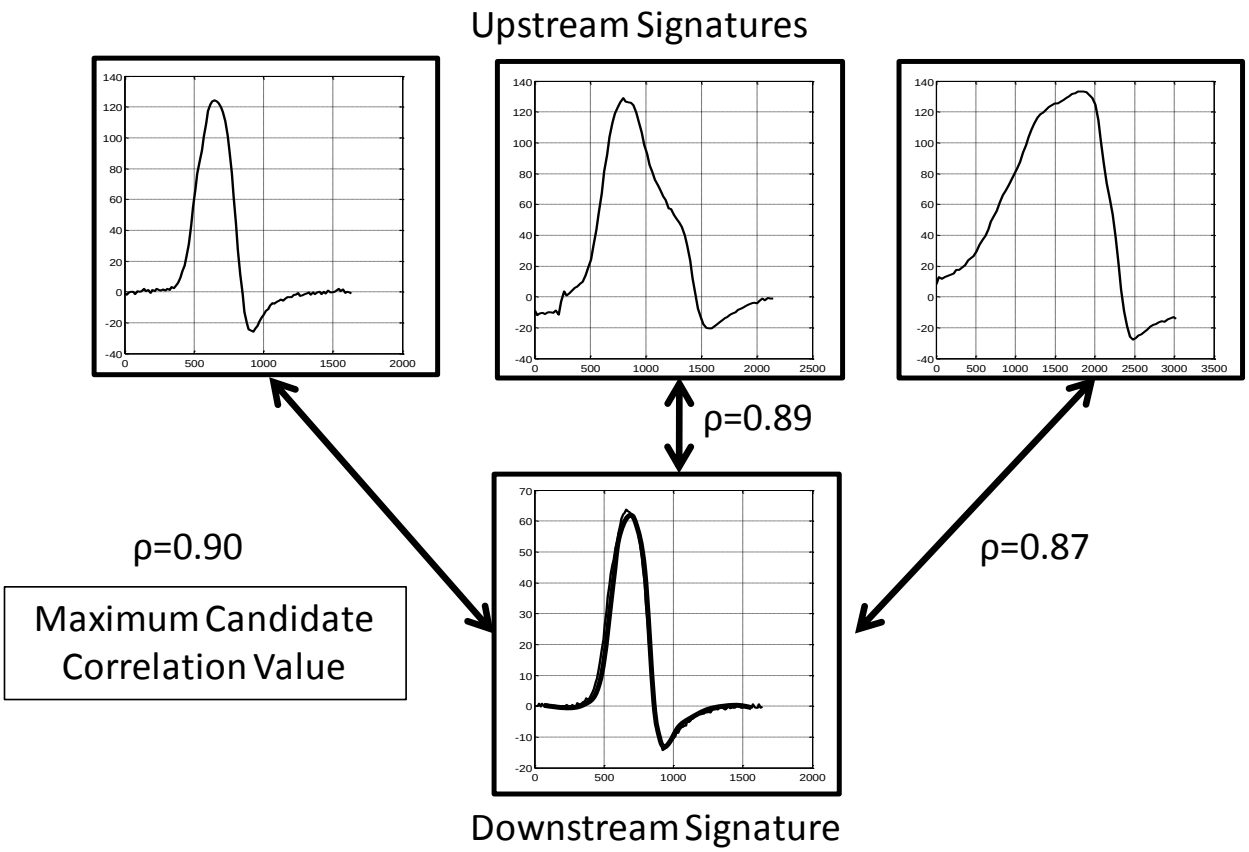


Figure 3.6: The upstream signature with the highest correlation coefficient, when correlated with a given downstream signature, will be selected.

3.2.3 Travel Time Pairing

The set of feasible matches is then evaluated to generate travel times. For each downstream signature, the upstream signature with the highest correlation coefficient from the list of feasible upstream matches is chosen as the correct match. A travel time is generated from each match. This matching algorithm identifies a single travel time is associated with each detection at the downstream detector. The only exception is that no travel time is reported if there are no vehicles in the feasibility window. This window is designed to be large enough that this is very unlikely.

It is possible that two downstream vehicles are matched with the same upstream vehicle. This would not be good if the only (or most important) goal were re-identification. Here, however, the main goal is travel time estimation, which as will be shown, is relatively insensitive to matching errors. This is because the travel time distribution due to correctly matched vehicles can be statistically separated from the travel time distribution due to mismatched vehicles using a k-means clustering algorithm which will be discussed in Section 3.2.4. Situations where the correctly matched and incorrectly matched histograms are not as well separated may require a more complicated statistical analysis tool or higher matching percentages. An algorithm could be implemented for the travel time pairing to require upstream vehicles to be matched with at

most one downstream vehicle. Also, the order of the vehicles could be used with a model of the probability of vehicles switching order to significantly reduce matching error.

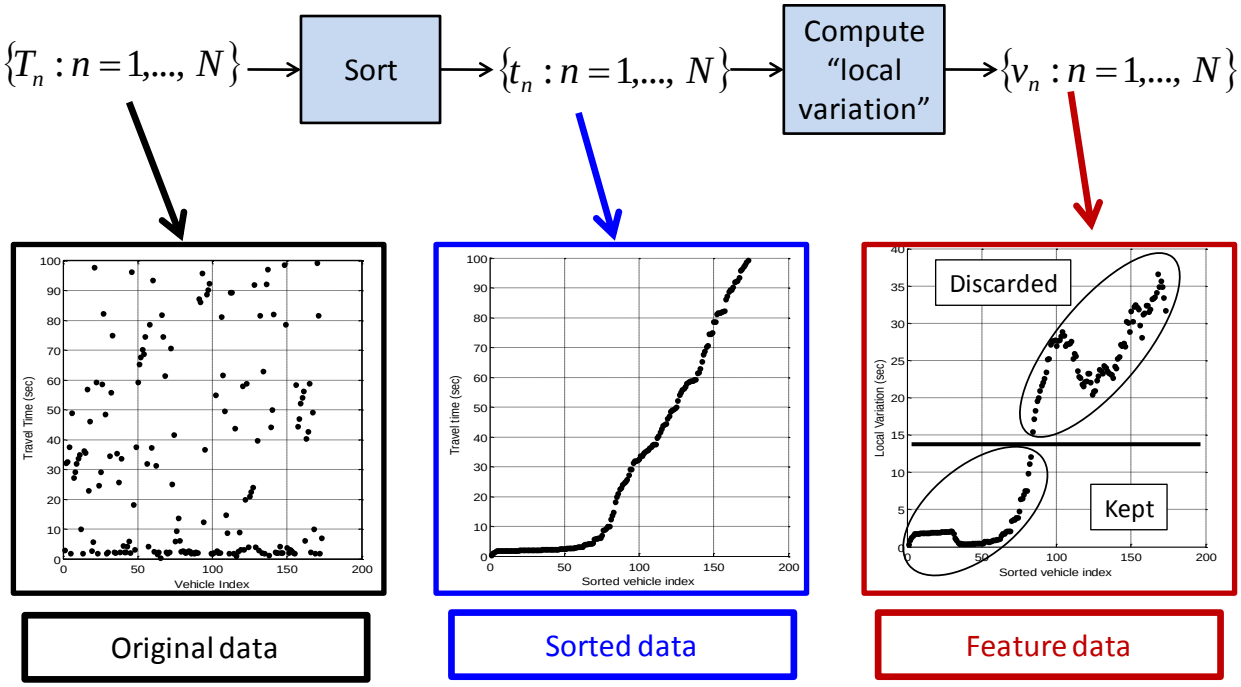
The implemented algorithm does not include any of these more complicated models for two reasons. One reason is that one goal of this project is to assess the quality of travel time estimation based on only matching. Using more complicated models of traffic flow could improve performance, but may reflect the quality of the traffic flow model and not the matching algorithm itself. Also, this simple algorithm can be directly applied at any upstream and downstream location without any training or modeling. A more specialized algorithm and more complicated models can work well for the location where they are designed, but are not generally transferable with the same quality to other locations.

3.2.4 Statistical Analysis

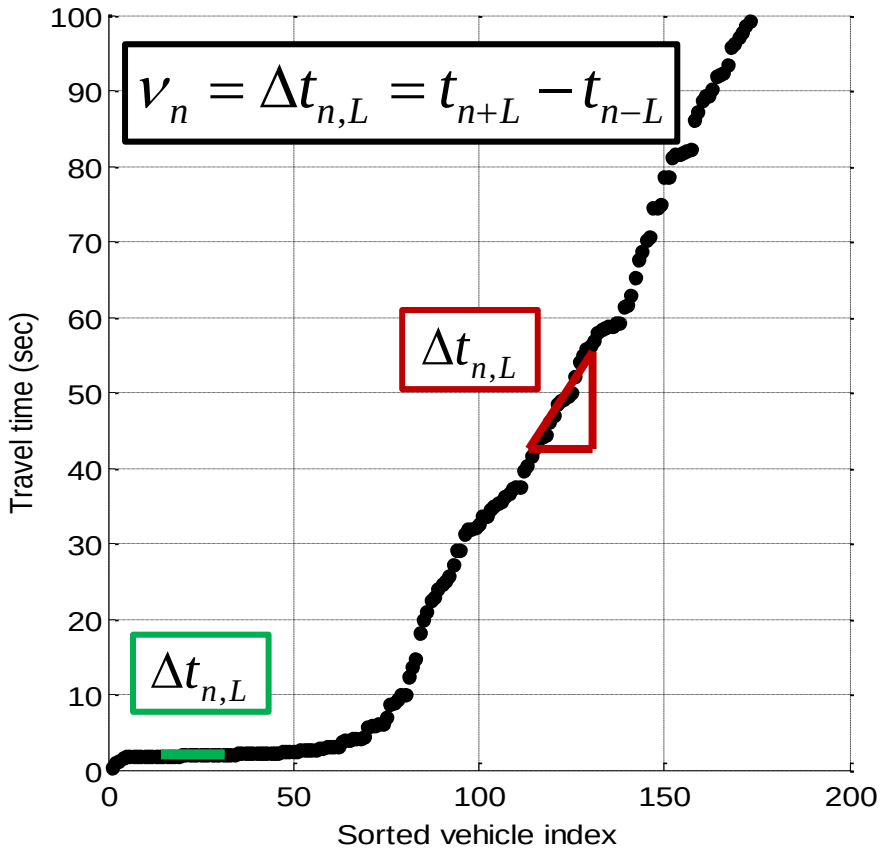
The statistical analysis part of the algorithm helps to remove erroneous travel time matches. The estimated travel time histogram is a composite of both a subset of the true travel time histogram and random travel times from erroneous matches. The random error histogram should tend to be uniform within the feasibility window since the mismatched vehicle is equally likely to be anywhere. The true travel times tend to be more Gaussian or at least clustered around a mean. This property that the correct travel times are grouped and the incorrect travel times are spread more evenly allows the two groups to be statistically separated. Figure 3.7 depicts this process.

Figure 3.7a shows the original travel times plotted in chronological order on the left. The middle figure shows the travel times sorted by travel time where the fastest vehicles are first and the slowest vehicles are last. The slope of this line at each point is estimated by the local variation of a group of points as shown in Figure 3.7b, $\upsilon_n = \Delta t_n = t_{n+L} - t_{n-L}$ where L is the size of the window over which the slope is estimated.

The feature data for each point, shown in Figure 3.7a, is then grouped into two clusters using k-means clustering. K-means clustering is an algorithm that groups the points so that the sum of the squared error between each point and the mean of its group is minimized [53]. The feature data plot, v_n, in Figure 3.7a shows that the points in the lower cluster correspond to speed common speed estimates and are therefore kept. Those estimates that fall in the upper cluster correspond to less frequent estimates that are more likely to be erroneous. The points in the other cluster are removed because they are unlikely to be true travel times.



a) K-means algorithm steps



b) Local variation

Figure 3.7: Local variation K-means clustering method

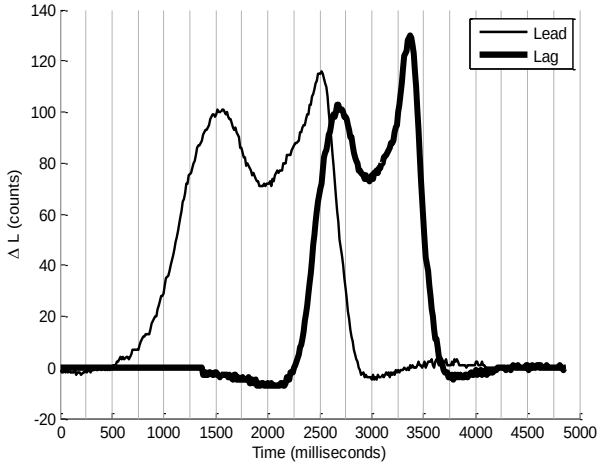
4 Acceleration/Deceleration Compensation

The signatures collected from electromagnetic sensors are collected at a rate that is a function of the detector card's settings and physical characteristics of each installation. For any given data collection, these samples are equally spaced in time at about 100 Hz. If a vehicle travels a constant speed over the two sensors, then a vehicle's spatially indexed signature can be generated by scaling the time axis by the speed of the vehicle. The speed of a vehicle can be calculated by finding the travel time that maximizes the cross-correlation function [50] of the signatures collected by the lead and lag sensors [41]. This speed compensation allows vehicles to be matched that travel a different speed over the upstream detector than over the downstream detector. Because this speed estimation is based on the cross-correlation function, its complexity is low and can be used for real time applications.

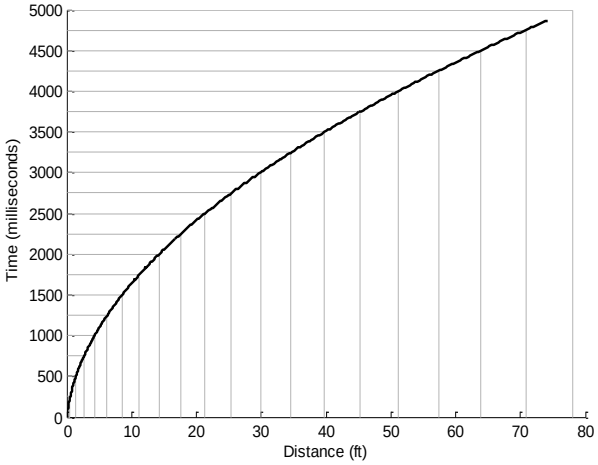
When the sensors are on arterials, it is probable that the vehicles will change speed not only between the upstream and downstream data collection locations, but also while they are traversing each sensor pair. When there is significant acceleration/deceleration, the speed compensation model needs to be augmented with additional information to generate improved spatially indexed signatures. To do this exactly, the speed of the vehicle at every point in time needs to be known to uniquely identify the non-linear scaling of the time axis $x = f(t)$. To reduce the search space, $f(t)$ is restricted to functions representing vehicles that maintain a constant acceleration/deceleration. These functions are of the form: $f(t) = \frac{1}{2}at^2 + v_0 t$ where a is the vehicle's acceleration/deceleration and v_0 is the velocity of the vehicle at the beginning of the signature.

An example pair of lead and lag signatures that have been distorted by vehicular acceleration/deceleration are shown in Figure 4.1a. The acceleration/deceleration distortion is evidenced by the fact that the lead signature is longer than that of the lag signature. A graph showing the relationship between the vehicle's position and time is shown in Figure 4.1b. The uniform horizontal lines correspond to the uniform vertical lines in Figure 4.1a, which represent the equally spaced samples. The vertical lines in Figure 4.1b represent how the axis must be stretched to compensate for the acceleration/deceleration distortion. These vertical lines and the corresponding spatially indexed lead and lag signatures are shown in Figure 4.1c. Figure 4.1d shows the same spatially indexed lead and lag signatures except that the lead signature has been shifted 16 ft to the right (the distance between the lead and lag sensors). This shows that the discrepancies between signatures from the lead and lag sensors have been removed by the acceleration/deceleration compensation.

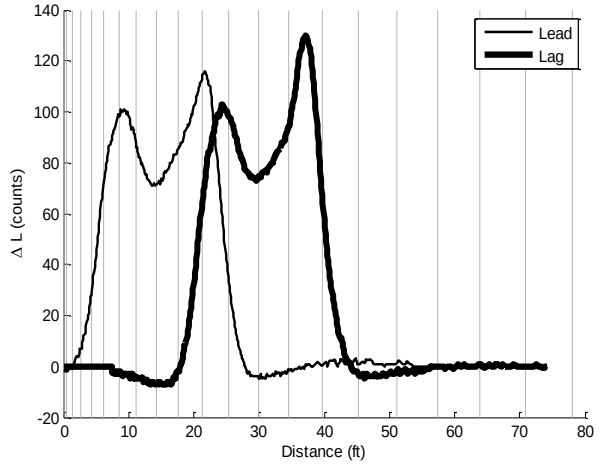
Figure 4.1b represents the relationship between time, t, and location, x, for a given acceleration/deceleration, a and initial velocity, v_0. For each combination of acceleration/deceleration and initial velocity, a correlation coefficient can be calculated. The correlation coefficient is maximized by the correct acceleration/deceleration and initial velocity. Figure 4.2 shows a contour plot of the correlation coefficients for each combination of acceleration/deceleration and initial velocity. The optimal choice for this example is an initial velocity of 0.84 miles per hour and 0.18 g's. This slow initial velocity is due to the vehicle being stopped just before it enters the detection zone and accelerating while passing over the sensor.



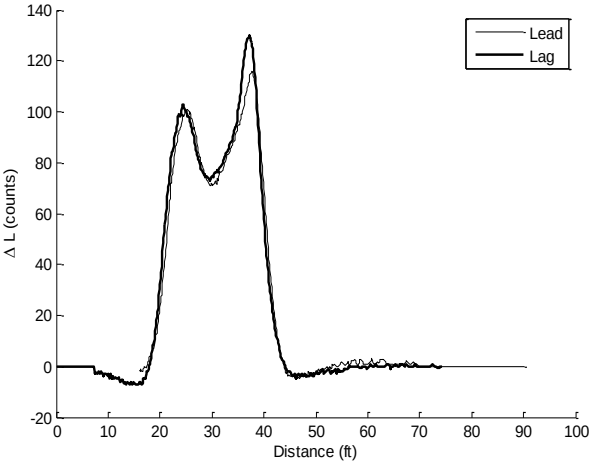
a) Example Lead and Lag Signature



b) Quadratic relationship between time and distance



c) Spatially indexed Lead and Lag Signatures



d) Spatially indexed Lead and Lag Signatures with 16 ft shift

Figure 4.1 Temporally and Spatially indexed example signatures.

There are several interesting characteristics in Figure 4.2. The first is that there are no data points below a diagonal line extending from the origin to a point at about 40 mph initial velocity and -0.2 g's acceleration/deceleration. This is because of the constraint that the vehicle is always travelling forward. For each initial velocity, there is a minimum acceleration/deceleration that meets this constraint. Another characteristic is the convex shape of the majority of the contour. This means that if noise alters the location of the maximum correlation coefficient, it is likely that the new maximum will have a very similar initial velocity and acceleration/deceleration to the true maximum. This also means that various convex optimization algorithms can be applied to find the maximum without performing a global search. The optimization used to find the acceleration/deceleration and initial velocity is described in the following section.

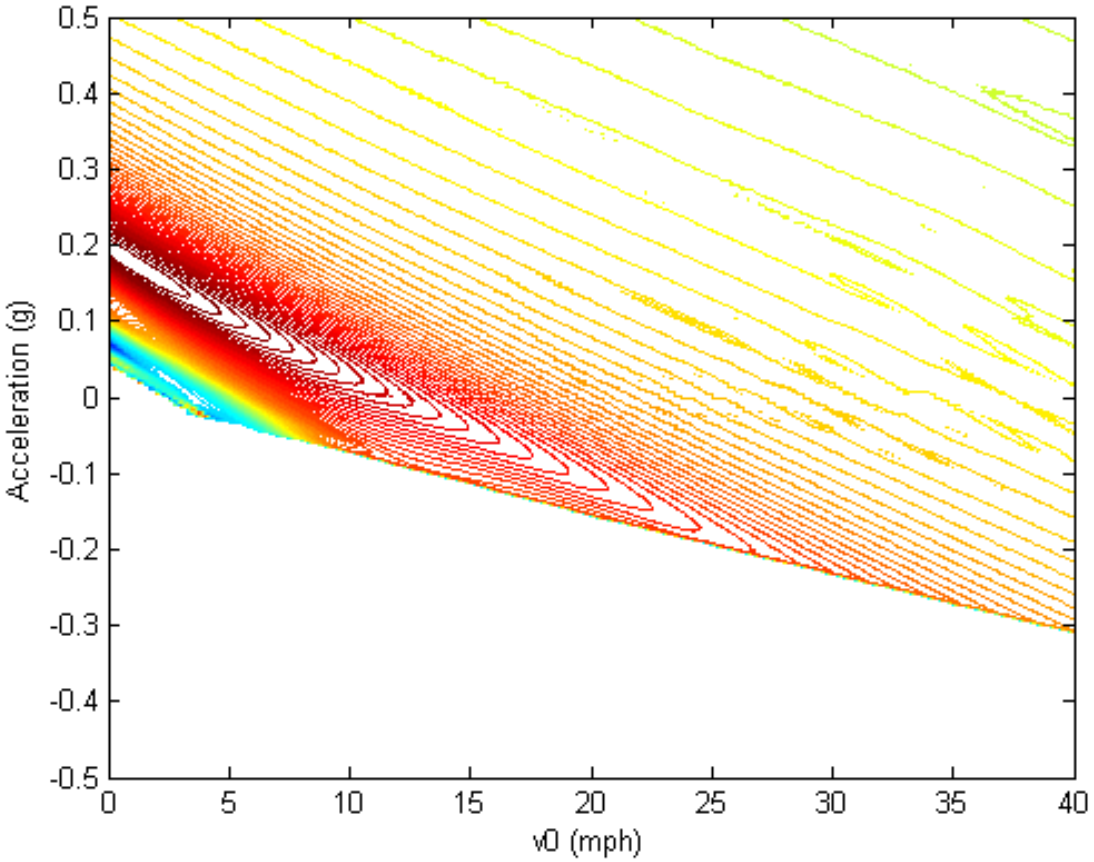


Figure 4.2 Contour plot of correlation coefficient between the lead and lag signatures for a given acceleration/deceleration and initial velocity.

The Nelder-Mead optimization algorithm is a maximization algorithm that is also called the simplex search algorithm. A simplex is a set of $n + 1$ points that enclose a convex region in an n-dimensional space. In a one-dimensional space, a simplex is a line segment. In a two dimensional space, a simplex is a triangle and in a three dimensional space, a simplex is a triangular prism. This idea extends to spaces of any number of dimensions. Since our search space is two dimensional, a triangle is used. First, the general two-dimensional Nelder-Mead maximization algorithm is explained. Its application to this specific search is then discussed.

The Nelder-Mead optimization algorithm is an iterative algorithm that begins with three points. The goal of each iteration is to replace the worst point by a better point. This continues until all three points converge on a location within a specified tolerance.

For each iteration the value of the function at each point is evaluated and the points are ordered such that the best point is p_1, the second best point is p_2, and the last (and worst) point is called p_3. A new point p_0 is then defined as the midpoint of a line segment between p_1 and p_2. This point is then used to calculate several other points as shown in the following equations:

$$p_r = p_0 + \alpha(p_0 - p_3) \quad (7)$$
$$p_e = p_0 + \gamma(p_0 - p_3) \quad (8)$$
$$p_c = p_3 + \rho(p_0 - p_3) \quad (9)$$

The flow chart in Figure 4.3 shows the four possible outcomes of each iteration: reflection, expansion, contraction, and reduction. Figure 4.4a shows the locations of the points that will be discussed. The reflected point, p_r, is the first new point evaluated. If this point is the best point thus far, then the algorithm decides that this is a particularly good direction and evaluates a point further out than the reflected point. This point, p_e, is called the extension point. If the extension point is better than the reflection point, then it replaces the worst point in the current simplex. The next simplex then includes this new set of three points as shown in Figure 4.4b. If the extension point is not better than the reflection point, then the reflection point is used, as shown in Figure 4.4c. If the reflection point is not as good as the best original point, p_1, but is better than the second best point, p_2, then the reflection point is used without evaluating the extension point. If p_r is worse than p_2, then the reflection point is probably not the right direction. Therefore, a point is chosen between p_0 and p_3 called the contraction point, p_c. If p_c is better than p_3 then it replaces p_3, as shown in Figure 4.4d. Otherwise, none of the new points are used and the size of the simplex is decreased by moving p_2 and p_3 toward $p_1$as shown in Figure 4.4e. This is called reduction.

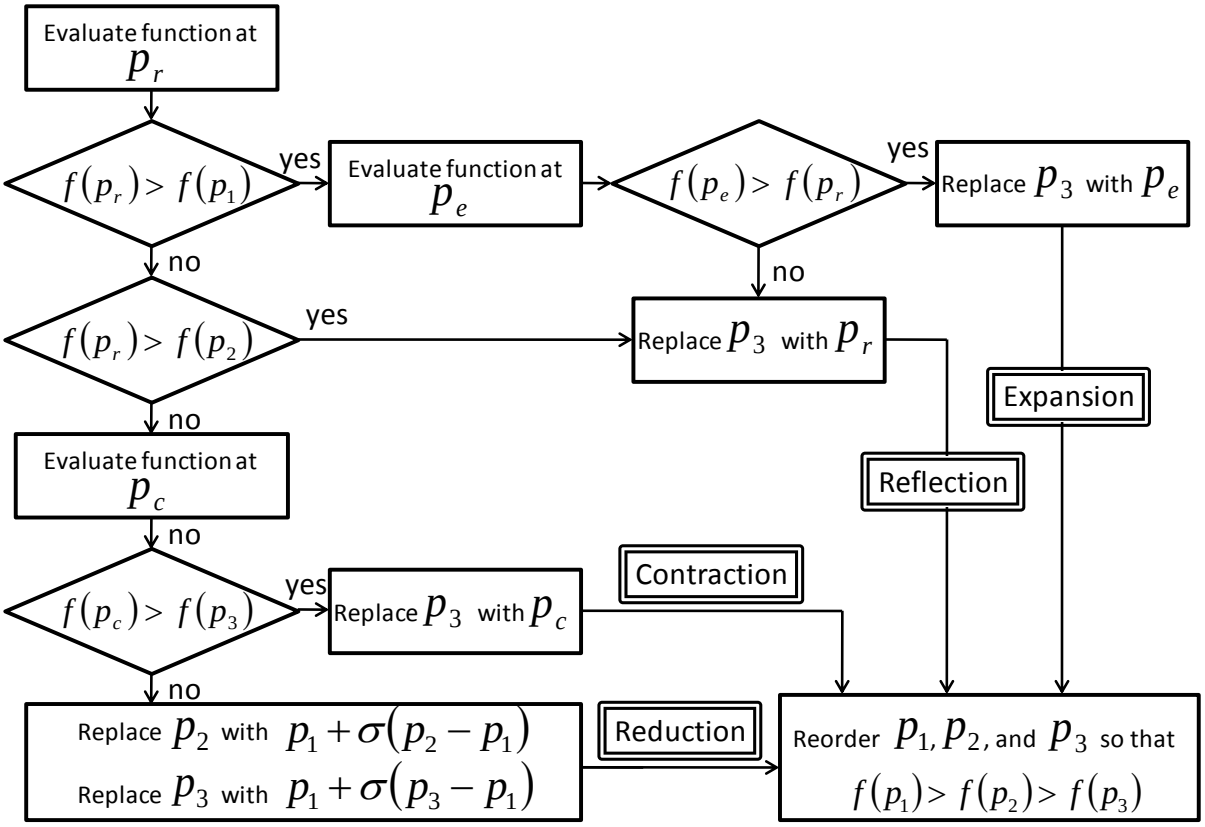
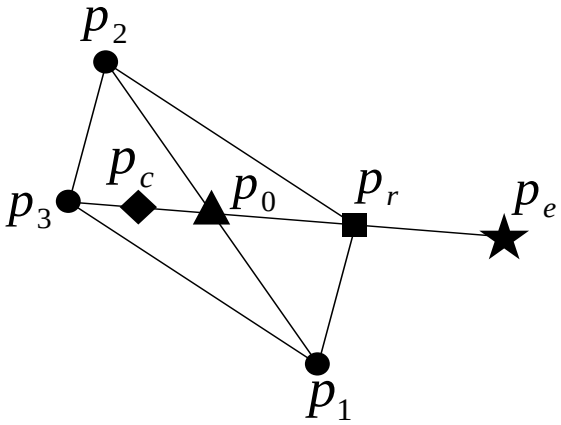
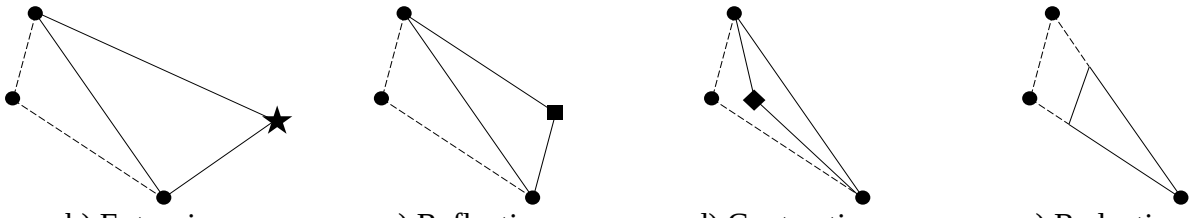


Figure 4.3 Nelder-Mead flow chart.



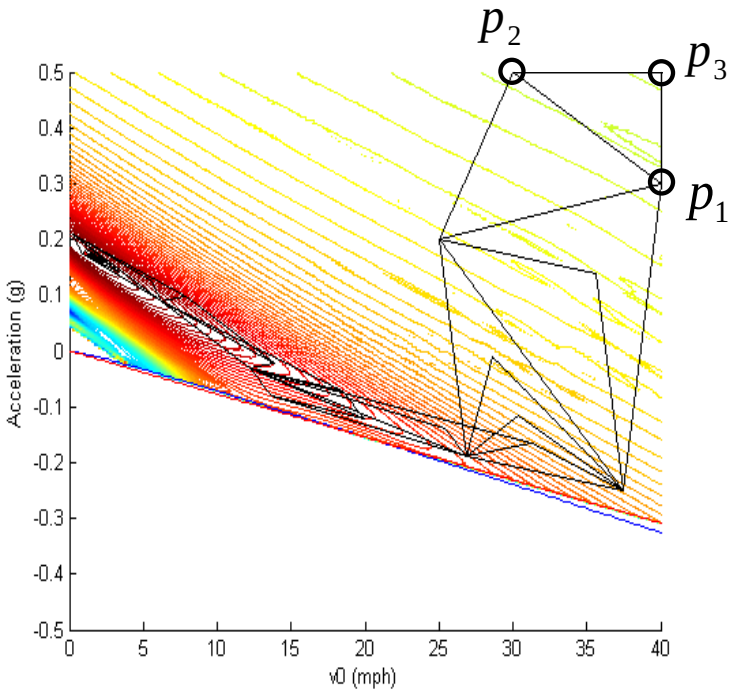
a) Definitions of points



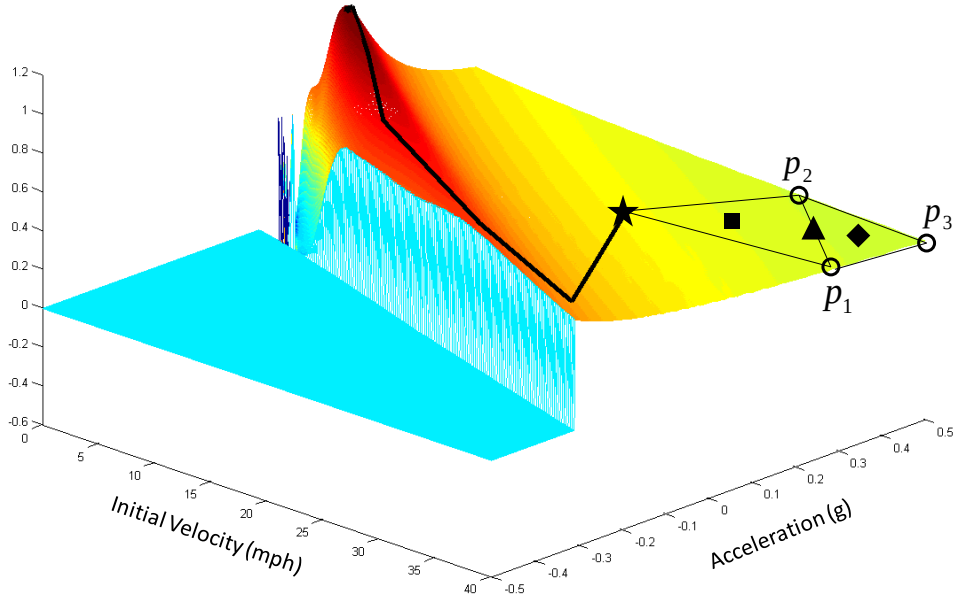
b) Extension c) Reflection d) Contraction e) Reduction

Figure 4.4 Nelder-Mead example of simplex search options.

To find the acceleration/deceleration and velocity, the function to be maximized is the correlation coefficient of the lead and lag signatures. An example contour plot is shown in Figure 4.5a. The initialization points are shown along with all steps of the algorithm. Figure 4.5b shows the same optimization example as a line overlaid on a three dimensional surface. The line connects the best point at each step of the iteration. The calculation of the first point is shown in detail. In this case, the extension point is chosen.



a) Example of all simplex triangles chosen during Nelder-Mead maximization on a contour plot with initialization points designated



b) Three dimensional plot of correlation coefficient for each acceleration/deceleration value and initial velocity. The line represents the best point of each simplex. The initialization points and the points calculated in the first step are also shown.

Figure 4.5 Example of Nelder-Mead maximization.

5 Data Collection

The travel time estimation algorithm can be applied to any two pairs of speed traps. Both test sites in this study (Lafayette and Noblesville) are at intersections that have a speed trap at the stop bar and a second speed trap in advance of the intersection on the Northbound approach. The advanced speed trap is therefore the upstream data collection site and the stop bar speed trap is the downstream data collection site. Data was collected from the Lafayette and Noblesville test sites and stored in a multimedia database that was specifically designed for this purpose. Each data collection site is now described in detail along with a description of the database used to store the data.

5.1 Data Collection Sites

5.1.1 West Lafayette

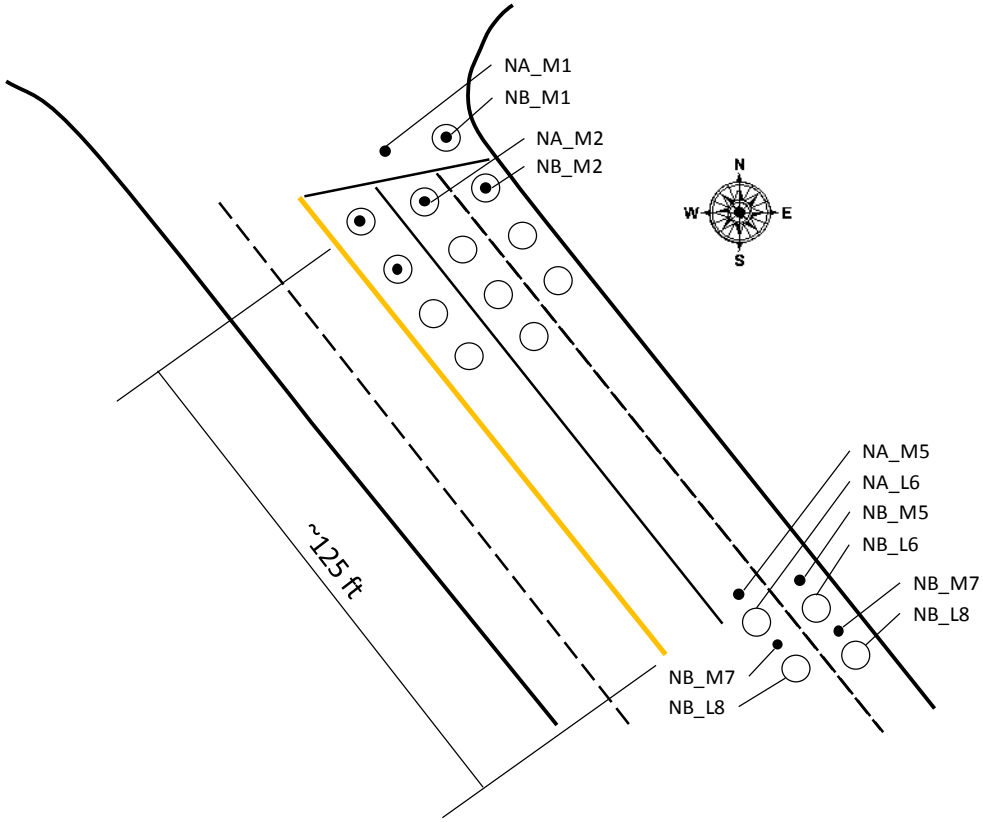
The northbound approach of the Northwestern and Stadium intersection in West Lafayette is shown in Figure 5.1a. The Northbound approach is the chosen approach for this study because it is the only approach at this intersection with advanced detection.

In each of the two Northbound through lanes, data can be collected from

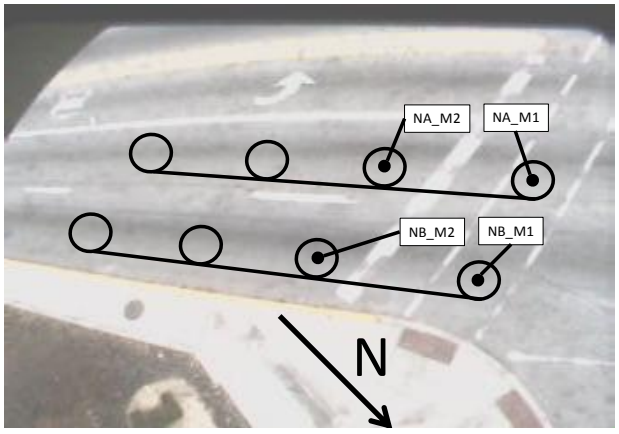
- two stop bar microloop detectors (referenced as NA_M1, NA_M2, NB_M1, and NB_M2 in Figure 5.1a and Figure 5.1b)
- two advanced mircoloop detectors (referenced as NA_M5, NA_M7, NB_M5, and NB_M7 in Figure 5.1a and Figure 5.1c),
- two inductive loop detectors (referenced as NA_L6, NA_L8, NB_L6, and NB_L8 in Figure 5.1a and Figure 5.1c)

for a total of 8 microloops and 4 inductive loops. Data can be collected from all 12 sensors concurrently. The inductive loops at the stop bar are not suitable for this study because they are connected in series to one detector channel instead of each being connected to an individual detector channel.

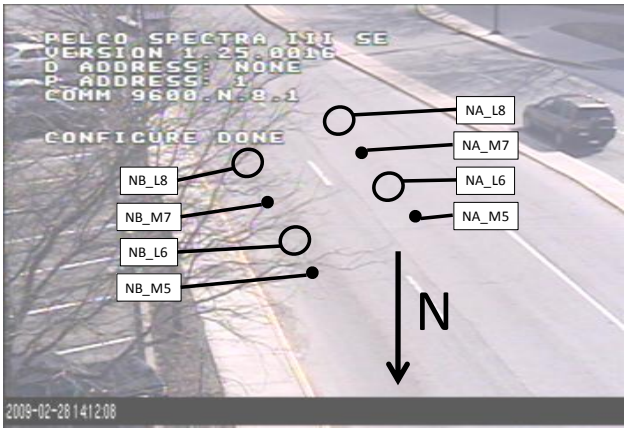
The distance between the advanced detectors and the detectors at the stop bar is about 125 ft. This is sufficient distance for velocity change between sensors and some lane change. It is not a large enough distance for significant re-ordering of vehicles.



a) Northbound Approach



b) Stopbar carmera view



c) Advanced Camera View

Figure 5.1: Northwestern and Stadium intersection.

5.1.2 Noblesville

The second data collection site is in Noblesville, IN at the intersection of State Highway 32 and 37. The northbound approach was selected because it has advanced detection. A diagram of the northbound approach is shown in Figure 5.7a. The sensors at this intersection also include

- two stop bar microloops (referenced as NA_M1, NA_M2, NB_M1, and NB_M2 in Figure 5.7a and Figure 5.7b),
- two advanced microloops (referenced as NA_M5, NA_M7, NB_M5, NB_M7 in Figure 5.7a and Figure 5.7c), and
- two advanced inductive loops in each lane (referenced as NA_L6, NA_L8, NB_L6, NB_L8 in Figure 5.7a and Figure 5.7c).

Similar to the West Lafayette site, the stop bar inductive loops are not suitable for this study because they are connected in series to one detector channel instead of each being connected to an individual detector channel. The most obvious advantage of the Noblesville site over the West Lafayette site is that the distance between the stop-bar detectors and the advanced detectors is about 400 ft instead of about 125 ft at the West Lafayette site which gives slightly longer travel times. While this is the most obvious difference between the intersections, it does not affect the performance of the algorithm. The performance of the matching algorithm is based on the size of the feasibility window, which is 100 seconds at both data collection sites. The Noblesville site does however have other more subtle differences that make it more challenging. At the Noblesville intersection there are a high number of vehicles accelerating and changing lanes while over the advanced loops.

5.1.3 Highway

A highway data collection site was used for some preliminary results. This data was collected from speed traps on I-70 near Indianapolis. The upstream speed trap is located at mile marker 66.6 and the downstream speed trap was located at mile marker 67.3. Data was collected for a duration of one hour and fifteen minutes from the center lane.

5.2 Crosstalk

The quality of the signatures collected is important to the matching performance. Care was taken to minimize noise in the system. Several of the channels were experiencing crosstalk that was causing a hundred detections per second on some channel. Noise and crosstalk removal is an important part of any signature matching system. For this reason, characteristics of crosstalk and an automatic crosstalk detection algorithm are presented here.

5.2.1 Characterization of Signatures with Crosstalk in the Time Domain

The time varying waveform corresponding to the change in the oscillating frequency of the sensor circuit due to a passing vehicle was termed as a signature in the introduction. Ideally the detector's oscillation frequency is influenced only by the presence of a vehicle over the sensor. But in some situations, the signals of independent detector circuits may be coupled to each other causing a spurious change in the oscillation frequency which we will call crosstalk. This is illustrated in Figure 5.2 which shows a 2.5 hour signature train containing varying levels of crosstalk. Figure 5.2(a-c) shows three 10-second segments extracted from the longer train of (d). In Figure 5.2a, no crosstalk is evident while (b) and (c) show increasing levels of crosstalk interference. The crosstalk in this example manifests itself as random noise added to the vehicle induced signatures.

Figure 5.3 shows another form of crosstalk, where the crosstalk manifests as sinusoidal harmonics in the signal. This is similar to the beating effect of two frequencies observed from two instruments at very close frequencies (of pitches). Both the more random and beating types of crosstalk include high frequency noise. This can be separated from the signature influenced by the vehicles since the effect of vehicles on the sensor is limited by the speed of the vehicle and the size of the metallic components in the vehicle that affect the sensor. These factors generate waveforms that tend to stay below 10 Hz. Therefore, the energy in the signatures at frequencies over 10 Hz must be primarily due to crosstalk interference.

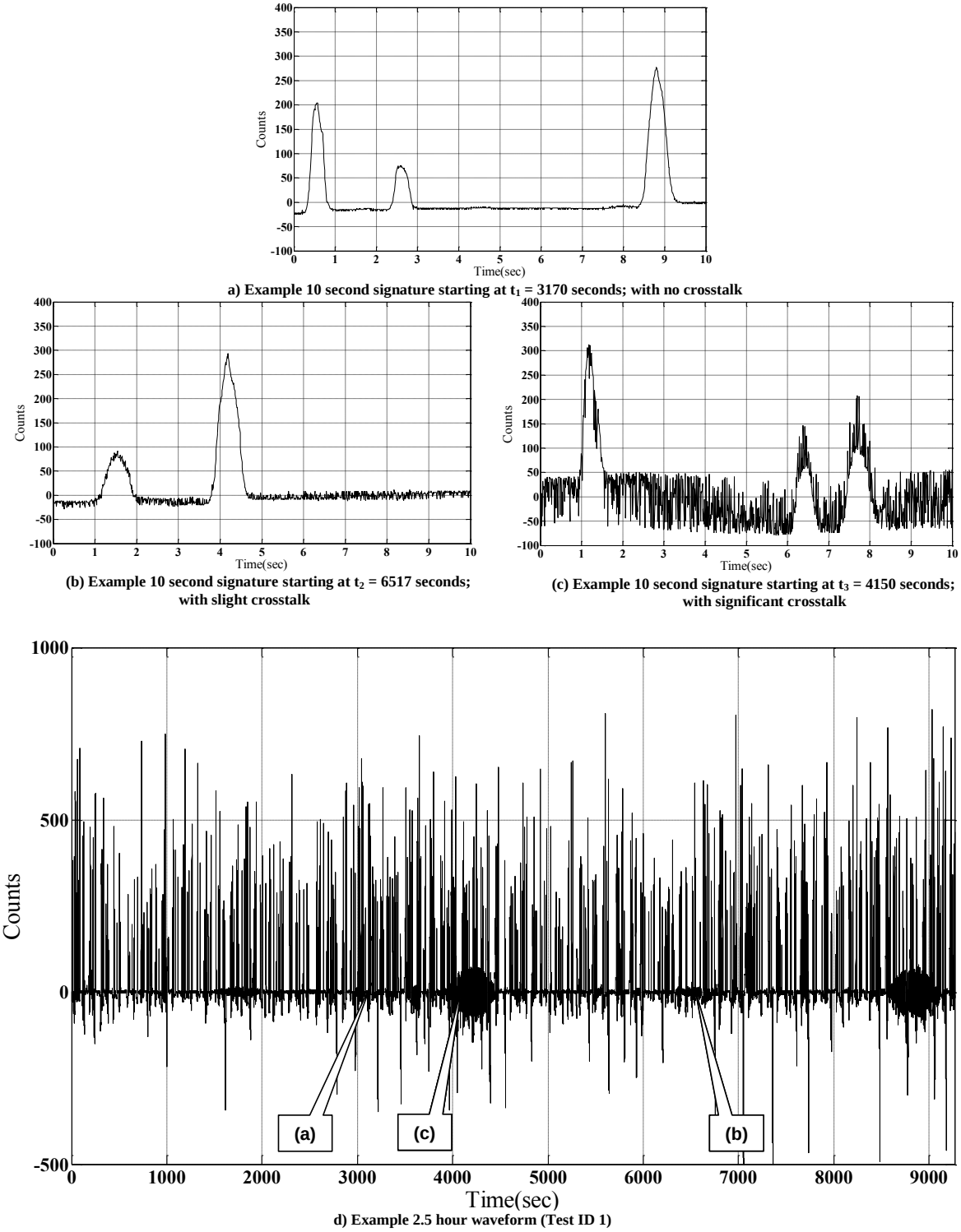
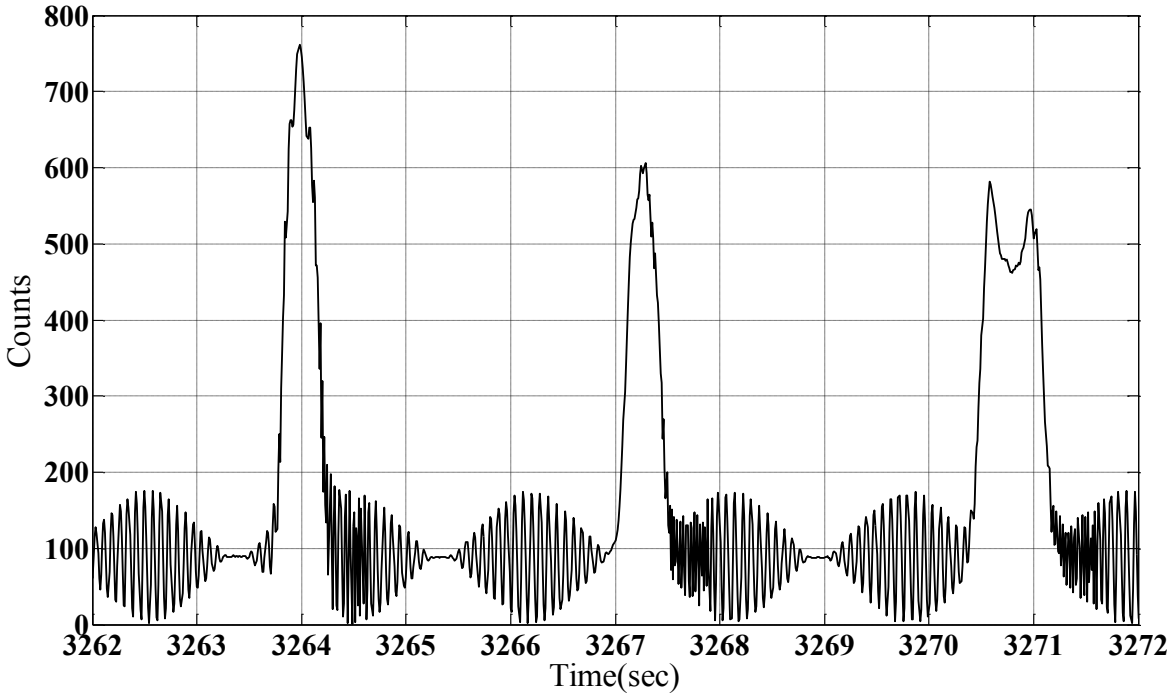
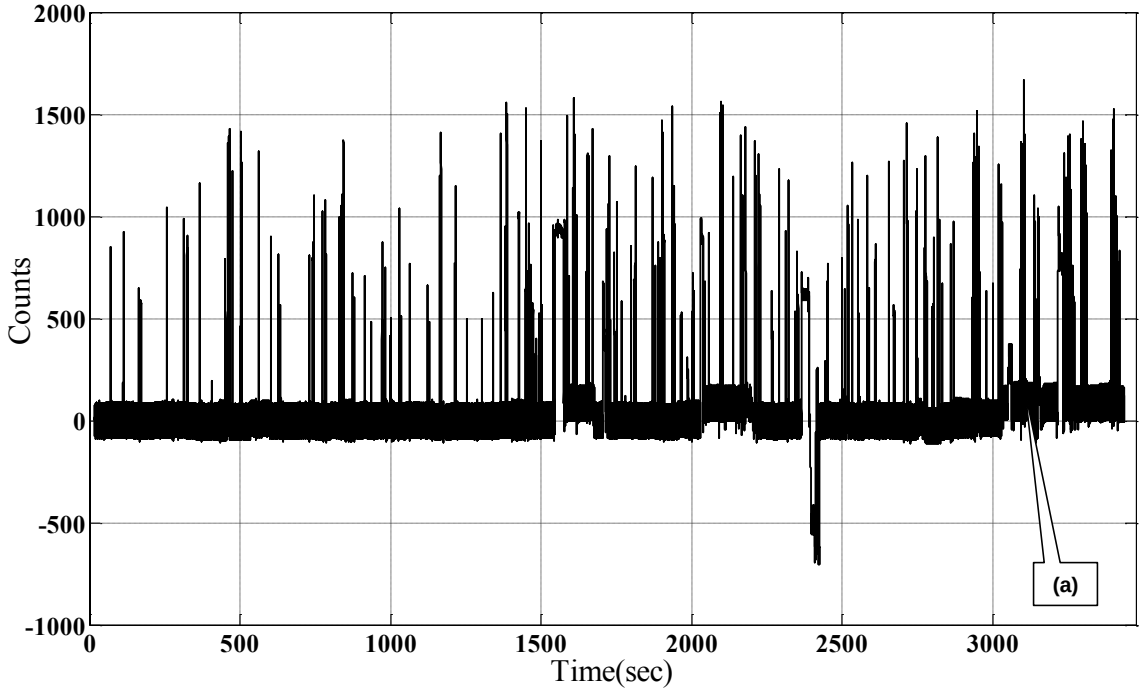


Figure 5.2: Example signatures from showing varying levels of the effect of crosstalk (Test ID 1)



(a) Example 10 second signature starting at t = 3262 seconds; with substantial amount of crosstalk-caused beats



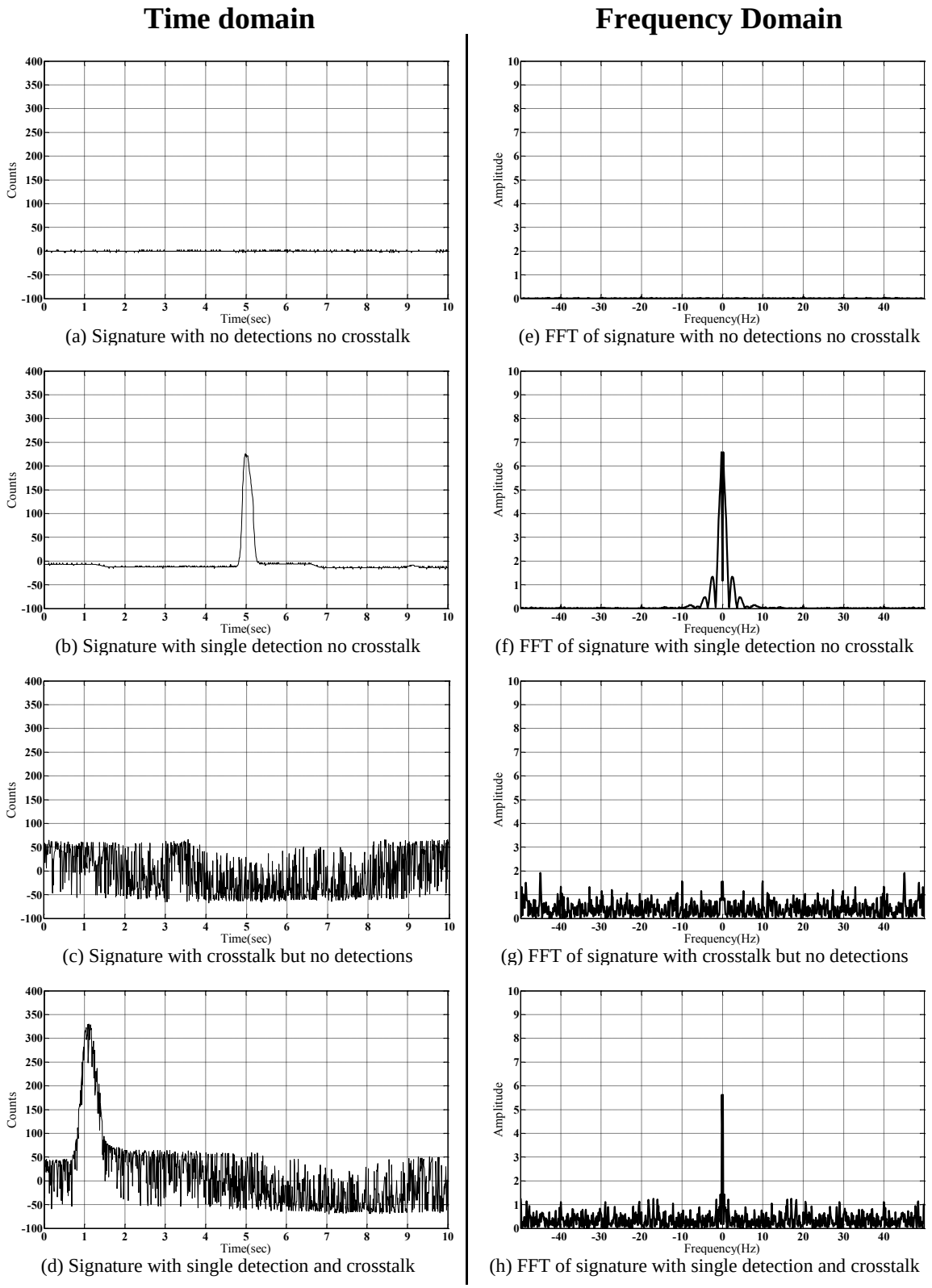
b) Example 57 minute waveform (Test ID 7)

Figure 5.3: Example signature showing harmonic crosstalk beats (Test ID 7)

5.2.2 Characterization of Signatures with Crosstalk in the Frequency Domain

Since crosstalk and vehicular signatures can be separated in frequency, a frequency domain analysis is useful. The Fast Fourier Transform (FFT) is used to determine the contribution of each frequency to the signal. The frequency domain representation of each 1,000 sample block of the signal is then generated. This corresponds to approximately 10 seconds of data. The magnitude of the frequency content of the signal can then be analyzed. Figure 5.4 (a) (b) (c), and (d) are four 10 second signatures and Figure 5.4 (e) (f) (g), and (h) are their respective frequency domain representations as calculated by the FFT.

The top pair of plots shows that if there is no vehicle and no crosstalk that there is nothing in the time domain or frequency domain. The second pair shows the time and frequency domain plots of a signature that is only affected by a vehicle. As discussed in the previous section, the frequency of the vehicle is limited to about 10 Hz. The purely crosstalk signature and frequency representation in the third row show that the crosstalk influenced signature has frequency components in all possible frequencies. Since the time domain signature is sampled, the possible frequencies are from zero Hz to half the sampling frequency. The last signature has crosstalk effects and is also affected by a passing vehicle. The effects are superimposed in the time and frequency domain such that the fourth pair resembles the second and third pair added together. In the following we will develop an algorithm to detect crosstalk.



(a) Signature with no detections no crosstalk

(e) FFT of signature with no detections no crosstalk

(b) Signature with single detection no crosstalk

(f) FFT of signature with single detection no crosstalk

(c) Signature with crosstalk but no detections

(g) FFT of signature with crosstalk but no detections

(d) Signature with single detection and crosstalk

(h) FFT of signature with single detection and crosstalk

Figure 5.4: Short duration (10 second) data sets illustrating difference in spectral content between regions with and without crosstalk (Test ID 1)

5.2.3 Spectral Energy Analysis

Based on the observations presented in the first two sub-sections, a metric to characterize the amount of crosstalk is now proposed. First, the signature is broken into vectors of length N, where N = 1000 samples. Each vector $\vec{x}[n]$ is then processed by an algorithm depicted in the block diagram in Figure 5.5. The algorithm begins by multiplying $\vec{x}[n]$ with a 1000 point Gaussian window $\vec{w}_G[n]$. The Gaussian window is chosen as it has the minimum time-bandwidth product amongst all tapered windows, allowing for maximum resolution in the frequency domain. A rectangular window was not chosen because of its high side lobes in the frequency domain which reduce spectral resolution.

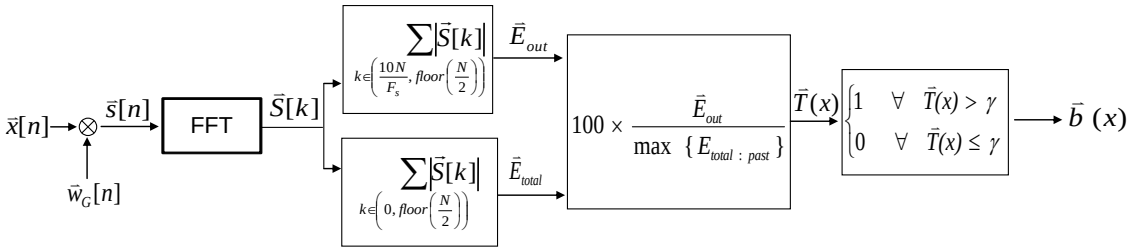


Figure 5.5: Block Diagram of algorithm to provide a metric for out of band energy to identify crosstalk

The resulting windowed data vector $\vec{s}[n]$ is then considered in the Frequency Domain using a Fast Fourier Transform (FFT) to yield the frequency domain signal $\vec{S}[k]$.
Based on the frequencies of signatures shown in Figure 5.4(e) and (f), it may be deduced that detections usually lie in the 0-10 Hz range while crosstalk affects the band outside this range. Hence, the out of band spectral energy is used as a measure for the amount of crosstalk. The 0-10 Hz frequency range corresponds to the discrete range:

$$k \in \left(\frac{10N}{F_s}, \text{floor}\left(\frac{N}{2}\right)\right) \tag{10}$$

where k is the FFT index, $F_s = n_{\text{total}}/T$, is the sampling rate, n_{total} is the total number of samples in the data set and T is the total duration of the data set. The out of band energy for $\vec{x}[n]$ is labeled as $\vec{E}_{\text{out}}$in Figure 5.5, which is found by summing $\vec{S}[k]$ over the range shown in (Equation 1). $\vec{E}_{\text{total}}$ is then found as the total spectral energy from positive frequencies in $\vec{S}[k]$, corresponding to the discrete range:

$$k \in \left(0, \text{floor}\left(\frac{N}{2}\right)\right) \tag{11}$$

$\vec{E}_{\text{out}}$ is then divided by the maximum of $\vec{E}_{\text{total}}$ over the previous M vectors, where M is large enough that $\vec{E}_{\text{total}}$ includes a window with a vehicle and thus characterizes the total possible spectral energy in a signature. This number, M can be set based on traffic conditions. The ratio $/\max\{E_{\text{total:past}}\}$ is expressed as a percentage to make the crosstalk index less dependent upon the installation.

This percentage is labeled $\vec{T}(x)$ in Figure 5.5, and referred to as the crosstalk index throughout the paper. The evaluation of threshold γ and the binary crosstalk indicator $\vec{b}(x)$ is presented in the next subsection.

5.2.4 Determination of Crosstalk Index Threshold

This crosstalk index is evaluated for 156 hours of data collected from 12 sensors at an installation that has shown no signs of crosstalk. A histogram of the points for these crosstalk indices is shown in Figure 5.6. This is a statistically significant data set with which to characterize the algorithm's performance over data with no crosstalk. In the Neyman Pearson framework, this is our null hypothesis. Even though the distribution does not appear to be Gaussian, a Gaussian distribution is assumed to characterize this crosstalk index.

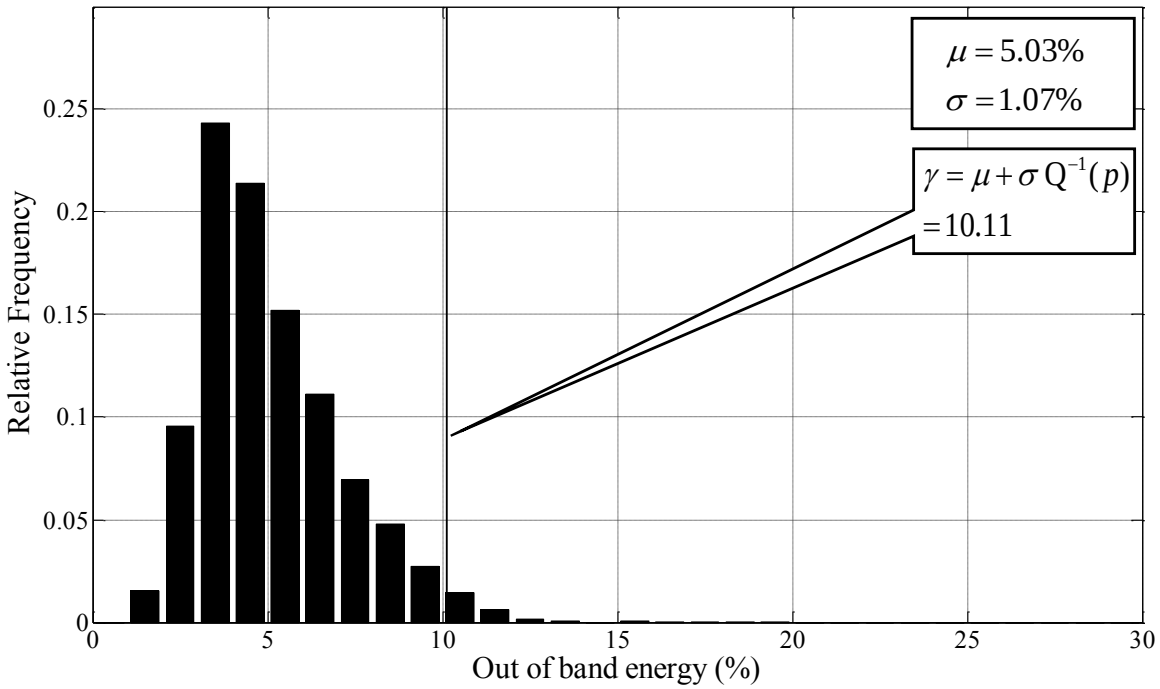


Figure 5.6: Histogram of crosstalk index for 108 sample waveforms (156 hours total) to estimate threshold for detecting crosstalk

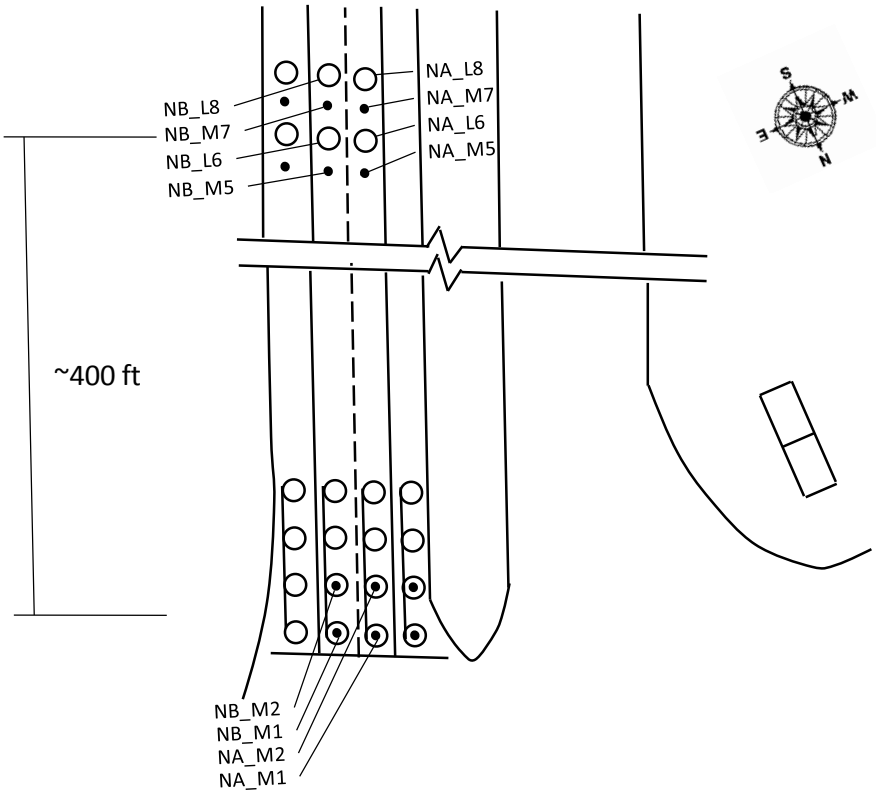
Neyman-Pearson is a theory framework from statistics. The property that we are using for this analysis is that the false alarm rate can be chosen through the threshold with knowledge only of the null hypothesis.

The distribution is found to have a mean of $\mu = 5.03$, and a standarad deviation of $\sigma = 1.07$. By using these values and the Gaussian assumption, we can set a threshold by limiting the probability of Type I error (e.g. detecting crosstalk when it is not present) to .001%. The threshold is found by applying the following formula:

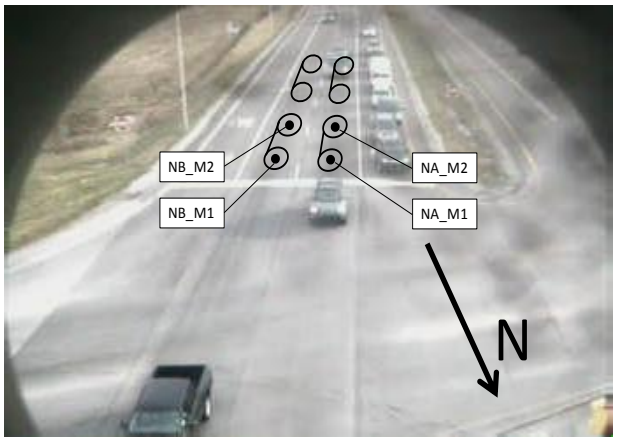
$$\gamma = \mu + \sigma Q^{-1}(p) \quad (12)$$

where p is the probability of Type 1 error as defined above and *Q(x)* is the complementary CDF of a Gaussian random variable *x* [50]. A threshold of $\gamma = 10.11\%$ is thus obtained, which is shown as the solid vertical line in Figure 5.6. This means that any crosstalk index above 10.11% will be flagged as an indication of crosstalk.

Finally, the crosstalk index is passed through a threshold function based on the estimated threshold γ, that produces a binary crosstalk indicator $\vec{b}(x)$. The binary indicator turns ON when the crosstalk index $\vec{T}(x)$ crosses the calculated threshold $\gamma = 10.11\%$, and stays OFF otherwise. This indicates that crosstalk is present when $\vec{T}(x)$ is larger than γ and absent otherwise.



a) Northbound Approach



b) Stopbar camara view

c) Advance camera view

Figure 5.7: Noblesville Intersection

6 Results and Analysis

This section shows the results of the travel time algorithm described in Chapter 2 and Chapter 3. The data collection sites are described in Chapter 4. First, the percentage of correctly matched vehicles is discussed for each data set. Next, the true travel time histogram for each data collection set is compared with the estimated travel time histogram before and after the k-means filtering. Specific examples are then presented to explain causes of matching errors.

6.1 Matching Percentage

The matching percentage for each data set is defined as:

$$\text{matching percentage} = \frac{\text{correctly matched vehilces}}{\text{ground truthed vehicles}}$$

The overall matching percentages for each lane are shown in Table 6-1. The Northwestern and Stadium data set was collected over many smaller data sets. The matching percentage is broken into individual data sets for this data collection site as shown in Table 6-2. These matching percentages range from 38.5% to 78.4%. These matching percentages will be shown to be sufficient for travel time estimation and that these percentages can be improved by applying additional filters.

Table 6-1: Matching Performance summary over all 12 days

Site	Lane	Counts				Ground Truth Matches	Algorithm Correct Matches	
		Upstream		Downstream				
		Lead	Lag	Lead	Lag		Number	Percent
Lafayette	A	3251	3227	3228	3160	2671	1286	48.1
Lafayette	B	3746	3798	3721	3844	2750	1737	63.2
Noblesville	A	1624	1632	1797	1865	1084	550	50.7
Noblesville	B	1442	1447	1651	1638	1228	722	62.9

Table 6-2: Northwestern and Stadium Matching Results

Date	Lane	Counts				Ground Truth Matches	Correct Matches	
		Upstream		Downstream				
		Lead	Lag	Lead	Lag		Number	Percent
5/25/2010	A	179	176	178	173	144	82	56.9
5/25/2010	B	212	213	220	216	203	121	59.6
5/26/2010	A	290	288	287	279	210	102	48.8
5/26/2010	B	308	315	324	323	269	156	58
5/27/2010	A	219	219	217	211	178	81	45.5
5/27/2010	B	248	251	209	243	150	87	58
5/28/2010	A	219	208	222	217	135	52	38.5
5/28/2010	B	214	216	215	219	182	97	53.3
6/10/2010	A	207	205	205	201	160	78	49.1
6/10/2010	B	255	260	223	266	163	109	66.9
6/11/2010	A	426	424	418	411	370	171	46.2
6/11/2010	B	519	530	536	537	392	231	58.9
6/12/2010	A	215	210	209	205	179	69	38.6
6/12/2010	B	248	253	262	264	206	128	62.1
6/14/2010	A	96	96	95	93	90	67	74.4
6/14/2010	B	106	108	117	113	102	80	78.4
6/16/2010	A	323	318	322	316	264	120	45.5
6/16/2010	B	390	391	377	412	270	182	67.4
6/17/2010	A	428	435	429	415	368	184	50
6/17/2010	B	487	493	478	458	291	204	70.1
7/1/2010	A	649	648	646	639	573	280	48.9
7/1/2010	B	759	768	760	793	522	342	65.5

One way to increase the matching performance is to only use travel times that have a high correlation coefficient. Figure 6.1 shows the relationship between the correlation coefficient chosen and the percentage of correct matches. The correct match percentage stays fairly constant until the threshold reaches about 0.95. After this, the percentage of correct matches rises quickly.

Conversely, as the threshold increases, fewer estimates are used. Figure 6.2 shows that the number of estimates with correlation coefficients above the threshold descends slightly until it reaches a correlation coefficient of about 0.95 and then it descends sharply. Figure 6.3 shows the relationship between the percentage of correct matches and the percentage of estimates that are kept. This shows a fairly linear relationship where the percentage of travel time estimates that are correct descends with increasing numbers of estimates.

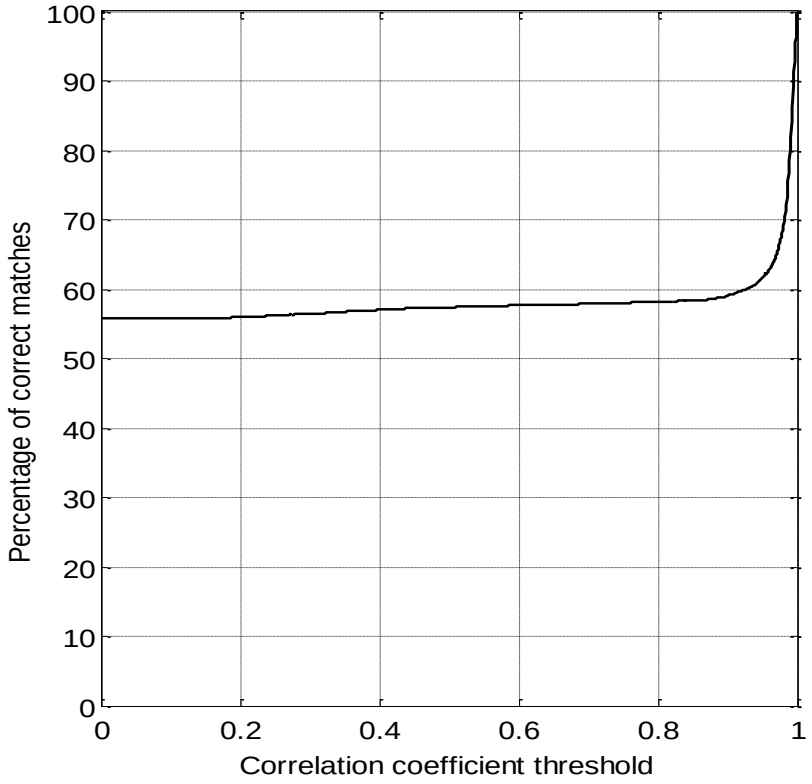


Figure 6.1: Percentage of correct matches vs. correlation coefficient threshold.

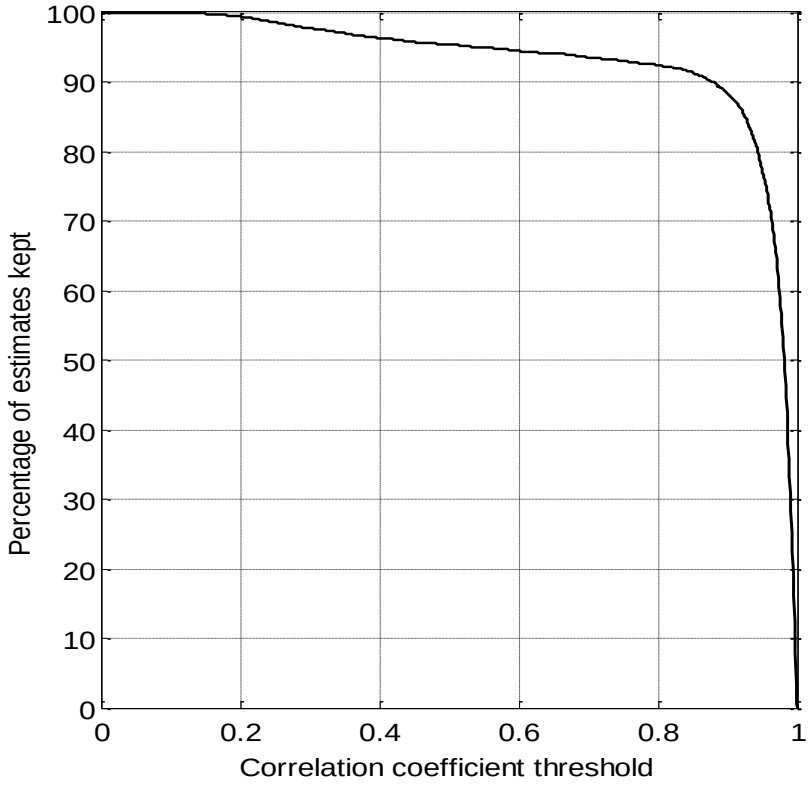


Figure 6.2: Percentage of matches kept vs. correlation coefficient threshold.

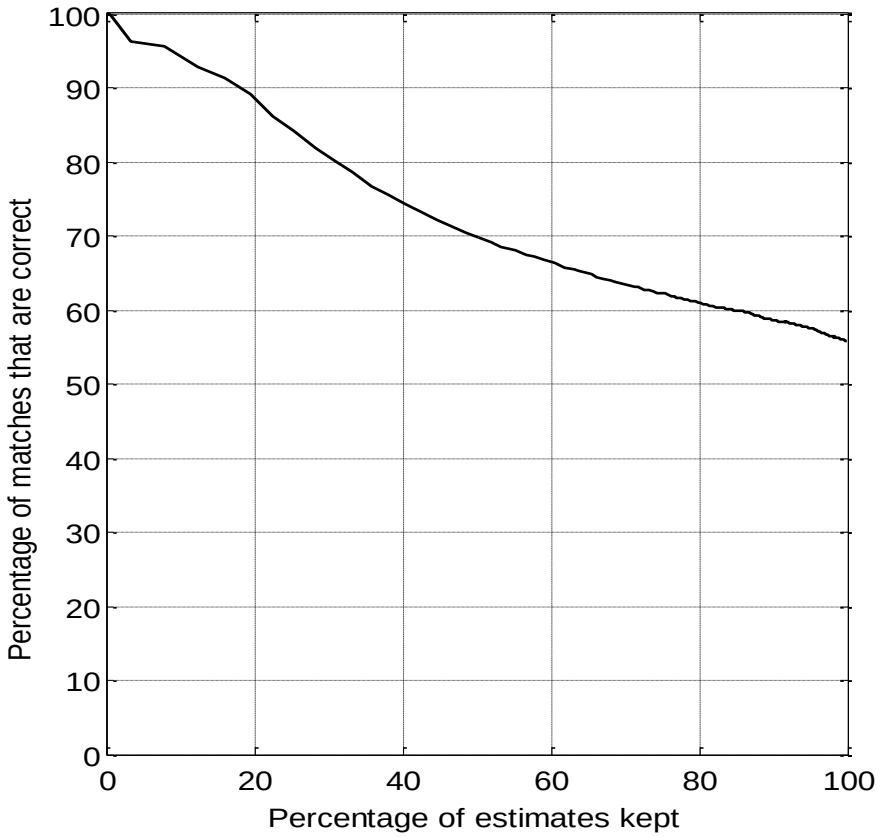


Figure 6.3: Percentage of correct matches vs. percentage of matches kept

The choice of where to operate on this graph depends on the application, but some guidance is given here for a couple example optimizations. The travel time estimates are a combination of both true and false travel times. This means that the travel time distribution, $f(t)$, can be described by a combination of the true travel time distribution, $f_1(t)$, the false travel time distribution, $f_2(t)$, and the percentage of estimates that are correct, α, in the following equation:

$$f(t) = \alpha f_1(t) + (1-\alpha) f_2(t)$$

The average travel time can be estimated by $\hat{\mu} = \frac{1}{n}\sum_{i=0}^{n-1} T(i)$ where each $T(i)$ is a travel time estimate from one of n vehicles. In this case, the bias is related to both the percentage of estimates that are from the true distribution and the difference between the true and false average travel time estimates through the following equation: $\text{bias}(\hat{\mu}) = (1-\alpha)(\mu_t - \mu_f)$. The most unbiased estimator is found by using only the most certain estimates.

The bias of an estimate is not usually sufficient to characterize the best estimate. As seen in this example, the most unbiased estimate is found by using only the estimate with the highest correlation coefficient. A more common metric used to evaluate an estimator is the mean square error. This metric is a combination of both the bias and variance of the estimate. By using a

linearization of the relationship between the percentage of estimates kept and the percentage of kept matches that are correct in Figure #, a point can be chosen that minimizes the mean square error. This analysis shows that the minimum mean square error (MMSE) estimator is found when the best 16.8% of the travel time estimates are used.

6.2 Causes of matching errors

Any signature matching travel time histogram will have errors because of the anonymous nature of signature matching. The matching performance at this data collection site is degraded because the sensors are not ideally located for signature matching. Ideally, the sensors should be placed at midblock or some location where vehicles are unlikely to be accelerating. Both the upstream and downstream sensors at this installation are close to the intersection, so a significant number of vehicles are decelerating or accelerating as they cross at least one pair of sensors. An example of this is shown in
Figure 6.4. The amount of time the vehicle spends over the lag sensor is significantly greater than the amount of time spent over the lead sensor. This lowers the correlation coefficient between the signature collected at the lead sensor and the signature collected at the lag sensor from about 0.98 to 0.8879. A more extreme case of this distortion is shown in Figure 6.5 where the vehicle stops on the sensor. These degraded signatures lead to false matches when attempting to match with signatures from another pair of sensors.

Sometimes signatures don't match well even though there is no obvious reason that the signature should be degraded. This may be due to slight lateral deviation causing one sensor to be affected by different parts of the vehicle. One example of prominent details in the lag signature that are diminished in the lead signature is shown in Figure 6.6. Figure 6.6a shows a continuous data stream from both the lead and lag detectors. If the algorithm worked perfectly, it would match the part labeled A in the lead signature to the part labeled B in the lag signature. This pair is shown in Figure 6.6b. Figure 6.6c shows the same part of the lead data stream paired with the part of the lag signature D. Since the three peaks in A are much less pronounced than they are in B, A and D have a higher correlation coefficient than A and B. This type of mismatch will occur regardless of whether there is deterioration due to acceleration/deceleration.

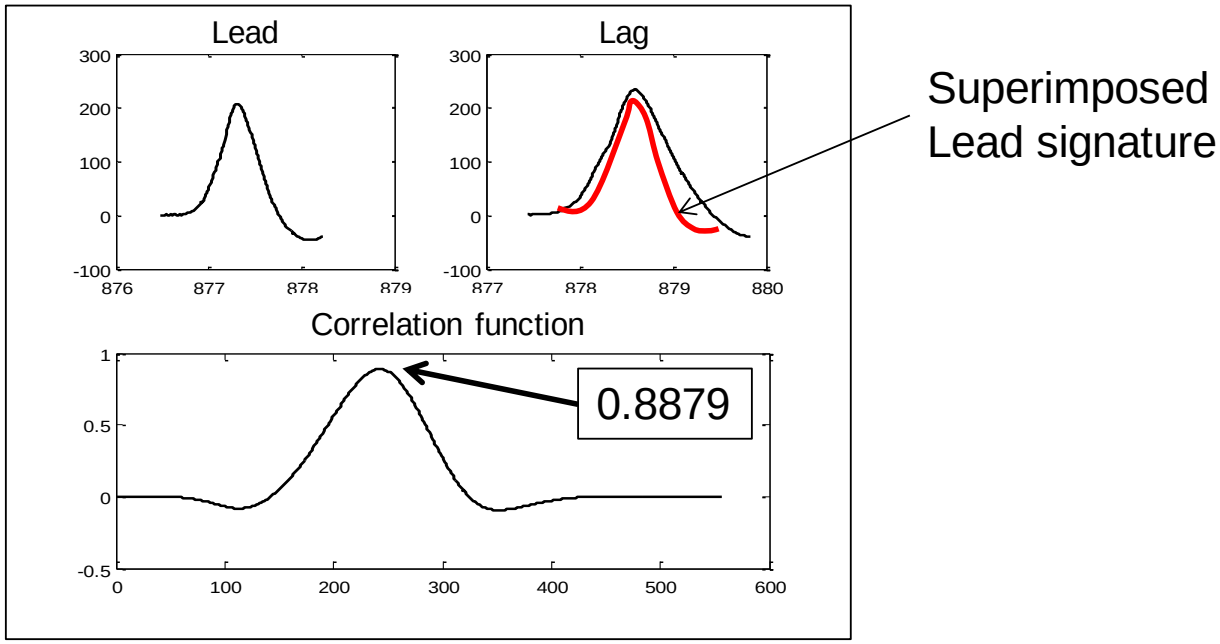


Figure 6.4: Effect of deceleration (or acceleration in general) on speed trap cross-correlation function

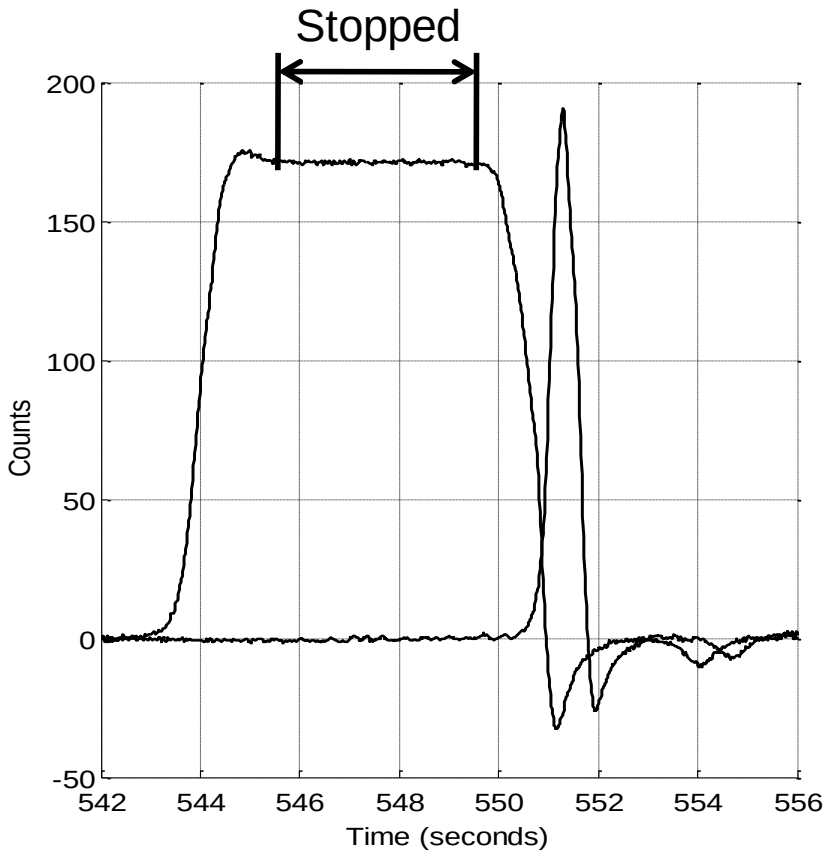
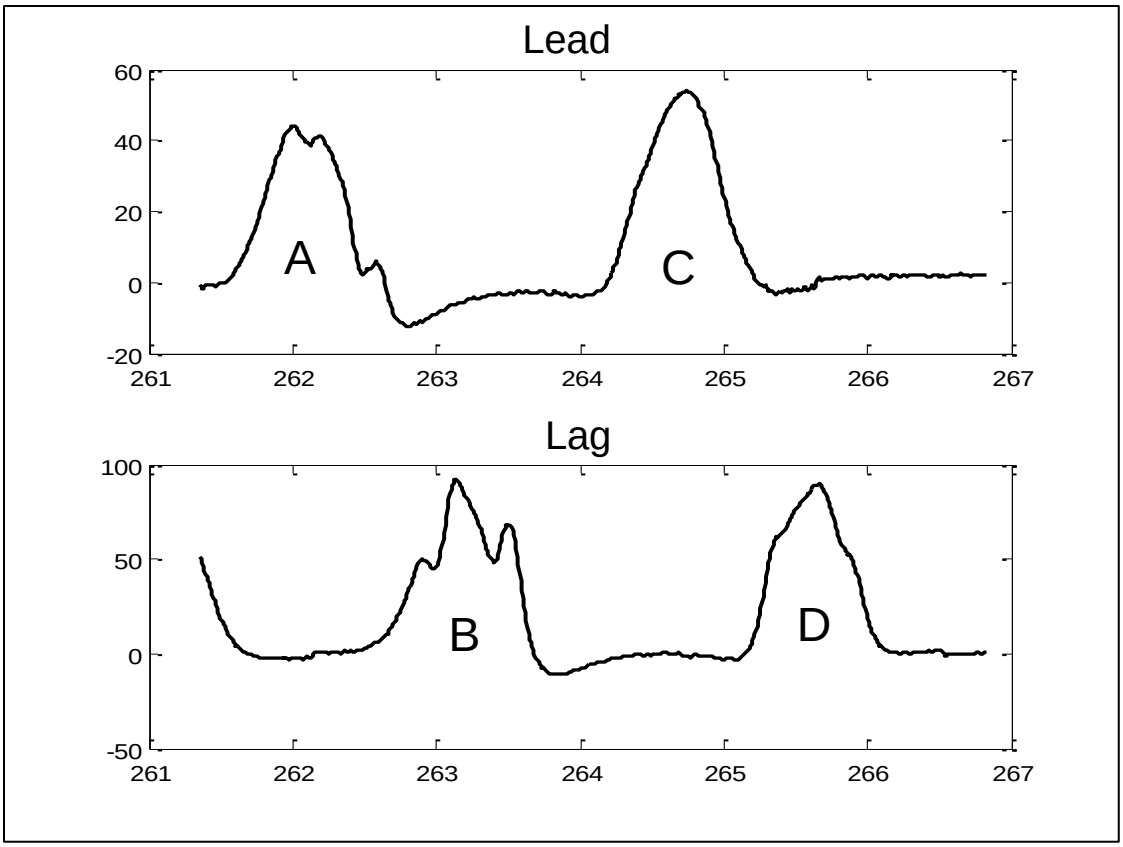
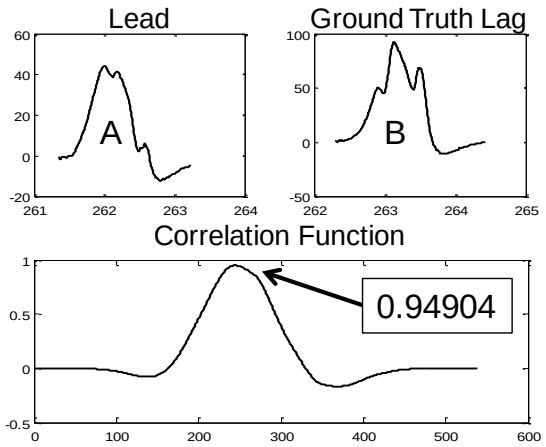


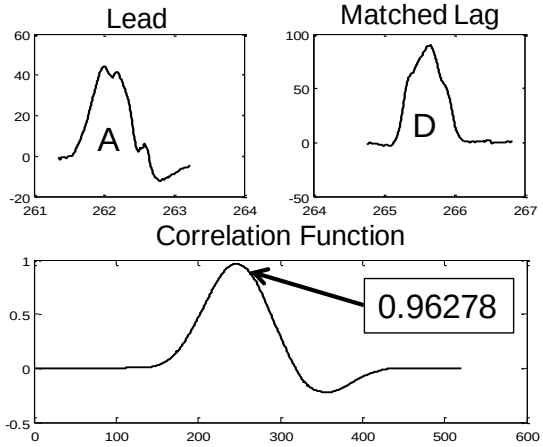
Figure 6.5: Effect of Stopping on the Sensor



a) Example Mismatched Lead/Lag travel time pair



b) Ground True Lead/Lag Match



c) Algorithm Lead/Lag Match

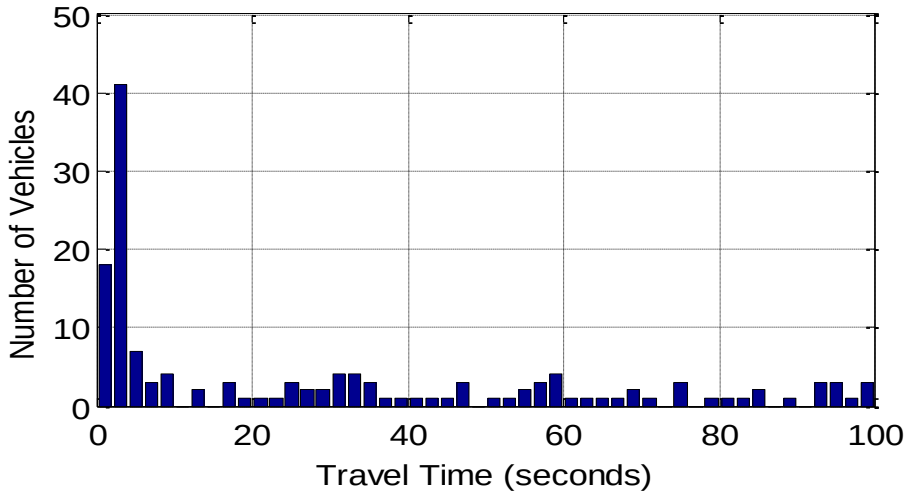
Figure 6.6: Signature Variability

6.3 Travel Time Histograms

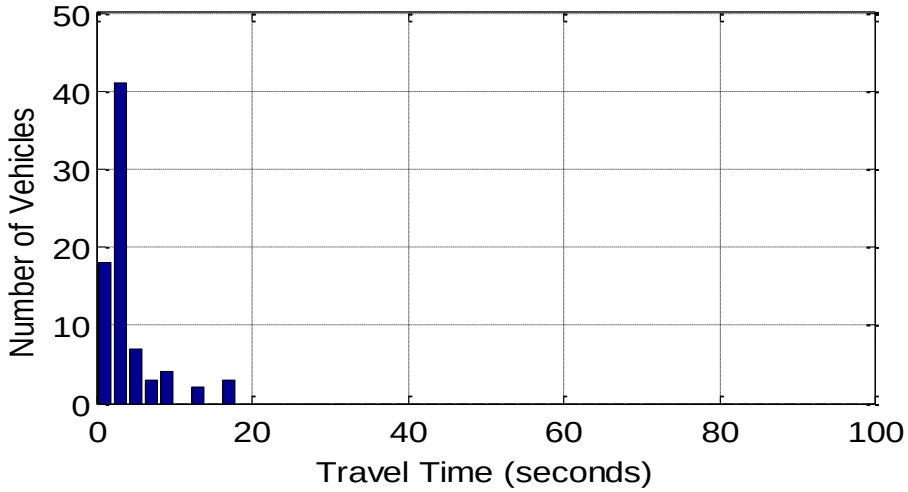
The data presented thus far is meant to analyze the matching percentages. While it is desirable have a high matching percentage, it is more important that the estimated travel time histogram be similar to the actual histogram. Figure 6.7 to Figure 6.30 show the histograms for each of the data sets. Part (a) of each figure shows the estimated histogram with no filtering. Part (b) shows the estimated histogram after the k-means filtering is applied. Part (c) is the true travel time histogram.

Figure 6.7c shows that the true travel times have a spike centered at about 4 seconds with a more uniform spread of travel times extending to about 40 seconds. The travel times extend up to 100 seconds from vehicles stopped between the upstream and downstream sensors due to cueing from the intersection. Figure 6.7a shows that the estimated travel times have a similar spike at about 4 seconds, but that there are noisy estimates that extend through the entire feasibility window. The k-means clustering attempts to remove the false travel time estimates. The result of this filtering is shown in Figure 6.7b. The spike in travel times is retained and many of the noisy travel times are successfully purged, but the longer true travel times are also removed. These longer travel times tend to be somewhat uniform and are therefore more difficult to separate from the erroneous travel time estimates.

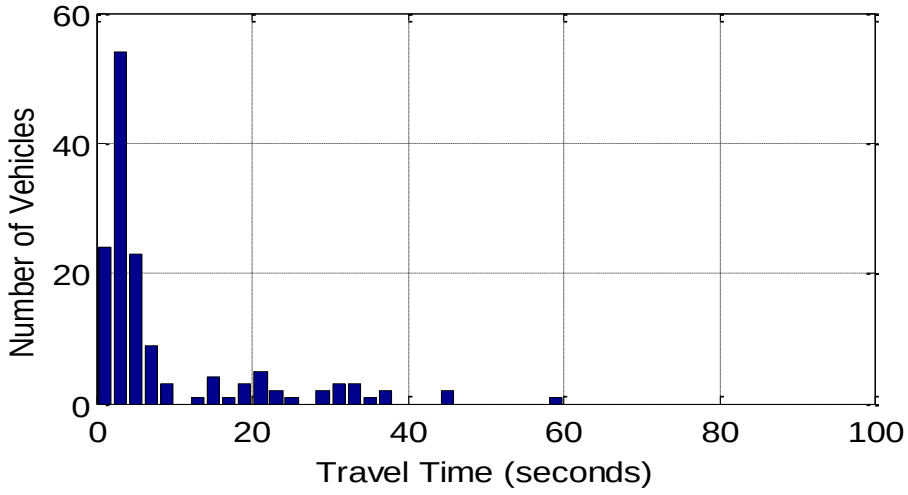
This same trend is followed through many of the other data sets. Occasionally, the k-means clustering keeps some of the noisy travel time estimates as seen in Figure 6.12. Instead of the travel time estimates being somewhat uniform, they are unusually dense around 60 seconds. This causes the k-means algorithm to keep these travel times. This error may be avoided by properly choosing the number of samples used to calculate the local variation for each point in the k-means clustering algorithm.



a) Unfiltered travel time histogram estimate

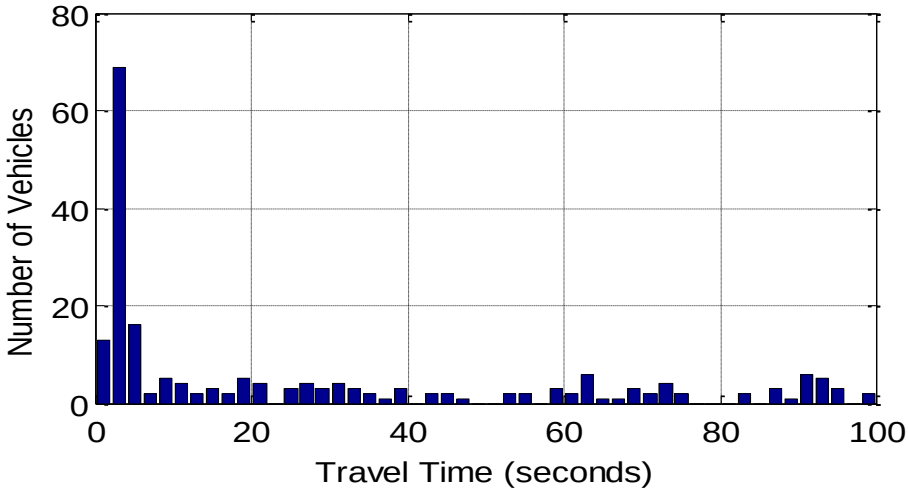


b) Filtered travel time histogram estimate

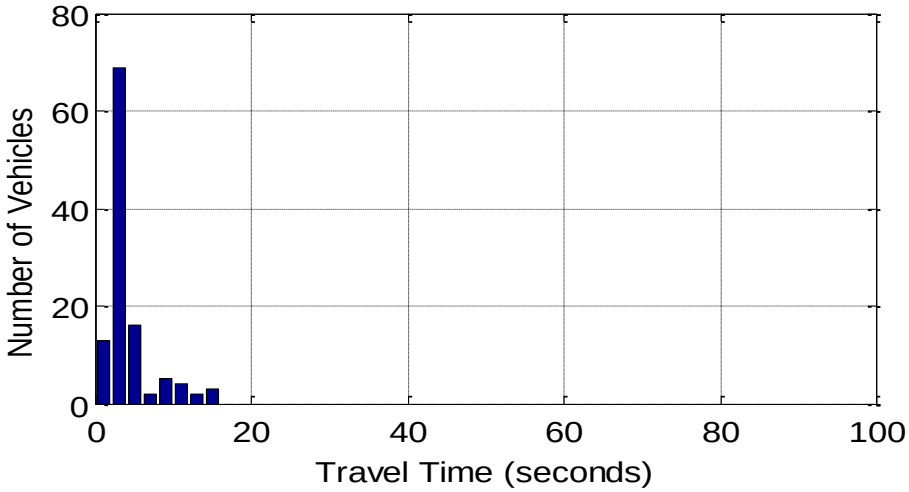


c) True travel time histogram

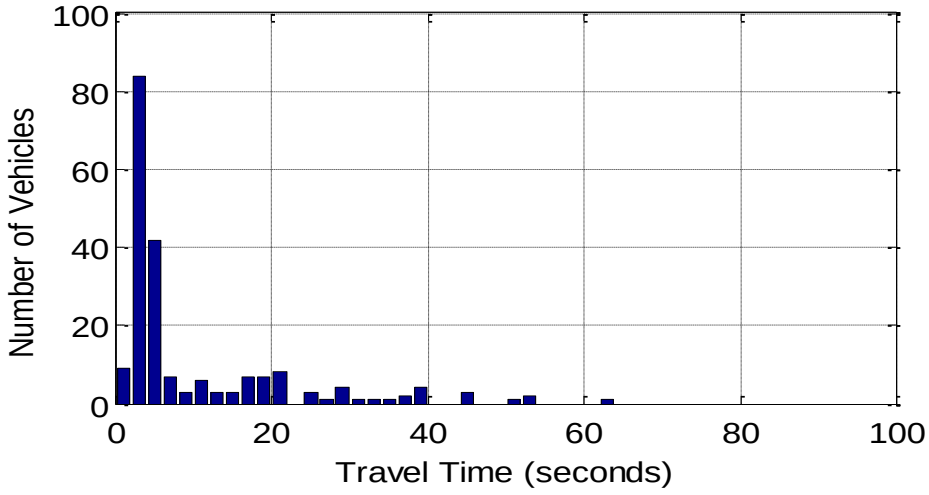
Figure 6.7: Travel time histogram for data set collected on 05/25/2009 Lane A (6)



a) Unfiltered travel time histogram estimate

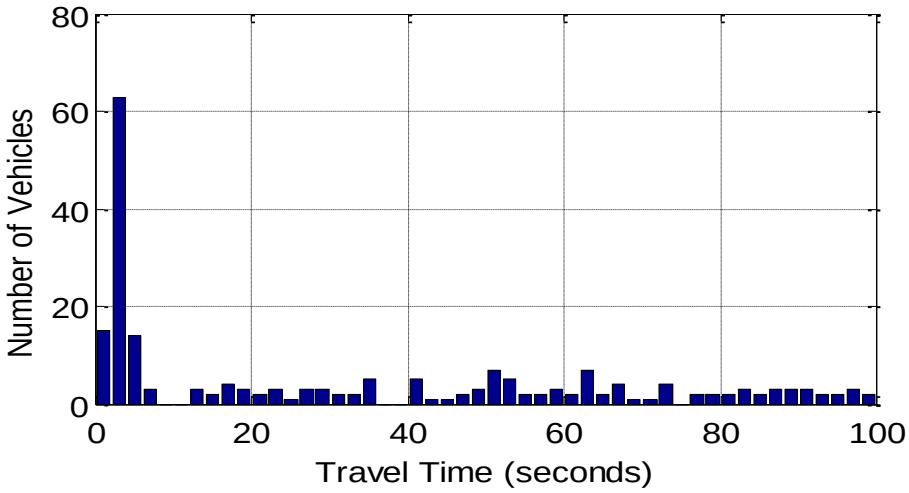


b) Filtered travel time histogram estimate

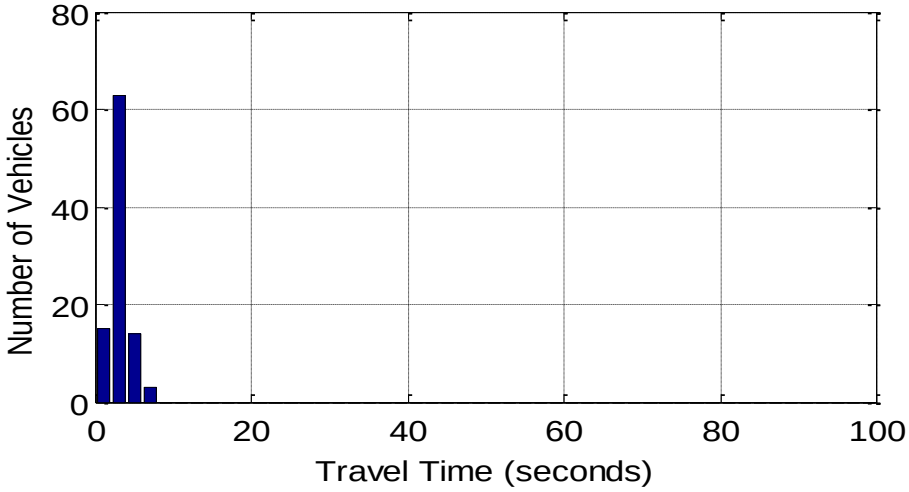


c) True travel time histogram

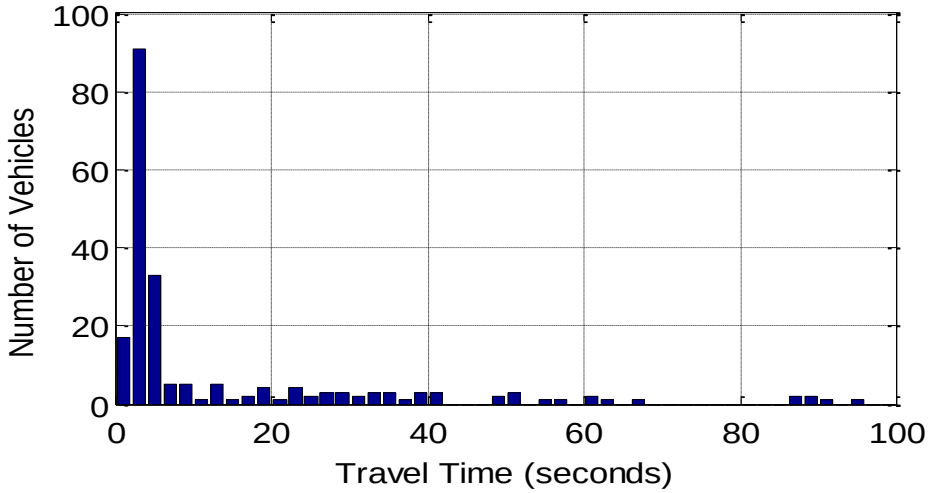
Figure 6.8: Travel time histogram for data set collected on 05/25/2009 Lane B (12)



a) Unfiltered travel time histogram estimate

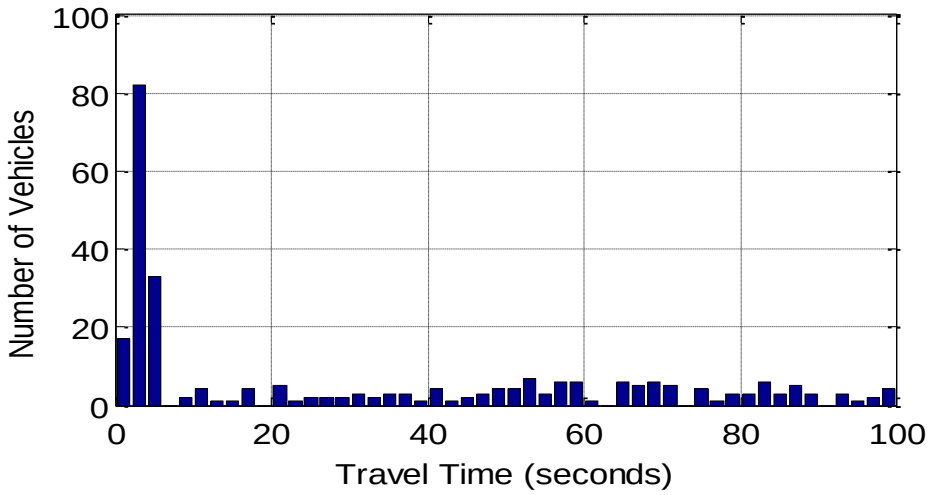


b) Filtered travel time histogram estimate

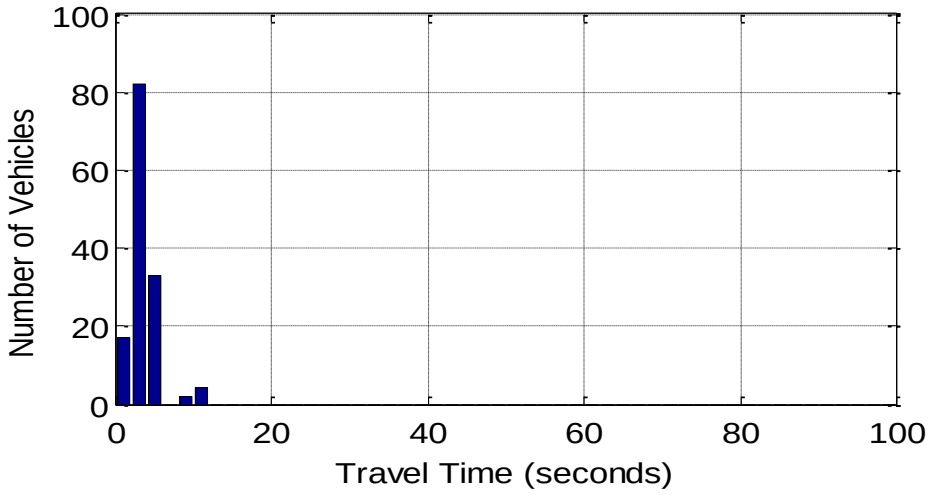


c) True travel time histogram

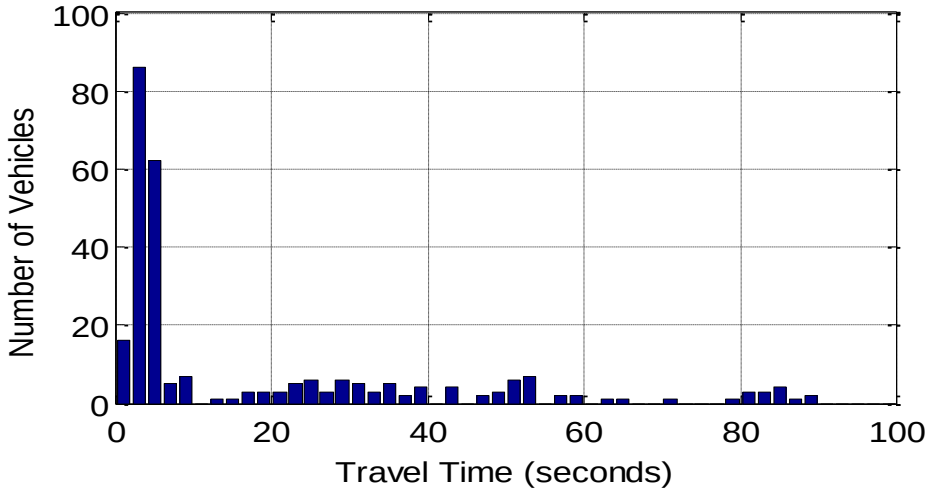
Figure 6.9: Travel time histogram for data set collected on 05/26/2009 Lane A (18)



a) Unfiltered travel time histogram estimate

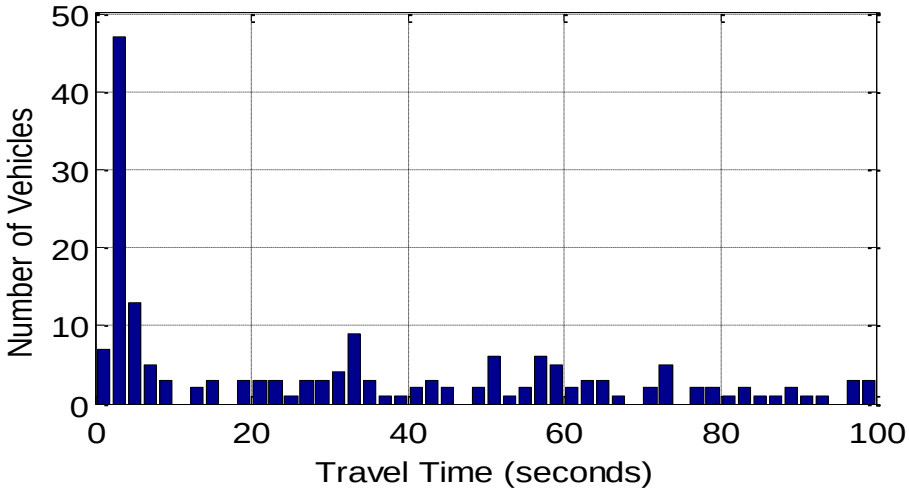


b) Filtered travel time histogram estimate

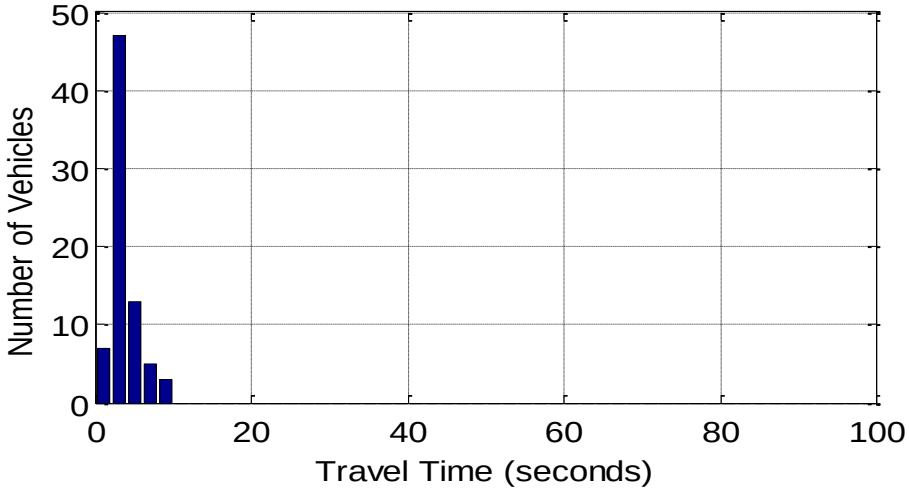


c) True travel time histogram

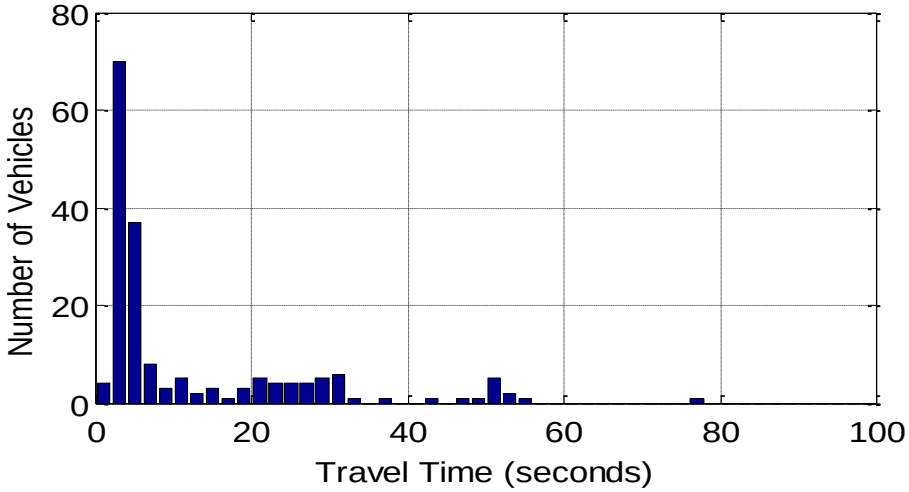
Figure 6.10: Travel time histogram for data set collected on 05/26/2009 Lane B (24)



a) Unfiltered travel time histogram estimate

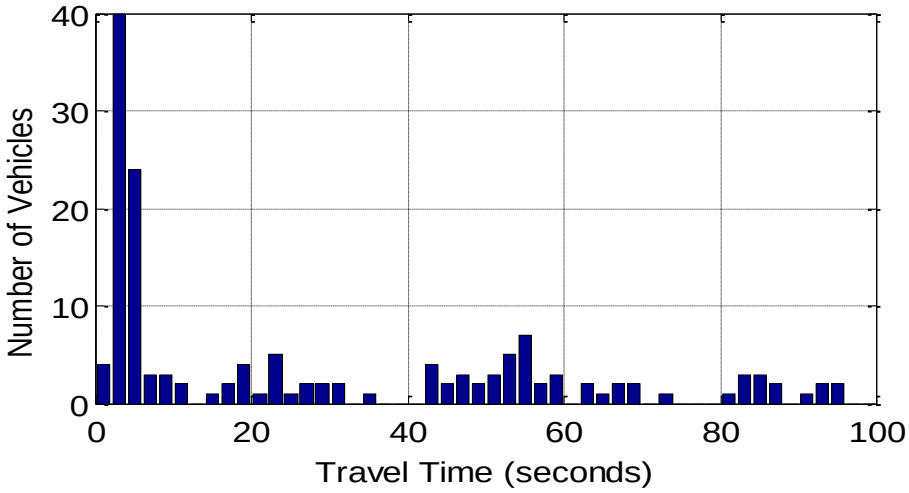


b) Filtered travel time histogram estimate

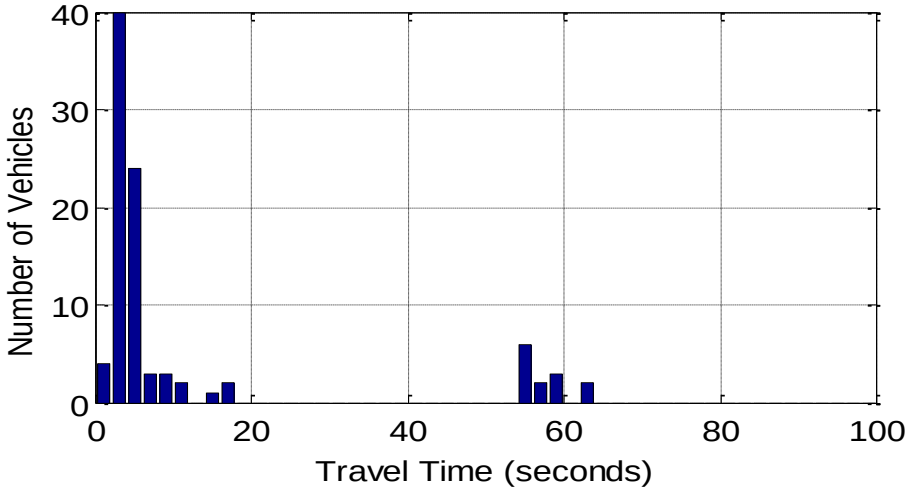


c) True travel time histogram

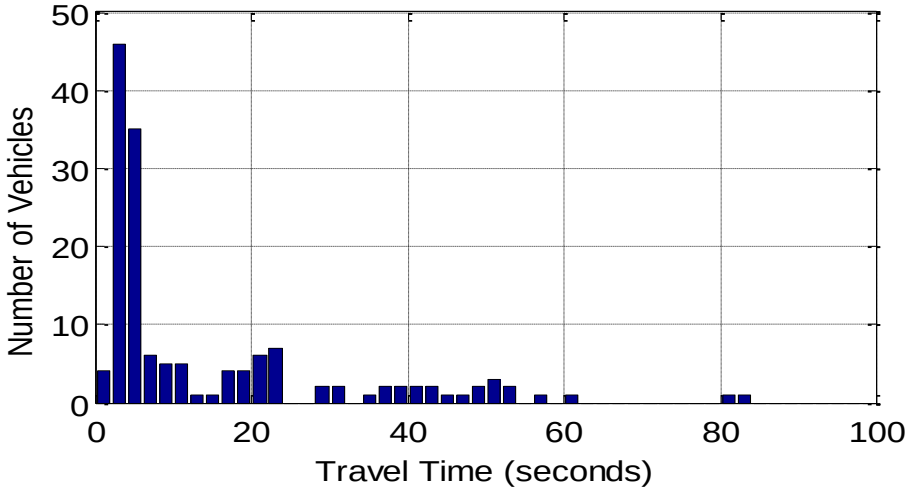
Figure 6.11: Travel time histogram for data set collected on 05/27/2009 Lane A (30)



a) Unfiltered travel time histogram estimate

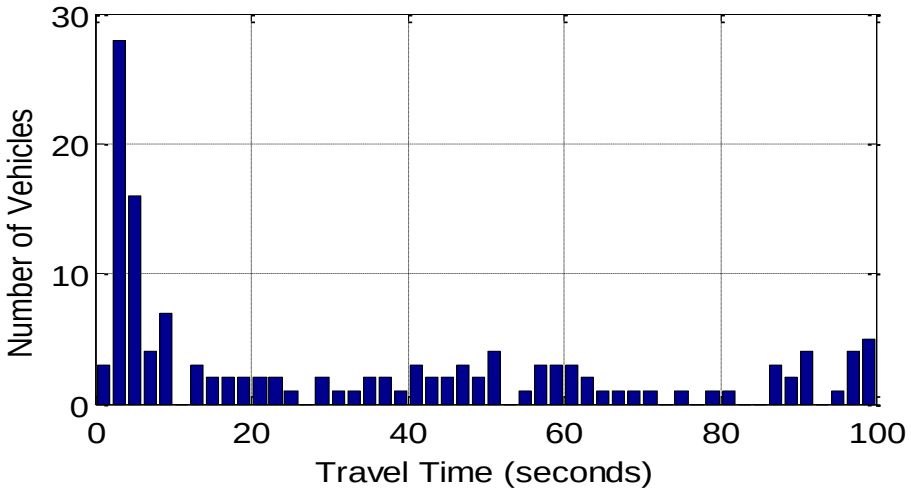


b) Filtered travel time histogram estimate

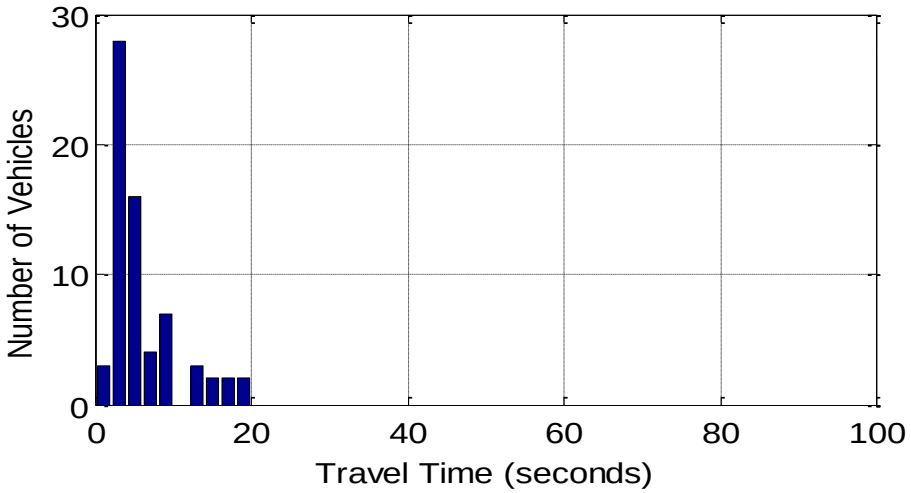


c) True travel time histogram

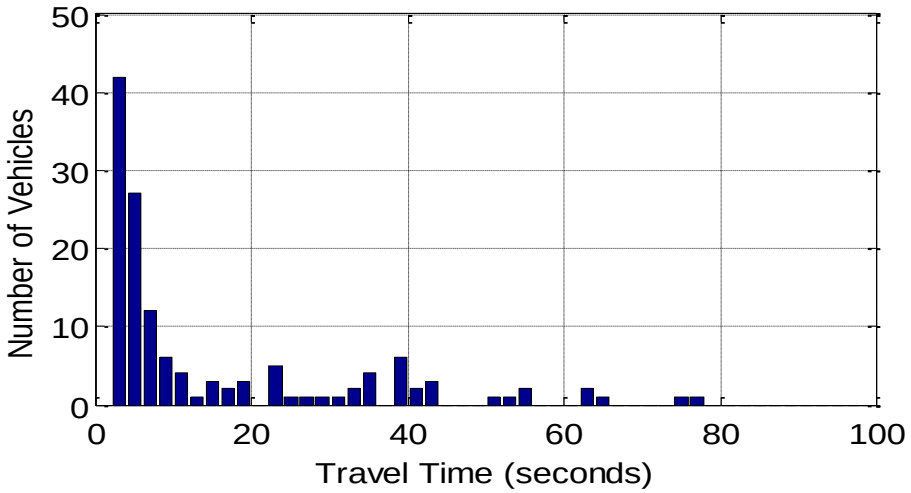
Figure 6.12: Travel time histogram for data set collected on 05/27/2009 Lane B (36)



a) Unfiltered travel time histogram estimate

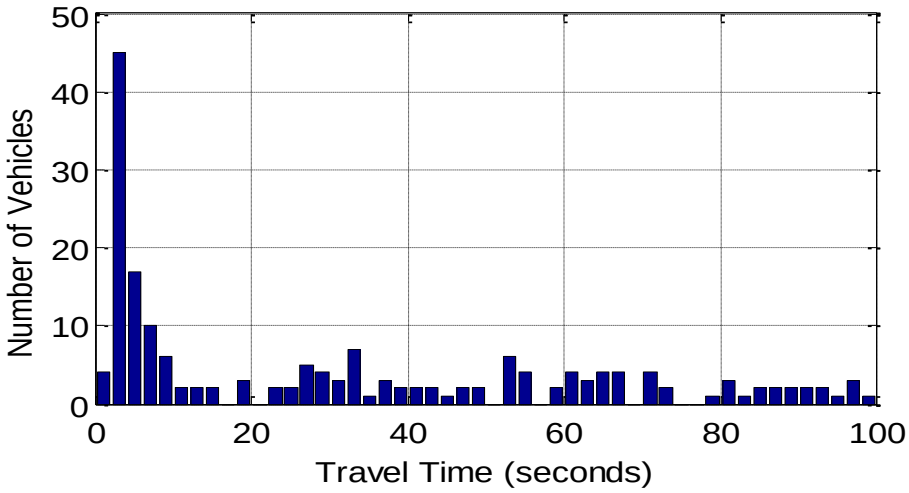


b) Filtered travel time histogram estimate

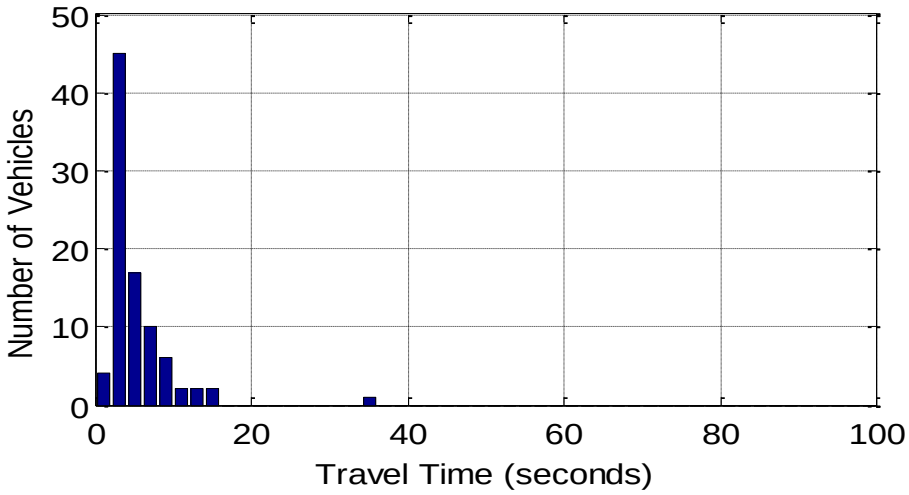


c) True travel time histogram

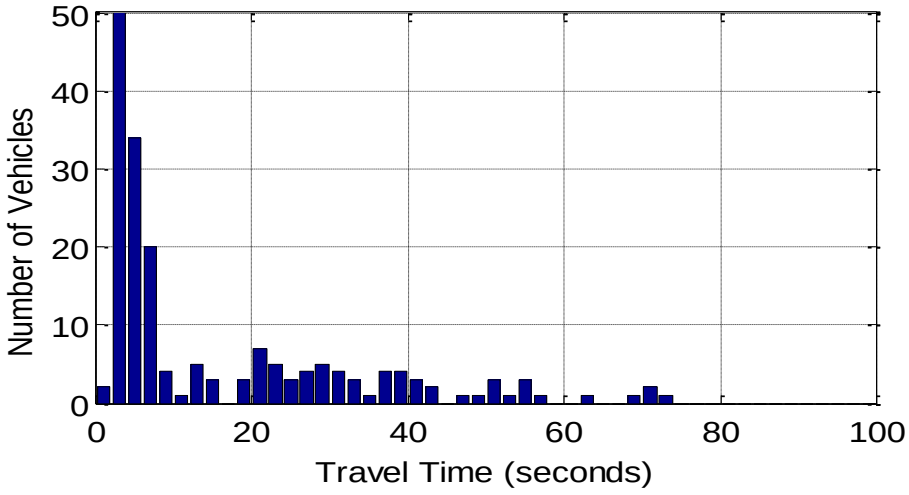
Figure 6.13: Travel time histogram for data set collected on 05/28/2009 Lane A (42)



a) Unfiltered travel time histogram estimate

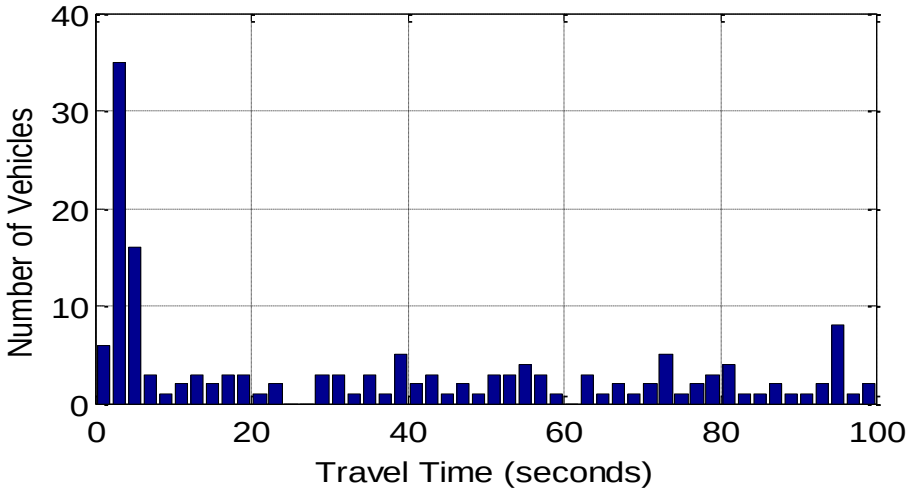


b) Filtered travel time histogram estimate

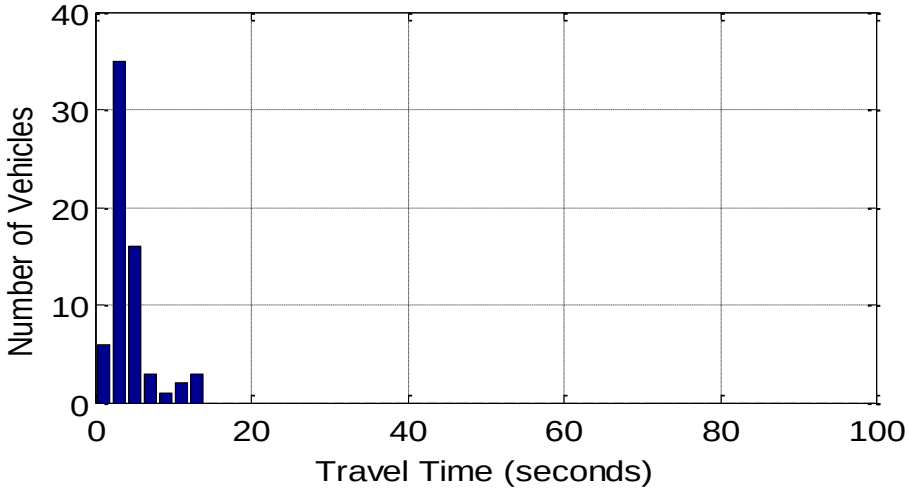


c) True travel time histogram

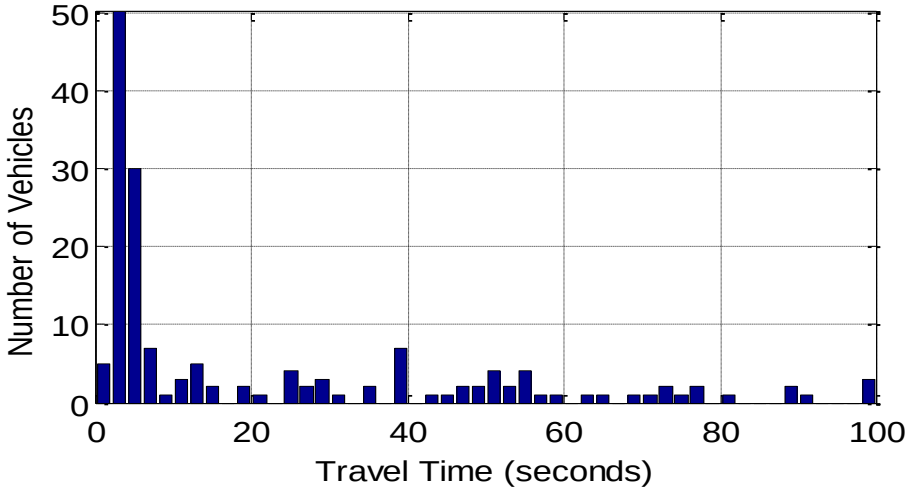
Figure 6.14: Travel time histogram for data set collected on 05/28/2009 Lane B (48)



a) Unfiltered travel time histogram estimate

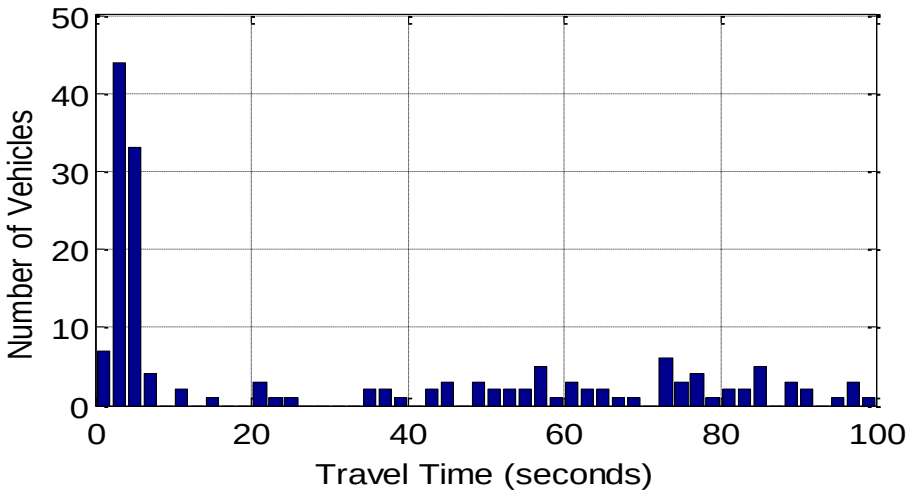


b) Filtered travel time histogram estimate

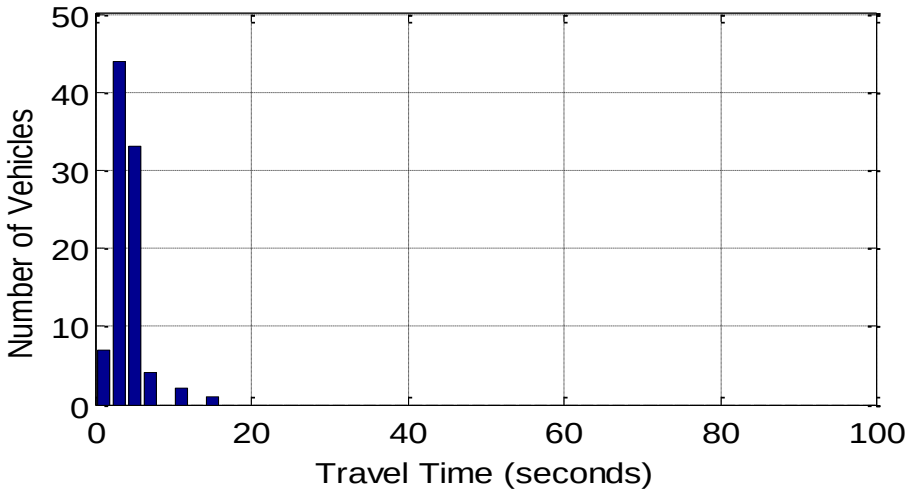


c) True travel time histogram

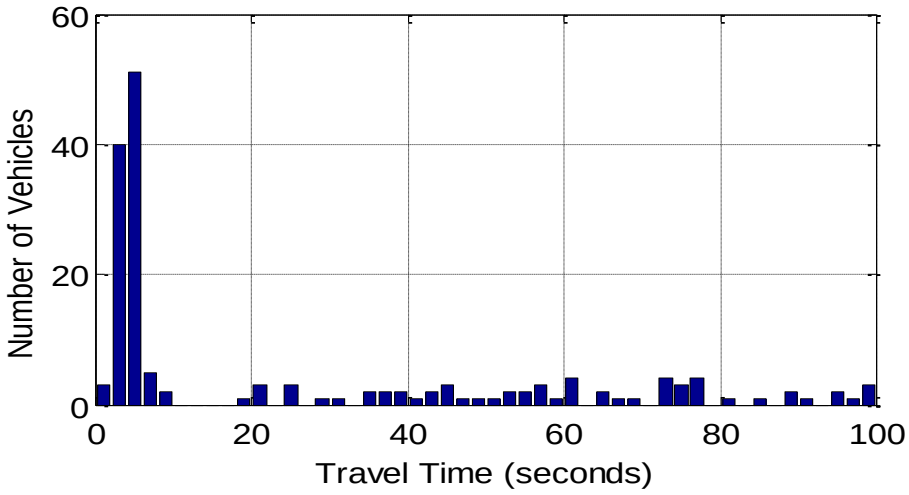
Figure 6.15: Travel time histogram for data set collected on 06/10/2009 Lane A (54)



a) Unfiltered travel time histogram estimate

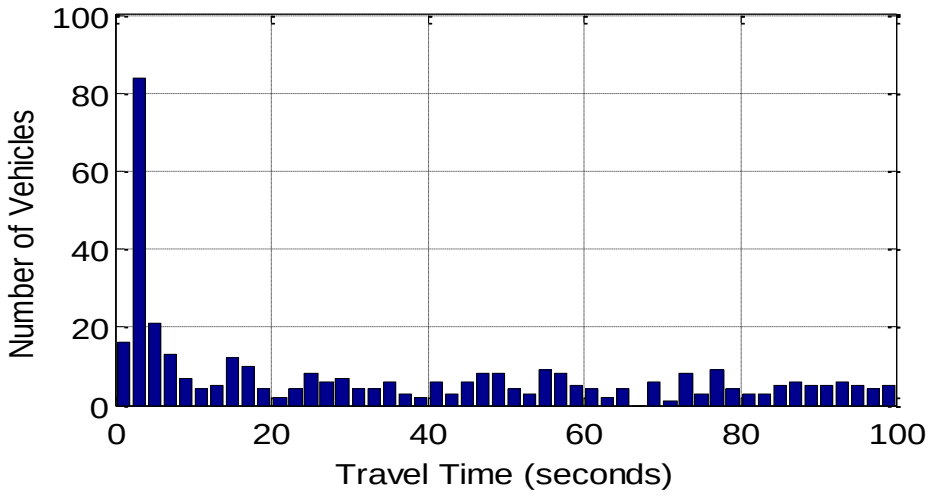


b) Filtered travel time histogram estimate

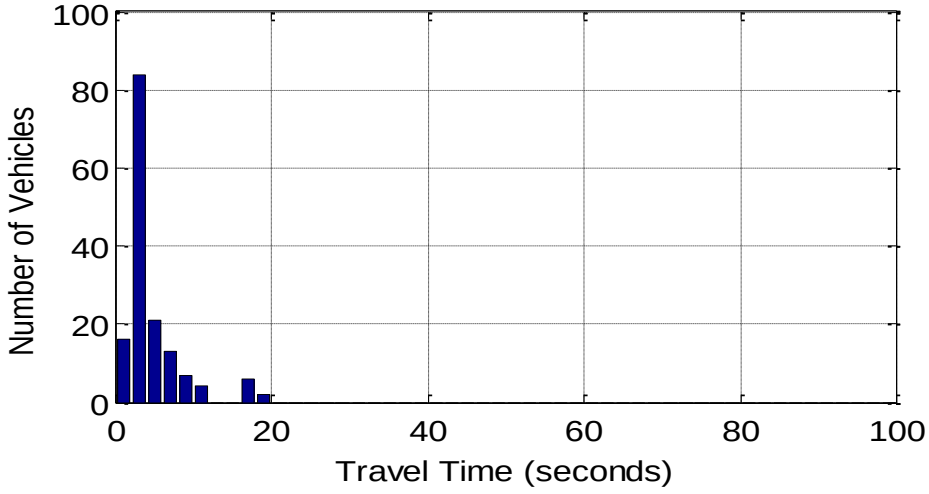


c) True travel time histogram

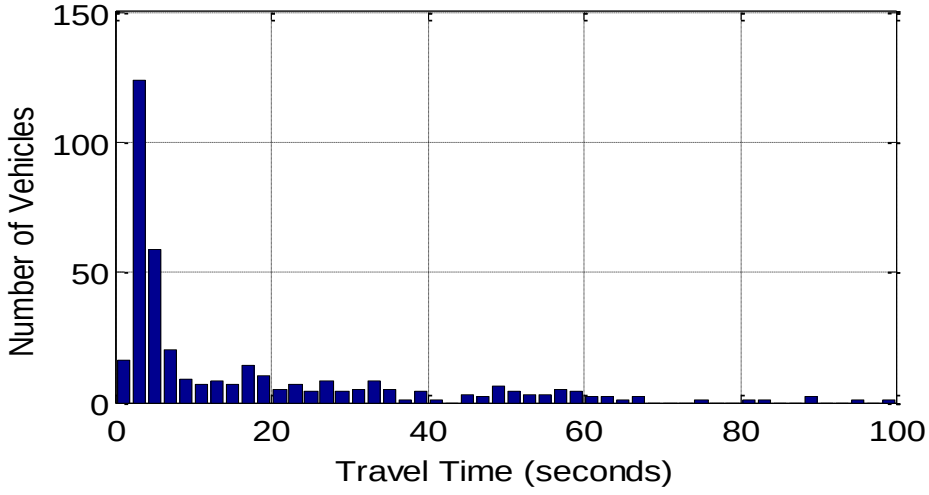
Figure 6.16: Travel time histogram for data set collected on 06/10/2009 Lane B (60)



a) Unfiltered travel time histogram estimate

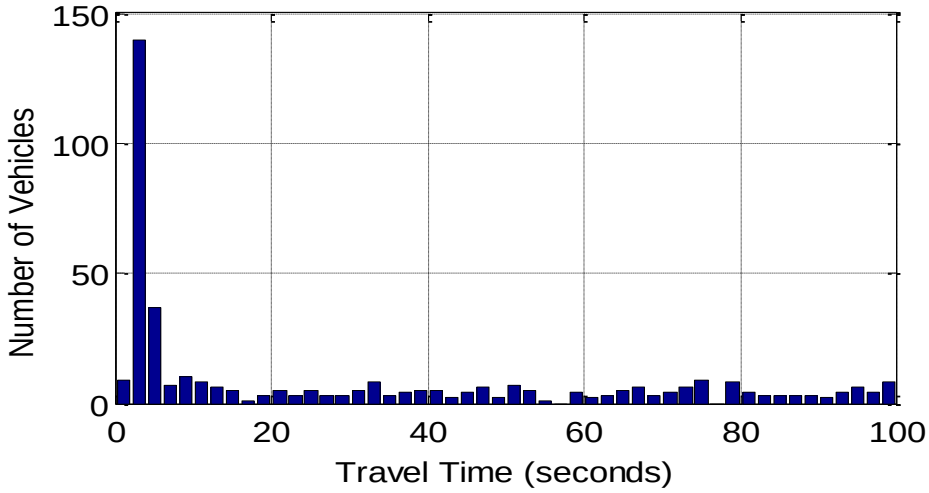


b) Filtered travel time histogram estimate

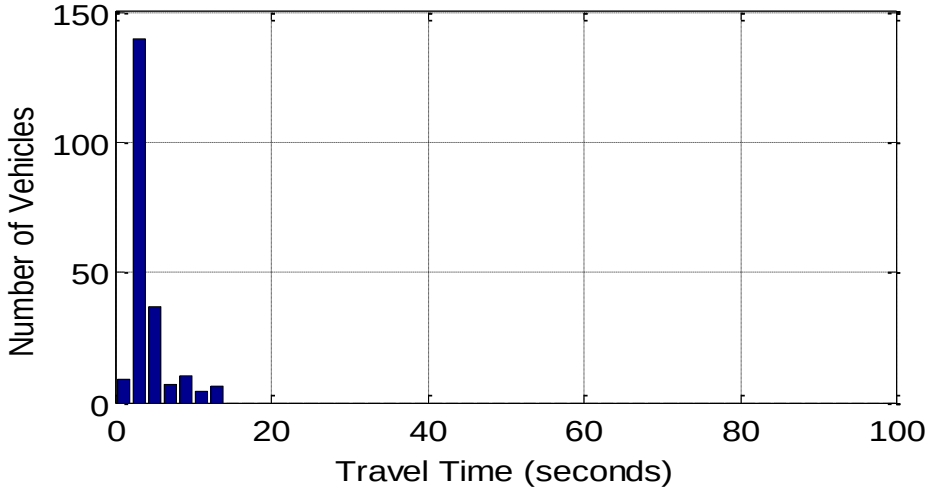


c) True travel time histogram

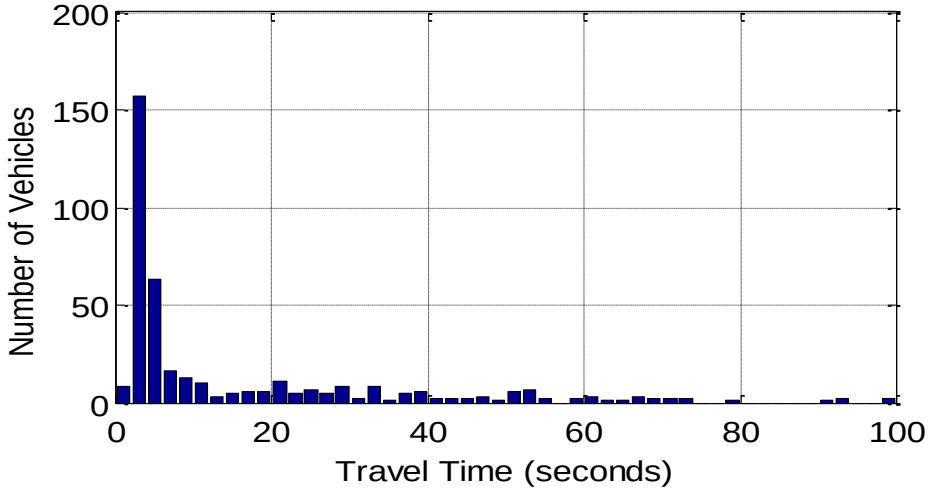
Figure 6.17: Travel time histogram for data set collected on 06/11/2009 Lane A (66)



a) Unfiltered travel time histogram estimate

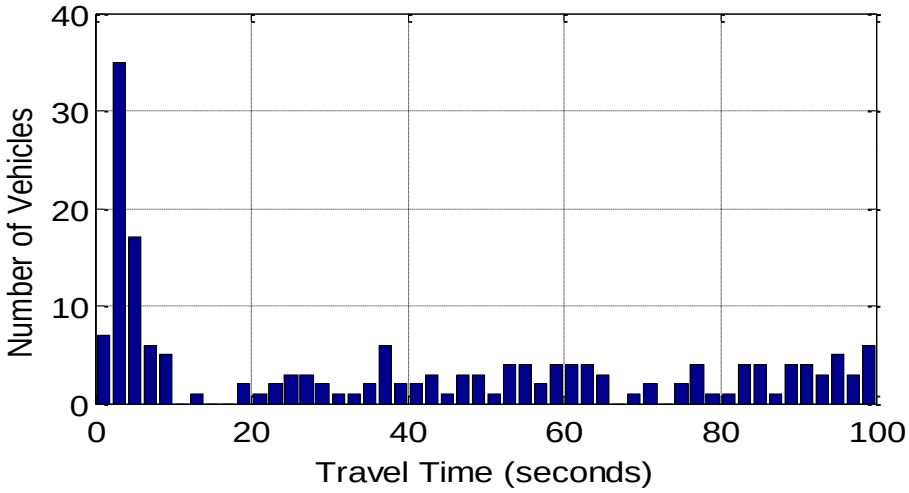


b) Filtered travel time histogram estimate

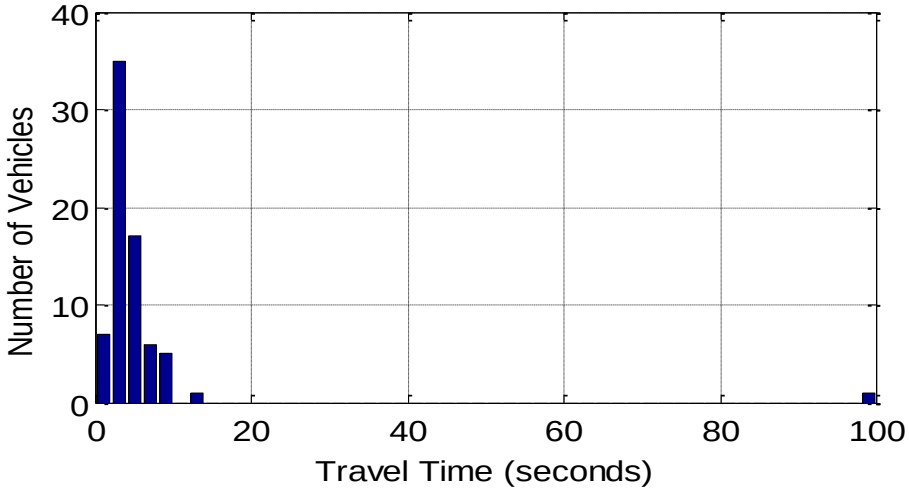


c) True travel time histogram

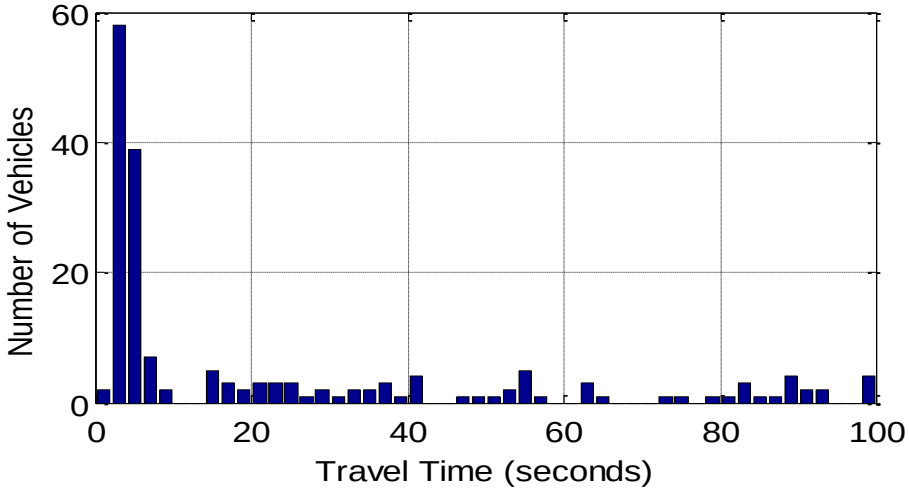
Figure 6.18: Travel time histogram for data set collected on 06/11/2009 Lane B (72)



a) Unfiltered travel time histogram estimate

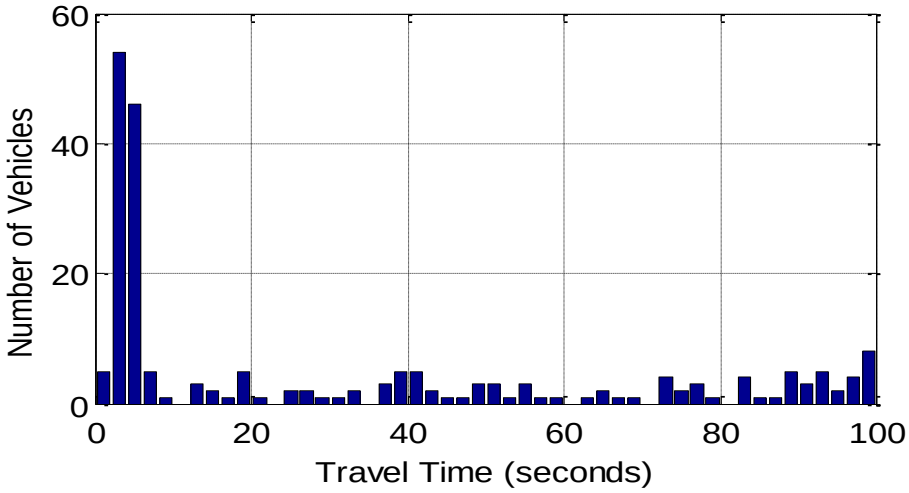


b) Filtered travel time histogram estimate

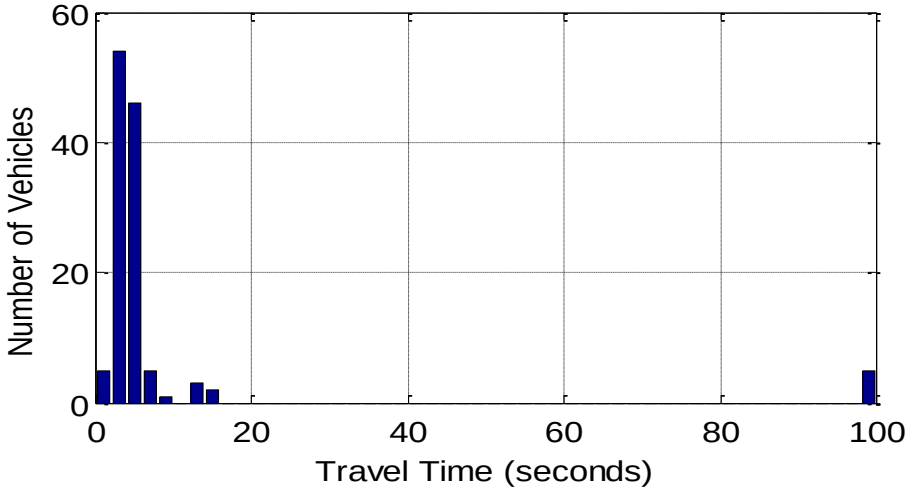


c) True travel time histogram

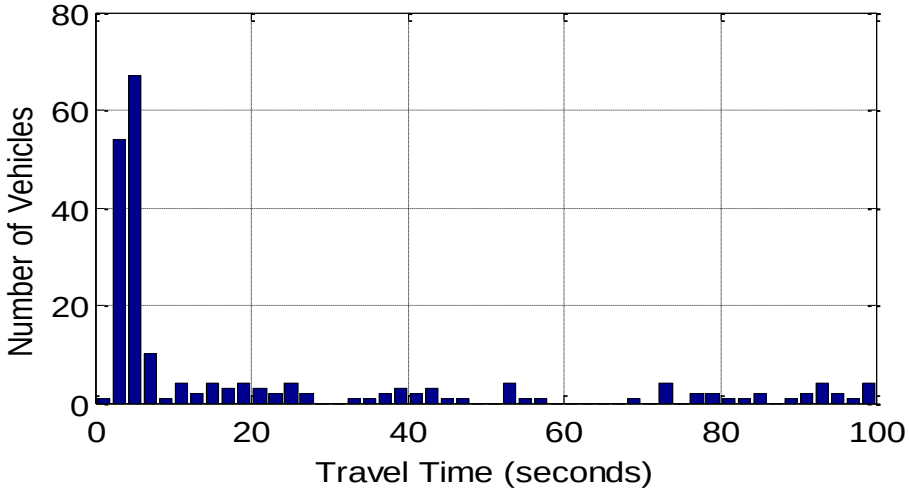
Figure 6.19: Travel time histogram for data set collected on 06/12/2009 Lane A (78)



a) Unfiltered travel time histogram estimate

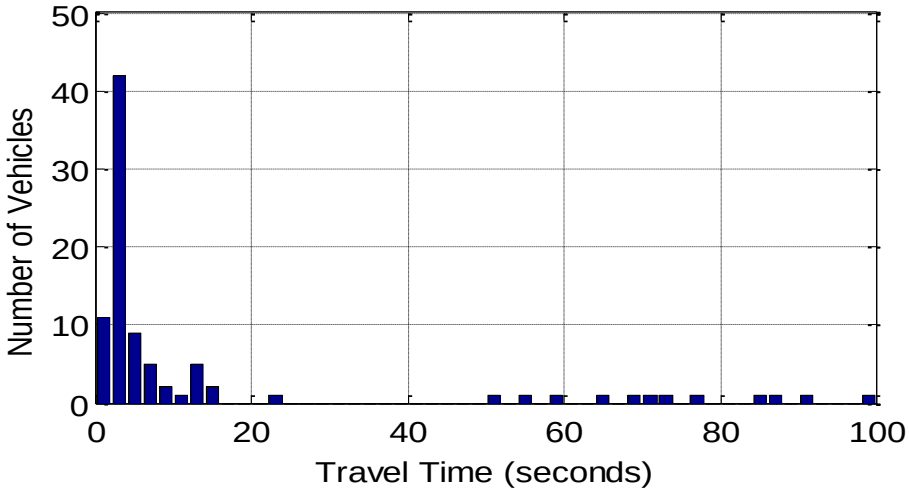


b) Filtered travel time histogram estimate

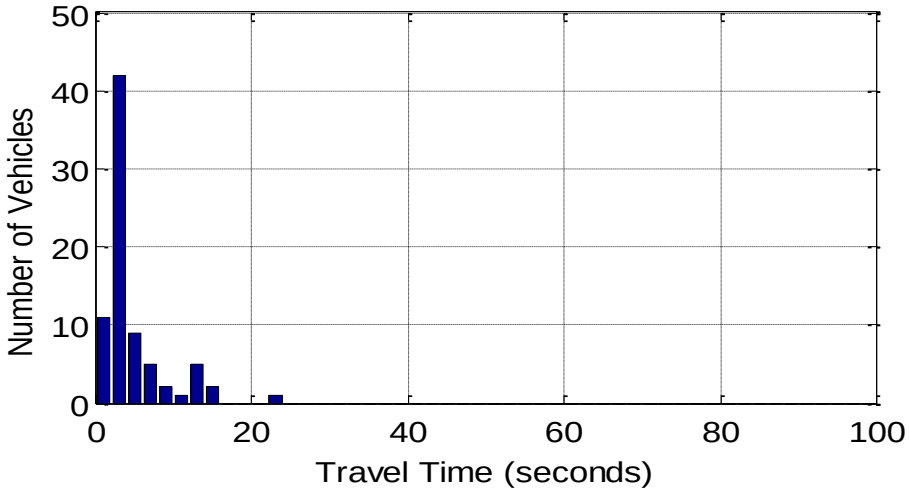


c) True travel time histogram

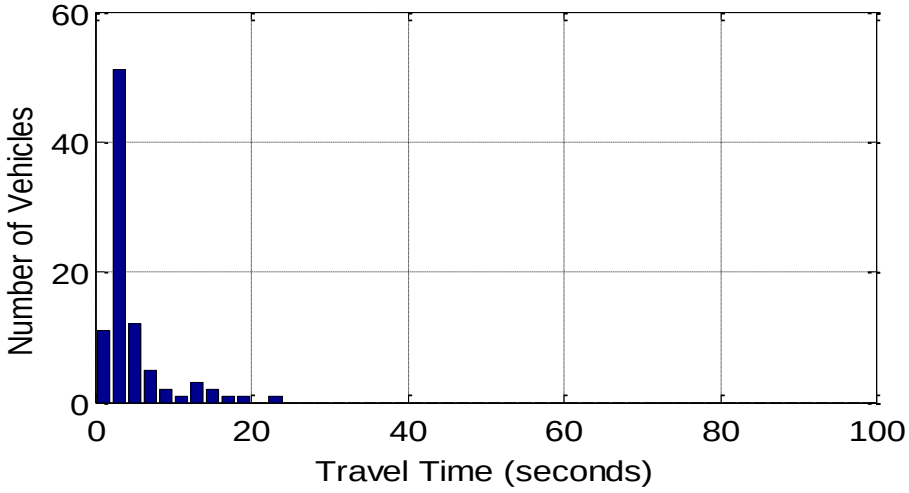
Figure 6.20: Travel time histogram for data set collected on 06/12/2009 Lane B (84)



a) Unfiltered travel time histogram estimate

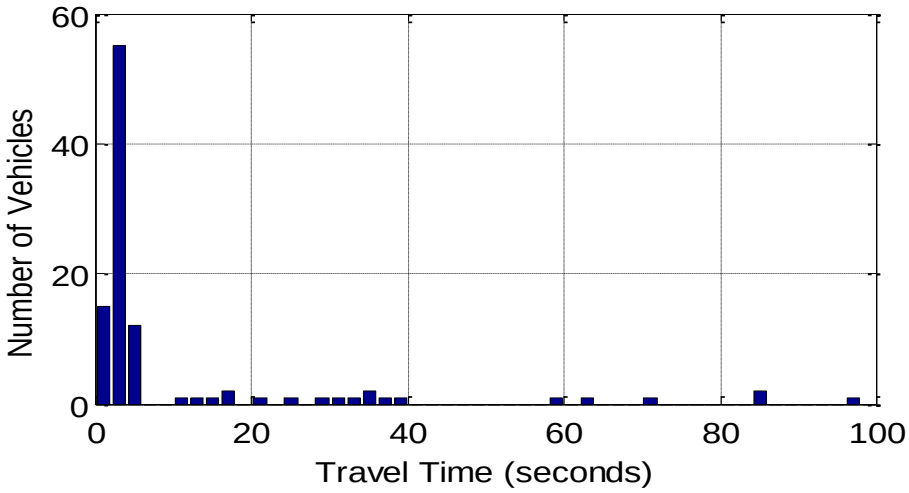


b) Filtered travel time histogram estimate

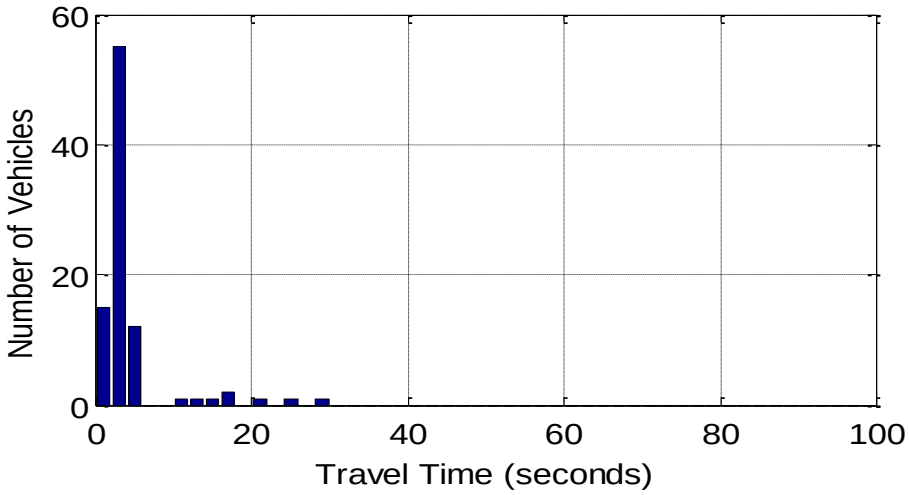


c) True travel time histogram

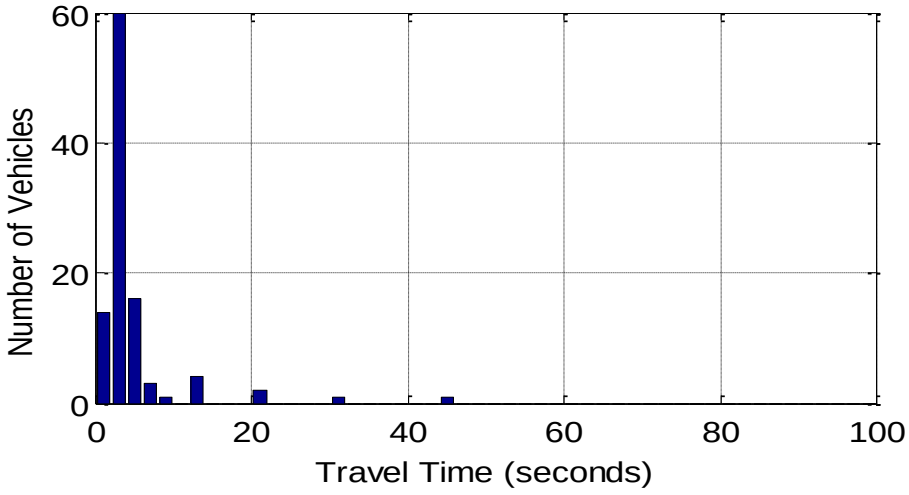
Figure 6.21: Travel time histogram for data set collected on 06/14/2009 Lane A (90)



a) Unfiltered travel time histogram estimate

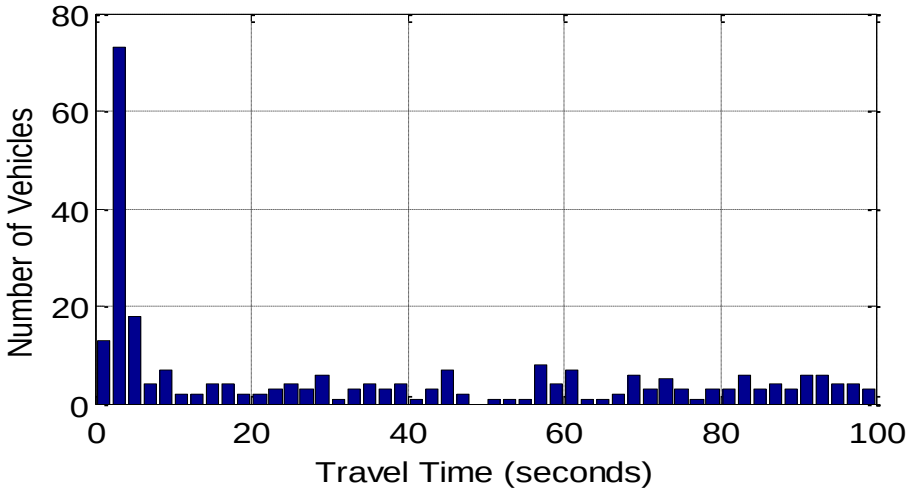


b) Filtered travel time histogram estimate

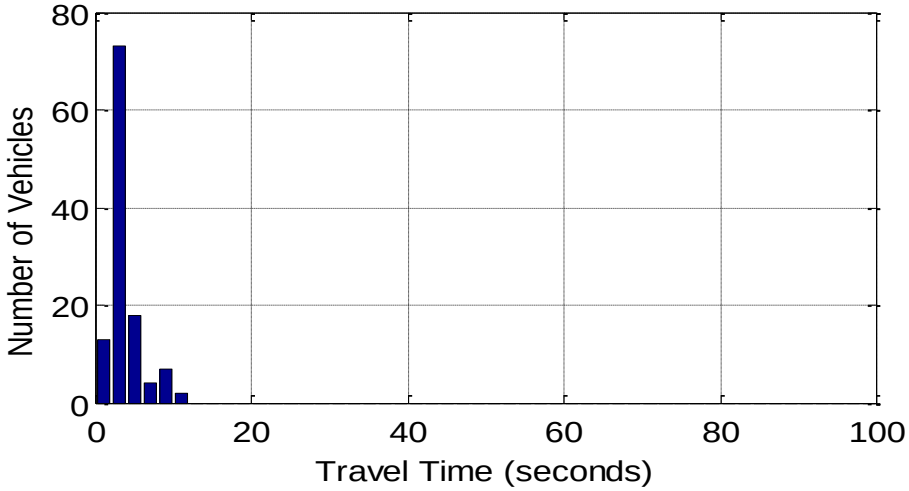


c) True travel time histogram

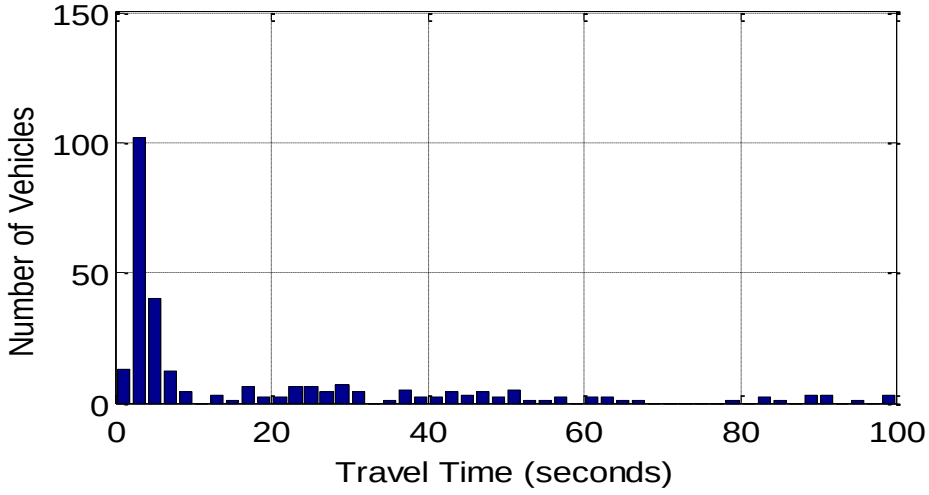
Figure 6.22: Travel time histogram for data set collected on 06/14/2009 Lane B (96)



a) Unfiltered travel time histogram estimate

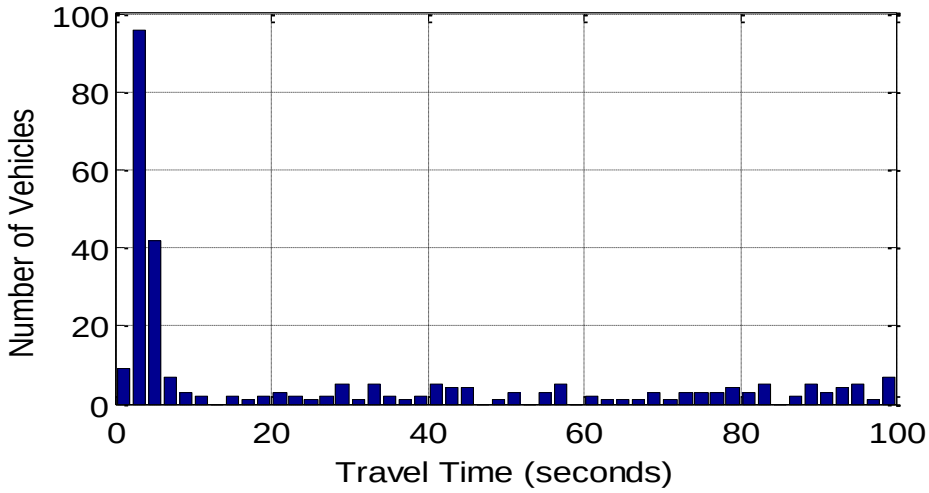


b) Filtered travel time histogram estimate

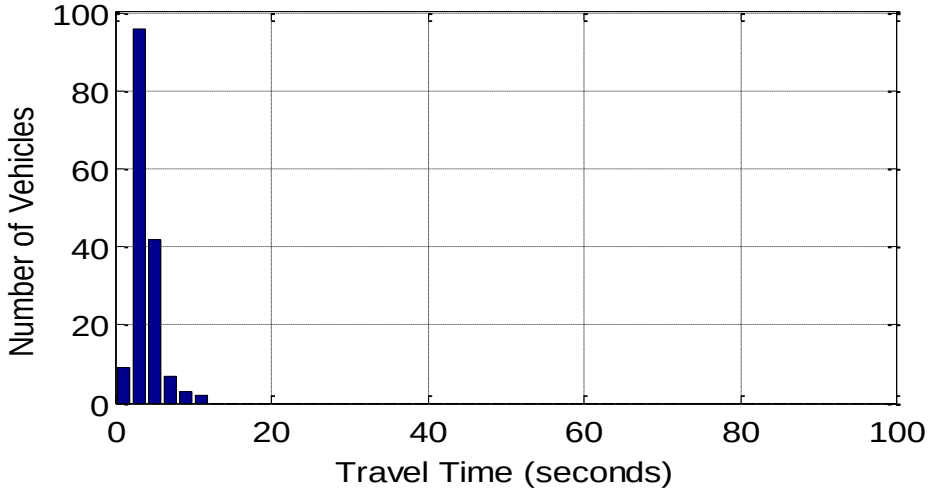


c) True travel time histogram

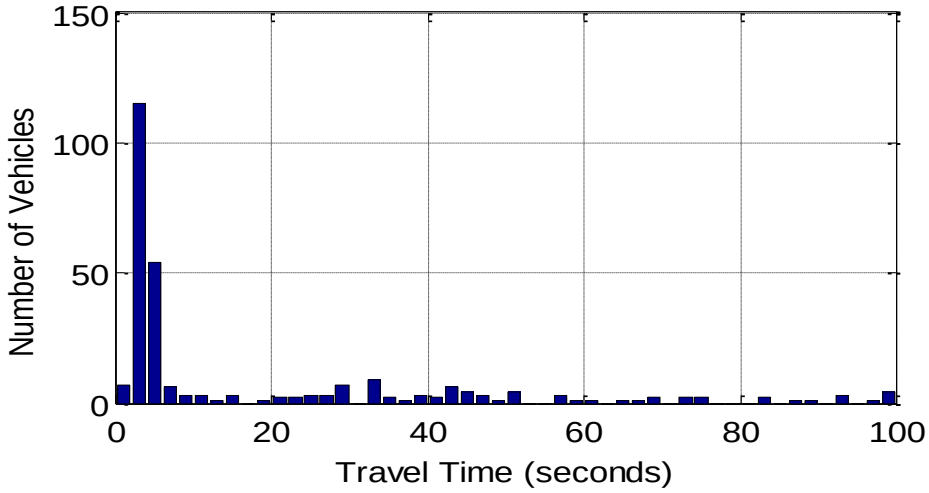
Figure 6.23: Travel time histogram for data set collected on 06/16/2009 Lane A (102)



a) Unfiltered travel time histogram estimate

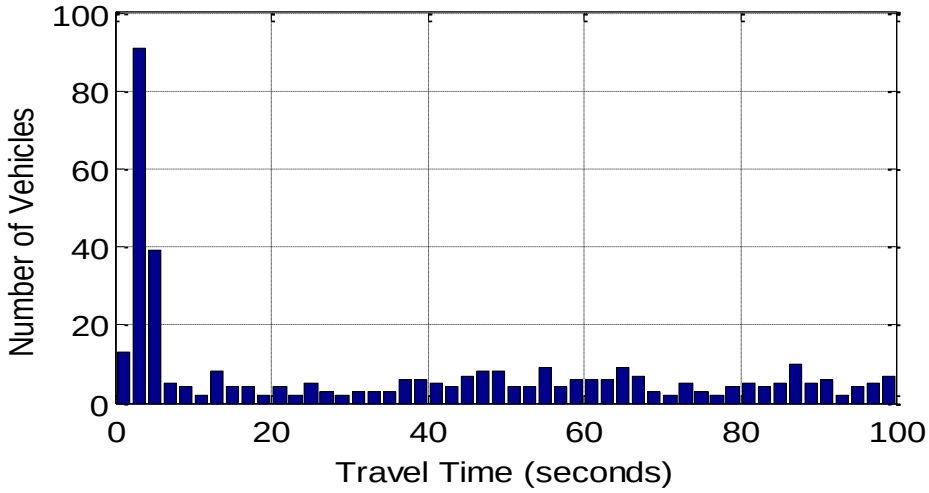


b) Filtered travel time histogram estimate

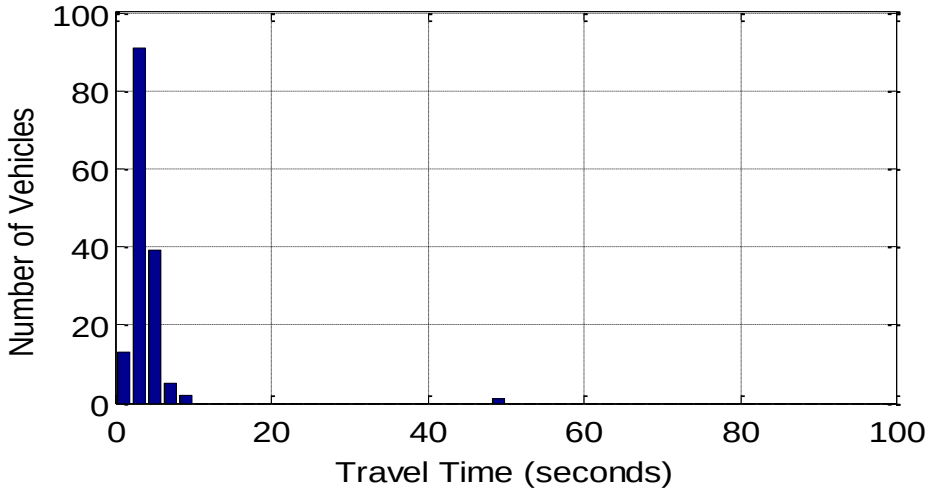


c) True travel time histogram

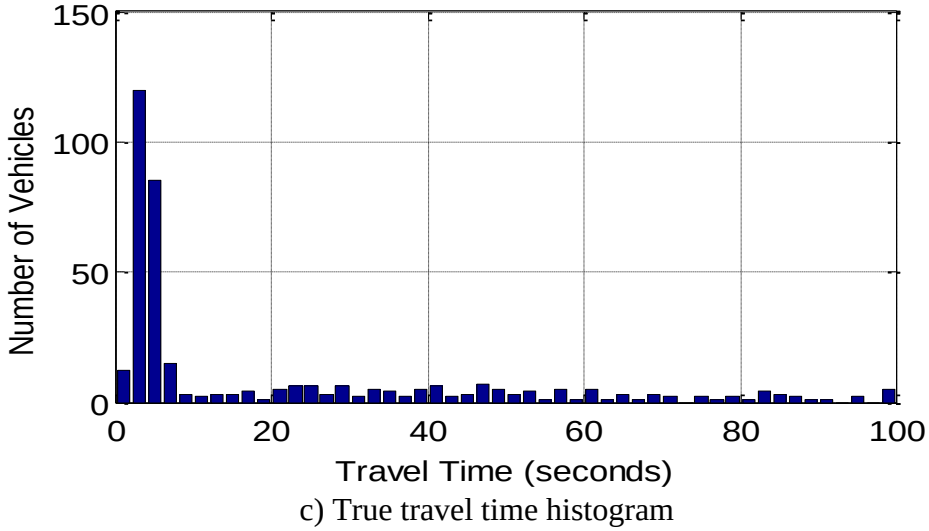
Figure 6.24: Travel time histogram for data set collected on 06/16/2009 Lane B (108)



a) Unfiltered travel time histogram estimate

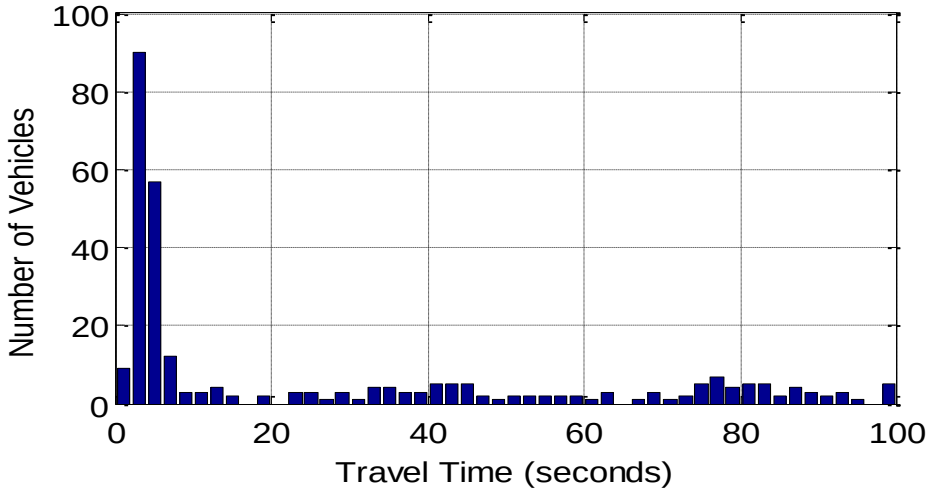


b) Filtered travel time histogram estimate

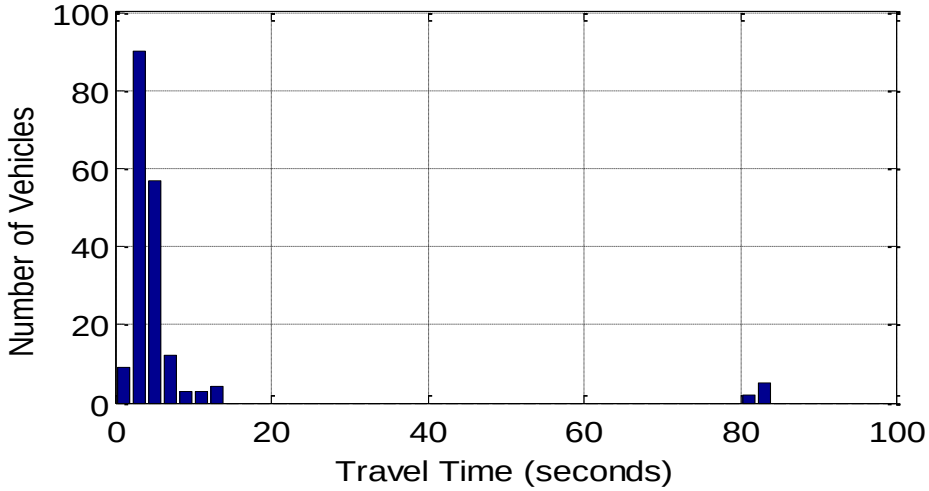


c) True travel time histogram

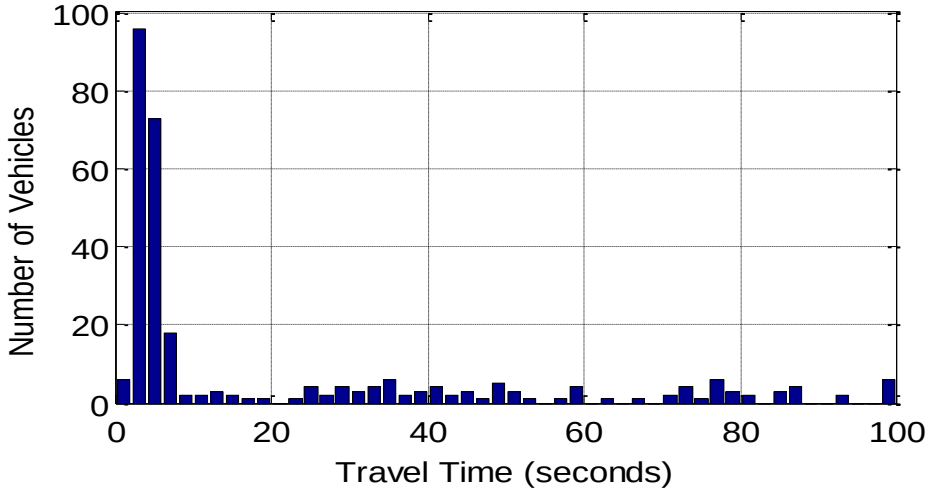
Figure 6.25: Travel time histogram for data set collected on 06/17/2009 Lane A (114)



a) Unfiltered travel time histogram estimate

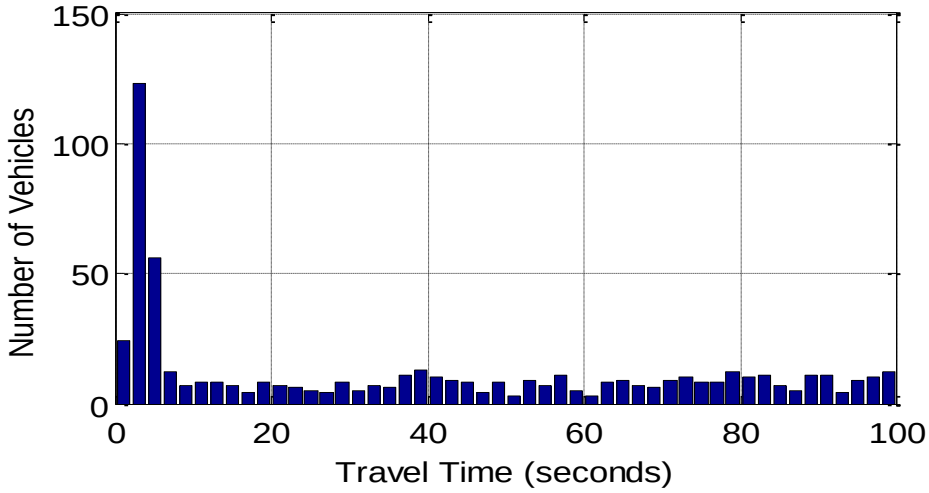


b) Filtered travel time histogram estimate

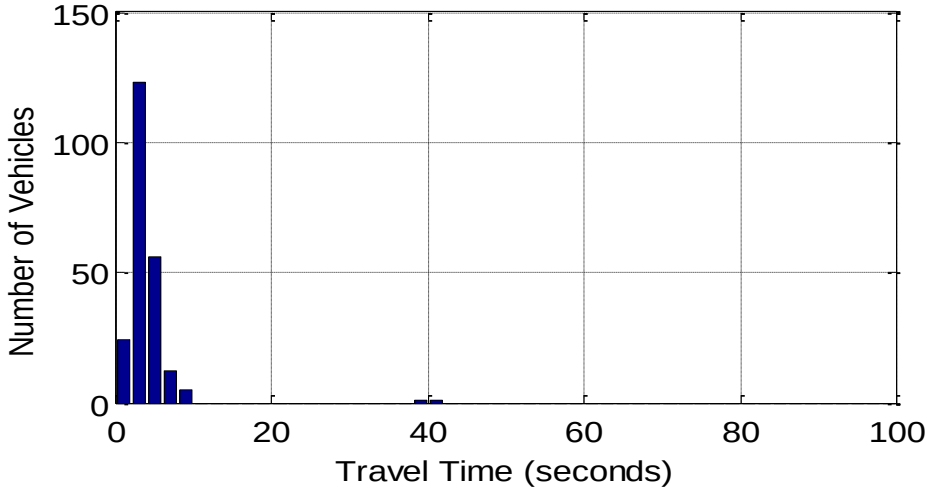


c) True travel time histogram

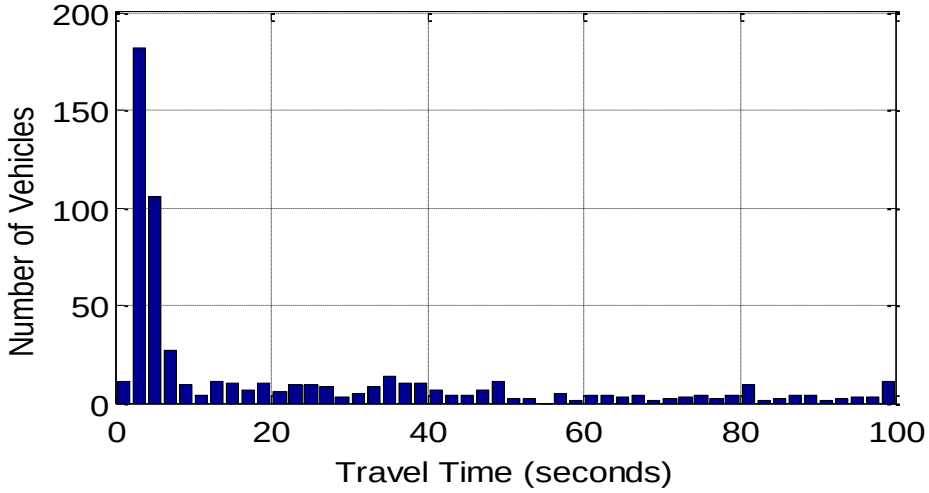
Figure 6.26: Travel time histogram for data set collected on 06/17/2009 Lane B (120)



a) Unfiltered travel time histogram estimate

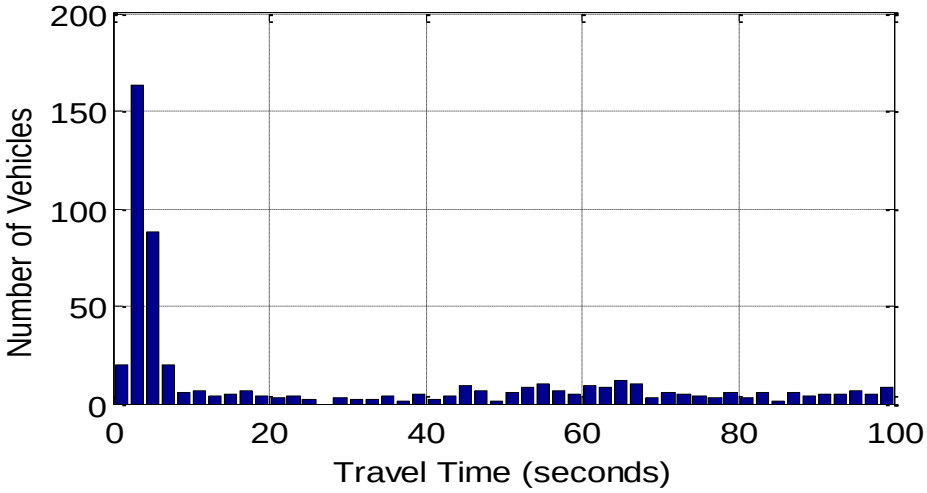


b) Filtered travel time histogram estimate

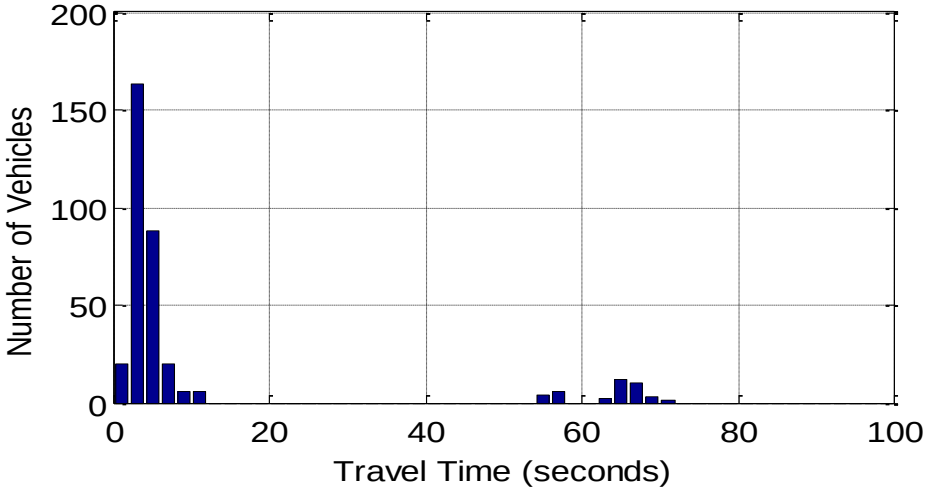


c) True travel time histogram

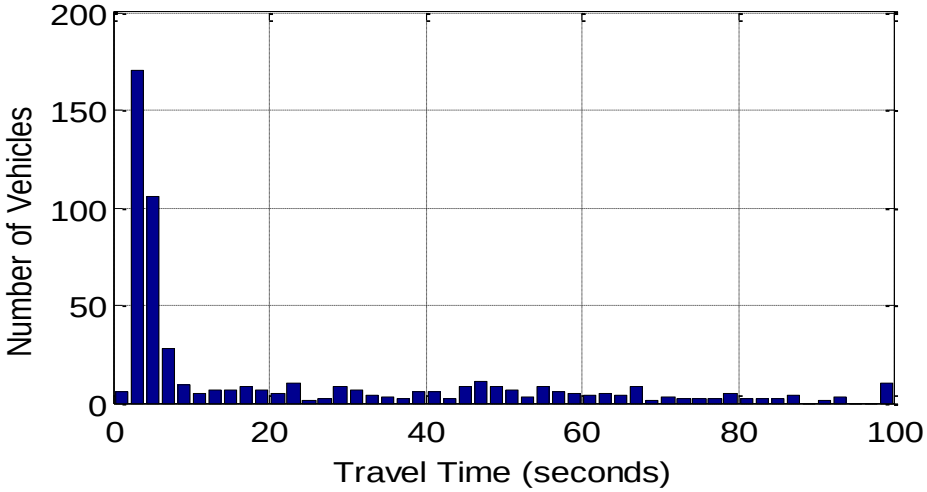
Figure 6.27: Travel time histogram for data set collected on 07/01/2009 Lane A (126)



a) Unfiltered travel time histogram estimate

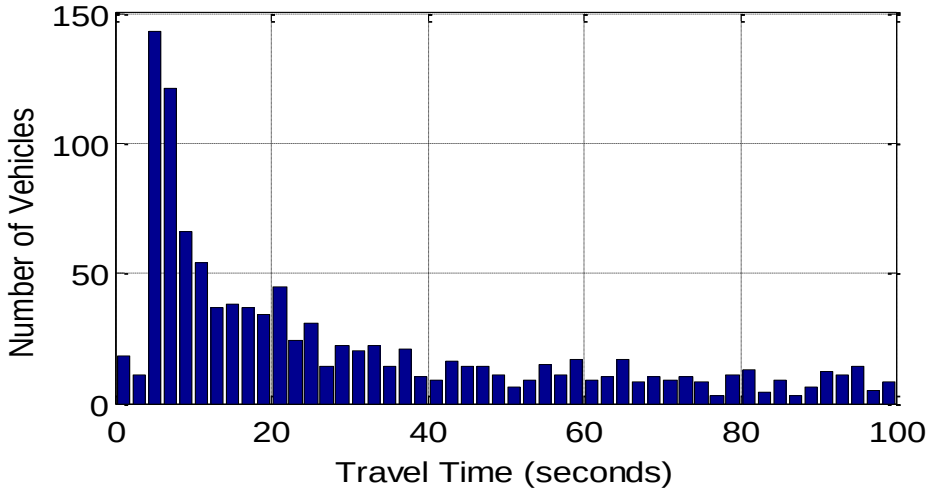


b) Filtered travel time histogram estimate

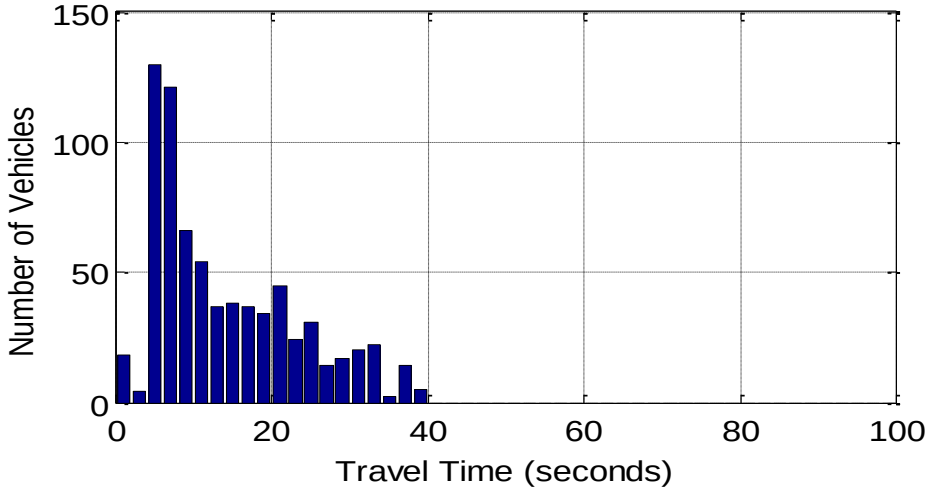


c) True travel time histogram

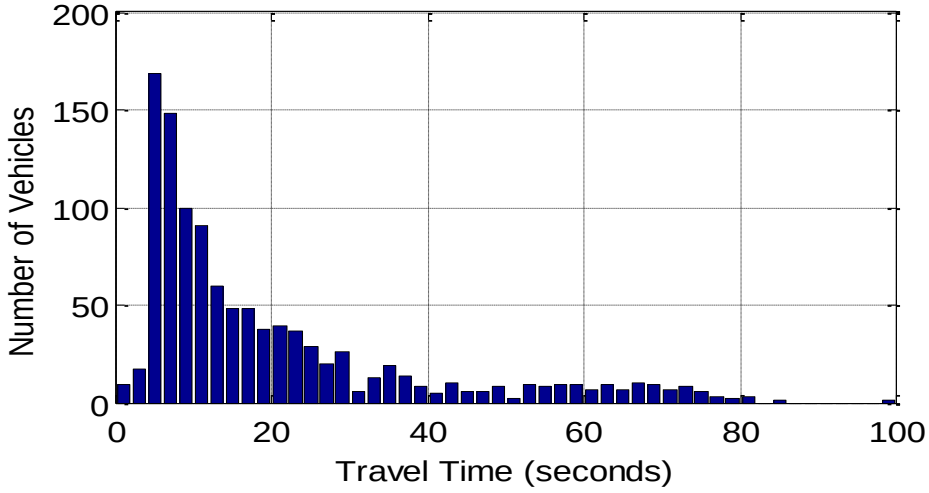
Figure 6.28: Travel time histogram for data set collected on 07/01/2009 Lane B (132)



a) Unfiltered travel time histogram estimate

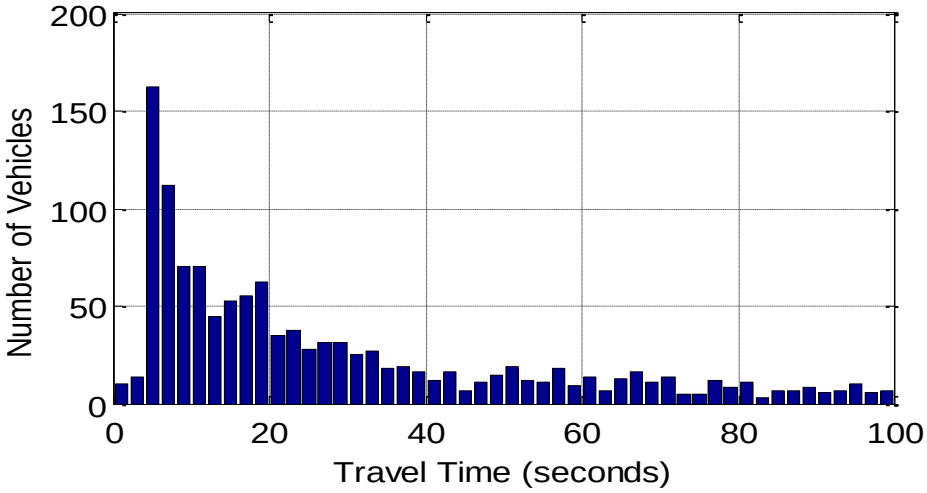


b) Filtered travel time histogram estimate

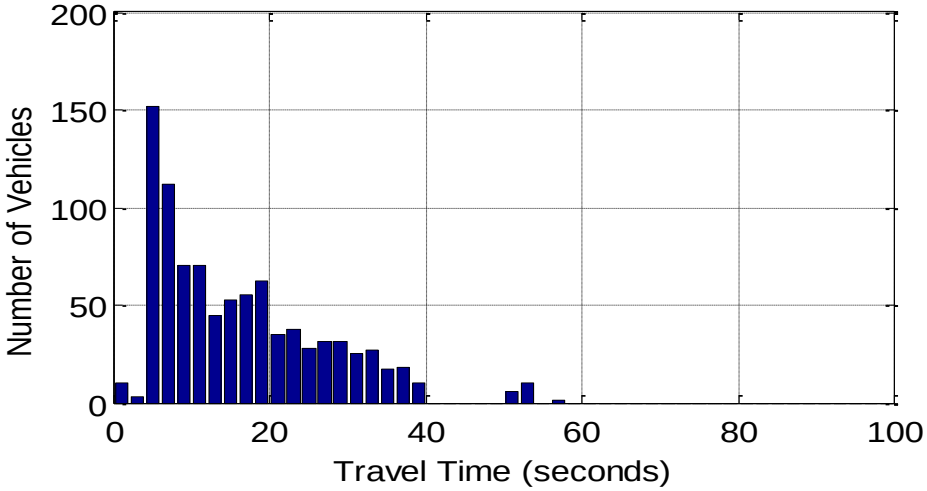


c) True travel time histogram

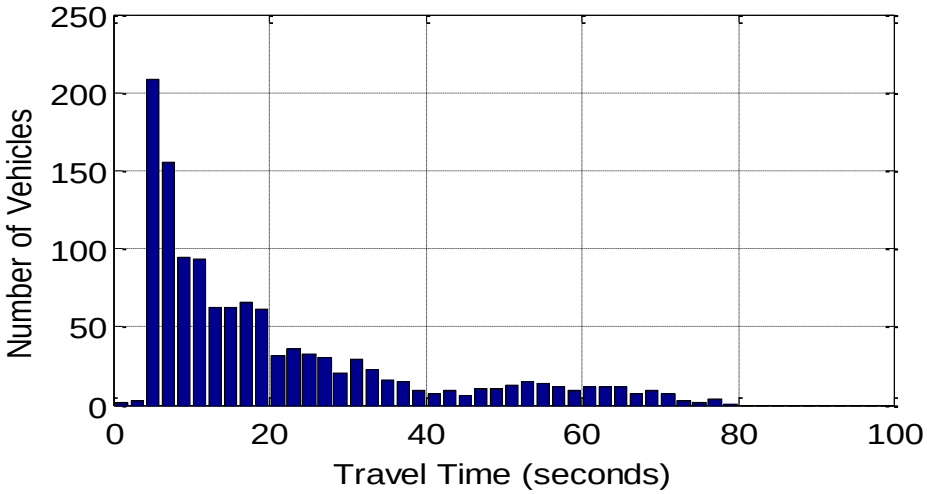
Figure 6.29: Travel time histogram for data set collected on 11/20/2009 Lane A (138)



a) Unfiltered travel time histogram estimate



b) Filtered travel time histogram estimate



c) True travel time histogram

Figure 6.30: Travel time histogram for data set collected on 11/20/2009 Lane B (144)

6.3.1 Acceleration/Deceleration Compensation

The acceleration/deceleration compensation described in Section 4 increases the performance of the signature matching. This improvement can be analyzed both in terms of matching performance and the underlying improvements based on a receiver operating characteristic analysis.

The receiver operating characteristic is a graph commonly used in communication systems to characterize the tradeoff in the system between the true positive rate and the false positive rate. In a good system, there is a high probability of correctly re-identifying a vehicle (the true positive rate) for any chosen probability of creating a false match (the false positive rate).

The algorithm described in Section 4 is not appropriate for analysis by the receiver operating characteristic. Since a travel time is estimated for every passing vehicle, there is no way to adjust the false positive rate. To complete this analysis a thresholding method of detecting matches is now described. If the correlation coefficient between an upstream and downstream signature is above a threshold, it is detected as a correct match. If the correlation coefficient is below the threshold, it is determined to be an incorrect match.

For each threshold, a false positive and true positive rate is calculated. The distributions of correlation coefficients for signatures from the same vehicle and from different vehicles are estimated from their corresponding empirical distributions. The true positive rates are plotted against the false positives rates to generate three curves as shown in Figure 6.31. In the first curve, the correlation coefficients are calculated with no compensation. The second curve uses speed compensation and the third curve uses acceleration/deceleration compensation. The curve shows improvement from the speed compensation and further improvement from the acceleration/deceleration compensation. For example, if a 20% false positive rate is acceptable, the original algorithm with no compensation would correctly match about 40% of the vehicles, while the speed normalization algorithm would correctly match a little over 60% and the acceleration/deceleration compensation algorithm would correctly match about 75% of the vehicles.

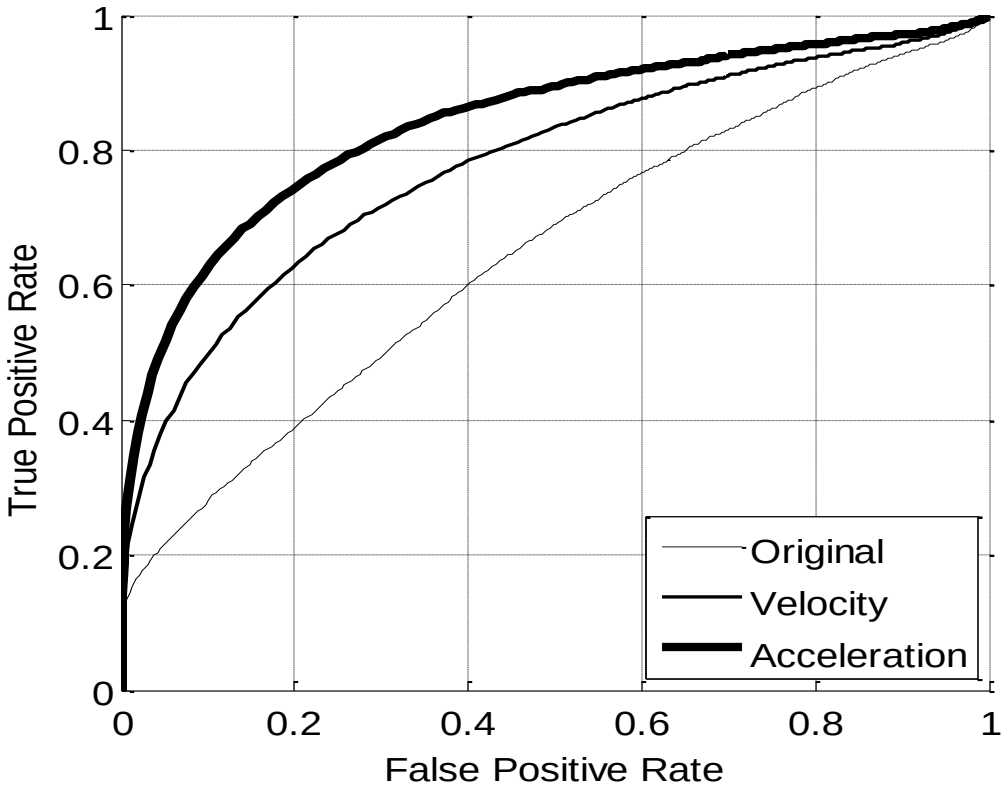


Figure 6.31 Receiver Operating Characteristics for signature matching with various levels of compensation.

While the efficiency of the algorithm and the receiver operating characteristic are informative, the more compelling results are related to how acceleration/deceleration compensation affects the percentage of vehicles correctly matched. This information is summarized in Figure 6.32. The percentage of correct matches is shown for each data set and for each of the three compensation algorithms: none, velocity, and acceleration/deceleration. The acceleration/deceleration compensation yields the highest percentage for every data set. The overall percentage correctly matched for no compensation, velocity compensation, and acceleration/deceleration compensation were 36.8%, 55.1%, and 64.5% respectively.

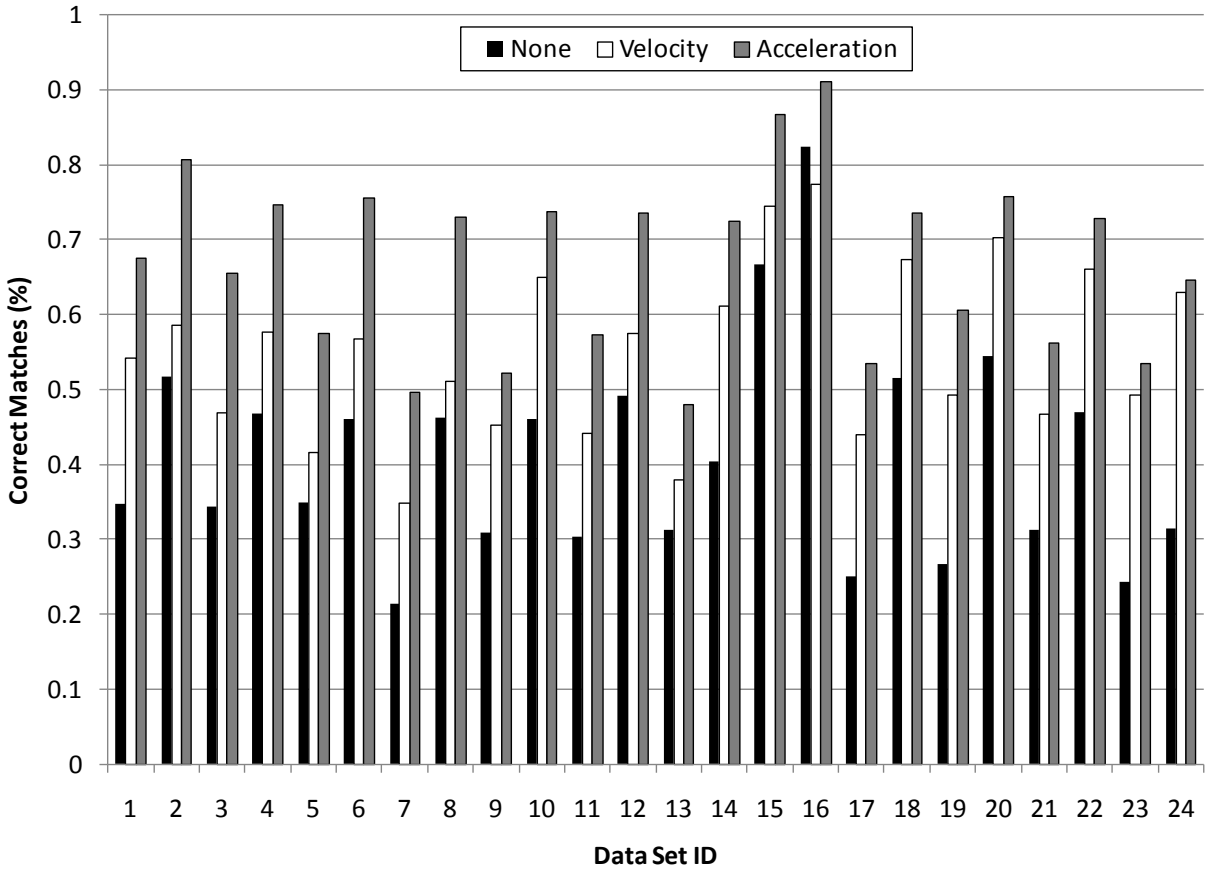
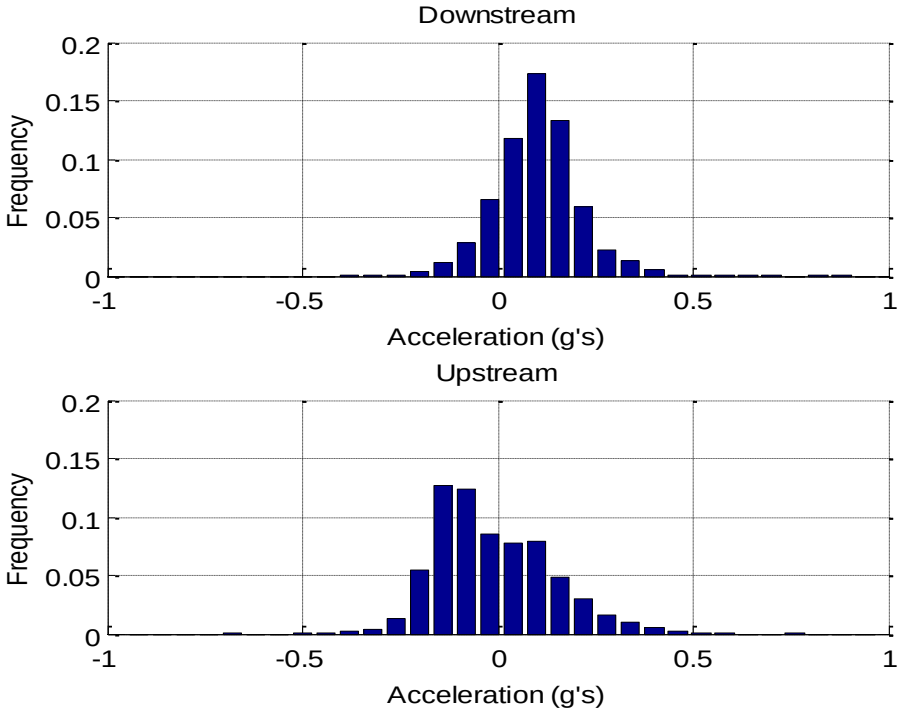
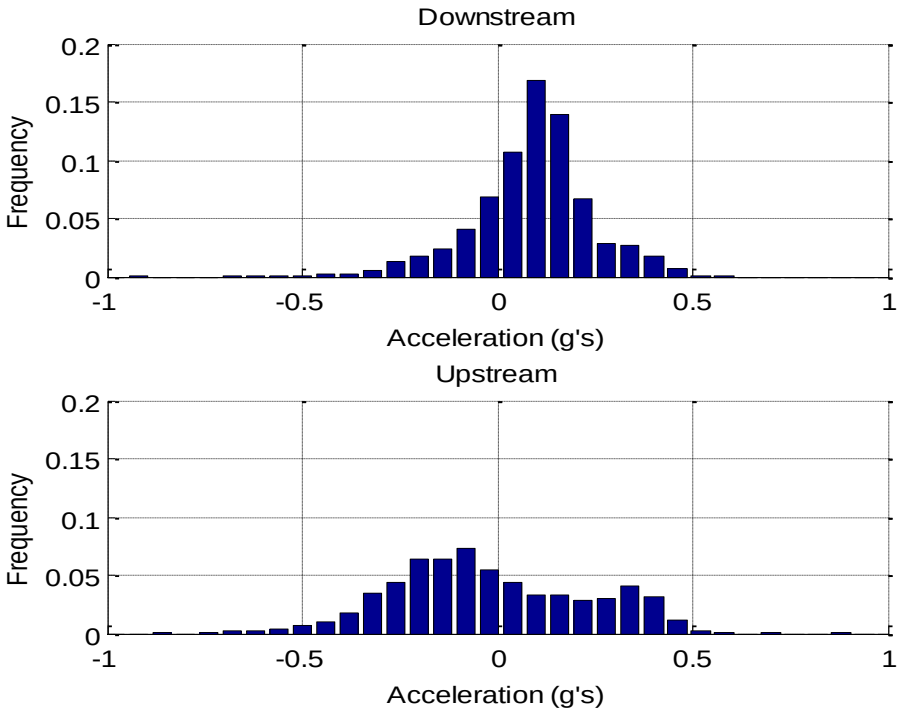


Figure 6.32 Matching rates for each of the data sets with no compensation, velocity compensation, or acceleration/deceleration compensation.

The distributions of accelerations/decelerations are important for measuring the safety of the corresponding intersections. Figure 6.33 shows the histograms of accelerations/decelerations estimated from both intersections. Figure 6.33a shows the histograms from the West Lafayette intersection. This shows that on average most vehicles were accelerating through the downstream speed trap and decelerating through the upstream speed trap. These results are also true for the histograms shown in Figure 6.33b from the Noblesville site. Due to higher operation speeds of the corridor, the acceleration/deceleration from the Noblesville site is more pronounced than the West Lafayette site in both the upstream and downstream data sets. There is a considerable amount of acceleration/deceleration measured at the upstream Noblesville site indicating that many of the vehicles are experiencing a queue that extends 400 ft from the intersection to the advanced detectors.



a) West Lafayette



b) Noblesville

Figure 6.33 Acceleration/deceleration estimate histograms.

6.4 Emerging Technologies

Many similarities exist between the study conducted by Sensys [37] and in this study. Both rely on using a distance (or similarity) metric to choose which vehicle signatures are most likely to correspond to the same vehicle. Unfortunately, because of the anonymous nature of signature matching there is always some matching uncertainty. The data sets in each study present different types of limitations. The Sensys study starts with a much richer data set than is collected in this study. Instead of a single one-axis magnetometer, data is collected from 7 three-axis magnetometers [37]. The positioning of the Sensys sensors is also better for signature matching because they are downstream of the intersection [37]. The Northwestern and Stadium intersection does not have Sensys sensors. The location of these sensors, if installed is shown in Figure 6.34. This downstream location limits queuing, stopping, or accelerating over the sensors. Even if acceleration/deceleration occurs over the sensors that are downstream of the intersection, the distortion will be limited since the effect of acceleration/deceleration distortion is characterized to a first order approximation by the ratio of the vehicle's acceleration/deceleration to its speed. Even if the vehicle is experiencing some acceleration/deceleration, the speed should be sufficient to reduce the effect [38]. The disadvantage of this downstream sensor location is that it cannot be used for controlling the intersection. In the Sensys study, the full dataset is not transmitted to the base station, perhaps to increase battery life. One advantage of microloops and inductive loops is that the full waveforms from the detectors are collected.

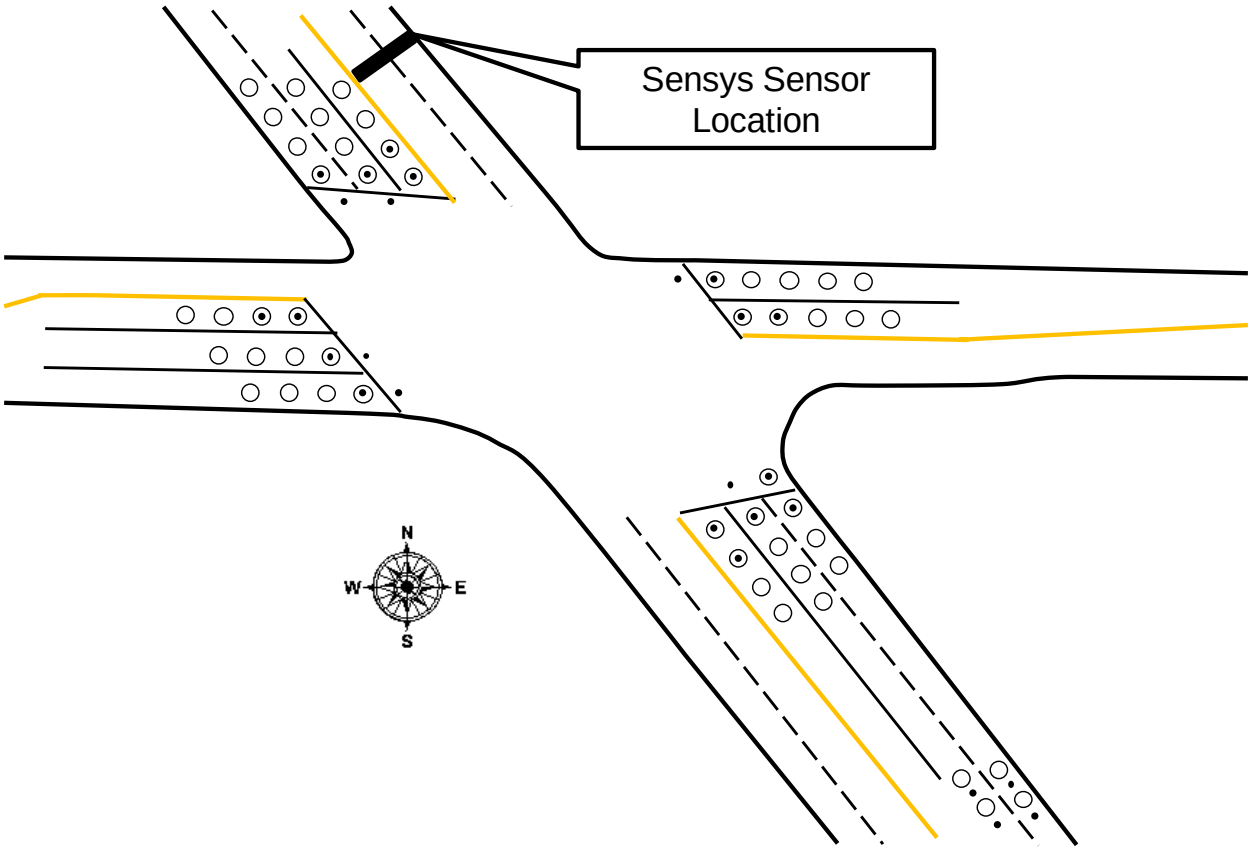


Figure 6.34: Desired Sensys sensor location

Both the algorithm presented in the Sensys study and the matching algorithm described in this report use a distance (or similarity) metric. The metric chosen is greatly influenced by the data available. When using microloop sensors, the amplitude can be greatly affected by installation depth and orientation, which requires using a distance metric that normalizes the amplitude and therefore does not take advantage of this feature to distinguish vehicles. The speed trap configuration of the microloops does allow for speed normalization to produce signatures that are speed invariant. The distance metric described in the Sensys study relies on the amplitude which would imply that the amplitudes of the signatures are more reliable between sensors or normalized in a way that makes the amplitudes sensor invariant.

Another significant difference between the two matching algorithms is with respect to the constraints placed on which vehicles can be matched. Constraints can greatly increase matching percentages, but can also cause systematic errors. The algorithm developed in this project does not constrain the matching except to define the time window of feasible matches. This feasibility window is chosen large enough to include all reasonable travel times. Each downstream vehicle is then matched to the most similar vehicle in the feasibility window. The algorithm described in the Sensys study applies additional constraints to the matching process. One constraint is that each vehicle can only be matched once [37]. While this assumption is reasonable, this constraint may either increase or decrease the number of errors. When signatures are correctly paired, they are eliminated from the list of possible signatures and therefore cannot be incorrectly matched. However, when an incorrect match occurs, this removes two signatures from the candidate signatures that have true pairs that are still in the candidate list. In this case the single erroneous match can lead to further erroneous matches.

The other constraint applied is that vehicles must arrive in the same order at the downstream data collection site as the upstream data collection site. This is a very powerful constraint that narrows the estimated travel time distribution by discarding any correct matches of abnormally fast or slow estimated travel times. The algorithm developed in this study first generates the unconstrained histogram and presents both this histogram along with a statistically filtered histogram.

The final difference between the two approaches is the method of evaluating the travel time estimation histogram. The Sensys study reports matching percentage based on the number of vehicles that were matched (correctly or incorrectly) versus the number of detections. The study described in this report has generated a large ground truth data set which was used to evaluate the matching algorithm. The error percentages presented are the percentage of correct matches of the algorithm compared to ground truth. The Sensys Networks study [39] reports a 47%, 59%, and 51% match rates for the three sub segments in their study and a 49% match rate for the full 0.9 mile segment. The percentage of vehicles correctly matched in this speed trap sensor study with acceleration/deceleration distortion compensation is 65%.

In summary, the Sensys study relies on multiple sensors placed in an unconventional location. Their sensors' signatures consist only of amplitude measurements without any time information. The microloop sensors used in this study have unreliable amplitudes from sensor to sensor, but their speed trap configuration allows the signatures from their sensors to be normalized for

vehicles traveling different speeds. The matching algorithms themselves follow two different philosophies. In this study unconstrained matches are aggregated in a travel time distribution which is then statistically filtered. The Sensys study applies constraints, which assist in higher vehicle match rates, but systematically remove travel time estimates that do not conform to the model that the constraints are based on. Future research could look at evaluating these algorithms in a common test bed with ground truth.

7 Application to a Longer Distance Matching Problem

Signature matching for travel time estimation is more difficult in situations where vehicles can enter and leave a system without traveling over the upstream and downstream detectors. While this is not the case for the data sets discussed thus far, it is likely to be the case when sensors are separated by longer distances (e.g., 0.5 miles to a mile). This is the case for the Noblesville corridor that has been studied in Objective 1 of this project. Using the turning movement data from the Objective 1 part of this study and the matching percentages discussed in this paper, this section discusses the predicted matching percentages in this more complicated corridor scenario.

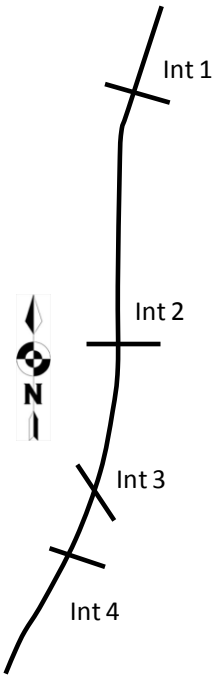


Figure 7.1: Noblesville Corridor

Hypothetical upstream and downstream collection locations are shown on the Noblesville SR 37 corridor in Figure 7.1. The upstream station is north of intersection 1 and the downstream station is south of intersection 4. In this configuration the southbound travel time is measured. Travel times are only possible from vehicles that traverse the entire corridor. Vehicles will enter and leave the corridor at all four intersections. Since the algorithm will attempt to match each vehicle that crosses the downstream location, the maximum possible matching percentage is the percentage of vehicles at the downstream location that also travelled over the upstream location:

$$\text{Match \%} = \frac{N_{Upstream} * p_1 * p_2 * p_3 * p_4}{N_{Downstream}}$$

where p_i is the percentage of southbound vehicles that do not turn at the i^{th} intersection.
The number of vehicles in each southbound approach from 6:00 AM to 9:00 AM is shown in

Table 7-1. For the intersections where the through and right turn lanes are counted together, the right turn volumes are assumed to be the same as the left turn volumes. Using this assumption the following probabilities are calculated: p_1 = 0.8745, p_2 = 0.8437, p_3 = 0.9238, and p_4 = 0.8307. Therefore the number of southbound vehicles that travelled the entire corridor is approximately 56.6% of the vehicles that entered the first intersection from the southbound approach, which is about 1488 vehicles. The downstream detector volume includes the southbound through vehicles, the eastbound right turn vehicles, and the westbound left turn vehicles at Intersection 4. To estimate this volume, the eastbound right turn volume is assumed to be the same as the westbound left turn volume. The total number of vehicles that travel southbound from intersection 4 is then (2681-248)+417*2=3268. The percentage of vehicles that travel southbound from intersection 4 that originated north of intersection 1 is 1488/3268= 45.5%.

The expected travel time through a corridor is the sum of the segment travel times and the delay incurred at each intersection. In this example the intersections are fairly close, therefore the difference between the longest and shortest travel time are assumed to be mainly affected by the delay incurred by an intersection. In a timed corridor, this additional delay is assumed to be at most the longest cycle length. Since the longest cycle length in the corridor is slightly over 100 seconds, the 100 second travel time feasibility window used in Objective 2 of the study is an appropriate choice of a travel time. Using the same travel time window should yield similar matching percentages of vehicles that travelled through the entire corridor. From the results of the ground truth matching, about 65% of these possible matches can be expected to be matched correctly by using acceleration/deceleration and speed normalization. This means that approximately 30% of the volume exiting the southernmost part of the corridor would yield a correct travel time estimated through the corridor. This is approximately 300 vehicles per hour in this 3 hour window.

To estimate the average travel time through a corridor, a sufficient number of estimates are required. The number of estimates required is a function of the distribution of travel times, the acceptable tolerance and the confidence interval. A study to determine the number of travel times for accurate travel time estimation [40] recommends that most accurate estimate of the number of required travel times is $n = \left[\frac{t_\alpha s}{\varepsilon}\right]$ where t_α is a t distribution statistic, s is the sample standard deviation, and ε is the interval half length. For the required sample analysis, travel times are converted to corresponding average speed. For this example, a common confidence interval of 95% is chosen. If the average speed changes by more than 5 mph, it is likely to noticeably affect the travel time. For this reason, 5 mph is used for the interval half-length. A relatively large standard deviation of 10 mph is assumed so that the estimated number of vehicles is conservative. For this example, 19 vehicles are required for an accurate average travel time estimate. This means that a new reliable average travel time estimate can be generated about every 10 minutes."

Table 7-1: Southbound volumes at each intersection

Intersection	Movement	
	Left	Right/Through
Intersection 1	165	2464
Intersection 2	276	3255
Intersection 3	105	2654
Intersection 4	248	2681

8 Conclusion

8.1 Overview

This study evaluates the feasibility of using existing speed trap sensor infrastructure for travel time estimation. It develops an algorithm for signature matching that includes both velocity and acceleration/deceleration compensation to improve matching performance. These algorithms are tested on an extensive dataset from two intersections including pictures, signatures, and contact closures from every vehicle and ground truth matching from the pictures for algorithm performance evaluation.

This study also includes a careful analysis of possible factors that can degrade the quality of sensor matching including crosstalk and sensor placement. The algorithm is presented in its most basic form for use with any sensor with sufficiently detailed information (e.g., inductive loops, magneto-inductive sensor, magneto-resistive sensors, acoustic sensors, radar sensors, laser ranging profilers, etc.).

Two methods of statistical filtering and an analysis of the expected performance along a long corridor are also included. These statistical filtering and analysis methods indicate that signature matching will yield a statistically reliable average travel time estimate about every 5 minutes on the example 4 intersection corridor. An explanation of the tradeoff between estimate bias, sample size, and estimate variance are described and a minimum mean square error metric is used to choose the optimal threshold for this tradeoff.

While the results of this study are promising, other emerging technologies also provide ways to estimate travel time. Crowd sourcing methods like Inrix and Bluetooth tracking provide very reliable travel time estimates. Since these alternatives are already commercialized, they can be easily deployed. The advantage to signature matching is the use of the existing infrastructure and more frequent travel time estimates.

The following section summarizes the signature matching travel time estimation algorithm. Future works are then discussed.

8.2 Signature Matching for Travel Time Estimation

This study has described and analyzed a signature matching algorithm for travel time estimation. Vehicles first cross the upstream pair of sensors and then the downstream pair of sensors. Energy and speed normalized signatures are generated from the raw data streams from each pair of sensors. When a vehicle crosses the downstream pair, its normalized signature is compared with a list of feasible upstream normalized signatures. The most similar upstream signature is chosen as the correct match. The only constraint on the list of possible upstream signatures is that it falls within the feasibility time window. This window is chosen large enough that it includes any reasonable travel times from the upstream pair of sensors to the downstream pair of sensors.

The correlation coefficient is used as the similarity metric to decide which signatures are the most similar. This was motivated by a communications model of the vehicles as a random signal embedded in independent and identically distributed Gaussian noise. Based on this model, the estimate that minimizes the probability of error is the match with the highest correlation coefficient. Statistical processing is used on the histograms of the travel time estimates to eliminate travel times that are likely to be in error.

Generating the normalized signatures before computing the correlation coefficient has two main steps: segmentation and normalization. The segmentation portion relies on the detector card to identify when vehicles are present, but this segmentation tends to remove useful data from the beginning and end of each signature. This segmentation is therefore modified to include each vehicle's complete signature while not including parts of the data stream from other vehicles.

Speed normalization greatly increases the correlation coefficient between signatures from the same vehicle by allowing all the signatures to be analyzed as if they were collected from vehicles travelling the same speed. This is an added advantage over some other matching algorithms in that the length of each vehicle is part of the data set that can be used to distinguish vehicles. The energy normalization is performed because the amplitude of the same vehicle's signature can vary between sensors and therefore is not data that should be used as part of the similarity metric.

Through experimental results, the matching algorithm described in this report has been shown to be useful for travel time estimation but not reliable enough to confidently match individual vehicles. The data that has been collected in the multimedia database can be used to explore the options discussed in the next section to enhance the matching percentage.

8.3 Future Research

The most beneficial further research would attempt to improve the quality of the collected signatures. Possible research includes a segmentation algorithm that does not rely on the presence indication from the detector card. This segmentation does sometimes have false or double detections. This leads to the discrepancies shown in Table 6-1. In congested conditions, it can be very difficult to tell a long vehicle with an interesting signature from two short vehicles.

One possible algorithm could use an energy window to detect vehicles, but this algorithm may detect significant energy in the signal when no vehicle is present if the nominal frequency of the detector card is not accurately estimated. Another possible segmentation algorithm could rely on a sliding window of the standard deviation. The variance based segmentation algorithm will not detect an inaccurate estimate of the nominal frequency as a vehicle because the error in the estimate of zero has no significant variance. The downside of this segmentation algorithm is that it tends to break a single vehicle in to two signatures if there are areas in the vehicle's signature that are somewhat constant.

Segmentation is fairly simple if the sensors are placed in a location that will see free flow traffic, but the close spacing of vehicles due to queuing at the intersection can make the segmentation very difficult. A more advanced detection algorithm would use both the lead and lag sensors jointly to estimate the presence of a vehicle.

The signature matching algorithm implemented in this project is desirable because of its simplicity. This simplicity means that it is easily transferable between locations and types of sensors. More complicated algorithms that include models of traffic flow or ordering information can greatly increase correct matching probability, but their complexity can also create a bias from the models and constraints used. The effect of these additional constraints should be characterized.

The current model is based on the assumption that the signatures are corrupted by additive Gaussian noise. The data shows that there are additional reasons that the signatures from the same vehicle are different when received by different sensors. A model that accounts for these additional differences should be developed. This may include distortion due to a vehicle's lateral deviation. Algorithms that use feature detection methods from speech and image processing could be useful.

The k-means clustering algorithm is a good way to separate two Gaussian distributions. It does not seem to work as well separating more complicated distributions of travel times. In the simplest model, it is assumed that the erroneous travel times are uniform and that the true travel times are Gaussian. A more complicated model includes another uniform histogram in the true travel times to simulate the delay experienced by vehicles being caught at a red light at the intersection. In order to keep more of the true travel times, a more sophisticated algorithm can be developed.

The existing database has thousands of signatures from various vehicles. In that sense, this is a very rich data set. However, the travel times are only from two locations. These signatures can be used to simulate many different traffic scenarios. First, probabilistic distributions of travel times could be tested. This could also be used to test the ability of the matching algorithm to track vehicles through various paths between two endpoints. Traffic simulators can be used to generate the arrival times of various vehicles at stations throughout a network. Signatures can then be applied to the vehicles at random to generate any test scenario.

9 Appendix

An overview of the data collection infrastructure is shown in Figure 9.1. Each data collection site includes two lanes with six sensors (four microloops and two inductive loops) in each lane. The lanes carrying northbound traffic and are labeled A and B starting from the lane adjacent to the left turn lane and increasing alphabetically toward the curb. In each lane the sensors are numbered 1, 2, 5, 6, 7 and 8 for a total of twelve sensors: NA_M1, NA_M2, NA_M5, NA_L6, NA_M7, NA_L8, NB_M1, NB_M2, NB_M5, NB_L6, NB_M7, and NB_L8. The numbering is used so that the microloop numbering is consistent with other sensors at this data collection sites.

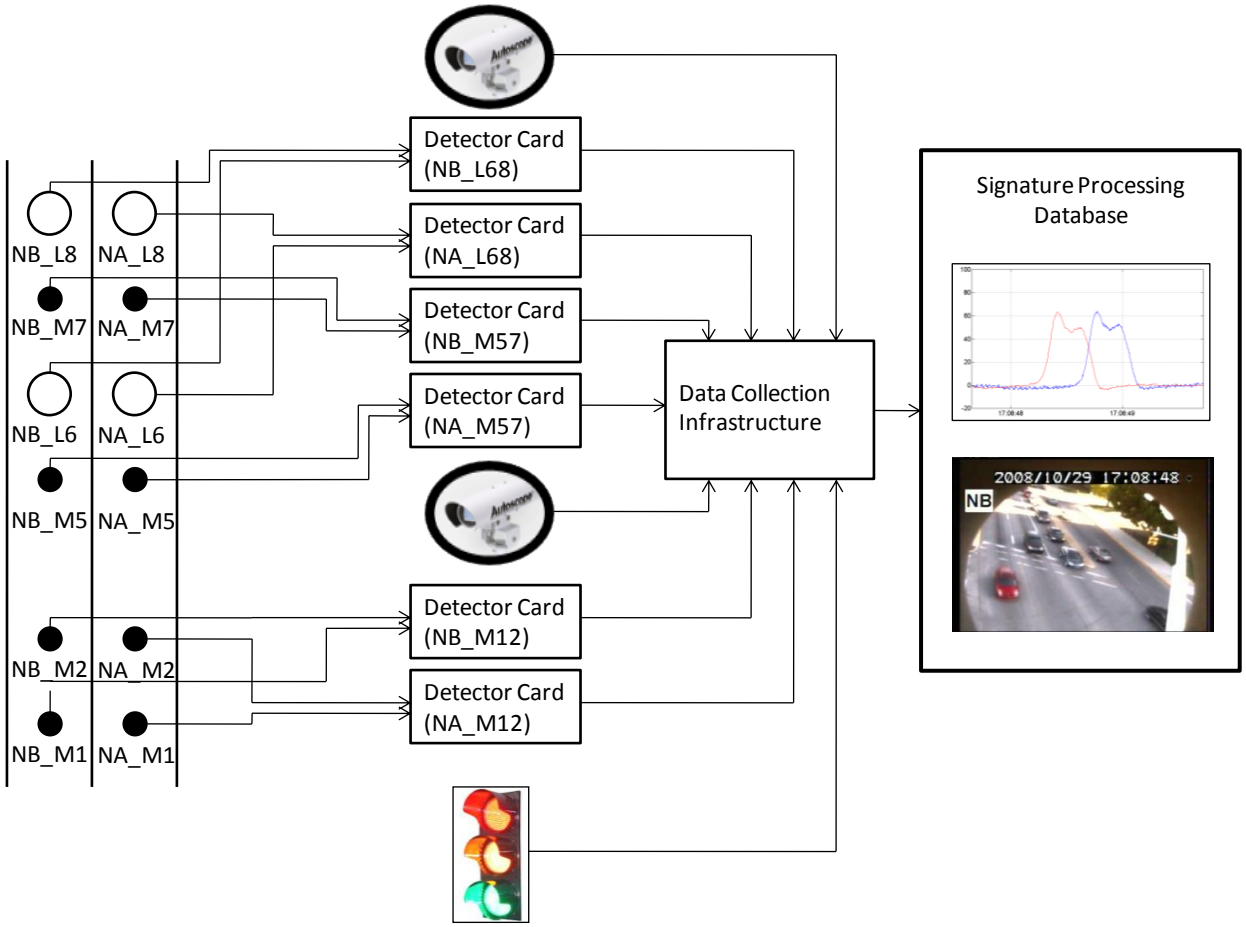


Figure 9.1: Database Infrastructure.

There are four types of data collected about the vehicles that pass through the data collection site:

- the normalized frequency change of the loops due to a vehicle (signatures),
- pictures of the vehicles,
- the status of the intersection including vehicle detections,
- and phase information.

The database is designed for evaluating vehicle re-identification algorithms. Datasets in some studies become less useful because notes about the data set are lost. For this reason, all meta-data required to interpret the raw data is stored in the database. Also, data is stored in the most fundamental unprocessed form. Since the data sets are large, it is important that the raw data tables are both efficient with respect to the data's size and designed such that the data is easily retrievable.

The relational diagram for the database is shown in Figure 9.2. The relationship diagram builds from the bottom to the top in three major groupings:

- Infrastructure Tables
- Data Tables
- Analysis Tables

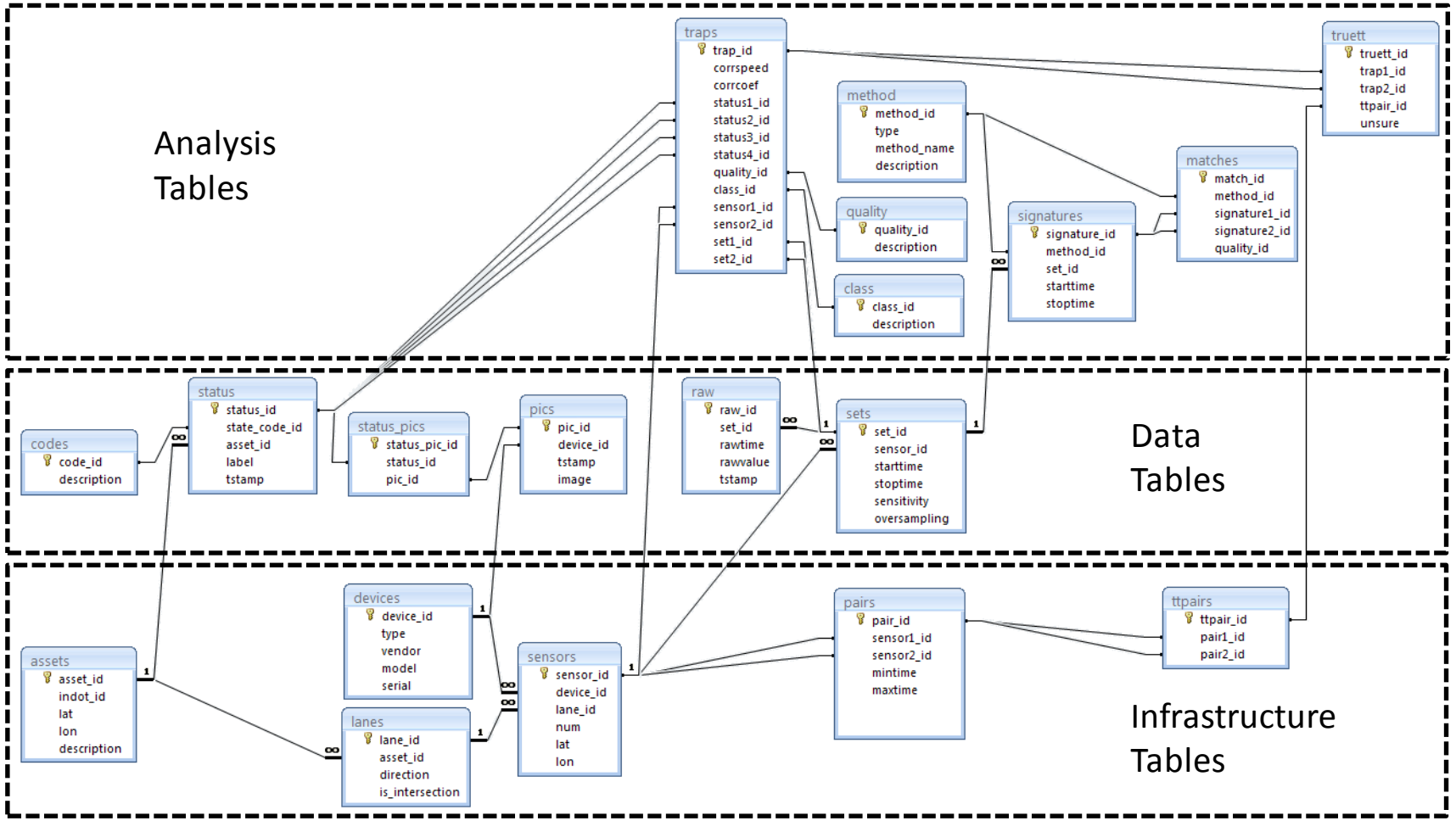


Figure 9.2: Database Diagram

9.1 Infrastructure Tables

The infrastructure tables record information about the collection devices at the collection site and the site itself. The data tables hold the raw data from the magneto-inductive sensors, the detections and intersection phase information and the pictures of the vehicles. The analysis tables hold the results of algorithms applied to the database.

Example infrastructure tables are shown in Table 9-1 to Table 9-6. The assets table contains information about the intersection or highway site including its location. Each lane has an entry in the lanes table that records its direction, its label, and which phase it corresponds to. The sensors table refers to the microloops, inductive loops and cameras. This table tells the location of the sensor, the type of device, which video source provides ground truth and which data collection site the sensor is located by linking to other tables. For example in Table 9-4, sensor_id 6 is labeled NA_M7 and is located at the latitude and longitude coordinates 40.430914° North and 86.913506° West. Information about the device itself is found in the example devices table shown in Table 9-3 in the row with device_id 3. The sensors table also shows that this sensor is linked to video_id 8. Looking at the devices table reveals that this is an Autoscope camera.

The last two infrastructure tables describe the logical relationship between sensors. The pairs table designates which sensors are paired as lead/lag sensors in a speed trap. An example of this table is shown in

Table 9-5. The TTPairs table shows which speed traps are being evaluated for upstream and downstream locations for travel time estimation. An example of this table is shown in Table 9-6. The description of each field in all the infrastructure tables is shown in Table 9-7 to Table 9-12.

Table 9-1: Example Assets Table

asset_id	lat	lon	Description	is_intersection
1	40.43134	-86.91414	Northwestern and Stadium	True
2	40.045656	-85.993703	Noblesville 32 and 37	True

Table 9-2: Example Lanes Table

lane_id	asset_id	direction	label	phase
1	1	N	A	2
2	1	N	B	2
3	2	N	A	2
4	2	N	B	2

Table 9-3: Example Devices Table

device_id	type	Vendor	model	Serial
1	microloop	3M	C800 V1.3	0044606CG2F
2	microloop	3M	C800 V1.3	0044646CG2F
3	microloop	3M	C800 V1.3	0044617CG2F
4	microloop	3M	C800 V1.3	0044616CG2F
5	loop	3M	C800 V1.3	0044662CG2F
6	loop	3M	C800 V1.3	0044607CG2F
7	camera	Autoscope	AIS Camera	00000000002
8	camera	Autoscope	AIS Camera	00000000003

Table 9-4: Example Sensors Table

sensor_id	device_id	video_id	lane_id	ch	label	lat	lon
1	1	7	1	1	NA_M1	40.431251	-86.913787
2	1	7	1	2	NA_M2	40.431203	-86.913751
3	2	7	2	1	NB_M1	40.431239	-86.913836
4	2	7	2	2	NB_M2	40.431222	-86.913814
5	3	8	1	1	NA_M5	40.430983	-86.913575
6	3	8	1	2	NA_M7	40.430914	-86.913506

Table 9-5: Example Pairs Table

pair_id	sensor1_id	sensor2_id	mintime	maxtime
1	2	1	0	20
2	4	3	0	20
3	6	5	0	20
4	8	7	0	20

Table 9-6: Example TTPairs Table

ttpair_id	pair1_id	pair2_id
1	2	1
2	4	3
3	6	5
4	8	7

Table 9-7: Assets Table Description

Attribute	Data Type	Description
asset_id	integer	Unique asset identifier
lat	double precision	Latitude
lon	double precision	Longitude
description	text	Describes the location of the asset
is_intersection	Boolean	True if asset is an intersection False if asset is a highway collection site

Table 9-8: Lanes Table Description

Attribute	Data Type	Description
lane_id	integer	Unique lane identifier
asset_id	integer	Link to assets table
direction	text	N,S,E,W direction
label	text	Name for lane
phase	integer	Phase of intersection associated with lane

Table 9-9: Devices Table Description

Attribute	Data Type	Description
device_id	integer	Unique id for a device
type	text	Microloop/Inductive Loop
vendor	text	Vendor device was purchased from
model	text	Model number of device
serial	text	Serial number

Table 9-10: Sensors Table

Attribute	Data Type	Description
sensor_id	integer	Unique sensor identifier
device_id	integer	link to device table
video_id	integer	link to device table (for the camera)
lane_id	integer	link to lanes table
ch	integer	Channel on device that sensor is connected to
label	text	Name for sensor
lat	double precision	Latitude
lon	double precision	Longitude

Table 9-11: Pairs Table

Attribute	Data Type	Description
pair_id	integer	Unique id for travel time pairing
sensor1_id	integer	Link to sensor table for first sensor being paired
sensor2_id	integer	Link to sensor table for second sensor being paired
mintime	interval	Minimum amount of time to travel from one sensor to the other
maxtime	interval	Maximum amount of time to travel from one sensor to the other

Table 9-12: TTPairs Table

Attribute	Data Type	Description
ttpair_id	integer	Unique id for travel time pairing
pair1_id	integer	Link to pairs table for the upstream pair of sensors
pair2_id	integer	Link to pairs table for the downstream pair of sensors

9.2 Data Tables

The data tables are where all of the raw data is stored. The status, pics, and raw tables hold the data while the codes, status_pics, and sets tables hold information that helps to interpret the data in the first three tables. An example plot of the raw data waveforms from NA_M1 (the lag sensors) and NA_M2 (the lead sensor) is shown in Figure 9.3. The sets table stores the start time, end time, and other information regarding data collected in the raw table. An example of the sets table is shown in Table 9-13. Table 9-14 and Table 9-15 show small samples of the raw data table from the NA_M1 and NA_M2 sensors. These small samples correspond to the areas specified by the arrows in Figure 9.3. The lead data set has set_id 4 and the lag data set has set_id 3.

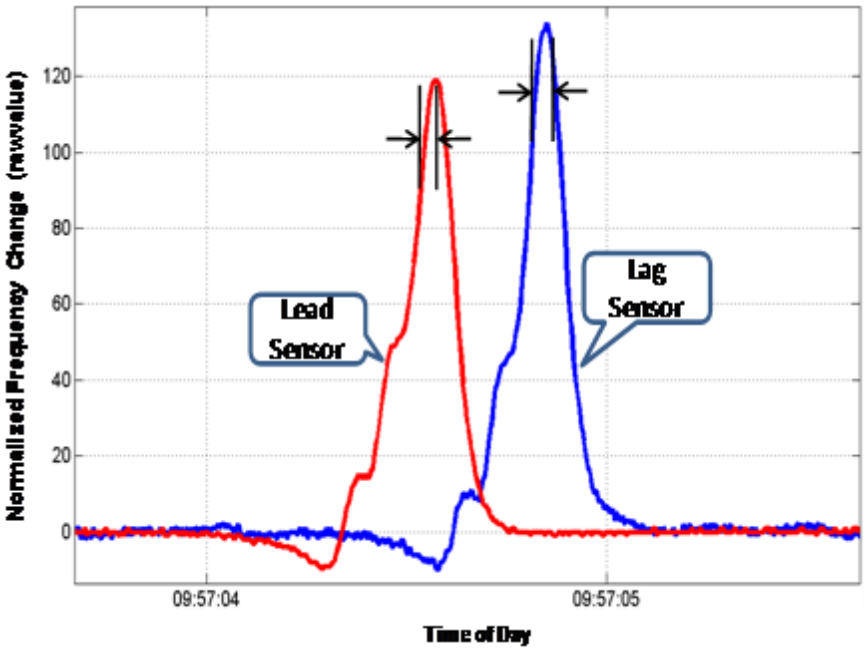


Figure 9.3: Sample raw magnetic signature for lead and lag sensors

Table 9-13: Example Sets Table

set_id	sensor_id	starttime	stoptime	sensitivity	oversampling
1	1	2008-11-10 09:56:50	2008-11-10 10:06:54	3300	M
2	2	2008-11-10 09:56:50	2008-11-10 10:06:54	3300	M
3	3	2008-11-10 09:56:53	2008-11-10 10:06:56	3300	M
4	4	2008-11-10 09:56:53	2008-11-10 10:06:56	3300	M

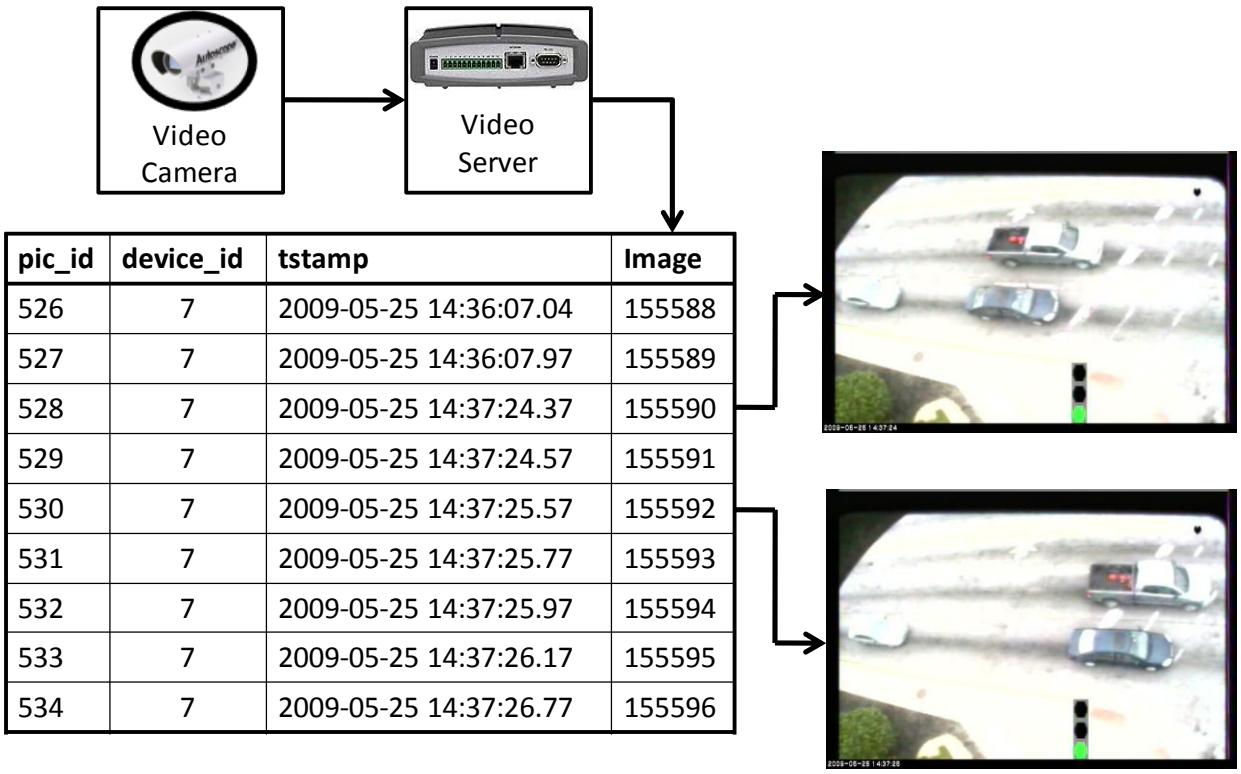
Table 9-14: Example raw table from lead sensor between vertical lines

raw_id	set_id	rawtime	rawvalue	tstamp
366880	4	15804	94	2008-11-10 09:57:04.539
366881	4	15809	100	2008-11-10 09:57:04.544
366882	4	15814	106	2008-11-10 09:57:04.549
366883	4	15819	111	2008-11-10 09:57:04.554
366884	4	15824	115	2008-11-10 09:57:04.559
366885	4	15830	118	2008-11-10 09:57:04.565
366886	4	15835	119	2008-11-10 09:57:04.570
366887	4	15840	119	2008-11-10 09:57:04.575

Table 9-15: Example raw table from lag sensor between vertical lines

raw_id	set_id	rawtime	rawvalue	tstamp
249081	3	16085	112	2008-11-10 09:57:04.82
249082	3	16090	119	2008-11-10 09:57:04.825
249083	3	16095	125	2008-11-10 09:57:04.83
249084	3	16101	129	2008-11-10 09:57:04.836
249085	3	16111	132	2008-11-10 09:57:04.841
249086	3	16116	132	2008-11-10 09:57:04.846
249087	3	16121	134	2008-11-10 09:57:04.851
249088	3	16126	131	2008-11-10 09:57:04.856

Figure 9.4 illustrates how the pictures of vehicles are stored in the database. The video server saves the images of the vehicles both when they arrive at the sensor and when they leave the sensor. The images are then loaded into the database with a device_id to identify the video source that captured the picture and a timestamp of when the picture was taken. The table in Figure 9.4 includes pic_id 528 where the vehicle is travelling over the lead loop and pic_id 530 where the vehicle is travelling over the lag loop.



pic_id	device_id	tstamp	Image
526	7	2009-05-25 14:36:07.04	155588
527	7	2009-05-25 14:36:07.97	155589
528	7	2009-05-25 14:37:24.37	155590
529	7	2009-05-25 14:37:24.57	155591
530	7	2009-05-25 14:37:25.57	155592
531	7	2009-05-25 14:37:25.77	155593
532	7	2009-05-25 14:37:25.97	155594
533	7	2009-05-25 14:37:26.17	155595
534	7	2009-05-25 14:37:26.77	155596

Figure 9.4: Picture Capture Procedure

Each picture corresponds to either a vehicle arriving or departing the detection zone of a sensor. These vehicle detections are stored in the status table. Table 9-16 shows an example status table. The example state code table shown in

Table 9-17 helps interpret the state_code_id column of the status table. For example a state_code_id of 9 means that a vehicle has entered the detection zone of a sensor and state_code_id 8 means that the vehicle has left the detection zone. The status_pics table links the status changes in the status table to pictures stored in the pics table. An example status_pics table is shown in Table 9-18. The fifth row of this example table has pic_id 528 and status_id 1947. This means that the top picture in Figure 9.4 is linked to the status_id 1947. Table 9-16 shows that status_id 1947 is from NA_M2 with state_code_id 9. This state_code_id means that a vehicle has been detected at the sensor. Another thing to notice about the status_pics table is that the first two rows have the same pic_id. The camera only takes a picture 10 times per second. Therefore, if two events occur close together, the same picture is used for two different events. Descriptions of all the fields of the Data Tables are found in Table 9-19 to Table 9-24.

Table 9-16: Example Status Table

status_id	asset_id	state_code_id	label	tstamp
1941	1	8	NB_M7	2009-05-25 14:37:09.564
1942	1	9	NB_M5	2009-05-25 14:37:09.564
1943	1	8	NB_L6	2009-05-25 14:37:09.965
1944	1	8	NB_M5	2009-05-25 14:37:10.264
1945	1	8	N_R	2009-05-25 14:37:22.166
1946	1	9	N_G	2009-05-25 14:37:22.166
1947	1	9	NA_M2	2009-05-25 14:37:24.266
1948	1	9	NB_M2	2009-05-25 14:37:24.515
1949	1	9	NA_M1	2009-05-25 14:37:25.566
1950	1	9	NB_M1	2009-05-25 14:37:25.716

Table 9-17: Example State Code Table

state_code_id	Description
0	Phase Off
1	Phase Green
2	Phase Yellow
3	Phase Red Clear
4	Ped Off
5	Ped Walk
6	Ped Clear
8	Detector Off
9	Detector On

Table 9-18: Example StatusPics Table

status_pic_id	pic_id	status_id
1777	2333	1941
1778	2333	1942
1779	2334	1943
1780	2335	1944
1781	528	1947
1782	529	1948
1783	530	1949
1784	531	1950

Table 9-19: Raw Table

Attribute	**Data Type**	**Description**
raw_id	integer	Unique id for raw data point
set_id	integer	Link to sets table
rawtime	integer	Millisecond time from detector card in datastream
rawvalue	integer	Raw measured value from detector card in datastream
tstamp	timestamp	Time data was received by collection computer

Table 9-20: Sets Table

Attribute	Data Type	Description
set_id	integer	Unique data set identifier
sensor_id	integer	Link to sensor table
starttime	timestamp	Start time of data collection
stoptime	timestamp	Stop time of data collection
sensitivity	integer	Sensitivity setting on 3M card for data collection
oversampling	text	Oversampling setting on 3M card for data collection

Table 9-21: Pics Table

Attribute	Data Type	Description
pic_id	integer	Unique picture identifier
device_id	integer	Link to device table
tstamp	timestamp	Time picture was taken
image	oid	Picture of vehicle

Table 9-22: Status Table

Attribute	Data Type	Description
status_id	integer	Unique id for status
asset_id	integer	Link to assets table
state_code_id	integer	Link to state_code table
label	text	String that identifies the type of signals received
tstamp	timestamp	Time event occurred

Table 9-23: State Codes Table

Attribute	Data Type	Description
state_code_id	integer	Unique id for state_codes
description	text	Description of code

Table 9-24: Status Pics Table

Attribute	Data Type	Description
status_pic_id	integer	Unique id for status pic pairs
pic_id	integer	Link to pics table
status_id	integer	Link to status table

9.3 Analysis Tables

The analysis tables help to group information to make it more easily accessible. They also store ground truth travel time information. Each row in this table represents a vehicle travelling over a speed trap. It has links to the status table to record which vehicle detections on the lead sensor correspond to the same vehicle on the lag sensor. There are also links to the sets table to access the signatures collected from the vehicle. Several fields were manually filled for each vehicle including its class (car, bike, or other) and its quality. The quality field designates if the lead/lag pairing was done correctly and whether or not the vehicle maintained a constant speed throughout the speed trap. These quality and class designations are stored as integers that link to the quality and class tables. Table 9-25 to Table 9-27 are descriptions of these three tables. Table 9-28 and Table 9-29 show the quality table and the class table respectively.

Table 9-25: Traps Table Description

Attribute	Data Type	Description
trap_id	integer	Unique id for each vehicle over the speed trap
corrspeed	double precision	speed calculated from correlation method
corrcoef	double precision	correlation coefficient for speed calculation
status1_id	integer	link to status table for vehicle detected at lead sensor
status2_id	integer	link to status table for vehicle no longer detected at lead sensor
status3_id	integer	link to status table for vehicle detected at lag sensor
status4_id	integer	link to status table for vehicle no longer detected at lag sensor
quality_id	integer	link to quality table
class_id	integer	link to class table
sensor1_id	integer	link to sensors table for lead sensor
sensor2_id	integer	link to sensors table for lag sensor
set1_id	integer	link to sets table for lead sensor data
set2_id	integer	link to sets table for lag sensor data

Table 9-26: Quality Table Description

Attribute	Data Type	Description
quality_id	integer	Unique id for each quality type
description	text	Description of each quality type

Table 9-27: Class Table Description

Attribute	Data Type	Description
class_id	integer	Unique id for each class of vehicles
description	text	Description of vehicles in the class

Table 9-28: Quality Table

quality_id	description
1	Unmarked
2	Good
3	Stopped at Stop Bar
4	Lane Issues
5	Invalid Match
6	Poor Picture Quality

Table 9-29: Class Table

quality_id	description
1	Unmarked
2	Car
3	Bike
4	Other

The TrueTT table stands for true travel times. This table holds the ground truth information about which upstream vehicle matches which downstream vehicle. The ttpair_id field designates which travel time pairing the travel time is a part of (which lane). The unsure flag allows the person executing the ground truthing to designate that the video is unclear for that particular travel time pairing. The details of the TrueTT table are shown in Table 9-30.

Table 9-30: TrueTT Table Description

Attribute	Data Type	Description
truett_id	integer	Unique id for each ground truth travel time
trap1_id	integer	link to trap table for upstream vehicle
trap2_id	integer	link to trap table for downstream vehicle
ttpair_id	integer	link to ttpair table
unsure	boolean	flag for a possibly incorrect ground truth travel time

Each row in the method table designates a method of either segmenting the data streams to create signatures or pairing signatures to create matches. The signatures and matches tables store the results of the algorithm. The matches and TrueTT tables can then be compared to see whether the travel time estimation algorithm matched the correct pair of signatures.

10 References

1. Palen, J., "The need for surveillance in intelligent transportation systems," Intellimotion, Vol. 6, No. 1, pp. 1-3, 10, 1997.
2. G. Urbanek and R. Rogers, "Alternative Surveillance Concepts and Methods for Freeway Incident Management", Volume 1, Executive Summary. Federal Highway Administration, 1978.
3. B. Coifman, "Estimating travel times and vehicle trajectories on freeways using dual loop detectors," Transportation Research Part A: Policy and Practices, Vol. 36, No. 4, pp. 351–364, 2002.
4. Louu Shen and Mohammed Hadi, "Estimation of Segment Travel Time Based on Point Traffic Detector Measurements," In Proceedings of the 87th Annual Meetings of the Transportation Research Board, 2009 Washington, DC
5. Nale Zhao, Lei Yu, Hui Zhao, Jifu Guo and Huimin Wen, "Analysis of Traffic Flow Characteristics on ring-road expressways in Beijing using FCD and RTMS Data," In Proceedings of the 87th Annual Meetings of the Transportation Research Board, 2009 Washington, DC.
6. Louu Shen and Mohanned Hadi, "Freeway Travel Time Prediction with Dynamic Neural Networks," In Proceedings of the 88th Annual Meeting of the Transportation Research Board, January 2010 Washington, DC.
7. Ruimin Li, Geoffrey Rose, and Majid Sarvi, "Evaluation of Speed-Based Travel Time Estimation Models," Journal of Transportation Engineering, vol. 132, pp. 540-547, July 2006
8. J. Kwon, B. Coifman, and P. Bickel, "Day-to-day travel time trends and travel time prediction from loop detector data," Transportation Research Record, Vol. 1717, No. 15, pp. 120–129, 2000.
9. E. Pfannerstill, "Automatic monitoring of traffic conditions by re-identification of vehicles," in IEEE Second International Conference on Road Traffic Monitoring, (London, UK), February 07-09 1989.
10. Y. Malinovskiy, Y.-J. Wu, Y. Wang, and U. K. Lee, "Field Experiments on Bluetooth-based Travel Time Data Collection," Transportation Research Board Annual Meeting, January 2010.
11. Y. Tanaka and F. Nishimura, "Multiroute travel-time data provision systems operation in Osaka," in Proc. 1994 IEEE Vehicle Navigation and Information Systems, pp. 351–356, 1994.
12. Y. Cui and Q. Huang, "Character extraction of license plates from video," in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (IEEE), pp. 502–507, 1997.
13. J. Schlaich, "Analyses of route choice behavior using mobile phone trajectories," In Proceedings of the 88th Annual Meeting of the Transportation Research Board, January 2010.

14. R. Cayford, "Characteristics of cell phone probe technologies and field testing of very high volume probe system," In Proceedings of the 88th Annual Meeting of the Transportation Research Board, January 2010.
15. B. L. Smith, H. Zhang, M.Fontaine, and M. Green, "Cell phones probes as an ATMS tool," Tech. Rep. Research Report No. UVACTS-15-5-79, Center for Transportation Studies at the University of Virginia, Charlottesville, VA, 2003.
16. S. Quayle, P. Koonce, D. Bullock, and D. DePencier, "Arterial performance measures using media access control readers: Pilot study in Portland, Oregon," In Proceedings of the 88th Annual Meeting of the Transportation Research Board, January 2010.
17. C. M. Day, R. Haseman, H. Premachandra, T. M. Brennan Jr., J. S. Wasson, J. R. Sturdevant, and D. M. Bullock, "Visualization and assessment of arterial progression quality using high resolution signal event data and measured travel time," In Proceedings of the 88th Annual Meeting of the Transportation Research Board, January 2010.
18. Wasson, Jason S., James R. Sturdevant, Darcy M. Bullock, "Real-time travel time estimates using media access control address matching," ITE Journal, vol. 78, no. 6, 2008.
19. Hagnahi, M. Hamedi, K. F. Sadabadi, S. Young, and P. Tarnoff, "Freeway Travel Time Ground Truth Data Collection Using Bluetooth Sensors," In Proceedings of the 88th Annual Meeting of the Transportation Research Board, January 2010.
20. W. Y. Kan, J. V. Krogmeier, and P. C. Doerschuk, "Model-based motion estimation from images sequences with an application to road surveillance," Optical Engineering, Vol. 35, No. 6, pp. 1723–1729, 1996.
21. B. Coifman, D. Beymer, and P.McLauchlan, "A real-time computer vision system for vehicle tracking and traffic surveillance," Transportation Research Part C: Emerging Technologies, Vol. 6, No. 4, pp. 271–288, 1998.
22. H. van Zuylen, F. Zheng, and Y. Chen, "Investigation of Urban Link Travel Time Estimation Based on Field Sparse Probe Vehicle Data," In Proceedings of the 88th Annual Meeting of the Transportation Research Board, January 2010.
23. J.-S. Yang, "Travel time prediction using the GPS test vehicle and Kalman filtering techniques," In Proceedings of the American Control Conference, June 2005.
24. W. Pu, J. Lin, and L. Long, "Estimation of urban street segment travel time using buses as real-time speed probes," In Proceedings of the Annual Meeting of the Transportation Research Board, January 2010.
25. Kornhauser, "Methodology for monitoring highway performance across extensive corridors using probe vehicle GPS data with application to Québec-Windsor corridor," In Proceedings of the 88th Annual Meeting of the Transportation Research Board, January 2010.
26. Christiansen and L. Hauer, "Probing for travel time: Norway applies AVI and WIM technologies for section probe data," in Traffic Technology International, (Surrey, UK), pp. 41–44, UK and International Press, Aug/Sep 1996.

27. C. Oh, S. G. Ritchie, and S. T. Jeng, "Vehicle re-identification using heterogeneous detection systems," in Proceedings of the 83rd Annual Meeting of the Transportation Research Board, (Washington, DC), pp. 98–106, TRB, National Research Council, Jan. 11-15 2004.

28. V. Totten, Application of Vehicle Re-Identification and Evaluating Detector Installation Performance. PhD thesis, Purdue University, West Lafayette, IN, May 2008.

29. D. J. Dailey, "Travel-time estimation using cross-correlation techniques," Transportation Research Part B, Vol. 27B, No. 2, pp. 97–107, 1993.

30. B. Coifman, "Vehicle reidentification and travel time measurement in real-time on freeways using the existing loop detector infrastructure," Transportation Research Record, Vol. 1643, No. 22, pp. pp. 181–191, 1998.

31. B. Coifman and S. Krishnamurthy, "Vehicle re-identification and travel time measurement across freeway junctions using the existing detector infrastructure," Transportation Research Part C, 2007.

32. B. Coifman and E. Ergueta, "Improved vehicle reidentification and travel time measurement on congested freeways," Journal of Transportation Engineering, Vol. 129, No. 5, pp. 475–483, 2003.

33. Haoui, R. Kavaler, and P. Varaiya, "Wireless Magnetic Sensors for Traffic Surveillance," Transportation Research Part C, November 2007.

34. M. Day, H. Premachandra, T. M. Brennen Jr., J. R. Sturdevant, and D. M. Bullock, "Operational evaluation of wireless magnetometer vehicle detectors at a signalized intersection," Transportation Research Board Annual Meeting, January 2010.

35. L. H. Orcutt and M. Y. AlKadri, "Overcoming roadblocks to innovation," Transportation Research Record, vol. 2109, no. 10.3141/2109-08, pp. 65–73, 2009.

36. R. Kavaler, K. Kwong, and A. Raman, "Arterial performance measurement with wireless magnetic sensors," Transportation Research Board Annual Meeting, January 2010.

37. K. Kwong, R. Kavaler, R. Rajagopal, and P. Varaiya, "A Practical Scheme for Arterial Travel Time Estimation Based on Vehicle Re-identification Using Wireless Sensors," In Proceedings of the 88th Annual Meeting of the Transportation Research Board, Washington, DC, January 2009.

38. Ernst, J.M., J.V. Krogmeier, and D.M. Bullock, "Non-Linear Compensation of Vehicle Signatures Captured from Electromagnetic Sensors with Application to Vehicle Re-identification," Accepted to the 13th International IEEE Conference on Intelligent Transportation Systems , September 19-22,2010, Madiera Island.

39. K. Kwong, R. Kavaler, R. Rajagopal and P. Varaiya. IEEE Trans Intelligent Transportation Systems, in press, 2010.

40. Quiroga, C. and D. Bullock, "Determination of Sample Sizes for Travel Time Studies," Institute of Transportation Engineers Journal on the Web, Volume 68, Number 8, pp. 92-98, August 1998.

41. Ernst, Joseph M., Mandoye Ndoye, James V. Krogmeier, and Darcy M. Bullock, "Maximum-likelihood speed estimation using vehicle-induced magnetic signatures", International transportation systems conference, 2009.

42. J. M. Ernst, A. Ault, J. V. Krogmeier, and D. M. Bullock, "Recommended tolerances for magnetometer orientation and field calibration procedure," Transportation Research Record, vol. 2128, no. 10.3141/2128-07, pp. 66–75, 2009.

43. R. López-Valcarce, C. Mosquera, and F. Pérez-Gonzoláz, "Estimation of Road Vehicle Speed Using Two Omnidirectional Microphones: A Maximum Likelihood Approach," no. 8, pp. 1059–1077, 2004.

44. L. E. Y. Mimbela and L. A. Klein, "Summary of vehicle detection and surveillance technologies used in intelligent transportation systems," tech. rep., Federal Highway Administration's Intelligent Transportation Systems Joint Program Office, November 2000.

45. S. Y. Cheung, S. Coleri, B. Dundar, S. Ganesh, C.-W. Tan, and P. Varaiya, "Traffic measurement and vehicle classification with a single magnetic sensor," Transportation Research Record, vol. 1917, pp. 173–181, Feb 2005.

46. N. Zhao, L. Yu, H. Zhao, J. Guo, and H. Wen, "Analysis of traffic flow characteristics on ring-road expressways in Beijing using FCD and RTMS data," In Proceedings of the Annual Meeting of the Transportation Research Board, January 2009.

47. L. Klein, M. Kelley, and M. Mills, "Evaluation of overhead and in-ground vehicle detector technologies for traffic flow measurement," J. Test. Eval., Vol. 25, No. 2, pp. 215–224, 1997.

48. J. R. Smith, Modern Communication Circuits. New York: McGraw-Hill, second ed., 1998.

49. L. A. Klein, M. K. Mills, and D. R. P. Gibson, "Traffic detector handbook: Third edition," Tech. Rep. FHWA-HRT-06-108, Federal Highway Administration, October 2006.

50. Kay, S. M., "Fundamentals of Statistical Signal Processing: Detection Theory", pp. 53–55,142–143. Upper Saddle River, NJ: Prentice-Hall, Inc., 1st ed., 1993.

51. C. Sun, S. G. Ritchie, K. Tsai, and R. Jayakrishnan, "Use of vehicle signature analysis and lexicographic optimization for vehicle re-identification on freeways," Transportation Research Part C, Vol. 7, pp. 167–185, 1999.

52. Tok, Andre. "Correcting acceleration-distorted inductive signatures to improve traffic information", In Proceedings of the 88th annual meeting of the transportation research board, Washington, DC January 2010.

53. R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification. New York: John Wiley and Sons, 2000.