NTRCI

NATIONAL TRANSPORTATION RESEARCH CENTER, INCORPORATED
University Transportation Center

# U09: License Plate Recognition (Phase B)

This project was funded by the NTRCI University Transportation Center under a grant from the U.S. Department of Transportation Research and Innovative Technology Administration (#DTRT06G-0043)



Professor Lee D. Han, Ph.D.
The University of Tennessee

June 2010

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| 4. Title and Subtitle<br>**U09: License Plate Recognition (Phase B)** | | 5. Report Date<br>**June 2010** |
| | | 6. Performing Organization Code |
| 7. Author(s)<br>**Lee D. Han, Ph.D. – University of Tennessee** | | 8. Performing Organization Report No. |
| 9. Performing Organization Name and Address<br>**National Transportation Research Center, Inc.**<br>**University Transportation Center**<br>**2360 Cherahala Blvd.**<br>**Knoxville, TN 37932** | | 10. Work Unit No. (TRAIS)<br>11. Contract or Grant No.<br>**DTRT06G-0043** |
| 12. Sponsoring Agency Name and Address<br>**U.S. Department of Transportation**<br>**Research and Innovative Technology Administration**<br>**1200 New Jersey Avenue, SE**<br>**Washington, DC 20590** | | 13. Type of Report and Period Covered<br>**Final Report November 2008 – June 2010**<br>14. Sponsoring Agency Code<br>**RITA** |
| 15. Supplementary Notes | | |


16. Abstract

**License Plate Recognition (LPR) technology has been used for off-line automobile enforcement purposes. The technology has seen mixed success with correct reading rate as high as 60 to 80% depending on the specific application and environment. This limitation can be, and is often, remedied through human verification after the fact and before a citation is issued.**

**Armed with advanced text-mining algorithms, this study enables LPR technology for real-time enforcement by matching plates whether correctly or incorrectly read at various locations in a network or along a corridor and, hence, tracking the movement and speed of vehicles. The focus of the project is on heavy vehicles as they are required to enter weigh stations, where the LPR tracking information can be used, in real time, for speed enforcement and/or as a triggering factor for other inspection activities.**

**The initial objective of the study was to devise an inexpensive and effective means for helping improve air quality in non-attainment metropolitan areas through speed enforcement. The successful deployment of such a measure can also potentially lead to improved highway safety, fuel efficiency, and national security. It is desirable to deploy this technology in a larger scale to realize the benefits and potential.**


| 17. Key Word | 18. Distribution Statement |
|---|---|
| **License Plate Recognition (LPR), speed enforcement, heavy vehicles, weigh stations, air quality, emissions, text-mining algorithms, national security, safety inspection, weight compliance, vehicle profiling** | **No restrictions** |

| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of Pages | 22. Price |
|---|---|---|---|
| **Unclassified** | **Unclassified** | **60** | |

# Table of Contents

# List of Figures

# List of Tables

# List of Equations

# Executive Summary

License Plate Recognition (LPR) technology has been used for off-line automobile enforcement purposes. The technology has seen mixed success with correct reading rates as high as 60 to 80% depending on the specific application and environment. This limitation can be, and is often, remedied through human verification after the fact and before a citation is issued.

Armed with advanced text-mining algorithms, this study enables LPR technology for real-time enforcement by matching plates whether correctly or incorrectly read at various locations in a network or along a corridor and, hence, tracking the movement and speed of vehicles. The focus of the project is on heavy vehicles as they are required to enter weigh stations, where the LPR tracking information can be used, in real time, for speed enforcement and/or as a triggering factor for other inspection activities.

The first phase of the project, which was reported previously, saw the development of the enabling text-mining algorithm and the demonstration of plate matching rate improvement from less than 60% to better than 90%. However, the false-matching rate is still relatively high. The second phase, which is reported herein, deployed the LPR technology and matching algorithm in a continuous fashion capturing plates throughout the day. The improved algorithms developed in this phase of the study improves the positive matching rate to over 97% while keeping the false-matching rate less than 1%. The deployed LPR system is now capturing plates 24/7 on I-640 and I-40 in Knoxville, TN.

The initial objective of the study was to devise an inexpensive and effective means for helping improve air quality in non-attainment metropolitan areas through speed enforcement. The successful deployment of such a measure can also potentially lead to improved highway safety, fuel efficiency, and national security. It is desirable to deploy this technology in a larger scale to realize the benefits and potential.

# Chapter 1 – General Overview

## *Background*

The purposes of this study are to field-deploy LPR technology and develop necessary technologies for tracking large trucks, via license plate recognition, in real time. During Phase A of this project, LPR technology was tested in a "mobile" setup (tripod-mounting) with limited deployment (one hour or so of data collection for each data collection period). The main goal of Phase B, detailed herein, is to install a pair of LPR units "permanently" on Interstate 40 to capture truck license plates continuously. This 24/7 operation would require much more stringent configuration in a real-world operational environment. The data have to flow via a wireless network in real-time. Better algorithm development is a key outcome of this phase.

The main application of LPR technologies, from the perspective of local government, is speed monitoring and enforcement. Yet other aspects such as homeland security, safety inspection, weight compliance, and vehicle profiling are also enabled because of the maturing technologies, including license plate recognition and automated plate matching.

A United States Department of Transportation (US DOT) study in 2003 [Tang et al] found that reducing large truck speed by 10 mph could reduce NOx emission by 18% per large truck. As a result, a number of metropolitan areas reduced the large truck speed limit on their urban Interstate highways. Knoxville, TN is one of them. In April 2006, Knoxville Regional Transportation Planning Organization lowered the speed limit for all large trucks with gross weight over 10,000 pounds on Interstate 40 (I-40) from 65 to 55 mph as a countermeasure to combat harmful emissions and improve air quality. Before effectiveness of this action is assessed, an important question to ask is whether the lowered speed limit was enforced.

According to Federal Highway Administration (FHWA) statistics, there is an estimated total of 12 million large trucks passing through the Knoxville section of I-40 annually. Among these, based on a previous study conducted by Han et al, at least 50% of all trucks were speeding. To enforce the newly enacted speed limit on I-40, a major increase in the number of highway patrol officers, patrol vehicles, citations, and overall resource commitment is essential. Otherwise, a lowered speed limit that does not yield an actual 10 mph speed reduction will have little benefit on air quality. In light of limited state resources and the heightened desire to reduce large truck operational speed, the National Transportation Research Center, Inc. (NTRCI) funded this study to assess the feasibility of deploying automated License Plate Recognition (LPR) technology for tracking large trucks, for the purpose of speed measurement and enforcement. Since large trucks are expected to pass through weigh stations for weighing and inspection purposes, it makes sense to track these trucks along the Interstates and, later, as they pull into the weigh station where their average speed can be used to initiate inspection and/or enforcement actions.

## *Project Team*

Phase B of this project was carried out by the University of Tennessee. The principal investigator (PI) of the project was Dr. Lee D. Han, who was also the PI of Phase A of the project. Francisco Moraes Oliveira-Neto, a Ph.D. candidate, assisted Dr. Han with algorithm development tasks while Stephanie Hargrove, also a Ph.D. student, assisted Dr. Han with fieldworks. Some of the students involved in the fieldwork or data analysis aspects include Elliott Moore, Scott DeNeale, Sam Moss, Jonathan Liu, and Steven Han.

A team of Tennessee Department of Transportation (TDOT) staff and PIPS engineers also assisted with the field deployment tasks.

## *Project Description*

Phase A of this study found the current LPR technology to be less than perfect, largely due to the multitude of design differences of license plates from different states. These include colors of the plates and the characters, fonts used, syntax, reflectivity, etc. As a result, plates from some states are more challenging to recognize correctly than those from others. LPR technology providers do offer ways to calibrate the internal LPR algorithms or the sensitivity of the imaging mechanism to favor certain colors (e.g. wave lengths), reflectivity, and syntaxes, but the improvement of readability of certain plates are typically at the cost of others. Figure 1 shows actual results from Phase A of this study.

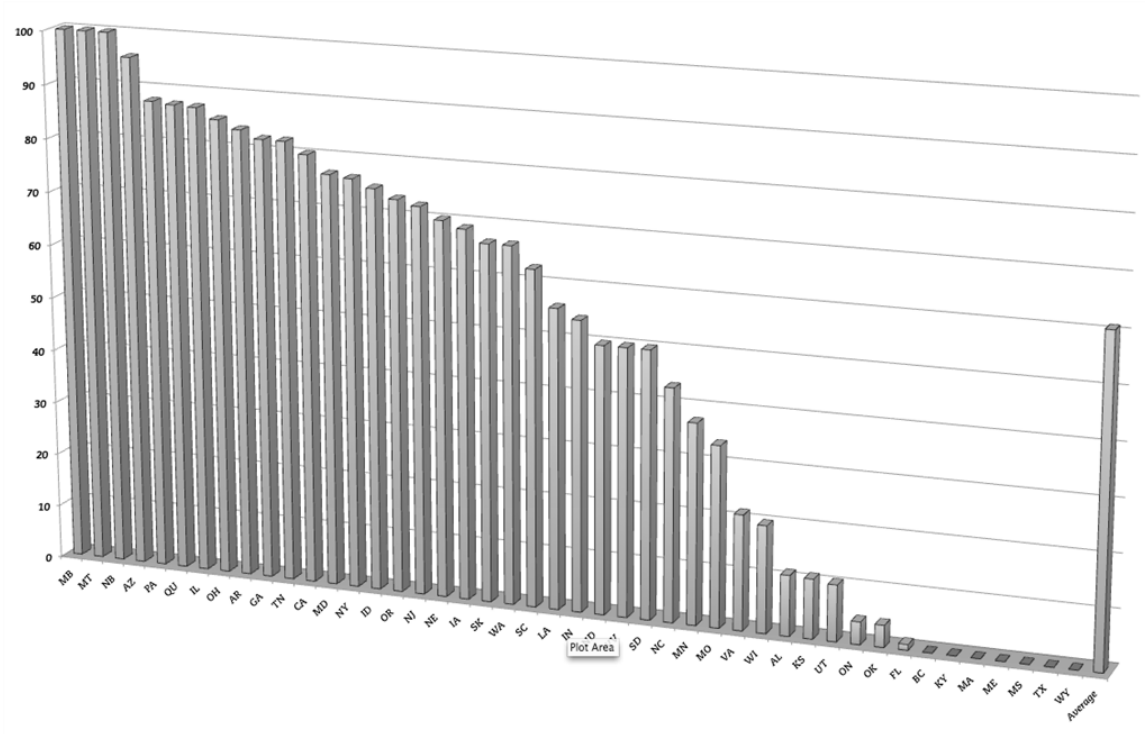


**Figure 1. Chart. Plate Recognition Rates for Different US States and Canadian Provinces.**

For the plates with high recognition rates, matching the same plate at multiple locations in the highway network is relatively easy. However, this is not the case for plates with low LPR recognition rates. It is not unusual to have very low to near-zero correct matching rates for plates from some states because of the dismal LPR recognition rates. By employing advanced text-mining algorithms, license plates can be improved and, thus, enable the deployment of large truck tracking and monitoring in real-time. During Phase A of this study, the matching algorithm relied heavily on the development of "truth matrices," which are essentially look-up tables of the likelihood of one character being read as something else. Truth matrices are difficult to establish, as they are different from one station to the next, and may change over time. A significant amount of effort in Phase B went into addressing the issues of improving the correct license plate matching rate and reducing the false matching rate. Chapter 2 of this study details a text-mining algorithm called Edit Distance that is the foundation of the plate matching algorithms.

The single most challenging aspect of this research, albeit non-technical, was related to the field deployment of the LPR hardware. To this end, Chapter 3 presents some of the activities of this task, the related challenges, and efforts to overcome these challenges that transpired during this study.

Chapter 4 details scholarship research involved in improving the basic matching algorithm presented in Chapter 2. The new and improved algorithms were subsequently implemented on the data collected after the deployment mentioned in Chapter 3. The results are presented in Chapter 5 of this report.

When a wide area deployment of the technology is realized, the objectives of air quality, safety, and fuel efficiency improvement would follow. Moreover, the functionality of tracking large trucks on the nation's highway network can also have functional implications on national security.

# Chapter 2 – Plate Matching with Edit Distance

The process of matching two strings involves a sequence of comparisons of individual characters to determine the degree of similarity between the two. Consider, for example, a license plate with the string "4455HZ," which is read by two LPR machines at two different locations. Suppose that at the first location, the plate was read as "4455IIZ" and at the second, "4455HZ." Neither LPR unit "knows" whether it has read the plate correctly. By looking at the two reports, one can either declare no match, or perhaps speculate a potential match since the two strings differ by only two pairs of characters: "I"-"H" and "I"-"" (where "" represents a null or empty character). If there were another plate that was read as "445OHZ" earlier at the first location, one may speculate that it is less likely that the "O"-"5" pair is a match. The task here is to "teach" the computer to make such speculations.

Techniques for measuring the similarity or dissimilarity between two strings have been developed in the past and have found application in areas such as handwritten character recognition and computation biology [Mei]. The pioneer in this field is Vladimir Levenshtein, who developed Edit Distance (ED), also known as Levenshtein distance, which is a metric that computes the distance between two strings as measured by the minimum-cost sequence of edit operations [Levenshtein]. Given two strings ***x*** and ***y***, their Edit Distance describes how many fundamental operations are required to transform ***x*** into ***y***. These fundamental operations are termed as follows:

- ***Substitutions:*** A character in ***x*** is replaced by the corresponding character in ***y*.**
- ***Insertions:*** A character in ***y*** is inserted into ***x***, thereby increasing the length of ***x*** by one character.
- ***Deletions:*** A character in ***x*** is deleted, thereby decreasing the length of ***x*** by one character.

To relate the definition of Edit Distance to the problem at hand, we return to the example of the plate "4455HZ" being captured by two LPR stations. Let ***x*** = "4455IIZ" and ***y*** = "4455HZ"; the task is to compute the number of fundamental operations to transform ***x*** into ***y***. (Note that ***x*** and ***y*** could have been assigned in reverse order since the "true" plate string is unknown.) In this case, it can be established that the minimum number of operations is 2, which corresponds to the substitution of the first "I" in ***x*** by "H" and the deletion of the second "I" in ***x***. Therefore, the Edit Distance ***d*(*x*,*y*)** between ***x*** and ***y*** is 2.

To understand why 2 is the minimum number of operations to transform ***x*** into ***y*** in our example, imagine the two strings disposed in a two-dimensional grid, as shown in Figure 2. The points on the axes represent the corresponding sequence of characters, with the sequence ***x*** on the *j* axis and the ***y*** sequence on the *i* axis. Let a move on this grid be represented by a link that ends on a point associated with the two characters $(x_{i_k}, y_{j_k})$. A diagonal downward move is defined as a *substitution*; a horizontal move to the right represents a *deletion*; and a vertical downward move

represents an *insertion*. Each node of the grid is associated with a function $\gamma(i_k, j_k)$, which measures the cost of each move along the grid. For the original construct of ED, this cost is set to 1 for insertions and deletions; in the case of substitutions, $\gamma(i_k, j_k)$ is 0 if the corresponding characters are the same, i.e., $x_{i_k} = y_{j_k}$, or 1 if they are not the same. If we "walk" from the origin point (0,0) to the end point ($i_m$,$j_m$) on the grid, each potential path is associated with an overall cost, *d*, defined as:

$$d(i_m, j_m) = \sum_{k=0}^{n} \gamma(i_k, j_k)$$

**Equation 1. Edit Distance from the Origin to an End Point.**

where,

*n* is the number of nodes of a path between ($i_0$, $j_0$) = (*0,0*) and ($i_m$, $j_m$) = (|*x*|,|*y*|)*;* and |*x*| and |*y*| are the lengths (number of characters) of ***x*** and ***y***.

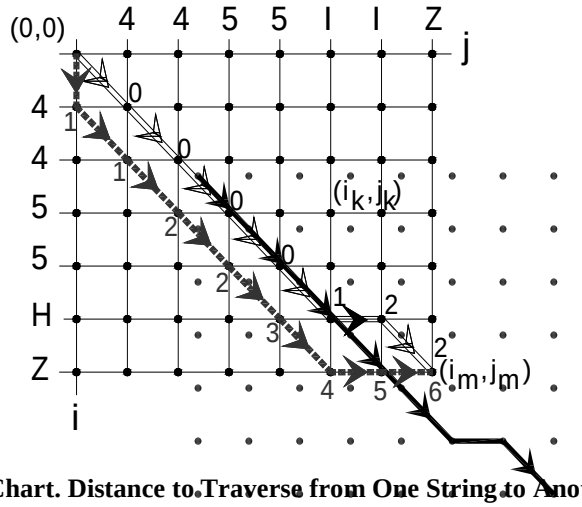


**Figure 2. Chart. Distance to Traverse from One String to Another.**

As an example, consider two paths (drawn by the solid and dashed lines) reaching the point ($i_m$, $j_m$) as shown in Figure 2. Computing the number of editing operations performed by these two paths will result in $d_{solid}(i_m, j_m) = 2$ and $d_{dashed}(i_m, j_m) = 6$.

To obtain the shortest path, one could exhaust all possible combinations of paths. Fortunately, there is a less computationally expensive procedure called *dynamic programming*, proposed by [Wagner and Fisher]. A detailed description of this procedure can be found in the book *Pattern Classification* [Duda et al]. As a result of applying dynamic programming to the Edit Distance problem, ***d(x,y)*** is determined to represent the minimum cost to reach the point ($i_m$, $j_m$), or $d(x,y) = \min\{d(i_m, j_m)\}$.

In many other applications, string ***y*** is provided by a list of words that has the maximum likelihood of containing the "true" value of the given string, ***x***. This pre-specified list of words is called a lexicon or reference for matching. Using this list of words, it is possible to detect errors, generate candidate corrections, and rank these candidates. However, the plate-matching problem at hand presents a significantly tougher challenge as neither ***x*** nor ***y*** is necessarily a true value from a limited pool of lexicon.

## *Matching Methodology*

Since this study considered plates read at two different locations, in the remaining text they will be referred to as *LPR Station 1* and *LPR Station 2* where LPR Station 2 is downstream of LPR Station 1. As such, for a given plate read at LPR Station 2, there are a number of candidate plates already read, correctly or not, at LPR Station 1 for matching purposes. The number of candidate plates can be constrained by an imposed threshold value, $\tau$, for the edit distance. There can also be a time window constraining the allowable travel time between the two stations. Therefore, the number of candidate plates read at LPR Station 1 for each plate at LPR Station 2 is defined by the following constraints. Figure 3 illustrates this.

$$ED \leq \tau$$

$$tt_i \in [tt_{lower}, tt_{upper}]$$

**Equation 2. Eligible Plates for Matching within a Time Window.**

where,

$tt_i$ = travel time observed for a potential match *i*;
$[tt_{lower}, tt_{upper}]$ = lower and upper limits of the time window.

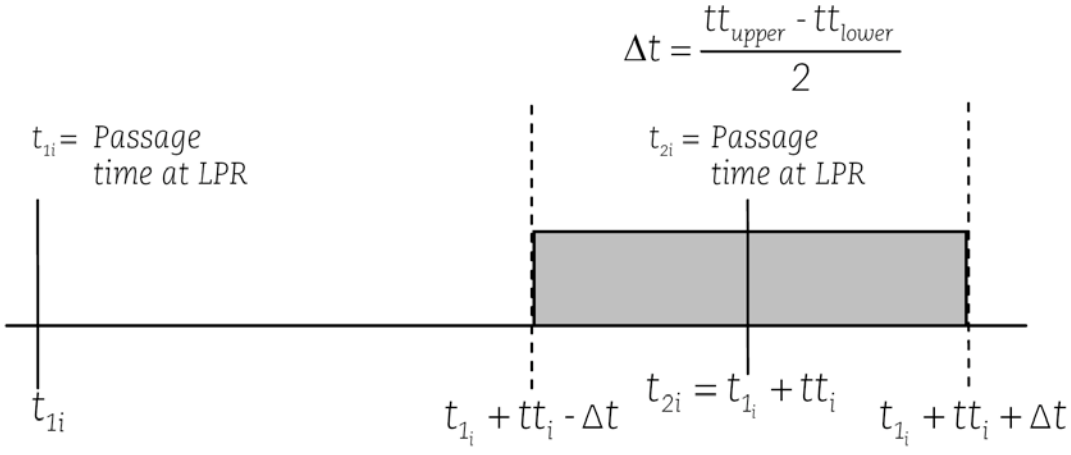


**Figure 3. Chart. Time Window of Matching Eligibility**

Using these constraints, all string candidates read from LPR Station 1 are ranked, and the candidate with the least ED not exceeding $\tau$ is selected. In case there are multiple candidates tied with the least ED value, the first appearing in chronological order is selected.

The time window initially includes the entire period of data collection. In the end, two time windows were used to reduce comparison operations with limits defined as follows:

$$tt_{uper} = tt\_mean + z \times tt\_std$$

$$tt_{lower} = tt\_mean - z \times tt\_std$$

**Equation 3. Upper and Lower Limits Defining a Time Window.**

where,

*tt_mean* = sample mean travel time;
*tt_std* = sample standard deviation of travel time;
*z* = the number of standard deviations above or below the mean.

## *Case Study and Results*

The LPR equipment used in Phase A of this study was manufactured by PIPS Technology. Two older versions of the equipment were used to capture license plates of westbound trucks on I-40; one at Campbell Station Road (LPR Station 1) and the other downstream at the weigh station (LPR Station 2). Both units used internal detection (plate-finder) software to trigger the camera and an infra-red-based illuminator, which was activated when a vehicle was within the camera's field of view. The two cameras were set up to capture plates in the rightmost lane of the road. Data were collected on weekdays, between 1:00 PM and 4:00 PM, excluding days of abnormal traffic patterns. The distance between the two stations was about 1.4 miles. During five days of data collection, 2,671 plates were captured at the first station and 1,530 were captured at the second station. Among these, a total of 787 were manually verified as identical. In addition to reading plates, the equipment also "stamped" each plate image with time information, which was useful for later comparisons.

### *LPR Performance*

The raw images stored in the LPR system database were viewed manually to compare with the detection reports. The results show an average accuracy of 61% for Station 1 and 63% for Station 2. Since the cameras were not permanently mounted (they were mounted on heavy tripods), the accuracies could potentially be higher.

In spite of the moderate accuracy, the equipment was able to read most characters of the license plates. Figure 4 shows the failure rate distribution, a chart of relative frequency of plates versus the number of characters misread per plate, for each LPR station.

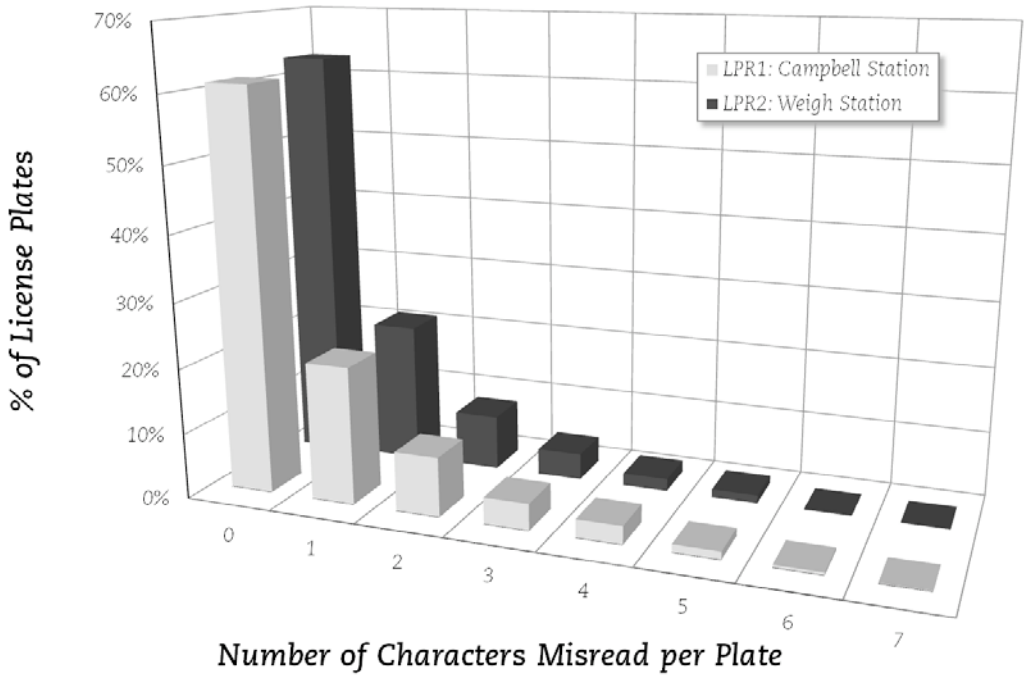


**Figure 4. Chart. Percent of Plates with Number of Misread Characters.**

*Truck Speed*

The histograms of the sample speeds of the 787 trucks captured at both stations are shown in Figure 5. Figure 6 shows truck-speed comparisons before (measured by tube) and after (using radar gun and LPR) the speed limit changed. As observed in Figure 5, after the new speed limit went into effect, truck speed ranged from 40 mph to 75 mph, and most of the speed values (the 20th percentile was approximately 55 mph) were higher than the actual speed limit of 55 mph. In Figure 6, comparing the before- and after-speed distributions, a shift of only about 8 mph in the average speed was observed, with no change in the variance.

*ED Performance Results*

To assess the performance of the matching methods implemented, modules in the MATLAB programming language were written to perform the calculations automatically. The number and percentage of positive matches, the number of false-positive matches, and the average number of candidates per plate were used as performance measures.

Four different threshold values, 0, 1, 2, and 3, were used to constrain edit distance. The results are shown in Figure 7. As the threshold value decreases, the number of candidates per plate gets smaller. In fact, as the threshold gets smaller it is unlikely to have more than one candidate from LPR Station 1 match a given plate at LPR Station 2.

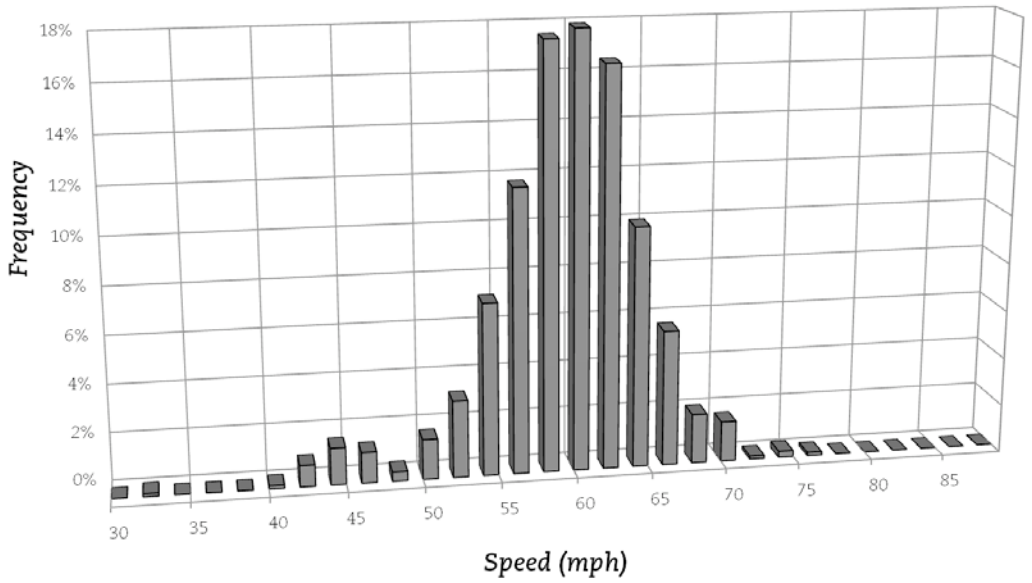


**Figure 5. Chart. Frequency of Sample Truck Speeds.**

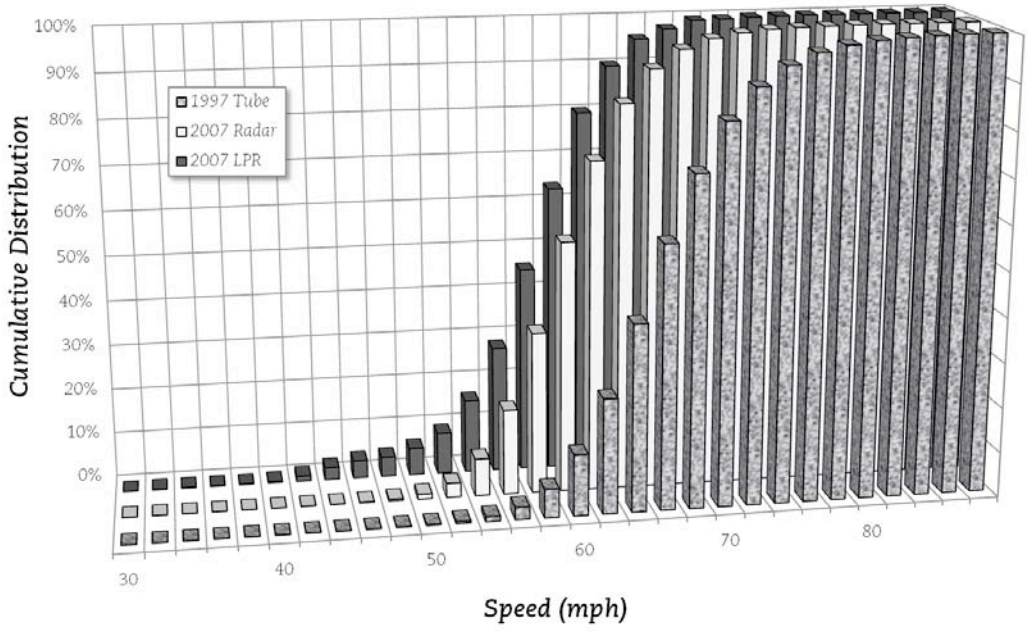


**Figure 6. Chart. Cumulative Distribution of Sample Truck Speeds.**

Table 1 shows the results obtained when both the top-rank and the first chronological candidates are selected. As can be seen, although smaller threshold values result in fewer false-positive matches, they also result in fewer positive matches. Moreover, without considering the travel time information, false-positive matches are very likely to occur for threshold values of 2 and 3.

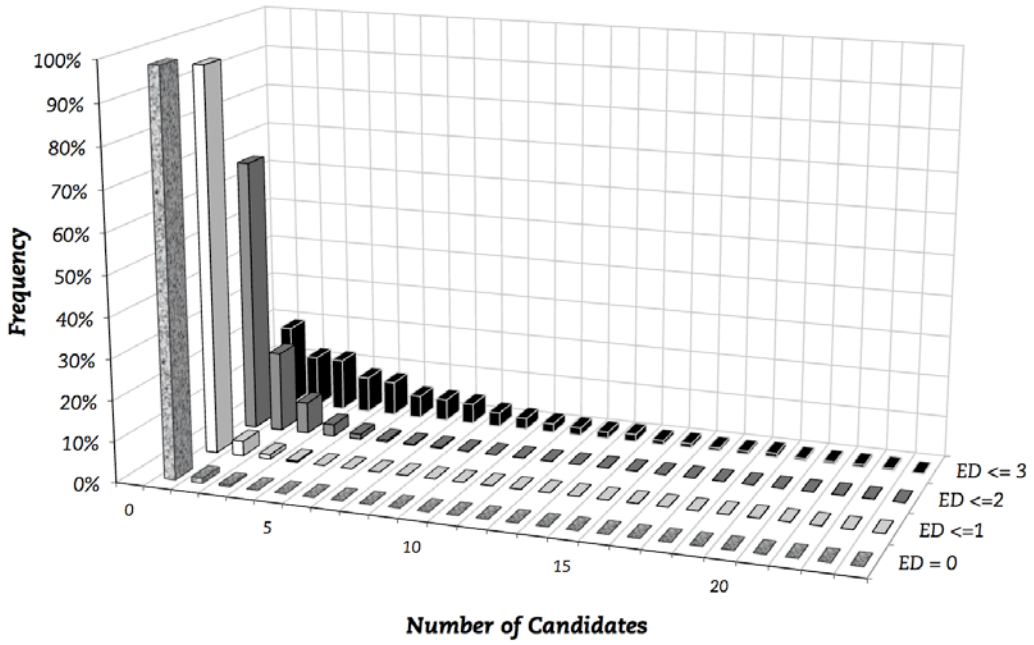


**Figure 7. Chart. Number of Plate Candidates without Time Window.**

**Table 1. Performance of ED without Travel Time Constraints.**

| SIMILARITY MEASURE USED | THRESHOLD | # OF MATCHES | # OF POSITIVE MATCHES | # FALSE POSITIVE MATCHES | AVERAGE NUMBER OF CANDIDATES | PERCENTAGE DETECTED |
|---|---|---|---|---|---|---|
| ED | 0 | 497 | 497 | 0 | 1.02 | 60% |
| ED | 1 | 692 | 667 | 25 | 1.08 | 81% |
| ED | 2 | 921 | 737 | 184 | 1.59 | 89% |
| ED | 3 | 1309 | 754 | 555 | 5.74 | 91% |

Constraining the number of candidates using only deterministic travel time intervals on each day, with $z = 3$ and 4, the distributions of candidates were computed as shown in Figure 8. Table 2 presents the results of combining the two constraints, Edit Distance and Travel Time, for each day of data collection. Note that the time constraint (time window) was different for each day, as it depends on the mean and standard deviation observed on each day sampled.

In Table 2, for all threshold values tested it seems very unlikely (with the highest average number of candidates being 1.03) to have more than one candidate per plate. In fact, most candidates turned out to be the true values (with only about 2% of mismatches in the worst case when a 97% plate-matching rate is also achieved). It is worth noting that all cases of false matches in Table 2 came from pairs of plates where either one or both plates were misread by the LPR machines, or were not read at all at one of the two LPR stations. Such could happen, for example, when a truck changes lane between the two LPR stations, either entering or exiting the monitored lane. This means that any plate captured at both stations, correctly or not, has a high chance of being indentified and leading to a positive match; whereas a plate not read at one

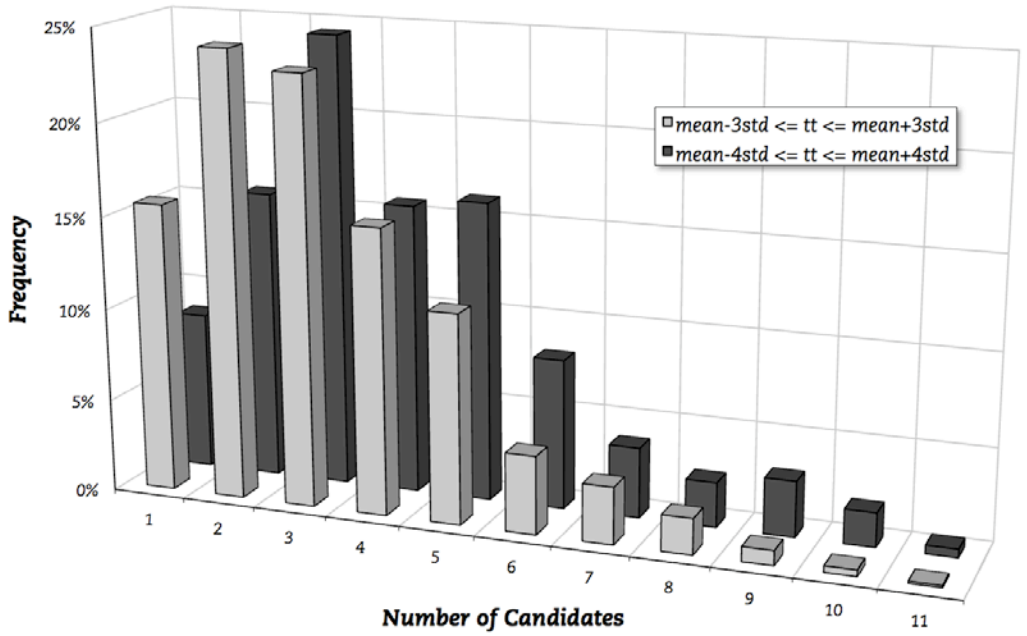


**Figure 8. Chart. Matching Candidate with 3 and 4 Standard Deviations of Travel Time.**

**Table 2. Performance of ED with Time Window Constraints.**

| SIMILARITY MEASURE USED | THRESHOLD | # OF MATCHES | # OF POSITIVE MATCHES | # FALSE POSITIVE MATCHES | AVERAGE NUMBER OF CANDIDATES | PERCENTAGE MATCHED |
|---|---|---|---|---|---|---|
| $ED + tt_1$ | 0 | 471 | 471 | 0 | 1.00 | 61% |
| $ED + tt_1$ | 1 | 636 | 636 | 0 | 1.00 | 82% |
| $ED + tt_1$ | 2 | 719 | 716 | 3 | 1.01 | 93% |
| $ED + tt_1$ | 3 | 762 | 746 | 16 | 1.02 | 97% |
| $ED + tt_2$ | 0 | 477 | 477 | 0 | 1.00 | 61% |
| $ED + tt_2$ | 1 | 645 | 645 | 0 | 1.00 | 82% |
| $ED + tt_2$ | 2 | 731 | 726 | 5 | 1.01 | 93% |
| $ED + tt_2$ | 3 | 779 | 756 | 23 | 1.03 | 97% |

of the stations has an elevated probability of resulting in a false match. Therefore, if the LPR machines were configured to aim at multiple lanes and, hence, capture more plates, the false matches may be further reduced. To illustrate this, Table 3 presents the five false-positive matches for the case of ED ≤ 2 and the second time window constraint, z = 4.

**Table 3. False Matches for ED ≤ 2 + $tt_2$.**

| LPR Station 2 | | | LPR Station 1 | | | Hypothetical Speed (mph) |
|---|---|---|---|---|---|---|
| Time | Recognized Characters | Plate Number | Time | Recognized Characters | Plate Number | |
| 13:30:26 | "1561" | "1561" | 13:28:46 | "15S7" | "15157" | 111.28 |
| 14:46:37 | "1297D" | "12990" | 14:43:37 | "12905" | "12905" | 40.62 |
| 15:19:36 | "1234" | "1234" | 15:16:57 | "9214" | "9214" | 48.74 |
| 15:14:12 | "2JZ294" | "2JZ294" | 15:13:21 | "2JF204" | "2JF204" | 70.74 |
| 14:14:39 | "9713" | "9713" | 14:13:13 | "9214" | "9214" | 59.18 |

## *Discussions*

The “first generation” matching algorithm presented herein is not expected to achieve perfection with perfect plate matching rate and zero false matches. Nevertheless, improvement can still be accomplished through further research on better plate similarity measures, dynamic travel time constraints, and improved configuration of LPR hardware.

Concerning the similarity measure employed in this study, the main drawback of using the formulation of edit distance with unitary cost functions, in the case of comparing distinct characters, is that it does not account for the probabilities of LPR machine misreading certain characters. For example, there is a relatively high chance of the characters "1," "0," and "B" being misread as "I," "O," and "8," respectively. The probabilities of such incidences were not considered in Phase A of the project. However, this information can be obtained by constructing a matrix of error probabilities taken by each LPR unit used. Once the matrix is constructed, the challenge becomes how to design the weights (or the cost function) to be used in the edit distance calculation. For example, what is the cost for transforming "0" into "O" given that the odds that "O" is misread as "0" are, for example, 50%. Some initial work by the author suggests that using a cost function would increase the number of positive matches and reduce false-positive matches. For example, in the first row of Table 3, "1561" and "15S7" would not have been falsely matched if it were known that the character "6" is very unlikely to be recognized as "S," or vice versa.

The travel time constraint used in this study is a simple and deterministic one. A more suitable way of using travel-time information would take advantage of travel time distributions, which are believed to be a Gaussian function. This way, the deterministic and chronological method used herein for selecting candidate plates would be replaced by a probabilistic method, where a candidate would be selected if it had both the lowest ED value and the highest travel time probability.

As for the equipment setup on the roadside, past deployment experience indicates that a permanent rather than a mobile setup would lead to improved accuracy in plate reading in the first place. To this end the newest version of LPR machines were mounted permanently on dynamic message signs (DMS) on Interstate 40.

Another issue of interest is the effect of the distance between LPR units. The closer two LPR units are located, the smaller in general the time window, and the less the chance of the presence of trucks with similar license plate numbers during the same time frame. On the other hand, very closely located LPR units tend to capture more of the instantaneous speeds and not the average trip speed. Trucks can simply slow down for the “speed trap” and then get back to the preferred cruising speed. In this regard, LPR units spaced farther apart could be more effective as a speed deterrent. Therefore, this distance should be one that neither compromises the desirable reliability of the algorithm nor affects traffic behavior.

The LPR pairs can be located anywhere along the main section of a roadway. In fact, a design with LPR units every couple of miles along the Interstate has been studied recently. One consideration though is the LPR pairs should not be too close to the exit ramp for the weigh station, where trucks may have already begun to slow down.

# Chapter 3 – Field Deployment of LPR Technology

During Phase A of the study, two relatively dated and different LPR units were mounted on tripods and situated on the side of I-40 to collect data for a total of five hours during daylight. The data were saved on a computer tethered to each LPR unit in the field and analyzed later on. A crucial task of Phase B of the study is to install state-of-the-art LPR units "permanently" over I-40 to continuously collect data for months under all lighting and weather conditions. The data are to be relayed over a 3G cellular network for real-time filtering and analysis. This is to prepare for the real-time vehicle tracking and enforcement applications in Phase C of the study.

Due to the nature of the endeavor, a prohibitive amount of effort of Phase B was devoted to the actual field deployment of the LPR technology, even though equally important to the success of the project is the development of a plate-matching algorithm of high accuracy and low false-matching rates. The field deployment effort turned out to be a drawn-out and painstaking process with many delays and the eventual no-cost extension of the project. We will not dwell on these challenges for the benefit of the reader. Instead, the following section "fast-forwards" through the process just to give the reader a glimpse of the scale of the task.

## *Key Project Partners*

The field deployment effort of this phase depended on the generous support of Tennessee Department of Transportation (TDOT), which granted us permission to mount LPR hardware on their existing hardware structures and the convenience of using their uninterrupted electricity, which is essential to the continuous deployment of LPR. In comparison, gasoline powered generators were used for short-term data collections in Phase A. TDOT also paid for and provided assistance of Tennessee Highway Patrol personnel for traffic control, a road crew for lane closure and reopening operations, and, most importantly, a trained electrician certified and experienced to work on the catwalk over the Interstate structures where the LPR cameras are mounted.

Another invaluable partner of the field deployment effort is PIPS Technology, a Federal Signal company. They provided two brand new cutting-edge P382 Spike-HD ALPR cameras for the use of this effort for free. They also provided technical personnel in coordination with TDOT engineers for the installation, maintenance, and data networking tasks.

## *Pre-Deployment Field Test*

A one-day equipment test of the PIPS P382 Spike-HD camera, which is to be used for field deployment later, was held in 2009. The test site was on I-40 between Eblen Cave Road and Old Poplar Springs Rd (see Figure 9). A mobile configuration, which involves essentially tripods and power generators, was used similar to that in Phase A of this study (see Figure 10). The LPR units were collecting data within minutes after the completion of the setup, which was far

superior to the LPR units used previously, where it would take an hour or more to configure the LPR system.

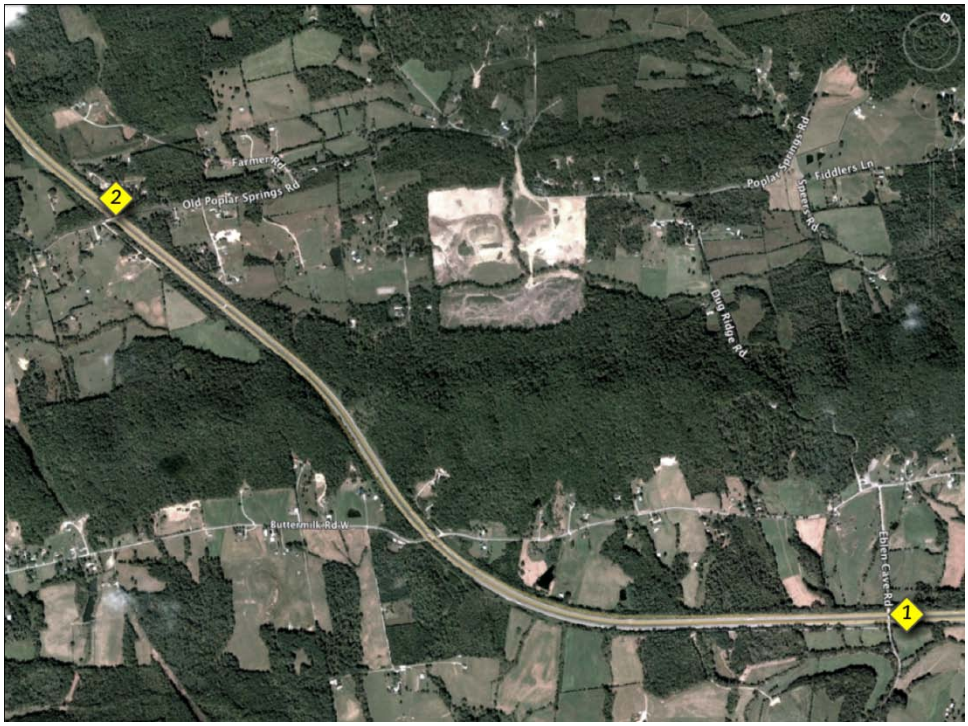


**Figure 9. Map. Pre-deployment Field Test on I-40.**

**Figure 10. Photographs. Actual Set-up at Pre-Deployment Test Site 1.**

## *Assessment of Deployment Sites*

Before the actual deployment of the LPR units, proper site or sites need to be selected. Some of the aspects to consider include:

- Travel Direction – This study needs capture trucks and then recapture the same trucks again at the weigh station on the west end of Knoxville, TN. As such, westbound direction is favored.
- Truck Traffic – It is important that ample truck traffic is present in the travel lanes where the LPR units are to be installed.
- Mounting Considerations – The LPR units should be mounted securely in an elevated angle with minimal likelihood of occlusion. The location should be very close to the moving traffic. To this end, roadside poles and overhead sign locations are most desirable.
- Availability of Power – The LPR units need reliable and continual power supply. The light poles along I-40 are many, but they are switched on and off by the utility company depending on time of the day. When the lights are off, no power is supplied to the pole. Existing Remote Traffic Microwave Sensor (RTMS) devices along I-40 rely primarily on solar energy, which is not desirable for guaranteed 24/7 operations. TDOT's traffic cameras are mounted on very high poles typically over 60 feet off the side of the travel lanes. Although these poles have continuous power supply, the poles are too thick in cross section for mounting and their locations too far from the traffic. The Dynamic Message Signs (DMS) are right on top of all lanes of the moving traffic and provide one of the few options with continuous and reliable power supply.
- Coupling of Locations – Multiple (at least two) sites are needed for LPR plate capturing and matching purpose. A reasonable distance between each pair of sites is a consideration. If the distance is very short, vehicle travel time may fluctuate significantly. If the distance is too great, many vehicles may enter and exit the stretch of the roadway and only a low percentage of vehicles would be traversing the entirety of the study site.

After pouring over the map of the TDOT ITS infrastructural locations (see Figure 11), consulting with TDOT and PIPS Technology engineers, and many site visits and meetings, Dynamic Message Signs #3 (see Figure 12), and #7(see Figure 13), were selected for permanent installation. The use of DMS sign structures, instead of the originally proposed ITS camera locations, was determined with the input from TDOT and PIPS Technology. DMS sign structures arch over the travel lanes and provide good clearance and appropriate angle for LPR applications. In addition, a continuous and reliable electrical power supply is available.

The next issue concerning these sites was selecting the exact lane at each site where the LPR cameras should be installed to maximize the number of trucks captured at both sites under the constraint of using only one camera per site. For the I-640W location( see Figure 13) it is clear after some field observations that the second (middle) lane is the most suitable for LPR installation as this lane sees the greater number of trucks than the other lanes.

The decision was more challenging for the I-40W DMS site (see Figure 12) where there are five travel lanes with the right two lanes merging from I-640 and continuing toward the Papermill

Exit. After studying and extracting truck volume information from the video of traffic traversing this section of I-40, we found that lane 3 typically has more trucks than lane 2 (from left) (see Figure 14). Therefore, the LPR camera was installed over lane 3.

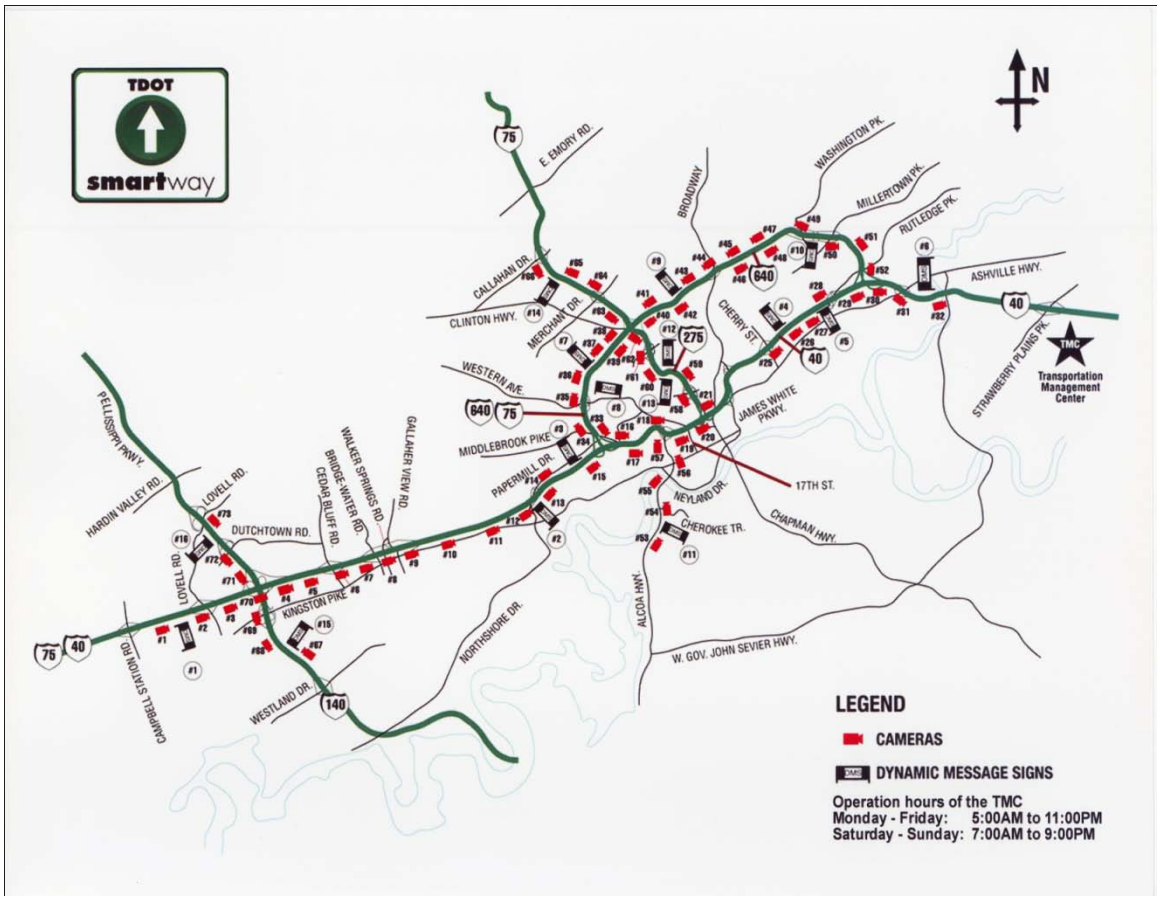


**Figure 11. Map. TDOT ITS Infrastructure Considered for LPR Study.**


**Figure 12. Photograph. TDOT DMS #3 on I-40 Near Papermill Road.**


**Figure 13. Photograph. TDOT DMS #7 on I-640W Near Pleasant Ridge Road.**

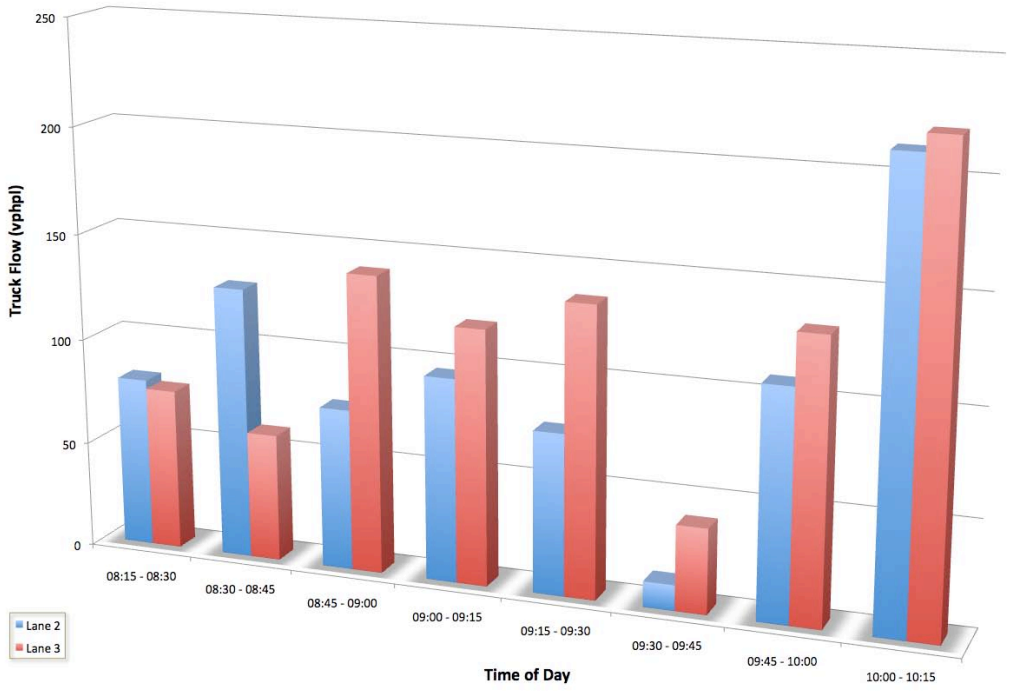


**Figure 14. Chart. Lane Distribution of Truck Traffic on I-40 Near DMS #3.**

## *Deployment Approval and Preparations*

Multiple meetings were held to determine the study sites, the LPR camera to be used, equipment mounting options, electricity and access concerns, etc. After these meetings, TDOT authorized the work to move forward with installation of two sets of LPR cameras at Station 1 (DMS #7) and Station 2 (DMS #3)(see Figure 15). With aerial photos, we measured the travel distance between the two stations to be 3.00 miles. This was also verified with field measures. With a posted speed limit of 55 mph throughout the study area, the expected legal maximum travel time for vehicles to traverse the study section can be calculated to be about 3.27 minutes or 196 seconds.

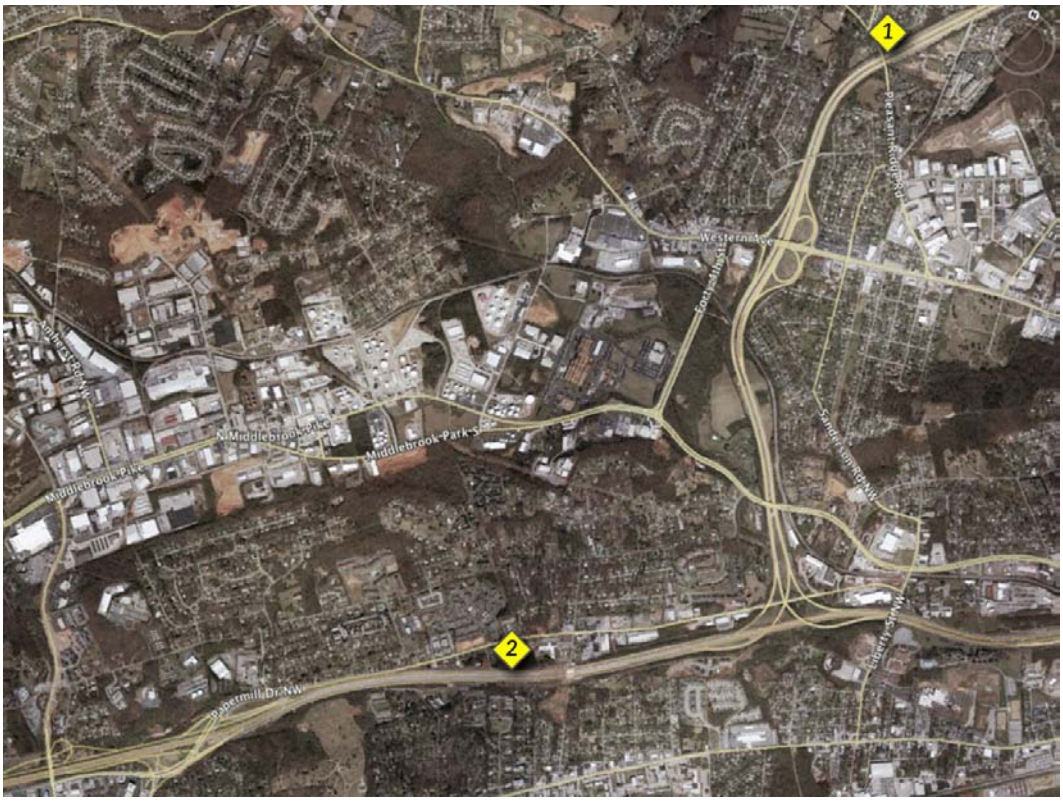


**Figure 15. Map. Deployment Site between Stations 1 and 2.**

A significant amount of effort subsequently went into coordinating various partners and their help to have all the hardware manufactured, the mounting mechanism designed and approved, the necessary wireless communication means procured, the data network configured, etc. Ultimately, all logistics had to be planned out so that the installation effort would cause the least amount of disruption to the traffic and ensure maximum safety to field workers as well as the motoring public.

## *Field Installation of LPR Hardware*

After a very long delay due to TDOT maintenance contract issues and LPR hardware availability, field installation was finally underway. A total of about 6 hours at each site was required for traffic control, utility installation, site preparation, camera mounting, real-time calibration, etc. The cameras were finally installed and are broadcasting data 24/7 via 3G cellular networks. Figure 16 showcases the field installation effort.


**Figure 16. Photograph. Highlights of Field Installation Activities.**

## *Post-installation Data Verification*

Multiple tests were conducted on the LPR data from the two stations. Known license plates were used on probe vehicles (see Figure 17) to verify the recognition accuracy at the two camera locations for two weeks after the installation and before the full-scale data analysis tasks.

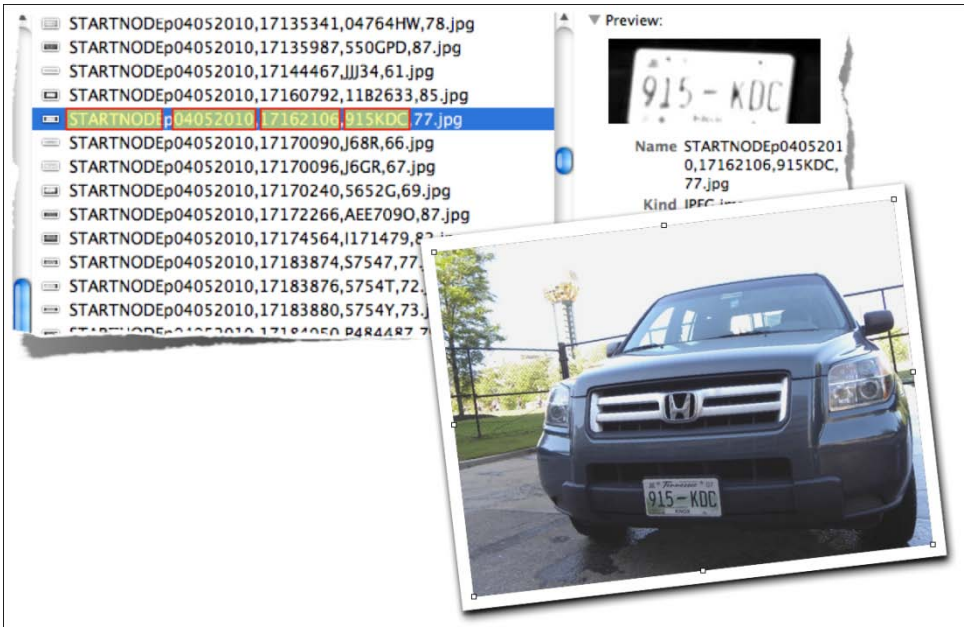


**Figure 17. Photograph. Probe Vehicle and Corresponding LPR Results.**

# Chapter 4 – Plate Matching with Weighed Edit Distance

An efficient plate-matching algorithm based on Edit Distance was presented in Chapter 2 of this report. Before One potential improvement on the aforementioned algorithm is the recognition of the "distance" between two strings should be treated differently in different cases. That is, the Edit Distance should be weighed in the algorithm to improve matching rate, and more importantly, reduce false-matching cases. To this end, the algorithm was revised and improved as presented herein.

Most of the work presented here was performed prior to or in parallel to the field deployment effort presented in Chapter 3. As such, data from the newly deployed system did not make their way into this portion of the study. Nevertheless, the algorithms presented herein were applied to the data later on and the results are presented in Chapter 5 of this report.

## *Background*

*Similarity Measures between Two Strings*

The process of matching two strings involves a sequence of comparisons of individual characters to determine the degree of similarity between two strings. In the literature of text mining, the edit distance is one popular technique to measure the similarity between two strings. Given two strings *x* and *y*, the edit distance calculates how many fundamental operations are required to transform *x* into *y*. These fundamental operations are termed as *substitutions* when a character in *x* is replaced by the corresponding character in y; *insertions* when a character in *y* is inserted into *x*, thereby increasing the length of *x* by one character; and d*eletions* when character in *x* is deleted, thereby decreasing the length of *x* by one character [Duda].

The edit distance $d(x \to y)$ between two strings *x*, *y*, can be calculated based on the following recurrent equation, as proposed by Wagner and Fischer (1974):

$$\begin{aligned} d(i, j) = \min\{ & d(i-1, j-1) + \gamma(x_i \to y_j), \\ & d(i-1, j) + \gamma(x_i \to \varepsilon), \\ & d(i, j-1) + \gamma(\varepsilon \to y_j)\}, \end{aligned}$$

**Equation 4. Edit Distance between Two Strings x and y.**

where $d(i, j)$ is the edit distance between *x*[1..i] and *y*[1..j], and *d(0,0)* = 0. The $\gamma$'s are the cost functions. For example, $\gamma(x_i \to y_j)$ is the cost for the change (substitution) from $x_i$ to $y_j$. The $\gamma(x_i \to \varepsilon)$, where $\varepsilon$ represents the empty character, is the cost incurred by a deletion of $x_i$. $\gamma(\varepsilon \to y_j)$ is the cost incurred by an insertion of $y_j$. Thus, the edit distance $d(x \to y)$ would be given by $d(|x|, |y|)$, where the notation |·| corresponds to the length of a string.

Various extensions of the original edit distance measure have been proposed to account for different situations. The original assignment for the cost functions as proposed by Levenshtein (1966) was to set $\gamma(x_i \to y_j) = 0$ if $x_i = y_j$, or $\gamma(x_i \to y_j) = 1$, otherwise ($x_i$ and $y_j$ cannot be $\varepsilon$ at the same time). Ocuda et al. (1976) proposed the generalized edit distance (*GED*) to assign different weights to the edit operations as a function of the character or the characters involved. For example, a cost associated with the edit substitution $"U" \to "V"$ could be smaller than the edit substitution $"Q" \to "V"$. The error rates can be reduced by adjusting the values of the weight for each fundamental edit operation in accordance with the kinds and occurrence probabilities. In addition to weight assignments, Oommen (1986) also proposed to constrain the *ED* by the number and type of edit operations to be included in the optimal edit transformation, and he named this new approached as constraint edit distance (*CED*). The main idea of the *CED* is to search for the optimal *ED* subject to a certain number of substitutions, insertions, and deletions.

The most recent advance in *ED* calculation was made by Wei (2004) who proposes the *Markov edit distance* (*MED*). The main idea is to calculate *ED* according to lengths of sub-patterns and a simple measure that compares how close the histograms of the two sub-patterns are. The cost function in the *MED* is defined as $\gamma(p_1 \to p_2)$, where $p_1$ and $p_2$ are two sub-patterns, which at least one of them is not a single symbol of the alphabet. Wei pointed out that in working with sub-patterns the statistical dependencies among the values assumed by adjacent positions in patterns can be better exploited in such way that a variety of string operations are incorporated, in addition to all operations already defined in previous literatures.

The weight (or cost) functions can play an important role in the calculation of *GED* and *CED* measures. Several authors proposed different ideas to consider the type of errors that may be present in a given application domain. In an application of handwritten text recognition, Seni et al. (1996) introduced additional operations (merge, split and pair-substitution), refined these set of operations as unlikely, likely and very likely, and established the order of importance of the new classification of operations relative to each other. Then, they assigned the cost for each of the classes of operation, e.g., an unlikely deletion is more penalized than a likely deletion.

Marzal and Vidal (1993) computed the weight function using the estimated probability matrix for substitutions, insertions and deletions of any pair symbols of the alphabet for the application of hand written digit recognition. They transformed the probability matrix into weight function by computing the negative logarithm of each probability value.

The *MED*, as proposed by Wei (2004), defines the probability of a certain sequence of operations to convert a string, *x*, into another string, *y*, as a Gibbisian probability distribution function, which in turn is defined as $P(x \to y) = \exp(-U(x \to y)/T)/Z$, where $T$ and $Z$ are constant parameters to be calibrated. The term $U(x \to y)$ is the energy that is required to overcome the distance, i.e., the edit distance, between string *x* and string *y*. The most desirable configuration

for transforming *x* into *y* would be the one that maximizes $P(x \to y)$, which in turn implies the minimization of $U(x \to y)$. *ED* and *GED* are special cases of the *MED*.

*License Plate Matching Application*

We have already presented the concept of license plate matching using Edit Distance in Chapter 2. But it bears to review it some of the key points. Consider, for example, a license plate whose lettering is constituted by the string "4455HZ" which should be read by two LPR machines at different locations. Suppose that at the first location, the plate was read as "4455IIZ" and at the second, "4455HZ." Note that neither LPR unit knows whether it has read the plate correctly. Applying the original idea of *ED* to the LPR example above, for *x* = "4455HZ" and *y* ="4455IIZ" we obtain $d(x \to y) = 2$, whereas for *x* = "4455HZ" and *y* ="445OHZ" we obtain $d(x \to y) = 1$. In the event that *y* ="445OHZ" has been observed earlier at the first station and belongs to the set of candidates to match *x* = "4455HZ," one would claim that "4455HZ" and "445OHZ" is a genuine match, whereas, in reality, they may belong to different plates, since it is unlikely that the pair "O"-"5" would form a match. Thus, in order to improve the matching performance, the *ED* method and the cost (or weight) functions $\gamma$'s should consider the LPR mistakes in reading certain characters. This can be achieved using the extensions of *ED* as found in the literature, combined with proper cost functions for the LPR application.

All LPR misinterpretations can be translated into a matrix of error probabilities where each cell is given the likelihood of certain pair-wise character symbol occurrence. During the LPR machine operation there is a relatively high chance of certain characters (e.g. "1," "0," and "B") being misread (e.g. "I," "O," and "8," respectively), by the LPR machine. Such information can be obtained by constructing a matrix of reading probabilities taken by each LPR unit used. Once the matrix is constructed, the challenge becomes how to design the weights (or the cost function) to be used in the edit distance calculation. The basic idea is that the higher the probability of the error occurrence, the smaller the weight to compensate for the corresponding character error. Therefore, in Equation 4, the unitary values of the cost functions would be replaced by appropriate weight functions based on the confusion matrices of LPR machines.

In designing the weight function, however, one should have in mind that the LPR application is different from common applications in the sense that there is no reference or list of true values to match the target value. For each recognized string in one location there are a set of other recognized strings for matching in another location, and the true plate number is unknown. Therefore, the designed weight function should associate both probability matrices of each LPR machine. It is expected that using such a cost function would increase the number of positive matches and reduce false-positive matches. In this chapter, we propose a suitable weight assignment for comparing strings read by LPR machines located at two points, based on error probabilities of the LPR machines in misreading certain characters.

Analyzing the *GED* and *CED* formulations it seems that *CED* is a more suitable measure to match pair strings recognized by LPR systems, thus reflecting more adequately the most common errors made by the LPR machine used in this study. Considering the possible errors that may happen when recognizing characters of USA plates, the most common errors are as follows: first, some of the plate characters are missed by the LPR machine, consequently, decreasing the length of the original string; second, stickers around the plate frame can be recognized as additional characters, thus resulting in recognized string with longer length; and finally, some of the characters are misread by the LPR machine, thus keeping the same length as the genuine string. As can be seen reversal errors are never expected, and any compensation technique should not reflect such errors.

The *GED* framework on the other hand does not completely account for the actual errors encountered in LPR applications and may sometimes compensate for reversal errors. For example, given two strings *x* = "ABC123" and *y* = "BCA123" which certainly come from different vehicles, under the *GED* framework, $d(x \to y)$ would be computed making a deletion of the first "A" from *x* and a insertion of "A" after "C" into *x*, which would result in two editing operations. Whereas, if compensation for reversal errors is prohibited, one would say that the number of operations in this example would be the 3 substitutions instead: $"A" \to "B"$, $"B" \to "C"$ and $"C" \to "A"$. The latter result could be achieved only if *ED* were constrained by the number of editing operations, such as allowing only substitutions without deletions and insertions in this case.

## *Methodology*

*Weight Scheme Proposed*

In this study we deal with the problem of matching vehicle plates for a single origin-destination, or two-point survey, referred to as station *g* and station *h*. Station *h* is located downstream of station *g*. For any given plate read at station *h*, there are a number of candidate plates already read at station *g* for matching purposes. As will be seen in the following sections, every two recognized strings are matched up to find the best assignment that minimizes an overall cost. To measure the cost of each pair-wise match, the *ED* formulation will be applied with different weight functions.

In designing the new weight functions for the application of vehicle tracking, we assume that the sequence of edit operations to convert a string *x* into a string *y* is independent of each other, i.e., there is no dependence relationship between neighborhood characters of the strings *x* and *y*. This means that in recognizing characters on the plate, the readings of LPR machine may not be affected by the position of the character or by the other surrounding characters.

It is also assumed that matrices containing the likelihoods of character misinterpretation by each machine are available or can be estimated from a dataset containing both reading and true values of the license plate numbers. We name such matrices as confusion matrices.

**Remark**: The confusion matrix is denoted by $\boldsymbol{C}^{l}$ where the element $C_{ij}^{l}$ can represent either the conditional probability $p(r_i \mid t_j)$ that a given true character $t_j$ was recognized as $r_i$ by a LPR machine *l*, or the inverse conditional probability $p(t_j \mid r_i)$ that for a given recognized character $r_i$ its ground truth character is $t_j$. The matrix has as its diagonal elements the probabilities that a character is correctly read and as its off-diagonal elements the misreading probabilities. In our problem of vehicle tracking, each matrix $\boldsymbol{C}^{l}$ is a N by N square matrix where N is the total number of possible alpha-numeric (plus the empty one) characters in which either $t_j$ or $r_i$ may assume (in our application, N is 37 which means 36 alphanumeric characters plus the empty one, representing the missing character, and that makes possible deletion and insertion operations).

*Weigh Function*

Let $x = x_1 x_2 ... x_i ... x_{l_x}$ and $y = y_1 y_2 ... y_j ... y_{l_y}$ be any two sequence of characters read at stations *g* and *h* with string lengths equal to $l_x$ and $l_y$, respectively. Suppose that the two strings are disposed along the axes of a grid, as illustrated in Figure 18, with editing operations represented as the following moves on the grid: downward along the diagonal for substitution, eastward for deletion, and vertical downward for insertion. There is a multitude of editing operation combinations to convert $x$ into $y$, which can be adequately represented by all possible directed paths from the point *(0, 0)* to the point $(l_x, l_y)$ on the grid. If the first assumption above holds, the probability of a given sequence of editing operations to compare $x$ and $y$ is given by the following formulation

$$p(x \rightarrow y) = \prod_{k=0}^{n} p(i_k, j_k)$$

**Equation 5. Probability of a Sequence of Editing Operations for Comparing x and y.**

where, *n* is the number of nodes of an path between $(i_0, j_0) = (0,0)$ and $(i_n, j_n) = (l_x, l_y)$. The $p(i_k, j_k)$ is the probability of the corresponding editing operation associated with the point $(i_k, j_k)$ on the grid, that is the likelihood to observe a character outcome $y_{j_k}$ at station *h*, for a given character outcome $x_{i_k}$ obtained at station g. On the grid, the moves $(i_k - 1, j_k - 1) \rightarrow (i_k, j_k)$, $(i_k - 1) \rightarrow (i_k, j_k)$ and $(i_k, j_k - 1) \rightarrow (i_k, j_k)$ represent substitution, deletion and insertion, respectively.

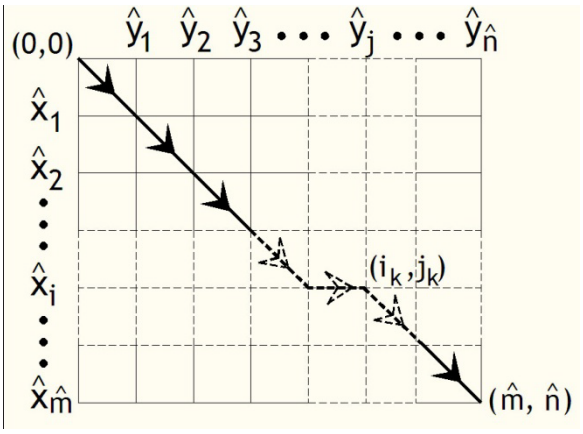


**Figure 18. Chart. Sample Grid of Edit Distance between Two Strings x and y.**

If one makes the negative logarithm in both sides of Equation 5 and minimize the result, we will obtain the following expression

$$d(x \to y) = \min\left\{\sum_{k=0}^{n} \log\left(\frac{1}{p(i_k, j_k)}\right)\right\}$$

**Equation 6. Minimization of Probability for a Given Editing Sequence.**

Indeed, to find the most likelihood match or sequence of editing operations, Equation 5 should be maximized, which implies to minimize its negative natural logarithm.

If $p(i_k, j_k)$ can be estimated from the collected dataset, the proposed weight function can be calculated as $\gamma(i_k, j_k) = \log\left(\frac{1}{p(i_k, j_k)}\right)$. This formulation can be used in existing edit distance measures such as *GED* and *CED*.

The problem now becomes how to estimate $p(i_k, j_k)$. As mentioned before, the context presented in this research differs from existing situations in the sense that there is no true reference string (plate). As will be seen, the method proposed to overcome this problem consists in applying conditional probability theory to associate the misreading probabilities given by the confusion matrices $\boldsymbol{C}^g$ and $\boldsymbol{C}^h$ of station *g* and *h*, respectively, and obtain estimates of $p(i_k, j_k)$ for any possible character association.

***Computation of the conditional probability of association character outcomes***

To estimate the key probability $p(i_k, j_k)$ for the weight function of Equation 6, we need to estimate the probability that the corresponding pair of character outcomes $x_{i_k}$ and $y_{j_k}$ at station *g*

and *h* is actually the exact same character. Such character association likelihood can be estimated on the basis of conditional probability $p(y_{j_k} \mid x_{i_k})$ of observing $y_{j_k}$ at *h* given $x_{i_k}$ at *g*.

To simplify the subsequent description let *x* and *y* now be any character outcome at station *g* and *h*, respectively. Furthermore, let *t* be a true character. Assuming that any character is equally likely to appear anywhere on a plate and that the machines work independently, we can estimate the conditional probability of observing the character outcome *y* at *h*, given a character outcome *x* at *g*, for true characters *t*, as the following expression:

$$p(y \mid x) = \frac{p(x, y)}{p(x)} = \frac{\sum_{t} p(x, y \mid t)\, p(t)}{\sum_{y,t} p(x, y \mid t)\, p(t)}$$

**Equation 7. Probability of Observing y at h given Observing x at g.**

It can be shown that Equation 7 can be rewritten as

$$p(y \mid x) = \sum_{t} p(y \mid t)\, p(t \mid x)$$

**Equation 8. Simplified form of Equation 7.**

where,

$$p(t \mid x) = \frac{p(x \mid t)\, p(t)}{p(x)} \text{ and } p(x) = \sum_{t} p(x \mid t)\, p(t)$$

**Equation 9. Basic Relationships of Conditional Probability.**

Notice that Equation 8 is composed by a summation of products with two factors $p(y \mid t)$ and $p(t \mid x)$ each, which can be viewed as entries of two confusion matrices. Let us define $\boldsymbol{C^g}$ as the confusion matrix whose entries are the values $p(t \mid x)$ and $\boldsymbol{C^h}$ as the confusion matrix whose entries are the values $p(y \mid t)$. Therefore, since by definition the ground truth characters are referred to the columns in matrices $\boldsymbol{C^g}$ and $\boldsymbol{C^h}$, an estimate of all possible character associations, or conditional probabilities $p(y \mid x)$, is given by the following matrix multiplication:

$$\mathbf{C} = \mathbf{C}^{g}.(\mathbf{C}^{h})^{T}$$

**Equation 10. Association Matrix as a Function of Confusion Matrices.**

where,

$(\mathbf{C}^{h})^{T}$ is the transpose of $\boldsymbol{C^h}$.

With index notation, each element $C_{ij}$ of $\boldsymbol{C}$ is therefore given by $C_{ij} = p(y_j \mid x_i)$, where $i = 1, \ldots, N$; and $j = 1, \ldots, N$.

Finally, the probability $p(i_k, j_k)$ in Equation 6 should be approximated by $p(y_j \mid x_i)$ and can be obtained directly by simply searching for the cell in matrix $\boldsymbol{C}$ in which the associated characters correspond to those involved at the editing operation at node $(i_k, j_k)$ on the grid of Figure 18.

*Matching Methodology without Using Passage Time Information*

In this section we describe the problem of matching plate number observations collected by a two-point survey, or dual LPR setup, using an offline procedure without using passage time information. In such, edit distance is calculated for all pair-wise matches between any two datasets provided by the LPR machines so that the set of assignments that minimizes the overall cost is determined. The motivation of finding this matching was to assess discriminative power of different similarity measures. Since the number of pair-wise combinations is expected to be large (number of outcomes in station *g* multiplied by the number of outcomes in station *h*), if any similarity measure is capable of discriminating genuine from false matches in this worst case scenario, we may claim that it is a good similarity measure for LPR application.

Our proposed vehicle tracking based on LPR technology, which can be viewed as a weighted bipartite matching problem, can be summarized as follows. First, for each outcome at station *h*, a vector with length equal to the number of pair-wise matches formed with all outcomes at station *g* is constructed, where each element is the similarity measure (edit distance) between the corresponding outcomes; second, the assignment with the least cost value is selected as a potential match; finally, a threshold on *ED-values* is used to discriminate the resulting pair-wise matches between potential positive and false matches. Figure 19 shows a flowchart of this procedure.

Notice that the number of observations in the two sets can differ as some vehicles either do not pass through the two stations or they may not have their plates recognized by either one of the two LPR stations. The result of this is an increasing chance of having false matches being detected as genuine.

*Vehicle Tracking Considering Passage Time Information*

Considering the two-point survey again, in this section we propose a matching procedure incorporating the passage time information, or time stamps, to improve the performance of the template matching. This procedure is to be used in situations where it is needed to decide whether or not a plate currently detected at downstream station *h* can be matched to a subset of plate detected at upstream station *g*. Such applications can involve speed enforcement or online estimation of travel time for information systems.

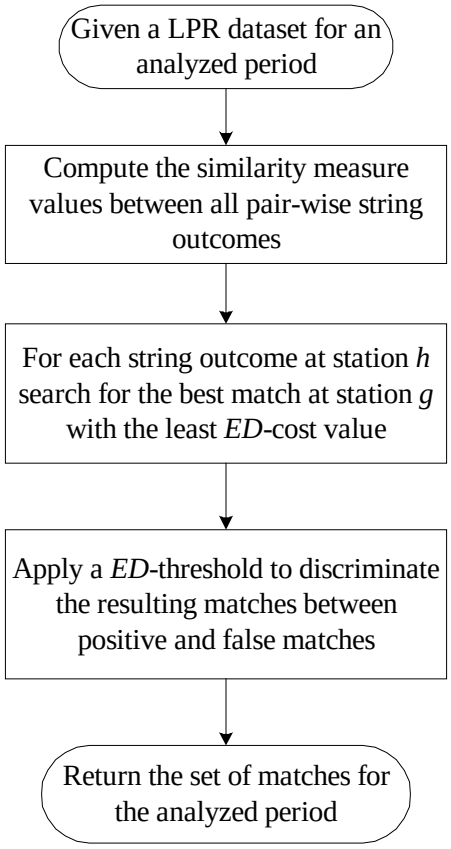


**Figure 19. Chart. Procedure for Identifying the Most Likely Match Pair.**

In essence, our proposed matching procedure consists in matching any current outcome $y_j$ (*j* is an index to keep track of the outcome location in the dataset) at station *h* to a subset of the earliest previous observations at station *g*. The subset of candidates at station g is formed by those outcomes whose corresponding passage time falls within a time window constraint. Such time window constraint is bounded by the upper and lower limits of the expected travel times on the road. Furthermore, the width of this time window constraint is adequately reduced every time there exists a high chance for the similarity measure to classify a match as genuine, when in reality it was false.

Initially, let us define the notation used. Set the pair-wise string $(x_i, y_j)$, as a potential match – where $x_i$ is the *i*th outcome observed at station *g* and $y_j$ is the *j*th outcome read at station *h*. That means that $x_i$ is the best earliest previous observation at station *g* to match $y_j$. In this case, the journey time of the corresponding vehicle will be estimated by the time-difference between the passage times recorded at station *g* and *h*, i.e. $t_{ij} = v_j - u_i$, where $u_i$ and $v_j$ are the corresponding time stamps at station *g* and *h*, respectively. Furthermore, let us define the range of values that the similarity measure $d_{ij} = d(x_i \to y_j)$ can assume as $[0, \tau^{\max}]$, where $\tau^{\max}$ is the maximum

possible value of $d_{ij}$ in which we may still declare $(x_i, y_j)$ as a genuine match. Finally, let us define the range $[0, \tau^*]$, such that $\tau^* \in (0, \tau^{\max})$, in which it is very likely that $(x_i, y_j)$ constitutes a genuine match.

The matching procedure, as shown in Figure 20 and Figure 21, can be described by the following steps: 1) Match any current observation $y_j$ at station *h* to a subset of the earliest previous observations at station *g* corresponding to the survey period $\Delta t^g$, and search among the candidates for the best string $x_i$ with the least edit distance; 2) If $d_{ij} \le \tau^*$, we declare the match $(x_i, y_j)$ as genuine; 3) Otherwise, if $\tau^* < d_{ij} \le \tau^{\max}$, we declare $(x_i, y_j)$ a valid match only if the estimated journey time $t_{ij}$ lies within a time constraint estimated from a sample of previously observed passage times from matches already classified as genuine. Such sample is collected looking at all matches obtained for a survey period $\Delta t^h$ in station *h*.

The survey period $\Delta t^g$, or time window constraint, at *g* to establish each subset of outcome candidates is given by $\Delta t^g = (v_j - jt_u, v_j - jt_l)$, where $jt_u$ is an upper bound for the journey time while $jt_l$ is a lower bound. Therefore, if $v_j - jt_u \le t_{ij} \le v_j - jt_l$, the corresponding match $(x_i, y_j)$ is classified as potential match, but not as genuine yet.

Assuming that the genuine journey times come from a symmetric density function, such as the normal distribution, it is quite true that the closer $t_{ij}$ is to the mean of the distribution, more likely the match $(x_i, y_j)$ is to be genuine. Also, it is known that the likelihood of having a genuine match increases when the similarity measure decreases. Therefore, to define the travel time constraint acting on the *ED* domain $(\tau^*, \tau^{\max}]$ we define the following inequality constraint

$$\left| \frac{t_{ij} - \mu^{jt}(\Delta t^h)}{\sigma^{jt}(\Delta t^h)} \right| \le z\left(d_{ij}\right), \quad \tau^* < d_{ij} \le \tau^{\max}$$

**Equation 11. Travel Time Constraint.**

where, $\mu^{jt}(\Delta t^h)$ and $\sigma^{jt}(\Delta t^h)$ are the moving average and standard deviation of the journey times for the corresponding earliest period $\Delta t^h$ of analysis and $z(d_{ij})$ is the number of standard deviation to define the interval limits which is a monotonically decreasing function of the similarity measure value $d_{ij}$.

Parameters $\mu^{jt}(\Delta t^h)$ and $\sigma^{jt}(\Delta t^h)$ can be estimated from a sample of passage times from matches classified as genuine during a previous survey period $\Delta t^h$ at *h*, or from matches

obtained in previous days during the same survey period $\Delta t^h$. Outliers should be eliminated from the sample if they lie outside a constraint interval calculated around the sample median [Clark].

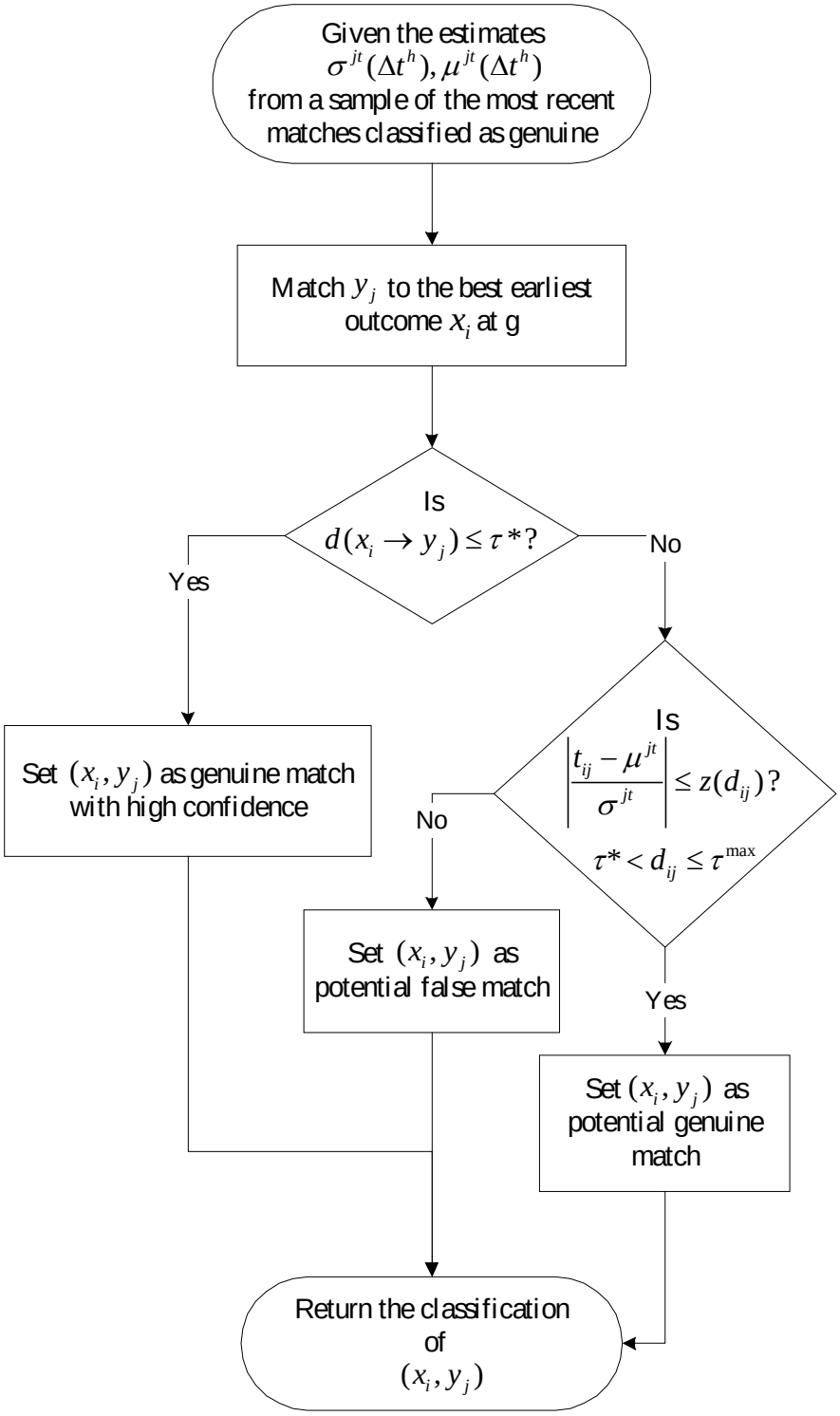


**Figure 20. Illustration. Proposed Matching Procedure.**

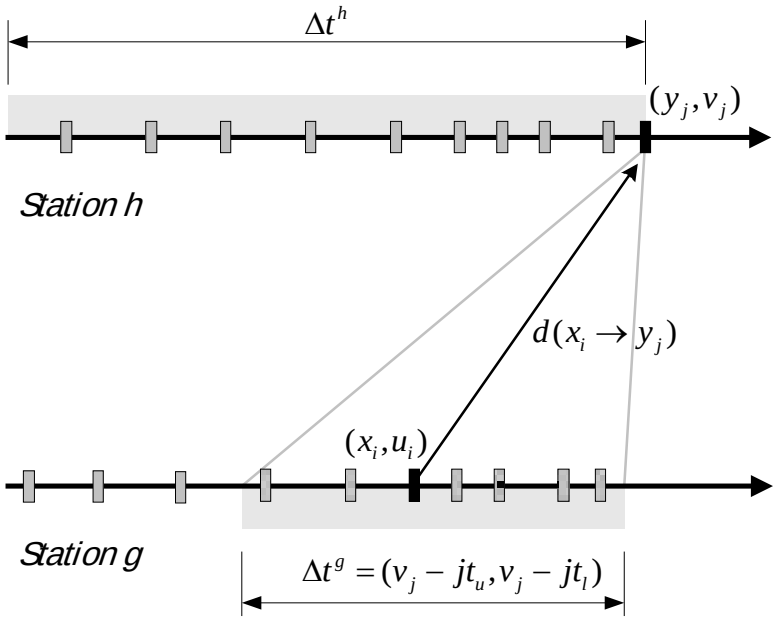


**Figure 21. Illustration. Travel Time and Time Window Considerations.**

*New Editing Constraints*

LPR machines usually do not reverse the characters on the plates. For this reason it is very likely that any pair of read strings can have its sequence of characters lined up if they come from the same vehicle. Thus, considering that reversal errors are not made by LPR machines, the *CED* with editing constraints defined as a function of the string lengths may potentially eliminate false positive matches that otherwise would be obtained if a *GED* formulation were used.

Therefore in this research, it is worth noting that the edit operation constraints used in *CED* are defined based on the length differences of the strings being compared. Hence, for any pair of read strings $x$ and $y$, with lengths given by $l_x$ and $l_y$, we propose the following constraint sets *(i,e,s)* of insertions, deletions and substitutions to transform $x$ into $y$.

$$(i,e,s) = (l_y - l_x, 0, l_x), \ \text{if} \ l_y > l_x;$$

$$(i,e,s) = (0, l_x - l_y, l_y), \ \text{if} \ l_y < l_x;$$

$$(i,e,s) = (0, 0, l_x), \ \text{if} \ l_y = l_x.$$

**Equation 12. Restrictions of Editing Operations.**

The three restrictions above state that insertions or deletions will be allowed only if the lengths of two strings are different, otherwise only substitutions will be allowed. Hence, the use of *CED* with these constraints enables us to find the most likely character alignments between a pair of strings.

## *Case Study and Experimental Results*

The aforementioned methods were implemented to match plates from our Phase A dataset, which consists of large truck plates captured at two sequential stations along I-40.

*Performance of Vehicle Tracking Procedures*

The dataset was divided into two parts: one for calibration of the model parameters and the other for comparison of the performances of the similarity measures. Since there were 5 days of data, all combinations of 3 datasets out of 5 were used as calibration data, with the remaining combinations with 2 datasets as validation data. Thus, each of the 10 combinations with three days of data was used to estimate 20 confusion matrices, i.e., 10 matrices of type $\boldsymbol{C^1}$ and 10 matrices of type $\boldsymbol{C^2}$ for LPR stations 1 and 2, respectively. These confusion matrices were then included into the formulation of *CED* and *GED* in combination with our proposed weight functions defined previously.

Considering the possible ways of defining the editing weights into the recurrent calculation of *ED* (see Figure 18), four procedures were indentified to calculate the *ED* between pair of strings, as follows:

- **$D_1$**: Edit distance with zero or one cost assignments, which corresponds to the original idea of Levenshtein;
- **D2**: *GED* using weight function as in Equation 6, with $p(i_k, j_k)$ estimated by $p(y \mid x)$, as defined in Equation 8;
- **D3**: Original *CED* with zero or one cost assignments and constrained by the editing sets defined in Equation 12;
- **D4**: *CED* using weight function as in Equation 6 and $p(i_k, j_k)$ as in Equation 8, and constrained by the editing sets defined in Equation 12.

The performance of our proposed procedures, D2 and D4, were then compared to the popular *ED* and *CED* methods, D1 and D3. All four procedures above were then applied to all 10 combinations of two remaining days of data, used as validation period.

As mentioned earlier, the performance of the similarity measures was investigated under a worst case scenario that consisted in matching up the two sets of plates for each remaining day, without using passage time information or the recorded time stamps. The main premise there was that under this worst-case scenario the most suitable measure for LPR application should be able to accurately match every two set of plates with fewer false matches.

Regarding the measures of performance, the percentage of positive matches and the percentage of false matches were calculated for a range of *ED*-thresholds covering the domain of all possible *ED* values, ranging from 0 to 20. In order to derive the performance measures, it was

necessary to obtain the ground truth values of the plate numbers by manually recording them when visualizing their images provided by the LPR datasets. The efficiency of each similarity measure was then established by drawing curves relating the percentage of correct matches to the percent of false matches over the domain of *ED* values. Thus, ten such charts were determined and all of them presented similar results as the chart shown in Figure 22 for one of the possible combination of datasets.

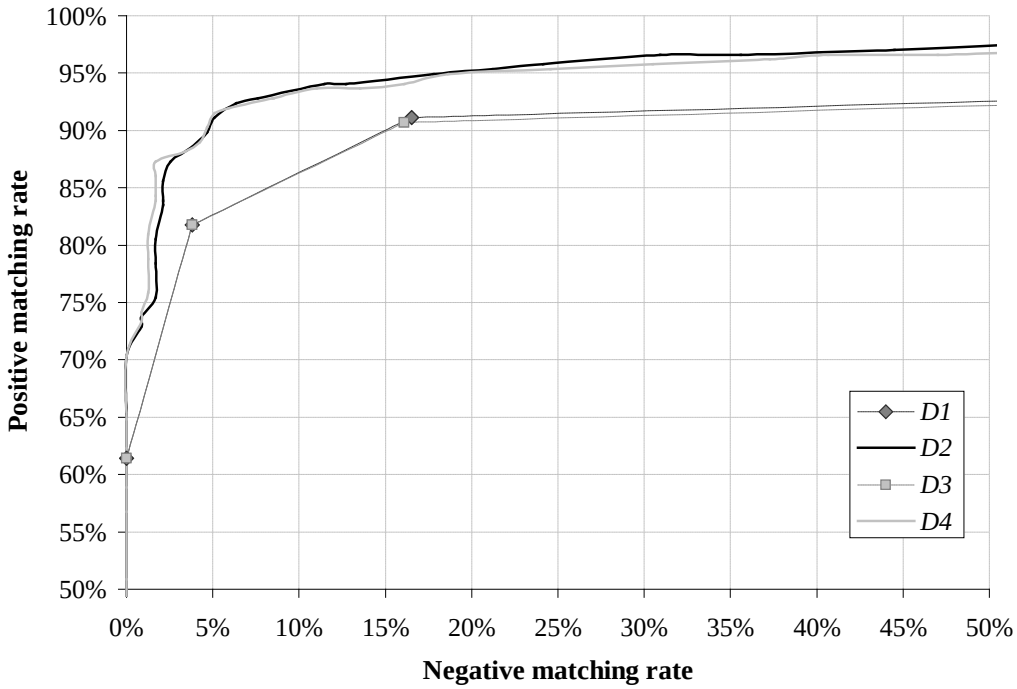


**Figure 22. Chart. Efficiency of Similar Measures.**

The first thing to notice in Figure 22 is that existing *ED* or *CED* combined with our proposed weight functions yielded considerably improved performance for vehicle tracking. Second, with respect to two measure frameworks, *GED* and *CED*, there was not any evidence of difference in performance between these two measures over all 10 analysis performed. Therefore, there is not empirical evidence yet to state that *CED* equipped with the proposed editing operation constraints, as defined in Equation 12, is a better procedure to compensate for the mostly common errors made by LPR machines.

In general, either D2 or D4 measures were able to achieve around 90% of positive matches with about 5% to 8% of false matches. In addition, it is worth noting that these measures achieved almost 80% of positive matches with approximately 1% to 2% of false matches. Thus, it seems that any *ED* formulation equipped with the proposed weight functions has the most discriminative power to match data from LPR systems when the target or reference values for matching are unknown.

*Performance of the Online Vehicle Tracking Procedure*

In this section the efficiency of the proposed online matching procedure, which incorporates the passage time information was assessed. This procedure was combined with the most suitable similarity measure chosen before, calibrated and evaluated using the 10 sets of data combinations.

As for the parameters of the online procedure summarized at the flowchart of Equation 10, it was observed from the calibration datasets that the best threshold values $\tau*$ and $\tau^{\max}$ would be 5 and 20, respectively. The time window constraint $\Delta t^g$ was defined assuming an upper bound for the vehicle speed of 90 mph and a lower bound of 35 mph. The moving averages and standard deviations were calculated from every previous time block $\Delta t^h$ containing at least 5 matches classified as genuine. From the calibration datasets, was established that $\Delta t^h = 10$ min was a suitable survey period for this purpose. Outliers were removed based on the procedure proposed by Clark et al. (2002) where any journey time value lying outside the interval determined by Equation 13 was classified as an outlier.

$$M_e \pm 3 \times \frac{\sum_{i=1}^{n} | JT_i - M_e |}{n}$$

**Equation 13. Outlier Boundaries Determination.**

where

$JT_i$ = journey time for vehicle *i*,

$M_e$ = median for each 10-min block of journey times, and

$n$ = number of observations inside each block.

The number of standard deviations given by the function $z(d_{ij})$, which establishes the time window constraint acting over the domain $(\tau*, \tau^{\max}]$ was defined by the quadratic function of Equation 14 below

$$z(d_{ij}) = \sqrt{9 \times \frac{\tau^{\max} - d_{ij}}{\tau^{\max} - \tau*}}$$

**Equation 14. Number of Standard Deviations for Time Window Constraints.**

Equation 14 guarantees that at the largest time constraint, when $z = 3$, $d_{ij}$ is equal to its lowest value i.e. $d_{ij} = \tau*$. Whereas when $d_{ij} = \tau^{\max}$ the time window constraint vanishes, meaning that when the similarity measure approaches its highest possible value the analyzed match will be

considered genuine only if the time-difference $t_{ij}$ is exactly equal to the mean estimated journey time.

After applying the above procedure combined with measure D2 to all 10 combinations of two days of data used for validation, it was in general achieved between 95% and 98% of genuine matches and about 0.5% to 1.5% of false matches. As a base scenario for comparison, we calibrated and applied the same procedure using measure D1. As a result, for the calibrated parameters $\tau^* = 0$ and $\tau^{\max} = 5$, we achieved a positive matching rate between 93.5% and 96.5% but with false matching rate of about 2.2% to 3.6%.

## *Conclusions*

We proposed a new procedure to compute the weight functions that can be used in existing distance measures when the confusion matrix of the LPR machines are available or can be estimated. The experimental results and analyses showed that the most suitable procedures for vehicle tracking based on the plates recognized by a dual LPR setup are either *GED* or *CED* formulation combined with the weight function and editing constraints proposed in this paper. These procedures have achieved about 90% of positive matches with only 5% to 8% of false matches. This represents a promising result for transportation planning purposes, or even for automatic traffic speed monitoring.

When the travel information is incorporated as second level constraint into the matching procedure, it observed a great gain in performance. The proposed method achieved about 95% to 98% of positive matches with about 0.5% to 1.5% of false matches. Although the matching rate was increased significantly when using the travel time information, further work is needed to validate this procedure for other situations of traffic conditions and setups of the LPR machine. The procedure was applied during a period with slight traffic variation resulting in small dispersion of travel times, which might have contributed to this good performance. In addition, the stations were setup relatively close to each other, so that there was no major source of traffic disturbance to disperse the travel times as most vehicles (trucks in this case) travelled in direct fashion through two station sections. Further studies are needed to extend this study for the case of multiple setups of LPR units, such as in a large area, to estimate origin-destination demand.

Another issue of interest is the sample size (number of outcomes) needed to estimate the character misreading matrices $\boldsymbol{C}^{\boldsymbol{g}}$ and $\boldsymbol{C}^{\boldsymbol{h}}$. These estimates are site-dependent as they reflect the characteristic (plate, vehicle, environment, etc) of the locations where the LPR machines are installed. Therefore, if there are many sources of variation, noise, to estimate the misreading probabilities, it would be necessary to have a large amount of data to achieve a required error precision. Such, estimation would also require more human intervention since the ground truth values of the plates are determined manually.

In situations where the LPR equipments are installed permanently, the human effort can be eliminated if there is a mechanism of estimating the conditional probabilities for the association matrix $\boldsymbol{C}$ by means of a learning process without human intervention. The sample size is still a concern in this case, however, now the system can keep learning until an error precision is finally reached, or the conditional probabilities converge to values within a threshold. This will be subject of further studies.

# Chapter 5 – Analysis Results

Chapter 2 of this report gave an in-depth presentation on how text-mining approaches, e.g. Lichtenstein Edit Distance, were applied with success to the plate-match task at hand. Edit Distance algorithm is usually applied to situations where text strings with uncertain accuracy were compared against "true" string values. The task of this research, however, is to match pairs of text strings with no "true" values available. In other words, we constantly face the challenge of not knowing for sure if a match, based on the algorithm, was correct or not. Efforts were concentrated on 1) raising correct matching rate, 2) reducing false matching rate, and 3) increasing matching speed.

During this study, we built on the algorithm we developed previously in Chapter 2 and developed a number of improved algorithms, which were presented in Chapter 4. These algorithms were implemented on the captured plated data from the new "permanent installation" as detailed in Chapter 3. The process of painstaking data analyses and the results are presented in this Chapter.

## *Collection of Ground Truths*

An automated license plate data importing and editing software within Microsoft Excel® was developed to allow the gathering of ground truth by students (see Figure 23). Ground truths of license plates are crucial to this research for three reasons:

1. Establish baseline LPR accuracy.
2. Derive "truth matrices" (as well as "confusion matrices") for LPR.
3. Provide positive matching rates.

A total of 120 hours, or about 40,000 license plates were processed individually taking about 150 hours of manual groundtruthing effort altogether.

## *Character Recognition Accuracy*

The accuracies of the two LPR units were around 26% and 54%. These are significantly lower than the 61% and 63% in Phase A of the study. The reason for the low accuracies is we did not calibrate the image processing units after installation. LPR accuracy can be improved if one calibrated the unit to favor plates from certain states or certain colors. However, we purposely kept the LPR units in their generic form to emulate scenarios where 1) they are deployed without prior knowledge of state mixes, plate colors, reflectivity, designs, etc., 2) designs of license plates change significantly over time, and 3) the accuracy of LPR units deteriorate naturally over time. On one hand, this certainly makes the matching task much more challenging than those presented in Chapters 2 and 4. On the other hand, we have a much more realistic situation to work with. The individual character reading error rates are shown here in Figure 24. It should be noted that even though the frequencies of zero misread characters (the first columns on the left of the chart) are relatively low, the frequencies of zero, one, and two misread characters (the

first three columns on the left of the chart) are quite high collectively. With the subpar reading accuracy, we strive to attain a matching rate of over 90%.

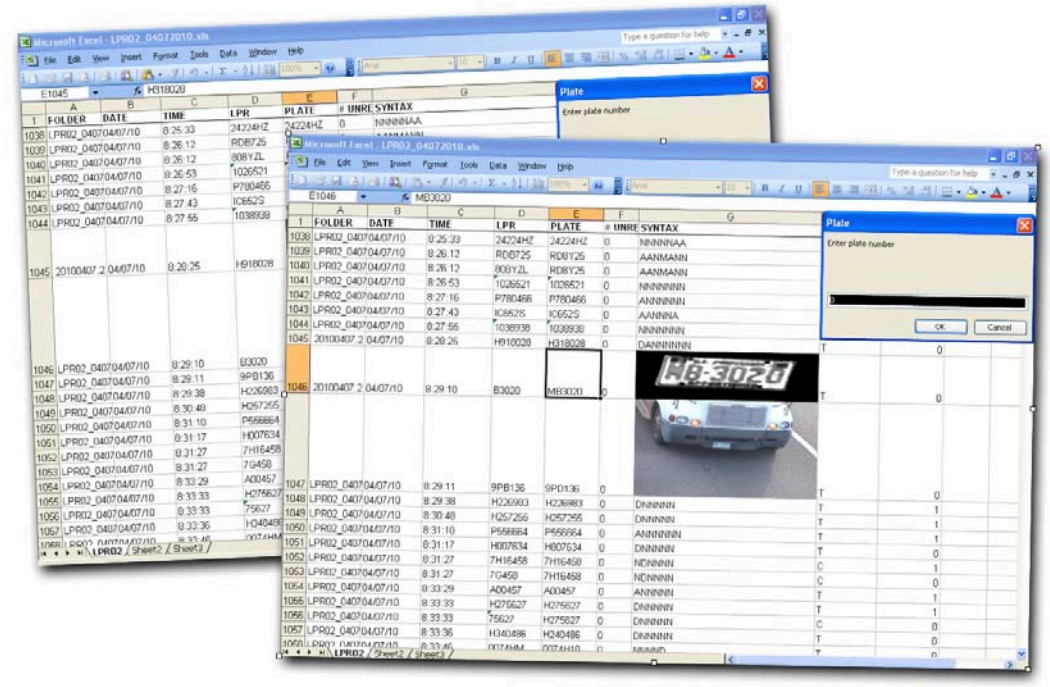

**Figure 23. Photograph. Excel-based Ground Truth Collection Software.**

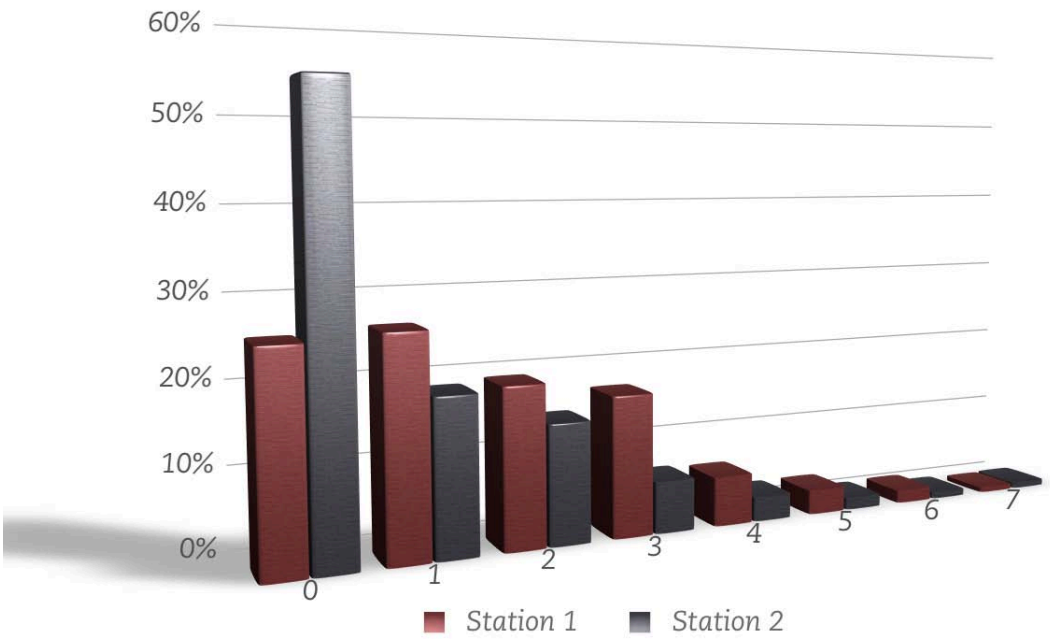


**Figure 24. Chart. Frequencies of Number of Characters Recognized Erroneously.**

## *Development of Truth Matrices and Association Matrix*

With the ground truth extracted, individual truth matrices were developed for the LPR stations. Based on the truth matrices and Bayesian probability, we derived an association matrix for the probability of a character being read as X at station 1 and Y at station 2. The association matrix is the key component for implementing non-integer Edit Distance calculations.

It should be noted that the association matrix may be different for a different time of day, day of week, month of year and can change over time. The association matrix provided here(see Table 4) is actually of significant value to LPR manufacturers. It is unfortunate that for most cases users of LPR technology do not have an association matrix to work with because of the time and resources required to derive them. The use of an association matrix to help improve plate-matching rate is detailed in Chapters 2 and 4.

**Table 4. Association Matrix Derived from Ground Truths.**

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | _ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 73 | 1 | | | | | 1 | | 2 | 1 | | 1 | | 3 | | | | | 1 | | | | | | 10 | | 1 | | | | 1 | | | | | | | 4 |
| 1 | 1 | 67 | | 2 | 1 | | | 3 | 1 | 1 | 1 | | | | 1 | | | 1 | 7 | 1 | | 1 | 2 | | | | | | 1 | | | 1 | | | | | 4 |
| 2 | | | 85 | 1 | | | | 1 | | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | 7 | 1 |
| 3 | | 2 | 1 | 89 | | | | 1 | | 2 | | | | | | | | | | 1 | | | | | | | | | | | | | | | | | 1 |
| 4 | | 1 | | | 83 | 2 | 3 | | | 1 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | 1 |
| 5 | | | | 1 | 2 | 86 | 1 | | | 1 | | | | | | | | 1 | | | | | | | | | | 4 | | | | | | | | | 1 |
| 6 | 1 | 1 | | | 1 | | 83 | | 2 | | 1 | 1 | | | 1 | 1 | 4 | | | | | | | | | | | | | | | | | | | | 1 |
| 7 | | 1 | 2 | 2 | | | | 82 | | | | | | | | | | 1 | 1 | | | | | | | | | | | 2 | | | | | | 4 | 1 |
| 8 | 1 | | | | 1 | | 1 | | 80 | 1 | 1 | 8 | | | | | | 1 | | | | | | | 1 | | 1 | | | | | | | | | | 1 |
| 9 | 1 | 1 | 1 | 1 | 1 | | | | 2 | 81 | | 1 | | | | | | 4 | | | | | | | 1 | | | | 1 | | | | | | | | 1 |
| A | 2 | 3 | 1 | 2 | 17 | 2 | 9 | 1 | 2 | 1 | 42 | 1 | | | 1 | 1 | | 1 | 1 | 1 | 3 | | | 1 | 1 | | | 1 | | | | | | 1 | 1 | | 2 |
| B | 2 | 2 | 1 | 2 | 1 | | 1 | | 41 | 2 | 1 | 31 | | 1 | 1 | | | 1 | 1 | | | | | 1 | 1 | | | 3 | | | 2 | | | | | | 2 |
| C | 5 | 1 | 1 | 1 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | | 71 | | 1 | 1 | 2 | | 1 | | | | | | 2 | | | | 1 | | | | | | | | 3 |
| D | 13 | 1 | | | | | 1 | | | 2 | 1 | 2 | | 65 | | | | | 1 | 1 | | | | | 4 | 1 | | 2 | | | 1 | | | | | | 3 |
| E | | 2 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 1 | 3 | | 1 | | 71 | 2 | | | 1 | 1 | 3 | | 1 | | | 1 | | | | | 1 | | | 1 | | | 3 |
| F | | 1 | 1 | | | 3 | 3 | 2 | | | 1 | 1 | 1 | 1 | 8 | 69 | | 1 | | | | | | | | 2 | | | | | | | | | | 1 | 4 |
| G | 1 | 2 | 2 | | 1 | 1 | 48 | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 | 23 | 1 | 1 | 1 | | | | 2 | 1 | 1 | | 1 | 1 | | 1 | | 1 | | | | 2 |
| H | | | | | | | | | 2 | | 1 | 2 | | | | | | 80 | 1 | | | | 3 | 3 | | | | 1 | | | | | 4 | | | | 3 |
| I | 4 | 42 | 2 | 4 | 3 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | | 1 | 2 | 1 | | 2 | 11 | 2 | | 1 | 1 | | 1 | | | 1 | | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 4 |
| J | 2 | 24 | 1 | 11 | 3 | | 1 | 3 | 1 | 1 | 1 | 1 | | | | | 1 | 1 | 3 | 40 | | | 1 | | | | | | | | | | | | | | 3 |
| K | 2 | 1 | | | 2 | 1 | 2 | 2 | | 1 | 1 | | | | 8 | | | | | | 62 | | | 2 | | | | 3 | | | | 1 | 2 | | | 8 |
| L | 1 | 9 | 2 | 1 | 5 | 3 | 3 | 1 | | 1 | 2 | 1 | 1 | 1 | 2 | 1 | | | 3 | 1 | | 50 | | | 1 | | | | | 1 | 2 | | 1 | | | 1 | 6 |
| M | | | | 1 | 2 | | 1 | 2 | 4 | 2 | 1 | 1 | | | | 1 | | 1 | 1 | | 3 | | 55 | 5 | | | | | | | 1 | 5 | 4 | | | 6 |
| N | | 2 | | | | 1 | | | 3 | | 6 | | | | 1 | | | 2 | 1 | | | | 2 | 74 | | | | 1 | | 1 | 1 | | 3 | | | | 1 |
| O | 52 | 3 | 1 | 1 | 2 | | 2 | 1 | 2 | 2 | 1 | 1 | | 5 | | | | 2 | 1 | | | | 1 | | 17 | | 1 | | | | | | | | | | 4 |
| P | | | 2 | | | 1 | | 1 | | 2 | 1 | | | 1 | 1 | 1 | 1 | | | | | | | | | 81 | | | | | | | | | | 1 | 4 |
| Q | 22 | 5 | | | | | | | 12 | 33 | | 1 | | | | | | | | | 1 | | | | 15 | | 6 | | | | | | | | | | 2 |
| R | 2 | 2 | 2 | | | | 1 | | 11 | 2 | 1 | 4 | | | 1 | | | | 1 | 1 | 1 | | | 2 | | 1 | | 56 | | 1 | | | 1 | | | | 8 |
| S | 1 | 4 | 2 | 3 | 2 | 35 | 1 | 1 | | 7 | | 1 | | | | 1 | | 3 | | 1 | | | | | | | | | 32 | | | | | | | | 3 |
| T | | 4 | 3 | 2 | 1 | 1 | 2 | 23 | | | 2 | 1 | 1 | | 1 | 1 | | 2 | 4 | 1 | | 2 | | 1 | | | | | 1 | 28 | 1 | 1 | | | 7 | 2 | 7 |
| U | 14 | 4 | 1 | 1 | | | | 1 | 2 | 2 | 2 | | | 1 | | | | 2 | 2 | | | 1 | 1 | | 4 | | | 1 | | | 55 | 1 | | | | | 3 |
| V | 1 | 2 | 1 | | | | 2 | | | 4 | 2 | 1 | | | | | | 2 | 1 | 1 | | | | | | | | | | | | 1 | 73 | 2 | | 2 | | 3 |
| W | | 5 | 1 | 1 | 2 | 1 | 1 | 1 | 3 | 1 | 1 | 1 | | | 1 | | 1 | 8 | 1 | 1 | 1 | | 8 | 5 | | | | 1 | | | 1 | 2 | 46 | | 1 | 1 | 4 |
| X | 1 | 3 | 1 | 1 | 1 | | 1 | 2 | | | 1 | | | | | | | 1 | | | 1 | | | | | | | 1 | | 1 | 1 | 1 | | 83 | 2 | | 1 |
| Y | | 3 | | 1 | | | | 2 | | 2 | 1 | 1 | | | | | | 4 | 1 | | | | 1 | | | | | | | 3 | | 4 | 1 | 1 | 55 | | 21 |
| Z | | 1 | 15 | 1 | 1 | 1 | | 7 | 1 | 1 | | | | | | | | 1 | | | | | | | | | | | | | | | | | | 66 | 2 |
| _ | 8 | 10 | 13 | 4 | 9 | 4 | 4 | 3 | 3 | 2 | 1 | 1 | | 1 | 1 | 1 | 1 | 14 | 2 | 1 | | 1 | 1 | 1 | 1 | 2 | | 1 | 1 | | | | 1 | | 1 | 2 | 3 |

## *Truck Travel Time and Speed Calculation*

We used our plate-matching algorithm to calculate the travel times and estimate the speeds of individual vehicles traversing the 3-mile section from I-640 to I-40. The speed data fluctuated throughout the day and dipped during morning peak hours(see Figure 25). The speed data were verified against TDOT's ITS RTMS speed data, which we also collected as a part of this research project.

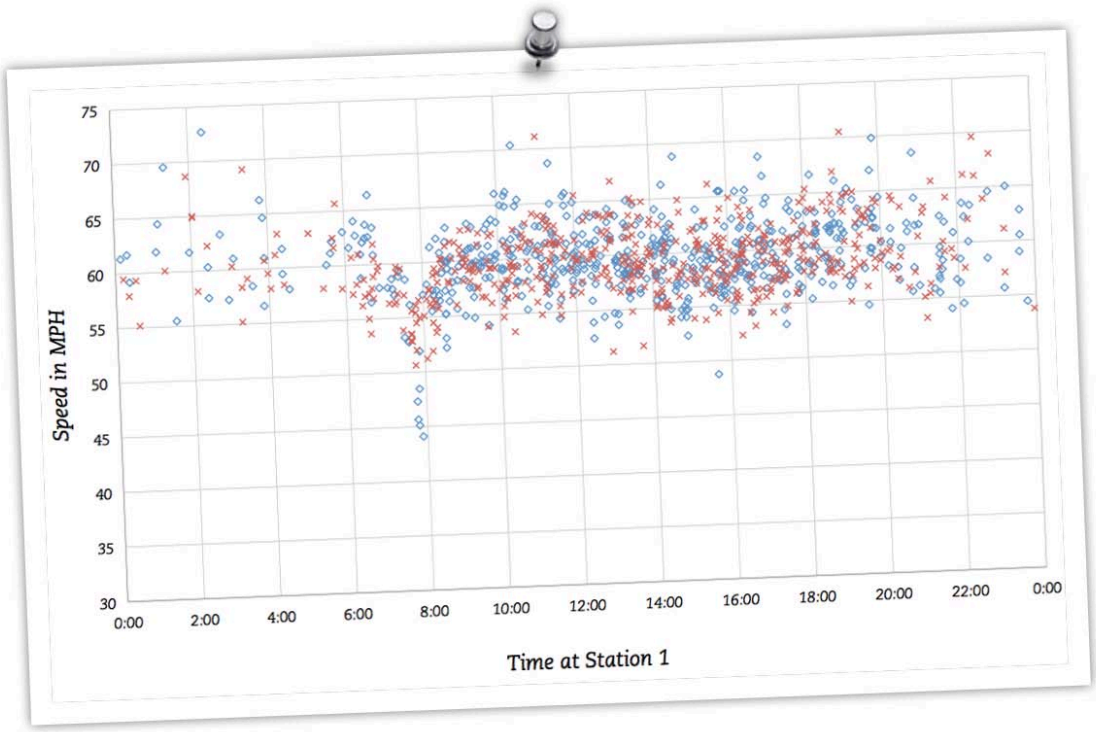


**Figure 25. Chart. Truck Speed Fluctuation over a 24-hour Period.**

Even though the 3-mile study section has a significant 90-degree right turning curve, a major merging area, and at least one lane-changing maneuver required for the large trucks, 93% of the trucks were speeding over the posted speed limit of 55 miles per hour (see Figure 26). If this two-LPR unit configuration were installed on a straight section of the Interstate system, one would probably expect near 100% of the trucks speeding over the 55 mph limit and a significant portion of them exceeding the speed limit by more than 10 mph.

## *Improved Matching Rates*

The matching rate without using our series of matching algorithms is less than 40%. With the algorithms we developed in Chapter 4 they improved significantly to achieve 97% or higher matching rates and very low false matching rates (see Table 5). These results are better than any achieved in literature. Even more impressive than the matching rates are the very low false matching rates we have achieved with the more sophisticated matching algorithms. With a false matching rate of less than 1%, tremendous economy is accomplished.

The *Olieveira-Han Learning Algorithm* not only yielded the best performance in both high positive matching rates and low false matching rates, but the performance continues to improve over time as the algorithm continues to learn on its own in the field (see Figure 27). The performance is also very stable over time without unexpected fluctuation

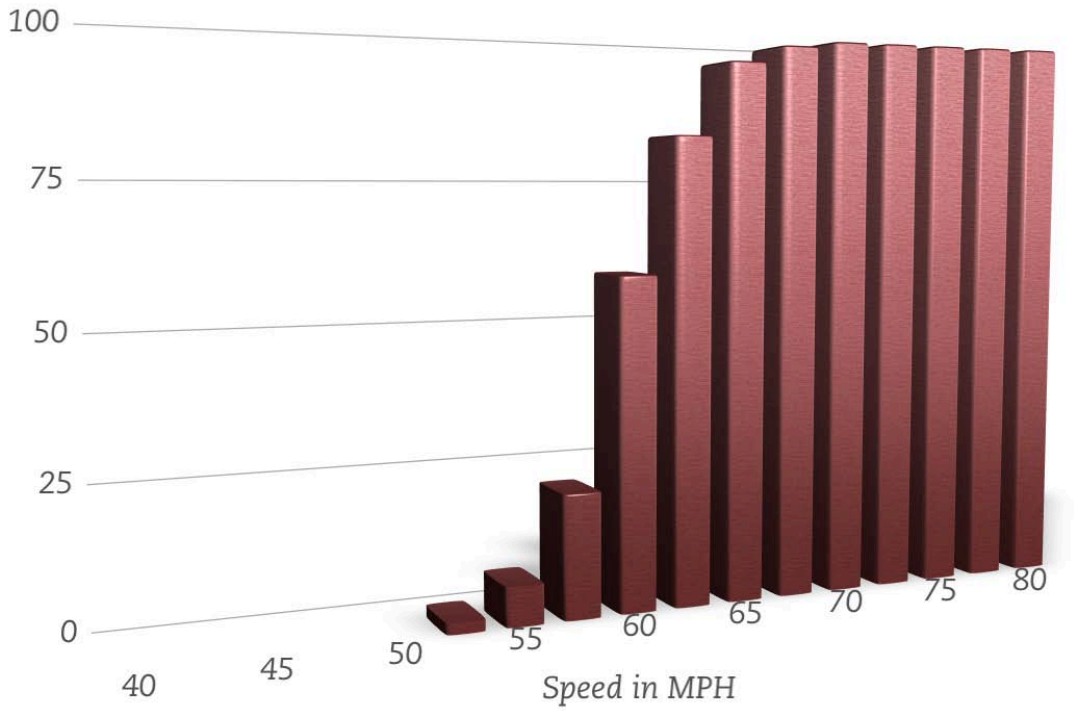


**Figure 26. Chart. Truck Speed Distribution in Study Area**

**Table 5. Plate Matching Accuracy Results.**

| | | Test Period | True Matches | Correctly Matched | Falsely Matched | Positive Match Rate | False Match Rate |
|---|---|---|---|---|---|---|---|
| Base | Exact matches only | 1 | 623 | 239 | 0 | 38.36% | 0.00% |
| | | 2 | 576 | 212 | 0 | 36.81% | 0.00% |
| ED | Edit Distance with ground truth, GT* | 1 | 623 | 588 | 42 | 94.38% | 6.67% |
| | | 2 | 576 | 562 | 27 | 97.57% | 4.58% |
| GED | Generalized ED with ground truth | 1 | 623 | 610 | 12 | 97.91% | 1.93% |
| | | 2 | 576 | 570 | 8 | 98.96% | 1.38% |
| GEDL | GED without GT using OH auto-learning | 1 | 623 | 611 | 5 | 98.07% | 0.81% |
| | | 2 | 576 | 567 | 4 | 98.44% | 0.70% |

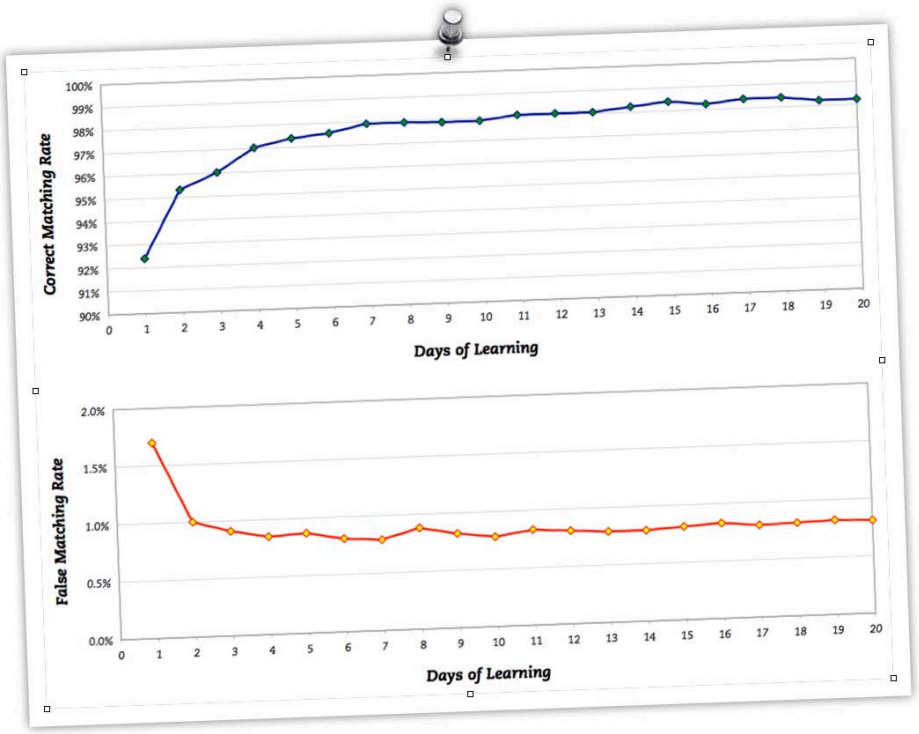


**Figure 27. Chart. Performance of Oliveira-Han Automated Learning over Time.**

# Chapter 6 – Conclusions

This study has successfully accomplished the proposed tasks and concluded that automated truck speed enforcement on Interstate highways using license plate recognition technology (LPR) is highly feasible (with over 97% positive-matching rate and less than 1% false-matching rate), even when LPR performance is, at times, less than desirable (less than 30% of accuracy). The points below suggest that we move forward towards deploying the algorithms.

- Many metropolises reduced the posted speed limit for large trucks for the purposes of air quality and safety;
- To derive benefits from such action, the reduced speed limit has to be enforced;
- A large-scale enforcement of the new speed limit is often extremely challenging due to fiscal and human resource constraints;
- Large trucks are required to enter weigh stations;
- By tracking these trucks at various locations along the Interstate highways, warnings and citations could be issued when they stop at the weigh station;
- Most tracking systems are either too expensive, non-universal, or still in development;
- All trucks are required to have license plates;
- Even though LPR technology is not perfect, the text-mining algorithms developed in this study can match a high percentage of truck plates and, also, determine their speeds;
- It would be desirable to proceed with a real-time LPR large truck tracking and speed monitoring study before the eventual deployment.

# Chapter 7 – References


- Clark, S. D., Grant-Muller, S., Chen, H., 2002. Cleaning of Matched License Plate Data. Transportation Research Record 1804, 1-7.
- Duda, R.O., P.E. Hart, and D.G. Stork (2000). *Recognition with Strings. Pattern Classification*. 2nd Ed. Wiley Inter Science. Ch. 8, p. 413-420.
- Han, L.D., M.K. Jeong, and F.M. Oliveira-Neto (2008). *Phase A – Final Report on License Plate Recognition and Plate Matching*, project report submitted to National Transportation Research Center, Inc., Knoxville, TN.
- Han, L.D., F.J. Wegmann, and A. Chatterjee (1997). *Using License Plate Recognition Technology for Transportation Study Data Collection*, project report submitted to Tennessee Department of Transportation, Nashville, TN.
- Levenshtein, V.I. (1966). Binary Codes Capable of Correcting Deletions, Insertions, and Reversals, *Soviet Physics Doklady*, Vol. 10, No. 8, pp. 707-710.
- Marzal, A., Vidal, E., 1993. Computation of Normalized Edit Distance and Applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15 (9), 926-932.
- Mei, J. (2004). Markov Edit Distance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, No. 3, 2004, pp. 311-320.
- Nakanishi, Y. J., Western, J., 2005. Ensuring the Security of Transportation Facilities: Evaluation of Advanced Vehicle Identification Technologies. Transportation Research Record 1938, 9-16.
- Nelson, L. J., 2000a. Identification/Recognition: A Change of Order. ITS World, 32-33.
- Nelson, L. J., 2000b. Snap Decisions. Traffic Technology International, 50-52.
- Nelson, L. J., 2003. An Avid Reader. Traffic Technology International, 72-74.
- Ocuda, T., Tanaka, E., Kasai, T., 1976. A method for Correction of Garbled Words based on the Levenstein Metric. IEEE Transactions on Computers, C-25(2), 172-177.
- Oliveira-Neto, F.M., L.D. Han, and M.K. Jeong (2009). Tracking Large Trucks in Real Time with License Plate Recognition and Text-Mining Techniques, *Transportation Research Records: Journal of the Transportation Research Board*, No. 2121, Transportation Research Board of National Academies, Washington, D.C., pp. 121-127.
- Oommen, B. J., 1986. Constrained String Editing. Information Sciences, 40 (3), 267-284.
- Tang, T, M. Roberts, and C. Ho (2003). *Sensitivity Analysis of MOBILE6 Motor Vehicle Emission Factor Model*, Federal Highway Administration, FHWA Resource Center, Atlanta, GA.
- Rossetti, M. D., Baker., J., 2001. Applications and Evaluation of Automated License Plate Reading Systems. In 11th ITS America Meeting. CD-ROM. Conference proceedings. Miami Beach, Fla.
- Seni, G., Kripasundar, V., Srihari, R., 1996. Generalizing Edit Distance to Incorporate Domain Information: Handwritten Text Recognition as a Case Study. Pattern Recognition, 29(3), 405-414.
- Wagner, R. A. and M. J. Fischer (1974). The String-To-String Correction Problem, *Journal of Association Computer Machinery,* Vol. 21, No. 1, pp. 168-173.
- Watling, D. P., 1994. Maximum Likelihood Estimation of an Origin-Destination Matrix from a Partial Registration Plate Survey. Transportation Research Part B, 28B(4), 289-214.

- Watling, D. P., Maher, M. J., 1988. A Graphical Procedure for Analyzing Partial Registration Plate Data. Traffic Engineering & Control 29, 515-519.
- Watling, D. P., Maher, M. J., 1992. A Statistical Procedure for Estimating a Mean Origin-Destination Matrix from Partial Registration Plate Survey. Transportation Research Part B, 26B(3), 171-193.
- Wei J., 2004. Markov Edit Distance. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6(3), 311-320.
- Wiggins, A., 2006. ANPR Technology and Applications in ITS. 2006. Research into practice: 22nd ARRB Conference Proceedings. CD-ROM. Australia Road Research Board, ARRB, Canberra, Australia, October of 2006.