University of Tennessee, Knoxville

TRACE: Tennessee Research and Creative Exchange

Doctoral Dissertations

Graduate School

8-2010

Matching Vehicle License Plate Numbers Using License Plate Recognition and Text Mining Techniques

Francisco Moraes Oliveira Neto
University of Tennessee - Knoxville, foliveri@utk.edu

Follow this and additional works at: https://trace.tennessee.edu/utk_graddiss

Part of the Civil Engineering Commons

Recommended Citation

Oliveira Neto, Francisco Moraes, "Matching Vehicle License Plate Numbers Using License Plate Recognition and Text Mining Techniques. " PhD diss., University of Tennessee, 2010.
https://trace.tennessee.edu/utk_graddiss/836

This Dissertation is brought to you for free and open access by the Graduate School at TRACE: Tennessee Research and Creative Exchange. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of TRACE: Tennessee Research and Creative Exchange. For more information, please contact trace@utk.edu.

To the Graduate Council:

I am submitting herewith a dissertation written by Francisco Moraes Oliveira Neto entitled "Matching Vehicle License Plate Numbers Using License Plate Recognition and Text Mining Techniques." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Civil Engineering.

Lee D. Han, Major Professor

We have read this dissertation and recommend its acceptance:

Stephen H. Richards, Christopher R. Cherry, Xueping Li

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Francisco Moraes de Oliveira-Neto entitled "Matching Vehicle License Plate Numbers Using License Plate Recognition and Text Mining Techniques." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Civil Engineering.

Lee D. Han, Major Professor

We have read this dissertation
and recommend its acceptance:

Stephen H. Richards

Christopher R. Cherry

Xueping Li

Accepted for the Council:

Carolyn R. Hodges
Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Matching Vehicle License Plate Numbers Using License Plate Recognition and Text Mining Techniques

A Dissertation Presented for
the Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Francisco Moraes de Oliveira-Neto
August 2010

ACKNOWLEDGEMENTS

Along this journey I was glad to meet many people who helped me either professionally or personally. In special, I would like to thank my advisor Lee Han for the guidance and support. He is a creative person who knows very well how to explore the potential and skills of his students. I also want to thank Dr. MK Jeong for co-advising my research, and Dr. Stephen Richards, Dr. Christopher Cherry and Dr. Xueping Li for serving on my committee.

While some persons helped me with ideas and insight others spent their time to provide data for my work. In special, I am thankful to Stephanie Hargrove for coordinating all field experiments and data management, providing valuable data which enriched my work and guided me to new directions. I am also grateful to Elliot Moore, Scott and Michael Han for collecting data for my research.

I would like thank my good friend Manoel Mendonça who even being younger than me always gives me good advices for life as if I were his younger brother. I also want to thank Felipe Loureiro for the professional guidance and personal support since I started my career in transportation under his supervision. He is one of the most inspiring and dedicated persons I have ever met, who loves his work and spreads his inspiration to other students and colleagues. I also want to thank everyone who made my life happier in Knoxville. In special, I want to thank Lina Schlachter, Bruno Wichmann and Marna Wichmann. I also want to thank my friends Daniela, Mara and Luciana from the coffeehouse, Golden Roast, where I used to stay for hours working; and my Brazilian roommates Faete Jacques and Marcos de Lucca.

I want to acknowledge the financial support from the National Transportation Research Center, Inc. (NTRCI) and from the U.S. Dept. of Transportation's Eisenhower Transportation Graduate Scholarship Program.

Finally, I want to thank my beloved family for turning me into the person that I am today. My parents Antônio França Moraes and Maria do Céu Guerra Moraes for the love and all effort they made to educate and raise me; and My brothers João Batista de Oliveira Neto and Carlos Eduardo Guerra Moraes for the love, faith and incentive they have been transferring to me to fulfill my life.

ABSTRACT

License plate recognition (LPR) technology has been widely applied in many different transportation applications such as enforcement, vehicle monitoring and access control. In most applications involving enforcement (e.g. cashless toll collection, congestion charging) and access control (e.g. car parking) a plate is recognized at one location (or checkpoint) and compared against a list of authorized vehicles. In this research I dealt with applications where a vehicle is detected at two locations and there is no list of reference for vehicle identification.

There seems to be very little effort in the past to exploit all information generated by LPR systems. In nowadays, LPR machines have the ability to recognize most characters on the vehicle plates even under harsh practical conditions. Therefore, although the equipment is not perfect in terms of plate reading, it is still possible to judge with certain confidence if a pair of imperfect readings, in the form of sequenced characters (strings), most likely belongs to the same vehicle. The challenge here is to design a matching procedure in order to decide whether or not they originated from the same vehicle.

In view of the aforementioned problem, this research intended to design and assess a matching procedure that takes advantage of a similarity measure called edit distance (*ED*) between two strings. The *ED* measures the minimum editing cost to convert a string to another. The study first attempted to assess a simple case of a dual LPR setup using the traditional *ED* formulation with 0 or 1 cost assignments (i.e. 0 if a pair-wise character is the same, and 1 otherwise). For this dual setup, this research has further proposed a symbol-based weight function using a probabilistic approach having as input parameters the conditional probability matrix of character association. As a result, this new formulation outperformed the original *ED* formulation. Lastly, the research sought to incorporate the passage time information into the procedure. With this, the performance of the matching procedure improved considerably resulting in a high positive matching rate and much lower (less than 2%) false matching rate.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 RESEARCH PROBLEM

Among intelligent transportation systems, automated vehicle identification (AVI) is a powerful tool for electronic toll and traffic management, commercial vehicle operations, motor vehicle law enforcement, origin-destination survey, and access control, among other applications. All these applications require a unique identification of a vehicle in a checkpoint, and some of them also need a vehicle to be tracked in several points (e.g. in speed enforcement). To be identifiable a vehicle should be equipped with a device that emits the vehicle (plate number and VIN – Vehicle Identification Number) and owner information to a reader in a checkpoint. Although this is the most accurate method of identification it raises some concerns about privacy and depending upon the application (e.g. electronic toll collection) is unreasonably to believe that all vehicle targets will possess such devices.

There is another way of indentifying vehicles, which consists in automatically reading the characters of their plate numbers using License Plate Recognition (LPR) systems. These systems were developed with the main objective of interpreting the alphanumeric characters on vehicle plates without human intervention. Thus they rely on three main components: an imaging acquisition processor, a character recognition engine and a computer to store the data. Basically, the LPR operation consists in capturing, recognizing and storing information such as images, plate numbers, passage times and location on a database for online verification or posterior analysis.

LPR systems have been applied in different transportation applications since its launch into the commercial world in the early 1980s (Nelson, 2003). Such applications involve enforcement, vehicle monitoring and access control. As for enforcement, LPR can operate as a background system for electronic toll collection to indentify violators, or can be used to enforce the speed limit over a road segment. In access control, vehicle plates are recognized and verified against a database to allow or deny access into a facility. In traffic monitoring, vehicles are detected in multiple points and data such as travel time, origin destination (OD) demand, and route choice can be estimated for different purposes.

Although LPR systems have the advantage of not requiring new devices to be installed inside the vehicles, there are still some concerns with respect to their accuracy. Some developers claim that the character recognition engine is able to achieve almost 100% of accuracy. However, such claims can hide important assumptions, since the equipment is not usually tested under all possible conditions found in practical applications (Nelson, 2000). Thus the performance of the system should be evaluated by field testing under varying conditions of illumination, vehicle speed, camera offset angle, precipitation and so on. Usually limited resources prevent developers to perform such rigorous tests, but they should provide the conditions under which systems achieve the stated accuracy.

In reality the potentialities of the LPR equipment are not quite realizable. Depending on the type of internal technology, the installation, the on-site calibration, the weather, the lighting, the plate configuration, and a host of other conditions (Nakanishi and Western, 2005), LPR rarely recognizes more than 80% of the plates and often does worse than 60%. Fortunately, all is not lost; even when LPR fails to read a plate, meaning that not every single character is recognized correctly, the system usually returns very valuable and mostly correct individual character information. By comparing the imperfectly read plate against another such plate, or against a given database, one may still be able to render reasonable judgment in terms of whether there is actually a match. For instance, if two strings (sequence of characters) differ from each other by only one character, they may well have originated from the same plate.

It is simple for a system to recognize a plate that it has seen before, or when a reference database containing all possible plates is available. A database subscriber reduces the universe of plates and make statistically easier to recognize the true plate. The confidence of the compensation method is inversely proportional to the database size. Sometime a simple syntax checker can easily handle failures. Of course, the absence of meaningful plate confirmation database points to a certain need for human involvement.

The problem is not trivial for applications involving vehicle tracking at two or at multiple checkpoints (e.g. OD estimation and travel time studies). Considering that the universe of possible vehicle plates is immense or not available, the system should be capable of matching observations of imperfect readings without any reference. For example, under the hypothetical assumption that every character on a plate has equally and independently likelihood of being misinterpreted by the LPR machine, a single reading could have been originated from several ground truth plates, making the task of matching plates under this universe of uncertainty and possibilities quite remarkable.

This research focused on the aforementioned problem which deals with matching plate readings captured by a dual setup of LPR equipment. It is proposed a matching procedure that compensates for the recurrently interpretation errors made by the equipment under practical conditions. The matching method is based on a technique of text mining named edit distance (*ED*), which aims to measure how close two strings (sequence of characters) are from each other using weight functions (which can be subjective score values or estimated from statistical data) designed for comparison between pair of characters.

1.2 RESEARCH OBJECTIVE AND PREMISES

The main goal of this research was to assess the problem of matching outcomes (readings) from a dual setup of LPR equipment. In this study I explored the concept of text mining techniques and weighted matching algorithms to the problem of tracking vehicles whose plate numbers have been recorded by a two-point setup of LPR units.

The research is based upon the following premises:

- LPR machines can never achieve perfect reading rate. Due to the varying conditions in which the LPR system should operate it is very common to have character misinterpretation. Even under ideal conditions the equipment is not flawless;

- Most errors made by a LPR machine, even under the harsh conditions of operation, are in some extent recurrent and therefore predictable. Such recurrence can be captured into a probabilistic framework that can be used for matching purpose;
- Different technologies operate distinctly and due to the variety of conditions under which the equipment operate a vehicle plate may generate different outcomes at different checkpoints;
- The character recognition algorithms recognize the individual characters on a plate independently, meaning that the position or whether the character is numeric or alphabetic does not affect the recognition of a surrounding character.

1.3 HYPOTHESIS

The hypothesis of this research is that: *using a history that shows the recurrent probability LPR errors (such as O and 0 or Q, B and 8, 1 and I, K and X, W and V, and so on), as well as using additional information (passage time stamps), it is possible to infer with certain degree of confidence the likelihood of any two imperfect readings being originated from the same vehicle. Such odds can be used to decide towards genuine and false matches.*

1.4 RESEARCH METHOD AND SCOPE

Since I was dealing with sequence of character outcomes (strings) provided by the LPR machines, it seemed reasonable to use a technique of string alignment to compare pair of strings. Thus, at first, the study identified a technique to compare sequence of characters in order to establish how close two of strings are from each other. This technique was then used in the matching procedure to identify vehicles traveling through a two-point LPR setup. Secondly, armed with the assumption that LPR machines can recognize most characters on the vehicle plates, even with low reading accuracy per plate number, a refinement of the matching procedure was proposed. Such improvement consisted in allowing new weights or cost functions (calculated using the LPR character reading probabilities) to compare character by character of a dual string alignment. Finally, the passage time information (passage time stamps) was included in the matching procedure as an additional constraint to restrict the number of candidates considered for matching. Thus, to accomplish all established goals, this research was summarized on the following activities.

1. Conduct a literature review on methods to measure similarity between strings and identify one that is suitable for LPR application. As will be seen in Chapter 3, the widely used method of comparing pair of strings is called edit distance;
2. Propose a weight function that incorporates the LPR odds of misreading characters and that can be used in different formulations of the selected similarity measure;
3. Assess different strategies to match plates from a dual LPR setup and establish a methodology to determine the most suitable method;

4. Investigate how the passage time information can help to improve the performance of the matching procedures. Two methods were tested:
    a. Fixed Time Window Constraint (FTWC) whose limits are defined by the lower and upper bounds of the vehicle journey times;
    b. Varying Time Window Constraint (VTWC) whose limits change with the travel time variation and the edit distance magnitude.
5. Perform statistical and simulation analyses to determine the required sample size to estimate the character association probabilities to be used in the weight functions.

It is worth noting that the matching procedures, not incorporating passage time information, can be applied to any dual LPR setup, either in freeways or urban areas. However, all matching procedures using passage time information, as additional constraint, were investigated only for dual LPR setup in freeways (such as interstates), with no major traffic disturbance that may cause too much variation on the vehicle travel times. Although they can still be applied to other conditions not assessed here, no data analysis is presented to support such applications. Different traffic conditions (e.g. with high travel time variation) other than the ones presented here will be object of further studies.

The matching procedure presented here can be also extended to sequential, and multiple LPR setups (with multiple entry and exit checkpoints); but all these issues are out of the scope of this research and will be subject for further studies.

1.5 DISSERTATION ORGANIZATION

Besides this introductory chapter, this dissertation work is composed of more six chapters. Chapter 2 presents a brief discussion about LPR operation, application and accuracy. The next four chapters were written in paper format for further publication. Chapter 3 presents a first attempt to apply a similarity measure to the problem of matching vehicle plates recognized by LPR systems. In Chapter 4, a refinement (new weight functions) of the method proposed in Chapter 3 is presented. Such enhancement consists in using the odds of LPR units in misreading characters to better discriminate positive matches from negative matches. Chapter 5 presents how the passage time information was included into the matching procedure to increase the likelihood of finding a positive match. In Chapter 6, a study on the sample size necessary to estimate the odds used in the weight function proposed in Chapter 4 is presented. Finally, Chapter 7 contains the conclusions and recommendations of this dissertation work.

CHAPTER 2

LICENSE PLATE RECOGNITION: OPERATION, APPLICATIONS AND ACCURACY

2.1 LPR OPERATION

License-plate recognition technology was originally developed to read license-plate characters on moving vehicles. The process of capturing a plate image and recognizing the characters involves vehicle detection, image processing, and optical character recognition, which have all been documented in detail in past literature (Rossetti and Baker, 2001; Wiggins, 2006). As Han et al. (1997) have pointed out all LPR systems take advantage of the basic pattern-recognition technology to identify the alphanumerical characters on license plates.

LPR units usually consist of the following main components: an illumination source, a camera, a vehicle sensing device, an image processor, and a computer to store images and the reading plates (Rossetti and Baker, 2001). An infra-red based illuminator source is normally required when operating in low light condition or at night time, as well as to overpower sunlight and eliminate shadows. A digital camera with fast shutter speed must be triggered by an internal or external vehicle sensing device. The image processor locates the plate from an image view of the vehicle and uses the embedded pattern recognition algorithm to indentify the plate number. The following is a summary of how a LPR system operates:

1. As a vehicle enters the system's field of view it initiates a sequence process. At first, a vehicle presence is detected by the sensing device (which could be an external loop detector or an internal trigger, wherein the signal from the video subsystem alerts the processor that a moving object may be present). After that, the video camera, with synchronized shutter and illuminator, captures an image or a series of images of the passing vehicle.
2. Once the image is digitized, the next step is to determine if and where the license plate is located within the captured image. The image processor must search for the plate number among a bunch of other similar objects such as sticker bumper, phone numbers, and other extraneous items. Thus, several tests are usually necessary to isolate and confirm that a plate is present and submit it for character recognition.
3. In the next step, the LPR pattern recognition algorithm segments and recognizes each character on the plate. The characters font, as well as the plate syntax, can be subsequently used to refine the determination. Finally, the recognized characters and images can be retained locally for examination against a database or transferred to remote server for further analysis.

The pattern recognition algorithm is the most important component of the image processor subsystem of a LPR system. There are three types of techniques commonly employed by LPR image processors: template matching or correlation method, structural analysis, and neural networks (Nelson, 1997; Rossetti and Baker, 2001; Wiggins, 2006). These methods are described as follows.

The template matching is the method used by the Optical Character Recognition (OCR) methods designed for operation with scanned documents. The OCR method takes each character of the plate and attempts to match it to a set of predefined standards. Since any deviation from the standard can cause questionable results, this method is not very tolerant to misaligned, obscured, dirty and damaged characters.

Structural analysis uses a decision tree to assess the geometric features of the character's contour. This method is a little more tolerable to poor quality of the shape of the characters.

Neural networks are methods based on training and learning process rather than programming. While learning to recognize a recurring pattern, the network constructs a statistical model that adapt to unique features of the characters. It seems that this method is the most tolerable to noise caused by changes under diverse operational conditions, however the process of training a neural network can be very time consuming and is usually required any time a new plate is released.

2.2 LPR APPLICATIONS

The first commercial available LPR system was implemented 25 years ago. Since then this type of system has been used for different applications which can be classified into three categories: access control, traffic studies and enforcement (Nakanishi and Western, 2005; Wiggins, 2006).

2.2.1 Access Control

Access control covers all examples of application such that a LPR system is used to read the license plate numbers of vehicles entering or leaving a checkpoint and automatically compared them against a list of authorized or registered vehicles. If there is a match an automated gate or other physical barrier will open to permit the access into/out the facility. Car park is a good example of application of LPR systems, especially those where drivers are required to pay for the permanence time. In this case, as the vehicle enters the facility their license plate numbers are recorded and associated to the entry ticket.

2.2.2 Traffic Studies and Planning

The main applications of LPR in traffic studies and planning involve traffic demand estimation (Origin-Destination (OD) demand estimation) and travel time studies. With regard to OD estimation survey, the traditional plate survey method consists in assigning at least one person to monitor each checkpoint (entries and exits of the studied area) and record part of the plate number of the passing vehicles. Depending on the studied area scale, it has been reported that this conventional method is a costly exercise, and even the best staffs can only record license plate numbers with an accuracy of about 70%. Furthermore, depending on the environmental conditions lower accuracy are normally reported (Wiggins, 2006). Replacing the observers by LPR machines can save time and increases the accuracy of the survey.

Another common application is travel time studies. LPR systems can be used to record the location and passage time of a vehicle in two different points of a roadway;

and from this data the average speed can be determined. The information of vehicle speeds can be used either for traffic flow studies or can be transmitted back to drivers, under an information system architecture, to inform journey times to complete certain desired trips. Regarding the use of LPR integrated with an information system, Buisson (2006) has developed a methodology to assess the impact of the number of devices on the precision of the displayed travel times in a congested road.

For transportation planning, LPR can be used in the determination of vehicle route choice in an urban network. In this application vehicle identification devices are distributed on the links of the network in order to reconstruct the main paths chosen by the drivers that use the urban network for different purposes. An OD demand matrix is also obtained from such studies. To this end, Casttilo et al. (2008) proposed two optimization programming formulations to select the device's locations and to reconstruct the trip patterns on a given network.

2.2.3 Enforcement

LPR system can help with the enforcement of bus lanes, the prevention of fraud in cashless toll collection systems, enforcement in vehicle charging systems and in speed enforcement over distance. In all these applications, except for speed enforcement, a vehicle is tracked in two points or detected in one point and its plate number is compared against a white list of vehicles that are allowed to use the facility. Normally in toll and in vehicle charging systems LPR operates as complementary system to catch those vehicles that are not registered to the system, i.e. not equipped with radio frequency identification (RFID) tag or that not possess a permission to travel on the road (Wiggins, 2006).

Speed enforcement over distance is an application, just as travel time studies, where two devices are located in two different checkpoints, upstream and downstream over a road portion, to measure the average speeds (Wiggins, 2006). At any time the average speed exceeds the road speed limit the corresponding data is stored and verified, and after confirmation the driver receives a warning or fine. The great advantage of LPR over the normally used speed cameras is that LPR system is non-invasive vehicle detection, since the drivers may not have any knowledge where the speed traps are located, what avoids disturbance on the traffic behavior.

The largest scale use of LPR technology is in London. Aiming to reduce congestion in the central part of the city, London became one the first municipalities worldwide to use LPR technology on large scale when implementing its congestion charge system (Eberline, 2008). The system was implemented in February, 2003. In 2007 it covered an area of 40 square kilometers (about 15.5 square miles). When the congestion charge began, about 700 cameras were situated in and around the charging zone. Photographs of vehicle plates are taken when they enter the charging zone and sent to a central computer that indentifies the plates using a LPR system. At the end of the day all recognized plates are matched with the payments. Tickets are issued to those drivers that do not pay the charge. According to the report carried out by Eberline (2008), the system has been a great success in many aspects. Mainly, it has reduced congestion and provoked a shift of demand from road usage to public modes of transportation.

2.3 LPR ACCURACY

The accuracy of the pattern recognition algorithms used in the image processor is important concern when evaluating LPR systems. The need for high accuracy also implies in higher prices of the systems since better algorithms and cameras are necessary (Rossetti and Baker, 2001). Fortunately, the required accuracy depends on the application and less expensive equipments can be acquired. For example, enforcement applications such as speed enforcement may require a high degree of accuracy (all characters on a plate should be identified) to avoid notify innocent users, while applications of traffic monitoring such as OD estimation may only require that an image obtained at an entry point be matched to an image obtained at an exit point.

One method of quantifying the accuracy of these systems is to measure the percentage of license plates correctly identified by the machine that could be verified by a person (Nelson, 1999; Rossetti and Baker, 2001). However, one should have in mind that this method does not assess completely the real capabilities of the LPR systems since it throws away part of information available, which could be used for matching purposes for example. Hence, if a LPR machine misreads only one or two characters on a plate, the retrieved information can still be useful depending on the application. In Chapter 3 the capabilities of a LPR system is also measured in terms character reading rate. As will be seen, this measure better reflects the LPR potentialities as far as this study is concerned.

According to the literature, the accuracy of the equipment is affected by three factors: the quality of the images captured in the field, the internal settings of the equipment used and the light conditions under which the plate images are acquired

(Nakanishi and Western, 2005). The quality of the images are affected by the traffic speed, the weather, the installation, the on-site calibration, the plate condition, the variety of plate syntax, the plate mounting location, vehicle type and other conditions. Due to theses factors LPR rarely recognizes more than 80% of the plates and often does worse than 60%. Hence, the internal technology may not distinguish between some characters (e.g. O, D and Q). However, if these mistakes or interchanges can be predicted and isolated, it is possible to compensate for them in order to track vehicle in a dual LPR setup.

CHAPTER 3

MATCHING VEHICLE PLATES USING LPR AND EDIT DISTANCE

This chapter presents the first nuances of the proposed application of similarity measures for strings to the problem of matching imperfect readings from a dual LPR setup. The application described herein is for speed monitoring; however, the methodology can be applied to any application related to a two-point LPR survey or multiple entry-exit setups. This chapter also proposes a method to calculate the character-based accuracy of LPR machines. There have been efforts to study the problem of matching plates read by LPR units at multiple locations of a highway (Han et al., 1997; Bertini et al., 2005; Buisson, 2006); however, few of them formalized the methodology for matching imperfectly read plates or for efficiently exploiting the LPR data. To this end, this chapter reports the first attempt to employ a text-mining technique called Edit Distance (*ED*) to improve the matching efficiency of imperfectly read plates. In the following sections, this chapter presents the fundamentals of the Edit Distance technique and how it is applied to license plate matching. A case study and its results are also presented, followed by discussions and conclusions.

3.1 LICENSE PLATE RECOGNITION FOR SPEED MONITORING

Using LPR for speed monitoring is similar to the traditional license-plate survey technique that has been widely employed for decades. In essence, field observers are placed at key points to record part of the license plates (e.g. the last three of the six characters) on vehicles passing the locations. The list of plates, in the form of sequenced character strings, is then compared with lists from other locations in order to match or pair the strings together. When two identical strings are found on different lists, a match is declared, and it is assumed that the same vehicle has traversed both locations over time. Information such as route choice, origin/destination, or average speed can subsequently be derived from the matches.

The concept of an LPR-based speed enforcement system is alluring: with simple replacement of the field observers from the old plate-survey technique, real-time vehicle monitoring seems easily attainable. The reality is not so simple, and, hence, the potentialities of LPR are not quickly realizable. Depending on the type of internal technology, the installation, the on-site calibration, the weather, the lighting, the plate configuration, and a host of other conditions (Nakanishi and Western, 2005), LPR rarely recognizes more than 80% of the plates and often does worse than 60%. Fortunately, all is not lost; even when LPR fails to read a plate, meaning that not every single character is recognized correctly, the system usually returns very valuable and mostly correct individual character information. By comparing the imperfectly read plate against another such plate, one may still be able to render reasonable judgment in terms of whether the two plates are a match. For instance, if two strings (sequence of characters) differ from each other by only one character, they may well have originated from the same plate. Therefore, a measure of similarity between two strings can be established to indicate the likelihood of a match.

3.2 MEASURE OF SIMILIRATY BETWEEN TWO STRINGS

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, as “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 (Wei, 2004). The pioneer in this field is Vladimir Iosifovich 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, 1966). 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, I returned 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 3-1. 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 move corresponds to a substitution; a move to the right represents an insertion; and a vertical move represents a deletion. 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 identical, i.e., $x_{i_k} = y_{j_k}$, or 1 if they are dissimilar. If I "walk" from the origin point (0, 0) to the end point (i_n, j_n) on the grid, each potential path is associated with an overall cost, *d*, defined as:

$$d(i_n, j_n) = \sum_{k=0}^{n} \gamma(i_k, j_k) \tag{3.1}$$

where,

n is the number of nodes of a path between $(i_0, j_0) = (0,0)$ and $(i_n, j_n) = (l_x, l_y)$; and

l_x and l_y are the lengths (number of characters) of x and y, respectively.

As an example, consider two paths (drawn by the solid and dashed lines) reaching the point (l_x, l_y) as shown in Figure 3-1. Computing the number of editing operations performed by these two paths will result in $d_{solid}(i_n, j_n) = 2$ and $d_{dashed}(i_n, j_n) = 6$.

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 (1974). A detailed description of this procedure can be found in the book Pattern Classification by Duda, Hart, and Stork (Duda et al., 2000). 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_n, j_n), or $d(x, y) = \min\{d(i_n, j_n)\}$.

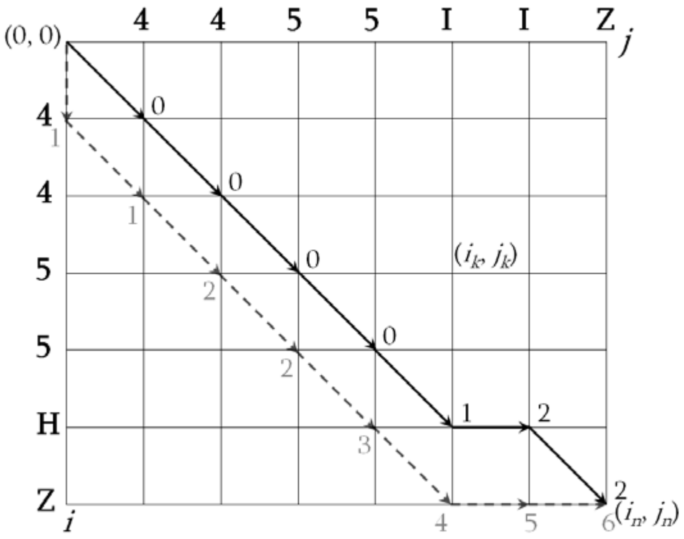


Figure 3-1 Example of Editing Paths on a Grid

In many 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 lexicon of reference words.

3.3 MATCHING PROCEDURE

In this study I 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. Thus, every pair of recognized strings is 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 0 or 1 cost values.

The matching procedure consists on a post-processing process 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. Then the set of assignments that minimizes the overall cost and such that all *ED* values are less than a given threshold τ is determined.

Mathematically, finding the least cost assignment requires solving the following assignment problem:

$$z = Minimize \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} z_{ij}$$

Subject to,

$$\sum_{i=1}^{m} z_{ij} \leq 1 \qquad (j = 1,2, \ldots, n),$$

$$\sum_{j=1}^{n} z_{ij} \leq 1 \qquad (i = 1,2, \ldots, m),$$

$$z_{ij} = 0 \text{ or } 1 \qquad (i = 1,2, \ldots, m; j = 1,2, \ldots, n)$$

where,

d_{ij} is the similarity measure between the i^{th} and j^{th} outcomes from stations *g* and *h* respectively;

m and *n* are the numbers of reading outcomes at station *g* and *h*, respectively;

z_{ij} is a indicator variable that equals to 1 whenever a match is declared.

The computational solution of the problem above requires the construction of a matrix whose entries are the *ED* costs obtained from the association between all pair of strings (Munkres, 1957, and Bazaraa et al. 2005). Thus, for a large database computing this matrix can be computationally expensive as it involves huge combination of data entries. Assuming that the similarity measure performs well in classifying the matches in true or false, an approximate solution to the assignment problem can be found by simply assigning for each outcome at station *h* that outcome at station *g* with least value of *ED*, and applying a checking procedure to prevent an outcome of being matched more than once. The final assignment is determined in order to meet the constraint that all *ED* values must be less than a threshold value.

The motivation of finding this set of matches without using the help of passage time information was to assess discriminative power of the similarity measure used. 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*) for a given survey period, if any similarity measure is capable of discriminating genuine from false matches in this worst case scenario, I may claim that it is a good similarity measure for LPR application.

In sum, my 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 (each element of this vector is the edit distance between the corresponding outcomes); second, the assignment with the least *ED*-value is selected as a potential match; third, a test is performed to verify if any of the outcomes in this current match has been already matched before, where in such case the match with higher *ED* cost is eliminated from the matching list; finally, a threshold on the *ED*-values is used to discriminate the resulting pair-wise matches between potential positive and false-positive matches. Figure 3-2 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 classified as genuine.

3.4 CASE STUDY AND RESULTS

In April 2006, Knoxville, Tennessee joined an increasing number of cities in reducing speed limits for large trucks (with gross weights over 10,000 pounds) on the interstate highways in its metropolitan area. In recent years, reducing large-truck speed limits in urban areas has become one of the preferred countermeasures for combating urban air-quality problems. The rationale for this is supported by a 2003 Federal Highway Administration (FHWA) study, which found that reducing large-truck speed by 16 km (or 10 mph – from 65 to 55 mph) can reduce emissions of NOx by 18% per large truck (Tang et al., 2003).

While reducing truck speed limits is a relatively simple act for metropolitan planning agencies, the subsequent enforcement effort often meets with more challenges. This is the case for Tennessee Highway Patrol (THP), which has jurisdiction over Interstates 40 and 75 (I-40 and I-75), both passing through the Knoxville metropolitan

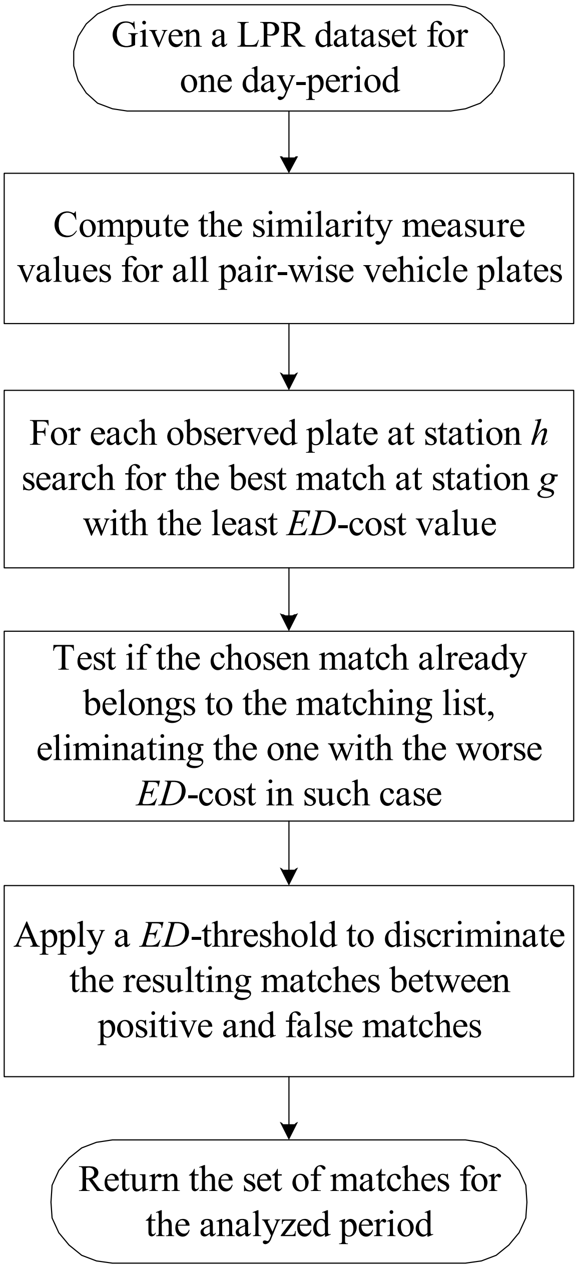

Given a LPR dataset for one day-period
Compute the similarity measure values for all pair-wise vehicle plates
For each observed plate at station *h* search for the best match at station *g* with the least *ED*-cost value
Test if the chosen match already belongs to the matching list, eliminating the one with the worse *ED*-cost in such case
Apply a *ED*-threshold to discriminate the resulting matches between positive and false matches
Return the set of matches for the analyzed period

Figure 3-2 Matching Procedure

area. After the new speed law was enacted, THP found itself facing 12 million large trucks annually, most exceeding the 88 km/hour (55 mph) speed limit, on this stretch of interstate. Furthermore, THP received no budget or manpower increases for the purpose of enforcing the new speed limit. The aforementioned FHWA report did not state this specifically, but it would be difficult to expect emission reductions if the new speed limit were not diligently enforced.

To this end, the University of Tennessee conducted a study using license-plate recognition (LPR) technology to automatically track large trucks as they traverse through the metropolitan area. Taking advantage of the existence of a weigh station on the west end of the area on I-40 where all trucks are required to stop for inspection, an LPR speed-enforcement system, with equipment strategically located along I-40, could issue warnings or citations as the perpetrating trucks stop on the weigh scale, with a THP officer stationed in the weigh house. This system would function in real time without the need for mailing out speeding tickets after the fact or pulling trucks over after dangerous high-speed pursuit, both alternatives resource- and labor-intensive.

3.4.1 Experiment Design

The LPR equipment used in this first experiment was manufactured by PIPS Technology. Two versions of the equipment were used to capture license plates of westbound trucks on I-40, one at Campbell Station Road (Station *g*) and the other downstream at the weigh station (Station *h*). 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 in 2007, 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.

3.4.2 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 *g* and 63% for Station *h*. Since the cameras were not permanently mounted (they were mounted on heavy tripods), the accuracies could potentially be higher.

In terms of character reading rate, the LPR units presented a better performance. Figure 3-3 illustrates the failure rate distribution, a chart of relative frequency of plates versus the number of characters misread per plate, for each LPR station. Calculating the average reading rate per plate (number of correct characters divide by the license plate length) for each machine, it was observed a rate of about 0.88 for Station *g* and 0.90 for Station *h*. Thus, in spite of the moderate plate accuracy, the equipment was able to read most characters of the license plates.

Out of the 787 plates that were manually identified as identical, 53% were read correctly at both stations, 20% were misread at both stations and the remaining 27% were misread at either one of the two stations. These results show that there is a propensity for a character being recognized correctly at the second station given that it has been read correctly at the first station. Had the LPR units worked totally independently (e.g. two different LPR units with distinct pattern recognition algorithms) I would expect a rate around 0.61 x 0.63 = 0.38 of plates recognized at both stations.

3.4.3 Truck Speed and Journey Time

The histogram of the sample speeds for the 787 trucks captured at both stations is shown in Figure 3-4a. Figure 3-4b shows the corresponding cumulative distribution of truck-speeds after the speed limit changed. As observed in Figure 3-4a, after the new speed limit went into effect, truck speed ranged from 40 mph to 75 mph, and as shown in Figure 3-4b most of the speed values (the 19th percentile was approximately 55 mph) were higher than the actual speed limit of 55 mph.

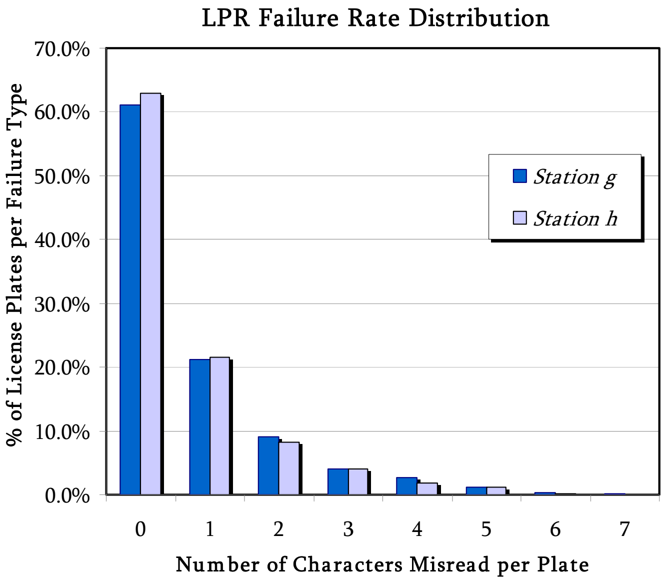
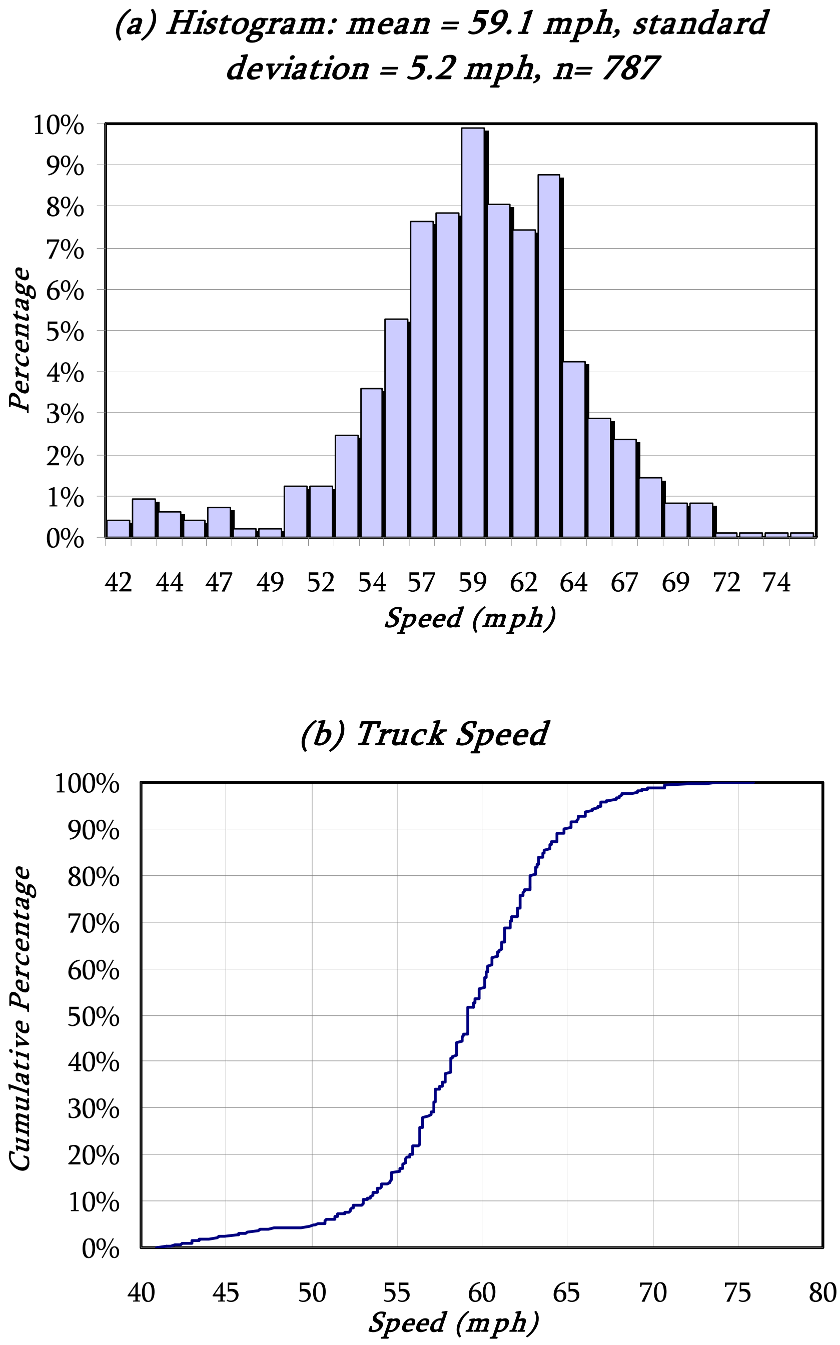


Figure 3-3 LPR Failure Rate Distribution in Recognizing Plate Characters



(a) Histogram: mean = 59.1 mph, standard deviation = 5.2 mph, n= 787

(b) Truck Speed

Figure 3-4 Histogram and Cumulative Distribution of the Sample Truck’s Speed

The truck's journey time variation along two survey days is presented in Figure 3-5, which shows two moving average profiles, one for each survey period. As can be seen, for the time period presented, there was not much traffic disturbance. Looking at the disaggregated data, the journey time variable presented low amplitudes of around 0.3 minutes (18 seconds).

3.4.4 ED Performance and Results

To assess the performance of the proposed matching method, 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.

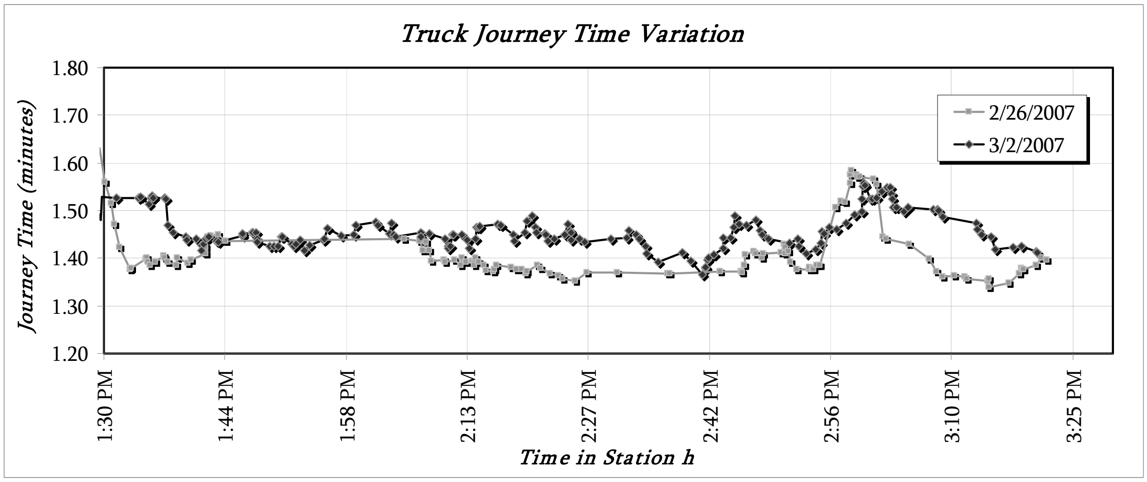


Figure 3-5 Truck's Journey Time along Two Survey Periods

Four different threshold values, 0 to 3, were used to constrain edit distance. Table 3-1 shows the results obtained when the top-rank candidates with least *ED-values* were 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 passage time, false-positive matches are very likely to occur for threshold values of 2 and 3.

3.5 RESULTS DISCUSSION

The algorithm presented herein is not expected to achieve perfection with 100% plate matching rate and zero false matches. Nevertheless, improvement can still be accomplished through further research on better plate similarity measures (Chapter 4), fixed and dynamic travel time constraints (Chapter 5), and improved configuration of LPR hardware.

Table 3-1 Performance of *ED*

Threshold	Number of Matches	Number of Positive Matches	Number of False-positive Matches	Average Number of Candidates	Percentage of Vehicles Detected	Percentage of False Matches
0	497	497	0	1.02	60%	0%
1	692	667	25	1.08	81%	4%
2	921	737	184	1.59	89%	20%
3	1309	754	555	5.74	91%	42%

Regarding to the similarity measure employed in this study, the main drawback of using the formulation of edit distance with 0 or 1 cost assignments, in the case of comparing distinct characters, is that it does not account for the expected likelihood of LPR units in 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 odds of such incidences were not considered in this chapter. However, this information is available and can be obtained by constructing a matrix of error probabilities for each LPR unit used. Once the matrix is constructed, the challenge becomes in designing 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 “O” is misread as "0" with probability of 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, the two outcomes "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. Such issues will be object of Chapter 4.

In this chapter, the passage time information was not used to restrict the number of candidates for matching. Two methods are proposed in Chapter 5 with fixed and dynamic time window constraints; with the assumption in the second method that the travel time variable follows a symmetric density function, such as the Gaussian function. This way, the chronological method used herein for selecting candidate plates should be replaced by a probabilistic method, where a potential match is selected if it has both the least *ED* value and whose passage time difference is very likely to be valid travel time.

As for the equipment setup on the roadside, it is believed that a permanent rather than a mobile setup would lead to improved accuracy in plate reading. In the second phase of the study, LPR machines were mounted “permanently” on the structure of variable message panels. Higher percentage of correctly read plates should result, as is a higher plate-matching rate.

3.6 CHAPTER CONCLUSIONS

In this chapter, a technique of text mining (called Edit Distance) was introduced to handle the problem of matching plates recognized by a dual setup of LPR units. A field study using two LPR units was conducted on I-40/75 in the vicinity of Knoxville, Tennessee in 2007. The original idea of Edit Distance (with 0 or 1 cost values) was used to match license plates not correctly recognized by the LPR units. This represents the first attempt of using a text mining technique to the classical transportation problem of plate matching.

While the accuracy of the LPR units was less than perfect, most of the plate characters were recognized, even if incorrectly sometimes; the use of edit distance with no passage time constraint resulted in a increasing number of positive matches but with considerable percentage of false matches. Therefore, some improvements are necessary to make this technique feasible for plate matching.

In Chapter 4, results of further experiments will be presented with the objective of improving the reliability of LPR system in reducing false-positive matches. One of the research directions the author continued on was the use of a probability matrix based on the odds of one character being read, or misread, as another. This matrix was used to develop a symbol-based cost function, instead of the 0 or 1 cots employed in this chapter, for the edit distance calculation. As will be seen, this sophistication increased the percentage of positive matches and reduced the likelihood of false-positive matches.

Another direction, presented in Chapter 5, is the use of the passage time stamps to restrict the number of possible candidates in Station *g* to match a given outcome from Station *h*. To this end, two methods are employed, resulting in much better matching performances.

CHAPTER 4

PROPOSED WEIGHT FUNCTION

In Chapter 3 the original formulation of Edit Distance (*ED*) has been applied to compare two sets of strings generated from two LPR machines, which were set up (about 1.4 miles apart) to recognize plate numbers of trucks driving westbound on the junction of Interstate 40 and 75 in Knoxville, Tennessee. The results show that the proposed procedure could identify a high percentage of vehicles travelling through the two stations but with high likelihood of finding false-positive matches. In this chapter, a new symbol-based weight function is proposed to estimate the probability of having a genuine match for a certain sequence of editing operations when comparing pair of strings read by a dual LPR setup. Therefore with this refinement, an unlikely alignment of pair-wise strings will be more penalized than a likely one in the calculation of the weight function. This chapter is organized into 4 sections. Section 4.1 presents extensions of the *ED* calculation and some formulations for the weight (or cost) function. In Section 4.2, the proposed weight function is presented. In Section 4.3, the performance of the proposed procedure is compared with some popular approaches using a real-life case study. Finally, discussion and conclusions are presented in Section 4.4.

4.1 WEIGHTED EDIT DISTANCE

As seen in Chapter 3, the edit distance *d(x, y)* between two strings *x* and *y*, can be calculated based on the following recurrent equation, as proposed by Wagner and Fischer (1974):

$$\begin{aligned} d(i,j) = \min\{ & d(i-1,j-1)+\gamma(x_i \rightarrow y_j), \\ & d(i-1,j)+\gamma(x_i \rightarrow \varepsilon), \\ & d(i,j-1)+\gamma(\varepsilon \rightarrow y_j)\} \end{aligned} \tag{4.1}$$

Where $d(i, j)$ is the edit distance between x[1..i] and y[1..j], and $d(0,0) = 0$. The γs represent the cost functions. For example, the $\gamma(x_i \rightarrow y_j)$ is the cost for the change (substitution) from x_i to y_j. The $\gamma(x_i \rightarrow \varepsilon)$, where ε represents the empty character, is the cost incurred by a deletion of x_i. The $\gamma(\varepsilon \rightarrow y_j)$ is the cost incurred by an insertion of y_j. Thus, the edit distance *d(x, y)* would be given by $d(l_x , l_y)$, where the notation l_x and l_y correspond to the lengths of the *x* and *y* strings, respectively.

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 $\gamma(x_i \rightarrow y_j) = 0$ if $x_i = y_j$, otherwise $\gamma(x_i \rightarrow y_j) = 1$ (x_i and y_j cannot be ε 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” → “V” could be smaller than the edit substitution “Q” → “V”. The error rates can be reduced by adjusting the values of the weight for each fundamental edit operation in accordance with the associated character 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 last major advance in the *ED* calculation was made by Wei (2004) who proposed 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 $\gamma(p_1 \rightarrow p_2)$ such that 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. Therefore, *CED* and *GED* represent special cases of the *MED*.

The weight 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 handwritten digit recognition. They transformed the probability matrix into the 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 x into y as a Gibbisian probability distribution function, which in turn is defined as $P(x \rightarrow y) = \exp(-U(x \rightarrow y) / T)/Z$, where T and Z are constant parameters to be calibrated, and $U(x \rightarrow y)$ is the energy involving in any of the sequence of edit operations to transform x into y. The most desirable configuration for transforming x into y would be the one that maximizes $P(x \rightarrow y)$ which is equivalent to minimize $U(x \rightarrow y)$.

4.2 PROPOSED WEIGHT SCHEME FOR LPR APPLICATION

Similarly to Chapter 3, I still deal with the problem of matching two plate datasets from a dual LPR setup, with the two locations named Stations g and h, located upstream and downstream, respectively.

In order to improve the matching performance, the *ED* method and the cost (or weight) functions γs 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 LPR application.

All LPR misinterpretations can be translated into a matrix of error probabilities where each cells is given by the likelihood of certain pair-wise character symbol occurrence (e.g. “1” – “I”, “D” – “O”, “B” – “8”). Such information can be obtained by constructing a matrix of reading probabilities for each LPR unit. Once the matrix is constructed, the next task is to associate the two matrices of character misinterpretations into a designed weight function (or cost function) to be used in the edit distance calculation. The basic idea is that the higher the probability of a character association occurrence (likelihood of a pair-wise character come from the same truth character), the smaller the weight to compare the two characters. Therefore, in Equation 4.1, the 0 or 1 cost values should be replaced by appropriate weight values for each editing operation involved.

In designing the weight function, however, one should have in mind that the LPR application is different from common *ED* applications in the sense that there is no reference or list of ground truth 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 error probability matrices of each LPR machine.

The formulation of the weight function is based on the assumption, stated in Section 1.2, that the edit operations to convert a string x into a string y are independent of each other, i.e., there is no dependence relationship between neighboring characters of the patterns x and y. This means that the expected value for each individual character outcome observed from a LPR machine is not affected by the position of the characters or by the other surrounding characters.

It is also assumed, as stated in Section 1.2, that matrices containing the likelihoods of character interpretation by each LPR unit are available or can be estimated from a dataset containing both readings and ground truth values of the license plate numbers. I named such matrices as truth matrices, as defined in Subsection 4.2.2.

4.2.1 Weight Function

Let $x = x_1x_2...x_i...x_{lx}$ and $y = y_1y_2...y_j...y_{ly}$ 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 4-1, with the edit operations represented as the following moves on the grid: along the diagonal for substitutions, to the right for insertions, and vertical for deletions. There are 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

$$p(x \to y) = \prod_{k=0}^{n} p(i_k, j_k) \tag{4.2}$$

where, n is the number of nodes of a path between (i_0, j_0) = (0,0) and $(i_n, j_n) = (l_x, l_y)$. I defined the $p(i_k, j_k)$ as the probability of the corresponding edit 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 $(i_k\text{-}1, j_k\text{-}1) \rightarrow (i_k, j_k)$, $(i_k\text{-}1, j_k) \rightarrow (i_k, j_k)$, and $(i_k, j_k\text{-}1) \rightarrow (i_k, j_k)$ represent substitution, deletion and insertion, respectively.

If I make the negative logarithm in both sides of Equation 4.2 and minimize the result, I will obtain the following expression

$$d(x, y) = \min\left\{\sum_{k=0}^{n} \log\left(\frac{1}{p(i_k, j_k)}\right)\right\} \tag{4.3}$$

Indeed, to find the most likely alignment or sequence of edit operations, Equation 4.2 should be maximized, which implies to minimize its negative natural logarithm.

Finally, 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 character association probability $p(i_k, j_k)$ can be estimated from the collected dataset.

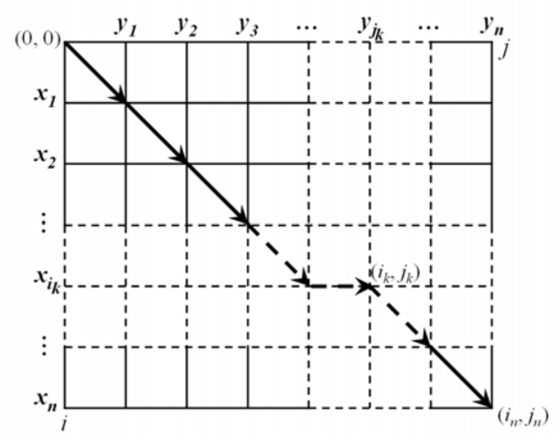


Figure 4-1 Path on a Grid for a General Comparison between Two Strings

4.2.2 Computation of the Conditional Probability *p*(*y* | *x*)

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 number). As will be seen, the method proposed to overcome this problem consists in applying conditional probability theory to associate the character interpretation probabilities given by two matrices, denoted by $\mathbf{C^g}$ and $\mathbf{C^h}$, of station *g* and *h*, respectively, and obtain estimates of $p(i_k, j_k)$ for any possible character association.

To estimate the key probability $p(i_k, j_k)$ for the weight function of Equation 4.3, I need to estimate the probability that the corresponding pair of character outcomes x_{i_k} and y_{j_k} at station *g* and *h* came from the exact same character. I proposed to calculate such character association likelihood in the basis of the 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* be now any character outcome at station *g* and *h*, respectively. Furthermore, let *z* be a ground truth character. Knowing that same brands of LPR units (with similar pattern recognition algorithm) work similarly, it is possible to estimate the conditional probability of observing the character outcome *y* at *h*, given a character outcome *x* at *g*, for a known character *z*, as the following expression:

$$p(y \mid x) = \frac{p(x,y)}{p(x)} = \frac{\sum_{z} p(x,y \mid z) p(z)}{\sum_{y,z} p(x,y \mid z) p(z)} \qquad (4.4)$$

Alternatively, Equation 4.4 can be also written as

$$p(y \mid x) = \sum_{z} p(y \mid z) p(z \mid x) \tag{4.5}$$

where,

$$p(z \mid x) = \frac{p(x \mid z) p(z)}{p(x)},$$

$$p(x) = \sum_{z} p(x \mid z) p(z)$$

A simple way to calculate the conditional probabilities for any character association is to use matrix manipulation. To this end, notice that Equation 4.5 is composed by a summation of products with two factors $p(y \mid z)$ and $p(\mathrm{z} \mid x)$ each, which can be viewed as entries of two probability matrices. Let us denote these two matrices as $\mathbf{R^g}$ containing the reverse probabilities $p(\mathrm{z} \mid x)$ and as $\mathbf{C^h}$ whose entries are the character interpretation probabilities denoted by $p(y \mid z)$.

Let us define the matrix denoted by $\mathbf{C^l}$, named character interpretation or truth matrix, as a matrix whose cells C_{ij}^{l} represents the conditional probability $p(r_i \mid z_j)$ that a given true character z_j was recognized as r_i by a LPR machine l. 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 $\mathbf{C^l}$ is a K by K square matrix where K is the total number of possible alpha-numeric (plus the empty one) characters for the variables z_j and r_i (in our application, K is 37 which means 36 alphanumeric characters plus the empty one). The empty character, denoted by the symbols ε or “ ” represents the missing character and makes possible deletion and insertion operations.

Let us now define the matrix denoted by $\mathbf{R^l}$, named reverse matrix, whose entries are the conditional probabilities, denoted by $p(z_j \mid r_i)$, that for a given recognized character r_i its ground truth character is z_j. As will be seen in the example, this matrix can be calculated for a LPR machine l using the corresponding truth matrix $\mathbf{C^l}$ and the information about the likelihoods of character occurrence on the plates.

In the matrices $\mathbf{R^g}$ and $\mathbf{C^h}$, the ground truth characters correspond to the columns of the matrices, while the read characters to the rows. Therefore, the computation of all possible character associations, or conditional probabilities $p(y \mid x)$, is given by the following matrix multiplication:

$$\mathbf{C} = \mathbf{R^g}.(\mathbf{C^h})^{\mathrm{T}} \tag{4.6}$$

where,

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

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

Finally, the probability $p(i_k, j_k)$ in Equation 4.3 should be approximated by $p(y_j \mid x_i)$ and can be obtained directly by simply searching for the cell in the matrix $\mathbf{C}$ in which the associated characters correspond to those involved in the editing operation at node (i_k, j_k) on the grid of Figure 4-1.

As an example, I demonstrated how to determine the association matrix **C** using Equation 4.4 for a hypothetical case. Let us assume that only two characters are possible to be observed, saying “A” and its complement “Ã” (not “A”). Let the corresponding truth matrices be as follows:

Matrix $\mathbf{C^g}$ of LPR Station *g*:

$$\mathbf{C^g} = \begin{bmatrix} p(x = \text{A} \mid z = \text{A}) & p(x = \text{A} \mid z = \tilde{\text{A}}) \\ p(x = \tilde{\text{A}} \mid z = \text{A}) & p(x = \tilde{\text{A}} \mid z = \tilde{\text{A}}) \end{bmatrix} = \begin{bmatrix} p_1 & 1 - \widetilde{p}_1 \\ 1 - p_1 & \widetilde{p}_1 \end{bmatrix}$$

Matrix $\mathbf{C^h}$ of LPR Station *h*:

$$\mathbf{C^h} = \begin{bmatrix} p(y = \text{A} \mid z = \text{A}) & p(y = \text{A} \mid z = \tilde{\text{A}}) \\ p(y = \tilde{\text{A}} \mid z = \text{A}) & p(y = \tilde{\text{A}} \mid z = \tilde{\text{A}}) \end{bmatrix} = \begin{bmatrix} p_2 & 1 - \widetilde{p}_2 \\ 1 - p_2 & \widetilde{p}_2 \end{bmatrix}$$

Let us also assume that the information about character occurrence on the plates is available and given by

$$\begin{aligned} p(z = \text{A}) &= p \\ p(z = \tilde{\text{A}}) &= \widetilde{p} = 1 - p \end{aligned}$$

Therefore whenever a character is recognized by both stations, the possible expected combinations of character association are presented in Table 4-1.

Table 4-1 contains the following information: the first column contains the possible ground truth characters; the second and third columns contain the possible character outcomes that must be observed at stations *g* and *h*; the fourth column shows the joint probabilities of observing the character x at g, y at h, for a given ground truth z; and the last column has the resulting conditional probabilities of pair-wise character association. An alternative presentation for Table 4-1 is presented in Figure 4-2, which shows a probability tree diagram.

In reality, since I deal with more than two possible characters it is more convenient to calculate the conditional probabilities using matrix manipulations. Thus, the expressions in the fourth column of Table 4-1 for each cell C_{ij} of the association matrix $\mathbf{C}$ can be obtained by firstly converting the entries $p(x \mid z)$ of the truth matrix $\mathbf{C^g}$ to their reverse conditional probabilities $p(x \mid z)$, and then applying Equation 4.6.

To obtain the reverse representation of matrix $\mathbf{C^g}$ I applied Bayesian theory. First, multiplying each column of $\mathbf{C^g}$ by the corresponding character likelihood $p(z)$ results in the following matrix

$$\begin{bmatrix} p(x=\mathrm{A} \mid z=\mathrm{A})p(z=\mathrm{A}) & p(x=\mathrm{A} \mid z=\tilde{\mathrm{A}})p(z=\tilde{\mathrm{A}}) \\ p(x=\tilde{\mathrm{A}} \mid z=\mathrm{A})p(z=\mathrm{A}) & p(x=\tilde{\mathrm{A}} \mid z=\tilde{\mathrm{A}})p(z=\tilde{\mathrm{A}}) \end{bmatrix} = \begin{bmatrix} pp_1 & (1-p)(1-\tilde{p}_1) \\ p(1-p_1) & (1-p)\tilde{p}_1 \end{bmatrix}$$

Then, dividing each row entry of the above matrix by the corresponding row-sum gives $\mathbf{R^g}$ whose entries are the probabilities $p(z \mid x)$, as follows:

$$\mathbf{R^g} = \begin{bmatrix} \dfrac{pp_1}{pp_1+(1-p)(1-\tilde{p}_1)} & \dfrac{(1-p)(1-\tilde{p}_1)}{pp_1+(1-p)(1-\tilde{p}_1)} \\ \dfrac{p(1-p_1)}{p(1-p_1)+(1-p)\tilde{p}_1} & \dfrac{(1-p)\tilde{p}_1}{p(1-p_1)+(1-p)\tilde{p}_1} \end{bmatrix}$$

Finally, the final conditional probability matrix, or association matrix $\mathbf{C}$, is determined using Equation 4.6.

Table 4-1 Conditional Probabilities for a Two Character Example

Ground Truth Character (z)	Reading at Station g (x)	Reading at Station h (y)	Joint Probability $p(z, x, y)$	Conditional Probability $p(y \mid x)$
A	A	A	pp_1p_2	$\frac{pp_1p_2 + \tilde{p}(1-\tilde{p}_1)(1-\tilde{p}_2)}{pp_1 + \tilde{p}(1-\tilde{p}_1)}$
	A	Ã	$pp_1(1-p_2)$	$\frac{pp_1(1-p_2) + \tilde{p}(1-\tilde{p}_1)\tilde{p}_2}{pp_1 + \tilde{p}(1-\tilde{p}_1)}$
	Ã	A	$p(1-p_1)p_2$	$\frac{p(1-p_1)p_2 + \tilde{p}\tilde{p}_1(1-\tilde{p}_2)}{p(1-p_1) + \tilde{p}\tilde{p}_1}$
	Ã	Ã	$p(1-p_1)(1-p_2)$	$\frac{p(1-p_1)(1-p_2) + \tilde{p}\tilde{p}_1\tilde{p}_2}{p(1-p_1) + \tilde{p}\tilde{p}_1}$
Ã	A	A	$\tilde{p}(1-\tilde{p}_1)(1-\tilde{p}_2)$	
	A	Ã	$\tilde{p}(1-\tilde{p}_1)\tilde{p}_2$	
	Ã	A	$\tilde{p}\tilde{p}_1(1-\tilde{p}_2)$	
	Ã	Ã	$\tilde{p}\tilde{p}_1\tilde{p}_2$	

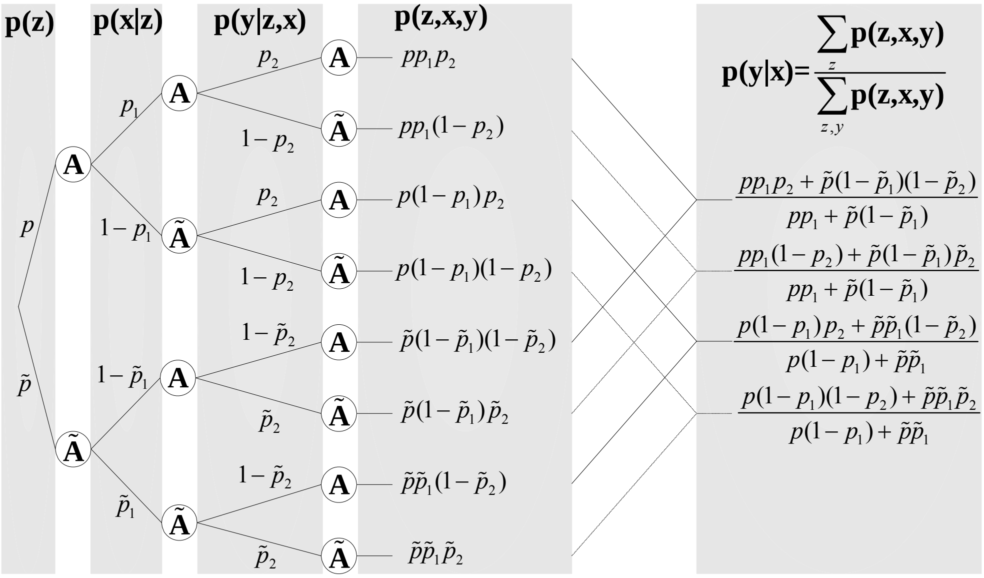

p(z)
p(x|z)
p(y|z,x)
p(z,x,y)
p
$\tilde{p}$
A
Ã
p_1
$1-p_1$
$1-\tilde{p}_1$
$\tilde{p}_1$
p_2
$1-p_2$
p_2
$1-p_2$
$1-\tilde{p}_2$
$\tilde{p}_2$
$1-\tilde{p}_2$
$\tilde{p}_2$
pp_1p_2
$pp_1(1-p_2)$
$p(1-p_1)p_2$
$p(1-p_1)(1-p_2)$
$\tilde{p}(1-\tilde{p}_1)(1-\tilde{p}_2)$
$\tilde{p}(1-\tilde{p}_1)\tilde{p}_2$
$\tilde{p}\tilde{p}_1(1-\tilde{p}_2)$
$\tilde{p}\tilde{p}_1\tilde{p}_2$
$\mathbf{p(y|x)}=\frac{\sum_z \mathbf{p(z,x,y)}}{\sum_{z,y} \mathbf{p(z,x,y)}}$
$\frac{pp_1p_2+\tilde{p}(1-\tilde{p}_1)(1-\tilde{p}_2)}{pp_1+\tilde{p}(1-\tilde{p}_1)}$
$\frac{pp_1(1-p_2)+\tilde{p}(1-\tilde{p}_1)\tilde{p}_2}{pp_1+\tilde{p}(1-\tilde{p}_1)}$
$\frac{p(1-p_1)p_2+\tilde{p}\tilde{p}_1(1-\tilde{p}_2)}{p(1-p_1)+\tilde{p}\tilde{p}_1}$
$\frac{p(1-p_1)(1-p_2)+\tilde{p}\tilde{p}_1\tilde{p}_2}{p(1-p_1)+\tilde{p}\tilde{p}_1}$

Figure 4-2 Probability Tree for a Two Character Example

4.2.3 Estimation of *p*(*y* | *x*) Based on Ground Truth (GT) Method

This section presents a method to estimate the conditional probabilities $p(y|\,x)$ based on the availability of ground truth values for a dataset of plates. From a data sample of plates captured during a period of the LPR machine operation, the corresponding ground truth values for each plate can be verified manually, and a matrix with character interpretation occurrences can be determined. Let us denote this matrix as $\mathbf{F^l}$, generated from a dataset collected from LPR machine *l*, as follows:

$$\mathbf{F^l}=\begin{bmatrix} f(r_1,z_1) & f(r_1,z_2) & \ldots & f(r_1,z_k) \\ . & . & . & . \\ . & . & . & . \\ f(r_k,z_1) & f(r_k,z_2) & \ldots & f(r_k,z_k) \end{bmatrix}$$

where $f(r_i, z_j)$ is a function which returns the number of times on the sample where a true character labeled as z_j is recognized as r_i by the machine. In a dual LPR setup there must be two matrices, $\mathbf{F^g}$ and $\mathbf{F^h}$, from the upstream and downstream stations, respectively.

From the occurrence matrix above, I can either estimate the character interpretation probabilities or the reverse probabilities as follows:

a) Character interpretation probability (cell frequency divide by column-sum):

$$\hat{p}(r_i \mid z_j) = \frac{f(r_i, z_j)}{\sum_{i=1}^{k} f(r_i, z_j)}$$

b) Reverse probability (cell frequency divided by row-sum):

$$\hat{p}(z_j \mid r_i) = \frac{f(r_i, z_j)}{\sum_{j=1}^{k} f(r_i, z_j)}$$

Thus, the estimates $\hat{\mathbf{C}}^{\mathbf{h}}$ and $\hat{\mathbf{R}}^{\mathbf{g}}$ of the truth matrix $\mathbf{C^h}$ and of the reverse matrix $\mathbf{R^g}$ can be computed from the character occurrence matrices $\mathbf{F^h}$ and $\mathbf{F^g}$, respectively. Finally, an estimate of the association matrix is given by

$$\hat{\mathbf{C}} = \hat{\mathbf{R}}^{\mathbf{g}}.\left(\hat{\mathbf{C}}^{\mathbf{h}}\right)^T$$

where each of its elements are calculated as follows

$$\hat{C}_{ij} = \sum_{s=1}^{k} \hat{p}(y_j \mid z_s)\hat{p}(z_s \mid x_i)$$

The expression above is therefore an estimator of the probability p(y_j | x_i), with the restriction that the two outcomes were originated from the same character, but not necessarily came from the same plate.

4.2.4 Editing Constraints for *CED*

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.

In this research, it is worth noting that the edit operation constraints used in *CED* are defined in the basis of 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, I proposed the following constraint sets *(i,e,s)* of insertions, deletions and substitutions to transform *x* into *y*.

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

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

c) $(i,e,s) = (0, 0, l_x),\ if\ l_y=l_x$.

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.

4.3 CASE STUDY AND RESULTS

The proposed matching methods were applied to the same LPR dataset used in Chapter 3, which involved tracking large trucks (with gross weights over 10,000 pounds) by a two-point setup on the interstate highways in Knoxville's metropolitan area.

4.3.1 Analysis Method

Since there were five days of data, all combinations of three days of data out of five were used as calibration data, with the remaining combinations of two days as validation data. Thus, each of the 10 combinations with three days of data was used to estimate 20 conditional probability matrices, i.e. 10 matrices of type $\mathbf{R^g}$ and 10 matrices of type $\mathbf{C^h}$ for LPR stations g and h, respectively. These probability matrices were then included in the formulation of *CED* and *GED* in combination with my proposed weight function defined in Subsection 4.2.1.

Considering the possible ways of defining the editing weights into the recurrent calculation of *ED* (Equation 4.1) four procedures were indentified to calculate the *ED* between pair of strings, as follows:

- $\mathbf{D_1}$: Edit distance with 1 or 0 cost assignments, which corresponds to the original idea of Levenshtein;
- **D2**: *GED* using weight function as in Equation 4.3, with the association probability $p(i_k, j_k)$ estimated by $p(y \mid x)$, as defined in Equation 4.5;
- **D3**: Original *CED* with 1 or 0 cost assignments and constrained by the editing sets defined in Subsection 4.2.4;

- **D4**: *CED* using weight function as in Equation 4.3 and $p(i_k, j_k)$ as in Equation 4.5, and constrained by the editing sets defined in Subsection 4.2.4.

The performance of my 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.

The same matching procedure described in Chapter 3 was used to assess the performance of the proposed refinement in this chapter. Thus the performance of the similarity measures was investigated under a worst case scenario that consisted in matching up two sets of plates for each remaining day, without using passage time information or the recorded time stamps. The main premise here was that under this worst case scenario the most suitable measure should be able to accurately match any two sets of plates with the least number of false matches.

Regarding the measures of performance, the percentage of positive matches and the percentage of false-positive matches was 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 percentage of false-positive matches over the domain of *ED* values.

4.3.2 Comparison of Matching Procedures

In sum, all ten profile curves of performance were similar to those presented in Figure 4-3 that contains the aggregated profile curves, or average profiles.

As can be seen in Figure 4-3, the existing *ED* or *CED* combined with my proposed weight functions yielded considerably improved performance for vehicle tracking. Second, with respect to the two measure frameworks, *GED* and *CED*, there was not any evidence of difference in performance between these two measures over all 10 analyses performed. Therefore, there is no empirical evidence yet to state that *CED* equipped with the proposed editing operation constraints, as defined in Subsection 4.2.4, is a better procedure to match outcomes from dual LPR setups.

In general, either D2 or D4 measures were able to achieve around 90% of positive matches with about 5% to 8% of false-positive 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.

4.4 CHAPTER CONCLUSIONS AND RECOMMENDATIONS

This chapter assessed the performance of different vehicle tracking procedures using a dual setup (two-point setup) of LPR units. I have proposed a general and simple procedure to compute the weight function that can be used in existing distance measures when the truth matrix of the LPR machines are available or can be estimated. A field study using two LPR units was conducted on I-40 in the vicinity of Knoxville, Tennessee in 2007. The experimental results and analyses show that the most suitable procedures for vehicle tracking on a dual LPR setup are either *GED* or *CED* formulation combined with the weight function and editing constraints proposed in this chapter. These procedures achieved about 90% of positive matches with only 5% to 8% of false-positive matches.

It is worth noting that the performance results obtained with the application of the proposed weight function in the *ED* formulation outperformed the original cost assignments. This partly confirmed the hypothesis of this research that the recurrent LPR errors can be used to infer about the likelihood of two imperfect readings being originated from the same vehicle.

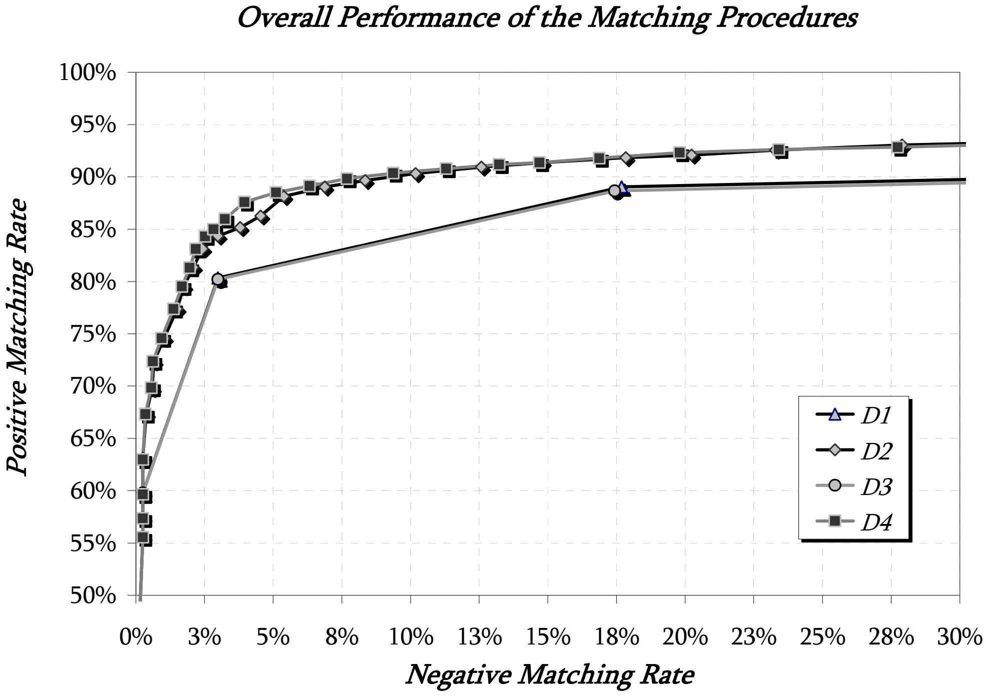


Figure 4-3 Performance of the Matching Procedures

In this analysis, the matching procedures were applied with no consideration of passage time stamps. In reality, there will always be a time limitation (one day, one hour, etc) corresponding to the survey period or to an arbitrary restriction (e.g. maximum number of candidates for matching) that constrains the possible pair-wise matches. The question is how to take judicious advantage of the time stamps to further improve the matching procedure. This will be subject of Chapter 5.

A very important issue that has not been yet considered here in the present analysis is the sample size (number of outcomes) needed to estimate the character association matrix **C**. A bad estimate of this matrix can deteriorate the matching performance. This matrix is site-dependent as it reflects the characteristic (plate, vehicle, environment, etc) of the locations where the LPR machines are installed. Therefore, if there are many sources of variation, noise, affecting the LPR operation, it would be necessary a large amount of data to achieve a required error precision. In such case, the estimation would require also more human intervention since the ground truth values of the plates are determined manually.

In situations where the LPR units are installed permanently, the human effort can be eliminated if there is a mechanism of estimating the conditional probabilities of the association matrix **C** by means of a 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 Chapter 6.

CHAPTER 5

MATCHING PROCEDURE COMBINED WITH PASSAGE TIME INFORMATION

The passage time stamps in registration plate surveys are an important part of the information collected in this type of study, since they are used to estimate the travel times of vehicles travelling from origin to the destination checkpoints. Whenever it is possible to record the vehicle plate number exactly, the travel times are derived directly from the exact matches. However, if there is uncertainty about the validity of matches, passage times are essential information to infer about the likelihood of genuine matches, since they follow a certain pattern or distribution. In the previous chapters 4 and 5, the database from LPR setup was matched up without using passage time information. In this chapter, I take advantage of this information, available in the LPR data, to help improving the performance of the matching framework.

5.1 EXISTING PLATE MATCHING PROCEDURES CONSIDERING PASSAGE TIME

Most of the earliest research on matching plate data was concerned with correcting for spurious matches of partial plate surveys. Beyond this, relatively little research has been also undertaken on methods to filter out erroneous records from matched LPR data. It seems that little research has been conducted about the use of passage time information in a matching procedure for LPR data.

5.1.1 Partial Plate Problem

Most of the research in vehicle plate matching has focused on the problem of estimating or correcting for spurious matches generated from partial plate surveys. To this end, many statistically based methods have been proposed to mitigate the false matching problem in the case of two observation point survey (i.e. a single origin and destination), such as those of Hauer (1978), Shewey (1983), Maher (1985), and Watling and Maher (1988). It seems that to handle the case of multiple origin and destinations the main contributions are due to Watling and Maher (1992) and Watling (1994).

Regarding to partial plate problem, researchers strived in the beginning to find the most probable set of matches with no use of passage time information, other than the restriction of the survey period (see for example: Makowski and Sinha, 1976; Hauer, 1978; and Maher, 1985). In the case of single origin-destination, they assumed Poisson arrivals in the destination for vehicles not detected in the origin, as well as assumed that the number of vehicles that pass through both stations is a binomial variable. Hauer (1976) proposed a heuristic approach to estimate the number of spurious matches and an interactive procedure to correct the data from license plate surveys. Later Maher (1985) proposed two statistical procedures (Maximum Likelihood Estimation and Least Square Estimation) to estimate the parameters (proportion of vehicles detected at first station that travel to the destination and arrival rate for vehicles at the second station not detected at the first one) of the distributions functions used into their models.

Shewey (1983) was the first to propose the use of the passage times to improve the performance of the matching problem for the single origin-destination. In his approach each set of possible matches must initially satisfy a fixed time window constraint which is determined by a priori estimate of the minimum and maximum possible journey times. Besides obeying the time window constraint, the chosen match would be the one whose passage time difference were the closest to the mid point of the time interval. This additional restriction was therefore an attempt to consider the shape of the travel time density function, assumed to be a symmetric density function such that it is more likely to observe a travel time close to the median or the mean value.

Watling and Maher (1992) assumed that the journey time between stations follows a normal distribution, in addition to the other assumptions proposed before, to derive a solution method for the problem based on a well known linear optimization programming. At first the objective function seeks to maximize a likelihood function of the most probable combination of vehicles conditional on the data. The problem is further transformed into a minimization problem by taking the negative logarithm of the likelihood function, becoming a problem similar to transportation problem (Hitchcock, 1941.), or the assignment problem (Bazaraa et al., 2005).

Since the parameters of model are unknown a priori, Watling and Maher (1992) proposed an interactive method to estimate them and find the most probable matches. Subsequently, Waitling (1994) proposed a maximum likelihood approach to estimate the model parameters. The models proposed by Watling and Maher (1992), as well as by Watling (1994), therefore use information of traffic flow and passage times to find the most probable vehicle combination for a given dataset of partial plate numbers. It is also demonstrated that their models are easily extended to the case of multiple origin and destinations.

5.1.2 Cleaning of Matched LPR Data

More recent studies involving the treatment of journey times estimated from LPR data focused on the problem of removing outliers from a set of matched license plate data (Clark et al, 2002; Robison and Polak, 2006). In such studies only ground truth matches (matches from pair-wise outcomes with the same value) were analyzed. It is important to notice that this is different from what has been proposed in early studies, in the case of partial plate surveys, where the aim was to use journey time as an additional constraint or embedded model parameter to classify matches in genuine and spurious.

Although cleaning methods are not directly of interest in this research, they offer insights on how to identify valid journey time estimates, or journey times of vehicles travelling in direct fashion from the origin point to the destination point. Therefore, a cleaning method could be combined with the proposed similarity measure into a new matching procedure.

The literature offers some reports on methods to clean up travel times estimated using LPR equipment. Clark et al. (2002) presented three methods to identify outliers in journey time observations. In this context, outliers are defined as any observation generated from either errors in database due to the inaccuracy of the equipment or unexpected travel behavior (such as stop en-route, and vehicle not restricted to traffic regulations). In order to distinguish outliers from incidents or normal variation in traffic behavior they pointed out that the journey time records should be analyzed on small time blocks within the survey period. They applied the three methods to clean up a dataset collected on motorway around Manchester, United Kingdom, and found that the most robust method, first proposed by Fowkes (1983), would be the one using the median as a measure of location and the interquartile range as spreading measure.

For LPR setups on urban area, Robinson and Polak (2006) have proposed a method which uses the serial temporal structure of the data. They emphasized that previous methods ignore such serial temporal structure by assuming that the traffic conditions are stationary within 5- or 15-min period. Hence, they suggested an overtaking rule which requires that if a vehicle is overtaken on a multiple lane link of an urban environment, then it will not have a travel time very distinct from the overtaking vehicle. They investigated their method using simulation and achieved better performances (defined in terms of the ability to identify valid vehicles and the ability to estimate the expected mean and standard deviation of valid journey times) compared to existing methods.

5.1.3 LPR Plate Survey

All models proposed to estimate the number of false matches from partial plate surveys assumed that the observers do not make any mistake in reading the plate numbers. In reality, the accuracy of such surveys is usually poor in harsh conditions of weather, lightening, and high traffic volumes. In addition, they are very time consuming and require high amounts of manpower to collect, tabulate and analyze the data. Therefore, as an alternative to overcome these problems, LPR machines can replace the observers and, by knowledge of the possible character errors made by the machines, the whole process to obtain the set of genuine matches between stations can be automated.

Data from LPR surveys have distinct characteristics compared to the data generated from partial plate surveys. First, LPR machines try to record all characters on the plates and whenever there is an error or a missing character it does not happen at the same location on the plate. Therefore, a database of readings collected from a single LPR unit is composed of distinct reading outcomes; rather than consisted of blocks of reading replications usually encountered in partial plate surveys, where only part of the plate number is recorded. Recall that models to deal with the partial plate problem are not useful, or needed, when replications are not observed.

Although in the LPR surveys the resulting outcomes seem to be distinct for a same station, there is no guarantee that the same outcome will be observed in different stations for a given plate number. As seen in Chapter 3 and 4, measures of similarity between strings proved to be a suitable method to determine the likelihood of a genuine match when two strings are paired up. Furthermore, as was the case for the partial plate problem, the likelihood of finding false matches can be decreased using an additional constraint which takes advantage of the passage times available in the LPR database.

5.2 PROPOSED METHODS

In this section two methods that incorporate the passage time information into a matching framework combined with the similarity measure proposed in Chapter 4 are presented. The two procedures were named Fixed Time Window Constraint (FTWC) and Varying Time Window Constraint (VTWC) methods.

Initially, let us define the notation used. Set the pair-wise strings (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*. Define u_i and v_j as the corresponding time stamps at station *g* and *h*, respectively. So, the difference in passage time is denoted by $t_{ij} = v_j - u_i$. Furthermore, $d_{ij} = d(x_i , y_j)$ denotes the similarity measure between x_i and y_j.

5.2.1 Fixed Time Window Constraint (FTWC)

In this section the fixed time window constraint method is described. In this method, for each outcome y_j (the j^{th} record) from the downstream station (Station *h*) a set of candidates in the upstream station (Station *g*) is first limited to those outcomes such that the passage time stamps fall within a time window constraint. This time constraint is a fixed time interval defined, as illustrated in Figure 5-1, by the following expression:

$$v_j - jt_u \le u_i \le v_j - jt_l \tag{5.1}$$

where:

u_i and v_j, are the time stamps of the i^{th} and j^{th} outcomes, recorded at stations *g* and *h*, respectively;

jt_l and jt_u are an estimate of the upper and lower bounds of the journey time.

Once the set of candidates is selected for a given outcome of Station *h*, the further procedure, illustrated in Figure 5-2, is similar to the matching procedure described in Section 3.3 of Chapter 3. First, the candidate x_i (i^{th} record at station *g*) whose match (x_i, y_j) has the least weighted edit distance d_{ij} is then chosen for further analysis. Second, a test is applied to verify if the chosen x_i outcome in this current match has been already matched before, where in such case the match with higher d_{ij} cost is removed from the matching list. Finally, if d_{ij} is less than a threshold τ, the chosen match is potentially genuine, otherwise may be false.

Recall that the edit distance d_{ij} here can be calculated using any procedure already described in Chapters 3 and 4, i.e. either with 0 or 1 cost assignments or using weights estimated from the recurrent character interpretations occurred during the LPR operation, as described in Chapter 4.

The time window constraint is defined according to the application objective. For example, if the goal is to identify speeding vehicles, the upper limit can be defined in terms of the minimum speed limit of the road, whereas the lower limit as function of a typical vehicle acceleration capacity.

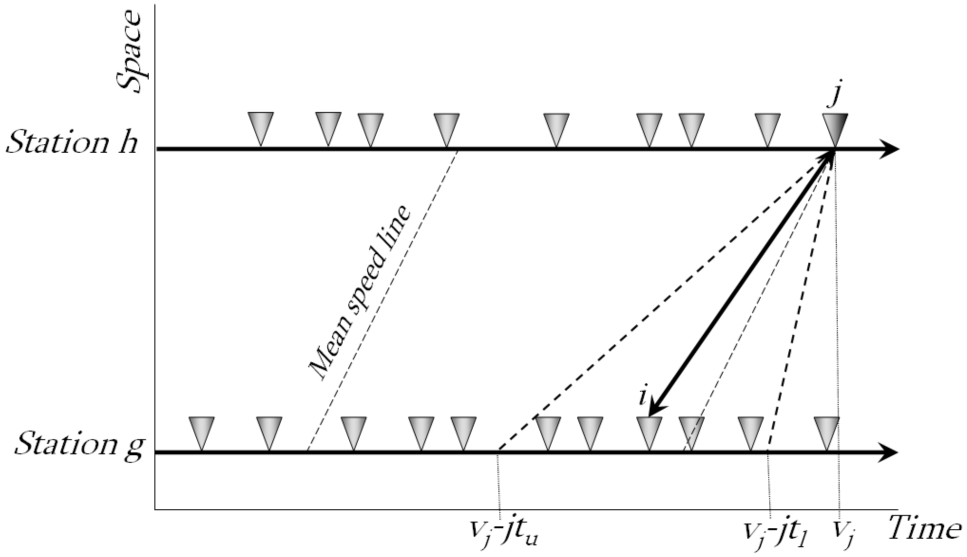


Figure 5-1 Time Space Diagram with Fixed Time Window Constraint

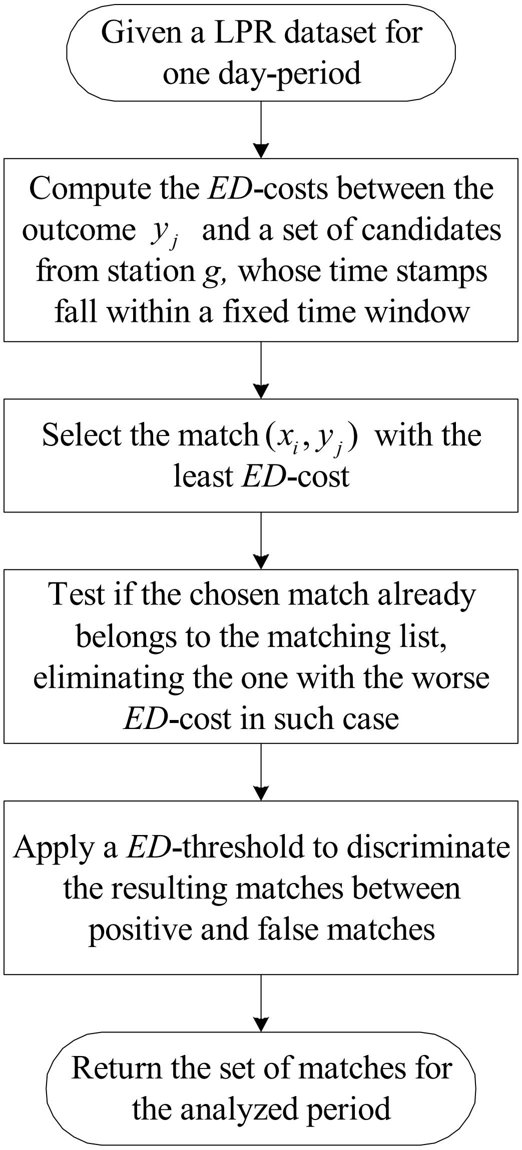

Given a LPR dataset for one day-period
Compute the *ED*-costs between the outcome y_j and a set of candidates from station *g,* whose time stamps fall within a fixed time window
Select the match (x_i, y_j) with the least *ED*-cost
Test if the chosen match already belongs to the matching list, eliminating the one with the worse *ED*-cost in such case
Apply a *ED*-threshold to discriminate the resulting matches between positive and false matches
Return the set of matches for the analyzed period

Figure 5-2 Flowchart Process for the FTWC Method

On the other hand, when the objective is to monitor the traffic conditions or to find the trip patterns on the urban environment, the limits can be defined arbitrarily by assuming that the maximum vehicle speed is infinity (corresponding to a lower journey time of zero) and that the minimum vehicle speed is very low such as 1 or 2 mph (so that the upper journey time is much larger than the expected journey time). This arbitrary restriction can also be used with the purpose of comparing the discriminative power of different similarity measures.

5.2.2 Varying Time Window Constraint (VTWC)

Regarding to the two-point LPR survey again, in this section a second matching procedure incorporating the passage time information is proposed to improve the performance of the template matching. Similarly to the FTWC method, this procedure is thought to be used in situations such that it is needed to decide whether or not a plate currently detected at the downstream station *h* can be matched to a subset of plates already detected at the upstream station *g*.

Again, my proposed matching procedure consists in matching any current outcome y_j at Station *h* to a subset of the earliest previous observations at Station *g*. As in the FTWC method, the subset of candidates at Station *g* is formed by those outcomes whose corresponding passage times fall within a fixed time window constraint. As before, such time window constraint is bounded by the upper and lower limits of the expected travel times on the road. However, in the VTWC method here there is an additional time window constraint whose width varies according to the likelihood of classifying a match in either genuine or false. The procedure will be detailed in the sequel.

Let us first define the range of values for the similarity measure d_{ij} as $[0, \tau^{\max}]$, where $\tau^{\max}$ is a critical value above which is very unlikely to declare (x_i, y_j) as a genuine match. The limit $\tau^{\max}$ is also viewed as a value below which is very unlikely to have a false math. As can be seen, there is a tradeoff on the determination of $\tau^{\max}$.
Finally, let us define the range $[0, \tau^*]$, such that $\tau^* \in (0, \tau^{\max})$, within which it is very likely that (x_i, y_j) constitutes a genuine match.

The matching procedure, as illustrated in the flowchart of Figure 5-3 and in the time-space diagram of Figure 5-4, 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* whose passage times fall within a fixed time window, and search among the candidates for the best string x_i with the least edit distance; 2) If $d_{ij} < \tau^*$, declare the match (x_i, y_j) as genuine; 2) Otherwise, if $\tau^* < d_{ij} < \tau^{\max}$, declare (x_i, y_j) a valid match only if the corresponding passage time difference t_{ij} lies within a varying time window constraint whose width varies with the magnitude of d_{ij} and with the expected mean and standard deviation of the vehicle journey time.

By assuming that the genuine journey times come from a symmetric density function, such as the Gaussian density function, the closer t_{ij} is to the mean of the distribution, more likely is the match (x_i, y_j) 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 $(\tau^*, \tau^{\max}]$, the

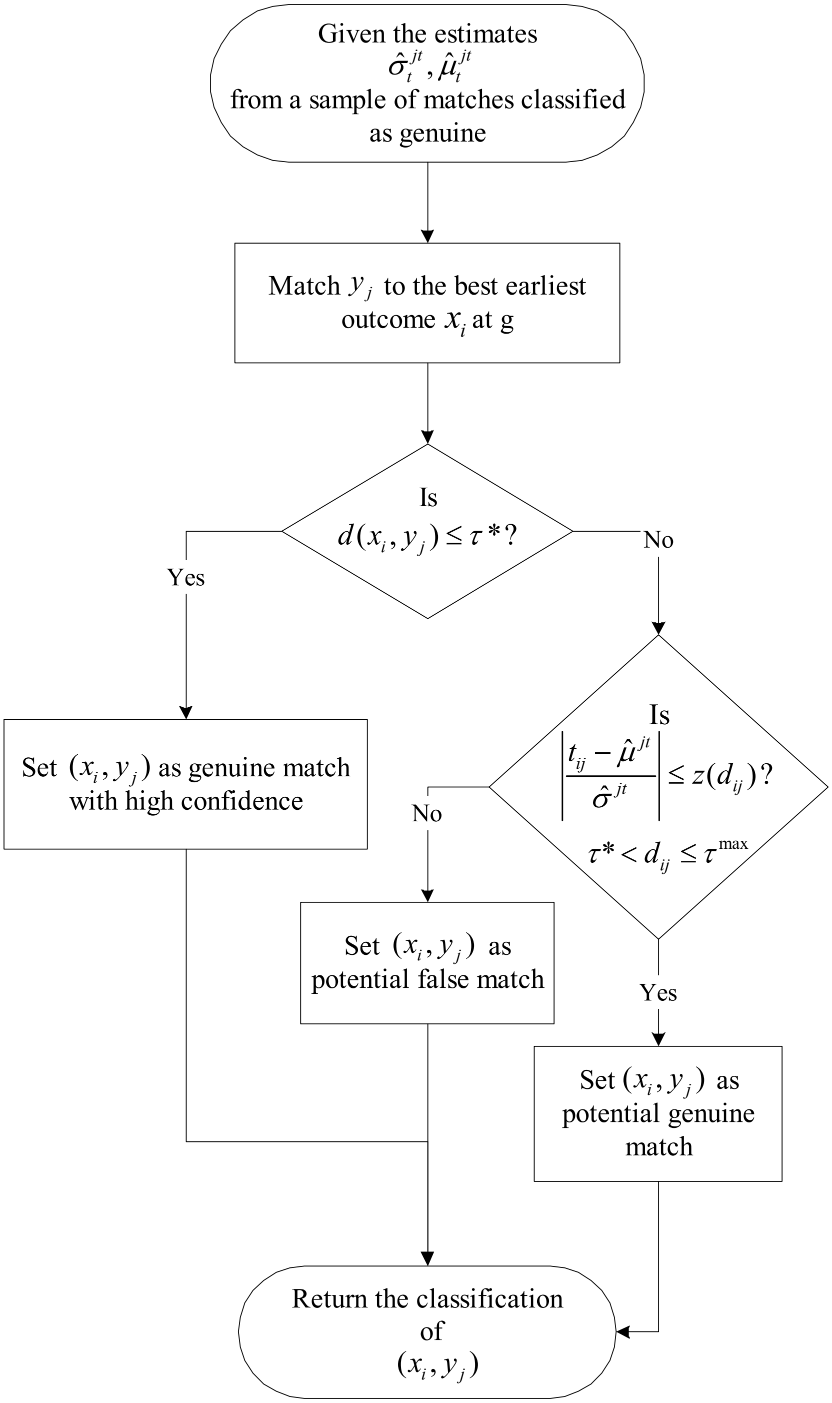

Given the estimates
$\hat{\sigma}_t^{jt}, \hat{\mu}_t^{jt}$
from a sample of matches classified as genuine
Match y_j to the best earliest outcome x_i at g
Is $d(x_i, y_j) \le \tau^*$?
Yes
No
Set (x_i, y_j) as genuine match with high confidence
Is $\left|\frac{t_{ij} - \hat{\mu}^{jt}}{\hat{\sigma}^{jt}}\right| \le z(d_{ij})$?
$\tau^* < d_{ij} \le \tau^{\max}$
No
Set (x_i, y_j) as potential false match
Yes
Set (x_i, y_j) as potential genuine match
Return the classification of (x_i, y_j)

Figure 5-3 Flowchart Process for the VTWC Method

following inequality constraint was defined

$$\left|\frac{t_{ij}-\mu_t^{jt}}{\sigma_t^{jt}}\right| \leq z(d_{ij}), \qquad \tau^* < d_{ij} < \tau^{\max} \tag{5.2}$$

where,

μ_t^{jt} and σ_t^{jt} are the expected mean and standard deviation of the journey times for the corresponding time *t* of a typical day;

z(d_{ij}) is a monotonically decreasing function of the similarity measure value d_{ij} and used to define the limits of the varying time window constraint.

Now, the upper and lower bounds of the journey time vary according to the value of d_{ij} and with the expected journey time parameters μ_t^{jt} and σ_t^{jt}. Although it seems to be a method to remove outliers (cleaning data method), it is just a way to classify matches in genuine or spurious. The upper and lower bounds, shown in Figure 5-4, for the journey times can be redefined in terms of the following functions:

$$jt_u(d_{ij},\mu_t^{jt},\sigma_t^{jt}) = \begin{cases} \Delta x / s_l, & if\ d_{ij} \leq \tau^* \\ \mu_t^{jt} + z\left(d_{ij}\right).\sigma_t^{jt}, & if\ \tau^* < d_{ij} \leq \tau^{\max} \end{cases}$$

$$jt_l(d_{ij},\mu_t^{jt},\sigma_t^{jt}) = \begin{cases} \Delta x / s_u, & if\ d_{ij} \leq \tau^* \\ \mu_t^{jt} - z\left(d_{ij}\right).\sigma_t^{jt}, & if\ \tau^* < d_{ij} \leq \tau^{\max} \end{cases}$$

Where,

s_l and s_u are estimates of the lower and upper limits of the vehicle speeds;

Δx is the distance separation between the two LPR unit locations;

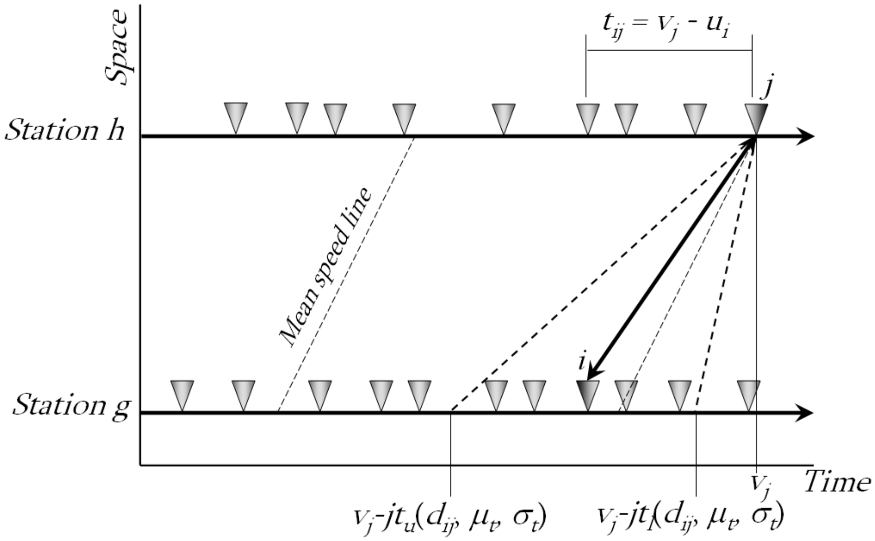

$t_{ij} = v_j - u_i$
j
Space
Station h
Mean speed line
i
Station g
v_j
Time
v_j-$jt_u(d_{ij}, \mu_t, \sigma_t)$
v_j-$jt_l(d_{ij}, \mu_t, \sigma_t)$

Figure 5-4 Time Space Diagram with Varying Time Window Constraint

The constraint of Equation 5.2 is thus sensitive to both variation on the journey time through a day and to the likelihood (given by the similarity measure) of a match (x_i, y_j) being genuine. Figure 5-5 shows an example of travel time variation over time for a typical day, with the upper and lower journey time limits defined by the top and bottom dashed lines.

In practice, true values for μ_t^{jt} and σ_t^{jt} are impossible to obtain. An alternative is to estimate these two parameters from historical data of journey times obtained in time intervals near to the time point *t* or use previous journey time values from matches already classified as genuine.

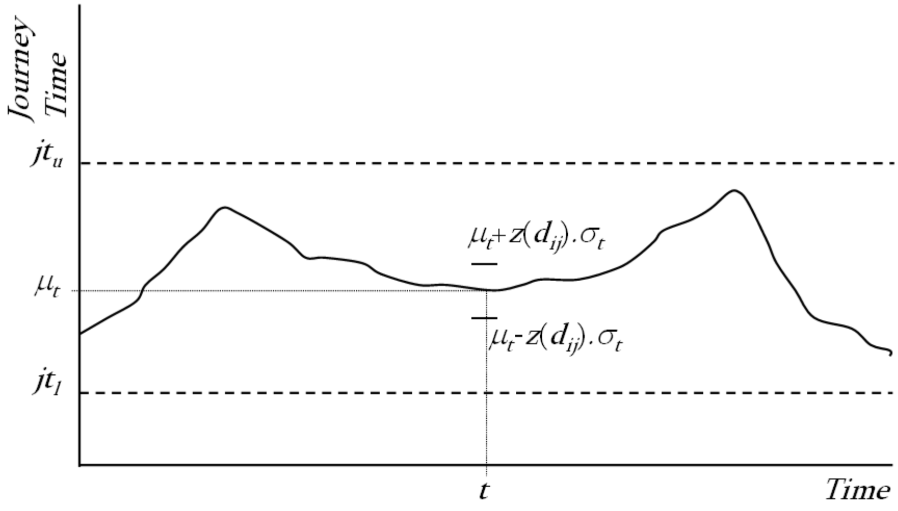


Figure 5-5 Profile Example of Journey Time Variation

5.3 LPR DATA

Two LPR setups were deployed to test the proposed matching procedures. The first setup (*LPR Setup 1 – 2007 Data*) was the same used in Chapters 3 and 4 to evaluate different formulations of the similarity measure. Remind that the main objective of this application was to monitor the speed of large trucks.

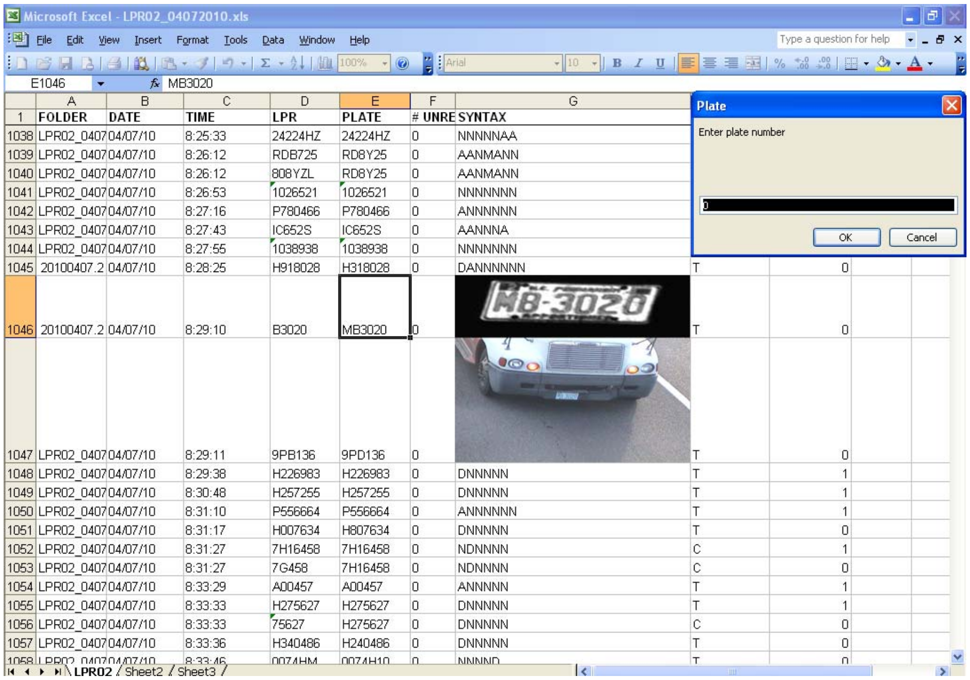
The second setup (*LPR Setup 2*) corresponded to a permanent dual LPR setup recently installed in Knoxville metropolitan area. Two newer versions of the PIPS cameras were set up three miles apart. The first one located in I-640 W (Station *g*) at Pleasant Ridge Rd and the second one located in I-40 W (Station *h*) before the exit ramp with Papermill Dr. Both LPR units were mounted on the existing structure of variable message panels, and aimed to the right middle lane to capture plates on the front of vehicles travelling westbound.

They were installed on April of 2010, and have been operating day-and-night since then. All plate data collected have been also stored for posterior analysis. Five complete days of operation, April 6th and 7th, as well as May 25th, 26th and 27th, were selected to evaluate the LPR operation and the performance of the matching procedures. Subsection 5.3.1, next, presents the results from these five days of operation.

5.3.1 Permanent LPR Setup: Data Collection

The new versions of the LPR equipment used at *LPR Setup 2* have the ability to continuously take pictures at a rate of 30 pictures per minute, so that each vehicle is caught 5 times on average. Although multiple pictures may be taken for a single vehicle, only those pictures resulting in distinct outcomes are saved. Thus, one single vehicle may generate multiple outcomes with distinct values. For each outcome, two image files are created, one with a closed view of the plate and another one showing the whole vehicle front. All information retrieved (LPR station, image view, time stamps, outcome values, reading confidence) is saved into a code of strings separated by commas, which is the name of each image file.

A script in Excel Visual Basic was written with the objective of compiling the LPR data and assisting in the process of comparing the raw images with the detected reports. Figure 5-6 shows a snapshot view of the spreadsheet used. This script made possible to automatically retrieve the reading values and time stamps for each outcome. More important, it also made possible to sequentially view the images of the plates and of the vehicle fronts, from which the true plate values were verified and recorded into the


Microsoft Excel - LPR02_04072010.xls
File Edit View Insert Format Tools Data Window Help
Type a question for help
100%
Arial
10
LPR02
Sheet2
Sheet3

E1046 MB3020

Plate

Enter plate number

0

OK Cancel

	A	B	C	D	E	F	G		
1	FOLDER	DATE	TIME	LPR	PLATE	# UNRE	SYNTAX		
1038	LPR02_0407	04/07/10	8:25:33	24224HZ	24224HZ	0	NNNNNAA		
1039	LPR02_0407	04/07/10	8:26:12	RDB725	RD8Y25	0	AANMANN		
1040	LPR02_0407	04/07/10	8:26:12	808YZL	RD8Y25	0	AANMANN		
1041	LPR02_0407	04/07/10	8:26:53	1026521	1026521	0	NNNNNNN		
1042	LPR02_0407	04/07/10	8:27:16	P780466	P780466	0	ANNNNNN		
1043	LPR02_0407	04/07/10	8:27:43	IC652S	IC652S	0	AANNNA		
1044	LPR02_0407	04/07/10	8:27:55	1038938	1038938	0	NNNNNNN		
1045	20100407.2	04/07/10	8:28:25	H918028	H318028	0	DANNNNNN	T	0
1046	20100407.2	04/07/10	8:29:10	B3020	MB3020	0		T	0
1047	LPR02_0407	04/07/10	8:29:11	9PB136	9PD136	0		T	0
1048	LPR02_0407	04/07/10	8:29:38	H226983	H226983	0	DNNNNN	T	1
1049	LPR02_0407	04/07/10	8:30:48	H257255	H257255	0	DNNNNN	T	1
1050	LPR02_0407	04/07/10	8:31:10	P556664	P556664	0	ANNNNNN	T	1
1051	LPR02_0407	04/07/10	8:31:17	H007634	H807634	0	DNNNNN	T	0
1052	LPR02_0407	04/07/10	8:31:27	7H16458	7H16458	0	NDNNNN	C	1
1053	LPR02_0407	04/07/10	8:31:27	7G458	7H16458	0	NDNNNN	C	0
1054	LPR02_0407	04/07/10	8:33:29	A00457	A00457	0	ANNNNN	T	1
1055	LPR02_0407	04/07/10	8:33:33	H275627	H275627	0	DNNNNN	T	1
1056	LPR02_0407	04/07/10	8:33:33	75627	H275627	0	DNNNNN	C	0
1057	LPR02_0407	04/07/10	8:33:36	H340486	H240486	0	DNNNNN	T	0
1058	LPR02_0407	04/07/10	8:33:46	0074HM	0074H10	0	NNNND	T	0

Figure 5-6 Screen Snapshot View of the Excel File used to Process the LPR Data

designed cells. In addition, the plate syntax and type of vehicles (car or trucks) were also recorded.

During the selected five days of operation, a total 10450 and 21266 plates were captured at stations *g* and *h*, respectively. Among these, a total of 2924 were manually verified as identical. Thus, around 28% of the vehicles detected at the first station were also detected at the second station.

5.3.2 LPR Performance

With respect to the LPR performance, Table 5-1 shows the plate-based accuracy and character-based reading rate for both LPR setups. As can be seen, the performance of the newer setup was poorer than that observed for the first application, even with newer versions and permanently mounted equipments. The main reason for the worse performance of the second LPR setup, compared to the first LPR setup, was that the machine settings in the second setup were not calibrated to recognize specifically plates from Tennessee which are the majority of plates found.

Table 5-2 shows the matching rates separated per machine failure (read or misread) of both 787 and 2924 plates manually identified as identical, from *LPR Setups 1* and *2*, respectively. As already pointed out for *LPR Setup 1* in Chapter 3, there was a propensity for a plate being correctly recognized at the second station given that it has been correctly read at the first station.

5.3.3 Analysis of the Vehicle Speeds and Journey Times

Regarding to the first LPR database, all data was collected during the afternoon off-peak (1:00-4:00 PM). The histogram of the sample speeds of the 787 trucks captured at both stations is replicated at Figure 5-7a. The empirical distribution had a kurtosis of 4.4599 and skewness of -0.6131, thus revealing a sample distribution highly concentrated around the mean and spread out more to the left. Therefore, for the analyzed period, it was more likely to find journey times around the mean value and eventually there were observations with larger journey times, compared to the expected values.

Table 5-1 LPR Performance

LPR Setup:	*1) 2007 – Mobile*		*2) 2010 – Permanent*	
Station:	*Campbell*	*Weight*	*I-640W*	*I-40W*
Plate reading accuracy:	61%	63%	26%	57%
Character reading rate:	0.88	0.90	0.78	0.85

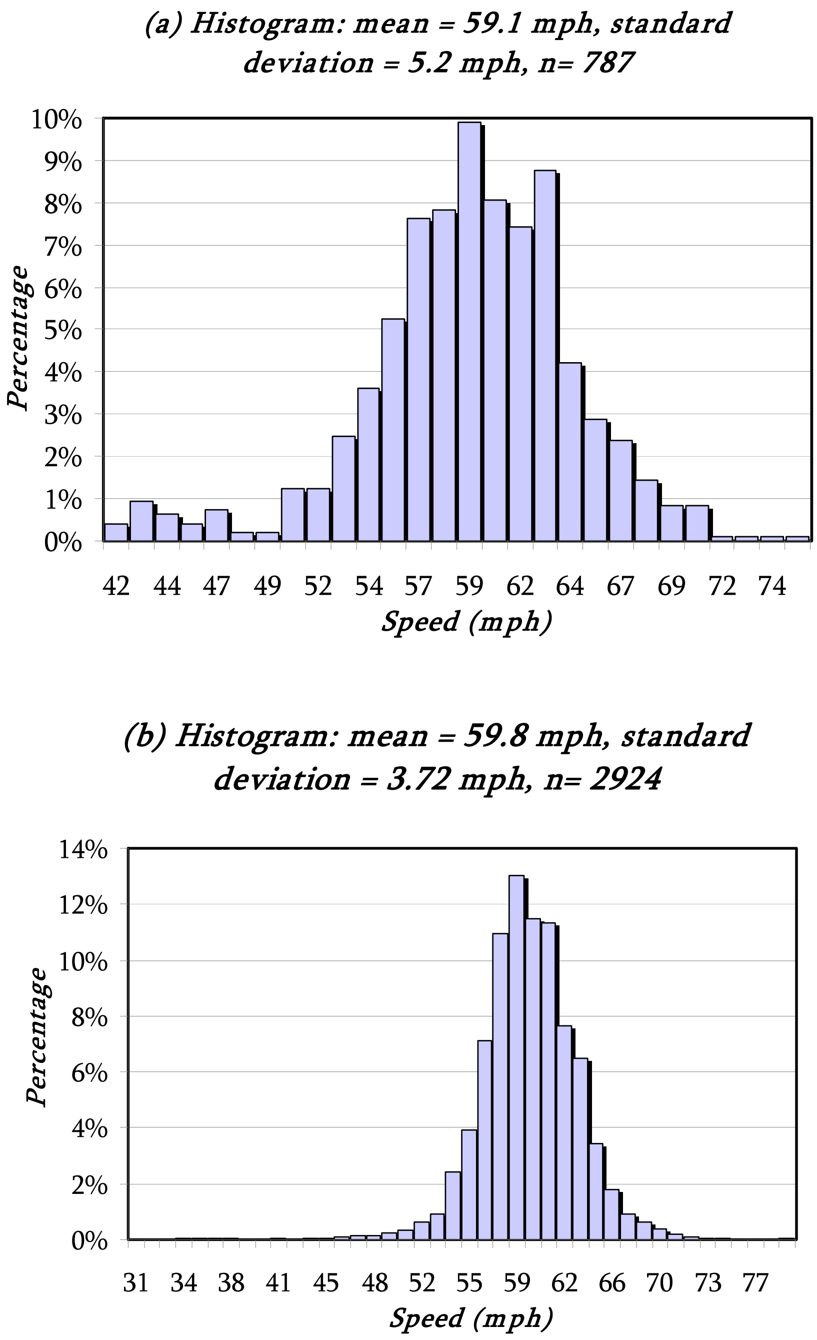
Table 5-2 Cross Table of Expected Matching Rates per Machine Failure

2007 Setup			2010 Setup		
787 plates	*h – misread*	*h – read*	***2924 plates***	*h – misread*	*h – read*
g – misread	19%	12%	*g – misread*	32%	36%
g – read	15%	53%	*g – read*	4%	27%

With respect to the second LPR database, Figure 5-7b shows a histogram of the sample speeds for the sample of 2924 vehicles travelling through both stations, and manually verified as identical. Again, the empirical distribution of speeds was slightly asymmetric to the left and with high density around the mean (kurtosis of 11.4704 and skewness of -0.9367).

Figure 5-8a shows the scattering plot and the moving average profile (for blocks of 10 observations) of the journey time for one day period of analysis of the first LPR database. This chart is only to illustrate a time profile view of the journey time variation. More data would be necessary to characterize statistically a typical profile. Observe that between 1:40-1:50 PM the peak traffic is ending (which explains in part the asymmetry of the speed distribution). Besides, notice that between 3:00-3:10 PM, as well as for other shorter periods, no vehicle was detected at both stations, and therefore the moving average was unchanged.

Again as an illustration, the scattering plot and the 24-hour profile of the moving average of the journey time for one day of analysis of the second LPR database are presented in Figure 5-8b. As before, this chart provided a disaggregated view of the travel time variation not possible to observe by a simple histogram.


(a) Histogram: mean = 59.1 mph, standard deviation = 5.2 mph, n= 787

(b) Histogram: mean = 59.8 mph, standard deviation = 3.72 mph, n= 2924

Figure 5-7 Histogram of the Sample Vehicle's Speeds

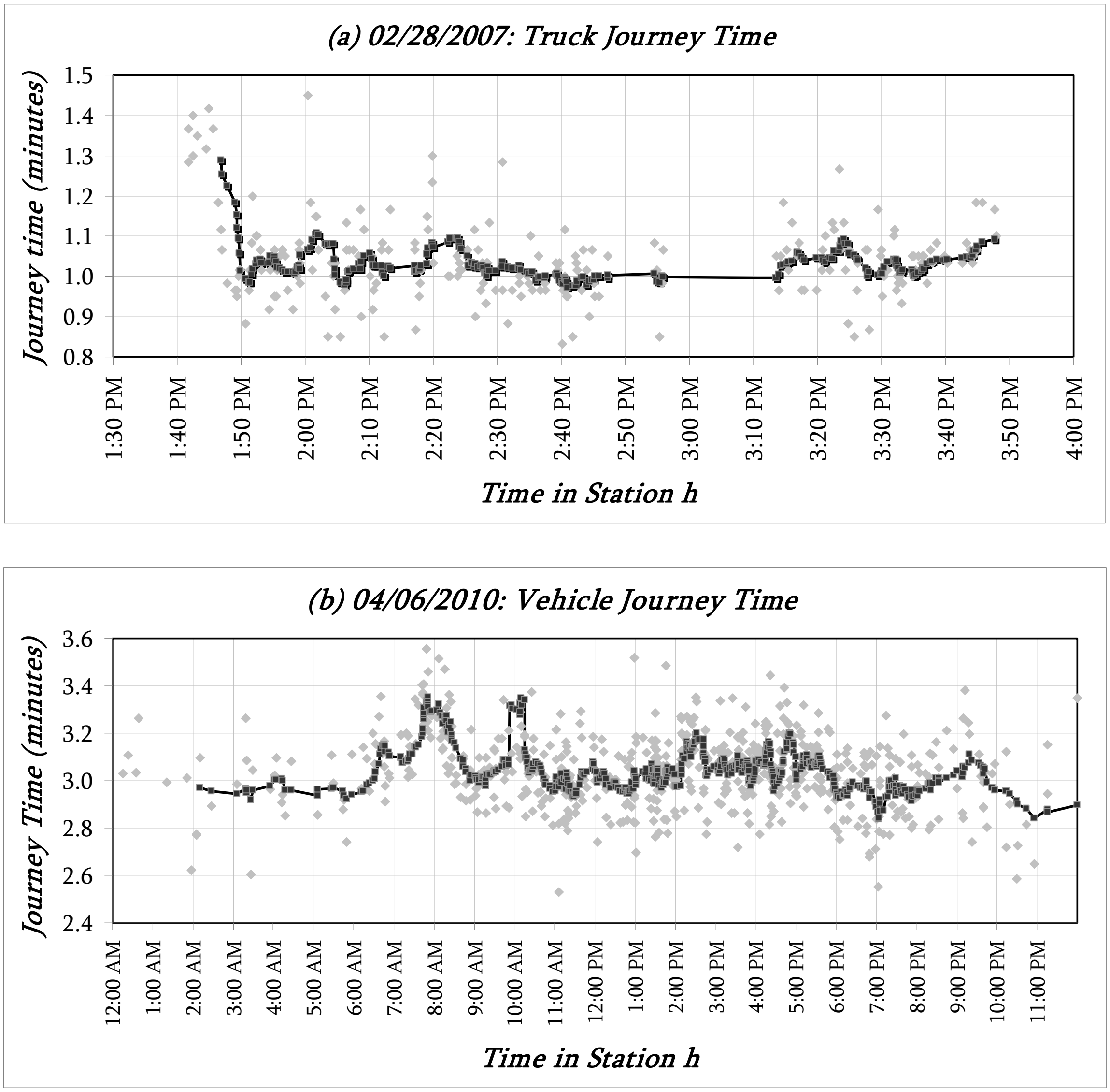


Figure 5-8 Scattering Plot and Moving Average of the Journey Times

5.4 MATCHING PROCEDURES EVALUATION

Each database (from *LPR Setup 1* and *LPR Setup 2*) was divided into two parts: one for calibration of the model parameters and the other for comparison of the performances of the similarity measures. The same procedure described in Chapter 4 was used to separate the data. Therefore, for each five days of data, all possible combinations of three datasets out of five were used as calibration data, with the remaining combinations of two datasets as validation data. Thus, each of the 10 combinations containing three days of data was used to estimate 20 conditional probability matrices, i.e. 10 matrices of type $\mathbf{R^g}$ and 10 matrices of type $\mathbf{C^h}$ for LPR stations *g* and *h*, respectively. These probability matrices were then used as parameter arguments for the proposed weight function (Chapter 4, Subsection 4.2.1) in the *GED* formulation, with the additional passage time constraint defined by either FTWC or VTWC methods.

5.4.1 Fixed Time Window Constraint Performance

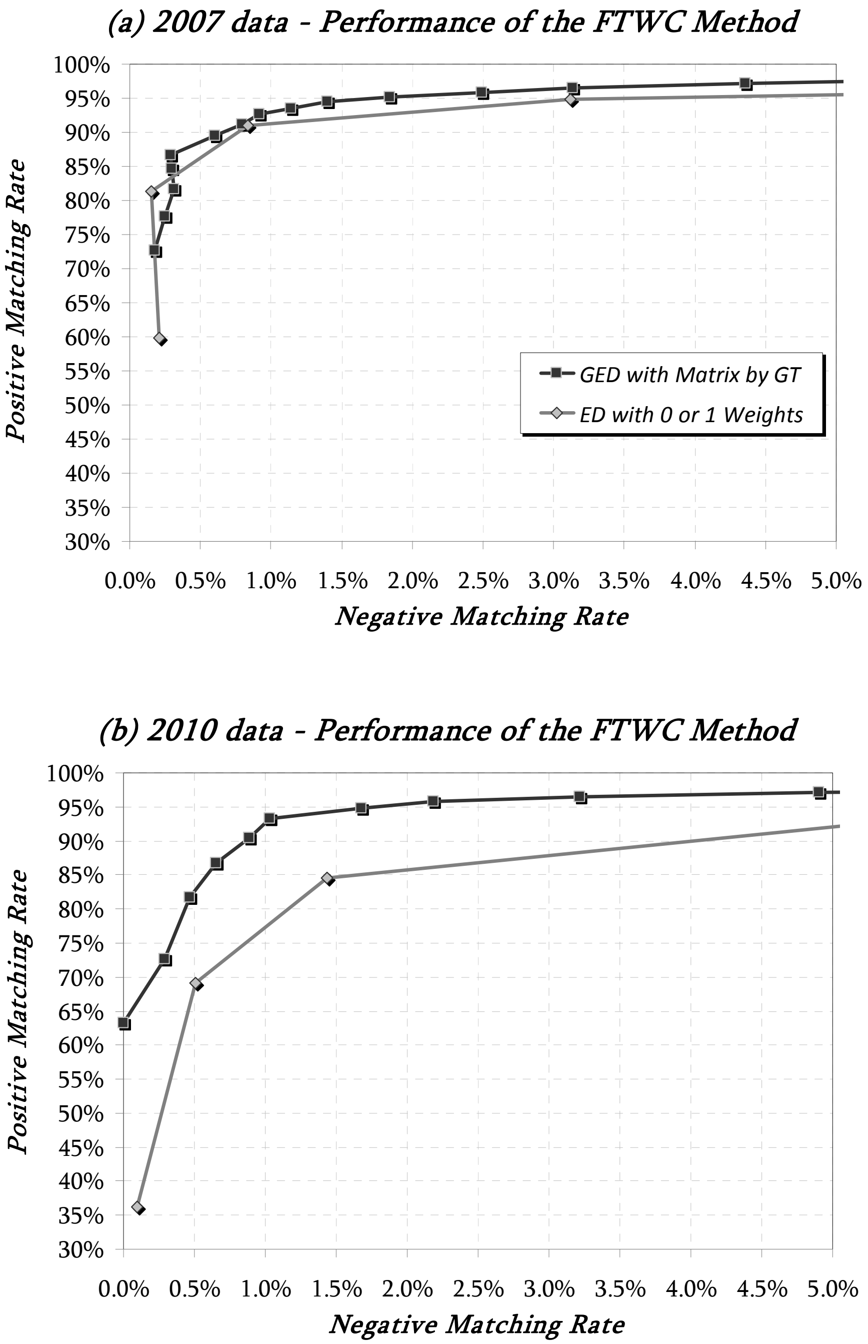
The FTWC method was applied to all 10 dataset combinations used for validation from the two LPR databases. Taking into account that this study dealt with LPR setups in freeways, the fixed time window constraint was defined assuming an upper bound for a vehicle speed of 90 mph and a lower bound of 35 mph. Two similarity measurement formulations were used for comparison: *GED* with weights as proposed in Chapter 4 (Section 4.2), and, as a base scenario, the original *ED* with 0 and 1 weights.

Figure 5-9 shows the average curves of performance, positive versus false matching rates, of the *GED* (at a threshold value ranging from 5 to 20) and of the original *ED* (for a range of 0 to 5) formulation. Compared to the performance without using passage time information (Chapter 4 results), merely using a fixed time window as secondary restriction in the matching procedure significantly reduced the false matching rate.

Furthermore, the *GED* formulation outperformed the original *ED* for both databases. However, the difference in performance was more evident for *LPR Setup 2*. This result may be explained by noting that in the *LPR Setup 2* there were two major sources (coming from two major freeways) of traffic flow feeding the second station. Thus, in this case the arrival rate (λ) of vehicles detected at the second station not detected at the first one was much higher. This variable, as well as the proportion (α) of vehicles detected at first station that traveled to the destination, directly affected the probability of a false match. Increasing the λ value increases the occurrence of false matches. Therefore, it was more likely to find a false match during the operation of the second LPR setup.

In the second LPR setup, the *ED* formulation generated more failures in properly classifying pair-wise matches. Therefore, it seems that the *GED* measure is a more robust measure of similarity in the sense that it is more capable of determining if any given match (x_i, y_j) is true or false.

At this point in the study estimates of the association matrix were computed using two matrices of character interpretation occurrences (Subsection 4.2.3), one per machine. However, if the LPR units work differently the use of only one occurrence matrix may deteriorate the performance of the matching procedure. Such an analysis is important because in the case of a system with many LPR units it would be very costly to collect


(a) 2007 data - Performance of the FTWC Method

(b) 2010 data - Performance of the FTWC Method

Figure 5-9 Aggregated Performance of the FTWC Method

samples from all LPR units.

Although an extensive analysis requires more data, Figure 5-10 illustrates what happened when the association matrix for *LPR Setup 2* is estimated using only one occurrence matrix, from either station *g* or *h*. As can be seen the performance of the FTWC deteriorated. However, the decrease in performance was not enough to reach the worst level of performance for the original *ED*.

5.4.2 Parameter Calibration of the VTWC method

As stated before, two parameters of the VTWC method should be calibrated. These corresponded to the two limit bounds for the *GED* value, which are used to delineate the range of the varying time window constraint. In order to obtain an estimate of these two parameters some preliminary analyses of the calibration data were performed.

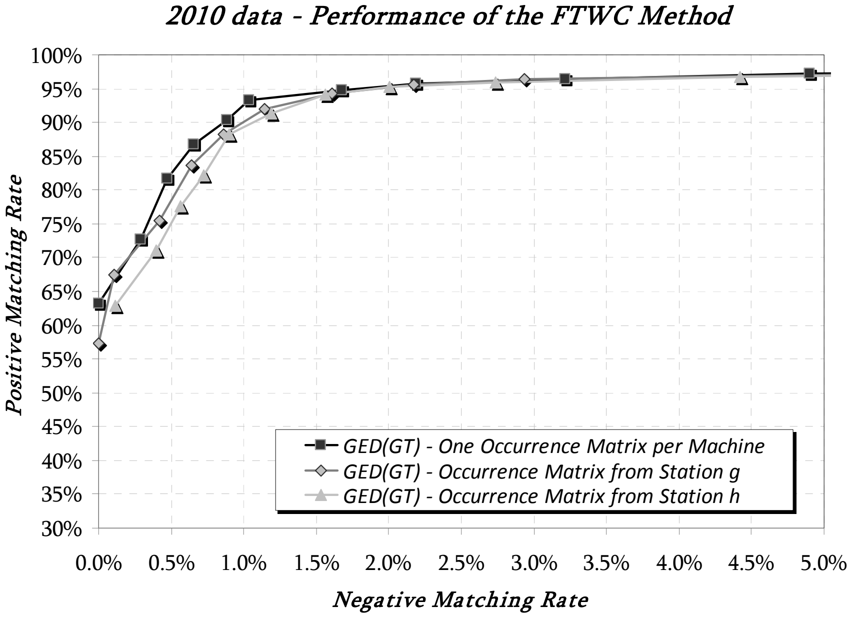
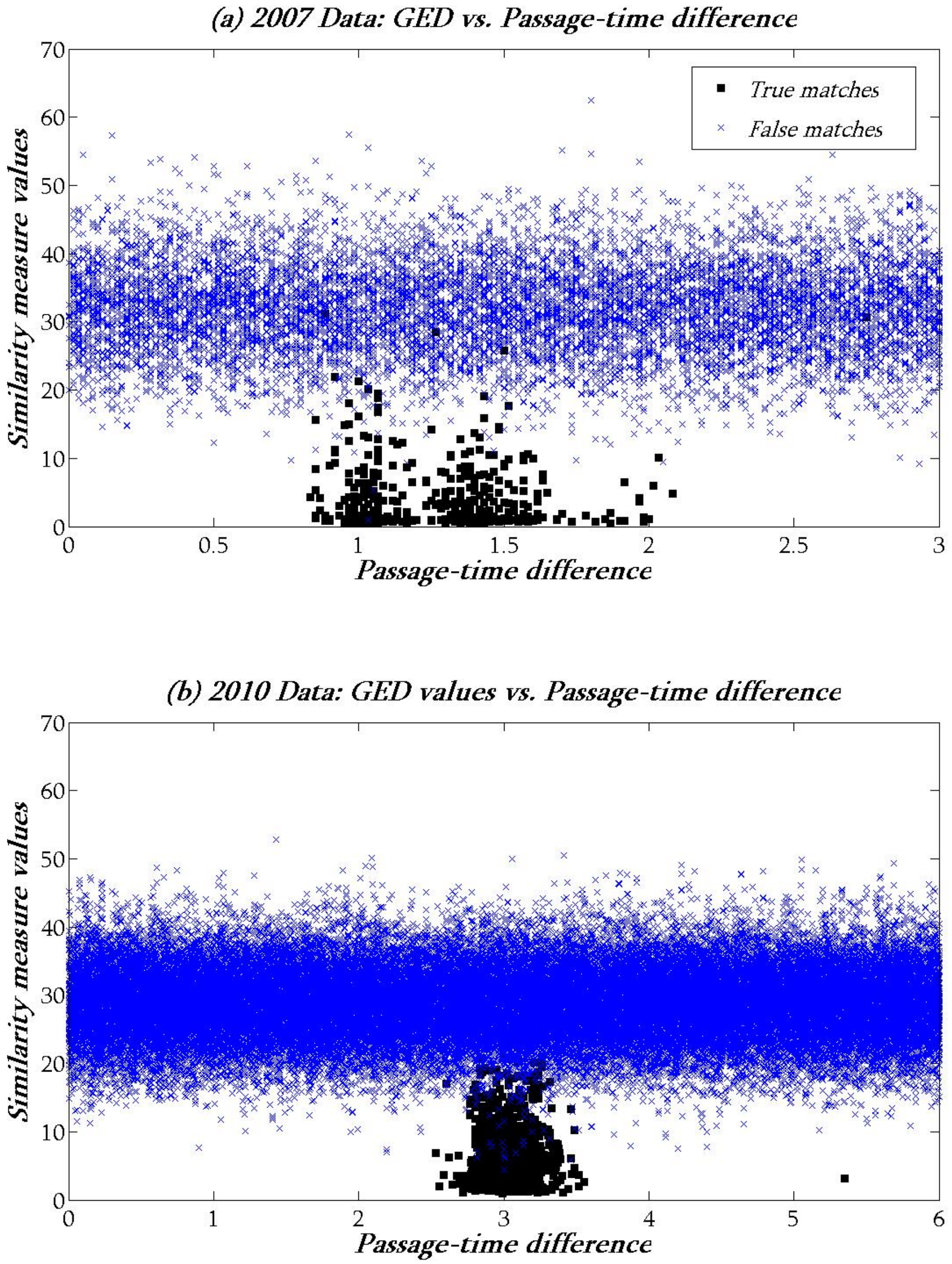


Figure 5-10 Aggregated Performance of the FTWC Method for Different Estimates of the Occurrence Matrices

At first, graphical analyses were performed to assess the variation of the *GED* values for all possible matching combinations. Figure 5-11 shows the scattering plots of the *GED* values, calculated for all possible combinations in both databases of exact matches and false matches, versus the passage time differences. Figure 5-11a shows the results of three days of data from *LPR Setup 1*, whereas Figure 5-11b presents one complete day of data from *LPR Setup 2*. As expected, the *GED* values for true matches are clustered around the mean and presented lower magnitude, whereas for false matches they were randomly distributed over time with higher magnitude.

A view of the distribution of the *GED* variable for true and false matches, as presented in Figure 5-12, demonstrates that the distribution shape for false matches is highly concentrated to the left and more spread out to the right (approximately resembling an exponential density function) whereas for true matches it is symmetric (resembling a normal distribution). This result reveals how the similarity measure is performing in classifying true versus false matches. The larger the overlapping area (intersection area) under both curves, the less the ability of the similarity measure to classify each match.

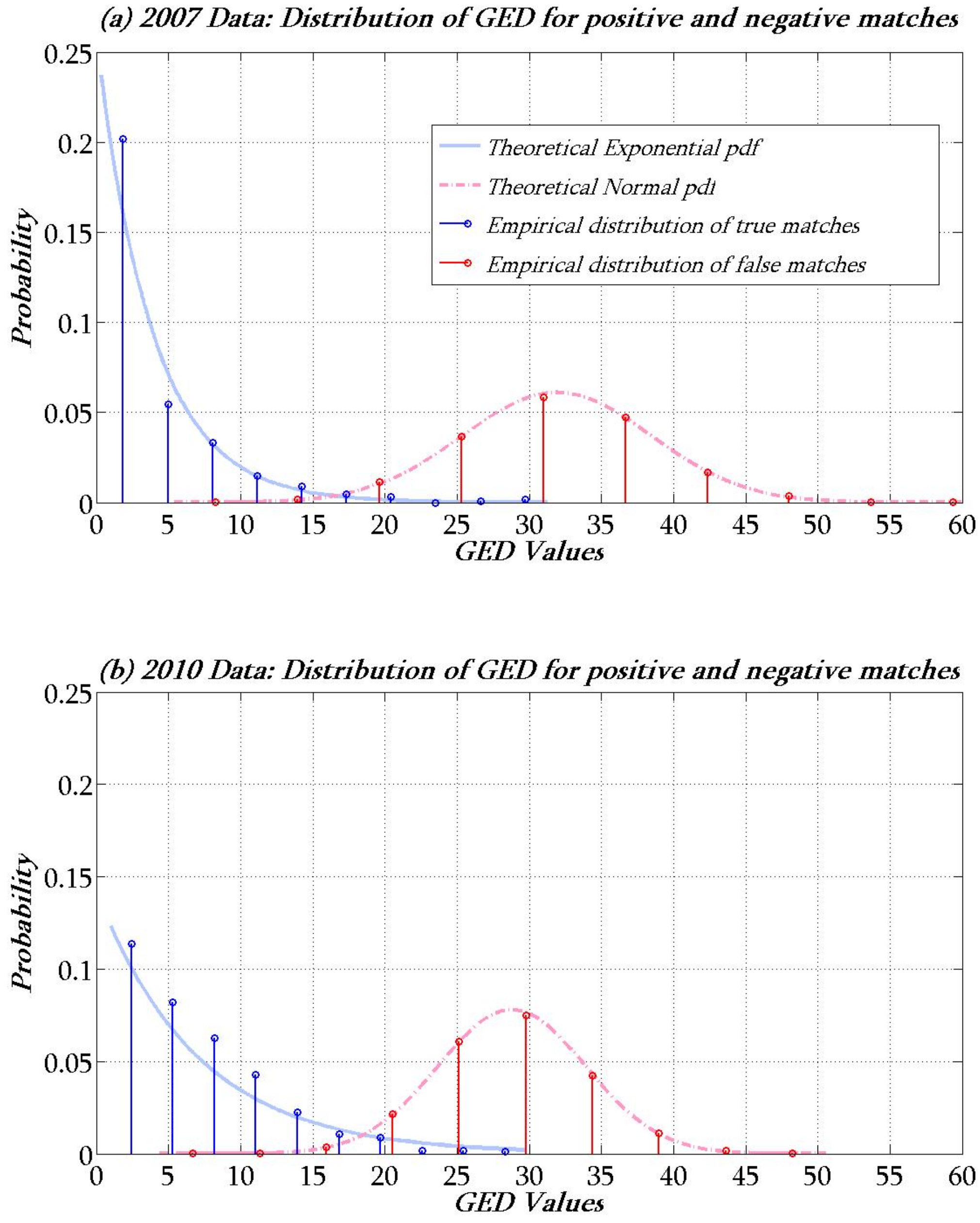
Analyzing the distributions of *GED* for the two databases, it can be seen that the overlapping area is greater for *LPR Setup 2*. This may be a consequence of the worse LPR accuracy for the second setup. Therefore, the model parameters τ^* and τ^{max} should have different values for the two LPR data. In other words, the width of the range (τ^*, τ^{max}] is expected to be shorter for *LPR Setup 2*.



(a) 2007 Data: GED vs. Passage-time difference

(b) 2010 Data: GED values vs. Passage-time difference

Figure 5-11 GED Values for both Ground Truth and False Matches versus Passage Time Difference


(a) 2007 Data: Distribution of GED for positive and negative matches

(b) 2010 Data: Distribution of GED for positive and negative matches

Figure 5-12 GED Distribution for True and False Matches

As mentioned before, the parameter τ^* should be a value below which is very unlikely to have a false match. The parameter τ^{max}, on the other hand, is defined as a value above which is quite unlikely to find a true match, and below which there is still a low likelihood of a false match. After some preliminary experiments and based on the graphical analysis presented, I came up with the following empirical values: $\tau^* = 5$ and $\tau^{max} = 20$ for *LPR Setup 1*, and $\tau^* = 5$ and $\tau^{max} = 16.5$ for *LPR Setup 2*.

Observing the sample data, the proportion of false matches with *GED* lower than 5 was 0.0013% for *LPR Setup 1*, and 0.0002% for *LPR Setup 2*. Whereas, with regard to the parameter τ^{max}, the observed proportion of true matches with *GED* above it was 1.4% and 3.3%, and the proportion of false matches with *GED* lower than it was 3.84% and 0.69%, for *LPR Setups 1* and *2*, respectively.

Notice that the proportion of false matches with *GED* lower than τ^{max} was much lower for *LPR Setup 2*. This result does not mean that the absolute number of false matches for the second setup is also much lower. Remind that, as mentioned before, the number of false matches is affected by the proportion α and by the arrival rate λ, and the likelihood of a false match increases when α decreases and λ increases. Since λ was much higher for *LPR Setup 2* one may still expect more false matches in this case.

5.4.3 Varying Time Window Constraint Application and Performance

In this section the performance of the proposed VTWC matching procedure was assessed. This procedure was combined with the most suitable similarity measure chosen before, calibrated and evaluated using the 10 sets of data combinations from each database.

Regarding to the parameters of the procedure, the τ* and the τ^{max} have been already determined in previous Subsection 5.4.2. The initial time window constraint was defined assuming an upper and lower bounds of the vehicle speed of 90 mph and 35 mph, respectively. Taking into account that no historical data was available, the sampling moving averages and standard deviations of the journey time were calculated from the nearest 10 previous matches classified as genuine.

The number of standard deviations given by the function $z(d_{ij})$, which establishes the time varying window constraint acting over the domain $(\tau^*,\tau^{max}]$, was defined by a quadratic function as in Equation 5.3 below:

$$z(d_{ij}) = \sqrt{9 \times \frac{\tau^{\max} - d_{ij}}{\tau^{\max} - \tau *}} \tag{5.3}$$

The function defined by Equation 5.3 varies from z = 3 (largest time window), where $d_{ij} = \tau^*$, to z = 0 (the time window vanishes), where $d_{ij} = \tau^{max}$.

The VTWC method was applied to all 10 dataset combinations used for validation from the two LPR databases. The performance of the *GED* similarity measure with the weights proposed in Section 4.2 of Chapter 4 was then assessed. The original *ED* with 0 and 1 weights was used as base scenario for comparison. To this end, it was assumed that *ED* could vary from 0 to 5. Table 5-3 shows the performance results for the 2007 database (*LPR Setup 1*), whereas Table 5-4 presents the results for the 2010 database (*LPR Setup 2*).

Table 5-3 2007 Data: Performance of *ED* and *GED* combined with VTWC

Validation Data - Combination	*Expected Number of Matches*	*Number of Positive Matches*	*Number of False Positive Matches*	*Number of Exact Matches*	*Percentage of Exact Matches (%)*	*Percentage of False Positive Matches (%)*	*Percentage Positive Matches (%)*
			(a) ED with 0 or 1 Weights				
1	315	296	8	187	59.4%	2.6%	94.0%
2	366	343	9	231	63.1%	2.6%	93.7%
3	417	389	11	244	58.5%	2.8%	93.3%
4	272	257	5	171	62.9%	1.9%	94.5%
5	323	303	7	184	57.0%	2.3%	93.8%
6	374	350	8	228	61.0%	2.2%	93.6%
7	228	216	6	143	62.7%	2.7%	94.7%
8	279	262	8	156	55.9%	3.0%	93.9%
9	330	309	9	200	60.6%	2.8%	93.6%
10	236	223	5	140	59.3%	2.2%	94.5%
			(b) GED with Association Matrix Estimated by GT Method				
1	315	309	3	187	59.4%	0.95%	98.1%
2	366	346	3	231	63.1%	0.82%	94.5%
3	417	395	4	244	58.5%	0.96%	94.7%
4	272	258	0	171	62.9%	0.00%	94.9%
5	323	307	2	184	57.0%	0.62%	95.0%
6	374	346	3	228	61.0%	0.80%	92.5%
7	228	223	1	143	62.7%	0.44%	97.8%
8	279	272	2	156	55.9%	0.72%	97.5%
9	330	313	3	200	60.6%	0.91%	94.8%
10	236	225	0	140	59.3%	0.00%	95.3%

Table 5-4 2010 Data: Performance of *ED* and *GED* combined with VTWC

Validation Data - Combination	*Expected Number of Matches*	*Number of Positive Matches*	*Number of False Positive Matches*	*Number of Exact Matches*	*Percentage of Exact Matches (%)*	*Percentage of False Positive Matches (%)*	*Percentage Positive Matches (%)*
(a) ED with 0 or 1 Weights							
1	1131	1021	78	407	36.0%	7.1%	90.3%
2	1125	1018	84	384	34.1%	7.6%	90.5%
3	1038	925	64	379	36.5%	6.5%	89.1%
4	1232	1123	88	444	36.0%	7.3%	91.2%
5	1145	1030	68	439	38.3%	6.2%	90.0%
6	1139	1027	74	416	36.5%	6.7%	90.2%
7	1185	1095	76	415	35.0%	6.5%	92.4%
8	1098	1002	56	410	37.3%	5.3%	91.3%
9	1092	999	62	387	35.4%	5.8%	91.5%
10	1199	1104	66	447	37.3%	5.6%	92.1%
(b) GED with Association Matrix Estimated by GT Method							
1	1131	1089	22	405	35.8%	2.0%	96.3%
2	1125	1083	27	381	33.9%	2.4%	96.3%
3	1038	990	23	376	36.2%	2.3%	95.4%
4	1232	1192	25	443	36.0%	2.1%	96.8%
5	1145	1101	19	438	38.3%	1.7%	96.2%
6	1139	1090	27	414	36.3%	2.4%	95.7%
7	1185	1142	25	414	34.9%	2.1%	96.4%
8	1098	1048	17	409	37.2%	1.6%	95.4%
9	1092	1043	24	385	35.3%	2.2%	95.5%
10	1199	1150	20	447	37.3%	1.7%	95.9%

Analyzing the results, it seems that the performance of the VTWC method was slightly better than the performance of the FTWC method. Perhaps, this result is due to a better adjustment of the VTWC to the variation in travel time. However, there is no enough empirical evidence to support this assertion, and a more detailed study should be performed to confirm this hypothesis.

Notice that, as happened with FTWC method, the *GED* method outperformed the original *ED*. However, as before, the performance was more evident for 2010 database, for the same reasons already explained for the FTWC method.

5.5 CHAPTER CONCLUSIONS AND RECOMMENDATIONS

In this chapter the passage time information has been incorporated into the matching framework proposed in Chapter 4. As seen in the literature, passage time information has been used before to predict the number of spurious matches from partial plate surveys, and to clean up matched LPR data. In this research it has been combined to the proposed similarity measure to decide towards a genuine or false match.

Two simple procedures were proposed to take advantage of the passage time information. The first was a simple time restriction over the passage time difference that constraints the number of outcomes selected from an upstream station dataset to match any given outcome of a downstream station dataset. In the second one, the time window constraint was a function of the similarity measure magnitude and of how close the difference in passage time is to the expected vehicle journey time between stations.

As expected, the results show that merely including the passage time information in the matching procedure considerably increased its performance. Although the matching rate was significantly increased when using the travel time information, further work is needed to validate this procedure for other situations of traffic conditions. The procedure was applied during a period and/or roadway with slight traffic variation resulting in small dispersion of travel times, what might have contributed to this good performance. In addition, the stations were set up relatively closed to each other, so thus there was no major source of traffic disturbance to disperse the travel times.

The second procedure (VTWC) had a slight better performance than the first one (FTWC). The VTWC framework is expected to adjust better to the variation in travel time, thus providing better performance results. However, the difference in performance was moderate due to two reasons. First, the lack of historical data prevented to estimate adequately the journey time parameters. Instead, sample blocks containing the earliest 10 observations were adopted. As such, the use of portions or time blocks of the data period might ignore the temporal variation at the journey time. Second, the moderate variation of the journey times for the period or road analyzed may have contributed to the similar performance observed when comparing the two methods.

Regarding to the measure of similarity used, the *GED* with weights calculated from the LPR machine errors outperformed the original *ED* with 0 or 1 weights. Furthermore, the difference in performance was more apparent for the second LPR setup. It was point out that the probably reason for this latter result was that the two databases had different likelihoods of finding a false match, mainly due to the much higher arrival rate, in the second LPR setup, of vehicles detected at the second station not detected at the first one. This results in a higher possibility of having a false match. Therefore, it seems that *ED* with the proposed weight function is more robust to variation in traffic conditions.

Further studies are still needed to assess different traffic conditions and LPR setups, as well as machine accuracies, not yet observed in this research and likely to exist in real world. Such sensitive analyses can be performed resorting to computational simulation. Several scenarios can be created, with different journey time profiles, machine installations and accuracies, thus allowing assessing the robustness of the proposed matching procedure under several situations. Moreover, the impact of using a single occurrence matrix to estimate the association matrix can be assessed in more details. All these analyses are out of this research scope and were left out for future studies.

CHAPTER 6

LEARNING PROCESS AND SAMPLE SIZE TO ESTIMATE THE CONDITIONAL PROBABILITIES

In chapter 4 the association matrix **C** containing the conditional probabilities of character association occurrences was estimated resorting to the availability of ground truth values for the reading outcomes. Two probability matrices, $\mathbf{C^h}$ and $\mathbf{R^g}$, for Station *g* and *h*, were defined to obtain an estimate of **C**. The proposed method to estimate these two matrices was very time-consuming as it required to manually indentifying the ground truth values from a sample of LPR data. As discussed before, **C** is depended of the LPR accuracy, which is affected by many factors related to the LPR technology and the installation configuration. It is expected that a bad estimate of **C** will deteriorate the matching procedure performance. Therefore, if there are many sources of variation, noise, affecting the LPR operation, it would be necessary a large amount of data to achieve a required error precision. In such case, the estimation would require also more human intervention. So far the sample size of plates or number of characters in which each LPR machine were exposed in order to obtain the association matrix was defined arbitrary. Fortunately, it seems that the sample size used in chapter 4 was sufficient to give a good matching performance. However, it is still unknown what amount of data should be collected to obtain a precise estimate of the association matrix. In this chapter, it is firstly discussed how to estimate the sample size (number of training characters) to obtain a statistically significant estimate of the association matrix **C**. Then, a second approach to estimate the association matrix, which consists in a learning process without human intervention, is proposed. Finally, the convergence of the proposed methods to estimate **C** and the impact of sequentially estimates of **C** on the matching procedures are analyzed.

6.1 SAMPLE SIZE FOR THE ASSOCIATION MATRIX

In this section, a theoretical approach on the problem of determining the sample size to estimate a significant association matrix is discussed. The sample size is defined in terms of the number of characters that the LPR machines should be exposed in order to have a good estimate. For simplicity, in the following derivations, it is assumed that all vehicle plates are detected at both stations, and hence the sample size of characters is approximately equal for both observation points. In other words, the arrival rate of license plates detected at second station not detected at the first one is zero.

The process of estimating **C** involves estimation of multinomial random variables. Notice that the entries of each row i of the matrix **C** can be seen as parameters $p_{i1}, p_{i2}, \ldots, p_{ik}$ of a multinomial random variable. Therefore, if n_i identical and independent multinomial trials are obtained with parameters given by the probabilities at row i of **C**, the corresponding final outcomes are multinomial random variables with parameters $n_i, p_{i1}, p_{i2}, \ldots, p_{ik}$.

The variable n_i is the expected number of times the truth character outcome r_i is observed at Station g, with the total number $\sum_i n_i$ of characters denoted by N. Notice also that the variable n_i is another outcome from a multinomial random variable, which is related to the possible reading outcomes at Station *g* for a sample of *N* trials of ground truth characters being recognized at both stations. Hence, for a character outcome x_i at first station, the corresponding expected value of n_i is given by:

$$n_i = Np(x_i) = N\sum_j p(x_i \mid z_j)p(z_j) \tag{6.1}$$

Where,

z_j is the j^{th} true character from a list of possible alphanumeric (plus the empty character) characters.

The main interest here is in estimating a minimum number of trials in order to obtain a precise estimate of the matrix **C**. I separated this problem into two steps. First, what should the sample size (n_i trials) be to obtain a good estimate of the *k* probabilities in any row category *i* of **C**? Furthermore, what should the total number of trials *N* be in order to guarantee that at least n_i trials will be observed for category *i* during the experiment? In the next three subsections, statistical solutions are discussed to deal with these two questions and a simple simulation procedure is described to estimate the sample size *N*.

6.1.1 Statistical Approach to Obtain the Sample Size of a Multinomial Variable

In past studies, solutions based on the calculation of large sample size for multinomial frequencies were proposed to this problem. Angers (1984) offers a solution (based on the Bonferroni inequality) to determine the smallest sample size required to estimate simultaneously *k* multinomial parameters with a set of *k* symmetrical confidence intervals with given simultaneous significance level α . As stated by Angers, the Bonferroni inequality ensures that the set of intervals has a probability of at least *1-* α of simultaneously containing the true *k* parameters, where α is the sum of the significance levels of the *k* individual intervals.

By applying the Bonferroni inequality and the central limit theorem, it has been shown that for large random sample size *n* from a multinomial distribution with unknown parameters θ_1, θ_2, …,θ_k, where $\sum_{i=1}^{k}\theta_i = 1$ a set of symmetrical confidence intervals can be calculated by

$$\left|\hat{\theta}_i - \theta_i\right| \leq Z_{\alpha_i/2}\sqrt{\frac{\theta_i(1-\theta_i)}{n}}; i = 1,2,\ldots, k \tag{6.2}$$

where $\hat{\theta}_i$ is the observed proportion of observations falling in the i^{th} category; $Z_{\alpha_i/2}$ is the $(1-\alpha_i)\times 100^{th}$ percentile of the standard normal distribution; α_i is the significance level of the i^{th} interval; and $\sum_{i=1}^{k}\alpha_i$ is the simultaneous significance level.

An estimate of the required sample size can be obtained by solving Equation 6.2 for *n*. Then, if the half width and significance level of each interval is specified by d_i andα_i, with the required simultaneous level calculated by $\sum_{i=1}^{k}\alpha_i$, the sample size is given by the smallest integer greater than or equal to:

$$n = \max\left\{Z_{\alpha_i/2}^2 \frac{\theta_i(1-\theta_i)}{d_i^2}; i = 1,2,\ldots, k\right\} \tag{6.3}$$

In practice, the θ_i values are unknown. As stated by Angers (1984), one way of overcoming this difficult is applying a two-stage procedure as proposed by Mamrak and Amer (1980) for estimating proportion:

(1) Collect an initial sample of size n_o and obtain the initial estimates $\hat{\theta}_i, i = 1,2,\dots,k$;

(2) From the initial estimates calculate the sample size using Equation 6.3. If $n > n_0$, collect an additional sample of size n-n_0 and pool the two samples.

Assuming that the only restriction specified about the alphas is that $\sum_{i=1}^{k} \alpha_i = \alpha$, the required sample size can be actually reduced (Angers, 1984). With this assumption, the minimum sample size is given by solving

$$n = \min_{(\alpha_1,\dots,\alpha_k)} \left\{ \max\left\{ Z^2_{\alpha_i/2} \frac{\theta_i(1-\theta_i)}{d_i^2} ; i = 1,2,\dots,k \right\} \right\}$$

Such that,

$$\alpha_i > 0; \qquad i = 1,2,\dots,k$$

$$\sum_{i=1}^{k} \alpha_i = \alpha\,, \;\; 0 < \alpha < 1$$

According to Angers (1984) a simple computer programming can be then written to solve the above optimization problem. Such programming can be based on the following steps:

(1) Select an initial sample of size n, and obtain the estimates:

$\hat{\alpha}_i = 2\left[1 - \Phi\left(\frac{d_i\sqrt{n}}{\sqrt{\theta_i(1-\theta_i)}}\right)\right]$; *i = 1, ... , k*. Where $\Phi(.)$ is the cumulative density function of the standard normal distribution;

(2) If $\alpha < \sum_{i=1}^{k} \hat{\alpha}_i$, *n* must be increased. Repeat steps 1 and 2 until $\alpha \geq \sum_{i=1}^{k} \hat{\alpha}_i$;

As mentioned before, the multinomial parameters are not known in advance. A worst case scenario would be to test all possible parameter vectors, as discussed in Thompson (1987), and estimate the largest sample size. However, this alternative is computationally tedious and might lead to an unnecessary large sample size. Thus, a less costly method, would be to apply the sequential sampling procedure, as described by Mavridis and Aiyken (2009), where the algorithm is initiated with small sample size, the sample is incremented gradually with new certain amount of sampling units, and new parameter vectors are estimated until $\alpha \geq \sum_{i=1}^{k} \hat{\alpha}_i$.

Regarding to the present study, the main interest is to estimate the proportions on row *i*, row-vector $\mathbf{C_i}$, of the conditional probability matrix **C** with a given marginal error vector of $\mathbf{d_i}$ = [d_{i1}, d_{i2}, ..., d_{ij}, ..., d_{ik}] at a pooled significance level $\alpha_i = \sum_{j=1}^{k} \alpha_{ij}$. In other words, determine a sample size that ensures that the set of confidence intervals of the conditional probabilities given by p_{ij} ± d_{ij}, where *j = 1,2, ..., k*, for the *i^th^* row of **C** will meet, for large n_i, the required simultaneous significance level α_i, as expressed by

$$\Pr\left\{\bigcap_{j=1}^{k} \left|\hat{p}_{ij} - p_{ij}\right| \leq d_{ij}\right\} \geq 1 - \alpha_i$$

It was assumed that the only restriction specified about the alphas was that they should be equal to a required pooled significance level for each row. Therefore, the sequential sampling procedure described by Mavridis and Aiyken (2009) can be applied to find a minimum sample size n_i for each category i of matrix **C**.

6.1.2 Sample Size to Observe at Least n_i for Each Category i

Having the required sample size for each category in hand, the next step is to find a minimum total sample size N that ensures the set of required sample sizes n_i, where $i = 1, 2, \ldots, k$, will be observed with a given confidence level. This means to find a minimum N that ensures a required simultaneous confidence level β of having at least n_i observations for each category i. Let N_i denote the number of trials at category i. Then, using Bonferroni inequality, I must have that

$$\Pr\left\{\bigcap_{j=1}^{k} N_i \geq n_i\right\} \geq \beta$$

Remind that each outcome n_i should be an observation of a binomial random variable with parameters N and $p(x_i)$, where x_i is a character outcome at Station g. Thus N can be found by solving the following problem:

$N = argmin\{n\}$

Such that,

$1-\sum_{u=1}^{n_i}\binom{n}{u}p(x_i)^u[1-p(x_i)]^{n-u} \geq \beta$; for all $i = 1,2,\ldots,k$

$0 < \beta < 1$

A computer programming can be written to solve the problem above. The algorithm can be initiated with $n = \max\{n_i / p(x_i)\}$ and terminated when the required confidence level is attained.

6.1.3 Sample Size by a Simulation Approach

An alternative procedure to the method of sequential sampling above, it is the use simulation. In this case, a sequential sampling procedure is still used, but now with the estimation of the significance levels at any time the sample is increased. The procedure can be summarized as follows:

(1) Use the LPR machines to collect an initial sample of plates, and estimate the character misreading matrices $\mathbf{C^g}$, $\mathbf{C^h}$ and the likelihood of truth character occurrences, vector $\mathbf{p} = [p_1, p_2, \ldots, p_k\ ]^T$;

(2) Generate alphanumeric characters, and emulate the process of recognizing characters by the dual LPR setup. To this end, randomly generate multiple samples of *n* characters using multinomial distribution with parameters given by vector **p**. Then, based on the expected machine operation, given by the estimates of $\mathbf{C^g}$ and $\mathbf{C^h}$, estimate for each sample what would be the resulting outcomes at each station for each of the *n* characters;

(3) Calculate the association matrices for all samples of size *n*, and compute the corresponding significance levels $\hat{\alpha}_i$ of each row *i* of the association matrix, for a given marginal error vector $\mathbf{d_i}$;

(4) If $\alpha_i \geq \hat{\alpha}_i$, for all *i = 1,2,...k,* stop the process. Otherwise, gradually increase the sample size by a certain amount of sampling units and repeat steps 1 to 3 until $\alpha_i \geq \sum_{j=1}^{k} \hat{\alpha}_{ij}$, for all *i*, where α_i is the required simultaneous significance level of row *i*;

A disadvantage of the proposed procedure above is that it depends on the estimation of the matrices $\mathbf{C^g}$ and $\mathbf{C^h}$ which requires the manual determination of the ground truth characters. Whereas the statistical procedures described earlier are not restricted to this condition.

6.2 PROPOSED LEARNING PROCESS (LP)

This section presents a method to estimate the conditional probabilities *p(y | x)* from a sample of genuine pair-wise matches. Suppose that it is possible to obtain a set of genuine matched plates from a period of operation of a dual LPR setup, such as a cross table, matrix, associating the character readings is derived. Let us denote this matrix as **F**, as follows:

$$\mathbf{F} = \begin{bmatrix} f(x_1, y_1) & f(x_1, y_2) & \dots & f(x_1, y_k) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ f(x_k, y_1) & f(x_k, y_2) & \dots & f(x_k, y_k) \end{bmatrix}$$

where each element $f(x_i, y_j)$ represents the absolute frequency of the corresponding pair-wise character association (x_i , y_j).

The conditional probability matrix, for all pair-wise character associations, can be estimated directly from the frequency matrix **F**, as follows:

$$\hat{\mathbf{C}} = \begin{bmatrix} \frac{f(x_1,y_1)}{\sum_{j=1}^{k} f(x_1,y_j)} & \frac{f(x_1,y_2)}{\sum_{j=1}^{k} f(x_1,y_j)} & \cdots & \frac{f(x_1,y_k)}{\sum_{j=1}^{k} f(x_1,y_j)} \\ . & . & . & . \\ . & . & . & . \\ \frac{f(x_k,y_1)}{\sum_{j=1}^{k} f(x_k,y_j)} & \frac{f(x_k,y_2)}{\sum_{j=1}^{k} f(x_k,y_j)} & \cdots & \frac{f(x_k,y_k)}{\sum_{j=1}^{k} f(x_k,y_j)} \end{bmatrix}$$

Thus, $\hat{C}_{ij} = \frac{f(x_i,y_j)}{\sum_{j=1}^{k} f(x_i,y_j)}$ is approximate for the probability $p(y_j \mid x_i)$. Observe that the probabilities in $\hat{\mathbf{C}}$ above are estimated from a set of plates captured at both stations. Therefore, they are estimates of conditional probabilities of the pair-wise character association with the restriction that the outcomes at both stations were not only originated from the same character, but also came from the same plate. Therefore, to obtain the estimated matrix $\hat{\mathbf{C}}$ I need to make sure the every dual outcome (x, y) came from the same plate, or at least it is very likely that they were originated from the same plate. To this end, I proposed an automated process to find such set of “genuine” matches which uses the FTWC method proposed in Chapter 5.

The proposed procedure to estimate all character association probabilities $p(y_j \mid x_i)$ consists in an iterative learning algorithm as described below:

(1) Apply the FTWC matching method (Chapter 5) to find a set of matches from a sample of LPR data with the initial settings:

a. Assume initially that **C** = **I**, where **I** is 37 x 37 identity matrix;

b. Use 0 or 1 assignments to calculate *ED*;

c. Set a threshold τ such that if $d(x, y) \leq \tau$ the string outcomes x and y may have come from the same truth string.

(2) From the set of matched plates obtained at step 1, tabulate the pair-wise character occurrences under a matrix format to obtain **F**, e.g. one cell of the matrix is for example the number of times a character "A", read at Station *g*, happens to match the character "4", read at Station *h*;

(3) Compute an updated approximation for the matrix **C**, where each cell will be the probability $\hat{C}_{ij} = \hat{p}(y_j \mid x_i)$;

(4) Test if the updated matrix at the current iteration is similar to that one from previous iteration. If so, stop; otherwise go back to step 1 using as new input the updated matrix estimated in step 3 and the appropriate threshold value. Thus, at step 1 the *ED* with 0 or 1 weights is replaced by *GED* with weights calculated using the updated matrix.

As seen before, the FTWC matching method described in Chapter 5 uses a fixed time window constraint on the passage time information to reduce the number of possible candidates at Station *g* to match a certain outcome at Station *h*. In other words, the potential matches are constrained to a shorter time window within the survey period. This actually increases the likelihood of finding a genuine match when applying the edit distance formulation, even with a poorer estimation of **C**. This way, the iterative procedure proposed is expected to reach a quite good approximation of **C**.

6.3 CONVERGENCE OF THE ASSOCIATION MATRIX

For the case of permanent LPR setup, where the units operate continuously, such as in speed enforcement, it is possible to estimate the association matrices sequentially by incrementing the data with additional data units, or small portions of data, until the process converges. Either the GT (Chapter 4) or the LP method proposed in Section 6.2 can be used for this purpose. The convergence criterion can be defined by a performance measure (sum of absolute differences or overall square root of the differences between the cell values) to compare the similarity between each two consecutive matrix estimates. The point of convergence is then reached when either additional data should not improve the estimates or when the convergence criterion is met. The chart of Figure 6-1 shows an example of an expected curve of convergence in the process of estimating **C**.

Regarding to the LP method the number characters, or sample size, used for each sequential estimate is equal to the sum of all cell values of each sequential matrix **F**. Thus, the estimation of **C** requires that both LPR machines operate simultaneously. Whereas for the GT procedure this requirement is not necessary, since the interpretation matrices can be obtained by observing the operation of each machine independently. Thus, for GT method the total number of characters, or sample size, was defined as the minimum number of character outcomes observed at each station.

As more portions of data is used to estimate the association matrix both methods should converge to the matrix **C** plus an error matrix **E** due to the noise involved in the estimation procedure, as represented by the following limit expression:

$$\underset{t\to\infty}{\hat{\mathbf{C}}_{\mathbf{GT}}} \to \mathbf{C} + \mathbf{E}_{\mathbf{GT}}$$

$$\underset{t\to\infty}{\hat{\mathbf{C}}_{\mathbf{LP}}} \to \mathbf{C} + \mathbf{E}_{\mathbf{LP}}$$

where *t* is time unit for the incremental amount of data used, which can represent number of hours, days, or weeks. GT stands for Ground Truth and LP for Learning Process. The matrix noise $\mathbf{E_{GT}}$ is due to error involved in the manual identification of ground truth plates. Whereas the matrix $\mathbf{E_{LP}}$ is due to the false matching rate in the FTWC method.

Observe that the required amount of data can differ between the two methods since, as mentioned before, the LP method requires a set of genuine matches from the dual LPR setup while for the GT this requirement is relaxed.

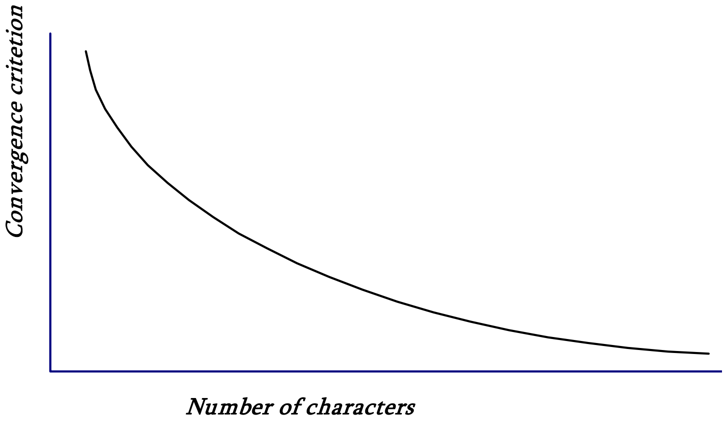


Figure 6-1 Example of Convergence to a Limit Matrix

6.4 APPLICATION TO REAL DATA

In this section, the *LPR Setup 2* (2010 data) was used to assess the sample size and convergence of the association matrix. First, the procedures described in Section 6.1 were applied to find the sample size with different significance levels and precisions. Then, the convergence of the matrix **C** was assessed. As described in Chapter 5, the *LPR Setup 2* has been operating continuously and provides sufficient data, more than one moth of data, to estimate a significant matrix and also to assess the impact of sequentially estimates on the matching framework.

6.4.1 Association Matrix Estimation

The association matrix for *LPR Setup 2* was estimated using the two different procedures described earlier. Five complete days of operation in 2010, April 6th and 7th, and May 25th, 26th and 27th, were selected to collect the ground truth plates and apply the GT procedure. The learning process (LP) method was applied to estimate another matrix from 39 days of operation in 2010, from April 8th to May 19th. Both matrices are shown in the Appendix A.

During the selected five days for estimation of **C** by the GT method, a total of 10450 and 21266 plates were captured at stations *g* and *h*, respectively. This amount of plates resulted in a total of 69475 and 141668 characters for estimation of the probability matrices $\mathbf{R^g}$ and $\mathbf{C^h}$, respectively. All ground truth values of the plate numbers were verified and recorded using the Excel spreadsheet described in Chapter 5. The process of recording the plate values from image views took on average 15 seconds per plate, resulting in a total of 43 hours and 88 hours of manpower, for stations *g* and *h*, respectively. This was a very exhaustive procedure and not free of mistakes.

Regarding to the LP method, after three interactions a total of 18152 pair-wise plates were classified as genuine by the FTWC method, which resulted in 120740 associated characters. The effort involved in the estimation of the association matrix by the LP method required only computational processing and the availability of enough data. The estimation error involved in this process corresponded to the number of misclassifications by the FTWC matching procedure, which depended on the *ED* threshold used.

Assuming that both matrices obtained were reliable estimates, the sample sizes (number of characters detected at both stations) needed to estimate them at a certain precision *e* and for different significance levels are shown in Figure 6-2.

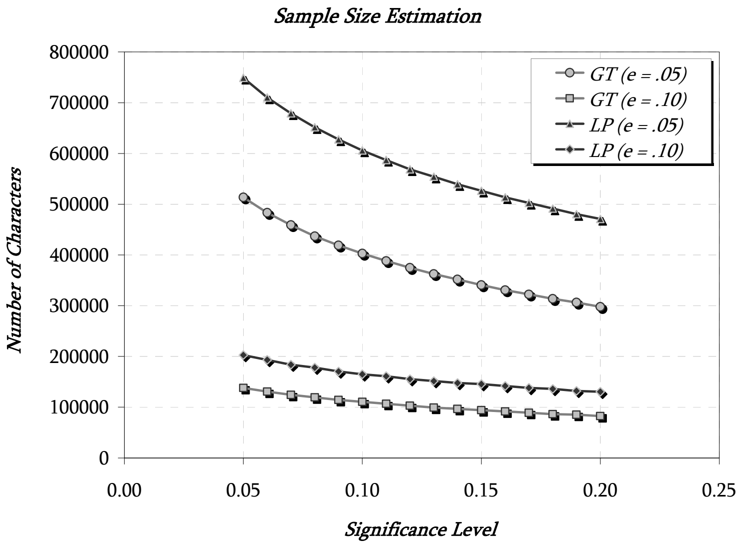


Figure 6-2 Sample Size for Estimation of the Association Matrix

As can be seen, a very accurate matrix would require a huge amount of data. However, after a carefully analysis of the estimated matrices I noticed that the cause of such huge sample was that the character outcome “Q” rarely happened (with chance of about 1:10000). Thus eliminating this outcome from consideration, I obtained an update of the sample size as shown in Figure 6-3.

According to Figure 6-3 a large amount of data is still required to obtain a significant estimate. For the matrix estimated using the GT method, the collected sample of 69475 characters would give a cell precision of 0.05 with significance level of less than 5%. Whereas for the matrix estimated by LP method the sample of 120740 characters would give a cell precision of 0.025 with significance level between 5% and 10%. Remind that the significance level is a simultaneous parameter implying that most cells had individual significance levels much less than the simultaneous level.

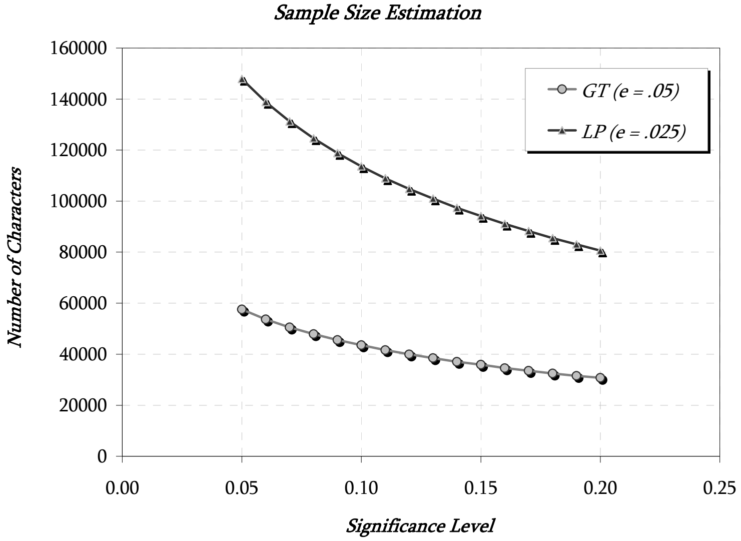


Figure 6-3 Sample Size for Estimation of the Association Matrix after Eliminating the Unlikely Outcome “Q”

As an illustration of the simulation procedure, I generated multiple samples of N = 69475 characters (corresponding to the minimum sample used to estimate the ground truth matrices) by a multinomial experiment with parameter vector equal to the estimated vector $\hat{\mathbf{p}}$ of character likelihood. With this, I simulated the possible outcomes from the LPR units using the estimated ground truth matrices $\hat{\mathbf{C}}^{\mathbf{g}}$ and $\hat{\mathbf{C}}^{\mathbf{h}}$. Assuming a cell precision of 0.05, the simulation process resulted in a simultaneous significance level of about 3%.

6.4.2 Association Matrix Convergence

In this section, the convergence of the association matrix is assessed. The hypothesis to test is that: as more data is added in the estimation procedure a point of convergence should be reached where the estimate can not further be improved.

Regarding to the GT method, five days of operation resulted in a convergence curve as shown in Figure 6-4 (adopting as convergence measure the overall squared root of the cell differences between every two consecutive estimates). Notice that the matrix estimate tended to converge with final convergence measure of 0.012.

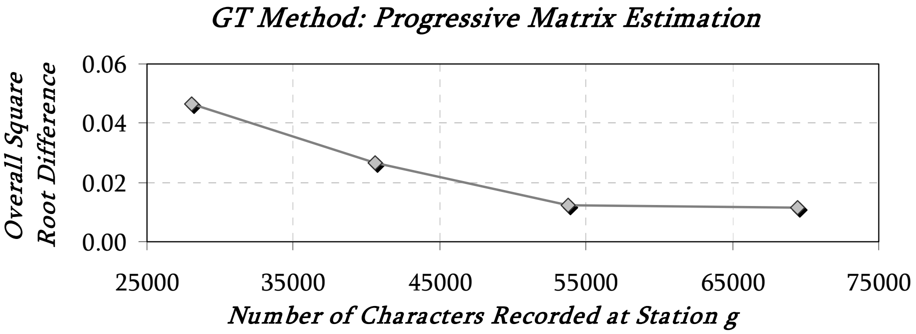


Figure 6-4 Association Matrix Convergence for GT Method

The curve of convergence for the LP method is presented in Figure 6-5. It seems that the method converged with sample size of around 60000 characters and with a final convergence measure less than 0.003.

In the chart of Figure 6-6, I compared the sequentially matrices estimated by the LP method with the final matrix estimated by GT method. This was to test the hypothesis that both estimation methods should converge to approximately the same association matrix **C**, but differing by error amount inherent to the estimation procedure. As can be seen in Figure 6-6, the matrix by LP method approached the final one by GT method as more data was added in the LP procedure. Therefore, when enough data is used it seems that the two procedures should provide final estimates close to each other, but still separated by an error matrix.

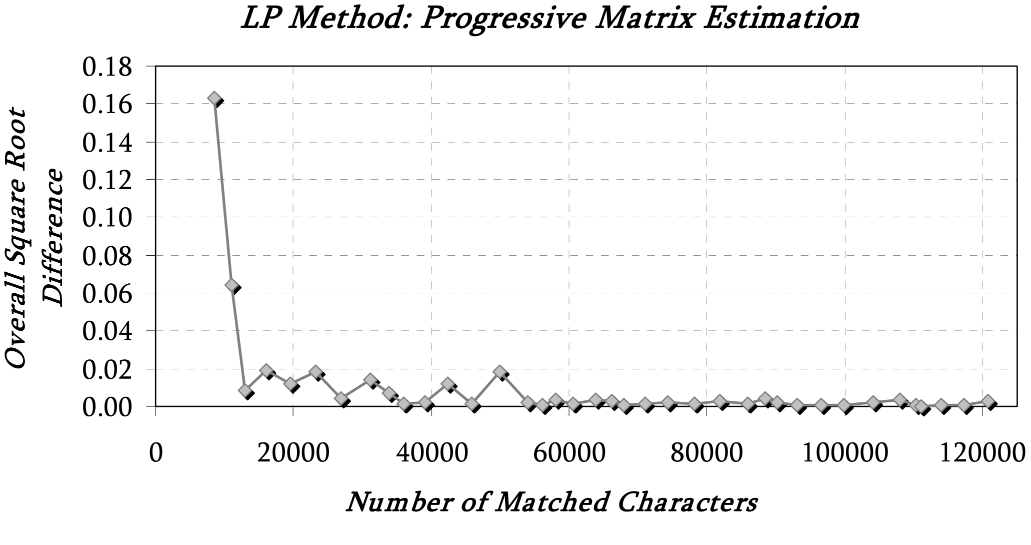


Figure 6-5 Association Matrix Convergence for LP Method

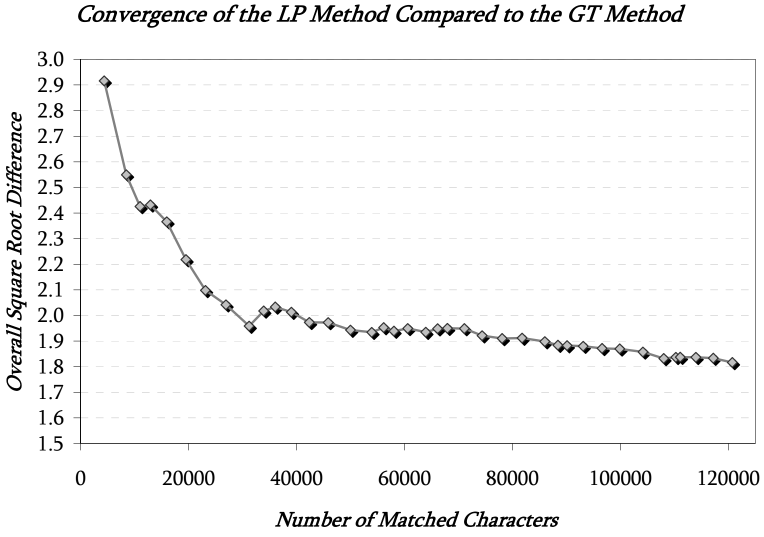
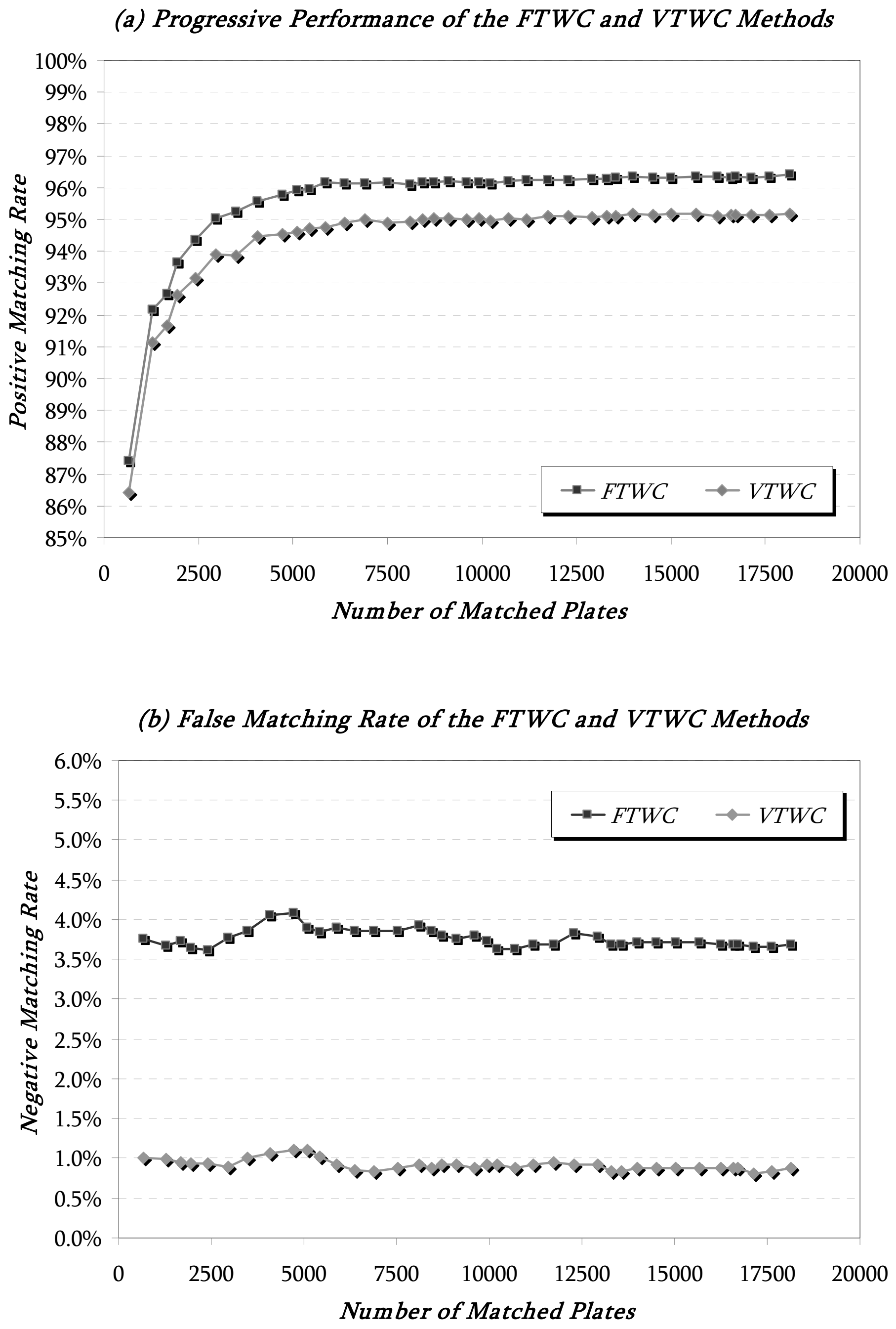


Figure 6-6 Comparison Between the Sequentially Estimates of C Using LP Method and the Final Estimate Using GT Method

6.4.3 Matching Results using Association Matrix Estimated by the Learning Process

The sequentially 39 estimates of **C** using the LP method were consecutively used in the *GED* formulation combined with the two matching procedures proposed in Chapter 5: the FTWC and the VTWC methods. Curves of performance related the positive matching rate, as well as the false positive matching rate, versus the corresponding number of characters used to estimate **C** are presented in Figure 6-7. As can be seen, as the sample size for estimation of **C** was incremented, the performance of the matching procedures considerably improved. For both methods, the positive matching rate highly increased while the false matching rate kept the same level. It is worth noting that the VTWC presented a better performance with much lower false matching rates.



(a) Progressive Performance of the FTWC and VTWC Methods

(b) False Matching Rate of the FTWC and VTWC Methods

Figure 6-7 Performance of the Matching Procedures for Sequentially Estimates of C using LP Method

6.5 CHAPTER CONCLUSIONS AND RECOMMENDATIONS

In this chapter two procedures were presented to estimate statistically significant sample sizes (number of training characters at which the LPR machines should be exposed) for the estimation of the association matrix **C**. Besides, a new method named learning process (LP) was proposed to estimate the association matrix **C**. The chapter ended with a convergence analysis of the estimation procedures and the effect of sequentially sampling on the matching procedures.

The main drawback of the GT method is that it requires determining an occurrence matrix for each individual LPR unit in the system. This may call for a large amount of data. Moreover, to extract the ground truth values from a LPR dataset requires many hours of painstaking work.

It is worth noting that the estimation procedure should be performed periodically to accompany any change of plate syntax due to changing in trip patterns or creation of new plate patterns, as well as due to the natural deterioration of LPR accuracy. Therefore, the task of estimating the association matrix can be more costly than expected.

One way of overcoming part of the problems with the GT method would be to determine a single occurrence matrix, from a single unit, and transfer the results to the remaining units in the system. This option may be feasible only if all LPR units in the system work similarly, with the same pattern recognition algorithm. The results of a preliminary analysis presented in Chapter 5 indicated that this alternative might be feasible.

Another way to overcome the burden of the GT method, as proposed in this chapter, is to use the LP method. The LP method requires only a set of pair-wise matches classified as genuine, which can be automatically determined by one of the proposed matching procedures described in Chapter 5. This can save precious time and money whenever the association matrix needs to be updated. Therefore, the LP method is a less costly and less time-consuming procedure and should be preferable.

The only drawback of the LP method is that when only a small proportion of vehicles travel through both stations several days of operation may be needed to observe an enough sample of genuine matches. This is not a limitation of the GT method since the association matrix is estimated by a matrix multiplication of two probability matrices which are independently estimated from each machine operation.

With respect to the required sample size to estimate **C**, the analysis results demonstrated that a large amount of data would be required to obtain a highly precise and accurate estimate of **C**, using either the GT or the LP method. Obviously, such sample size or amount of data depends on the cell precision and the simultaneous significance level used. On the other hand, from the convergence analysis, it seems that a good approximate would be reached with less data.

Regarding the convergence of the estimation procedures, GT and LP methods, the results showed first that the two methods tended to converge to a limit matrix. Mathematically both estimation methods should result in almost the same matrix estimation when the estimation error is small and all vehicles are detected at both stations. The results demonstrated that in both methods the estimates tended to approach to the same matrix, plus a small noise caused by the error in the estimation process (i.e. collecting error in the GT method and false matching rate in the LP method).

Regarding to the affect of sample size on the matching procedures, it has been demonstrated that a poor estimate of **C** can really deteriorate the discriminative power of the similarity measure used. In other words, as more data were added, the corresponding estimated matrix significantly increased the performance of the matching procedures.

In this study only one type of LPR technology was deployed. More work would be needed to evaluate different LPR accuracies. It is believed that the rate at which the estimation procedures converge to a limit matrix depends on the accuracy of the LPR technology. If more accurate equipment is used less data is needed and the estimation process should converge faster.

CHAPTER 7

CONCLUDING REMARKS

In this research I faced with the problem of matching readings from a dual License Plate Recognition (LPR) setup without any reference. A two-point setup has the objective of tracking vehicles passing trough two locations. To this end, the vehicle plate numbers, as well the time of passages, are captured at an upstream and a downstream station. However, since the equipment is not flawless, the reading outcomes stored at each station database are not accurate implying that only a portion of the data can be matched exactly. Therefore, the challenge was to indentify vehicles passing through both stations from inaccurate pair of readings. Taking into account that the outcomes generated by LPR systems are formed by sequence of characters (strings) I sought to solve this problem by applying a procedure to measure the proximity between two strings, emulating how a human would speculate that any two strings are close to each other.

7.1 MEASURE OF SIMILARITY BETWEEN STRINGS

Searching in the specialized literature, there is a technique to measure the similarity between strings named Edit Distance (*ED*), that in its original formulation measure how many characters a target string is dissimilar from a reference string. Basically, the procedure consists in aligning a pair of strings and finds the alignment that gives the least number of editing operations to convert a target string into a reference string. Since the edit distance is a symmetrical measure, there is not actually any distinction between target and reference string. Thus, this measure can be used to compare two strings and decide whether or not they were originated from the same source.

This measure is largely used in the field of text mining in applications such as handwritten recognition and computation biology. This research represented a first attempt to apply such technique to solve a problem in transportation science.

The original idea with weights of 0 (for identical pair-wise characters) and 1 (for distinct pair-wise characters) was put into practice to match plates from a dual LPR setup (*LPR Setup 1* – 2007 Data) on a short segment of a freeway, in the vicinity of Knoxville metropolitan area. To this end, the *ED* was calculated for all possible pair of matches between the two stations, and the best assignment with minimum overall cost was found. The method proved to be suitable to indentify most vehicles passing trough both stations. However, the false matching rate was too high.

The *ED* with original weights does not actually measure the likelihood of two strings coming from the same ground truth value. In this case, whenever it is somehow likely to have pair-wise outcomes that differ from each other by more than one character the traditional *ED* does not work very well.

In a dual LPR setup for example, if there is a considerable proportion of vehicles detected at first station that do not travel or are not detected at the second station, as well as if there is high arrival rate of vehicles at the second station not detected at the first one, it is very likely to have a false match. Although this has not been fully analyzed in this research, I believe that it is the main reason for the poor performance of original *ED* formulation.

7.2 WEIGHTED EDIT DISTANCE

The LPR units can actually read most characters on the vehicle plates, even with low to moderate plate reading rate. One of the hypotheses in this research was that the LPR errors in misreading certain characters can be estimated and used to infer with certain degree of confidence the likelihood of every two imperfect readings being originated from the same vehicle. To account for the LPR mistakes in reading certain characters, I devised a weight function reflecting these recurrent errors. As could be seen, the literature offers extension of the original *ED* allowing the use of symbol-based weight functions. The main two extensions referred to the Generalized Edit Distance (*GED*) and the Constrained Edit Distance (*CED*), as described in Chapter 4.

I proposed a new weight function reflecting the probability of two character outcomes being originated from the same ground truth character. The calculation of the weight function was based on conditional probability theory. I defined the odds of character misreading into matrices whose cell values were the conditional probabilities associating ground truth characters to reading characters. Thus, using Bayesian theory I found out that all possible odds of association character outcomes can be calculated by a simple matrix multiplication of the estimates of the character interpretation matrices (one per LPR machine).

In an empirical study, on matching plates from a dual LPR setup (*LPR Setup 1*), either *GED* or *CED* formulation with the new weight function outperformed the original

ED. The new proposed procedures achieved about 90% of positive matches with only 5% to 8% of false matches. This result is encouraging considering that in all these initial experiments the passage time stamps were left out.

7.3 FIXED TIME WINDOW VERSUS VARYING TIME WINDOW CONSTRAINT

In all initial experiments I assessed the performance of different similarity measures with purposely no consideration of passage time stamps. This decision was to see how these procedures performed under high uncertainty. In reality, there will always be a time limitation (one day, one hour, etc) corresponding to the survey period or to an arbitrary restriction (e.g. maximum number of candidates for matching) that constraints the possible pair-wise matches.

As seen in the literature, the vehicle passage times have been used in earlier studies to help indentifying spurious matches in partial plate surveys and more recently to clean up LPR matched data from outliers. In this research, I incorporated the passage times in the matching procedure as additional constraint on the number of candidates at the upstream station to match up the current outcome from the downstream station.

Two methods were proposed. The first method, denoted Fixed Time Window Constraint (FTWC), consisted in simply restricting the number of candidates for matching by a fixed time window over the passage time differences. This method has been used before for the partial plate problem. The second one, denoted Varying Time Window Constraint (VTWC), consisted in changing the width of the time window according to the variation in the estimated travel time and in the magnitude of the similarity measure used.

Two LPR setups (*LPR Setup 1* - 2007 Data and *LPR Setup 2* - 2010 Data) were used to assess the two matching procedures, as described in Chapter 5. As expected, the experimental results showed that merely including the passage time information in the procedure considerably increased its performance. The two methods presented similar performance results with slight superiority of the VTWC method. However, it is expected that the VTWC method adjusts better to highly disperse traffic situations, not fully analyzed in this study.

Regarding to the two analyzed measures, the *GED* with the new weight functions outperformed the original *ED*. The difference in performance was more apparent for the setup (*LPR Setup 2* – 2010 Data) with higher likelihood of having a false match. It seems that the new weight functions make the similarity measure more robust in deciding whether or not a given pair of imperfect readings constitutes a false or true match.

7.4 SAMPLE SIZE ESTIMATION

In this research two procedures were proposed to estimate the association matrix **C** containing the conditional probabilities. The first procedure, denoted Ground Truth (GT) method, required the availability of ground truth values for a sample of recognized plates. The manual verification of the plate numbers from the image view of vehicles turned out to be a very time consuming process. To overcome this problem, a new approach, denoted Learning Process (LP) method, was proposed that consisted in collecting a set of "genuine matches" and derive the association matrix from the corresponding character association occurrences. The set of matches potentially genuine were found using one of the matching frameworks, either FTWC or VTWC, by an interactive process.

In systems where the LPR units operate permanently, it is more attractive to use the LP method to update the association matrix than spending a large amount of human effort and time using GT method. The only drawback of the LP method is that it may require a large amount of data, or many days of observation, considering that in most cases only a small proportion of vehicles detected at the first station are also detected at the second station.

An issue of interest was the sample size to obtain a statistically significant estimate of the association matrix. The results pointed to a large amount of data to obtain a highly precise and accurate estimate of **C**, using either the GT or the LP method. Obviously, the sample size or amount data depends on the cell precision and the simultaneous significance level used.

Alternatively, the sample size can be determined by sequentially adding data until the estimate converges, with small convergence criterion. The *LPR Setup 2* which was set up to operate continuously provided the data needed for such analysis. It seems that the estimation procedures converge to an approximate estimate using less data than what would be required for a highly precise and accurate estimate, i.e. with 0.01 cell precision and 5% confidence level. The two estimation procedures also tended to converge to the same matrix.

7.5 APPLICATIONS OF THE PROPOSED PROCEDURES

Speed enforcement over distance and travel time studies are the two main applications of the methods developed in this research. Speed enforcement requires better confidence of the matching procedures, thus calling for more accurate equipment and highly precise and accurate estimates for the association matrix. The latter is not a problem, since for speed enforcement the LPR setups should be installed permanently providing enough data for estimation. However, higher accuracy of the LPR units implies in higher prices of the equipment.

Speed enforcement over distance consists in recording the time of passage of vehicles at two checkpoints and subsequently calculates the vehicle speeds using the distance between the stations. Warnings or fines are issued to transgressors later. To this end, LPR units can be deployed to automatically record passage times and derive and store the vehicle speeds. Using LPR can significantly reduce the manpower needed to perform such enforcement. Moreover, in weight station locations where all trucks are required to stop for inspection, an LPR speed-enforcement system, with equipment strategically located, could issue warnings or citations as the perpetrating trucks stop on the weigh scale, with an officer stationed in the weigh house. This system could function in real time without the need for mailing out speeding tickets after the fact or pulling trucks over after dangerous high-speed pursuit, both alternatives resource- and labor-intensive.

In travel time studies the process of deriving vehicle speeds is similar to a LPR speed-enforcement system. The information generated in such studies can be used for example in level of service analyses of roadways. Furthermore, when the LPR system is part of an information system the expected travel times can be transmitted back to the drivers through message panels.

In real time systems such as information systems, the number of matches, or vehicles captured to estimate the travel times is crucial to increase the reliability of the information. This is true because vehicle travel times change over time due to traffic variation and if only part of vehicles are sampled the resulting estimations may not be representative of the time slots sent out to the drivers. The proposed matching procedures using passage time information, FTWC and VTWC, can be used to increase the sample size for the estimation of the travel times.

7.6 FURTHER STUDIES

In this section I present some recommendations for future work as follows.

7.6.1 Sensitive Analysis

Regarding to the dual LPR setup, it is believed from the empirical analyses that the performance of the matching procedures are affected by several factors. First, it is directly influenced by the accuracy of the LPR units, which is also affected by external factors. Second, the likelihood of finding a false match by chance in the database can also interfere in the matching performance. Such likelihood is a function of two traffic variables: the proportion of vehicles detected at the first station that travels to the second station and the arrival rate of vehicles detected at the second station not detected at the first one.

A further study could deal with the impact assessment of all variables cited above on the matching performance. Such sensitivity analysis could be carried out by means of a controlled experiment using simulation. For instance, the accuracy of the LPR units can be altered by randomly increasing the percentage of character mistakes; that is inserting or eliminating character mistakes from an existing LPR dataset containing the ground truth values. The traffic variables can be also altered by generating random plate numbers and includes them into the dataset.

Besides allowing assessing the performance of the matching procedures under several hypothetical scenarios, a sensitivity analysis would also allow to evaluate the impact of using only one occurrence matrix to estimate the association matrix by GT method. In multiple LPR setup, if all LPR units operate in different fashion would be necessary to estimate one matrix for each machine. This would require a large amount of manpower since the process of estimating **C** by GT is very time consuming. Using only one single matrix from a given LPR unit reduces the amount of work spent to estimate **C** in the expensive of a worse matching performance.

7.6.2 Sample Size Estimation

An additional analysis that has not been performed yet is the impact of the LPR accuracy on the association matrix estimation. The amount of data needed to estimate the association matrix is highly correlated with the accuracy of the LPR units. Therefore, the rate of convergence of the estimation procedures should vary with the changing in the LPR performance.

7.6.3 Online matching

The matching procedures proposed in this research can be used for online matching. In this case, the system tries to identify a vehicle at the moment it is detected at the downstream station and the information generated is immediately retrieved and analyzed. Further work still should be done to validate the proposed matching frameworks for an online application.

7.6.4 Extension of the Matching Procedures to Multiple LPR Setups

The matching procedures can be extended to multiple LPR setups such as multiple entry-exit points for OD estimation surveys or sequential setups in the case of route determination in urban areas. These issues were left out for further studies.

The arrangement of LPR units on multiple entries and exits is needed for origin-destination surveys where it is desired to estimate the trip patterns over a certain urban subarea. A set of n_o LPR units is assigned to cover the entry points while another set of n_d is assigned to cover the exit points, as illustrated in Figure 7-1. It is assumed here that all destinations are accessible from all origins, making possible $n_o \times n_d$ origin-destination combinations during the survey period.

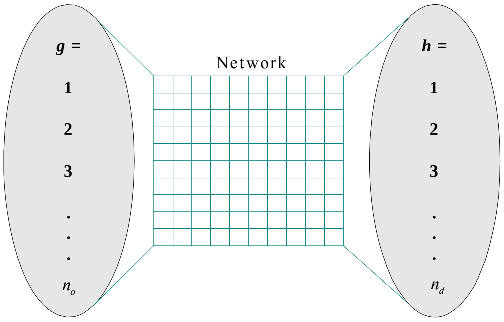


Figure 7-1 Multiple Exit-Entry LPR Setup

The problem here consists in finding an assignment, a set of matched plates, among all entry- and exit- points that minimize a overall cost function (the similarity measure) restricted to a set of time window constraints. Therefore, all methods already developed for dual LPR setup can be easily extended.

The case of sequential setup, where LPR units are disposed consecutively (Figure 7-2), is more challenged. The interest is to track a vehicle at multiple points in order reconstruct its route. In this case, a single vehicle usually generates multiple outcomes and can be also missed by a few or all stations. Thus the number of combinations can be immense, depending on the number of stations (s) involved. There is an extension of edit distance to deal with multiple string alignments (Carrillo and Lipman,1988; Gupta et al., 1995) and that may help to determine the set of stations where a given vehicle was detected.

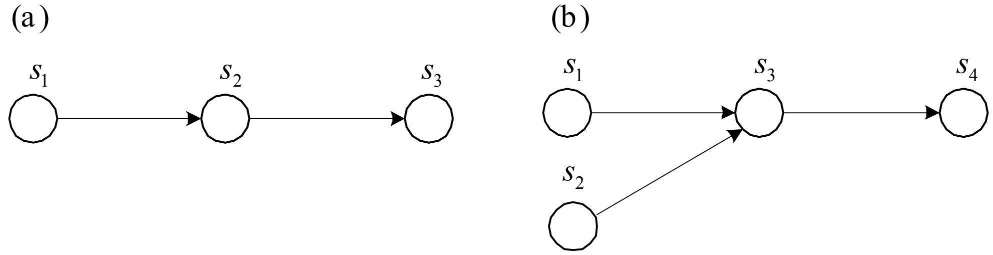


Figure 7-2 Sequential LPR Arrangements

LIST OF REFERENCES

Angers, C, 1984. Large Sample Sizes for the Estimation of Multinomial Frequencies from Simulation Studies. *Simulation 43*, 175-177.

Bazaraa M. S., Jarvis, J. J., Sherali, H. D., 2005. The Transportation and Assignment Problem. *Linear Programming and Network Flows*, 3rd Ed. Wiley Inter science, Chapter 10, 527-542.

Bertini, R. L, Lasky, M., Monsere, C. M., 2005. Validating Predicted Rural Corridor Travel Times from an Automated License Plate Recognition System: Oregon's Frontier Project. *The 12th World Congress of Intelligent Transport Systems*. CD-ROM. San Francisco, California, United States, November of 2005.

Buisson, C, 2006. Simple Traffic for a Simple Problem: Sizing Travel Time Measurement Devices. *Transportation Research Record 1965,* 210-218.

Carrillo, H., Lipman, D., 1988. The Multiple Sequence Alignment Problem in Biology. *SIAM Journal of Applied Mathematics*, 48, 1073-1082.

Casttilo, E., Menendez, J. M., Jimenez, P., 2008. Trip Matrix and Path Flow Reconstruction and Estimation based on Plate Scanning and Link Observations. *Transportation Research Part B*, Vol. 42, 455-481.

Clark, S. D., Grant-Muller, S., Chen, H., 2002. Cleaning of Matched License Plate Data. *Transportation Research Record:* 1804, 1-7.

Duda, R.O., Hart, P.E., Stork, D.G., 2000. Recognition with Strings. Pattern Classification. 2nd Ed., Wiley Inter Science, Chapter 8, 413-420.

Eberline, A., 2008. *Cost/Benefits Analysis of Electronic License Plates*. Final Report 637, Arizona Department of Transportation.

Fowkes, A. S., 1983. The Use of Number Plate Matching for Vehicle Travel Time Estimation. PTRC, *Proc., 11th Annual Conference on Transportation Planning Methods*, University of Sussex, London, 141-148.

Gupta, S. K. , Kececioglu, J., Schaffer, A. A., 1995. Making the Shortest-Paths Approach to Sum-of-Pairs Multiple Sequence Alignment More Space Efficient in Practice. *Proceedings of the 6th Annual Symposium on Combinatorial Pattern Matching*.

Han, L. D., Wegmann, F. J., Chatterjee, A., 1997. *Using License Plate Recognition (LPR) Technology for Transportation Study Data Collection*. Report submitted to the Tennessee Department of Transportation, TDOT.

Hauer E, 1978. Correction of License Plate Surveys for Spurious Matches. *Transportation Research Part A*, 13A, 71-78.

Hitchcock, F. L., 1941. Distribution of Product from Several Resources to Numerous Localities. *Journal of Mathematical Physics*, Vol. 20, 224-230.

Levenshtein V.I., 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10(8), 707-710.

Maher, M. J. The Analysis of Partial Resgistration-Plate Data, 1985. *Traffic Engineering & Control* 26, 495-497.

Makowski, G. G., Sinha, K. C., 1976. A Statistical Procedure to Analyze Partial License Plate Numbers. *Transportation Research 10*, 103-132.

Mamrak, S. A., Amer, P. D., 1980. Estimating Confidence Intervals for Simulations of Computer Systems. *Simulation*, 35(6), 199-205.

Marzal, A., Vidal, E., 1993. Computation of Normalized Edit Distance and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15 (9), 926-932.

Munkres, J., 1957. Algorithms for Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1).

Mavridis, D., Aitken, C. G. G., 2009. Sample Size Determination for Categorical Responses. *Journal of Forensic Sciences*, 54(1), 135-150.

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., 1997. License Plate Recognition Systems. In *ITS World,* 2(1), 26-29.

Nelson, L. J., 1999. Seeing is Believing. In *ITS International 23*, 38-40.

Nelson, L. J., 2000. Snap Decisions. In *Traffic Technology International*, 50-52.

Nelson, L. J., 2003. An Avid Reader. In *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.

Oommen, B. J., 1986. Constrained String Editing. *Information Sciences*, 40 (3), 267-284.

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.

Robinson, S., Polak, J., 2006. Overtaking Rule Method for the Cleaning of Matched License-Plate Data. *Journal of Transportation Engineering*, 132(8), 609-617.

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.

Shewey P. J. H, 1983. An Improved Algorithm for Matching Partial Registration Numbers. *Transportation Research Part B*, 17B(5), 391-397.

Tang, T, Roberts, M., Cecilia Ho, 2003. *Sensitivity Analysis of MOBILE6 Motor Vehicle Emission Factor Model.* Federal Highway Administration, FHWA Resource Center, Atlanta.

Thompson, S. K., 1987. Sample Size for Estimating Multinomial Proportions. *The American Statistician*, 41(1), 42-46.

Wagner, R. A., Fischer, M. J., 1974. The String-To-String Correction Problem. *Journal of the Association Computer Machinery*, 21(1), 168-173.

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. *Research into practice: 22nd ARRB Conference Proceedings*. CD-ROM. Australia Road Research Board, ARRB, Canberra, Australia, October of 2006.

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.

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.

APPENDICES

APPENDIX A

Association Matrices

LIST OF ACRONYMS

CED	Constrained Edit Distance
ED	Edit Distance
FTWC	Fixed Time Window Constraint
GED	General Edit Distance
GT	Ground Truth
LPR	License Plate Recognition
LP	Learning Process
VTWC	Varying Time Window Constraint

VITA

Moraes Oliveira-Neto was born in 1977, in Fortaleza, Ceará, Brazil, where he grew up. In 2002, he completed a B. S. in Civil Engineering at the Federal University of Ceará. In 2004, he obtained a M. S. in Civil Engineering with concentration in Transportation Engineering at the Federal University of Ceará. Between 2002 and 2006, he worked as a Traffic Engineer at the Advanced Urban Traffic Control Center of Fortaleza, Ceará. In 2010, he was granted a doctoral degree in Civil Engineering with concentration in Transportation Engineering at the University of Tennessee, Knoxville. As a Ph.D. student Moraes was recipient of the U.S. Dept. of Transportation's Eisenhower Graduate Fellowship. His research interests include computational transportation science, transportation modeling, applied statistics and operation research.