Technical Report Documentation Page

1. Report No.	2.	3. Recipients Accession No.
2004-29		

4. Title and Subtitle	5. Report Date
TMC Traffic Data Automation for Mn/DOT's Traffic Monitoring Program	July 2004
	6.

7. Author(s)	8. Performing Organization Report No.
Taek Mu Kwon	

9. Performing Organization Name and Address	10. Project/Task/Work Unit No.
University of Minnesota Duluth Northland Advanced Transportation Systems Laboratories 1023 Unversity Drive Duluth, Minnesota 55812	11. Contract (C) or Grant (G) No.

12. Sponsoring Organization Name and Address	13. Type of Report and Period Covered
Minnesota Department of Transportation 395 John Ireland Boulevard Mail Stop 330 St. Paul, Minnesota 55155	Final Report
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract (Limit: 200 words)

The Minnesota Department of Transportation (Mn/DOT) has been responsible for collecting, analyzing, and publishing traffic count data from the various roadway systems throughout the state. The traffic reporting system mainly developed by the Traffic Forecasting and Analysis Section (TFAS) of Mn/DOT has been used in several federal programs, internal Mn/DOT applications, and many private sectors. The objective of this project was to continue the TFAS' automation efforts by automating the TMC portion of traffic data (ITS generated data) contributed to the Mn/DOT's Traffic Monitoring System. The focus was given to develop an Internet based system that produces computerized reports on continuous and short-duration count data.

One of the challenges of utilizing ITS-generated traffic data for computing continuous and short-duration count was in dealing with missing and incorrect data produced by faulty conditions of traffic data collection systems including detectors and communication links. This study found that data imputation techniques based on spatial and temporal inferences of traffic flow can overcome the difficulties and produce accurate statistical data. This report describes the details on actual implementation of the algorithms developed, analysis utilities, and practical system integration examples. One unresolved issue in this project was dealing with the stations in which nearly no data is available for the entire year, which was observed from 2-3% of the short-duration count stations. This problem is left for future work.

17. Document Analysis/Descriptors	18. Availability Statement
ATR, Continuous Count, Short-Duration Count, Imputation, Automated Reporting, Missing Data, Spatial Inference, Temporal Inference, Bayesian Imputation	No restrictions. Document available from: National Technical Information Services, Springfield, Virginia 22161

19. Security Class (this report)	20. Security Class (this page)	21. No. of Pages	22. Price
Unclassified	Unclassified		

TMC Traffic Data Automation for Mn/DOT's Traffic Monitoring Program

Final Report

Taek Mu Kwon, Ph.D.

Department of Electrical and Computer Engineering

University of Minnesota Duluth

July 2004

Published by

Minnesota Department of Transportation
Office of Research Services
Mail Stop 330
395 John Ireland Bld.
St. Paul, Minnesota 55155

Acknowledgements

The author would like to acknowledge several Mn/DOT specialists and managers, and UMD students who contributed to this project. Dr. Eil Kwon and Marthand Nookala provided help in developing the basic concept of this project during the initial funding process. Thanks to Jonette Kreidewis for organizing and leading many Technical Advisory Panel (TAP) meetings throughout the project period. Frequent participants in the TAP meetings include Mark Flinner, George Cepress, Jonette Kreidewis, Dave Berg, John Bieniek and Jim Aswegan. They provided many constructive suggestions for this project. Mark Flinner provided many details for the project requirements and guided the overall project. In addition, he provided invaluable assistance and made suggestions for the development of algorithms throughout the project. The author would like to extend special thanks to him for his friendship and many contributions. In addition, thanks extend to Doug Lau at Traffic Management Center (TMC) for providing daily traffic data to the UMD Data Center, which served as the data source for this project and other applications. Finally, thanks to UMD students: Dan Rogahn for writing web interface, relational database and the Automatic Traffic Recorder (ATR) automation program, and Nirish Dhruv and Siddhath Patwardhan for testing imputation algorithms.

TABLE OF CONTENTS

List of Figures

Executive Summary

Computer automation of traffic data reporting can significantly reduce the reporting time and human errors. However, many challenges exist due to missing and incorrect data produced by traffic sensors. The goal of this project was to automate importing, filtering, analysis and computation of continuous and short-duration count data from Mn/DOT's Office of Transportation Data and Analysis (TDA) for the portion of traffic data collected by the Traffic Management Center (TMC). The TMC traffic data, which is part of the Mn/DOT's Intelligent Transportation Systems (ITS), are collected from an extensive network of loop detectors (about 4,000 loops) in 30-second intervals. Three parties, TDA, TMC, and the University of Minnesota Duluth (UMD) Data Center, collaborated on an integrated system that covers to do the following: TMC transmits the raw traffic data to the UMD Data Center where the data is archived; UMD data center processes and transmits the computed data to TDA; TDA then performs the final analysis, adjusts the data and saves them into a database for various applications. The objective was to make the overall system work with minimal human intervention.

As with most real-world data, TMC traffic data contains missing and/or incorrect data points that could potentially lead to error or bias in estimating true traffic conditions. Missing and incorrect data are frequently caused by constructions, power outage, temporary maintenance operations, calibration deviations, etc., which are unavoidable aspects of any traffic system. Identifiable missing or incorrect data points were marked as bad data and treated altogether as missing data points. Identification was done through a rejection rule that tests boundaries of each data point and a detection routine that checks occurrences of repeated patterns of identical volume counts based on maximum likelihood probability.

In this project, two new approaches were developed to cope with the missing and incorrect data. The first approach is based on predesignation of multiple detector sets per each station that were chosen according to an equivalency relation of traffic flow. The detector sets are then prioritized as primary, secondary, tertiary, etc. If primary does not pass the acceptance test, secondary is used, and similarly from the secondary to tertiary. In the actual implementation, detector sets were defined up to tertiary, which resulted in significant reduction of missing data due to replacements of missing data from the additionally available data from the equivalent detector sets. This approach essentially utilizes spatial relation of traffic flow for the recovery of missing data.

The second approach utilizes temporal trends of traffic flow. Traffic patterns tend to repeat day to day. For example, a Monday traffic pattern is similar to the Monday of the previous or following weeks except for the case when the same day of week is a holiday or near holidays. In addition, during the day, traffic patterns tend to follow repeating trends, typically showing two peaks: one in the morning rush hour and another in the afternoon rush hour. These temporal trends were used for constructing a probability model, which was in turn used for simulating the candidate values of missing data through multiples of Bayesian selections. A statistical average of the simulated candidates replaces the missing data. These temporal inferences were integrated as an algorithm referred to as the Nonnormal Bayesian Linear Regression Imputation and used

for imputation. In short, although missing data is lost data, it was found that temporal and spatial inferences of traffic flow could provide key statistical information for estimating the lost data in order to improve the validity of Annual Average Daily Traffic (AADT) and other annual computations.

The result of new data processing method was that since over 300 days of data at most stations are available after spatial and temporal treatment of missing data, there is no longer a need to factor the short-duration count for day of week and month of year as was traditionally done. The original approach of selecting only 48 or 72 hours of representative traffic data was modified to directly compute AADT from the available daily traffic data. In addition, data visualization tools were developed, so that analysts can view and study the daily trend of data for the selected year.

As a part of the overall automation system, a relational database was constructed for the maintenance of detector sets defined for each station. Along with the database, sophisticated web interfaces were designed and implemented using Python scripts in order to facilitate remote maintenance of data by Mn/DOT personnel. However, it was later found that the complexity of the web application required training and high cost of maintenance thus hindering the use of the system. Although this database is presently used for the computation of continuous-count data, the TDA and UMD research team elected to change it to a simpler system. Recognizing the shortcomings, a simple text-based format for the detector sets of short-duration count stations was developed. It was found that this simple text format along with a single step file transfer was a better approach in terms of requiring no training and no cost of maintenance. This is an interesting finding. While using relational databases with web interfaces is a popular method today, it is not necessarily a good approach for accessing remote data that does not require such overhead.

In summary, this project demonstrated that ITS generated traffic data could be effectively used in conventional data reporting applications with high accuracy. The challenge was coping with missing and incorrect data found in ITS traffic data. For that, we successfully developed and implemented data imputation techniques based on spatial and temporal inferences of traffic flow. These data processing techniques served as the key component for the overall system automation. However, the problem of dealing with missing data is still unresolved for those stations in which almost no data is available for the entire year, which was observed from 2-3% of the short-duration count stations. We leave this problem for future work, but also recommend reducing such cases by early detection of detector problems and improved maintenance.

CHAPTER 1
INTRODUCTION

1.1 Mn/DOT's Traffic Monitoring Program[1]

Traffic monitoring programs have been one of the important functions in state and federal level transportation departments. The Minnesota Department of Transportation (Mn/DOT) has been maintaining an active traffic monitoring, forecasting and analysis program. Mn/DOT has been responsible for collecting, analyzing and publishing traffic count, classification and weight data from the various roadway systems throughout the state. These traffic data have a wide variety of users including five of the six federally mandated management systems in Mn/DOT. Elements of today's Mn/DOT Traffic Monitoring System (TMS) are administered cooperatively through the efforts of three separate Divisions[2] and all the District Offices:

- Program Support Group,
- Program Delivery Group,
- State Aid for Local Transportation Group, and
- Metropolitan Division.

The Traffic Forecasting and Analysis Section (TFAS) of the Program Support Group has been planning and administering the Department's traffic monitoring program. Today's Mn/DOT's Traffic Monitoring System (TMS) is a product of ongoing automation activities designed to improve traffic data quality and timeliness for traffic volume data users. Although the objectives may vary over time, the central premises of these efforts are the following:

[1] This portion was written based on a two-page summary of the Mn/DOT's traffic counting program published in Summer 1996 and additional information provided by the Traffic Forecasting and Analysis Section of Mn/DOT (Mark Flinner).

[2] At the time of this writing, Mn/DOT was going through reorganization, thus names and divisions provided here only represent divisions as they were before the reorganization.

- The TMS must be based on statistically valid principles
- The TMS must use data systems that integrate all necessary data types
- Traffic data should be collected, processed and reported in electronic form. Manual aspects of TMS operation should be minimized.
- Lines of communication must be established and maintained between those involved with the TMS and the customers using information coming from it.
- The TMS must be dynamic and flexible in order to take advantage of new methodologies and technologies that bear on traffic data.

These premises and TFAS' long-term objectives served as the main impetus for creating a project that would extend the degree of automation for the traffic volume data collected by the MN/DOT's Traffic Management Center (TMC) in the Metropolitan Division.

One measure of roadway use is the Annual Average Daily Traffic (AADT) volume. These estimates represent how many vehicles are traveling on the state's roadway segments (in both directions) on an average day of the year. These traffic volume data are derived from two kinds of traffic counting activities. The first involves continuous traffic counting devices or ATRs (automatic traffic recorders), which record hourly volume data 24 hours a day throughout the year. The second involves short-duration counting devices such as road tubes and manual or portable automatic vehicle classification devices. Data collected from these counting activities are screened, factored if necessary, and analyzed to create AADT volumes that are mapped and distributed for use by the Federal Highway Administration (FHWA), MN/DOT, county and local highway departments, and area planning organizations. Additionally, private sector business consultants, engineering firms, and real estate interests, among others frequently request and use the department's traffic volumes in their work.

Mn/DOT's ATRs are located primarily on trunk highways throughout the state. Traffic volumes are retrieved from these ATRs by the TFAS staffs several times a week. The ATR data are nominally screened using a SAS program. They are then output into a format that is suitable for loading into the MN/DOT's Traffic Analysis Expert System (TAES). Analysts then edit the ATR data using the TAES to check for equipment malfunctions, to cull out bad data, and to synthesize data where data are missing. After

the ATR data have been edited, they are ready to be used for reports and to create seasonal/day-of-week adjustment factors for the short-duration count data collected at approximately 32,000 locations throughout the state.

The short-duration count data are collected using portable data collection devices such as pneumatic road tubes for a minimum of 48 hours duration. Where there are permanent sensors available (such as are managed by the TMC), short duration samples are manually taken from the loop sensor data files and sent to the TFAS staff. Every short duration count is manually entered into a relational data base management system programmed in R:BASE[3], and is further adjusted by the seasonal/day-of-week factors that are derived from the ATR data. After the short-duration count data are entered into the database, they are evaluated against past AADT estimates, and recounts are ordered when anomalous data values, equipment malfunctions, or tube set failures indicate the need for a recount.

At the end of the counting season, the short-term counts are evaluated for spatial and temporal coherency and placed on draft traffic volume maps. The draft maps are circulated to Mn/DOT district and/or county and municipality engineers for feedback. Final traffic volume maps are then prepared and distributed to Mn/DOT's traffic volume data users. The Department's TFAS has already published the Department's first automated traffic volume map, and anticipates automating its entire traffic volume mapping process in the future using a combination of CADD technology, database integration, and correlation to the department's GIS base map. The automation process will also enhance the Department's ability to examine concurrently both spatial and temporal changes in trunk highway use. AADT are automatically loaded into the department's computerized Transportation Information System (TIS) annually. In addition, as the supporting GIS map base is completed, TFAS plans to make its traffic volume data available electronically throughout the state. Traffic volume maps are already available on the Department's web site to facilitate dissemination of the department's AADT traffic volume information.

[3] Information on R:BASE can be found in the web site "www.rbase.com."

1.2 Project Goals and Tasks

1.2.1 Background

The Metropolitan Division's Traffic Management Center (TMC) monitors and manages traffic on the metro area freeways and arterial highways and is responsible for collecting the traffic data for the roads under their management. The TMC maintains and collects volume and occupancy data from about 4,000 inductive loop detectors at a constant rate of 30 seconds through an extensive network of detectors and computerized data communication. This type of traffic data is often referred to as ITS (Intelligent Transportation Systems) traffic data and has been used for traffic control and monitoring operations at TMC. For most state and local transportation departments, ITS traffic data is a largely untapped resource for traditional traffic counting programs, although it provides a rich set of data that could be used for traffic counting [4]. The problem lies in that ITS data is susceptible to outliers, missing values and other types of data anomalies that are not easy to resolve. Moreover, since the data is collected at 30– second intervals, the amount of data is substantially large and difficult to manage using simple desktop PC tools. Nevertheless, since the 1980s, Mn/DOT has tapped into the ITS traffic data and has produced the short-duration and continuous count data through manual compilation for the locations along instrumented metro freeways. As part of the TFAS's on-going efforts to integrate and automate the Department's traffic monitoring program, the present project was developed in collaboration with the Transportation Data Research Lab at the University of Minnesota Duluth. Simply stated, the project aimed to provide well screened and high quality data for the TMC portion of the ATR and short-duration traffic data.

The TFAS has established unique sequence numbers (different from the TMC's station numbers) that identify traffic counting locations throughout the state. The traffic locations from TMC consist of a primary set of detectors that typically represent a segment of the road. Along with the sequence numbers, a new concept was introduced in this project, which allows designation of alternative sets of detectors for each station in order to improve the quality and reliability of the data. Therefore, a station, whether it is

ATR or Short-duration Court (SC) station, was allowed to be assigned up to three sets of detectors, which are referred to as primary, secondary and tertiary detector sets. If the volume count collected from the primary of a station is disqualified from the acceptance test, the data from its secondary replaces the data of primary and works similarly for the secondary and tertiary. The central mechanisms of the methodology employed in this project are prioritized choices of detector sets for spatial inferences along with multi-level Bayesian data imputations utilizing temporal relations.

1.2.2 Project Goals

The objective of this project was to develop an automated system for the TMC portion of ITS traffic data contributed to the Mn/DOT's TMS. The system should provide all automated acceptance tests required by TFAS for both continuous-count and short-duration count volume data. The system should provide data formatted according to the Mn/DOT's SAS and R:BASE application input requirements and should automatically transfer data to the TFAS server. The automation system should work with minimum human intervention.

1.2.3 Project Tasks

This section describes the original tasks and the modifications that occurred during the period of project for further improvement. All modifications of tasks were determined collectively based on the analysis of benefits by the Mn/DOT and TDRL research team.

Task 1: Development of station identification database

This task involves the design and implementation of a station identification database that would be used as a detector look-up table for the corresponding data files from the TMC traffic data archives. A primary station (whether it is continuous or short-duration) consists of multiple traffic detectors in a segment of a road, at which a unique sequence number is assigned. A safe fallback mechanism is in place by designating the secondary stations, i.e., when the traffic data collected from the primary does not pass the acceptance test, the data collected from its secondary replaces the data. The same relation

is applied from the secondary to tertiary detectors. The detector assignment of a station may change over time due to construction of roads or other reasons. Therefore, there is a need to develop an efficient way for the TFAS staff to freely change and maintain the detector/station assignments. This is achievable if a database maintains the station identification information from which the automation system retrieves the detector assignments in order to perform the required processing. Hence, the research team will rigorously work with the TFAS staff in defining a sound database system for the station information.

Modification of Task 1: Task 1 was successfully completed by developing a flexible database using a relational database management system (RDBMS). The research team also developed a web interface using Python scripting to allow database access from Internet. Presently, all ATR data are produced based on this database lookup table. However, later the research team observed that using RDBMS had a few drawbacks from the maintenance point of view. It required training of TFAS staff on how to use and manage the web interface. Moreover, tracking and retrieving the history of database change information along with rollback computation was too complicated for ordinary analysts with medium level computer skills. Maintaining different levels of user access rights based on the security level assigned to the users posed additional inconvenience. Therefore, the TDRL and Mn/DOT research team decided to abandon the complicated RDBMS integrated web approach and to develop a remote file-transfer approach based on formatted text files. This approach was implemented for the AADT computation of SC stations, and the research team decided to extend it to accommodate for the ATR stations.

Task 2: Continuous-count data automation

This task involves the design and implementation of an automated reporting system for continuous-count data. Continuous count data, which are packaged into the SAS application acceptable input format, consists of records of hourly volume counts. Because the TMC data are collected at 30-second intervals, a conversion to hourly volume is required as a pre-computation process. The first stage of the automated-test is

to check whether any detectors in the primary station failed due to a hardware or a communication error. This test is conducted through checking the flags available from the TMC data source. If the station passes the first stage test, the second stage test is conducted. The second-stage test checks whether the data stays within the statistical acceptance range by analyzing the tolerance range of the volume and occupancy relationship. If the primary station fails from one of the two tests, the data from the secondary station are examined. If the secondary station passes the two-stage acceptance test, its data are used to construct the continuous count data, otherwise, the data are set to zeros.

The present SAS application input format of the continuous-count data consists of the following fields: AM/PM, Month, Day of the Week, Station ID, Lane Direction, and a Set of Four-Digit Hourly Volumes. This format may be modified in the future by the TFAS staff, so the project team will be closely working with the TFAS staffs to accommodate needed revisions. Where the automation system will reside will be determined based on the progress of the UMD data center. If a complete data archive is constructed at UMD, the automation system will be placed at the UMD data center. However, if the progress is not sufficient to accommodate the continuous-count automation, it will need to reside at TMC or in another office within MN/DOT.

Modification of Task 2: This task was mostly implemented as planned. Utilizing volume/occupancy relation as an acceptance test was rigorously studied, but it was determined to be unreliable. Later, the research team proposed to include an imputation process for missing data before computing hourly data.

Task 3: Beta-test and correction for the continuous-count data automation

In order to ensure that the continuous-count automation works correctly, a collaborative beta test will be conducted with the TFAS staff. This test will be performed in two stages. First, past data will be used to check the system's correctness in the data acceptance test and timely delivery function of the data to the TFAS server. Second, present live traffic data will be used to emulate the real conditions. Any deficiency found will be corrected during this period.

Modification of Task 3: This task was completed as planned; no modification was made.

Task 4: Short-duration count data automation

Short-duration (SC) count data for a station is defined as a 24-hour (noon to noon) volume average computed over the qualified three consecutive days (48-hour period, noon to noon). The range of qualified days was defined to be in between *April 1 to November 1***.** During this period, days with holidays, near holidays, detour, incidents, severe weather, and special events are excluded from the qualified pool of days. For any given week within this pool of dates, three qualified 48-hours periods, *Monday noon to Wednesday noon* (this period is denoted by the middle-date, Tuesday), or from Tuesday noon to Thursday noon (middle-date=Wednesday), or from *Wednesday noon to Friday noon* (middle-date=Thursday) are selected. The intent of these choices is to find a typical weekday traffic volume for any given location. In order to allow controllable automation, the operator will be able to set (1) the number of allowable low and high volume flags in certain time periods of the day (Stage One test of Task 2), (2) the acceptance parameters for the volume/occupancy test (Stage Two test of Task 2) and (3) (for the short duration data only) disqualified days, from which the automation system creates the acceptance test. The operators will be able to see these preset parameters, so that they would know in what acceptance basis the data were produced.

Modification of Task 4: Several modifications were made for Task 4. First, imputation techniques were introduced to resolve issues of various missing patterns in the raw data. By utilizing the availability of most of the 365 days of data per station, AADT was directly computed using the TMC traffic data. In addition, daily traffic volume data with graphic utilities were provided instead of the originally planned data for 48-hour periods. Consequently, Task 4 was extended up to the final step of computing a true AADT, resulting in a much higher level of data automation. Due to this modification, the old way of deducing AADT based on an adjusted short-duration (48 hours) count (i.e., seasonal and day-of-week adjustment) is no longer needed for the TMC locations.

Task 5: Beta-test and correction of short-duration count data automation

In this task, a beta-test will be conducted for the short-duration automation portion to ensure that the overall system works correctly. Because the short-duration count data are generated from a very large data set (seven months of 30-second detector-by-detector data), the beta test is only possible with the past data. A thorough test against the last year's data will be conducted in collaboration with the TFAS and TMC staffs. Any deficiencies found will be corrected during this period.

Modification of Task 5: Due to the modification in Task 4, the beta test was directly conducted against the past AADT data. An extensive test going back to 1999 was conducted to verify the data.

Task 6: Report and system manual write-up

A thorough report will be written to provide the detailed design and algorithms of the completed automation system. A user manual will be completed during this task period.

1.2.4 Report Organization

The rest of report is organized as follows. Chapter 2 describes an overview of the overall system at a block diagram level. Chapter 3 provides detailed descriptions on background, theoretical aspects, and classification issues on missing data. In addition, several new imputation algorithms developed based on the characterization of missing patterns are described. Chapter 4 describes how the actual system was implemented for computing continuous count and short-duration count data. Chapter 5 provides conclusions, recommendations, and future work.

CHAPTER 2
OVERVIEW OF THE SYSTEM

As described in Section 1.2.2, the goal of this project was to extend the present Mn/DOT's TMS automation efforts for the TMC's ITS traffic data portion. In order to achieve this objective, the overall system was designed around on-line availability of data through Internet connections. The main linkage to establish was an on-line relation between the data production capability of the Data center at the UMD TDRL and the servers at the Mn/DOT TMS.

Although the overall system was designed based on a multi-tiered architecture, at the conceptual level, it may be described as a blackboard concept in a classroom. This relationship is illustrated in Figure 1. Data can be written to or read from the blackboard server by TDRL Data Center or Mn/DOT TMS servers. The arrow lines indicate Internet data connections, and the sequence of data flow works as follows. Files that include the detector lists that specify primary, secondary and tertiary detectors for ATR and SC stations are posted by Mn/DOT on the blackboard server. The TDRL Data Center servers download the detector list files from the blackboard server in order to use them to compute the SC and ATR data. If multiple versions of detector list files were uploaded, Mn/DOT may specify which file to use. The Data Center servers then produce the required data (AADT and ATR data), and post the data on the blackboard server. For clear distinction, the data is always transferred in a file form with a file name that includes year, moth, and day information. The Mn/DOT servers or analysts regularly monitor the data files and download the files into the TMS when the data are available. In all cases, the file names on the blackboard include date information along with data type to prevent any conflict or confusion in the data version or usage of the data.

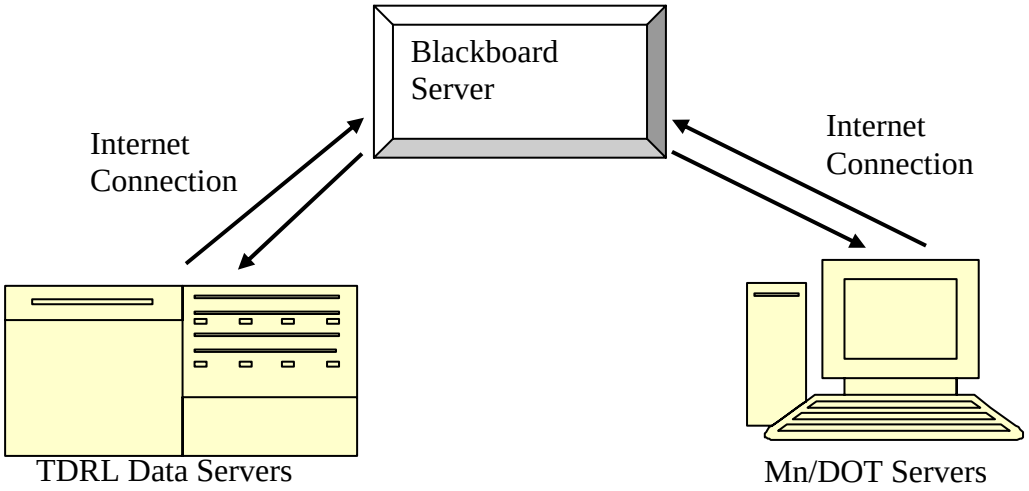


Figure 1: System level concept of data automation

CHAPTER 3
TREATMENT OF MISSING DATA

3.1 Introduction on Missing Data

As with most real-world data, ITS traffic data contains missing and incorrect data. In fact, since ITS traffic data are collected 24 hours a day throughout the year using computerized data collection systems, presence of data-loss due to hardware malfunction at the site or along the transmission lines is a high probability. More specifically, construction, power outage and temporary maintenance operations are unavoidable aspects, which mostly likely lead to a data loss. Missing data itself could provide us a great deal of information about the loop detectors, reliability, maintenance requirements and the expected quality of data. However, for traffic counting purposes, estimating the missing data is essential.

Attempts have been made with some success, to estimate missing data in a collection of ITS traffic data. Research at the Texas Transportation Institute (TTI) has explored regression analysis in combination with the Expectation-Maximization (EM) algorithm and compared the results with those from simple techniques such as straight-line interpolation and "*factor-up*" on traffic data [3]. The results are very encouraging. The EM algorithm, however, is rather computationally intensive and, as the researchers conclude, the marginal improvement in performance did not weigh well against the time and effort that goes into the implementation of the EM algorithm. Moreover, treatment of larger blocks of missing data has not been addressed in their study, a potential problem with EM. Schmoyer et al. [4] proposed a simple filtering approach for detecting missing data and linear regression estimates for the treatment of missing data. Again, this approach does not address large blocks of missing data. A school of time series estimation and filtering approaches exists, which has been known to be effective in recovering missing data or removing noise from band-limited signals [9,10,11,12]. Since most ITS traffic data are obtained by sampling data at a constant rate such as 30 seconds or 5 minutes, they are indeed a time series and could be applied to the vast array of

available time-series algorithms. However, no direct study results on traffic data are presently available to the best knowledge of the author.

Many rigorous research works on imputing missing data have been conducted in the field of statistics for applications in social science survey data, since such data most likely contain non-responses. Little & Rubin [5] essentially developed and laid foundations on the analysis of multiple imputation approaches on non-response survey data and suggested a number of statistical models based on historical inferences. These pioneering works are mostly based on likelihood estimates derived from formal statistical models. Schafer extended the analysis to incomplete multivariate datasets with continuous and discrete variables and applied EM algorithms and Monte-Carlo based Markov chain approaches. In a broad sense, the approaches mentioned can be called Bayesian approaches, since they explicitly use probability for quantifying uncertainty in inferences based on statistical data analysis [8].

This chapter describes classification of missing data patterns and the treatment of them as developed in this project.

3.2 Classification of Missing Data Patterns

3.2.1 Spatial and Temporal Characteristics of Traffic Data

Before investigating the missing traffic data patterns, it is important to recognize that traffic data inherently holds spatial and temporal relationships if it is comprised of data from multiple detectors in multiple locations. Spatial relation refers to a geographical relation of detectors, and it may be characterized using the size of geographical area from a smaller to a larger scale. For example, detectors could be characterized as detectors in a station[4], a road, a county, or a state. Similarly, temporal relation may be described using an increasing time-scale such as seconds, hours, days, months, and years. These inherent relations could be used as a reference for how to classify the missing data patterns. For example, data may be missing at a different spatial level such as a detector (lane) or a station (directional total) level, or at a different time

[4] A station is formed at a location of road where loop detectors are installed at each lane to observe the sampled view of the traffic flow in that road.

scale such as minutes or hours. The challenge is how to effectively combine both the spatial and temporal characteristics into one uniform representation.

3.2.2 Classification by a Tree Structure of Missing Data Patterns

In order to investigate missing patterns in the TMC traffic data, we observed statistics on stations for year 2001. Figure 2 shows missing data statistics for a typical station based on counting of days for missing percentage per day for the year 2001. Notice that the number of days containing more missing data in a year decreases as the percentage of missing increases. Based on this observation and the characteristics of traffic data discussed in Section 3.1.1, we found that the missing patterns fall into a leaf of a tree structure illustrated in Figure 3. This tree structure of missing data patterns was taken into account in designing the overall imputation strategy.

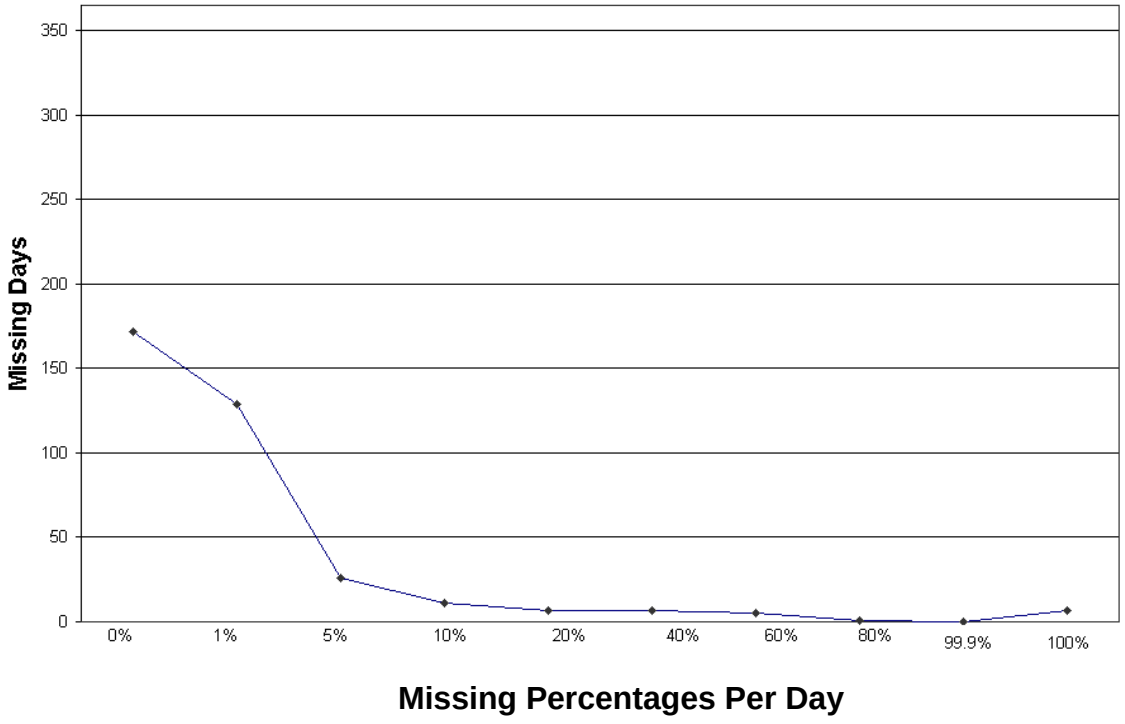


Figure 2: Typical annual missing percentages of a station (station number 1078E)

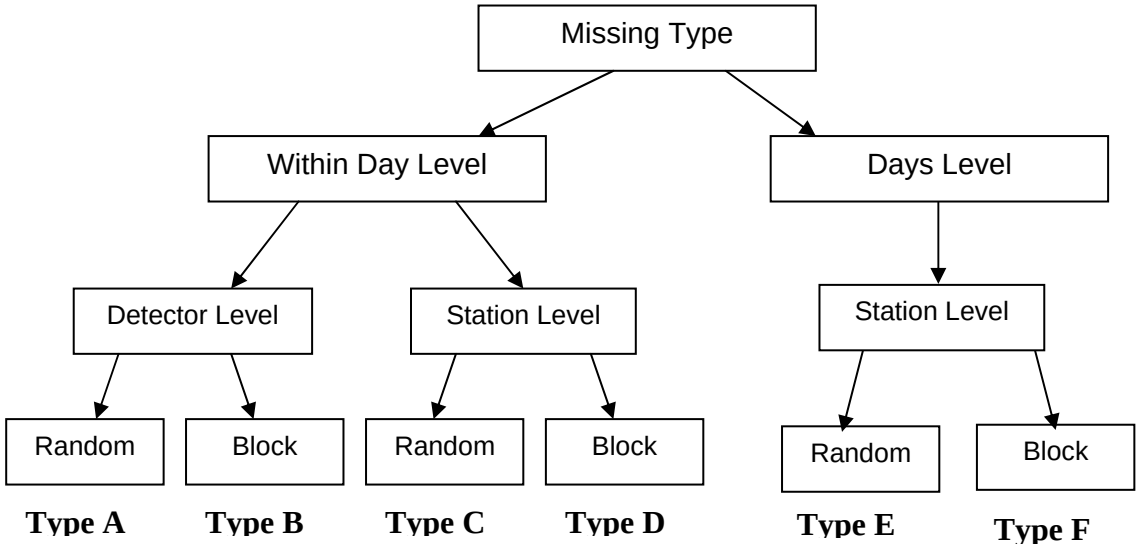


Figure 3: Classification of missing patterns in a tree structure

As shown in Figure 3, missing patterns fall into one of the branches of the tree. At the top level, we classify the missing data types into two types, either the whole days are missing, or a part of a day is missing. If only a part of the day's data is missing, we further divide it into two missing types in spatial relation, i.e., a part of detectors data is missing, or the whole directional station data is missing. The next level down is classified based on the occurrences of random missing or blocks of missing (a block means a group of consecutive data). For the day level, the missing data patterns occur either at random or in blocks of days but are only classified at the station level, since the detector level overlaps. Also, since our objective is computing AADT at the station level, treatment of station level covers the needs of the algorithm development and missing data pattern analysis. For convenience of description, we name each leaf of the tree from *Type A* to *F* from left to right branches. The basic idea of our imputation strategy is the following: when data imputation is started from *Type A* and progressed towards *Type F*, each stage ends up supplying more data for the next level, providing further inference. Below, we further clarify missing data relations in detector/station level and random/block level.

Detector or Station Level Missing

This distinction occurs due to the spatial relationship of detectors. In a station, only one or two detectors could be broken and produce missing or incorrect data. Such cases exist due to a partial construction or maintenance operation of roadways or breakage of loop wires by cracks. In other cases, all of the detectors in a particular station can be broken, which leads to station-level missing data patterns. Station level missing data also happens because the detectors in a station are usually connected to a single controller box that sends data to the central data collection server. Therefore, if a controller malfunctions (e.g., looses power or communication link), the result becomes station-level missing data pattern.

Random or Block Level Missing

Random or block level missing data is determined using a temporal relationship of missing data patterns. Random missing data refers to missing values that occur completely randomly. This is equivalent to ignorable non-response data in statistics where many multiple imputation techniques have been applied [5]. In general, random missing data are caused by transient hardware or software problems that are difficult to identify and correct. Therefore, we always have to expect existence of random missing data patterns in traffic data. Block missing data refers to missing values that occur in a consecutive blocks of data in temporal relationship. Although a high density of random missing data theoretically can lead to block missing data, such rarely happens in real data. Most block missing data occurs in a long sequence of data such as half day, few months, or whole year in some cases according to our observations. In the real world, construction of a segment of a road frequently occurs for an extended time period during the summer construction season, which leads to a long sequence of block missing. This type of missing data pattern cannot be imputed using the techniques used in random missing data [5,7]. This type of missing data pattern is more difficult to impute or estimate due to limited inferences.

3.3 Multiple Imputation

Multiple imputation (MI) is a statistical technique for analyzing incomplete data sets, that is, data sets for which some entries are missing. Each missing datum is replaced by *m*> *1* simulated values, producing *m* simulated versions of the complete data. Each version is analyzed by standard complete-data methods, and the results are combined using simple rules to produce inferential statements that incorporate missing data uncertainty [7].

Rubin [5,7] presented a method for combining results from a data analysis performed *m* times, once for each of *m* imputed data sets, to obtain a single set of results. From each analysis, one must first calculate and save the estimates and standard errors. Let *Q* be the quantity of interest, such as the mean of population. Suppose that $\hat{Q}_j$ is an estimate of a scalar quantity of interest (e.g. a regression coefficient) obtained from data set *j* (*j*=1, 2, ... , *m*) and U_j is the standard error associated with $\hat{Q}_j$. The overall estimate is the average of the individual estimates,

$$\bar{Q} = \frac{1}{m}\sum_{j=1}^{m}\hat{Q}_j \qquad (1)$$

For the overall standard error, one must first calculate the within-imputation variance,

$$\bar{U} = \frac{1}{m}\sum_{j=1}^{m}U_j \qquad (2)$$

and the between-imputation variance,

$$B = \frac{1}{m-1}\sum_{j=1}^{m}(\hat{Q}_j - \bar{Q})^2. \qquad (3)$$

The total variance of $(Q-\bar{Q})$ is

$$T = \bar{U} + \left(1 + \frac{1}{m}\right) B. \qquad (4)$$

The overall standard error is the square root of *T*. Confidence intervals are obtained by taking the overall estimate plus or minus a number of standard errors, where that number is a quantile of Student's t-distribution with degrees of freedom

$$df = (m-1)\left(1 + \frac{m\bar{U}}{(m+1)B}\right)^2. \qquad (5)$$

A significance test of the null hypothesis Q=0 is performed by comparing the ratio

$$t = \frac{\bar{Q}}{\sqrt{T}} \qquad (6)$$

to the same t-distribution. Additional methods for combining the results from multiply imputed data are reviewed by Schafer [6].

3.4 TDRL Algorithms

Little and Rubin suggested several imputations that are defined statistically proper [7]. In this project, one of them referred to as the nonnormal Bayesian imputation procedure that is proper for the standard inference was adapted. This section describes the detailed algorithms developed for this project.

3.4.1 Nonnormal Bayesian Imputation Algorithm

According to Rubin’s analysis, many Bayesian models beside the normal approximately yield the standard inference with complete data, and thus many such models can be used to create proper imputations for ignorable nonresponse. He suggested the following algorithm:

Algorithm 1: Nonnormal Bayesian Imputation

Input: Observed Values $(Y_1,...,Y_n)$

Output: *M* Imputed Values

Step1: Draw (*n-1*) uniform random numbers between 0 and 1, and let their ordered values be $(a_1,...,a_{n-1})$; also let $a_0 = 0$ and $a_1 = 1$.

Step2: Draw each of the M missing values by drawing from $(Y_1,...,Y_n)$ with probabilities $(a_1 - a_0),(a_2 - a_1),...,(1 - a_{n-1})$.

3.4.2 Imputation of Randomly Missing Data Patterns

Whether data is at the detector or station level, random data missing implies randomness of the occurrences and thus availability of observable data in the neighborhood of missing data patterns. While missing data samples are randomly located and unpredictable, traffic volume counts during the day approximately follow distinctive patterns that repeat over and over again. More specifically, it has a camel back pattern; that is, traffic volume is generally very low from mid night to about 5:00am, and then it is gradually increased as time approaches towards morning rush hour. During the morning rush hour, traffic volume reaches the morning peak and then it is decreased again but not as much as the midnight. In the afternoon it reaches another peak. In order to incorporate such time dependent patterns while maintaining the variability, we devised an algorithm that combines linear regression with a Nonnormal Bayesian imputation [7] for imputing randomly missing data patterns. We refer to this algorithm as the Nonnormal Bayesian Linear Regression (NBLR) algorithm and it is presented below. The basic idea follows Rubin's suggestion on creating nonignorable imputed values using ignorable imputed models [7]. Let a sequence of volume counts in *n* elements that includes *m* missing values be denoted by

$$V = (V_{x_1}, V_{x_2}, ..., V_{x_k}, V_{x_{k+1}}, ..., V_{x_{k+m}}, ..., V_{x_n}).$$

It is a consecutive portion of volume data taken around the missing values where one or more observed data exist. The observed *(n-k)* values are denoted as $V_{obs} = (V_{x_1}, V_{x_2}, ..., V_{x_n})$, and the missing values are denoted as $V_{mis} = (V_{x_k}, V_{x_{k+1}}, ..., V_{x_{k+m}})$.

Algorithm 2: Nonnormal Bayesian Linear Regression (NBLR) Imputation

Input: *V*

Output: estimate of missing values $\hat{V}_{x_k}, \hat{V}_{x_{k+1}}, ..., \hat{V}_{x_{k+m}}$

Step 1: Find the parameters of a linear regression model given by $\hat{y}_{x_i} = \hat{\beta}_0 + \hat{\beta}_1 x_i$ using V_{obs}.

Step 2: Construct a random variable D_{obs} using the difference between the regression estimate and the observed values, that is,

$$\begin{aligned} D_{obs} &= (V_{x_1} - \hat{y}_{x_1}, V_{x_2} - \hat{y}_{x_2}, ..., V_{x_n} - \hat{y}_{x_n}) \\ &= (d_{x_1}, d_{x_2}, ..., d_{x_n}) \end{aligned}.$$

Step 3: Draw *M* imputed values for each missing values by applying D_{obs} to Algorithm 1 and then compute the estimate of missing values as:

$$\hat{V}_{x_k} = \hat{y}_{x_k} + \tilde{d}_{x_k}$$

where $\tilde{d}_{x_k}$ is the average of *M* imputed values.

This algorithm essentially utilizes the inferences in time trend of traffic volume using the observed values through the linear regression model while the nonnormal Bayesian drawing of values capture the statistical inference of the observed values. The effect of the algorithm is illustrated using a real data example in Figure 4 by showing before and after imputation. The data used is station data with 5-minute intervals for a day. Notice that the algorithm clearly captures the time trend as well as the statistical variability and fills in the missing values. Many other cases tested resulted in a similar outcome.

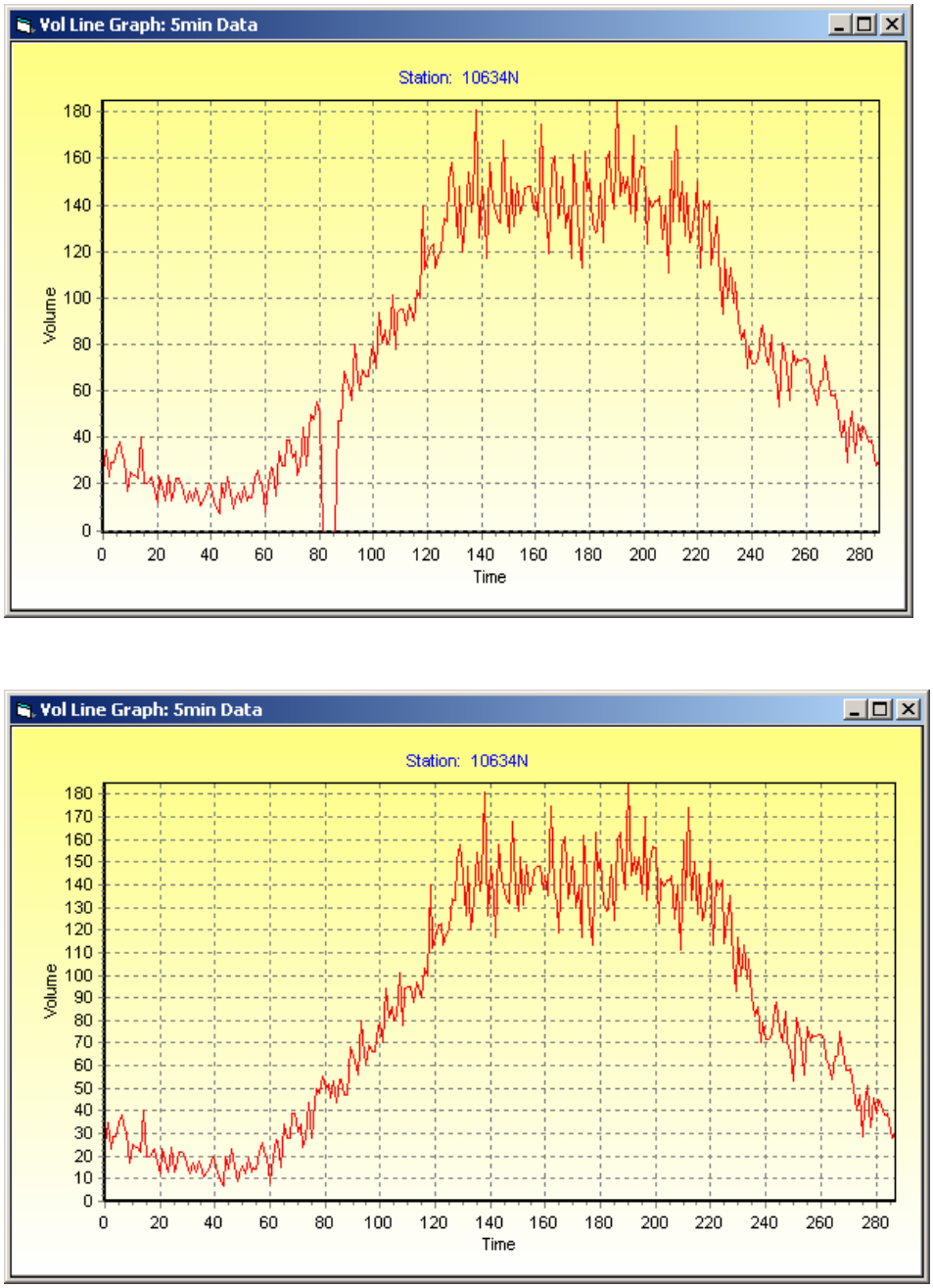


Figure 4: Effect of NBLR: before imputation (top) and after imputation (bottom)

3.4.3 Imputation of Block Missing Data Patterns

Block missing data refers to existence of a large amount of consecutive missing values in the data, such that neighboring values can no longer provide enough time trend inferences. In this case, the NBLR algorithm in Algorithm 2 cannot be used since the time trend inferences are not available. Therefore, some other inferences must be used. In traffic volume data, one can easily observe repeated patterns in the same day-of-week in surrounding weeks except for holidays and near holidays. For example, if a block of data is missing on Monday of 13th week of the year, the traffic during the missing block is likely similar to Mondays of 10th, 11th, 12th, 14th,15th and 16th weeks as long as the Monday is not holiday or near holiday. Based on these existing inferences, block missing data patterns are imputed using the following algorithm.

Algorithm 3: Block Level Nonnormal Bayesian Imputation

Step 1: Identify the beginning and end time of the block of missing data.
Step 2: Create an array of observed vectors using the same time block of the missing block on the same day-of-week from *M* previous weeks and *M* following weeks (*M* is usually a small number such as four or five), i.e., $B_{obs} = (B_{w_1}, B_{w_2}, ..., B_{w_{2M}})$ where B_{w_i} denotes the same time block of the volume data on the same weekday of previous or following weeks. If the same weekday of any of the chosen weeks is a holiday or near holiday, the data from that week is excluded.
Step 3: Using B_{0bs} draw *m* blocks by applying the NBI algorithm (Algorithm 1) and replace the missing block with the average of the *m* drawn blocks.

Again, the effectiveness of Algorithm 3 is illustrated using an example. Notice from Figure 5 that block of missing data (about six hours) was restored with high fidelity, which can be seen from the continuity of the data at the beginning and end of the day (or see another day like this one containing all “good” data).

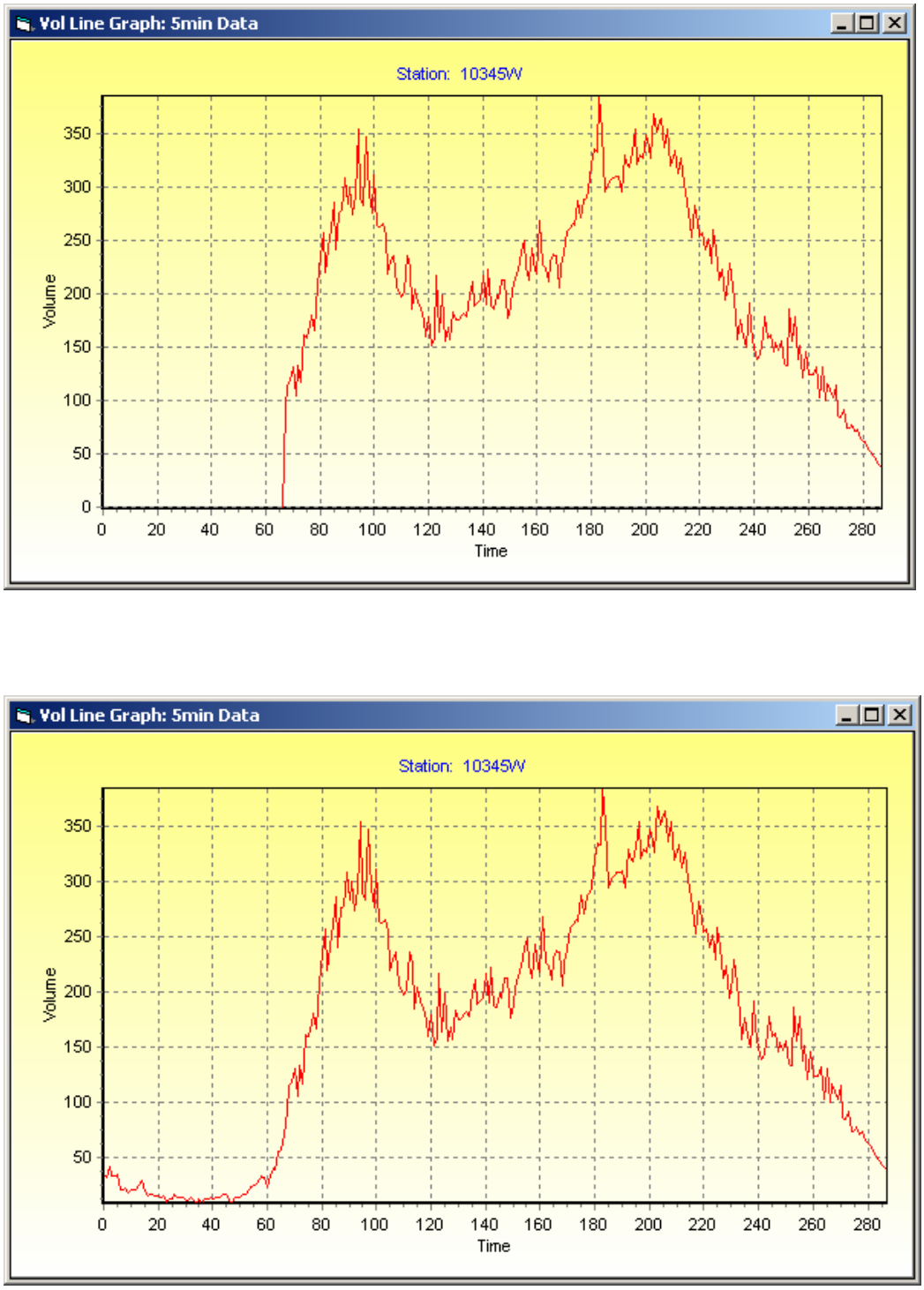


Figure 5: Effect of Block Imputation by Algorithm 3: top graph shows before block imputation and bottom graph shows after block imputation.

CHAPTER 4
IMPLEMENTATION

This chapter describes implementation details of this project for two purposes. First, it is intended to serve as a record on how the actual data were produced. Second, it is intended to clarify what has been done and what has not been done so future developments and modifications can be made with reference to this work. This chapter will also include a description of software tools that have been developed for this project.

4.1 Continuous Count Data

Continuous count data from ATR stations are simply ordered lists of hourly traffic volume counts that consists of 12 entries for AM and 12 entries for PM per day. Each hour's data represents the total volume count of a station during the corresponding hour, which is the total volume count of individual loops for that station in an hour. The data are recorded seven days a week throughout the year on a continuous basis.

4.1.1 Continuous-Count Data Source

Traffic data has been supplied by TMC to the Data Center of TDRL through an automated on-line daily uploading. These data sets contain a binary form of volume and occupancy collected at 30-second intervals from all detectors that TMC manages. For each detector, there are two files consisting of 2,880 elements. One file contains volume and the other, occupancy. Since TMC manages 3,500 to 4,000 detectors (it varies over time), the total number of data files per day is between 7,000 to 8,000. For exchange and archive, this large number of files is zip-compressed into a single file that is then transmitted to the TDRL Data Center. The compressed file size is typically 15 MB (Mega Bytes); when uncompressed its size becomes about 32MB.

4.1.2 Detectors in Continuous Count Stations

For continuous stations, three prioritized detector sets are always defined according the equivalency relation of traffic flow. These detector sets are referred to as

primary, secondary and tertiary detector sets denoting higher to lower priorities, respectively. In principle, the three detector sets must have the same amount of traffic flow in spatial relation. A lower priority detector set is used as an alternative detector set if the acceptance tests on the higher priority detector set fail. Detector identification numbers are expressed as either negative or positive integers. The positive numbers instruct the computing algorithm to add the detector volume to the station volume; the negative numbers instruct the algorithms to subtract the detector volume from the station volume. However, station volume must always be a non-negative integer.

4.1.3 Station Identification Database

The location of traffic measurement, that is the location of a station, may change over time. Likewise detectors assigned to a station may change. Since the detector locations and numbers as well as the stations themselves go through modifications, there is a need for a flexible means that could keep track of those changes, provide easy maintenance, and allow retrieval of any necessary combination of station information. For this purpose, a relation database (Microsoft SQL Database Engine™) was selected for the first choice of technology. This database was called the station identification database. During Task 1, this database was designed and developed to accommodate the required station management functions for both ATR and SC stations as well as for future applications such as a Geographical Information System (GIS). The database comprises two linked tables: Station Table and Detector Table (the detailed columns are shown in Appendix A). The Station Table maintains the geographical data, names, identification numbers and who and when modified. The Detector Table maintains all detectors allocated for all stations that are linked to the Station Table. Since the main users and custodians of this database are located at the central office of Mn/DOT in St. Paul, the database must be accessible from remote locations with good security measures. Therefore, a web interface that allows only indirect access to the database was developed. The tools used for this web application are a Zope server, Python language and Java Scripts. Example screen captures of these web interfaces are shown in Figures 6 - 8. The development of this part of the project was completed during the summer of 2002.

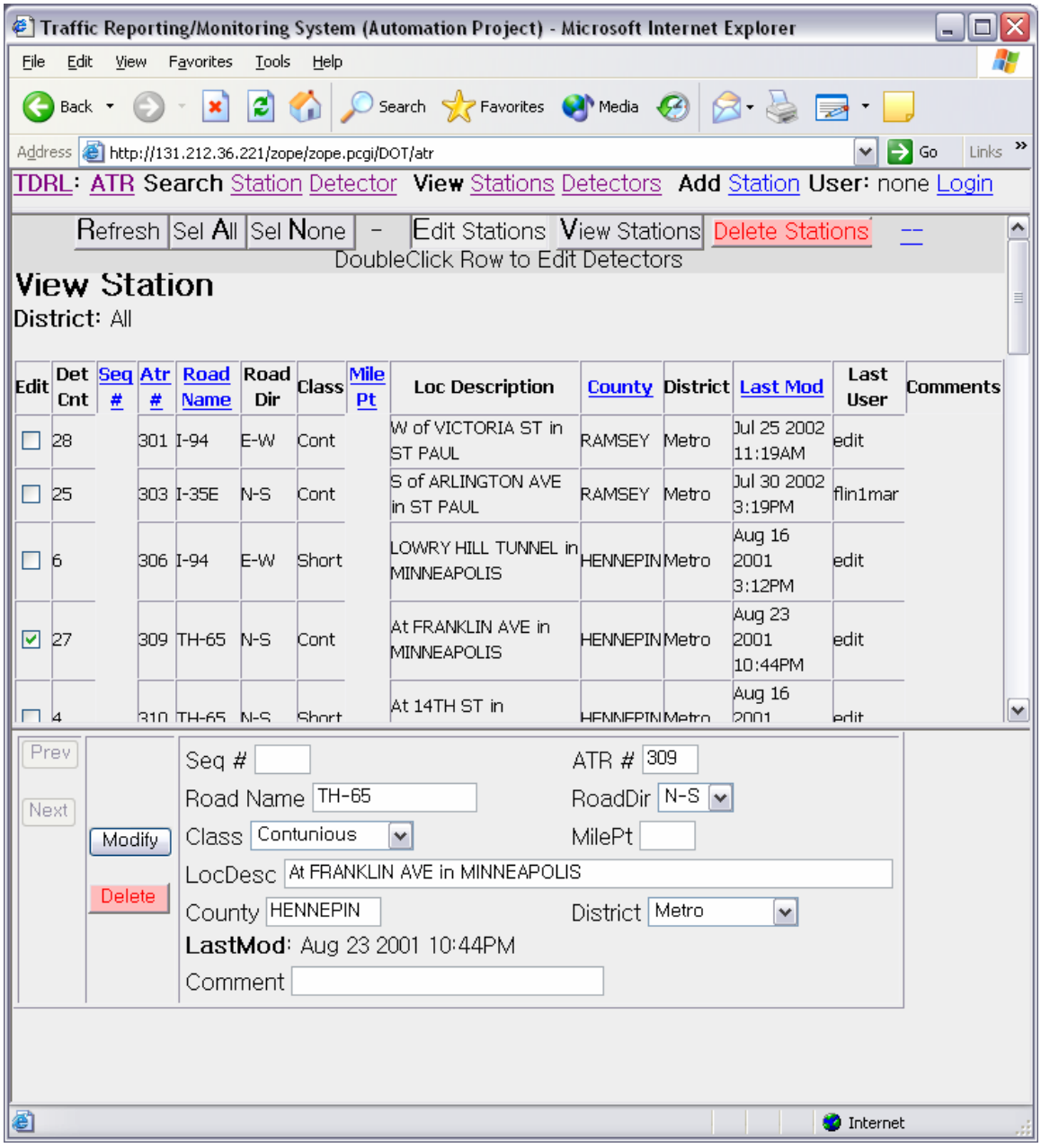


Figure 6: A sample screen capture of web interface: station table edit function

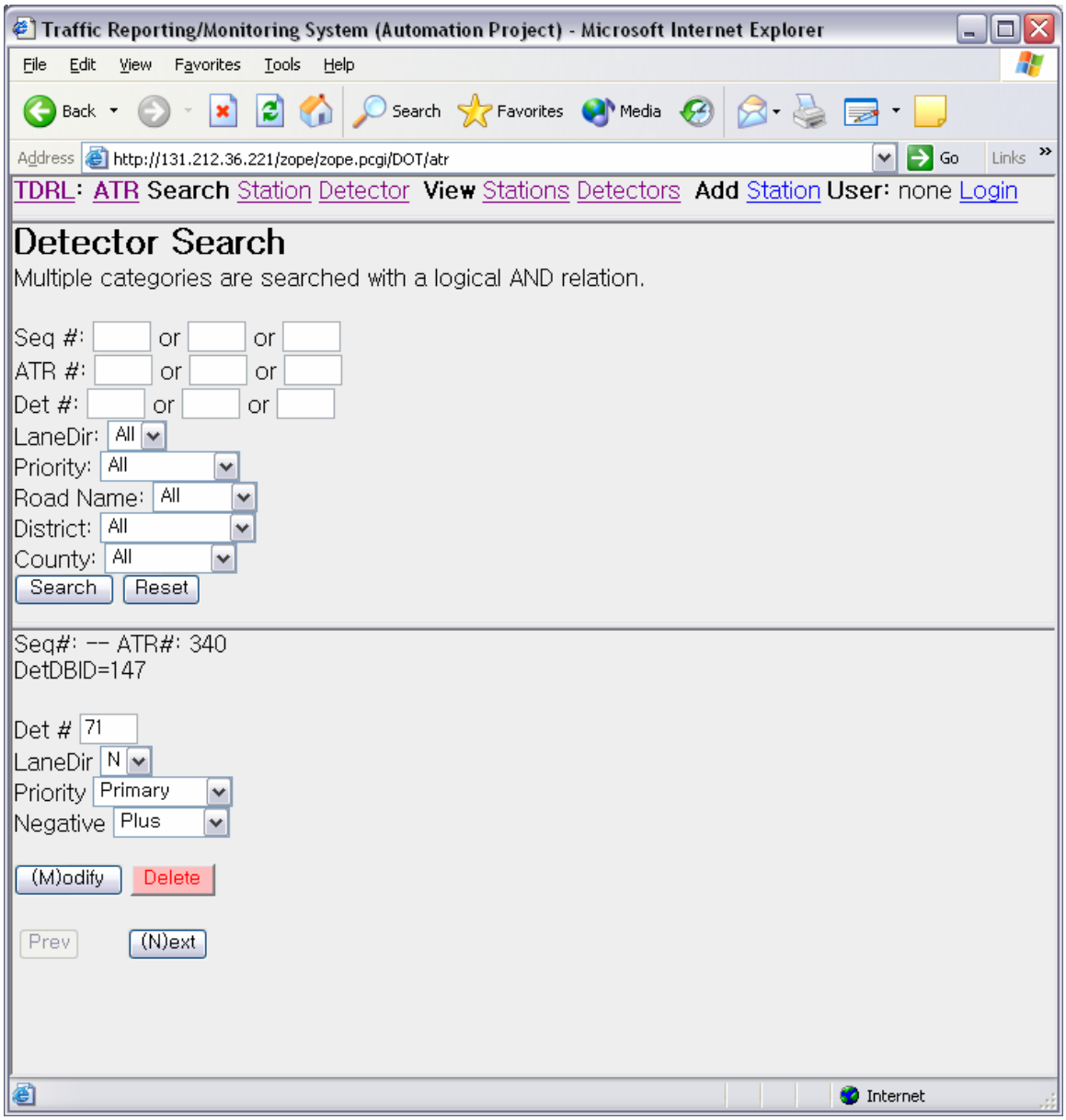


Figure 7: Sample screen capture of web interface: Detector edit

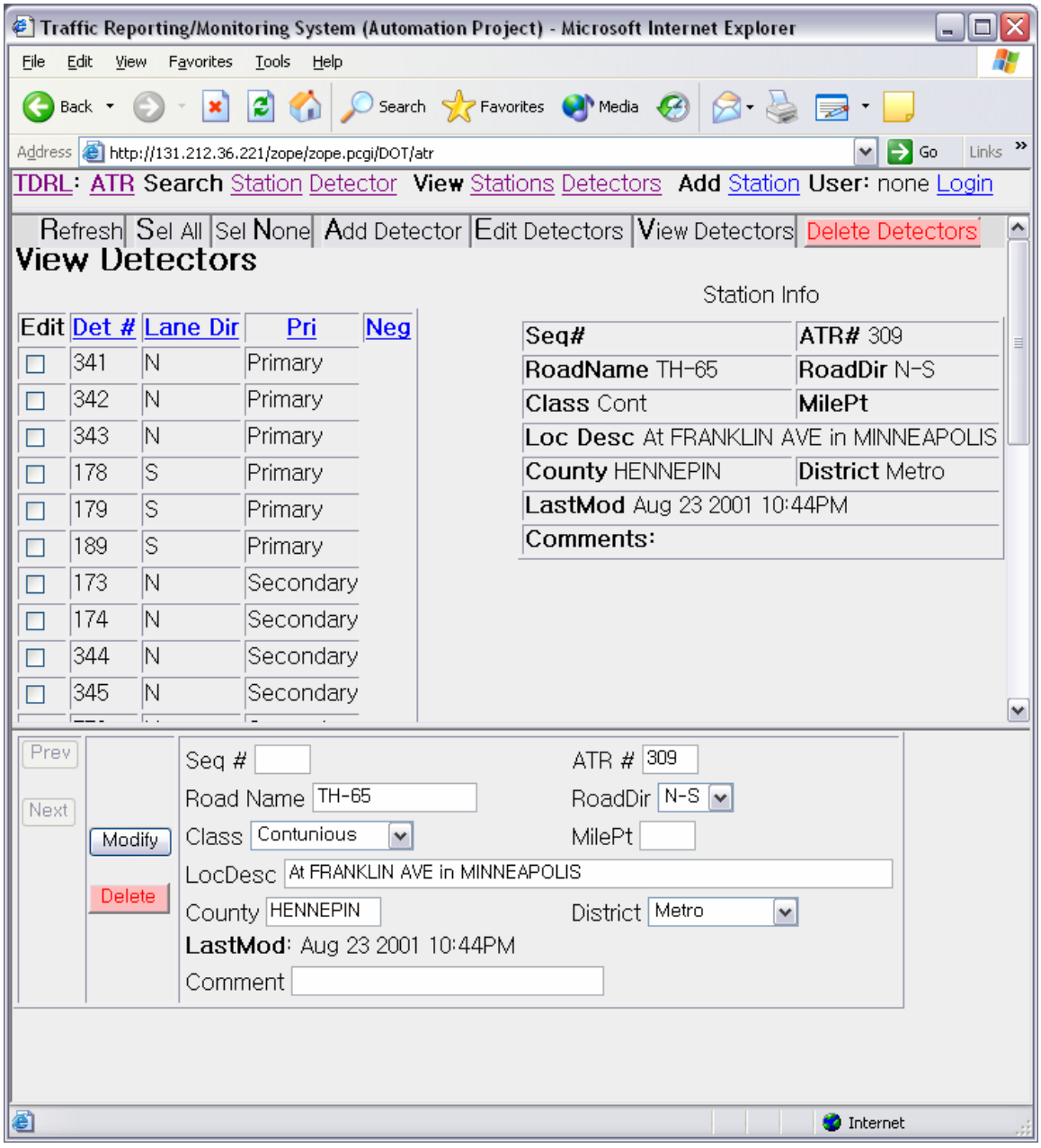


Figure 8: Sample screen capture of web interface: Selected station and detector view

Although this web application was developed with sophisticated web programming techniques and easy to use graphical user interfaces, in the final analysis it was concluded that it created additional burdens to the Mn/DOT users in training and learning and to the TDRL developers for user and database maintenance. This clearly goes against our initial spirit of creating simple and easy to maintain automated system. Therefore, for future version of this system, Mn/DOT and TDRL decided to develop a

simple ftp based exchange of formatted text for the maintenance of station and detector information. This modification was planned to be carried out during the 2003-2004 fiscal year.

4.1.4 Data Format

The final output of continuous count data is formatted according to the existing SAS application input requirement for internal screening by Mn/DOT. All characters in the data file are ASCII characters. Each line contains a half-day of data for a station in one direction. Thus a full one-day amount of data in one direction occupies two lines: 12 hours per each row corresponding to AM and PM of the day. The field of each line and the digit positions are summarized in the table below. Considering two directions in most roads, a complete data for a single station per day would occupy four lines of data. This format was determined during the years that data were aggregated manually with the assistance of spreadsheets.

Digit Position	**Number of Digits**	**Description**
2	1	AM=1, PM=2
3-4	2	Month, 1-12
5-6	2	Day of the month, 1-31
7-8	2	Year
9	1	Day of the Week: Sun=1, Mon=2, Tue=3, Wed=4, Thu=5, Fri=6, Sat=7
10-12	3	Station ID*
13	1	Lane direction of the station, E,W,S,N,R
14-73	60	A set of five digits represents the hourly volume. Twelve of five digit sets (12*5=60) are consecutively concatenated in the order representing 1st to 12th hour depending on AM or PM.

* Presently, ATR ID is used as a Station ID.

Below two rows of data was taken from top two rows of a sample file."

```
210131002301E00662004980030900235002760089703106058400577204091038880421 7
220131002301E04678048380567206988071240657605002033480298203326021790149 7
210131002301W00631004260030000324005830230105530068960692805005044130456 5
220131002301W04565047570541505826066410684704897029360252802314018480107 3
```

As an example, the interpretation of the first line from the above data is illustrated in the following table:

Digit Position	Value	Meaning
2	1	AM
3-4	01	January
5-6	31	31st day
7-8	00	Year 2000
9	2	Day of the Week: Monday
10-12	301	Station ID = ATR ID
13	E	Lane Direction, East
14-73	00662 …	A set of five digits represents the hourly volume. Twelve four digits are consecutively listed

4.1.5 Log File Data Format

Along with the data file, a log file was produced to document missing data statistics and the choice of which detector set the algorithm selected. The log file consists of text readable ASCII codes, and they are mostly self-explanatory. The first line records the information on which day's data on what day it was processed, then it proceeds with station by station reporting of information on missing percent, directional volume differences, missing detector file, the hourly choice of detector sets and the hourly missing percent. The hourly choice of detector sets are denoted as: P=Primary, S=Secondary, and T=Tertiary. A sample log file can be found in Appendix B.

4.1.6 File Name Convention

Since the reported data are delivered to Mn/DOT electronically, a consistent file name convention was developed to denote which data type and the day of year it represents. For daily continuous count data and log files, a prefix "ATR" along with the date of the data is used as the file name.

File name format for daily ATR data: ATRyyyymmdd.dat

where yyyy denotes 4 digit year; mm denotes 2 digit month; and dd denotes 2 digit day. For example, for Feb 6, 2000, the data file should have the name “ATR20000206.dat” and the log file “ATR20000206.log.”

To reduce the number of files, data sets are often packaged into a single file that may contain one or more weeks of data. The weekly data file name is denoted by appending a letter “w” followed by a numeric number that represents the number of weeks contained in that file. The date in the file name then represents the ending date of the data (mostly Sunday). A week is defined by seven days starting from Monday and ending after the final hour of Sunday. The following example further illustrates the name convention.

Example:	ATR20020206w1.dat	One week of data ending Feb 6, 2002.
	ATR20020206w1.log	Log file for ATR20020206w1.dat
	ATR20020206w2.dat	Two weeks of data ending Feb 6, 2002.
	ATR20020206w2.log	Log file for ATR20020206w2.dat

4.1.7 Software Developed

The software that computes the continuous count data was written in Microsoft Visual Basic with a few ActiveX tools. The code was relatively complex because it must handle unzipping, network file-transfer coordination, relational database access through network, missing file handling, calendar functions and the scheduled runs. However, the user interface is extremely simple as shown in Figure 7.

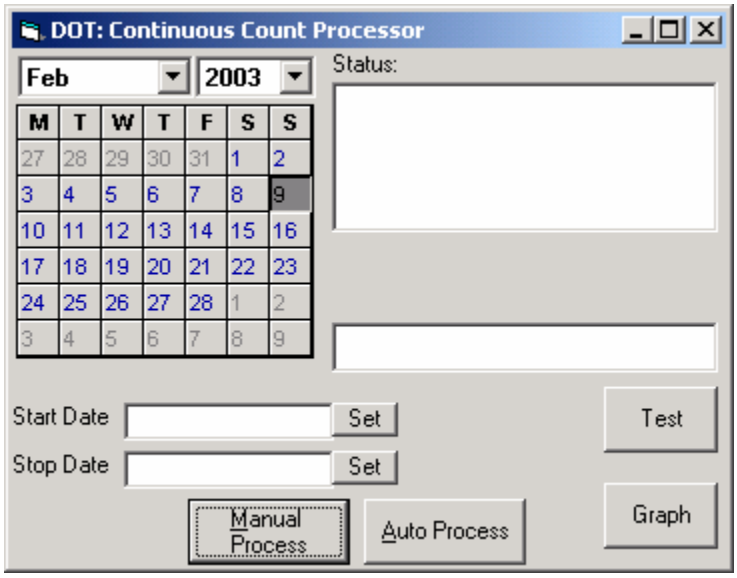


Figure 9 : Short duration count computation software

The code may run manually by entering the start and stop date or automatically by a scheduler. For manual entry, the dates should be entered by clicking the calendar buttons instead of typing to help eliminate typographical errors. However, a manual run should be used only if an error condition requires human intervention. During an automated run, the internal registry keeps track of which date was last completed so that it automatically determines which week to run next. It is presently scheduled to run daily to check whether the data from TMC arrived. If it finds enough data, then the computation process is activated.

An additional piece of utility software that can read and analyze the ATR data was developed since the ATR data in the form defined in Section 4.1.4 is hard to read. This software tool is named "ATRViewer" and includes the following functions:

- Reads multiple weeks of data
- Displays various forms of hourly graphs (line, bar, area, point)
- Calculates statistics
- Computes and plots daily volume
- Graphs hourly color grids and histogram
- Exports to an Excel file

- Converts and loads from binary source file

Figures 10-15 shows a sample screen of the functions listed above. Although the examples shown display only one week's worth of data, it can read an unlimited amount of data (such as a whole year) and can create the same plots and statistics.

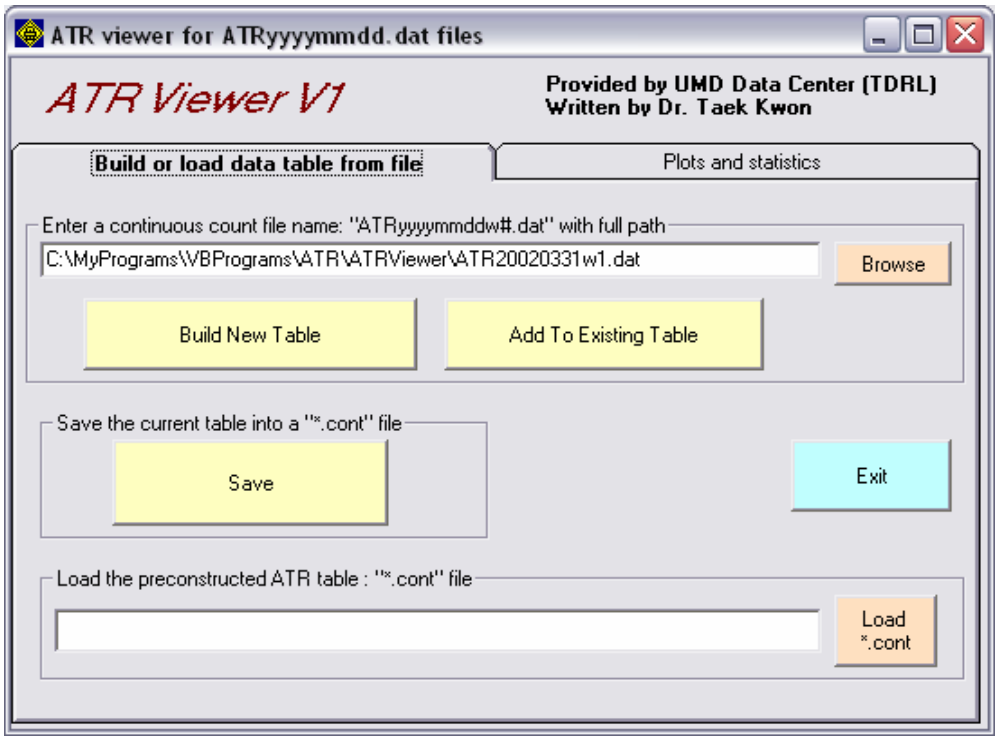


Figure 10: Screen Capture of the ATR Viewer Program

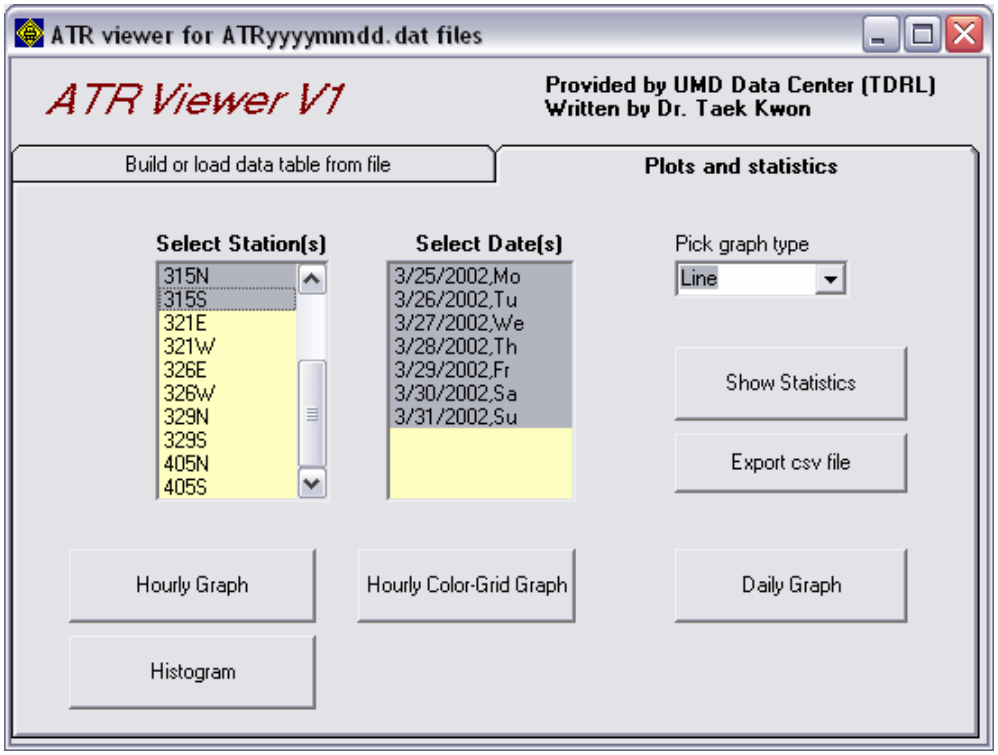


Figure 11: Plot and statistics tab

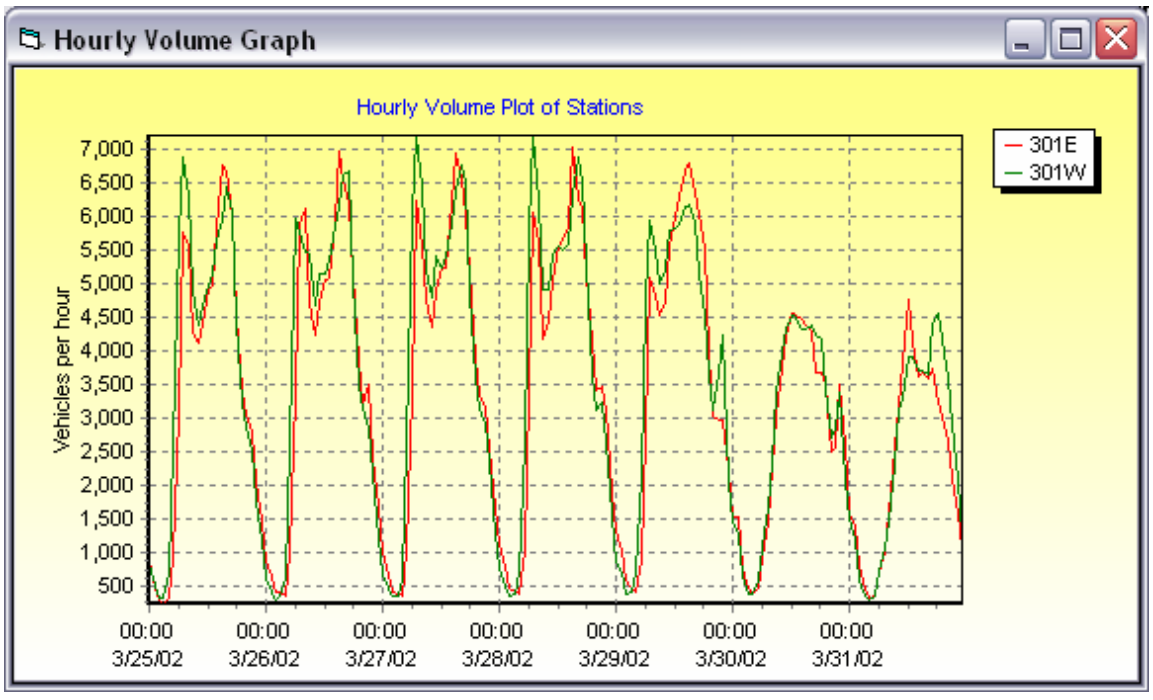


Figure 12: Line plot example

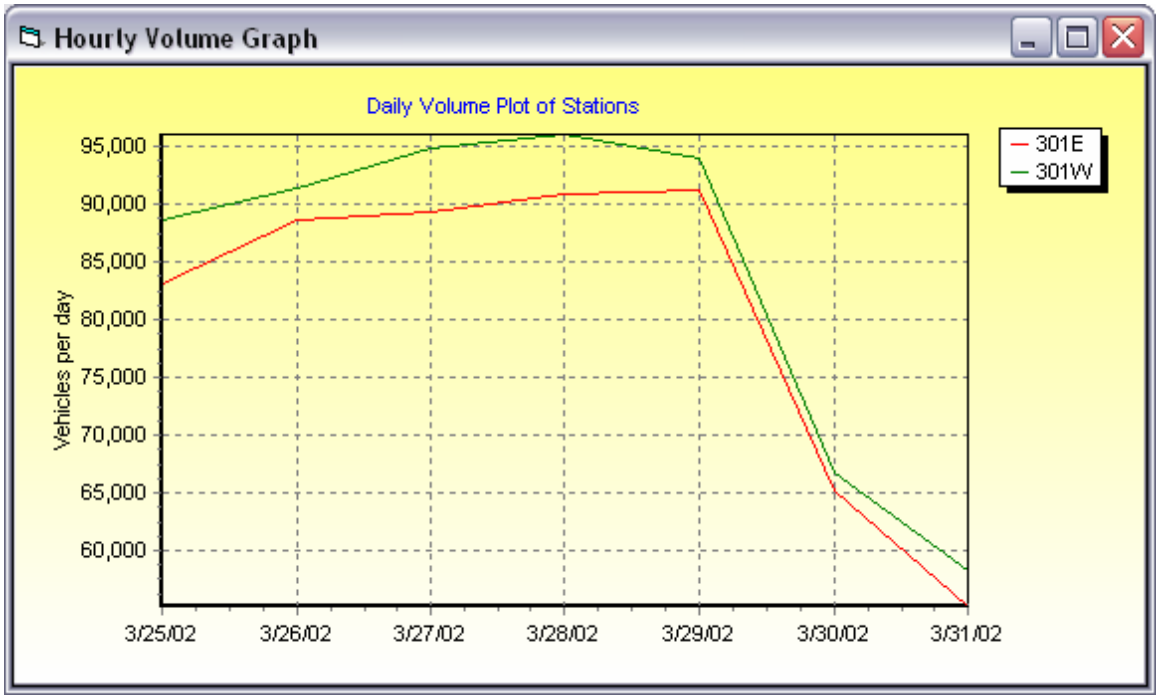


Figure 13: Daily volume plot example

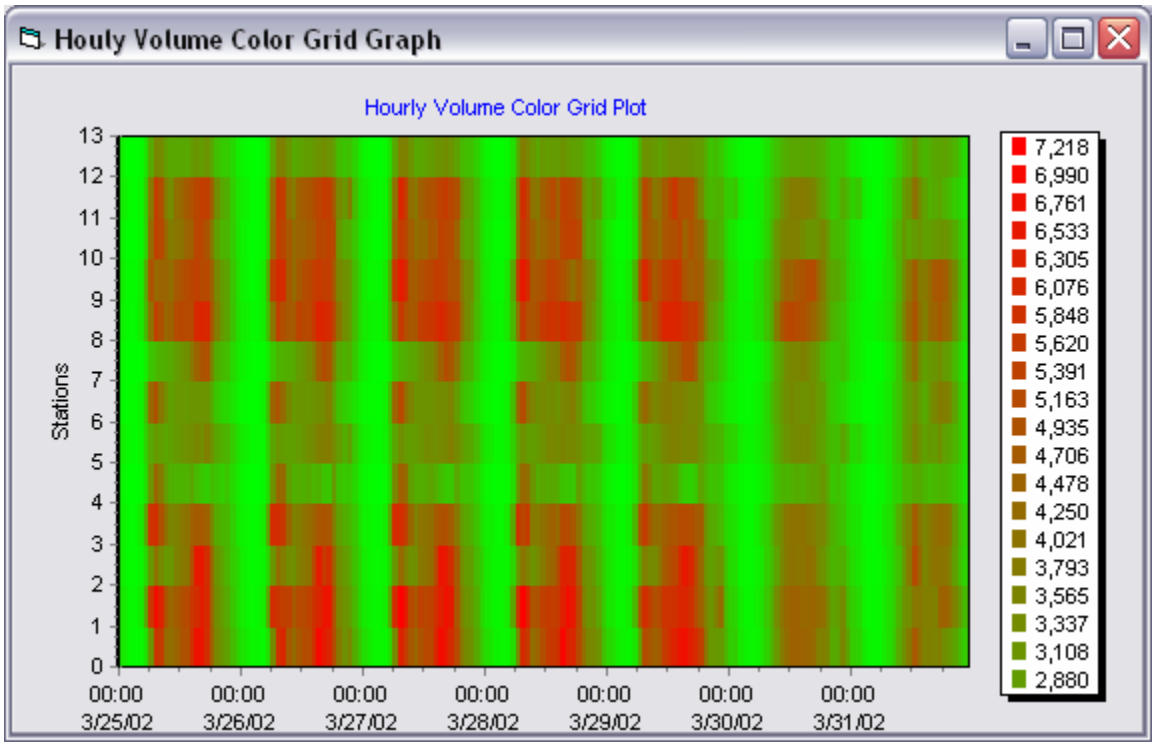
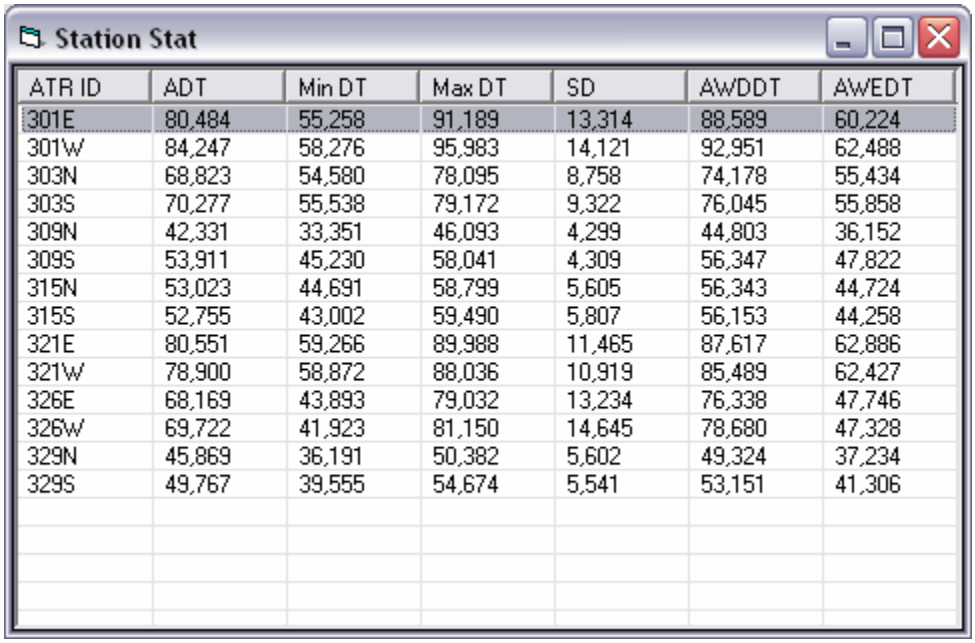


Figure 14: Hourly color grid example: Morning and afternoon high traffic times can be observed.



ATR ID	ADT	Min DT	Max DT	SD	AWDDT	AWEDT
301E	80,484	55,258	91,189	13,314	88,589	60,224
301W	84,247	58,276	95,983	14,121	92,951	62,488
303N	68,823	54,580	78,095	8,758	74,178	55,434
303S	70,277	55,538	79,172	9,322	76,045	55,858
309N	42,331	33,351	46,093	4,299	44,803	36,152
309S	53,911	45,230	58,041	4,309	56,347	47,822
315N	53,023	44,691	58,799	5,605	56,343	44,724
315S	52,755	43,002	59,490	5,807	56,153	44,258
321E	80,551	59,266	89,988	11,465	87,617	62,886
321W	78,900	58,872	88,036	10,919	85,489	62,427
326E	68,169	43,893	79,032	13,234	76,338	47,746
326W	69,722	41,923	81,150	14,645	78,680	47,328
329N	45,869	36,191	50,382	5,602	49,324	37,234
329S	49,767	39,555	54,674	5,541	53,151	41,306

Figure 15: Statistics of the selected period: ADT (average daily traffic), Min DT (minimum daily traffic), Max (maximum daily traffic), SD (standard deviation), AWDDT (average weekday daily traffic), AWEDT (average weekend daily traffic)

4.2 Short-Duration Count Data

This subsection describes the present version of AADT computation implemented for short-duration count stations.

4.2.1 Traditional Definition of Short-Duration Count In Mn/DOT

Short-duration count (SC) of a station is defined as a 24-hour (noon to noon) volume average computed over qualified three consecutive days, two 24 hour periods of noon to noon resulting 48 hours. In any given week, three qualified 48-hour periods are selected as from Monday noon to Wednesday noon (this period is denoted by the middle-date, Tuesday), from Tuesday noon to Thursday noon (middle-date=Wednesday), and from Wednesday noon to Friday noon (middle-date=Thursday). The qualified pool of dates for the short-duration count is typically selected from the period between *April 1 and November 1***.** During this period, dates with holidays, near holidays, detour, incidents, severe weather, and special events are excluded from the qualified pool of days to avoid any severe deviation from normal traffic patterns. These choices are made essentially to obtain a typical daily traffic count for a week day, from which the station's AADT can be

estimated by seasonal/day-of-week adjustments. The adjustment factors for seasonal and day-of-week are derived from the ATR data on the same road or clustered ATRs exhibiting similar traffic characteristics as the SC station.

4.2.2 AADT Computation of SC Stations from ITS Traffic Data

Traffic data at a selected location is traditionally collected using portable vehicle counting devices such as pneumatic tubes by sampling typical days. On the other hand ITS traffic data is typically collected using pavement imbedded inductive loop detectors (not portable) at a much higher data-sampling rate (typically 30 seconds of samples) for real-time traffic monitoring and control. Also, the data is collected seven days a week, all year round. Therefore, it makes more sense to use the entire set of available data than to use a sampled set of just a few days as it was traditionally done for the computation of AADT using ITS traffic data. Based on this reasoning, the original task was modified to directly compute AADT from the TMC traffic data. Unfortunately, like other ITS traffic data, TMC data contains many missing values. If the amount of missing data is small, it does not present too much difficulty since they can be readily imputed. However, if the size of a missing data block is very large and thus only few good days are available for the entire year, imputation is more challenging.

4.2.3 New Station Definition Text Format

Although development and implementation of a relational database for managing detectors and stations (described in Section 4.1.3) has been completed, the final analysis indicates that the use of a simple text file is better for the personnel in Mn/DOT as discussed before. A simple text format for the SC station definition has been developed, which will be extended to the ATR stations in the future.

The line entry of the form has the following format:

StationID, DirCode, P, detP1, detP2, …, S, detS1,detS2,…, T, detT1,detT2, …, End

StationID is a unique identification number assigned to each station by Mn/DOT. DirCode denotes a direction code of a station that is determined by the direction of the

road where the station is located. This code is represented by a number between 1 and 8 where the codes are defined in clockwise direction, i.e., 1=N, 2=NE, 3=E, 4=SE, 5=S, 6=SW, 7=W, 8=NW, 0= “All other directions such as reversible or both”. The rest of fields can be more easily explained by an example. Let us define a station 321 direction 7 (west) with primary detectors at 843, 844 and 845, secondary detectors at 854, 855, 856, and 857, and tertiary detectors at 826, 827, 828 and 830. The line entry for this station would be written as:

```
321,7,P,843,844,845,S,854,855,856,857,T,826,827,828,830,End
```

Comments may be attached after the End statement or any line starting with the character “;” or the line left blank for legibility. The detailed editing rules and file name conventions are shown in Appendix C.

This format was developed to support a simple parser for programming purposes as well as for creating a human readable text format. Both objectives were accomplished and the present SC computing was implemented based on parsing of the detector lists written in this new format. The result was a significant performance increase against the database approach since the SC software no longer had to access the database to request the detector list each time it computed an AADT for a station. An ftp site has already been established from which Mn/DOT analysts can upload the station definition files. When the TDRL SC software is activated it runs using the most recent station definition file available from the ftp site.

4.2.4 Short-Duration Count Data Format

After computing the AADT for SC stations the program produces a final output following a certain format that can be directly fed into the Mn/DOT’s TMS database. The format presently accepted by Mn/DOT is:

StationID, DirCode, EndingDate, AADT, “ValDays TMC”

The number of columns allocated for each field is:

7 columns, 2 columns, 11 columns, 7 columns, "Minimum 7 columns"

All fields are separated by a comma and are right justified. Null data is left as blank. The number of columns indicated for each data field includes spaces but excludes the separating comma. The meanings of the fields are:

- *StationID*: a unique sequence number defined for the station
- *DirCode*: direction code
- *EndingDate*: ending date of AADT computation period. Usually it is usually12/31/yyyy, but it can be also 10/31/yyyy, for example, if AADT was computed between 11/01/2001 – 10/31/2002, the ending date is 10/31/2002.
- *AADT*: Annual Average Daily Traffic (AADT) volume counts computed for one year ended by the *EndingDate*
- *ValDays*: the number of days that had useable and valid data for the AADT computing duration.
- *TMC*: It is a string constant that indicates "It was computed from the TMC data"
- "…": This field is a commenting area and is incorporated into the TMS database and shows up on analyst reports.

A sample data is shown in Figure 16. The resulting file name follows the format ADTSampleyyyy.txt where yyyy is the year of AADT.

```
 10069, 5, 06/26/2002,  41210, " 21 TMC"
 10182, 3, 10/31/2002,  37170, "350 TMC"
 10182, 7, 10/31/2002,  37095, "350 TMC"
 10286, 1, 10/31/2002,  44310, "322 TMC"
 10286, 5, 10/31/2002,  45375, "315 TMC"
 10287, 1, 10/31/2002,  41600, "329 TMC"
 10287, 5, 10/31/2002,  43332, "161 TMC"
 10288, 3, 10/31/2002,  33969, "294 TMC"
 10288, 7, 10/31/2002,  35226, "294 TMC"
```

Figure 16 : Sample AADT data formatted according to Mn/DOT specification

4.2.5 Detection of Missing and Incorrect Volume Counts

Before the imputation algorithm is implemented, the first step required is identification of missing and incorrect values. These missing data and incorrect values become candidates for imputation.

When a TMC traffic file is unzipped, it produces daily volume and occupancy files, each of which contains 2,880 values representing 30-sec samples of a single detector for a single day. In the data, all hardware errors are already flagged as a negative value during the data packaging process. These negative values become missing values in our algorithm. In addition, any volume counts greater than 39 per 30-second period are considered as incorrect values and are treated as missing values since such values are physically impossible. Yet another type of values screened are consecutive repeating values. In traffic data, there is a high probability of repeating 0 or 1 (or low number) during the low traffic hours such as 2:00 – 5:00 AM. However, the repeating is less likely to appear during the high traffic hours. Repeating of high numbers such as a number greater than 10 is highly unlikely to appear during any time of the day. In general, the probability that repeated numbers appears in a daily detector file diminishes as the volume count becomes larger.

Based on this principle, we can construct a probability model for the detection of incorrect data. Theoretically, its distribution should follow a Poisson distribution. However, it was not clearly observed in the real data. A simple but practical rule of detecting repeated values was established as follows. Repeated zeros or ones are considered normal during the low volume hours 2:00 – 5:00 AM. During any other period, if repeated values are observed more than four hours, it is considered as incorrect data and replaced with imputed values.

In addition to the repeating value problem, there are other types of incorrect count values that exist in loop data. When the threshold of loop detector sensitivity is set to a wrong value, volume counts can be too high or too low. Very often mutual coupling causes over counting due to detection of adjacent lanes. In general, undercount or over-count problems are extremely difficult to detect just from the loop data alone. In this

project, no attempts have been made to detect or correct over- or under- count problems in loops.

4.2.6 Implementation of Imputation

The basic premise of the overall imputation algorithm developed in this project was that missing data patterns (types classified in Section 3.2.2) supply recursive inferences to the next level as imputation moves from the Type-A missing data patterns and progress toward the Type-F missing data patterns. For example, after imputation of Type-A and Type-B missing data patterns there will be less Type-C missing data patterns, therefore more inferences are available for imputing Type-C missing data patterns, which would result in imputation with more information. The overall data processing cycle is implemented beginning with the proper identification of missing data types and then applying corresponding imputation algorithms. Figure 17 illustrates the steps implemented through a block diagram.

The imputation process starts with treating the detector-level random missing data cases, i.e., Type-A missing data patterns as shown in Figure 17. Since Type-A patterns are a class of random missing data patterns, the NBLR algorithm described in Section 3.4.2 was used for imputation. After extensive experiments, it was determined that up to 16 consecutive missing values of 30-second data can be effectively imputed using the NBLR algorithm. In the overall processing, Type B missing data patterns were not imputed since they are eventually imputed during the process of Type-C and D patterns.

After imputation of Type-A missing data patterns, the detector data was converted into station data with 5- minute interval. This was necessary to create a smaller memory requirement, so that a whole year of data could be loaded into the computer RAM and processed. Without this conversion, about 10 GB (giga bytes) of data must be loaded into RAM to process one year of data. Such a large memory-capacity is not presently available from the computers at the TDRL data center. Type-C missing data patterns were determined by less than six consecutive missing data points, which would correspond to 30 minutes. However, for future implementations 12 consecutive missing data points that correspond to one hour is recommended for Type-C missing patterns since 5 minute data can easily infer the time trend up to one hour. Imputation of Type-C

patterns was implemented using the NBLR algorithm since Type-C patterns are random missing data patterns at the station level.

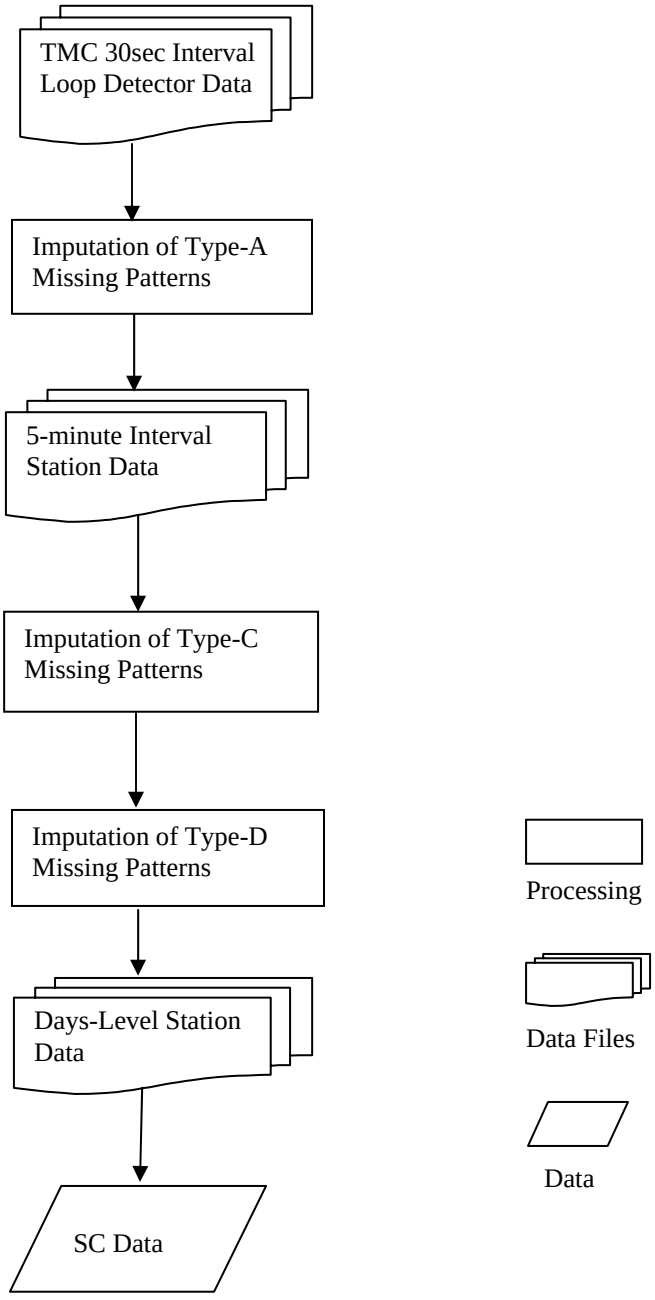


Figure 17: Block Diagram of Imputation Steps Implemented

Upon completion of Type-C imputation, block level imputations are applied to Type-D missing data patterns. Type-D missing data patterns were determined when the size of missing block is less than 60% of the day. The Algorithm 3 (Block Level Nonormal Bayesian Imputation) discussed in Section 3.4.3 was used for imputing the Type-D missing data patterns.

As shown in Figure 17, after completion of Type-D imputation day-level station data was produced for several purposes: for the final computation of AADT, days-level imputations, and also to provide a type of data similar to the ATR stations. In the present state of implementation (at the time of this writing), imputations up to Type-D missing data patterns were completed, but imputation of Type-E and F missing data patterns have not been implemented and left for future study due to limited time and the need for further studies.

4.2.7 Software Developed

The imputation software written implements the processing steps of the block diagram shown in Figure 17. This software does not have any interesting user interface since it was written for scheduled runs without manual commands or user interventions. However, an important utility tool that allows analysis of days-level data for any selected year has been developed for Mn/DOT analysts and is described in this section.

The days-level station data in Figure 17 contains information on daily traffic between the start date and end date of the year defined in a self-descriptive form. This effort was made to make the data file transportable between different operating systems and computer systems. The days-level station data is organized in the following order:

(1) a “magic code” used for software confirmation and identification of the days-level data type; the code is 56789yyyy where yyyy is the year of the ending date, e.g., for 2003 the magic code is stored as 567892003. (4 byte long integer)

(2) starting date of the year (8 byte date type)

(3) ending date of the year (8 byte date type)

(4) number of stations (2 byte integer)

(5) number of days (2 byte integer)

(6) an array of station numbers (4 byte array)

(7) direction code for the stations (2 byte integer array)

(8) two dimensional long (4 bytes) array of daily traffic count of the year; an array element x(i,j) represents total volume of ith station on jth day.

This days-level station data file is produced for every year as the SC automation program finishes the run. A tool utilizing this data named "Daily Traffic Data Analyzer", has been developed and is illustrated in Figure 18. Upon loading of data using this software tool, it immediately computes AADT (Average Annual Daily Traffic), AWDDT (Average Weekday Daily Traffic), AWEDT (Average Weekend Daily Traffic), PDT (Peak Daily Traffic of the Year), ValDays (number of valid days), SDs (Standard Deviations), and number of outliers. Users can select or double click on any of the stations available to see the graphs of daily traffic for the entire year. An example graph is shown in Figure 19. The graphs can be zoomed in or out to see the details by dragging a mouse on the region of interest. A zoomed example is shown in Figure 20. Using this graph, analysts can see how the traffic has been changed during the year or what the trends are. For example, one can see that Mondays have least traffic while Fridays have the most traffic during weekdays. Another trend that can be noticed is that, during the summer months, traffic increases, but in December traffic is significantly decreased.

Using this tool, the users can also see the actual data by clicking the left axis and by selecting the data tab. The data screen example is shown in Figure 21. In addition, there are a number of graphical editing tools that are available for users which are not described here but can be easily learned by playing with the software. According to Mn/DOT analyst's comments, "This tool was an invaluable tool" for their traffic analysis. This tool is presently available for download from the TDRL web site (http://tdrl1.d.umn.edu). In the future, a Geographical Information System (GIS) tool that implements a traffic map will be developed to provide spatial map of daily traffic counts. This GIS application will be available as a web tool, such that the users can easily see the traffic estimates.

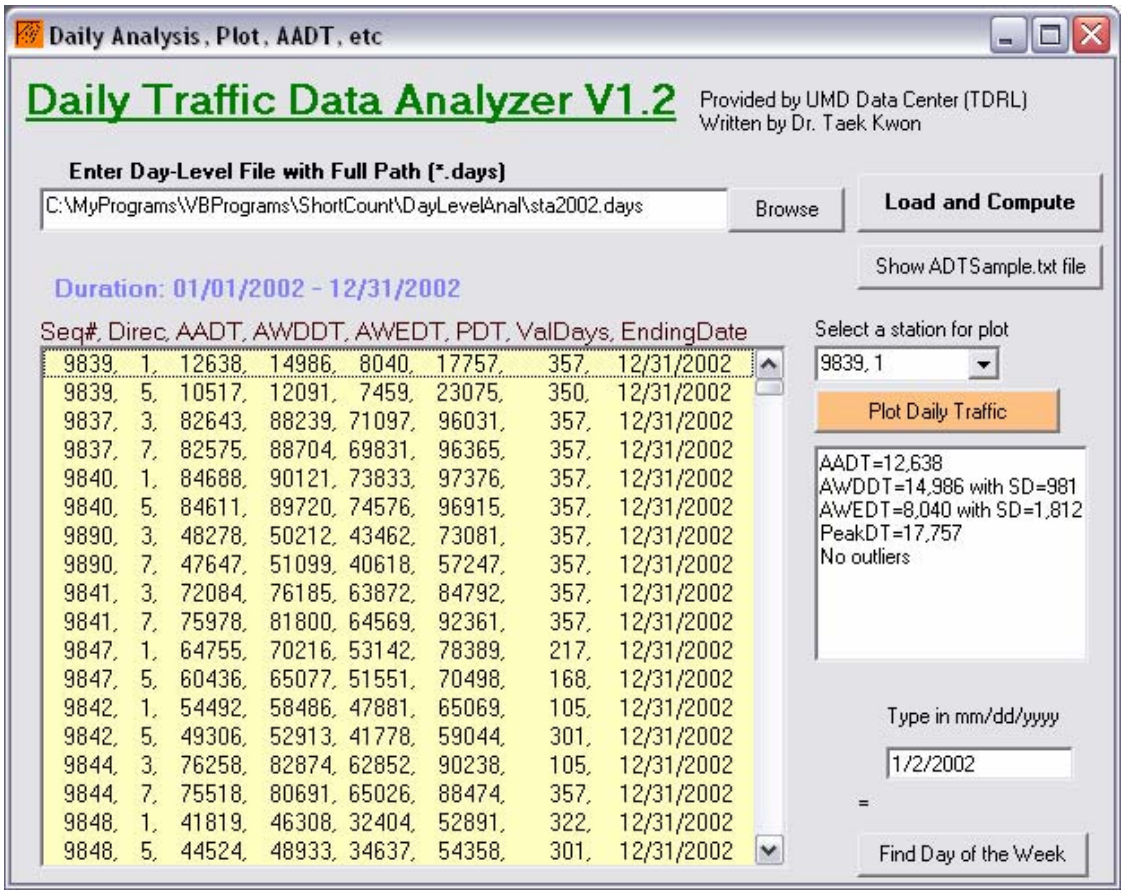


Figure 18: A sample screen of Daily Traffic Data Analyzer

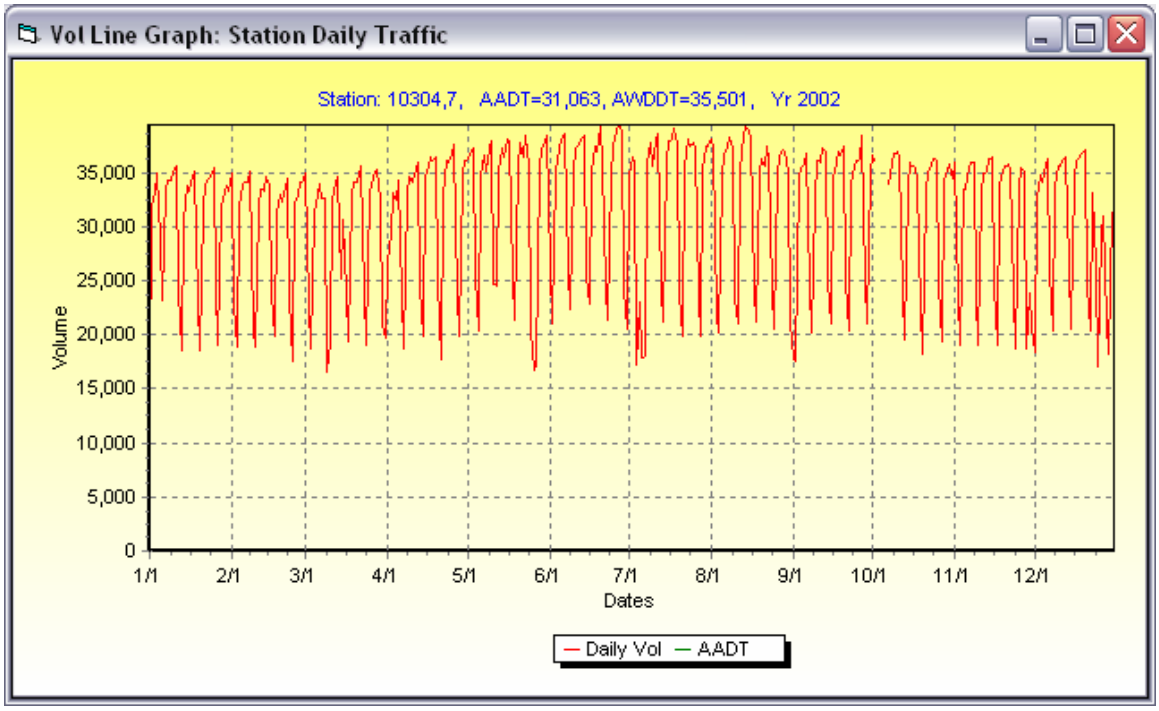


Figure 19: A sample graph of daily traffic, station 10304, NE, year 2002.

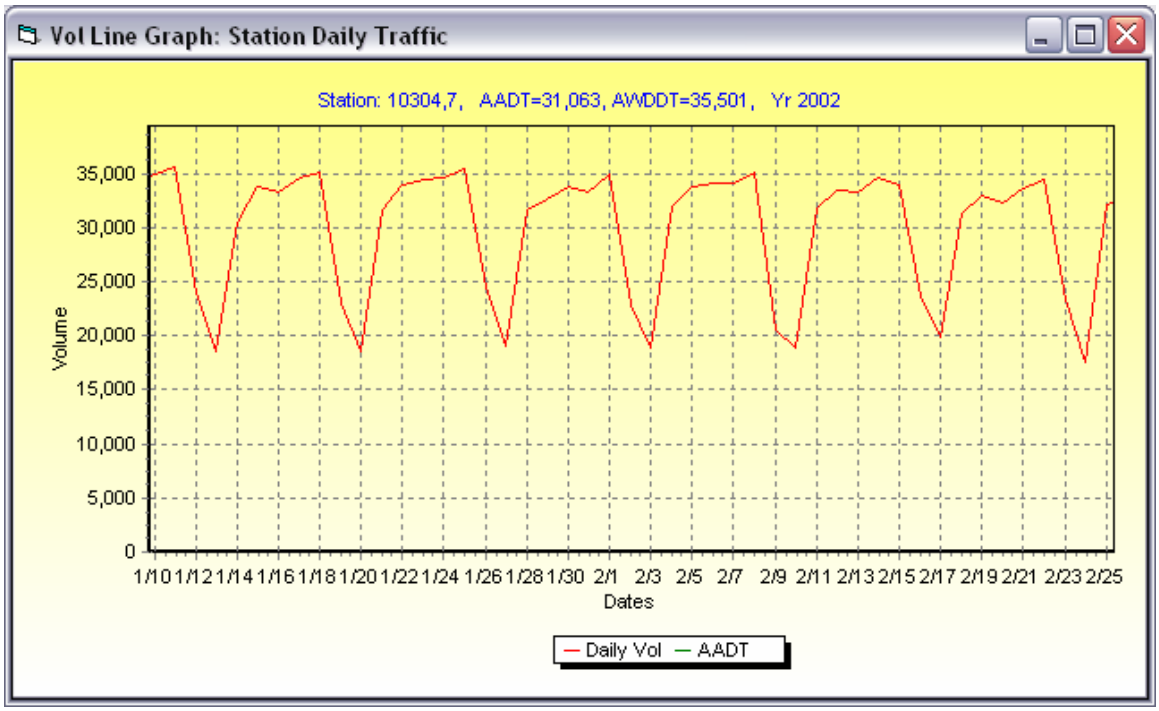


Figure 20: A zoomed in graph of Figure 19 by dragging a mouse on the region of interest.

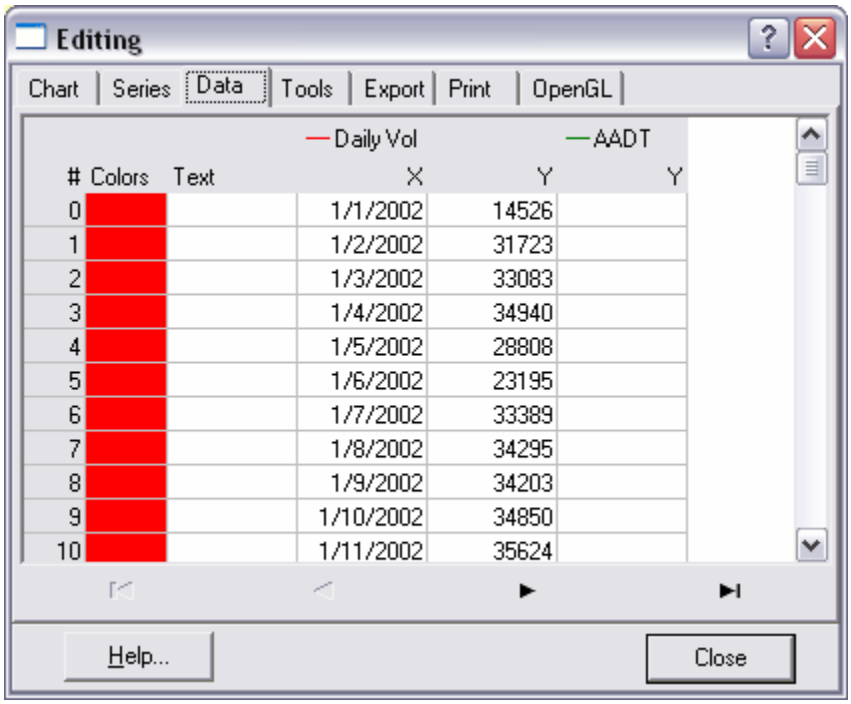


Figure 21: Graph editing tool that allows to see the actual data as well as various editing functions of the graph.

CHAPTER 5
CONCLUSION AND FUTURE WORK

This project was started with a goal of automating the continuous and short-duration count data for the portion of TMC traffic data. In the process, one important issue which the research team spent the most amount of time on was how to deal with the missing and incorrect data that exist in the TMC traffic data. First, all identifiable incorrect data points were simply treated as missing data since we do not know the amount of incorrectness. Thus the problem was simplified to dealing with only missing data. The finding is that missing data can be effectively imputed with the values that are very close to the real values if we utilize observable spatial and temporal relations of traffic flow. For utilization of spatial relation, this project introduced multiple redundant sets of detectors that are defined (or allocated) for each station by locating the detector sets that have an equivalency relation in terms of traffic flow. Since this approach is essentially equivalent to creating redundancy in data through availability of additional data from the vicinity of the primary detectors, it enabled replacement of missing data from the equivalent set of data. It should be noted that the use of spatial redundancy was possible because the TMC traffic sensor network was densely implemented (loops were installed at every 0.5 miles), which is one of the advantages of using ITS generated traffic data for traffic counting program. This strategy significantly improved the data quality by reducing the number of missing data when three sets of detectors (primary, secondary and tertiary) were assigned.

However, this approach alone did not produce the desired “completely healthy” traffic data that do not contain any missing data points. In some cases, all three sets of detectors contained missing data within the same time span. Therefore, there is a need for additional means to treat the missing data. Another important relation of traffic flow that was utilized was temporal relation of traffic flow. It was found that, except for the days with special events or holidays and near holidays, traffic patterns tend to repeat during the same day of week. Since this repetition does not occur precisely but in a statistical sense, algorithms that utilize Bayesian approach of quantifying uncertainty in temporal

inferences (trends in this case) based on statistical properties of the data were developed. These algorithms successfully imputed the missing values when temporal inferences are available. In summary, the problems of missing data in ITS generated traffic data could be overcome through imputation based on temporal and spatial inferences that exist in the traffic flow.

Another finding from this project was that conventional way of using 48 hours or 72 hours of representative samples as a short-duration count and then adjusting it for AADT is no longer necessary for ITS generated traffic data. After spatial and temporal imputation of data, an ample amount of data was available for directly computing AADT and other summary statistics of traffic volume. Thus, the initial project tasks were modified to directly compute AADT rather than selecting short-duration samples.

There are some outstanding issues that require further exploration. One of them is estimating AADT when stations have absolutely no data for the entire year. It occurs in about 10 to 15 stations out of 483 stations every year. Since such stations do not have any data for imputation based on temporal inference, spatial inference such as the locations having equivalent traffic demand is the only available information that can be incorporated. Future work should study how spatial inferences in such cases can be automatically incorporated to come up with a reasonable estimate of AADT. Having no data for the entire year suggests that the loops, controller, or communication links have been failing for the entire year, which could have been prevented through proper maintenance. Therefore, there is a need for developing an automatic notification system that reports suspected loop failures to Mn/DOT maintenance personnel or finding other ways of reducing long term failures.

Through this project, the research team learned that an automated system on this scale must be an evolving system. Better concepts and methods can be tested as the system is being developed, implemented, and used, as this was demonstrated in this project. That is, many initial concepts and methods in the proposal evolved into improved concepts and methods. It is expected that the three parties of the project team, TDA, TMC, and UMD Data Center, will continue to work together on improving the present system.

REFERRENCES

[1] U.S. DOT ITS, Archived Data User Service (ADUS), "ITS Data Archiving: Five-Year Program Description," March 2000, Published by U.S. DOT, ADUS Program.

[2] Margiotta, Richard, *ITS as a Data Resource: Preliminary Requirements for a User Service.* Report FHWA-PL-98-031, Federal Highway Administration, Washington, DC, April 1998.

[3] Gold D., S. Turner, B. Gajewski and C. Spiegelman, "Imputing Missing Values in ITS Data Archives for Intervals Under 5 Minutes," *TRB 80th Meeting CD- ROM*, Paper No. 01-2760, 2001.

[4] Schmoyer, R., P. Hu, and R. Goeltz, "Statistical Data Filtering and Aggregation to Hour Totals of ITS Thirty-Second and Five-Minute Vehicle Counts," *TRB 80th Annual Meeting CD-ROM*, Paper No 1769, 2001.

[5] Little, R. J. A. and D. B. Rubin, *Statistical Analysis with Missing Data*, Wiley Series in Probability and Mathematical Statistics, 1987.

[6] Schafer, J. L., *Analysis of Incomplete Multivariate Data*, Chapman & Hall/CRC Publication, 1997.

[7] Rubin, Donald B., *Multiple Imputation For Non-Response in Surveys*, Wiley Series in Probability and Mathematical Statistics, 1987.

[8] Gelman A., J. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis, Texts in Statistical Science*, Chapman & Hall/CRC, 1995.

[9] Box, G.E.P., G.M. Jenkins, and G.C. Reinsel, *Time Series Analysis – Forecasting and Control*, 3rd ed., Englewood Cliffs, NJ; Prentice Hall, 1994.

[10] Chatfield, C., *The Analysis of Time Series – An Introduction*, 5th ed., London, UK; Chapman and Hall, 1996.

[11] Naidu, P.S., *Modern Spectrum Analysis of Time Series*, Boca Raton, FL; CRC Press Inc., 1996.

[12] Warner, R. M., *Spectral Analysis of Time-Series Data*, New York, NY; Guilford Press, 1998.

APPENDIX A

Station Identification Database

Station Table

Column Name	**Description**
StaDBID	DB generated index
SeqNum	Mn/DOT defined sequence number
ATRNum	Mn/DOT defined ATR number
RoadName	Road name that the station belongs to
RoadDir	Road direction at the station
Class	Continuous or Short-Duration
MilePoint	Station mile point
LocDesc	Location description generally used
County	County name that the station belongs to
District	District name that the station belongs to
LasdMod	Last modified date and time
LastUser	The user who last modified
Comments	Any comments on the station

Detector Table

Column	Description
StaDBID	Index of station table
DetDBID	Index of detector generated by database
DetNum	TMC assigned detector number
Priority	Primary, secondary, or tertiary
Negative	Negative signed detector (volume is subtracted)
LaneDir	Lane direction of the detector

Appendix B

Sample Log Data for Continuous Count Data

```
Processing: ATR 301 - 12/16/2002 - Seq 0
 ---Checking Primary---
  Data Missing:  E .03%   W .03%
  Direction Volume Difference =  8.9%
 ---Checking Secondary---
  Data Missing:  E .02%   W .03%
  Direction Volume Difference =  6.2%
 ---Checking Tertiary---
  Data Missing:  E .02%   W .03%
  Direction Volume Difference =  9.8%
 Selected det set:
 E   P   P   P   P   S   P   P   P   P   P   P   P
     P   P   P   P   P   P   P   P   P   P   P   P
 W   P   P   P   P   S   P   P   P   P   P   P   P
     P   P   P   P   P   P   P   P   P   P   P   P
 Hourly Missing %:
 E   0   0   0   0 .50   0   0   0   0   0   0   0
     0   0   0   0   0   0   0   0   0   0   0   0
 W   0   0   0   0 .67   0   0   0   0   0   0   0
     0   0   0   0   0   0   0   0   0   0   0   0

Processing: ATR 303 - 12/16/2002 - Seq 0
 ---Checking Primary---
  Data Missing:  N .03%   S .03%
  Direction Volume Difference =  3.8%
 ---Checking Secondary---
  Data Missing:  N .03%   S .03%
  Direction Volume Difference =  2.8%
 ---Checking Tertiary---
  Data Missing:  N  17%   S  22%
  Direction Volume Difference =  .36%
 Selected det set:
 N   P   P   P   P   S   P   P   P   P   P   P   P
     P   P   P   P   P   P   P   P   P   P   P   P
 S   P   P   P   P   P   P   P   P   P   P   P   P
```

```
     P   P   P   P   P   P   P   P   P   P   P   P
 Hourly Missing %:
 N   0   0   0   0 .63   0   0   0   0   0   0   0
     0   0   0   0   0   0   0   0   0   0   0   0
 S   0   0   0   0 .63   0   0   0   0   0   0   0
     0   0   0   0   0   0   0   0   0   0   0   0

Processing: ATR 309 - 12/16/2002 - Seq 0
 ---Checking Primary---
  Data Missing:  N .01%   S .01%
  Direction Volume Difference =   18%
 ---Checking Secondary---
  *unZip_1Det: Detector Not Found! D:\DOT\Traffic\20021216.traffic
95.v30
  Data Missing:  N .02%   S  25%
  Direction Volume Difference =   65%
 ---Checking Tertiary---
  Data Missing:  N  31%   S  31%
  Direction Volume Difference =  1.3%
 Selected det set:
 N   P   P   P   P   P   P   P   P   P   P   P   P
     P   P   P   P   P   P   P   P   P   P   P   P
 S   P   P   P   P   P   P   P   P   P   P   P   P
     P   P   P   P   P   P   P   P   P   P   P   P
 Hourly Missing %:
 N   0   0   0   0 .28   0   0   0   0   0   0   0
     0   0   0   0   0   0   0   0   0   0   0   0
 S   0   0   0   0 .28   0   0   0   0   0   0   0
     0   0   0   0   0   0   0   0   0   0   0   0

Processing: ATR 315 - 12/16/2002 - Seq 0
 ---Checking Primary---
  Data Missing:  N .03%   S .03%
  Direction Volume Difference =  1.6%
 ---Checking Secondary---
  Data Missing:  N .01%   S .01%
  Direction Volume Difference =  1.6%
 ---Checking Tertiary---
```

```
  Data Missing:  N .03%   S .03%
  Direction Volume Difference =  1.2%
 Selected det set:
 N   P   P   P   P   S   P   P   P   P   P   P   P
     P   P   P   P   P   P   P   P   P   P   P   P
 S   P   P   P   P   S   P   P   P   P   P   P   P
     P   P   P   P   P   P   P   P   P   P   P   P
 Hourly Missing %:
 N   0   0   0   0 .21   0   0   0   0   0   0   0
     0   0   0   0   0   0   0   0   0   0   0   0
 S   0   0   0   0 .28   0   0   0   0   0   0   0
     0   0   0   0   0   0   0   0   0   0   0   0

Processing: ATR 321 - 12/16/2002 - Seq 0
 ---Checking Primary---
  *unZip_1Det: Detector Not Found! D:\DOT\Traffic\20021216.traffic
850.v30
  Data Missing:  E  33%   W .02%
  Direction Volume Difference =   100%
 ---Checking Secondary---
  *unZip_1Det: Detector Not Found! D:\DOT\Traffic\20021216.traffic
854.v30
  *unZip_1Det: Detector Not Found! D:\DOT\Traffic\20021216.traffic
855.v30
  *unZip_1Det: Detector Not Found! D:\DOT\Traffic\20021216.traffic
856.v30
  *unZip_1Det: Detector Not Found! D:\DOT\Traffic\20021216.traffic
857.v30
  Data Missing:  E .02%   W  100%
  Direction Volume Difference =  100%
 ---Checking Tertiary---
  *unZip_1Det: Detector Not Found! D:\DOT\Traffic\20021216.traffic
849.v30
  *unZip_1Det: Detector Not Found! D:\DOT\Traffic\20021216.traffic
850.v30
  *unZip_1Det: Detector Not Found! D:\DOT\Traffic\20021216.traffic
851.v30
```

*unZip_1Det: Detector Not Found! D:\DOT\Traffic\20021216.traffic
853.v30

Data Missing: E 100% W .01%

Direction Volume Difference = 100%

Selected det set:

E	S	S	S	S	S	S	S	S	S	S	S	S
	S	S	S	S	S	S	S	S	S	S	S	S
W	P	P	P	P	T	P	P	P	P	P	P	P
	P	P	P	P	P	P	P	P	P	P	P	P

Hourly Missing %:

E	0	0	0	0	.42	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0
W	0	0	0	0	.21	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0

Appendix C

Format for Station Detector List Files

The detector list files for both the Short-duration Count (will be referred to as SC) stations and Continuous Count (will be referred to as ATR) stations must follow strict rule in order to allow the files to be used by the intended software.

Notation Convention:

yyyy	four digit number representing year, 0000-9999
mm	two digit number representing month, 01-12
dd	two digit number representing date, 01-31

File name:

For ATR stations, ATRDetsyyyymmdd.txt
Example) ATRDets20020101.txt

For SC stations, SCDetsyyyymmdd.txt
Example) SCDets20030109.txt

It is important to exactly use 8 digits for the year, month, and date along with the prefix as specified by above. The file name must not include any spaces. **This date should correspond to the date you are uploading the file to the TDRL server.** The reason for including the date information in the file is to keep track of changes may occur over time and to allow to go back and be able to rerun the program using the old detector lists. More importantly, the date information allows the developed software to recognize the most recent version of the detector lists and use them.

Primary, Secondary, and Tertiary Detector Sets

- For ATR stations, **all three prioritizes sets of detectors must be listed.**
- For SC stations, **only the primary set of detectors are required**, and the secondary and tertiary detector sets may be optionally added.

This and file names are the only difference between the ATR and SC stations.

Direction Code

The road direction at a station is numerically coded as follows:

- 1=N
- 2=NE
- 3=E
- 4=SE
- 5=S

- 6=SW
- 7=W
- 8=NW
- 0=All others such as reversible directions

Rules for the Entry of Detector List

- Any line starting with semicolon ";" is considered as a comment line and ignored by the software. The semicolon does not have to start from the first column. Comment area is especially useful for indicating the detector states and changes.
- Blank lines are considered as a comment line and ignored by the software.
- The detector list for a single station must be specified within a single line no matter how long the line is.
- The data entry line must start with the numeric station ID number and end with the "End" statement.
- All entries are **not case sensitive**.
- Any entries appended after the "End" statement is considered as a comment and ignored by the software.
- Each entry within the line must be separated by comma except for the comments.
- Spaces are allows after the commas, but not allowed between the numeric numbers. For example, "`,  24535`" is OK, but "`,  24  535`" will cause an error.

The Line Entry Format:

The primary detectors are listed after the letter "P"; the secondary detectors are listed after the letter "S"; and the tertiary detectors are listed after the letter "T". The line format is:

StationID, DirCode, P, detP1, detP2, …, S, detS1,detS2,…, T, detT1,detT2, …, End

where

StationID: numeric number of the station ID
DirCode: numeric number representing the direction
P, S, T: indicates start of list of primary, secondary, and tertiary detectors
detP1, detP2, …: list of numeric numbers representing primary detectors
detS1,detS2,…: list of numeric numbers representing secondary detectors
detT1,detT2, …: list of numeric numbers representing tertiary detectors
End: indicates the end of the detector list for the station

Example Detector List File:

```
; Comments start with ";"
; Detector list for ATR stations
301,3,P,3176,3177,3178,3179,S,2638,2639,2640,2641,2642,T,2643,2644,2645,2646,3180,End
301,7,P,3218,3219,3220,3221,S,2658,3222,3223,3224,3225,T,2663,2664,2665,2666,3217,End
303,1,P,2393,2394,2395,2396,S,2397,2398,2399,2400,T,2389,2390,2391,-2392,2396,End
303,5,P,2457,2458,2459,2460,S,2450,2451,2452,2456,T,-2461,2462,2463,2464,End
309,1,P,341,342,343,S,173,174,344,345,570,T,335,336,337,338,-339,-340,End
309,5,P,178,179,189,S,94,95,176,177,T,-181,-182,183,184,185,186,End
315,1,P,494,495,496,S,266,267,525,1038,T,156,268,269,270,533,End
315,5,P,256,257,1003,S,1006,1007,1008,T,110,254,255,1002,End
321,3,P,846,847,850,S,837,838,839,841,T,849,850,851,853,End
321,7,P,843,844,845,S,854,855,856,857,T,826,827,828,830,End
326,3,P,793,794,795,S,1730,1731,1732,T,783,784,785,786,End   Comments can be here

; blank lines are allowed. Blank lines may improve readability.
326,7,P,788,789,790,S,1740,1741,1742,T,777,778,779,787,End
; comments can be added between the lines
329,1,P,339,340,S,-2130,2131,2132,T,335,336,337,338,-341,-342,-343,End
329,5,P,181,182,S,2243,2244,T,-178,-179,-180,183,184,185,186,End
405,1,P,1926,1927,1928,S,1929,1930,1931,T,1922,1923,1924,-1925,1928,End
405,5,P,1970,1971,S,1972,1973,1974,T,1972,-1975,1976,1977,1978,End
; comments can be added at the end as well.
; all of the above are valid format
```