2007). These technologies, along with their predominant initiating sources, are outlined in Table 2-1. While considered highly reliable, there is a considerable delay between the occurrence of an incident and the initiation of a traffic management response where low tech technologies are engaged (Singliar and Hauskrecht, 2006).

Table 2-1. Human-Based Incident Detection Technologies

Technology	Initiating Source
Cellular Phone Calls	Passing Motorists
Freeway Service Patrols	Police and Other Official Vehicles
Peak Period Patrols	Law Enforcement Agencies
Fleet Operators	DOT Patrol Agents
Closed Circuit TV	Traffic Center Employees
Motorist Call Boxes	Passing Motorists
Aircraft Patrols	Law Enforcement and DOT
Fixed Observers	Law Enforcement and DOT
CB Radio Monitoring	Law Enforcement Agencies

Automatic incident detection (AID) is generally founded in a series of algorithms intended for the recognition of freeway incidents. While this approach has been around since the beginning of intelligent transportation systems, studies indicate that the effectiveness of AID is, at best, poor (Parkany and Xie, 2002). The lack of AID operational usefulness is primarily related to an unacceptably high false alarm rate, resulting in many automatic incident alarms being either disabled or ignored.

For a period of time, the poor performance of AID was of no consequence, since traffic management centers (TMCs) were able to provide marginal detection capability through human-based systems (Sobhi and Kelly, 1999). However, the increasing size and range of freeway transportation networks under the management of TMCs are growing at rates faster than human-

Table 2-5 AFIDS-ICM Performance Data by Road Section for Low Congestion

LOC	Low					
Rd. Sec.	TTC_O (m)	TTC_C (m)	DR (%)	FCR (%)	AR (%)	ACC (%)
I-85	12.8	32.2	100	0	100	100
I-185	13.8	31.2	100	0	100	100
I-385	13.8	31.2	100	0	100	100

Table 2-6 AFIDS-ICM Performance Data by Road Section for Moderate Congestion

LOC	Moderate					
Rd. Sec.	TTC_O (m)	TTC_C (m)	DR (%)	FCR (%)	AR (%)	ACC (%)
I-85	12.9	31.1	100	0	100	100
I-185	13.8	31.2	100	0	100	100
I-385	13.8	31.2	100	0	100	100

Table 2-7 AFIDS-ICM Performance Data by Road Section for High Congestion

LOC	High					
Rd. Sec.	TTC_O (m)	TTC_C (m)	DR (%)	FCR (%)	AR (%)	ACC (%)
I-85	24.4	20.6	100	0	100	100
I-185	24.9	20.1	100	0	100	100
I-385	24.9	20.1	100	0	100	100

Table 2-8 AFIDS-ICM Cumulative Performance Data by Road Section

LOC	All LOC's					
Rd. Sec.	TTC_O (s)	TTC_C (s)	DR (%)	FCR (%)	AR (%)	ACC (%)
All Rd. Sec.	17.2	27.65	100	0	100	100

Results indicated in Tables 2-5 through 2-7, and summarized in 2-8, support the use of AFIDS-ICM and AFIDS-RM as a means of characterization of traffic incidents and the identification of alternate routes as an integral part of traffic related decision support. Using a computer based platform, the methodologies described result in characterization and re-routing within 24.4 minutes for low and high priority incidents. And, 32.2 minutes for cleared incidents.

ACC results support that all three objectives of the study were obtained and that the two modules operate at maximum efficiency. This is somewhat expected due to the incorporation of computer logic into the process that all but eliminates the possibility for error.

The TTC_n is relatively high compared to other approaches. However, this relates to the efficiency of the AFIDS-IDM, which is tuned for maximum identification of incidents with limited tolerance for false alarms. Such tuning is achieved through a calculation process that waits longer to make a decision, but is less prone to error.

There is a noticeable difference between the times to characterize a cleared incident (TTC_C) and the time to characterize an incident that has occurred (TTC_O). This is explained on the basis of fuzzy clustering theory where an incident can only be declared “cleared” when the decision variable reaches an incident free state. That is the degree of belongingness associated with the incident-free membership is greater than the degree of belongingness associated with the incident membership. This, greater than relationship, can only be obtained at the expense of increased time.

2.5. Conclusion and Recommendations

Advances in AID have lead to a number of approaches to automatic incident detection. These approaches were generally restricted to incident detection and stopped short of incident characterization and re-routing options. This has created a void in traffic control and management efforts, leading to increased legislation directed toward the creation of advanced information systems linked to freeways and vehicles.

This paper introduces AFIDS-ICM as the incident characterization module of the AFIDS decision support system. The system is augmented with a re-routing module, AFIDS-RM, employed when high priority incidents are recognized. The modules presented engage traffic management personnel through a series of user friendly computer screens that are color coded to simplify the recognition and characterization of traffic incidents.

Performance tests of the two modules indicate that the modules meet their intended objectives through the:

1. Maximization of the real time characterization of traffic incidents
2. Minimization of the number of false characterizations
3. Maximization of the correct number of correct alternate routes reported

In comparison to alternative approaches to incident characterization and re-routing, the two modules described offer two relative advantages:

1. The two modules presented are accompanied with an incident detection capability. Working in unison, the three modules form a complete decision support system directed toward the identification and characterization of traffic incidents, supported with re-routing capability.

2. The two modules are supported with color coded computer graphic output for incident detection and graphical mapping for re-routing.

Current efforts are underway to enhance the computer graphic capability of the two modules. Through these efforts, traffic managers will be able to access additional screens detailing historical data, weather conditions, traffic flows is areas surrounding the incident, and weather conditions surrounding the incident. This is expected to increase the value of the tool by providing additional decision support data.

REFERENCES

Busch, F. (Eds). (1987). “Incident Detection”. In M. Papageorgiou (Ed). *Concise Encyclopedia of Traffic and Transportation Systems,* pp. 219-225. Oxford.

Chen, K. & Galler, B. (1990). “An Overview of Intelligent Vehicle-Highway Systems (IVHS) Activities in North America”. *Proceedings of the Fifth Jerusalem Conference on Information technology***.** Jerusalem, Israel. Pp. 694 – 701.

Chow, E. Y., Gershwin, S. B., Greene, C. S., Houpt, P. K., Kurkijian, A. and Willsky, A. S. (1977). Dynamic Detection and Identification of Incidents on Freeways Volume I : Summary. ESL-R-764 Electronic Systems Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Courage, K. G. and Levin, M. (1968). A Freeway Corridor Surveillance, Information and Control System. Research Report 488-8 Texas Transportation Institute, Texas A&M University, College Station.

Dudek, C.L. and Messer, C.J. (1974). “Incident Detection on Urban Freeways”. *Transportation Research Record,* No. 495, pp. 12–24.

Eisele, W.L., and Lomax, T.J. (2004). ODOT Operations Performance Measures: Final Report. Sponsored by the Oregon Department of Transportation, June 2004. Available at: http://mobility.tamu.edu/resources.

Ishak, S. S. and Al-Deek, H. M. (1998). “Freeway Incident Detection Using Fuzzy Art Adaptive Resonance Theory”. *Proc. Fifth International Conference on Applications of Advanced Technologies in Transportation Engineering*, pp. 59-66.

Lomax, T.J., Turner, S.M., and Margiotta, R. (2003). Monitoring Urban Roadways in 2001: Examining Reliability and Mobility with Archived Data. Report FHWA-OP-03-041. Pp. 64.

Monahan, T. (2007). War Rooms of the street: Surveillance practices in transportation control centers. *The Communication Review* Iss. 10, No. 4, pp. 367-389.

Nowakowski, C., Green, P., Kojima, M. (1999). *Human Factors in Traffic Management Centers: A LiteratureRreview. (Technical Report UMTRI-99-5), Ann Arbor, MI: The* University of Michigan Transportation Research Institute.

Parkany, E. and Xie, C.. (Aug. 2002). “Use of Driver-Based Data for Incident Detection”. *Proceedings of the 7th International Applications of Advanced Technologies in Transportation Conference,* pp. 143-150.

Payne, H. J. and Tignor, S. C. (1978). "Freeway Incident-Detection Algorithms Based on Decision Trees with States". *Transportation Research Record*, No.682, pp. 30-37.

Roy, P., and Abdulhai, B.(2004). "GAID: Genetic Adaptive Incident Detection for Freeways". *Transportation Research Record,* No. 1856, pp 96-106.

U.S, Government Printing Office. (2005). Safe, Accountable, Flexible, Efficient Transportation Equity Act: A Legacy for Users (SAFETEA-LU), Title V, Subtitle C, Sec. 5301-5310. Washington, DC: U.S. Government Printing Office.

Shue, J. (2002). A Fuzzy Clustering-Based Approach to Automatic Freeway Incident Detection and Characterization. *Fuzzy Sets and Systems,* Iss. 128, pp. 377–388.

Singliar, T. and Hauskrecht, M. (2006). "Towards a Learning Traffic Incident Detection System". *ICML 2006 Workshop on Machine Learning Algorithms for Surveillance and Event Detection,* Pittsburgh.

Sobhi, N. and Kelly, M. (1999). Human Factors Recommendations for TMC Design. *Public Roads,* Vol. 62, No.6, pp. 36-48.

Transportation Research Board. (2008). *Highway Capacity Manual* (3rd ed.), Washington, DC: National Academy of Sciences.

Williams, B.M. and Guin, A. (June 2007). "Traffic Management Center use of Incident Detection Algorithms: Findings of a Nationwide Survey". *IEEE Transactions on Intelligent Transportation Systems,* Vol. 8, no. 2, pp. 351-358.

ARTICLE THREE

Comparison of Five Algorithms for Automatic Freeway Incident Detection

3.1. Introduction

Non-recurring traffic incidents can be costly, resulting in increased traffic congestion, fuel consumption, and environmental pollution. Some estimates place these costs in excess of $35 billion per year (Lindley, 1996). Early recognition can help control the escalation of costs by both reducing the time frame to clear an incident and reducing the potential for secondary incidents (Busch, 1987). It is not unexpected that a considerable body of research has evolved on quick detection methodologies centered on automatic incident detection (Williams and Guin, 2007).

3.2. Background Studies

The beginnings of Automatic Incident Detection (AID) can be traced back to the 1960's with early approaches restricted to the development of statistical and pattern-based algorithms (Dudek and Messer, 1974; Courage and Levin, 1968; Levin and Krause, 1978). Since that time, other efforts, based on numerous approaches, have evolved. These include comparative algorithms (Payne and Tignor, 1978), time series and filtering algorithms (Chow et al.1977; Cook and Cleveland, 1974), and traffic theory based algorithms (Chow et al.1977; Gall and Hall, 1989; Greene et al., 1977; Kurkijian et al., 1977).

More recently, AID research has centered on artificial intelligence and soft computing techniques producing a number of methodologies including fuzzy logic/fuzzy set theory (Hsiao et al., 1994; Chang and Wang, 1994; Lin and Chang, 1998; Shue, 2002), artificial neural networks (Dia and Rose, 1997), fuzzy logic in conjunction with neural networks (Ishak and Al-Deek, 1998; Srinivasan et al, 2001), fuzzy expert systems (Lin and Chang, 1998), wavelet transformations (Samant and Adeli, 2000), and generic algorithms over neural networks (Roy and Abdulhai, 2003).

AID research has evolved with the introduction of one technique after another, with no single methodology assuming a dominate role in incident detection. This can be somewhat perplexing for traffic management practitioners who must distinguish the applicability of one approach over another in actual field applications. Compounding the issue further is that many methods were developed in simulated environments where actual traffic conditions were designed to fit the algorithm, giving a greater degree of control of the experiment than would be found in actual implementation. Further, AID methodologies are generally presented as standalone approaches rather than as comparisons against baseline values.

One barrier to comparison by field practitioners is the complex calibrations that many AID are based on. The calibration parameters have to be fine-tuned in practice. Failure to do so properly can result in poor performance and abnormally high false alarm rates (Balke, 1993).

Data collection appears to be the single commonality to AID. Lane traffic counts and density, collected at successive traffic sensors, are the two primary types of input data. Capacity is then calculated mathematically from inputs.

The evaluation of incident detection algorithms is a somewhat time consuming process, due to the need to calculate threshold values for each methodology. However, since incident detection algorithms seldom perform at the levels they were developed under (Abdulhai and Ritchie, 1999), it is a necessary undertaking. The standard approach that has evolved to meet this need produces three metrics (Dia et al., 996): a) detection rate (DR), false alarm rate (FAR), and time to detect (TTD). The Comparison of AID's between detection zones is through a methodology assigning weighted priorities to each of the three metrics.

3.3. Research Objectives and Scope

It is the purpose of this paper to survey current literature related to AID and to list present methods in this field of study. Further, this paper evaluates each method against a set of criteria to short-list algorithms that are applicable to field practice. The latter are then implemented using real data collected by the South Carolina Department of Transportation (SCDOT) to determine the FAR, DR, and TTD.

3.3.1 Identification of Current Algorithms

The literature search included a wide variety of electronic and print resources to identify algorithms for inclusion in this study, including the *Transportation Research Board (TRB),* the *Journal of Transportation Research (JTR),* and the *Intelligent Transport Systems Journal (ITS).* Additionally, holdings from seven university libraries and *Dissertation Abstracts International* were searched along with reference sections of the innumerable studies that were collected were reviewed to identify other potential pertinent research. Finally, several researchers and practitioners, currently working in the field, were contacted and asked to provide pertinent research or to identify sources of studies.

A total of twenty-two algorithms, indicated in Table 3-1, were identified from the

Table 3-1. AID Algorithms Identified Through Literature Search

Standard Normal Deviate Algorithm	Exponential Smoothing Algorithm
Low Volume Algorithm	Dynamic Model (MM and GLR)
The California Algorithms	Bayesian
Decision Logic Units-based Algorithm	HIOCC and PATREG
ARIMA	Multi-Layer Feed-Forward
DELOS	Image Processing
Fuzzy Logic	Principal Component Analysis
Cumulative Sum of Occupancy	Probabilistic Neural Network Algorithm
Fuzzy Radial Neural Network Algorithm	Adaptive ANN-Wavelet Algorithm
Wavelet Energy-Radial Basis Algorithm	Discrete Wavelet Transform Algorithm
CUSUM based Algorithm	Support Vector Machine

literature review. Inclusion criteria for the initial list centered on its ability to recognize traffic incidents and not on its applicability to TMC's.

3.3.2 Short-List of Applicable Algorithms

The twenty-two algorithms were examined for five unique identifiers to determine if they met criteria for inclusion in the study. The criteria, listed below, identify those algorithms that can be implemented with real traffic data:

- Over-saturation is not included as a category of analysis
- The capability to analyze bulk data on an hour-by-hour basis
- Incidents are predicted within a specific zone at a given point in time
- No traffic incident data was available for the fine tuning of the algorithm
- Lane occupancy could be calculated mathematically

Five algorithms met the criteria for inclusion:

- California Algorithm #8
- Exponential Smoothing Algorithm
- McMaster Incident Detection Algorithm
- Shue Fuzzy Logic Algorithm
- Alabama Freeway Incident Detection System Algorithm

3.3.2.1 California Algorithm #8

The California Algorithm Family (Payne and Tignor, 1978) is a set of 10 algorithms that are founded in decision tree analysis. The original algorithm is a straightforward approach recognizing a potential incident has occurred when three tests on the measured occupancy from two adjacent stations surpass preset threshold values (T_n) associated with each test. Definitions associated with the general logic of the California Algorithm Family are presented in Table 3-2 followed by a graphical representation of the logic in decision tree form in Figure 3-1.

Table 3-2. Definitions Associated with California Algorithm Family Logic

Measure	Description	Definition
OCC(i,t)	Occupancy at detector station i at time t	
DOCC(i,t)	Downstream occupancy	OCC(i+1,t)
OCCDF(i,t)	Spatial occupancy difference	OCC(i,t) - DOCC(i,t)
OCCRDF(i,t)	Relative spatial occupancy difference	OCCDF(i,t) / OCC(i,t)
DOCCTD(i,t)	Relative temporal downstream occupancy difference	(DOCC(i,t-2)-OCC(i,t) /OCC(i,t-2))
T_n	Predetermined threshold value	

An incident state is terminated when threshold value T_2 is no longer exceeded. Threshold values are calibrated from empirical data.

The simplicity of the original algorithm resulted in an unacceptably high false alarm rate. Subsequent algorithms reduce the false alarm metric but are more complex.

The two that have demonstrated higher performance ratings are the #7 and #8. The California #7 replaces the use of relative temporal differences in downstream occupancy values with occupancy measurements (DOCCTD(i,t)). This reduces the false alarm rate through the recognition of recurring compression waves commonly found in heavy traffic. The #7 recognizes that simple downstream occupancy data dropping below a certain threshold, usually 20 percent is more indicative of an incident. This algorithm also incorporates a persistence check that requires traffic discontinuity persist for a specified period of time before a potential incident is recognized.

The California #8 is the most complex of the California family and it has been shown to be the best performer (Cohen and Ketselidou, 1993). The algorithm recognizes nine states and requires the calibration of five threshold values, two more than previous algorithms. This algorithm introduces a repetitive test for compression waves and a suppression of alarms related to these waves for up to five minutes. Through this approach, the #8 is less likely to give a false alarm in normal traffic congestion. The modified decision logic of the California #8 is presented in Figure 3-2 as suggested by Payne et al. (1976).

3.3.2.2 Exponential Smoothing Algorithm

The Exponential Smoothing Algorithm was developed by Cook (1974) using data from the John C. Lodge Freeway in Detroit. The method is an extension of the Standard Normal Deviate

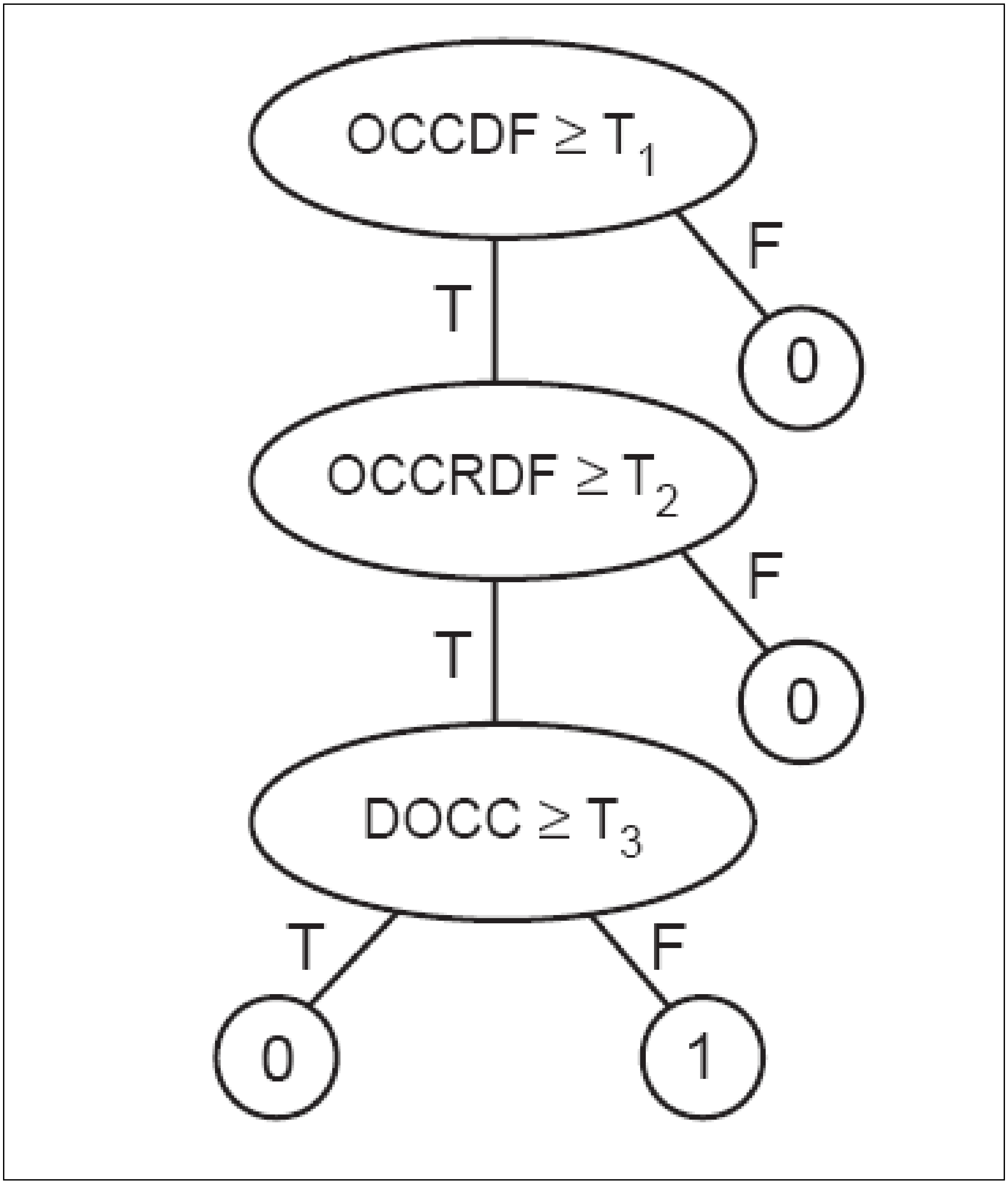
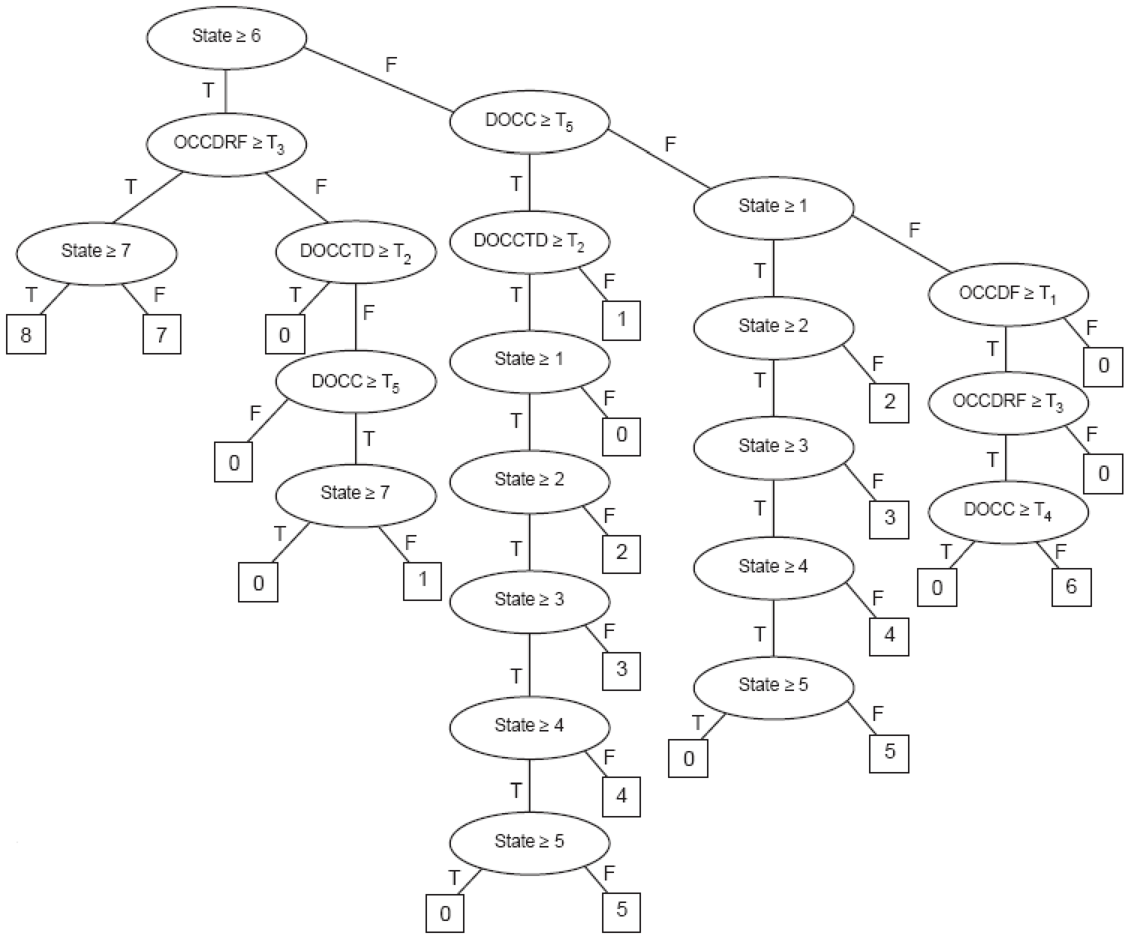


Figure 3-1. Decision Tree for General Logic of California Algorithm Family



States:

0- Incident Free Condition
1- Compression wave this minute
2- Compression wave 2 minutes ago
3- Compression wave 3 minutes ago
4- Compression wave 4 minutes ago
5- Compression wave 5 minutes ago
6- Tentative incident
7- Incident confirmed
8- Incident continuing

Figure 3-2. California #8 Decision Tree Logic

Algorithm (Dubek, and Messer, 1974) but differs in the use of a more sophisticated forecasting method. The smoothing feature of the algorithm gives a heavier weight to recent traffic data than past records reducing false alarms related to traffic volumes. The algorithm uses a double smoothing approach indicated in Equations 3-1 and 3-2.

$$S_1(k) = X(k) + (1- S_1(k-1) \qquad (3\text{-}1)$$

$$S_2(k) = S_1(k) + S_2\ (k-1) \qquad (3\text{-}2)$$

Where:

X= A smoothing constant determined from the weight of past data

S_1= The first set of smoothed data

S_2= The second set of smoothed data

k= Time step

The smoothed data set is then used to generate a tracking signal as the algebraic sum of all previous estimate errors to the present minute, divided by the current estimate of the standard deviation. An incident is indicated when the tracking signal deviates from zero beyond a pre-specified threshold. The threshold can be computed based on either the variability of the data or likelihood of a false alarm.

Cook used a set of 13 traffic variables resulting from the basic traffic variables of volume, occupancy, and speed to test the performance of the algorithm (Cook and Cleveland, 1974). The variables are indicated in Table 3-3.

Table 3-3. Exponential Smoothing Basic Traffic Variables

Station volume	Subsystem volume
Station occupancy	Subsystem occupancy
Station speed (volume/occupancy)	Subsystem speed
Station volume-occupancy	Subsystem kinetic energy
Station speed-occupancy	Volume-occupancy discontinuity
Station kinetic energy	Speed-occupancy discontinuity
Station discontinuity	

3.3.2.3 McMaster Incident Detection Algorithm

The McMaster Incident Detection Algorithm is a catastrophe theory algorithm developed using data from Queen Elizabeth Way, Mississauga, Ontario. The algorithm is based on the belief that flow and occupancy, unlike speed, change smoothly when moving from a congested to an uncongested state. The algorithm starts by identifying congested states, and then, it attempts to determine if the congestion is the cause of a traffic incident or a permanent bottleneck. As suggested by Persaud et al. (1981) a volume-occupancy template (Figure3-3) is derived from historical flow-occupancy data collected at times of change from congested to uncongested conditions. Traffic conditions are classified into one of four states, calibrated at each detector station:

State 1- Uncongested

State 2- Congestion

State 3- Congestion

State 4- Permanent bottleneck congestion

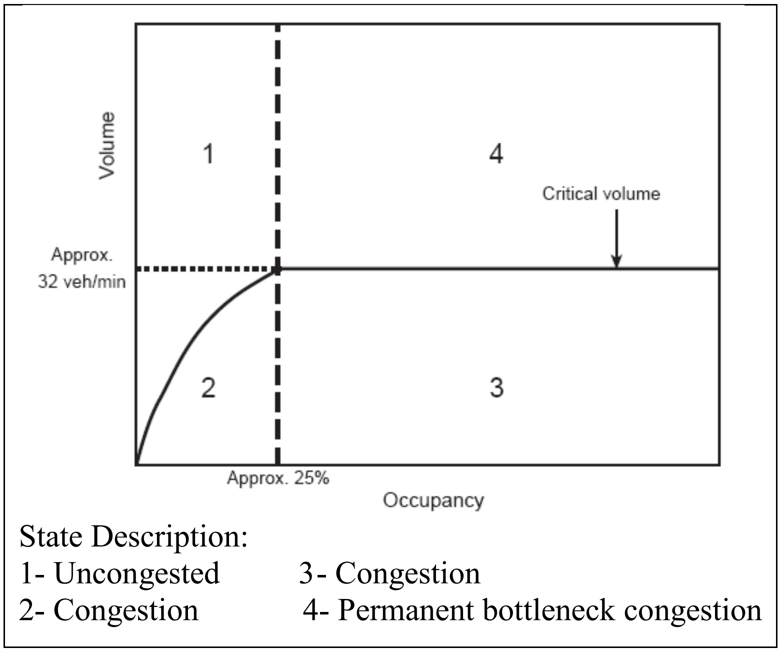


Figure 3-3. McMaster Catastrophe Theory Template

Once the volume-occupancy template is determined the following logic is applied:

1. If State 2 or 3 is indicated the algorithm examines the traffic condition at the downstream station.
2. An alarm is triggered if the downstream station is in State 1 or 2 on the belief that recurring and incident congestion result in different downstream traffic patterns.
3. If State 3 is detected the algorithm looks at the next downstream detector station using the same logic.
4. If State 4 is detected at the downstream detector station, the congestion is classified as recurring.

The initial McMaster algorithm was refined to include additional states intended to decrease the vulnerability of the algorithm to incident related traffic patterns resulting from non-incident conditions. This modified the original logic to include separate templates to discriminate between detector stations, depending on their location with respect to recurring bottlenecks. As suggested by Hall et al. (1993), Figure 3-4 indicates the templates, intended for normal and recurrent congestion stations.

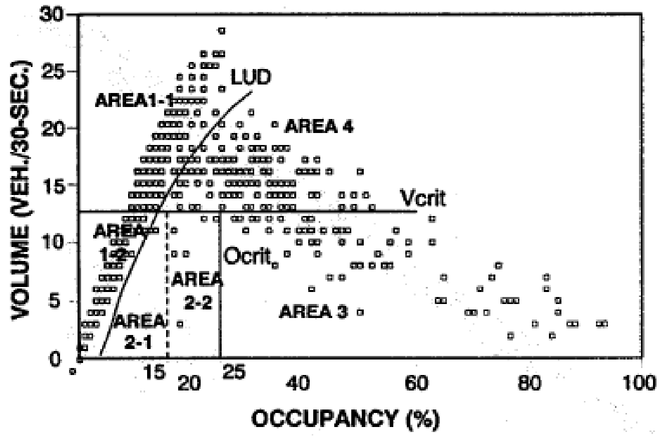
Calibration of the algorithm involves distinguishing between the congested and un-congested regions. The minimum non-congested speed is estimated for the station. This is used to create the boundary between States 1 and 3. A quadratic equation is then estimated to obtain flow as a function of occupancy at the station, and a constant flow value is estimated, which is to be subtracted from the function to create the boundary between States 1 and 2.

3.3.2.4 Shue Fuzzy Logic Algorithm

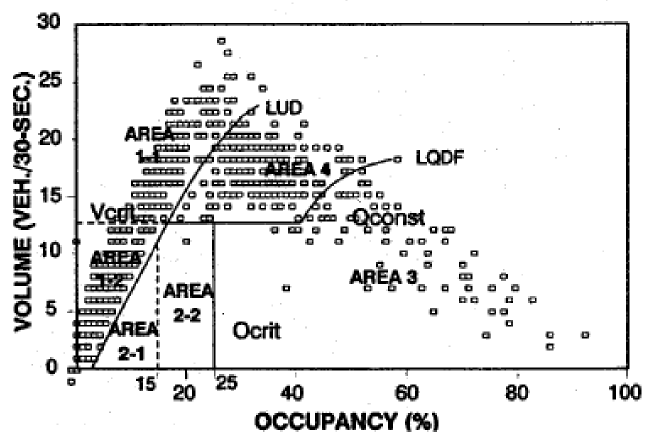
The Shue algorithm (Shue, 2002) is a fuzzy clustering approach to freeway incident detection and characterization developed under simulation of traffic patterns through CORSIM. The method identifies lane blocking incidents from comparisons of time varying patterns of incident induced and incident free traffic states. Lane traffic counts and density, collected at successive traffic sensors, are the two primary types of input data.

The Shue algorithm is carried out in four steps:

1. Identification of traffic flow conditions
2. Determination of decision variable



Template Indicating Typical Parameters for Normal Station



Template Indicating Typical Station Affected by Recurrent Congestion

Figure 3-4. Templates Intended for Normal and Recurrent Congestion

3. Determination of fuzzy set membership
4. Determination of a lane blocking incident

Step one is a pre-classification of traffic flow conditions as low, moderate, heavy, or over congestion. Each condition is assigned an occupancy based threshold determined through historical data. At a given time-step, the time varying occupancy based correlation associated with each type of traffic condition is calculated with lane occupancies collected at the upstream sensor along with the occupancy-based threshold. Comparisons of the time-varying occupancy based correlation values indicate the specific type of flow condition in accordance with the highest correlation value identified by the time varying occupancy values.

Incident occurrence is recognized through the execution of Steps 2 through 4. In Step 2, a decision variable is determined for the previously identified traffic condition. The decision variables for low, moderate, and heavy congestion are determined through equations 3-3, 3-4, and 3-5 respectively. No decision variable is determined for over congestion:

$$v_i^5\ (k) = \frac{[o_i^u(k) - o_i^d(k)]}{o_i^u(k)} \tag{3-3}$$

$$v_i^4\,(k) = \ \{[[\textstyle\sum_{j=1}^{J} f_j^d\ (k)]/J] - f_i^d(k)\}\,\{[\,\textstyle\sum_{j=1}^{J} f_j^d\ (k)]/J\} \tag{3-4}$$

$$v_i^{10}(k) = \left\{\textstyle\sum_{\tau}^{T} f_i^d\ (k-\tau)\right\} - \ f_i^u(k-T) - \frac{\sum_{j=1}^{J} f_j^d(k)}{J} \tag{3-5}$$

Where:

$f_i^u\ (k) and\ f_i^d\ (k)$= the upstream and downstream traffic counts collected at target lane *i* and time step *k*,
$o_i^u\ and\ o_i^d$ = collected occupancies,

J = total number of adjacent lanes,

T = the maximum time lag predetermined for consideration of the travel time taken from the upstream detector station to the downstream detector station on the relationship between the upstream and downstream traffic data,

τ = time lag index.

Step 3 computes a time varying decision-variable correlation, Equation 3-6, for upstream, midstream, and downstream locations on a lane by lane basis. The decision variable measures the association between decision variables calculated in Step 2 to pattern clusters determined through historical data.

$$\omega_{m,n}^{p.q}(k) = 1 - \frac{1}{B}\sqrt{\sum_{\tau=0}^{3}(v_n^s(k-\tau) - \mu_{m,n}^{p.q})^2} \tag{3-6}$$

Where:

β = a pre-determined value set for the boundaries of $\omega_{m,n}^{p.q}(k)$,

m = level of congestion,

n = lane code,

p = location parameter indicating the location being investigated,

q = a binary digit (q_1 = incident occurrence and q_0 = no incident),

$\mu_{m,n}^{p.q}$ = pattern of the decision variable pre clustered on the basis of historical traffic data associated with attributes *p, q, m, and n.*

An incident is recognized in Step 4 when the following condition is met:

IF $\{\mathrm{Max}[\omega_{m,n}^{1,1}(k), \omega_{m,n}^{2,1}(k), \omega_{m,n}^{3,1}(k)] - \omega_{m,n}^{p,0}(k)\} > \lambda_1$

THEN A lane-blocking incident with the attributes m and n is recognized at time step k

Where:

m = low, moderate, or high level of congestion category;
n = lane code,
k = Specified time frame.

3.3.2.5 Alabama Freeway Incident Detection System (AFIDS) Algorithm

The Alabama Freeway Incident Detection System (AFIDS) is a three module approach to incident detection, characterization, and re-routing in emergency evacuation situations. Incident detection is through a single module, Alabama Freeway Incident Detection System- Incident Detection Module (AFIDS-IDM). AFIDS-IDM incorporates a fuzzy cluster approach to incident detection is an extension of the Shue algorithm. The primacy differences between the two algorithms lies in the environments in which they were developed and the intended application. The Shue algorithm was developed through a simulated environment using Corridor Simulation (CORSIM) software for use in normal freeway incident detection. AFIDS, on the other hand, was developed from freeway data from the South Carolina Department of Transportation (SCDOT) for use in emergency evacuation situations. This difference is compounded in that AFIDS centers on real world traffic patterns and states while the Shue was developed on traffic patterns and states defined through simulation. The patterns and states defining AFIDS are identified through the *Transportation Research Record*.

AFIDS-IDM logic, Table 3-4, is carried out in seven steps. Like most fuzzy set algorithms, AIDS does not give a clear incident or no-incident signal but rather the likelihood that an incident has occurred. The fuzzy logic incorporated into the algorithm is designed to approximate human thinking in situations of imperfect knowledge.

Table 3-4. AFID-IDM Incident Detection Process

Step	Description
1	Input data
2	The determination of a Level of Congestion Index (LOC $_{\text{Index}}$)
3	The determination of the Level of Congestion (LOC) from the index
4	The determination of the decision variable v^s associated with the specified LOC
5	The determination of the comparison variable lambda (λ) associated with the input data and posted speed limit
6	The determination of the fuzzy set membership, ω_m, associated with the specified v^s
7	A lane blocking incident exists when $\omega_m > \lambda$.

Step 1 is data input. From the data, the LOC $_{\text{Index}}$ is determined in Step 2, from Equation 3-7.

$$LOC_{ijk} = (SF_{ik} / c_j) * (1/f_p) \qquad (3\text{-}7)$$

Where:

LOC_{ijk} = Level of congestion for i lane of traffic in evacuation route j at time period k,

SF = Service flow rate for LOC i under prevailing roadway and traffic conditions for I, lanes in one direction, in vehicles per hour. This value is obtained from the input data, and it is the total number of actual vehicles across all bin numbers for that ATR for that hourly update,

c_j = Capacity for the road section under study. This value is obtained from the Capacity field in the ATR data,

(SF_{jk} / c_j) = Utilization at time factor k,

f_p= factor for further adjustments due to time of day determined from a look-up table.

The LOC $_{\text{Index}}$ is applied to a look-up table to determine the Level of Congestion (LOC). Posted speed limit and LOC $_{\text{Index}}$ are the two variables in determining the LOC. Congestion categories are rated as: a) low, b) moderate, c) heavy, and d) over congested.

LOC Categories are derived from Level of Service Categories (LOS) A through F, described in the Transportation Research Board's publication Highway Capacity Manual, where:

LOC Category Low = LOS Categories A and B
LOC Category Moderate = LOS Categories C and D
LOC Category High = LOS Category E
LOC Category Over Congestion = LOS Category F.

Decision variable v^s is determined through the application algorithms v^1, v^2, and v^3, each representing the LOC's low, moderate, and high, respectively.

$$v_i^1(k) = \frac{o^u(k-n) - \mathrm{o}^{\mathrm{d}}(k)}{o_i^u(k-n)} \tag{3-8}$$

$$v_i^2(k) = \left\{\left[\frac{f^d(k)}{J}\right] - f^d(k)\right\} - \{f^d(k)/I\} \tag{3-9}$$

$$v_i^3(k) = \{f^d(k) * Min(1.0, F_p)\} - \{f^u(k_{k-n}) * (C_{fs})\} - (f^d(k))/I \tag{3-10}$$

Where:

$f_i^u(k)$ and $f_i^d(k)$ = the upstream and downstream traffic counts collected at target lane i and time step k,

$o_i^u(k)$ and $o_i^d(k)$ = collected occupancies,

I = total number of adjacent lanes,

T = the maximum time lag predetermined for consideration of the travel time taken from the upstream detector station to the downstream detector station on the relationship between the upstream and downstream traffic data,

F_p = time lag index defined as the posted speed limit / distance between $f_i^u(k)$ and $f_i^d(k)$.

Comparison variable λ is determined by offsetting the traffic count at the upstream ATR by a correction factor adjusting for the minimum speed expected to navigate each detection zone based on the appropriate LOC.

The comparison value, λ, is arrived through the following equation:

$$\lambda = (\text{Upstream traffic count at ATR}_i) * (C_{fs}) \qquad (3\text{-}11)$$

Where:

Upstream traffic count at ATR_i *is* the vehicle count at time period k determined from the upstream ATR;

C_{fs} is a correction factor derived from dividing the minimum expected speed for a detection zone at a specified LOC by the posted speed for the detection zone.

Fuzzy set membership is determined by applying Equation 3-12:

$$\omega_m(k) = 1 - (v^s(k - f_p) - \mu_m) \qquad (3\text{-}12)$$

Where:

ω_m = Fuzzy set membership value,

μ_m = pattern of the decision variable pre clustered on the basis of historical traffic data associated with attribute m.

AFIDS-IDM recognizes a lane blocking incident when the following fuzzy rule is fired:

IF $\omega_m(k) > \lambda$
THEN a lane blocking incident with attribute *m* is recognized at time step *k*
ELSE *k*= *k*+1 and go back to previous procedure

3.3.3. Comparison of Short-Listed Algorithms

Twenty-two incident detection algorithms were identified through an extensive literature review. The algorithms were evaluated against a set of criteria to determine a short-list of algorithms applicable to field application. This list included the California Algorithm #8, Exponential Smoothing Algorithm, McMaster Incident Detection Algorithm, Shue Fuzzy Logic Algorithm, and Alabama Freeway Incident Detection System Algorithm. The effectiveness of the five algorithms was compared against a set of metrics accepted in the evaluation of incident detection algorithms.

3.3.3.1 Data

Data from 269 sensors across South Carolina was made available for the study. Out of these, 12 detection zones, representing continuous sections of the highway, were selected. Data were obtained in terms of vehicle speed and traffic counts from upstream and downstream sensors. Two sensors, or Automatic Traffic Recorders (ATR), in succession define a detection zone. Occupancy values are mathematically calculated from the data. The format in which the collected data was made available for the study is presented in Table 3-5.

ID is the road identification number. Hour refers to the hour of the day the data was reported number sequentially. Lanes indicates the specific lane under consideration and the number lanes in the specific section of road. Bin numbers indicate the number of vehicles traveling at a given speed. Total Volume is the summation of vehicles across all Bins.

Road sections under study, Figure 3-5, were primarily north and south of Greenville, S.C., along Interstate Highway 185, north and south of Spartanburg, S.C., along Interstate Highway 85, and north and south of Laurens County, S.C., along Interstate 385. Data was collected over a period of two and a half months by SCDOT.

Table 3-5. Sample of Collected Traffic Data

ID	ATR_ID	HOUR	LANE	Bin_0_5	Bin_11_5	Total Volume	ID1
137085	0	1	1	0	0	237	1
137086	0	1	2	0	0	150	2
137087	0	1	3	0	0	31	3
137088	0	1	4	0	0	9	4
137089	0	1	5	0	0	87	5

3.3.3.2 Comparative Evaluation

Each of the short-listed algorithms was evaluated to determine the number of incidents each identified across four road sections taken from the data set. Each road section formed a detection zone and was bounded by upstream and downstream ATR’s. The road sections and ATR’s forming the detection zones are presented in Table 3-6. Incident detection results are presented in Tables 3-7 through 3-11.

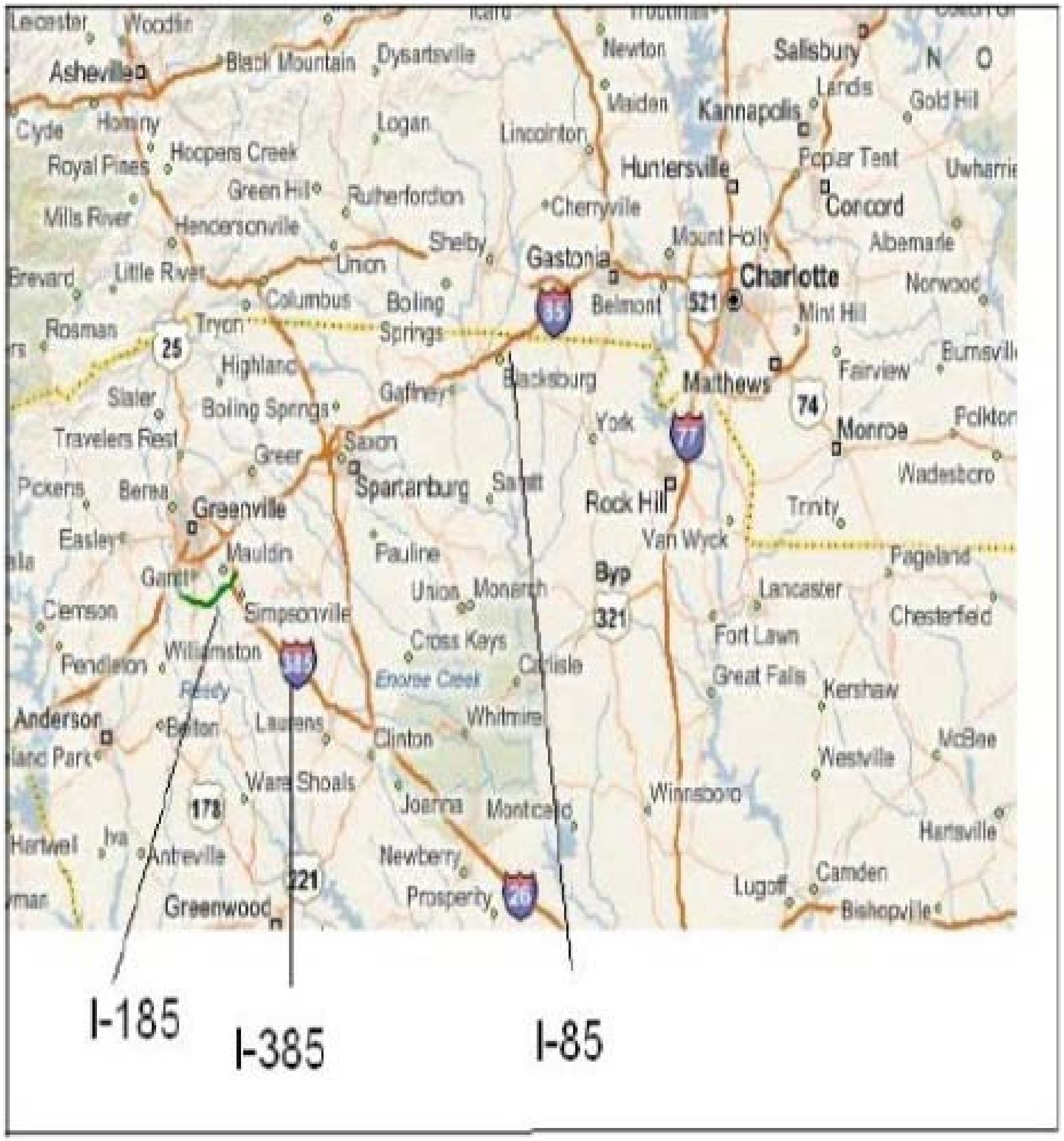


Figure 3-5. Road Sections Considered

Table 3-6. Road Sections by ATR's

Road Section #	Upstream ATR	Downstream ATR
1	194	196
2	196	197
3	242	243
4	243	244

Table 3-7. Performance Results for California #8 Algorithm

Road Section	Total Number of Incidents
ATR 194-196	6
ATR 196-197	26
ATR 242-243	2
ATR 243-244	2

Table 3-8. Performance Results for Exponential Smoothing Algorithm

Road Section	Total Number of Incidents
ATR 194-196	5
ATR 196-197	38
ATR 242-243	2
ATR 243-244	2

Table 3-9. Performance Results for McMasters Algorithm

Road Section	Total Number of Incidents
ATR 194-196	4
ATR 196-197	16
ATR 242-243	1
ATR 243-244	2

Table 3-10. Performance Results for Shue Algorithm

Road Section	Total Number of Incidents
ATR 194-196	5
ATR 196-197	36
ATR 242-243	2
ATR 243-244	2

Table 3-11. Performance Results for AFIDS-IDM

Road Section	Total Number of Incidents
ATR 194-196	5
ATR 196-197	39
ATR 242-243	3
ATR 243-244	2

Performance results indicate that the Exponential Smoothing, Shue, and AFIDS-IDM algorithms report similar results. This indicates that the sensitivity of the three algorithms is tuned to pick up similar categories of incidents across a range of severities. The California #8 and McMasters algorithms, on the other hand, indicate that they are tuned to detect only more severe incidents.

Results indicate that the rankings of incident occurrence for the five algorithms are identical, with ATR's 196-197 reporting the most incidents followed by ATR's 194-196. ATR's 243-244 reported the least number of incidents in the AFIDS-IDM algorithm, while the McMasters reported the least number of incidents through ATR's 242-243. The remaining three algorithms reported equal incidents from ATR's 242-243 and ATR's 243-244.

3.4 Conclusion and Recommendations

This research surveyed current literature related to AID, to list present methods in this field of study. These methodologies were then compared to a set of criteria to determine a short-list of algorithms relevant to field practice. Each of the five algorithms represented a different methodology in automatic incident detection. The five algorithms identified were then implemented with data made available from the South Carolina Department of Transportation (SCDOT) to determine the number of incidents detected by each algorithm.

Of the short-listed algorithms, the Exponential Smoothing (ES), Shue, and AFIDS-IDM algorithms performed at similar sensitivities. The California #8 and McMasters algorithms were noticeably less sensitive than other algorithms.

A limitation of this research lies in its reliance on a single data set. More research is needed with the short-listed algorithms using more diverse data sets demonstrating more varied traffic patterns across significantly different weather patterns.

REFERENCES

Abdulhai, B. and Ritchie, S.G. (1999). "Enhancing the Universality and Transferability of Freeway Incident Detection using a Bayesian-Based Neural Network". *Transportation Research. Part C: Emerging Technologies*, pp. 261–280.

Balke, K.N. (1993). An Evaluation of Existing Incident Detection Algorithms, Technical Report 1232-20, Texas Transportation Institute, Texas A&M University System, College Station, Tex.

Busch, F. (Eds). (1987). "Incident Detection". In M. Papageorgiou (Ed). *Concise Encyclopedia of Traffic and Transportation Systems.* pp. 219-225. Oxford.

Chang, E.-P. and Wang, S. H. (1994). "Improved Freeway Incident Detection Using Fuzzy Set Theory". *Transportation Research Record*, No.1453, pp. 75-82.

Chow, E. Y., Gershwin, S. B., Greene, C. S., Houpt, P. K., Kurkijian, A. and Willsky, A. S. (1977). Dynamic Detection and Identification of Incidents on Freeways Volume I : Summary. ESL-R-764 Electronic Systems Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Cohen, S. and Ketselidou, Z. (1993). A Calibration Process for Automatic Incident Detection Algorithms. *Microcomputers in Transportation.* Pp. 7-254.

Cook, A. R. and Cleveland, D. E. (1974). "Detection of Freeway Capacity-Reducing Incidents by Traffic-Stream Measurements". *Transportation Research Record*, No.495, pp.1-11.

Courage, K. G. and Levin, M. (1968). A Freeway Corridor Surveillance, Information and Control System. Research Report 488-8 Texas Transportation Institute, Texas A&M University, College Station.

Dia, H., Rose, G., and Snell, A. (1996). "Comparative Performance of Freeway Automated Incident Detection Algorithms". *Proceedings of 18th ARRB Conference*, 18(7), pp. 359-374.

Dia, H. and Rose, G. (1997). Development and Evaluation of Neural Network Freeway Incident Detection Models Using Field Data. *Transportation Research Part C.* Vol.1, No. 3, pp. 203-217.

Dudek, C.L. and Messer, C.J. (1974). "Incident Detection on Urban Freeways". *Transportation Research Record,* No. 495, pp. 12–24.

Gall, A. I. and Hall, F. L. (1989). "Distinguishing Between Incident Congestion and Recurrent Congestion: A Proposed Logic". *Transportation Research Record*, No.1232, pp. 1-8.

Greene, C. S., Houpt, P. K., Willsky, A. S. and Gershwin, S. B. (1977). Dynamic Detection and Identification of Incidents on Freeways Volume 3: The Multiple Model Method. ESL-R-766 Electronic Systems Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Hall, F. L., Shi, Y., and Atala, G. (1993). Off-Line Testing of the McMaster Incident Deteciton Algorithm Under Recurrent Congestion. *Transportation Research Record,* No. 1394, pp. 1-7, TRB, National Research Council, Washington D.C.

Hsiao, C-H., Lin, C-T., and Cassidy, M. (1994). Application of Fuzzy Logic and Neural Networks to Automatically Detect Freeway Traffic Incidents. *Journal of Transportation Engineering.* Vol. 120, No. 5, pp. 753-771.

Ishak, S. S. and Al-Deek, H. M. (1998). "Fuzzy Art Neural Network Model for Automated Detection of Freeway Incidents". *Transportation Research Record*, No.1634, pp. 56-63.

Kurkijian, A., Gershwin, S. B., Houpt, P. K. and Willsky, A. S. (1977). Dynamic Detection and Identification of Incidents on Freeways Volume 2 : Approaches to Incident Detection using Presence Detectors. ESL-R-765 Electronic Systems Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Lin, C.-K. and Chang, G.-L. (1998). Development of a Fuzzy-Expert System for Incident Detection and Classification. *Mathematical and Computer Modeling.* Vol. 27, No. 9-11, pp. 9-25.

Lindley, J. A. (1996). Qualication of Urban Freeway Congestion and Analysis of Remedial Measures, *Technical Report RD/87-052*, FHWA, Washington, D.C.

Payne, H.J., Helfenbein, E.D., and Knobel, H.C. (1976). Development and Testing of Incident-Detection Algorithms: Vol. 2, Research Methodology and Detailed Results, Technical Report FHWA-RD-76-12, Federal Highway Administration, Washington, DC.

Payne, H. J. and Tignor, S. C. (1978). "Freeway Incident-Detection Algorithms Based on Decision Trees with States". *Transportation Research Record*, No.682, pp. 30-37.

Persaud, B.N., Hall, F.L., and Hall, L.M. (1981). Congestion Identification Aspects of the McMaster Incident Detection Algorithm. Transportation Research Record No. 1287. Pp. 167-175.

Roy, P., and Abdulhai, B.(2004). “GAID: Genetic Adaptive Incident Detection for Freeways”. *Transportation Research Record No. 1856*, pp 96-106,

Samant, A. and Adeli, H. (2001). "Enhancing Neural Network Traffic Incident-Detection Algorithms Using Wavelets". *Computer-Aided Civil and Infrastructure Engineering*, Vol.16 No.4, pp. 239-245.

Shue, J. (2002). A Fuzzy Clustering-Based Approach to Automatic Freeway
Incident Detection and Characterization. *Fuzzy Sets and Systems,* No. 128, pp. 377–388.

Srinivasan, D., Cheu, R.L., and Poh, Y.P. (2001). “Hybrid Fuzzy Logic-Genetic Algorithm Technique for Automated Detection of Traffic Incidents on Freeways”. *2001 IEEE Intelligent Transportation Systems Conference Proceedings.* pp. 25-29. Oakland (CA).

Williams, B.M. and Guin, A. (June 2007). “Traffic Management Center use of Incident Detection Algorithms: Findings of a Nationwide Survey”. *IEEE Transactions on Intelligent Transportation Systems,* No. 8, Vol. 2, pp. 351-358.

CONCLUSIONS AND RECOMMENDATIONS

Introduction

This study, written in journal article form, presents three papers centered on automatic incident detection. Working in building block fashion, each article addresses the subject from a different perspective. As a whole the three papers form a coherent body of knowledge that introduce a new methodology, define a method for its implementation, and compare the method to others in the field.

The Three Journal Articles

Article One- *Traffic Incident Detection Algorithm for Emergency Evacuation Using Real Data*

This article introduces the Alabama Freeway Incident Detection System-Incident Detection Module (AFIDS-IDM) as a methodology for the detection of freeway incidents. AFIDS-IDM incorporates fuzzy cluster analysis with traffic research reported in the *2008 Highway Capacity Manual* in the identification of lane blocking incidents from comparisons of time varying patterns of incident induced and incident free traffic states. Lane traffic counts and density, collected at successive traffic sensors, are the two primary types of input data. State variables are defined from the spatial and temporal relationships of the raw data, and then evaluated quantitatively and qualitatively to determine the decision variables necessary for the determination of lane blocking incidents. The specified decision variable is then compared to a fuzzy cluster analysis algorithm to determine the existence of a lane blocking incident.

AFIDS-IDM was evaluated using data supplied through the South Carolina Department of Transportation (SCDOT). Performance results indicate that AFIDS-IDM was able to identify traffic incidents across a range of road sections.

Article Two- *Incident Characterization and Re-routing After Automatic Incident Detection*

Article Two defines the Alabama Freeway Incident Detection System- Incident Characterization Module (AFIDS-ICM) as a methodology for the characterization of traffic incidents identified by the AFIDS-ID. The method characterizes incidents through the firing of a series of fuzzy based rules that prioritize traffic zones as “green”, “yellow”, or “red” to indicate severity of traffic conditions, with red conditions consider the highest priority. The Alabama Freeway Incident Detection System-Rerouting Module (AFIDS-RM) is employed when "red” incidents are recognized. This module selects alternate routes using Geo-Mapping. Users are presented with a decision support system with computer based interface as a means of implementing AFIDS-ICM and AFIDS-RM.

Article Three- *Comparison of Five Algorithms for Automatic Freeway Incident Detection*

It is the purpose of Article Three to survey current literature related to Automatic Incident Detection (AID) and to list present methods in this field of study. A short list of algorithms is determined from a set of criteria designed to identify algorithms capable of implementation to real-world traffic incident detection. The algorithms are then implemented using real data collected by the South Carolina Department of Transportation (SCDOT) to determine the number and location of traffic incidents identified by each algorithm. The five algorithms selected for comparison are the California Algorithm #8, the Exponential Smoothing Algorithm, the McMaster Incident Detection Algorithm, the Shue Algorithm, and the Alabama Freeway Incident Detection System- Incident Detection Module Algorithm.

Further Development

Further development of the study presented through this dissertation lies in its boundaries. As such, a number of limitations become apparent. Chief among these is the use of the single South Carolina Department of Transportation (SCDOT) data set, and the limited resources made available to Department of Transportation (DOT) users.

The chief limitation of the study lies in the South Carolina Department of Transportation (SCDOT) data set, which was somewhat incomplete in that it did not provide the actual traffic incident data necessary for the determination of real-time false alarm rates (FAR), detection rates (DR), and mean time to detect incident (MTTD), nor did it provide data in any means other than at hourly intervals. While this data is seldom maintained by Department of Transportations (DOTs), simulation offers the prospect of making reasonable determinations from combinations of computer-generated traffic conditions and actual DOT data sets that allow the determination of these metrics. It would be expected then, that this work would be extended using this methodology.

A second limitation of the study lies in the use of a single data set. This restricts the testing of the proposed methodology by presenting a limited exposure to climatic and seasonal changes that impact traffic patterns. Additional research is needed in testing the methodology with data sets that exhibit extreme conditions to test the limits of the methodology.

While the methodology presented is in a plain and uncomplicated form, it is the result of the infant nature of the proposed system. This is obvious from the limited support provided to end users. Additional development is needed in the identification and incorporation of supporting information such as weather conditions surrounding occurred incidents, congestion of adjoining traffic arteries, and location of first responders. This information can be invaluable in the routing of first responders to incident areas.

Potential Use in Education

The proposed methodology has a number of uses within the educational community. Chief among these are the disproportional number of evacuations that plague the educational community and the vulnerability of school age students to real or perceived danger.

Because the majority of school age students are congregated in relatively close spaces without immediate transportation, issues related to inclement weather and other issues often necessitate their evacuation to safer locations. This creates a need to insure that any resulting traffic incidents be recognized as quickly as possible to facilitate their clean up and return of traffic to normal conditions.

Failure to insure the safe evacuation of school age students can result in an increased perception of the immediate situation that can grow into hysteria, adding to the value of the propose system to the school system. The immediate recognition and characterization of incidents can lead to an increased sense of security for those being evacuated.

REFERANCES

Bezdek, J.C. (1973). Fuzzy Mathematics in Pattern Classification. Ph.D. diss., Cornell University.

Hall, L.O., Bensaid, A.M. Clarke, L.P., Velthuizen, R.P., Silbiger, M.S., and Bezdek, J.C. (1992). A Comparison of Neural Network and Fuzzy Clustering techniques in Segmenting Magnetic Resonance Images of the Brain. *IEEE Trans. Neural Networks,* Vol. 3, No. 5, pp. 672–682.

Nowakowski, C., Green, P., Kojima, M. (1999). *Human Factors in Traffic Management Centers: A Literature Review.* (Technical Report UMTRI-99-5), Ann Arbor, MI: The University of Michigan Transportation Research Institute.

Sugeno, M. and Yasukawa, T. (1993). A Fuzzy-Logic-Based Approach to Qualitative Modeling, *IEEE Trans. Systems* Vol.24, No. 6, pp. 1279–1284.

Transportation Research Board. (2008). *Highway Capacity Manual* (3rd ed.), Washington, DC: National Academy of Sciences.

APPENDIX A

CODE FOR CALIFORNIA ALGORITHM #8

```
j=1;
for i=1:3:4103
    occa(j)=a(i)/(a(i)+a(i+1)+a(i+2));
    occa(j+1)=a(i+1)/(a(i)+a(i+1)+a(i+2));
    occa(j+2)=a(i+2)/(a(i)+a(i+1)+a(i+2));
    j=j+3;
end
j=1;
for i=1:3:4103
    occb(j)=b(i)/(b(i)+b(i+1)+b(i+2));
    occb(j+1)=b(i+1)/(b(i)+b(i+1)+b(i+2));
    occb(j+2)=b(i+2)/(b(i)+b(i+1)+b(i+2));
    j=j+3;
end
```

```
j=1;
for i=1:3:4103
    lanea1(j)=a(i);
    lanea2(j)=a(i+1);
    lanea3(j)=a(i+2);
    j=j+1;
end
j=1;
for i=1:3:4103
    laneb1(j)=b(i);
    laneb2(j)=b(i+1);
    laneb3(j)=b(i+2);
    j=j+1;
end
    k=1;
    j=1;
    l=1;
```

```
for i=1:24:660
        for k=1:24
           olaneaa1(l,k)=occa(i);
           olaneaa2(l,k)=occa(i);
           olaneaa3(l,k)=occa(i);
           k=k+1;
           i=i+1;
        end
    l=l+1;
end
k=1;
j=1;
l=1;
for i=1:24:660
        for k=1:24
           olanebb1(l,k)=occb(i);
           olanebb2(l,k)=occb(i);
```

```
        olanebb3(l,k)=occb(i);
        j=j+1;
        k=k+1;
        i=i+1;
      end
   l=l+1;
end
k=1;
j=1;
l=1;
for i=1:24:660
      for k=1:24
        laneaa1(l,k)=lanea1(i);
        laneaa2(l,k)=lanea2(i);
        laneaa3(l,k)=lanea3(i);
        k=k+1;
        i=i+1;
```

```
        end
    l=l+1;
end
k=1;
j=1;
l=1;
for i=1:24:660
        for k=1:24
          lanebb1(l,k)=laneb1(i);
          lanebb2(l,k)=laneb2(i);
          lanebb3(l,k)=laneb3(i);
          j=j+1;
          k=k+1;
          i=i+1;
        end
    l=l+1;
end
```

```
    %
    %   THRESHOLD VALUES
    %
% T1=0;
count=0;
k=1;
 l=1;
 pt1=0
 pt2=0
for i=1:28
   for j=3:24
      OCCDF(i,j)=lanebb1(i,j)-laneaa1(i,j);
      if OCCDF(i,j)>398
         OCCRDF(i,j)=OCCDF(i,j)/laneaa1(i,j);
         if OCCRDF(i,j)>5
            DOCCTD(i,j)=(lanebb1(i,j-2)-lanebb1(i,j))/lanebb1(i,j-2);
            if DOCCTD(i,j)>0.1540
```

```
                count=count+1;
                i;
                j;
                pt1(l)=olanebb1(i,j);
                pt2(l)=lanebb1(i,j);
                l=l+1
%                   plot(olanebb1(i,j),lanebb1(i,j),'-o')
%                    hold on
            end
         end
      end
   end
end
count
pt1
pt2
% a=[0.4124,0.4097,0.4066]
```

```
% b=[554,496,456]

% %       FOR SECTION OF ROAD FROM ATR_ID_196 TO ID_197   %

%

% j=1;

% for i=1:3:5399

%       occa(j)=a(i)/(a(i)+a(i+1)+a(i+2));

%       occa(j+1)=a(i+1)/(a(i)+a(i+1)+a(i+2));

%       occa(j+2)=a(i+2)/(a(i)+a(i+1)+a(i+2));

%       j=j+3;

% end

% j=1;

% for i=1:3:4103

%       occb(j)=b(i)/(b(i)+b(i+1)+b(i+2));

%       occb(j+1)=b(i+1)/(b(i)+b(i+1)+b(i+2));

%       occb(j+2)=b(i+2)/(b(i)+b(i+1)+b(i+2));

%       j=j+3;

% end
```

```
% j=1
% for i=1:3:5399
%       lanea1(j)=a(i);
%       lanea2(j)=a(i+1);
%       lanea3(j)=a(i+2);
%       j=j+1;
% end
% j=1;
% for i=1:3:4103
%       laneb1(j)=b(i);
%       laneb2(j)=b(i+1);
%       laneb3(j)=b(i+2);
%       j=j+1;
% end
% % j
%       k=1;
%       j=1;
```

```
%       l=1;
%       for i=1:24:1368
%           for k=1:24
%               olaneaa1(l,k)=occa(i);
%               olaneaa2(l,k)=occa(i);
%               olaneaa3(l,k)=occa(i);
%               k=k+1;
%               i=i+1;
%           end
%         l=l+1;
%       end
%       k=1;
%       j=1;
%       l=1;
%       for i=1:24:1368
%           for k=1:24
%               olanebb1(l,k)=occb(i);
```

```
%                 olanebb2(l,k)=occb(i);
%                 olanebb3(l,k)=occb(i);
%                 j=j+1;
%                 k=k+1;
%                 i=i+1;
%               end
%          l=l+1;
%       end
%        k=1;
%       j=1;
%       l=1;
%       for i=1:24:1368
%               for k=1:24
%                 laneaa1(l,k)=lanea1(i);
%                 laneaa2(l,k)=lanea2(i);
%                 laneaa3(l,k)=lanea3(i);
%                 k=k+1;
```

```
%                 i=i+1;
%               end
%           l=l+1;
%         end
%         k=1;
%         j=1;
%         l=1;
%         for i=1:24:1368
%               for k=1:24
%                 lanebb1(l,k)=laneb1(i);
%                 lanebb2(l,k)=laneb2(i);
%                 lanebb3(l,k)=laneb3(i);
%                 j=j+1;
%                 k=k+1;
%                 i=i+1;
%               end
%           l=l+1;
```

```
%       end
% %       l
% %       %
% %       %
%
% % T1=0;
% count=0;
%  k=1;
% for i=1:57
%    for j=3:24
%       OCCDF(i,j)=lanebb1(i,j)-laneaa1(i,j);
%        if OCCDF(i,j)> 268
%           OCCRDF(i,j)=OCCDF(i,j)/laneaa1(i,j);
%           if OCCRDF(i,j)>3.92
%               DOCCTD(i,j)=(lanebb1(i,j-2)-lanebb1(i,j))/lanebb1(i,j-2);
%                if DOCCTD(i,j)> 0.1358
%                    count=count+1;
```

```
%                    i;

%                    j;

%

%               end

%           end

%       end

%    end

% end

% count

% % T1

%

% %        FOR SECTION OF ROAD FROM ATR_ID_242 TO ID_243   %

%

% j=1;

% for i=1:3:4127

%        occa(j)=a(i)/(a(i)+a(i+1)+a(i+2));

%        occa(j+1)=a(i+1)/(a(i)+a(i+1)+a(i+2));
```

```
%       occa(j+2)=a(i+2)/(a(i)+a(i+1)+a(i+2));

%       j=j+3;

% end

% j=1;

% for i=1:3:4127

%       occb(j)=b(i)/(b(i)+b(i+1)+b(i+2));

%       occb(j+1)=b(i+1)/(b(i)+b(i+1)+b(i+2));

%       occb(j+2)=b(i+2)/(b(i)+b(i+1)+b(i+2));

%       j=j+3;

% end

% j=1;

% for i=1:3:4127

%       lanea1(j)=a(i);

%       lanea2(j)=a(i+1);

%       lanea3(j)=a(i+2);

%       j=j+1;

% end
```

```
% j=1;
% for i=1:3:4127
%       laneb1(j)=b(i);
%       laneb2(j)=b(i+1);
%       laneb3(j)=b(i+2);
%       j=j+1;
% end
% %  j
%       k=1;
%       j=1;
%       l=1;
%       for i=1:24:1377
%           for k=1:24
%               olaneaa1(l,k)=occa(i);
%               olaneaa2(l,k)=occa(i);
%               olaneaa3(l,k)=occa(i);
%               k=k+1;
```

```
%               i=i+1;
%             end
%         l=l+1;
%     end
%     k=1;
%     j=1;
%     l=1;
%     for i=1:24:1368
%             for k=1:24
%               olanebb1(l,k)=occb(i);
%               olanebb2(l,k)=occb(i);
%               olanebb3(l,k)=occb(i);
%               k=k+1;
%               i=i+1;
%             end
%         l=l+1;
%     end
```

```
%       k=1;
%       j=1;
%       l=1;
%       for i=1:24:1377
%           for k=1:24
%               laneaa1(l,k)=lanea1(i);
%               laneaa2(l,k)=lanea2(i);
%               laneaa3(l,k)=lanea3(i);
%               k=k+1;
%               i=i+1;
%           end
%         l=l+1;
%       end
%       k=1;
%       j=1;
%       l=1;
%       for i=1:24:1368
```

```
%                   for k=1:24
%                      lanebb1(l,k)=laneb1(i);
%                      lanebb2(l,k)=laneb2(i);
%                      lanebb3(l,k)=laneb3(i);
%                      k=k+1;
%                      i=i+1;
%                   end
%              l=l+1;
%          end
% %         l
% % %         %
% % %         %   THRESHOLD VALUES
% % %         %
% %
% % T1=0;
% count=0;
%  k=1;
```

```
% for i=1:57
%    for j=3:24
%       OCCDF(i,j)=lanebb1(i,j)-laneaa1(i,j);
%        if OCCDF(i,j)> 229           %T1
% %             T1=OCCDF(i,j)
%           OCCRDF(i,j)=OCCDF(i,j)/laneaa1(i,j);
%            if OCCRDF(i,j)>   3.6
% %                T1=OCCRDF(i,j)
%               DOCCTD(i,j)=(lanebb1(i,j-2)-lanebb1(i,j))/lanebb1(i,j-2);
%               if DOCCTD(i,j)>   0.2578
%                   count=count+1;
%                   i;
%                   j;
%               end
%           end
%       end
%    end
```

```
% end
%  count
% %T1
%
%
% %       FOR SECTION OF ROAD FROM ATR_ID_243 TO ID_244   %
% j=1;
% for i=1:3:5831
%        occa(j)=a(i)/(a(i)+a(i+1)+a(i+2));
%        occa(j+1)=a(i+1)/(a(i)+a(i+1)+a(i+2));
%        occa(j+2)=a(i+2)/(a(i)+a(i+1)+a(i+2));
%        j=j+3;
% end
% j=1;
% for i=1:3:5831
%        occb(j)=b(i)/(b(i)+b(i+1)+b(i+2));
%        occb(j+1)=b(i+1)/(b(i)+b(i+1)+b(i+2));
```

```
%       occb(j+2)=b(i+2)/(b(i)+b(i+1)+b(i+2));
%       j=j+3;
% end
% j=1;
% for i=1:3:5831
%       lanea1(j)=a(i);
%       lanea2(j)=a(i+1);
%       lanea3(j)=a(i+2);
%       j=j+1;
% end
% j=1;
% for i=1:3:5831
%       laneb1(j)=b(i);
%       laneb2(j)=b(i+1);
%       laneb3(j)=b(i+2);
%       j=j+1;
% end
```

```
% %   j
%       k=1;
%       j=1;
%       l=1;
%       for i=1:24:1944
%             for k=1:24
%               olaneaa1(l,k)=occa(i);
%               olaneaa2(l,k)=occa(i);
%               olaneaa3(l,k)=occa(i);
%               k=k+1;
%               i=i+1;
%             end
%          l=l+1;
%       end
%       k=1;
%       j=1;
%       l=1;
```

```
%       for i=1:24:1944
%              for k=1:24
%                 olanebb1(l,k)=occb(i);
%                 olanebb2(l,k)=occb(i);
%                 olanebb3(l,k)=occb(i);
%                 j=j+1;
%                 k=k+1;
%                 i=i+1;
%              end
%          l=l+1;
%       end
%       k=1;
%       j=1;
%       l=1;
%       for i=1:24:1944
%              for k=1:24
%                 laneaa1(l,k)=lanea1(i);
```

```
%               laneaa2(l,k)=lanea2(i);
%               laneaa3(l,k)=lanea3(i);
%               k=k+1;
%               i=i+1;
%             end
%          l=l+1;
%       end
%       k=1;
%       j=1;
%       l=1;
%       for i=1:24:1944
%             for k=1:24
%               lanebb1(l,k)=laneb1(i);
%               lanebb2(l,k)=laneb2(i);
%               lanebb3(l,k)=laneb3(i);
%               j=j+1;
%               k=k+1;
```

```
%                    i=i+1;
%                 end
%            l=l+1;
%        end
% %        l
% %        %
% %        %   THRESHOLD VALUES
% %        %
%
% T1=0;
% count=0;
DOCCRDF
%  k=1;
% for i=1:81
%     for j=3:24
%        OCCDF(i,j)=lanebb1(i,j)-laneaa1(i,j);
%         if OCCDF(i,j)>  217   %T1
```

```
% %           T1=OCCDF(i,j)

%           OCCRDF(i,j)=OCCDF(i,j)/laneaa1(i,j);

%           if OCCRDF(i,j)> 0.3036

% %                 T1=OCCRDF(i,j)

%             DOCCTD(i,j)=(lanebb1(i,j-2)-lanebb1(i,j))/lanebb1(i,j-2);

%             if DOCCTD(i,j)>   0.21235

% %                   T1 = DOCCTD(i,j)

%                 count=count+1;

%                 i;

%                 j;

%

%             end

%           end

%       end

%     end
```

% end

% % T1

% count

APPENDIX B

CODE FOR EXPONENTIAL SMOOTHING ALGORITHM

```
% SMOOTHENING ALGORITHM

function f1=smotheningalgorithm()

fid=fopen('thesis1.xls');

%       ALGORITHM FOR FINDING THE INCIDENTS ON THE LANE        %

%    KEEPS TRACK OF NUMBER OF INCIDENTS

%       FOR SECTION OF ROAD FROM ATR_ID_194 TO ID_195   %

j=1;
for i=1:3:4103
        lanea1(j)=a(i);
        lanea2(j)=a(i+1);
        lanea3(j)=a(i+2);
        j=j+1;
end
j=1;
```

```
for i=1:3:4103
        laneb1(j)=b(i);
        laneb2(j)=b(i+1);
        laneb3(j)=b(i+2);
        j=j+1;
end
        k=1;
        j=1;
        l=1;
        for i=1:24:660
                 for k=1:24
                    laneaa1(l,k)=lanea1(i);
                    laneaa2(l,k)=lanea2(i);
                    laneaa3(l,k)=lanea3(i);
                    k=k+1;
                    i=i+1;
                 end
            l=l+1;
        end
        k=1;
        j=1;
        l=1;
        for i=1:24:660
                 for k=1:24
                    lanebb1(l,k)=laneb1(i);
                    lanebb2(l,k)=laneb2(i);
                    lanebb3(l,k)=laneb3(i);
```

```
                j=j+1;
                k=k+1;
                i=i+1;
              end
          l=l+1;
        end
        %
        %   THRESHOLD VALUES
        %

T1=0;
T2=0;
count=0;
k=1;
for i=1:28
   for j=1:24
     OCCDFA(i,j)=laneaa2(i,j)-laneaa1(i,j);
     OCCDFB(i,j)=lanebb2(i,j)-lanebb1(i,j);
     if OCCDFA(i,j)>  50
        if OCCDFB(i,j)>  100
                 count=count+1;
                 i
                 j
        end
     end
   end
end
```

```
% T1=0
count
count=0;
for i=1:28
   for j=1:24
      OCCDFA(i,j)=laneaa3(i,j)-laneaa2(i,j);
      OCCDFB(i,j)=lanebb3(i,j)-lanebb2(i,j);
      if OCCDFB(i,j)> 85
%           T1=OCCDFB(i,j)
         if OCCDFB(i,j) >  69
               count=count+1;
               i
               j
         end
      end
   end
end
% T1
count

%      FOR SECTION OF ROAD FROM ATR_ID_196 TO ID_197   %

j=1;
for i=1:3:5399
      lanea1(j)=a(i);
      lanea2(j)=a(i+1);
      lanea3(j)=a(i+2);
```

```
        j=j+1;
end
j=1;
for i=1:3:5399
        laneb1(j)=b(i);
        laneb2(j)=b(i+1);
        laneb3(j)=b(i+2);
        j=j+1;
end
% j
        k=1;
        j=1;
        l=1;
        for i=1:24:1368
                 for k=1:24
                   laneaa1(l,k)=lanea1(i);
                   laneaa2(l,k)=lanea2(i);
                   laneaa3(l,k)=lanea3(i);
                   k=k+1;
                   i=i+1;
                 end
            l=l+1;
        end
        k=1;
        j=1;
        l=1;
        for i=1:24:1368
```

```
                for k=1:24
                  lanebb1(l,k)=laneb1(i);
                  lanebb2(l,k)=laneb2(i);
                  lanebb3(l,k)=laneb3(i);
                  j=j+1;
                  k=k+1;
                  i=i+1;
                end
             l=l+1;
          end
% l
%        %   THRESHOLD VALUES
%        %

% T1=0;
count=0;
k=1;
for i=1:57
   for j=1:24
      OCCDFA(i,j)=laneaa2(i,j)-laneaa1(i,j);
      OCCDFB(i,j)=lanebb2(i,j)-lanebb1(i,j);
       if OCCDFA(i,j)>  166
%           T1=OCCDFB(i,j);
           if OCCDFB(i,j)>  151
                 count=count+1;
                 i
                 j
```

```
            end
        end
    end
end
count
count=0;
for i=1:57
    for j=1:24
       OCCDFA(i,j)=laneaa3(i,j)-laneaa2(i,j);
       OCCDFB(i,j)=lanebb3(i,j)-lanebb2(i,j);
        if OCCDFA(i,j)>   61
%            T1=OCCDFB(i,j);
%             if OCCDFB(i,j)>  151
                count=count+1;
                i
                j
%             end
        end
    end
end
count
%  T1
%
%       FOR SECTION OF ROAD FROM ATR_ID_242 TO ID_243   %
a=xlsread('thesis1.xls','Z21668:Z25795');
b=xlsread('thesis1.xls','Z25796:Z29923');
j=1;
```

```
for i=1:3:4127
        lanea1(j)=a(i);
        lanea2(j)=a(i+1);
        lanea3(j)=a(i+2);
        j=j+1;
end
j=1;
for i=1:3:4127
        laneb1(j)=b(i);
        laneb2(j)=b(i+1);
        laneb3(j)=b(i+2);
        j=j+1;
end
%  j
        k=1;
        j=1;
        l=1;
        for i=1:24:1377
                 for k=1:24
                    laneaa1(l,k)=lanea1(i);
                    laneaa2(l,k)=lanea2(i);
                    laneaa3(l,k)=lanea3(i);
                    k=k+1;
                    i=i+1;
                 end
            l=l+1;
        end
```

```
     k=1;
     j=1;
     l=1;
     for i=1:24:1377
          for k=1:24
            lanebb1(l,k)=laneb1(i);
            lanebb2(l,k)=laneb2(i);
            lanebb3(l,k)=laneb3(i);
            j=j+1;
            k=k+1;
            i=i+1;
          end
       l=l+1;
     end
%      l
%      %
%      %   THRESHOLD VALUES
%      %

T1=0;
count=0;
%  k=1;

for i=1:57
  for j=1:24
    OCCDFA(i,j)=laneaa2(i,j)-laneaa1(i,j);
    OCCDFB(i,j)=lanebb2(i,j)-lanebb1(i,j);
```

```
      if OCCDFA(i,j)>   129
%            T1=OCCDFB(i,j);
           if OCCDFB(i,j)>  149
                  count=count+1;
                  i
                  j
           end
      end
   end
end
%T1
count
count=0;
T1=0;
for i=1:57
   for j=1:24
     OCCDFA(i,j)=laneaa3(i,j)-laneaa2(i,j);
     OCCDFB(i,j)=lanebb3(i,j)-lanebb2(i,j);
      if OCCDFB(i,j)>  217
%            T1=OCCDFB(i,j);
          if OCCDFB(i,j)>  150
                 count=count+1;
                 i
                 j
          end
      end
   end
```

```
end
count
%T1
%
%
% %     FOR SECTION OF ROAD FROM ATR_ID_243 TO ID_244   %
j=1;
for i=1:3:5831
    lanea1(j)=a(i);
    lanea2(j)=a(i+1);
    lanea3(j)=a(i+2);
    j=j+1;
end
j=1;
for i=1:3:5831
    laneb1(j)=b(i);
    laneb2(j)=b(i+1);
    laneb3(j)=b(i+2);
    j=j+1;
end
%   j
    k=1;
    j=1;
    l=1;
    for i=1:24:1944
        for k=1:24
          laneaa1(l,k)=lanea1(i);
```

```
            laneaa2(l,k)=lanea2(i);
            laneaa3(l,k)=lanea3(i);
            k=k+1;
            i=i+1;
          end
       l=l+1;
     end
     k=1;
     j=1;
     l=1;
     for i=1:24:1944
          for k=1:24
            lanebb1(l,k)=laneb1(i);
            lanebb2(l,k)=laneb2(i);
            lanebb3(l,k)=laneb3(i);
            j=j+1;
            k=k+1;
            i=i+1;
          end
       l=l+1;
     end
%       l
%       %
%       %   THRESHOLD VALUES
%       %

T1=0;
```

```
count=0;
%  k=1;
for i=1:81
   for j=3:24
      OCCDFA(i,j)=laneaa2(i,j)-laneaa1(i,j);
      OCCDFB(i,j)=lanebb2(i,j)-lanebb1(i,j);
      if OCCDFA(i,j)>  100
%            T1=OCCDFB(i,j);
            if OCCDFB(i,j)>  140
                 count=count+1;
                 i
                 j
            end
      end
   end
end
% T1
% T1=0;
count
count=0
for i=1:81
   for j=3:24
      OCCDFA(i,j)=laneaa3(i,j)-laneaa2(i,j);
      OCCDFB(i,j)=lanebb3(i,j)-lanebb2(i,j);
      if OCCDFA(i,j)> 12
%             T1=OCCDFB(i,j);
           if OCCDFB(i,j)>  35
```

```
                count=count+1;
                i
                j
            end
        end
    end
end
% T1
 Count
```

APPENDIX C

CODE FOR MCMASTER ALGORITHM

```
% McMaster ALGORITHM

function f1=mcmaster()



%                                          %
%                                          %
%       ALGORITHM FOR FINDING THE INCIDENTS ON THE LANE        %
%                                          %
%                                          %



%    KEEPS TRACK OF NUMBER OF INCIDENTS



%      FOR SECTION OF ROAD FROM ATR_ID_194 TO ID_195   %



a=xlsread('thesis1.xls','Z2660:Z6763');
b=xlsread('thesis1.xls','Z6764:Z10867');
j=1;
for i=1:3:4103
     occa(j)=a(i)/(a(i)+a(i+1)+a(i+2));
     occa(j+1)=a(i+1)/(a(i)+a(i+1)+a(i+2));
```

```
        occa(j+2)=a(i+2)/(a(i)+a(i+1)+a(i+2));
        j=j+3;
end
j=1;
for i=1:3:4103
        occb(j)=b(i)/(b(i)+b(i+1)+b(i+2));
        occb(j+1)=b(i+1)/(b(i)+b(i+1)+b(i+2));
        occb(j+2)=b(i+2)/(b(i)+b(i+1)+b(i+2));
        j=j+3;
end
j=1;
for i=1:3:4103
        lanea1(j)=a(i);
        lanea2(j)=a(i+1);
        lanea3(j)=a(i+2);
        j=j+1;
end
j=1;
for i=1:3:4103
        laneb1(j)=b(i);
        laneb2(j)=b(i+1);
        laneb3(j)=b(i+2);
        j=j+1;
end
        k=1;
        j=1;
        l=1;
```

```
for i=1:24:660
        for k=1:24
          olaneaa1(l,k)=occa(i);
          olaneaa2(l,k)=occa(i);
          olaneaa3(l,k)=occa(i);
          k=k+1;
          i=i+1;
        end
    l=l+1;
end
k=1;
j=1;
l=1;
for i=1:24:660
        for k=1:24
          olanebb1(l,k)=occb(i);
          olanebb2(l,k)=occb(i);
          olanebb3(l,k)=occb(i);
          j=j+1;
          k=k+1;
          i=i+1;
        end
    l=l+1;
end
k=1;
j=1;
l=1;
```

```
for i=1:24:660
        for k=1:24
          laneaa1(l,k)=lanea1(i);
          laneaa2(l,k)=lanea2(i);
          laneaa3(l,k)=lanea3(i);
          k=k+1;
          i=i+1;
        end
   l=l+1;
end
k=1;
j=1;
l=1;
for i=1:24:660
        for k=1:24
          lanebb1(l,k)=laneb1(i);
          lanebb2(l,k)=laneb2(i);
          lanebb3(l,k)=laneb3(i);
          j=j+1;
          k=k+1;
          i=i+1;
        end
   l=l+1;
end
%
%   THRESHOLD VALUES
%
```

```
% T1=0;
count=0;
%   THE NO. OF ACCIDENTS TAKEN THESE THRESHOLD VALUES IS 6
 k=1;
 l=1;
 pt1=0
 pt2=0
for i=1:28
   for j=3:24
      OCCDF(i,j)=lanebb1(i,j)-laneaa1(i,j);
      if OCCDF(i,j)>398
         OCCRDF(i,j)=OCCDF(i,j)/laneaa1(i,j);
         if OCCRDF(i,j)>5
            DOCCTD(i,j)=(lanebb1(i,j-2)-lanebb1(i,j))/lanebb1(i,j-2);
            if DOCCTD(i,j)>0.1540
                count=count+1;
                i;
                j;
                pt1(l)=olanebb1(i,j);
                pt2(l)=lanebb1(i,j);
                l=l+1
%                   plot(olanebb1(i,j),lanebb1(i,j),'-o')
%                     hold on
            end
         end
      end
```

```
    end
end
count
pt1
pt2
% a=[0.4124,0.4097,0.4066]
% b=[554,496,456]
% plot(a,b,'o')

plot(pt1,pt2,'o')



%         % %        FOR SECTION OF ROAD FROM ATR_ID_196 TO ID_197   %
%
% j=1;
% for i=1:3:5399
%         occa(j)=a(i)/(a(i)+a(i+1)+a(i+2));
%         occa(j+1)=a(i+1)/(a(i)+a(i+1)+a(i+2));
%         occa(j+2)=a(i+2)/(a(i)+a(i+1)+a(i+2));
%         j=j+3;
% end
% j=1;
% for i=1:3:4103
%         occb(j)=b(i)/(b(i)+b(i+1)+b(i+2));
%         occb(j+1)=b(i+1)/(b(i)+b(i+1)+b(i+2));
%         occb(j+2)=b(i+2)/(b(i)+b(i+1)+b(i+2));
%         j=j+3;
```

```
% end
% j=1
% for i=1:3:5399
%       lanea1(j)=a(i);
%       lanea2(j)=a(i+1);
%       lanea3(j)=a(i+2);
%       j=j+1;
% end
% j=1;
% for i=1:3:4103
%       laneb1(j)=b(i);
%       laneb2(j)=b(i+1);
%       laneb3(j)=b(i+2);
%       j=j+1;
% end
% % j
%       k=1;
%       j=1;
%       l=1;
%       for i=1:24:1368
%             for k=1:24
%               olaneaa1(l,k)=occa(i);
%               olaneaa2(l,k)=occa(i);
%               olaneaa3(l,k)=occa(i);
%               k=k+1;
%               i=i+1;
%             end
```

```
%       l=l+1;
%    end
%    k=1;
%    j=1;
%    l=1;
%    for i=1:24:1368
%          for k=1:24
%            olanebb1(l,k)=occb(i);
%            olanebb2(l,k)=occb(i);
%            olanebb3(l,k)=occb(i);
%            j=j+1;
%            k=k+1;
%            i=i+1;
%          end
%       l=l+1;
%    end
%     k=1;
%    j=1;
%    l=1;
%    for i=1:24:1368
%          for k=1:24
%            laneaa1(l,k)=lanea1(i);
%            laneaa2(l,k)=lanea2(i);
%            laneaa3(l,k)=lanea3(i);
%            k=k+1;
%            i=i+1;
%          end
```

```
%           l=l+1;
%       end
%       k=1;
%       j=1;
%       l=1;
%       for i=1:24:1368
%               for k=1:24
%                   lanebb1(l,k)=laneb1(i);
%                   lanebb2(l,k)=laneb2(i);
%                   lanebb3(l,k)=laneb3(i);
%                   j=j+1;
%                   k=k+1;
%                   i=i+1;
%               end
%           l=l+1;
%       end
% %       l
% %       %
% %       %   THRESHOLD VALUES
% %       %
%
% % T1=0;
% count=0;
%  k=1;
% for i=1:57
%    for j=3:24
%       OCCDF(i,j)=lanebb1(i,j)-laneaa1(i,j);
```

```
%         if OCCDF(i,j)> 268
%           OCCRDF(i,j)=OCCDF(i,j)/laneaa1(i,j);
%           if OCCRDF(i,j)>3.92
%              DOCCTD(i,j)=(lanebb1(i,j-2)-lanebb1(i,j))/lanebb1(i,j-2);
%               if DOCCTD(i,j)> 0.1358
%                  count=count+1;
%                  i;
%                  j;
%
%              end
%           end
%        end
%     end
% end
% count
% % T1
%
% %        FOR SECTION OF ROAD FROM ATR_ID_242 TO ID_243   %
%
% a=xlsread('Book1.xlsx','Z21668:Z25795');
% b=xlsread('Book1.xlsx','Z25796:Z29923');
% j=1;
% for i=1:3:4127
%         occa(j)=a(i)/(a(i)+a(i+1)+a(i+2));
%         occa(j+1)=a(i+1)/(a(i)+a(i+1)+a(i+2));
%         occa(j+2)=a(i+2)/(a(i)+a(i+1)+a(i+2));
%         j=j+3;
```

```
% end
% j=1;
% for i=1:3:4127
%       occb(j)=b(i)/(b(i)+b(i+1)+b(i+2));
%       occb(j+1)=b(i+1)/(b(i)+b(i+1)+b(i+2));
%       occb(j+2)=b(i+2)/(b(i)+b(i+1)+b(i+2));
%       j=j+3;
% end
% j=1;
% for i=1:3:4127
%       lanea1(j)=a(i);
%       lanea2(j)=a(i+1);
%       lanea3(j)=a(i+2);
%       j=j+1;
% end
% j=1;
% for i=1:3:4127
%       laneb1(j)=b(i);
%       laneb2(j)=b(i+1);
%       laneb3(j)=b(i+2);
%       j=j+1;
% end
% %  j
%       k=1;
%       j=1;
%       l=1;
%       for i=1:24:1377
```

```
%               for k=1:24
%                 olaneaa1(l,k)=occa(i);
%                 olaneaa2(l,k)=occa(i);
%                 olaneaa3(l,k)=occa(i);
%                 k=k+1;
%                 i=i+1;
%               end
%            l=l+1;
%        end
%        k=1;
%        j=1;
%        l=1;
%        for i=1:24:1368
%               for k=1:24
%                 olanebb1(l,k)=occb(i);
%                 olanebb2(l,k)=occb(i);
%                 olanebb3(l,k)=occb(i);
%                 k=k+1;
%                 i=i+1;
%               end
%            l=l+1;
%        end
%        k=1;
%        j=1;
%        l=1;
%        for i=1:24:1377
%               for k=1:24
```

```
%                   laneaa1(l,k)=lanea1(i);
%                   laneaa2(l,k)=lanea2(i);
%                   laneaa3(l,k)=lanea3(i);
%                   k=k+1;
%                   i=i+1;
%                 end
%             l=l+1;
%         end
%         k=1;
%         j=1;
%         l=1;
%         for i=1:24:1368
%                 for k=1:24
%                   lanebb1(l,k)=laneb1(i);
%                   lanebb2(l,k)=laneb2(i);
%                   lanebb3(l,k)=laneb3(i);
%                   k=k+1;
%                   i=i+1;
%                 end
%             l=l+1;
%         end
% %         l
% % %         %
% % %         %   THRESHOLD VALUES
% % %         %
% %
% % T1=0;
```

```
% count=0;
%  k=1;
% for i=1:57
%     for j=3:24
%        OCCDF(i,j)=lanebb1(i,j)-laneaa1(i,j);
%         if OCCDF(i,j)> 229             %T1
% %              T1=OCCDF(i,j)
%            OCCRDF(i,j)=OCCDF(i,j)/laneaa1(i,j);
%             if OCCRDF(i,j)>   3.6
% %                  T1=OCCRDF(i,j)
%                DOCCTD(i,j)=(lanebb1(i,j-2)-lanebb1(i,j))/lanebb1(i,j-2);
%                 if DOCCTD(i,j)>   0.2578
%                      count=count+1;
%                      i;
%                      j;
%                  end
%            end
%        end
%     end
% end
%  count
% %T1
%
%
% %       FOR SECTION OF ROAD FROM ATR_ID_243 TO ID_244   %
%
% j=1;
```

```
% for i=1:3:5831
%       occa(j)=a(i)/(a(i)+a(i+1)+a(i+2));
%       occa(j+1)=a(i+1)/(a(i)+a(i+1)+a(i+2));
%       occa(j+2)=a(i+2)/(a(i)+a(i+1)+a(i+2));
%       j=j+3;
% end
% j=1;
% for i=1:3:5831
%       occb(j)=b(i)/(b(i)+b(i+1)+b(i+2));
%       occb(j+1)=b(i+1)/(b(i)+b(i+1)+b(i+2));
%       occb(j+2)=b(i+2)/(b(i)+b(i+1)+b(i+2));
%       j=j+3;
% end
% j=1;
% for i=1:3:5831
%       lanea1(j)=a(i);
%       lanea2(j)=a(i+1);
%       lanea3(j)=a(i+2);
%       j=j+1;
% end
% j=1;
% for i=1:3:5831
%       laneb1(j)=b(i);
%       laneb2(j)=b(i+1);
%       laneb3(j)=b(i+2);
%       j=j+1;
% end
```

```
% %   j
%         k=1;
%         j=1;
%         l=1;
%         for i=1:24:1944
%               for k=1:24
%                 olaneaa1(l,k)=occa(i);
%                 olaneaa2(l,k)=occa(i);
%                 olaneaa3(l,k)=occa(i);
%                 k=k+1;
%                 i=i+1;
%               end
%           l=l+1;
%         end
%         k=1;
%         j=1;
%         l=1;
%         for i=1:24:1944
%               for k=1:24
%                 olanebb1(l,k)=occb(i);
%                 olanebb2(l,k)=occb(i);
%                 olanebb3(l,k)=occb(i);
%                 j=j+1;
%                 k=k+1;
%                 i=i+1;
%               end
%           l=l+1;
```

```
%       end
%       k=1;
%       j=1;
%       l=1;
%       for i=1:24:1944
%           for k=1:24
%               laneaa1(l,k)=lanea1(i);
%               laneaa2(l,k)=lanea2(i);
%               laneaa3(l,k)=lanea3(i);
%               k=k+1;
%               i=i+1;
%           end
%         l=l+1;
%       end
%       k=1;
%       j=1;
%       l=1;
%       for i=1:24:1944
%           for k=1:24
%               lanebb1(l,k)=laneb1(i);
%               lanebb2(l,k)=laneb2(i);
%               lanebb3(l,k)=laneb3(i);
%               j=j+1;
%               k=k+1;
%               i=i+1;
%           end
%         l=l+1;
```

```
%        end
% %        l
% %        %
% %        %   THRESHOLD VALUES
% %        %
%
% T1=0;
% count=0;
DOCCRDF
% % %   THE NO. OF ACCIDENTS TAKING THESE THRESHOLD VALUES
IS 2
%  k=1;
% for i=1:81
%     for j=3:24
%        OCCDF(i,j)=lanebb1(i,j)-laneaa1(i,j);
%        if OCCDF(i,j)>  217   %T1
% %            T1=OCCDF(i,j)
%             OCCRDF(i,j)=OCCDF(i,j)/laneaa1(i,j);
%             if OCCRDF(i,j)> 0.3036
% %                  T1=OCCRDF(i,j)
%               DOCCTD(i,j)=(lanebb1(i,j-2)-lanebb1(i,j))/lanebb1(i,j-2);
%               if DOCCTD(i,j)>   0.21235
% %                   T1 = DOCCTD(i,j)
%                    count=count+1;
%                    i;
%                    j;
%
%               end
```

```
%            end
%       end
%    end
% end
% % T1
%  count
```

APPENDIX D

CODE FOR SHUE ALGORITHM

```
% Shue Fuzzy Logic Algorithm

CONGESTION = 0;
Low = 1;
Medium = 2;
High = 3;

% Assign threshold level to Congestion based on historical data
% Congestion = f(lane occupancy, occupancy threshold)

% Using congestion level, determine Incident occurrence
% Each congestion level has a unique equation
% Used to compute a decision variable at time step k

Decision(k) = 0;

% U(i,k) and D(i,k) represent upstream and downstream counts
% at lane i and time k
```

```
% Ou(i,k) & Od(i,k) = collected occupancies of up and downstream
% at lane i and time k

% J = total number of adjacent lanes
% T = maximum time lag
% Lag = time lag index

if Congestion == Low
   Decision(k) = ((Ou(k) - Od(k)) / Ou(k));
end

if Congestion == Medium
   Sum = 0;
   for j=1:J
      Sum = Sum + D(j,k);
   end
   Decision(k) = {((Sum / J) - D(i,k)) * (Sum / J));
end

if Congestion == High
   Sum = 0;
   for j=Lag:T
      Sum = Sum + D(i,k-Lag);
```

```
    end
    Sum1 = 0;
    for j=1:J
        Sum1 = Sum1 + D(j,k);
    end
    Decision(k) = Sum - U(i,k-T) - (Sum1 / J);
end


% Now compute a time varying decision-variable correlation


% B = predterimined value set for the boundaries of w(p.q,m,n,k)
% m = level of congestion
% n = lane code
% p = location parameter
% q = binary digit represting incident or no incident ( 1 or 0)
% k still represents current time step


% u(p.q,m,n,k) = pattern of the decision variable
% on basis of historical data


Sum2 = 0;
for Lag=0:3
    Sum2 = Sum2 + ((Decision(n)*(k-Lag)-u(p.q,m,n,k))^2;
```

```
end
Incident_Decision(k) = 1 - (1/B)*(sqrt(Sum2));

% Recognize an Incident if the following condtion is met
% Judgement is made against a threshold value
% We will use 0.5 for Demonstrative Purpose

Threshold = 0.5;

if (Max(w(1.1,m,n,k),w(2.1,1,m,n,k),w(3.1,m,n,k)) - w(p.0,m.n,k) > Threshold)
   disp('A lane blocking incident with the attributes m and n is recognized at time
step k')
end.
```

APPENDIX E

CODE FOR AFIDS-IDM ALGORITHM

```
% AFIDS ALGORITHM


%j=evacuation route


%SF(i,k)=service flow rate
%c(j)=capacity for the road section under study
%f(p)=further adjustment
%LOC(i,j,k)=level of conjestion index
%LOC = level of congestion


LOC(i,j,k) = (SF(i,k)/c(j))*(1/f(p));


LOC = table(LOC(i,j,k),sl);


% U(i,k) and D(i,k) represent upstream and downstream counts
% at lane i and time k


% Ou(i,k) & Od(i,k) = collected occupancies of up and downstream
% at lane i and time k


%I=total number of adacent lanes
```

```
%T=time lag
%F(p)=speed limit

if LOC == LOW
   Vs(k)=(Ou(k-n)-Od(k))/Ou(k-n)
end

if LOC == MEDIUM
   Vs(k)=((D(k)/J)-D(k))-(D(k)/I)
end

if LOC == HIGH
   Vs(k)=(D(k)*min(1.0,F(p))-(U(k(k-n))*(Cfs)))-(D(k))/I
end

%delta = comparison value
%Cfs = correction factor

delta = (U(ATRi)*(Cfs)
```

%Wm = fuzzy set membership value

%Um = pattern of decision variable

Wm(k)=1-(Vs(k-fp)-Um)