

1. Report No. FHWA/TXDOT-94/1928-1F		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle  AUTOMATED INCIDENT MANAGEMENT PLAN USING GEOGRAPHIC INFORMATION SYSTEMS TECHNOLOGY FOR TRAFFIC MANAGEMENT CENTERS				5. Report Date July 1993	
				6. Performing Organization Code	
7. Author(s) Robert H. Siegfried and Nitin Vaidya				8. Performing Organization Report No.  Research Report 1928-1F	
9. Performing Organization Name and Address Texas Transportation Institute The Texas A&M University System College Station, Texas 77843-3135				10. Work Unit No. (TRAIS)	
				11. Contract or Grant No. Study No. 7-1928	
12. Sponsoring Agency Name and Address Texas Department of Transportation Research and Technology Transfer Office P. O. Box 5080 Austin, Texas 78763-5080				13. Type of Report and Period Covered Final: August 1992 - July 1993	
				14. Sponsoring Agency Code	
15. Supplementary Notes Research performed in cooperation with the Texas Department of Transportation and the U.S. Department of Transportation, Federal Highway Administration. Research Study Title: Feasibility of Developing an Automated Incident Management Plan (AIM)					
16. Abstract This study addresses the need to incorporate automated incident management in a traffic management center to respond to accidents, disabled vehicles, and other capacity-reducing incidents. The strategy examined to address this need involved developing an automated incident management plan, by customizing personal computer (PC) software and special programs, as a tool to assist operators in the decision making process. The objective of the study was to develop a database management system. To accomplish the objective, Geographical Information System (GIS) technology was customized for incident management. The plan was developed using a GIS development platform approach. PC-ARC/INFO was used to develop prototype/demonstration applications of two types. The first type pertains to incident management operations applications, which addresses Alternate Routing, Incident Response, and Resource Management. The second type pertains to planning and analysis applications, which includes a Roads Database Query, and Incident Database Query. Commercially available programs such as PASSER-II and CAMEO have been integrated into the plan to enhance the capabilities of the automated incident management plan. The development of GIS applications for incident management has identified issues that must be addressed prior to implementing a large scale automated incident management system.					
17. Key Words Geographical Information System (GIS), Automated Incident Management, GIS in Transportation, Intelligent Vehicle Highway System (IVHS), Automated Traffic Management System (ATMS), Route Guidance, Service Patrol Management, Automated Traveler Information System (ATIS)			18. Distribution Statement No Restrictions. This document is available to the public through NTIS: National Technical Information Service 5285 Port Royal Road Springfield, Virginia 22161		
19. Security Classif.(of this report) Unclassified		20. Security Classif.(of this page) Unclassified		21. No. of Pages 378	22. Price



**AUTOMATED INCIDENT MANAGEMENT PLAN USING  
GEOGRAPHIC INFORMATION SYSTEMS TECHNOLOGY  
FOR TRAFFIC MANAGEMENT CENTERS**

by

Robert H. Siegfried  
Engineering Research Associate  
Texas Transportation Institute

and

Nitin Vaidya  
Assistant Research Scientist  
Texas Transportation Institute

Research Report 1928-1F  
Research Study Number 7-1928  
Study Title: Feasibility of Developing an  
Automated Incident Management Plan (AIM)

Sponsored by the  
Texas Department of Transportation  
In Cooperation with  
U.S. Department of Transportation  
Federal Highway Administration

July 1993

TEXAS TRANSPORTATION INSTITUTE  
The Texas A&M University System  
College Station, Texas 77843-3135





## IMPLEMENTATION STATEMENT

The applications developed during the course of this research can be implemented in traffic management centers with some modifications/additions for practical use. Incident management can be improved in several ways.

Upon integration with the Global Positioning System (GPS), the applications can be used to provide vehicle tracking and dispatching for incident response services such as the Motorist Assistance Program (MAP) in the Houston area. The applications can be interfaced with variable message signs and used to provide estimated delay information as well as alternate route information to the motorists through the variable message signs. The applications can be used to improve the response time by quickly identifying the appropriate incident response personnel and resources. Additional improvements in incident management can be brought about by using a combination of the applications for incident location and roads network features analysis.

To implement the research applications, some additional work may be required. Porting of applications to a UNIX workstation platform, integration with GPS, interfacing with variable message signs, enhancement of the data model for delay estimation and collection of data on incident management resources and personnel, as well as detailed up-to-date data on the roads network and incidents may be required for implementation of all features.

The applications, when implemented, will save energy by reducing traffic delays through route guidance. In addition, the applications will not only improve incident response time, but also have the potential to save lives.



## **DISCLAIMER**

The contents of this report reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily represent the official view or policies of the Texas Department of Transportation or the Federal Highway Administration. This report does not constitute a standard, specification or regulation. The software and programs used for this research project were utilized to examine their feasibility; their use by the researchers is not intended to endorse those particular products.

## **ACKNOWLEDGEMENT**

Appreciation is due to the Texas Department of Transportation and the Federal Highway Administration for providing the resources to conduct this research. Appreciation is especially expressed to Steven Z. Levine of the Texas Department of Transportation, District 12 Transportation Operations Department, for his efforts in bringing this project to fruition and for his input as the project technical chairman; and to Chris Olavson of the Texas Department of Transportation, District 12 Advanced Transportation Planning Department, for providing digital maps for this research study. The efforts of TTI staff members who provided valuable research, reviews, and comments, are also appreciated. They are: Dick McCasland, Darrell Borchardt, Dan Hickman, Kelly Parma, Gene Goolsby, Dennis Smalley, Lisa Patke, Emily Sheen, Paul Berry Lively and former TTI staff members Sabas Avila and Tom Nugyen.

## TABLE OF CONTENTS

	<u>Page</u>
<b>LIST OF FIGURES</b>	xii
<b>LIST OF TABLES</b>	xiii
<b>SUMMARY</b>	xv
<b>1. INTRODUCTION</b>	1
1.1. Requirements for an Automated Incident Management Plan	2
1.2. Commercial Software	3
1.3. Geographic Information Systems (GIS) Technology	4
<b>2. THE GIS APPROACH</b>	7
2.1. Automated Incident Management Plan System Architecture	9
<b>3. MAP AND DATABASE DEVELOPMENT FOR INCIDENT MANAGEMENT</b>	13
3.1. Application and Map Analysis	14
3.2. Preparing Maps to Develop ARC/INFO <sup>(C)</sup> Coverages	16
3.3. Creating Application Coverages	16
3.4. Editing Maps	17
3.5. Database Development	19
3.5.1 Design and Development of the Congestion Management Arc Attribute Table (CMAAT) Database	19
3.5.2 Developing Databases for Network Analysis	23
3.6. Summary of Coverage Development	25

	<u>Page</u>
<b>4. APPLICATIONS DEVELOPMENT FOR INCIDENT MANAGEMENT</b>	<b>27</b>
4.1. Applications for Incident Management	28
4.1.1. Dynamic Route Guidance Applications	29
4.1.1.1. Wide Area Alternate Routing	30
4.1.1.2. Local Area Frontage Road Alternate Routing	32
4.1.1.3. Alternate Routing Around an Area of Hazardous Material Spill	33
4.1.1.4. Travel Time Simulation System	35
4.1.1.5. Comparison of Travel Times Using Real and Historical Data	37
4.1.1.6. Other Commercial Software in the Automated Incident Management Plan	45
4.1.2. Incident Response Applications	47
4.1.2.1. Fleet Management for Service Patrols	48
4.1.2.2. Lane Blocking	49
4.1.2.3. Hazard Area Blocking	52
4.1.3. Resource Management Applications	56
4.2. Applications for Planning and Analysis	63
4.2.1. Incident Database Geographic Query System	63
4.2.2. Roads Database Geographic Query System	65
4.3. Benefits of Applications	71
 <b>5. ISSUES AND LIMITATIONS IN GIS DEVELOPMENT USING PC-ARC/INFO</b>	 <b>73</b>
5.1. Slow Data Manipulation	73
5.2. Limitations for Real-Time Congestion Management	73

	<u>Page</u>
5.3. Maintaining Consistent Orientation of Arcs	74
5.4. Lack of Standard Syntax Across Modules	75
5.5. Lack of Dynamic Segmentation	75
5.6. SML <sup>(C)</sup> Limitations	77
5.7. PC versus WorkStation	78
5.8. Deletion of Pseudo-Nodes and Combination of Data	82
5.9. Display of Messages on Screen	82
5.10. Data Availability and Collection	83
<b>6. CONCLUSIONS</b>	<b>85</b>
6.1. Summary of Development Efforts	85
6.2. Recommendations for Developing an Automated Incident Management Plan Using GIS	87
6.3. Related Research, Alternative Approaches, and Future Work	88
<b>REFERENCES</b>	<b>91</b>
<b>APPENDICES</b>	
<b>A. Terminology</b>	<b>A-1</b>
<b>B. INCMAN: INCident MANagement Support Tool — A Traffic Management Center User's Guide</b>	<b>B-1</b>
<b>C. Structure of the Databases</b>	<b>C-1</b>
<b>D. Programs Written to Demonstrate Applications of the Automated Incident Management Plan</b>	<b>D-1</b>
<b>E. A Report on Benchmarking of Workstations for GIS</b>	<b>E-1</b>

## LIST OF FIGURES

Figure 2.1	A Geographic Information System Application	8
Figure 2.2	System Architecture of an Automated Incident Management Plan Using ARC/INFO <sup>(C)</sup> GIS Platform	10
Figure 4.1	Directions Generated for a Wide Area Route	31
Figure 4.2	Frontage Roads and Ramp Interchange for Local Area Alternate Routing	34
Figure 4.3	Example of Routing Around a Hazard Area	36
Figure 4.4	A Result of a Travel Time Simulation	38
Figure 4.5	Van Locations in Fleet Management Application	50
Figure 4.6	List of Map Van Locations	51
Figure 4.7	Menu for Lane / Area Blocking	53
Figure 4.8	An Example of Hazard Areas Needing Evacuation	55
Figure 4.9	Graphical Query System on Hospitals	58
Figure 4.10	Graphical Query on HOV and Park and Rides	59
Figure 4.11	Graphical Query on Flood Zones and Pumps	60
Figure 4.12	Fire Station Operation Zones	61
Figure 4.13	Address of Fire Station for Street Serviced	62
Figure 4.14	Incident Database Geographic Query System	66
Figure 4.15	Results of Monthly Query on Incident System	67
Figure 4.16	Locational Query on Incident System	68
Figure 4.17	Roads Database Geographic Query System	70



## LIST OF TABLES

	<u>Page</u>
Table 3.1 Representation of a Unique Identifier (GISCODE) for Database Links	21
Table 4.1 Travel Time Simulation vs. Real-Time Data for Case I	41
Table 4.2 Travel Time Simulation vs. Real-Time Data for Case II	42
Table 4.3 Travel Time Simulation vs. Real-Time Data for Case III	43



## SUMMARY

This study addresses the need to incorporate automated incident management in a traffic management center so that the operators can quickly assess the probable traffic impact of incidents and take appropriate response actions to reduce such impact.

The strategy to address this need involved developing an automated incident management plan by customizing personal computer (PC) software. Geographic Information System (GIS) technology, which features integration of database with a graphic map to create an intelligent map, was customized for incident management. Using the most comprehensive PC based GIS system available at the time of this study, PC-ARC/INFO<sup>(C)</sup> [1], the processes to develop a database, maps, incident management applications, program integration, and user-friendly interfaces were effectively accomplished.

Prototype/demonstration incident management applications have been developed which have identified issues that must be addressed prior to implementing a large scale automated incident management system. The applications developed for the automated incident management plan are grouped into two types. The first type pertains to incident management operations which includes Alternate Routing, Incident Response, and Resource Management. The second type pertains to planning and analysis for incident management which includes Roads Database Query, and Incident Database Query. In addition, the capabilities of an automated incident management plan can be enhanced by integrating the applications with commercially available programs such as PASSER-II<sup>(C)</sup>[2] and CAMEO<sup>(C)</sup>[3].

GIS technology can be used as an effective platform to feasibly develop an automated incident management plan. It is recommended that a GIS approach to incident management applications development should be based on a multi-tasking workstation environment using a GIS development tool with support for (a) a relational database management system, (b) functionality for incident management, (c) a programming language that parallels a higher level programming language, (d) capability for map editing that is similar to a CAD system, (e) a variety of data format for import/export of maps and databases, and (f) truly integrated modularity and uniformity of design.



# **AUTOMATED INCIDENT MANAGEMENT PLAN USING GEOGRAPHIC INFORMATION SYSTEMS TECHNOLOGY FOR TRAFFIC MANAGEMENT CENTERS**

## **I. INTRODUCTION**

One of the objectives of the Intelligent Vehicle Highway Systems (IVHS) program is the development of urban area Advanced Traffic Management Systems. A key ingredient of traffic management operations is the ability to identify and react to freeway corridor incidents. Timely and effective detection and response can result in significant travel savings and enhanced safety.

The Houston District of the Texas Department of Transportation is developing a freeway incident response and management program to reduce the time required to clear an incident and to reduce the impacts on motorists. One of the techniques being considered is an automated incident management plan. When major incidents occur, possibly involving hazardous materials, large trucks, or other unusual situation, the operator of a traffic management system must make decisions on what controls to employ, what to display on variable message signs, who to notify, and how to reduce the effect of the incident on traffic operations. The automated incident management plan would enable the operator to assess the impact of the incident and to develop alternative routes.

This study was conducted to address freeway incident management to be performed in a traffic management center for the Houston metropolitan area, and specifically, to assess the use of Geographic Information Systems (GIS) as the locational basis and database for the automated incident management plan. The results and conclusions of this study are intended to assist traffic management centers in Houston and other urban areas with transportation system management.

It was found that it was possible to use a GIS to relate incident locations with the transportation network and to make decisions and calculations for incident management. This system could provide computerized mapping and database management. By using a

GIS development platform, it is possible to develop interrelated maps, databases, and incident management applications. This platform also provides the ability to integrate other software for data sharing and execution through a common user-friendly interface.

This report documents the development of an automated incident management plan in order to assess the feasibility of developing and implementing the results of this study as a strategy to reduce and manage congestion.

### **1.1. REQUIREMENTS FOR AN AUTOMATED INCIDENT MANAGEMENT PLAN**

Implementation of an automated incident management plan in a traffic management center would require a computer and computer interface devices such as: graphical display, keyboard, mouse, printer, and plotter. The computer would be loaded with the GIS software and the applications that make up the automated incident management plan. Once the GIS platform is operational, the operator at a traffic management center would be able to operate the system through a user-friendly interface to assist with incident management. The applications developed for this study can provide dynamic route guidance, incident response, resource management, and roads and incident database spatial queries.

The operator and a traffic management center will execute applications for real-time traffic management to integrate and support with the system's traffic surveillance, communication, and control capabilities. When incidents occur, the operator will be able to develop traffic control and routing plans and provide information to variable message signs to indicate: incident locations, expected incident delay times, and suggested alternative routes. The operator will also have access to fleet management for service patrols by providing: directions to incident location, estimated time of arrival, contact of clean up crews assigned to incident location, closest backup assistance, resource center assistance, and information to nearest exit and turn ramps. Finally, the traffic management center manager will be able to evaluate and plan for incident management operations by determining: incident problem site identifications and assessments, optimal service patrol routes based

on incident frequencies, and contingency planning for roadway construction, maintenance, and repair sites.

## 1.2. COMMERCIAL SOFTWARE

There are two approaches to consider in developing computer systems for automated incident management. In the first approach, incident management software may be developed "from scratch" using traditional high level programming languages after an in-depth incident response systems analysis. In the second approach, commercial software may be utilized to develop incident management applications in an interpretive environment. Generally, the software purchase cost for the first approach is low, while the services of a programmer and development time are high. Conversely, the software purchase costs for the second approach are high, while the services of a programmer and development time are low. Development time is affected by: application design requirements, functionality needed, coding or programming, and testing and debugging.

A traditional programming language, such as C<sup>(C)</sup> or FORTRAN<sup>(C)</sup>, provides the power and flexibility to develop large and sophisticated software. Such languages provide tools for a software engineer or programmer to develop applications tailored to meet the needs of the user. However, development of applications in traditional programming languages is time-consuming because these languages only have generic "low-level" functions which are not specific to an application.

On the other hand, commercially available software generally includes various customizable analysis programs specific to the applications to be developed. Use of such programs reduces the development time and effort. Unlike traditional programming languages, the user need not develop the features inherent in the analysis programs. Commercial software that is specific to certain kinds of applications is generally more user-friendly and applications may be developed without using the services of a conventional programmer.

Commercially available software may also be of two types. The first includes specialized programs which perform specific tasks rather than providing advanced functionality to develop applications which may perform these tasks. Such software may or may not be customizable. Examples of such software are PASSER-II<sup>(C)</sup>, HAZTRANS<sup>(C)</sup>, and CAMEO<sup>(C)</sup>. The second type includes software which provide features and functionality to develop applications to perform specific tasks. These software generally provide their own programming/macro language for development of applications. Examples of such software are ARC/INFO<sup>(C)</sup> which is a GIS development tool, and dBASE<sup>(C)</sup> which is a database management systems development tool.

### **1.3. GEOGRAPHIC INFORMATION SYSTEMS (GIS) TECHNOLOGY**

GIS is an advanced technology that was developed with the ability to perform spacial analysis with tabular, vector, and image data. The power of this analysis lies in the data model provided by the program. The data model consists of topological data structures which are together capable of recording data information associated with points, lines, and polygons. This data model allows the development of applications which facilitate a variety of graphical analysis using a geographic database. GIS is described by the Environmental Systems Research Institute as: "An organized collection of computer hardware, software, geographic data, and personnel designed to efficiently capture, store, update, manipulate, analyze, and display all forms of geographically referenced information." In simpler terms, it is: "A computer system capable of holding and using data describing places on the earth's surface" [4]. A GIS application in ARC/INFO<sup>(C)</sup> consists of coverages and a set of programs to manipulate those coverages. A *coverage*, in turn, consists of a map layer associated with a set of databases.

Transportation agencies have implemented, or are considering the use of, GIS technology to provide computer-based mapping with database management to relate and describe intersections, roadways, and resource areas.



PC-ARC/INFO<sup>(C)</sup>, version 3.4d, a commercially available geographic information systems development tool, was used in this study as a platform to develop the automated incident management plan. PC-ARC/INFO<sup>(C)</sup> was chosen because: (a) it had extensive support for transportation applications, (b) it had a programming language to build custom GIS applications, (c) it was available on a variety of computer and operating systems (only GIS available on PC at the time of this study), and (d) it was the most widely used commercial GIS development tool. In general, ARC/INFO<sup>(C)</sup> was considered to be a general purpose GIS development system that could develop applications for transportation.

PC-ARC/INFO<sup>(C)</sup> has a modular structure, where different operations are performed in each module. The PC-ARC/INFO<sup>(C)</sup> modules include:

- ARCEDIT for creating editing and querying map and related databases;
- ARCPLOT for displaying, printing and plotting and querying of coverages;
- NETWORK which includes ROUTE and ALLOCATE for dealing with special GIS issues of networks such as roads or pipelines;
- OVERLAY for complex GIS queries involving 2 or more coverages;
- DATA-CONVERSION for converting GIS/database from other formats to ARC/INFO<sup>(C)</sup> and vice-versa; and
- TABLES for manipulation of databases in dBASE-IV<sup>(C)</sup> file formats.

Macro Languages are provided with application development platforms so that application designers can develop and implement applications and interfaces. These macro languages generally support much of the same functionality that high level programming languages provide.

PC-ARC/INFO<sup>(C)</sup> uses the Simple Macro Language (SML<sup>(C)</sup>) to create GIS applications. While it does not have the capability to develop some complex application, it was effectively used to examine the capabilities of GIS in transportation for the automated incident management plan.



## 2. THE GIS APPROACH

Development of an automated incident management plan has been investigated by developing demonstration/prototype applications using commercially available software. The study utilizes PC-ARC/INFO<sup>(C)</sup> as the development platform for the automated incident management plan. This development platform can:

- perform relational database management;
- provide a color coded graphical display of the transportation network;
- develop customized incident management applications for user-friendly operator interaction.

Research activities included: 1) development of maps, 2) development of map-related databases, and 3) development of prototype/demonstration incident management applications. The applications can be executed on a PC-486 with the aid of user-friendly interfaces. Figure 2.1 shows the elements that make up a GIS application.

Applications were developed using a Simple Macro Language (SML<sup>(C)</sup>) provided for customized application development. SML<sup>(C)</sup> is an interpreted language. This means that coding is limited by the underlying interpretive architecture, i.e., PC-ARC/INFO<sup>(C)</sup>. For application modifications there were limitations due to lack of programming capabilities such as: modularity, encapsulation, and code reusability. Nevertheless, SML<sup>(C)</sup> provided a means to develop incident management applications to examine GIS program requirements.

PC-ARC/INFO<sup>(C)</sup> provided application capability for: origin/destination routing, vehicle lane blocks, data item calculations (e.g., Distance/Rate ratio), data item comparisons, data item updates, address-matching capability, object oriented data access (e.g., click on roadway to get capacity), restricted routing (e.g., one-way ramps), and others.

Two types of applications have been defined for the automated incident management plan. The first type pertains to incident management operations in a traffic management center, whereas, the second pertains to planning and analysis for incident management. In addition, these applications were designed to integrate with PASSER-II<sup>(C)</sup> for intersection

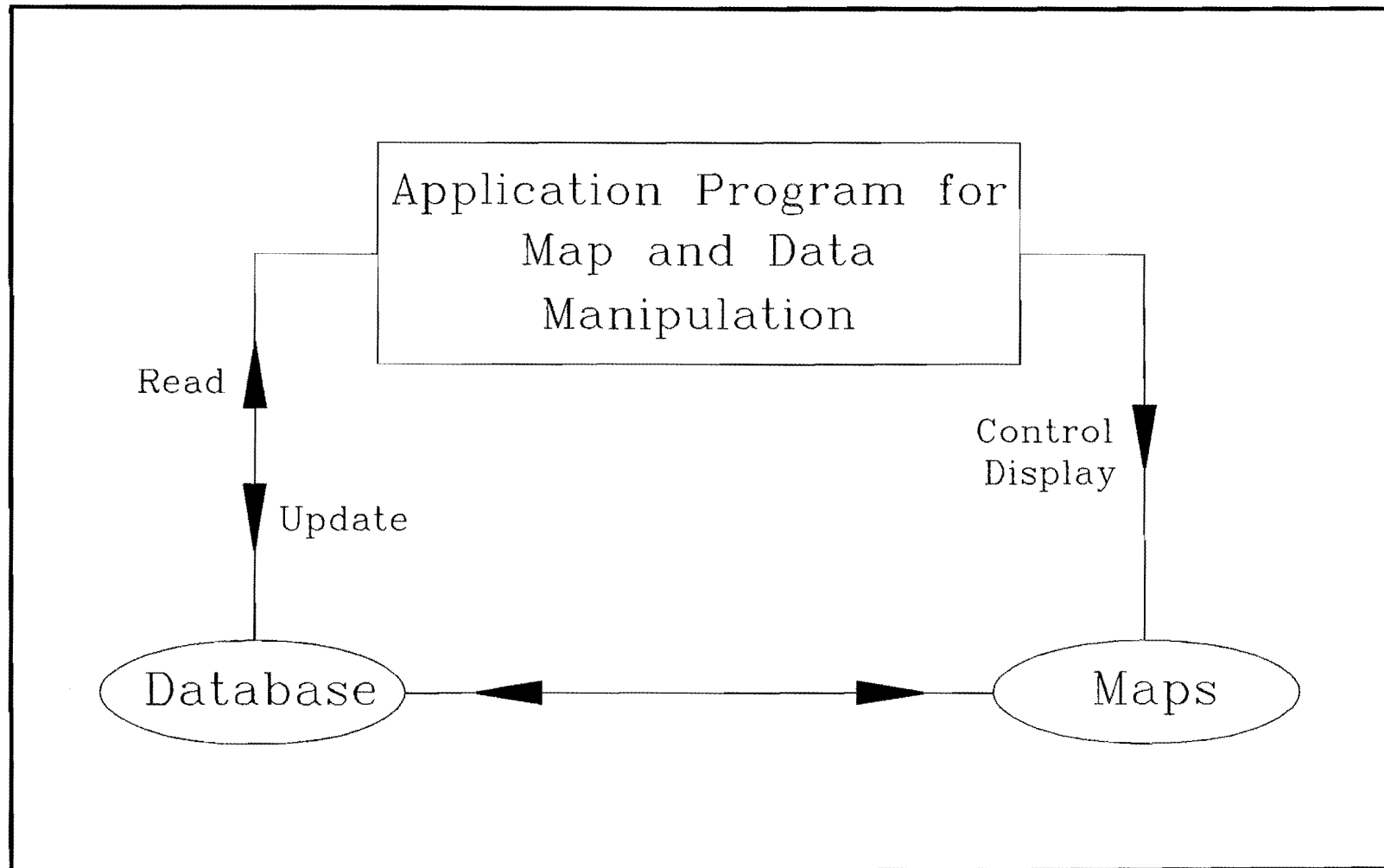


Figure 2.1 A Geographic Information System Application

operational data updates, and CAMEO<sup>(C)</sup> for independent execution to perform area wide chemical emergency management.

The applications for incident management operations include:

- Dynamic Route Guidance
- Incident Response
- Resource Management

The applications for planning and analysis for incident management include:

- Roads Database Query
- Incident Database Query

## **2.1. AUTOMATED INCIDENT MANAGEMENT PLAN SYSTEM ARCHITECTURE**

The system architecture of the automated incident management plan envisages the use of ARC/INFO<sup>(C)</sup> GIS development package as the development platform. On this platform, several prototype/demonstration GIS applications have been developed using various modules of ARC/INFO<sup>(C)</sup>. Figure 2.2 shows the system architecture of an automated incident management plan using ARC/INFO<sup>(C)</sup> GIS platform.

Two types of applications have been proposed for the automated incident management plan based on a set of coverages for the Houston metropolitan area. The first type deals with incident management operations in a traffic management center, whereas, the second type deals with planning and analysis for incident management. In addition, it is proposed that these applications integrate (i.e., to share data) with other commercially available software such as PASSER-II<sup>(C)</sup> which is representative of intersection analysis software, and CAMEO<sup>(C)</sup> which is representative of software for management of chemical emergencies.

Applications for incident management operations include the following:

- *Dynamic Route Guidance*: Depending on the need, alternative routing could be performed over: (a) a local area using frontage roads for incident affected freeways, and/or (b) a wide area using corridor arterials, and/or (c) hazard areas.

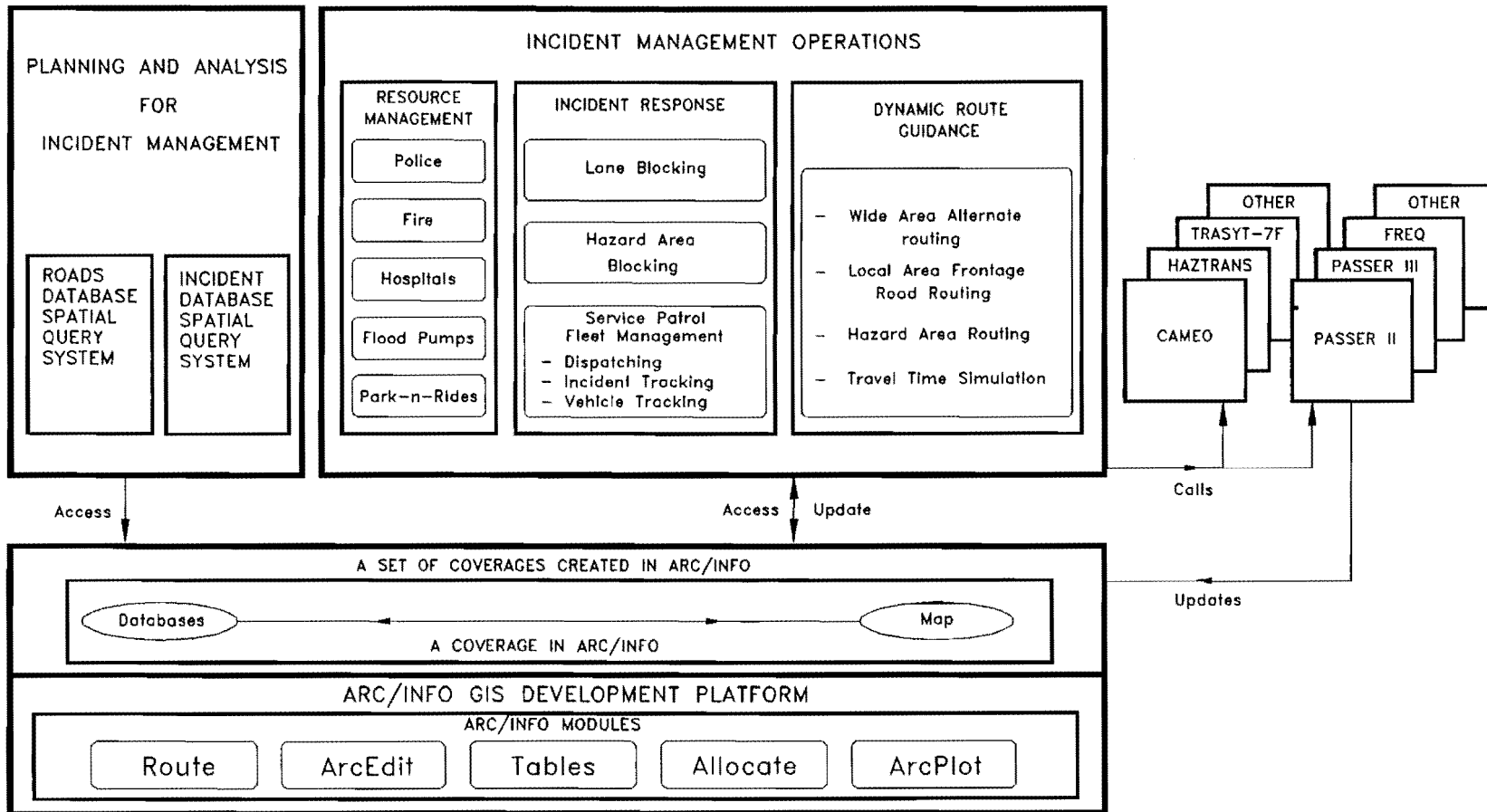


Figure 2.2 System Architecture of an Automated Incident Management Plan Using ARC/INFO<sup>(c)</sup> GIS Platform

In addition, a travel time simulation could be used to estimate freeway delays as well as provide dynamic route guidance to motorists through variable message signs, highway advisory radio, and/or other motorist information media. For this purpose, travel time simulation could be performed using superposition of real incidents on historical data.

- *Incident Response:* Incident response plans could be provided for: (a) blocking of individual freeway lanes for the purpose of alternate routing, or (b) automated blocking of a hazard area, such as an area affected by a chemical spill, by generating a list of streets to be blocked and providing alternate routing around the area, or (c) automated tracking of incidents and dispatching of one or more of available vehicles from an incident response fleet, such as vans of the Houston Motorist Assistance Program in Harris County, Texas [5].
- *Resource Management:* Some of the resources for incident management operations are police and fire stations, hospitals, and critical flood pump locations. These resources could be managed by: (a) providing spatial queries to obtain names, addresses and phone numbers of organizations/individuals to contact when required; and (b) providing spatial queries on the features, capacities and availabilities of resources.

Applications for planning and analysis for incident management are provided for engineers and planners who could analyze roads and incidents, separately or in conjunction, to devise plans for improving traffic flow.

Such applications include:

- *Roads database query:* The roads database query system will permit spatial queries on the roadway inventory database as well as graphical results of queries on the database. Such queries could be of the kind "Show graphically all sections of North Freeway with dual shoulders" or "How many lanes are available in the outbound direction of an interactively selected section of a freeway." Queries may be performed on a variety of attributes in the roadway inventory.

- *Incident database query:* The incident database query system will provide the ability to perform spatial queries on the number and types of incidents on any section of freeway. Conversely, response to some queries may be shown graphically. For example, "List all incidents in the month of June 1992 on a graphically selected section of a freeway" will list the incidents date, time, and type of all such incidents. On the other hand, "Graphically show all sections of Southwest Freeway with more than 20 incidents in the year 1991" will provide a graphical response.



### 3. MAP AND DATABASE DEVELOPMENT FOR INCIDENT MANAGEMENT

It is significant to realize that a GIS is in fact a database management system that can draw a map and relate feature and attribute information [6]. A map and related database together are called a *coverage* in ARC/INFO<sup>(C)</sup> terminology. There are five basic elements in the ARC/INFO<sup>(C)</sup> data model: 1) *cartographic database*, 2) *georelational data model*, 3) *coverage concept*, 4) *feature class concept*, and 5) *topological data structure*.

- 1) The *cartographic database* organizes data using a related topological approach. The topology enables the use of two types of spacial data: locational data to describe points, lines, and areas; and attribute data, the information describing these features.
- 2) The *georelational data model* allows the separation of geographic information into layers or coverages to represent features (e.g., roads, park and rides, incidents, construction zones). The flexibility of combining these features is made possible through this georelational layering capability.
- 3) The *coverage concept* is a way to organize a single map layer that usually describes one type of map feature, such as freeway network, resource centers (e.g., police stations, fire stations), or incident locations. A coverage is stored as a set of feature class files which describe the area (boundary, arcs, nodes, polygons, and attributes).
- 4) The *feature class concept* is the basic unit in the cartographic database (1 above). The basic feature classes include: Arc, line segment having length but no area; Node, endpoints of an arc, nodes link all arcs to provide arc/node topology; Polygon, an area defined by a series of arcs; Points, used to represent point features, e.g., bus stops, traffic signs, bridges, mile posts; and Route, linear feature made up of arcs or part of arcs.
- 5) The *topological data structure* provides the relationship between connecting coverage features. For example, the topology of an arc includes the "from

node" and "to node" and its right and left polygons. Topology provides the basis for geographic analysis by providing information about the location of features relative to other features. This topological data structure provides the power and ingenuity needed for transportation applications.

The processes involved in developing a coverage for the automated incident management plan included: 1) application and map analysis, 2) preparing maps, 3) creating the coverage, 4) editing the map coverage, and 5) developing the database.

### **3.1. APPLICATION AND MAP ANALYSIS**

The applications required for incident management were analyzed to determine maps that would be required. The NETWORK module used to perform routing and allocating applications dictated the feature limitations for the incident management applications. The Network module limited roadway centerline maps to a maximum of 4,300 arcs. Working with the application requirements and limitations, the general map requirements were determined for the automated incident management plan.

The major feature requirements that guided the development of maps were:

- 1) available mapping of Harris County;
- 2) center-line map;
- 3) maps that were up to date;
- 4) freeway, principal, arterial, frontage road, freeway ramp, interchange ramp features; and
- 5) maximum 4,300 arcs.

Several sources were considered for development of maps, as given below:

- 1) TRAF maps from Texas Department of Transportation (TxDOT)
- 2) QUAD maps from TxDOT
- 3) METROCOM maps from City of Houston
- 4) TTI in-house digitized maps
- 5) TIGER<sup>(C)</sup> database files from the US Census Bureau
- 6) ETAK<sup>(C)</sup> maps from Etak Inc.

TxDOT TRAF maps are very detailed highway and street maps that include representation of individual freeway lanes, frontage roads and freeway interchanges. TxDOT QUAD maps are less detailed, but provide center line roadways. METROCOM maps are detailed maps with representations of roadway medians, shoulders, and area (polygon) features. TTI in-house CADD maps included various layered AUTOCAD<sup>(C)</sup> and MICROSTATION<sup>(C)</sup> maps with varying levels of detail, accuracy, and coverage of Harris county. The Census Bureau's TIGER<sup>(C)</sup> files are detailed maps with USGS census data that can be used to directly generate ARC/INFO<sup>(C)</sup> coverages. ETAK<sup>(C)</sup> maps are based on TIGER<sup>(C)</sup> files.

A digitized AUTOCAD<sup>(C)</sup> map that was developed by TTI, using a Rand-McNally<sup>(C)</sup> road map of Harris County, was examined. However, a number of digitizing errors were found on the map, and the Rand-McNally<sup>(C)</sup> map did not cover Harris County in its entirety. The map had not been updated to include the latest developments in Harris County, most notably, Sam Houston Tollway. Updating this map would prove to be too costly and time consuming, given the scope of this project.

A review of the TxDOT TRAF maps and the TIGER<sup>(C)</sup> map indicated that each had too many features for practical network analysis applications in ARC/INFO. The TRAF maps included detailed representations of freeway interchanges and freeway lanes. The TIGER<sup>(C)</sup> maps were examined by extracting data for a limited area due to their extensive size; but this was not sufficient, because of the requirement that applications cover a larger area. Additionally, both of these maps had a few errors and were incomplete. Similar findings also eliminated the use of ETAK<sup>(C)</sup> files for application development.

The METROCOM maps had too many polygon features and no centerline features. Also, the files were not available in a suitable format for conversion at the time of this study. These maps may prove to be useable in the future as they are being converted to ARC/INFO<sup>(C)</sup> format with center line features. These maps also have attribute data associated with features.

After identifying the sources and characteristics of the maps that were available to create a coverage, it was determined that the best map for prototype/demonstration development were the TxDOT QUAD maps. These maps provided the best solution for

application coverage development because they provided center-line data and were available in quadrants for selecting the limited features needed.

### **3.2. PREPARING MAPS TO DEVELOP ARC/INFO<sup>(C)</sup> COVERAGES**

The TxDOT QUAD maps were used as the basis for most application coverages. The TxDOT QUAD maps were edited to meet the requirements of the applications and ARC/INFO<sup>(C)</sup>.

There were forty-four TXDOT QUAD maps obtained from TxDOT in AUTOCAD<sup>(C)</sup> ASCII Drawing Interchange format (DXF) which were exported from INTERGRAPH's IGDS format files. These maps were then combined to generate one single map in AUTOCAD<sup>(C)</sup>. Features on the maps such as streams, railroads and annotations of street names, were excluded from this file since they were not necessary for the applications requirements. The combined map that was generated from the TxDOT QUAD maps is referred to as the "Congestion Management" (CM) map.

### **3.3. CREATING APPLICATION COVERAGES**

The DATA CONVERSION module provided with PC-ARC/INFO<sup>(C)</sup> was used to import existing vector file formats. There are eight vector file formats supported: ATLAS, DIME, DLG, DXF, ETAK, IGES, MOSS, and TIGER.

The CM map that was created in AUTOCAD<sup>(C)</sup> from the TXDOT QUAD maps was exported to DXF format for conversion into an ARC/INFO<sup>(C)</sup> coverage. The three steps taken to create an ARC/INFO<sup>(C)</sup> coverage were as follows:

- 1) The DXFINFO command was used to review the structure and contents of the CM map which was in DXF format. This review provided the information required to select the layers to use when converting to ARC/INFO<sup>(C)</sup>.
- 2) The DXFARC command was used to interactively specify layer names and options to create the coverage.



I-610 (W.Loop 610N, W.Loop 610S, N.Loop 610W, N.Loop 610E, E.Loop 610N, E.Loop 610S, S.Loop 610W, S.Loop 610E).

Principals: Navigation, Alameda-Genoa, U.S. 90, Bellaire, Allen Parkway/Kirby, Bingle/Voss/Hillcroft, Clinton, CE King/Federal/Shaver, FM 1960, John F Kennedy, Little York, Red Bluff, SH 35, Shepherd, SH 6, Cullen, Wilcrest, Brittmore, Westheimer, Bissonnet, Cavalcade, Eldridge, Alameda, Memorial, West Montgomery, Hardy Toll, Tidwell, Homestead, Wallisville.

### *Color Coding of Maps*

Maps have been color coded to reflect the classification of roadway. The classifications are based on the Houston-Galveston Regional Transportation Study [7]. Generally, each class also corresponds to a default maximum permissible speed limit, though this is not universally true and could vary from section to section in each class. The network colors include:

- Interstates and other freeways - green;
- ramps, freeway interchanges, and frontage roads - yellow;
- principals - blue;
- major arterials - gray; and
- minor arterials - red.

### *Annotation*

Annotation was provided to label the main roadways and resource centers. Annotation was kept to a minimum due to screen cluttering on a twelve inch display.

### *Orientation of arcs*

To model origin-destination routing, it was important that each arc, representing a street segment, be assigned an appropriate orientation. This is necessary since ARC/INFO<sup>(C)</sup> requires bi-directional impedance values for each arc, only one of which is used to find the route with least impedance. For example, if an outbound route from

downtown to the suburbs is required, all the arcs on the routes should have the same orientation, i.e., either all arcs be oriented towards downtown or away from downtown. Special efforts were made to orient the arcs consistently (arcs are mostly oriented outbound); however, on roads which loop around downtown or those which do not lead to downtown, it was difficult to maintain consistency and there are cases where two adjacent arcs on the same roadway have opposite orientations.

### **3.5. DATABASE DEVELOPMENT**

Once the CM map coverage was created, it was necessary to enter the arc feature data into the database. Also, additional databases and data items had to be created for network analysis applications.

The databases that were developed for the CM coverage include:

- 1) Congestion Management Arc Attribute Table (CMAAT) database, for arc data information;
- 2) Address database, for address-matching analysis;
- 3) Turntable database, for intersection analysis;
- 4) Incident database, for incident location analysis; and,
- 5) Point Attribute Table (PAT) databases, for police, hospitals, and fire stations information.

#### **3.5.1 Design and Development of the Congestion Management Arc Attribute Table (CMAAT) Database**

##### *Design of the CMAAT*

The initial approach to structuring the CMAAT was to represent each road segment using three separate databases. That is, each arc would be represented with a record in each of the three databases. This approach was taken because of the anticipated size of the database which would slow data access. It was believed that the use of three smaller databases would possibly result in faster data access. Using such a structure, only the

required data would be loaded into memory using a swapping mechanism. However, it was a relatively complex data structure that would require frequent JOIN operations on the three database design. ARC/INFO<sup>(C)</sup> was not efficient in handling such operations. As such, the CMAAT data structure was redesigned to include all data for an arc in one record. Each record is about 1200 bytes and includes over 150 attributes for each arc.

Once the structure of CMAAT database was set, the next step was to add data items for every arc in the CMAAT. The process of adding data items was done, in most cases, automatically and/or semi-automatically. In other cases, data was manually entered with an interactive process provided by the relational database management system. The CMAAT information was assembled from the following available databases:

- 1) TxDOT Roadway Inventory Database (TRID)
- 2) TTI Travel Time Run Database (TTRD)
- 3) TTI 24-Hour Volume Database (24HVD)
- 4) TIGER<sup>(C)</sup> Database

#### *Development of the CMAAT*

The TRID is a comprehensive roadway inventory database acquired from the TxDOT. The database has over 100 items that describe such features/events as roadway geometries, signage, travel patterns, and so forth. The database was obtained in a format that could be accessed directly by ARC/INFO<sup>(C)</sup>. However, importing this data into CMAAT database was a major issue. The TRID street segments did not always correspond to the street segments in the CMAAT database. In addition, arcs in the TRID database did not exist on the CMAAT and vice versa.

One approach to add items to the CMAAT was to manually match the CMAAT and TRID arcs and then to enter the TRID data into the CMAAT database. However, a better approach was taken which performed a "relational database JOIN using multiple items" on the two databases and then performed an update to the CMAAT database. This was done by matching three data items that existed in both databases: STREET name, FROM STREET name, and the TO STREET name. As these three items also existed on the TRID, the roadway inventory could be incorporated semi-automatically into the CMAAT



database. However, as there was a potential for error in this approach, (for example because of a difference in spelling or because of an error in any one of the two databases), care needed to be taken in naming the streets in the CMAAT before performing a match. Random checking of arcs indicated this approach to be successful.

Once the TRID database had been successfully added to the CMAAT, a unique identifier was created from the predominant data items of all databases considered for input to the CMAAT. This unique identifier, called the GISCODE, would allow easier data transfers from the TTRD and the 24HVD database [8]. The unique GISCODE approach would also allow for eventual updates to the CMAAT database.

The GISCODE is a nine digit number, as shown in Table 3.1. The first digit was created from the general direction of travel of the arc. The second digit represents the classification of the roadway; this digit was found in the TRID and TTRD databases. The next five digits are the street code found in the TRID and the TTRD databases. The last two digits represent the sequence in which arcs are arranged; this was found in the TTRD database.

Having created the GISCODE, the next step in entering data into the CMAAT was to match and/or enter the GISCODE on the TTRD and the 24HVD databases. Having done so, the data from these databases was added to the CMAAT database. The last data items to be entered into the CMAAT were the latitude/longitude values from the TIGER database.

GISCODE	ITEM CLASSIFICATION	DIGIT
1 2 34567 89 	direction roadway classification street code arc number in sequence	1 2 3-7 8,9

Table 3.1. Representation of a Unique Identifier (GISCODE) for Database Links

The TIGER<sup>(C)</sup> files from the Census Bureau were examined for both map and data. ARC/INFO<sup>(C)</sup> has utilities to directly convert TIGER<sup>(C)</sup> files to an ARC/INFO<sup>(C)</sup> GIS. Upon conversion to ARC/INFO<sup>(C)</sup>, it was found that about 135,000 arcs were being used to represent Harris County. This large number of arcs significantly exceeded the limit on the number of arcs allowed in PC-ARC/INFO<sup>(C)</sup>. Thus, the TIGER<sup>(C)</sup> file for Harris County could not be used directly for generating a useful PC-ARC/INFO<sup>(C)</sup> coverage. But, the database in TIGER<sup>(C)</sup> files was found to be useful in developing separate maps generated using C programs and dBASE-IV<sup>(C)</sup>. Three approaches were attempted to incorporate latitude and longitude values from TIGER<sup>(C)</sup> files in the CMAAT.

The first approach was similar to the approach used for importing roadway inventory. "Relational database JOIN on multiple items" was attempted using street names and names of intersecting streets. This approach was not suitable, however, because of aliases, differences in spellings, an incomplete pre-census TIGER<sup>(C)</sup> file database, and differing street segments between GIS database and the TIGER<sup>(C)</sup> files database.

The second approach involved writing a program to translate screen coordinates to latitude and longitude values. For this purpose, the extents of the GIS map were related to the extents in the TIGER<sup>(C)</sup> files. After the extents were related, a mouse was used to generate the latitude and longitude for any intersection selected. These latitudes and longitudes were then compared with the data values in TIGER<sup>(C)</sup> files where near perfect matches in the values were obtained.

As this approach seemed suitable, the processes were further automated in a third approach by extending the program to translate screen coordinates. The algorithm involved obtaining a list of intersections from the CMAAT database, and automatically seeking the screen coordinates from ARC/INFO<sup>(C)</sup> for each intersection (this is possible in ARC/INFO<sup>(C)</sup> by translating these screen co-ordinates and updating the CMAAT database to include the latitude and longitude values). In this way, latitude and longitude values were semi-automatically incorporated into the CMAAT database.

### 3.5.2 Developing Databases for Network Analysis

Once the CMAAT was developed, the remaining databases were created to perform the necessary network analysis for applications development. These databases are related to the CMAAT database through common locational and feature attributes. The databases include the address database, turntable database, incident database, and point attribute table databases.

#### *Building an Address Database*

The address database was created with the ACREATE command. Address data items from the CMAAT are grouped into a separate database to facilitate the handling of multiple addresses for various features. In addition, several features can be represented by more than one address, e.g., Interstate 45 and I-45.

The addresses for arcs are represented with the street name and address number ranges. There is an address associated with the from-node and an address associated with the to-node.

In order to perform address-matching with this database, ARC/INFO<sup>(C)</sup> assigns some reserved or key words, such as NORTH, EAST, SOUTH and WEST to represent direction, and ROAD, STREET, DRIVE, FRWY, AVENUE, BLVD, and LOOP for the type of street. ARC/INFO<sup>(C)</sup> reads the street names from the database and looks for the keyword in the name to be able to generate a soundex to perform a match. However, problems occur when streets are named as "NORTH LOOP WEST." Since each word in such a street name is a keyword, there is no name for ARC/INFO<sup>(C)</sup> to perform a match on. As a result, ARC/INFO<sup>(C)</sup> indicates an inability to match. This issue needs to be considered when naming the streets for ARC/INFO<sup>(C)</sup>. For example, "NORTH LOOP WEST" was renamed to "NORTH LOOP 610W."

#### *Building a Turntable (TRN) database*

The TRN database identifies every possible turn from one arc onto another arc for the entire CM coverage, and calculates the angle and azimuth of each turn. This turn

information is used during generation of least-impedance route. The TRN database was generated in the NETWORK module. For each intersection several records are written into this database, one record corresponding to each turn. The TRN database generates items for the node number, the identifications for the two intersecting arcs, the azimuth or the orientation of the arcs, and the angle of turn.

The TRN database, however, initially does not include any impedance attributes or data. In addition to the items generated, it was required to add the impedance items and provide the values for these items. For a 1000 arc road network, there would be between 4000 to 8000 turn records, so an automated mechanism was required for generating the generic turn impedances. For this purpose, additional items have been added to the turntable database. A program was written to generate a generic turn impedance for each intersection depending on the classes of the intersecting roads and on the angle of turn.

The turn impedance with negative values prohibit turns; a zero value would provide no impedance; and greater than zero would indicate the time in minutes to make the turn. For example, the turn impedance from freeway to frontage road is infinite, as the only way a vehicle can go from a freeway to a frontage road is via a ramp. On the other hand, freeway to freeway turn is near zero as long as it is not a U-turn. For some specific intersections, turn impedance values were manually entered. This TRN database has enabled the integration of ARC/INFO<sup>(C)</sup> travel time application with the output of PASSER<sup>(C)</sup>, as well as enabled frontage road, ramp, and freeway interchange routing.

#### *Building an Incident Database*

The incident database is an external database that was created from the Houston Motorist Assistance Program quarterly reports [9]. This report is prepared for incident data provided from service patrols. This database includes a number of items that are used for incident analysis. The data items include: detection time, response time, clearance times, incident location, type of incident, type of assistance, and other information.

The incident database was created to provide daily and cumulative incident data for display and analysis on the CM map coverage. A number of other external databases that

include accident information, vehicle mile trends, and construction zones can be used in a similar manner to provide a visual display for analysis.

#### *Point and Polygon Resource Databases*

The resource databases were created for point and polygon features. Point features that were included on the CM map coverage include: fire stations, police stations, hospitals, variable message signs, automatic vehicle identification readers, flood pump stations, park and ride lots, and Texas reference markers. Polygon features include: zip code areas, phone number areas, and flood zone areas.

Both locational and attribute data is stored for each resource. Typical item information included are: NAME, ADDRESS, ZIPCODE, PHONE NUMBER, NUMBER OF BEDS, NUMBER OF AMBULANCES, STATION NUMBER, CONTACT NAME, and so forth. The purpose for these databases is to allow the operator to select these features, points, or polygons interactively to provide valuable on-line information to assist with incident management.

### **3.6. SUMMARY OF COVERAGE DEVELOPMENT**

A complete description of the database is provided in APPENDIX C. A brief summary of the CMAAT and network analysis databases are discussed in this section.

Each arc, which represents a street segment, is stored as a record. Each record includes location identification internal fields for relational management. The CMAAT has items for length of arc in miles, the location of the arc, the class of road to which that arc belongs, and the name of the arc. Each arc represents traffic flow in both directions, except for roads which are "one-way" (e.g., ramps). To achieve bi-directional representation, data for both directions is stored in the same record. This includes the maximum permissible speed in each direction, the number of lanes in each direction, the capacity per lane in each direction, the direction of traffic, the status of congestion in each direction, the number of shoulders in each direction, and an impedance field which is an indicator of expected time to travel in that direction on that arc. There are a number of other items which are used

to compute this impedance item. These include forty-eight items which contain 24-hour volume data, hour by hour, for each direction of the arc. This volume data is later used to compute the estimated time to travel on the arc at a given time of day.

The database also includes the roadway inventory acquired from TxDOT. This roadway inventory could be used for GIS query on the roads database and could be helpful both in planning for incident management and for incident prevention. It includes information such as from and to street address, names of cross streets, number of lanes, number of shoulders, daily volume, and road width. Road classification has been based on the Houston Galveston Regional Transportation Study classification system. The color-coding of roads for display has been performed using this classification system.

Latitude and longitude information has been included for referencing roads and intersections. This latitude and longitude information could also be used for reference using Texas Reference Marker locations.

For use in network analyses, address, turntable, incident, and resource databases were created. The turntable database contains descriptions of turns allowed at all intersections within the network. This database also records which arcs are one way. ARC/INFO<sup>(C)</sup> representation inherently allows two way traffic on any arc. By restricting turns, the turntable makes the system realistic. In addition, it includes the expected delay due to the turn for each turn. This delay number has been entered for each turn in the database. The address database was generated to facilitate the handling of multiple addresses. With this database, the operator can perform address-matching with the aid of a built-in soundex. The incident database was integrated into the CM map coverage by using an external service patrol database to provide locational incident information for daily and cumulative operations. Finally, various resource databases were developed for points and polygons to provide on-line information associated with fire stations, police stations, park and ride lots, zip code areas, and so forth.

#### 4.0 APPLICATIONS DEVELOPMENT FOR INCIDENT MANAGEMENT

Applications development refers to development of computer software that would perform a specific task or a set of tasks. In the context of the automated incident management plan, applications development refers to development of programs to automate the tasks required for incident management, including tasks for traffic center operations as well as tasks for planning and analysis of incidents.

An applications development platform or environment is a software that includes a set of programming tools for development of specific applications. The programs have been written for the automated incident management plan using a PC-ARC/INFO<sup>(C)</sup> GIS applications development platform.

The prototype/demonstration applications described in this section were developed using the PC-ARC/INFO<sup>(C)</sup> GIS development package on a PC/486 microcomputer environment. These applications, either singularly or in combination, demonstrate the functionality expressed in the system architecture (shown in Figure 2.2) for the automated incident management plan.

All applications have been developed using the Simple Macro Language (SML<sup>(C)</sup>) of PC-ARC/INFO<sup>(C)</sup>. Several PC-ARC/INFO<sup>(C)</sup> modules have been used for the development and operation of these applications. DATA CONVERSION and OVERLAY modules were required only for development and, therefore, these were not used for operations. Most SML<sup>(C)</sup> programs that have been written operate within the ROUTE module and call other modules whenever needed. A call to another module generally involves the following:

- 1) Exiting from one module, such as ROUTE, after saving the state of the current executing application;
- 2) Entering another module, such as ARCPLOT, to perform a set of tasks in that module;
- 3) Exiting the second module; and
- 4) Entering the original module to update the state of the application based on tasks completed in the second module.

User-friendly interfaces have been created for each application. The purpose of a user interface is to provide the operator a way to interact with the computer in general, and with the application in particular, without needing to know the internal architecture of either the computer or the application. User interfaces should be user friendly so that the operator can learn and operate the applications with minimal training and prior knowledge of the application itself.

All interfaces are mouse and menu driven. Keyboard interaction has been kept to a minimum; wherever possible, mouse selectable menus have been provided. Input forms have been largely eliminated to minimize mistakes in entering the names of streets and other information. Pull down menus have been emulated in most applications.

The menus are white on black background. Most menus have between 3 and 8 items to prevent confusion. Menus do not normally exceed a depth of 5 levels to allow the operator to maintain coherence. Consistency has been the goal in the menus so that an item on one menu has the same meaning across all menus. The only user-interface limitation in these applications is the lack of support for *graphical user-interfaces* in PC-ARC/INFO<sup>(C)</sup>.

All the dynamic route guidance and incident response functions have also been brought together in a prototype application which has been named INCMAN (acronym for INCident MANagement). A user's guide for INCMAN is attached to this report as Appendix B.

#### **4.1 APPLICATIONS FOR INCIDENT MANAGEMENT**

Applications for incident management operations will be used in traffic management centers by operators for real-time traffic management. The applications have been classified into three categories:

- 1) Dynamic Route Guidance applications, for alternate routing and variable message sign control;
- 2) Incident Response applications, for blocking incident areas and providing fleet management to service patrols; and



- 3) Resource Management applications, which allow the operator to:
  - (a) perform graphical query on incident management resources, such as police, fire and hospital, and
  - (b) contact incident management resources, such as police, fire and hospital.

#### **4.1.1 Dynamic Route Guidance Applications**

Applications for dynamic route guidance may be classified into the following four categories:

- 1) Wide Area Alternate Routing for routing over a general area;
- 2) Local Area Alternate Routing over specific areas using frontage roads and interchange ramps;
- 3) Hazard Area Alternate Routing; and
- 4) Travel Time Simulation for estimating expected travel times on incident affected roads.

All of the above dynamic route guidance applications share the following underlying models:

- 1) Dijkstra's shortest path algorithm, which is built into PC-ARC/INFO<sup>(C)</sup>. This algorithm has been implemented in PC-ARC/INFO<sup>(C)</sup> to accept arc length or travel time for the purpose of finding the shortest path.
- 2) A data model where:
  - (a) volume-to-capacity-ratios are related to speeds, as specified in the Highway Capacity Manual; and
  - (b) the volume-to-capacity-ratios change for each street segment depending on either of the following:
    - (i) Time of the day, using historical hourly volume data; or
    - (ii) Blockage of one or more lanes caused by incidents.

Route guidance applications developed for this study are discussed in the following sections, including the program purpose, function, and development.

#### 4.1.1.1 Wide Area Alternate Routing

*Purpose:*

Wide Area Alternate Routing refers to alternate routing over a large area, in general, involving travel of several miles. Such routing is required in situations where a freeway or major arterial is experiencing severe non-recurrent congestion and routing over other roads would help relieve congestion. This program is designed to provide routing directions as well as expected travel times for the route.

*Function/Capability:*

Alternate routing over a wide area requires that the operator specify one or more of the following before attempting to generate an alternate route:

- 1) Simulate an incident location either:
  - (a) by placing a block on an intersection for incidents which block all lanes; or
  - (b) by blocking one or more lanes on a segment of a road; or
- 2) Specify the time of the day by calling the computer system clock or entering the time of day.

The user will specify the origin and destination on the computer screen. The system will then generate the following:

- 1) Optimal route between the origin and destination. This route will automatically be generated as an alternate route whenever the expected travel time on the freeway or regular route is higher than the travel time on the alternate route.
- 2) The directions for alternate route and expected travel time will also be generated, which in turn may be communicated to motorists through variable message signs or HAR.

An example of the directions generated for alternate route are shown in Figure 4.1.

Directions for route 1

Start at NORTH FWY

Travel	2.95 miles	Turn right to BELTWAY 8N
Travel	7.10 miles	
Total	10.06 miles	

Travel time is  
10.971  
minutes

Click Anywhere to Exit



Figure 4.1 Directions Generated for a Wide Area Route

*Method of Development:*

This application has been developed using commands in the ROUTE and ARCEDIT modules of PC-ARC/INFO<sup>(C)</sup>. In the ROUTE module, the commands include the simulated blocking of a roadway and generation of least-time route. In the ARC/EDIT module, the commands are used to update the travel times in the database.

4.1.1.2 Local Area Frontage Road Alternate Routing

*Purpose:*

Local area alternate routing refers to alternate routing in a limited area using frontage/feeder roads to relieve the congestion on freeways. This application uses the frontage roads, ramps and freeway interchanges on the base map to demonstrate local area alternate routing.

*Function/Capability:*

The application is similar to the wide area routing application and uses much of the design of the wide area routing application. The operator must specify one or more of the following before attempting to generate an alternate route:

- 1) Simulate an incident location either:
  - (a) by placing a block on an intersection for incidents which block all lanes; or
  - (b) by blocking one or more lanes on a segment of a road; or
- 2) Specify the time of the day by calling the computer system clock or by entering the time of day.

The user will then specify the origin and destination on a freeway on the computer screen. The system will generate the following:

- 1) Optimal route between the origin and destination. This route will automatically use frontage roads, ramps, and freeway interchanges as alternate routes whenever the expected travel time on the freeway mainlanes is higher than the expected travel time on frontage roads;

- 2) The directions indicating exit and entrance ramps, as well as expected travel time, for alternate route will also be generated. These in turn may be communicated to motorists through variable message signs or HAR.

An example of a typical computer display of frontage roads and ramp interchanges, which may be used for local area alternate routing, is shown in Figure 4.2.

The application is different in that if an intersection is blocked or if a freeway segment has a severe congestion due to blockage of one or more lanes, the system will generate a route using one or more of frontage roads, ramps, and freeway interchanges. The system has the intelligence to:

- 1) Consider the correct entrance and exit ramps; as well as
- 2) Consider a U-turn at a freeway underpass/overpass.

*Method of Development:*

This application has been developed using commands in the ROUTE and ARCEDIT module of PC-ARC/INFO<sup>(C)</sup>. In the ROUTE module, the commands include the blocking of the roadway or individual intersection and generation of least-time route. In the ARC/EDIT module, the commands are used to update travel times impedance in the underlying database. In addition, for this application appropriate turn restrictions and resistances have been built into the underlying data model to provide intelligence for entrance/exit ramps and overpass/underpass.

4.1.1.3. Alternate Routing Around an Area of Hazardous Material Spill

*Purpose:*

This application has been developed in conjunction with hazard area blocking application. The focus of the application is to provide an alternate route around an area wide incident, such as a chemical plant explosion or gas leak where all roads leading into or passing through this area have been blocked.

A.I.M.S.  
Frontage Road Simulation

ROUTE	TRAFFIC UPDATE	UTILITIES	QUIT
-------	----------------	-----------	------

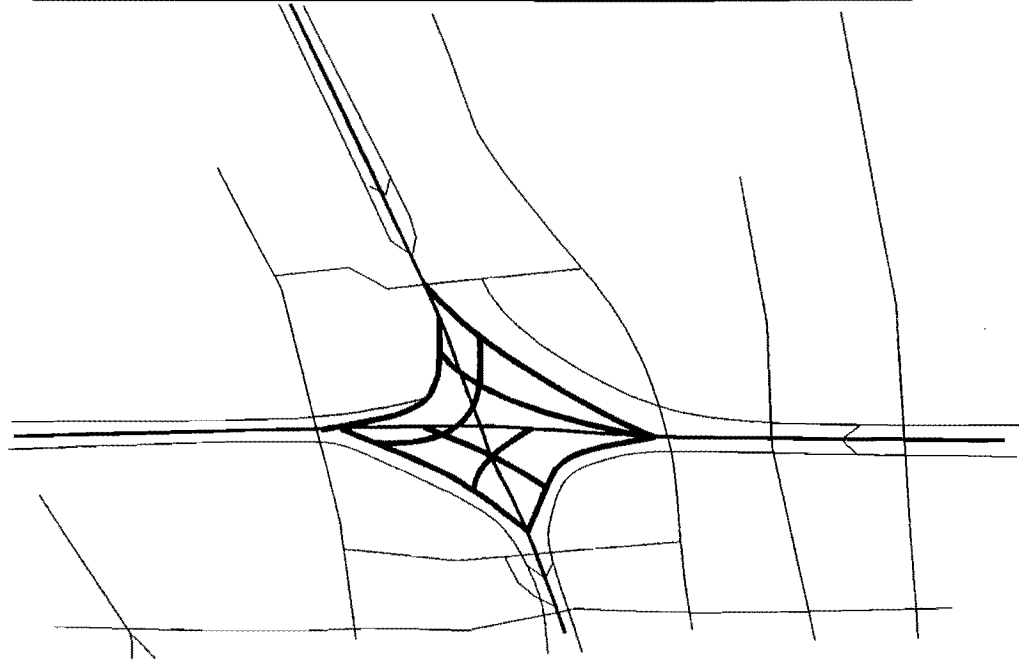


Figure 4.2 Frontage Roads and Ramp Interchange for Local Area Alternate Routing

### *Function/Capability*

When an incident requiring evacuation occurs, the affected area is blocked. The underlying coverage is updated accordingly. The user can then perform wide area origin-destination alternate routing to generate various alternate routes for motorists on roads which pass through the area. Figure 4.3 shows an example of an alternate route generated around a blocked hazard area. The motorists can then be informed of these alternate routes over HAR or variable message signs or other means.

### *Method of Development:*

This application has been developed using commands in the ROUTE module of PC-ARC/INFO<sup>(C)</sup>. The commands include the blocking of individual intersections and generation of least-time route.

#### 4.1.1.4. Travel Time Simulation System

### *Purpose:*

The Travel Time Simulation system provides estimated travel time and delay on freeways. These times can be communicated to motorists over variable message signs or HAR, or other media.

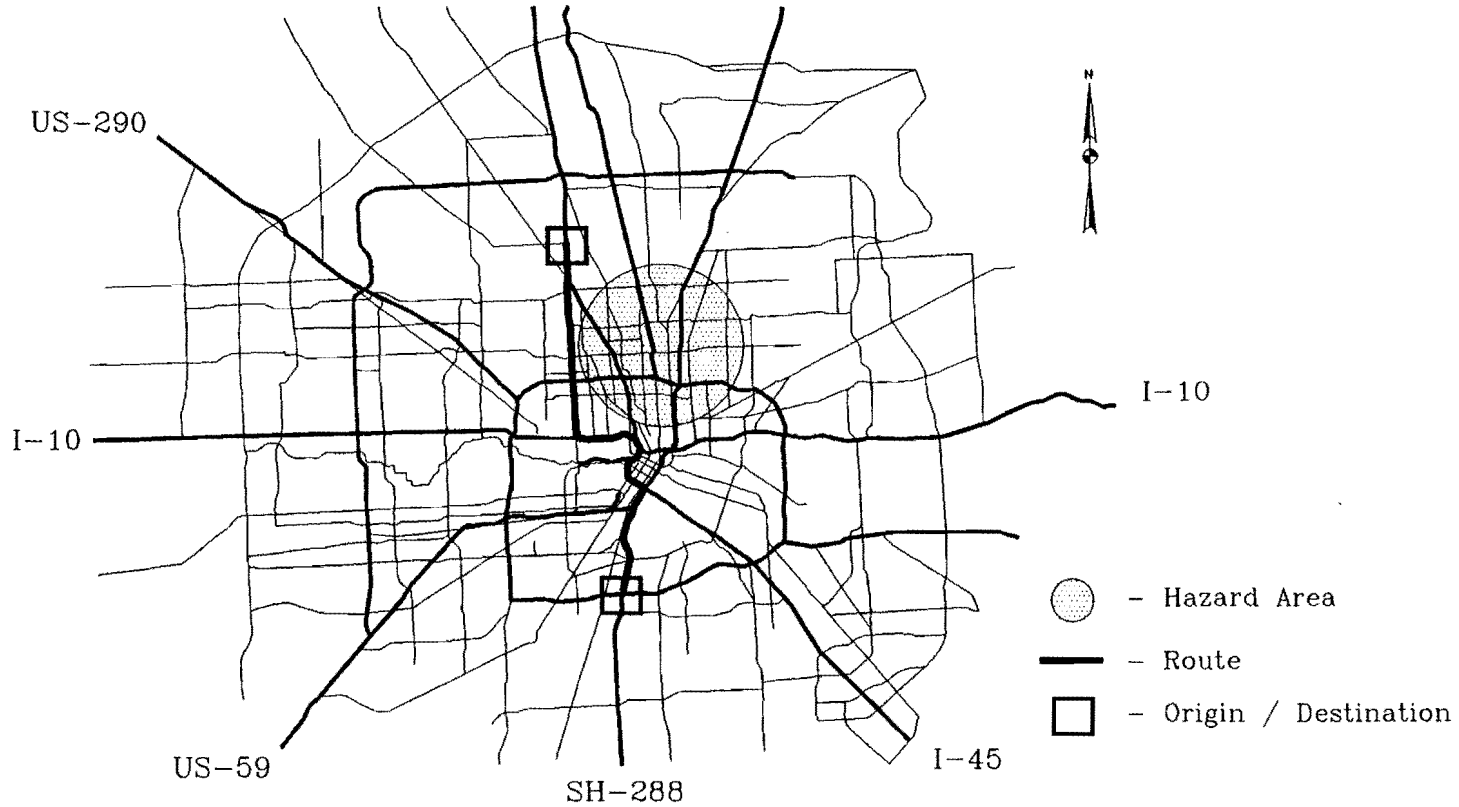
### *Function/Capability*

The simulation is similar to alternate routing applications. Before generating the travel times and delays, the operator must perform one of the following:

- 1) Simulate an incident location either:
  - (a) By placing a block on an intersection for incidents which block all lanes; or,
  - (b) By blocking one or more lanes on a segment of the freeway; and,
- 2) Specify the time of the day by calling the computer system clock or by entering the time of day.

MAIN MENU  
 A.I.M.S.:INCMAN (INCident MANAgement Support Tool)

INCIDENTS VAN UPDATE DISPATCH TRAFFIC UPDATE UTILITIES QUIT



36

Figure 4.3 Example of Routing Around a Hazard Area



The operator will then identify a number of locations on the freeway network to specify the path for a route. The system will then generate the travel time on the route. An example of the result of a travel time simulation is shown in Figure 4.4.

*Method of Development:*

This application has been developed using commands in the ROUTE and ARCEDIT module of PC-ARC/INFO<sup>(C)</sup>. In the ROUTE module, the commands include generation of least-time route. In the ARCEDIT module, the commands are used to update travel times in the database.

The relationship expressed in the Highway Capacity Manual between speed and volume-to-capacity-ratio has been used for estimating the travel time. Underlying this estimate of travel time is a data model where volume to capacity ratios change for each roadway segment depending on either of the following:

- 1) Historical volumes corresponding to the time of the day; or
- 2) Blockage of one or more lanes caused by incidents.

4.1.1.5. Comparison of Travel Times Using Real and Historical Data

*Purpose:*

The dynamic route guidance applications provide the operators with estimated travel times in addition to generating routes and directions. Estimated travel times should approximate the real travel times to be useful in traffic center operations. The suitability of dynamic route guidance with the proposed approach has also been investigated for the automated incident management plan. The estimated travel times generated by the simulation for various incident conditions have been compared with a sample of data collected during a real-time cellular phone demonstration study (CPS) [10]. In that study, travel time information was collected in real-time using cellular phones. The data was entered into a database on a local area network where real-time information could be used for review. Comparisons of the estimated travel times to measured trend times are provided in the following section.

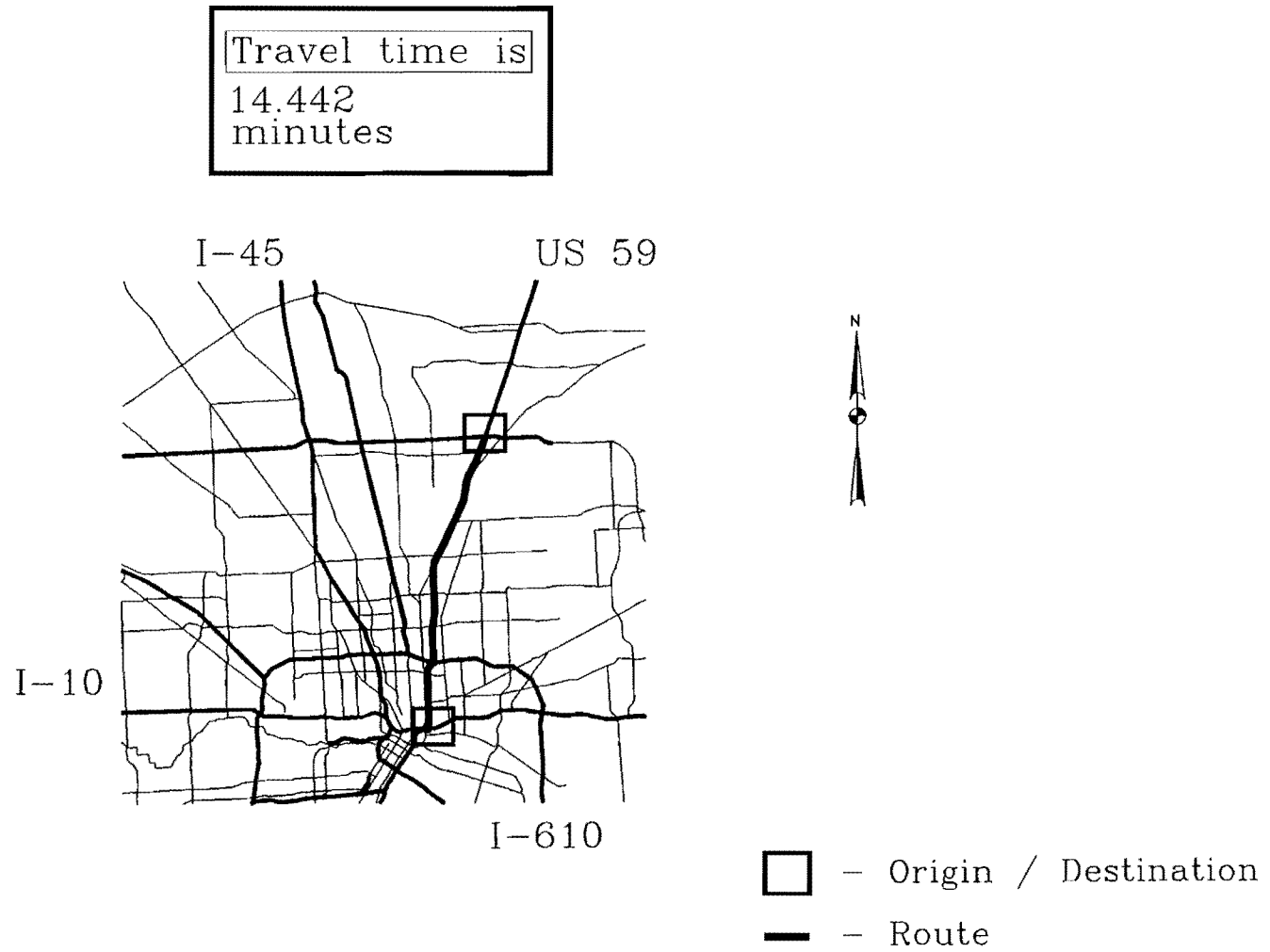


Figure 4.4 A Result of a Travel Time Simulation

*Data Model:*

- 1) Volume-to-Capacity (V/C) vs. Speed, were based on the relationship used in Highway Capacity Manual. Discrete speed values are applied to each freeway V/C threshold.
- 2) Volume of normal traffic is the historical hourly volume in the data. As the time of day changes, the volume changes. This, in turn, changes the V/C ratio, and therefore the speed.
- 3) When an incident occurs or is cleared, the capacity of an affected freeway segment is updated in proportion to the number of lanes available.

*Available Capacity =*

$$\text{Maximum Capacity} \times \frac{\text{Number of Lanes Available}}{\text{Maximum Number of Lanes}}$$

Then the V/C ratios are recomputed as are the speeds corresponding to the V/C ratio.

- 4) The travel time depends upon the distance and the speed. The distance is given by the length of the arc in the GIS database. Thus, new travel times are computed every time there is a change in volume or a change in capacity for any street segment.

*Approach:*

The travel time simulation was performed on the following freeways on which data was generated during the cellular phone study:

- 1) Freeway (I-45), covering 23.7 miles (38.13 km) which overlapped 23.1 miles (37.17 km) used for CPS operation, and
- 2) Freeway (US-59), covering 16.5 miles (26.55 km), which overlapped 17.2 miles (27.67 km) of CPS operation.

The travel time simulation was performed for each of 3 hours in the morning and 3 hours in the evening, as below:

- 1) Morning hours were 6 AM, 7 AM and 8 AM, and
- 2) Evening hours 4 PM, 5 PM and 6 PM.

Wherever possible, for each hour, the travel time simulation was performed for the following three cases:

- 1) Case I. Where no incidents are reported;
- 2) Case II. Where incidents blocking one lane are reported; and
- 3) Case III. Where incidents blocking 2 lanes are reported.

Data collected in 1992 during the cellular phone study were used. Wherever possible, more than 5 random samples from cellular phone study data were selected and averaged for each case. These were then compared with the simulated travel times.

*Case I: No Incidents*

Table 4.1 compares the cellular phone real-time data with the simulated data for Case I. The average difference between real speeds and simulated speeds are:

North Freeway	13.3%
Eastex Freeway	8.4%
Combined	10.7%

However, the variation in individual cases indicates inconsistency. The difference between real and simulated values is greater for the North Freeway. This may be attributed to unavailability of real hourly volume data for most segments of North Freeway. The hourly volumes for North Freeway are mostly extrapolated as compared to measured volumes, whereas for Eastex Freeway, most hourly volumes are as measured in other studies.

Case I: No Incidents Reported						
North Fwy				Eastex Fwy		
Time	CPS Avg. Speed	Simulated Speed	Diff. % Speed	CPS Avg. Speed	Simulated Speed	Diff. % Speed
All Speeds in Miles Per Hour (MPH)						
6am	53.90	47.20	12.40	47.10	39.10	17.08
7am	47.60	46.30	2.72	42.50	46.70	9.90
8am	52.50	46.90	10.69	50.50	48.00	4.97
4pm	55.20	41.00	25.79	49.70	46.10	7.39
5pm	49.00	40.30	17.75	43.80	41.80	4.62
6pm	52.50	48.50	7.56	47.80	50.80	6.30
All Speeds in Kilometers Per Hour (KPH)						
6am	86.70	75.90	12.40	75.80	62.90	17.08
7am	76.60	74.50	2.72	68.40	75.10	9.90
8am	84.50	75.50	10.69	81.30	77.20	4.97
4pm	88.80	66.00	25.79	80.00	74.20	7.39
5pm	78.80	64.80	17.75	70.50	67.30	4.62
6pm	84.50	78.00	7.56	76.90	81.70	6.30

Table 4.1. Travel Time Simulation vs. Real-Time Data for Case I

*Case II: Incident reported with any one lane blocked*

In this case, incidents are superimposed on the model. The simulated speeds may differ more significantly from real speeds because the effect of incidents are inherently more unpredictable. Although the same number of lanes may be blocked, the delays due to incidents vary with time and location. Moreover, the delay could vary from lane to lane even for the same location. Table 4.2 compares the cellular phone real-time data with the simulated data for Case II.

Case II: Incidents with One Lane Blocked						
North Fwy				Eastex Fwy		
Time	CPS Avg. Speed	Simulated Speed	Diff. % Speed	CPS Avg. Speed	Simulated Speed	Diff. % Speed
All Speeds in Miles Per Hour (MPH)						
6am	39.00	45.00	15.38	33.60	37.90	12.79
7am	36.20	45.40	25.41	34.70	43.00	23.64
8am	*	*	*	*	*	*
4pm	*	*	*	*	*	*
5pm	35.60	39.10	9.80	28.50	37.50	31.60
6pm	*	*	*	*	*	*
All Speeds in Kilometers Per Hour (KPH)						
6am	62.80	72.40	15.38	54.10	61.00	12.79
7am	58.20	73.00	25.41	55.80	69.20	23.64
8am	*	*	*	*	*	*
4pm	*	*	*	*	*	*
5pm	57.30	62.90	9.80	45.90	60.30	31.60
6pm	*	*	*	*	*	*

\* data not available

Table 4.2. Travel Time Simulation vs Real-Time Data for Case II

*Case III: Incidents reported with 2 lanes blocked.*

The limitations for Case II also apply here. Table 4.3 compares the cellular phone real-time data with simulated data for Case III. Comparisons are limited because of availability of data as can be seen in Table 4.3. Using the speeds indicated in the table, the model predicts a delay of 3 to 5 minutes for an incident that causes blockage of two lanes. However, the delays recorded in the real-time database for an incident condition involving blockage of 2 lanes are significantly higher; between 12 and 20 minutes. This indicates a deficiency in the model.

Case III: Incidents with 2 Lanes Blocked						
North Fwy				Eastex Fwy		
Time	CPS Avg. Speed	Simulated Speed	Diff. % Speed	CPS Avg. Speed	Simulated Speed	Diff. % Speed
All Speeds in Miles Per Hour (MPH)						
6am	34.70	43.40	24.80	*	*	*
4pm	35.40	39.30	11.01	*	*	*
5pm	*	*	*	31.70	34.70	9.46
All Speeds in Kilometers Per Hour (KPH)						
6am	55.80	69.80	24.80	*	*	*
4pm	57.00	64.00	11.01	*	*	*
5pm	*	*	*	51.00	55.80	9.46

\* data not available

Table 4.3. Travel Time Simulation vs. Real-Time Data for Case III

*Limitations:*

First, there are major limitations in the simulation model itself. The effect of an incident with one or more lanes blocked shows considerable differences between real-time data and simulated data. The following must be considered as limitations in the model:

- 1) The travel time model is dependent upon the length of arcs which are affected by incidents. This is because:

- (a) V/C ratios are related to speed; and
  - (b) the travel time is a function of length of street segment and the speed. Since the length of arc varies considerably, the affect on travel time differs considerably for the same V/C ratios. To offset the effect of length of arc, dynamic segmentation or some other suitable mechanism should be used.
- 2) The V/C ratio to speed relationship has been expressed by discrete values in the model. This does not adequately express the relationship. Using a curve as shown in the Highway Capacity Manual instead of discrete values to represent the V/C to speed relationship could improve the model in this respect.
  - 3) The capacity of a freeway segment is linearly related to the number of lanes in the model. Incident impact relationships are actually exponentially related for shoulder activity, one-lane block, two blocks and so forth [11].
  - 4) In the "V/C to Speed" model, as expressed in the Highway Capacity Manual, the speed becomes unstable as freeway flow approaches capacity. In the model, discrete speed values have been used for the unstable region. This issue also needs to be addressed.
  - 5) No redistribution of volumes were provided for the simulated model.

Second, CPS data was collected over segments totalling 16.5 miles (26.55 km) on EASTEX FWY and 23.1 miles (37.17 Km) on NORTH FWY, whereas the computer generated data over 17.2 miles (27.67 km) on EASTEX FWY and 23.7 (38.13 km) miles on NORTH FWY. The street segments were 96% overlapped due to the unavailability of dynamic segmentation, (refer Section 3.5). Consequently, comparing travel times was not appropriate. Therefore the speeds were compared instead.

Third, the time delay due to incidents is more difficult to predict because the effects of incidents depends upon the type of incident, location, time of day, weather, and other factors.

Finally, hourly volumes were not available on all street segments used for this study. However, daily volumes were available for all segments and hourly volumes had to be extrapolated from the daily volumes for several street segments.



*Summary:*

An approach for dynamic route guidance using real and historical data on a GIS platform has been proposed. During preliminary tests on this approach, it has been found that some simulated travel times closely approximate real data, whereas others (particularly incident conditions) are noticeably higher or lower. Further study is needed on the usefulness and validity of this approach with the following enhancements:

- 1) using larger data sets of hourly volumes; and
- 2) the refinement of the underlying data model to overcome some of the limitations in the model that has been used.

4.1.1.6. Other Commercial Software in the Automated Incident Management Plan

*Purpose:*

The automated incident management plan can be enhanced by integrating automated incident management applications with other commercially available software that may be suitable for incident management applications such as CAMEO<sup>(C)</sup>, HAZTRANS<sup>(C)</sup>, and PASSER<sup>(C)</sup>. CAMEO<sup>(C)</sup> is a software for management of chemical emergencies; HAZTRANS<sup>(C)</sup> is a software for routing and risk management of hazardous materials; and PASSER-II<sup>(C)</sup> and PASSER-III<sup>(C)</sup> [12] are software for signal timing analysis.

*Function/Capability*

PASSER-II<sup>(C)</sup> and PASSER-III<sup>(C)</sup> are two software for signal timing analysis and optimization. While PASSER-II<sup>(C)</sup> is designed for arterials, PASSER-III<sup>(C)</sup> is designed for fixed sequence signalized diamond interchanges. The two softwares require different types of inputs and have different interfaces; however, the outputs of both of these include the average delay for intersections. It was noted that this delay generated by PASSER-II<sup>(C)</sup> or PASSER-III<sup>(C)</sup> could be used to update the turn table for GIS applications in PC-ARC/INFO<sup>(C)</sup>. It was also noted that both PASSER-II<sup>(C)</sup> and PASSER-III<sup>(C)</sup> outputs would require similar algorithms; integrating one also indicates the integratability of the other.

The output of PASSER-II<sup>(C)</sup> was found to be more suited to development of an algorithm. Hence, an algorithm was developed and implemented for PASSER-II<sup>(C)</sup> output.

PASSER-II<sup>(C)</sup> has been integrated with automated incident management plan applications written in PC-ARC/INFO<sup>(C)</sup>. In this application, the output of PASSER-II<sup>(C)</sup> is post processed and converted to a format suitable for use by dynamic route guidance applications. The turn restrictions and average vehicle delay for any single intersection generated by PASSER-II<sup>(C)</sup> are used to update the corresponding turn data in the PC-ARC/INFO<sup>(C)</sup> database. Any subsequent routing based on that database will take into account the updated intersection delays read into the database.

The integration algorithm involves matching the names of intersecting streets as available in the output of PASSER-II<sup>(C)</sup> with the names of intersecting streets, as given in the turn table. If the names are misspelled in either of the two or if aliases are used, the system will not find a match and will not perform an update operation on the turn table.

PASSER-III<sup>(C)</sup> output data for frontage road/arterial interchanges can also be similarly read into PC-ARC/INFO<sup>(C)</sup> database. While this has not been implemented, the methodology would be the same as for PASSER-II.

CAMEO<sup>(C)</sup>, a software for Computer Aided Management of Emergency Operations, was obtained at no cost from National Oceanic and Atmospheric Administration (NOAA)/Hazardous Materials Response Branch, Seattle, Washington. CAMEO<sup>(C)</sup> has been designed for emergency planning for chemical incidents. It includes an independent GIS, a complete database of chemicals, along with their hazards and treatments, and a database of organizations involved in production, marketing, and transportation of chemicals. All databases are in dBASE<sup>(C)</sup> format and are based on TIGER<sup>(C)</sup> files. It was noted that the need to share data between CAMEO<sup>(C)</sup> and PC-ARC/INFO<sup>(C)</sup> automated incident management plan was minimal because of the existence of an independent GIS in CAMEO<sup>(C)</sup>. For this reason, CAMEO<sup>(C)</sup> and other PC-ARC/INFO<sup>(C)</sup> automated incident management plan applications were only brought under a common user interface.

HAZTRANS<sup>(C)</sup>, a software for Hazardous Material Routing and Risk Management from Abkowitz and Associates, was only considered during the initial stage of this study. The software design group of Abkowitz and Associates was contacted for information.

HAZTRANS<sup>(C)</sup> is an independent GIS package written in C (DOS version). HAZTRANS<sup>(C)</sup> is exclusively on PC platform and is being ported to Windows and Windows/NT. However, it cannot be accessed externally with any programming language; nor does it have a macro language. HAZTRANS<sup>(C)</sup> uses RAIMA database format and cannot be exported to DBF or other commercial formats. The only way data can be shared with PC-ARC/INFO<sup>(C)</sup> is through ASCII files to which HAZTRANS<sup>(C)</sup> can export its database. HAZTRANS<sup>(C)</sup> thus appeared to have limited features to support integration with other software. Moreover, even this could not be investigated as a regular site license for \$17,500 was required and short term loan was not available. Considering these two factors, integration of HAZTRANS<sup>(C)</sup> was not pursued further.

*Development:*

The application for integrating PASSER-II<sup>(C)</sup> with the automated incident management plan runs off-line in the traffic management center. Only one intersection is handled at one time, though the application can be extended to handle multiple intersections. When the operator/engineer selects PASSER-II<sup>(C)</sup> through the user-interface for the automated incident management plan, the program updates average vehicle delay in the turn-table file automatically following execution of PASSER-II<sup>(C)</sup>. The output file is first read and keywords are searched to locate the names of intersections whose signal timing has been generated by PASSER-II<sup>(C)</sup>. The delay data is then read from the output data file of PASSER-II<sup>(C)</sup>. The program now calls the TABLES module and searches the names of the intersections in the turn table. Following this, the directions of turns are matched for the intersection. The turn table is then updated with the delay times read from the output data file.

#### **4.1.2. Incident Response Applications**

Applications for Incident response have been classified into three categories:

- 1) Fleet Management for Service Patrols;
- 2) Lane Blocking; and,
- 3) Hazard Area Blocking.

Incident response applications developed for this study are discussed in the following sections, including the program purpose, function, and development.

#### 4.1.2.1. Fleet Management for Service Patrols

*Purpose:*

Freeway service patrols are typically on call 24 hours per day to provide assistance during incidents. This prototype application has been designed to provide improved response times through fleet management of the service patrols. This prototype application provides the following to the control center for service patrols:

- vehicle tracking with or without Automatic Vehicle Location;
- incident monitoring; and
- vehicle dispatch guidance and directions.

*Function/Capability:*

This application provides a graphical display of all incident locations as well as color-coded locations of all service patrol vehicles showing for each patrol vehicle whether it is patrolling the freeways or providing assistance at an incident location. Associated with this is the location information in text form which can be displayed as and when required.

The operator in the control center can perform one or more of the following functions with this application:

- 1) Enter and retract incident information;
- 2) Enter and update location and status of service patrol vehicles;
- 3) Find and dispatch the closest service patrol vehicle available to an incident site using travel time simulation application;
- 4) Update variable message signs in conjunction with dynamic route guidance as well as incident response applications;
- 5) When the service patrols are equipped with AVL transponders, track service patrol vehicles automatically without need for manual entry. However, AVL functionality is limited to the display of moving vehicles in the demonstration

program. More research is needed on the use of PC-ARC/INFO<sup>(C)</sup> for an advanced AVL system; and

- 6) Other functions such as graphical query on street segment database, on-screen display and printing of travel directions as well as of incident and service patrol vehicle locations.

Figure 4.5 shows a display of a prototype fleet management application, INCMAN, which also includes features of dynamic route guidance and incident response applications. In addition, Figure 4.6 shows the locations of MAP vans on the screen as well as their location in text form.

*Development:*

In order to display the symbols denoting service patrol vehicles, an additional symbol table was created in ARC/INFO<sup>(C)</sup>. This symbol table is used throughout the application. This application operates in the ROUTE module. However, several external programs that have been written for this application are called from the ROUTE module for this application. These programs are used to maintain consistency in the location of service patrol vehicles. A number of temporary files and SML<sup>(C)</sup> programs are generated by these external programs during the course of execution of this application. These temporary data files are used to record service patrol vehicle locations and other data required for display functions used in the application. In addition, dynamic route guidance has been used to dispatch service patrol vehicles and to control variable message signs.

4.1.2.2. Lane Blocking

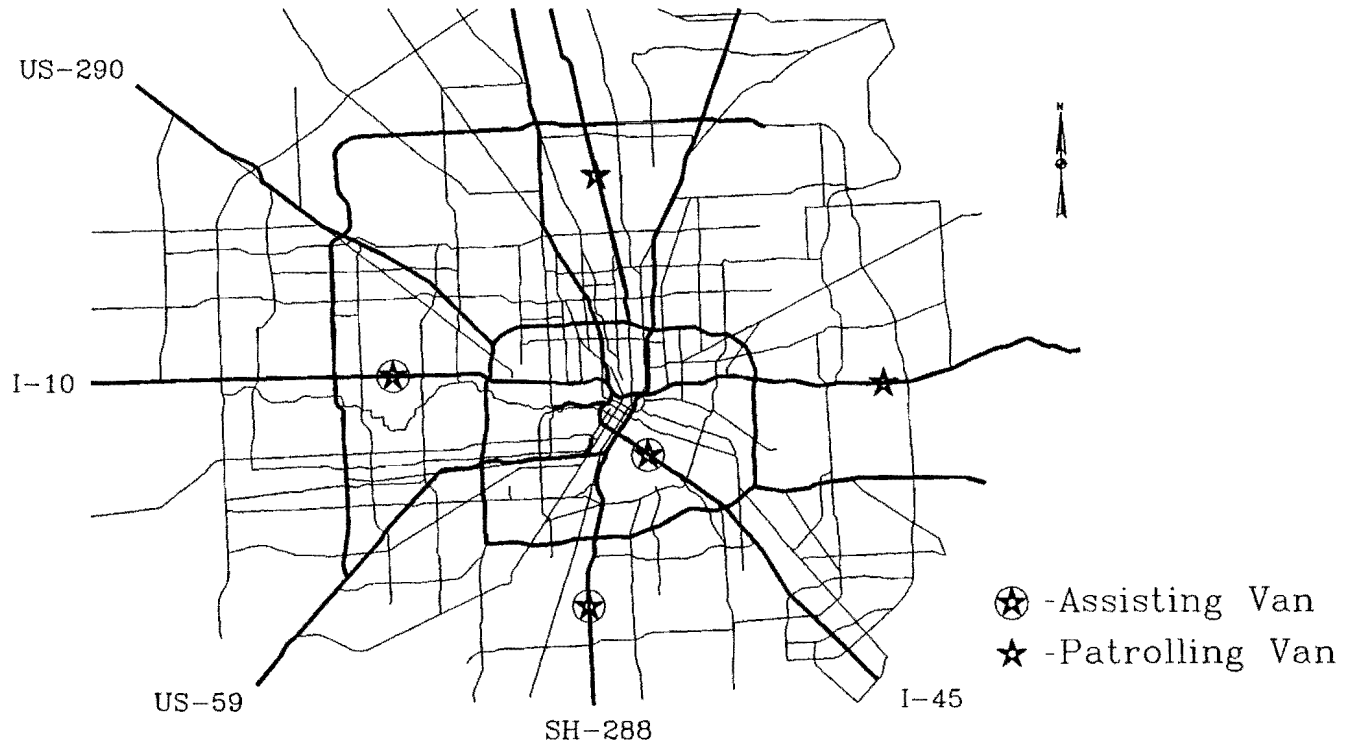
*Purpose:*

The purpose of this application is to allow the operator in a traffic management center to perform the following:

- 1) Estimate the delay caused by an incident using travel time simulation for the incident condition as indicated by the number of lanes blocked; and,

MAIN MENU  
A.I.M.S.:INCMAN (INCident MAnagement Support Tool)

INCIDENTS VAN UPDATE DISPATCH TRAFFIC UPDATE UTILITIES QUIT



50

Figure 4.5 Van Locations in Fleet Management Application

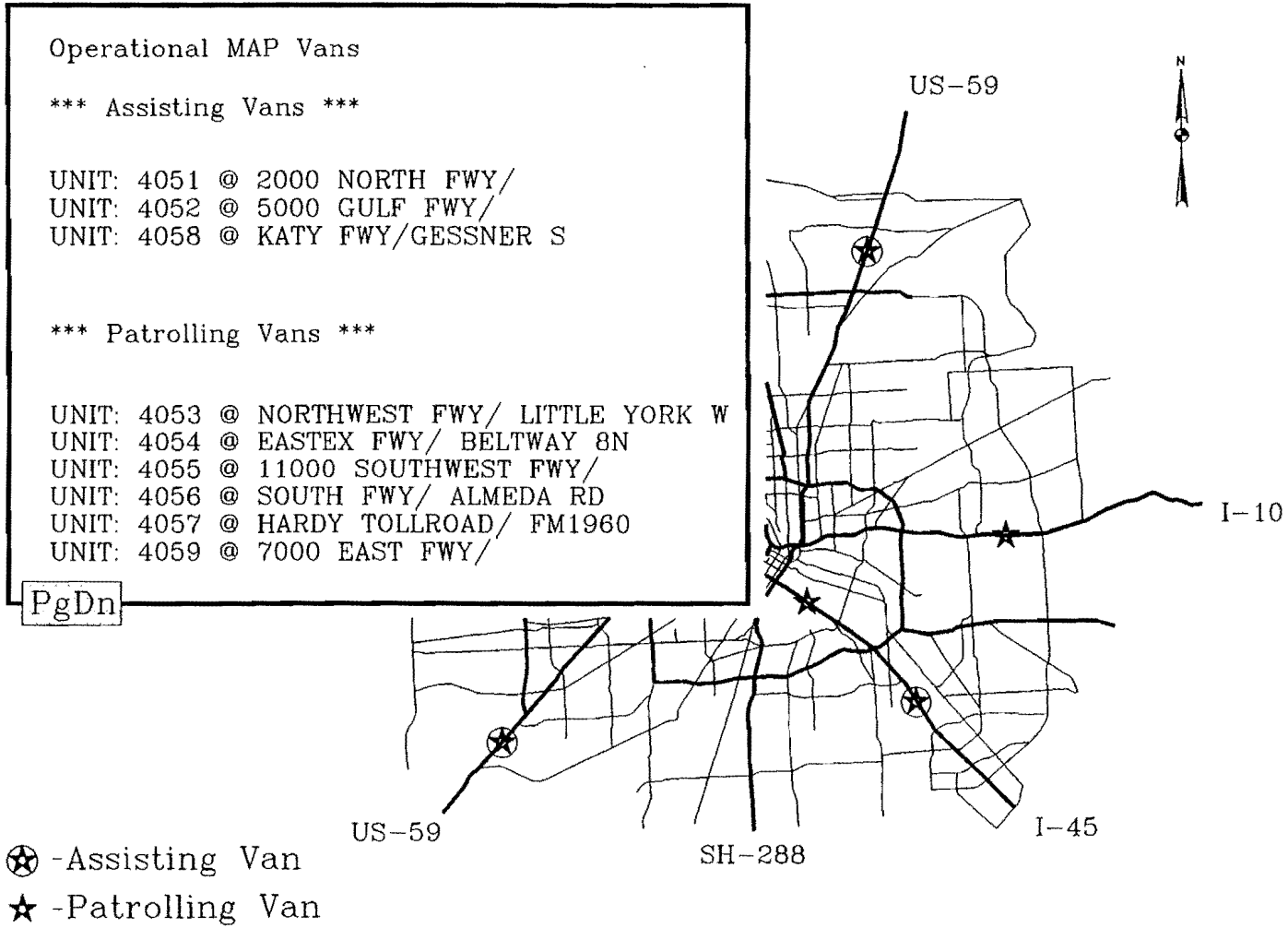


Figure 4.6 List of Map Van Locations

- 2) Estimate if there is a need for alternate routing, and, if the need exists, generate alternate routes and communicate alternate routes, to motorists, via variable message signs or HAR.

*Function/Capability:*

The system allows an operator to simulate lane blocking incidents. The blocked lanes may be simulated on any segment of the road network. The direction of traffic on the lane must be specified by the operator because a single line on the map on the computer screen usually represents both directions of flow of traffic. The operator/user may block one or more lanes of the same segment at any time and may also block several different lanes in different segments at the same time. Once this has been done, the operator may estimate the delay using travel time simulation or generate alternate route using wide or local area alternate routing programs.

A typical display for the lane/area blocking application is illustrated in Figure 4.7.

*Development:*

This application operates in the ROUTE module. However, to simulate blocked lanes, it calls the ARCEDIT module. The user specifies the lanes to be blocked and the system updates the underlying database accordingly. The control, then, returns to the ROUTE module where the operator could perform further alternate routing.

#### 4.1.2.3 Hazard Area Blocking

*Purpose:*

This application has been developed in conjunction with hazard area alternate routing applications. The purpose of the application is to block an area where a major incident, such as a chemical plant explosion or gas leak, has occurred. This application allows the authorities to quickly identify the area and block the area in order to generate alternate routing and information for the motorists.



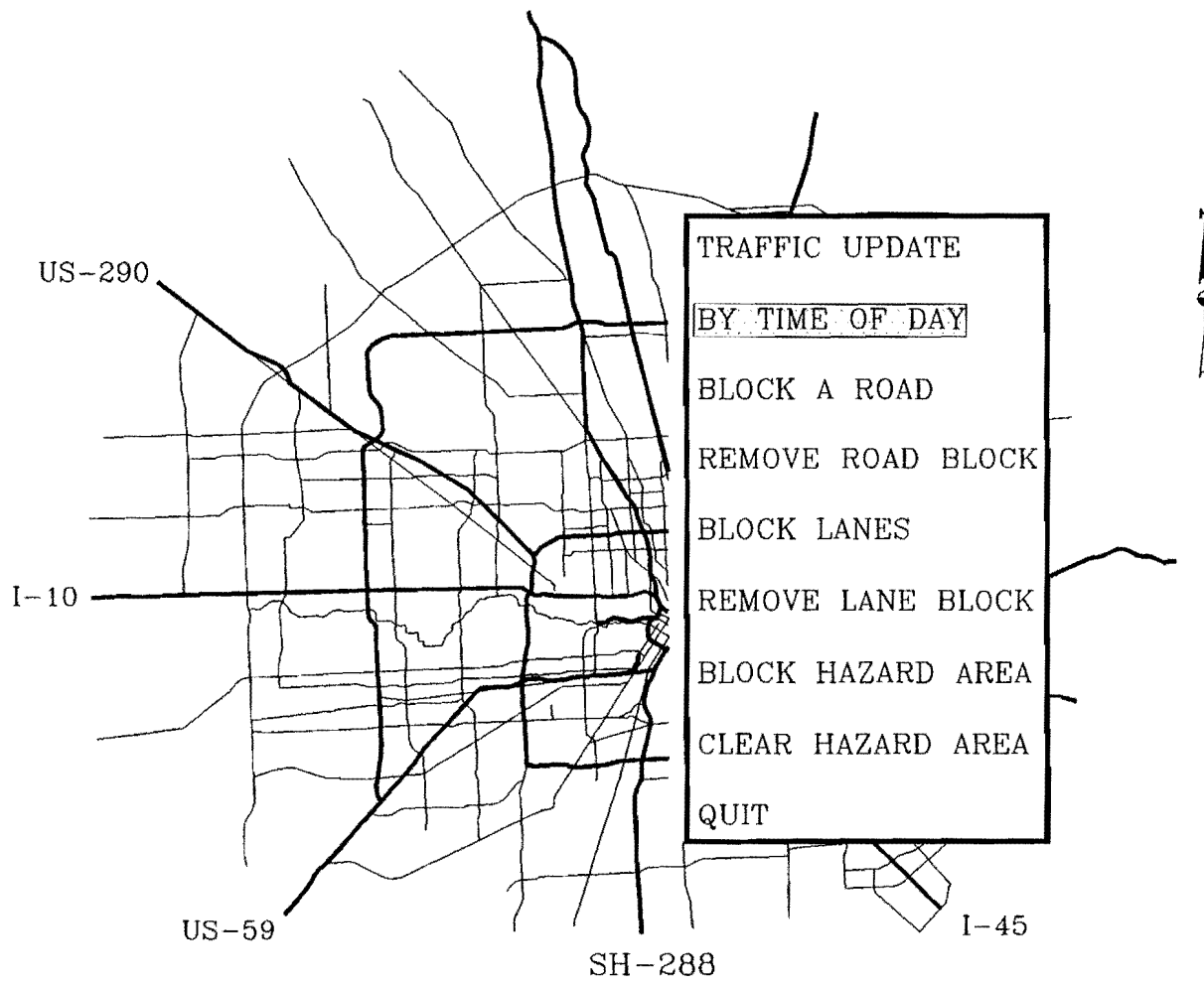


Figure 4.7 Menu for Lane / Area Blocking

*Function/Capability:*

When an area wide incident occurs, the operator can locate the incident on the map and specify a radius for the area affected. The system will automatically simulate the blocking of all roads into or through the area. The user can then perform hazard area alternate routing to generate various alternate routes for motorists. The motorists can then be informed of these alternate routes over HAR, variable message signs, or other means.

This application may be extended to generate a list of all intersections on which physical road blocks may need to be placed by police. A list of intersections to be blocked will help the operator and authorities to quickly take an appropriate action. In addition, a listing of residents in the areas affected could be generated.

Figure 4.8 shows a display of a hazard area.

*Development:*

PC-ARC/INFO<sup>(C)</sup> does not directly support this application. However, an algorithm has been designed for this application which combines features of different modules by sharing and transferring data between the modules. This application uses the TABLES, ALLOCATE, and ROUTE modules. The ROUTE module allows blocking of individual intersections. This application automatically generates a list of affected intersections within a certain radius of the center of a hazard using features of ALLOCATE and TABLES. The center and radius is specified in ALLOCATE module and information on the intersections to be blocked is generated. This information is pre-processed in TABLES and the result of this pre-processing is suitable for input into ROUTE. Processed output is read in ROUTE and the corresponding intersections are blocked automatically. Following this, any least-impedance route, performed in the ROUTE module, will bypass all blocked intersections.

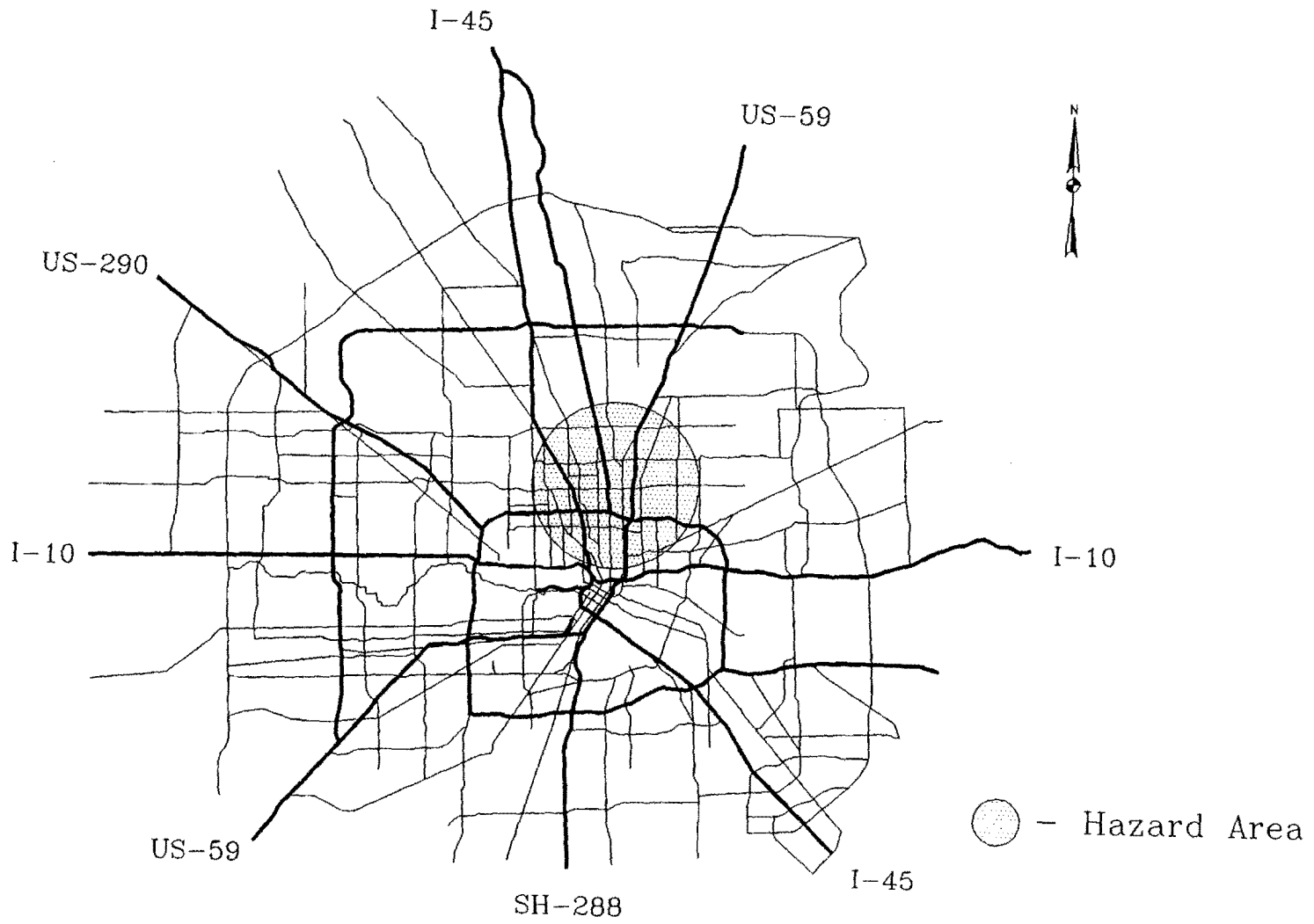


Figure 4.8 An Example of Hazard Areas Needing Evacuation

### 4.1.3. Resource Management Applications

*Purpose:*

When an incident occurs, one of the functions of a traffic management center operator is to communicate with appropriate resources such as police, fire, and hospitals. For these resources, the names, phone numbers, and other relevant data must be available to the operator. The applications for resource management are intended to provide the operator with a graphical directory of resources and a set of geographic databases on the features of the resources.

*Function/Capability:*

All resource management applications, except one, have a base map showing freeways and major arterials of the Houston metropolitan area. There are two kinds of applications that have been developed related to the base map:

- 1) Associated with each arc is data on the resources for that arc. For example, the names and phone numbers of police, fire, hospital, flood pump managers, and park-and-ride resources are associated with each arc. When incidents occur, the operator only needs to select that arc graphically and the names and phone numbers of each resource will be displayed on the screen. Conversely, the resource and its area of operation for each resource may be queried by the operator and displayed on the screen, if needed.
- 2) On the top of this map are superimposed locations of various incident management resources. The user is allowed to conduct queries on these resources. The queries could be spatial, where the user selects a resource location and the database for the resource location is displayed. In addition, queries could be specified in terms of the requirements of the user, and the system will display graphically all locations which meet the criteria specified by the user in his query. A listing of the database of each location will also be displayed. The user can then decide to use appropriate incident management resources. The following major resources are available for graphical queries:

Hospitals: The user may query on data such as address, emergency phone number, whether it has life-flight helicopters, or whether a burns unit is available.

Police: The user may query on data such as address, emergency phone number, contact person, number of police cars on staff, number of accident investigators, or other service characteristics.

Fire Stations: The user may query on data such as address, emergency phone number, contact person, number of fire fighters, or number of fire fighting vehicles.

Flood Zones and Pumps: The user may query on information such as address of appropriate pump location, capacity of pump, emergency phone number, and contact person for repair.

Park and Rides and HOV Lanes: The user may query on information on Park and Ride locations as well as HOVs. A Park and Ride query may be performed on information such as the capacity, the hours of operation, the number of buses allocated, or other service elements. On HOVs, the query may be performed on information such as the hours of operation inbound as well as outbound. This demonstration application is based on a map of the NORTH freeway which shows all Park and Rides and the HOV along the freeway.

Demonstration of a graphical query on hospital resources is shown in Figure 4.9. Figure 4.10 shows a graphical query on HOVs and Park-n-Rides. Figure 4.11 provides a graphical query on flood zones and flood pumps. Figure 4.12 shows a fire station's area of operation, and Figure 4.13 shows the address of the fire station responsible for a specific street address.

Other resources that may be useful for inclusion in either of the above kinds of application, are:

- 1) Zip Code areas;
- 2) Texas reference marker locations;
- 3) Phone areas/districts;

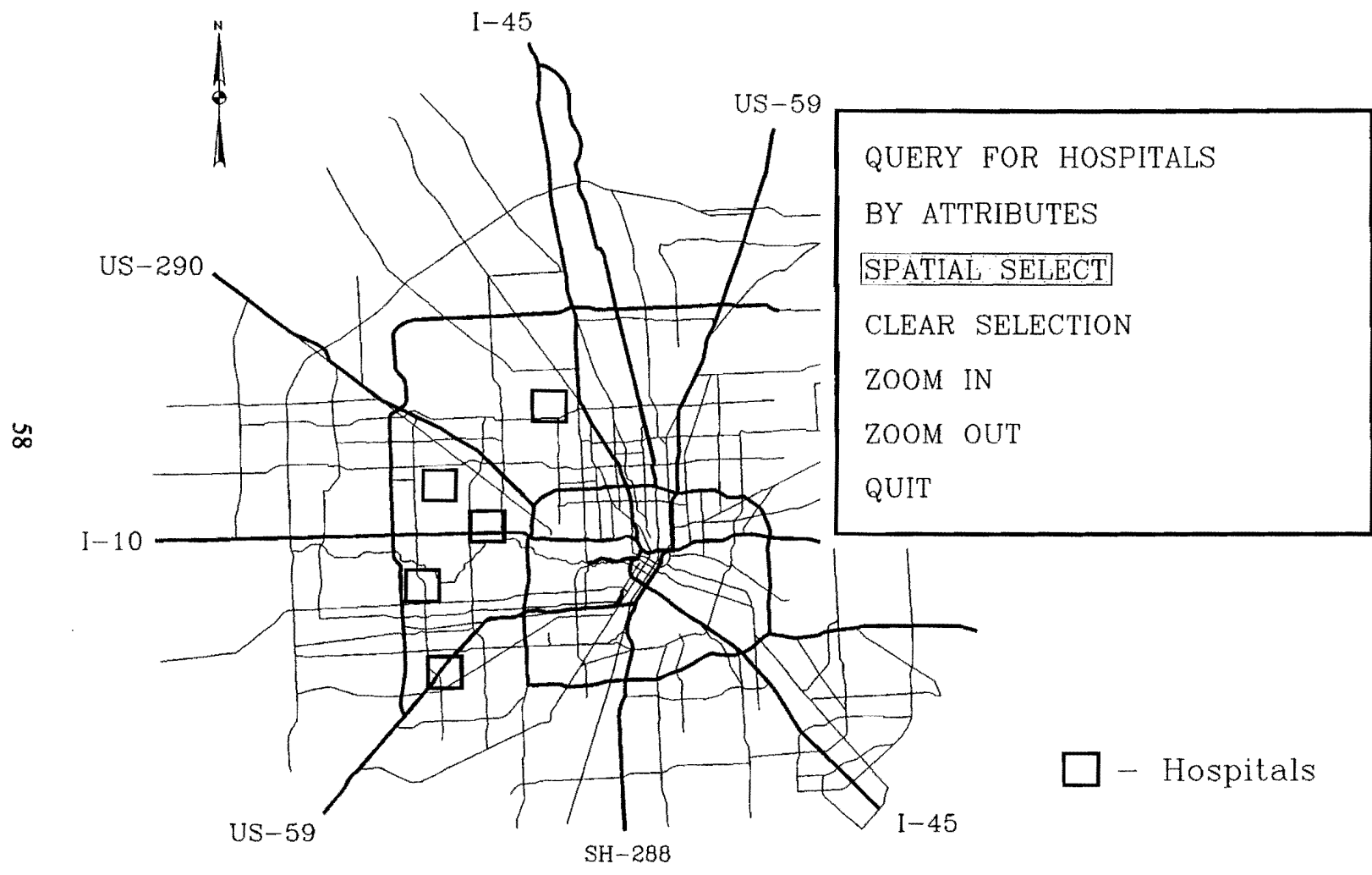


Figure 4.9 Graphical Query System on Hospitals

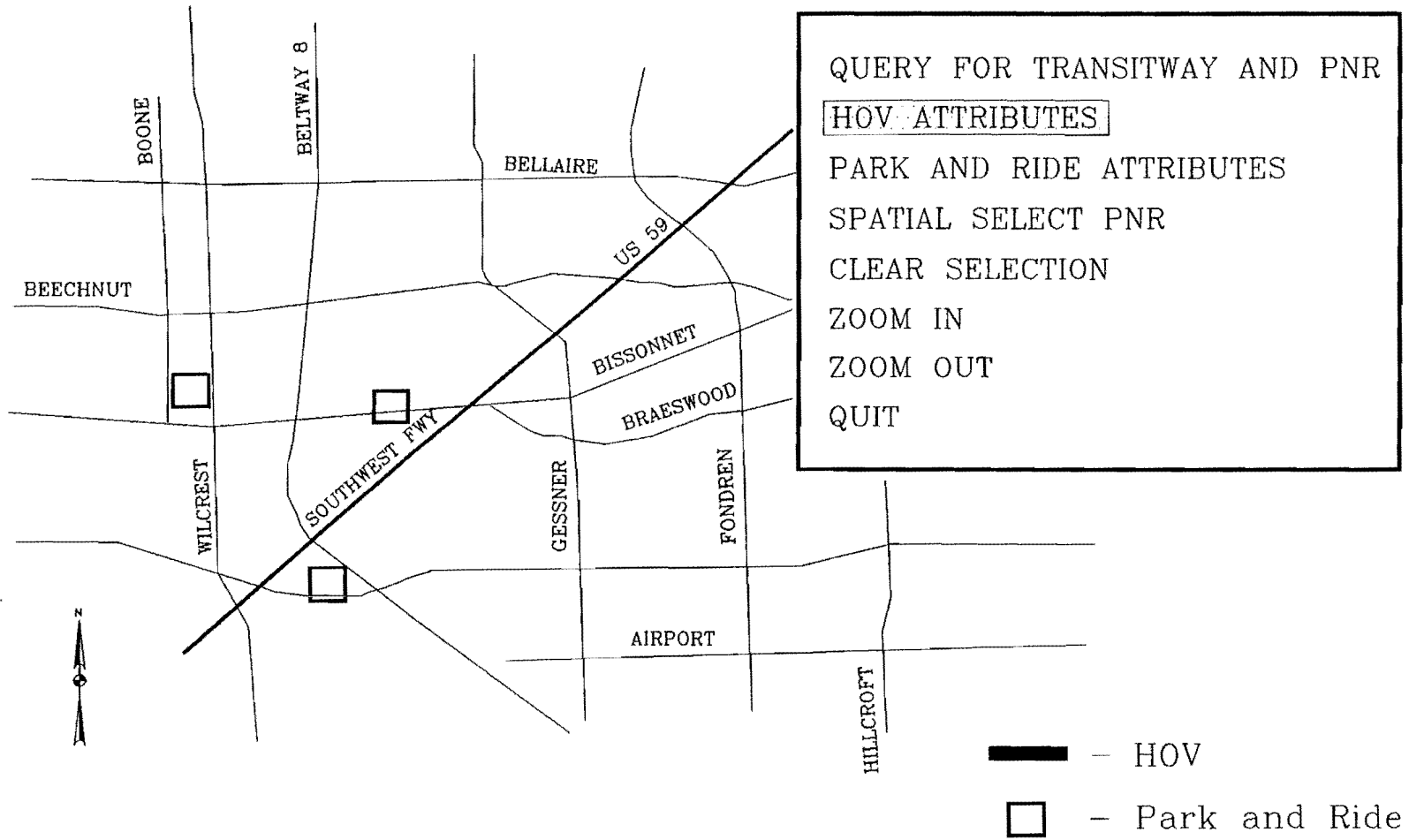


Figure 4.10 Graphical Query on HOV and Park and Rides





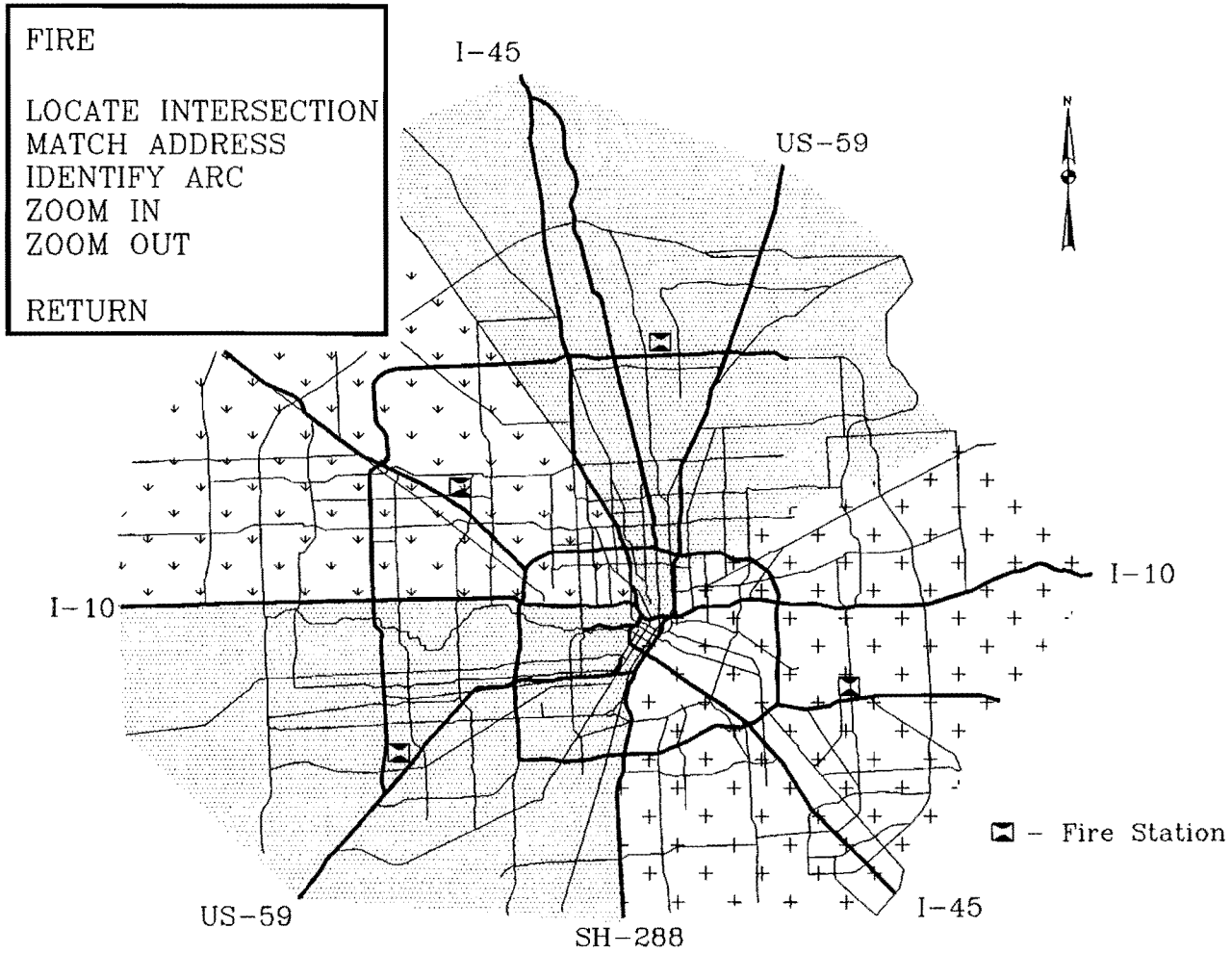


Figure 4.12 Fire Station Operation Zones

FIRE RESOURCE  
STATION #3 333-3333 333 MAIN  
CONTINUE

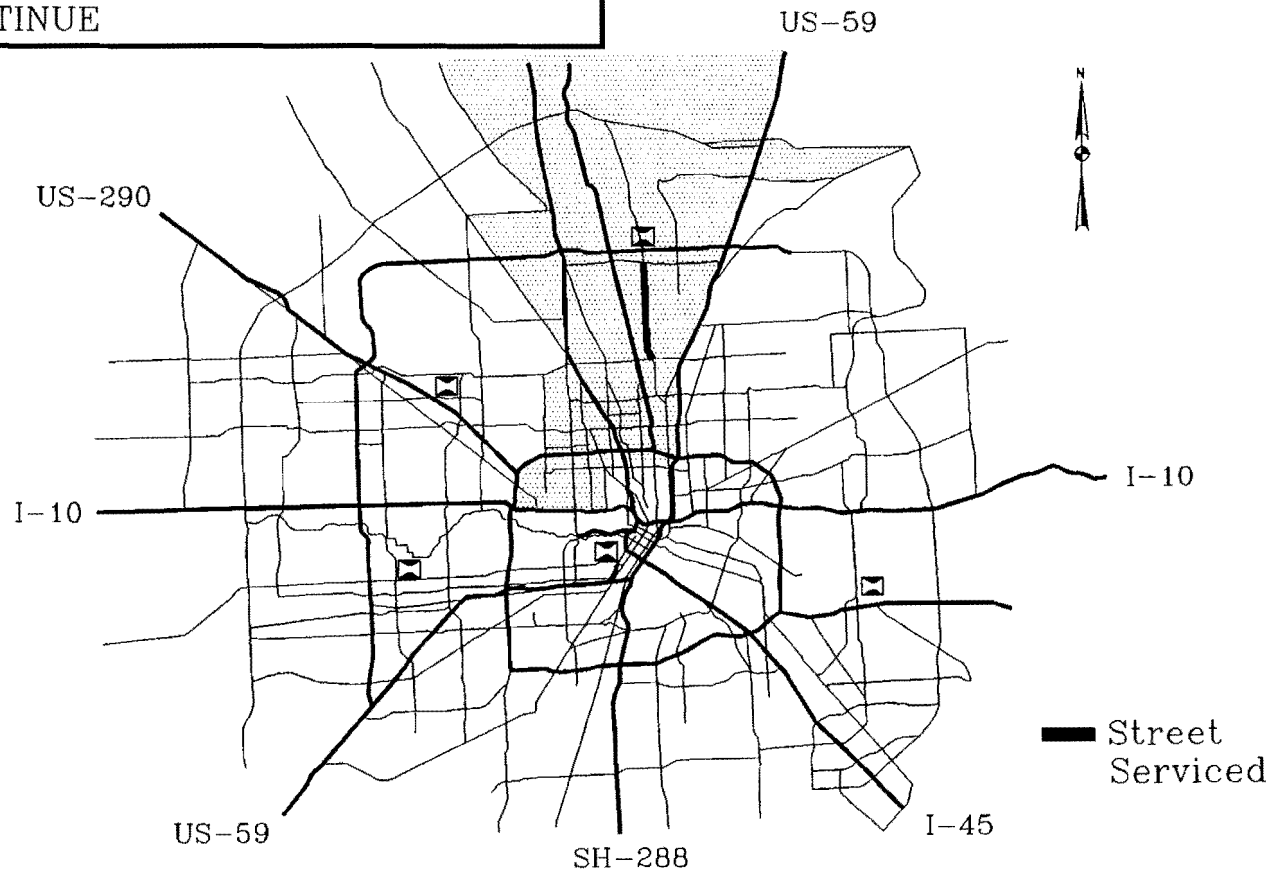


Figure 4.13 Address of Fire Station for Street Serviced

- 4) AVI card reader locations;
- 5) School locations; and
- 6) Variable message sign locations.

*Development:*

Applications which indicate the area of operation of resource, and those in which individual road segments may be queried for resources, operate within the ALLOCATE module. These applications are based on the arc attribute table for the coverage. The applications for query on resource database operate in the ARCEDIT and/or ARC PLOT modules. These are based on point and/or polygon attribute tables for each resource which list the features of the resources.

## **4.2. APPLICATIONS FOR PLANNING AND ANALYSIS**

Engineers and planners need to analyze roads and incidents to devise plans for incident management. While applications for planning and analysis are off-line, these serve a valuable purpose in incident management. These applications support development of incident management operations by pre-incident or post-incident analysis. Two applications have been designed for planning and analysis purposes:

- 1) Incident Database Geographic Query; and
- 2) Roads Database Geographic Query.

### **4.2.1. Incident Database Geographic Query System**

*Purpose:*

An incident database query system is required by planners and engineers to ascertain the time, location, type, and cause of incidents. A geographic query on the incident database allows planners to graphically determine locations which are more prone to incidents and relationship of incidents to such factors as time of day, visibility, and rain. This application is designed to provide the ability to analyze incidents.

*Function/Capability:*

This incident analysis system uses the incident data collected for the Houston Motorist Assistance Program (MAP). For demonstration, the area of operation was restricted to the SOUTHWEST Freeway (US 59).

The system allows the operator to conduct queries to perform the following:

- 1) Graphically display month by month, incidents at each location;
- 2) Graphically display cumulative incident data for each location; and,
- 3) List the incidents reported for each month on the screen.

Such analysis helps the operator identify sections of freeway where assistance has been predominant. This system could be extended to employ traditional accident analysis techniques such as correlating accident frequency/severity to operational and physical roadway characteristics.

A typical screen for the incident database geographic query system is shown in Figure 4.14. Figure 4.15 shows graphically the results of a monthly query and Figure 4.16 shows the results of a locational query in text form.

*Development:*

In this application, the map of the system is annotated with street names to aid in the analysis and has a colored legend for the ranges in number of incidents. The freeways are highlighted and other features include street identification and address location.

The application runs within the ARCPLOT module. To list the incidents for a specific location and month, an external MAP database needs to be accessed. Though ARCPLOT can access the attribute tables, it does not support external databases. To access the database of incidents for streets and display incident data (graphics and text), ARCPLOT is tricked into believing that the MAP data file is actually the AAT of congestion management coverage corresponding to the month and year. For this purpose, a "pseudo-coverage" is created for each month of data and the MAP database for that month is renamed as the AAT in the directory representing the month and year. Thereafter, a selection is performed on this "pseudo attribute table" using the name of the desired street intersection. The relevant data is then written to a text file for display.

The application can be extended easily to analyze the occurrence of major incidents with respect to time and location. TRASER software could possibly be used in place of the MAP database for this purpose. Additional queries may be created to account for factors such as time of the day, rain, and visibility.

#### **4.2.2. Roads Database Geographic Query System**

*Purpose:*

The roads database query system has been designed, for analysts and planners, to be used in conjunction with the incident database query system. Using these two systems, the engineers and planners can relate the incidents to various conditions on the roads such as the number of lanes, shoulders, pavement conditions, daily volume, and other characteristics.

*Function/Capability:*

Analysts can perform a variety of queries to analyze the road network and identify potentially hazardous road segments. The road segments which satisfy the criteria are graphically displayed and are also listed on a screen. The results of the queries are shown graphically on the screen if the query is text based. On the other hand, a freeway segment may be selected graphically and its database may be viewed by the user.

Examples of the spatial or graphical queries that can be performed are as follows:

- 1) Number of lanes on roads;
- 2) Daily volume of traffic;
- 3) Number of annual accidents;
- 4) Surface conditions;
- 5) Road width; and,
- 6) Last resurface.

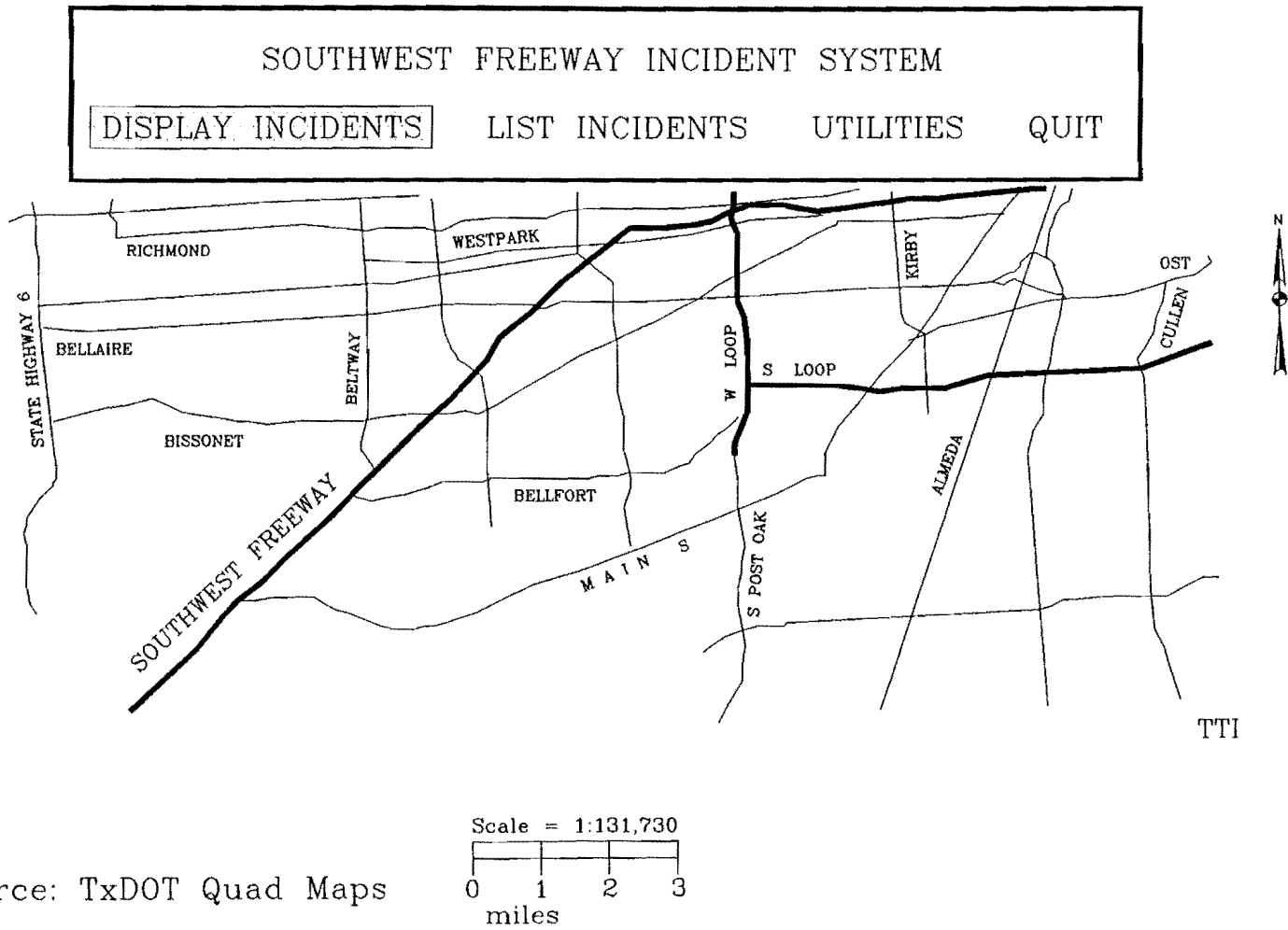


Figure 4.14 Incident Database Geographic Query System

SOUTHWEST FREEWAY INCIDENT SYSTEM  
 DEC. 1992  
 INCIDENT : 240  
 (Bi-directional)

67

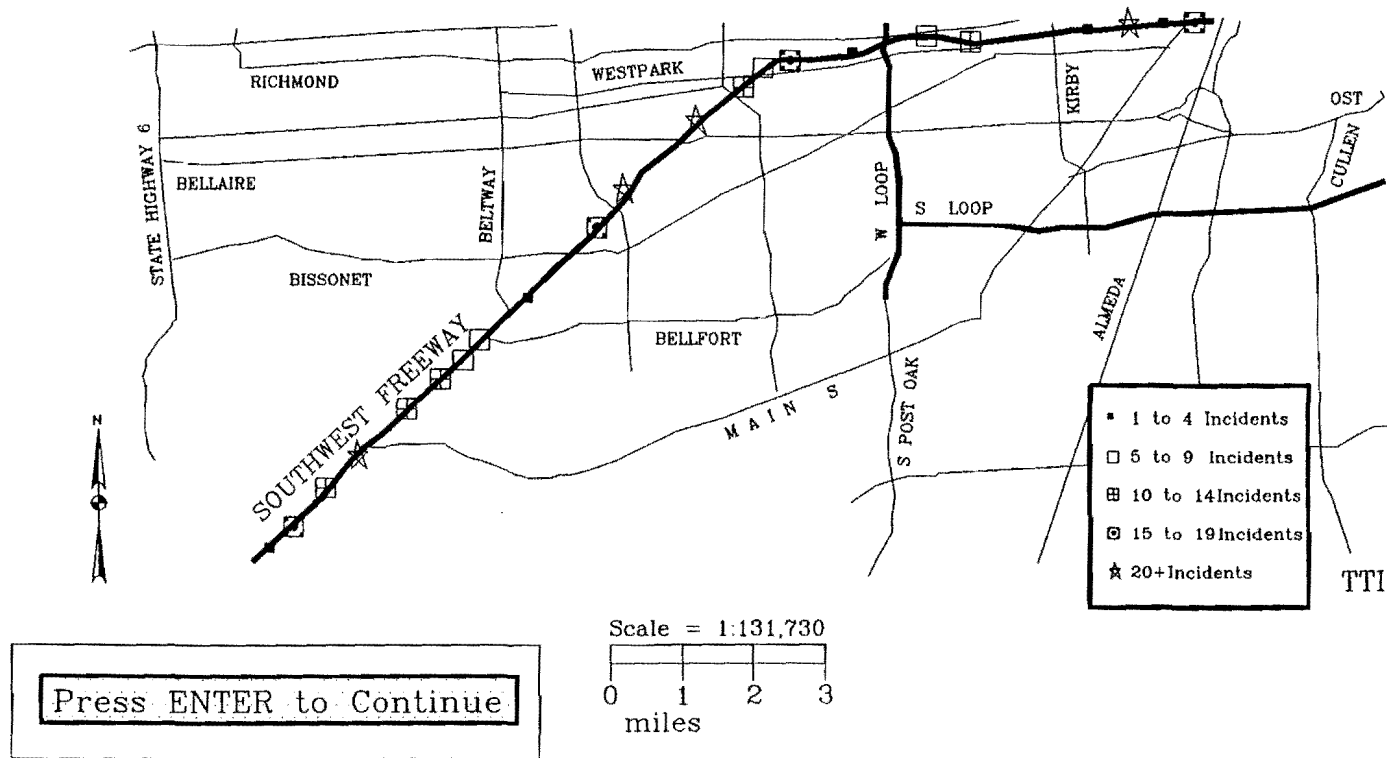


Figure 4.15 Results of Monthly Query on Incident System

### SOUTHWEST FREEWAY INCIDENT SYSTEM

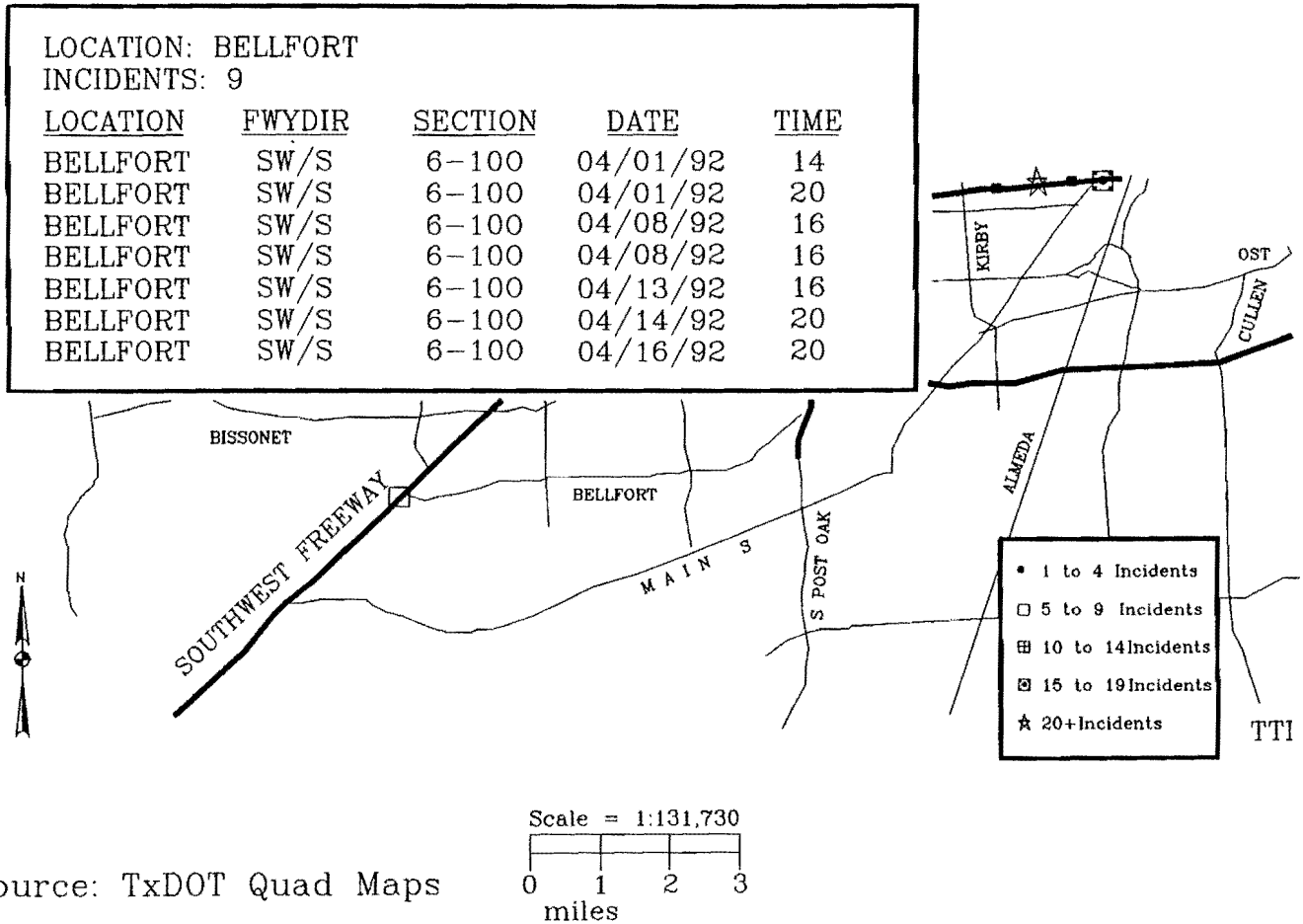


Figure 4.16 Locational Query on Incident System



A number of other queries are possible using the roadway inventory. However, for simplicity not all queries have been included in this demonstration system. Figure 4.17 shows the screen, with menu, of the roads database geographic query system.

For example, a text based query could be, "Show all sections of freeways with daily volume greater than 175,000 vehicle per day and the number of lanes less than 8 inches (20.32 cm). The response would be shown graphically on the screen. On the other hand a query such as "How many lanes are available in the outbound direction of North Freeway at Tidwell" can be performed graphically by selecting that section of North Freeway and the result is viewed by the user in text form.

*Development:*

This application runs in the ARCPLOT module. The entire congestion management coverage is used for this application. The programs call commands in the ARCPLOT module to perform queries on the roadway inventory attributes of the congestion management arc attribute table.

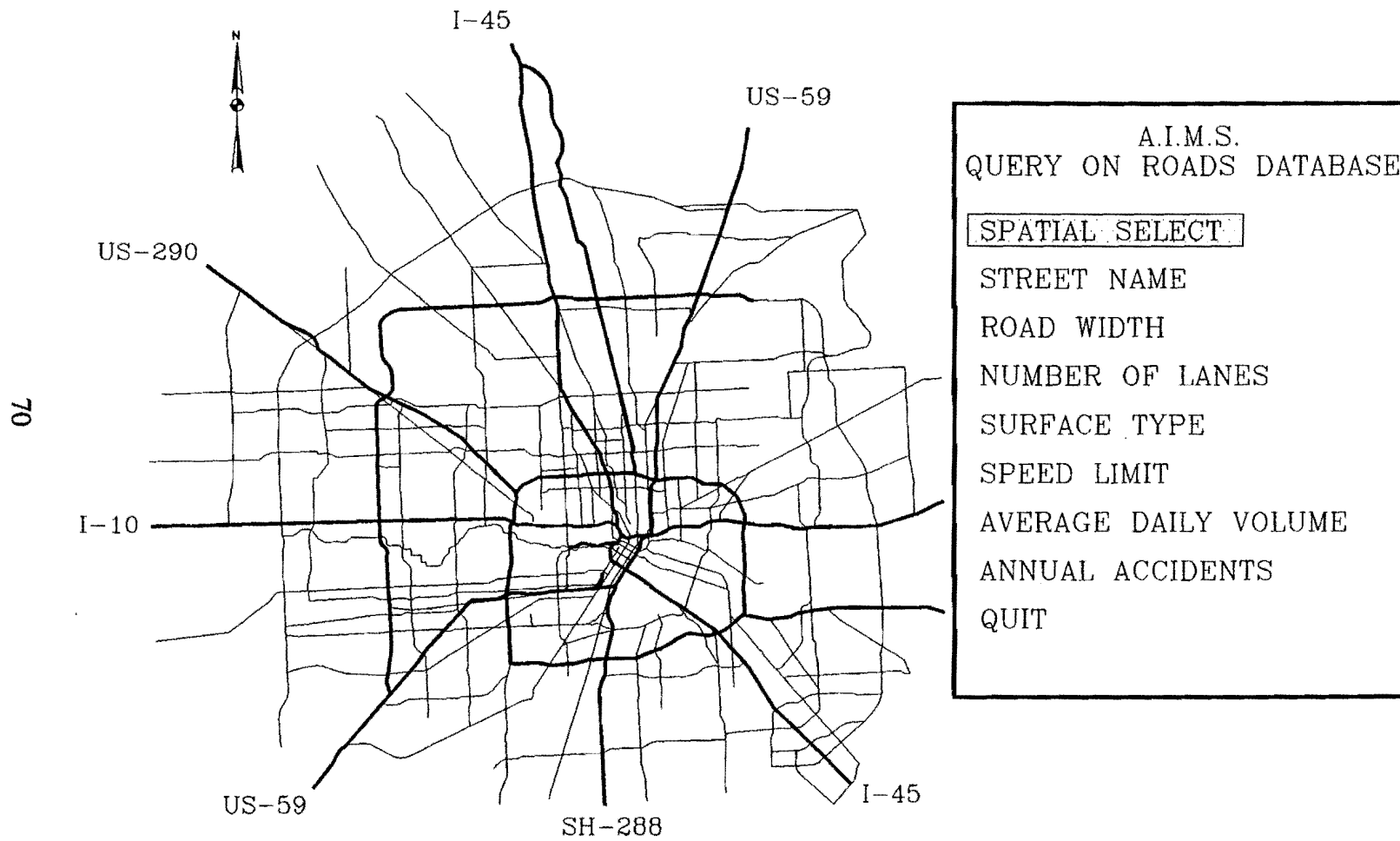


Figure 4.17 Roads Database Geographic Query System

### 4.3. BENEFITS OF APPLICATIONS

Development of the demonstration/prototype applications has demonstrated that commercially available software may be used for an automated incident management plan. In particular, a GIS development tool such as PC-ARC/INFO<sup>(C)</sup> could be an appropriate platform for such a plan. The plan may include incident management applications developed on such a platform as well as other stand-alone applications which may interchange data with these applications. Such an automated incident management plan could serve two purposes.

One, real-time incident management may be performed through a combination of one or more of:

- 1) Dynamic route guidance through alternate routing, travel time simulation, and variable message signs;
- 2) Incident response through fleet management of incident assistance vehicles such as police, fire, ambulance, MAP as well as blocking of lanes and hazard areas; and,
- 3) A resource database which provides a list of contacts, address, and phone numbers as well as other relevant information for various kinds of incident management resources.

Two, planning for incident management could be performed through a combination of one or more of:

- 1) Geographic queries and results on roadway inventory/database; and
- 2) Geographic analysis of incidents, locations, and time. This will help in identifying the improvements required on the roadways to reduce the likelihood of the occurrence of incidents.

APPENDIX D includes the SML code for the applications developed.



## **5. ISSUES AND LIMITATIONS IN GIS DEVELOPMENT USING PC-ARC/INFO<sup>(C)</sup>**

### **5.1. SLOW DATA MANIPULATION**

Data manipulation in PC-ARC/INFO<sup>(C)</sup> is rather slow when compared with other commercially available relational databases. Although PC-ARC/INFO<sup>(C)</sup> features a relational database management, its data manipulation performance is slow including updates to data, relational operations, and changes in structure. For example, adding an attribute to the arc attribute table which is a dBASE<sup>(C)</sup> file, could take up to several times longer than it would take if the same attribute was added using dBASE<sup>(C)</sup>. As a result, this application takes longer to develop and run.

A simple update of one record takes considerable time as PC-ARC/INFO<sup>(C)</sup> rewrites the entire file. For large databases, this could take several seconds or even minutes. Moreover, most PC-ARC/INFO<sup>(C)</sup> applications developed for transportation involve entering and exiting a number of modules which means repeated loading of GIS database and the need for even longer run time.

### **5.2. LIMITATIONS FOR REAL-TIME CONGESTION MANAGEMENT**

PC-ARC/INFO<sup>(C)</sup> applications do not appear to be suitable for real-time traffic management systems such as automatic vehicle identification based traffic monitoring where new data is received at short intervals of a few seconds.

The automated incident management plan does not envisage real-time incident detection. Such detection could be based on the analysis of volume and occupancy data received every few seconds from the detectors. However, PC-ARC/INFO<sup>(C)</sup> is not able to handle frequent updates to large databases within the time constraints for new data from the detectors.

The automated incident management plan does not envisage real-time arterial signal control. While applications for automated incident management plan can be modified to update input data files for PASSER<sup>(C)</sup> to compute new signal timings, these can not be accomplished within the time constraints of real-time control. However, a separate study to investigate the feasibility of extending PASSER-II<sup>(C)</sup> or PASSER-III<sup>(C)</sup> for real-time signal control may be useful.

Nevertheless, PC-ARC/INFO<sup>(C)</sup> may be suitable for applications where new information is received in intervals of several minutes, such as incidents. For example, for the entire Houston metropolitan area, an average of one incident, such as an accident or stall, is reported an average of every 5 to 10 minutes.

### 5.3. MAINTAINING CONSISTENT ORIENTATION OF ARCS

To perform origin/destination routing, it is important that each arc, representing a street segment, be assigned an appropriate orientation. Each arc on the map on the computer screen represents both travel directions unless it has been specifically restricted to be one way. Each arc has a *from node* where it begins and a *to node* where it ends. As a result, an arc has one impedance for the *from-to-to* direction and another for the *to-to-from* direction. Only one of the two impedances is used to find the route with least impedance. For example, if an outbound route from downtown to suburbs, is required all the arcs on the routes should have the same orientation, i.e., either all arcs be oriented towards downtown or away from downtown. An attempt was made to orient the arcs consistently; with most arcs oriented to be outbound. However, on roads which run in a loop around downtown or those which do not lead to downtown, it has been difficult to maintain consistency; there are cases where two adjacent arcs on one road have opposite orientation. This is an issue that needs to be dealt with in the future development of an implementation of the automated incident management plan.

#### **5.4. LACK OF STANDARD SYNTAX ACROSS MODULES**

PC-ARC/INFO<sup>(C)</sup> has been implemented in modules and several functions are common to these modules. However, for each module, the functions are not identical. In some cases, there are differences in syntax; in some other cases, there is a difference in function. In other cases, the names of the programs that provide the same function are different in different modules. For example, MAPEX command allows "zoom out" in ARCEDIT module but not in ROUTE module. This lack of standardization implies that PC-ARC/INFO<sup>(C)</sup> requires more memory on both the hard disk and in the main memory. In addition, it worsens the learning curve for a new user. Although menu based applications hide the syntax from the user, it is a drawback in the development of applications.

#### **5.5. LACK OF DYNAMIC SEGMENTATION**

For linear measures, such as road networks, the route-measure methodology is often used to represent associated data. In this methodology a position is located based on a known point and a distance along a linear object. Data such as pavement quality, incidents, and speed zones are often maintained using the route-measure method. For example, a route together with a mile-post on it uniquely identify a location without needing a coordinate representation scheme; in addition, attributes can be associated using this referencing mechanism.

Geographic Information Systems (GIS), such as ARC/INFO<sup>(C)</sup>, often feature the arc-node topological method to record spatial data. This topology restricts attributes to either 1) nodes between two or more arcs, or 2) arcs between two nodes. However, this may not be compatible with the route-measure format used in external databases. Dynamic segmentation provides this compatibility.

Dynamic segmentation refers to mechanisms that provide a method for representing and analyzing linear features. It has tools to link multiple databases in route-measure format to existing linear features and to store, display, query, and analyze the attributes independently of the beginning and ending of arcs. In other words, existing arcs or linear

features in a GIS need not be broken with pseudo-nodes when such analysis is performed; it is this feature that makes dynamic segmentation a powerful analytical tool.

For example, a GIS with dynamic segmentation can display road segments based on queries such as "Show all of Hwy 59 which is paved in concrete, has not been repaired in the last 5 years, and does not have sufficient lighting." Such a query coupled with another query "Show all accident locations on Hwy 59" can help the planners relate accidents to condition of roads. As another example, consider the query "Show all of IH-45 with roadway width less than 48 feet and daily traffic volume greater than 200,000 and where the average travel speed is less than 20 mph between 7 AM and 8 AM." This query could help relate congestion with surface width and traffic volume.

Without dynamic segmentation, DOTs often would need to maintain databases of small road segments in their GIS. This would make the databases very large, and referencing and maintaining such a database would be costly. For the example queries, in a traditional GIS, DOTs may need to split even a small stretch of the highway, say 5 miles (8 km), into several hundred small arcs so that it conforms to the pavement data. On the other hand, in a GIS with dynamic segmentation, the pavement data can be related to the arc-node topology without the need to split the highway representation into small arcs.

Dynamic segmentation has the following features:

- 1) Ability to define routes within a line coverage; a route is a linear feature to which attributes are attached and may comprise one or more arcs, each of which may either be completely a part of the route or be partially a part of the route, i.e., the start and end points of a route need not coincide with the start and end points (nodes) of arcs.
- 2) Ability to define sections, each of which represents all or part of an arc and has a starting point and an end point based on its position within a route. User defined attributes may be associated with a section.
- 3) The defined routes have linear measures such as mile-posts associated with them and attributes may be attached to these routes using the measures.
- 4) The attributes attached to the routes may belong to multiple data sets, i.e., many different measures may be used along a route. For example, there



could be different route-measures for pavement type, lighting, repair records, overpasses or underpasses, number of lanes, etc. The attributes attached to these route-measures may not be a part of the GIS data model and may be stored in a relational database system. In other words, the attributes in each different route-measures may be maintained separately in databases corresponding to these route-measures and only linked to GIS when needed.

Dynamic segmentation is useful for applications such as geocoding or referencing, pavement management, incident analysis, shortest path routing, and other linear data needs. Several DOTs including Wisconsin, Missouri, Florida, North Carolina, Saskatchewan (Canada), and Arizona are using dynamic segmentation for such applications [13].

PC-ARC/INFO<sup>(C)</sup> does not support dynamic segmentation, although the UNIX workstation version does support dynamic segmentation. The lack of dynamic segmentation has been particularly noticed in the travel time simulation system. In PC-ARC/INFO<sup>(C)</sup> a user may only perform travel time simulation on a route between any two nodes. To better model the effect of incidents in the travel time simulation system, a need has been identified for dynamic segmentation.

## 5.6. SML<sup>(C)</sup> LIMITATIONS

PC-ARC/INFO<sup>(C)</sup> uses the Simple Macro Language (SML<sup>(C)</sup>) to create GIS applications. It does not have the capability to develop complex applications; however, it was effectively used to examine the capabilities of GIS for the automated incident management plan.

The SML<sup>(C)</sup> provides a set of tools to the applications designer to perform a variety of tasks using PC-ARC/INFO<sup>(C)</sup> commands, such as:

- 1) building user interfaces;
- 2) building macros for frequently used commands; and
- 3) performing geoprocessing operations and data manipulation.

The SML<sup>(C)</sup> is rather limited in its scope and function in the following areas:

- 1) Variable manipulation is possible, but the variables are not named; the variables are limited to 50 and local variables are not supported.
- 2) Branching is possible, but there is little support for modular and structured programming.
- 3) Variable passing is not possible. All variables are global.
- 4) Looping, whether conditional or unconditional, is not supported, but an application designer may simulate the same.
- 5) There is no support for advanced data structures such as arrays and linked lists.
- 6) The user interface capabilities are also limited. Menus and forms can be created, but graphical interfaces are not possible. Even pull-down menus are not possible; these have to be mimicked.
- 7) There is no support for programming advances like data abstraction, encapsulation and code-reusability.
- 8) SML<sup>(C)</sup> is an interpreted language where source program statements are translated into machine statements and executed one by one after translation of each statement.
  - a) This requires memory overhead as well as reduces execution speed of the programs.
  - b) Source programs are required for the application and are, thus, not secure.

SML<sup>(C)</sup> is, thus, more similar to a lower level language than to traditional programming languages. In summary, custom applications and interfaces can be built, but SML<sup>(C)</sup> is not well suited for creating complex applications.

## **5.7. PC VERSUS WORKSTATION**

A limitation in GIS development, and a very important one, is that GIS applications are typically very intensive in their need for computer system resources. GIS maps/images

and databases are large and require substantial memory, graphics, and storage capabilities. Often, the applications require frequent fixed disk access and data transfer.

In this respect, the PC-ARC/INFO<sup>(C)</sup> is rather limited. Largely, the limitations are due to the architecture of PCs. However, the limitations are worsened by the design of PC-ARC/INFO<sup>(C)</sup>.

First, DOS does not have the capability to directly address main memory larger than 640Kb. PC-ARC/INFO<sup>(C)</sup> has been designed to work within this 640Kb of memory and does not have any memory management capabilities of its own. PC-ARC/INFO<sup>(C)</sup> needs a minimum of 490Kb of memory to run. The individual modules need an extra 35Kb to 60Kb each of memory for the graphics device drivers to work with PC-ARC/INFO<sup>(C)</sup>. Such memory requirement leaves little room for other devices and memory conflicts are not uncommon. A disk caching device driver is often required because GIS applications are disk intensive. This driver takes additional memory. If, in addition, a disk-doubler is used to meet the requirements of fixed disk space, memory becomes a major constraint. Even with expanded and extended memory managers, which PC-ARC/INFO<sup>(C)</sup> does not support, memory contention remains an issue. In such cases, the memory managers allow the use of reserved or upper, extended and expanded memory, where some drivers may be loaded into these memory areas. Also, larger memory, if installed, becomes available for disk caching and some operations are speeded up. However, there still is a very high requirement for disk read/writes, causing applications to remain slow.

Second, DOS does not have multi-tasking architecture, this means that only one PC-ARC/INFO<sup>(C)</sup> module may run at a time. Often an application that is being developed for use in the ROUTE module needs to be tested in that module, but has to be updated in the ARCEDIT or TABLES modules. This requires frequent entry and exit into each module as well as repeated loading and unloading of the GIS applications. This results in time overhead for development of applications.

Third, graphics capabilities in the PCs are limited compared to UNIX workstations. This also affects the speed at which PC-ARC/INFO<sup>(C)</sup> handles the graphical displays. In recent times, some graphics accelerators have been introduced for PCs. These should improve the response times.

Fourth, PC-ARC/INFO<sup>(C)</sup> has a modular design requiring frequent entry and exit into modules. Moreover, the size of maps that could be handled by each module may vary. For example, the number of arcs in the ROUTE module is limited to 4300 which implies that it cannot handle networks larger than one of medium complexity, whereas other modules can support over 20,000 arcs.

Fifth, PC-ARC/INFO<sup>(C)</sup> has a limited user interface environment. For example, pull-down menus are not supported, though these can be emulated. There is no support for graphical user interfaces. The look and feel of applications is not sophisticated.

Sixth, SML<sup>(C)</sup> the application programs language for PC-ARC/INFO<sup>(C)</sup>, is limited. Because of this, developing applications is very time consuming. The application programs are not structured and self documenting and SML<sup>(C)</sup> has a limited set of commands and features. Developing applications and debugging errors in these programs is very time consuming.

In contrast, ARC/INFO<sup>(C)</sup> on workstation is a far better tool for developing GIS applications. The UNIX workstations do not suffer from the limitations of DOS. An application may address as much memory as is installed on the system. The environment is multi-tasking. Graphics accelerators are available. Fixed disk drives are much faster. These enhancements in the computing environment improve development time as well as the speed of execution of applications. In addition, ARC/INFO<sup>(C)</sup> has a much richer set of commands. The ARCPLOT and ROUTE/ALLOCATE modules are integrated into one module. PC-ARC/INFO<sup>(C)</sup> works in the X-windows environment and supports, to some extent, development of graphical user interfaces through AML<sup>(C)</sup> which is similar to traditional high level programming languages and has a significantly higher level of features and functions. A description of AML<sup>(C)</sup> for the purpose of comparison with SML<sup>(C)</sup> is given below:

The ARC Macro Language (AML<sup>(C)</sup>) is a programming language for developing GIS applications in the ARC/INFO<sup>(C)</sup> environment on mainframes, minis, and UNIX workstations. It provides a set of tools to the user to perform a variety of tasks such as building user interfaces, specifying user environment, building macros for frequently used commands, and even for performing geoprocessing operations and

data manipulation. The AML<sup>(C)</sup> is similar, in principle, to traditional high level programming languages and provides similar functions, such as branching, variable manipulation, and argument passing. The language is standardized across a variety of hardware platforms and provides for portability of ARC/INFO<sup>(C)</sup> applications. Graphical user interfaces and custom applications could be built with AML<sup>(C)</sup>.

To compare the performance of PC-ARC/INFO with that of workstation ARC/INFO, a set of macros and GIS databases were developed on the PCs. These applications were then ported to the workstation environment. The execution times for these applications were clocked with a stop watch on a Sun UNIX workstation and a PC/386. The same applications executed on an average over 25 times faster on the Sun workstation. It is estimated that on a "high end" 486 workstation, the ratio of Sun to PC would be reduced somewhat. However, the difference would still be substantial. A benchmarking report on the above tests and on ARC/INFO<sup>(C)</sup> for various workstations as well as workstation versus personal computers is attached in APPENDIX E. While the report is about 2 years old, the methodology used for benchmarking is considered applicable to future purchases of workstations for GIS development.

It has been learned from ESRI that further development of PC-ARC/INFO<sup>(C)</sup> is being terminated. Instead, workstation ARC/INFO<sup>(C)</sup> will be ported to a Windows-NT operating environment. This operating environment will run on both PCs as well as workstations. Thus, there will be a single product with nearly uniform features across PCs and workstations. As PC-ARC/INFO<sup>(C)</sup> limitations will be eliminated, it is expected that the development time of applications as well as execution time of applications will be considerably improved on this new operating environment. Computer technology has also evolved during the course of this project. New high performance processors are available for both PCs and workstations. DEC's Alpha, Motorola's PowerPC, and Intel's Pentium high performance microprocessors have become available. Windows-NT will be available for each of these processors. The distinction between PC-ARC/INFO<sup>(C)</sup> on personal computers and ARC/INFO<sup>(C)</sup> on workstations will be somewhat blurred, though execution times on each of these will still be different. Nevertheless, the development effort conducted will be preserved.

## **5.8. DELETION OF PSEUDO-NODES AND COMBINATION OF DATA**

This problem occurs during development of an application when a pseudo or unnecessary node is to be deleted. A pseudo node is one where only two arcs meet and where there is no intersecting arc. The problem is in combining the database. By default, PC-ARC/INFO<sup>(C)</sup> deletes the database of one arc. The arc that is deleted is the one which has a lower record number. This is an arbitrary mechanism and is a deficiency in PC-ARC/INFO<sup>(C)</sup>. Upon combination of data of the two arcs, all length oriented PC-ARC/INFO<sup>(C)</sup> default data items are adjusted to reflect the change in size. Other items may need changes by the user depending on the error caused by the combination of data. For example, the number of lanes may be the same for both original arcs and will not be affected. But, the number of street lights on the original arc should be the sum of the number of street lights on the two new arcs; the user must update the database accordingly. This issue must be handled with care during development of a GIS application for traffic management.

## **5.9. DISPLAY OF MESSAGES ON SCREEN**

During the course of incident management applications, several modules may be entered or exited depending on the need of the applications. However, every time a module is entered or exited, messages are displayed on the screen. For example, on entering a module, a message indicating the product name and the ESRI logo is displayed on the screen. Another example is that every time a selection is performed on the database, a 'records selected' message appears on the screen. These messages do not affect the applications themselves, but the look and feel of the applications is affected; the applications do not appear seamless and the user could become confused by messages which pertain to internal operations and are not relevant for the application itself. These messages are useful to some extent for developing applications. However, once applications have been developed there is no way to turn these messages off.

## **5.10. DATA AVAILABILITY AND COLLECTION**

A major requirement of any GIS is the availability of data. For example, in the applications developed for the automated incident management plan, each record comprised approximately 1,200 bytes of data. There were over 800 street segments on the map without including minor arterials, frontage roads, ramps, and other minor streets. This meant over 800 kilobytes of data. A comprehensive GIS will require thousands of street segments. For example, TIGER files of the census bureau have over 135,000 street segments representing Harris county. Before a GIS is developed, a decision needs to be made on the streets to be represented. For example, with 5,000 records there will be 6 megabytes of data which would not only be difficult to obtain and/or collect but also will make the application slow. Worse, in the event of a lack of data for some street segments, the GIS may not handle incident management function accurately.





## 6. CONCLUSIONS

### 6.1. SUMMARY OF DEVELOPMENT EFFORTS

Feasibility of an automated incident management has been investigated using commercially available software. In particular, commercial software based on geographic information systems technology has been investigated as a development platform for an automated incident management plan.

Commercial software reduces application development time with its in-built tools for development of a specific class of applications. The research approach has been the development of demonstration/prototype incident management applications using the development platform and other commercial software on a personal computer. PC-ARC/INFO<sup>(C)</sup> has been used as the development platform. It provides support for a mapping system with an attached database management system and tools to perform various tasks on this map/database.

A GIS development platform provides a means to import/export existing Computer Aided Drafting Design maps with an automated means to develop an underlying topological (arc/node) database structure. This topology can subsequently be filled in with the databases needed for incident management. The topology developed provides connectivity or intelligence to a map. A GIS development package provides several programs and commands (functions) to assist with developing incident management applications including system calls to integrate other software/programs. The functionality assimilated into the automated incident management plan consisted of spatial data analysis, geographic data management, and network analysis.

A number of demonstration/prototype applications have been developed. These may be classified into one of the following two types. The first are those that address real-time incident management issues. The second are those that address incident management planning issues.

Incident management issues are addressed by the following:

- *Dynamic Route Guidance Applications:* Depending on the need, alternate route guidance could be performed over (a) a local area using frontage roads for incident affected freeways, (b) a wide area using corridor arterials, and/or (c) hazard area routing. In addition, a travel time simulation could be used to estimate delays on major freeways for providing dynamic route guidance to motorists using a superposition of real incidents on historical data.
- *Incident Response Applications:* Incident response could be provided using one or more of (a) simulation of blocking of individual freeway lanes for the purpose of alternate routing, or (b) simulated automated blocking of a hazard area, such as an area affected by a chemical spill, by generating a list of streets to be blocked and providing alternate routing around the area, or (c) automated tracking of incidents and dispatching one or more available vehicles from an incident response fleet, such as vans of the motorist assistance program in Harris County, Texas.
- *Resource Management Applications:* Some of the resources for incident management operations are police, fire, hospitals, flood pumps, etc. These resources could be managed by (a) providing spatial queries to obtain names, addresses, and phone numbers of organizations/individuals to contact when required, and (b) providing spatial queries on the features, capacities, and availabilities of resources.

Issues of planning and analysis for incident management are addressed by the following:

- *Roads Database Geographic Queries:* The roads database query system will permit spatial queries on the roadway inventory database as well as graphical results of queries on the database.
- *Incident Database Geographic Queries:* The incident database query system will provide the ability to perform spatial queries on the number and types of incidents on any section of a freeway.

Additionally, these applications can be enhanced by integrating these with other commercial software such as PASSER-II<sup>(C)</sup> and CAMEO<sup>(C)</sup> which may share data with these applications.

## **6.2. RECOMMENDATIONS FOR DEVELOPING AN AUTOMATED INCIDENT MANAGEMENT PLAN USING GIS**

As a result of this study, it was determined that careful consideration must be given to the selection of a GIS development environment and platform. A recommended development approach would include:

- 1) A windowed multi-tasking computing environment (workstation).
- 2) A GIS development platform with support for:
  - (a) A programming language that parallels a higher level programming language;
  - (b) Adequate modelling of road networks for incident management;
  - (c) Functionality for map editing which is similar to a CAD system;
  - (d) Data conversion from a variety of data format;
  - (e) User selectable relational database management system; and
  - (f) Truly integrated modularity and uniformity of design.

Recommendations to develop an automated incident management plan can be made based on the assumption that a GIS development approach is acknowledged to be feasible. Given this assumption, it is recommended that future development of an automated incident management plan include the following sequential tasks:

### **I. Needs Assessment:**

- Coordination\availability of information;
- Determine requirements for maps and databases;
- Determine incident management applications needed;
- Determine functionality needed;
- Determine other software needed for support of plan;

- Provide a basis for system planning;
  - Provide a basis for long-range implementation planning;
  - Determine costs involved.
- II. Long-range implementation plan:
- Outline objectives;
  - Select development software;
  - Select development environment;
  - Describe development tasks;
  - Outline priorities;
  - Outline schedules.
- III. Applications Development:
- Develop coverages (maps and databases);
  - Develop applications.
- IV. Applications testing:
- Test applications software functions;
  - Test for programming bugs;
  - Refine programs.
- VI. Implementation:
- Install programs;
  - Train operators.
- VII. Maintenance:
- Software support;
  - Develop upgrades.

### **6.3. RELATED RESEARCH, ALTERNATIVE APPROACHES, AND FUTURE WORK**

#### *Related Research*

Dynamic Route Guidance has been approached by T.N.Lam and C.O.Tong [14] using similar mechanism of historical and current travel pattern. They have proposed a dynamic network model using Dijkstra's shortest path algorithm and speed density relationship. The

model may be used for forecasting traffic flows by simultaneously monitoring traffic and updating the historical travel pattern whenever required.

#### *Alternative Approaches*

Expert Systems approach to incident management has been taken in two separate studies. Edmond C.P.Chang and K.Huarng have designed an Incident Management Expert System (IMES) using CLIPS expert system development tool which is commercially available from NASA [14]. The software includes incident detection, confirmation, identification, assessment, and response. Stephen G. Ritchie and Neil A. Prosser have separately developed a Freeway Real-Time Expert System Demonstration (FRED) for managing non-recurrent freeway congestion [15]. The system includes incident detection, verification, identification, response, and recovery monitoring. This system has been developed using G2 real-time expert system building tool, which is commercially available from GENSYM Corporation.

#### *Future Research*

An area in which the GIS could be extended for incident management is assisting incident management operators in identifying the measures to be taken to respond to an incident including the number and locations of variable message signs as well as the number and locations of incident management personnel at and about the site of an incident. Another area could be addition of incident detection capability by integration with expert systems technology.

In the future, a comprehensive study may also be taken up using an improved model for travel time simulation/forecasting system to better simulate traffic flow impacts. This study may include significantly more volume of real travel time data for analytical purposes.

In addition, the model may include improvements to:

- 1) minimize the effect of length of arcs;
- 2) improve the representation of the V/C to speed relationship;
- 3) realistically model the effect of lane blocking on capacity; and

4) incorporate recent and real historical hourly volume data as well as use of the computer clock for automatic selection of this historical data.

The results generated using this improved model may then be compared with real-time travel time data. If the simulated results closely match the real travel times, then this model could become the foundation for a dynamic route guidance system.

The Incident Analysis application may be extended to use traffic incident data available through TRASER software instead of the service patrol database, which has been used presently in the demonstration application. The application could also be enhanced to cover the entire Harris County.

Other GIS functionality that may be added to supplement the automated incident management plan are: polygon overlay and buffering, image display and management, linear modeling (dynamic segmentation), raster modeling and management, and surface modeling.

There are other commercially available software such as FREQ, TRANSYT-7F, NETSIM, CORFLO, HAZMAP, TRANSCAD/GISPLUS, and MGE which could have potential use for automated incident management. However, these are beyond the scope of this work and could be subject to separate studies. In particular, separate studies may be conducted to compare MGE and TRANSCAD/GISPLUS vis-a-vis ARC/INFO<sup>(C)</sup> as all three are GIS development environments.

Finally, there are a variety of technologies that can be examined for their interface and support to the automated incident management plan. These include Global Positioning System, Automatic Vehicle Identification, and Automatic Vehicle Location. The technologies would provide a means to: track vehicles, detect incidents, and provide real-time traffic data.

## REFERENCES

- [1] Environmental Systems Research Institute, *ARC/INFO Users Manuals*, Redlands, CA, 1991.
- [2] Texas Transportation Institute, *Arterial Signal Timing Optimization Using PASSER II-90 Program User's Manual*, Research Report 467-2F, Texas A&M University System, College Station, Texas, June 1991.
- [3] U.S. Environmental Protection Agency and NOAA Hazardous Materials Response Branch, *Computer Aided Management of Emergency Operations (CAMEO)*, Washington, D.C., November 1992.
- [4] Environmental Systems Research Institute, Inc., *Understanding GIS: The ARC/INFO Method*, revision 6, 1992.
- [5] Siegfried, Robert H., *Houston Motorist Assistance Program, Annual Report*, Texas Transportation Institute, Texas A&M University System, College Station, Texas, November 1991.
- [6] Siegfried, Robert H., *An Evaluation of METRO's Geographical Information System (GIS) Functional Requirements*, Texas Transportation Institute, Texas A&M University System, College Station, Texas, March 1993.
- [7] Texas Transportation Institute, *Houston-Galveston Regional Transportation Study, Travel Time and Speed Survey*, Texas A&M University System, College Station, Texas, July 1985.
- [8] Borhardt, Darrell, *Listing of 24-Hour Traffic Counts Completed in District No. 12 Since September 1989*, Texas Transportation Institute, Texas A&M University System, College Station, Texas, 1993.
- [9] Siegfried, Robert H., *Houston Motorist Assistance Program, Quarterly Report, February 1993 - March 1993*, Texas Transportation Institute, Texas A & M University System, College Station, Texas, July 1993.

- [10] McCasland, W.R. and Dennis Smalley, *A Public/Private Sector Partnership To Provide Traffic Information by Cellular Telephone Reports*, Texas Transportation Institute, Texas A&M University System, Draft Report, 1993.
- [11] Institute of Transportation Engineers, *Traffic Engineering Handbook*, Washington D.C., 1992.
- [12] Texas Transportation Institute, *Application Guide for the Microcomputer Version of PASSER-III-88*, Research Report 478-2F, Texas A&M University System, College Station, Texas, April 1990.
- [13] Nyerges, Timothy L. and Kenneth J. Dueker, *Geographic Information Systems in Transportation*, The Center for Urban Studies, Portland State University, July 1988.
- [14] Lam, T.N. and C.O. Tong, *A Dynamic Route Guidance System Based on Historical and Current Traffic Pattern*, in proceedings of Intelligent Vehicles 1992, Detroit pp 365-369.
- [15] Chang, Edmond C.P. and K. Huarng, *Freeway Incident Management Expert System Design*, in proceedings of 72nd Annual meeting of TRB, Washington D.C., January 1993.
- [16] Ritchie, Stephen G. and Neil A. Prosser, *A Real-Time Expert System Approach to Freeway Incident Management*, Transportation Research Board, 70th Annual Meeting, Washington D.C., 1991.

#### ADDITIONAL READINGS

- \* Abkowitz, Mark, Paul Der-Ming Cheng and Mark Lepofsky, *Use of Geographic Information Systems in Managing Hazardous Material Shipments*, Transportation Research Record #1261, GIS 1990, pp. 35-43.



- \* Albert, Steven, Robert H. Siegfried and Steven Levine, *Application of Geographic Information System to Houston Freeway Incident Management Program*, in proceedings of the 60th Annual Meeting of ITE, Orlando, Florida, 1990, pp. 131-132.
- \* Burns, Ken, *911 Services: Issues in Preparation and Use of the Geographic Data Base to Support Emergency Response*, in proceedings of NCGA GIS'90, Houston Texas, 1990, pp. 277-280.
- \* Date, C.J., *An Introduction to Database Systems*, Addison Wesley Publishing Co., 1990.
- \* Dudek, C. L. and G.L. Ullman, *Freeway Corridor Management*, National Co-operative Highway Research Program, Synthesis of Highway Practice #177, March 1992.
- \* Dueker, Kenneth J., *Geographic Information Systems: Research Issues*, The Center for Urban Studies, Portland State University, November 1988.
- \* Fletcher, D., *Modelling GIS Transportation Networks*, in proceedings of URISA 1988, Los Angeles, CA, Vol.2, 1987, pp. 84-92.
- \* O'Neill, Wende and Balakrishna Akundi, *Automated Conversion of Milepoint Data to Intersection/Link Network Structure: An Application of GIS in Transportation*, Transportation Research Record #1261, GIS 1990, pp. 27-34.
- \* Rex, Bruce, *Computational Geometry in Geographic Information Analysis*, The Center for Urban Studies, Portland State University, August 1989.
- \* Siegfried, Robert H., Nitin Vaidya, Sabas J. Avila, William McCasland and Steven Levine, *Incident Management Using Geographic Information Systems*, in proceedings of the 62nd Annual Meeting of ITE, Washington, D.C., 1992, pp. 433-436.
- \* Simkowitz, Howard J., *Recent Advances in Transportation GIS*, in proceedings of NCGA GIS'90, Houston Texas, 1990, pp. 277-280.
- \* Smith, Patti, *State Transportation Department Empowered by GIS*, in GIS World, April 1993, pp. 34-37.

- \* Texas Transportation Institute, *Freeway Design and Operations*, Texas A&M University System, College Station, Texas, 1989.
- \* Transportation Research Board, *Highway Capacity Manual*, Washington, D.C., 1985.
- \* Wiggins, Linda L. and Steven P. French, *GIS: Assessing Your Needs and Choosing A System*, Planning Advisory Service Report #433, American Planning Association, Chicago, Il., August 1991.

## **APPENDIX A**

### **TERMINOLOGY**



## TERMINOLOGY

Applications development refers to development of computer software that would perform a specific task or a set of tasks. In the context of the AIM Plan applications development refers to development of programs to automate the tasks required for incident management, including tasks for traffic center operations as well as tasks for planning and analysis of incidents.

An applications development platform or environment is a software that includes a set of programming tools for development of specific applications. The programs have been written for the AIM plan using PC-ARC/INFO GIS applications development platform.

The following computer terminology is used in context of the AIM plan:

**Application :** Computer software applied to a specific task or specific tasks for the operator, and is a collection of one or more programs.

**Software :** Software is a set of one or more programs designed to work together to perform computational tasks. Software may be classified into one or more of the following sub-categories:

*System Software:* Low level software used to operate the computer system. MS-DOS operating system, NOVELL Netware network operating system, compilers, interpreters, device drivers, and linkage loaders are examples of system software.

*Application Software:* Software that performs a specific task or a set of tasks in a general category. Examples are DBASE-IV, which is a database management software and Lotus 1-2-3, which is spread-sheet software.

*Application Development Software:* Software which consists of programming tools for development of specific application software. Example is PC-ARC/INFO GIS development software.

*Commercial Software:* Software available off-the-shelf with capabilities to perform certain tasks in a general category. Examples are WordPerfect which is a word-processing software and Lotus which is spread-sheet software.

*Custom Software:* Software developed for the specific needs of a specific user.

**Program :** A set of instructions for the operation of a computer or computer application.

**Customization :** Modification of a general purpose application software, such as WordPerfect, to the needs of a specific user by the developer of the software or by the user of the software.

**Computer Programming Language :**

A language in which instructions may be written for operation of the computer or a computer application. Computer languages may be classified into one or more of the following sub-categories:

*Low Level Language:* A programming language which closely resembles the internal operating instructions of a computer. These languages generally are used for systems software development. Assembly language such as ASSEMBLER is an example of such a language.

*High Level Language:* A high level language is more user friendly and can be written using English-like instructions hiding the internal operations of the computer from the programmer. An example is the BASIC language.

*General Purpose Programming Language:* General purpose programming language is one which has a variety of instructions for a variety of tasks. These languages provide high flexibility and power to an application programmer. These are generally high level languages. An example is the C programming language.

*Macro Language:* A macro language is one which provides instructions for a limited number of tasks within a specific application environment. Generally, macros are used to perform repetitive tasks in application software such as WordPerfect. However, macros may be used for customizing an application software or for development of other custom applications using application development software.

**Tools :** Tools are computer software or computer instructions which may be used for a number of tasks during application development, such as writing and executing macros, application debugging, and computer system information utilities

**Platform :** A platform is an application development software on the top of which an application may be developed and may be operated. The application may use the underlying functionality of this development platform. For this reason, the platform is also referred to as an *environment*.

**User Interface:** A means to provide the user with a way to interact with the computer in general and with the application in particular without needing to know the internal architecture of either the computer or the application. User interfaces should be user friendly so that the user can learn and operate the applications with minimal training and prior knowledge of the application itself.

**User Friendly Interface (UFI):**

A user friendly interface is one which allows the user to interact with a computer or application with menus and mouse such that the user requires no or limited knowledge as well as no or limited training of the commands and syntax which are required to operate the computer or the application without such an interface.

**Graphical User Interface (GUI):**

A graphical user interface has mouse selectable icons to represent menu choices besides having pull-down menus. Because of the symbolic nature of icons, these are often self explanatory and several icons may be available at once on a screen giving the user numerous alternative choices on one screen. In addition, a GUI supports other interface advances such as buttons, scroll bars, and toggle switches.

**Geographic Information System Development Tools:**

Computer software that could be customized to develop specific GIS applications. Some such tools are PC-ARC/INFO, MGE, TRANSCAD/GISPLUS, MAPINFO, and ATLAS-ARC.



**File:**

A file is a section of permanent memory on a computer used to store or record data or computer instructions as desired by the operator. Each file has a name by which it is referenced and is listed in a *directory*, which itself has a fixed location and which records its name and physical location on the permanent memory.



## **APPENDIX B**

**INCMAN: INCident MANagement Support Tool**

**A TRAFFIC MANAGEMENT CENTER USER'S GUIDE**



## **ACKNOWLEDGEMENT**

This user's guide is an update on the user's guide which was submitted previously as an interim report on this project by TTI staff Robert H. Siegfried and Nitin Vaidya.

## **INTRO: INTROduction**

INCMAN (INCident MANagement Support Tool) is a computer software that will provide fleet management for service patrols more effectively by providing the following broad functions:

1. Vehicle tracking,
2. Incident tracking, and
3. Vehicle dispatch and directions.

Additionally, INCMAN encompasses all of the incident response and dynamic route guidance functions of an automated incident management plan, which has been outlined in the main body of the report. In particular, INCMAN has been tailored to meet the fleet management needs of the Motorist Assistance Program (MAP) which operates fleet vans to provide assistance at incident sites.

The following is a brief description of this tool. This document covers the background, functionality, mechanics of the application, configurations and setup, terminology, and operating instructions to understand and implement the INCMAN application.

## **BACKG: BACKGround**

The following background describes the entities and their functions as they relate to the implementation of this application, followed by an outline of specific benefits INCMAN has as a tool in a traffic management center.

### **Motorist Assistance Program (MAP)**

As part of a Freeway Incident Management and Response Program, TxDOT has organized and managed the Houston Motorist Assistance Program (MAP) designed to assist minor incidents on designated freeways in Harris County. MAP is sponsored by public inter-agency and private support. The objective of MAP is to reduce the delay due to minor incidents that cause non-recurrent congestion while providing safety and traffic control to the motoring public. MAP consists of 9 vans patrolling designated freeways 16 hours/day, 5 day/week. The vans are equipped with jumper cables, gas, water, tools, and traffic control devices. MAP is managed from a Central Control Center (CCC) by TxDOT administrators. The vans are operated by Harris County Sheriff's deputies whose salaries are paid by the Metropolitan Transit Authority of Harris County. The program has been very effective, assisting over 1,600 incidents/month and averaging a total clearance time of 30 minutes.

### **Central Control Facility (CCF)**

The Central Control Facility (CCF) is the communication link between the MAP vans, incidents, and the motoring public. The following functions are performed by dispatchers in the CCF and MAP vans on the transportation network:

#### **TxDOT Dispatchers**

- Track and dispatch MAP vans to incidents using radio communication. Dispatching is based on assigned freeways and not actual vehicle locations.
- Operate an incident Hot line (CALLMAP) to receive calls from motorist/agencies.
- Control a variable message sign system through modem to inform the motoring public of traffic conditions.
- Dispatch forms to record: log in/off, turn points, incidents, others.

### **MAP Vans**

- Communicate to TxDOT dispatcher by radio providing ID #'s, location, incident data.
- Detect and respond to minor incidents to provide assistance and traffic control.
- Incident forms are filled out on scene.



### **BENEF: BENEFits**

The benefits of the INCMAN as a tool for the CCF to assist with incident and response management are:

- Color coded display of the transportation network showing dynamically MAP van and incident locations. This gives TxDOT dispatcher an eye in the sky to effectively dispatch and monitor traffic conditions for the variable message sign system.
- Computerized MAP van dispatching, providing direction to incidents and estimated time of arrival to scenes.
- Immediate accessibility to digital data sets to allow operations planning and analysis, such as clearance times, number of incidents, number of hours patrolled, turning point assessments, etc.
- Relieve volumes of paper work done daily.
- Ad-hoc database queries and alternative routing based on real time network and traffic conditions.

In sum, this tool will assist the operators of the CCF with efficient incident and traffic management in the areas of daily dispatching and tracking, decision support, operations' analysis, and variable message sign coordination.

### **FUNC: FUNCtionality**

INCMAN was developed using the PC-ARC/INFO GIS development package and the C programming language. The INCMAN consists of a map showing various freeway, major arterial, and frontage roads that are constantly displayed on your VGA display; a database that is not visible unless you choose to display it; and a user-interface, menu and mouse driven. The user-interface allows the operators of the CCC to perform the following functions:

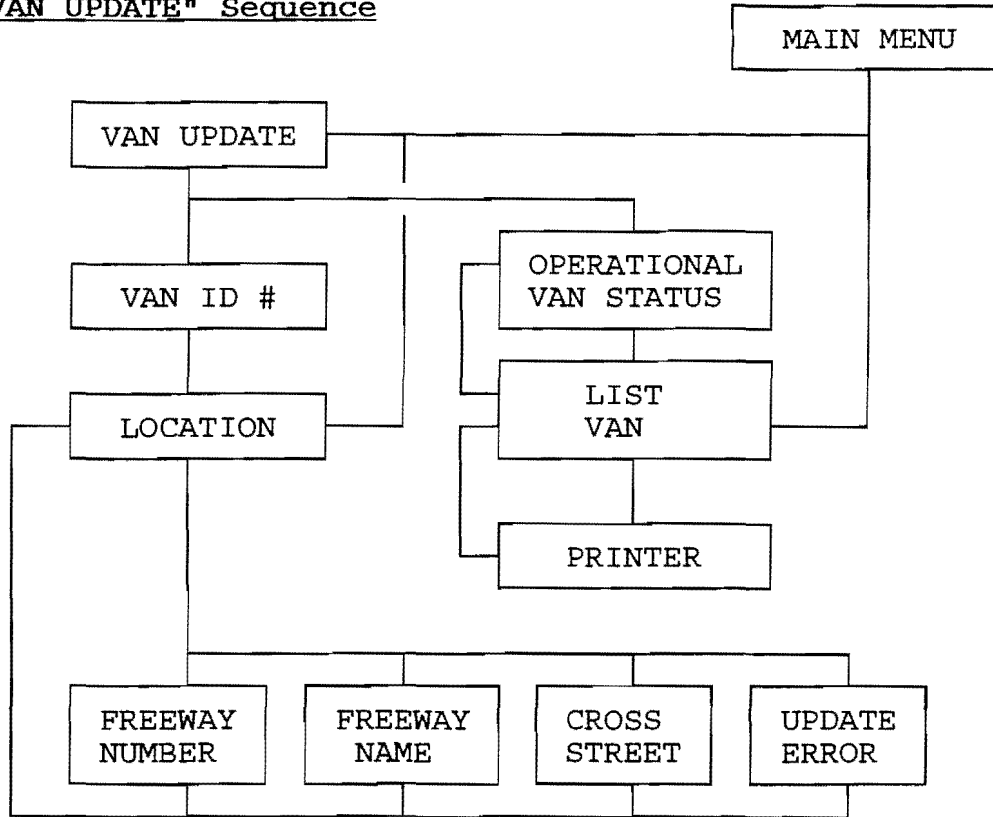
1. Enter and retract, using symbols and color coded displays,
  - (a) incident information, as well as
  - (b) updated locations and status of MAP vans.
  
2. Optimal routing on the map to find the closest MAP van based on travel time to any given incident, while providing the dispatched van with directions and estimated time of arrival to the incident.
  
3. Update freeway traffic conditions such as recurrent congestion volumes by time of day, and blocking of freeway capacity by specified number of lanes. These updates provide pseudo-real travel time information for routing coordination with the variable message sign system and MAP vans.
  
4. Dynamic route guidance using:
  - (a) Routing on any road segment, to estimate the travel time delay, or
  - (b) Alternate routing for non-recurrent and recurrent congestion, or
  - (c) Routing around a hazard area during conditions such as a chemical spill.
  
5. Graphical database queries on any roadway segment.

### **MECH: MECHAnics of the Application**

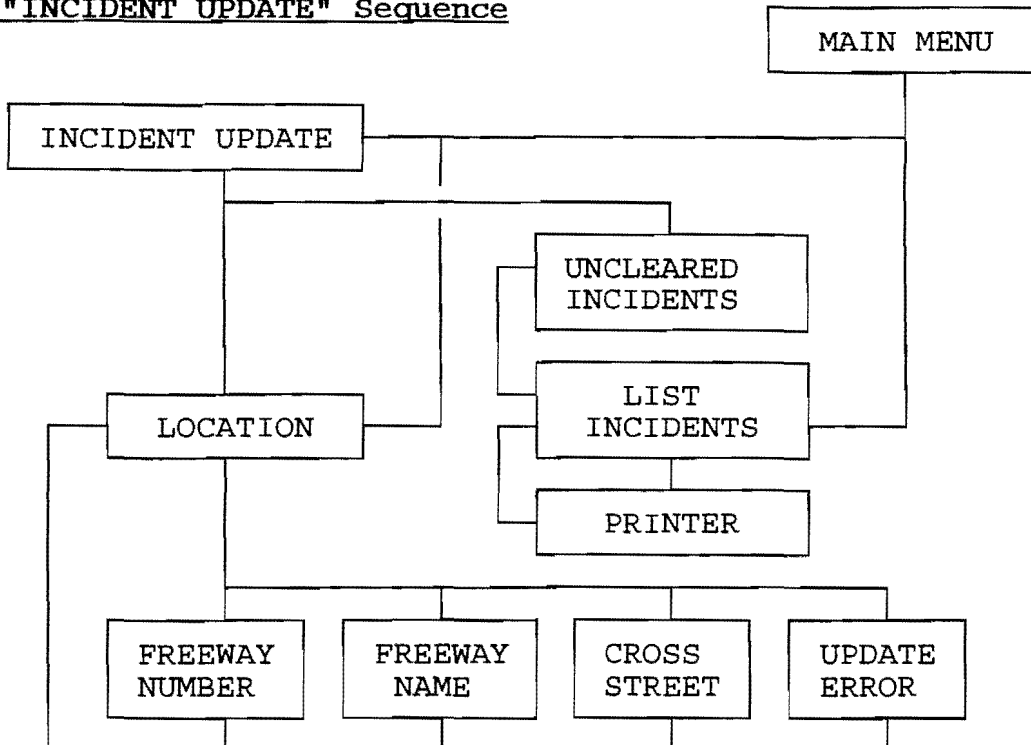
How does it work? After the user executes the INCMAN application by either turning the PC on or typing INCMAN at the C:\> prompt, the entire session is mouse and menu-driven with occasional keyboard response. That is, your PC VGA display will show you a set of menus that will enable you to enter incidents, enter van locations, and perform routing from any origin/destination. To perform these functions, you will select menu items using your mouse or keyboard arrow keys. There are variety pop-up menus and items/actions to choose from. The items/actions are selected by highlighting and then pressing the mouse keys. Some items perform actions invisible to the operator, while most invoke pop-up menus to eventually add/remove symbols from the Harris county map. The INCMAN flow chart sequences below graphically describe the flow control of the application to assist in gaining some perspective on what you are doing throughout the application.

## INCMAN Flow Chart Sequences

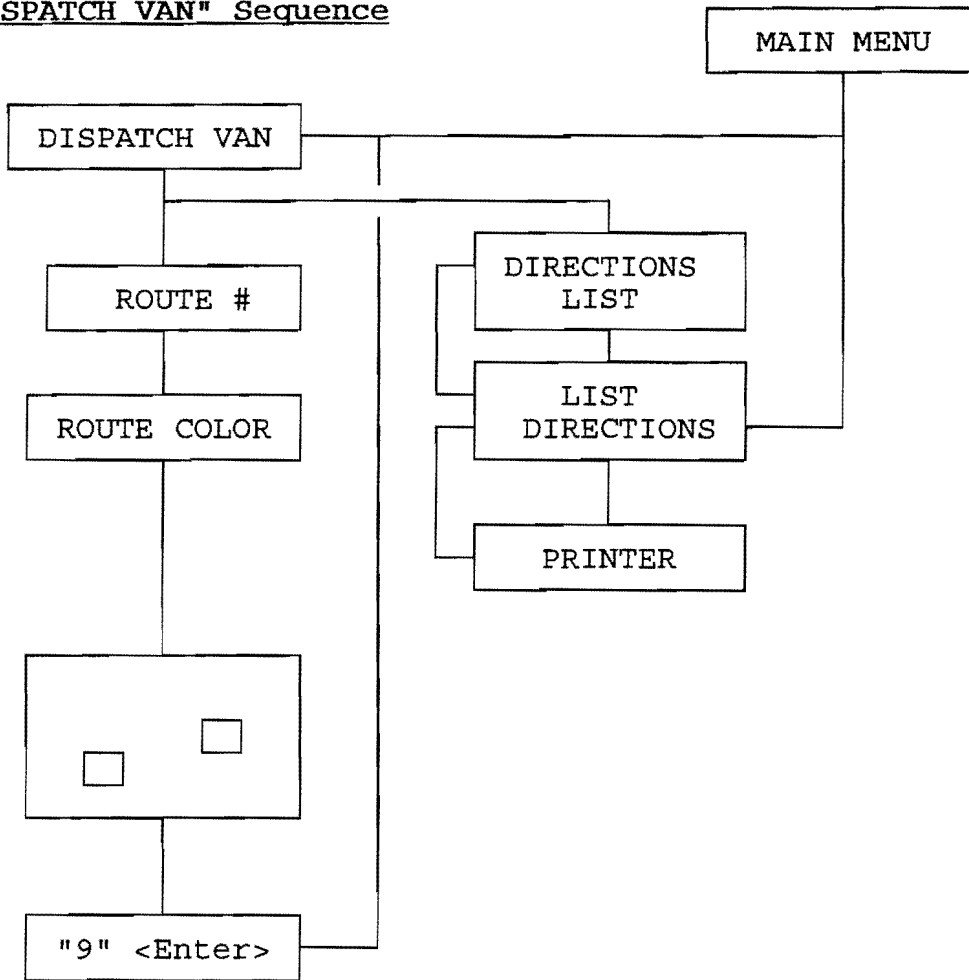
### "VAN UPDATE" Sequence



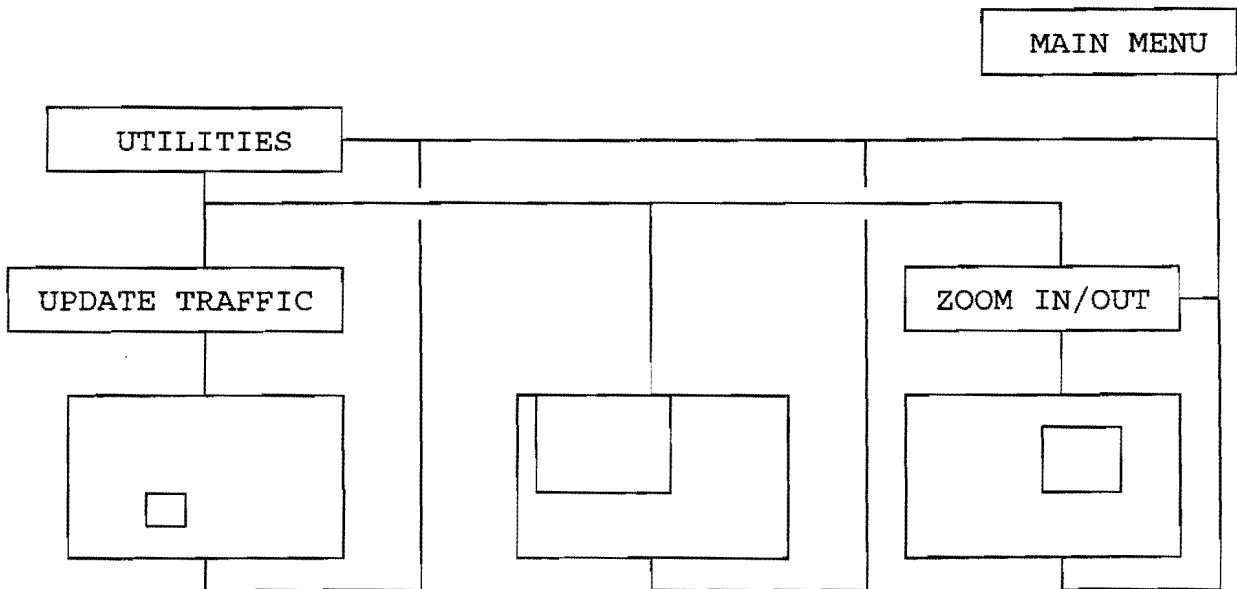
### "INCIDENT UPDATE" Sequence



"DISPATCH VAN" Sequence



"UTILITIES" sequence



In sum, the operator of INCMAN will interactively track and dispatch MAP vans and interactively enter incident information through the user-friendly interface provided. The updates performed for non-recurrent operations will provide an efficient and effective computerized automated incident management system.

## **CONFIG: CONFIGurations and Setup**

The INCMAN tool has been designed to run on a 80386 processor or better. The recommended operating configurations are as follows:

PC	: 80486 or better, DX2
Display	: SVGA, > =14"
Keyboard	: ASCII
Hard drive	: > =300Mb
Memory	: > =4Mb
Speed	: > =50 Mhz
Mouse	: Bus Type
DOS	: > =5.0
PC-Arc/Info	: 3.4d

## **TERM: TERMinology, Colors and Symbols**

This section briefly explains various application jargon and the color and symbol design for the roadway, MAP van status, and incident status to help you understand the operating instructions and their effects as you run the application.

### **Terminology**

- <Enter> - Enter key on your keyboard.
- Application - a programmed set of command statements provided in a text file to perform specific tasks within the interpreter program (PC-Arc/Info).
- Arc - a road segment bounded by two intersections or nodes.
- Assisting Van - a MAP van that has been notified of an incident and is in route, or a MAP van that is at the scene of an incident assisting stranded motorists.
- DOS - the operating system, Disk Operating System, used to run PC-Arc/Info.
- Incident - a non-recurrent minor incident where traffic control and/or minor repair/assistance is needed.
- MAP Van - a Motorist Assistance Program incident response van patrolling designated freeways 16 hours/day, 5 day/week.
- Menu - a boxed window with a list of menu actions/items to select.
- Mouse - a device used to select menu actions/items.
- Node - the intersection of 2 arcs or road segments.



- Patrolling Van** - a MAP van that is providing surveillance on the transportation network.
- Pop-up Menu** - a menu that replaces a menu on the screen with its own list of menu actions/items to select.
- Routing** - an algorithm that performs an origin/destination routine based on road segment length or time of travel.
- Select** - an interactive request to use the mouse or arrow keys on the keyboard to highlight the desired menu action/item; press the left button key or <Enter> key, respectively.
- User** - Person running the application, i.e., operator.

## Colors and Symbols

Assisting Van	- a yellow star.
Freeway Segment	- a green line/curve.
Incident	- a red square with an X inscribed.
Major Arterial	- a blue line/curve.
Minor Arterial	- a grey line/curve.
Patrolling Van	- a white star.
Road Block	- a white square with an X inscribed and node number.

**OPER: OPERating Instructions**

To run the INCMAN application, type "INCMAN <Enter>" at the C:\> prompt.

After some processing and messages on the screen, the screen changes to display a map of Harris County freeways and major arterial. The top of the screen shows the "MAIN MENU, A.I.M.S.:INCMAN (An Incident Management Support Tool)" (Figure B-1.0).

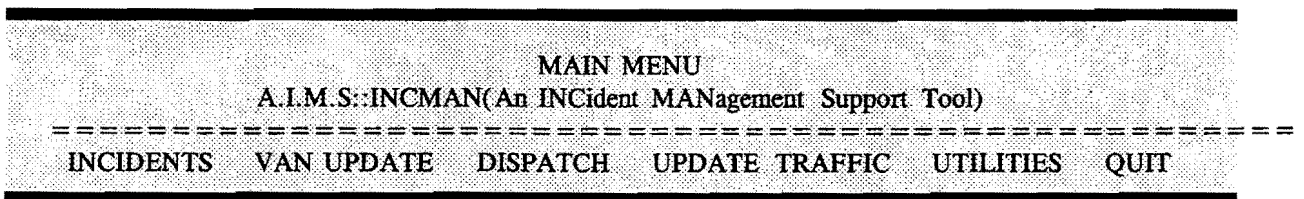


Figure B-1.0. Main Menu for INCMAN

The map of Harris County is color-coded. Green color represent the freeways, blue color represent the major arterial, and grey color represent other major roads. Also visible on the screen are white stars. Each of these signifies the starting location of the 9 MAP vans.

The entire application is self-prompting with on-screen instructions which leads the operator through each procedure to provide response team tracking/dispatching, incident detection, and variable message sign coordination.

On the "MAIN MENU," the operator has 6 menu item options to select from. These are "VAN UPDATE", "INCIDENT UPDATE", "DISPATCH VAN", "UPDATE TRAFFIC", "UTILITIES", and "QUIT". Each of these items will take the user through a series of sequences, as outlined in the following sections.

The "INCIDENT UPDATE" sequence allows the operator to update incidents detected and cleared, and provides a listing of current incidents not yet cleared.

The "VAN UPDATE" sequence allows the operator to update patrolling and assisting van status, and provides a listing of current van status.

The "DISPATCH VAN" sequence allows the operator to dispatch MAP vans to detected incidents providing directions and estimated time of arrival.

The "UPDATE TRAFFIC" sequence allows the operator to update traffic conditions, such as updation of recurrent conditions by time of day, or incident conditions on the network.

The "UTILITIES" sequence allows the operator to update traffic conditions, examine road segment information, and zoom in/out of the transportation network.

The last item on the "MAIN MENU", "QUIT", takes the user back to DOS.

## "VANUPDATE" Sequence

The first item on the "MAIN MENU", "VAN UPDATE", allows the operator to update van locations. Selecting this item will cause a pop-up menu called "VAN UPDATE" to be displayed on the screen (Figure B-2.0).

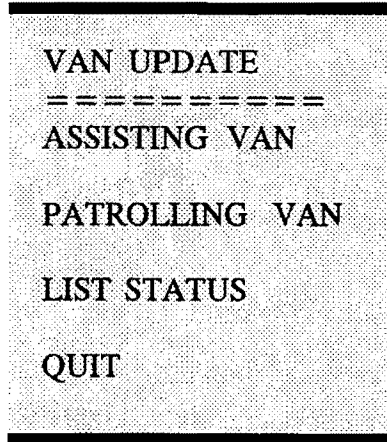


Figure B-2.0. "VAN UPDATE" Pop-Up menu

The "VAN UPDATE" Pop-Up menu has four items: "ASSISTING VAN", "PATROLLING VAN", "LIST STATUS", and "QUIT".

To update a patrolling van that is assisting a motorist or who is in-route to an incident, select the first item, "ASSISTING VAN". This selection will automatically invoke a "VAN ID NUMBER" pop-up menu (Figure B-2.1).

In the "VAN ID NUMBER" pop-up menu, you have the option to select a van unit number or to "CANCEL".

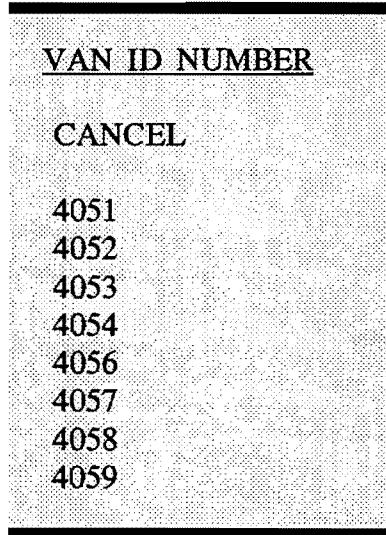


Figure B-2.1. "VAN ID NUMBER" pop-up menu

Selecting the appropriate van identification number from this menu will pop-up a "LOCATION" menu (Figure B-2.2).

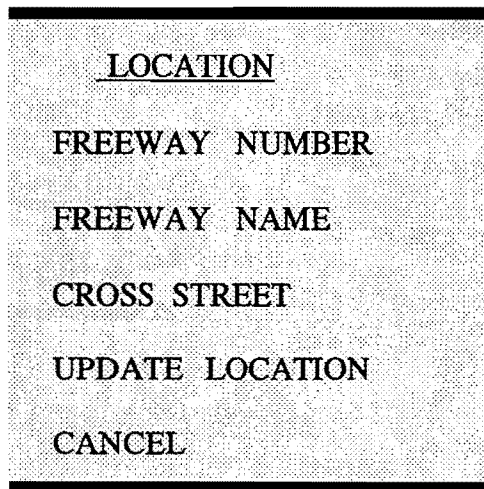


Figure B-2.2. "LOCATION" pop-up menu

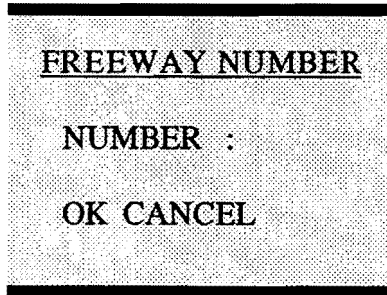
Use this menu to enter a location of the van/incident. This menu has five items: "FREEWAY NUMBER", "FREEWAY NAME", "CROSS STREET", "UPDATE LOCATION", and "CANCEL".

The operator may enter a location in one of the two following two methods:

- (1) By specifying the freeway block number and name of the freeway:
  - a. Select "FREEWAY NUMBER" and enter the freeway block number.
  - b. Select "FREEWAY NAME".
  
- (2) By specifying the intersection of the freeway with a cross-street:
  - a. Select the "FREEWAY NAME".
  - b. Select the "CROSS STREET".

Once a location has been selected, select "UPDATE LOCATION" on the "LOCATION" pop-up menu. If a valid location has been selected, after some processing, the van location will be visible on the screen. As previously noted, a yellow star indicates a van that is assisting a motorist, whereas a white star indicates that the van was last reported patrolling at that location. If the location is not valid, an error message will appear on the screen. If you do not want to enter a location, select the fifth item, "CANCEL", to return to the "MAIN MENU".

To select a location by the method (1) above, highlight the first item in the "LOCATION" pop-up menu, "FREEWAY NUMBER"; this selection pops up a "FREEWAY NUMBER" form (Figure B-2.3).



FREEWAY NUMBER  
NUMBER :  
OK CANCEL

Figure B-2.3. "FREEWAY NUMBER" form

Enter the desired freeway block number (just the number !) with your keyboard and then press <Enter>. Next select "OK"; or "CANCEL" to disregard number. After you select "OK" or "CANCEL" you will return to the "LOCATION" pop-up menu to continue with the location entry, method (1), or to "CANCEL" and return to the "MAIN MENU".



The second item on the "LOCATION" pop-up menu, "FREEWAY NAME", is used for providing the freeway corresponding to a block number entered for method (1), or for entering the freeway name followed by a cross street in method (2). Selecting it will pop-up a menu, "FREEWAY NAME" (Figure B-2.4), of freeway names. Select the appropriate freeway name at this menu.

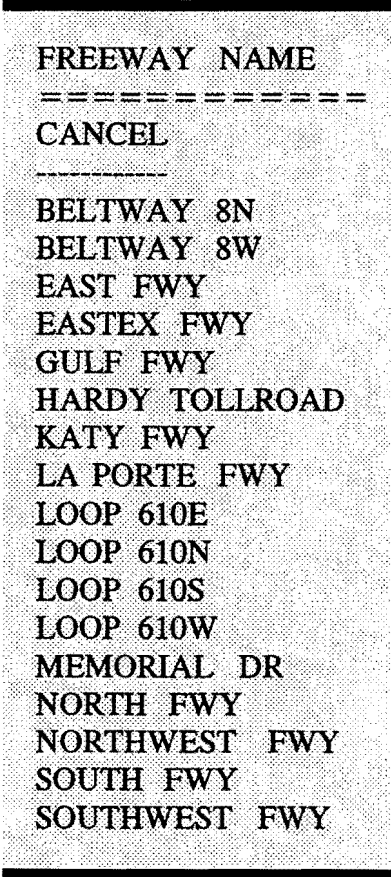


Figure B-2.4. "FREEWAY NAME" pop-up menu

The third item on the "LOCATION" pop-up menu, "CROSS STREET", is for entering a location as described in method (2) above. Selecting this item pops-up a menu, "CROSS STREET" (Figure B-2.5), of street names. Select the appropriate street name at this menu.

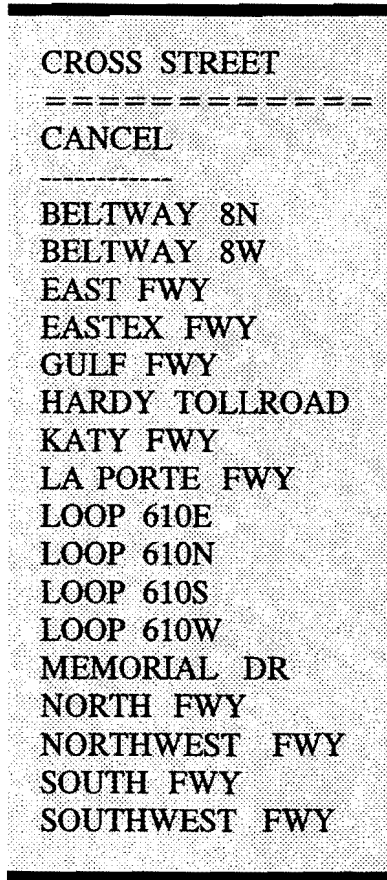


Figure B-2.5. "CROSS STREET" pop-up menu

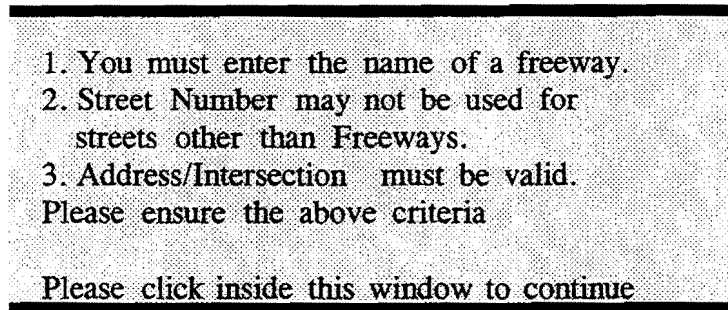


Figure B-2.6. "ERROR/WARNING" message

The fourth item on the "LOCATION" pop-up menu, "UPDATE LOCATION", updates the van/incident locations according to methods (1) or (2) above, and returns you to the "MAIN MENU" menu with the updates drawn on screen. If, however, the location specified is not valid an error/warning message as shown in Figure B-2.6 appears on the screen.

The last item on the "LOCATION" pop-up menu, "CANCEL", cancels this menu and returns you to the "MAIN MENU".

The second item on the "VAN UPDATE" pop-up menu, "PATROLLING VAN", will take you through the same sequence of menus as "ASSISTING VAN". Use the same procedures as above. The only visible difference would be a yellow star displayed to screen vs. a white star.

The third item on the "VAN UPDATE" pop-up menu, "LIST STATUS", lists the status of all the vans on the freeway network. Selecting this, pops a window (Figure B-2.7) on the left hand side of the screen.

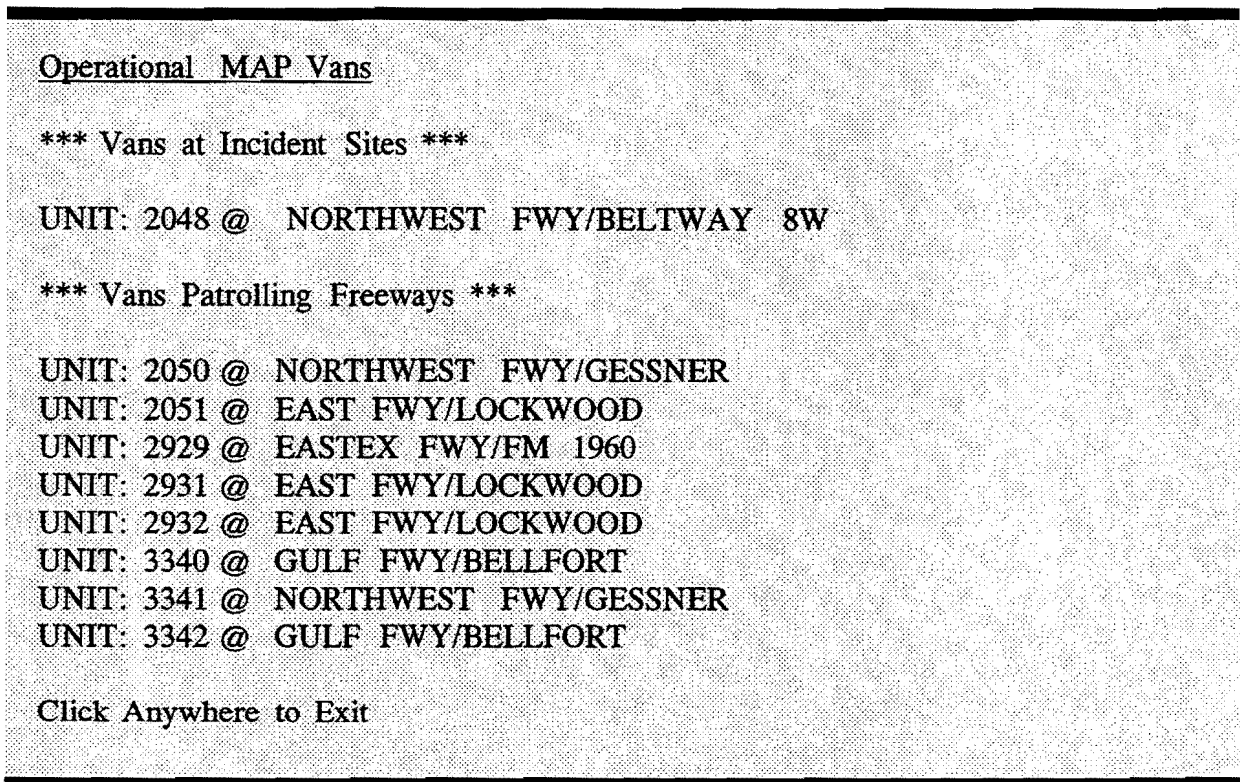


Figure B-2.7. "OPERATIONAL MAP VANS" window

At the top of the window it lists all the assisting vans under the heading, "\*\*\*\*\*Assisting MAP Vans\*\*\*\*\*", in the form "UNIT:ID# @ Location". Following that, it lists all patrolling vans under the heading "\*\*\*\*\*Patrolling Vans \*\*\*\*\*", in the same format. To

view the entire list you can click on "PgDn", for page down, or "Home", for page up. To exit from this window click anywhere on the screen.

Upon clicking the screen when the "Operational MAP Van" window is displayed, a "LIST VANS" pop-up menu (Figure B-2.8) provides the option to print van status to the screen again or to the printer.

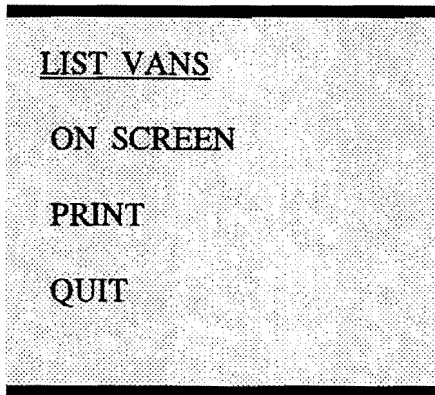


Figure B-2.8. "LIST VANS" pop-up menu

Select "ON SCREEN", to print to screen. Select "PRINT" from this menu to print a listing of all current van locations to a printer.

The fourth item on the "LIST VAN" pop-up screen, "QUIT" takes the user back up to the "MAIN MENU".

## "INCIDENT UPDATE" Sequence

The second item on the "MAIN MENU", "INCIDENT UPDATE", allows the operator to update incidents detected. Selecting this item will invoke a pop-up menu called "INCIDENT UPDATE" (Figure B-3.0).

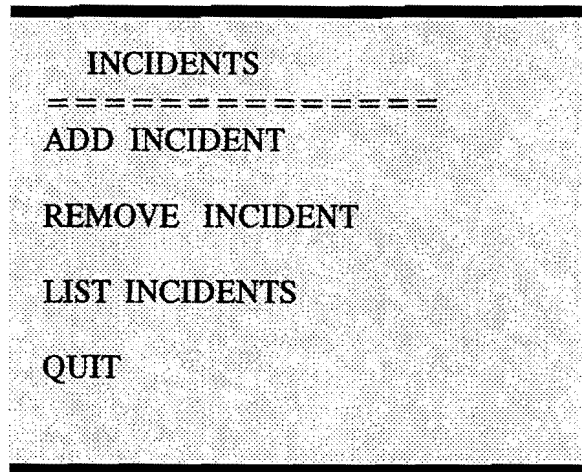


Figure B-3.0. "INCIDENT UPDATE"

The "INCIDENT UPDATE" menu has four items: "ADD INCIDENT", "REMOVE INCIDENT", "LIST INCIDENTS", and "QUIT". Incident updating works similar to the updating of van locations. This menu allows the operator to indicate incidents detected and clear those assisted. In addition, the operator can get a listing of the uncleared incidents. If you do not wish to update an incident location or view the locations of incidents in tabular form, select the fourth item, "QUIT", to return to the "MAIN MENU".

To update an incident detected select the first item, "ADD INCIDENT". This pops up the "LOCATION" menu (refer Figure B-2.2 section "LOCATION" pop-up menu). Use this menu to enter the desired location. Once the location of the incident is entered, the incident shows up on the screen in the form of a red box with an inscribed red X.

To update a cleared incident select the second item, "REMOVE INCIDENT". Again, use the "LOCATION" menu (refer Figure B-2.2) to specify the location to be removed (refer to the section above, "LOCATION" pop-up menu).

The third item on the "INCIDENT UPDATE" pop-up menu, "LIST INCIDENTS", is for a listing of current incidents detected and awaiting clearance. Selecting it pops a window (Figure B-3.1) on the screen.

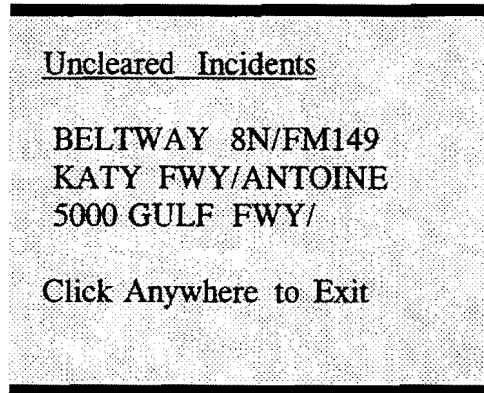


Figure B-3.1. "UNCLEARED INCIDENTS" window

Within this window are listed all currently detected incidents. The listing format for incidents is either; the street address such as "Street\_Number Street\_Name"; or an intersection such as "Freeway\_Name/Street\_Name". Click anywhere within this window to exit this window. Upon clicking a "LIST INCIDENTS" pop-up menu (Figure B-3.2) provides the option to print a listing of the current incidents to screen or to a printer.

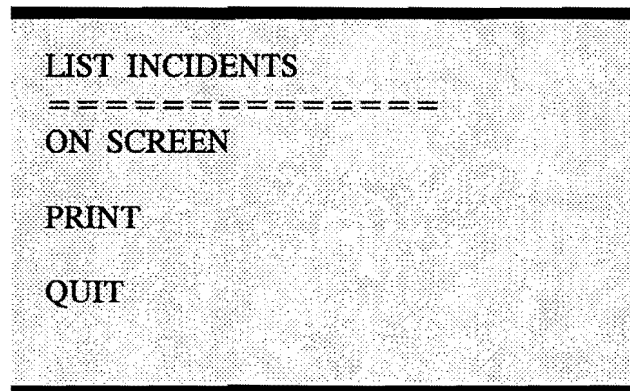


Figure B-3.2. "LIST INCIDENTS" pop-up menu

The fourth item on the "LIST INCIDENTS" pop-up menu, "QUIT", will take you back to the "MAIN MENU".

## "DISPATCH VAN" Sequence

The third item on the "MAIN MENU", "DISPATCH VAN", allows the operator to dispatch MAP vans to detected incidents. The response unit is provided with the optimal route, directions to his destination, and estimated time of arrival to the incident. In addition, this item can be used to coordinate variable message sign information. Selecting this item will cause a pop-up menu called "DISPATCH VAN" to be displayed on the screen (Figure B-4.0).

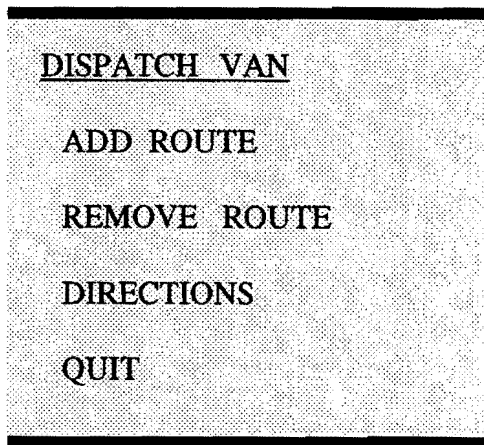


Figure B-4.0. "DISPATCH VAN"

The "DISPATCH VAN" pop-up menu has four items: "ADD ROUTE", "REMOVE ROUTE", "DIRECTIONS" and "QUIT".



To dispatch a MAP Van to the scene, select the first item, "ADD ROUTE". Selecting this will cause a pop-up "ROUTE NUMBER" menu (Figure B-4.1). If you do not wish to dispatch a van or perform a routing routine, select the fourth item, "QUIT", to return to the "MAIN MENU".

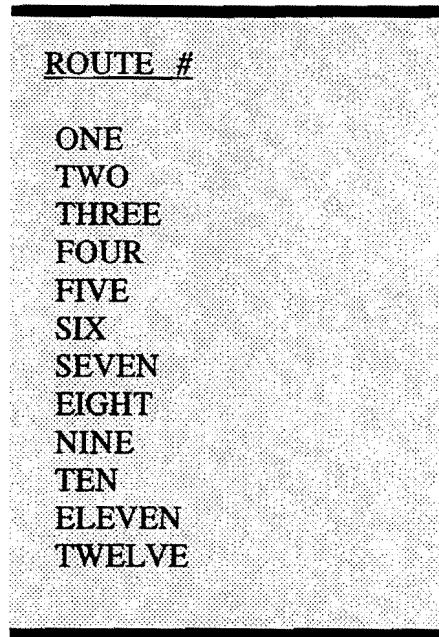


Figure B-4.1. "ROUTE #" pop-up menu

Every time a route needs to be calculated, the operator must select a new route number. The selection of a used number will cause the generation of an erroneous route. The system allows up to 12 route numbers. After 12 routes have been generated, some of the routes must be removed, before new routes can be generated.

After a route number has been selected, a "ROUTE COLOR" pop-up menu (Figure B-4.2) pops up.



Figure B-4.2. "ROUTE COLOR" pop-up menu

The operator may draw a route in any of the 15 colors listed on the menu. Once the route color is selected, the operator is taken to the Harris County map. Next, you will select a node for the origin, followed by a node for the destination of the desired route. To select a node, point and click the mouse on the closest street intersection. Once all nodes have been selected, press "9" <Enter>. The route then will display on the screen in the selected color and you will return to the "MAIN MENU".

To remove the last route, select the second item, "REMOVE ROUTE" from the "DISPATCH VAN" pop-up menu. Selection of this item causes removal of this route both from the list of routes and from the screen. The screen is redrawn following removal of the route.

The third item on the "DISPATCH VAN" pop-up menu, "DIRECTIONS", allows display and/or printing of the directions for the last route. Selecting this pops a window showing the directions (Figure B-4.3).

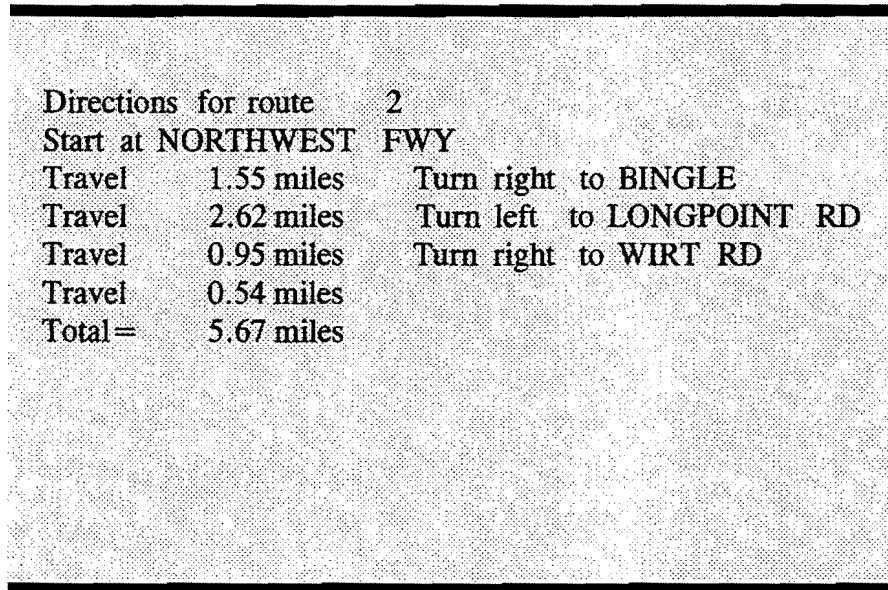


Figure B-4.3. "DIRECTIONS" window

At the top of the window is the heading, "Directions for route #(number)". Following that, Start at "street" following with Travel miles/time, turning directions, and Total miles/time. To view the entire list you can click on "PgDn", for page down, or "Home", for page up. To exit from this window click anywhere on the screen. Upon clicking, a "DIRECTIONS" pop-up menu appears (Figure B-4.4).

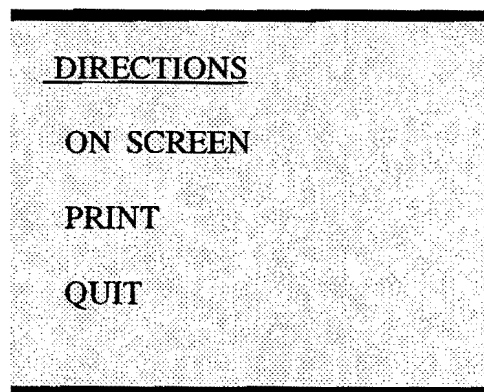


Figure B-4.4. "DIRECTIONS" pop-up menu

Select "ON SCREEN" for display on screen, or "PRINT" for hard copy of the directions.

The fourth item on the "DIRECTIONS" pop-up menu, "QUIT", takes the user back to the "MAIN MENU".

## "UPDATE TRAFFIC" Sequence

The fourth item on the "MAIN MENU" allows the operator to update traffic conditions. Selecting this option pops up a "UPDATE TRAFFIC" menu (Figure B-5.0).

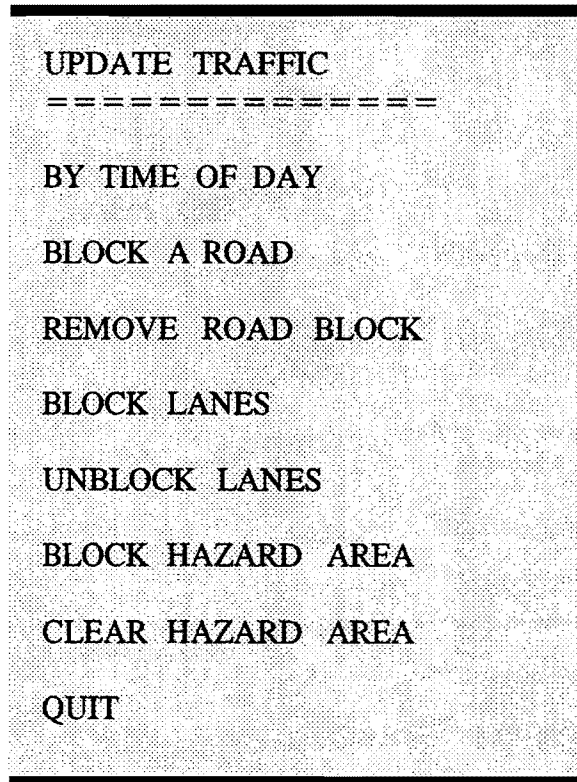


Figure B-5.0. "UPDATE TRAFFIC" pop-up menu

The "UPDATE TRAFFIC" menu has eight options: "BY TIME OF DAY", "BLOCK A ROAD", "REMOVE ROAD BLOCK", "BLOCK LANES", "REMOVE LANE BLOCK", "BLOCK HAZARD AREA", "CLEAR HAZARD AREA" and "QUIT". These options allow the user to update the traffic conditions on the freeway network for recurrent and non-recurrent congestion. If you do not wish to use this menu, select the last item, "QUIT", to return to the "MAIN MENU".

To block a freeway segment completely, select the second item, "BLOCK A ROAD". Following this selection the operator needs to select the street intersection to be blocked with the mouse. Once selected, the blocked intersection is shown on the screen with a white

box ,an X inscribed, and a node number. No traffic can pass through at this intersection. After a "BLOCK" has been made, you will return to the "MAIN MENU".

To clear a freeway segment that was previously blocked, select the third item, "REMOVE A ROAD BLOCK" on the "UPDATE TRAFFIC" pop-up menu. Following this selection, the operator needs to select the street intersection to be cleared with the mouse. The white box with an X is removed from the screen, and redrawn. After a "BLOCK" has been removed you will return to the "MAIN MENU".

To block individual lanes of a freeway segment, select the fourth item, "BLOCK LANES", on the "UPDATE TRAFFIC" pop-up menu. This selection causes the system to change from "ROUTE" module to "ARCEDIT" module. A menu pops up (Figure B-5.1) requiring the user to specify the number of lanes to block. After this, the user is required to zoom in on the area in which the desired street segment exists. The user then selects one or more street segments in the area as desired.

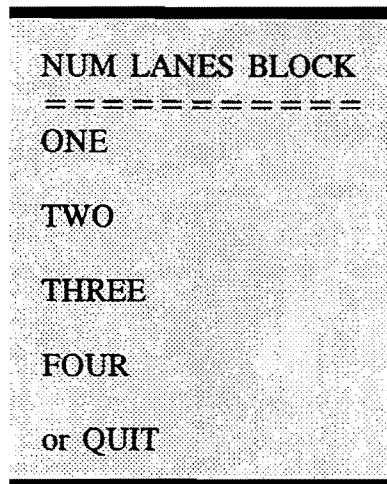


Figure B-5.1. "NUM LANES BLOCK" pop-up menu

After the number of lanes has been specified, another menu pops up (Figure B-5.2). The user is now required to specify the direction in which the lanes are to be blocked. "BLOCK FROM LANES" refers to-to-from direction of the arcs representing the street and "BLOCK TO LANES" refers to from-to-to direction.

Once this has been done, another menu pops up (Figure B-5.3) where the user is given the option, which is "YES", to start blocking of lanes on some other street segments. If a "NO" is selected, then the attribute table is updated using volume to capacity ratios and "ARCEDIT" module is exited and the "ROUTE" module re-entered reflecting the updated conditions.

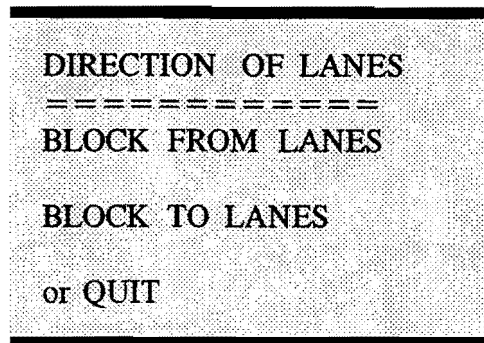


Figure B-5.2. "DIRECTION OF LANES" pop-up menu

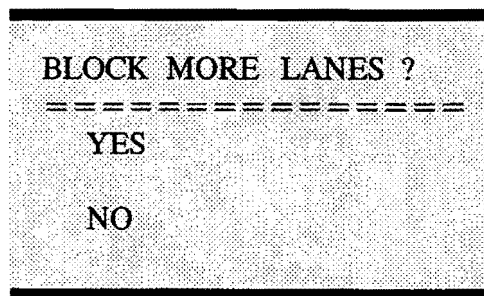


Figure B-5.3. "BLOCK MORE LANES ?" pop-up menu

To clear lanes blocked, select the fifth item, "REMOVE LANE BLOCK". All lane blocks are removed once this item is selected. For this purpose, "ROUTE" module is first exited, the "TABLES" module is entered, and the attribute table is appropriately updated before re-entering "ROUTE" module.

To block a hazardous area, such as one with chemical spill, select the sixth item "BLOCK HAZARD AREA". Upon such selection, the "ROUTE" module is exited and the "ALLOCATE" module is entered. A form pops up on the screen and the user must enter the radius of the area affected (Figure B-5.4). After this, the user must point and click the mouse to select the center of the area affected. Once this has been done, the system draws the segments to be blocked on the screen; to process the selections, it exits "ALLOCATE" module and enters "TABLES" module for further processing. Subsequently, the "ROUTE" module is re-entered and the appropriate intersections are automatically blocked. Further routing on the screen will bypass the affected area.

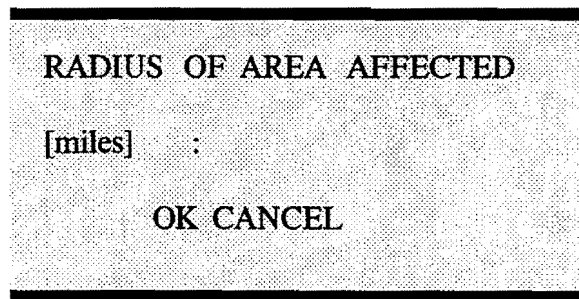


Figure B-5.4. "RADIUS OF AREA AFFECTED" pop-up form

To clear a hazard area selected the seventh item "CLEAR HAZARD AREA" on the menu. This would remove all the road blocks, redraw the map, and reset the system for further routing.

The first item on the "UPDATE TRAFFIC" pop-up menu, "BY TIME OF DAY", is to update traffic according to recurrent conditions. In a full implementation, the system will check the time of day (computer clock) and automatically update the traffic conditions according to historical data sets. However, for this prototype the "ROUTE" module is exited and the "TABLES" module is entered where a "PLEASE SELECT A TIME ZONE" menu pops up (Figure B-5.5). The user may select either "AM" or "PM".



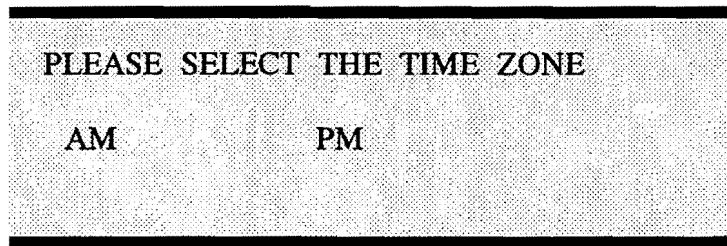


Figure B-5.5. "PLEASE SELECT A TIME ZONE" pop-up menu

After selection of time zone, a "TIME (AM)" or "TIME (PM)" menu will pop-up (Figures B-5.6 or B-5.7). The user may select the suitable time of the day. Following this, the attribute table is updated to recurrent traffic conditions corresponding to the user selected time. Thereafter, "ROUTE" module is re-entered for any other tasks.

The last item, "QUIT", takes the user back to the "MAIN MENU" menu.

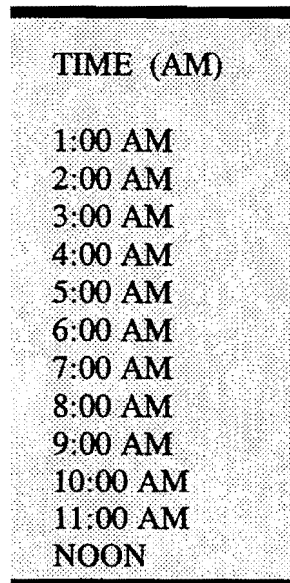


Figure B-5.6. "TIME (AM)" popup menu

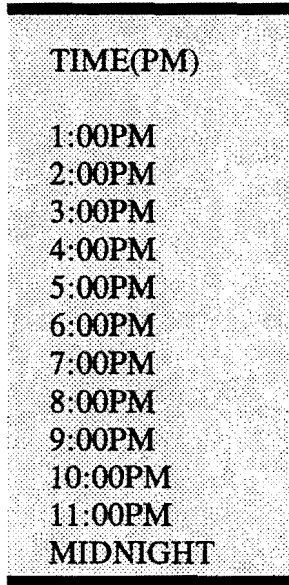


Figure B-5.7. "TIME (PM)" popup menu

## "UTILITIES" Sequence

The fifth item on the "MAIN MENU, "UTILITIES", allows the operator to examine street information, show a location on the street network, and zoom in/out of the displayed network. Selecting this item will cause a pop-up menu called "UTILITIES" to be displayed on the screen (Figure B-6.0).

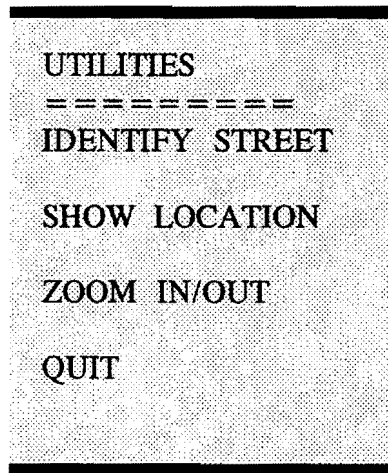


Figure B-6.0 "UTILITIES" pop-up menu

The "UTILITIES" pop-up menu has four items: "IDENTIFY STREET", "SHOW LOCATION", "ZOOM IN/OUT", and "QUIT". If you do not wish to use the "UTILITIES" pop-up menu, select the fourth item, "QUIT", to return to the "MAIN MENU".

The first item on the "UTILITIES" pop-up menu, "IDENTIFY STREET", is for graphical query on the streets database. If this option is selected, the user will be able to access the database of any street on the map by selecting the desired street with the mouse. After the street is selected, it is highlighted in white on the screen, and the street database values for items pertaining to names, street address and cross street are displayed in a window on the screen (Figure B-6.1). To clear this window click anywhere and you will return to the "MAIN MENU".

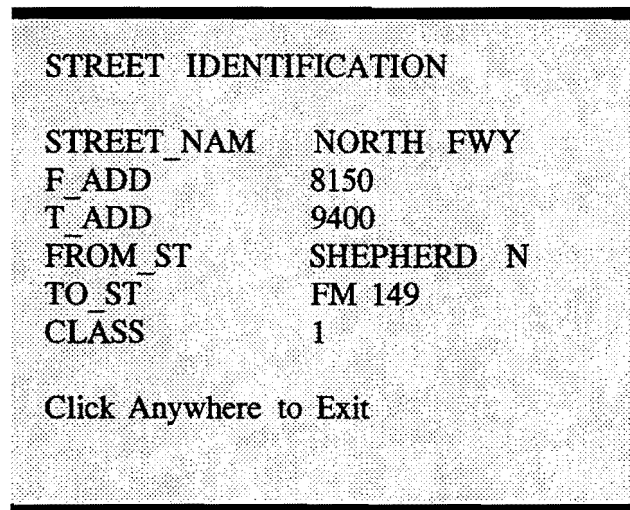


Figure B-6.1. "STREET IDENTIFICATION" window

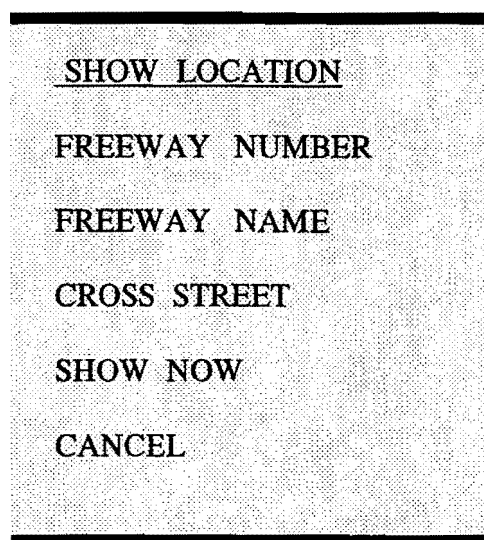


Figure B-6.2. "SHOW LOCATION" pop-up menu

The second item on the "UTILITIES" pop-up menu, "SHOW LOCATION", allows user to show any location on the screen. Selection of this item causes the pop-up of a "SHOW LOCATION" menu (Figure B-6.2.).

On the "SHOW LOCATION" menu the user/operator may show location by the street address or by specifying an intersection. This menu is very similar to Figure B-2.2, which is the menu for entering incident/van location. The only difference is that the fourth item, "SHOW NOW", instantly displays the location specified.

The third item on the "UTILITIES" pop-up menu, "ZOOM IN/OUT", allows the user to zoom in or zoom out on the map. Selection of this item causes the pop-up of a "ZOOM IN/OUT" menu (Figure B-6.3).

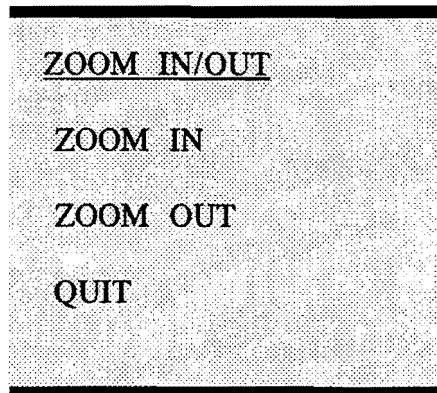


Figure B-6.3. "ZOOM IN/OUT" pop-up menu

The "ZOOM IN/OUT" menu has three items: "ZOOM OUT", "ZOOM IN" and "QUIT". If you do not wish to zoom the map, select the third item "QUIT", to return to the "MAIN MENU".

The first item, "ZOOM IN", would require the operator to select the two diagonal ends of the area to be zoomed in on. The operator only needs to point and click at any two points on the screen. The dynamic rectangular box represents the area you will zoom.

The second item, "ZOOM OUT", would zoom out on the map to the original extent of the map.

The third item, "QUIT", would take the operator back to the "MAIN MENU".

## **QUIT sequence**

The sixth and the last item on the "MAIN MENU", "QUIT", takes the operator back to DOS.





## **APPENDIX C**

### **STRUCTURE OF THE DATABASES**



**CONGESTION MANAGEMENT ARC ATTRIBUTE TABLE (AAT)**

Field	Field Name	Type	Width	Dec	Index	Comments
						Fields #1 to #7 are Arc/Info generated
1	FNODE_	Numeric	11		N	Node number of "from node"
2	TNODE_	Numeric	11		N	Node number of "to node"
3	LPOLY_	Numeric	11		N	Number of left polygon
4	RPOLY_	Numeric	11		N	Number of right polygon
5	LENGTH	Numeric	13	6	N	length in feet
6	INCMAN_	Numeric	11		N	Internal Arc number
7	INCMAN_ID	Numeric	11		N	TTI entered unique identification number
						TTI added additional fields #8 to #166
						Fields #8 to #19 were used to import LATITUDE and LONGITUDE values
8	F_X_COORD	Numeric	12	2	N	
9	F_Y_COORD	Numeric	12	2	N	
10	T_X_COORD	Numeric	12	2	N	
11	T_Y_COORD	Numeric	12	2	N	
12	X_COORD_F	Numeric	13	6	N	
13	Y_COORD_F	Numeric	13	6	N	
14	X_COORD_T	Numeric	13	6	N	
15	Y_COORD_T	Numeric	13	6	N	
16	F_COORD_X	Numeric	12	2	N	
17	F_COORD_Y	Numeric	12	2	N	
18	T_COORD_X	Numeric	12	2	N	
19	T_COORD_Y	Numeric	12	2	N	
						Fields #20 to #23 represent unique codes for importing data from external sources
20	F_GISCODE	Numeric	9		N	
21	T_GISCODE	Numeric	9		N	

Field	Field Name	Type	Width	Dec	Index	Comments
22	F_SUPCODE	Numeric	10			N Generated for accuracy
23	T_SUPCODE	Numeric	10			N
24	ST_CODE	Numeric	5			N From TTI 24-hour traffic count study.
25	ISITDUP	Character	1			N Used for generating the SUPCODE if duplicate GISCODE is found
26	SLENGTH	Numeric	13	6		N Used for updating GIS database during development.
27	CLASS	Numeric	9			N Class of road segment
28	STREET_NAM	Character	20			N Name of road segment
29	FROM_ST	Character	20			N Name of "from" intersection
30	TO_ST	Character	20			N Name of "to" intersection
31	SYMBOL	Numeric	11			N Color for display
32	ADDRESS	Character	45			N Includes Name, From, To numbers
						Field #28 and #33 to #38 are used for generating an address coverage for address matching
33	F_ADD	Numeric	6			N Street number of "from node"
34	T_ADD	Numeric	6			N Street number of "to node"
35	PRE_DIR	Character	2			N Name prefix; North, South, East or West
36	STREET_TYP	Character	4			N Street type; road, boulevard, avenue, street, drive, parkway, circle, gate, loop spur etc.
37	SUF_DIR	Character	2			N Name suffix; North, South, East or West
38	DISTANCE	Numeric	13	6		N Length in miles
39	ANACCIDENT	Numeric	6			N # of accidents annually
						Fields #40 to #53 are data items in the "to-to-from" direction
40	F_MILES	Numeric	13	6		N length in files
41	F_MAX_SPD	Numeric	11			N Max Speed
42	F_IMPEDANC	Numeric	13	6		N Impedance
43	F_IMP_FACT	Numeric	6	3		N Impedance Factor; used to update database in the model

Field	Field Name	Type	Width	Dec	Index	Comments
44	F_TOT_LANE	Numeric	11			N Total # lanes
45	F_AV_LANES	Numeric	11			N Available lanes
46	F_CAP_LANE	Numeric	11			N Capacity per lane
47	F_TOT_CAP	Numeric	11			N Total capacity
48	F_AV_CAP	Numeric	11			N Available capacity
49	F_VOLUME	Numeric	6			N Current volume of traffic
50	F_VOL_CAP	Numeric	6	3		N Volume to Capacity Ratio Travel Time Study
51	F_TTR_1	Numeric	4	2		N Morning travel time
52	F_TTR_2	Numeric	4	2		N Off-peak travel time
53	F_TTR_3	Numeric	4	2		N Evening travel time
Fields #44 to #67 are data items in the "from-to-to" direction						
54	T_MILES	Numeric	13	6		N length in miles
55	T_MAX_SPD	Numeric	11			N max speed
56	T_IMPEDANC	Numeric	13	6		N impedance
57	T_IMP_FACT	Numeric	6	3		N impedance factor; used to update database in the model
58	T_TOT_LANE	Numeric	11			N total number of lanes
59	T_AV_LANES	Numeric	11			N available lanes
60	T_CAP_LANE	Numeric	11			N capacity per lane
61	T_TOT_CAP	Numeric	11			N total capacity
62	T_AV_CAP	Numeric	11			N available capacity
63	T_VOLUME	Numeric	6			N current volume of traffic
64	T_VOL_CAP	Numeric	6	3		N volume to capacity ratio Travel Time Study
65	T_TTR_1	Numeric	4	2		N morning travel time
66	T_TTR_2	Numeric	4	2		N off-peak travel time
67	T_TTR_3	Numeric	4	2		N evening travel time
Fields #68 to #73 were used during development of the database.						
68	CENTER	Numeric	11			N
69	ROAD_DIR	Character	20			N
70	LOCATION	Character	15			N
71	MAPKEY	Character	5			N
72	MAPORDER	Numeric	3			N

Fields #73 to #96 are 24-hour hourly volumes in the "to-to-from" direction.

Field	Field Name	Type	Width	Dec	Index	Comments
73	F_AM1200	Numeric	5			N
74	F_AM0100	Numeric	5			N
75	F_AM0200	Numeric	5			N
76	F_AM0300	Numeric	5			N
77	F_AM0400	Numeric	5			N
78	F_AM0500	Numeric	5			N
79	F_AM0600	Numeric	5			N
80	F_AM0700	Numeric	5			N
81	F_AM0800	Numeric	5			N
82	F_AM0900	Numeric	5			N
83	F_AM1000	Numeric	5			N
84	F_AM1100	Numeric	5			N
85	F_PM1200	Numeric	5			N
86	F_PM0100	Numeric	5			N
87	F_PM0200	Numeric	5			N
88	F_PM0300	Numeric	5			N
89	F_PM0400	Numeric	5			N
90	F_PM0500	Numeric	5			N
91	F_PM0600	Numeric	5			N
92	F_PM0700	Numeric	5			N
93	F_PM0800	Numeric	5			N
94	F_PM0900	Numeric	5			N
95	F_PM1000	Numeric	5			N
96	F_PM1100	Numeric	5			N
						Fields #97 to #120 are 24-hour hourly volumes in the "from-to-to" direction.
97	T_AM1200	Numeric	5			N
98	T_AM0100	Numeric	5			N
99	T_AM0200	Numeric	5			N
100	T_AM0300	Numeric	5			N
101	T_AM0400	Numeric	5			N
102	T_AM0500	Numeric	5			N
103	T_AM0600	Numeric	5			N
104	T_AM0700	Numeric	5			N
105	T_AM0800	Numeric	5			N
106	T_AM0900	Numeric	5			N
107	T_AM1000	Numeric	5			N
108	T_AM1100	Numeric	5			N
109	T_PM1200	Numeric	5			N
110	T_PM0100	Numeric	5			N

Field	Field Name	Type	Width	Dec	Index	Comments
111	T_PM0200	Numeric	5			N
112	T_PM0300	Numeric	5			N
113	T_PM0400	Numeric	5			N
114	T_PM0500	Numeric	5			N
115	T_PM0600	Numeric	5			N
116	T_PM0700	Numeric	5			N
117	T_PM0800	Numeric	5			N
118	T_PM0900	Numeric	5			N
119	T_PM1000	Numeric	5			N
120	T_PM1100	Numeric	5			N
						Fields #121 to #166 are items from the roadway inventory
121	DATE	Character	5			N Date of data entry
122	RDATE	Character	8			N Date of roadway inventory
123	COUNTY_NUM	Character	3			N County number
124	STATION	Character	8			N Station
125	STREET	Character	20			N Street name
126	RFROM	Character	20			N Name of "from street"
127	RTO	Character	20			N Name of "to street"
128	STREETCODE	Numeric	5			N Street Code
129	SECTIONNUM	Numeric	3			N Section number
130	TYPE	Character	1			N Type; divided, one way etc.
131	FUNCTIONCL	Character	2			N Functional class; freeway , arterial etc.
132	STATESYS	Character	2			N State System; Interstate, SH or FM
133	RLOCATION	Character	3			N Location code
134	MAINTENANC	Character	1			N Maintenance by; County, State or City
135	COUNCILDIS	Character	1			N Council District
136	PRECINCT	Character	1			N Precinct
137	RLENGTH	Character	5			N Length in miles
138	ROWWIDTH	Character	3			N Right of way width
139	ROADWIDTH	Numeric	3			N Road width
140	NOLANES	Numeric	2			N Number of lanes
141	MEDIANWID	Character	3			N Median width
142	MEDIANDES	Character	1			N Median design; curb, guard rail, painted, ditch etc.
143	SIDEWALKS	Character	1			N Sidewalks, none, or one or both sides
144	CURBS	Character	1			N Curbs; none or continuous
145	SHOULDERS	Character	2			N Shoulder width
146	SURFACETYP	Numeric	1			N Surface type; concrete, asphalt etc.

Field	Field Name	Type	Width	Dec	Index	Comments
147	SURFACECON	Character	1			N Surface condition; excellent to poor.
148	ILLUMINATN	Character	1			N Illumination; none to adequate
149	PARKING	Character	1			N Parking restrictions
150	MARKINGS	Character	1			N Marking; center lines, lane lines etc.
151	TRAFFICSIG	Character	2			N Number of traffic signals
152	STOPSIGNS	Character	2			N Number of stop signs
153	YIELDSIGNS	Character	1			N Number of yield signs
154	CAUTIONLTS	Character	1			N Number of caution lights
155	SPEEDLMT	Numeric	2			N Speed limit
156	UNPROTECTE	Character	1			N Number of unprotected railroad crossings
157	CROSSBUCKS	Character	1			N Number of crossings with crossbucks
158	FLASHLGTS	Character	1			N Number of crossings with flash lights
159	FLASHGATE	Character	1			N Number of crossings with flash lights and gates
160	DAILYVOLUM	Numeric	6			N Daily volume over 24 hour period
161	DATECOUNT	Character	8			N Date of volume count
						Fields #162 to #166 in the roadway inventory pertain to the average daily travel (ADT).
162	ADT	Numeric	6			N
163	ADTFACOR	Character	5			N
164	RINGAREA	Character	1			N
165	QUADRANTA	Character	1			N
166	ADTCALC	Character	1			N
** Total **			1156			





## CONGESTION MANAGEMENT TURN TABLE

Field	Field Name	Type	Width	Dec	Index	Comments
						Fields #1 to #7 are PC-ARC/INFO generated
1	NODE_	Numeric	11			N Node number
2	ARC1_	Numeric	11			N PC-ARC/INFO number of from arc
3	ARC2_	Numeric	11			N PC-ARC/INFO number of to arc
4	AZIMUTH	Numeric	13		3	N Azimuth of from arc
5	ANGLE	Numeric	13		3	N Angle of turn
6	ARC1_ID	Numeric	11			N ID # of "from" arc
7	ARC2_ID	Numeric	11			N ID # of "to" arc
						Additional Fields
8	NCFRONT_ID	Numeric	11			N Used for relating to AAT
9	CLASS1	Numeric	2			N Class of "from" arc
10	CLASS2	Numeric	2			N Class of "to" arc
11	NAME1	Character	20			N Name of "from" arc
12	NAME2	Character	20			N Name of "to" arc
13	TURNTIME	Numeric	5		2	N Turn impedance in minutes
** Total **			142			

**CONGESTION MANAGEMENT POINT ATTRIBUTE TABLE (PAT) FOR "HOSPITAL"**

Field	Field Name	Type	Width	Dec	Index	Comments
						Fields #1 to #4 are PC-ARC/INFO generated
1	AREA	Numeric	13	6	N	
2	PERIMETER	Numeric	13	6	N	
3	HOSPITAL_	Numeric	11		N	
4	HOSPITAL_I	Numeric	11		N	TTI entered unique ID number TTI entered additional fields from #5 to #14
5	LOCATION	Character	20		N	Location of hospital
6	ADDRESS	Character	25		N	Address of hospital
7	NAME	Character	35		N	Name of hospital
8	ZIPCODE	Numeric	5		N	Zipcode for hospital
9	PHONE_NO	Character	8		N	Emergency phone number
10	NUM_BEDS	Numeric	6		N	Number of beds
11	NUM_AMBULA	Numeric	3		N	Number of ambulances
12	LIFEFLIGHT	Character	1		N	Whether lifeflight
13	COUNTY	Character	1		N	County of location
14	HOSP_TYPE	Numeric	2		N	State or private
** Total **			155			





**ADDITIONAL CONGESTION MANAGEMENT AAT ITEMS FOR "FIRE"**

<b>Field</b>	<b>Field Name</b>	<b>Type</b>	<b>Width</b>	<b>Dec</b>	<b>Index</b>	<b>Comments</b>
167	DXF_LAYER	Character	8			N Used for creation of coverage
168	CONTACT	Numeric	2			N Contact Name
169	CONTSYM	Numeric	2			N Symbol for display of contact
170	FIRE	Numeric	11			N Fire station address
171	FIRESYM	Numeric	2			N Symbol for display of fire station
** Total **			1181			

## **APPENDIX D**

**PROGRAMS WRITTEN TO DEMONSTRATE APPLICATIONS OF THE  
AUTOMATED INCIDENT MANAGEMENT PLAN**





```

rem *****
rem  FileName:          AIMS.BAT
rem  LastUpdate:       6/1/93
rem  Written By:       Nitin Vaidya
rem  Calls:            Arc/Info, AIMS.SML
rem  CalledBy:         None
rem  DataFiles:        None
rem  Variables:        None
rem  Purpose:          Batch file to start Arc/Info
rem                   as well as AIMS application
rem *****

rem *****
rem  Change drive and directory
rem *****
d:
cd \incman\pulldown

rem *****
rem  Start Arc/Info and run AIMS.SML
rem *****
arc aims

```

```

&rem *****
&rem      FileName:      AIMS.SML
&rem      LastUpdate:   6/1/93
&rem      WrittenBy:    Nitin Vaidya
&rem      Calls:        AIMS.COL
&rem      CalledBy:     AIMS.BAT
&rem      DataFiles:    None
&rem      Variables:    %21 -> variable used as
&rem                    condition for looping
&rem      Purpose:      To popup the AIMS.COL menu
&rem                    in a conditional loop and
&rem                    exit when user selects
&rem                    'QUIT' on the menu
&rem *****

```

```

&rem *****
&rem      Clear screen and initialize variable
&rem *****
cls
&sv 21 ""

```

```

&rem *****
&rem      Loop for conditional popup of AIMS.COL menu
&rem *****
&label CONTINUE
  popup aims.col sml
  &goto CONTINUE &if &nc %21 "QUIT"
q &return

```

```

&rem 21 2 1 13 18 53
&rem          'A.I.M.S
&rem          '=====
&rem          'Please`Select`An`Option
&rem          '=====
&rem
&rem  INCMAN`APPLICATION          SW`FWY`INCIDENTS
&rem
&rem  FRONTAGE`ROAD`ROUTING      SHOW`DETAILED`MAP
&rem
&rem  RESOURCE`DATABASE          RESOURCE`MANUAL
&rem
&rem  TRAVEL`TIME`SIMULATION      ROADS`DATABASE
&rem
&rem  AVL`DEMO`ARCINFO           AVL`DEMO`C
&rem
&rem  PASSER-II-90               PASSER-III-90
&rem
&rem  CAMEO                       QUIT
&rem EOF

&rem *****
&rem  In Arc/Info menu files, the comments can only be made
&rem  after the menu choices are indicated. Hence, the comments
&rem  are given here
&rem *****

&rem *****
&rem  FileName:      AIMS.COL
&rem  LastUpdate:   6/1/93
&rem  WrittenBy:    Nitin Vaidya
&rem  Calls:        \incman\pulldown\MAPLOT.SML
&rem              \incman\pulldown\AVLARC.SML
&rem              \incman\pulldown\SHOWDET.SML
&rem              \incman\pulldown\PASSER2.SML
&rem              \incman\pulldown\PDMAPH.SML
&rem              \dvolume\pulldown\PDMAPH.SML
&rem              \dbase34d\ncfront\pulldown\PDMAPH.SML
&rem              \swinc\pulldown\PLOTSW.SML
&rem              \resource\AGENCIES.SML
&rem              \remanual\RSETUP.SML
&rem              \cameo\CAMEO.EXE
&rem              \dbase34d\p2-90\PASSER2.BAT
&rem              \dbase34d\p3-90\PASSER3.BAT
&rem              \avl_c\CAR4.EXE
&rem  CalledBy:     AIMS.SML
&rem  DataFiles:    \dbase34d\p2-90\NEE.OUT
&rem              FILEVAR.SML
&rem  Variables:    %21 -> used to store the selection
&rem              by the user
&rem  Purpose:      Main menu for AIMS applications;
&rem              Allows user to select any application
&rem              and calls that application
&rem *****

&rem *****
&rem  Save all variables to a file FILEVAR.SML
&rem  Goto the label as per the menu choice
&rem *****
&save filevar.sml 1 50.

```

```

&jump %21

&rem *****
&rem INCMAN application
&rem *****
&label INCMAN
  &run pdmaph
  &goto end'aims

&rem *****
&rem FRONTAGE ROAD ROUTING
&rem *****
&label FRONTAGE'ROAD'ROUTING
  &sys "CD \dbase34d\ncfront\pulldown"
  &run pdmaph
  &sys "Cd \incman\pulldown"
  &goto end'aims

&rem *****
&rem ROADS DATABASE
&rem *****
&label ROADS'DATABASE
  arcplot maplot
  &goto end'aims

&rem *****
&rem SOUTHWEST FREEWAY INCIDENT SYSTEM
&rem *****
&label SW'FWY'INCIDENTS
  &sys "cd \swinc\pulldown"
  &run plotsw
  &sys "cd \incman\pulldown"
  &goto end'aims

&rem *****
&rem RESOURCE DATABASE
&rem *****
&label RESOURCE'DATABASE
  &sys "CD \resource"
  &run agencies
  &sys "cd \incman\pulldown"
  &goto end'aims

&rem *****
&rem RESOURCE MANUAL; Allocation of areas to Centers
&rem *****
&label RESOURCE'MANUAL
  &sys "cd \remanual"
  &run rsetup
  &sys "cd \incman\pulldown"
  &goto end'aims

&rem *****
&rem CAMEO
&rem *****
&label CAMEO
  &sys "cd \cameo"
  &sys "cameo"
  &sys "cd \incman\pulldown"
  &goto end'aims

```

```

&rem *****
&rem PASSER-II; update turntable with PASSER2.SML
&rem *****
&label PASSER-II-90
    &sys "CD \dbase34d\p2-90"
    &sys "passer2"
    &sys "cd \incman\pulldown"
    &sys "copy \dbase34d\p2-90\nee.out"
    &run passer2
    &goto end\aims

&rem *****
&rem PASSER-III
&rem *****
&label PASSER-III-90
    &sys "CD \dbase34d\p3-90"
    &sys "passer3"
    &sys "cd \incman\pulldown"
    &goto end\aims

&rem *****
&rem AVL DEMO; written in C language
&rem *****
&label AVL\DEMO\C
    &sys "cd\avl_c"
    &sys "car4"
    &sys "cd \incman\pulldown"
    &goto end\aims

&rem *****
&rem AVL DEMO; written in Arc/Info
&rem *****
&label AVL\DEMO\ARCINFO
    route avlarc
    &goto end\aims

&rem *****
&rem TRAVEL TIME SIMULATION
&rem *****
&label TRAVEL\TIME\SIMULATION
    &sys "cd \dvolume\pulldown"
    &run pdmaph
    &sys "cd \incman\pulldown"
    &goto end\aims

&rem *****
&rem DETAILED MAP; as obtained from TxDOT quad maps
&rem *****
&label SHOW\DETAILED\MAP
    route showdet
    &goto end\aims

&rem *****
&rem THE END
&rem *****
&label end\aims
    &goto QUIT
&label QUIT
    clear
    &return

```

```

&rem *****
&rem   FileName:      MAPLOT.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        QUERYMAP.COL
&rem   CalledBy:     AIMS.COL
&rem   DataFiles:    None
&rem   Variables:    None
&rem   Purpose:      Draw coverage D:\INCMAN in ArcPlot
&rem                  and run queries on the ROADS DATABASE
&rem *****

```

```

disp 4 3
setpage 1 1
clear
mapextent d:\incman
linesymbol 2
arcs d:\incman
linesymbol 3
&rem *****
&rem redraw freeways to highlight the freeways
&rem *****
reselect d:\incman arcs symbol eq 11
arcs d:\incman

&label CONTINUE
  popup QUERYMAP.COL sml
  &goto CONTINUE &if &ne %35 "QUIT"
q &return

```

```

&rem 35 1 1 55 21 24
&rem      `A.I.M.S
&rem `QUERY`ON`ROADS`DATABASE
&rem
&rem SPATIAL`SELECT
&rem
&rem STREET`NAME
&rem
&rem ROAD`WIDTH
&rem
&rem NUMBER`OF`LANES
&rem
&rem SURFACE`TYPE
&rem
&rem SPEED`LIMIT
&rem
&rem AVERAGE`DAILY`VOLUME
&rem
&rem ANNUAL`ACCIDENTS
&rem
&rem QUIT
&rem
&rem EOF

```

```

&rem *****
&rem      FileName:      QUERYMAP.COL
&rem      LastUpdate:    6/1/93
&rem      WrittenBy:     Nitin Vaidya
&rem      Calls:         None
&rem      CalledBy:      MAPLOT.SML
&rem      DataFiles:     None
&rem      Variables:     %35 -> User selection of query type
&rem                   %50 -> User specification of query parameters
&rem      Purpose:      Menu that allows user to perform
&rem                   graphical as well as text based
&rem                   queries on the ROADS DATABASE
&rem                   and displays the results in both
&rem                   text and graphics
&rem *****

```

```

linesymbol 5
&jump %35

```

```

&rem *****
&rem      Select street segments based on user specified street name
&rem *****
&label STREET`NAME
  &response 50 "STREET NAME IS :"
  reselect d:\incman arcs street_nam cn %50
  arcs d:\incman
  list d:\incman arcs street_nam from_st to_st
  &goto REDRAW

```

```

&rem *****
&rem      Select street segments based on user specified road width
&rem *****
&label ROAD`WIDTH
  &response 50 "ROADWIDTH :"
  reselect d:\incman arcs ROADWIDTH %50
  arcs d:\incman
  list d:\incman arcs street_nam from_st to_st roadwidth

```

```

&goto REDRAW

&rem *****
&rem Select street segments based on user specified number of lanes
&rem *****
&label NUMBER'OF'LANES
  &response 50 "NUMBER OF LANES : "
  reselect d:\incman arcs NOLANES %50
  arcs d:\incman
  list d:\incman arcs street_nam from_st to_st NOLANES
  &goto REDRAW

&rem *****
&rem Select street segments based on user specified surface type
&rem *****
&label SURFACE'TYPE
  &response 50 "SURFACE TYPE : "
  reselect d:\incman arcs SURFACETYP %50
  arcs d:\incman
  list d:\incman arcs street_nam from_st to_st surfacety
  &goto REDRAW

&rem *****
&rem Select street segments based on user specified speed limit
&rem *****
&label SPEED'LIMIT
  &response 50 "SPEED LIMIT : "
  reselect d:\incman arcs SPEEDLMT %50
  arcs d:\incman
  list d:\incman arcs street_nam from_st to_st speedlmt
  &goto REDRAW

&rem *****
&rem Select street segments based on user specified daily volume
&rem *****
&label AVERAGE'DAILY'VOLUME
  &response 50 "DAILY VOLUME : "
  reselect d:\incman arcs dailyvolum %50
  arcs d:\incman
  list d:\incman arcs street_nam from_st to_st dailyvolum
  &goto REDRAW

&rem *****
&rem Select street segments based on user specified number of accidents
&rem *****
&label ANNUAL'ACCIDENTS
  &response 50 "ANNUAL ACCIDENTS : "
  reselect d:\incman arcs ANACCIDENT %50
  arcs d:\incman
  list d:\incman arcs street_nam from_st to_st anaccident
  &goto REDRAW

&rem *****
&rem Select street segments graphically and list database info
&rem *****
&label SPATIAL'SELECT
  reselect d:\incman arcs BOX *
  arcs d:\incman
  list d:\incman arcs street_nam from_st to_st
  &goto REDRAW

&label ZOOM'IN

```



```

mapextent *
&goto REDRAW

&label ZOOM`OUT
mapextent d:\incman
&goto REDRAW

&rem *****
&rem Redraw map after any selection based display
&rem *****
&label REDRAW
clear
aselect d:\incman arcs incman_id gt 0
linesymbol 2
arcs d:\incman
reselect d:\incman arcs symbol eq 11
linesymbol 3
arcs d:\incman
aselect d:\incman arcs incman_id gt 0
&goto END`QUERY

&rem *****
&rem End of Queries
&rem *****
&label END`QUERY
&goto QUIT
&label QUIT
&return

```

```

&rem *****
&rem   FileName:      AVLARC.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     AIMS.COL
&rem   DataFiles:    D:\ARCEXE\SYMBOLS\COLOR\WHERE.DBF
&rem   Variables:    %1 ->  Variable for address
&rem                  %2 ->  temporary variable for
&rem                      time lag
&rem   Purpose:      Draw coverage D:\DVOLUME and
&rem                  simulate a moving vehicle on
&rem                  the screen
&rem *****

```

```

readnetwork d:\dvolume
addresscov d:\dvolume
disp 4 3
setpage 1 1
mapex d:\dvolume
drawnetwork symbol

```

```

&rem *****
&rem   Use newly defined markerset file;
&rem   this file is WHERE.DBF in the
&rem   D:\ARCEXE\SYMBOLS\COLOR directory
&rem *****
markerset color.where

```

```

&rem *****
&rem   the variable %1 stores the simulated address
&rem   of the location of the vehicle
&rem *****
&sv 1 2000

```

```

&rem *****
&rem   This loop increments the address of the
&rem   location of the vehicle and shows the
&rem   vehicle at the new location allowing display
&rem   of up to 60 locations of the vehicle
&rem *****
&label LOOP

```

```

&rem *****
&rem   Draw the symbol of the van at the
&rem   new location of the van
&rem *****
whereis 3 %1 NORTH FWY

```

```

&rem *****
&rem   Variable %2 is used to induce a time lag
&rem   in the display of new location
&rem *****
&cv 2 0
&label PASSTIME
  &cv 2 ( %2 + 1 )
  &goto PASSTIME &if &rn %2 0 40

```

```

&rem *****
&rem   Erase the symbol of moving van from
&rem   current location

```

```
&rem *****
whereis 2 %1 NORTH FWY

&rem *****
&rem Redraw the freeway which has been erased by
&rem the moving vehicle
&rem *****
whereis 11 NORTH FWY

&rem *****
&rem Increment %1 , i.e., the street address by
&rem 100 and repeat process up to address 12000
&rem *****
&cv 1 ( %1 + 100 )
&goto LOOP &if &rn %1 2000 11900

&rem *****
&rem End of program
&rem *****
q &return
```

```
&rem *****
&rem   FileName:      SHOWDET.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        SHOWDET.COL
&rem   CalledBy:     AIMS.COL
&rem   DataFiles:    None
&rem   Variables:    None
&rem   Purpose:      To show D:\DBASE34D\NEW coverage
&rem                which shows how detailed the
&rem                map was originally.
&rem *****
```

```
readnetwork d:\dbase34d\new
disp 4 3
setpage 1 1
mapex d:\dbase34d\new
drawnet symbol
```

```
&rem *****
&rem   Popup a window to prevent the application from
&rem   exiting and closing the display
&rem *****
popup showdet.col sml
q return
```

```
&rem 29 1 1 58 1 21
&rem PRESS ANY KEY TO EXIT
&rem EOF
```

```
&rem *****
&rem   FileName:      SHOWDET.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     SHOWDET.SML
&rem   DataFiles:    None
&rem   Variables:    %29 -> temporary
&rem   Purpose:      To prevent the display
&rem                 from closing
&rem *****
```

```
&rem
&rem Execute the selected option
&return
```

```

&rem *****
&rem   FileName:      PASSER2.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        TABLES, D:\READPASS.SML
&rem   CalledBy:     AIMS.COL
&rem   DataFiles:    NEE.OUT
&rem                 FILEVAR.SML
&rem   Variables:    %50 -> temporary variable for looping
&rem                 %2  -> first name of arterial
&rem                 %3  -> second name of arterial
&rem                 %4  -> temporary variable to search
&rem                 for keywords
&rem                 %5  -> first name of intersecting street
&rem                 %6  -> second name of intersecting street
&rem                 %7  to %10 -> phase directions for arterial
&rem                 %11 to %14 -> phase directions for cross street
&rem                 %15 to %22 -> delay in veh/sec per
&rem                 phase direction
&rem   Purpose:      This program reads the output generated
&rem                 by the execution of PASSER-II-90 and
&rem                 searches for keywords to read the names
&rem                 of arterials as well as intersecting street,
&rem                 the phase direction and the delay in
&rem                 veh/sec per phase direction. These are stored
&rem                 in Arc/Info variables and saved to a file
&rem                 FILEVAR.SML for later use. Finally, it
&rem                 invokes TABLES and executes READPASS.SML
&rem                 program to incorporate the data read by this
&rem                 program into turn table of AIMS coverage.
&rem *****

```

```
&open nee.out NOTOPEN
```

```
&label GET'ARTERIAL'NAME
```

```

&cv 50 1
&rem *****
&rem   Loop to reach the line which has the name of
&rem   the arterial
&rem *****
&label ARTERIAL'LOOP
&read 1 ENDFILE
&cv 50 %50 + 1
&goto ARTERIAL'LOOP &if &rn %50 1 6

```

```

&rem *****
&rem   %2, %3 combine to form the name of ARTERIAL
&rem *****
&extract 2 1 2
&extract 3 1 3
&goto GET'CROSS'STREET

```

```
&label GET'CROSS'STREET
```

```

&rem *****
&rem   Conditional loop to find the keywords
&rem   "**** INTERSECTION" for
&rem   finding the name of intersecting street
&rem *****
&label CROSS'STREET'LOOP
&read 1 ENDFILE

```

```

&extract 4 1 1
&goto CROSS'STREET'LOOP &if &ne %4 "*****"
&extract 4 1 2
&goto CROSS'STREET'LOOP &if &ne %4 "INTERSECTION"

&rem *****
&rem %5, %6 combine to form the name of CROSS STREET
&rem *****
&extract 5 1 4
&extract 6 1 5
&goto GET'PHASE'DIRECTION

&label GET'PHASE'DIRECTION

&rem *****
&rem Conditional loop to find the keyword
&rem "PHASE DIRECTION" for reading phase directions
&rem *****
&label PHASE'LOOP
&read 1 ENDFILE
&extract 4 1 1
&goto PHASE'LOOP &if &ne %4 "PHASE"
&extract 4 1 2
&goto PHASE'LOOP &if &ne %4 "DIRECTION"

&rem *****
&rem read the phase directions into variables from
&rem %7 to %14; where %7 to %10 correspond to
&rem arterial phase directions and %11 to %14
&rem correspond to cross street phase directions
&rem *****
&extract 7 1 3
&extract 8 1 4
&extract 9 1 5
&extract 10 1 6
&extract 11 1 7
&extract 12 1 8
&extract 13 1 9
&extract 14 1 10
&goto GET'DELAY

&label GET'DELAY

&rem *****
&rem Conditional loop to find the keyword "DELAY (SECS/VEH)"
&rem for reading delay numbers per phase direction
&rem *****
&label DELAY'LOOP
&read 1 ENDFILE
&extract 4 1 1
&goto DELAY'LOOP &if &ne %4 "DELAY"
&extract 4 1 2
&goto PHASE'LOOP &if &ne %4 "(SECS/VEH) "

&rem *****
&rem read the delay in secs/veh into variables from
&rem %15 to %22; where %15 to %18 correspond to
&rem arterial delays and %19 to %22 correspond to
&rem cross street delays

```

```

&rem *****
&extract 15 1 3
&extract 16 1 4
&extract 17 1 5
&extract 18 1 6
&extract 19 1 7
&extract 20 1 8
&extract 21 1 9
&extract 22 1 10
&goto END'PASSER2

&label NOTOPEN
  &type "File Could not be opened"
  &goto END'PASSER2

&label ENDFILE
  &type "End of File Reached"
  &goto END'PASSER2

&label END'PASSER2
  &rem *****
  &rem Close files, save variables to file
  &rem FILEVAR.SML, copy it to "\",
  &rem Change directory to "\", invoke tables
  &rem and execute READPASS.SML
  &rem *****
  &close
  &save filevar 1 50
  &sys "copy filevar.sml \"
  &sys "cd\"
  tables readpass
  &sys "cd \incman\pulldown"
  &return

```



```

&rem *****
&rem   FileName:      READPASS.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     D:\incman\pulldown\PASSER2.SML
&rem   DataFiles:    FILEVAR.SML
&rem   Variables:    %3  ->  variable to remove unwanted second name
&rem                                     of the arterial
&rem                                     %39 -> Stores command to select turn table records
&rem                                     based on names of arterial and the
&rem                                     cross street
&rem                                     %40 to %43 -> variables for direction
&rem                                     %44 -> Keyword "ARTERIAL" or "CROSS_STREET" used
&rem                                     for accessing label %44_BACK_TO_LOOP
&rem                                     %45 -> Name of turn table
&rem                                     %46 -> temporary variable points to variable,
&rem                                     i.e. variables %7 to %14,
&rem                                     in which the phase direction is stored
&rem                                     %47 -> temporary variable for phase direction
&rem                                     %48 -> temporary variable points to variable,
&rem                                     i.e., variables %40 to %43,
&rem                                     in which the direction is stored
&rem                                     %49 -> temporary variable for direction
&rem                                     %50 -> temporary variable for storing the delay
&rem                                     in veh/sec corresponding to phase dir.
&rem   Purpose:      This program uses the previously read names, delays
&rem                                     and phase directions to update the turn table.
&rem                                     To find matched turn table records the program
&rem                                     searches the records for names of arterial
&rem                                     and cross-street. Thereafter the orientation or
&rem                                     azimuth information in turn table is used to
&rem                                     reduce the search to match with the phase direction.
&rem                                     Finally the angle of turn information in the turn
&rem                                     table is used to match with the phase direction
&rem                                     to match the turn table record with PASSER generated
&rem                                     output. Once this has been done the turn time in
&rem                                     the turn table is updated. In the matching process
&rem                                     an assumption is that if arterial is NB/SB, the
&rem                                     cross street is EB/WB and vice-versa
&rem *****
&run filevar

&rem *****
&rem %3 initialized to remove unwanted
&rem second name of the arterial
&rem *****
&goto CONTINUE &if &ne %3 "DISTRICT"
  &sv 3 ""

&label CONTINUE
&rem *****
&rem save directions to variables
&rem %40 to %43
&rem *****
&sv 40 NB
&sv 41 SB
&sv 42 EB
&sv 43 WB

&rem *****

```

```

&rem save name of turn table to variable
&rem *****
&sv 45 D:\NCFRONT.TRN

&sv 44 ARTERIAL
select %45
&rem *****
&rem select for arterial as the from-arc
&rem save command to variable %39
&rem *****
&sv 39 "reselect for ( ( NAME1 eq '%2 %3' ) and ( NAME2 eq '%5 %6' ) )"

&label ARTERIAL_TURN
&rem *****
&rem reinitialize the phase direction
&rem variable to the first arterial phase dir.
&rem *****
&cv 46 7

&rem *****
&rem Copy phase direction (as in %7 -> %46) to %47
&rem *****
&value 47 %46
&cv 48 40

&rem *****
&rem Copy direction (as in %40) to %49
&rem *****
&value 49 40

&rem *****
&rem Outer Loop
&rem *****
&label ARTERIAL_OUT

&rem *****
&rem Inner Loop
&rem *****
&label ARTERIAL_IN
&rem *****
&rem Jump to direction label specified in %49
&rem if phase direction and direction match
&rem *****
&goto %49 &if &cn %47 %49

&label ARTERIAL_BACK_TO_LOOP
&cv 46 %46 + 1
&rem *****
&rem Get new value of phase direction into %47
&rem (as in %46 which has been incremented)
&rem *****
&value 47 %46

&rem *****
&rem %7 to %10 have phase directions for the
&rem arterial; jump to inner loop ARTERIAL_IN
&rem if any arterial phase directions are still
&rem to be processed
&rem *****
&goto ARTERIAL_IN &if &rn %46 7 10

&rem *****

```

```

&rem reinitialize arterial phase direction
&rem variable to the first arterial phase dir.
&rem *****
&cv 46 7
&value 47 %46
&cv 48 %48 + 1

&rem *****
&rem Get new value of direction into %49
&rem (as in %48 which has been incremented)
&rem *****
&value 49 %48

&rem *****
&rem jump to outer loop ARTERIAL_OUT
&rem if any directions are still
&rem to be processed
&rem *****
&goto ARTERIAL_OUT &if &rn %48 40 43

&sv 44 CROSS_STREET
select %45
&rem *****
&rem select for cross-street as the from-arc
&rem save command to variable %39
&rem *****
&sv 39 "reselect for ( ( NAME1 eq '%5 %6' ) and ( NAME2 eq '%2 %3' ) )"

&label CROSS_STREET_TURN
&rem *****
&rem reinitialize phase direction variable
&rem to the first arterial phase direction.
&rem *****
&cv 46 11

&rem *****
&rem Copy phase direction (as in %11 -> %46) to %47
&rem *****
&value 47 %46
&cv 48 40

&rem *****
&rem Copy direction (as in %40) to %49
&rem *****
&value 49 40

&rem *****
&rem Outer Loop
&rem *****
&label CROSS_STREET_OUT

&rem *****
&rem Inner Loop
&rem *****
&label CROSS_STREET_IN

&rem *****
&rem Jump to direction label specified in %49
&rem if phase direction and direction match
&rem *****
&goto %49 &if &cn %47 %49

```

```

&label CROSS_STREET_BACK_TO_LOOP
&cv 46 %46 + 1
&rem *****
&rem Get new value of phase direction into %47
&rem (as in %46 which has been incremented)
&rem *****
&value 47 %46

&rem *****
&rem %11 to %14 have phase directions for the
&rem cross-street; jump to inner loop
&rem CROSS_STREET_IN if any
&rem cross_street phase directions are still
&rem to be processed
&rem *****
&goto CROSS_STREET_IN &if &rn %46 11 14

&rem *****
&rem reinitialize cross-street phase direction
&rem variable to the first cross street phase dir.
&rem *****
&cv 46 11
&value 47 %46
&cv 48 %48 + 1

&rem *****
&rem Get new value of direction into %49
&rem (as in %48 which has been incremented)
&rem *****
&value 49 %48

&rem *****
&rem jump to outer loop CROSS_STREET_OUT
&rem if any directions are still
&rem to be processed
&rem *****
&goto CROSS_STREET_OUT &if &rn %48 40 43
q &return

&rem *****
&rem Process by each direction
&rem *****

&rem *****
&rem NORTH direction
&rem *****
&label NB
select %45

&rem *****
&rem execute command saved in %39
&rem *****
%39
&type "Reached NB"

&rem *****
&rem select all turn records which have matching
&rem orientation
&rem *****
reselect for ( ( AZIMUTH LT 45 ) or ( AZIMUTH GE 315 ) )

&rem *****

```

```

&rem  if then phase direction variable indicates
&rem  a thru turn then jump to label for thru
&rem  otherwise process for left turn
&rem  *****
&goto NB_THRU &if &cn %47 'THRU'

&rem  *****
&rem  left turn
&rem  *****
  reselect for ( ANGLE GE 45 )
  &goto UPDATE_RECORD

&label NB_THRU
  &rem  *****
  &rem  select all turn records which match the
  &rem  requirements of a thru direction of turn
  &rem  *****
  reselect for ( ( ANGLE LT 45 ) and ( ANGLE GE -45 ) )
  &goto UPDATE_RECORD

&rem  *****
&rem  SOUTH direction
&rem  *****
&label SB
&type "Reached SB"
select %45

&rem  *****
&rem  execute command saved in %39
&rem  *****
%39

&rem  *****
&rem  select all turn records which have matching
&rem  orientation
&rem  *****
reselect for ( ( AZIMUTH GE 135 ) and ( AZIMUTH LT 225 ) )

&rem  *****
&rem  if then phase direction variable indicates
&rem  a thru turn then jump to label for thru
&rem  otherwise process for left turn
&rem  *****
&goto SB_THRU &if &cn %47 'THRU'

&rem  *****
&rem  left turn
&rem  *****
  reselect for ( ANGLE GE 45 )
  &goto UPDATE_RECORD

&label SB_THRU
  &rem  *****
  &rem  select all turn records which match the
  &rem  requirements of a thru direction of turn
  &rem  *****
  reselect for ( ( ANGLE LT 45 ) and ( ANGLE GE -45 ) )
  &goto UPDATE_RECORD

&rem  *****
&rem  EAST direction
&rem  *****

```

```

&label EB
&type "Reached EB"
select %45

&rem *****
&rem   execute command saved in %39
&rem *****
%39

&rem *****
&rem   select all turn records which have matching
&rem   orientation
&rem *****
reselect for   ( ( AZIMUTH GE 45 ) or ( AZIMUTH LT 135 ) )

&rem *****
&rem   if then phase direction variable indicates
&rem   a thru turn then jump to label for thru
&rem   otherwise process for left turn
&rem *****
&goto EB_THRU &if &cn %47 'THRU'

&rem *****
&rem   left turn
&rem *****
reselect for   ( ANGLE GE 45 )
&goto UPDATE_RECORD

&label EB_THRU
&rem *****
&rem   select all turn records which match the
&rem   requirements of a thru direction of turn
&rem *****
reselect for   ( ( ANGLE LT 45 ) and ( ANGLE GE -45 ) )
&goto UPDATE_RECORD

&rem *****
&rem   WEST direction
&rem *****
&label WB
&type "Reached WB"
select %45

&rem *****
&rem   execute command saved in %39
&rem *****
%39

&rem *****
&rem   select all turn records which have matching orientation
&rem *****
reselect for   ( ( AZIMUTH GE 225 ) or ( AZIMUTH LT 315 ) )

&rem *****
&rem   if the phase direction variable indicates
&rem   a thru turn then jump to label for thru
&rem   otherwise process for left turn
&rem *****
&goto WB_THRU &if &cn %47 'THRU'

&rem *****
&rem   left turn

```

```

&rem *****
reselect for ( ANGLE GE 45 )
&goto UPDATE_RECORD

&label WB_THRU
&rem *****
&rem select all turn records which match the
&rem requirements of a thru direction of turn
&rem *****
reselect for ( ( ANGLE LT 45 ) and ( ANGLE GE -45 ) )
&goto UPDATE_RECORD

&rem *****
&rem Update the records in the turn table as processed
&rem *****
&label UPDATE_RECORD

&rem *****
&rem point %50 to the variable number for phase direc.
&rem *****
&value 50 46

&rem *****
&rem re-point %50 to the variable number for delay
&rem *****
&CV 50 ( %50 + 8 )

&rem *****
&rem store the delay in %50; the delay is in seconds
&rem then update the appropriate field in the turn
&rem table converting the delay into minutes
&rem *****
&VALUE 50 %50

&rem *****
&rem divide by 60 to get delay in minutes
&rem *****
calc TURNTIME = ( %50 / 60 )

&rem *****
&rem jump to the label (arterial or cross_street)
&rem to process other phase directions and delays
&rem *****
&goto %44_BACK_TO_LOOP

```

```

&rem *****
&rem   FileName:      AGENCIES.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        AGENCIES.COL
&rem   CalledBy:     d:\incman\pulldown\AIMS.COL
&rem   DataFiles:    None
&rem   Variables:    %39 -> As selected by user in AGENCIES.COL
&rem   Purpose:      Brings up the main menu of RESOURCE AGENCIES
&rem                  repeatedly in a loop
&rem *****

&rem *****
&rem   initialize the user response
&rem *****
&SV 39 ""

&rem *****
&rem   Conditional loop to repeatedly
&rem   pop up the RESOURCES main menu.
&rem *****
&LABEL CONTINUE
  POPUP AGENCIES.COL SML
  &GOTO CONTINUE &IF &NC %39 "QUIT"
  &RETURN

```



```

&rem 38 1 1 25 16 30
&rem      'A.I.M.S
&rem `INCIDENT`MANAGEMENT`RESOURCES
&rem
&rem POLICE`STATIONS
&rem FIRE`STATIONS
&rem HOSPITALS
&rem FLOODING`ZONES`AND`PUMPS
&rem PARK`AND`RIDES
&rem CHANGEABLE`MESSAGE`SIGNS
&rem AUTO`VEHICLE`ID`LOCATIONS
&rem SCHOOLS
&rem ZIP`CODES
&rem PHONE`AREAS
&rem REFERENCE`MARKERS
&rem
&rem QUIT
&rem EOF

```

```

&rem *****
&rem      FileName:      AGENCIES.COL
&rem      LastUpdate:   6/1/93
&rem      WrittenBy:    Nitin Vaidya
&rem      Calls:        ARC PLOT, PUMPS.SML, PNR.SML, ZIPCODES.SML
&rem                  PLOTARCS.SML
&rem      CalledBy:     AGENCIES.SML
&rem      DataFiles:    None
&rem      Variables:    %21 -> Name of base coverage/map
&rem                  %28 -> Copy of user selection
&rem                  %38 -> User selection from menu; changed to
&rem                        the name of directory D:\RESOURCE
&rem                  %39 -> Copy of user selection from menu;
&rem                        changed to the name of resource
&rem      Purpose:      This is the main menu for the RESOURCE
&rem                  database and invokes the user selected
&rem                  resource for the query system. There is
&rem                  base map or coverage common to most
&rem                  applications and 'point' coverages are
&rem                  superimposed on the top of the base map.
&rem                  There are some exceptions, however, and
&rem                  a different set of coverages are used
&rem                  for those resources.
&rem *****

```

```

&sv 39 %38
&sv 28 %38
&sv 38 d:\resource
&SV 21 D:\INCMAN
&SAVE FILEVAR 1 50

```

```

&rem *****
&rem      Jump to the label as per user selection
&rem *****
&jump %39

```

```

&label POLICE`STATIONS
      &sv 39 police
      &goto GOTO`ARC PLOT

&label FIRE`STATIONS

```

```

&sv 39 fire
&goto GOTO'ARC PLOT

&label HOSPITALS
&sv 39 hospital
&goto GOTO'ARC PLOT

&rem *****
&rem This is handled differently
&rem because it is not a 'point' coverage
&rem *****
&label FLOODING'ZONES'AND'PUMPS
&sv 39 pumps
&save filevar 1 50
ARC PLOT PUMPS
&goto QUIT

&rem *****
&rem This is handled differently
&rem because it is a different coverage
&rem *****
&label PARK'AND'RIDES
&sv 39 pnr
&save filevar 1 50
ARC PLOT PNR
&goto QUIT

&label CHANGEABLE'MESSAGE'SIGNS
&sv 39 cms
&goto GOTO'ARC PLOT

&label CELLULAR'PHONE'STATIONS
&sv 39 cps
&goto GOTO'ARC PLOT

&label AUTO'VEHICLE'ID'LOCATIONS
&sv 39 avil
&goto GOTO'ARC PLOT

&label SCHOOLS
&sv 39 schools
&goto GOTO'ARC PLOT

&rem *****
&rem This is handled differently
&rem because it is not a 'point' coverage
&rem *****
&label ZIP'CODES
&sv 39 zipcodes
&save filevar 1 50
ARC PLOT ZIPCODES
&goto QUIT

&label PHONE'AREAS
&sv 39 phone
&goto GOTO'ARC PLOT

&label REFERENCE'MARKERS
&sv 39 markers
&goto GOTO'ARC PLOT

&rem *****

```

```
&rem  Invokes ARC PLOT and runs PLOTARCS.SML
&rem  PLOTARCS.SML is common to all
&rem  resources except as specified
&rem  *****
&label GOTO`ARC PLOT
    &save filevar 1 50
    ARC PLOT PLOTARCS
    &goto QUIT

&label QUIT
    &return
```

```

&rem *****
&rem   FileName:      PLOTARCS.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        QUERYPLOT.COL
&rem   CalledBy:     AGENCIES.COL
&rem   DataFiles:    None
&rem   Variables:    %21 -> Name of base coverage\map
&rem                 %38 -> Name of directory under which
&rem                 the resource coverages reside
&rem                 %39 -> temporary for name of coverage
&rem   Purpose:      Runs in ARCPLOT and draws the map in
&rem                 color. Repeatedly invokes QUERYPLOT.COL for user
&rem                 queries. QUERYPLOT.COL is the common menu for
&rem                 most resources in this application
&rem *****

disp 4 3
setpage 1 1
clear
&run filevar
mapextent %21
lineset color.lin

&rem *****
&rem draw the base coverage specified
&rem by coverage %21
&rem *****
arclines %21 symbol

textcolor 0
markersymbol 8
markercolor 2

&rem *****
&rem draw the 'point' coverage specified in
&rem directory %38 and coverage name %39
&rem *****
points %38\%39

&rem *****
&rem Conditional loop to pop up the
&rem QUERYPLO.COLmenu for user selection
&rem *****
&label CONTINUE
  popup QUERYPLOT.COL sml
  &goto CONTINUE &if &ne %35 "QUIT"
q

```

```

&rem 35 1 1 45 14 34
&rem 'QUERY'FOR'%28
&rem
&rem BY'ATTRIBUTES
&rem
&rem SPATIAL'SELECT
&rem
&rem CLEAR'SELECTION
&rem
&rem ZOOM'IN
&rem
&rem ZOOM'OUT
&rem
&rem QUIT
&rem
&rem EOF

&rem *****
&rem   FileName:      QUERYPLO.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     PLOTARCS.SML
&rem   DataFiles:    None
&rem   Variables:    %35 -> User selection
&rem                %38 -> Name of directory for resources
&rem                %39 -> Name of resource coverage
&rem                %49 -> Users query
&rem   Purpose:      Allows the user to enter a query on the
&rem                current resource database. The selections
&rem                satisfying user query are highlighted on
&rem                the screen and the database is listed.
&rem                In future the query could be extended to
&rem                a menu driven system so that the user
&rem                need not remember the syntax for query
&rem *****

&jump %35

&rem *****
&rem User query is stored in %49, the query is performed
&rem on the database and the selections highlighted on
&rem the screen; simultaneously the database is listed
&rem on back screen.... Need to update this so that
&rem the data can be displayed on the main screen
&rem *****
&label BY'ATTRIBUTES
&response 49 "ENTER QUERY :"
  reselect %38\%39 points %49
  markercolor 5
  points %38\%39
  list %38\%39 points address zipcode phone_no

  &rem *****
  &rem clear the previous selection
  &rem *****
  aselect %38\%39 points %39_id gt 0
  &goto END'QUERY

&rem *****
&rem User selects objects using a mouse, the selections

```

```

&rem are highlighted on
&rem the screen; simultaneously the database is listed
&rem on back screen.... Need to update this so that
&rem the data can be displayed on the main screen
&rem *****
&label SPATIAL`SELECT
  reselect %38`%39 points BOX *
  markercolor 5
  points %38`%39
  list %38`%39 points address zipcode phone_no
  &goto END`QUERY

&label CLEAR`SELECTION
  &rem *****
  &rem clear the previous selection
  &rem *****
  aselect %38`%39 points %39_id gt 0
  markercolor 2

  &rem *****
  &rem Redraw previously highlighted
  &rem selections in normal color
  &rem *****
  clear
  arclines %21 symbol
  points %38`%39
  &goto QUIT

  &goto END`QUERY

&label ZOOM`IN
  mapextent *
  clear
  arclines %21 symbol
  points %38`%39
  &goto END`QUERY

&label ZOOM`OUT
  mapextent %21
  clear
  arclines %21 symbol
  points %38`%39
  &goto END`QUERY

&label END`QUERY
  &goto QUIT

&label QUIT
  &return

```

```

&rem *****
&rem   FileName:      PUMPS.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        QUERYPUM.COL
&rem   CalledBy:     AGENCIES.COL
&rem   DataFiles:    None
&rem   Variables:    %21 -> Name of base coverage/map
&rem                %38 -> Name of directory under which
&rem                the resource PUMP resides
&rem   Purpose:      Similar to PLOTARCS.SML. It draws the map in color
&rem                as well as draws the polygons for flood zones. Also
&rem                invokes QUERYPUM.COL repeatedly for user selection
&rem *****

```

```

disp 4 3
setpage 1 1
clear
&run filevar
mapextent %21
lineset color.lin

```

```

&rem *****
&rem draw the base coverage arcs
&rem *****
arclines %21 symbol
textcolor 0
linesymbol 4

```

```

&rem *****
&rem draw the polygons that represent the flood zones;
&rem this is where it differs from PLOTARCS.SML
&rem *****
polys %38\pumps

```

```

textcolor 0
markersymbol 8
markercolor 2

```

```

&rem *****
&rem draw the point locations for the pumps
&rem *****
points %38\pumps

```

```

&rem *****
&rem Conditional loop to pop up the
&rem QUERYPUM.COL menu for user selection
&rem *****
&label CONTINUE
  popup QUERYPUMP.COL sml
  &goto CONTINUE &if &ne %35 "QUIT"
q

```

```

&rem 35 1 1 56 18 23
&rem 'QUERY'FOR'%28
&rem
&rem PUMP'ATTRIBUTES
&rem
&rem PUMP'SPATIALLY
&rem
&rem FLOOD'ZONE'ATTRIBUTES
&rem
&rem FLOOD'ZONE'SPATIALLY
&rem
&rem CLEAR'SELECTION
&rem
&rem ZOOM'IN
&rem
&rem ZOOM'OUT
&rem
&rem QUIT
&rem
&rem EOF

```

```

&rem *****
&rem   FileName:      QUERYPUM.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     PUMPS.SML
&rem   DataFiles:    None
&rem   Variables:    %35 -> User selection from menu
&rem                %38 -> Name of directory
&rem                %39 -> Name of the coverage/map for pumps and
&rem                    flood zones
&rem                %49 -> The query by user
&rem   Purpose:      Similar to QUERYPLO.COL. Allows the user to
&rem                perform queries on both PUMPS and FLOOD ZONES
&rem                databases, either graphically or by text input
&rem                The selections are highlighted and listed on the
&rem                back screen
&rem *****

```

```

linesymbol 1
&jump %35

```

```

&rem *****
&rem   User query on PUMP attributes, stored in %49,
&rem   is performed and the resulting selections highlighted
&rem   and the database listed on the back screen
&rem *****
&label PUMP'ATTRIBUTES
  &response 49 "ENTER QUERY :"
  reselect %38%\%39 points %49
  markercolor 5
  points %38%\%39
  list %38%\%39 points
  aselect %38%\%39 points %39_id gt 0
  &goto END'QUERY

```

```

&rem *****
&rem   User query on FLOOD ZONE attributes, stored in %49,
&rem   is performed and the resulting selections highlighted
&rem   and the database listed on the back screen
&rem *****

```



```

&label FLOOD'ZONE'ATTRIBUTES
  &response 49 "ENTER QUERY : "
  reselect %38%\%39 polys %49
  linesymbol 5
  polys %38%\%39
  list %38%\%39 polys
  aselect %38%\%39 polys %39_id gt 0
  &goto END'QUERY

&rem *****
&rem Spatial query on PUMP attributes, stored in %49,
&rem is performed and the resulting selections highlighted
&rem and the database listed on the back screen
&rem *****
&label PUMP'SPATIALLY
  reselect %38%\%39 points BOX *
  markercolor 5
  points %38%\%39
  list %38%\%39 points
  aselect %38%\%39 points %39_id gt 0
  &goto END'QUERY

&rem *****
&rem Spatial query on FLOOD ZONES, stored in %49,
&rem is performed and the resulting selections highlighted
&rem and the database listed on the back screen
&rem *****
&label FLOOD'ZONE'SPATIALLY
  reselect %38%\%39 polys BOX *
  linesymbol 5
  polys %38%\%39
  list %38%\%39 polys
  aselect %38%\%39 polys %39_id gt 0
  &goto END'QUERY

&label ZOOM'IN
  mapextent *
  clear
  &goto CLEAR'SELECTION

&label ZOOM'OUT
  mapextent %21
  clear
  &goto CLEAR'SELECTION

&rem *****
&rem Clear previous selections and redraw
&rem *****
&label CLEAR'SELECTION
  aselect %38\pumps points pumps_id gt 0
  aselect %38\pumps polys pumps_id gt 0
  clear
  arclines %21 symbol
  linesymbol 4
  polys %38\pumps
  markercolor 2
  points %38\pumps
  &goto END'QUERY

&label END'QUERY
  &goto QUIT

```

```
&label QUIT  
&return
```

```

&rem *****
&rem   FileName:      PNR.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        QUERYPNR.COL
&rem   CalledBy:     AGENCIES.COL
&rem   DataFiles:    None
&rem   Variables:    %38 -> Name of directory under which
&rem                 the PNR resource resides.
&rem   Purpose:      Similar to PLOTARCS.SML, but uses a different
&rem                 base coverage for PNR. Database has information on
&rem                 both transitways and Park-N-Ride locations
&rem *****

disp 4 3
setpage 1 1
clear
&run filevar

&rem *****
&rem   draw the coverage/maps
&rem *****
mapextent %38\pnr
linesymbol 3
arcs %38\pnr

&rem *****
&rem   highlight the transitway
&rem *****
linesymbol 8
reselect %38\pnr arcs A_PM_H_BV GT 0
arcs %38\pnr

&rem *****
&rem   draw PNR locations
&rem *****
linesymbol 3
aselect %38\pnr arcs pnr_id gt 0
textcolor 0
markersymbol 8
markercolor 2
points %38\pnr

&rem *****
&rem   draw street names
&rem *****
annotext %38\pnr

&label CONTINUE
  popup QUERYPNR.COL sml
  &goto CONTINUE &if &ne %35 "QUIT"
q

```

```

&rem 35 1 1 45 16 34
&rem 'QUERY' FOR 'TRANSITWAY' AND 'PNR
&rem
&rem TRANSITWAY' ATTRIBUTES
&rem
&rem PARK 'AND' RIDE' ATTRIBUTES
&rem
&rem SPATIAL 'SELECT' PNR
&rem
&rem CLEAR 'SELECTION
&rem
&rem ZOOM 'IN
&rem
&rem ZOOM 'OUT
&rem
&rem QUIT
&rem
&rem EOF

&rem *****
&rem   FileName:      QUERYPNR.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        TRANSATT.COL
&rem   CalledBy:     PNR.SML
&rem   DataFiles:    None
&rem   Variables:    %35 -> User selection of query type
&rem                %38 -> Directory under which PNR resources reside
&rem                %39 -> Name of PNR resource coverage/map
&rem                %49 -> User entered query
&rem   Purpose:      Similar to QUERYPLO.COL. Allows queries on both
&rem                transitway attributes as well as Park-n-Rides.
&rem                The selections are highlighted upon query and
&rem                the data is listed on back screen. Spatial
&rem                selections are only possible on PNRs
&rem *****

&jump %35

&rem *****
&rem   Query on Transitway database is performed through
&rem   another menu TRANSATT.COL
&rem *****
&label TRANSITWAY'ATTRIBUTES
  &label CONTINUE
  popup transatt.col sml
  &rem *****
  &rem   DANGER !!! this menu and TRANSATT.COL
  &rem   both use same user selection variable
  &rem   Reset variable after this loop has
  &rem   been exited.
  &rem *****
  &goto CONTINUE &if &ne %35 "QUIT"
  &sv 35 ""
  &goto END'QUERY

&rem *****
&rem   Query on PNR database. The user response is stored
&rem   in %49 and the query performed to highlight selections
&rem   and list appropriate data
&rem *****
&label PARK'AND'RIDE'ATTRIBUTES

```

```

&response 49 "ATTRIBUTE : "
reselect %38\pnr points %49
markercolor 5
points %38\pnr
list %38\%39 points name capacity daily_use opertn_in opertn_out
aselect %38\%39 points %39_id gt 0
&goto END'QUERY

&rem *****
&rem Spatial query on PNR database.
&rem The query is performed to highlight selections
&rem and list appropriate data
&rem *****
&label SPATIAL'SELECT'PNR
reselect %38\%39 points BOX *
markercolor 5
points %38\%39
list %38\%39 points name capacity daily_use opertn_in opertn_out
aselect %38\%39 points %39_id gt 0
&goto END'QUERY

&label ZOOM'IN
mapextent *
clear
&goto CLEAR'SELECTION

&label ZOOM'OUT
mapextent %38\pnr
clear
&goto CLEAR'SELECTION

&rem *****
&rem Clear any pervious selections and redraw the map
&rem *****
&label CLEAR'SELECTION
clear

&rem *****
&rem draw all street segments
&rem *****
linesymbol 3
aselect %38\pnr arcs pnr_id gt 0
arcs %38\pnr

&rem *****
&rem highlight transitway
&rem *****
linesymbol 8
reselect %38\pnr arcs a_pm_h_bv gt 0
arcs %38\pnr
aselect %38\pnr arcs a_pm_h_bv gt 0

&rem *****
&rem draw Park-n-Ride locations
&rem *****
textcolor 0
markersymbol 8
markercolor 2
points %38\pnr

&rem *****
&rem write street names

```

```
&rem *****  
annotext %38\pnr  
&goto END\QUERY  
  
&label END\QUERY  
&goto QUIT  
  
&label QUIT  
&return
```

```

&rem 35 1 1 50 10 29
&rem 'QUERY' 'TRANSITWAY' 'ATTRIBUTES'
&rem
&rem AM'PEAK' HOUR 'BUS' 'VEHICLES
&rem AM'PEAK' PERIOD 'BUS' 'PERSONS
&rem PM'PEAK' HOUR 'VAN' 'VEHICLES
&rem PM'PEAK' PERIOD 'VAN' 'PERSONS
&rem DAILY'TOTAL' 'CAR' 'VEHICLES
&rem DAILY'TOTAL' 'CAR' 'PERSONS
&rem
&rem QUIT
&rem EOF

```

```

&rem *****
&rem   FileName:      TRANSATT.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     QUERYPNR.COL
&rem   DataFiles:    None
&rem   Variables:    %35 -> User selection from menu
&rem                %38 -> Name of directory where PNR resource
&rem                resides
&rem   Purpose:      This allows user to perform queries on the
&rem                transitway database. When the user selects
&rem                a query type the result is listed on the back
&rem                screen. This is because there is only one
&rem                set of data in all the records matching the
&rem                transitway in this database and the data is
&rem                effectively hard coded. Not all possible
&rem                queries are shown in the menu.
&rem                Record number 62 is selected all the time
&rem                so clearance of previous selection is not
&rem                a requirement.
&rem *****

```

```

aselect %38\pnr arcs pnr_id gt 0
reselect %38\pnr arcs pnr_ eq 62
&jump %35

```

```

&label AM'PEAK' HOUR 'BUS' 'VEHICLES
  list %38\pnr arcs A_AM_H_BV
  &goto QUIT

```

```

&label AM'PEAK' PERIOD 'BUS' 'PERSONS
  list %38\pnr arcs A_AM_P_BV
  &goto QUIT

```

```

&label PM'PEAK' HOUR 'VAN' 'VEHICLES
  list %38\pnr arcs A_PM_H_VV
  &goto QUIT

```

```

&label PM'PEAK' PERIOD 'VAN' 'PERSONS
  list %38\pnr arcs A_PM_P_VP
  &goto QUIT

```

```

&label DAILY'TOTAL' 'CAR' 'VEHICLES
  list %38\pnr arcs A_D_CV
  &goto QUIT

```

```

&label DAILY'TOTAL' 'CAR' 'PERSONS
  list %38\pnr arcs A_D_CP

```

```
&goto QUIT  
&label QUIT  
&return
```



```

&rem *****
&rem   FileName:      ZIPCODES.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:         QUERYZIP.COL
&rem   CalledBy:     AGENCIES.COL
&rem   DataFiles:    None
&rem   Variables:    %21 -> Name of base coverage/map
&rem                 %38 -> Name of directory where ZIPCODES map
&rem                 %39 -> Stores 'ZIPCODES'
&rem   Purpose:      Similar to PLOTARCS.SML, except that zipcodes
&rem                 make a polygon coverage and attribute table.
&rem                 This draws the base map and zipcodes on the
&rem                 screen and repeatedly calls QUERYZIP.COL for
&rem                 queries on the zipcodes database.
&rem *****

```

```

disp 4 3
setpage 1 1
clear
&run filevar

```

```

&rem *****
&rem draw base map
&rem *****
mapextent %21
lineset color
arclines %21 symbol

```

```

&rem *****
&rem draw zipcode areas
&rem *****
textcolor 0
linesymbol 4
polys %38\%39
annotext %38\%39

```

```

&rem *****
&rem repeatedly call QUERYZIP.COL
&rem *****
&label CONTINUE
  popup QUERYZIP.COL sml
  &goto CONTINUE &if &ne %35 "QUIT"
q

```

```

&rem 35 1 1 45 14 34
&rem `QUERY`FOR`%28
&rem
&rem BY`ATTRIBUTES
&rem
&rem SPATIAL`SELECTION
&rem
&rem CLEAR`SELECTION
&rem
&rem ZOOM`IN
&rem
&rem ZOOM`OUT
&rem
&rem QUIT
&rem EOF

&rem *****
&rem   FileName:      QUERYZIP.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     ZIPCODES.SML
&rem   DataFiles:    None
&rem   Variables:    %35 -> User selection from menu
&rem                %38 -> Name of directory where ZIPCODES resides
&rem                %39 -> Variable stores `ZIPCODES`
&rem                %49 -> User query as entered
&rem   Purpose:      This is similar to QUERYPLO.COL except that the
&rem                queries here are on polygon type map. The user
&rem                may enter the attribute required and the selected
&rem                polygon representing zipcode will be highlighted
&rem                and its attributes will be displayed.
&rem *****

&jump %35

&rem *****
&rem   Read user query into %49, perform
&rem   query and show/list the results
&rem *****
&label BY`ATTRIBUTES
  &response 49 "ENTER QUERY :"
  reselect %38\%39 polys %49
  linesymbol 5
  polys %38\%39
  list %38\%39 polys
  aselect %38\%39 polys %39_id gt 0
  &goto QUIT

&rem *****
&rem   Spatial query, perform
&rem   query and show/list the results
&rem *****
&label SPATIAL`SELECTION
  reselect %38\%39 polys BOX *
  linesymbol 5
  polys %38\%39
  list %38\%39 polys
  aselect %38\%39 polys %39_id gt 0
  &goto QUIT

&label ZOOM`IN

```

```

mapextent *
clear
&goto CLEAR`SELECTION

&label ZOOM`OUT
mapextent d:\incman
clear
&goto CLEAR`SELECTION

&rem *****
&rem  Clear previous selections and
&rem  redraw base map and zipcode areas
&rem *****
&label CLEAR`SELECTION
  aselect %38\zipcodes polys zipcodes_id gt 0
  clear
  arclines d:\incman symbol
  linesymbol 4
  polys %38\zipcodes
  annotext %38\zipcodes
  &goto QUIT

&label QUIT
&return

```

```
&rem PLOTSW.BAT  
cd\  
CD SWINC  
CD PULLDOWN  
arc plotsw
```

```
&rem *****
&rem * PLOTSW.SML
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTSW.BAT
&rem * CALLS: PLOTLOAD.SML, FILEVAR.SML, PLOTEXIT.COL
&rem * OBJECTIVE: SETS ENVIRONMENT TO ENTER/EXIT PROGRAM
&rem *****
```

```
clear
&SV 1 D:\SWINC
&SAVE filevar.sml 1 1
ARC PLOT PLOTLOAD
```

```
CLEAR
POPUP PLOTEXIT.COL SML
  &GOTO SKIP &IF &EQ %50 "DOS"
  &RETURN
```

```
&LABEL SKIP
  Q
  &RETURN
```

```
&REM 50 1 5 10 5 10
&REM `EXIT`TO`
&REM `=====`
&REM DOS
&REM
&REM ARC/INFO
&REM EOF
```

```
&rem *****
&rem * PLOTEXIT.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTSW.SML
&rem * CALLS: none
&rem * OBJECTIVE: ALLOWS USER TO SELECT WHERE TO QUIT TO
&rem *****
```

```
&RETURN
```

```

&rem *****
&rem * PLOTLOAD.SML
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTSW.SML
&rem * CALLS: FILEVAR.SML, PLOTDRAW.SML, PLOTMENU.MNU
&rem * OBJECTIVE: SET GRAPHICS ENVIRONMENT
&rem *          PULL IN MAIN MENU
&rem *****

&run filevar
DISP 4 3
SETPAGE 1 1
LINESET COLOR.LIN
LINESYMBOL 8
MAPEX %1
addresscov %1
&RUN PLOTDRAW

&sv 2 n
&SV 6 F
&label WHILE
    &goto EXIT &if &eq %2 "Y"
    popup plotmenu.mnu SML
    &goto WHILE

&label EXIT
q
&RETURN

```

```
&rem *****
&rem * PLOTDRAW.SML
&rem * LAST UPDATE: 5-6-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTLOAD.SML, REDRAW2.SML
&rem * CALLS: SCALE.SML, SOURCE.SML, TTI.SML
&rem * OBJECTIVE: DRAWS THE BASE MAP
&rem *****
```

```
CLEAR
LINESET COLOR.LIN
ARCS %1
whereis 9 southwest
whereis 9 loop 610w
whereis 9 loop 610s
ANNOTEXT %1
MARKERSET COLOR.SW
SHADESET COLOR.SHD
SHADESYMBOL 100
&run scale
&run source
&run tti
&RETURN
```



```
&rem *****
&rem * TTI.SML
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTDRAW.SML, PLOTDISP.SML, STR_NAM.SML, STR_LOC.SML,
&rem *             REDRAW.SML, PLOTINZ1.SML, PLOTINZ2.SML
&rem * CALLS: none
&rem * OBJECTIVE: DRAWS TTI IN LOWER RIGHT HAND CORNER OF SCREEN
&rem *****
```

```
shadeset color.shd
shadesymbol 100
patch 8.85 .15 9.35 .45
move 8.9 .2
textsize .2 .1
  text 'TTI'
&return
```

```
&rem *****
&rem * SOURCE.SML
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY KELLY PARMA
&rem * CALLED BY: PLOTDRAW.SML, PLOTDISP.SML, STR_NAM.SML, PLOTINZ1.SML,
&rem *           STR_LOC.SML, REDRAW.SML, PLOTINZ2.SML
&rem * CALLS: SOURCE.TXT
&rem * OBJECTIVE: DRAWS SOURCE OF MAP TEXT TO SCREEN
&rem *****
```

```
shadeset color.shd
shadesymbol 100
patch 0 0 1.85 .3
move 0 .1
textsize .13 .05
  textfile source.txt
```

```
&rem *****
&rem * SCALE.SML
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTDRAW.SML, PLOTDISP.SML, STR_NAM.SML
&rem * CALLS: none
&rem * OBJECTIVE: DRAWS SCALE LEGEND ON FULL SCREEN
&rem *****
```

```
shadeset color.shd
shadesymbol 100
patch 2.9 0 4.6 .75
```

```
box 3 .4 4.443 .5
line 3 .3 3 .4
line 3.481 .3 3.481 .5
line 3.962 .3 3.962 .5
line 4.443 .3 4.443 .4
move 3.1 .6
textsize .13 .05
  text 'Scale = 1:131,730'
textsize .11 .05
move 3.004 .15
  text 0
move 3.485 .15
  text 1
move 3.966 .15
  text 2
move 4.447 .15
  text 3
move 3.1 0
textsize .15 .07
  text miles
&return
```

```

&REM 3 2 1 1 4 75
&REM          'SOUTHWEST\FREEWAY\INCIDENT\SYSTEM'
&REM
=====
&REM  DISPLAY\INCIDENTS          LIST\INCIDENTS          UTILITIES          QUIT
&REM
&REM EOF
&JUMP %3

&rem *****
&rem * PLOTMENU.MNU
&rem * LAST UPDATE: 5-6-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTLOAD.SML
&rem * CALLS: PLOTWHAT.COL, PLOTCUML.SML, PLOTMNTN.COL, PLOTYEAR.COL,
&rem *          PLOTDISP.SML
&rem *          PLOTHOWW.COL
&rem *          PLOTUTIL.COL
&rem *          PLOTQUIT.COL
&rem * OBJECTIVE: SETS THE MAIN MENU FOR THE USER
&rem *****

&LABEL DISPLAY\INCIDENTS
  POPUP PLOTWHAT.COL SML
    &GOTO REPEAT &IF &EQ %2 "Y"
  &GOTO TOTALINC &IF &EQ %20 "CUMULATIVE"
  POPUP PLOTMNTN.COL SML
    &GOTO REPEAT &IF &EQ %2 "Y"
  POPUP PLOTYEAR.COL SML
    &GOTO REPEAT &IF &EQ %2 "Y"
  &RUN PLOTDISP
  &GOTO REPEAT

&LABEL LIST\INCIDENTS
  POPUP PLOTMNTN.COL SML
    &GOTO REPEAT &IF &EQ %2 "Y"
  POPUP PLOTYEAR.COL SML
    &GOTO REPEAT &IF &EQ %2 "Y"
  POPUP PLOTHOWW.COL SML
  &GOTO REPEAT

&LABEL UTILITIES
  POPUP PLOTUTIL.COL SML
    &GOTO QUIT2 &IF &EQ %46 "YES"
  &GOTO REPEAT

&LABEL TOTALINC
  &RUN PLOTCUML
  &GOTO REPEAT

&LABEL REPEAT
  &SV 2 N
  &RETURN

&LABEL QUIT
  popup plotquit.col sml
  &RETURN

&LABEL QUIT2
  &RETURN

```

```
&REM 20 1 5 5 7 12
&REM `DISPLAY
&REM `=====
&REM MONTHLY
&REM
&REM CUMULATIVE
&REM
&REM CANCEL
&REM EOF
```

```
&rem *****
&rem * PLOTWHAT.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTMENU.MNU
&rem * CALLS: none
&rem * OBJECTIVE: OPTION GIVEN TO CHOOSE WHETHER TO DISPLAY A MONTH'S DATA ON
&rem * THE SCREEN OR CUMULATIVE DATA
&rem *****
```

```
&GOTO QUIT &IF &EQ %20 "CANCEL"
&RETURN
```

```
&LABEL QUIT
&SV 2 Y
&RETURN
```

```
&REM 7 1 6 15 16 7
&REM `MONTH
&REM `=====
&REM JAN
&REM FEB
&REM MAR
&REM APR
&REM MAY
&REM JUN
&REM JUL
&REM AUG
&REM SEP
&REM OCT
&REM NOV
&REM DEC
&REM
&REM CANCEL
&REM EOF
```

```
&rem *****
&rem * PLOTMONTH.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTMENU.MNU
&rem * CALLS: none
&rem * OBJECTIVE: ALLOWS USER TO SELECT MONTH FOR INCIDENT LOCATION DISPLAY
&rem *****
```

```
&GOTO EXIT &IF &EQ %7 "CANCEL"
&SV 6 T
&RETURN
```

```
&LABEL EXIT
&SV 6 F
&SV 2 Y
&RETURN
```

```
&REM 15 1 5 5 9 8
&REM `YEAR
&REM `=====
&REM 89
&REM 90
&REM 91
&REM 92
&REM 93
&REM
&REM CANCEL
&REM EOF
```

```
&rem *****
&rem * PLOTYEAR.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTMENU.MNU
&rem * CALLS: none
&rem * OBJECTIVE: ALLOWS USER TO SELECT YEAR FOR INCIDENT LOCATION DISPLAY
&rem *****
```

```
&GOTO EXIT &IF &EQ %15 "CANCEL"
&SV 6 T
&RETURN
```

```
&LABEL EXIT
&SV 6 F
&SV 2 Y
&RETURN
```

```
&rem 25 1 5 45 7 10
&rem 'QUIT
&rem '=====
&rem
&rem QUIT
&REM
&rem CANCEL
&REM
&REM EOF
&JUMP %25
```

```
&rem *****
&rem * PLOTQUIT.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTMENU.MNU
&rem * CALLS: none
&rem * OBJECTIVE: ALLOWS USER TO CHOOSE WHETHER TO QUIT OR RETURN TO MENU
&rem *****
```

```
&LABEL QUIT
&SV 2 Y
&RETURN
```

```
&LABEL CANCEL
&SV 2 N
&RETURN
```



```
&REM 21 1 5 5 6 12
&REM 'SELECT'BY:
&REM '=====
&REM
&REM LOCATION
&REM
&REM POINTING
&REM EOF
&jump %21
```

```
&rem *****
&rem * PLOTHOWW.COL
&rem * LAST UPDATE: 5-6-93
&rem * CALLED BY: PLOTMENU.MNU
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLS: PLOTLIST.COL, STR_NAM.SML
&rem *          PLOTPONT.COL, STR_LOC.SML
&rem * OBJECTIVE: ALLOWS USER TO DETERMINE HOW TO CHOOSE THE INCIDENT
&rem *          LOCATION TO LIST THE INCIDENTS.
&rem *****
```

```
&label LOCATION
  popup plotlist.col sml
  &GOTO END &IF &EQ %2 Y
  &run str_nam
  &goto END
```

```
&label POINTING
  &SV 45 "YES"
  popup plotpont.col sml
  &GOTO END &IF &EQ %45 NO
  &run str_loc
  &goto END
```

```
&label END
  &sv 6 F
  &RETURN
```

```

&REM 22 1 5 5 16 19
&REM 'STREET'NAME
&REM '-----'
&REM CANCEL
&REM '-----'
&REM AIRPORT
&REM ALABAMA
&REM ALMEDA
&REM AUSTIN
&REM BEECHNUT
&REM BELLAIRE
&REM BELLFORT
&REM BELTWAY '8W
&REM BISSONNET
&REM BUFFALO 'SPDWAY
&REM CHIMNEY 'ROCK
&REM COMMERCE PARK
&REM CORPORATE
&REM CRAWFORD
&REM DUNLAVY
&REM EDLOE
&REM ELGIN
&REM FANNIN
&REM FONDREN
&REM FOUNTAINVIEW
&REM GESSNER
&REM GRAY
&REM GREENBRIAR
&REM GULF 'FWY
&REM HARWIN
&REM HAZARD
&REM HILLCROFT
&REM KIRBY
&REM KIRKWOOD
&REM LA 'BRANCH
&REM LARKWOOD
&REM LOOP '610W
&REM MAIN
&REM MANDELL
&REM MC 'GOWEN
&REM MONTROSE
&REM NEWCASTLE
&REM NORTH 'FWY
&REM RICE
&REM SAN 'JACINTO
&REM SAVOY
&REM SH '6
&REM SHEPHERD
&REM SOUTH 'FWY
&REM SPUR '257
&REM SUGAR 'CREEK
&REM SUGAR 'GROVE
&REM SWEETWATER
&REM TIMMONS
&REM WESLAYAN
&REM WESTPARK
&REM WHEELER
&REM WILCREST
&REM WILLIAM 'TRACE
&REM WOODHEAD
&REM EOF

```

```
&rem *****
&rem * PLOTLIST.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: NITIN VAIDYA, KELLY PARMA
&rem * CALLED BY: PLOTHOWW.COL
&rem * CALLS: none
&rem * OBJECTIVE: SELECTS THE STREET TO LIST INCIDENT DATA FOR A MONTH
&rem *****

&GOTO EXIT &IF &EQ %22 "CANCEL"
&return

&LABEL EXIT
  &SV 2 Y
  &RETURN
```

```

&rem *****
&rem * STR_NAM.SML
&rem * LAST UPDATE: 5-6-93
&rem * WRITTEN BY: NITIN VAIDYA
&rem * CALLED BY: PLOTHOWW.COL
&rem * CALLS: FILEVAR.SML, SCALE.SML, SOURCE.SML, TTI.SML
&rem *          SWTITLE.TXT, SWLEG.TXT, C:\SWBAK51\PULLDOWN\COUNTS\%7%15CNT.TXT
&rem * OBJECTIVE: 1) DISPLAYS SELECTED INCIDENT LOCATION TO SCREEN
&rem *                2) LISTS EACH INCIDENT IN WINDOW ON SCREEN
&rem *****

```

```

&RUN FILEVAR
DISP 4 3
SETPAGE 1 1
LINESET COLOR.LIN
LINESYMBOL 8
mapex %1
addresscov %1
ARCS %1
WHEREIS 9 SOUTHWEST
WHEREIS 9 LOOP 610W
WHEREIS 9 LOOP 610S
ANNOTEXT %1
TEXTSIZE .1 .15
MOVE 1 6.5
    TEXTFILE SWTITLE.TXT
MOVE 3.5 6.25
    TEXT %7.19%15
&run scale
&run source
&run tti

```

```

MARKERSET COLOR.SW
SHADESET COLOR.SHD
SHADESYMBOL 100
PATCH 6.8 0.2 8.7 2.2
KEYBOX 0 0
KEYPOSITION 7. 2.
BOX 6.8 0.2 8.7 2.2
TEXTSIZE .11 .05
KEYMARKER SWLEG.TXT NOBOX

```

```

&open D:\SWINC\PULLDOWN\COUNTS\%7%15CNT.TXT NOTOPEN

```

```

&label READDATA
    &read 18 NOTREAD
    &extract 16 18 1
    &extract 17 18 2
    &extract 19 18 3
    &goto CROSSTREET &IF &CN %22 %17
    &goto READDATA

```

```

&label CROSSTREET
    &goto LOWINCIDENT &if &rn %16 0 4
    &goto LOWMEDINCID &if &rn %16 5 9
    &goto MEDINCIDENT &if &rn %16 10 14
    &goto MEDHIHINCID &if &rn %16 15 19
    &goto HIHINCIDENT &if &rn %16 20 200

```

```

&label LOWINCIDENT
    whereis 5 SOUTHWEST\FWY/%17%19

```

```

&goto SHOWLOC

&LABEL LOWMEDINCID
WHEREIS 30 SOUTHWEST\FWY/%17%19
&GOTO SHOWLOC

&label MEDINCIDENT
whereis 72 SOUTHWEST\FWY/%17%19
&goto SHOWLOC

&LABEL MEDHIHINCID
WHEREIS 48 SOUTHWEST\FWY/%17%19
&GOTO SHOWLOC

&label HIHINCIDENT
whereis 100 SOUTHWEST\FWY/%17%19
&goto SHOWLOC

&rem *****
&rem * To get to the MAP database of incidents for streets and
&rem * display incident data (graphics and text) the following
&rem * routine is used.
&rem
&rem * The said routine joins STREETNA.DBF data file with the
&rem * MAP datafile for the MonthYear (%7%15). ARCPLOT is tricked
&rem * into believing that the MAP data file is actually the AAT of
&rem * coverage MonthYear. The join takes place over the CORRECTLOC
&rem * field which has field names common to the two files. This
&rem * field has the converted addresses of street numbers into
&rem * nearest intersection; to ensure uniformity.
&rem
&rem * Thereafter a selection is performed on the joined database
&rem * using the name of the desired street intersection.
&rem
&rem * The relevant data is then written to a textfile for display
&rem
&rem * At the end the selection is cleared for further queries.
&rem *****

&label SHOWLOC
JOIN D:\SWINC\PULLDOWN\STREETNA D:\SWINC\PULLDOWN\%7%15\AAT CORRECTLOC
reselect D:\SWINC\PULLDOWN\%7%15 arcs correctloc cn '%17'
&openw LOCFILE
&write "LOCATION: %17 %19"
&write "INCIDENTS: %16"
list D:\SWINC\PULLDOWN\%7%15 arcs location fwydir section date toh
&write
&write "Click Anywhere to EXIT"
&closew
popup LOCFILE 24 1 1 1 10 60
clearselect
&goto READDATA

&label NOTREAD
&close showinc
&goto ENDPROG

&label NOTOPEN
&type "SHOWINC could not be opened"
popup july90.col sml
&goto ENDPROG

```

```
&label ENDPROG  
&return
```

```

&rem *****
&rem * STR LOC.SML
&rem * LAST UPDATE: 5-6-93
&rem * WRITTEN BY: NITIN VAIDYA
&rem * CALLED BY: PLOTHOWW.COL
&rem * CALLS: FILEVAR.SML, SOURCE.SML, TTI.SML
&rem *          SWTITLE.TXT, SWLEG.TXT, C:\SWBAK51\PULLDOWN\COUNTS\%7%15CNT.TXT
&rem * OBJECTIVE: 1) DISPLAYS INCIDENT LOCATION FOR LOCATION SELECTED
&rem *                2) LISTS INDIVIDUAL INCIDENTS IN WINDOW
&rem *****

```

```

&RUN FILEVAR
LINESET COLOR.LIN
LINESYMBOL 8
addresscov %1
ARCS %1
WHEREIS 9 SOUTHWEST
WHEREIS 9 LOOP 610W
WHEREIS 9 LOOP 610S
ANNOTEXT %1
TEXTSIZE .1 .15
MOVE 1 6.5
    TEXTFILE SWTITLE.TXT
MOVE 3.5 6.25
    TEXT %7.19%15
&run source
&run tti

```

```

MARKERSET COLOR.SW
SHADESET COLOR.SHD
SHADESYMBOL 100
PATCH 6.8 0.2 8.7 2.2
KEYBOX 0 0
KEYPOSITION 7. 2.
BOX 6.8 0.2 8.7 2.2
TEXTSIZE .11 .05
KEYMARKER SWLEG.TXT NOBOX

```

```

&open D:\SWINC\PULLDOWN\COUNTS\%7%15CNT.TXT NOTOPEN

```

```

&label READDATA
    &read 18 NOTREAD
    &extract 16 18 1
    &extract 17 18 2
    &extract 19 18 3
    &goto CROSSTREET &IF &CN %22 %17
    &goto READDATA

```

```

&label CROSSTREET
    &goto LOWINCIDENT &if &rn %16 0 4
    &goto LOWMEDINCID &if &rn %16 5 9
    &goto MEDINCIDENT &if &rn %16 10 14
    &goto MEDHIHINCID &if &rn %16 15 19
    &goto HIHINCIDENT &if &rn %16 20 200

```

```

&label LOWINCIDENT
    whereis 5 SOUTHWEST'FWY/%17%19
    &goto SHOWLOC

```

```

&LABEL LOWMEDINCID
    WHEREIS 30 SOUTHWEST'FWY/%17%19

```

```

&GOTO SHOWLOC

&label MEDINCIDENT
  whereis 72 SOUTHWEST\FWY/%17%19
  &goto SHOWLOC

&LABEL MEDHIHINCID
  WHEREIS 48 SOUTHWEST\FWY/%17%19
  &GOTO SHOWLOC

&label HIHINCIDENT
  whereis 100 SOUTHWEST\FWY/%17%19
  &goto SHOWLOC

&rem *****
&rem * To get to the MAP database of incidents for streets and
&rem * display incident data (graphics and text) the following
&rem * routine is used.
&rem
&rem * The said routine joins STREETNA.DBF data file with the
&rem * MAP datafile for the MonthYear (%7%15). ARCPLOT is tricked
&rem * into believing that the MAP data file is actually the AAT of
&rem * coverage MonthYear. The join takes place over the CORRECTLOC
&rem * field which has field names common to the two files. This
&rem * field has the converted addresses of street numbers into
&rem * nearest intersection; to ensure uniformity.
&rem
&rem * Thereafter a selection is performed on the joined database
&rem * using the name of the desired street intersection.
&rem
&rem * The relevant data is then written to a textfile for display
&rem
&rem * At the end the selection is cleared for further queries.
&rem *****

&label SHOWLOC
  JOIN D:\SWINC\PULLDOWN\STREETNA D:\SWINC\PULLDOWN\%7%15\AAT CORRECTLOC
  reselect D:\SWINC\PULLDOWN\%7%15 arcs correctloc cn '%17'
  &openw LOCFILE
    &write "LOCATION: %17 %19"
    &write "INCIDENTS: %16"
    list D:\SWINC\PULLDOWN\%7%15 arcs location fwydir section date toh
    &write
    &write "Click Anywhere to EXIT"
  &closew
  popup LOCFILE 24 1 1 1 10 60
  clearselect
  &goto READDATA

&label NOTREAD
  &close showinc
  &goto ENDPROG

&label NOTOPEN
  &type "SHOWINC could not be opened"
  popup july90.col sml
  &goto ENDPROG

&label ENDPROG
  &return

```



```

&rem 28 1 5 5 4 18
&rem      `CHOOSE:
&REM `-----
&REM
&REM ZOOM`AND`SELECT
&REM EOF
&JUMP %28

&rem *****
&rem * PLOTPONT.COL
&rem * LAST UPDATE: 5-6-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTHOWW.COL
&rem * CALLS: REDRAW.SML, PLOTINZ1.SML, PLOTMORE.COL, PLOTNOTE.COL
&rem * OBJECTIVE: ALLOWS USER TO ZOOM IN ON A SPECIFIC LOCATION TO LIST
&rem *              INDIVIDUAL INCIDENTS FOR THAT MONTH
&rem *****

&LABEL ZOOM`AND`SELECT
  MAPEX *
    &GOTO PLOTINC &IF &EQ %6 T
  &RUN REDRAW
  &GOTO MORE

&LABEL PLOTINC
  &RUN PLOTINZ1
  &GOTO QUIT &IF &EQ %45 NO
  &GOTO MORE

&LABEL MORE
  POPUP PLOTMORE.COL SML
  &GOTO ZOOM`AND`SELECT &IF &EQ %49 "YES"
  &GOTO IDENTIFY

&LABEL IDENTIFY
  popup plotnote.col sml
  reselect %1 arcs box *
  MOVEITEM %1 ARCS STREET_NAM TO 22
  &return

&LABEL QUIT
  &RETURN

```

```
&rem *****  
&rem * REDRAW.SML  
&rem * LAST UPDATE: 5-6-93  
&rem * WRITTEN BY: KELLY PARMA  
&rem * CALLED BY: PLOTUTIL.COL, PLOTMAPZ.COL, PLOTPONT.COL  
&rem * CALLS: none  
&rem * OBJECTIVE: REDRAWS MAP AFTER ZOOMING IN  
&rem *****
```

```
clear  
LINESET COLOR.LIN  
ARCS %1  
whereis 9 SOUTHWEST  
whereis 9 LOOP 610W  
WHEREIS 9 LOOP 610S  
ANNOTEXT %1  
&run source  
&run tti  
&return
```

```

&rem 8 1 5 58 8 12
&rem `ZOOM`IN/OUT
&rem `=====
&rem ZOOM`IN
&rem
&rem ZOOM`OUT
&rem
&rem CANCEL
&rem
&rem EOF
&rem
&JUMP %8

&rem *****
&rem * PLOTMAPZ.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA, NITIN VAIDYA
&rem * CALLED BY: PLOTUTIL.COL
&rem * CALLS: REDRAW.SML, PLOTINZ1.SML
&rem * REDRAW2.SML, PLOTINZ2.SML
&rem * OBJECTIVE: ALLOWS USER TO EITHER ZOOM IN OR OUT ON THE MAP
&rem *****

&label ZOOM`IN
mapextent *
  &GOTO PLOTINC &IF &EQ %6 "T"
  &run redraw
  &goto QUIT

&label ZOOM`OUT
mapextent %1
  &GOTO PLOTINC &IF &EQ %6 "T"
  &RUN REDRAW2
  &goto QUIT

&label PLOTINC
  &GOTO PLOTINC2 &IF &EQ %20 "CUMULATIVE"
  &RUN PLOTINZ1
  &GOTO QUIT

&LABEL PLOTINC2
  &RUN PLOTINZ2
  &GOTO QUIT

&label CANCEL
  &goto QUIT

&label QUIT
  &return

```

```
&rem *****
&rem * REDRAW2.SML
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTMAPZ.COL
&rem * CALLS: FILEVAR.SML, PLOTDRAW.SML
&rem * OBJECTIVE: DRAWS WHOLE MAP WHEN ZOOMING OUT
&rem *****

&run filevar
DISP 4 3
SETPAGE 1 1
&RUN PLOTDRAW

&sv 2 n
&label WHILE
    &goto EXIT &if &eq %2 "Y"
    &RETURN

&label EXIT
Q
```

```

&rem *****
&rem * PLOTINZ2.SML
&rem * LAST UPDATE: 5-4-93
&rem * WRITTEN BY: KELLY PARMA, NITIN VAIDYA
&rem * CALLED BY: PLOTMAPZ.COL
&rem * CALLS: FILEVAR.SML, SOURCE.SML, TTI.SML
&rem *          SWTITLE.TXT, SWCUMUL.TXT, SWLEG2.TXT, TOTALINC.TXT
&rem * OBJECTIVE: DRAWS MAP and CUMULATIVE INCIDENTS WHEN ZOOMING IN
&rem *****

```

```

&RUN FILEVAR
DISP 4 3
SETPAGE 1 1
LINESET COLOR.LIN
LINESYMBOL 8
addresscov %1
ARCS %1
WHEREIS 9 SOUTHWEST
WHEREIS 9 LOOP 610W
WHEREIS 9 LOOP 610S
ANNOTEXT %1
TEXTSIZE .1 .15
MOVE 1 6.5
    TEXTFILE SWTITLE.TXT
MOVE 2 6.25
    TEXTFILE SWCUMUL.TXT
&run source
&run tti

```

```

MARKERSET COLOR.SW
SHADESET COLOR.SHD
SHADESYMBOL 100
PATCH 6.8 0.2 8.8 2.2
KEYBOX 0 0
KEYPOSITION 7. 2.
BOX 6.8 0.2 8.8 2.2
TEXTSIZE .11 .05
KEYMARKER SWLEG2.TXT NOBOX

```

```

&open TOTALINC.TXT NOTOPEN

```

```

&label READDATA
    &read 18 NOTREAD
    &extract 16 18 1
    &extract 17 18 2
    &extract 19 18 3
    &goto LOWINCIDENT &if &rn %16 0 9
    &GOTO LOWMEDINCID &IF &RN %16 10 49
    &goto MEDINCIDENT &if &rn %16 50 199
    &GOTO MEDHIHINCID &IF &RN %16 200 499
    &goto HIHINCIDENT &if &rn %16 500 1500
    &goto READDATA

```

```

&label LOWINCIDENT
    whereis 5 SOUTHWEST\FWY/%17%19
    &goto READDATA

```

```

&LABEL LOWMEDINCID
    WHEREIS 30 SOUTHWEST\FWY/%17%19
    &GOTO READDATA

```

```
&label MEDINCIDENT
  whereis 72 SOUTHWEST\FWY/%17%19
  &goto READDATA

&LABEL MEDHIHINCID
  WHEREIS 48 SOUTHWEST\FWY/%17%19
  &GOTO READDATA

&label HIHINCIDENT
  whereis 100 SOUTHWEST\FWY/%17%19
  &goto READDATA

&label NOTREAD
  &close showinc
  &SV 6 T
  &goto ENDPROG

&label NOTOPEN
  &type "SHOWINC could not be opened"
  &goto ENDPROG

&label ENDPROG
  &return
```

```
&rem 48 1 5 10 5 36
&rem 'TO' SELECT 'ARC' FOR 'INCIDENT' DATA---
&rem 'YOU' MUST 'DRAW' A 'BOX' AROUND 'THE' ARC
&rem 'YOU' WANT 'TO' IDENTIFY.
&rem 'DRAW' THE 'BOX' AROUND 'ONLY' PART 'OF
&rem 'THE' DESIRED 'ARC'.
&rem EOF

&rem *****
&rem * PLOTNOTE.COL
&rem * LAST UPDATE: 4-16-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTPONT.COL
&rem * CALLS: none
&rem * OBJECTIVE: TELLS THE USER HOW TO SELECT THE ARC TO LIST INCIDENT DATA
&rem *****

&RETURN
```

```
&REM 49 1 10 3 6 13
&REM 'ZOOM'MORE?
&REM '=====
&REM
&REM YES
&REM
&REM NO
&REM EOF
&JUMP %49
```

```
&rem *****
&rem * PLOTMORE.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTPONT.COL
&rem * CALLS: none
&rem * OBJECTIVE: LETS USER SELECT WHETHER IT IS NEEDED TO ZOOM IN ANY MORE
&rem * WHEN LISTING INCIDENTS.
&rem *****
```

```
&RETURN
```



```

&rem *****
&rem * PLOTINZ1.SML
&rem * LAST UPDATE: 5-6-93
&rem * WRITTEN BY: NITIN VAIDYA, KELLY PARMA
&rem * CALLED BY: PLOTMAPZ.COL, PLOTPOINT.COL
&rem * CALLS: FILEVAR.SML, SOURCE.SML, TTI.SML
&rem *          SWTITLE.TXT, SWLEG.TXT, C:\SWBAK51\PULLDOWN\COUNTS\%7%15CNT.TXT
&rem * OBJECTIVE: REDRAWS MAP and INCIDENTS WHEN ZOOMING IN
&rem *****

```

```

&RUN FILEVAR
DISP 4 3
SETPAGE 1 1
LINESET COLOR.LIN
LINESYMBOL 8
addresscov %1
ARCS %1
WHEREIS 9 SOUTHWEST
WHEREIS 9 LOOP 610W
WHEREIS 9 LOOP 610S
ANNOTEXT %1
TEXTSIZE .1 .15
MOVE 1 6.5
    TEXTFILE SWTITLE.TXT
MOVE 3.5 6.25
    TEXT %7.19%15
&run source
&run tti

```

```

MARKERSET COLOR.SW
SHADESET COLOR.SHD
SHADESYMBOL 100
PATCH 6.8 0.2 8.7 2.2
KEYBOX 0 0
KEYPOSITION 7. 2.
BOX 6.8 0.2 8.7 2.2
TEXTSIZE .11 .05
KEYMARKER SWLEG.TXT NOBOX

```

```

&open D:\SWINC\PULLDOWN\COUNTS\%7%15CNT.TXT NOTOPEN

```

```

&label READDATA
    &read 18 NOTREAD
    &extract 16 18 1
    &extract 17 18 2
    &extract 19 18 3
    &goto LOWINCIDENT &if &rn %16 0 4
    &GOTO LOWMEDINCID &IF &RN %16 5 9
    &goto MEDINCIDENT &if &rn %16 10 14
    &GOTO MEDHIHINCID &IF &RN %16 15 19
    &goto HIHINCIDENT &if &rn %16 20 200
    &goto READDATA

```

```

&label LOWINCIDENT
    whereis 5 SOUTHWEST`FWY/%17%19
    &goto READDATA

```

```

&LABEL LOWMEDINCID
    WHEREIS 30 SOUTHWEST`FWY/%17%19
    &GOTO READDATA

```

```
&label MEDINCIDENT
  whereis 72 SOUTHWEST\FWY/%17%19
  &goto READDATA

&LABEL MEDHIHINCID
  WHEREIS 48 SOUTHWEST\FWY/%17%19
  &GOTO READDATA

&label HIHINCIDENT
  whereis 100 SOUTHWEST\FWY/%17%19
  &goto READDATA

&label NOTREAD
  &close showinc
  &goto ENDPROG

&label NOTOPEN
  &type "SHOWINC could not be opened"
  popup july90.col sml
  &SV 45 "NO"
  &goto ENDPROG

&label ENDPROG
  &return
```

```

&rem *****
&rem * PLOTUML.SML
&rem * LAST UPDATE: 4-16-93
&rem * WRITTEN BY: NITIN VAIDYA, KELLY PARMA
&rem * CALLED BY: PLOTMENU.MNU
&rem * CALLS: FILEVAR.SML, PLOTCONT.COL
&rem *          SWTITLE.TXT, SWCUMML.TXT, SWLEG2.TXT, TOTALINC.TXT
&rem * OBJECTIVE: PLOTS TO THE SCREEN THE TOTAL NUMBER OF INCIDENTS OCCURRING
&rem *          ON THE SOUTHWEST FREEWAY FROM AUGUST 1989 - OCTOBER 1992.

```

```

&rem *****

```

```

&RUN FILEVAR
DISP 4 3
SETPAGE 1 1
LINESET COLOR.LIN
LINESYMBOL 8
mapex %1
addresscov %1
ARCS %1
WHEREIS 9 SOUTHWEST
WHEREIS 9 LOOP 610W
WHEREIS 9 LOOP 610S
ANNOTEXT %1
TEXTSIZE .1 .15
MOVE 1 6.5
  TEXTFILE SWTITLE.TXT
MOVE 2 6.25
  TEXTFILE SWCUMML.TXT
&run scale
&run source
&run tti
&setvar 26 0

```

```

MARKERSET COLOR.SW
SHADESET COLOR.SHD
SHADESYMBOL 100
PATCH 6.8 0.2 8.8 2.2
KEYBOX 0 0
KEYPOSITION 7. 2.
BOX 6.8 0.2 8.8 2.2
TEXTSIZE .11 .05
KEYMARKER SWLEG2.TXT NOBOX

```

```

&open TOTALINC.TXT NOTOPEN

```

```

&label READDATA
  &read 18 NOTREAD
  &extract 16 18 1
  &extract 17 18 2
  &extract 19 18 3
  &calcvar 26 %16 + %26
  &goto LOWINCIDENT &if &rn %16 0 9
  &GOTO LOWMEDINCID &IF &RN %16 10 49
  &goto MEDINCIDENT &if &rn %16 50 199
  &GOTO MEDHIHINCID &IF &RN %16 200 499
  &goto HIHINCIDENT &if &rn %16 500 1500
  &goto READDATA

```

```

&label LOWINCIDENT
  whereis 5 SOUTHWEST'FWY/%17%19

```

```

&goto READDATA

&LABEL LOWMEDINCID
WHEREIS 30 SOUTHWEST\FWY/%17%19
&GOTO READDATA

&label MEDINCIDENT
whereis 72 SOUTHWEST\FWY/%17%19
&goto READDATA

&LABEL MEDHIHINCID
WHEREIS 48 SOUTHWEST\FWY/%17%19
&GOTO READDATA

&label HIHINCIDENT
whereis 100 SOUTHWEST\FWY/%17%19
&goto READDATA

&label NOTREAD
&close showinc
&SV 6 T
textsize .1 .15
move 3 6
text INCIDENTS:%26
POPUP PLOTCONT.COL SML
&goto ENDPROG

&label NOTOPEN
&type "SHOWINC could not be opened"
&goto ENDPROG

&label ENDPROG
&return

```

&REM 27 1 23 1 1 25  
&REM Press 'ENTER' to 'Continue'  
&REM EOF  
&RETURN

&rem \*\*\*\*\*  
&rem \* **PLOTCONT.COL**  
&rem \* LAST UPDATE: 4-15-93  
&rem \* WRITTEN BY: KELLY PARMA  
&rem \* CALLED BY: PLOTUML.SML, PLOTDISP.SML  
&rem \* CALLS: none  
&rem \* OBJECTIVE: HOLDS THE SCREEN SHOWING THE INCIDENT LOCATIONS (MONTHLY/  
&rem \* CUMULATIVE) UNTIL USER PRESSES A KEY TO BRING BACK MENU.  
&rem \*\*\*\*\*

```

&rem 4 1 5 58 13 17
&rem 'UTILITIES
&rem '=====
&rem WHERE'IS
&rem
&rem IDENTIFY'STREET
&rem
&rem ZOOM'IN/OUT
&rem
&REM PLOT'DISPLAY
&rem
&REM CLEAR'INCIDENTS
&REM
&rem CANCEL
&rem EOF
&rem
&jump %4

&rem *****
&rem * PLOTUTIL.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: NITIN VAIDYA, KELLY PARMA
&rem * CALLED BY: PLOTMENU.MNU
&rem * CALLS: PLOTMAPZ.COL
&rem *          PLOTLOCA.COL
&rem *          PLOTWHER.COL
&rem *          PLOTEFFX.COL, PLOTWHA2.COL, PLOTMTH2.COL, PLOTYER2.COL, PLOTSURE.COL
&rem *          REDRAW.SML
&rem * OBJECTIVE: ALLOWS USER TO CHOOSE FROM A VARIETY OF OPTIONS TO PERFORM
&rem *          DIFFERENT FUNCTIONS
&rem *****

&label ZOOM'IN/OUT
  popup PLOTmapz.col sml
  &goto END_ROUTE

&label IDENTIFY'STREET
  linesymbol 14
  popup plotloca.col sml
  &openw streetid
  &write "STREET IDENTIFICATION"
  identify %1 arcs * street_nam f_add t_add from_st to_st class
  &write
  &write "Click Anywhere to Exit"
  &closew streetid
  popup streetid 5 1 1 1 10 25
  linesymbol 8
  &goto END_ROUTE

&label WHERE'IS
  popup PLOTwher.col sml
  &goto END_ROUTE

&LABEL PLOT'DISPLAY
  POPUP PLOTEFFX.COL SML
  POPUP PLOTWHA2.COL SML
  &goto end route &if &eq %20 "CANCEL"
  &GOTO PLOT2 &IF &EQ %20 "CUMULATIVE"
  POPUP PLOTMTH2.COL SML
  POPUP PLOTYER2.COL SML
  POPUP PLOTSURE.COL SML

```

```
      &GOTO PLOTMONTH &IF &EQ %46 "YES"  
      &GOTO END_ROUTE  
  
&LABEL PLOT2  
  POPUP PLOTSURE.COL SML  
  &GOTO PLOTTOTAL &IF &EQ %46 "YES"  
  &GOTO END_ROUTE  
  
&LABEL PLOTMONTH  
  CLEAR  
  DISP 2  
  PLOT %7%15.PLT  
  &GOTO END_ROUTE  
  
&LABEL PLOTTOTAL  
  CLEAR  
  DISP 2  
  PLOT TOTALINC.PLT  
  &GOTO END_ROUTE  
  
&label CLEAR`INCIDENTS  
  &SV 6 F  
  &RUN REDRAW  
  &GOTO QUIT  
  
&label END_ROUTE  
  &goto QUIT  
  
&label CANCEL  
  &goto QUIT  
  
&label QUIT  
  &return
```

```

&rem *****
&rem * PLOTDISP.SML
&rem * LAST UPDATE: 5-6-93
&rem * WRITTEN BY: NITIN VAIDYA, KELLY PARMA
&rem * CALLED BY: PLOTMENU.MNU
&rem * CALLS: FILEVAR.SML, SCALE.SML, SOURCE.SML, TTI.SML, PLOTCONT.COL
&rem *          SWTITLE.TXT, SWLEG.TXT, C:\SWBAK51\PULLDOWN\COUNTS\%7%15CNT.TXT
&rem * OBJECTIVE: DISPLAYS THE MONTHLY ACCIDENT LOCATIONS TO SCREEN
&rem *****

```

```

&RUN FILEVAR
DISP 4 3
SETPAGE 1 1
LINESET COLOR.LIN
LINESYMBOL 8
mapex %1
addresscov %1
ARCS %1
WHEREIS 9 SOUTHWEST
WHEREIS 9 LOOP 610W
WHEREIS 9 LOOP 610S
ANNOTEXT %1
TEXTSIZE .1 .15
MOVE 1 6.5
    TEXTFILE SWTITLE.TXT
MOVE 3.5 6.25
    TEXT %7.19%15
&run scale
&run source
&run tti
&setvar 26 0

```

```

MARKERSET COLOR.SW
SHADESET COLOR.SHD
SHADESYMBOL 100
PATCH 6.8 0.2 8.7 2.2
KEYBOX 0 0
KEYPOSITION 7. 2.
BOX 6.8 0.2 8.7 2.2
TEXTSIZE .11 .05
KEYMARKER SWLEG.TXT NOBOX

```

```

&open D:\SWINC\PULLDOWN\COUNTS\%7%15CNT.TXT NOTOPEN

```

```

&label READDATA
    &read 18 NOTREAD
    &extract 16 18 1
    &extract 17 18 2
    &extract 19 18 3
    &calcvar 26 %16 + %26
    &goto LOWINCIDENT &if &rn %16 0 4
    &GOTO LOWMEDINCID &IF &RN %16 5 9
    &goto MEDINCIDENT &if &rn %16 10 14
    &GOTO MEDHIHINCID &IF &RN %16 15 19
    &goto HIHINCIDENT &if &rn %16 20 200
    &goto READDATA

```

```

&label LOWINCIDENT
    whereis 5 SOUTHWEST`FWY/%17%19
    &goto READDATA

```



```
&LABEL LOWMEDINCID
WHEREIS 30 SOUTHWEST\FWY/%17%19
&GOTO READDATA

&label MEDINCIDENT
whereis 72 SOUTHWEST\FWY/%17%19
&goto READDATA

&LABEL MEDHIHINCID
WHEREIS 48 SOUTHWEST\FWY/%17%19
&GOTO READDATA

&label HIHINCIDENT
whereis 100 SOUTHWEST\FWY/%17%19
&goto READDATA

&label NOTREAD
&close showinc
textsize .1 .15
move 3 6
text INCIDENTS:%26
POPUP PLOTCONT.COL SML
&goto ENDPROG

&label NOTOPEN
&type "SHOWINC could not be opened"
popup july90.col sml
&SV 6 F
&goto ENDPROG

&label ENDPROG
&return
```

```
&REM 15 1 5 5 7 8
&REM `YEAR
&REM `=====
&REM 89
&REM 90
&REM 91
&REM 92
&REM 93
&REM EOF
```

```
&rem *****
&rem * PLOTYER2.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTUTIL.COL
&rem * CALLS: none
&rem * OBJECTIVE: SELECTS YEAR FOR PLOTTING MONTHLY REPORTS TO PRINTER
&rem *****
```

```
&RETURN
```

```
&REM 7 1 6 15 14 7
&REM `MONTH
&REM `=====
&REM JAN
&REM FEB
&REM MAR
&REM APR
&REM MAY
&REM JUN
&REM JUL
&REM AUG
&REM SEP
&REM OCT
&REM NOV
&REM DEC
&REM EOF
```

```
&rem *****
&rem * PLOTMTH2.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTUTIL.COL
&rem * CALLS: none
&rem * OBJECTIVE: SELECTS MONTH TO USE FOR PLOTTING TO PRINTER
&rem *****
```

```
&RETURN
```

```
&rem 47 1 5 10 2 28
&rem 'POSITION'CURSOR'TO'DESIRED
&REM  'STREET'AND'PRESS'ENTER
&REM EOF
```

```
&rem *****
&rem * PLOTLOCA.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTUTIL.COL
&rem * CALLS: none
&rem * OBJECTIVE: LETS THE USER KNOW TO POSITION CURSOR TO SELECT STREET TO
&rem * IDENTIFY
&rem *****
```

.

```

&REM 9 1 5 15 8 14
&REM  'WHERE'IS
&REM  '=====
&REM
&REM  STREET
&REM
&REM  INTERSECTION
&REM
&REM  CANCEL
&REM EOF
&JUMP %9

```

```

&rem *****
&rem * PLOTWHER.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTUTIL.COL
&rem * CALLS: PLOTBLOK.FRM, PLOTSTRT.COL
&rem *          PLOTINT1.COL, PLOTINT2.COL
&rem * OBJECTIVE: ALLOWS USER TO SELECT WHICH TYPE OF LOCATION TO BE FOUND
&rem *****

```

```

&LABEL STREET
&SV 13 " "
POPUP PLOTBLOK.FRM SML
  &GOTO QUIT &IF &EQ %2 "Y"
POPUP PLOTSTRT.COL SML
  &GOTO QUIT &IF &EQ %2 "Y"
whereis 57 %13 %10
&RETURN

```

```

&LABEL INTERSECTION
POPUP PLOTINT1.COL SML
  &GOTO QUIT &IF &EQ %2 "Y"
POPUP PLOTINT2.COL SML
  &GOTO QUIT &IF &EQ %2 "Y"
WHEREIS 95 %11 / %12
&RETURN

```

```

&label CANCEL
  &goto QUIT

```

```

&LABEL QUIT
  &RETURN

```

```
POPUP 45 1 4 5 2 35
&REM `AFTER`PLOTING`TO`PRINTER
&REM `ARC/INFO`WILL`AUTOMATICALLY`QUIT
&REM EOF

&rem *****
&rem * PLOTEFFX.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTUTIL.COL
&rem * CALLS: none
&rem * OBJECTIVE: NOTIFIES USER THAT THE PROGRAM WILL AUTOMATICALLY QUIT
&rem *                WHEN PLOTING DISPLAY TO THE PRINTER
&rem *****

&RETURN
```

```
&REM 20 1 5 5 7 12
&REM 'PRINT
&REM '=====
&REM MONTHLY
&REM
&REM CUMULATIVE
&REM
&REM CANCEL
&REM EOF
```

```
&rem *****
&rem * PLOTWHA2.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTUTIL.COL
&rem * CALLS: none
&rem * OBJECTIVE: SELECTS WHICH TYPE OF INCIDENT REPORT TO SEND TO PRINTER
&rem *****
```

```
&GOTO QUIT &IF &EQ %20 "CANCEL"
&RETURN
```

```
&LABEL QUIT
&SV 2 Y
&RETURN
```

```
&REM 46 1 4 6 6 18
&REM 'ARE YOU SURE YOU
&REM 'WANT TO PRINT?
&REM '=====
&REM YES
&REM
&REM NO
&REM EOF
&JUMP %46
```

```
&rem *****
&rem * PLOTSURE.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: KELLY PARMA
&rem * CALLED BY: PLOTUTIL.COL
&rem * CALLS: none
&rem * OBJECTIVE: MAKES CERTAIN USER WANTS TO PLOT TO THE PRINTER AND QUIT
&rem *****
```

```
&LABEL YES
  &SV 2 Y
  &SV 47 Y
  &RETURN
```

```
&LABEL NO
  &RETURN
```



```

&REM 11 1 5 5 16 21
&REM 'SELECT MAIN STREET
&REM '=====
&REM CANCEL
&REM '-----
&REM AIRPORT
&REM ALABAMA
&REM ALIEF 'CLODINE
&REM ALMEDA
&REM ALMEDA 'GENOA
&REM AUSTIN
&REM BEECHNUT
&REM BELLAIRE
&REM BELLFORT
&REM BELTWAY '8W
&REM BISSONNET
&REM BUFFALO 'SPEEDWAY
&REM CHIMNEY 'ROCK
&REM COMMERCE 'PARK
&REM CORPORATE
&REM CRAWFORD
&REM CULLEN
&REM DUNLAVY
&REM EASTEX 'FWY
&REM EDLOE
&REM ELDRIDGE
&REM ELGIN
&REM FANNIN
&REM FONDREN
&REM FOUNTAINVIEW
&REM GESSNER
&REM GRAY
&REM GREENBRIAR
&REM GULF 'FWY
&REM HARRISBURG
&REM HARWIN
&REM HAZARD
&REM HILLCROFT
&REM HOLCOMBE
&REM KIRBY
&REM KIRKWOOD
&REM LA 'BRANCH
&REM LARKWOOD
&REM LOOP '610S
&REM LOOP '610W
&REM LOUISIANA
&REM M 'L 'KING
&REM MACGREGOR
&REM MAIN
&REM MANDELL
&REM MC 'GOWEN
&REM MONTROSE
&REM MURPHY
&REM NEWCASTLE
&REM NORTH 'FWY
&REM OLD 'SPANISH 'TRAIL
&REM POST 'OAK
&REM RICE
&REM RICHMOND
&REM SAN 'JACINTO
&REM SAVOY
&REM SH6

```

```
&REM SHEPHERD
&REM SOUTH`FWY
&REM SOUTHWEST`FWY
&REM SPUR257
&REM SUGAR`CREEK
&REM SUGAR`GROVE
&REM SWEETWATER
&REM TIMMONS
&REM VOSS
&REM WESLAYAN
&REM WESTHEIMER
&REM WESTPARK
&REM WHEELER
&REM WILCREST
&REM WILLIAM`TRACE
&REM WOODHEAD
&REM EOF
```

```
&rem *****
&rem * PLOTINT1.COL
&rem * LAST UPDATE: 4-15-93
&rem * WRITTEN BY: NITIN VAIDYA, KELLY PARMA
&rem * CALLED BY: PLOTWHER.COL
&rem * CALLS: none
&rem * OBJECTIVE: SELECT THE FIRST STREET IN FINDING WHERE TWO STREETS
&rem * INTERSECT
&rem *****
```

```
&GOTO EXIT &IF &EQ %11 "CANCEL"
&return
```

```
&LABEL EXIT
&SV 2 Y
&RETURN
```

```

&rem 14 2 5 3 5 20
&rem      'BLOCK'NUMBER
&rem '=====
&rem 'NUMBER :%13
&rem
&rem NONE      OK      QUIT
&rem EOF
&rem
&jump %14

&rem *****
&rem * PLOTBLOK.FRM
&rem * LAST UPDATE: 4-15-93
&rem * CALLED BY: PLOTWHER.COL
&rem * CALLS: none
&rem * OBJECTIVE: ALLOWS USER TO ENTER BLOCK NUMBER TO FIND A SPECIFIC
&rem *              LOCATION
&rem *****

&label OK
&return

&label NONE
&sv 13 " "
&return

&LABEL QUIT
&SV 2 "Y"
&RETURN

```

```
&REM *****  
&REM  Filename :    RSETUP.SML  
&REM  WRITTEN BY :  Dan Hickman  
&REM  Calls :      RSTART.SML  
&REM  Description : first SML called to start the resource manual application  
&REM *****
```

```
ROUTE RSTART  
QUIT
```

```

&REM *****
&REM
&REM FILENAME : RSTART.SML
&REM WRITTEN BY : Dan Hickman
&REM CALLS : MENU1.MNU MENU2.SML <resource name>.SML
&REM CALLED BY : RSETUP.SML
&REM DESCRIPTION : This SML displays a menu of resources and lets you
&REM choose which resource you want to work with.

READNET ALLMAP LENGTH LENGTH
DISP 4 2
ADDRESSCOV ALLMAP

&LABEL START
MAPEX 2993249 645000 3259839 850450
DRAWNET SYMBOL
POPUP MENU1.MNU 1 1 1 1 6 12
&JUMP QUIT &IF &EQ %1 QUIT
&JUMP EXIT &IF &EQ %1 EXIT
&JUMP NEXTMENU
&LABEL QUIT
QUIT
&LABEL NEXTMENU
&REM ***** ASSIGN VAR 3 TO AAT CENTER RECORD NAME *****
&GOTO %1
&GOTO ENDIF1
&LABEL POLICE
&GOTO START
&LABEL AMBULANCE
&GOTO START
&LABEL CONTACT
&SETVAR 4 CONTSYM
&SETVAR 3 CONTACT
&GOTO ENDIF1
&LABEL FIRE
&SETVAR 4 FIRESYM
&SETVAR 3 FIRE
&GOTO ENDIF1
&LABEL ENDIF1

&REM *** COLOR CODE ALL ARCS BY WHAT SECTION THEY ARE ALLOCATED TO *****
DRAWNET %4
&RUN %3.SML

&LABEL MENULoop
POPUP MENU2.SML SML
&JUMP MENULoop &IF &NE %2 RETURN
&JUMP START
&LABEL EXIT

```

POLICE  
AMBULANCE  
FIRE  
CONTACT

QUIT

Comments: File Name **MENU1.MNU**

MATCH INTERSECTION  
CHOOSE INTERACTIVELY

CANCEL  
Comments: File Name **MENU3.MNU**

ENTER STREET NUMBER :  
Comments: File Name **ADDIN.MNU**



```

&REM 2 1 1 1 9 20
&REM '%1
&REM
&REM LOCATE INTERSECTION
&REM MATCH ADDRESS
&REM IDENTIFY ARC
&REM ZOOM_IN
&REM ZOOM_OUT
&REM
&REM RETURN
&REM EOF

&REM *****
&REM  Filename :  MENU2.SML
&REM  Written By :  Dan Hickman
&REM  Calls :    <resource name>.SML STREETS.TXT OUTPUT.TX NOTFND2.MSG
&REM           ADDIN.MNU CENTNUM.DAT <resource name>.dat MATCH.DAT
&REM  Called By :  RSTART.SML
&REM  ***** VARIABLES  1 - TYPE OF RESOURCE CHOSEN
&REM  *****           2 - HOLDS FUNCTION CHOICE
&REM  *****           3 - CURRENT RESOURCE VARIABLE
&REM  *****           4 - LINE SYMBOL VARIABLE FOR CURRENT RESOURCE
&REM  DESCRIPTION :This SML displays a menu of options for the current
&REM                resource and executes commands depending on user input.

&REM  ***** PICK FUNCTION CHOICE *****
&GOTO %2
    &GOTO RETURN

&REM  ***** SHOW ALL CENTERS OF CURRENT TYPE ON MAP *****
&LABEL SHOW
    &RUN %3.SML
    &GOTO RETURN

&REM  ***** SHOW INTERSECTION ON SCREEN *****
&LABEL LOCATE
    POPUP STREETS.TXT 5 1 1 1 23 20
    POPUP STREETS.TXT 6 1 1 1 23 20
    &OPENW OUTPUT.TST
    WHEREIS 1 %5/%6
    &CLOSEW
    &OPEN OUTPUT.TST
    &READ 7 RETURN
    &CLOSE
    POPUP NOTFND2.MSG 0 1 1 1 2 15
    &GOTO RETURN

&REM  ***** SHOW STREET ADDRESS ON SCREEN *****
&LABEL MATCH
    POPUP STREETS.TXT 6 1 1 1 23 20
    POPUP ADDIN.MNU 5 3 1 1 1 30
    &OPENW OUTPUT.TST
    WHEREIS 1 %5 %6
    &CLOSEW
    &OPEN OUTPUT.TST
    &READ 7 RETURN
    &CLOSE
    POPUP NOTFND2.MSG 0 1 1 1 2 15
    &GOTO RETURN

&REM  ***** IDENTIFY AN ARC AND SHOW WHICH CENTER IT IS ALLOCATED TO *****
&LABEL IDENTIFY

```

```

&REM ***** WRITE IDENTIFY OPERATION OUTPUT TO FILE *****
&OPENW CENTNUM.DAT
IDENTIFY ALLMAP ARC * %3
&CLOSEW

&REM ***** READ IDENTIFIED ARC'S ALLOCATED RESOURCE NODE NUMBER *****
&OPEN CENTNUM.DAT
&READ 7 CHOOSE
&READ 7 CHOOSE
&CLOSE

&REM ***** SEARCH THROUGH RESOURCE DATA FILE FOR IDENTIFIED RESOURCE *****
&OPEN %3.DAT
&LABEL WHILE1
  &READ 8 NOTFOUND
  &EXTRACT 9 8 1
&GOTO WHILE1 &IF &NE %9 %7
&CLOSE
&GOTO FOUND

&REM ***** JUMPS TO HERE IF NO MATCH IS FOUND *****
&LABEL NOTFOUND
&OPENW MATCH.DAT
&WRITE "%1 RESOURCE NOT FOUND FOR THIS ARC"
&WRITE "CONTINUE"
&CLOSEW
POPUP MATCH.DAT 0 1 1 1 2 45
&GOTO RETURN

&REM ***** PUT RESOURCE INFO INTO A FILE TO BE SHOWN ON SCREEN *****
&LABEL FOUND
&EXTRACT 8 8 2
&OPENW MATCH.DAT
&WRITE "%1 RESOURCE"
&WRITE " %8"
&WRITE " CONTINUE"
&CLOSEW
POPUP MATCH.DAT 0 1 1 1 3 40

&GOTO RETURN

&REM ***** LET USER PICK AREA, ZOOM, AND REDRAW RESOURCES *****
&LABEL ZOOM_IN
MAPEX *
DRAWNET %4
&RUN %3.SML
&GOTO RETURN

&REM ***** ZOOM OUT TO ORIGINAL VIEW AND REDRAW RESOURCES *****
&LABEL ZOOM_OUT
MAPEX 2993249 645000 3259839 850450
DRAWNET %4
&RUN %3.SML
&GOTO RETURN

&LABEL RETURN
&LABEL CANCEL

```

```
&REM *****
&REM  FILENAME :      FIRE.SML
&REM  WRITTEN BY :    Dan Hickman
&REM  CALLED BY :     MENU2.SML
&REM  PURPOSE  :      This file places a marker on the map at each of the
&REM                   fire station resources.
&REM *****
```

```
WHEREIS 55 BELTWAY 8W / BISSONNET
WHEREIS 55 MAIN / WESTHEIMER
WHEREIS 55 NORTHWEST FWY / TIDWELL W
WHEREIS 55 BELTWAY 8N / ALDINE WESTFIELD
WHEREIS 55 SHAVER / LAPORTE FWY
```

```

&rem *****
&rem   FileName:      PDMAPH.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        ROUTE PDROUT.SML, ARCEDIT NBLOCLAN.SML,
&rem                 ALLOCATE NMAPALLO.SML, TABLES UNBLOCLN.SML
&rem                 TABLES TIME_TAB.SML, TABLES FILEBARR.SML
&rem   CalledBy:     AIMS.COL
&rem   DataFiles:    FILEVAR.SML
&rem   Variables:    %21 -> Name of base coverage
&rem                 %40 -> Flag for hourly volume
&rem                 %41 -> Flag for hazard are blocking
&rem                 %42 -> Flag for lane blocking
&rem   Purpose:      This program is common to three applications:
&rem                 INCMAN, TRAVEL TIME SIMULATION and FRONTAGE ROAD
&rem                 ROUTING. The only difference is in the base
&rem                 coverage/map which is stored in variable %21.
&rem                 The program first calls PDROUT.SML. Depending on
&rem                 flag/variable setting by the user in other programs
&rem                 called by PDROUT.SML, this program will call one of
&rem                 the above mentioned programs.
&rem *****

```

```

clear
&sv 21 d:\incman
&save filevar.sml 21 21
clear
&goto ROUTING'APPLICATIONS

```

```

&rem *****
&rem   Run INCMAN and call other application programs based
&rem   user selections in the simulation program.
&rem *****

```

```

&label ROUTING'APPLICATIONS
route pdROUT
&run filevar
&goto HOURLY'TIME &if &eq %40 TIMEUPDATE
&goto DISASTER &if &eq %41 HAZARD
&goto BLOCK'LANES &if &eq %42 BLOCKLANES
&goto UNBLOCKLANES &if &eq %42 UNBLOCKLANES
&goto end'incman

```

```

&rem *****
&rem   Call program to block individual lanes for further
&rem   routing in INCMAN
&rem *****

```

```

&label BLOCK'LANES
arcedit nbloclane
&sv 42 ""
&save filevar 1 50
abuild %21 line
&goto ROUTING'APPLICATIONS

```

```

&rem *****
&rem   Clear individual lanes that have been blocked
&rem   for INCMAN
&rem *****

```

```

&label UNBLOCKLANES
tables unblocln
&sv 42 ""
&save filevar 1 50
abuild %21 line

```

```

&goto ROUTING'APPLICATIONS

&rem *****
&rem Call program to update impedance values based on hourly volumes
&rem for INCMAN
&rem *****
&label HOURLY'TIME
    tables time tab
    &goto end'incman &if &ne %40 TIMEUPDATE
    &sv 40 ""
    &save filevar 1 50
    &goto ROUTING'APPLICATIONS

&rem *****
&rem Call application program for blocking of hazard areas;
&rem This is an INCMAN application that can also be used
&rem with TRAVEL TIME SIMULATION and FRONTAGE ROAD ROUTING
&rem *****
&label DISASTER
    allocate nmapallocate
    tables filebarrier
    &goto ROUTING'APPLICATIONS

&label end'incman
    clear
    &goto EXIT

&label EXIT
    clear
    &return

```

```

&rem *****
&rem   FileName:      PDROUT.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        PDREDRAW.SML, PDMENU.MNU, ACOPY.EXE
&rem   CalledBy:     PDMAPH.SML
&rem   DataFiles:    m$i00.sml, m$f00.sml, m$b00.sml, m$r00.sml,
&rem                 m$v00.sml, m$vf00.sml, m$i10.sml, m$v10.sml
&rem                 initvan.loc, initloc, vanxy
&rem   Variables:    %21 -> Name of coverage/map
&rem                 %39 -> Variable for conditional loop to popup
&rem                 PDMENU.MNU, which is the main INCMAN
&rem                 application menu.
&rem   Purpose:      Runs under the ROUTE module. Reads the coverage and
&rem                 draws the map in appropriate color. Initializes the
&rem                 temporary files that record van and/or incident
&rem                 locations before continuing execution through
&rem                 PDREDRAW.SML. Also pops up the main menu in a loop.
&rem *****

&run filevar
&rem *****
&rem   t_impedanc and f_impedanc are in reversed order because to-from-to
&rem   data in AAT is in reverse order
&rem *****
readnetwork %21 t_impedanc f_impedanc turmtime
addresscov %21
disp 4 3
mapex %21
setpage 1 1
markerset color.map

&rem *****
&rem   Initialize temporary files
&rem *****
&openw m$i00.sml
&closew m$i00.sml
&REM &openw m$f00.sml
&REM &closew m$f00.sml
&sys "acopy initloc m$f00.sml"
&openw m$b00.sml
&closew m$b00.sml
&openw m$r00.sml
&closew m$r00.sml
&openw m$vb00.sml
&closew m$vb00.sml
&REM &openw m$vf00.sml
&REM &closew m$vf00.sml
&sys "acopy initvan.loc m$vf00"
&sys "acopy initvan.loc m$vf00.sml"
&openw m$i10.sml
&closew m$i10.sml
&openw m$i10
&closew m$i10
&openw m$v10
&closew m$v10
&openw vanxy
&closew vanxy

&rem *****
&rem   Run the PDREDRAW.SML program
&rem *****

```

```
&run pdredraw

&rem *****
&rem Initialize temporary variable
&rem for conditional looping and pop
&rem up the main menu in the loop
&rem *****
&sv 39 N
&label WHILE
    &goto EXIT &if &eq %39 "Y"
    popup pdmenu.mnu sml
    &goto WHILE

&label EXIT
q
```

```

&rem *****
&rem   FileName:      PDREDRAW.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     PDROUT.SML, PDVANUP.COL, PDINCUP.COL, PDROUTEM.COL
&rem   DataFiles:    CENTERF
&rem   Variables:    %21 -> Name of coverage/map
&rem                %37 -> Counter for the number of routes that
&rem                may be redrawn.
&rem                %41 -> Flag for activation of blocking of hazard
&rem                area.
&rem                %42 -> Node number of intersection to be blocked
&rem   Purpose:      This is called by several programs. Redraws the map
&rem                and coverage in colors, while redrawing each
&rem                recorded route in color. If the hazard area flag is
&rem                set then the appropriate intersections are blocked
&rem                using the information generated by NMAPALLO.SML and
&rem                FILEBARR.SML programs.
&rem *****

```

```

clear
drawnetwork symbol
&cv 37 1

```

```

&rem *****
&rem   Block hazard areas if the flag %41 is set to
&rem   HAZARD
&rem *****
&goto DISASTER &if &cn %41 HAZARD

```

```

&rem *****
&rem   Conditional loop to find out and redraw all
&rem   routes in the current coverage/map
&rem *****
&label WHILE
  &goto CONTZOOM &if &nr %37 1 15
  &goto CONTROUTE &if &nn %37
  selectroute %37
  drawroute
  &cv 37 1 + %37

```

```

&label CONTROUTE
  &goto WHILE

```

```

&rem *****
&rem   This was used to execute temporary SML files.
&rem   Not used currently !!
&rem *****
&label CONTZOOM
  &run m$i00
  &run m$f00
  &run m$b00
&return

```

```

&rem *****
&rem   If the hazard flag is set then read each
&rem   intersection(node) to be blocked and perform
&rem   block action on the map.
&rem *****
&label DISASTER

```



```

&rem *****
&rem   A new plotter file is used so that side effect
&rem   of blocking the intersections are minimally
&rem   visible on the screen.
&rem *****
textset plotter.disaster
textsymbol 77

&rem *****
&rem   Read blocked nodes data from file CENTERF
&rem *****
&open %21\pulldown\centerf filenotopen
&rem enter the loop
&label listmap
  &read 42 endfile
  &extract 42 42 1

  &rem *****
  &rem   Filter irrelevant data
  &rem *****
  &goto incrvar &if &nn %42
  addbarrier %42 8
&label incrvar
  &goto listmap

&label endfile
  &close
  &run filevar

  &rem *****
  &rem   Reset the hazard flag/variable
  &rem *****
  &sv 41 ""
  &save filevar 1 50
  &goto CONTRROUTE

&label longfile
  &sv 42 1
  &goto listmap

&label filenotopen
  &type "unable to open file"
  &goto end_maproute

&label end_maproute
  &return

```

```

&rem 43 2 1 1 4 78
&rem
&rem          'MAIN'MENU'
&rem          'A.I.M.S.:INCMAN'(INCident'MANagement'Support'Tool)
&rem
'=====
&rem  INCIDENTS      VAN'UPDATE      DISPATCH      TRAFFIC'UPDATE      UTILITIES
QUIT
&rem EOF

&rem *****
&rem  FileName:      PDMENU.MNU (for INCMAN)
&rem  LastUpdate:   6/1/93
&rem  WrittenBy:    Nitin Vaidya
&rem  Calls:        PDVANUP.COL, PDINCUP.COL, PDUTIL.COL
&rem               PDROUTEM.COL, PDBARR.COL
&rem  CalledBy:     PDROUT.SML
&rem  DataFiles:    None
&rem  Variables:    %39 ->  Flag for conditional loop to popup this
&rem               menu repeatedly through PDROUT.SML
&rem               %43 ->  User selection from menu.
&rem  Purpose:      Main menu for INCMAN application. Calls other menus
&rem               based on user selection
&rem *****

&jump %43

&label VAN'UPDATE
  popup pdvanup.col sml
  &goto REPEAT

&label INCIDENTS
  popup pdincup.col sml
  &goto REPEAT

&label DISPATCH
  popup pdroutem.col sml
  &goto REPEAT

&label UTILITIES
  popup pdutil.col sml
  &goto REPEAT

&label TRAFFIC'UPDATE
  popup pdbarr.col sml
  &goto REPEAT

&rem *****
&rem  Pop this menu up again
&rem *****
&label REPEAT
  &sv 39 N
  &return

&label QUIT
  &sv 39 Y
  &return

```

```

&rem 43 2 1 13 4 54
&rem
&rem          'A.I.M.S
&rem          'Travel'Time'Simulation
&rem '=====
&rem DISPATCH      TRAFFIC'UPDATE      UTILITIES      QUIT
&rem EOF

&rem *****
&rem   FileName:      PDMENU.MNU (for TRAVEL TIME SIMULATION)
&rem   LastUpdate:    6/1/93
&rem   WrittenBy:     Nitin Vaidya
&rem   Calls:         PDUTIL.COL, PDROUTEM.COL, PDBARR.COL
&rem   CalledBy:      PDROUT.SML
&rem   DataFiles:     None
&rem   Variables:     %39 -> Flag for conditional loop to popup this
&rem                  menu repeatedly through PDROUT.SML
&rem                  %43 -> User selection from menu.
&rem   Purpose:       Main menu for TRAVEL TIME SIMULATION application.
&rem                  Calls other menus based on user selection
&rem *****

&jump %43

&label DISPATCH
  popup pdroutem.col sml
  &goto REPEAT

&label UTILITIES
  popup pdutil.col sml
  &goto REPEAT

&label TRAFFIC'UPDATE
  popup pdbarr.col sml
  &goto REPEAT

&rem *****
&rem   Pop this menu up again
&rem *****
&label REPEAT
  &sv 39 N
  &return

&label QUIT
  &sv 39 Y
  &return

```

```

&rem 43 2 1 13 4 52
&rem
&rem          'A.I.M.S
&rem          'Frontage'Road'Simulation
&rem '-----'
&rem ROUTE      TRAFFIC'UPDATE      UTILITIES      QUIT
&rem EOF

&rem *****
&rem   FileName:      PDMENU.MNU (for FRONTAGE ROAD ROUTING)
&rem   LastUpdate:    6/1/93
&rem   WrittenBy:     Nitin Vaidya
&rem   Calls:         PDUTIL.COL, PDROUTEM.COL, PDBARR.COL
&rem   CalledBy:      PDROUT.SML
&rem   DataFiles:     None
&rem   Variables:     %39 -> Flag for conditional loop to popup this
&rem                  menu repeatedly through PDROUT.SML
&rem                  %43 -> User selection from menu.
&rem   Purpose:       Main menu for FRONTAGE ROAD ROUTING application.
&rem                  Calls other menus based on user selection
&rem *****

&jump %43

&label ROUTE
  popup pdroutem.col sml
  &goto REPEAT

&label UTILITIES
  popup pdutil.col sml
  &goto REPEAT

&label TRAFFIC'UPDATE
  popup pdbarr.col sml
  &goto REPEAT

&rem *****
&rem   Pop this menu up again
&rem *****
&label REPEAT
  &sv 39 N
  &return

&label QUIT
  &sv 39 Y
  &return

```

```

&rem 33 1 5 42 17 18
&rem 'TRAFFIC'UPDATE
&rem '=====
&rem BY'TIME'OF'DAY
&rem
&rem BLOCK'A'ROAD
&rem
&rem REMOVE'ROAD'BLOCK
&rem
&rem BLOCK'LANES
&rem
&rem REMOVE'LANE'BLOCK
&rem
&rem BLOCK'HAZARD'AREA
&rem
&rem CLEAR'HAZARD'AREA
&rem
&rem QUIT
&rem EOF

&rem *****
&rem   FileName:      PDBARR.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        REDRAW.SML
&rem   CalledBy:     PDMENU.MNU
&rem   DataFiles:    None
&rem   Variables:    %33 -> Variable to store user selection
&rem                %40 -> Flag for changing time of day
&rem                %41 -> Flag for blocking or clearing of areas
&rem                %42 -> Flag for blocking or clearing of lanes
&rem   Purpose:      Menu to allow user to simulate traffic by either
&rem                using historical data by time of day or by
&rem                superposing incidents by blocking roads, lanes
&rem                or areas on the map. Upon user selection saves
&rem                values to flag variables which are used for
&rem                further processing.
&rem *****

&jump %33

&rem *****
&rem   Block or clear road intersection(s).
&rem *****

&label BLOCK'A'ROAD
  addbarrier *
  &goto QUIT

&label REMOVE'ROAD'BLOCK
  removebarrier *
  &goto QUIT

&rem *****
&rem   Block individual lanes or remove all lane blocks.
&rem *****

&label BLOCK'LANES
  &sv 42 BLOCKLANES
  &save filevar 1 50

```

```

quit

&label REMOVE`LANE`BLOCK
  &sv 42 UNBLOCKLANES
  &save filevar 1 50
  quit

&rem *****
&rem   Update traffic using volumes appropriate for the time of day
&rem *****

&label BY`TIME`OF`DAY
  &sv 40 TIMEUPDATE
  &save filevar 1 50
  quit

&rem *****
&rem   Block or clear a hazard area
&rem *****

&label BLOCK`HAZARD`AREA
  &sv 41 HAZARD
  &save filevar 1 50
  quit

&label CLEAR`HAZARD`AREA
  removebarrier all
  &run redraw
  &goto QUIT

&label QUIT
  &return

```

```

&rem *****
&rem   FileName:      REDRAW.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:         None
&rem   CalledBy:     PDMAPZ.COL
&rem   DataFiles:    None
&rem   Variables:    %37 -> Counter for the number of routes that
&rem                   may be redrawn
&rem   Purpose:      Used to redraw the coverage in the ROUTE module
&rem                   after a ZOOM operation. It finds and redraws all
&rem                   previous routes to maintain the state of the map.
&rem *****

```

```

clear
drawnetwork symbol
&cv 37 1

```

```

&rem *****
&rem Conditional loop to find out and redraw all routes
&rem in the current coverage/map
&rem *****

```

```

&label WHILE
  &goto CONTZOOM &if &nr %37 1 15
  &goto CONTROUTE &if &nn %37
  selectroute %37
  drawroute
  &cv 37 1 + %37

```

```

&label CONTROUTE
&goto WHILE

```

```

&rem *****
&rem Was used to run temporary SML files; no longer used
&rem *****

```

```

&label CONTZOOM
  &run m$i00
  &run m$f00
  &run m$b00

```

```

&return

```

```

&rem 32 1 5 30 9 14
&rem `DISPATCH
&rem `=====
&rem ADD`ROUTE
&rem
&rem REMOVE`ROUTE
&rem
&rem DIRECTIONS
&rem
&rem QUIT
&rem EOF

&rem *****
&rem   FileName:      PDROUTEM.COL
&rem   LastUpdate:    6/1/93
&rem   WrittenBy:     Nitin Vaidya
&rem   Calls:          ROUTENUM.COL, ROUTECOL.COL, PDREDRAW.SML, TDIR.SML
&rem                   ACOPY.EXE
&rem   CalledBy:      PDMENU.MNU
&rem   DataFiles:     TRAVTIME, TDIR, DIRECTIO
&rem   Variables:     %24 -> Variable to store route number; separately
&rem                   stores the travel time
&rem                   %25 -> Variable to store route color
&rem                   %32 -> Variable for user selection from this menu
&rem   Purpose:       Allows the user to add a new route, display or
&rem                   print directions for this route on screen, or
&rem                   remove an unused route. Uses temporary files
&rem                   to achieve some of the above.
&rem *****

&jump %32

&rem *****
&rem   Add a new route. Get the route number and color and write
&rem   travel time to temporary file `travtime`
&rem *****
&label ADD`ROUTE
  popup routenumber.col sml
  popup routecolor.col sml
  addroute %24 %25
  &openw travtime
  path *
  &closew travtime
  &open travtime NOTOPEN
  &read 24 NOTREAD
  &read 25 NOTREAD
  &extract 24 25 3
  &close travtime
  &goto QUIT

&rem *****
&rem   Remove an unused route
&rem *****
&label REMOVE`ROUTE
  removeroute
  &run pdredraw
  &goto QUIT

&rem *****
&rem   Write directions to temporary file `tdir`; append travel time
&rem   from variable 24 to `tdir` and call TDIR.SML to display
&rem   contents of this temporary file

```



```
&rem *****
&label DIRECTIONS
  &openw tdir
  &closew
  directions street nam distance miles tdir
  &sys "acopy tdir directio"
  &openw tdir A
  &write
  &write "Travel time is"
  &write %24
  &write "minutes"
  &write
  &write "Click Anywhere to Exit"
  &closew
  &run tdir
  &goto QUIT

&label NOTOPEN
  &type "Could not open file"
  &goto QUIT

&label NOTREAD
  &type "Could not read"
  &goto QUIT

&label QUIT
  &return
```

```

&rem *****
&rem   FileName:      TDIR.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        TDIR.COL
&rem   CalledBy:     PDROUTEM.COL
&rem   DataFiles:    tdir
&rem   Variables:    %38 -> Flag for popping up menu in conditional loop
&rem   Purpose:      Display directions for the current route in a
&rem                window and call TDIR.COL in a conditional loop
&rem                for other options on directions.
&rem *****

```

```

&rem *****
&rem   Display directions in window; as in contents of 'tdir' file
&rem *****
popup tdir 1 1 1 1 12 60

```

```

&sv 38 "N"
&rem *****
&rem . Conditional loop to popup directions menu
&rem *****
&label WHILE
  &goto EXIT &if &eq %38 "Y"
  popup tdir.col sml
  &goto WHILE

```

```

&label EXIT
&return

```

```

&rem 30 1 5 40 10 13
&rem
&rem `DIRECTIONS
&rem `=====
&rem ON`SCREEN
&rem
&rem PRINT
&rem
&rem QUIT
&rem EOF

&rem *****
&rem   FileName:      TDIR.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        PRASC.EXE
&rem   CalledBY:     TDIR.SML
&rem   DataFiles:    tdir, directio
&rem   Variables:    %30 -> User selection from this menu
&rem                %38 -> Flag for conditional looping
&rem   Purpose:      Allows the user to display directions on screen
&rem                or to print the directions using an external
&rem                program PRASC.EXE using temporary files 'tdir'
&rem                and 'directio'
&rem *****

&jump %30

&rem *****
&rem   Popup 'tdir' file to show directions on screen window
&rem *****
&label ON`SCREEN
  popup tdir 1 1 1 1 12 60
  &sv 38 "N"
  &return

&rem *****
&rem   Call PRASC.EXE to print directions on the printer using file
&rem   'directio'
&rem *****
&label PRINT
  &sys "prasc directio"
  &sv 38 "N"
  &return

&label QUIT
  &sv 38 "Y"
  &return

```

```

&rem 25 1 5 40 18 14
&rem 'ROUTE'COLOR
&rem '=====
&rem WHITE
&rem RED
&rem GREEN
&rem BLUE
&rem YELLOW
&rem CYAN
&rem MAGENTA
&rem DARK'GRAY
&rem LIGHT'GRAY
&rem LIGHT'RED
&rem LIGHT'GREEN
&rem LIGHT'BLUE
&rem DARK'YELLOW
&rem LIGHT'CYAN
&rem LIGHT'MAGENTA
&rem
&rem EOF

&rem *****
&rem   FileName:      ROUTECOL.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     PDROUTEM.COL
&rem   DataFiles:    None
&rem   Variables:    %25 -> Stores user selection from this menu
&rem   Purpose:      Allows the user to assign a color to a new route
&rem *****

&jump %25

&label WHITE
  &cv 25 1
  &goto end_color

&label RED
  &cv 25 2
  &goto end_color

&label GREEN
  &cv 25 3
  &goto end_color

&label BLUE
  &cv 25 4
  &goto end_color

&label YELLOW
  &cv 25 5
  &goto end_color

&label CYAN
  &cv 25 6
  &goto end_color

&label MAGENTA
  &cv 25 7
  &goto end_color

```

```
&label DARK_GRAY
  &cv 25 8
  &goto end_color

&label LIGHT_GRAY
  &cv 25 9
  &goto end_color

&label LIGHT_RED
  &cv 25 10
  &goto end_color

&label LIGHT_GREEN
  &cv 25 11
  &goto end_color

&label LIGHT_BLUE
  &cv 25 12
  &goto end_color

&label DARK_YELLOW
  &cv 25 13
  &goto end_color

&label LIGHT_CYAN
  &cv 25 14
  &goto end_color

&label LIGHT_MAGENTA
  &cv 25 15
  &goto end_color

&label end_color
  &return
```

```

&rem 24 1 5 40 15 9
&rem `ROUTE`#
&rem `=====
&rem ONE
&rem TWO
&rem THREE
&rem FOUR
&rem FIVE
&rem SIX
&rem SEVEN
&rem EIGHT
&rem NINE
&rem TEN
&rem ELEVEN
&rem TWELVE
&rem EOF

&rem *****
&rem   FileName:      ROUTENUM.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     PDROUTEM.COL
&rem   DataFiles:    None
&rem   Variables:    %24 -> User selection from this menu
&rem   Purpose:      Allows the user to assign a number to a new route
&rem *****

&jump %24

&label ONE
  &cv 24 1
  &goto end_number

&label TWO
  &cv 24 2
  &goto end_number

&label THREE
  &cv 24 3
  &goto end_number

&label FOUR
  &cv 24 4
  &goto end_number

&label FIVE
  &cv 24 5
  &goto end_number

&label SIX
  &cv 24 6
  &goto end_number

&label SEVEN
  &cv 24 7
  &goto end_number

&label EIGHT
  &cv 24 8
  &goto end_number

```

```
&label NINE
  &cv 24 9
  &goto end_number

&label TEN
  &cv 24 10
  &goto end_number

&label ELEVEN
  &cv 24 11
  &goto end_number

&label TWELVE
  &cv 24 12
  &goto end_number

&label end_number
  &return
```

```

&rem 43 1 5 59 9 16
&rem `UTILITIES
&rem `=====
&rem IDENTIFY`STREET
&rem
&rem SHOW`LOCATION
&rem
&rem ZOOM`IN/OUT
&rem
&rem QUIT
&rem EOF

&rem *****
&rem   FileName:      PDUTIL.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        PDMAPZ.COL, PDLOCAT.SML
&rem   CalledBy:     PDMENU.MNU
&rem   DataFiles:    STREETID, WHEREIS
&rem   Variables:    %14 -> Name of main intersecting road, if specified
&rem                %15 -> Name of cross street, if specified
&rem                %17 -> Street number, if specified
&rem                %21 -> Name of coverage
&rem                %43 -> User selection from menu
&rem   Purpose:      Utilities common to INCMAN, TRAVEL TIME SIMULATION
&rem                and FRONTAGE ROAD ROUTING. Allow the user to
&rem                (a) show location on the screen by specifying the
&rem                address or intersection, or (b) list the street
&rem                name and address information upon selection of
&rem                street segment with mouse, or (c) zoom in or zoom
&rem                out on the map displayed.
&rem *****

&jump %43

&rem *****
&rem   Calls PDMAPZ.COL to popup a 'Zoom' menu; the user may choose
&rem   to zoom in or zoom out; the screen is redrawn as per selection
&rem *****
&label ZOOM`IN/OUT
  popup pdmapz.col sml
  &goto QUIT

&rem *****
&rem   The user selects a street segment or arc with the mouse;
&rem   the street name and address information is written to a
&rem   file STREETID and this file is then displayed in a window
&rem *****
&label IDENTIFY`STREET
  &openw streetid
  &write "STREET IDENTIFICATION"

  &rem *****
  &rem   Select arc and write address information
  &rem *****
  identify %21 arcs * street_nam f_add t_add from_st to_st class
  &write
  &write "Click Anywhere to Exit"
  &closew streetid

&rem *****

```



```

&rem Show STREETID file on screen
&rem *****
popup streetid 1 1 1 1 10 25
&goto QUIT

&rem *****
&rem Calls PDLOCAT.COL for the user to specify the location.
&rem The user specified location is then displayed using symbol 16
&rem At least one road with street number must be specified or
&rem two intersecting roads must be specified.
&rem *****
&label SHOW'LOCATION
  &sv 31 'ANYLOC'
  &run pdloc

  &rem *****
  &rem If the main street name is not
  &rem specified process separately
  &rem write to file WHEREIS
  &rem *****
  &goto WHERE'IS &if &eq %14 ""
  &goto WHEREIS'MAIN &if &eq %15 ""
  &goto QUIT &if &ne %17 ""
  &openw whereis
  whereis 16 %17 %14/%15
  &closew whereis
  &goto POP_MATCH

&rem *****
&rem If 'NO MATCH' is found then popup an appropriate
&rem message after checking for above in the WHEREIS file
&rem *****
&label POP_MATCH
  &open whereis
  &cv 22 1
  &label WHERELOOP
    &read 16 NOTREAD
    &label EXTRACTLOOP
    &extract 17 16 %22
    &cv 22 %22 + 1
    &goto POP_MESSAGE &if &eq %17 "NO"
    &goto EXTRACTLOOP &if &rn %22 1 9
    &goto WHERELOOP

&rem *****
&rem Pop message in case of 'NO MATCH'
&rem *****
&label POP_MESSAGE
  &close whereis
  popup whereis 1 1 1 1 4 40
  &label NOTREAD
  &close whereis
  &goto QUIT

&rem *****
&rem Show the user specified location on screen when a street
&rem location is specified but which does not have the name of
&rem main intersecting arterial
&rem *****
&label WHERE'IS
  &openw whereis

```

```
whereis 16 %17 %15
&closew whereis
&goto POP_MATCH
```

```
&rem *****
&rem Show the user specified location on screen when a street
&rem location is specified but which does not have the name of
&rem the intersecting cross street.
&rem *****
```

```
&label WHEREIS'MAIN
  &openw whereis
  whereis 16 %17 %14
  &closew whereis
  &goto POP_MATCH
```

```
&rem *****
&rem Exit this menu
&rem *****
&label QUIT
  &return
```

```

&rem 30 1 5 58 8 12
&rem `ZOOM`IN/OUT
&rem `=====
&rem ZOOM`IN
&rem
&rem ZOOM`OUT
&rem
&rem QUIT
&rem
&rem EOF

&rem *****
&rem   FileName:      PDMAPZ.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        REDRAW.SML
&rem   CalledBy:     PDUTIL.COL
&rem   DataFiles:    None
&rem   Variables:    %21 -> Name of coverage
&rem                %30 -> User selection from menu
&rem   Purpose:     Allows the user to zoom in or zoom out on the
&rem                displayed map. The user is required to specify
&rem                the area to be zoomed in on using the mouse.
&rem                Zoom out is only possible to the original extents.
&rem *****

&jump %30

&rem *****
&rem   Zoom In using the mouse and redraw the screen
&rem *****
&label ZOOM`IN
  mapextent *
  &run redraw
  &goto QUIT

&rem *****
&rem   Zoom out to the original screen and redraw the screen
&rem *****
&label ZOOM`OUT
  mapextent %21
  &run redraw
  &goto QUIT

&label QUIT
  &return

```

```

&rem *****
&rem   FileName:      PDLOC.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        PDLOCAT.COL, PDLOCATI.COL, PDLOCATV.COL
&rem   CalledBy:     PDVANUP.COL, PDINCUP.COL, PDUTIL.COL
&rem   DataFiles:    None
&rem   Variables:    %14 -> Name of main street or freeway
&rem                %15 -> Name of cross street
&rem                %17 -> Street number of main street or freeway
&rem                %43 -> Flag for conditional looping to show menu
&rem   Purpose:      Calls popup menu for user specification of
&rem                location to be displayed.
&rem *****

&rem *****
&rem   Initialize variables for street names and numbers
&rem *****
&sv 17 ""
&sv 15 ""
&sv 14 ""
&sv 43 N

&rem *****
&rem   Loop to repeatedly pop up the menu for user specification of
&rem   a location to be displayed.
&rem *****
&label WHILE
  &goto EXIT &if &eq %43 "Y"
  &goto ANYLOC &if &eq %31 'ANYLOC'
  &goto INCLOC &if &eq %31 'INCLOC'
  &goto VANLOC &if &eq %31 'VANLOC'
  &goto WHILE

&label ANYLOC
  popup pdlocat.col sml
  &goto WHILE

&label INCLOC
  popup pdlocati.col sml
  &goto WHILE

&label VANLOC
  popup pdlocatv.col sml
  &goto WHILE

&label EXIT
  &return

```

```

&rem 43 1 5 58 12 20
&rem `LOCATION
&rem `=====
&rem STREET`NUMBER
&rem
&rem MAIN`STREET
&rem
&rem CROSS`STREET
&rem
&rem SHOW`NOW
&rem
&rem CANCEL
&rem EOF

&rem *****
&rem   FileName:      PDLOCAT.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        STREENUM, STREENAM
&rem   CalledBy:     PDLOC
&rem   DataFiles:    WARNING2
&rem   Variables:    %14 -> Name of main street or freeway
&rem                %15 -> Name of cross street
&rem                %17 -> Street number
&rem                %39 -> Flag for canceling display of location
&rem                %43 -> User selection from menu
&rem   Purpose:      This is similar to PDLOCATI.COL and PDLOCATV.COL
&rem                programs. This allows the user to enter the
&rem                location on the network. The location may be
&rem                either by address or by intersection. This menu
&rem                is repeatedly popped up for the user to enter
&rem                (a) the street number and name or (b) names of
&rem                two intersecting streets. The location is then
&rem                displayed on the screen as a white square.
&rem *****

&jump %43

&rem *****
&rem   Popup the form for user entry of street number; if the user enters
&rem   "CANCEL" then cancel this operation
&rem *****
&label STREET`NUMBER
  popup streenum.frm sml
  &goto end_location

&rem *****
&rem   Popup the menu for user selection of main street or freeway name;
&rem   if the user enters "CANCEL" then cancel this operation
&rem *****
&label MAIN`STREET
  popup streenam.col sml
  &goto end_location

&rem *****
&rem   Popup the menu for user selection of cross street name;
&rem   if the user enters "CANCEL" then cancel this operation
&rem *****
&label CROSS`STREET
  popup streenam.col sml
  &goto end_location

```

```

&rem *****
&rem Flag set to 'N' so that this menu may be popped up again.
&rem *****
&label end_location
    &sv 43 N
    &return

&rem *****
&rem Check if main street or freeway name has not been specified
&rem else check for address conflict or show the location specified
&rem *****
&label SHOW_NOW
    &goto STREET_NAM &if &eq %14 ""
    &goto CHECK_MULTIPLE_NAMES &if &ne %17 ""
    &goto COMPLETE_UPDATE

&rem *****
&rem Check for address conflict
&rem *****
&label CHECK_MULTIPLE_NAMES
    &goto WARN_USER &if &ne %15 ""
    &goto COMPLETE_UPDATE

&rem *****
&rem User entry of location is valid; return to calling program to
&rem show the location specified by the user.
&rem *****
&label COMPLETE_UPDATE
    &sv 43 Y
    &return

&rem *****
&rem If neither the main street name nor the cross street name have
&rem been specified then display warning to the user.
&rem *****
&label STREET_NAM
    &goto WARN_USER &if &eq %15 ""
    &goto COMPLETE_UPDATE

&rem *****
&rem Popup the warning file on the screen
&rem *****
&label WARN_USER
    popup warning2 1 1 1 1 7 50
    &goto end_location

&rem *****
&rem Set appropriate flags and return to calling program
&rem *****
&label CANCEL
    &sv 43 Y
    &sv 39 CANCEL
    &return

```

```

&rem 43 1 5 3 9 16
&rem 'INCIDENTS
&rem '=====
&rem ADD'INCIDENT
&rem
&rem REMOVE'INCIDENT
&rem
&rem LIST'INCIDENTS
&rem
&rem QUIT
&rem EOF
&rem

```

```

&rem *****
&rem   FileName:      PDINCUP.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        LISTINC.EXE, LOCMATCH.EXE, ACOPY.EXE,
&rem                PDLOCATI.SML, PDLISTIN.SML, PDREDRAW.SML
&rem   CalledBy:     PDMENU.MNU
&rem   DataFiles:    matcha, inclist, m$i10, m$i00.sml, m$r00.sml
&rem   Variables:    %14 -> Freeway name for incident location
&rem                %15 -> Cross Street name for incident location
&rem                %17 -> Street number for incident location
&rem                %18 -> Variable to read line from MATCHA file
&rem                %19 -> 'DETECTEDMATCH' or 'CLEAREDMATCH' flag
&rem                %20 -> Symbol for display of incident location
&rem                %39 -> Flag for repeated popup of main PDMENU
&rem                %43 -> User selection from this menu.
&rem   Purpose:      This is the main routine for display of locations
&rem                of incidents and listing of the incidents on the
&rem                screen. The program calls SML as well as C programs
&rem                to process and display the incidents in a red square
&rem                with a red cross inscribed in it. This program has
&rem                somewhat convoluted design as several temporary
&rem                files are used to accomplish the tasks.
&rem *****

```

```
&jump %43
```

```

&rem *****
&rem   This segment follows ADD'INCIDENT and CLEAR'INCIDENT
&rem   Execution transfers to DETECTEDMATCH or CLEAREDMATCH
&rem   depending on the setting of the flag '%19'
&rem *****
&label VALIDLOCATION
  &openw matcha

  &rem *****
  &rem   Write symbol and address to file MATCHA
  &rem *****
  whereis %20 %17 %14/%15
  &closew matcha
  &open matcha NOTOPEN

  &rem *****
  &rem   Jump to DETECTEDMATCH or CLEAREDMATCH if MATCHA cannot
  &rem   be read; else display contents of MATCHA on screen.
  &rem *****
  &read 18 %19

```

```

&close matcha
popup matcha 1 1 1 1 5 40
&goto QUIT

&rem *****
&rem  Calls PDLOCATI.SML for user to specify an incident location
&rem  Set incident flag '%19' to 'DETECTEDMATCH'.
&rem *****
&label ADD'INCIDENT
  &sv 31 'INCLOC'
  &run pdloc
  &goto QUIT &if &eq %39 "CANCEL"
  &sv 19 DETECTEDMATCH
  &sv 20 1
  &goto VALIDLOCATION

&rem *****
&rem  Display the incident location as well as write SML
&rem  instruction to file m$i00.sml for future reference
&rem *****
&label DETECTEDMATCH
  &close matcha
  &openw m$i00.sml A
  &write "WHEREIS 1 %17 %14/%15"
  &closew
  whereis 1 %17 %14/%15
  &goto QUIT

&rem *****
&rem  Calls PDLOCATI.SML for user to specify an incident location
&rem  that needs to be cleared; set incident flag '%19' to
&rem  'CLEAREDMATCH'.
&rem *****
&label REMOVE'INCIDENT
  &sv 31 'INCLOC'
  &run pdloc
  &goto QUIT &if &eq %39 "CANCEL"
  &sv 19 CLEAREDMATCH
  &sv 20 1
  &goto VALIDLOCATION

&rem *****
&rem  Add incident location to be cleared to file m$r00.sml ;
&rem  run LOCMATCH.EXE to remove this from m$i00.sml
&rem  and redraw the screen.
&rem *****
&label CLEAREDMATCH
  &close matcha
  &openw m$r00.sml A
  &write "WHEREIS 1 %17 %14/%15"
  &closew
  &sys "locmatch"
  &run pdredraw
  &goto QUIT

&rem *****
&rem  Calls C program LISTINC.EXE and copies from temporary file
&rem  to INCLIST data file; Calls PDLISTIN
&rem *****
&label LIST'INCIDENTS

```



```
&sys "listinc"  
&sys "acopy m$i10 inclist"  
&run pdlistin  
&goto QUIT  
  
&label NOTOPEN  
&type "File Cannot be Opened !!!"  
&goto QUIT  
  
&label QUIT  
&return
```

```

&rem 43 1 5 3 12 20
&rem 'LOCATION
&rem '=====
&rem FREEWAY'NUMBER
&rem
&rem FREEWAY'NAME
&rem
&rem CROSS'STREET
&rem
&rem UPDATE'LOCATION
&rem
&rem CANCEL
&rem EOF

```

```

&rem *****
&rem   FileName:      PDLOCATI.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        STREENUM, FREENAM, STREENAM
&rem   CalledBy:     PDLOC
&rem   DataFiles:    WARNING1
&rem   Variables:    %14 -> Name of main street or freeway
&rem                %15 -> Name of cross street
&rem                %17 -> Street number
&rem                %39 -> Flag for canceling display of location
&rem                %43 -> User selection from menu
&rem   Purpose:      This is similar to PDLOCAT.COL and PDLOCATV.COL
&rem                programs. This allows the user to enter the
&rem                location of an incident. The location may be
&rem                either by address or by intersection. This menu
&rem                is repeatedly popped up for the user to enter
&rem                (a) the street number and name or (b) names of
&rem                two intersecting streets. The incident is then
&rem                displayed on the screen in a red square.
&rem *****

```

```

&jump %43

```

```

&rem *****
&rem   Popup the form for user entry of street number; if the user enters
&rem   "CANCEL" then cancel this operation
&rem *****

```

```

&label FREEWAY'NUMBER
  popup streenum 29 2 5 3 5 21
  &goto end_location

```

```

&rem *****
&rem   Popup the menu for user selection of main street or freeway name;
&rem   if the user enters "CANCEL" then cancel this operation
&rem *****

```

```

&label FREEWAY'NAME
  popup freenam 14 1 5 3 18 21
  &goto end_location

```

```

&rem *****
&rem   Popup the menu for user selection of cross street name;
&rem   if the user enters "CANCEL" then cancel this operation
&rem *****

```

```

&label CROSS'STREET
  popup streenam 15 1 5 3 18 21
  &goto end_location

```

```

&rem *****
&rem Flag set to 'N' so that this menu can be repeatedly popped up
&rem *****
&label end_location
    &sv 43 N
    &return

&rem *****
&rem Check if freeway name has not been specified or if freeway number
&rem is not specified else show the location specified
&rem *****
&label UPDATE_LOCATION
    &goto WARN_USER &if &eq %14 ""
    &goto FREEWAY_NUM &if &ne %17 ""
    &goto COMPLETE_UPDATE

&rem *****
&rem User entry of location is valid; return to calling program to show
&rem the location specified for the incident.
&rem *****
&label COMPLETE_UPDATE
    &sv 43 Y
    &return

&rem *****
&rem If neither the main street name nor the cross street name have
&rem been specified then display warning to the user.
&rem *****
&label FREEWAY_NUM
    &goto WARN_USER &if &ne %15 ""
    &goto COMPLETE_UPDATE

&rem *****
&rem Pop up the warning file on the screen
&rem *****
&label WARN_USER
    popup warning1 1 1 1 1 7 50
    &goto end_location

&rem *****
&rem Set appropriate flags and return to calling program
&rem *****
&label CANCEL
    &sv 43 Y
    &sv 39 CANCEL
    &return

```

```

&rem *****
&rem   FileName:      PDLISTIN.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        PDLISTIN.COL
&rem   CalledBy:     PDINCUP.COL
&rem   DataFiles:    m$i10
&rem   Variables:    %38 -> Flag for repeatedly displaying, in a loop,
&rem                                     PDLISTIN.COL menu
&rem   Purpose:      Repeatedly displays the PDLISTIN.COL menu
&rem *****

&rem *****
&rem   Write message to temporary file
&rem *****
&openw m$i10 A
&write
&write "Click Anywhere to Exit"
&closew
popup m$i10 1 1 1 1 10 45

&rem *****
&rem   Set conditional loop flag to 'N' and
&rem   thereafter display in a loop the menu
&rem   PDLISIN.COL until the flag is set.
&rem *****
&sv 38 "N"
&label WHILE
  &goto EXIT &if &eq %38 "Y"
  popup pdlistin.col sml
  &goto WHILE

&label EXIT
  &return

```

```

&rem 30 1 5 19 9 16
&rem
&rem `LIST` INCIDENTS
&rem `=====
&rem ON`SCREEN
&rem
&rem PRINT
&rem
&rem QUIT
&rem EOF

&rem *****
&rem   FileName:      PDLISTIN.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        PRASC.EXE
&rem   CalledBy:     PDLISTIN.SML
&rem   DataFiles:    m$i10, inclist
&rem   Variables:    %38 -> Flag for repeatedly displaying, in a loop,
&rem                                     PDLISTIN.COL menu
&rem                                     %30 -> User selection from this menu
&rem   Purpose:      Allows the user to list the status of incidents
&rem                                     on the screen or as a printed hard copy.
&rem *****

&jump %30

&rem *****
&rem   List van status on screen using temporary file
&rem *****
&label ON`SCREEN
  popup m$i10 1 1 1 1 10 45
  &sv 38 "N"
  &return

&rem *****
&rem   Print van status on printer using file INCLIST
&rem *****
&label PRINT
  &sys "prasc inclist"
  &sv 38 "N"
  &return

&label QUIT
  &sv 38 "Y"
  &return

```

```

&rem 44 1 5 16 9 15
&rem `VAN`UPDATE
&rem `=====
&rem ASSISTING`VAN
&rem
&rem PATROLLING`VAN
&rem
&rem LIST`STATUS
&rem
&rem QUIT
&rem EOF

&rem *****
&rem   FileName:      PDVANUP.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        PDLOCATV.SML, PDUNIT.COL, PDREDRAW.SML, PDLISTV.SML
&rem                FINDVAN.EXE, LISTVAN.EXE, ACOPY.EXE
&rem   CalledBy:     PDMENU.MNU
&rem   DataFiles:    m$V10, matcha, m$b00.sml, m$vb00.sml, m$f00.sml,
&rem                m$vf00.sml, vanlist
&rem   Variables:    %1 ->
&rem                %2 ->
&rem                %3 ->
&rem                %14 -> Freeway name for van location
&rem                %15 -> Cross street name for van location
&rem                %16 -> MAP van unit number
&rem                %17 -> Street number for van location
&rem                %18 -> Variable to read from MATCHA file
&rem                %19 -> Flag for 'PATROLMATCH' or 'ASSISTMATCH'
&rem                %20 -> Symbol for display of van location
&rem                %39 -> Flag for repeated popup of main PDMENU
&rem                %44 -> User selection from this menu
&rem   Purpose:
&rem *****

&jump %44

&rem *****
&rem   Calls PDLOCATV.SML for user to specify a van location
&rem   Set van flag '%19' to 'ASSISTMATCH'. Gets a van number from
&rem   PDUNIT.COL and writes to temporary file M$V10.
&rem *****
&label ASSISTING`VAN
  popup pdunit.col sml
  &sv 31 `VANLOC'
  &run pdloc
  &goto QUIT &if &eq %16 "CANCEL"
  &goto QUIT &if &eq %39 "CANCEL"
  &openw m$V10
  &write "%16"
  &closew
  &sv 19 ASSISTMATCH
  &sv 20 3
  &goto VALIDLOCATION

&rem *****
&rem   This segment follows ASSISTING`VAN and PATROLLING`VAN
&rem   Execution transfers to ASSISTMATCH or PATROLMATCH
&rem   depending on the setting of the flag '%19'
&rem *****
&label VALIDLOCATION

```

```

&openw matcha

&rem *****
&rem Write symbol and address to file MATCHA
&rem *****
whereis %20 %17 %14/%15
&closew matcha
&openw matcha NOTOPEN

&rem *****
&rem Jump to ASSISTMATCH or PATROLMATCH if MATCHA cannot
&rem be read; else display contents of MATCHA on screen.
&rem *****
&read 18 %19
&closew matcha
popup matcha 1 1 1 1 5 40
&goto QUIT

&rem *****
&rem Run FINDVAN.EXE to update van location using temporary files
&rem Display the incident location as well as write SML
&rem instruction to file m$b00.sml and m$vb00.sml for
&rem future reference
&rem *****
&label ASSISTMATCH
&closew matcha
&rem &getxyc 1 2 3
&rem &openw vanxy A
&rem &write "%1 %2 %16"
&rem &closew
&sys "findvan"
&run pdredraw
&openw m$vb00.sml A
&write "%16 WHEREIS 3 %17 %14/%15"
&closew
&openw m$b00.sml A
&write "WHEREIS 3 %17 %14/%15"
&closew
whereis 3 %17 %14/%15
&goto QUIT

&rem *****
&rem Calls PDLOCATV.SML for user to specify a van location
&rem Set van flag '%19' to 'PATROLMATCH'. Gets a van number from
&rem PDUNIT.COL and writes to temporary file M$V10.
&rem *****
&label PATROLLING`VAN
popup pdunit.col sml
&sv 31 `VANLOC'
&run pdloc
&goto QUIT &if &eq %16 "CANCEL"
&goto QUIT &if &eq %39 "CANCEL"
&openw m$v10
&write "%16"
&closew
&sv 19 PATROLMATCH
&sv 20 2
&goto VALIDLOCATION

&rem *****
&rem Run FINDVAN.EXE to update van location using temporary files
&rem Display the incident location as well as write SML

```

```

&rem instruction to file m$f00.sml and m$vf00.sml for
&rem future reference
&rem *****
&label PATROLMATCH
  &close matcha
  &sys "findvan"
  &run pdredraw
  &openw m$vf00.sml A
  &write "%16 WHEREIS 2 %17 %14/%15"
  &closew
  &openw m$f00.sml A
  &write "WHEREIS 2 %17 %14/%15"
  &closew
  whereis 2 %17 %14/%15
  &goto QUIT

&rem *****
&rem Calls C program LISTVAN.EXE and copies from temporary file
&rem to VANLIST data file; Calls PDLISTV
&rem *****
&label LIST`STATUS
  &sys "listvan"
  &sys "acopy m$fb00 vanlist"
  &run pdlistv
  &goto QUIT

&label NOTOPEN
  &type "File Cannot be Opened !!!"
  &goto QUIT

&label QUIT
  &return

```



```

&rem 16 1 5 19 17 16
&rem `VAN`ID`NUMBER
&rem `=====
&rem CANCEL
&rem `-----
&rem 4051
&rem 4052
&rem 4053
&rem 4054
&rem 4055
&rem 4056
&rem 4057
&rem 4058
&rem 4059
&rem 4060
&rem 4061
&rem 4062
&rem 4063
&rem 4064
&rem 4065
&rem 4066
&rem 4067
&rem 4068
&rem EOF

&rem *****
&rem   FileName:      PDUNIT.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        none
&rem   CalledBy:     PDVANUP.COL
&rem   DataFiles:    none
&rem   Variables:    %16 -> User selection of a van number,
&rem   Purpose:      Allows the user to specify the van of which the
&rem                 location or status may have changed.
&rem *****

&return

```

```

&rem 43 1 5 19 12 20
&rem `LOCATION
&rem `=====
&rem FREEWAY`NUMBER
&rem
&rem FREEWAY`NAME
&rem
&rem CROSS`STREET
&rem
&rem UPDATE`LOCATION
&rem
&rem CANCEL
&rem EOF

&rem *****
&rem   FileName:      PDLOCATV.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        STREENUM, FREENAM, STREENAM
&rem   CalledBy:     PDLOC
&rem   DataFiles:    WARNING1
&rem   Variables:    %14 -> Name of main street or freeway
&rem                %15 -> Name of cross street
&rem                %17 -> Street number
&rem                %39 -> Flag for canceling display of location
&rem                %43 -> User selection from menu
&rem   Purpose:      This is similar to PDLOCAT.COL and PDLOCATI.COL
&rem                programs. This allows the user to enter the
&rem                location of a van. The location may be specified
&rem                either by address or by intersection. This menu
&rem                is repeatedly popped up for the user to enter
&rem                (a) the street number and name or (b) names of
&rem                two intersecting streets. The van is then
&rem                displayed on the screen as a yellow or a white
&rem                star depending on its operational status.
&rem *****

&jump %43

&rem *****
&rem   Popup the form for user entry of street number; if the user enters
&rem   "CANCEL" then cancel this operation
&rem *****
&label FREEWAY`NUMBER
  popup streenum 29 2 5 19 5 21
  &goto end_location

&rem *****
&rem   Popup the menu for user selection of main street or freeway name;
&rem   if the user enters "CANCEL" then cancel this operation
&rem *****
&label FREEWAY`NAME
  popup freenam 14 1 5 19 18 21
  &goto end_location

&rem *****
&rem   Popup the menu for user selection of cross name;
&rem   if the user enters "CANCEL" then cancel this operation
&rem *****
&label CROSS`STREET
  popup streenam 15 1 5 19 18 21
  &goto end_location

```

```

&rem *****
&rem Flag is set to 'N' so that this menu can be repeatedly popped up.
&rem *****
&label end_location
    &sv 43 N
    &return

&rem *****
&rem Check if main street or freeway name has not been specified
&rem else show the location specified
&rem *****
&label UPDATE_LOCATION
    &goto WARN_USER &if &eq %14 ""
    &goto FREEWAY_NUM &if &ne %17 ""
    &goto COMPLETE_UPDATE

&rem *****
&rem Valid user entry of location; return to calling program to show
&rem the location specified by the user.
&rem *****
&label COMPLETE_UPDATE
    &sv 43 Y
    &return

&rem *****
&rem If neither the main street name nor the cross street name have
&rem been specified then display warning to the user.
&rem *****
&label FREEWAY_NUM
    &goto WARN_USER &if &ne %15 ""
    &goto COMPLETE_UPDATE

&rem *****
&rem Popup the warning file on the screen
&rem *****
&label WARN_USER
    popup warning1 1 1 1 1 7 50
    &goto end_location

&rem *****
&rem Set appropriate flags and return to calling program
&rem *****
&label CANCEL
    &sv 43 Y
    &sv 39 CANCEL
    &return

```

```

&rem *****
&rem   FileName:      PDLISTV.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        PDLISTV.COL
&rem   CalledBy:     PDVANUP.COL
&rem   DataFiles:    m$fb00
&rem   Variables:    %38 ->  Flag for repeatedly displaying, in a loop,
&rem                                     PDLISTV.COL menu
&rem   Purpose:      Repeatedly displays the PDLISTV.COL menu
&rem *****

&rem *****
&rem   Write message to temporary file
&rem *****
&openw m$fb00 A
&write
&write "Click Anywhere to Exit"
&closew
popup m$fb00 1 1 1 1 16 45

&rem *****
&rem   Set conditional loop flag to 'N' and
&rem   thereafter display in a loop the menu
&rem   PDLISTV.COL until the flag is set.
&rem *****
&sv 38 "N"
&label WHILE
    &goto EXIT &if &eq %38 "Y"
    popup pdlistv.col sml
    &goto WHILE

&label EXIT
&return

```

```

&rem 30 1 5 3 10 13
&rem
&rem `LIST`VANS
&rem `=====
&rem ON`SCREEN
&rem
&rem PRINT
&rem
&rem QUIT
&rem EOF

&rem *****
&rem   FileName:      PDLISTV.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        PRASC.EXE
&rem   CalledBy:     PDLISTV.SML
&rem   DataFiles:    m$fb00, vanlist
&rem   Variables:    %38 -> Flag for repeatedly displaying, in a loop,
&rem                                     PDLISTV.COL menu
&rem                                     %30 -> User selection from this menu
&rem   Purpose:      Allows the user to re-display the status of vans
&rem                                     on the screen or as a printed hard copy.
&rem *****

&jump %30

&rem *****
&rem   List van status on screen using temporary file
&rem *****
&label ON`SCREEN
  popup m$fb00 1 1 1 1 16 45
  &sv 38 "N"
  &return

&rem *****
&rem   Print van status on printer using file VANLIST
&rem *****
&label PRINT
  &sys "prasc vanlist"
  &sv 38 "N"
  &return

&label QUIT
  &sv 38 "Y"
  &return

```

`STREET`NUMBER

`NUMBER`:%17

OK

Comments: File Name **STREENUM**

```
`FREEWAY`NAME
`=====
CANCEL
`-----
BELTWAY`8N
BELTWAY`8W
EAST`FWY
EASTEX`FWY
GULF`FWY
HARDY`TOLLROAD
KATY`FWY
LA`PORTE`FWY
LOOP`610E
LOOP`610N
LOOP`610S
LOOP`610W
MEMORIAL`DR
NORTH`FWY
NORTHWEST`FWY
SOUTH`FWY
SOUTHWEST`FWY
```

Comments: File Name **FREENAM**

` CROSS ` STREET  
`-----  
CANCEL  
`-----  
43RD ` CLAY ` RD  
AIRLINE  
ALDINE ` BENDER  
ALDINE ` WESTFIELD  
ALIEF ` CLODINE  
ALLEN ` PKWY  
ALMEDA ` RD  
ANTOINE  
ATASCOCITA ` RD  
BARKER ` CYPRESS ` RD  
BELLAIRE ` BLVD  
BELLFORT  
BELLFORT ` W  
BELTWAY ` 8N  
BELTWAY ` 8W  
BERRY ` RD  
BINGLE  
BISSONNET  
BROADWAY  
BROMPTON ` RD  
CAVALCADE  
CAVALCADE / 20TH  
CLAY ` RD  
CLINTON ` DR  
CROSSTIMBERS ` E  
CULLEN  
EAST ` FWY  
EASTEX ` FWY  
ELDRIDGE  
ELLA ` BLVD  
FM249  
FM149  
FM1960  
FM529  
FORTY ` THIRD ` W  
FULTON  
GALVESTON ` RD  
GESSNER ` S  
GULF ` FWY  
HARDY  
HARDY ` W  
HARDY ` TOLLROAD  
HARRISBURG  
HARWIN  
HEMPSTEAD  
HILLCROFT  
HILLCROFT / VOSS  
HOLCOMBE  
HOLCOMBE ` W  
IRVINGTON  
JENSEN ` DR  
JOHN ` F ` KENNEDY  
JONES ` RD  
KATY ` FWY  
KEMPWOOD  
KIRBY ` DR  
KUYKENDAHL  
LA ` PORTE ` FWY



LITTLE 'YORK  
LITTLE 'YORK 'W  
LOCKWOOD  
LOOP '184  
LOOP '610E  
LOOP '610N  
LOOP '610S  
LOOP '610W  
M. L. 'KING  
MACGREGOR 'WAY 'N  
MAIN 'N  
MAIN 'S  
MARTIN 'LUTHER 'KING  
MEMORIAL 'DR  
MOUNT 'HOUSTON  
NAVIGATION  
NORTH 'FWY  
NORTHWEST 'FWY  
OLD 'SPANISH 'TRAIL  
PARKER  
POST 'OAK 'S  
REVELLE  
RICHMOND 'AVE  
SAWYER  
SH '6  
SHEPHERD 'N  
SHEPHERD 'S  
SOUTH 'FWY  
SOUTHWEST 'FWY  
STUDEWOOD  
TANNER  
TELEPHONE 'RD  
THIRTY 'FOURTH 'W  
TIDWELL 'W  
TWENTIETH 'W  
US '90  
VETERANS 'MEMORIAL 'DR  
WALLISVILLE 'RD  
WAYSIDE 'S  
WESTHEIMER  
WESTPARK  
WHEATLEY  
WILL 'CLAYTON 'PKWY  
YALE

Comments: File Name **STREENAM**

```

&rem *****
&rem   FileName:      FILEBARR.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     PDMAPH.SML
&rem   DataFiles:    CENTERF
&rem   Variables:    %21 -> Name of coverage
&rem   Purpose:      Runs under TABLES module. Reads the AAT and
&rem                accesses information about the intersections
&rem                to be blocked as generated in NMAPALLO.SML.
&rem                This information is then written to an ASCII
&rem                file CENTERF which is subsequently read in the
&rem                ROUTE module for INCMAN, FRONTAGE ROAD ROUTING
&rem                or TRAVEL TIME SIMULATION applications to
&rem                generate alternate routes bypassing a blocked
&rem                hazardous area.
&rem *****

```

```

arc
clear
&run filevar
select %21.AAT

```

```

&rem *****
&rem   Read information for blocked intersections
&rem   and write to ASCII file CENTERF
&rem *****
reselect for CENTER ne 0
&openw centerf
list FNODE_
list TNODE_
&closew
q stop

```

```

&rem *****
&rem   FileName:      UNBLOCLN.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     PDMAPH.SML
&rem   DataFiles:    FILEVAR.SML
&rem   Variables:    %21 -> Name of coverage/map
&rem   Purpose:      Common to INCMAN, FRONTAGE ROAD ROUTING and
&rem                TRAVEL TIME SIMULATION
&rem                Program clears any and all lanes on the
&rem                coverage which have been previously blocked
&rem                and updates the impedances in both directions
&rem                using the volume to capacity ratios.
&rem                The program works under the TABLES module.
&rem *****

&run filevar
select %21.aat

&rem *****
&rem   Select only those records where one or
&rem   more lanes is blocked
&rem *****
reselect for f_av_lanes ne f_tot_lane

&rem *****
&rem   Reset the number of available lanes;
&rem   the available capacity and the volume to
&rem   capacity in the to-to-from direction
&rem *****
  calculate F_AV_LANES = F_TOT_LANE
  calculate F_AV_CAP = ( F_CAP_LANE * F_AV_LANES )
  calculate F_VOL_CAP = ( F_VOLUME / F_AV_CAP )

&rem *****
&rem   Calculate to-to-from impedance
&rem   based on volume to capacity ratios
&rem *****
  calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 0.25 ) * 0.9
  reselect for F_VOL_CAP < 1.25
  calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 0.4 ) * 0.9
  reselect for F_VOL_CAP < 1.0
  calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 0.5 ) * 0.9
  reselect for F_VOL_CAP < 0.85
  calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 0.7 ) * 0.9
  reselect for F_VOL_CAP < 0.7
  calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 0.85 ) * 0.9
  reselect for F_VOL_CAP < 0.5
  calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 1.0 ) * 0.9

&rem *****
&rem   Select all records again for reselection
&rem *****
select %21.aat

&rem *****
&rem   Select only those records where one or
&rem   more lanes is blocked
&rem *****
reselect for t_av_lanes ne t_tot_lane

```

```

&rem *****
&rem   Reset the number of available lanes;
&rem   the available capacity and the volume to
&rem   capacity in the from-to-to direction
&rem *****
  calculate T_AV_LANES = T_TOT_LANE
  calculate T_AV_CAP = ( T_CAP_LANE * T_AV_LANES )
  calculate T_VOL_CAP = ( T_VOLUME / T_AV_CAP )

&rem *****
&rem   Calculate from-to-to impedance
&rem   based on volume to capacity ratios
&rem *****
  calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 0.25 ) * 0.9
  reselect for T_VOL_CAP < 1.25
  calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 0.4 ) * 0.9
  reselect for T_VOL_CAP < 1.0
  calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 0.5 ) * 0.9
  reselect for T_VOL_CAP < 0.85
  calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 0.7 ) * 0.9
  reselect for T_VOL_CAP < 0.7
  calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 0.85 ) * 0.9
  reselect for T_VOL_CAP < 0.5
  calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 1.0 ) * 0.9

q stop

```

```

&rem *****
&rem   FileName:      TIME TAB.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:         TIME.MNU
&rem   CalledBy:     PDMAPH.SML
&rem   DataFiles:    None
&rem   Variables:    None
&rem   Purpose:      Program is common to INCMAN, TRAVEL TIME
&rem                 SIMULATION and FRONTAGE ROAD ROUTING.
&rem
&rem                 Program runs in the TABLES module and calls
&rem                 TIME.MNU to update impedance based on hourly
&rem                 volumes; primarily for travel time simulation.
&rem *****

arc
clear
&run filevar
popup %21\pulldown\time.mnu sml
q stop

```

```

&rem 36 2 1 20 6 40
&rem
&rem 'PLEASE' SELECT 'THE' TIME 'OF' THE 'DAY
&rem
&rem          AM                      PM
&rem EOF

&rem *****
&rem   FileName:      TIME.MNU
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        TIME_AM.COL, TIME_PM.COL
&rem   CalledBy:     TIME_TAB.SML
&rem   DataFiles:    None
&rem   Variables:    %36 -> User selection of AM or PM hours
&rem                %21 -> Name of coverage/map
&rem   Purpose:      Program is common to INCMAN, TRAVEL TIME
&rem                SIMULATION and FRONTAGE ROAD ROUTING.
&rem                The user may select 'AM' hours or 'PM' hours
&rem                depending on which the program will call
&rem                either TIME_AM.COL or TIME_PM.COL
&rem *****

&rem Execute the selected option

&jump %36

&label AM
  popup %21\pulldown\time_am.col sml
  &goto endfile

&label PM
  popup %21\pulldown\time_pm.col sml
  &goto endfile

&label endfile
  &return

```

```

&rem 1 1 1 65 15 14
&rem 'TIME'(AM)
&rem
&rem 1:00`AM
&rem 2:00`AM
&rem 3:00`AM
&rem 4:00`AM
&rem 5:00`AM
&rem 6:00`AM
&rem 7:00`AM
&rem 8:00`AM
&rem 9:00`AM
&rem 10:00`AM
&rem 11:00`AM
&rem NOON
&rem EOF

&rem *****
&rem   FileName:      TIME_AM.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        CALIMPED.SML
&rem   CalledBy:     TIME.MNU
&rem   DataFiles:    None
&rem   Variables:    %1 -> User selection of the hour
&rem                %36 -> Field name corresponding to the hour
&rem                in the from-to-to direction
&rem                %37 -> Field name corresponding to the hour
&rem                in the to-to-from direction
&rem   Purpose:      TIME_AM.COL and TIME_PM.COL are similar.
&rem                Call CALIMPED.SML to update the impedance
&rem                based on user selection made in this menu.
&rem *****

&jump %1

&label 1:00`AM
&sv 36 F_AM0100
&sv 37 T_AM0100
&goto endfile

&label 2:00`AM
&sv 36 F_AM0200
&sv 37 T_AM0200
&goto endfile

&label 3:00`AM
&sv 36 F_AM0300
&sv 37 T_AM0300
&goto endfile

&label 4:00`AM
&sv 36 F_AM0400
&sv 37 T_AM0400
&goto endfile

&label 5:00`AM
&sv 36 F_AM0500
&sv 37 T_AM0500
&goto endfile

&label 6:00`AM

```

```
&sv 36 F_AM0600
&sv 37 T_AM0600
&goto endfile

&label 7:00'AM
&sv 36 F_AM0700
&sv 37 T_AM0700
&goto endfile

&label 8:00'AM
&sv 36 F_AM0800
&sv 37 T_AM0800
&goto endfile

&label 9:00'AM
&sv 36 F_AM0900
&sv 37 T_AM0900
&goto endfile

&label 10:00'AM
&sv 36 F_AM1000
&sv 37 T_AM1000
&goto endfile

&label 11:00'AM
&sv 36 F_AM1100
&sv 37 T_AM1100
&goto endfile

&label NOON
&sv 36 F_PM1200
&sv 37 T_PM1200
&goto endfile

&label endfile
&run %21\pulldown\CALIMPED
&return
```



```

&rem *****
&rem   FileName:      CALIMPED.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     TIME_AM.COL, TIME_PM.COL
&rem   DataFiles:    None
&rem   Variables:    %21 -> Name of the coverage
&rem                %36 -> Field name, corresponding to hour
&rem                selected by the user, in the
&rem                from-to-to direction
&rem                %37 -> Field name, corresponding to hour
&rem                selected by the user, in the
&rem                to-to-from direction
&rem   Purpose:      Updates the impedance in both the from-to-to
&rem                and the to-to-from direction on the basis of
&rem                the hour selected by the user. The updation
&rem                algorithm uses the current-volume to capacity
&rem                ratios and relates these to the speeds based
&rem                on highway capacity manuals. The speeds and
&rem                distance are then used to estimate the travel
&rem                time which is the impedance value also. The
&rem                model has been adjusted for the real travel
&rem                speeds rather than the maximum permissible speed.
&rem                The model needs revision to more accurately
&rem                reflect the V/C curve (1) to have impedance
&rem                adjust for values between two V/C limits as now
&rem                and (2) to take into consideration the length of
&rem                the arcs for calculation of the impedance.
&rem *****

```

```
select %21.AAT
```

```

&rem *****
&rem   Update only those records which have hourly
&rem   volumes data.
&rem *****
reselect for F_am0700 ne 0

```

```

&rem *****
&rem   Update current volume in each direction using
&rem   the volume for the user selected hour. Then
&rem   Calculate the volume to capacity ratios in
&rem   each direction
&rem *****
calculate F_VOLUME = %36
calculate T_VOLUME = %37
calculate F_VOL_CAP = F_VOLUME / F_AV_CAP
calculate T_VOL_CAP = T_VOLUME / T_AV_CAP

```

```

&rem *****
&rem   Update from-to-to impedance using volume to
&rem   capacity ratios
&rem *****
calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 0.25 ) * 0.9
reselect for F_VOL_CAP LT 1.25
calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 0.4 ) * 0.9
reselect for F_VOL_CAP LT 1.00
calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 0.5 ) * 0.9
reselect for F_VOL_CAP LT 0.85
calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 0.7 ) * 0.9
reselect for F_VOL_CAP LT 0.70
calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 0.85 ) * 0.9

```

```

reselect for F_VOL_CAP LT 0.50
calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 1.0 ) * 0.9

&rem *****
&rem This needs to be done to again select all
&rem the records for updation in the from-to-to
&rem direction
&rem *****
select %21.AAT
&rem *****
&rem Update only those records which have hourly
&rem volumes data.
&rem *****
reselect for T_pm0500 NE 0

&rem *****
&rem Update to-to-from impedance using volume to
&rem capacity ratios
&rem *****
calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 0.25 ) * 0.9
reselect for T_VOL_CAP LT 1.25
calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 0.4 ) * 0.9
reselect for T_VOL_CAP LT 1.00
calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 0.5 ) * 0.9
reselect for T_VOL_CAP LT 0.85
calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 0.7 ) * 0.9
reselect for T_VOL_CAP LT 0.70
calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 0.85 ) * 0.9
reselect for T_VOL_CAP LT 0.50
calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 1.0 ) * 0.9

&RETURN

```

```

&rem 1 1 1 65 15 14
&rem `TIME` (PM)
&rem
&rem 1:00`PM
&rem 2:00`PM
&rem 3:00`PM
&rem 4:00`PM
&rem 5:00`PM
&rem 6:00`PM
&rem 7:00`PM
&rem 8:00`PM
&rem 9:00`PM
&rem 10:00`PM
&rem 11:00`PM
&rem MIDNIGHT
&rem EOF

```

```

&rem *****
&rem      FileName:      TIME_PM.COL
&rem      LastUpdate:   6/1/93
&rem      WrittenBy:    Nitin Vaidya
&rem      Calls:        CALIMPED.SML
&rem      CalledBy:     TIME.MNU
&rem      DataFiles:    None
&rem      Variables:    %1 -> User selection of hour
&rem                   %36 -> Field name corresponding to the hour
&rem                   in the from-to-to direction
&rem                   %37 -> Field name corresponding to the hour
&rem                   in the to-to-from direction
&rem      Purpose:      TIME_AM.COL AND TIME_PM.COL are similar.
&rem                   Call CALIMPED.SML to update the impedance
&rem                   based on user selection made in this menu.
&rem *****

```

```
&jump %1
```

```

&label 1:00`PM
&sv 36 F_PM0100
&sv 37 T_PM0100
&goto endfile

```

```

&label 2:00`PM
&sv 36 F_PM0200
&sv 37 T_PM0200
&goto endfile

```

```

&label 3:00`PM
&sv 36 F_PM0300
&sv 37 T_PM0300
&goto endfile

```

```

&label 4:00`PM
&sv 36 F_PM0400
&sv 37 T_PM0400
&goto endfile

```

```

&label 5:00`PM
&sv 36 F_PM0500
&sv 37 T_PM0500
&goto endfile

```

```
&label 6:00`PM
```

```
&sv 36 F_PM0600
&sv 37 T_PM0600
&goto endfile

&label 7:00`PM
&sv 36 F_PM0700
&sv 37 T_PM0700
&goto endfile

&label 8:00`PM
&sv 36 F_PM0800
&sv 37 T_PM0800
&goto endfile

&label 9:00`PM
&sv 36 F_PM0900
&sv 37 T_PM0900
&goto endfile

&label 10:00`PM
&sv 36 F_PM1000
&sv 37 T_PM1000
&goto endfile

&label 11:00`PM
&sv 36 F_PM1100
&sv 37 T_PM1100
&goto endfile

&label MIDNIGHT
&sv 36 F_AM1200
&sv 37 T_AM1200
&goto endfile

&label endfile
&run %21\pulldown\CALIMPED
&return
```

```

&rem *****
&rem   FileName:      NMAPALLO.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        MAPALLOC.FRM
&rem   CalledBy:     PDMAPH.SML
&rem   DataFiles:    None
&rem   Variables:    %21 -> Name of coverage/map
&rem                                     %31 -> Used temporarily to store a large number
&rem                                     such as 1000000 to bypass a limitation
&rem                                     of ALLOCATE command ADDCENTER
&rem                                     %32 -> Radius of area affected in miles
&rem   Purpose:      This program is executed under ALLOCATE. It draws
&rem                                     map and calls MAPALLOC.FRM for the user to enter
&rem                                     the center as well as the radius of the area
&rem                                     affected. Thereafter the
&rem                                     program shows the area to be blocked on the screen
&rem                                     writes the intersections to be blocked to the AAT
&rem                                     of the coverage.
&rem
&rem                                     A limitation of ALLOCATE is that it is designed for
&rem                                     bus routing type of applications where the radius
&rem                                     is an upper limit whereas for hazardous incidents
&rem                                     a lower limit is desirable. Another problem is that
&rem                                     a capacity is needed for bus routing but is not
&rem                                     required for incidents. To take care of this second
&rem                                     issue a large number is used for capacity in %31 so.
&rem                                     In reality, ALLOCATE needs refinement to handle
&rem                                     this application more appropriately
&rem *****

```

```

&run filevar
display 4 3
setpage 1 1
mapextent %21
readnetwork %21 t_miles f_miles
drawnetwork symbol

```

```

&rem *****
&rem   User must input the center of the area affected
&rem   as well as the radius of the area and the program
&rem   must display the area affected
&rem *****
popup mapallocate.frm sml
addcenter * %32 %31
run
grow

```

```

&rem *****
&rem   Add the information on the intersections to be
&rem   blocked to the AAT of the coverage/map
&rem *****
writeallocate center
q

```

```

&rem 30 2 1 24 6 31
&rem 'RADIUS'OF'AREA'AFFECTED
&rem
&rem '[miles]      :%32
&rem
&rem          OK CANCEL
&rem EOF

```

```

&rem *****
&rem   FileName:      MAPALLOC.FRM
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     NMAPALLO.SML
&rem   DataFiles:    None
&rem   Variables:    %30 -> User selection 'OK' or 'CANCEL'
&rem                %31 -> A high number such as 100000 used
&rem                to bypass limitation of ALLOCATE
&rem                command ADDCENTER in terms of
&rem                capacity
&rem                %32 -> Radius of area affected in miles
&rem   Purpose:      Accepts the input from the user on the radius of
&rem                the area affected due to a hazard. This input is
&rem                used in NMAPALLO.SML to determine the intersections
&rem                to be blocked for alternate routing.
&rem *****

```

```
&jump %30
```

```
&label OK
```

```

&rem *****
&rem   A temporary and high number for bypassing
&rem   ADDCENTER limitations
&rem *****
&cv 31 1000000
&return

```

```
&label CANCEL
```

```

&sv 32 ''
&sv 31 ''
&return

```

```

&rem *****
&rem   FileName:      NBLOCLAN.SML
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:         NBLOCLAN.COL, NBLOCKFT.COL, BLOKMORE.COL
&rem   CalledBy:     PDMAPH.SML
&rem   DataFiles:    None
&rem   Variables:    %21 -> Name of coverage/map
&rem                 %29 -> Flag for conditional looping
&rem                 to block more lanes as desired by user
&rem                 %34 -> Flag for the direction in which lane(s)
&rem                 are to be blocked as desired by user
&rem                 %37 -> Number of lanes to be blocked as desired
&rem                 by the user.
&rem   Purpose:      Common to INCMAN, FRONTAGE ROAD ROUTING and
&rem                 TRAVEL TIME SIMULATION. Runs under ARCEDIT module.
&rem                 This program may be used to block individual lanes
&rem                 on freeway segments in either direction. One or more
&rem                 lanes may be blocked at a time. The affect of such
&rem                 blocking is to reduce available capacity, the
&rem                 volume to capacity ratio and the speed; thereby
&rem                 increasing the impedance on the selected road
&rem                 segment in the specified direction.
&rem *****

&rem *****
&rem   read and draw the coverage in
&rem   color showing the orientation of
&rem   each arc with arrows
&rem *****
&run filevar
editcoverage %21
display 4 3
setpage 1 1
editfeature arcs
select all
calculate $SYMBOL = 2
drawenvironment arcs arrows
draw

&rem *****
&rem   initialize conditional looping flag to YES
&rem *****
&sv 29 YES

&rem *****
&rem   Conditional loop if lanes are to
&rem   to be blocked in more than one
&rem   segment
&rem *****
&label LOOP

&rem *****
&rem   Pop up the menu to select the direction
&rem   of the lanes to be blocked
&rem *****
popup nblockft.col sml

&rem *****
&rem   if the user does not specify direction
&rem   enquire if he wants to block lanes in
&rem   some other segment(s).

```

```

&rem *****
&goto MORE_LANES_TO_BLOCK &if &eq %34 "or'QUIT"

&rem *****
&rem Pop up the menu to select the direction
&rem of the lanes to be blocked
&rem *****
popup nbloclane.col sml

&rem *****
&rem if the user does not specify number of lanes
&rem enquire if he wants to block lanes in
&rem some other segment(s).
&rem *****
&goto MORE_LANES_TO_BLOCK &if &eq %37 "or'QUIT"

&rem *****
&rem Process lane blockage in the desired direction
&rem *****
&goto BLOCK_FROM_LANES &if &eq %34 "BLOCK'FROM'LANES"
&goto BLOCK_TO_LANES &if &eq %34 "BLOCK'TO'LANES"

&rem *****
&rem Process lane blockage in the to-to-from direction
&rem *****
&label BLOCK_FROM_LANES

&rem *****
&rem Zoom in on the segments to be blocked
&rem *****
mapex *
draw

&rem *****
&rem Select the desired street segments and
&rem reduce the number of available lanes,
&rem reduce the available capacity
&rem and reduce the volume to capacity ratio
&rem in the to-to-from direction
&rem *****
SELECT MANY
calculate F_AV_LANES = ( F_TOT_LANE - %37 )
calculate F_AV_CAP = ( F_CAP_LANE * F_AV_LANES )
calculate F_VOL_CAP = ( F_VOLUME / F_AV_CAP )

&rem *****
&rem update the impedance values using volume to
&rem capacity ratios in the to-to-from direction
&rem *****
calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 0.25 ) * 0.9
reselect for F_VOL_CAP < 1.25
calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 0.4 ) * 0.9
reselect for F_VOL_CAP < 1.0
calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 0.5 ) * 0.9
reselect for F_VOL_CAP < 0.85
calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 0.7 ) * 0.9
reselect for F_VOL_CAP < 0.7
calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 0.85 ) * 0.9
reselect for F_VOL_CAP < 0.5
calculate F_IMPEDANC = ( F_MILES * 60 ) / ( F_MAX_SPD * 1.0 ) * 0.9

```



```

&rem *****
&rem   check if the user wants to block any other
&rem   lanes in the same or any other street segments
&rem *****
&goto MORE_LANES_TO_BLOCK

&rem *****
&rem   Process lane blockage in the from-to-to direction
&rem *****
&label BLOCK_TO_LANES

&rem *****
&rem   Zoom in on the segments to be blocked
&rem *****
mapex *
draw

&rem *****
&rem   Select the desired street segments and
&rem   reduce the number of available lanes,
&rem   reduce the available capacity
&rem   and reduce the volume to capacity ratio
&rem   in the from-to-to direction
&rem *****
SELECT MANY
calculate T_AV_LANES = ( T_TOT_LANE - %37 )
calculate T_AV_CAP = ( T_CAP_LANE * T_AV_LANES )
calculate T_VOL_CAP = ( T_VOLUME / T_AV_CAP )

&rem *****
&rem   update the impedance values using volume to
&rem   capacity ratios in the from-to-to direction
&rem *****
calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 0.25 ) * 0.9
reselect for T_VOL_CAP < 1.25
calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 0.4 ) * 0.9
reselect for T_VOL_CAP < 1.0
calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 0.5 ) * 0.9
reselect for T_VOL_CAP < 0.85
calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 0.7 ) * 0.9
reselect for T_VOL_CAP < 0.7
calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 0.85 ) * 0.9
reselect for T_VOL_CAP < 0.5
calculate T_IMPEDANC = ( T_MILES * 60 ) / ( T_MAX_SPD * 1.0 ) * 0.9

&rem *****
&rem   check if the user wants to block any other
&rem   lanes in the same or any other street segments
&rem *****
&goto MORE_LANES_TO_BLOCK

&rem *****
&rem   Popup the menu BLOKMORE.COL to inquire if the
&rem   user desires to block other lanes in the same
&rem   or in other street segments; zoom out on the
&rem   map and redraw the map. If the user answers in
&rem   the affirmative then go to the top of the loop
&rem *****
&label MORE_LANES_TO_BLOCK
popup blokmore.col_sml

```

```
mapex %21  
draw  
&goto LOOP &if &equal %29 YES  
save  
q
```

```

&rem 34 1 1 59 9 20
&rem
&rem 'Direction'Of'Lanes
&rem
&rem BLOCK'FROM'LANES
&rem
&rem BLOCK'TO'LANES
&rem
&rem or'QUIT
&rem EOF

&rem *****
&rem   FileName:      NBLOCKFT.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     NBLOCLAN.SML
&rem   DataFiles:    None
&rem   Variables:    %34 -> Variable to store user selection of
&rem                   the direction of lane(s) to be blocked
&rem   Purpose:      Stores the selection by user in variable %34 for
&rem                   further processing in NBLOCLAN.SML
&rem *****

&label or'QUIT
&return

```

```

&rem 29 1 1 59 7 20
&rem
&rem 'BLOCK\MORE\LANES'?
&rem
&rem     YES
&rem
&rem     NO
&rem EOF

&rem *****
&rem   FileName:      BLOKMORE.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     NBLOCLAN.SML
&rem   DataFiles:    None
&rem   Variables:    %29 -> User selection "YES" or "NO"
&rem   Purpose:      Queries the user if any more lanes are to
&rem                 be blocked and stores it in variable %29
&rem                 for processing in NBLOCLAN.SML
&rem *****

&return

```

```

&rem 37 1 1 58 13 20
&rem
&rem `Num`Lanes`Block
&rem
&rem ONE
&rem
&rem TWO
&rem
&rem THREE
&rem
&rem FOUR
&rem
&rem or`QUIT
&rem
&rem EOF
&rem
&rem Execute the selected option

&rem *****
&rem   FileName:      NBLOCLAN.COL
&rem   LastUpdate:   6/1/93
&rem   WrittenBy:    Nitin Vaidya
&rem   Calls:        None
&rem   CalledBy:     NBLOCLAN.SML
&rem   DataFiles:    None
&rem   Variables:    %37 -> The number of lanes to be blocked as
&rem                                     selected by the user
&rem   Purpose:      Stores the number of lanes blocked in variable
&rem                                     %37 for further processing in NBLOCLAN.SML
&rem *****

&jump %37

&label ONE
  &cv 37 1
  &goto or`QUIT

&label TWO
  &cv 37 2
  &goto or`QUIT

&label THREE
  &cv 37 3
  &goto or`QUIT

&label FOUR
  &cv 37 4
  &goto or`QUIT

&label or`QUIT
  &return

```



## **APPENDIX E**

**A REPORT ON BENCHMARKING OF WORKSTATIONS FOR GIS**





## **ACKNOWLEDGEMENT**

This report was originally written as an internal TTI report, titled "A Report on Benchmarking of Workstations for GIS", by TTI staff Nitin Vaidya and Sabas Avila in July 1991.

## **INTRODUCTION**

With the emergence of Geographical Information Systems (GIS) in many industries, and the open architecture of many UNIX workstations, an endless combination of GIS packages and workstations can be used. However, while data on various standard tests on various workstations is available from different computer magazines and computer organizations, relatively little is known about the performance of these workstations for GIS.

Much of the benchmarking to date has been performed on software that requires operations which are either graphic, memory, or processing oriented. A GIS package is a combination of the most computer intensive processes of CAD (Computer Aided Design), and database software. For this reason, standard benchmarking tests may not be effective in evaluating the performance of GIS applications on these workstations. To better evaluate the relative merits of workstations made by various vendors, a need exists to benchmark these workstations for GIS applications, with emphasis on the functions required for applications in GIS for Transportation (GIS-T).

This report documents these efforts in that direction.

## **BACKGROUND**

GIS applications involve intensive computer processing of the CPU, disk, and graphics. Therefore, a GIS benchmark must include simultaneous operations of the CPU, hard drive, and graphics. Furthermore, the benchmark of these parameters must emphasize applications in transportation. For this reason, the areas to be evaluated include networking, CPU, disk, and graphic intensive applications as below:

**Networking applications.** Networks are systems of connected linear features that form a frame work in which resources flow. For example, a network can simulate a system of streets in which vehicles travel. NETWORK is a specialized product designed to provide network modelling and address matching capabilities within ARC/INFO. NETWORK serves two primary functions: routing and allocation. The ROUTING program determines optimal paths through the network. The ALLOCATE program finds the nearest center for links in the system.

The single most important task of a workstation involved in networking is reading the network. It is not uncommon for a network to have over 3000 arcs with attributes. Benchmarking of network applications would measure the ability of the workstation to read a network and successfully complete NETWORKING and ROUTING functions.

**CPU intensive applications.** GIS applications require memory intensive processes to read the network in order to complete a task in a timely manner. Therefore, the CPU benchmarking tests will measure the ability of the workstation to process both graphical and relational information.

**Disk intensive applications.** Since most GIS applications are data intensive which may exceed available memory, GIS applications frequently need to read from, and write to, disk. Disk benchmarking is, therefore, required.

**Graphics intensive applications.** ARC/INFO provides full cartographic output capabilities from simple screen displays to high quality cartographic plots to the monitor. It is latter which the workstations must support. Workstation graphics resolution must be high and the generation of plots should be quick.

## **BENCHMARKING METHOD**

ARC/INFO was the GIS development platform selected for use in benchmarking procedures. ARC/INFO is commercial software product released in mid-1980 by Environmental Systems Research Institute Inc. (ESRI). ARC/INFO was the most comprehensive GIS package for transportation, available on a variety of workstations and widely regarded as representative of GIS development platforms. ARC/INFO utilizes a relational database that supported dBASE-IV. The software included a macro language in which many commands could be written for benchmarking purposes. ARC/INFO also performed many function ideals to transportation, such as networking, geocoding, and address matching.

Based on studies conducted by other agencies, five leading vendors of workstations were shortlisted for benchmarking. These vendors are IBM, Sun Microsystems, HP, DEC, and Intergraph. While ESRI agreed to provide IBM and Sun workstations, HP, DEC, and Intergraph agreed to provide workstations for our benchmarking. However, Intergraph and DEC later declined to participate in the benchmarking process. The benchmarking tests were, therefore, conducted on IBM, HP, and Sun workstations.

Furthermore, the workstations were to meet the specifications listed in Table E-1. These specifications are representative of the requirements for GIS applications in transportation.

### **Benchmark Tests**

To maintain uniformity throughout benchmarking, the series of benchmark tests were performed to each workstation. These tests were a combination of macros written to perform ARC/INFO tasks.

The objectives of the benchmark were to measure the capabilities of the different workstations while running GIS applications. Three types of tests were devised:

Memory:	16 MB to 32 MB ( preferably 24 MB )
Disk Drive:	At least 300 MB available memory
Monitor:	19" or 16" color
Tape Drive:	150 MB streaming cartridge type
Windowing:	X-windows or equivalent
Price:	Maximum of \$35000

Table E-1. Workstation Specifications

1. Transportation Networks Tests,
2. General GIS Tests,
3. PC vs. Workstation Tests.

For each type of test, a weighted mean index was arrived at.

### **Index of Performance**

To compute an index of performances for the workstations, weights were assigned to each of the tasks of the three tests. These weights were assigned to reflect the expected use of the commands for applications. Approximate estimates of these weights have been made based on the past use of these commands in PC-ARC/INFO.

The performance times of each task of the tests are multiplied by these weights and the results are totalled. The resulting number reflects the performance time of the

workstations. As test times are used the lower the resulting number, the better the performance time of the machine.

The performance index is computed as follows:

1. The slowest time for each of the tests is multiplied by the corresponding weight and results for all tests are added up, and
2. The time for each test for each workstation is multiplied by the corresponding weight and results for each test are added, and
3. The total number for the slowest time is divided by the corresponding number for each of these machines to give the performance index for the machine.

The configurations of workstations and PC, that were benchmarked are given in Table E-2.

## **BENCHMARKING PROCEDURE**

### **Transportation Networks Tests**

This test was composed of those ARC/INFO commands which are essential for GIS applications in transportation.

### Workstation Configuration

#### SUN Sparc 2GX

Memory: 32 MB,  
Monitor: 16" (1152 X 900) color monitor  
Hard drive: 800 MB hard drive  
Windowing: SunView  
Miscellaneous: 2D graphics accelerator

#### HP 720 PVRX

Memory: 32 MB  
Monitor: 19" (1280 X 1024) color monitor  
Hard drive: 660 MB with only 150 MB available  
Windowing: X-windows  
Miscellaneous: 3D graphics accelerator

#### IBM RS 6000/320,

Memory: 16 MB  
Monitor: 19" (1280 X 1024) color monitor,  
Hard drive: 640 MB  
Windowing: X-windows

#### HP Vectra RS/25C Personal Computer

Memory: 8 MB  
Monitor: 12" SVGA color monitor  
Hard drive: 620 MB  
Windowing: None  
Miscellaneous: running ARC/INFO at DOS level

Table E-2. Workstation Configurations

Seven different tests were conducted on each of the three workstations. The same data set was used for each system. The data set as well as the tests have been provided by ESRI, Huntsville, Alabama. Each of these tests were timed to measure processing of the task. The following tests were conducted:

**Buffering and Polygon Overlaying.** The first in a series of GIS-T application needs tests included buffering and polygon overlay. Buffering is a spatial operation which determines spatial proximity or nearness of various geographic features. For example, in GIS-T, a buffer can be generated to identify the areas of a flooding zone or ROW of roadways. Polygon overlay is the process of merging overlapping polygons from two layers to create new polygons in an output layer. This process also combines the information used in the two polygon entities. This test was performed on a "parcel" coverage provided by ESRI. This is a CPU intensive task.

**Allocation of Arcs.** Allocation of arcs is the process of distributing links in the network to one or more centers. As links are assigned to the center, a portion of that center's resources are distributed to meet each link's demand. For example, in GIS-T, all streets within a ten mile radius of a chemical spill can be automatically identified. This test was performed on a "parcel" coverage provided by ESRI. This task, like buffering and polygon overlay, is also CPU intensive.

**Saving, to Disk, of a Coverage in Arcedit.** Arcedit is a unique graphics and database editor. It combines the capabilities of computer-aided design functions with a geographic database. These capabilities are important not only for automating and maintaining databases, but for the creation of high quality maps. This test was performed on coverage, provided by ESRI, with over 9000 arcs. This process is disk intensive.

**Clean a Coverage.** Clean is a program in ARC/INFO which edits and corrects geometric coordinate errors, assembles arcs into polygons, and creates feature attribute information for each polygon. This procedure is essential when creating any GIS application. Although this process is rarely used, this task must be undertaken at least once. This test was performed on coverage, provided by ESRI, with over 9000 arcs. Clean is a CPU intensive task.



**Plot or Redraw a Coverage.** The purpose of this task is to measure the speed in which a workstation could generate a drawing on the monitor and distinguish the best graphic outputs. This is important for those GIS-T applications involving real-time analysis. This test was performed on coverage, provided by ESRI, with over 9000 arcs. This a graphics intensive task.

**Route on a Coverage.** Probably the single most important task in GIS-T development is the routing applications. ROUTE is a path finding program used to model the movement of resources between two or more points. ROUTE will find optimal path between points. This is a fundamental task in GIS-T applications. This test was performed on a coverage, provided by ESRI, with about 3000 arcs. The task is CPU intensive.

This is relatively a simple task involving almost totally CPU processing. 80-99% of the process is CPU oriented, and 0-20% disk and graphics involved.

**Read Network.** Reading the network is the process reading the arcs of the network and its associated attributes into memory. This task determines speed in which the workstation can process an application. If a workstation is unable to read a network quickly or retain a considerable portion of the network in memory, much of a workstation's processing speed of this task is lost. This test was performed on a coverage, provided by ESRI, with about 3000 arcs. This procedure is disk intensive.

To compute a combined index of performance for the workstations, weights were assigned to each of the tests . Approximate estimates of these weights have been made based on the past use of these commands on PC-ARC/INFO for transportation GIS.

The weights are as follows:

Test #1: Buffering and Polygon Overlay: X 02

Test #2: Allocate Arcs:	X 08
Test #3: Save in ArcEdit:	X 10
Test #4: Clean Coverage:	X 05
Test #5: Plot/Redraw:	X 35
Test #6: Route:	X 25
Test #7: Read Network:	X 15

The performance times of each of these tests are multiplied by these weights and the results are added up. The resulting number reflects the performance of the workstations. As test times are used, the lower the resulting number, the better the performance of the machine.

Each of these tests involved timing with a stop watch. The benchmarking tests were video recorded for future reference.

### **General GIS Tests**

To measure the speed of a workstation to process GIS information, another of set of general tests was used in addition. These tests included task within ARC/INFO programs: ARCPLOT, ARCEDIT, INFO, and ARC.

These tests measure the degree of efficiency that the workstation perform instructions essential in developing the GIS maps and databases. Developing a GIS map and database requires continuous editing, updating, and other trouble shooting procedures. Within ARC/INFO exists four modules in which much of the developing procedures take place. The modules are ARCPLOT, ARCEDIT, ARC, and INFO.

These tests were provided by ESRI and were self timing, producing results in an output file. Moreover, the tests could be loaded only on SUN and IBM workstations as HP did not have a suitable tape drive. Hence, these were performed only on SUN and IBM workstations.

There were four tests:

Test #1 was a set of tasks in ARC PLOT

Test #2 was a set of tasks in ARCEDIT

Test #3 was a set of tasks at ARC level

Test #4 was a set of tasks in INFO

To compute a combined index of performance for the workstations, weights were assigned to each of the tests. Approximate estimates of these weights have been made based on the past use of these commands on PC-ARC/INFO.

The weights are as follows:

Test #1 X 05

Test #2 X 02

Test #3 X 01

Test #4 X 02

The performance times of each of these tests are multiplied by these weights and the results are added up. The resulting number reflects the performance of the workstations. As test times are used, the lower the resulting number, the better the performance of the machine.

The tests have been recorded on a video for future reference.

## PC vs. Workstation Tests

The final test compared a workstation to a personal computer to perform GIS in transportation applications. The same data set, a 3750 arcs coverage developed by TTI, was used on both the computers.

The tests were:

Test #1 Reading a Network

Test #2 Restoration of Network

Test #3 Routing on Network

To compute a combined index of performance for the workstations, weights were assigned to each of the tests. Approximate estimates of these weights have been made based on the past use of these commands on PC-ARC/INFO.

The following weights have been estimated for each of these tests:

Test #1: Reading of a Network	X 3
Test #2: Restoration of a Network	X 7
Test #3: Routing on a Network	X 5

The performance times of each of these tests are multiplied by these weights and the results are added up. The resulting number reflects the performance of the workstations. As test times are used, the lower the resulting number, the better the performance of the machine.

Each of these tests was timed with a stop watch and recorded on a video for future reference.

## OBSERVATIONS AND DISCUSSION

### Transportation Networks Tests

Performance ratios for each test are computed in the same way as the overall performance index and reflect the speed of the workstations. The observations have been tabulated and plotted graphically.

The following figures and tables show the observations for the tests:

1. Table E-3 shows the order of performance of the workstations in each test.
2. Table E-4 and Figure E-1 show the index of performance for each workstation.
3. Table E-5 shows the computation of the index of overall performance.
4. Table E-6 and Figure E-2 show the execution times for each of the seven tasks.
5. Table E-7 and Figure E-3 show the performance ratios for each task.

---

Buffering	HP	>>	IBM	>>	SUN
Allocation	IBM	>>	SUN	>>	HP
Restoration	IBM	>>	SUN	>>	HP
Saving	IBM	>>	HP	>>	SUN
Cleaning	SUN	>>	HP	>>	IBM
Network Reading	HP	>>	SUN	>>	IBM
Routing	HP	>>	SUN	>>	IBM

Table E-3. Transportation Network Tests Order of Performance

---

HP	1.57
SUN	1.42
IBM	1.33

Table E-4. Performance Indices

---

While each of these machines performed better in some area(s), HP 720 does appear to hold an edge overall. HP had the best times in 3 tests, IBM had the best times in 3 tests and SUN in 1 test.

However, the performance index of SUN is higher than that of IBM because of the weighted average for computation of performance index; the tests in which IBM did best are not the commands that are frequently used. On the other hand, SUN does best in the most frequently used commands. Tests 5,6 and 7 are commands which are used 75% of the time. In these tests, the IBM trails behind by a wide margin. This explains for SUN finishing marginally ahead of IBM overall.

The result for test #2 has limited accuracy. The reason is that this test could not be completed on HP workstation and so a correct time could not be obtained because of a system crash. As such, this time was estimated based on review of the video recording. The time for each of these machines was observed until a certain result was reached in each

BENCHMARK: TRANSPORT. NETWORK TESTS  
INDEX OF OVERALL PERFORMANCE

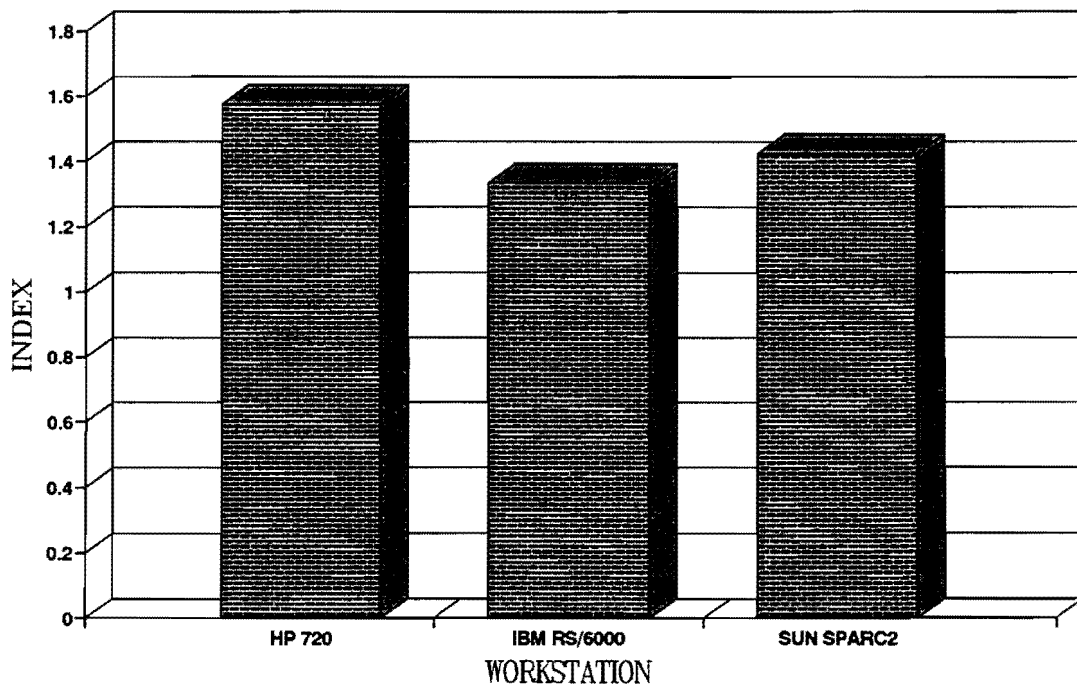


Figure E-1. Benchmark: Transportation Network Tests - Index of Performance

<b>BENCHMARK: TRANSPORTATION NETWORKS TESTS WEIGHTED TIMES &amp; PERFORMANCE INDEX</b>					
<b>FUNCTION</b>	<b>WEIGHTED FACTOR</b>	<b>WEIGHTED VALUES</b>			<b>EXTREME VALUES</b>
		<b>HP 720</b>	<b>IBM RS/600</b>	<b>SUN SPARC</b>	
<b>BUFFERING</b>	<b>2</b>	<b>115.94</b>	<b>121.32</b>	<b>123.26</b>	<b>123.26</b>
<b>ALLOCATION</b>	<b>8</b>	<b>560.00</b>	<b>351.52</b>	<b>441.04</b>	<b>560.00</b>
<b>RESTORATION</b>	<b>35</b>	<b>527.10</b>	<b>1003.10</b>	<b>480.20</b>	<b>1003.10</b>
<b>SAVING</b>	<b>10</b>	<b>130.60</b>	<b>37.50</b>	<b>39.70</b>	<b>130.60</b>
<b>CLEANING</b>	<b>5</b>	<b>349.20</b>	<b>292.70</b>	<b>742.20</b>	<b>742.20</b>
<b>NETWORK READI</b>	<b>15</b>	<b>37.05</b>	<b>69.75</b>	<b>69.30</b>	<b>69.75</b>
<b>ROUTING</b>	<b>25</b>	<b>223.25</b>	<b>427.50</b>	<b>258.50</b>	<b>427.50</b>
<b>TOTAL</b>	<b>100</b>	<b>1943.14</b>	<b>2303.39</b>	<b>2154.20</b>	<b>3056.41</b>
<b>INDEX</b>		<b>1.57</b>	<b>1.33</b>	<b>1.42</b>	<b>1.00</b>

Table E-5. Computation of Index of Performance



<b>BENCHMARK: TRANSPORT. NETWORK TESTS EXECUTION TIMES</b>			
<b>FUNCTION</b>	<b>TIME (seconds)</b>		
	<b>HP 720</b>	<b>IBM RS/6000</b>	<b>SUN SPARC2</b>
<b>BUFFER</b>	<b>57.97</b>	<b>60.66</b>	<b>61.63</b>
<b>ALLOCATE</b>	<b>70.00</b>	<b>43.94</b>	<b>55.13</b>
<b>RESTORE</b>	<b>15.06</b>	<b>28.66</b>	<b>13.72</b>
<b>SAVE</b>	<b>13.06</b>	<b>3.75</b>	<b>3.97</b>
<b>CLEAN</b>	<b>69.84</b>	<b>58.54</b>	<b>148.44</b>
<b>READ</b>	<b>2.47</b>	<b>4.65</b>	<b>4.62</b>
<b>ROUTE</b>	<b>8.93</b>	<b>17.10</b>	<b>10.34</b>

Table E-6. Benchmark: Transportation Network Tests - Execution Times

<b>BENCHMARK: TRANSPORT. NETWORK TESTS PERFORMANCE RATIOS</b>			
<b>FUNCTION</b>	<b>SPEED RATIO</b>		
	<b>HP 720</b>	<b>IBM RS/6000</b>	<b>SUN SPARC2</b>
<b>BUFFER</b>	<b>1.06</b>	<b>1.02</b>	<b>1.00</b>
<b>ALLOCATE</b>	<b>1.00</b>	<b>1.59</b>	<b>1.27</b>
<b>RESTORE</b>	<b>1.90</b>	<b>1.00</b>	<b>2.09</b>
<b>SAVE</b>	<b>1.00</b>	<b>3.48</b>	<b>3.29</b>
<b>CLEAN</b>	<b>2.13</b>	<b>2.54</b>	<b>1.00</b>
<b>READ</b>	<b>1.88</b>	<b>1.00</b>	<b>1.01</b>
<b>ROUTE</b>	<b>1.91</b>	<b>1.00</b>	<b>1.65</b>

Table E-7. Benchmark: Transportation Network Tests- Performance Ratios

# BENCHMARK: TRANSPORT. NETWORK TESTS EXECUTION TIMES

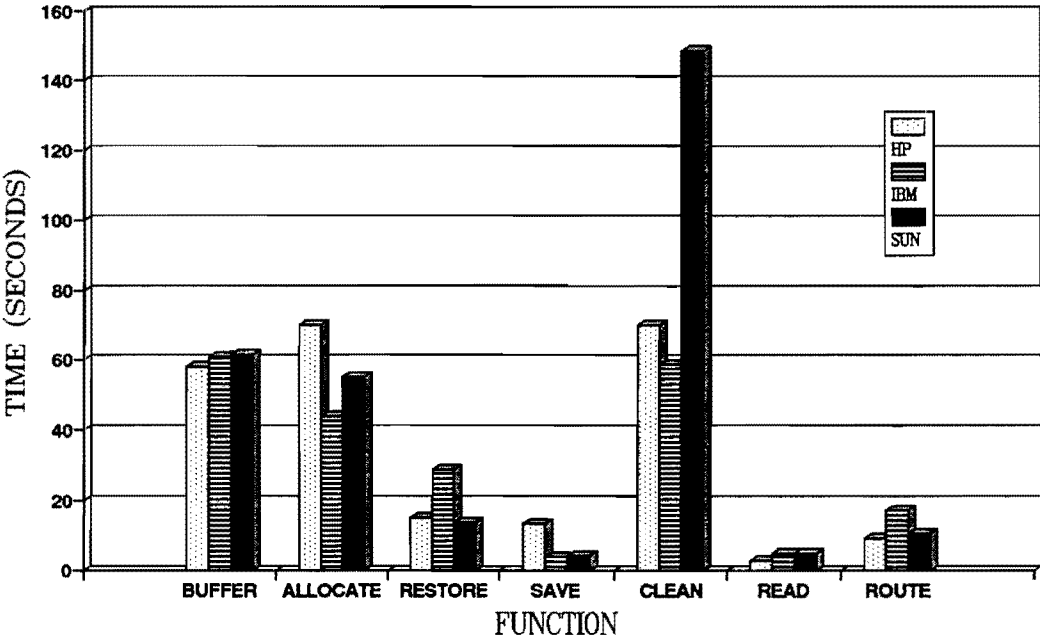


Figure 2. Benchmark: Transportation Network Tests - Execution Times

# BENCHMARK: TRANSPORT. NETWORK TESTS PERFORMANCE RATIOS

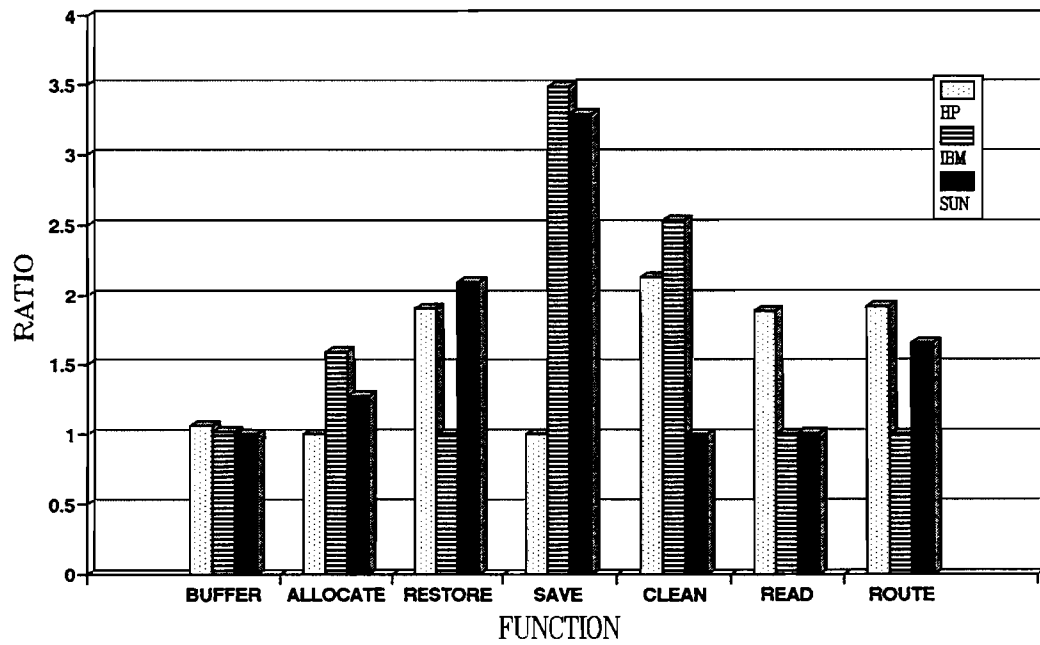


Figure E-3. Benchmark: Transportation Network Tests - Performance Ratios

case. Then the numbers were extrapolated to arrive at a time figure for HP. But this may not be accurate. Nevertheless, HP did turn out to be the slowest in this test. ( HP = 16.4 , IBM = 13.3, and SUN = 15.6, times in seconds). Thus, for the complete test it was estimated (IBM = 43.94, SUN = 55.13 times in seconds) that HP would take about 70 seconds. HP had taken more than 55 seconds when the crash of the system occurred. The cause of the crash could not be determined, however.

Representatives of the vendors, who were present during the benchmarking, stated that several factors could have affected the outcome of the results:

1. IBM had half the main memory of the other two.
2. IBM was not running accelerated graphics.
3. ARC/INFO running on HP had been compiled on another machine (HP 800) and was slower than it should have been.
4. HP had very little available disk space (this was observed by TTI staff).
5. HP had a 3D graphics card instead of a 2D card.

Some observations were unanticipated. In each of these observations, one computer performed significantly slower than the other two. The plausible explanation for each of these, as below, indicates that benchmarking may be affected by other factors.

1. HP took about 3 times as much time as the others to save an edited arc attribute table, perhaps because of an overloaded disk.
2. SUN took about twice as long to clean a coverage even though it performed well in buffering/overlay and routing both of which are also CPU intensive.
3. IBM took about twice as long to refresh a screen image; this may be attributable to the absence of an accelerated graphics card.
4. IBM took about twice as long to route on map even though its performance in some other CPU intensive tests was very good.

## General GIS Tests

Performance ratios for each test are computed in the same way as the overall performance index and reflect the speed of the workstations. The observations have been tabulated and plotted graphically.

The following figures and tables show the observations for these tests:

1. Table E-8 shows the order of performance in each test.
2. Table E-9 and Figure E-4 show the execution times for each test.
3. Table E-10 and Figure E-5 show the performance ratio for each test.
4. Table E-11 shows the computation of performance index for these tests.

ARC PLOT	SUN	>>	IBM
ARCEDIT	IBM	>>	SUN
ARC	IBM	>>	SUN
INFO	IBM	>>	SUN

Table E-8 General GIS Test Order of Performance

---

Again each of the two computers was better in one or more area(s). However, the overall index of performance for these tests, based on the above weights, was 1.14 in favor of the SUN workstations. This is consistent with the observation in Transportation Networks Tests. Though IBM performed better in 3 of the 4 tests, SUN was better by a wide margin in the test with the most weightage.

BENCHMARK: GENERAL GIS TESTS EXECUTION TIMES TIME (SEC)		
TEST TYPE	IBM	SUN SPARC2
ARC PLOT	42	26
ARC EDIT	40	43
ARC	34	57
INFO	16	20

Table E-9. General GIS Tests - Execution Times

BENCHMARK: GENERAL GIS TESTS PERFORMANCE RATIOS		
TEST TYPE	IBM	SUN SPARC2
ARC PLOT	1.00	1.61
ARC EDIT	1.08	1.00
ARC	1.68	1.00
INFO	1.25	1.00

Table E-10. General GIS Tests - Performance Indices

BENCHMARK: GENERAL GIS TESTS WEIGHTED TIMES & PERFORMANCE INDEX		
TEST TYPE	IBM	SUN SPARC2
ARC PLOT	210	130
ARC EDIT	80	86
ARC	34	57
INFO	32	40
TOTAL	356	313
INDEX	1.00	1.14

Table E-11. Computation of Index of Performance

# BENCHMARK: GENERAL GIS TESTS

## EXECUTION TIMES

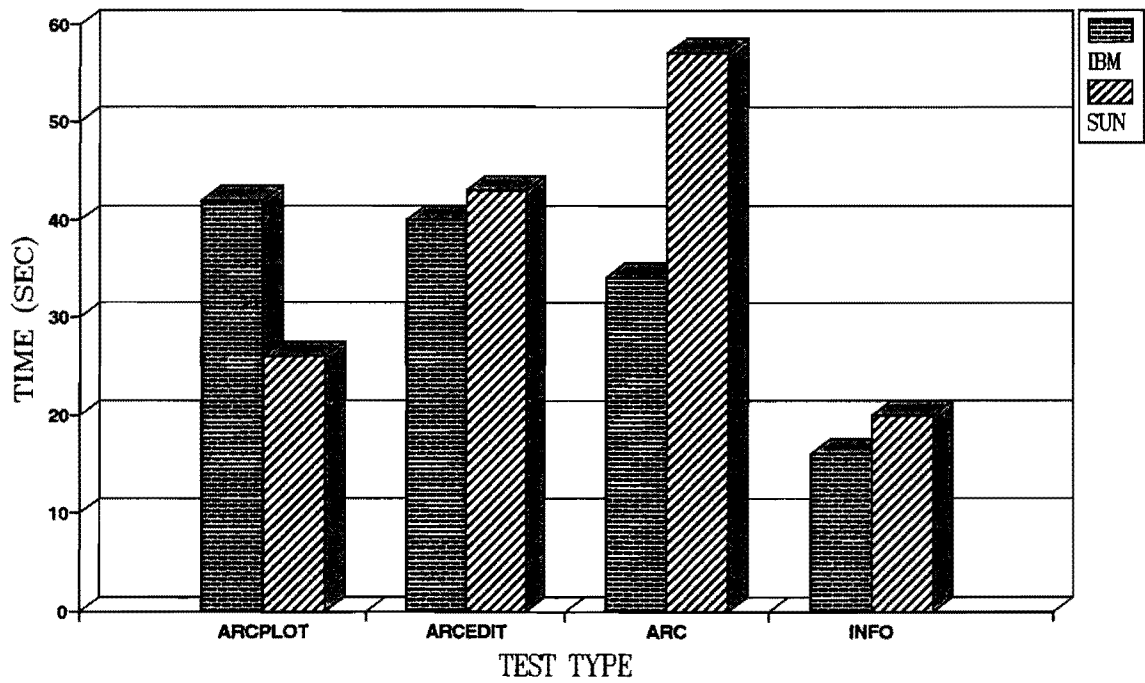


Figure 4. Benchmark: General GIS Tests - Execution Times

# BENCHMARK: GENERAL GIS TESTS PERFORMANCE RATIOS

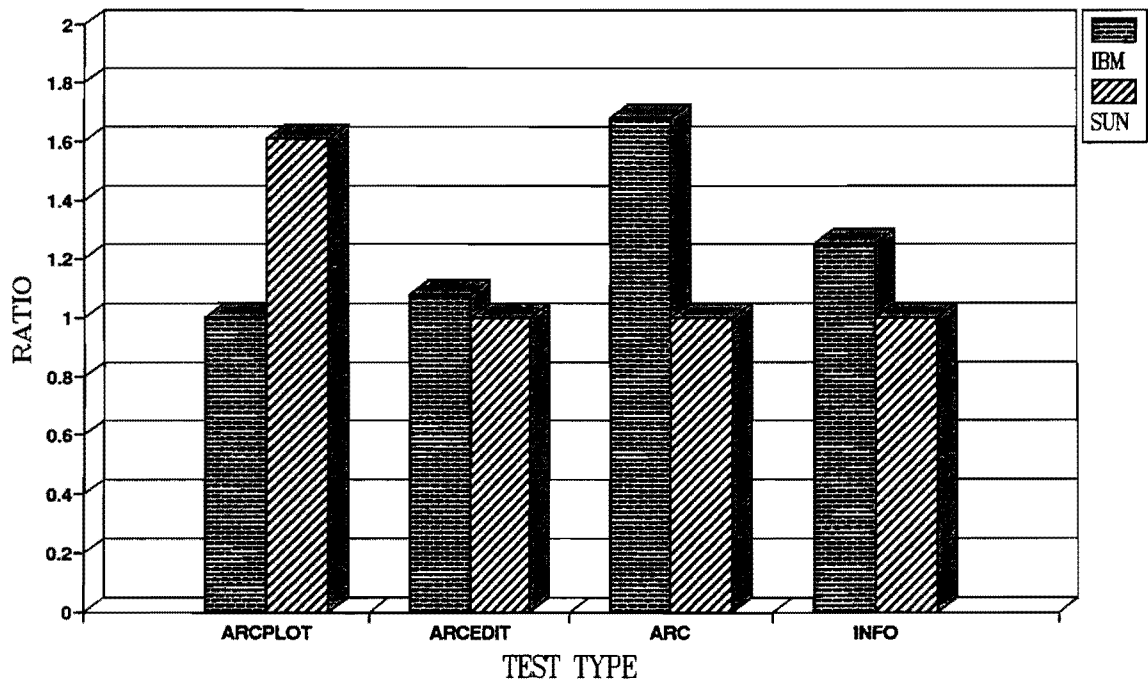


Figure E-5. Benchmark: General GIS Tests - Performance Ratios



## PC vs. Workstation Tests

Performance ratios for each test are computed in the same way as the overall performance index and reflect the speed of the workstations. The observations have been tabulated and plotted graphically.

---

Reading a Network	SUN	>>	HP-PC/386
Restoration of Network	SUN	>>	HP-PC/386
Routing on Network	SUN	>>	HP-PC/386

Table E-12 . PC vs. Workstation Tests Order of Performance

---

The following figures and tables show the observations for these tests:

1. Table E-12 shows the order of performance in each test.
2. Table E-13 and Figure E-6 show the execution times for each test.
3. Table E-14 and Figure E-7 show the performance ratios for each test

The SUN Sparc Workstation outperformed the PC/386 in each of these tests by a factor of 25 to 30. The overall index was 27.28 w.r.t the PC/386. Such a large performance

<b>BENCHMARK: WORKSTATION VS. PC EXECUTION TIMES</b>		
<b>FUNCTION</b>	<b>TIME (seconds)</b>	
	<b>SUN SPARC2</b>	<b>HP/386</b>
<b>RESTORATION</b>	<b>1.97</b>	<b>54.69</b>
<b>ROUTING</b>	<b>1.12</b>	<b>28.34</b>
<b>NETWORK READIN</b>	<b>2.50</b>	<b>72.00</b>

Table E-13. Benchmark: Workstation Vs. PC - Execution Times

<b>BENCHMARK: WORKSTATION VS. PC PERFORMANCE RATIOS</b>		
<b>FUNCTION</b>	<b>SUN SPARC2</b>	<b>HP/386</b>
<b>RESTORATION</b>	<b>27.76</b>	<b>1.00</b>
<b>ROUTING</b>	<b>25.30</b>	<b>1.00</b>
<b>NETWORK READIN</b>	<b>28.80</b>	<b>1.00</b>

Table E-14. Benchmark: Workstation Vs. PC - Performance Ratios

# BENCHMARK: WORKSTATION VS. PC EXECUTION TIMES

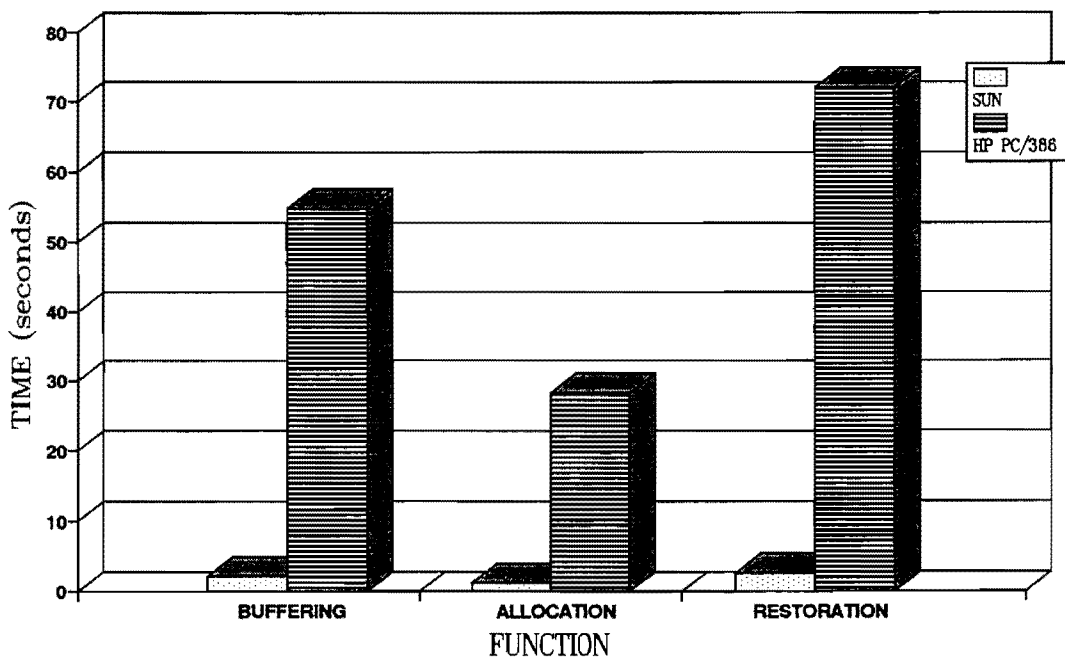


Figure 6. Benchmark: Workstation Vs. PC - Execution Times

# BENCHMARK: WORKSTATION VS. PC PERFORMANCE RATIOS

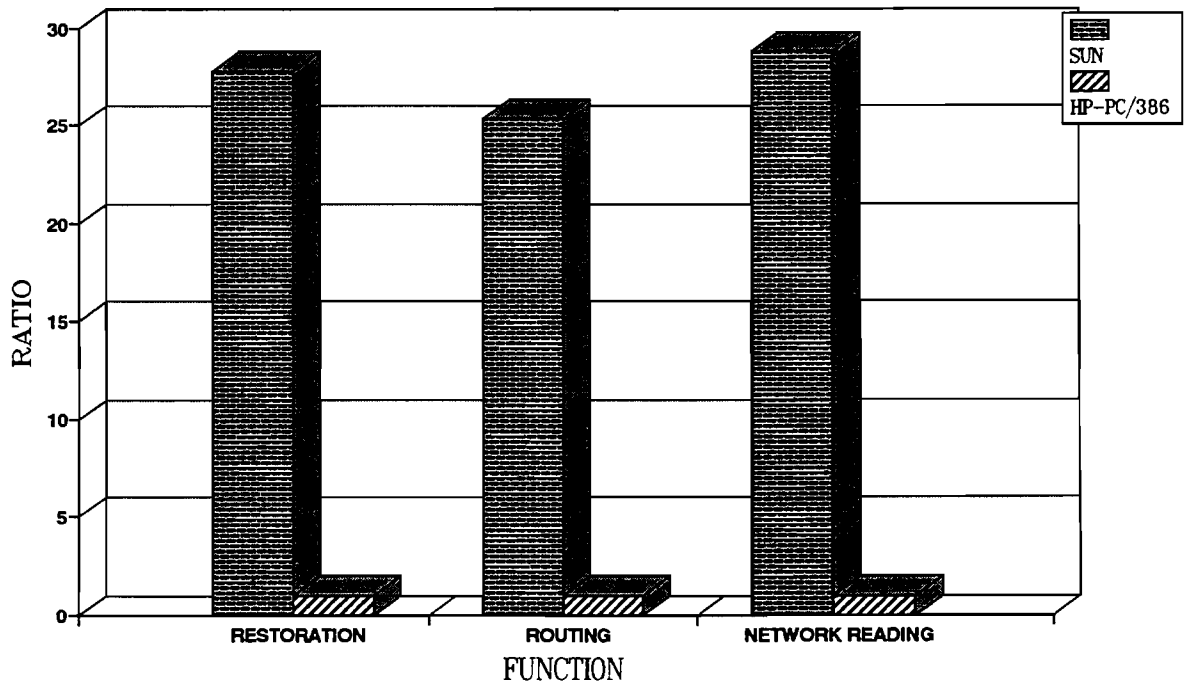


Figure 7. Benchmark: Workstation Vs. PC - Performance Ratios

index indicates significant difference in the capability of workstations and PCs in terms of GIS operations and this issue must be considered at the time of decision on a computing platform for GIS development.

## LIMITATIONS

First, the performance indices could well be different on some other GIS applications and under other test domains. The weighted performance indices that have been computed are based on past use of commands for GIS-T. For some other GIS applications, these weights would be different and these indices would not be valid.

Second, the workstations benchmarked may have performed differently with some other configurations. While these workstations did by and large meet the criteria of specification at a certain price, the vendors could have provided workstations, for the same price, with different configurations.

Third, the benchmarking results could have been different on a GIS development platform other than ARC/INFO.

Finally, the benchmarking was limited to the tests provided by ESRI. Some other tests may have yielded somewhat different results.

Thus, these tests are only indicative of the performance of the workstations for GIS, but are by no means conclusive.

## CONCLUSIONS

Workstations have been benchmarked, from three different vendors, for GIS applications. Emphasis has been laid on performance in the functions required for GIS-T.

In benchmarking each of the three workstations performed better in one or more areas. Overall, HP workstation had the best performance index.

The benchmarking tests had limitations and are only indicative of the performance of workstations. These tests may not be viewed as conclusive.