Automated Collision Diagrams

David M. Litvin, Goodell-Grivas, Inc., Southfield, Michigan
Tapan K. Datta, Wayne State University, Detroit

Collision diagrams, which are graphical representations of accident experience at intersections or along roadway segments, are an integral part of the analysis of hazardous roadway locations. Historically, collision diagrams have been prepared manually by a laborious search of accident files, organization of selected accident reports, and manual preparation of diagrams. Since the development of computerized accident data systems and computer graphics hardware, several software systems have been developed to prepare collision diagrams. Although a number of these efforts have not been reported, the ones that have are generally schematic in nature. Some procedures involve presenting groups of accidents, by type, by means of a single symbol. Presentation of a group of similar accidents at one location sometimes lacks visual impact, and visual impact is one of the most important features of the use of collision diagrams in accident analysis. The development of an automated system to produce collision diagrams that represent each accident by a single symbol is reported. The automated collision diagram system uses existing computer-based accident files directly, is easy and cost effective to use, and produces a high-quality collision diagram.

Accident experience at an intersection or along a roadway segment has traditionally been represented by collision diagrams. Preparation of such diagrams, as part of the analysis of a hazardous location, is a standard procedure for traffic and safety engineers (1). In the past, collision diagrams were prepared manually. This required searching accident files for the appropriate reports; sorting, reviewing, and tabulating the results; and drafting the resulting diagrams. With the advent of computer-based systems of accident inventory, the tasks of searching files and sorting and tabulating the results were taken over by computer software (2). The next logical step is to have the computer prepare the collision diagrams.

Several successful developments in computer-drawn collision diagrams have been reported (2, 3). Some of the systems that have been reported are schematic in nature: One symbol is drawn for each type of accident, and annotations are used to indicate the number of occurrences for each type of accident (2). Although this strategy of representing accident experience at an intersection is effective, it lacks the visual impact that is produced by representing each accident separately. Another advantage of drawing each accident separately is the ability to represent accident severity in the form of a diagram. This is particularly relevant as computerized accident information systems become available at the county and local levels to police traffic safety officers and citizens' traffic advisory committees. These systems and automated diagrams were available before only at the state level. To these local-level groups and even to experienced engineers, the visual impact of one symbol per accident can produce a striking difference in comparison with one symbol that represents several accidents of similar types (see Figure 1).

SOFTWARE FOR AUTOMATED COLLISION DIAGRAMS

Software to produce automated collision diagrams was developed to satisfy a need of the Traffic Improvement Association (TIA) of Oakland County, Michigan. TIA is a nonprofit organization supported by public and private donations that supplies traffic engineering services to more than 30 political jurisdictions in Oakland County, including the County Road Commission. It also maintains a comprehensive accident inventory. Any city, township, or county agency may request inventory reports and summaries from TIA for any intersection or roadway section in the county.

The software for the system is user oriented and requires only minimal familiarity with computer operations. The software, designated the automated collision diagram system (ACDS), is composed of two programs. The first, called SIEVE, performs a search of the accident file to extract the appropriate records. A user must specify a location by intersection or roadway segment identification number and may specify a combination of constraints depending on the type of analysis to be performed. The constraints may include but are not limited to dates (beginning and ending date), time of day (hour range), weather conditions (clear, cloudy, rainy, etc.), pavement condition (dry, wet, icy, etc.), lighting (day, dark, dark with lights, etc.), type of accident (rear-end, left-turn, etc.), location (direction and/or distance from the intersection), and any combination of the above.

The program accesses the accident file and sifts through all the records, retaining those reports that satisfy all of the user-specified criteria. The SIEVE program produces a subfile of accident reports by simply extracting the data from the accident history file but performs no processing on the data.

The second program, PACDS, is modular in nature and contains four sets of routines that perform the functions of (a) checking data consistency, (b) plotting intersections or roadway segments, (c) drawing collision symbols, and (d) producing associated printed reports.

The task of checking accident data for consistency is handled by a module called COPCHK, which detects and eliminates any errors that may result from inconsistency in the methods used by the different personnel who develop the data base. The most common problem here involves placing an accident in terms of direction at intersections whose "legs" do not lie directly north-south-east-west. One person may code an intersection leg as northwest, another simply north. These inconsistencies are flagged. If the user is operating the ACDS in time-share mode, the inconsistent data can be modified at the terminal and processing can continue. If the ACDS is being used in batch mode, an appropriate message is printed and processing continues on the set of data for the next intersection.

The second module, INTPLT, is a set of routines that draws and labels the intersections or roadway segments. If geometric data are available, the user can exercise the option to have the collision diagram indicate the number of lanes on each leg or segment, offset of approaches, and curb cuts. If the geometric data are not available, then a standard roadway segment, a four-leg intersection, or a "T-intersection" is specified by type. The data for each type of standard configuration are contained in a program library file. The T-intersection diagram can be rotated to resemble the real-world situation. The current version of the program stores only standard cross intersections and T-intersections. However, other intersection configurations, such as five legs or one leg skewed, can be coded and entered into the library for reference by type. Each type of intersection requires an average of 15 lines of data, each of which contains as many as 8 entries.

The third module, which contains the most extensive routines, is the ACCPLT module. The ACCPLT routine examines each accident record captured by the SIEVE program. The type of accident and the location and direction of the vehicles involved are determined by the following variables:

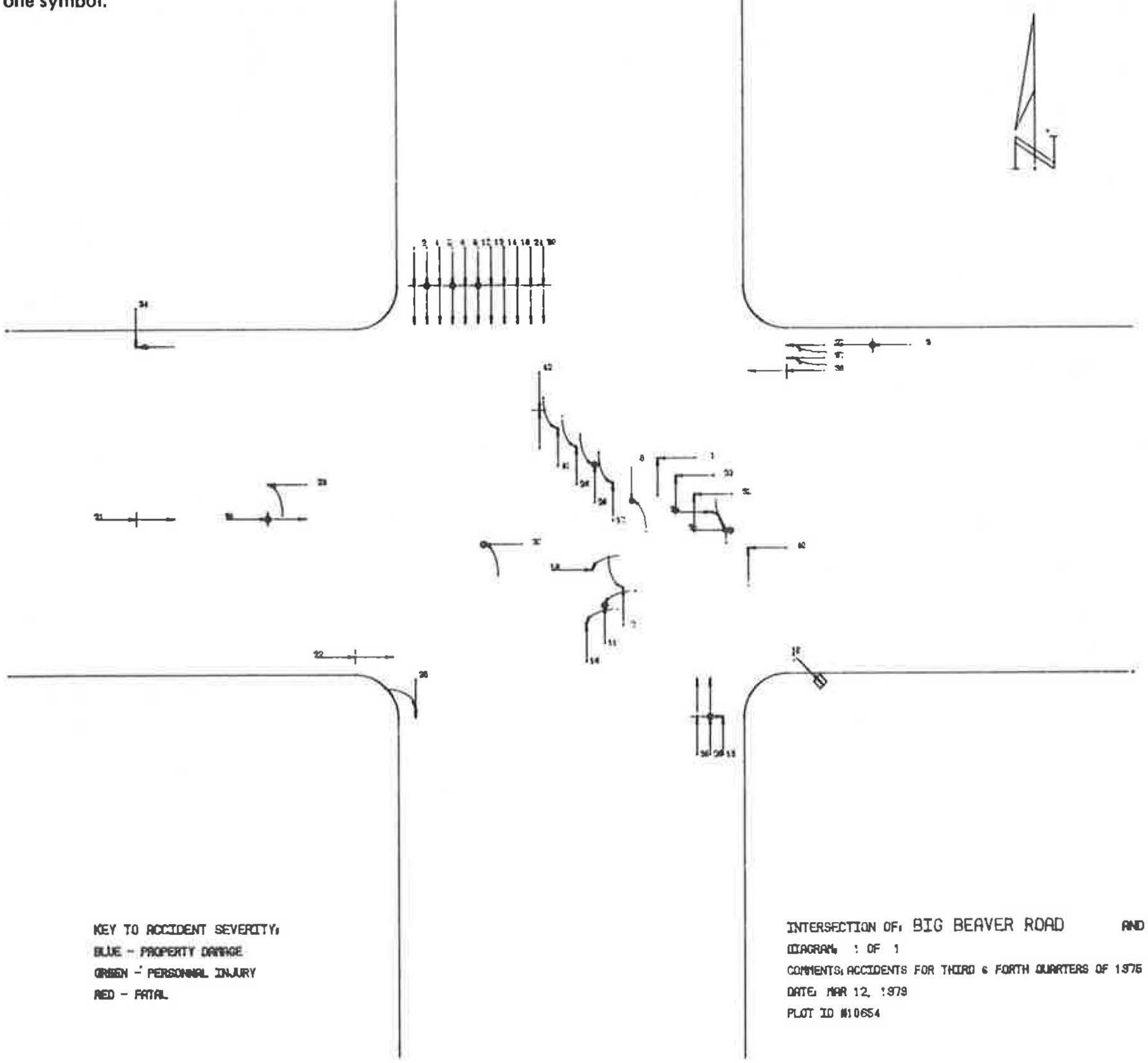
1. Direction of accident from intersection,
2. Distance of accident from intersection,
3. Two-vehicle accident code (for two-vehicle accidents only),
4. Vehicle or pedestrian direction of travel,
5. Driver or pedestrian intent,
6. Contributing circumstances,
7. Location of vehicle impact, and
8. Severity of accident.

Once the type, location, and direction of an accident have been determined, a sequence number is assigned to the accident to associate the plotted symbol with a printed report.

A pitfall of producing collision diagrams with one symbol per accident is the overcrowding that may occur in plotting locations that have a high number of accidents. To accommodate this situation, the ACDS software allows the user two options. The first option allows the user to reduce the number of accidents on each diagram by producing plots for quarterly or monthly time frames. The second option is to summarize each type of accident and to draw and annotate one symbol for each type of accident. The annotation includes the number of accidents of that type by severity.

Traditionally, to indicate the severity of an accident, a circle-type symbol was added to the collision diagram (4). The absence of a circle at the impact point indicated an accident that resulted in property damage only. The presence of a circle indicated an injury accident, and a filled-in (solid) circle indicated a fatal accident. Figure 2 shows examples of the traditional indicators of accident severity. The ACCPLT module indicates accident severity by drawing collision symbols in blue, green, or red. Blue indicates property-damage-only accidents, green indicates personal-injury accidents, and red indicates fatal accidents. The color coding of accident experience at a location is a valuable guide, particularly when fatal accidents are involved (the color coding is variable, and the selection of colors to designate the severity ratings described above represents the preference of TIA). The circle-type designation of accident severity has been maintained to allow easy copying of the collision diagram. As the costs of plotting diminish,

Figure 1. Automated collision diagram in which each accident is represented by one symbol.

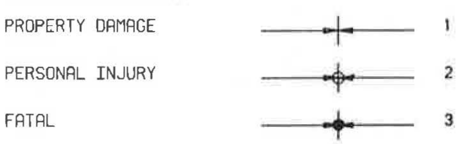


however, the cost of producing two, three, or any number of additional copies is continually being reduced. Since copying by other means—such as blue-line or photostatic techniques—may no longer be necessary, the need for the circle-type symbol will be eliminated.

The software to produce the collision symbols resides in a submodule of ACCPLT designated COLSYM. COLSYM contains a set of 15 subroutines that produce all necessary collision symbols. The coding of the COLSYM routines is optimized by following structured programming techniques to eliminate redundant coding. For example, many collision symbols require use of the symbol for a moving vehicle (a solid line with an arrowhead). All subroutines that require a symbol for a moving vehicle reference the same single moving vehicle (MVVEH) subroutine. The coordinates and orientation of the moving-vehicle symbol are determined by the calling subroutines. This minimization of code is used for such symbols as those for head-on and right-angle accidents, turning accidents in which the intent of one vehicle is to go straight, and run-off-the-road accidents. The routines in the COLSYM submodule require plotting subroutines that would be supported in any standard scientific plotting package. These include routines for arrowheads, lines, circles, and ellipses.

The fourth module, which is responsible for printing accident information associated with the collision diagram, is called ACPRNT. This module prints the necessary accident information for each accident, such as location, date and day of the week, time, environmental conditions (weather, light, and pavement conditions), vehicle accident severity, vehicle violation, and the accident sequence number as assigned by the ACCPLT module. Figure 3 shows a sample printout from the ACPRNT module (since the program is calibrated in U.S. customary units of measurement, no SI equivalents are given).

Figure 2. Traditional indicators of accident severity for head-on accident.



TRANSFERABILITY

The ACDS software package is written in G-level FORTRAN IV. The computer core requirements for the entire ACDS package, including associated plotting and FORTRAN on-line functions, are 110-k bytes. The program was developed on an IBM 360/67 and is currently operational on an AMDAHL 470V6 system at Wayne State University (WSU).

Because the core requirements are relatively limited and an overlay structure is feasible because of the modularity of the software, ACDS can be implemented, with few modifications, on any system that supports G-level FORTRAN IV and a plotter, has scientific function software (ellipse, circle, arrows), and has a moderate amount of core or virtual memory. The software, as described earlier, can operate in either batch or timeshare (terminal) environments.

COST

The cost of producing collision diagrams by using ACDS is significantly lower than it is for other methods. Interestingly, the most expensive aspect of using the ACDS software is in executing the SIEVE module. Data for an entire year or several years may need to be searched to capture the appropriate accident records. For example, if the SIEVE program is executed during prime hours (9:00 a.m. to 6:00 p.m. weekdays) on the WSU system, the cost of extracting an intersection's data from the file of total 1976 accidents in Oakland County (more than 37 000 accident records) would be approximately $9.00, including all tape drive, tape mount, and central pro-

Figure 3. ACDS printed report.

INTERSECTION: BIG BEAVER ROAD AND ROCHESTER ROAD

COMMENT: 1976 ACCIDENT EXPERIENCE

ACC. #	DATE	DOW	TOD	DIST.	WTHR.	PAVE.	VISI.	SEVRT	VEH. #	VEH. SEVRT	VEH. VIOL
.											.
11	7 21	M	0600	38' S	CLR	DRY	DAY	P-1	1	PDO	NA
									2	PI	TOO CLOSE
12	7 30	W	1730	38' S	CLR	DRY	DAY	PDO	1	PDO	LN USAGE
										PDO	NA
13	8 04	M	1215	19' N	CLR	DRY	DAY	PDO	1	PDO	LN USAGE
									2	PDO	NA
14	8 08	F	0300	0'	CLR	DRY	DRK	FTL	1	PDO	FLD ROW
									2	FTL	NA
15	8 11	M	1300	57' E	CLR	DRY	DAY	PDO	1	PDO	TOO FAST
.											.

cessing unit (cpu) costs. The number of records extracted does not significantly affect the cost. The cost per search can be reduced by requesting multiple searches during each run or by running batch tasks to extract data during nonprime hours. Late-hour execution of a batch task reduces the costs of a SIEVE run to 30 percent of prime-time costs.

Once the data have been extracted, the cost of producing a collision diagram is quite low. For a collision diagram for an intersection with 78 accidents (one year), executed during prime time, generating the plotter file (9.7 s in cpu time) costs $0.10, plotter charges are $3.50, and printer charges are $0.30. This results in a computer cost of $0.17/accident if the SIEVE module is executed during prime time and $0.09/accident if it is executed during a time when the lowest rates are in effect (these costs are for the IBM 360/67 system; costs for operation on the AMDAHL system have not yet been determined but should be lower than those on the IBM system).

Hourly personnel costs are also quite minimal. Batch tasks to execute the SIEVE module require 5 min or less to submit for processing. On-line terminal sessions are generally used to produce the actual collision diagrams. This is done to correct errors in consistency as they are determined by the software. Ten min of terminal time is the average for sets of accident data in which there are no inconsistencies to correct, whereas more than 20 to 30 min may be required for intersections that have a high number of accidents (more than 100) and a high number of errors in consistency. These time estimates include the time it takes the user to determine what corrections are to be made. If the generation of plots is to be performed by batch mode, 5 min/diagram is again an average time.

ALTERNATIVE INPUTS

As stated earlier, the ACDS software uses the TIA accident records as input without processing or modifying any information from the data base. Other data base formats can easily be accommodated by simply altering the input formats for the SIEVE and ACCPLT modules. The Michigan Department of State Highways and Transportation is currently completing a statewide accident location system, the Michigan accident location index (MALI). The purpose of the system is to maintain an accident data base for every reported accident in the state and locate each incident to the closest intersection. The ACDS can be modified to read the MALI data base directly and produce collision diagrams for any intersection in the state.

CONCLUSIONS

The ACDS software package provides a realistic representation of accident experience at a specific location—either an intersection or a roadway segment. The collision diagram produced by ACDS, by virtue of its representation of each accident by a symbol and its color coding of accident severity, produces better results and provides more insight into accident experience at a location than do those systems that produce only one symbol for each type of accident. The ACDS is easy to execute and, in view of the quality of the output produced, is certainly cost effective.

REFERENCES

1. P. C. Box. Accident Pattern Evaluation and Countermeasures. Traffic Engineering, Aug. 1976.
2. L. M. Hagmann. A Quick and Simple Way to Prepare Collision Diagrams. Traffic Engineering, Dec. 1974.
3. Traffic and Transportation Engineering Handbook. ITE, Arlington, VA, 1976.
4. L. J. Pignataro. Traffic Engineering Theory and Practice. Prentice-Hall, Englewood Cliffs, NJ, 1973.

Discussion

John J. Zogby, Bureau of Accident Analysis, Pennsylvania Department of Transportation

Litvin and Datta deserve recognition for developing a data-processing technique that is powerful, relatively easy to use, and within the budget of most agencies involved in accident analysis. I look at this tool as a further extension of the potential of a traffic records system. Many claim to have been using automated collision diagrams for some time, but there is no readily available documentation of such use. So, for those of us who are involved in developing a comprehensive traffic records system, this information is most helpful.

Some attractive features of this system that immediately came to my attention were the following:

1. The capability to choose a variety of constraints, such as a range of periods, times of day, weather conditions, pavement surface conditions, and types of accident, in addition to location selection;
2. The edit routine, which is especially attractive to those of us who have been criticized for having an eastbound vehicle on a north-south roadway;
3. The ability to generate a report of additional details for accidents that are represented in the collision diagram.

Those who use collision diagrams and have an automated accident file would do well to examine this package.

Before I get into more specific comments about the ACDS software package, a few philosophic observations are in order. In my own experience, in discussion of the uses of accident collision diagrams as part of engineering studies of accident sites, there are no passionate proponents or opponents. Even among agencies that have access to automated accident record systems, there is no universal acceptance of the need for automated collision diagrams. Some agencies seem to be satisfied with sifting through reams of paper and painstakingly drawing each collision diagram with annotations of pertinent accident characteristics. They reject automated collision diagrams as inaccurate because of the slightest inconsistency or discrepancy. Other organizations combine collision activity in a condition diagram (in some cases, on aerial photographs), thereby precluding any automated report generation unless the condition diagram can be automatically produced. And still others, who have the ability to automatically manipulate, aggregate, and mathematically analyze accident data, find the preparation of accident collision diagrams an unnecessary step in the study process. These observations are not presented to argue the legitimacy of the rationale of these agencies but rather to acknowledge that there are agencies and individuals that do not see the need for automated collision diagrams.

For those who do see the need for such diagrams, I will comment on the ACDS software package more specifically. There are several areas in which I anticipate problems. First, a suggestion: In the associated accident information report, since the accident sequence numbers are programmatically assigned, the accidents listed in the printed report could be arranged in an order compatible with their presentation in the diagram. For example, accidents of the same type within the same quadrant could be presented together. This would avoid paging back and forth through the printed listing and aid in the search of additional data for analysis.

In the problem areas, I must admit that, although I can suggest a solution in one instance, I do not have the answers. I must also admit to making some broad assumptions as to the limitations of the system.

The first area of concern is the presentation of each accident by symbol in the collision diagram. Unless there is a programmatical adjustment for the size of the diagram based on the frequency of accidents by quadrant or in total, at some point there will be a space limitation on the presentation of each accident. In Figure 1, for example, I see a problem in representing 50 rear-end collisions at any leg of this intersection. In addition, I found it difficult to read the accident sequence numbers associated with each symbol shown. There seems to be a diminishing return on the "visual impact" benefit when the frequency of accidents at a given location is high.

Another area of concern is how to handle a multileg intersection that has two or more approaches in the same quadrant. The presentation of data under this condition would be distorted. One possible solution may be to identify each variation of the multileg intersection in a four-leg format. Although this would solve the distortion in the collision activity of vehicles on different legs in the same quadrant, it means that all accidents at that particular location would not be presented in one diagram.

My final concern is the transferability of the ACDS package to other accident data bases. I will attempt to illustrate this by using the accident record system we have in Pennsylvania. In their paper, the authors state, "Other data base formats can easily be accommodated by simply altering the input formats for the SIEVE and ACCPLT modules." However, in the description of the SIEVE program, they mention that, "A user must specify a location by intersection or roadway segment identification number. . . ." Pennsylvania's accident record system does not identify intersections or road segments by a reference number. The identification of intersections in the Pennsylvania system is by intersecting route numbers or, in the case of local roads, by street names. Our system also has the capability of recording up to five differently identified legs for any one intersection. Road segments are identified on the state road system by route number and station and on the local road system by street name and the cross streets on either side.

The ACCPLT module may also present a "transfer" problem. This module requires variables that are not in the Pennsylvania accident file, such as, for intersection accidents, distance of accident from intersection. I am not sure what problems these examples would present. It may be that extensive modification to the SIEVE and ACCPLT modules would be required.

I hope that my criticism does not offer justification to those who are complacent about the manual preparation of collision diagrams. I say this because the ACDS software package can be a most advantageous tool for agencies that have an automated accident record system and use collision diagrams, especially in today's atmosphere of limited personnel resources in the highway and safety fields.

Authors' Closure

We would like to express our appreciation to Zogby for his discussion. His comments on and critique of ACDS are constructive and supportive of our efforts. He expresses three main concerns and offers one suggestion on the capabilities of the ACDS software. We would like to address these items briefly.

Zogby suggests that the printed accident report be organized by quadrant to avoid paging through the output to locate accidents in similar quadrants. The accidents are currently plotted and printed in chronological order. This is the order in which the PACDS program receives the data from the SIEVE program. It is possible to modify the SIEVE program or to use a utility sort program to rearrange (sort) the accident data by direction from intersection and direction of travel and then in chronological order. This procedure would be performed before the PACDS program is executed and would result in the printed output being organized by quadrant.

The first of Zogby's concerns is the overcrowding of symbols at locations where there are many accidents. This can be handled by ACDS by segmenting the data into quarterly or monthly diagrams if the number of accidents at a site becomes so large that it poses an overcrowding problem. In fact, presentation of total yearly accidents at a site in one diagram is not a mandatory practice for traffic engineers. As an alternative, a user could also revert to the method of displaying one symbol for each type of accident and a number adjacent to the symbol to indicate the number of occurrences.

Zogby's second comment concerns handling a multileg intersection. The ACDS programs reference a library of intersection types. The data in the library direct the INTPLT module to draw a specific configuration of an intersection. The library data also contain information for the COPCHK module for use in editing accident record data. Only five types of intersections are currently available. These are a standard cross intersection and a T-intersection rotated in four directions. Data that describe other configurations can be added to the library at any time and require the coding of an average of 15 lines of data, each of which has as many as eight data fields.

Zogby's last comment concerns the transferability of the ACDS software. He is concerned that the SIEVE program captures accident reports based on an intersection or roadway segment identification number. Actually, the SIEVE program examines a record's identification field. This field may contain alpha characters and can be modified to accept any reasonable number of characters. As it happens, the TIA and MALI data that we used in developing the system do use identification numbers. Zogby is further concerned that all of the required data such as distance and direction from intersection, for accidents recorded as "at intersection", may not be available on some already implemented systems such as Pennsylvania's. This may present a problem in transferability. However, a set of assumptions and other criteria may be used to locate the accident. Modification of the accident location procedure in a program like ACDS does not represent a significant reprogramming effort.

Finally, Zogby mentions that some of the numbers in Figure 1 that associate accident symbols with printed information were difficult to read. We apologize that we could not include a full-size diagram [approximately 35.6 × 35.6 cm (14 × 14 in)] in the manuscript Zogby reviewed. The diagram as shown in Figure 1 is not full size because of space limitations.

Publication of this paper sponsored by Committee on Traffic Records.