TECHNICAL REPORT STANDARD PAGE

Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA/LA-90/232		

Title and Subtitle	
TRAFFIC ACCIDENT SIMULATION USING INTERACTIVE COMPUTER GRAPHICS	5. Report Date December 1991 6. Performing Organization Code LTU-COE-SAIL-90081
Author(s) Dale O. Anderson, Ph.D., Leslie K. Guice, Ph.D., Patrick L. Rocke, M.S.	8. Performing Organization Report No. 232
Performing Organization Name and Address Louisiana Tech University College of Engineering P. O. Box 10348 Ruston, LA 71272-0046	10. Work Unit No. 11. Contract or Grant No. LTRC Proj. #88-1SS / State Proj. #736-12-71
Sponsoring Agency Name and Address Louisiana Department of Transportation & Development P. O. Box 94245 Baton Rouge, Louisiana 70804	13. Type of Report and Period Covered Final Report 1988 - 1990 14. Sponsoring Agency Code

Supplementary Notes
Conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration.

Abstract

The primary goal of this project was to assemble the hardware and software necessary to form a prototype sktop video animation workstation to support traffic accident reconstruction. This goal was accomplished. ie workstation is an IBM-PC/AT compatible (ISA) microcomputer based on the Intel 80386 Microprocessor and)287 floating point co-processor. A truevision Targa-16 graphics adapter and VDI/O RGB to NTSC converter as installed in the computer. A BCD-4000 animation controller was also installed and was connected to a Sony -5850 3/4" U-Matic editing recorder. TOPAS Animator from AT & T Graphics Software Labs was used as the imation software.

Two FORTRAN 77 interface subroutines were developed to interface to TOPAS. AMFTOPAS.FOR is an ASCII odel file interface code for building TOPAS models with custom codes. ANITOPAS.FOR is an animation script interface code for creating TOPAS animation scripts with custom codes. These subroutines have been ccessfully used to create vehicle and terrain model file generators, as well as vehicle trajectory animation ripts.

Key Words	18. Distribution Statement
traffic, accident, reconstruction, computer, video, animation, models	Unrestricted. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	91	

Traffic Accident Simulation Using Interactive Computer Graphics

Final Report

By

Dale O. Anderson, Ph.D.
Assistant Professor of Mechanical Engineering
Leslie K. Guice, Ph.D.
Professor and Head of Civil Engineering
Patrick L. Rocke, M.S.
Graduate Research Assistant
Louisiana Tech University
Ruston, LA 71272

Conducted for

LOUISIANA DEPARTMENT OF TRANSPORTATION AND DEVELOPMENT
LOUISIANA TRANSPORTATION RESEARCH CENTER
in Cooperation with
U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL HIGHWAY ADMINISTRATION

August, 1990

ACKNOWLEDGMENTS

Principal funding for this research project was provided by the Federal Highway Administration (FHWA), U. S. Department of Transportation, and by the state of Louisiana through the Louisiana Transportation Research Center (LTRC), Department of Transportation and Development (LADOTD).

The authors gratefully acknowledge the time, effort and assistance provided by William W. Temple, P.E., Steven L. Cumbaa, P.E., William King, P.E., and Deborah Boleware of LTRC. The helpful advice and direction provided by Sharon F. Lyles of LADOTD is also gratefully acknowledged.

The authors would also like to thank Paul A. Pisano of FHWA for helping to clarify the goals of the research project and pointing out potential implementation problems.

Drs. Anderson and Guice also wish to thank the graduate and undergraduate students who assisted with this project: Carey F. Cox, John S. D. Nguyen, Patrick L. Rocke, Kun Kenneth Liang, David G. Hart, Ross E. Bozeman, Mike S. Briggs and Mohsen M. Farahani.

ABSTRACT

This report is a summary of a three-year research effort to develop an integrated computer/video recorder system for conducting traffic accident reconstructions and generating real-time, three-dimensional, color animations of accident sequences on videotape. Major emphasis was placed on developing a system that could be run on a relatively low-cost workstation platform. Existing accident reconstruction codes and methods would be used to reconstruct accident sequences. A commercial desk-top broadcast video animation package was used to depict the reconstruction results graphically. Several custom codes were developed to interface accident reconstruction codes and various model-generating codes with the animation package.

The TOPAS (Three-dimensional Object Processing and Animation Software) code from AT&T Graphics Software Labs was selected for this research effort. TOPAS is a user-friendly, menu-driven, three-dimensional modeling and animation package that provides a wide range of building, editing and image enhancement tools. TOPAS runs on an IBM-PC/AT compatible computer using either a Truevision TARGA or ATVISTA frame buffer. Professional video editing hardware was also required to record animation onto videotape.

It was established that a relatively low-cost accident reconstruction system that produces high-quality video images is viable and can be a

valuable tool in roadway design or accident litigation. However, a specialist who is knowledgeable in accident reconstruction principles as well as computer graphics and production video is a necessity for effective utilization of the system.

Most serious accidents in Louisiana are single-vehicle accidents. The HVOSM-RD2 code was acquired to simulate 3-D single-vehicle trajectories. However, it was concluded that there was insufficient experimental validation using late-model vehicles to defend a trajectory simulator successfully in court. Other reconstruction tools were also evaluated.

Additional research is needed to develop improved model-generating tools and to provide experimental validation for vehicle trajectory simulation.

IMPLEMENTATION STATEMENT

The desktop engineering video animation workstation developed in this research effort has great long-term potential to assist the state of Louisiana or any other state wishing to utilize this technology in the evaluation of and defense against litigation arising from traffic accidents. The value of this technology lies in the ability to present traffic accident reconstructions as near photo-realistic animations depicting the reconstructed motions of the vehicle(s) involved in an accident. Once state personnel are trained in the use of the workstation and procedures are established for accident data collection, this technology could be routinely used to assist expert witnesses hired by the state to visualize their reconstruction results for presentation at planning sessions, settlement meetings or in the courtroom.

Desktop engineering video animation is a natural addition to the engineering design and analysis functions already performed by LADOTD for the state's roads. Video animation could be used to create "walk-around", "drive-through" or "fly-over" animations of proposed new transportation facilities or modifications to existing facilities for use in the design and evaluation process, public relations or for presentation to the state Legislature.

Desktop engineering video animation is a visualization technique that is rapidly maturing and has a wide variety of potential applications in many engineering design and analysis activities, not just in traffic accident reconstruction. Video animation is a logical choice for presenting the results of almost any dynamic simulation or analysis of a physical system. Even the casual observer can readily see every visible nuance in the structure and motion of a video animation.

A word of caution needs to be presented at this point. Engineering video animation is a very sophisticated visualization process involving very specialized skills and specialized hardware and software. It is not a turn-key black box that can be used successfully with little or no skill or experience. An individual involved in the creation of engineering animation requires the analytical skills necessary to develop the motion time histories of the system being modeled, the ability and artistic sense to create a dimensionally correct model of the system with the proper motion characteristics, a working knowledge of the animation software, and an understanding of production video equipment and processes used to record the animation. The development of such skills in an individual requires that an appropriate training program be made available.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

The purpose of this research effort was to develop desktop video animation techniques for application to traffic accident reconstruction. This required the merging of professional video editing technology with IBM-PC/AT-based computer graphics. It also required a survey of traffic accident reconstruction methods and an analysis of their utility for producing video animation.

DESKTOP ENGINEERING VIDEO ANIMATION

The desktop or personal computing environment, consisting of personal computers and engineering or graphics workstations, is the ideal medium for simulating the dynamics of systems of rigid bodies and creating engineering video animation from the results (1). The current computing power and versatility of the personal workstation rival, or even exceed, that of the traditional mainframe computer.

Computer-generated animation has been used for several years to visualize the results of complex dynamic simulations in research labs, but it has now matured sufficiently to find an increasing number of applications in more traditional engineering design and analysis endeavors. Until recently, most computer-generated animation has been done on supercomputers or "super graphics" workstations, but advances in personal workstation hardware and software have enabled them to offer similar 3-D graphics capabilities at a small fraction of the cost.

Desktop engineering video animation is the production of an animated sequence of video images using a desktop (personal) workstation. The animation is typically recorded onto videodisc or tape. This type of animation is not created in real time. Instead, it is created one frame at a time as fast as the computer is able to do it, and then it is played back in real time (2). This distinction is important because the animation can be created on relatively low cost microcomputers or engineering workstations instead of expensive supercomputers.

The fact that the generation of a video animation sequence does not have to happen in real time allows the engineer to work with very large and complex models and to use sophisticated rendering techniques to produce each frame image of the finished animation sequence. High rendering speed, while desirable, is not necessary for the production of video animation. Workstation cost is a direct function of rendering speed. Near real time rendering speeds (10 frames per second) are currently achievable only on very expensive super graphics workstations and even then only for relatively simple models.

ANIMATION HARDWARE

A basic desktop engineering video animation workstation (Fig. 1) consists of an animation computer connected to production video editing equipment. A color frame buffer, with 8 to 24 color "bit planes," must be installed in the computer to provide the image. The color image produced by a personal workstation is generally displayed on an RGB (Red, Green, Blue) color monitor. A video signal converter must be attached to the frame buffer to convert the RGB video image to an EIA (Electronics Industries Association) RS-170A/NTSC (National Television Standards Committee) video image suitable for recording. A videodisc recorder or a professional editing videotape recorder (S-VHS, U-Matic, SP U-Matic, MII, etc.) and controller must also be added to record the images onto videotape. In addition, a color printer can be included to provide hardcopy of system images.

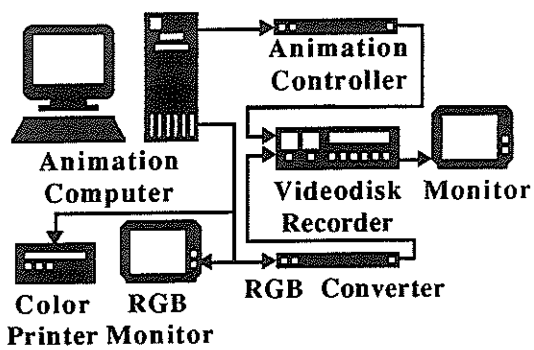

Figure 1. Basic engineering video animation workstation.

A more advanced engineering video animation workstation (Fig. 2) includes both a video disk recorder and a professional editing tape recorder, as well as a color film recorder. Animation sequences are recorded directly onto videodisc and then later edited onto videotape for presentation or distribution. The color film recorder can produce high-quality color slides or prints of individual images.

Video images are only medium resolution by computer graphics standards. An NTSC video image contains a maximum of approximately 492 horizontal lines of image information (out of 525 total). A video resolution (492 lines or less) color frame buffer is adequate for video animation provided that it has sufficient color bit planes (at least 8 and preferably 24) to provide smooth color variations across curved surfaces. Higher spatial resolution frame buffers, usually found on engineering or graphics workstations, may be used, but their images would have to be averaged down to fit within the spatial resolution of the NTSC video standard. While this averaging helps to control aliasing (jagged lines and boundaries), higher resolution images take much longer to compute.

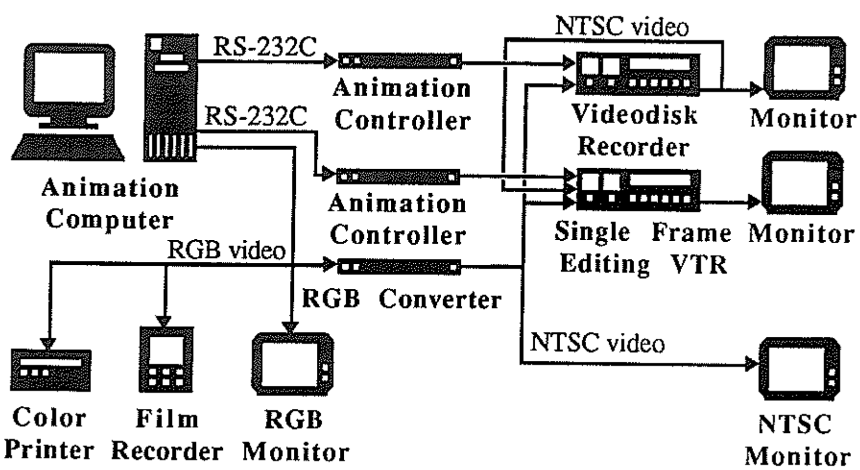

Figure 2. Advanced engineering video animation workstation.

Desktop engineering video animation must be recorded one frame at a time, since it is not created in real time. This requires a professional quality videodisc recorder or editing tape recorder. It cannot be done on consumer video recorders. In addition to the recorder, a controller is required which will interface to the computer. The controller allows the computer to operate the recorder automatically so that a video animation sequence can be rendered and recorded without human intervention once the process has been started.

The videodisc is the ideal medium for recording engineering video animation for several reasons: each frame has a unique address on the disc and can be accessed individually, the RGB image can be recorded directly, the video image is recorded in a digital format that does not degrade with usage or time, and the video disk itself is very small and easy to handle. Few of the commercial animation packages will currently drive a videodisc recorder, but support is expected to be forthcoming.

The ability to record specific individual video frames onto tape is made possible by encoding the videotape with a time or frame code that is synchronized with each video frame. By reading the time or frame code in real time, the controller can accurately locate any individual frame in the entire tape. SMPTE (Society of Motion Picture and Television Engineers) time code is recorded on an audio track and is probably the most common and least expensive time coding method. VIFC (Vertical Interval Frame Code) frame code is recorded directly in the vertical interval (invisible) portion of the video signal of each frame and does not require a separate audio track.

ANIMATION SOFTWARE

A video animation software package consists of two basic parts: a surface or solids model renderer and an animation sequencer. The renderer produces the finished image of the solids model for each frame to be recorded, while the sequencer modifies the model from frame to frame and records the finished images onto videotape in the proper order. In addition to these two basic functions, most commercial animation packages also include a model editor, an animation script editor and a real-time, wire-frame animation preview function.

Almost all the video animation software packages currently available for personal workstations are primarily intended for the video graphics artist, and the model creation and editing facilities usually do not have the tools to produce dimensionally correct models easily. It is therefore essential that the package have a published model file format, preferably ASCII and not binary, so that models can be created externally by a surface or solids modeling package or by a custom code and imported into the animation package using model file translators. Model files for most current commercial animation packages consist of simple polygon surface meshes for each object. IGES (Initial Graphics Exchange Standard) and DXF (AutoCAD) model file translators are commercially available for some of the more popular animation packages. The authors have found that specific classes of models, such as automobiles, are more easily generated by a custom code. If the model file format is known, then the writing of custom model generation codes is fairly straightforward.

An additional requirement of the model file format is that linkages between objects must be included as part of the file. Many models are created as an assemblage of individual component objects, and the connectivity relationships between objects must be preserved (3). This type of model structure is called hierarchical and consists of sibling, parent and child linkages. Sibling objects move independently of each other. A parent object imposes its own motion on its children. A child object moves relative to its parent.

Keyframe animation is probably the most common method of scripting rigid body motion. A Keyframe is simply a selected video frame in which the position and orientation of an object are specified. Each moving object in an image must have its own set of keyframes. The animation sequencer will smoothly interpolate between the keyframes to produce the position and orientation of the object at each intermediate video frame. Simple motions can be scripted with few keyframes. For some complex motions however, each video frame must be a keyframe to represent the motion accurately (4). The apparent velocity of object motion is controlled by the video frame rate (30 frames per second) and the spacing between keyframes.

The video animation software package should have a published animation script file format, again preferably ASCII and not binary, since an intermediate code must be written to create the animation script from dynamic position and orientation time histories produced by a simulator. For some simple cases, it is easier to animate objects using the animation script editor supplied with the package, but for long term use, a custom script generator code is the easiest and most efficient solution. Such a code must take the simulation time histories and match a position and orientation to each moving object at specific times. The animation script generator is probably the most complicated interface code to develop, since it must have information about the solids model as well as the dynamic time history of each moving object.

TRAFFIC ACCIDENT RECONSTRUCTION

Traffic accident reconstruction is the science of using the laws of physics and empirical data to determine a sequence of events in a traffic accident that best fits the evidence left at the scene and on the vehicle(s), as well as the statements of the witnesses. Each traffic accident is different. Some accidents will leave a substantial amount of physical evidence (debris, skid marks, gouges in the road, damage to roadside objects, etc.), while others will leave practically none. Some accidents may have several witnesses and in others not even the vehicle driver can recount what happened. As a consequence, there is no set formula for traffic accident reconstruction, only a set of tools that may be used. The knowledge and judgment of the reconstructionist are prime ingredients in any traffic accident reconstruction.

All witness statements must be viewed with some skepticism. Human visual perception of distance and speed is highly unreliable, since the human brain is not structured to process visual information in a quantitative fashion. However, human visual perception of vehicle position and orientation with respect to fixed landmarks at the scene can be quite accurate if the witness is not moving significantly with respect to the landmarks. Occupant injuries and extensive vehicle damage tend to "color" the impression of a witness concerning accident severity (i.e., impact speed).

Speed estimates offered by police officers are seldom really based on "experience" (since police officers seldom see an accident happen, and when they do there is no way to measure the impact speed). A police officer's speed estimate is usually based on a guess or on a simplified calculation, which may or may not apply to the particular accident. Police officers are trained primarily in public safety, not in reconstruction physics.

In general, most accidents can be divided into the following events: pre-impact dynamics (trajectory), impact (collision) and post-impact dynamics (trajectory). The position, orientation and velocity of each vehicle are usually required at a minimum of three points: impact, separation (initial post-impact) and rest (final post-impact). Vehicle collisions are usually assumed to be planar events. Vehicle pre- and post-collision trajectories, however, can often be nonplanar.

The physical evidence at the scene that is most important to reconstruction is the rest position and orientation of the vehicle(s), skid marks or other indications of vehicle pre-impact and post-impact trajectory, road surface and roadside conditions, visibility and the weather. The point of impact is also important to an accident reconstruction, but usually it must be estimated. The distance from impact to rest, along the assumed post-impact trajectory, combined with an estimate of the tire/surface friction coefficient, can be used in a work-energy calculation to estimate the post-impact velocity of the vehicle. Conservation of momentum or impulse-momentum methods may be used to model the collision event. Trajectory simulation may be used to model the pre- and post-impact trajectories of a vehicle. For non-impact accidents, work-energy calculations or trajectory simulation are the only approaches usable for reconstruction. Rollover accidents are generally approached using impulse-momentum and velocity profile techniques.

The physical evidence on the vehicle(s) that is most important to reconstruction is the vehicle VIN, make, model, year, body style, and damage measurements. External damage measurements can be used to estimate speed change during collision. Damage within the occupant compartment can indicate occupant motion and possible injury causation mechanisms.

Sir Isaac Newton is credited with proposing the fundamental Laws of Motion, in 1687, that are still used for most ground vehicle dynamic analyses (5). Newton's first law of motion deals with the conservation of momentum, in the absence of external forces. Newton's second law of motion deals with the change of momentum produced by the application of external forces. Finally, Newton's third law of motion states that contact forces between bodies (vehicles) must have equal magnitude and act in opposite directions along the same line of action. Also commonly used are the concepts of kinetic and potential energy of mechanical systems, as well as the dissipation of mechanical energy through work. The basic relationships between displacement, velocity and acceleration are commonly used, as are constant velocity and constant acceleration approximations. These are the fundamental principles of physics used in traffic accident reconstruction.

The Work-Energy Relationship

The work done by a vehicle as it moves along its trajectory dissipates the mechanical energy of the vehicle. A good example of this is a vehicle braking or skidding to rest. (Accelerating the vehicle can be thought of as negative work that adds energy to the vehicle.) For accidents where the vehicle does not roll over, the mechanical energy is composed of kinetic energy, from the in-plane linear and yaw velocities, and potential energy changes from moving on a sloped surface. The work done by the vehicle is usually produced at the tire/surface interface of each tire. In a rollover, the roll (or pitch) kinetic energy becomes far more important than the yaw kinetic energy, and the work is done by the vehicle body impacting the surface.

Equation 1 is the two-dimensional work energy relationship for a vehicle moving on a sloped surface, without rollover. Equation 1 is a single scalar equation in 10 parameters. Only one of the parameters can be calculated, so the other nine must be specified. Usually the external force and vehicle trajectory are specified or estimated and the vehicle properties (m & J) are measured or approximated. This leaves one of the four velocities as the unknown.

$$\frac{1}{2}mv_1^2 + \frac{1}{2}J\omega_1^2 - mgh_1 = \int_{trajectory} F \cdot ds + \frac{1}{2}mv_2^2 + \frac{1}{2}J\omega_2^2 - mgh_2 \,, \qquad (1)$$

Where m is the vehicle mass.

v_1 is the vehicle linear speed along the vehicle trajectory path at time #1.

J is the vehicle yaw mass moment of inertia about the CG. ($J=mk^2$, where k is the yaw radius of gyration)

ω_1 is the vehicle angular (yaw) speed at time #1.

g is the acceleration of gravity.

h_1 is the elevation of the vehicle at time #1. (Positive down!)

F is a resultant applied eternal force. (Usually tire or impact forces)

ds is the incremental distance along the vehicle trajectory path.

v_2 is the vehicle linear speed along the vehicle trajectory path at time #2.

ω_2 is the vehicle angular (yaw) speed at time #2.

h_2 is the elevation of the vehicle at time #2. (Positive down!)

(Note: The trajectory starts at time #1 and ends at time #2.)

The post-impact kinetic energy of the vehicle must be entirely used up in skidding or braking work for the vehicle to come to rest. A work-energy expression describing this relationship is presented in Equation 2.

$$\frac{1}{2}m_s v_s^2 + \frac{1}{2}J_s\omega_s^2 = \sum_{trajectory} m_k g\left(u_k\mu_k - \frac{dh}{ds} \right)\Delta s \,, \qquad (2)$$

Where m_s is the vehicle mass at separation.

v_s is the vehicle linear speed at separation.

J_s is the vehicle yaw mass moment of inertia about the CG at separation. ($J=mk^2$, where k is the yaw radius of gyration)

ω_s is the vehicle angular speed at separation.

m_k is the mass of the vehicle. (Usually $m_k = m_s$.)

g is the acceleration of gravity.

u_k is the friction utilization coefficient. ($0<u_i\leq 1$)

μ_k is the assumed average tire/surface friction coefficient.

$\frac{dh}{ds}$ is the average surface slope. (A positive slope is downhill.)

Δs is the length of the trajectory segment. (Arc length.)

Equation 2 is a single scalar equation in at least nine parameters, but only one of the parameters can be calculated at a time. Therefore, the other seven must be specified. Usually, the post-impact linear speed is sought, which means that the post-impact angular velocity, vehicle mass and yaw moment of inertia, tire/surface friction coefficient, surface slope and trajectory path length are all estimated. This equation can be very conveniently programmed into a spreadsheet to make it quick and easy to use.

Vehicle Trajectory Simulation

Vehicle trajectory simulation is a technique for computing the path of a vehicle's motion given the initial position, orientation and velocity of the vehicle, as well as driver control (steering and traction or braking) information as a function of time. Such a simulation is based on Newton's second law of motion and is expressed as the solution of a set of simultaneous ordinary differential equations (6).

$$\frac{dv}{dt} = \frac{1}{m}\,\Sigma\text{externally applied forces}\ , \tag{3}$$

$$\frac{d\omega}{dt} = I^{-1}\,\Sigma\text{externally applied torques}\ ,$$

Where $\frac{dv}{dt}$ is the linear acceleration vector.

$\frac{d\omega}{dt}$ is the angular acceleration vector.

t is time.

I is the mass moment of inertia matrix.

(Note: externally applied forces and torques are produced by tire/surface contact and by aerodynamic drag.)

These equations are first order in velocity and second order in position or orientation.

Equation 3 forms an initial value problem where the position, orientation and velocity at an initial time are specified and the position, orientation, velocity and acceleration of the vehicle at subsequent times are computed. Unfortunately, the initial velocity of the vehicle is usually not known, but position and orientation of the vehicle at the end and at intermediate points along the trajectory path can be estimated. This makes the solution of the differential equations an iterative process. First, the initial velocity of the vehicle is estimated. Then the differential equations are solved, producing a simulated vehicle path. If the simulated vehicle path is not sufficiently close to the path estimated from the evidence and witness statements, then the initial velocity estimate is revised and the equations are solved again. The process terminates when the simulated path is sufficiently close to the estimated path, and it results in an estimate of the vehicle speed at the beginning of the trajectory path.

If a two-dimensional simulation is performed, three second order ordinary differential equations (x, y & yaw) are required to describe the vehicle trajectory. If a three-dimensional simulation is performed, six second order ordinary differential equations (x, y, z, yaw, pitch & roll) are required to describe the vehicle trajectory, in addition to any tire deflection or suspension dynamics models. The differential equations are usually solved numerically using Runge-Kutta or predictor-corrector techniques.

In addition to the differential equations of motion, a simulation requires a model of tire/surface interaction to produce tire tractive and braking forces, as well as side forces for steering and skidding. A 3-D tire model would also include tire vertical forces caused by a nonplanar surface. These models are empirically derived based on the tire friction coefficient, rolling resistance and side force versus slip angle tests at various tire loads and camber angles.

For vehicle trajectory simulation, a substantial amount of vehicle, tire and terrain data is required to produce meaningful results. The vehicle mass is easily determined by weighing a similar vehicle (same make, model, year, body style, etc.) in a known condition, then adding the weight of the occupants and baggage and subtracting the weight of used fuel, etc. The vehicle yaw, pitch, roll and yaw-roll couple mass moments of inertia are difficult to measure, requiring specialized and expensive apparatus, but may be estimated. The tire/surface friction coefficient as a function of slip and camber angles may be measured experimentally using a specialized skid trailer or it can be estimated from existing tire test data. Terrain surface geometry must be gathered from a site survey.

Several vehicle trajectory simulation codes are currently available. Among them are HVOSM (7), SMAC (8), T3DRS (9, 10), VTS (11). HVOSM (Highway-Vehicle-Object Simulation Model) was developed by FHWA to simulate 3-D, single-vehicle trajectories over uneven surfaces. SMAC (Simulation Model for Automobile Collisions) was developed for NHTSA as a 2-D vehicle trajectory simulator with a vehicle collision model. T3DRS is a 3-D tractor-trailer simulator developed at the University of Michigan

Transportation Research Institute. VTS is a 2-D, single-vehicle trajectory simulator developed by Dale O. Anderson for Collision Safety Engineering.

Vehicle Crush Energy

In 1976, the CRASH 2 computer program (12, 13), developed under NHTSA sponsorship to perform crash severity calculations for national traffic accident statistical databases (NCSS & NASS), introduced a relatively simple computational procedure for calculating the amount of collision energy absorbed by vehicle crush (Equation 4).

$$CE = \int_{\text{damage}} \left(Ax(w) + \tfrac{1}{2}Bx(w)^2 + C \right) dw \,, \tag{4}$$

Where x(w) is the average residual crush depth versus width profile of the vehicle damage.
A is the linear coefficient of the damage energy correlation.
B is the quadratic coefficient of the correlation.
C is the constant of the correlation. $\left(\text{usually } C = \frac{A^2}{2B}\right)$
dw is the incremental width across the damaged area.

The coefficients A, B and C for CRASH 2 were estimated from crash test data, mostly frontal compliance tests, for general classes of automobiles and light trucks (12, 14). The values of these coefficients were later revised for the CRASH 3 program (15, 16, 17). While this is a simplified analysis (compared to finite or boundary element analyses) which treats the vehicle body as a homogeneous crush layer, it can provide a useful approximation of the energy dissipated by the collision event and can provide a check on impulse-momentum collision calculations. This model is easily programmed on a spreadsheet.

For a single-vehicle collision with a fixed, rigid object, a simple mechanical energy balance using the calculated vehicle crush energy relates the vehicle impact and separation velocities.

$$\frac{1}{2}mv_i^2 + \frac{1}{2}J\omega_i^2 = CE + \frac{1}{2}mv_s^2 + \frac{1}{2}J\omega_s^2 \ , \tag{5}$$

Where v_i is the vehicle linear speed at impact.
ω_i is the vehicle angular (yaw) speed at impact.
v_s is the vehicle linear speed at separation.
ω_s is the vehicle angular (yaw) speed at separation.

For a two-vehicle collision, Equation 5 must be expanded to include the kinetic energies of both vehicles at impact and separation and the total crush energy of both vehicles. Since vehicle collisions are rapid, nonequilibrium events involving two deformable objects, it is difficult to assume that the crush energy of each individual vehicle accounts for its own loss of kinetic energy during the collision.

Collision Models - 2-D Particle Conservation of Momentum

The collision event typically lasts less than 0.1 seconds (100 milliseconds). During this time, there are enormous forces and accelerations acting on the vehicle body, but neither the vehicle nor the occupants move very far during the actual collision event. Thus vehicle aerodynamic and tire skid forces can be neglected and the collision can be modeled as conserving momentum.

If the collision is modeled as a planar impact of two particles that do not have any rotation, the following system of equations (Equation 6a) applies:

$$m_1 v_{1i} + m_2 v_{2i} = m_1 v_{1s} + m_2 v_{2s}\,, \qquad (6a)$$

Where m_1 is the mass of vehicle #1.
m_2 is the mass of vehicle #2.
v_{1i} is the velocity (vector) of vehicle #1 at impact.
v_{2i} is the velocity (vector) of vehicle #2 at impact.
v_{1s} is the velocity (vector) of vehicle #1 at separation.
v_{2s} is the velocity (vector) of vehicle #2 at separation.

Equation 6a represents two linear scalar equations with eight potential unknown speeds, assuming the vehicle masses are known (Equation 6b).

$$m_1 vx_{1i} + m_2 vx_{2i} = m_1 vx_{1s} + m_2 vx_{2s}\,, \qquad (6b)$$
$$m_1 vy_{1i} + m_2 vy_{2i} = m_1 vy_{1s} + m_2 vy_{2s}\,,$$

Where vx_{1i} is the speed of vehicle #1 in the "X" direction at impact. ($vx_{1i} = V_{1i}\cos(\phi_{1i})$, where V_{1i} is the pre-impact speed and ϕ_{1i} is the direction of the pre-impact velocity vector)
vx_{2i} is the speed of vehicle #2 in the "X" direction at impact. ($vx_{2i} = V_{21}\cos(\phi_{2i})$)
vx_{1s} is the speed of vehicle #1 in the "X" direction at separation. ($vx_{1s} = V_{1s}\cos(\phi_{1s})$, where V_{1s} is the post-impact speed and ϕ_{1s} is the direction of the post-impact velocity vector)
vx_{2s} is the speed of vehicle #2 in the "X" direction at separation. ($vx_{2s} = V_{2s}\cos(\phi_{2s})$)
vy_{1i} is the speed of vehicle #1 in the "Y" direction at impact. ($vy_{1i} = V_{1i}\sin(\phi_{1i})$)
vy_{2i} is the speed of vehicle #2 in the "Y" direction at impact. ($vy_{2i} = V_{2i}\sin(\phi_{2i})$)
vy_{1s} is the speed of vehicle #1 in the "Y" direction at separation. ($vy_{1s} = V_{1s}\sin(\phi_{1s})$)
vy_{2s} is the speed of vehicle #2 in the "Y" direction at separation. ($vy_{2s} = V_{2s}\sin(\phi_{2s})$)

In some cases, the vehicles stick together after impact so that the post-impact velocity is the same for both vehicles. Equation 6a can then be simplified as follows:

$$m_1v_{1i} + m_2v_{2i} = (m_1 + m_2)v_s \,, \qquad (6c)$$

Where v_s is the velocity (vector) of both vehicles at separation.

If the post-impact vehicle velocities are calculated by the work-energy relationship (Equation 2), or by trajectory simulation, and the vehicles are assumed to be traveling without sliding sideways before the collision, the pre-impact velocity of each vehicle can be calculated from Equation 6a, as long as the vehicles are not travelling along the same line (a colinear collision).

In a colinear collision, Equation 6a reduces to a scalar equation and only one of the velocities may be calculated. In that case, some reconstructionists employ a "restitution" equation to augment Equation 6a and allow for the solution of two unknown velocities.

$$e(v_{1i} + v_{2i}) = v_{1s} + v_{2s} \,, \qquad (6d)$$

Where e is the coefficient of restitution: e=0 for perfectly plastic collisions (i.e., the vehicles stick together), e=1 for perfectly elastic collisions (i.e., kinetic energy is conserved). Usually e is given a small nonzero value.

Usable values for the coefficient of restitution for vehicle collisions must be determined experimentally and include relative vehicle mass and body stiffness effects. If the restitution equation is used in conjunction with a non-colinear collision, three unknown velocities may be calculated.

Computation of pre-impact velocities using particle momentum relationships (Equations. 6a-c) is a straightforward process of solving simultaneous linear equations. Note that any of the 10 parameters (two vehicle masses and eight velocity components) may be calculated using these equations provided that the other eight are specified. This model is readily programmed on a spreadsheet or an equation solver (18).

Collision Models - 2-D Rigid Body Impulse Momentum

Since cars and trucks are objects that can rotate and whose rotational kinetic energy can be quite significant, the D collision momentum analysis can be extended to include rotation by assuming that the vehicles are rigid bodies that rotate about their center of gravity. This leads to the system of equations shown in Equation 7 (18, 19).

The first vector equation in Set 7 is the familiar particle conservation of momentum equation (6a) presented earlier. The next two equations in Set 7 express the conservation of angular momentum of the system. The last equation in set 7 is a constraint equation requiring that the colliding vehicles reach a common velocity at the center of contact just before separation. This is a set of six simultaneous equations in 20 parameters; therefore, as many as six unknowns can be calculated. If the vehicle inertial properties (m & k) are measured or estimated and the damage radius (r) is estimated, then six of the 12 remaining velocity components must be estimated and the other six may be calculated.

$$m_1 v_{1i} + m_2 v_{2i} = m_1 v_{1s} + m_2 v_{2s} \,, \tag{7}$$
$$k_1^2 \cdot \omega_{1i} - r_1 \times v_{1i} = k_1^2 \cdot \omega_{1s} - r_1 \times v_{1s} \,,$$
$$k_2^2 \cdot \omega_{2i} - r_2 \times v_{2i} = k_2^2 \cdot \omega_{2s} - r_2 \times v_{2s} \,,$$
$$v_{1s} + \omega_{1s} \times r_1 = v_{2s} + \omega_{2s} \times r_2 \,,$$

Where k_1 is the yaw radius of gyration of vehicle #1. $\left(J_1 = m_1 k_1^2\right)$

k_2 is the yaw radius of gyration of vehicle #2. $\left(J_2 = m_2 k_2^2\right)$

ω_{1i} is the yaw speed of vehicle #1 at impact.
ω_{2i} is the yaw speed of vehicle #2 at impact.
ω_{1s} is the yaw speed of vehicle #1 at separation.
ω_{2s} is the yaw speed of vehicle #2 at separation.
r_1 is the radius vector from the cg of vehicle #1 to the center of damage.

Vehicle Velocity-Time Diagrams

A velocity-time (V-T) diagram provides an excellent consistency check for a traffic accident reconstruction (20). It is also an essential preliminary step to performing video animation from manual reconstruction calculations. When working from manual calculations, the V-T diagram will most likely be piecewise linear, as shown in Figure 3.

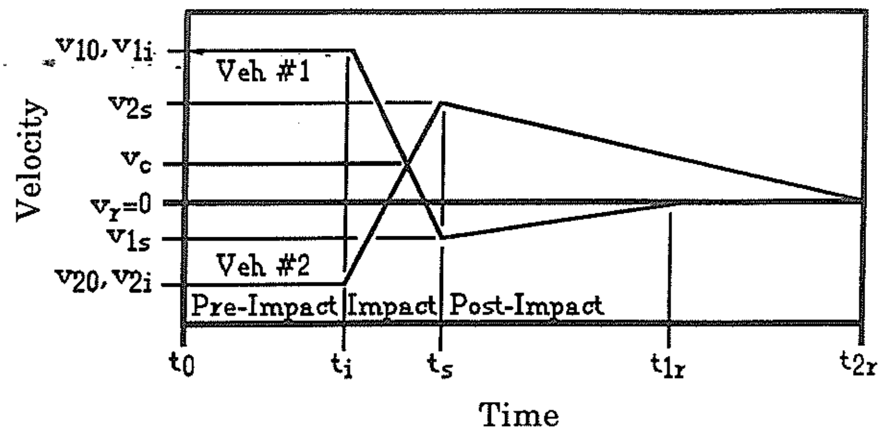

Figure 3. Velocity-time diagram for a two-vehicle head-on collision

Trajectory simulations produce continuous velocity versus time curves. The collision event (t_s - t_i) usually lasts less than 0.1 second. The time it takes each vehicle to come to rest following a collision (t_r - t_s) is not necessarily the same. The pre-impact time interval (t_i - t_0) depicted on the V-T diagram is essentially arbitrary. The smoothness of the V-T curve is a measure of the continuity of the solution. Abrupt changes in slope should only happen during an impact.

Usually a V-T curve depicts linear velocity along the trajectory path, which has been previously established. However, for complicated accidents, separate V-T diagrams with a common time axis can be drawn for the linear velocity components along the coordinate axes (x, y & z), as well as the rotational velocity components (yaw, pitch & roll). Once a consistent set of V-T diagrams has been established, vehicle position and orientation, as functions of time, can be calculated for animation scripting. Piecewise linear V-T curves calculations are easily programmed on a spreadsheet.

The area under a V-T curve is displacement. If the area is calculated in a step-wise manor, a displacement versus time curve is produced. The area under the V-T curve from the initial time to impact represents the pre-impact distance travelled by the vehicle. The area under the V-T curve from impact to separation represents the distance travelled by the vehicle during the collision (usually less than a few feet). The area under the V-T curve from separation to rest represents the post-impact or runout distance travelled by the vehicle and should be equal to the value estimated from scene measurements.

The slope of the V-T curve is vehicle acceleration. A straight-line segment on a V-T curve represents a period of constant acceleration. A curved-line segment represents a period of changing acceleration. The average slope of the V-T curve from the initial time to impact represents the average pre-impact vehicle acceleration, which is related to pre-impact braking. The average slope of the V-T curve from impact to separation represents the average collision deceleration experienced by the vehicle, from which can be calculated the collision impulse. The average slope of the V-T curve from separation to rest represents the average post-impact vehicle acceleration.

METHODOLOGY

The underlying purpose of this research effort was to develop a computer-aided traffic accident reconstruction capability emphasizing the presentation of the reconstruction as a video animation for litigation purposes. The IBM-PC/AT personal computer platform was selected as the target low-cost reconstruction and animation workstation. The description low-cost relates to the selected computer platform. Expensive, professional grade, video equipment is required for any approach considered.

The animation workstation developed in this project is based on an IBM-PC/AT compatible microcomputer with a 16-Mhz Intel 80386 microprocessor and a 80287 floating point co-processor, a Truevision TARGA-16 frame buffer with 512x482 resolution and 32,768 displayable colors, a Truevision VDI/O RGB-to-NTSC video signal converter, a BCD-4000 animation controller, and a Sony VO-5850 3/4-inch U-Matic editing tape recorder (Fig. 4). This system has performed well through over two years of intensive work.

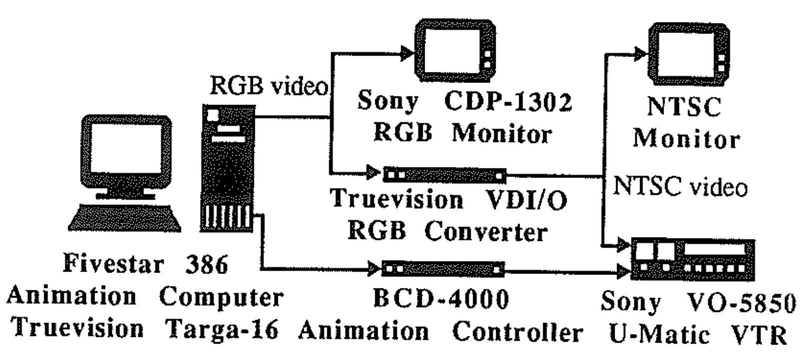

Figure 4. The LTRC traffic accident reconstruction workstation

Initially it was proposed to write the animation code as a major part of this research effort, but subsequent review of commercial computer graphics software revealed that affordable, general-purpose, video animation software was already beginning to appear on the market for personal computers. At that point it was decided to alter the focus of the research to provide connectivity between traffic accident reconstruction data and commercial video animation software. The basis of this decision was to provide a more mature and functional modeling, rendering and animation capability than could be provided by custom code within the time and resource limits of this project. The animation code selected for use in this project was TOPAS (Three dimensional Object Processing and Animation Software) (21) which is marketed by AT&T Graphics Software Labs in Indianapolis, Indiana.

All the commercial video animation software for IBM-PC/AT computers, which was reviewed in the early stages of this project, was found to be directed toward the broadcast video art market for creating "flying logos," program titles, etc. These programs were not designed as engineering modelers to produce dimensionally accurate vehicle or terrain models (3, 2). As a result, the primary research focus became the development of modeling and animation interfaces which would allow the creation of vehicle and terrain models and motion scripts for the vehicle models using external codes. This goal was accomplished by the development of the AMFTOPAS (22) model file generation and ANITOPAS (23) animation script file generation subroutines, which were written in FORTRAN 77. These subroutines produce ASCII text files that can be read by the TOPAS software.

Another facet of this project was to evaluate vehicle trajectory simulation as an accident reconstruction tool. The HVOSM-RD2 (Highway-Vehicle-Object Simulation Model - Roadside Design version) code (7), developed by the Federal Highway Administration in the 1970's, was selected as the 3-D, single-unit vehicle simulator for this project. It was ported to the IBM-PC/AT computer and slightly modified to produce a vehicle trajectory time-history file that could be used for animation.

In assessing the role of trajectory simulation for litigation, it was determined that there was insufficient experimental data available in the public literature to validate vehicle trajectory simulators successfully. For courtroom use, a trajectory simulation would need to be run on experimental vehicle trajectory data that were visibly similar to the reconstructed trajectory of the case vehicle. If a simulated trajectory could be shown as an interpolation of available experimental data rather than an extrapolation, it could probably be successfully defended in court. This suggests that if the state were to decide to use trajectory simulation as a primary reconstruction tool, it would have to be prepared to conduct a series of validation experiments with instrumented vehicles, which could then be presented as evidence that the simulator works accurately and reliably in the type of case on which it is used.

BRIEF CHRONOLOGY

The first phase of this research involved an extensive literature search of existing accident reconstruction tools and an evaluation of which tools might be most relevant to the types of accidents encountered within the state of Louisiana. Several codes including HVOSM RD-2 and VD-2 versions (Highway-Vehicle-Object Simulation Model) (7), CRASH-3 (Calspan Reconstruction of Accident Speeds on the Highway) (15), SMAC (Simulation Model of Automobile Collisions) (8), IMPAC (19), T3DRS (8, 10), and others were identified and evaluated for implementation. HVOSM-RD2 was determined to be most appropriate to this research effort because it is a 3-D single-vehicle trajectory simulation. The FORTRAN 4 source code was slightly modified for the IBM PC/AT platform.

Additional research conducted in the first phase included a thorough review of graphics hardware and software, particularly as it related to three-dimensional solid modeling and animation. The option of developing a customized, three-dimensional modeling and animation package was discarded early because of the availability of commercial codes with outstanding features. At the time this research was initiated, only a couple of companies were marketing PC-based modeling and animation packages, and those packages (TOPAS and CUBICOMP) were primarily developed for the broadcast and video art industries. Nevertheless, it was found that the software used published ASCII file formats which could be accessed through external codes. TOPAS was found to have better pricing and a larger user base.

DISCUSSION OF RESULTS

3-D DYNAMICS OF SINGLE UNIT GROUND VEHICLES

The standard coordinate system for vehicle dynamics is the SAE (Society of Automotive Engineers) system (see Fig. 3-3). It is a right-handed system with the Z axis pointing toward the center of the earth. This standard is used for both fixed and body-centered coordinate systems describing the motion of the vehicle body.

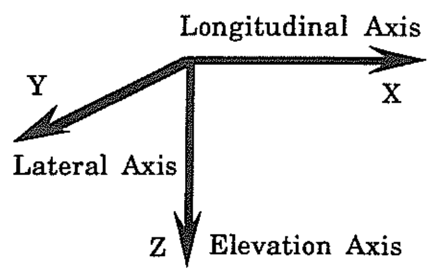

Figure 5. SAE standard coordinate system for vehicles and terrain.

The gross motions of the body of a single-unit ground vehicle can be described by the longitudinal (X), lateral (Y) and vertical (Z) displacements of the vehicle center of gravity with respect to a fixed inertial (terrain) coordinate system and by the Euler orientation angles of the vehicle body with respect to a body-centered inertial (vehicle) coordinate system at the vehicle CG.

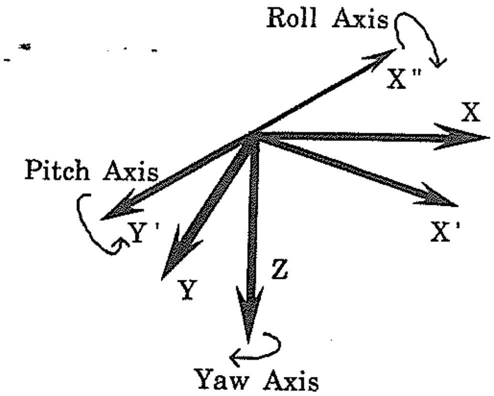

Figure 6. Vehicle Euler Rotation Axes - Yaw, Pitch & Roll.

In addition to the gross vehicle body motions, the motions of individual wheels are also usually desired, since they indicate the response of the vehicle suspension system. Wheel displacement and orientation is usually computed with respect to the vehicle body and consists of axle devlection, wheel steer, camber (tilt) and spin angles (see Fig. 3-5). The SAE standard coordinate system for tires is a hybrid system. As with the vehicle coordinate system, the Z axis points toward the center of the earth, but for tire forces, the system is right-handed, and for tire orientation, the system is left-handed. The left-handed system allows the wheel spin angles to be positive as the vehicle is moving forward.

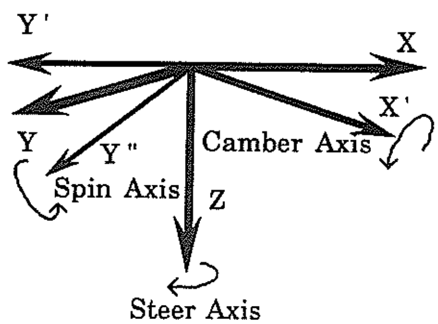

Figure 7. Tire Rotation Axes - Steer, Camber & Spin.

TOPAS

The animation software chosen for this research was TOPAS (Three-dimensional Object Processing and Animation Software) which is marketed by AT&T Graphics Software Labs (21) for use on IBM-PC/AT compatible microcomputers using a Truevison TARGA or ATVista frame buffer. TOPAS is a user-friendly, menu-driven, three-dimensional modeling and animation software package based on polygon mesh models. TOPAS allows a maximum of 1000 objects per model with up to 2800 polygons per object. Colors for objects and individual polygons can be assigned using preset color choices or using an RGB (Red-Green-Blue) color bar, allowing up to 32,768 colors to be displayed simultaneously. Image enhancements include Phong, Gouraud or flat shading, transparency, and specular highlights.

TOPAS provides a wide range of building and editing tools for model construction. These models can be stored as either binary model files or as ASCII (American Standard Code for Information Interchange) text model files. Using the ASCII text file format, these model files can be created outside of TOPAS by any standard text editor or by specialized external codes and then read in.

incremental rotations to their respective axes, and the transformed model will be properly oriented in each frame of the animation.

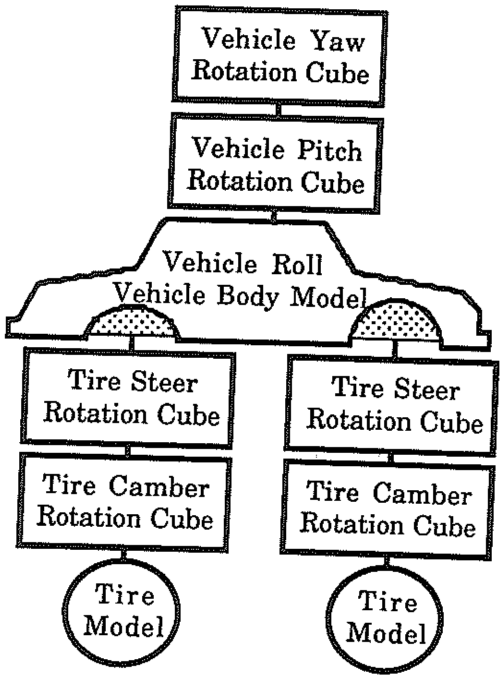

Figure 8. Hierarchical vehicle model.

A simple car model consisting of a body and four tires would require the creation of a hierarchical model composed of 15 separate objects (or object images) to represent vehicle and tire motions fully. The vehicle body would require two hidden rotation cubes in addition to the body model object, while each tire would require two hidden rotation cubes in addition to the tire model object. If axle or bogie motion is to be represented, more objects must be added to the model.

Modeling Terrain Surfaces

The terrain is created as a surface or solid model, but usually only the top surface is of importance. Road surfaces are readily created from a few simple measurements, while roadside surfaces (embankments, slopes, ditches, etc.) usually require a site survey to be adequately represented. Roadside appurtenances such as guard rails, utility poles, traffic signs, etc., usually have standard geometries and can be stored in a library for reuse once they are created.

Preliminary terrain models can be created from the terrain surface description used by the trajectory simulation. HVOSM (7) allows the definition of up to five terrain tables, each table having a maximum grid size of 21 by 21. The x and y coordinate positions as well as the elevation are given at each grid point. These grids can be used to form a polygonal mesh to model the terrain surface. Since TOPAS (21) requires that all polygons be flat, the rectangular grids provided by the terrain tables must be broken up into triangular polygons for the surface model. For completeness, four side surfaces and a bottom surface are usually added to the terrain model.

Topographical descriptions of the site can be determined from site surveys using traditional ground instruments (transits, levels, etc.), aerial photogrammetry or satellite-based global positioning systems. Each of these methods results in data of different scales and levels of accuracy that can be represented two-dimensionally as contour maps or three-dimensionally as terrain models. Recently, topographic data have been converted to forms that

REFERENCES CITED

1. Cox, C. F., "Microcomputer-Based Animation of Vehicle Trajectory Simulations," Master's Thesis, Louisiana Tech University, Ruston, LA, Nov. 1989, pp 33-36.

2. Guice, Leslie K., and Anderson, Dale O., "Development of a Traffic Accident Reconstruction Animation System," Proceedings of the 6th. National Conference on Microcomputers in Civil Engineering, Orlando, FL., Nov 1988, pp 296-300.

3. Anderson, Dale O., Cox, Carey F., and Guice, Leslie K., "Creating Video Animation from Off-Highway Vehicle Simulations," Computer Applications: Design, Analysis and Production of Off-Highway Equipment, SAE SP-795, Society of Automotive Engineers (SAE), Warrendale, PA, Sep 1989, pp 121-126, (also SAE paper 891924).

4. Nguyen, John S. D., "Development of a Post-Processor for Traffic Accident Reconstruction Animation," Master's thesis, Louisiana Tech University, Ruston, LA, 1988, pp 4-12.

5. Beer, Ferdinand P., and Johnston, E. Russell Jr., Mechanics for Engineers: Statics and Dynamics, fourth ed., McGraw-Hill, New York, 1987, pp 2-3.

6. D'Souza, A. Frank, and Garg, Vijay K., Advanced Dynamics: Modeling and Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1984, pp 111-114.

7. Segal, D. J., Highway-Vehicle-Object Simulation Model - 1976, Vol. 1: User's Manual, NTIS PB 267-401, FHWA-RD-76-162, Vol. 2: Programmer's Manual, NTIS PB 267-402, FHWA-RD-76-163, Vol. 3: Engineering Manual - Analysis, NTIS PB 267-403, FHWA-RD-76-164, Vol. 4: Engineering Manual - Validation, NTIS PB 267-404, FHWA-RD-76-165, National Technical Information Service, Springfield, VA, Feb. 1976.

8. Soloman, P. L., The Simulation Model for Automobile Collisions Operator's Manual, National Highway Traffic Safety Administration, Oct. 1974.

9. Gillespie, T. D., et. al., Truck and Tractor-Trailer Dynamic Response Simulation: Volume 3, User's Manual, NTIS PB 81-174542, National Technical Information Service, Springfield, VA, Dec. 1980.

10. Bernard, J., et al., Truck and Tractor-Trailer Dynamic Response Simulation: Volume 1, Summary Report, NTIS PB 81-174526, Volume 2, Technical Report, NTIS PB 81-174534, Programmer's Manual, NTIS PB 81-178360, National Technical Information Service, Springfield, VA, Dec. 1980.

11. Anderson, D. O., Woolley, R. L., Tagg, M. D., and Warner, C. Y., "The VTS Single-Vehicle Trajectory Simulation," Field Accidents: Data Collection, Analysis, Methodologies, and Crash Injury Reconstructions, SAE P-159, Society of Automotive Engineers, Warrendale, PA, Feb. 1985, pp 219-32, (also SAE paper 850252).

12. McHenry, Raymond R., and Lynch, James P., Crash2 User's Manual, Sept. NTIS PB 262-882, National Technical Information Service, Springfield, VA, 1976.

13. Warner, Charles Y., Anderson, Dale O., and Albrecht, Horst, "A Critique of the CRASH Accident Reconstruction Computer Program," Proceedings of the 6th. International Symposium on Automotive Technology and Automation, Wolfsberg, West Germany, 1978.

14. Woolley, Ronald L., Warner, Charles Y., and Tagg, Melany D., "Inaccuracies in the CRASH3 Program," Field Accidents: Data Collection, Analysis, Methodologies, and Crash Injury Reconstructions, Society of Automotive Engineers (SAE), Warrendale, PA, 1985, pp 265-283, (also SAE paper 850255).

15. Crash 3 Technical Manual, IBM-PC version, Accident Investigation Division, National Center for Statistics and Analysis, NHTSA, USDOT, 1986.

16. Jones, Ian S., and Jennings, P. W., "Development and Evaluation of the CRASH 2 Program for Use Under European Conditions," Proceedings of the 5th. IRCOBI, Oxford, England, 1980.

17. Monk, M., and Guenther, D., "Update of CRASH 2 Computer Model Damage Tables," Final Report for NHTSA Project SLR-16, to be published, Cited by Jones in 1980 (16).

18. Liang, Kun "Kenneth," "Evaluation of Commercial Equation Solvers and Spreadsheets for Performing Two Dimensional Vehicle Collision Calculations," Master's Thesis, Louisiana Tech University, Ruston, LA, 1990, pp 20-46.

APPENDIX A
ANIMATION OF TRAFFIC ACCIDENT RECONSTRUCTIONS

Video animation of traffic accident reconstructions is performed on an animation workstation similar to that shown in Figure A-1 (A1). The system is based on an IBM-PC/AT compatible (EISA or ISA) computer connected to a professional video editing system. The video system and animation software are expensive compared to traditional personal computer software and peripherals, but the system is low-cost when compared to workstation and supercomputer-based video animation systems.

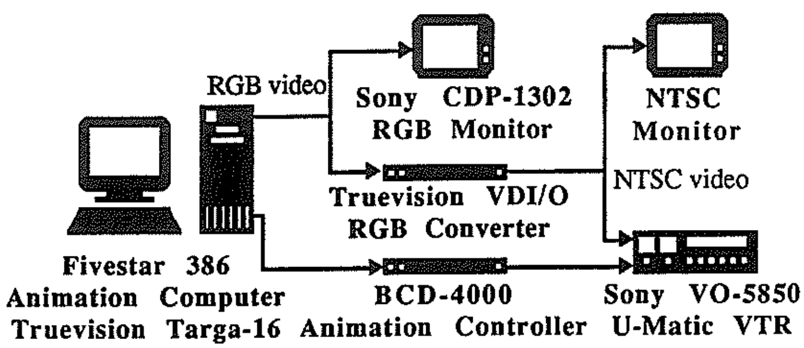

Figure A-1 - LTRC traffic accident reconstruction video animation workstation.

The animation software chosen for this research was TOPAS (Three-dimensional Object Processing and Animation Software) which is marketed by AT&T Graphics Software Labs (A2) for use on IBM-PC/AT compatible microcomputers using a Truevison TARGA or ATVista frame buffer. TOPAS is a menu-driven, 3-D, modeling and animation software package based on polygon mesh surface models. TOPAS allows a maximum of 1000 objects, 16,000 vertices and 16,000 polygons per model with up to 2000 polygons per object. Object colors can be chosen from a pallet of 32,768 colors. Surface rendering models include Phong, Gouraud or flat shading, transparency, and specular highlights.

TOPAS ASCII model (AMF) files are fairly straightforward ASCII text files that can be created and edited by a standard text editor or by a word processor in non-document mode. Although custom codes have been developed to create 2-axle, single-unit vehicle and terrain models, these and other models may be created or edited using a text editor. TOPAS also has the facility to create or edit models, but it is difficult to maintain dimensional correctness using the TOPAS model editor.

CREATING VEHICLE MODELS

Vehicle models should use inches as the standard unit of measure, since it is the most common unit used in published American vehicle data. However, as long as the units are specified in the model file, TOPAS will scale a model constructed in any measurement system to fit the terrain model for animation.

Vehicle models are TOPAS ASCII model files, which must be created with specific characteristics. The coordinate system used is the SAE standard vehicle coordinate system (A3). The vehicle model is created with the vehicle

CG lying at the origin. The vehicle x axis is the longitudinal axis going from the CG through the front of the vehicle (Fig. A-2). The vehicle y axis is the lateral axis going from the CG through the passenger (right) side of the vehicle. The vehicle z axis is the vertical axis going from the CG through the vehicle floor toward the center of the earth.

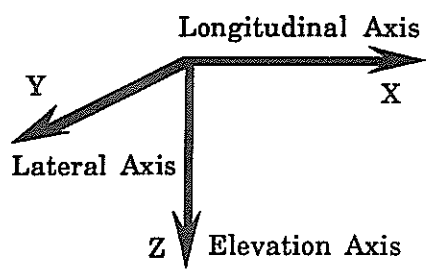

Figure A-2 - SAE vehicle body coordinate system.

An automobile or light truck model consists of 10 hidden rotation cubes (A4), the body model, and four tire model images (Fig. A-3). Only one rotation cube model need be defined since all rotation cubes required for the vehicle model can be images (instances) of the same model. The vehicle body yaw rotation cube is the parent object for the vehicle model and is named ViCGZ, where "i" is the vehicle number (i.e., 1, 2, ... , 9). It is defined with respect to the vehicle CG and has a rotation axis specified along the vehicle z (vertical) axis. The vehicle body incremental yaw rotation and the incremental linear CG displacements are applied to the yaw rotation cube. The vehicle body pitch rotation cube is next in the hierarchy and is named ViCGY. It is defined with respect to the vehicle CG and has a rotation axis specified along the vehicle y (lateral) axis. The vehicle body incremental pitch rotation is applied to the pitch rotation cube. The vehicle body model is next in the hierarchy and is named

Title Records

The title records are required only as file identification. They are not used by ANIGEN, but both records must be included even if they are blank.

Number of Vehicles Record

The number of vehicles record appears as follows:

NV

Where:

NV	- Number of Vehicles.

Vehicle Record

The vehicle record appears as follows:

X0 Y0 Z0 NT NA NW(1) NW(2) ...

Where:

X0	- Initial vehicle CG X coordinate (feet).
Y0	- Initial vehicle CG Y coordinate (feet).
Z0	- Initial vehicle CG Z coordinate (feet).
NT	- Number of time history data sets.
NA	- Number of axles on vehicle. (Numbered from front to back.)
NW(i)	- Number of wheels on axle #i. (1 or 2)

Vehicle Time History Records

The vehicles time history records appear as follows:

T	X	Y	Z	YAW	PITCH	ROLL
(axle 1)	STEERL	CAMBRL	ZL	STEERR	CAMBRR	ZR
:	:	:	:	:	:	:
(axle NA)	STEERL	CAMBRL	ZL	STEERR	CAMBRR	ZR

(Note: the field (axle i) is a comment and is not part of the record.)

Where:

T	- Time (sec.)
X	- Vehicle CG X coordinate (feet).
Y	- Vehicle CG Y coordinate (feet).
Z	- Vehicle CG Z coordinate (feet).
ROLL	- Vehicle body roll rotation angle (deg.).

PITCH - Vehicle body pitch rotation angle (deg.).
YAW - Vehicle body yaw rotation angle (deg.).
(NOTE: the following record is repeated for each axle in sequence)
STEERL - Steer angle of the left wheel on axle #i (deg.).
CAMBRL - Camber angle of the left wheel on axle #i (deg.).
ZL - Z-Coordinate of the left wheel's roll center on axle #i.
STEERR - Steer angle of the right wheel on axle #i (deg.).
CAMBRR - Camber angle of the right wheel on axle #i (deg.).
ZR - Z-Coordinate of the right wheel's roll center on axle #i.

Note that the vehicle body record comes first and contains the data set time. The wheel motion data is organized by axle. Axles are numbered in sequence from the front of the vehicle to the rear of the vehicle. A passenger car would therefore have two axles: front (#1) and rear (#2). On each axle there may be up to two wheels, which are ordered from left (driver's side) to right (passenger's side). Each axle must have at least one wheel, but no more than two.