Computer graphics displays of simulated automobile dynamics

by CALVIN M. THEISS
Cornell Aeronautical Laboratory, Inc.
Buffalo, New York

INTRODUCTION

Simulation of physical systems using digital computers has been accomplished by many people over the past few years. One such simulation program at Cornell Aeronautical Laboratory, Inc. constitutes an analytical representation of an automobile as it departs from the highway under various environmental conditions, especially under adverse ones where there is danger of collision with obstacles. As with most complex simulation programs where there are interactions among many components, the equations of motion, including the required restraints, become long and numerous. But foremost, the output data set in printed form is quite extensive and very difficult for the investigating engineer to completely comprehend. About thirty crowded pages of output are printed for a five second automobile test run.

The objective of the task described here was to convert some of these output data into a pictorial form; that is, pictures were desired for a simulated automobile undergoing various violent maneuvers. In this manner, the actions in any part of the vehicle can be easily and quickly determined. Since this objective could be applied to most simulation problems to advantage, a set of generalized computer subroutines were designed to be called in a manner similar to most plotting utility packages.

The pictures are then drawn by state-of-the-art peripheral plotting hardware. Several output methods are available to CAL users—

a. on-line 8-1/2 X 11 inch pictures by Xerox equipment,
b. off-line CALCOMP plots on 30" drawing paper,
c. off- or on-line film exposure through the use of a flying spot scanner built in-house, using a high precision cathode ray tube and a 16 mm movie camera. Commercial flying spot scanners are now available for this purpose.

Automobile simulation

The CAL Single Vehicle Accident Program simulates a moving automobile, its interaction with the terrain over which it travels and collisions with a variety of obstacles in its path. The program is very flexible in that approximately 100 input parameters describe the automobile, such as wheel base, weight, braking coefficients, nonlinear characteristics of shock absorbers, engine torque on driver wheels, etc. Likewise, various terrain profiles and obstacles can be specified for different simulation runs. For example, the radial tire force generated at each wheel is simulated by means of simple springs on smooth terrain; however, on rough terrain or when obstacles, such as curbs, are encountered, the tires are simulated by a set of compound springs radiating from the hub of the wheel. Originally, the output consisted of about sixty items of response printed in tabular form.

For a first effort at pictorial output, the automobile is considered as four different objects, namely, the two front wheels, the rear axle and the sprung mass (the body). The Single Vehicle Accident Program was modified to record dynamically on magnetic tape thirteen parameters which describe the actions of these four objects. They are—

1-3 position of the sprung mass,
4-6 attitude of the sprung mass,
7-8 deflection of each of the front wheels,
9-10 camber of each of the front wheels,
11 deflection of the rear axle,
12 roll of the rear axle,
13 steer angle applied to the front wheels.
This magnetic tape is the dynamic input to the picture generation program. Other objects, such as road markings, roadside barriers, ramps, etc., are static initial inputs to the program.

**Perspective picture generation**

A quick but incomplete review of the available literature revealed that a number of different approaches to producing three-dimensional computer graphics displays have been developed. However, none met our particular requirements to permit quick economical generation of both still and motion pictures with the existing in-house equipment available.

The pictures are drawn as line drawings using straight lines only. The periphery of a wheel, for example, is drawn as a 50-sided regular polygon. No attempt was initially made for "hidden line removal".

**Description of objects**

Each object is stored as one or more three-dimensional arrays; each entry defines a point of the object with respect to three-dimensional rectangular coordinates fixed in the object. Thus, after the appropriate coordinate transformations discussed below, a perspective picture can be drawn in two-dimensional space by starting at the first point of the first array, drawing a straight line to the second point and proceeding onto each point of the array in turn. This is done for each array of the object.

**Camera simulation**

The usual description of a simple pinhole camera as illustrated in Figure 1 is used. The position of the image, Q, on the picture plane of an object at point P in front of the camera with a focal length of F is:

\[
x_{Q} = 0
\]

\[
y_{Q} = -F\frac{y_{P}}{x_{P} - F}
\]

\[
z_{Q} = -F\frac{z_{P}}{x_{P} - F}
\]

Thus, the three-dimensional space in front of the camera is mapped onto the two-dimensional space of the picture plane (film) of the camera. The simulation of this simple camera is the crux of our picture generating program.

In addition, care must be taken to assure that attempts are not made to draw portions of the object image which are outside the picture boundary. At first thought one might expect this to occur only when

\[
|y_{Q}| \geq Y_{max}
\]

Actually this occurs also as

\[
x_{P} - F \to 0
\]

For drawing a single straight line, only the central portion, an end, or the whole line may be in view. Thus, intersections of lines and picture edges must be continually tested. For example, see Figure 2.

Line AB is defined in the object array by points A and B, the transformations of both lie outside the picture. However, the central portion of the line lies inside the boundary and must be drawn. Thus, for lines where at least one of the end points lies outside the boundary, algorithms were developed to determine which portion, if any, of the line is to be drawn.

**Definitions of coordinates and parameters**

As noted above, coordinate transformation of the arrays of points of an object in three-dimensional space...
to points of the picture in two-dimensional space is necessary. This is done with the help of the following coordinate systems and relating parameters.

a. **Inertial coordinate system**: a rectangular coordinate system fixed in free space. This is the fundamental reference base for spatial relationships among the various components.

b. **Object coordinate system**: a rectangular coordinate system fixed to the object. One exists for each object.

c. **Camera coordinate system**: a rectangular coordinate system fixed to the camera, with the origin in the center of the picture plane and the positive X axis extending toward and through the focal point.

d. **Picture coordinate system**: a two-dimensional rectangular coordinate system fixed to the picture frame.

e. **Object position**: the position of the origin of object coordinates, with respect to the inertial coordinate system (the row vector, B).

f. **Object attitude**: the angular relationship of the object coordinates with respect to the inertial coordinates in terms of Euler angles, yaw, Y; pitch, P; and roll, R (see Figure 3).

g. **Camera position**: the position of the origin of the camera coordinates with respect to the inertial coordinate system (the row vector, C).

h. **Camera attitude**: three angles, the azimuth, A, and elevation, E, of the camera line of sight (camera X axis) and the orientation of the picture (tilt), T, with respect to the inertial coordinates.

i. **Camera focal length**: (defined in Figure 1).

**Spatial relationships**

Motion and dynamic activity in the simulated scene to be "photographed" is reflected in changes over time of the position and attitude parameters (a total of six) for each object. Furthermore, change in the camera parameters are also possible. The camera could be mounted on a vehicle. Panning (attitude changing) and/or zooming (changes in focal length) are included.

Consider a point of an object, any point in the object description arrays, as a row vector, \( P_o \), described in object space. The position vector with respect to inertial space, \( P_I \), is obtainable by the transformation.

\[
P_I = P_o \times [I] + B
\]

where the transformation matrix is

\[
[I] = \begin{bmatrix}
\cos P \cos Y, & \sin P \cos Y \sin R - \cos R \sin Y, \\
-\sin P, & \cos P \sin R, \\
\end{bmatrix}
\]

Similarily, in camera space, the relationship is

\[
P_c = (P_I - C) \times [C]
\]

where the transformation matrix is

\[
[C] = \begin{bmatrix}
\cos A \cos E, & \sin A \cos E, & \sin E \\
-\sin A, & \cos A, & 0 \\
-\cos A \sin E, & -\sin A \sin E, & \cos E
\end{bmatrix}
\]

At this point the value of the x component of \( P_c \), \( x_{pc} \), is checked. If \( x_{pc} \leq F \) (see Figure 1), the point is (1) behind the camera, or (2) at the camera focal point, or (3) its corresponding picture image point will be a very large distance from the center of the picture, i.e., \( x_q \),

![Diagram](image-url)

**Figure 3—Euler angles relating body axes \((X_b, Y_b, Z_b)\) with respect to fixed axes \((X_f, Y_f, Z_f)\)**

From the collection of the Computer History Museum (www.computerhistory.org)
Thus, complete lines connecting to the point cannot be drawn.

The final transformation to the two-dimensional point in the picture plane is

$$Q = \frac{KF}{x_{ps}} P_x \times [C_P]$$

where $K$ is an enlargement factor and the transformation matrix is

$$[C_P] = \begin{bmatrix} 0 & -\cos T, \sin T \\ -\sin T, \cos T \end{bmatrix}$$

Now the point must be checked to determine whether it is within the picture frame by the following criteria—

$$|x_q| \leq W/2$$

$$|y_q| \leq H/2$$

where $W$ and $H$ are the prescribed width and height of the picture, respectively.

The picture generation program

The program was written using the FORTRAN IV source language and documented for general usage in the analysis of automobile dynamics.

Picture subroutine set

The subroutines were designed to supply the rudimentary requirements for drawing perspective line drawings. A brief description of each subroutine call follows.

a. CALL FRAME
   1. When output is to the flying spot scanner, the film is advanced one frame.
   2. When output is on the CALCOMP plotter, a new area of the paper is selected.

b. CALL FRAMESZ (W, H)
The size of the picture is established.

c. CALL CAMSET (POS, ITYPE, INVT)
The camera parameters are established.

   POS(n) n=1,2,3 specifies the camera position, C.
   POS(n) n=4, 5, 6 specifies the position of the camera focal point with respect to the inertial coordinates if ITYPE=1.
   POS(n) n=4, 5, 6 specifies the azimuth and elevation of the camera line of sight and the camera focal length if ITYPE=2.
   POS(7) specifies the rotation of the picture about the line of sight, i.e., its tilt angle.
   INVT specifies if either the vertical or horizontal axes (or both or neither) is to be reversed.

d. CALL OBAXIS (X, Y, Z, ROLL, PITCH, YAW)
The position and attitude of the object axis are established. All further calls to OBLINE refers to this object coordinate system until another call to OBAXIS occurs.

e. CALL OBLINE (X, Y, Z, N)
   X, Y, Z are arrays containing the respective components of N points with respect to the object coordinate system established by the last call to OBAXIS. All pertinent coordinate transformations are computed and a perspective line drawing is made connecting the N points in the order of the array. The drawing is “clipped” wherever it crosses the picture boundary as established by the previous call to FRAMESZ.

f. CALL CIRCLE (X, Y, Z, R, AZ, EL, TI, P, N)
The data for the arrays required by a call to OBLINE are prepared for drawing a circle centered at point X, Y, Z with radius R. The circle axis points in the direction with an azimuth AZ and elevation EL. Actually, the circle is approximated by an N-sided regular polygon. P is the returned data arrays. TI specifies the rotation or starting point of the polygon; for a true circle it has no significance.

g. CALL OBJECT (TITLE, X, Y, Z, ROLL, PITCH, YAW)
This call and the next two calls are included only to reduce the bookkeeping details for the user. TITLE is the name of an object to be drawn, whose object axis is defined by the last six arguments. The descriptive arrays for this object are stored with the object name in a common store. When the OBJECT is called, the store is searched for the object named, the coordinate transformations calculated and the object is drawn. Subroutines OBAXIS and OBLINE are called by this subroutine.

h. CALL OBJINP
This call reads data from cards giving the name and descriptive arrays of objects and stores them in the common store.

i. CALL DLTOBJ (TITLE)
The object TITLE is deleted from the common store, thus freeing storage space for new objects.
The executive routine

The executive routine reads the magnetic tape describing the thirteen parameters of the automobile dynamics. These in turn are used for determining the spatial relationships of the object coordinates for each of the four objects (the two front wheels, rear axle and sprung mass) of the automobile for each increment in time. Calls to the picture subroutines are then made to draw the pictures. Background objects such as road markings, barriers, ramps, etc., are read in from cards. Other features of the program include:

a. The frame rate (or time increment between pictures) is specified for each usage of the program. In fact, it may be changed at any point in the run. In this manner several “still pictures” may be chosen at arbitrary times throughout the simulated event.

b. Background objects may be added or deleted at any point in time.

c. Camera motion, if desired, may be specified or changed at any point in time. Further, panning and/or zooming, if desired, may be specified with given rates or automated to keep the automobile in the center of the picture.

d. The program can also write titles and subtitles in the film where desired.

Figure 4—Photographic and computer graphic display of GM parapet impact
Sample applications

Sample still pictures from several different simulated vehicle maneuvers are shown in Figures 4, 5 and 6, together with pictures of similar "real life" tests. Automatic panning is used in all cases.

CONCLUSIONS

The pictures illustrate the ease of assessing the capability of the simulation program. Through comparisons with similar "real life" test pictures, especially motion pictures, one may quickly verify the validity of a simulation program and establish a high degree of confidence in its usage.

These results encourage one to look ahead to future activity. The various other applications of this output technique to simulation is limited only by the user's imagination and needs. Improvements, of course, are desirable. First of all, the removal of hidden lines would improve the esthetic value. The literature contains several methods of varying levels of satisfying and financially feasible results. The economical inclusion of color and shading is challenging. Illustration of the deformation of objects, such as the crushing of an automobile fender, would be highly desirable.
Figure 6—Ramp jump at 44 mph
ACKNOWLEDGMENTS

I wish to express my gratitude to Raymond R. McHenry and Claron W. Swonger for their help and guidance in the development work leading to this paper.

The reported research was performed under Contract No. CPR-11-3988 with the Traffic Systems Division, Office of Research and Development, Bureau of Public Roads, U. S. Department of Transportation. The opinions, findings and conclusions expressed in this paper are those of the author and not necessarily those of the Bureau of Public Roads.

REFERENCES

1 R R McHERN Y  N J DeLEYS
Vehicle dynamics in single vehicle accidents—validation and extensions of a computer simulation.
December 1967

2 K G KNOWLTON
A computer technique for producing animated movies
Proc AFIPS Conference 1964

3 A M NOLL
Computer-generated three-dimensional movies

Computers and Automation November 1965

4 W A FETTER
Computer graphics
Annual Meeting of the American Society for Engineering Educators 1966

5 R I. MITCHELL
A computerized 3-D plotting program
Computerized Imaging Techniques, Proc of the Society of Photo-Optical Instrumentation Engineers June 1967

6 C M THEISS
Perspective picture output for automobile dynamics simulations

7 L G ROBERTS
Machine perception of three-dimensional solids
Lincoln Laboratory Technical Report 315 May 1963

8 R A WEISS
BE VISION, a package of IBM 7090 FORTRAN programs to draw orthographic views of combinations of plane and quadric surfaces
J ACM April 1966

9 P LOUTREL
A solution to the hidden-line problem for computer-drawn polyhedra