FHWA-NJ-2002-010

3D/4D SIGN REVIEW SUPPORT

FINAL REPORT
SEPTEMBER 2002

Submitted by

Dr. Gary Consolazio**
Dr. Hani Nassif, P.E.*

** Dept. of Civil & Coastal Engineering
University of Florida
Gainesville, FL, 32611

* Dept. of Civil & Environmental Engineering
Center for Advanced Infrastructure & Transportation (CAIT)
Rutgers, The State University
Piscataway, NJ 08854-8014

NJDOT Research Project Manager
Mr. Karl Brodtman

In cooperation with

New Jersey
Department of Transportation
Bureau of Research and Technology
and
U.S. Department of Transportation
Federal Highway Administration

DISCLAIMER STATEMENT

ACKNOWLEDGEMENTS

The authors wish to thank the New Jersey Department of Transportation (NJDOT) for funding the research described herein.

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA –NJ-2002-010		

4. Title and Subtitle	5. Report Date
3D/4D Sign Review Support	September 2002
	6. Performing Organization Code CAIT/Rutgers
7. Author(s) Dr. Gary Consolazio and Dr. Hani Nassif, P.E.	8. Performing Organization Report No. FHWA-NJ-2002-010
9. Performing Organization Name and Address New Jersey Department of Transportation P.O. Box 600 Trenton, NJ 08625	10. Work Unit No. 11. Contract or Grant No.
12. Sponsoring Agency Name and Address Federal Highway Administration U.S. Department of Transportation Washington, D.C.	13. Type of Report and Period Covered Final Report 14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This report discusses the development of a new software system that supports 3D/4D modeling and visualization efforts related to the review of roadside signage. By using 3D renderings of proposed signage projects and 4D "virtual drive-throughs", engineers can evaluate newly proposed signage projects directly from the point of view of a driver moving along the roadway, reading the signage.

In order to generate such visualization, the engineer must be able to model the roadway and signage in three-dimensional space and then render the model graphically from various points of view in the model. The software developed for this project addresses these needs and is made up of two components: a visualization model generation tool; and a visualization-rendering tool. The model generation component permits the user to import geometry data and image data and convert that data into a format appropriate for use by the rendering component of the system. The rendering component is capable of rendering a 3D model with sign faces, textures, etc. from any desired point of view.

A virtual drive-through of a proposed signage project can be simulated using the rendering component by defining a driver-motion path through the 3D model and moving along that path at a prescribed rate. Potential problems of excessive sign density, inadequate sight distance, and obstruction of sign faces can be identified prior to actually installing the signage in the field. By identifying such problem areas using 3D/4D modeling and visualization, expensive changes involving signage already installed in the field can be avoided.

17. Key Words	18. Distribution Statement
3D/4D modeling, visualization, software, signage	

19. Security Classif (of this report)	20. Security Classif. (of this page)	21. No of Pages	22. Price
Unclassified	Unclassified	33	

Form DOT F 1700.7 (8-69)

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This report discusses the development of a new software system that supports 3D/4D modeling and visualization efforts related to the review of roadside signage. By using 3D renderings of proposed signage projects and 4D “virtual drive-throughs”, engineers can evaluate newly proposed signage projects directly from the point of view of a driver moving along the roadway and reading the signage. In order to generate such a visualization, the engineer must be able to model the roadway and signage in three-dimensional space and then render the model graphically from various points of view in the model.

The software developed for this project addresses these needs and is made up of two components: a visualization model generation tool and a visualization rendering tool. The model generation component permits the user to import geometry data and image data and convert that data into a format appropriate for use by the rendering component of the system. The rendering component is capable of rendering a 3D model with sign faces, textures, etc. from any desired point of view. A virtual drive-through of a proposed signage project can be simulated using the rendering component by defining a driver-motion path through the 3D model and moving along that path at a prescribed rate. Potential problems of excessive sign density, inadequate sight distance, and obstruction of sign faces can be identified prior to actually installing the signage in the field. By identifying such problem areas using 3D/4D modeling and visualization, expensive changes involving moving signage already installed in the field can be avoided.

INTRODUCTION

Within the NJDOT environment, there is a frequent and regular need to quickly and easily visualize and illustrate alternative solutions to geometric highway problems in 3D that involve motorists' view of the highway and associated signage and traffic control devices. The primary objective of this project is the development of a software system capable of facilitating construction of 3D models for review of roadway geometry, signing, and roadway improvement/modification projects. The software system can be used by NJDOT personnel to review designs from an engineering design perspective, visually evaluate proposed designs, or make presentations of proposed projects to the public in a highly graphical, flexible, and easy to understand format.

At the time this project was initiated, no program existed for a personal computer that was capable of providing both easy to use model building features appropriate for representing highway objects and 4D drive-through simulation capabilities. For this reason, this project was undertaken to address the need for such software. The goal was to develop a software system that would aid engineers with the development of 3D models and in the generation of subsequent drive-through visualizations (animations).

Background

In 1997, 4D drive-through visualizations of Interstate I-280 eastbound through Newark ,New Jersey, and a proposed Rt. 21 ramp off I-280 into Newark (which has since been constructed and is now open to traffic) were built using a stand alone program developed by the Center for Advanced Infrastructure & Transportation (CAIT) at Rutgers University [(1)]. The I-280 and Rt. 21 ramp visualizations were both reviewed by NJDOT and were constructed for NJDOT during the study entitled "*4D Drive-Through Visualization of I-280 for Review of Proposed Signing*". The 3D models built for the project included 3D vertical and horizontal curvature, barriers, bridges, walls, sign structures, and signs. The 3D model of I-280 was constructed based on available data such as GPS and video log information provided by NJDOT. Data were collected from various digital, video and paper sources. GPS data provided by NJDOT consisted of latitude, longitude, and elevation data from an instrumented Automated Road Analyzer ARAN van (used as part of NJDOT's pavement management system) that was driven along the segment of roadway of interest (see Figure 1).

For the Route 21 ramp visualization, the ramp was *under construction* at the time of the project and therefore the NJDOT ARAN van could not be used. Instead, all geometrical data were taken from construction plans for the project (see Figure 2). A 3D model was then built using the horizontal and vertical layout data contained in the construction plans.

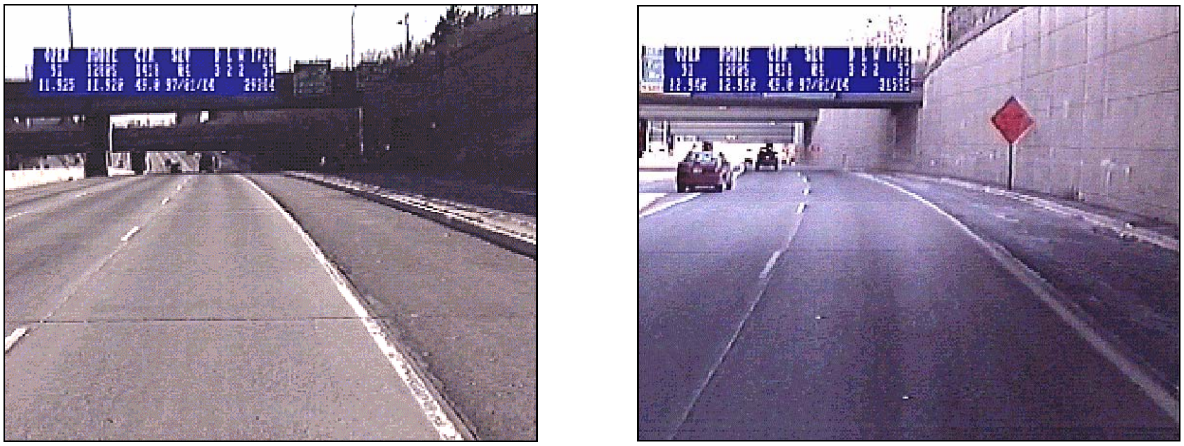


Figure 1. Video log information from the NJDOT ARAN van

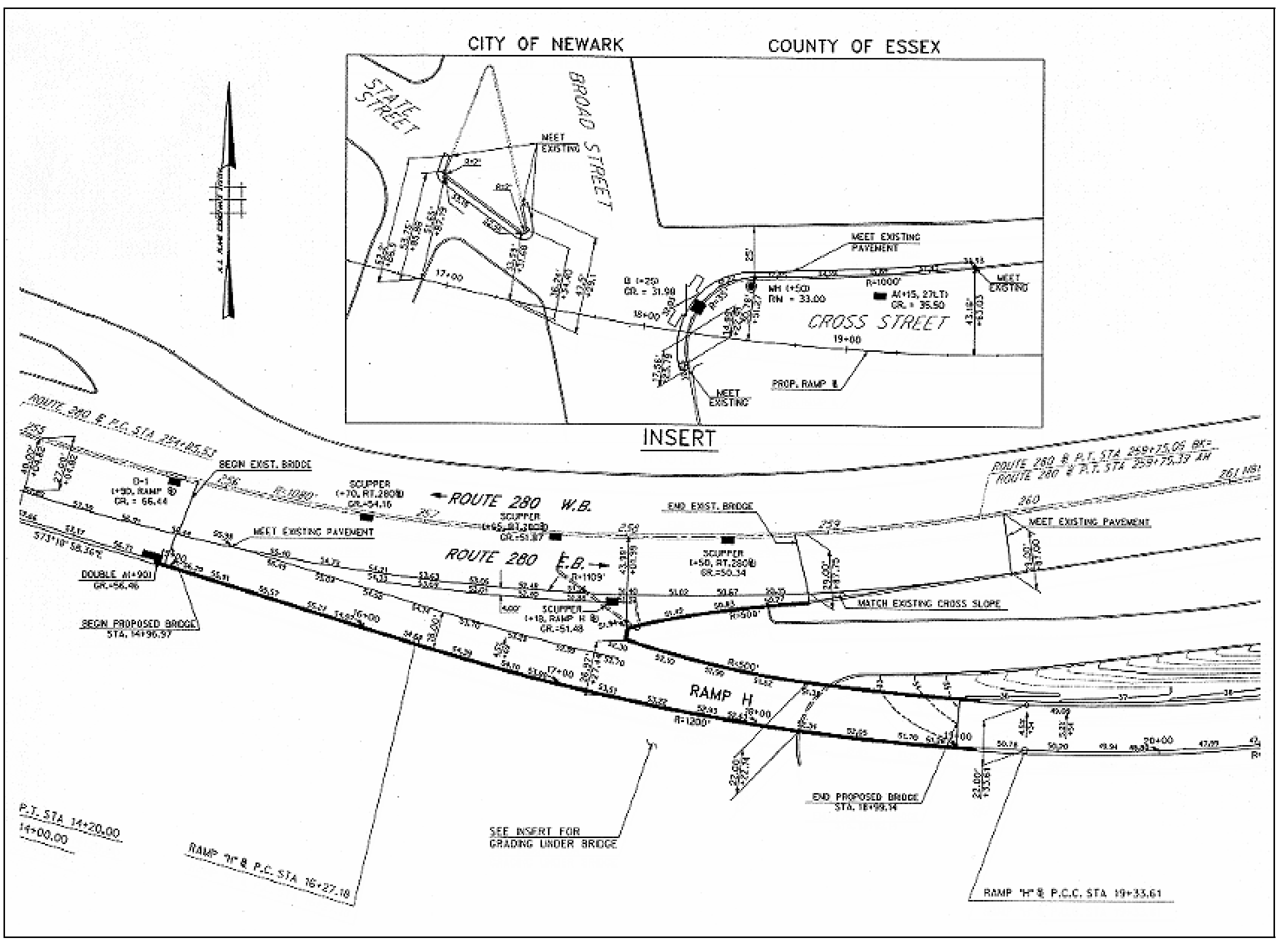


Figure 2. Construction plan for Rt. 21 ramp and associated portion of I-280.

Using the 3D reference line (constructed using GPS and other data) a complete 3D model of I-280 eastbound through Newark was generated. The total length of roadway modeled was approximately four miles. Barriers, bridges, and sign structures were also represented in the 3D model in addition to the roadway. This task took considerable effort since accurate modeling of the positions and size of these features was important in ensuring that the model could be used to evaluate aspects of the roadway such as sight-distances.

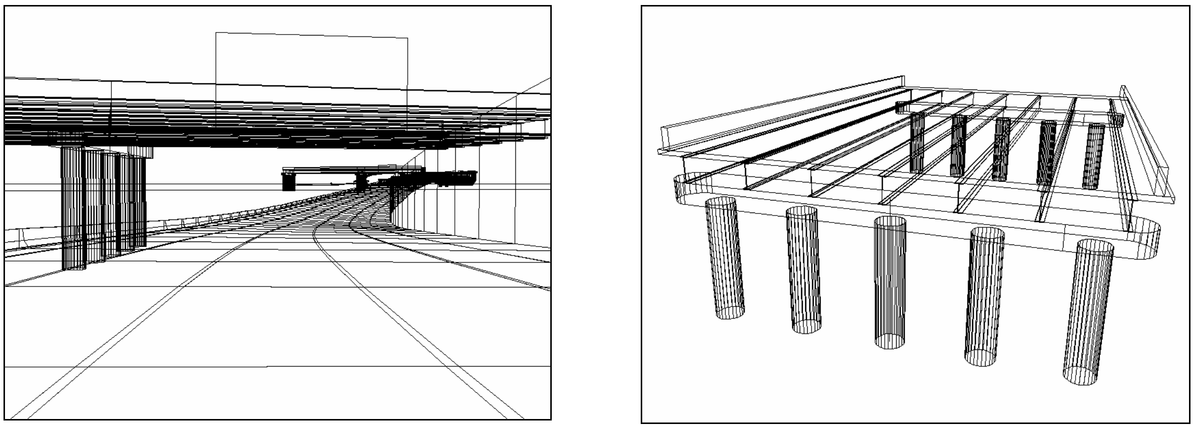


Figure 3. Wire-frame view of roadway, bridges, sign, and retaining wall.

Once the 3D geometrical roadway model, including bridges, barriers, walls, etc., was created (see Figure 3), the proposed signing was merged with the 3D roadway model. The sign faces were obtained from NJDOT in CAD (MicroStation) format and converted for use in the 4D drive-through visualization. Position and size data for the proposed signing were taken from the plans for the proposed signing. Rectangular sign objects, having the correct size, shape, and position relative to the roadway were merged with the 3D roadway model. These rectangular objects served as surfaces onto which the sign images were *texture mapped*—i.e. “digitally painted” (see Figure 4).

4D drive-through simulations were then created by defining viewer (driver) motion paths along the roadways in the 3D models and then moving the point of view of the driver along those motion paths in small increments. The entire 3D model was rendered at thousands of points along the motion path which resulted in thousands of “frame” images (similar to frames of a movie). The frame image files (see Figure 5) were then subsequently compiled and compressed into a digital movie format that can be viewed at the speed needed to correctly simulate a driver moving along a roadway at highway speeds.

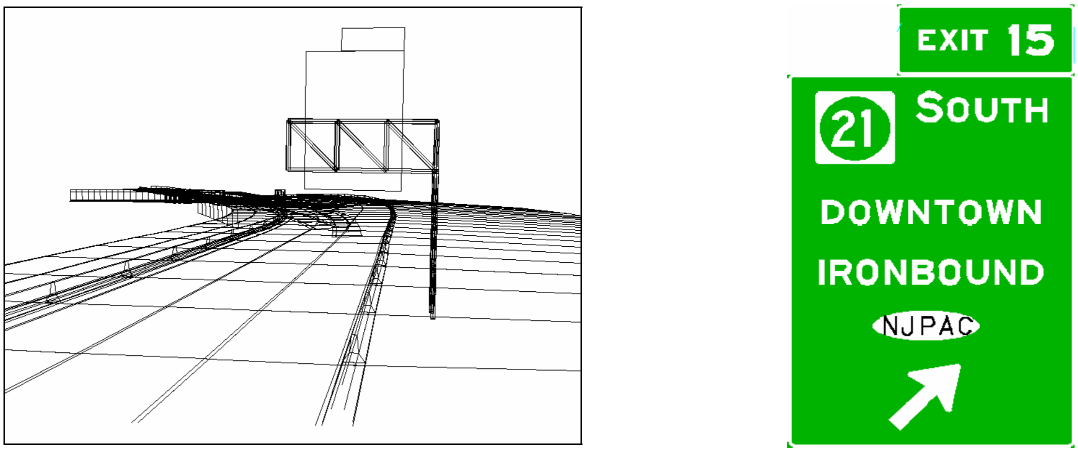


Figure 4. Wire-frame view and corresponding texture map of a sign.

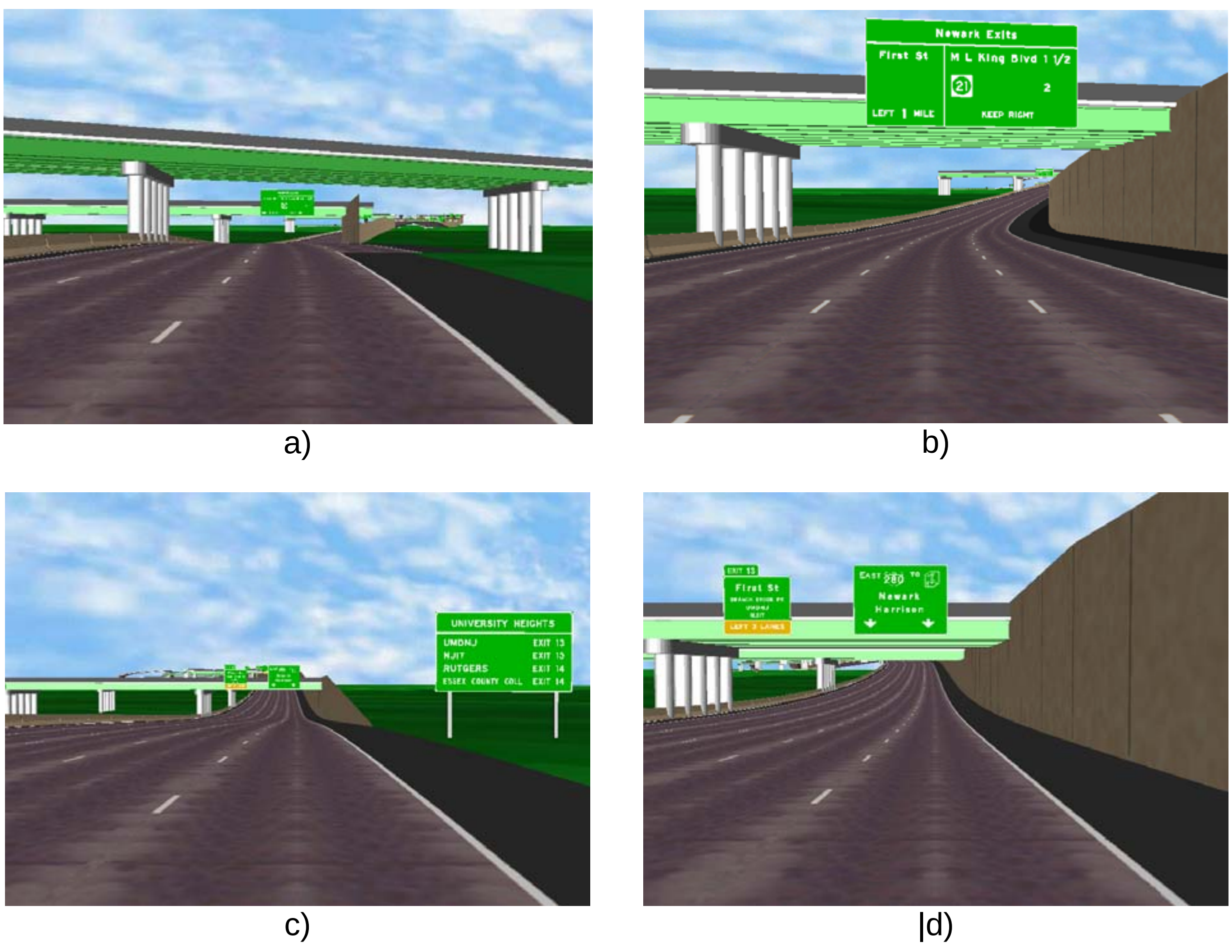


Figure 5. Movie frames representing motion along the roadway.

The drive through visualizations generated for the review of new signing plans for the Newark Performing Arts Center on I-280 and the Rt. 21 ramp visualization demonstrated how such a procedure can work in a time constrained NJDOT task force and public review environment. However, while the visualizations were found to be very useful to NJDOT, the need for better model building tools also became evident during this project. The process of converting image data—e.g. images of sign faces—was tedious and time consuming. In addition, it was concluded that it would be beneficial to also have the ability to import 3D geometric data for objects such as bridge components and sign structures, from CAD programs such as AutoCAD or MicroStation. In order to address these needs, the project being reported on herein was undertaken.

Project Goals

In order to make the process of building 3D models easier for purposes of visualizing highway signage projects, a new software system was developed. The primary goal in developing the new software was to make those portions of the model-building (or model-generation) process that were identified as most tedious in the I-280/Rt.21 project simpler and more manageable. The primary tasks identified as constituting the bulk of the effort involved in generating 3D highway models were:

- *Texture management* : Managing and converting image data into appropriate texture form for use in 3D model rendering.
- *Integrating 3D objects into the 3D model* : Importing 3D objects (such as bridge components, sign structures, and barriers) into the 3D model being generated.
- *Roadway model generation* : Processing highway alignment data (e.g. from the NJDOT ARAN van) for purposes of producing a 3D model of roadway components.
- *Rendering the 3D model* : Visualizing the proposed signage by rendering the entire 3D model from various driver points-of-view.

The newly developed software system, called VisView, is composed of three main components that aid the users in completing the tasks listed above. These components, which are described in more detail later in this report, are:

- *VisGen* : Visualization Generator which aids in management and manipulation of image/texture data and allows 3D objects to be imported from CAD programs and converted into VisRen (see below) formatted objects.
- *VisUtil* : A simple set of utilities that aid in generating 3D roadway model components from 3D alignment data.
- *VisRen* : Visualization rendering component that permits the user to render the model statically from a single point of view or dynamically (e.g. for a drive-through simulation) using real-time rendering on accelerated graphics video hardware.

The remainder of this report describes the development and use of the VisView system in more detail including individual descriptions of the three component programs listed above.

DESCRIPTION OF DATA TYPES

In order to perform a "virtual drive-through" of a proposed signage project, one must first construct a 3D model of the highway segment in question. Doing so involves the management, modification and use of several different types of data including image data, geometric object data, and roadway alignment data. The following sections describe in further detail the types of data that are compatible with the VisView system and how these various types of data can be merged with the assistance of the VisView software.

Image Data

Clearly, sign faces must be represented in the 3D model if the proposed signage is to be reviewed during the simulated drive-through. Methods of representing sign appearance range from vector-based methods to raster based methods. In vector-based methods, each geometric object in the sign face must be individually represented. For example, each stroke of each letter in the sign would need to be represented by either a straight line or curved line segment. In addition, information such as line color, background color, texture, etc. must also be represented. In contrast to this method, a raster-based representation consists of a large number of "dots" each having a single, unique color. The collection of all such dots forms an image of the same form as those routinely viewed using computer software such as web browsers and image editors.

For extremely simple sign faces, the vector-based format is more advantageous because very little data needs to be stored in order to represent the sign face. However, for complex signs, the advantages of vector-based storage are diminished. A large number of geometric objects (e.g. character strokes) must not only be stored but also individually rendered when the entire highway scene is rendered. During the past two decades, graphical images and icons have become a core component of modern computer operating systems, virtually all of which now provide a *graphical user interface*, or GUI, (e.g. the Microsoft "Windows" operating system). As a result of this saturation of image data, modern computer hardware and operating systems have been designed such that raster image data can be displayed with extremely high speed and efficiency. As a result, in the VisView system, raster based storage is used to represent image data such as sign faces and object textures.

A further advantage of using raster based image storage arises when it is desired that a 3D highway model be constructed that includes not only newly proposed signage but also *existing* signage. For the newly proposed signage, CAD drawings of the actual sign faces are likely available and can be incorporated into the 3D model in either raster format or vector format (if the vector data is available). However, for existing roadway signage, no such vector based CAD data is likely available. One option would be to redraw the signs that are currently installed in the field using a CAD program, but this is very time consuming and wasteful of resources. If sign faces are instead stored in raster format, digital photographs can be taken of the signage currently in the field and easily converted into a raster format that can be integrated into the 3D model. No redrawing of the data is necessary. Thus both proposed signs and existing signs could be easily included in the same 3D model through the use of raster image data storage (see Figure 6).

Similarly, image data representing the visual texture (appearance) of various surfaces in the 3D model can easily be accommodated using raster based image storage. For example, Figure 7 shows a rendering of the Rt. 21 ramp 3D model in which raster-based image data has been used to represent the texture of the roadway and the concrete retaining wall (as well as other objects). The roadway texture was generated from a photograph of the actual roadway in this area and the concrete panel texture was generated from a photograph taken at the ramp construction site.

a) Proposed sign face (from a CAD drawing)

b) Photo of existing sign as installed in the field

c) Cropped photo of existing sign

Figure 6. Use of raster image storage to represent both proposed signage and existing signage.

Due to the advantages of using raster based image data for storing and representing sign faces and object textures, this data format was adopted for all such uses in the VisView system. However, one must understand that there are still numerous raster image formats available for storing image data. Some of the most common image formats are JPG, TIFF, BMP, and PPM. In the VisView system the JPG and PPM formats have been adopted for two different purposes. The JPG image format is a highly efficient method of storing image data and very little storage space is needed to store relatively complex images. For this reason, JPG has become the single most widely used format used for digital cameras, web based image viewing, and related applications. As a result, a primary feature of the VisView system is that it allows very easy import of image data in JPG format. This import process is discussed in further detail later in this report.

While the JPG format is extremely efficient for purposes of storing and importing image data, it cannot be used internally by the 3D rendering software. Instead, the PPM portable pix-map (ppm) format is used for this purpose. In the I-280 and Rt. 21 visualization projects, it was found that the process of converting data from JPG format to PPM format was time consuming because not only did the images need to be converted, but they also had to be manually resized into standard “texture map” image sizes. Additionally, cropping information had be to specified for many of the images that were used to represent sign faces. To address this issue, features have been implemented in the VisView model generator program (called VisGen) to make these operations as easy as possible.

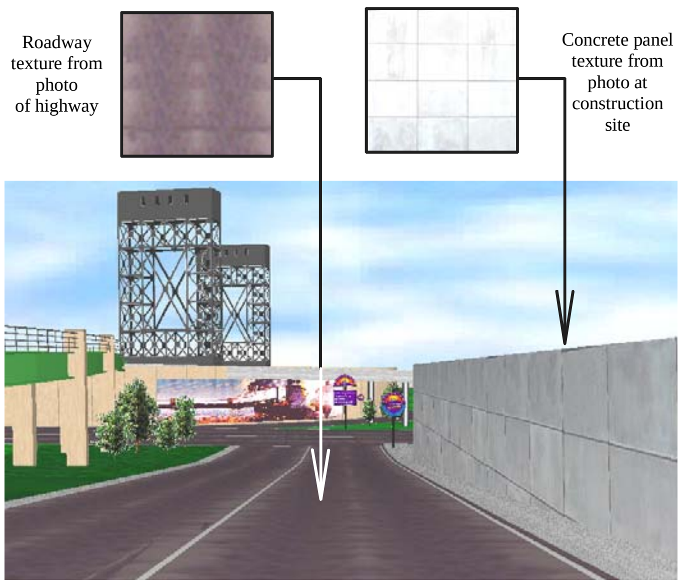


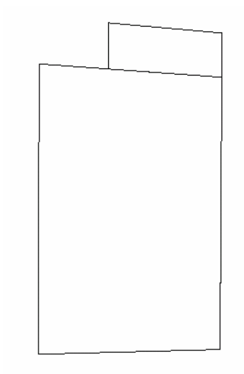
Figure 7. Photographic image data used to apply “texture” to objects in a 3D model.

Geometric Data

In addition to representing the visual surface appearance of objects in a 3D model (using image data to generate “textures” as was described above), the geometry or shape, of the objects must also be represented. Geometric data describes shape of objects in the model by breaking each object down into a finite number of flat surfaces. For example, consider the sign represented in Figure 8. In order to render the complete sign, both its geometry and appearance must be represented in the model. The geometry of this particular sign is modeled with two flat rectangles that represent the actual physical sizes of the two sign components (the main sign and the exit tab). The appearance of the sign is represented using image data. In order to represent the complete sign, the image data is “texture mapped” (or painted) onto the geometry.

In the VisView system, geometry may be described in a number of different ways. The VisRen sub-component (the rendering module) of the VisView system permits a variety of methods for describing geometric shapes. For example, arbitrarily shaped geometric “faces” can be used to model the shape of objects in the 3D model. Figure 9 illustrates some typical examples of simple geometric faces that may be used within the VisView system. Each face consists of a finite number of vertices (also called nodes) for which exact geometric positions in three-dimensional space have been established.

Each face (also called an “element face” because elements may have multiple faces in some situations) is rendered by connecting the vertices and subsequently “painting” the enclosed surface area with a color or texture. In Figure 10, we return to the previous example illustrating how geometry and texture are combined to render the final appearance of a 3D object. However, in this figure, the individual face edges and vertices are explicitly shown (normally these features would be hidden when creating a 3D rendering). It is clear from the figure, that two separate quadrilateral element faces have been used to render the overall sign object. Each of the quadrilateral faces has been textured (“painted”) with appropriate image data to produce the final rendering.

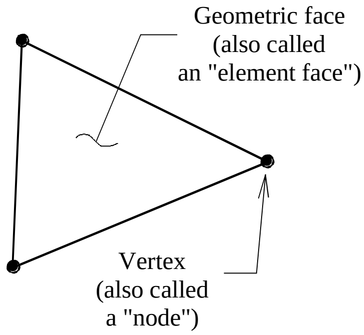


a) Geometric data representing shape of sign (perspective view)

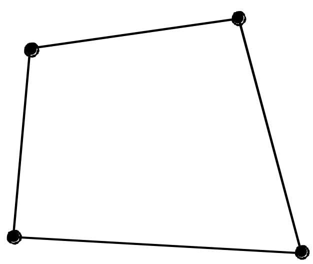
b) Image (texture) data representing sign appearance

c) Texture data applied to geometry to represent sign (perspective view)

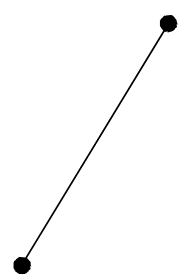
Figure 8. Combining geometric data and image data to render a 3D sign object.



a) Triangular face with three vertices



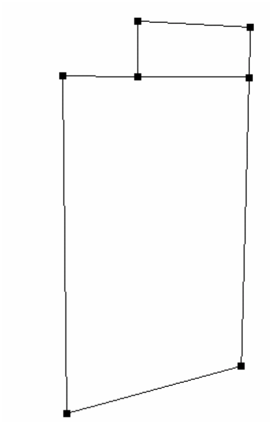
b) Quadrilateral face with four vertices



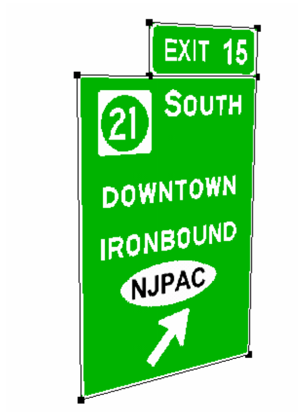
c) Line face with two vertices

Figure 9. Simple geometric faces that can be used to model complex 3D objects.

In the VisView system, the user may describe the shape of geometric objects by explicitly providing three dimensional coordinate data for the vertices and then stating the manner in which those vertices are to be connected to form faces. This process can be accomplished by providing the vertex and face data explicitly in the model description file used by the VisRen software. However, in addition, the VisRen software has a powerful *shape extrusion* feature that makes the process of describing the geometry of many standard shapes very simple. Figure 11 illustrates a typical application of cross-sectional extrusion. In order to build a model of a sign bridge structure, a simple square cross-sectional shape is defined. Each element in the model consists of two vertices which are connected by a single defined element face (i.e. the line connecting the two vertices). However, at each vertex, a cross-sectional shape is also applied to the element. By extruding this cross-sectional shape along the axis of the element, a three-dimensional tube (with a square cross-section) is automatically constructed (Figure 11-a) by the VisRen software. By applying this process repeatedly to each line element that is used to define the complete structure, a three-dimensional model of the sign bridge can be rapidly constructed (shown in a wire frame view in Figure 11-b). This same process can be used to model other regularly shaped objects such as concrete barriers, bridge piers, bridge girders, and poles supporting small roadside signs.



a) Quadrilateral geometric faces

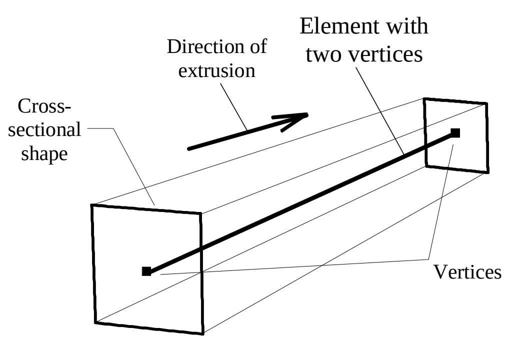


b) Rendered sign object (with face vertices and edges shown for emphasis)

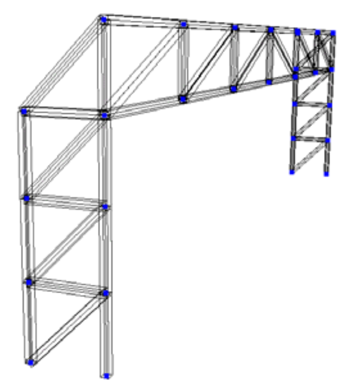
Figure 10. Use of two quadrilateral faces to render a sign object.

While the features provided in VisRen offer a variety of methods for describing geometric shapes, it was recognized at the outset of this project that having the ability to import data from other programs would also be highly beneficial. For example, a user might wish to model a 3D object such as a highway bridge or a sign structure in a CAD program such as AutoCAD or MicroStation. During the initial stages of the project, several options were explored with regard to importing geometric data from such programs into the VisView system. AutoCAD and Microstation both use unique—but different—methods for storing data in their native file formats. Since it was not feasible to implement import capabilities into VisView that would be capable of reading the native file formats for both of these programs, an alternate solution was sought.

AutoCAD and Microstation are both capable of exporting data from their native formats into the DXF (data exchange format) file format. Given this ability, it was decided that the VisView system would implement partial DXF import capabilities. The complete DXF standard is very complex and contains components that have no use in 3D modeling of highway signage projects. Therefore, import capabilities were added to VisView only for the most relevant components of the DXF standard. After examining DXF file format documentation and exporting selected 3D objects from both AutoCAD and MicroStation into DXF format, it was concluded that the DXF objects most relevant to 3D modeling of highway signage projects were the `3DFACE` and `POLYLINE` objects. Thus, features have been implemented in VisView which allow DXF `3DFACE` and `POLYLINE` objects to be imported into the system and then subsequently exported into the model-definition file format utilized by the VisRen software. Additional details regarding the DXF import capability are given later in this report.



a) A single two-vertex element rendered by extruding a defined cross-section along the axis of the element



b) 3D model of a sign bridge constructed using simple two-vertex elements with square cross-sections defined

Figure 11. Use of cross-sectional extrusion to model a sign bridge structure.

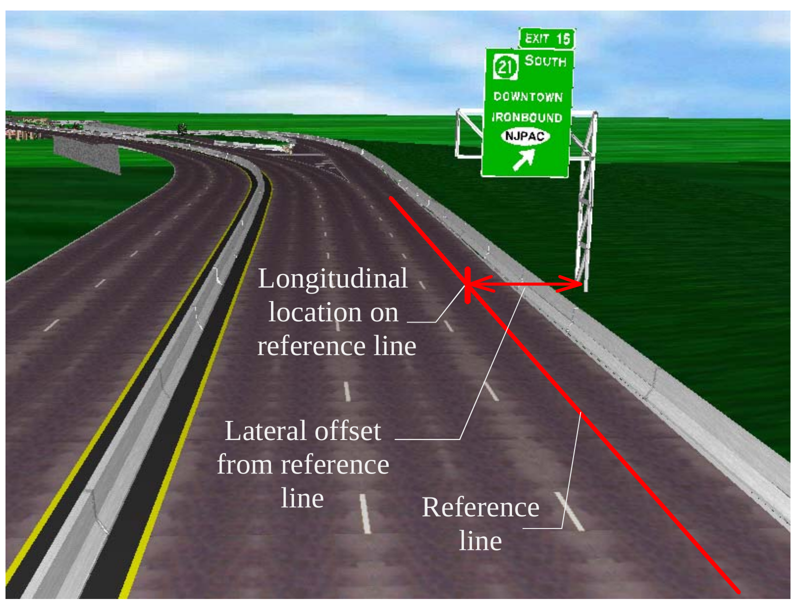
Roadway Alignment Data

Roadway alignment data is a specific type of geometric data that describes the shape of the roadway reference line. In general, it can be treated in precisely the same manner as the geometric data described in the previous section. However, there is one primary distinction which warrants giving this type of data special consideration. That distinction is the fact that roadway alignment data is often used to position (or locate in 3D space) *other* types of 3D objects that are contained within the overall roadway model. For example, consider the I-280 and Rt. 21 3D model renderings shown in Figure 12. The first part of the figure consists of an aerial view of the I-280 model. One can see clearly from this rendering that the only content present in the model consists of objects very close to the roadway itself. Objects farther away from the roadway do not need to be modeled for purposes of performing a simulated drive-through of the scene. The figure also alludes to the positioning method used to specify the locations of the bridge overpass structures. The exact positions of the bridges in general 3D space were never calculated for this model. Instead, their positions were specified as "milepost" and lateral offset positions relative to the roadway reference line.

The roadway reference line is generally taken to be the centerline of the rightmost lane on the roadway (this is not an absolute requirement, but rather a simple convention). The *geometry* of the reference line itself is generally determined from GPS data such as that produced by the NJDOT ARAN van mentioned earlier in this report. Using such GPS data sources, a reference line describing the roadway alignment is established. Once this task is completed, the alignment data (in the form of the reference line) can be used to position other 3D objects such as bridge and sign structures relative to the roadway. The alignment data can also be used to generate geometric faces that are used to represent the roadway surface.

Figure 12-b illustrates a typical use of roadway alignment data for the purpose of establishing the position of a roadside sign structure in 3D space. Rather than determining the exact location of the sign in space, its position has been established by specifying a longitudinal location (i.e., a "milepost" location) on the reference line and a lateral offset perpendicular to the reference line. This method of positioning roadside features *relative* to the reference line is much more convenient than having to establish *absolute* 3D coordinates for each object in the model. In the VisView system, utility software programs allow the user to work with alignment data in the ways just described.

a) Portion of I-280 model showing bridge overpass structures positioned relative to roadway



b) Portion of Rt. 21 ramp model illustrating how a sign structure has been positioned relative to the roadway alignment reference line

Figure 12. Using roadway alignment data to position roadside objects.

DESCRIPTION OF THE VISVIEW SOFTWARE SYSTEM

The primary goal in developing the VisView software has been to provide the user with a set of tools that can be used to manage 3D data resources, generate 3D model components, and perform virtual drive-through simulations using 3D rendering techniques. In order to accomplish this goal, the VisView system has been developed as three separate components: *VisGen*, *VisUtil*, and *VisRen*. Each of these programs will be described in detail later in this chapter. However, before doing so, an overview of the overall VisView system will be given. The purpose of this overview is to show the reader how each of the three VisView components are used to create complete 3D models and simulated drive-throughs.

System Overview

In the previous section, descriptions were given for the various types of data that are most often used in creating 3D highway signage models: image data, geometric data, and alignment data. In the VisView system, these data sources are managed and utilized by the VisGen and VisUtil components. The VisGen ("visualization generator") component assists the user in managing the first two types of data: image data and geometric object data. Using VisGen, the user can easily import, manage, and modify JPG images that will be used to represent sign faces and material textures in the 3D model. The VisGen component also permits the user to import 3D objects in DXF format from CAD programs such as AutoCAD and MicroStation. Subsequently, the program can be used to export both image data (in PPM format) and geometric data (in VisRen format) so that simulated drive-throughs of the model can be performed using VisRen.

The VisUtil (visualization utilities) component of the VisView system is a collection of "utility" programs that can be used to work with roadway alignment data. For example, the VisUtil software can be used to generate "meshes" of element faces that represent the roadway surface. In addition, coordinates for 3D objects positioned relative to the roadway can be computed using the VisUtil software. Animation paths for virtual drive-throughs of the highway model can be easily generated using the VisUtil programs once the user has established correct roadway alignment data.

Once all of the components of the 3D model have been created using VisGen and VisUtil, the VisRen (visualization rendering" component of the VisView system can be used to render the model. The model can be rendered, or visualized, using a variety of different rendering techniques : wire frame, simple surface rendering, surface rendering with color, surface rendering painted with texture (image) data, etc. In addition, VisRen can be used to perform virtual drive-throughs of 3D models along driver-motion paths previously established using VisUtil.

Figure 13 graphically summarizes the VisView system description that has been given above. The figure illustrates the various roles that the VisView components play in constructing 3D drive-through visualizations and the flow of data through the model-building and visualization processes.

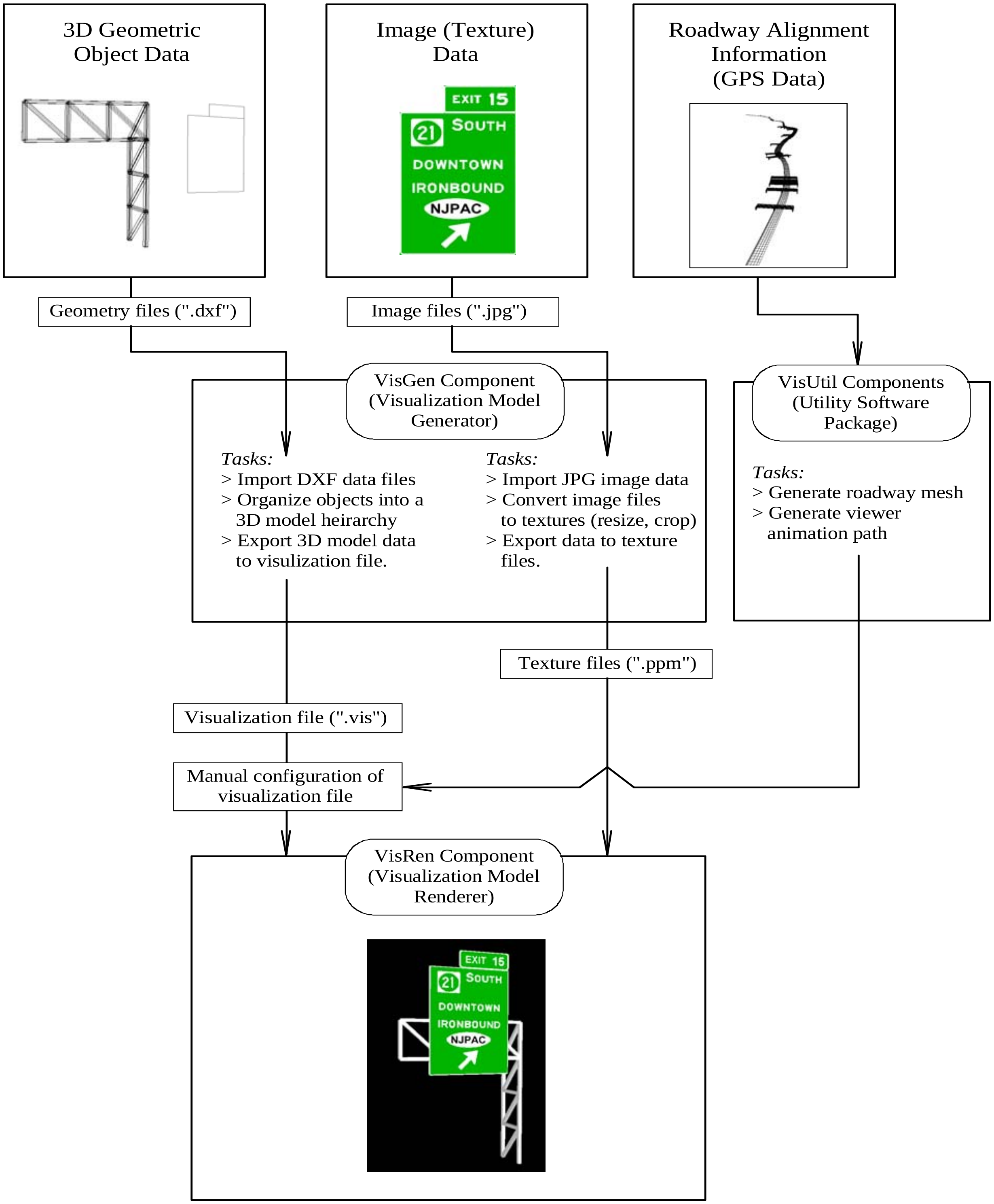


Figure 13. Relationship between data and VisView software system components

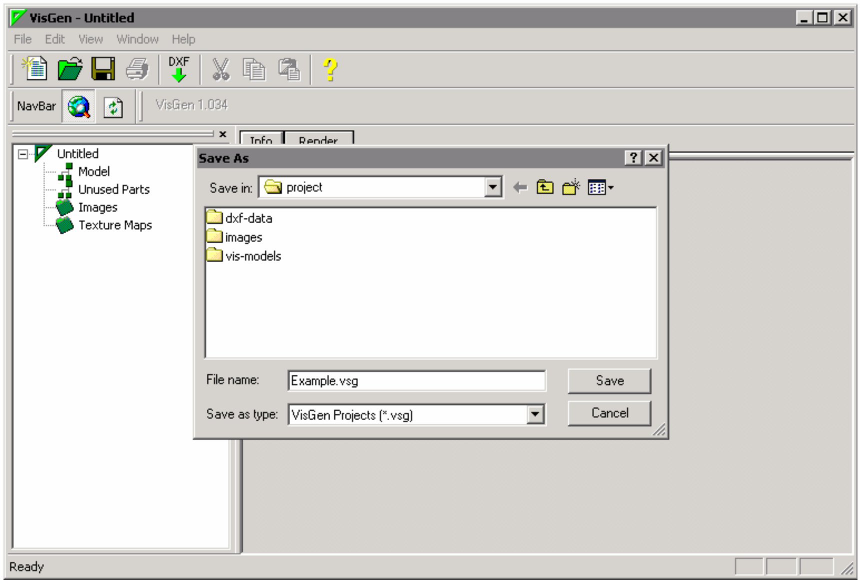
Visualization Generator Component (VisGen)

The VisGen component of the VisView system aids the user in importing, managing, manipulating, and exporting image data and DXF geometry data. In particular, the VisGen program can help the user perform the following tasks:

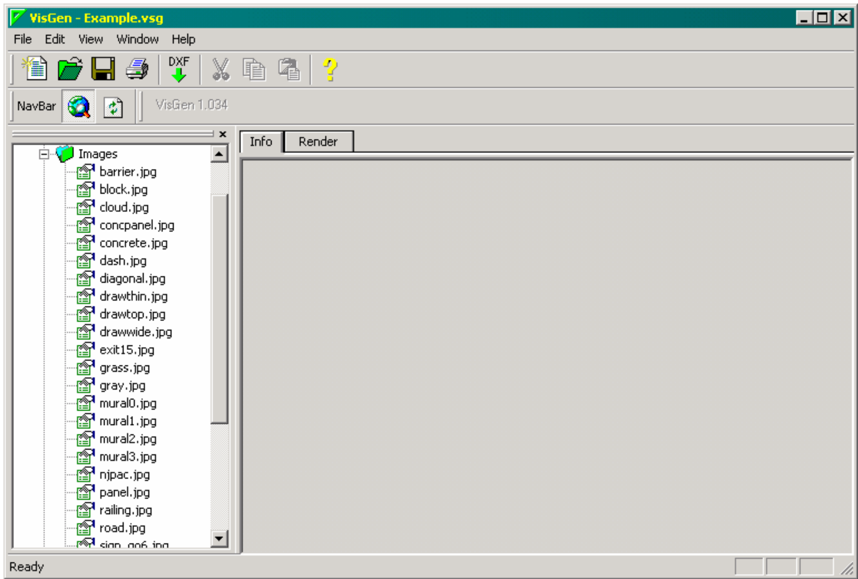
- Import image data stored in JPG format
- Preview imported image data
- Set cropping regions for image data
- Resize imported JPG data to sizes appropriate for use in texture mapping
- Export texture data to PPM files that can be processed by VisRen
- Import geometric DXF objects of type `3DFACE` and `POLYLINE`
- Export geometric data in VisRen compatible format

In general, the user should be able to perform all necessary image manipulation and management from within the VisGen program. Additionally, DXF data can be imported in VisGen and then exported in VisRen format (and later manually edited by the user if additional modifications are deemed necessary). In order to demonstrate the features and use of the VisGen program, a short tutorial is given below in which the reader is "walked through" the most common features of VisGen.

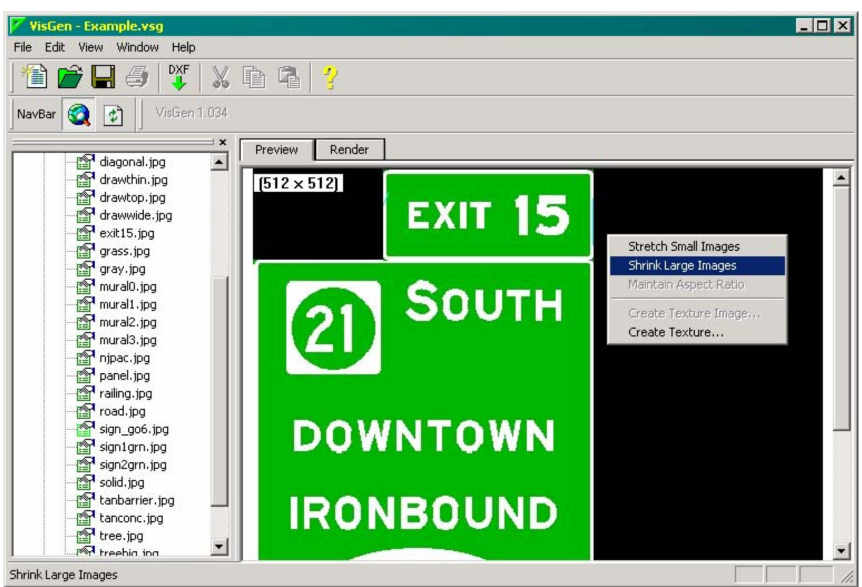
When the program is initially started, the program does not have an active project file loaded. To begin a ***new*** project, select File>Save (or click the File Save icon on the program's toolbar), browse the file system to the desired location, and save the project file. VisGen project files have a ".vsg" extension. In the example below, a project file called "example.vsg" is created in a directory called "project". Note that all image data is assumed to be stored in the "images" subdirectory of the directory that contains the project file.



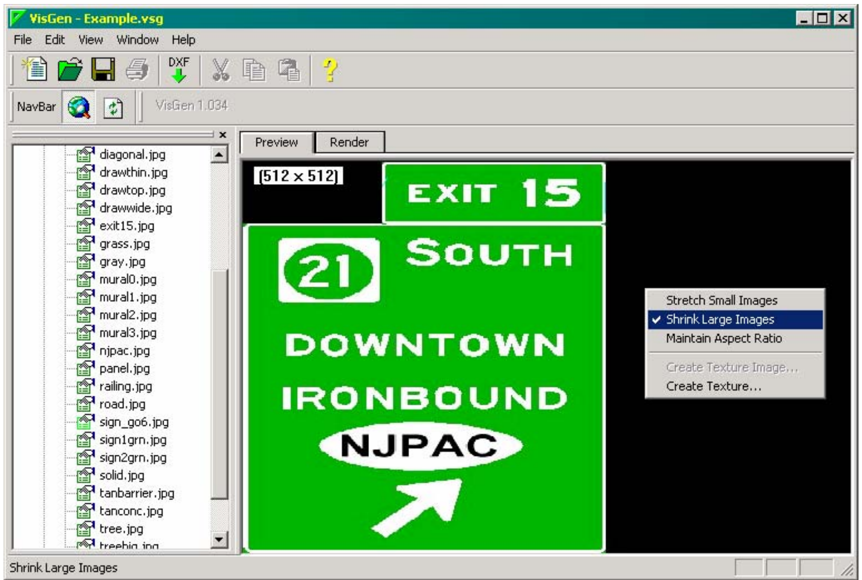
Once the project file has been created, VisGen will search the directory containing the project file for a directory called "images". If found, VisGen will load all of the image data contained in this directory. Note that the VisGen window contains a toolbar at top, a project navigation (browser) bar at the left (a tree view), and an information panel at the right side of the window. To view a list of the files present in the "images" directory, click the "+" symbol on the images entry of the project browser. In the screen image shown below, all of the images available to the project are listed.



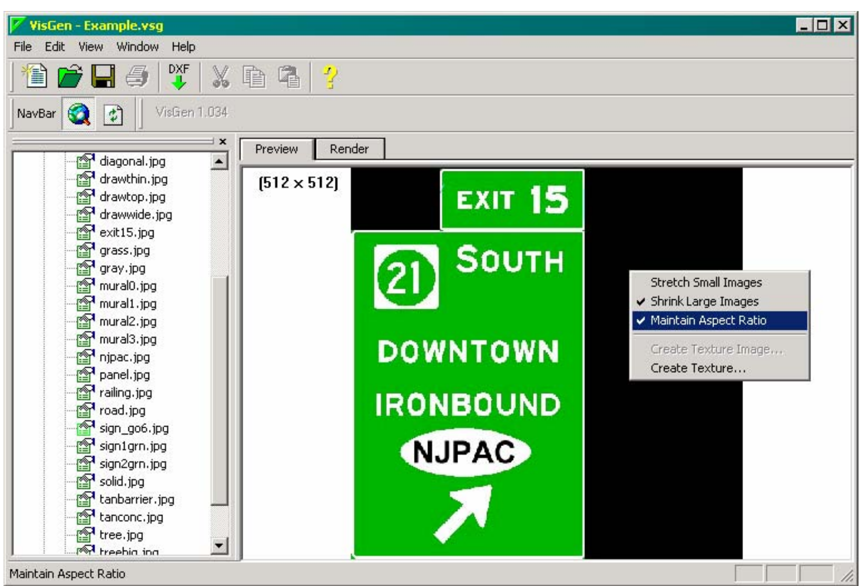
To preview a particular image, click on the image name in the image list section of the project browser toolbar. In the case shown below, the file "`sign_go6.jpg`" has been selected. Once selected, the image data will be displayed in the "preview" window. Because the image in this example is relatively large, it exceeds the size of the preview window. A scroll bar appears next to the preview image so that the entire image can scrolled vertically.



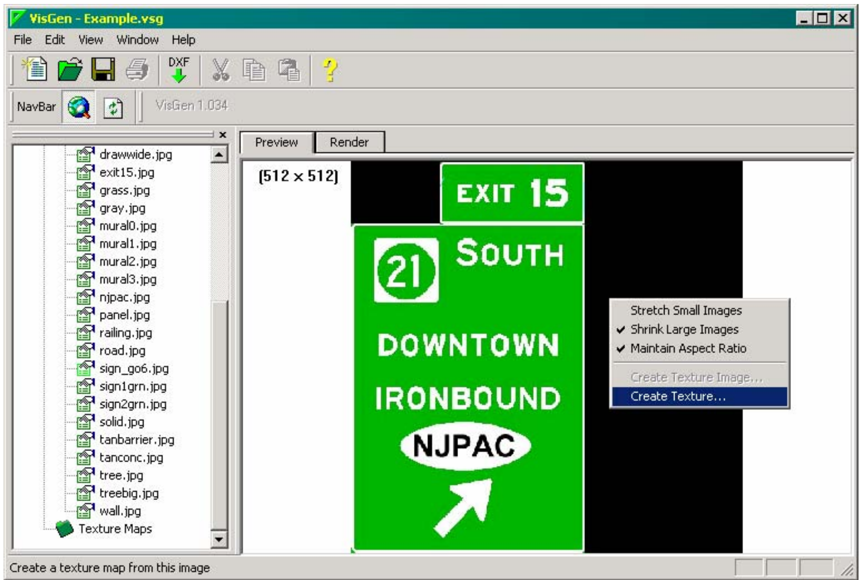
As an alternative to using the preview scroll bar, the user may also choose to shrink the image to fit the screen. Right-clicking in the preview window launches a preview window “options” dialog. Selecting “shrink large images” will permit the entire image to be previewed without the need for scrolling.



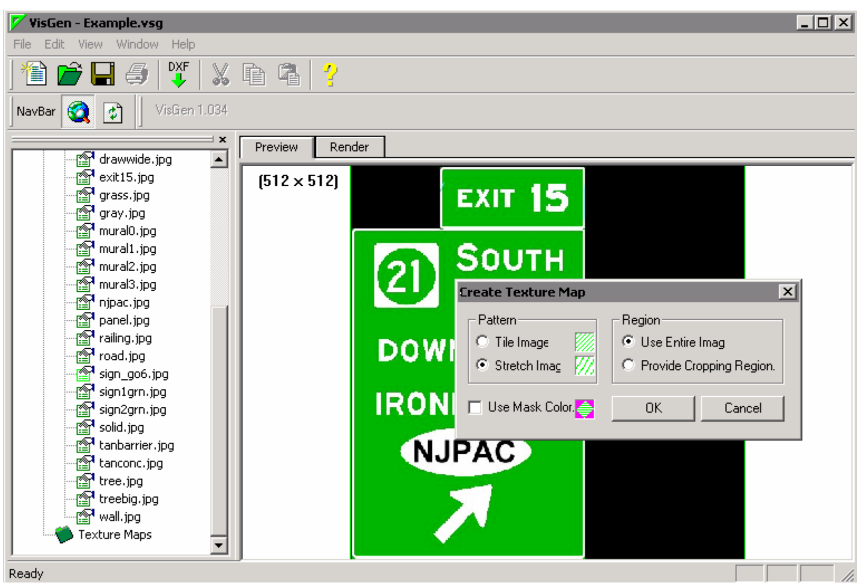
When large images are shrunk (or small images are expanded) to fit the available window size, they may be distorted (as is the case in the image shown above). To correct this problem, the user may select “maintain aspect ratio” on the options dialog of the preview window. Doing so will permit the entire image to be previewed but without the distortion previously shown. With this option now activated, the preview window now appears as shown below.



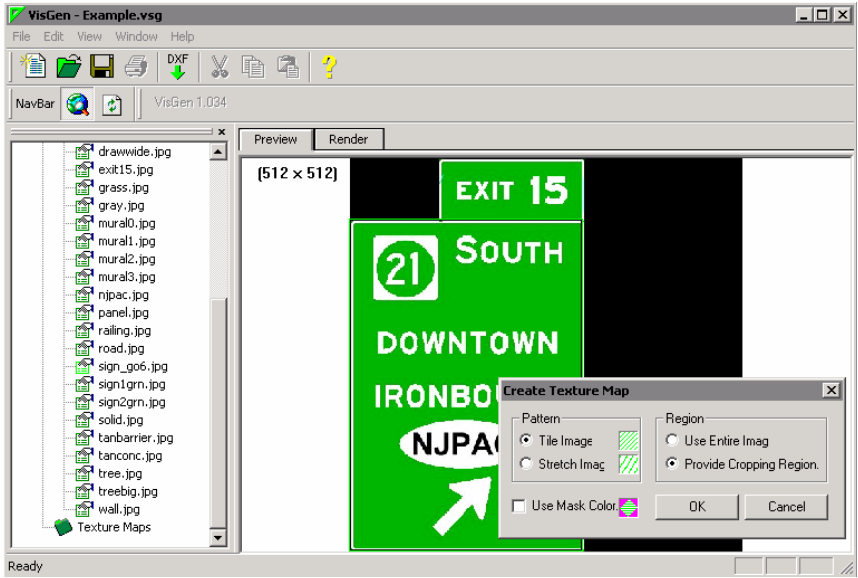
Now that the image data for the sign is more appropriately shown, we will create a texture object using a cropping region. "Texture" data is image data that is stored in a specific format that can be used by VisRen when rendering the 3D model. To create a texture object from image data, right click in the preview window and select "create texture".



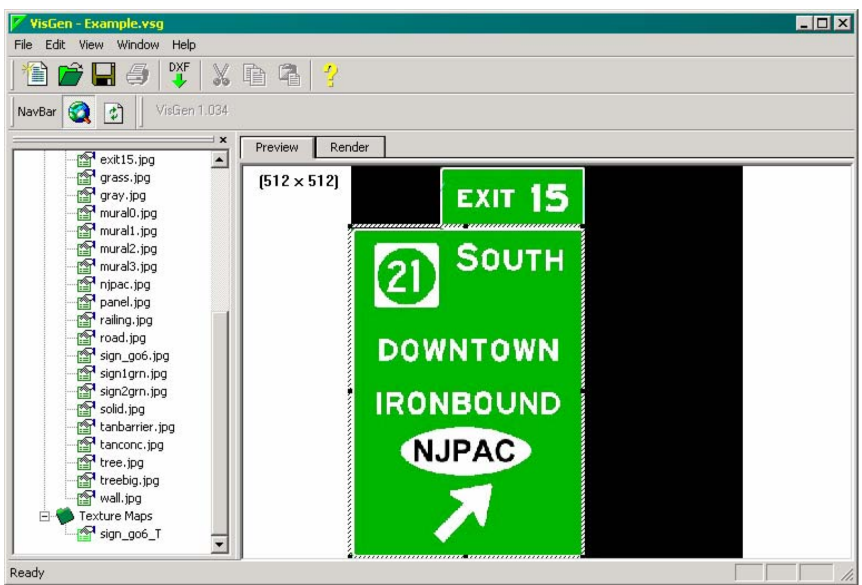
Once "create texture" is selected, the texture map creation dialog will be launched (see below). In this example, rather than simply creating a texture map corresponding to the entire image by pressing OK, we will create a texture map for a specific portion of the image data. This will be accomplished by selecting a cropping region. In the texture creation dialog, click "provide cropping region". Next, drag the mouse to create a "box" (cropping boundary) around the desired region.



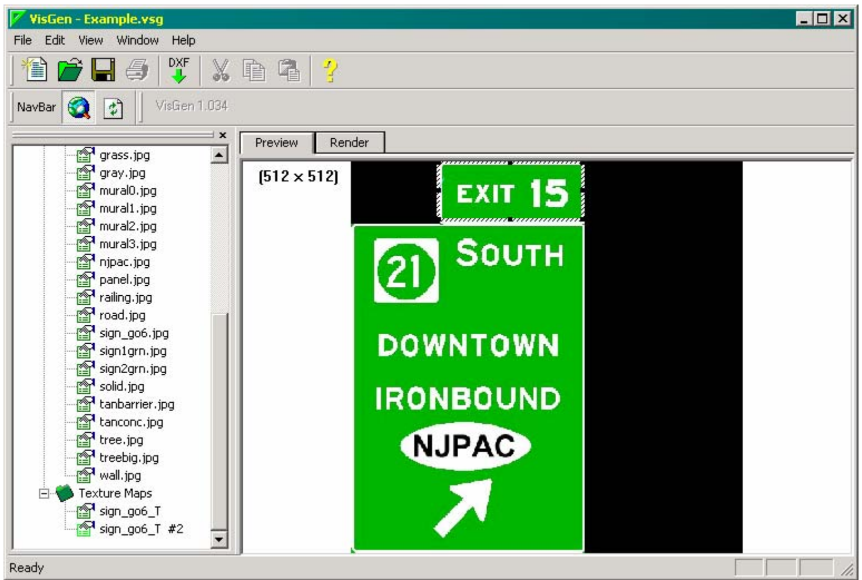
Once the cropping region has been selected, the texture creation dialog will once again be shown. At this point, press OK to create a texture object from the cropped part of the image data.



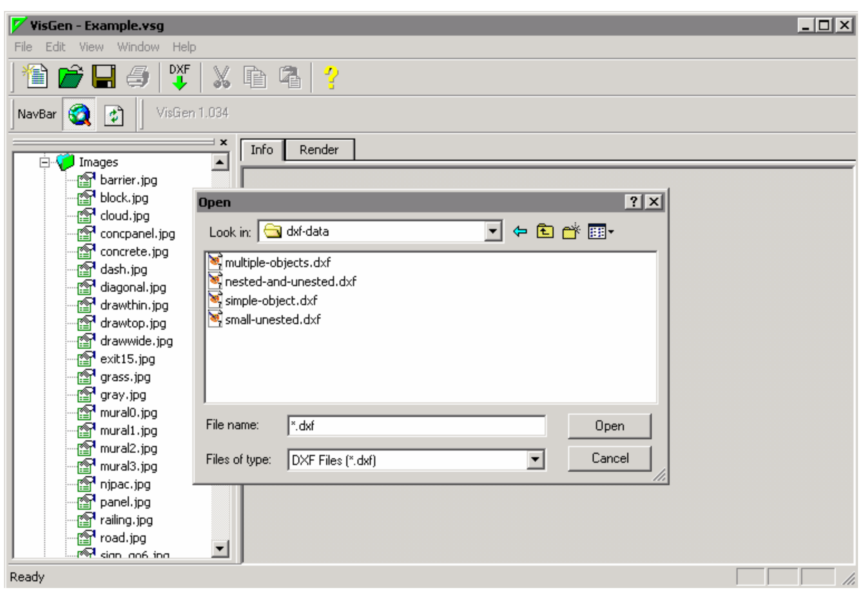
After pressing OK, VisGen will create a texture object corresponding to the cropped image region. In addition, a new entry, shown as "`sign_go6_T`" (where T=texture) is added to the "texture maps" section of the project browser window (at left in figure shown below). Image data in the "texture maps" section of this tree is now in the proper format for use in texturing (or painting) geometric objects in VisRen.



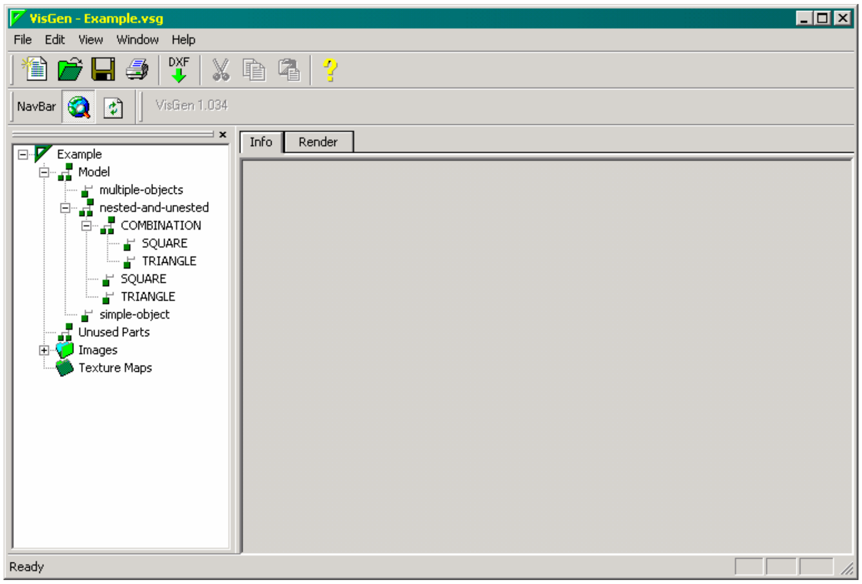
To further illustrate the creation of textures and the use of cropping regions, we continue with the same image. As shown below, the same image can be selected once again, and a different cropping region specified (in this case the region around the “exit 15” portion of the sign). When the new texture object is created, it is named “`sign_go6_T #2`” as is shown in the “texture maps” portion of the project browser.



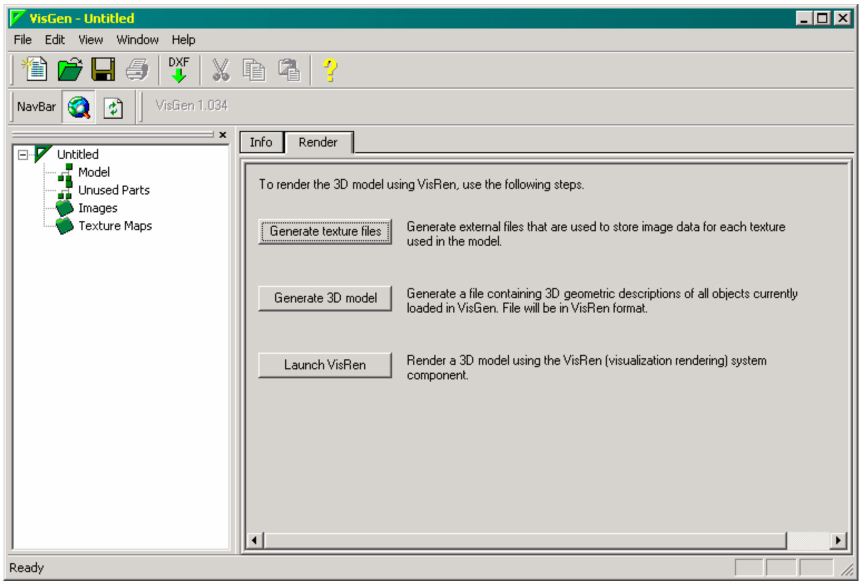
Next, the DXF import capabilities will be demonstrated. To import geometric data from a DXF file, press the “DXF” button on the VisGen toolbar. A “File>Open” dialog will appear which will allow the user to browse the file system to the desired location. Once the correct directory is located, select the “.dxf” that is to be imported and press “Open”.



After importing several of the DXF files shown in the File>Open dialog above, the VisGen project browser can be used to determine the organization of the data that has been imported. All DXF data is shown under the “model” portion of the project browser window. Nested DXF data is shown using a tree structure (in the same manner as the directory structure of a file system that is normally shown in the Windows Operating System). In the example below, we see several DXF objects that have been imported.



Once image data and geometric DXF data have been loaded, VisGen can be used to create the external files needed by VisRen to render the 3D model. Clicking on the “render” tab of the VisGen main window displays the following screen.



On this screen, the user can perform three different tasks:

- Create external texture map files (PPM data files that will be read by VisRen)
- Create portions of a 3D model by converting imported DXF data into VisRen compatible data and writing that data to a VisRen file (.vis).
- Launch VisRen to render a 3D model

Visualization Utility Component (VisUtil)

The VisUtil component of the VisView system consists of several simple command line driven utility programs. Since these are not graphical Windows programs, screen images will not be shown. The VisUtil programs can be used to perform the following tasks:

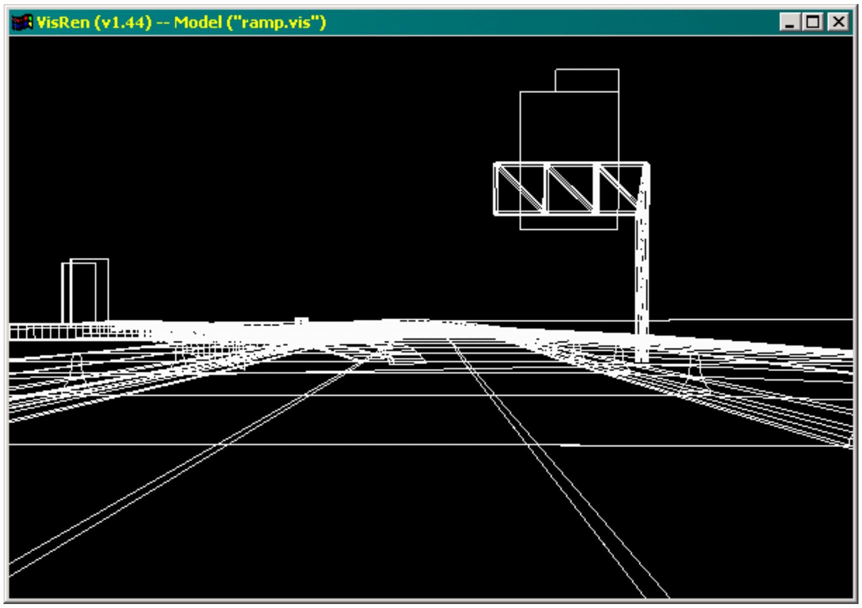
- Generate “meshes” of element faces that represent the roadway surface.
- Compute coordinates for 3D objects positioned relative to the roadway
- Generate animation paths for virtual drive-throughs of the highway model using roadway reference lines constructed from GPS data

The use of the VisUtil programs is not a required step in the generation of 3D highway models. In many instances, however, the use of these programs can be helpful to the user. Program usage information can be determined from the utility program source code.

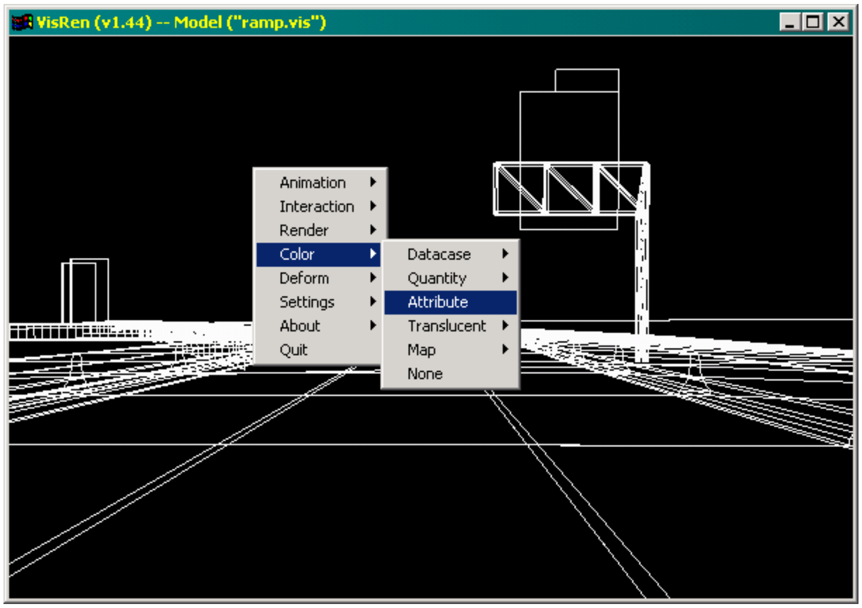
Visualization Rendering Component (VisRen)

Once a complete 3D model has been constructed and written to a VisRen “model definition” file, the VisRen component of the VisView system can be used to render the highway model and perform virtual drive-through simulations. The following section provides a brief tutorial on the use of the VisRen program. The 3D model of the Rt. 21 ramp project is used as an example model for purposes of discussing the features of VisRen.

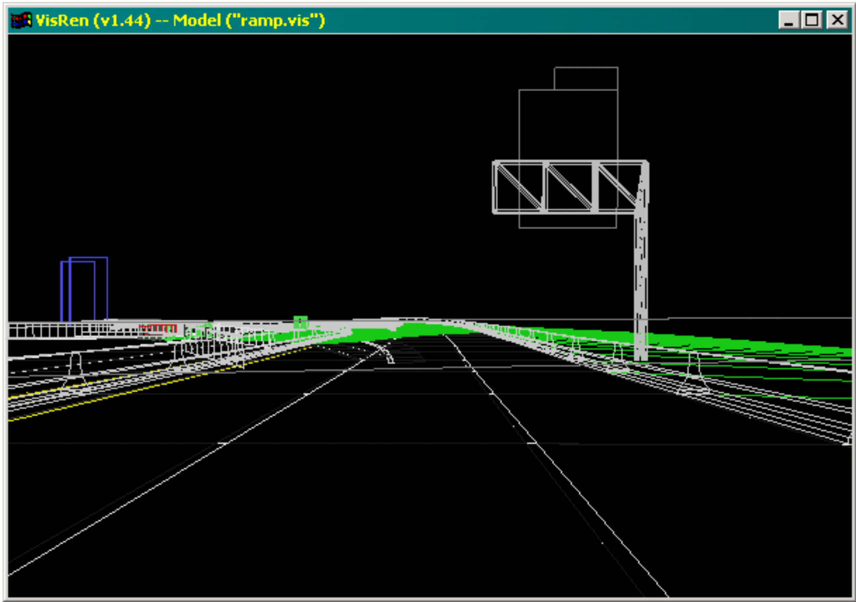
When a 3D model is initially loaded, VisRen scans the model definition file for “viewer motion path” data. Such a path would be used, for example, to describe the motion of a viewer performing a virtual drive-through of the model. If such path data is found, it is processed and the viewer is placed at the beginning of the motion path (as shown below). The “rendering mode” is set to “wire frame” initially.



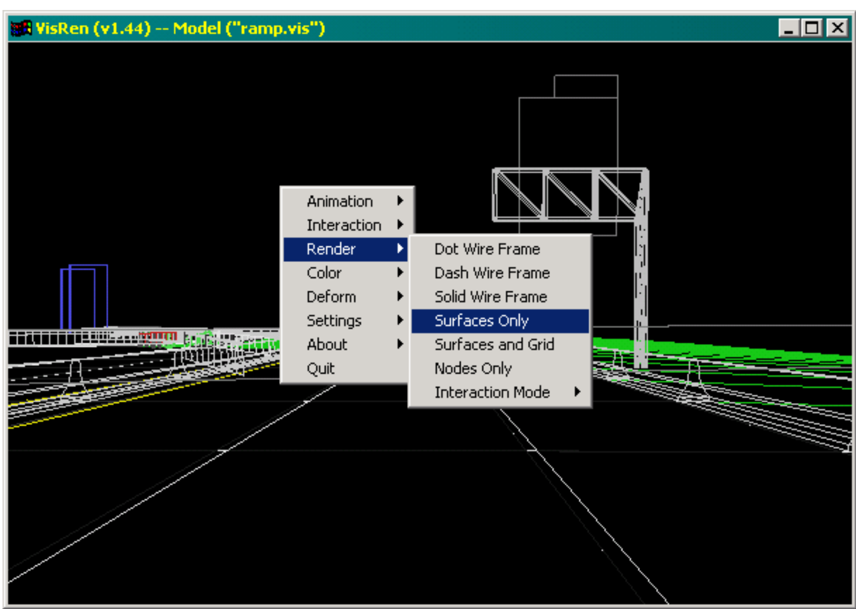
To render the model with a level of detail sufficient to review a proposed highway signage project, we must modify the rendering settings. The VisRen command menu is a pop-up menu system that is activated by right clicking in the rendering window. In the image shown below, the user has selected the “color” sub-menu on the main VisRen menu and has further selected “attribute” as the type of data to use for coloring objects in the model. Attribute data is loaded from the VisRen model definition file (files having an extension of “.vis”).



After selecting “attribute” based coloring, the model is automatically re-rendered, again using a wire frame view, but now with color. The resulting rendered image appears as shown below.

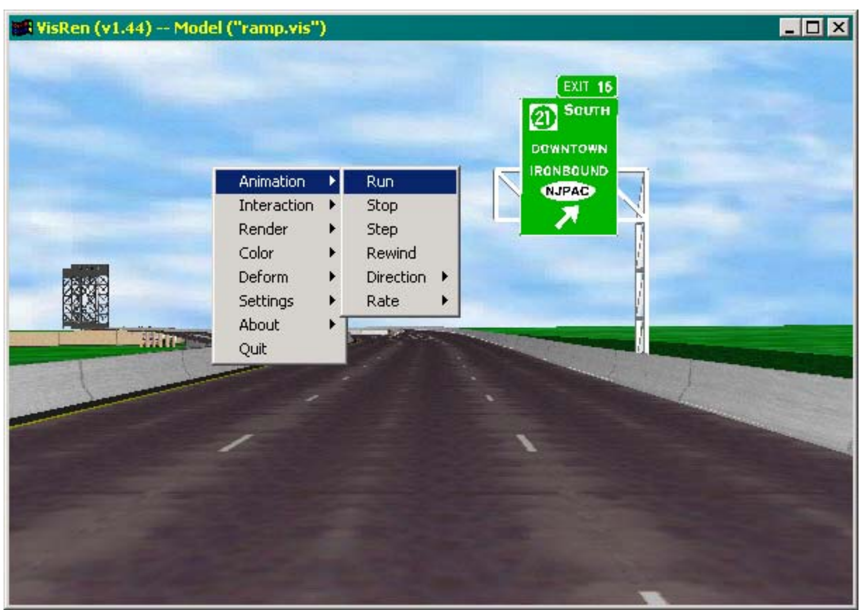


Next, the user needs to modify the “rendering mode” that is used to render the model, i.e. visualize the appearance and shapes in the model. From the main menu, the “render” sub-menu has been selected (see below). Next, the user selects “surfaces only” as the rendering mode to make active. This option will render the entire model as a series of painted surfaces but will *not* show the edge lines of element faces. If it is desired that the edge lines be shown (a rare case), the user may instead select the “surfaces and grid” rendering technique from the rendering menu.



Once the “surfaces only” rendering mode has been selected, the 3D model will be immediately re-rendered as shown below. The entire model is rendered by “painting” texture data (image data) onto surfaces that represent the shape of all objects in the model. This is the rendering condition most appropriate for reviewing proposed highway signage projects.

When operating in the “drive-through” interaction mode (which is the default condition when animation path data is detected in the model definition file by VisRen), simply pressing and holding the left mouse button will “drive” the user through the model. The speed at which the viewer moves through the model can be set on the “animation“ sub-menu of the VisRen main menu under the heading “rate”. The data in the “rate” sub-menu is obtained by VisRen from the model definition file.



In order to obtain realistic speed of motion through the 3D model, it is *strongly* suggested that the user make use of accelerated video hardware. All of the 3D rendering tasks performed by VisRen utilize the OpenGL 3D graphical rendering language. Numerous video cards currently available on the market have partial or full OpenGL hardware acceleration available at reasonable prices. VisRen is completely capable of rendering 3D models on non-accelerated video hardware, however, virtual drive-through animations will be very slow. Even in such cases, geometric problems such as excessive sign density and sign obstructions can be detected. However, making use of moderately priced accelerated graphics hardware will greatly enhance the ability of the user to perform real-time virtual drive-through simulations.

CONCLUSION

A new software system has been developed to support the process of reviewing proposed highway signage projects using computer rendering techniques. The system provides the user with the ability to import and manage image data and geometric data, generate 3D model components, and render the proposed highway signage project either statically (as a still image) or dynamically (as an animated virtual drive-through of the highway model).

REFERENCES

1. Consolazio, G.R. (1998), “4D Drive-Through Visualization of I-280 For Review of Proposed Signing”, Final Research Report, Center for Advanced Infrastructure Technology, Rutgers University, New Jersey, Submitted to New Jersey DOT, 1998.