Southwest Region University Transportation Center

Automated Generation of Virtual Scenarios in Driving Simulator from Highway Design Data

SWUTC/10/476660-00018-1

Texas Transportation Institute
Texas A&M University System
College Station, Texas 77843-3135

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
SWUTC/10/476660-00018-1		

4. Title and Subtitle	5. Report Date
AUTOMATED GENERATION OF VIRTUAL SCENARIOS IN DRIVING SIMULATOR FROM HIGHWAY DESIGN DATA	September 2010
	6. Performing Organization Code
7. Author(s) Xi Zhao, Alicia Nelson, Susan Chrysler, and Yunlong Zhang	8. Performing Organization Report No. Report 476660-00018-1
9. Performing Organization Name and Address Texas Transportation Institute The Texas A&M University System College Station, Texas 77843-3135	10. Work Unit No. (TRAIS) 11. Contract or Grant No. DTRT07-G-0006
12. Sponsoring Agency Name and Address Southwest Region University Transportation Center Texas Transportation Institute The Texas A&M University System College Station, Texas 77843-3135	13. Type of Report and Period Covered Final Project Report September 2009–August 2010 14. Sponsoring Agency Code

15. Supplementary Notes

Supported by a grant from the U.S. Department of Transportation, University Transportation Centers Program

16. Abstract

In 2008, the Texas Transportation Institute (TTI) began using a desktop driving simulator made by Realtime Technologies, Inc. This system comes with a library of different roadway segment types that can be pieced together to create driving scenarios. The current project explored ways to create custom roadway segments for use in the TTI driving simulator. The project initially attempted to use AutoCAD® Civil 3D® roadway drawings and export them for use in the simulator. This attempt was not successful because all roadway geometric design features could not be exported. Next, the project used commercially available three-dimensional modeling software (Road Tools by Presagis, Inc.) and was able to create new roadway segments that were drivable in the simulator. These new roadway segments are not as precise as those created in Civil 3D®, but for most driving behavior studies conducted in the TTI simulator, researchers judged them to be sufficient.

17. Key Words	18. Distribution Statement
Driver Distraction, Vehicle Entertainment Systems, Instrumented Vehicle Research	No restrictions. This document is available to the public through NTIS: National Technical Information Service Alexandria, Virginia 22312 http://www.ntis.gov

19. Security Classif.(of this report)	20. Security Classif.(of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	62	

Form DOT F 1700.7 (8-72) **Reproduction of completed page authorized**

AUTOMATED GENERATION OF VIRTUAL SCENARIOS IN DRIVING SIMULATOR FROM HIGHWAY DESIGN DATA

by

Xi Zhao
Graduate Student
Department of Civil Engineering
Texas A&M University

Alicia Nelson
Associate Research Specialist
Center for Transportation Safety
Texas Transportation Institute

Susan Chrysler, Ph.D.
Center for Transportation Safety
Senior Research Scientist
Texas Transportation Institute

And

Yunlong Zhang, Ph.D.
Assistant Professor
Department of Civil Engineering
Texas A&M University

Report SWUTC/10/476660-00018-1
Project 476660-00018
Project Title: Automated Generation of Virtual Scenarios in Driving Simulator from Highway Design Data

Southwest Region University Transportation Center
Texas Transportation Institute
The Texas A&M University System
College Station, Texas 77843-3135

September 2010

DISCLAIMER

ACKNOWLEDGMENTS

The authors recognize that support for this research was provided by a grant from the U.S. Department of Transportation University Transportation Centers Program to the Southwest Region University Transportation Center.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

Over the past decade, the Texas Transportation Institute (TTI) has operated two different driving simulators and has primarily used them for evaluations of drivers' reactions to traffic control devices. Other uses have included visualization of pedestrian facilities by landscape architects and studies of driver distraction. A driving simulator can be a safe and inexpensive alternative to on-road studies. Additionally, a simulator can condense the amount of time needed to see all the treatment combinations under study. Also from a feasibility standpoint, conducting a study with many different control variables can be impractical to implement and evaluate on open roadways. Simulator studies allow the research team to test a wide variety of drivers in controlled settings, and depending on the portability of the simulation, a broad demographic of participants can be studied. Traffic engineers often complain about the accuracy and realism of the roadways depicted in simulators. Indeed, since most simulators are used for human factors research, the system developers have not had to worry about the absolute precision of the roads; they simply have to look enough like a road to prompt normal driving.

TTI has elected to use commercially produced driving simulator systems because the institute does not have the computer programming expertise to develop and maintain its own system. One disadvantage to using a commercially produced simulator is that users are limited to the existing library of roadway segments provided by the vendor. The new simulator, purchased by TTI in 2008, has the capability to import user-created roadway segments. The current project explored methods to create these roadway segments using actual roadway design data and engineering drawings.

Two methods were identified. The first used a computer-aided design (CAD) package to specify a particular road in terms of cross section, superelevation, curvature, and other roadway design parameters. The package used was AutoCAD® Civil 3D®, which is used widely throughout the roadway design community and accepts inputs in standard engineering units. This software was written specifically for transportation engineers and uses terms and dimensions familiar to them. This package can export files in a particular format that should be readable by the driving simulator software. The project was successful in getting the files to be read, but several key design components were lost in translation. The superelevation and curvature specifications made in Civil 3D® did not survive the export/import process. When these problems became evident, the initial goal of automating the process of converting roadway design files to those usable by the simulator was abandoned.

The second method identified utilized a commercially available, three-dimensional modeling software called Road Tools by Presagis, Inc. (formerly Multi-Gen, Inc.). This software is aimed at those who use simulation for visualization and animation, rather than roadway designers and builders. The current project found that Road Tools can export a drivable file to the simulator system. Further, the new roadway segment (also called "tiles") created in Road Tools can be used in conjunction with existing segments in the simulator system's library. The creation of these tiles does offer a wider selection of tiles for the TTI library and the capability to customize tiles in the future.

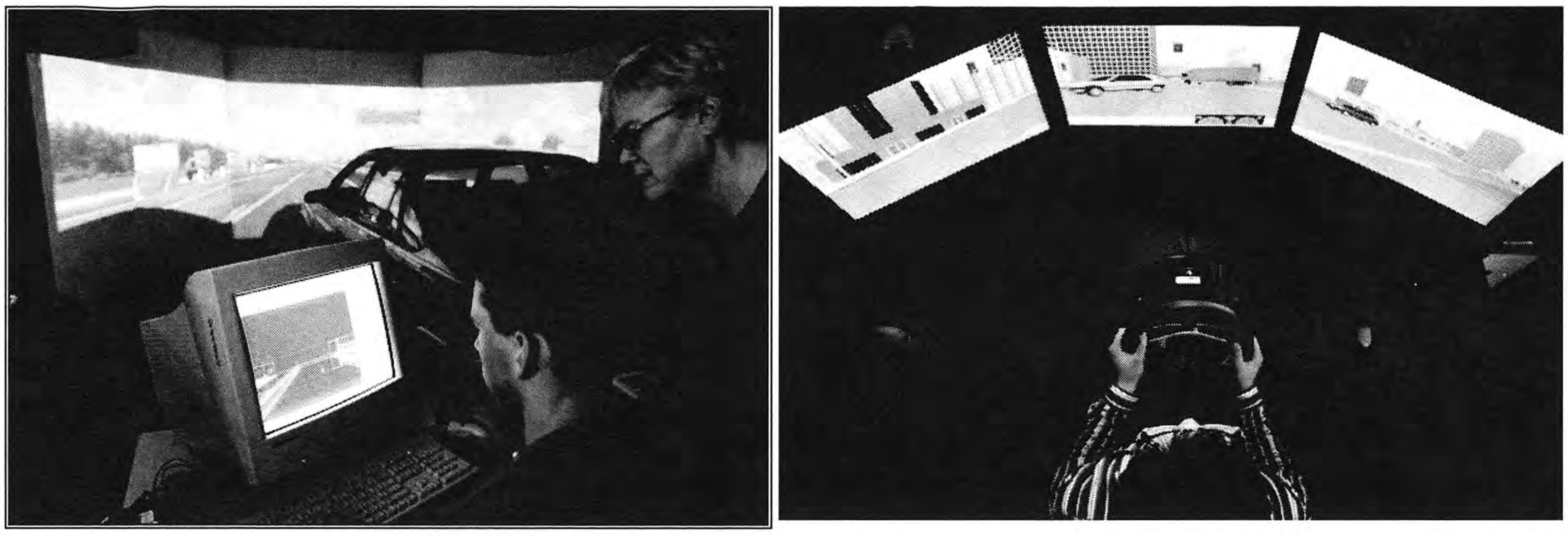
The methodology developed suits TTI's needs and the level of programming skill of its staff. Since most of TTI's work in the simulator is focused on driving behavior, the minor errors in the roadway tiles are not critical in these types of experiments. The simulator, however, can also be used to provide visualization of new roadway designs. For these types of applications, it is more critical that the exact geometry be rendered accurately. The next step in this research is to continue working on the first method mentioned above, even though it is doubtful that the expected resulting model is suitable for specific topics studied using driving simulation. Developing the algorithm and application to automatically model roadways based on design data in AutoCAD® Civil 3D® is meaningful for roadway design evaluation and visualization.

INTRODUCTION

Transportation human factors research often consists of testing drivers' reactions to new and innovative traffic control devices, roadway designs, or traffic operations. Simulation can also be used to test driver distractions and the effectiveness of new technology. Before devices are placed on the roadway, or new technology is placed in the cars or hands of drivers, researchers want to ensure there are no negative consequences to the use of these devices and that they provide a benefit compared to existing conditions. A driving simulator can be a safe and inexpensive alternative to on-road studies. Additionally, a simulator can condense the amount of time needed to see all the treatment combinations under study. Also, from a feasibility standpoint, conducting a study with many different control variables can become impractical to implement and evaluate on open roadways. Simulator studies allow the research team to test a wide variety of drivers in controlled settings, and depending on the portability of the simulation, a broad demographic of participants can be studied.

TTI'S SIMULATORS

Over the past decade, the Texas Transportation Institute (TTI) has operated two different driving simulators. The first was a full-size, fixed-based simulator produced by DriveSafety™ as shown in Figure 1a. The authoring software used to create custom driving environments was HyperDrive™. The newest and current simulator is a Realtime Technologies, Inc., three-screen desktop system shown in Figure 1b. Both authoring systems contain a large library of roadway segments and entities, or objects, such as signs and trees to create the driving environment. The DriveSafety™ system did not allow for the in-house creation of custom roadways and entities, while the newer Realtime system does allow custom tile and entity creation through the use of third-party software.



(a) Full-size simulator (b) Desktop simulator

Figure 1. TTI's two driving simulators: the full-size DriveSafety™ system and the desktop Realtime Technologies, Inc., system.

As TTI's simulator technology and knowledge have evolved, the two simulators have become a valuable place for work-zone research applications. In the beginning researchers were limited to only the items and features pre-existing in the software, resorting to much creativity in accomplishing what needed to be done. With newer software technology, researchers are now able to build much more customized scenarios and objects. In order to utilize this capability, expertise is needed in three-dimensional (3D) modeling and programming, which TTI has not previously had in-house.

For this project, TTI hired a graduate student with the necessary background to explore the flexibility of the new software in the Realtime Technologies, Inc., driving simulator. This report discusses the software and its capabilities, the methodology for creating new tiles, or roadways, and limitations and lessons learned throughout the process.

LITERATURE REVIEW

In the past, driving simulators have been used to study driving behavior, human factors, traffic accidents, and roadway design. Few transportation studies have focused on the computer science aspects of scenario generation or driving simulation itself because these topics are related to vehicle dynamics and image generation. Transportation researchers have been content to work within the confines of the library of roadways and entities provided by driving simulator vendors.

In TTI, a number of studies were completed in the past involving one of TTI's driving simulators. The Southwest Region University Transportation Center (SWUTC) project in 2004 evaluated the effectiveness of different display methods to assess traffic sign comprehension *(1)*. The full-size driving simulator allowed accurate measurements of driving performance, but the study showed that performance on sign comprehension was similar to that measured using simpler computer-based testing. Another SWUTC project in 2006 demonstrated the usefulness of a driving simulator in evaluating geometric designs for two-lane roads *(2)*. A Texas Department of Transportation (TxDOT) project in 2005 used the full-size driving simulator to study advance notification messages and use of sequential portable changeable message signs in work zones *(3)*. Also in 2007, a TxDOT project used the driving simulator to study how to improve temporary traffic control at urban freeway interchanges and pavement marking material selection in work zones *(4)*.

The University of Central Florida (UCF) and the University of Iowa have each developed their own driving simulator. A four-year project was initiated in 2000 to apply autonomous vehicles in UCF's driving simulator for real-time interaction with the simulator vehicle *(5)*. Using a commercial DriveSafety™ simulator at the University of Calgary, Laberge et al. studied the effects of passenger and cellular-phone conversations on driver distraction in 2004, arranging drivers to drive through residential and urban traffic environments in a fixed-based driving simulator where a variety of events occurred *(6)*. Andersen et al. studied the car-following model with a commercial driving simulator in 2004 *(7)*. Jenkins and Rilett used the TTI DriveSafety™ driving simulator with a microscopic simulation program and studied the application of simulated passing behavior in 2004 *(8)*. In 2007, the Western Transportation Institute (WTI) used a Realtime Technologies system with custom road tiles to simulate approximately 22 miles of US 191 on a driving simulator in order to develop and refine safety countermeasures for the roadway *(9)*. Olstam and Lundren developed a framework method to simulate realistic interactions between vehicles in 2008 *(10)*. Cha et al. presented a hybrid driving simulator with dynamics- and data-driven motion *(11, 12)*. Driving simulation was used in the project to evaluate warnings to prevent right-angle crashes at signalized intersections in 2008 *(13)*. Yan et al. used surrogate safety to validate a driving simulator in 2008 *(14)*. Dutta et al. used the driving simulator to evaluate and optimize factors affecting the understandability of variable message signs in 2004 *(15)*.

TTI'S SOFTWARE

REALTIME SIMCREATOR®

SimCreator®, a software tool provided by Realtime Technologies, Inc., is a graphical, real-time simulation and modeling system. It is the foundation on which most products of Realtime Technologies, Inc., are based, including the authoring package SimVista™, which will be discussed in the this section. As the core and fundamental software in the desktop driving simulation system, SimCreator® is currently used to run TTI's desktop simulator.

SimCreator® provides the environment to develop distributed simulation models conveniently and rapidly. Instead of spending time writing code, researchers can achieve the simulation with an intuitive graphical user interface that allows placement and connection of various components to build models, as shown in Figure 2. Each component, or square shown in the figure, can either be a group made up of components or a C/C++ code component. Once a model is developed, SimCreator® generates the code and simulates the model; meanwhile, the driving performance measures are outputted. A model in SimCreator® can be nested as a group component into other models and as an alias component that maintains its link to the original model.

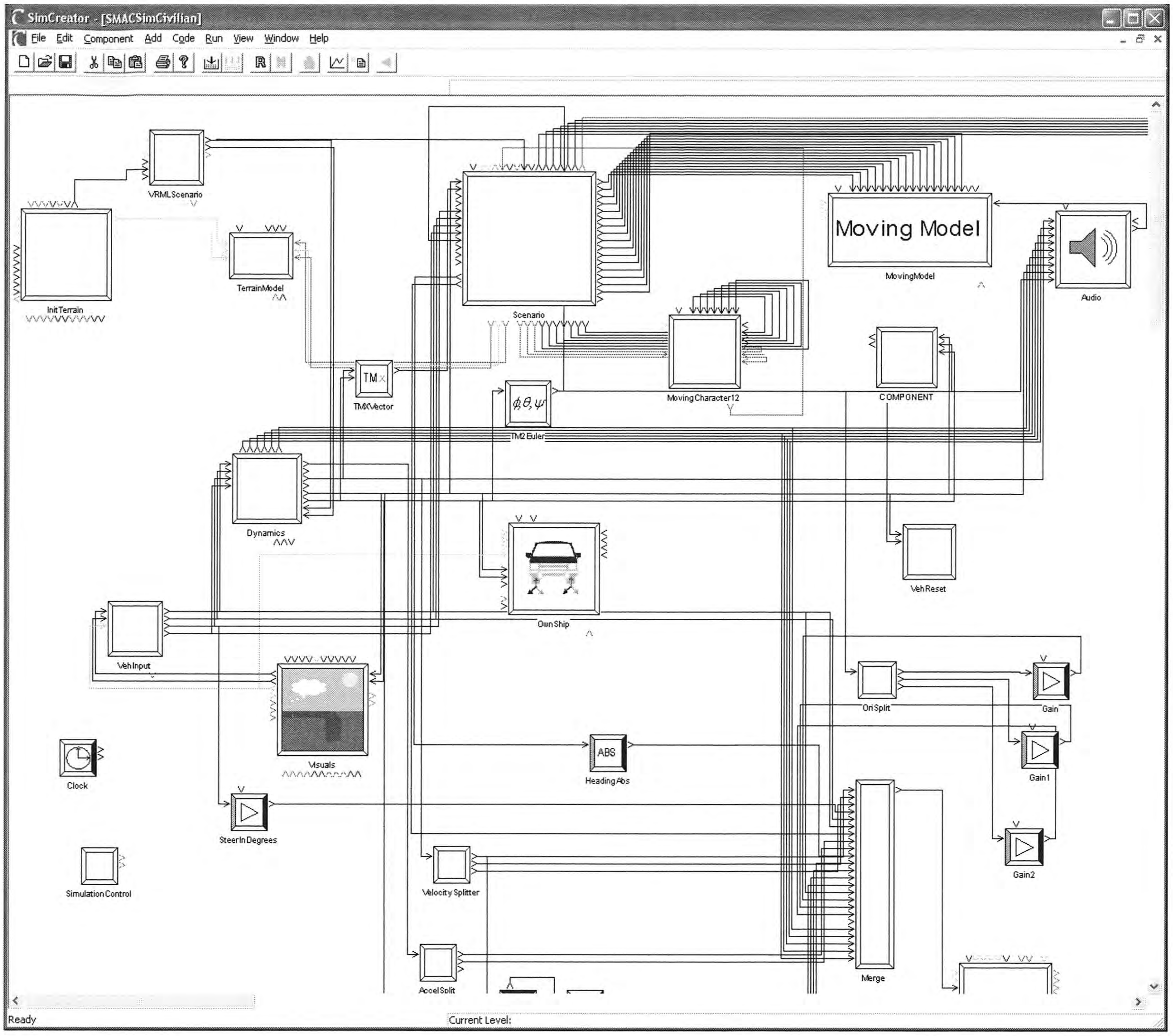


Figure 2. Realtime SimCreator® simulation and modeling system.

This project focused on the desktop driving simulation system from Realtime Technologies, Inc., which TTI utilizes for its advanced features. Researchers can rapidly develop distributed real-time simulations and have SimCreator® generate code to execute on embedded systems as well as distributed and multiprocessor systems. The model built graphically in SimCreator® is as efficient as manually coded models but requires less programming skill and time from researchers. The model can be executed on a special interpreter without compiling, or it can be compiled and then executed. For further development and advanced modeling application, the model built in SimCreator® can integrate an external application programmable interface (API) of other software and customized code.

REALTIME SIMVISTA™

SimVista™ is a tile-based scene and scenario authoring system provided by Realtime Technologies, Inc., and allows researchers to conveniently and quickly author customized virtual environments including traffic and pedestrians. It offers powerful tools to create comprehensive and complex scenarios for simulation because the Simulator Monitor and Control (SMAC) model is designed to support users with limited technical skills as well as more advanced users to create complex driving simulation scenarios for research and training.

SimVista™ provides a graphical user interface that allows researchers to drag and drop objects, including roadway "tiles" or road segments, into a virtual world and control them with scripting and parameters. With this concept, the user constructs driving environments by combining the tiles together like pieces of a puzzle, as shown in Figure 3. Tiles are dragged into the virtual space on a grid from a selection palette. Additional features such as supplemental trees, signage, buildings, etc. can also be added by dragging them from the objects palette. The resulting environment can either be published to the simulator on which it will be executed, or it can be republished back to the authoring tool as a new tile that can be included in the Object Gallery and can be used to create larger scenarios or driving environments. Once the static driving environment including the roadway network and visual features is finished, dynamic elements such as traffic signal controls, pedestrians, and vehicles will bring the scenario to life. These objects are dragged and dropped into the scene from the palette in the same way that tiles were added to create the scene. A set of special objects, which are called scenario control objects, controls the behaviors of the vehicles, pedestrians, and other controllable features during simulation. Scenario control objects include sensors, paths, markers, start and stop points, and similar scripting tools. Some of these objects come with scripts attached to them that allow the designer to issue commands to the simulator to control vehicles and other dynamic objects. The scripting language is a JavaScript-based language with customized commands created for simulation control functions. A combination of these elements can be used to evaluate where vehicles are and how the driver behaves.

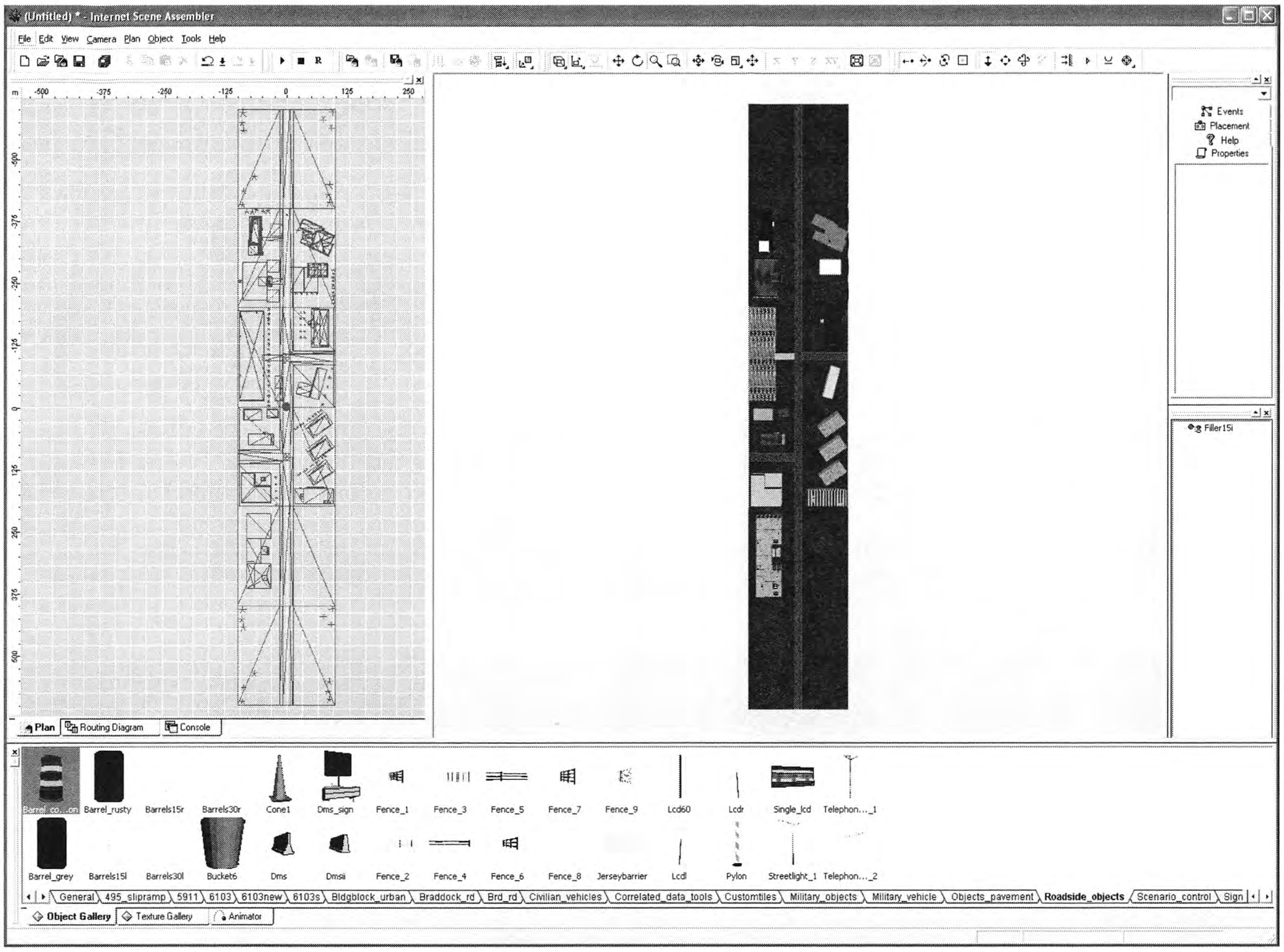


Figure 3. Realtime SimVista™ scenario authoring system.

SimVista™'s extensive software allows the user to control almost every aspect of the simulation. SimVista™'s open database format also allows users to import and configure customized visual and roadway models. The officially released Object Gallery only provides limited tiles; thus, TTI would like to be able to construct its own tiles for specific environments and embedded driving behavior for specific research topics. This is the advantage that the SimVista™ software and its open gallery provide.

PRESAGIS CREATOR

Creator (not to be confused with SimCreator®), which is developed by Presagis, Inc., is a highly specialized tool that helps modelers design high-fidelity and optimized 3D models and terrain for interactive real-time applications. The interactive applications supported by models created in Creator are diverse in nature, ranging from flight and vehicle simulations for military personnel to visual demonstrations of construction projects for architects.

More than a basic modeling tool, Creator integrates a set of powerful tools supporting hierarchical visual databases in a "what you see is what you get" environment, as shown in Figure 4. The key characteristic—and a major differentiator of Creator from other similar software—is its extremely high level of precision, control, and accuracy at all levels of the scene, from site level to vertex level. Extensible and multipurpose, the Creator software provides immersive control over the entire modeling process. The polygon-based authoring system in Creator generates optimized object models, high-fidelity terrain, and realistic synthetic environments.

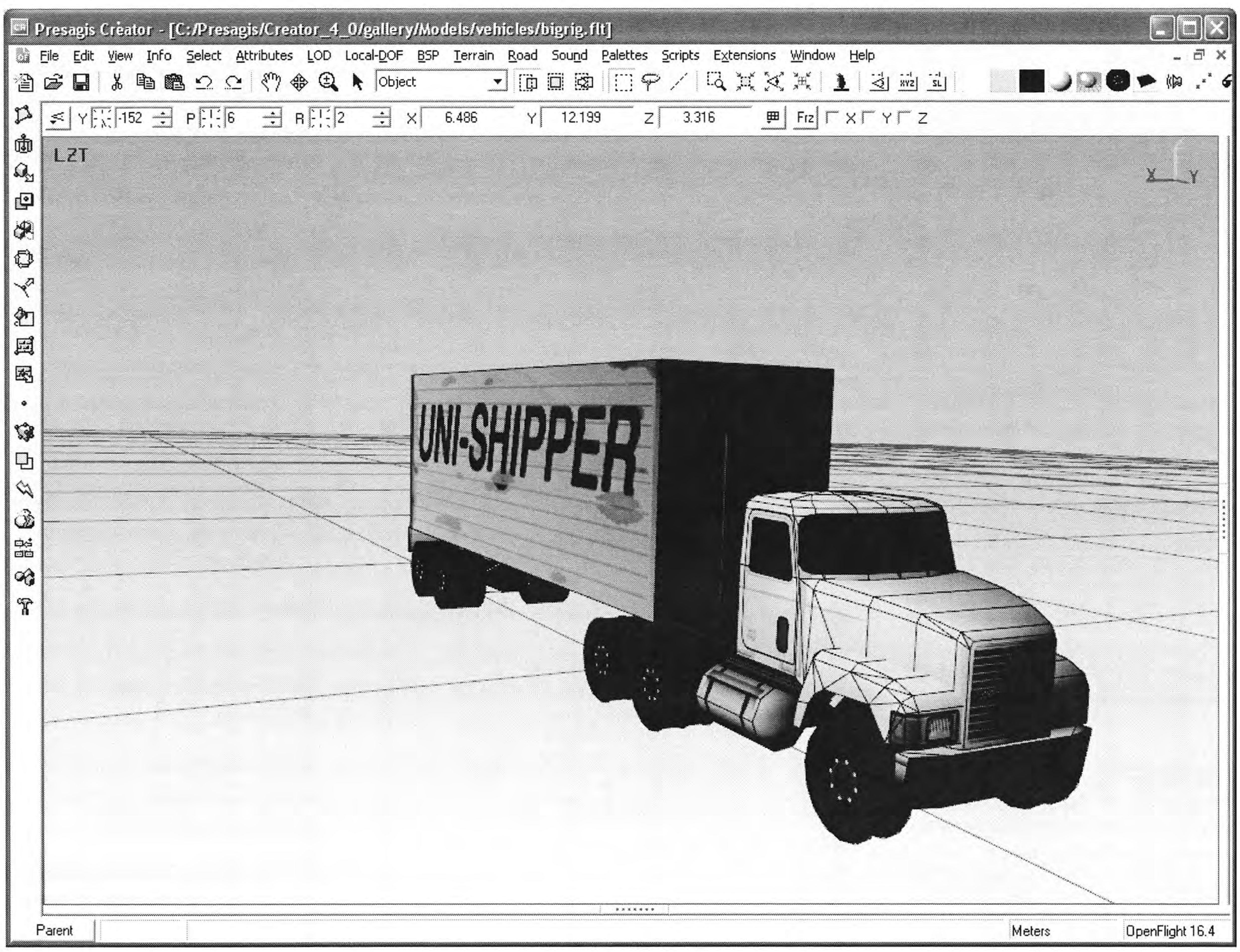


Figure 4. Presagis Creator modeling tools.

Creator can build optimized models that simplify and reduce programming requirements for the real-time application. It also provides a graphic user interface for constructing models, terrain, and scenes in a hierarchical visual database that conforms to the OpenFlight standard file format. OpenFlight is the standard real-time 3D database file format and is the native output file format of Creator, offering comprehensive interactive control. The customizable extension capabilities of the OpenFlight format provide a common ground for its diverse users and a mechanism to save and re-use objects or environments for future projects. The OpenFlight file becomes part of a real-time application after it is imported into run-time software such as Presagis Vega Prime.

PRESAGIS CREATOR ROAD TOOLS

Creator Road Tools, developed by Presagis, Inc. and the University of Iowa College of Engineering, is a powerful package that can rapidly create realistic roadways for high-fidelity real-time 3D graphics applications. It is purchased exclusively as an add-on module for Presagis Creator or Creator Pro (see Figure 5).

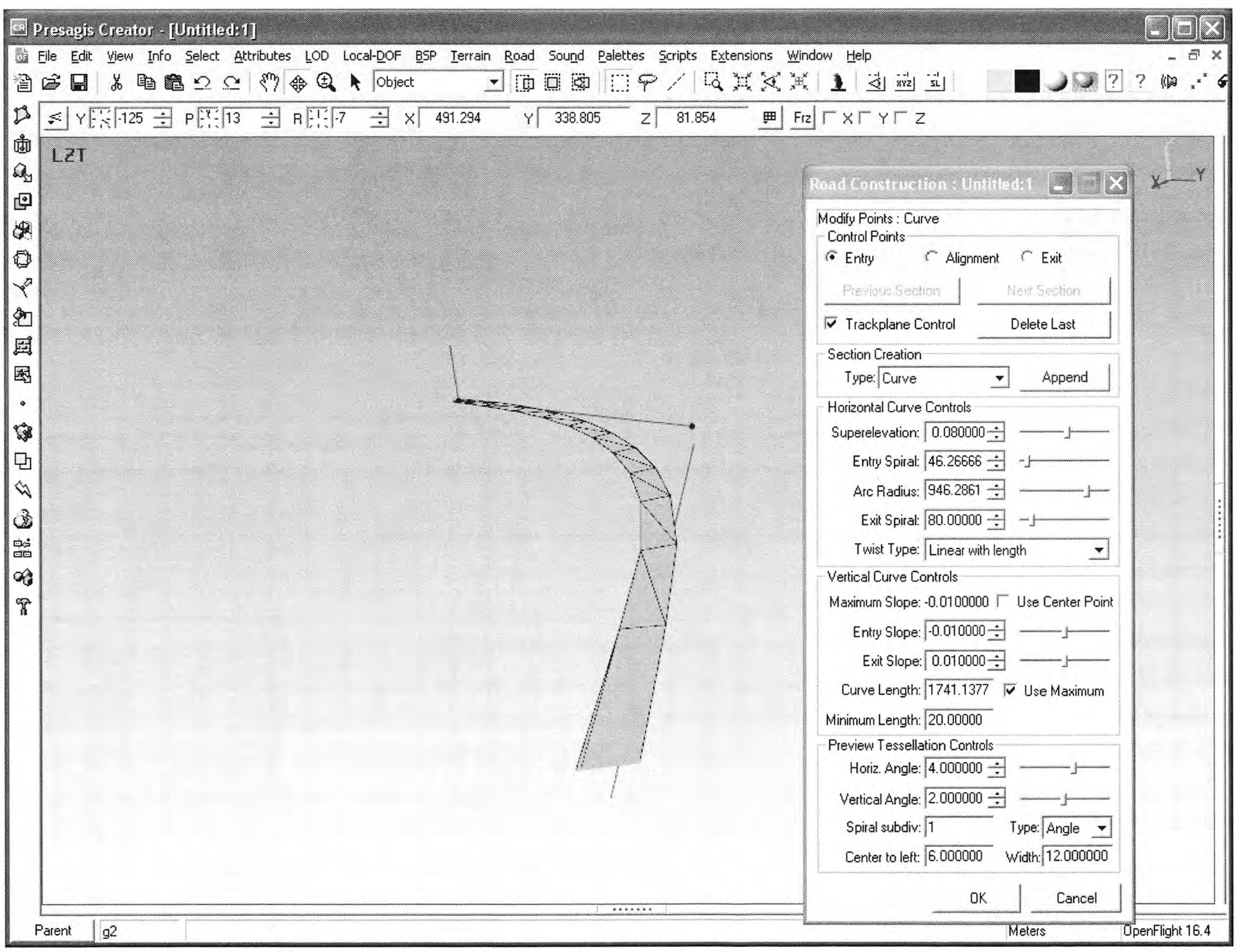


Figure 5. Presagis Creator Road Tools road-modeling add-on.

Creator Road Tools meets the stringent requirements of driving simulations for the purpose of vehicle design, specialized driver training, entertainment, driving behavior study, and accident re-enactment. Utilizing accurate and advanced algorithms for synthetic road generation to enhance the virtual environment of urban simulations and transportation simulation, Creator Road Tools can generate the placement of roads with automated attributes into a synthetic environment. The software previews them in a unique, pre-simulated "drive" mode in the modeling environment.

Road Tools can be broken down into three basic operations: road construction, road tessellation, and scenario data. Road construction defines the type of road section to be created. All new or existing road alignments and design parameters can be interactively manipulated or modified with dynamic updates of the resulting 3D geometry. Road tessellation defines and applies the road's attributes, including its appearance at various levels of detail. Regularly spaced roadway features, such as light poles and reflective markers, can be placed along the roadway automatically. An unlimited number of levels of detail (LODs) can be generated, each with its own features, cross section, texture, and optimized curvature-based polygonization. Scenario data are also created for use in the driving simulation. These data include lane and centerline definition.

TTI has chosen to utilize Creator Road Tools because it accurately models roadways that conform to highway design standards in transportation engineering. Although it cannot generate complex roadways or intersections and the generated roadway model needs to be adjusted manually, Creator Road Tools is still a powerful tool since it can significantly reduce the cost of roadway modeling and standardize the models themselves based on the OpenFlight hierarchy database.

CREATING NEW TILES

ORIGINAL METHODOLOGY USING AUTOCAD® CIVIL 3D®

The original methodology in the project proposal was to develop and implement an algorithm in order to generate a road model that could be used in the driving simulator based on the given road design data in DWG files designed in AutoCAD® Civil 3D®. AutoCAD® Civil 3D® is a widely accepted building-information modeling solution for civil engineering. The second step was to find a method to convert the scenarios in OpenFlight format to an executable format according to the specific driving simulator. With TTI's Realtime Technologies, Inc., simulator, the expected final executable format is a customized Virtual Reality Modeling Language (VRML) format for SimVista™'s Internet Scene Assembler (ISA), which is a 3D authoring tool that facilitates the creation of interactive and dynamic applications. These two steps are shown in Figure 6.

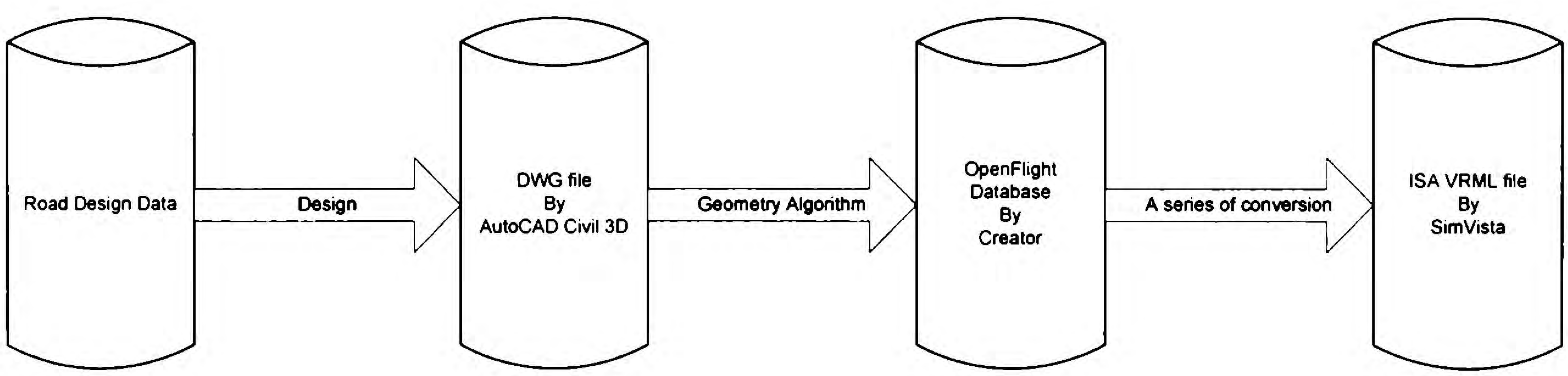


Figure 6. The original methodology in the project proposal.

The research team encountered two obstacles when they tried to use the original methodology. The first problem was that the API provided by Autodesk® for AutoCAD® Civil 3D® is not compatible with the API for the OpenFlight database. They are based on different languages and environments. Autodesk® did not provide free interface access data in DWG format for stand-alone application. However, the Open Design Alliance (ODA), formerly known as OpenDWG, represents a nonprofit membership-based association of software companies, software developers, and users that provides ODA software libraries that can access the files in DWG format. The second problem was that the models constructed in AutoCAD® Civil 3D® were not practical and executable in driving simulation because the simulation requires not only the exact geometry data and visible model but also the drivable, interactive, and dynamic environment. The research team tried to overcome these obstacles and explore alternative methods to achieve the goals of this project.

ALTERATIVE METHODOLOGY USING ROAD TOOLS

After investigating the use of Civil 3D® to generate DWG files, the research team discovered and obtained Presagis Creator Road Tools, a road modeling package. The investigation revealed that Road Tools is a powerful and convenient tool to generate 3D models for roadways and the features along the roadway when given specific road design data. The alternative methodology utilizes the commercially available Presagis Creator Road Tools, which allows the user to input road design variables directly. This eliminates the step of using AutoCAD® Civil 3D® to generate the roadway geometry DWG files. Since the road designed in AutoCAD® Civil 3D® is also based on the road data, as is Road Tools, using Presagis Creator Road Tools is a reasonable alternative method to develop an executable driving simulation environment. As shown in Figure 7, both methodologies were to develop an executable driving simulation environment and scenario according to the field road design data available. The alternative methodology means the existing road design file in DWG cannot be utilized, and the fidelity of the environment is limited because of the limitation of Road Tools. However, the complexity and fidelity of the models generated by Road Tools are reasonably low yet practical for the resources required for the simulation. Additionally, unlike designing in AutoCAD® Civil 3D®, it requires less cost and professional skill to design a road model with limited fidelity using Creator Road Tools. Thus, the alternative methodology is reasonable and more practical than the original methodology.

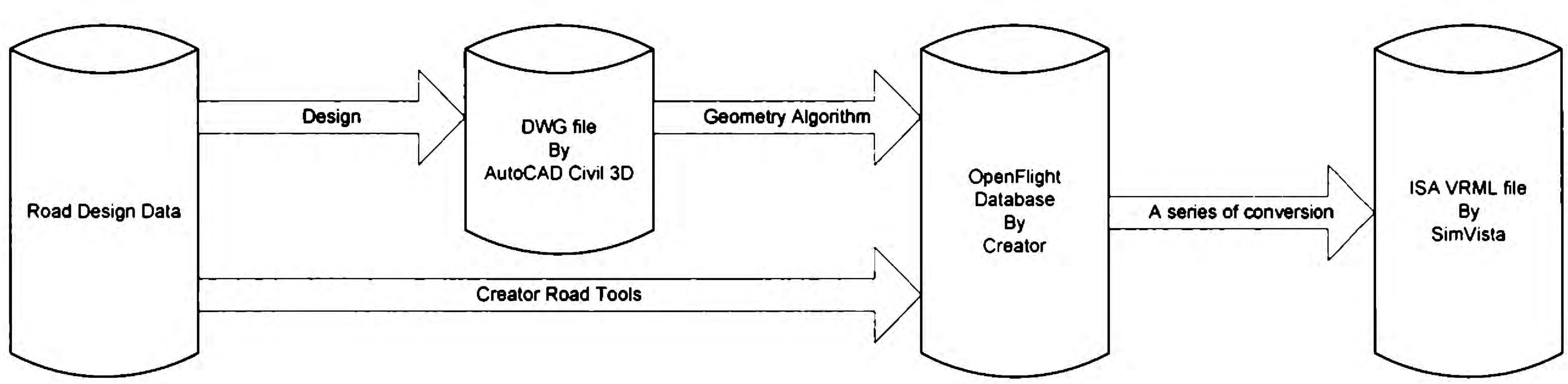


Figure 7. The alternative methodology for the project.

The detailed alternative methodology is illustrated in Figure 8. Based on the initial road design data, the model can be created in Presagis Creator Road Tools; then the OpenFlight file is converted into the published ISA tile by series transformation and adjustment. In addition to the visible geometry, additional data that are invisible can be edited manually in SimVista™ utilizing its customized VRML file format. This includes the texture, path, and correlated data of the model, which control the appearance and motion of the vehicles and other objects in the scenario.

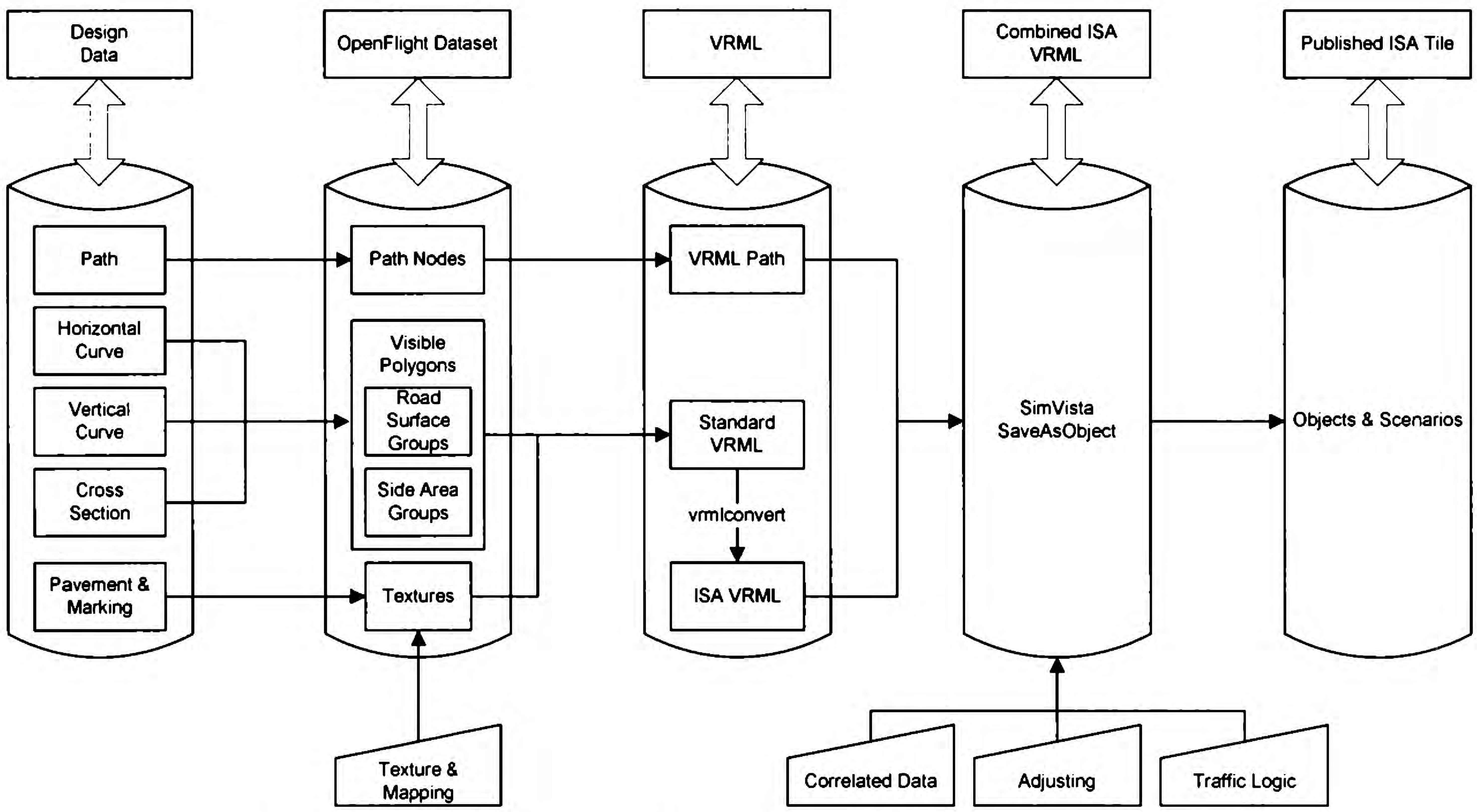


Figure 8. Detailed procedure of the alternative methodology.

The following sections describe the detailed process for developing new tiles in SimVista™ from road design data. Some steps of the process are optional based on what is included in the scenario. The development procedure can be summarized in the following steps:

1. Geometry design and adjustment based on road design data and existing tiles.
2. Texture development based on road design data.
3. Road construction in Presagis Creator Road Tools.
4. Road tessellation in Presagis Creator Road Tools.
5. Path definition and generation in Presagis Creator Road Tools.
6. Conversion of the OpenFlight file to a standard VRML file.

7. Conversion of the standard VRML file to an ISA VRML file.
8. Conversion of the PATH file to a VRML path file.
9. Combination of the VRML file and the VRML path file.
10. Adjustment of the data in the combined ISA VRML file including texture.
11. Addition of the traffic logic and driving behavior into the object.
12. Publishing the object or scene and adding it to the gallery as a new tile.

This process consists of the common steps for a single road section. Complex tiles with traffic control logic, such as intersection or traffic merge sections, cannot be directly developed by the function of these packages. Manual adjustment and editing in the 3D model are necessary after step 3. Even some textures may be required or may need to be redesigned. Step 11 is used for complex tiles in order to add vehicle control logic.

GEOMETRY DESIGN AND ADJUSTMENT

As discussed previously, the capability of Road Tools is limited. Only basic geometry parameters can be inputted into the software. Thus, it is necessary to find a feasible compromise to design the model within the limitations of the package and the purpose of the model.

If the model is designed based on field road data and simulated in order to examine or preview the road and the environment along the road, the fidelity of the model should be as high as the limitation of Road Tools. Practically, higher fidelity means more polygons in the model, which leads to a higher requirement for hardware resources at run time. Once the requirements overwhelm the available hardware resources, immersion and interaction can become unrealistic and therefore unacceptable.

If, on the other hand, the model is designed to study a small subset of possible road features such as signs or intersection studies, the model can run at a reasonably low fidelity and keep the features on which the study focuses. So, for instance, if the purpose of the study is to examine intersection signal operations, it is not important to perfectly model the superelevation of curves leading to the intersection.

If the new model needs to be compatible with existing tiles in the authoring software gallery, such as the officially released tiles by Realtime Technologies, Inc., the parameters of existing tiles must be investigated for seamless connection. Unfortunately, Realtime Technologies, Inc., does not provide accurate data for the library tiles. The tile descriptions in the user manual provide dimensions that represent the sum of the underlying roadway features and the textures mapped to them. This means that the roadway texture includes the width of both the travel lane and the shoulder. For modeling purposes, it is important to know these two entities' dimensions separately.

For a seamless simulation environment, the actual values of these parameters are necessary. For example, all officially released tiles ignore the cross section and assume the road surface is flat surface, which is 0.03 m higher than the ground. Comparing the cross section in the American Association of State Highway and Transportation Officials (AASHTO) *Green Book (16)*, as shown in Figure 9, the model used by driving simulation relatively reduced the fidelity of the roadway and complexity of the model. This is an example of simplifying models for simulation. If the new model is designed to work together with existing tiles, it has to present the same features.

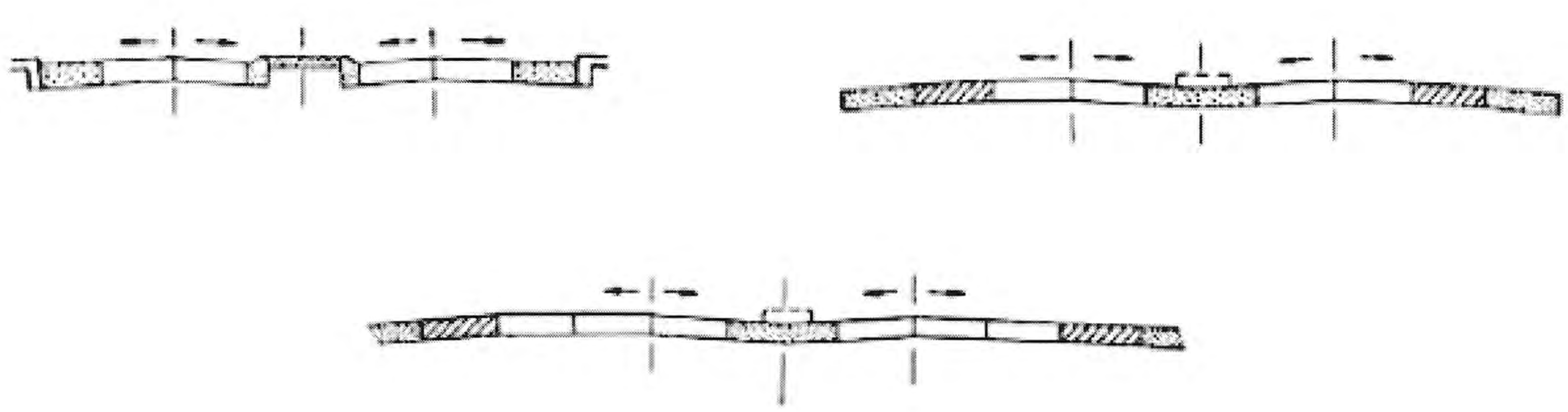
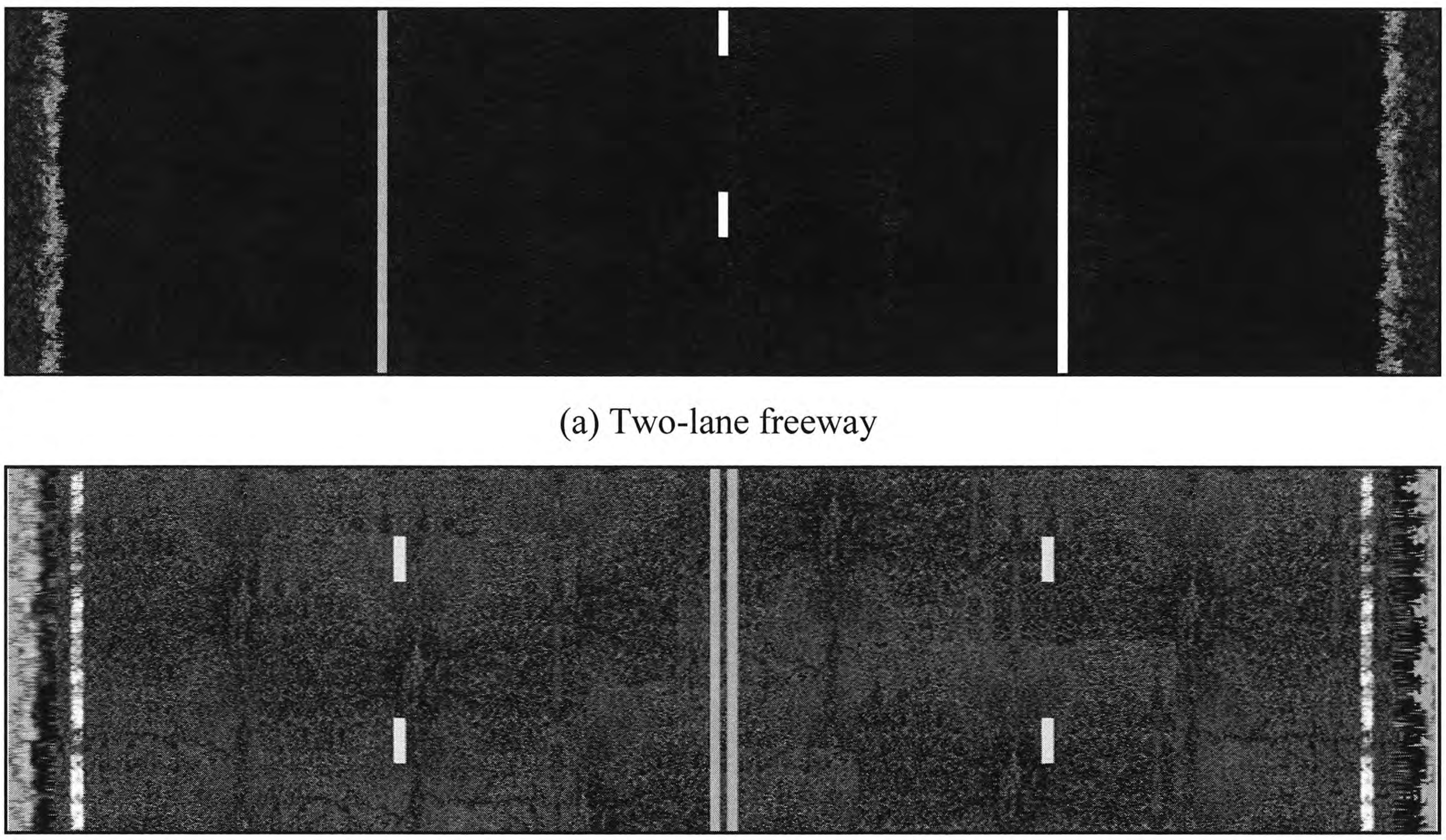


Figure 9. The basic cross-slope arrangements in the AASHTO *Green Book*.

TEXTURE DEVELOPMENT

The pavement and markings on roadways are represented by textures in a corresponding 3D model. Thus, the development of texture should be elaborate and accurate. Presagis, Inc. can support a number of picture formats as textures, while SimVista™ only supports GIF, JPG, and PNG formats. TTI prefers to develop textures in PNG format using a 24-bit mode in Photoshop® CS.

Normally, the texture of a road is a processed photograph of a specific road surface, as shown in Figure 10. The recommended length and height of the final texture file are 2^n pixels, such as 256 × 128. A transparent layer is necessary because in the real world the edge of the road is not fine. Obviously, the visible effect of the road surface is determined by the content of the texture. Since the pavement and marking design should follow the AASHTO *Green Book* and *Manual on Uniform Traffic Control Devices* (MUTCD) *(17)*, the textures need to be adjusted in order to correctly map textures following these standards. The most basic elements that require special attention are lane width, shoulder width, and the ratio of striping to gap for the centerline. Adjustments are closely correlated to the texture-mapping strategy.



(a) Two-lane freeway

(b) Four-lane freeway

Figure 10. The texture of two-lane and four-lane freeways.

ROAD CONSTRUCTION

Presagis Creator Road Tools provides powerful functions to construct road alignment, as shown in Figure 11. In the Road Construction dialog box, users can define the type of road section and set parameters to define the road's curve, superelevation, and slope. The location of the control points for the road section can be indicated in the Graphics view, displaying the road section. If the designed section is mathematically feasible, it appears green; if not, it appears red.

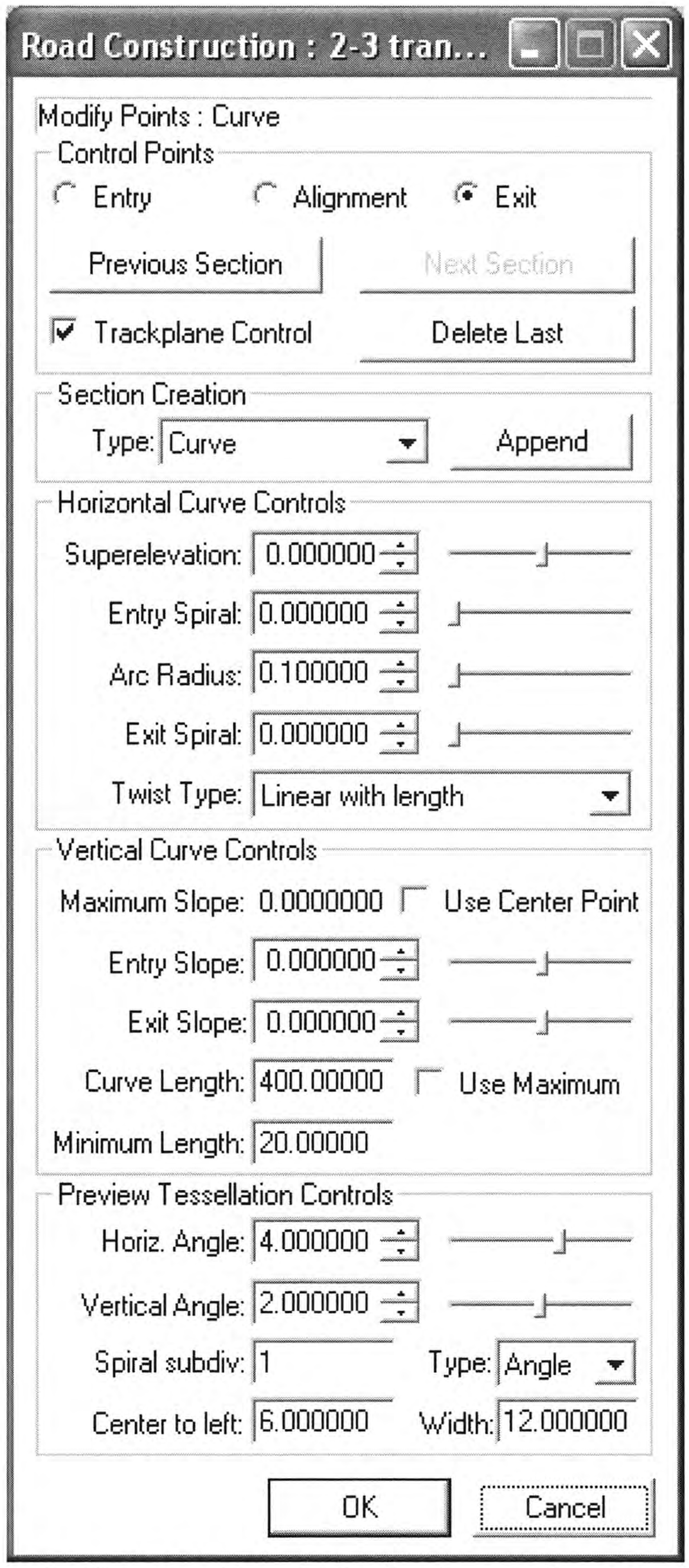


Figure 11. Presagis Creator Road Tools: Road Construction dialog box.

Road Tools provides a number of parameters to develop the alignment of the road. Most of them can be intuitively understood by a transportation engineer. The definitions of the parameters *(18)* are listed in Table 1. Road sections are placed as road nodes at the group level in the Hierarchy view. Each road section is comprised of at least one path node and a number of road subsections, represented by group nodes, that are children to the road node.

Table 1. Definitions of the parameters in the Road Construction dialog box.

Parameter	Definition
Control Points	Defines the limits of the road, specifies the target road section for editing, and deletes road sections.
Entry	Defines the section's beginning point in space (note that this is not necessarily where curvature begins). The Road Construction dialog box prompts the user for the Entry point only when the user is building the first section of a road. For subsequent sections, the Construction Tool assumes the Entry point is the same as the third point of the previous section. The user can always modify a section's assumed Entry point. If the user does so, the previous section's Exit point changes to match, ensuring continuity.
Alignment	Defines the intersection of the tangents leading into and out of the curve in the horizontal dimension. The Alignment point can also be used to define the incoming and outgoing slopes of the section.
Exit	Defines the end point of the section.
Previous Section	Selects the road section built before the current section for editing. The previous section then becomes the current section.
Next Section	Selects the road section built after the current section for editing. The next section then becomes the current section.
Trackplane Control	Prevents the tracking plane from automatically realigning for each new road section. By default this option is set.
Delete Last	Deletes the last road section built.
Section Creation	Defines the type of road section the user is creating and allows the user to add new sections to existing ones.
Type	Specifies what type of road section to create: straight sections, curves, or hills: • Curve—Creates simple horizontal curves and horizontal and vertical curve combinations. Horizontal road curves are constructed using a radial arc with spiral (Euler) transition sections at either end to provide easement. A curve section's horizontal alignment and vertical profile are defined by inputting three control points and setting the curve's design parameters. • Hill—Creates parabolic vertical curves with no horizontal curvature. • Straight—Creates sections with no horizontal or vertical curvature. Note that straight sections are not often required; when the user creates curves and hills, the Construction Tool automatically constructs the straight segments necessary to provide a smooth transition between each adjacent section. The user needs to construct a straight section only when the section does not provide a transition between successive curves or hills, e.g., using straight sections to build a network of straight streets.
Append	Adds the next section of road to the end of the section just created.
Horizontal Curve Controls	Together with the three control points, defines the horizontal alignment of a curve.
Superelevation	Determines the maximum amount of bank in a curve, expressed as a slope (rise/run).
Entry Spiral	Specifies the distance, in database units, through which the curvature and superelevation are introduced into a curve.
Arc Radius	Specifies the radius of the circle that defines the centerline of the circular part of the curve. Every curve is defined by a circle. The circular arc segment conforms exactly to the circumference of this circle. The spiral segments in the curve form transitions between the circular arc segment and the previous and subsequent road sections.

Table 1. Definitions of the parameters in the Road Construction dialog box (continued).

Parameter	Definition
Exit Spiral	Specifies the distance, in database units, through which the curvature and superelevation are reduced when exiting a curve.
Twist Type	Controls the way in which superelevation (banking) is distributed in spiral sections: • Linear with Length—Indicates that superelevation is added to the surface of the spiral at a linear rate with respect to the distance into the spiral. For example, at a point 80 m into a 100 m spiral, 80% of maximum superelevation has been attained. This is the method used in the design of many real roads. When the road has dramatic spirals that attain unusually high superelevation in short distances, this method can result in very abrupt transitions, producing a bumpy road surface. The other two methods may be better when the road has such radical banking. • Linear with Curve Angle—Computes superelevation as a function that increases linearly with angular deflection into the spiral. As the rate of curvature increases more rapidly toward the arc end of a spiral, so would the rate of twist or introduction of superelevation increase more rapidly toward that end. This concentrates most of the twist at the arc end of the spiral segment. • Cosine Function with Length—Computes superelevation based on a cosine function with length along the spiral, giving very gradual twisting rates at both ends of the segment and concentrating the twisting in the spiral segment's midsection.
Vertical Curve Controls	Curves can have both horizontal and vertical curvature; hills contain only vertical curvature. Vertical Curve Controls for curve sections are the same as those for hill sections. Note that the vertical curvature is defined and computed independently of the horizontal alignment; vertical and horizontal curvature can be constructed separately or can occur together (either fully or partially) within a road section.
Maximum Slope	Displays the maximum slope of a hill or curve. This field is informational and cannot be modified.
Use Center Point	Defines the Entry Slope and Exit Slope from the lines connecting the first and third points to the second (center) point.
Entry Slope	Defines the slope at the first control point. The Entry Slope of the current section must equal the Exit Slope of the previous section. If the user enters a different Entry Slope, the Exit Slope of the previous section automatically changes to match, ensuring that road sections are continuous.
Exit Slope	Defines the steepness of the slope at the third control point.
Curve Length	Specifies the horizontal distance covered by the vertical curve. If the curve specifications force two curves, it is constructed from two parabolas of equal length. The Curve Length is the length of one of these parabolas. In highway design, the length of the curve is dictated by design regulations and depends on the design speed of the road. If the user is building a road that simulates real-life conditions, he or she should clear Use Maximum and enter a specific value for the Curve Length.
Use Maximum	Specifies that the vertical height should be based on the longest parabola that is tangent to the beginning and ending slopes and falls within the defined section.

Table 1. Definitions of the parameters in the Road Construction dialog box (continued).

Parameter	Definition
Minimum Length	Applies when the intersection of a section's beginning and ending slopes occurs very close to either the first or third control point. This condition can severely restrict the length of the vertical curve because the maximum length of a curve is twice the distance from the slope intersection to the nearest end point. If the vertical curve is restricted to a length that is less than the minimum length the user specifies, two parabolas of equal length are used to construct the curve section, thus allowing the use of longer parabolas.
Preview Tessellation Controls	Allows the user to define the road's initial width and centerline location so that the user can view the road while it is being built. The user can enter numbers in the various fields or use the sliders and view the road as it automatically updates the information. The information in these fields is applied to the entire road. However, the user must use the Tessellation Tool to further define the road's attributes (texture, color, levels of detail, and features) or load a setup (RDS) file.
Horiz. Angle	Controls the number of horizontal polygons to create smoother transitions into and out of curves. As Horiz. Angle increases, the number of polygons decreases.
Vertical Angle	Controls the number of vertical polygons to create smoother or rougher curves and hills. As Vertical Angle increases, the number of polygons decreases.
Spiral Subdiv.	Allows a smoother transition through the spiral sections of curves, by dividing road surface polygons into smaller polygons. To create smaller polygons within the spiral segment, the Horiz. Angle and Vertical Angle values are divided by the subdivide value. For example, a subdivide value of 2 would double the number of polygons used to represent the spiral section for that level of detail. While this results in a smoother driving surface, the extra polygons will impact system performance.
Center to Left	Defines the distance between the left edge of the road's cross section and the centerline of the road's construction.
Type	Specifies how the spiral segments are divided into polygons. This has no effect on the basic geometry of the spiral, but only on how polygons are used to surface that geometry: • Angle—Creates polygons according to the Horiz. Angle curvature restraint in the Road LOD attributes dialog box. Using this method, more polygons are generated for areas of high curvature, and fewer polygons are generated for areas of low curvature. • Distance—Uses the same length along the centerline for each polygon in the spiral segment. This length is equal to the length of the polygons in the curved arc segment.
Width	Sets the distance from the left edge of the road to the right edge, on a vector that is perpendicular to the centerline of the road. This width is expressed in database units and does not include side areas.

ROAD TESSELLATION

The Road Construction Tool only defines the alignment of road sections such as length and horizontal and vertical characteristics. It appears in the Graphics view as a flat ribbon. The Road Tessellation Tool is used to define the road's attributes such as road and side area width, define appearance at different levels of detail, add features such as light and traffic signs, and then apply these definitions to the road.

In the Road Tessellation dialog box, the road components appear in a hierarchical tree structure, as shown in Figure 12. The Road Definition Tree provides the function to save, create, edit, and move the components in the Road Tessellation Tree.

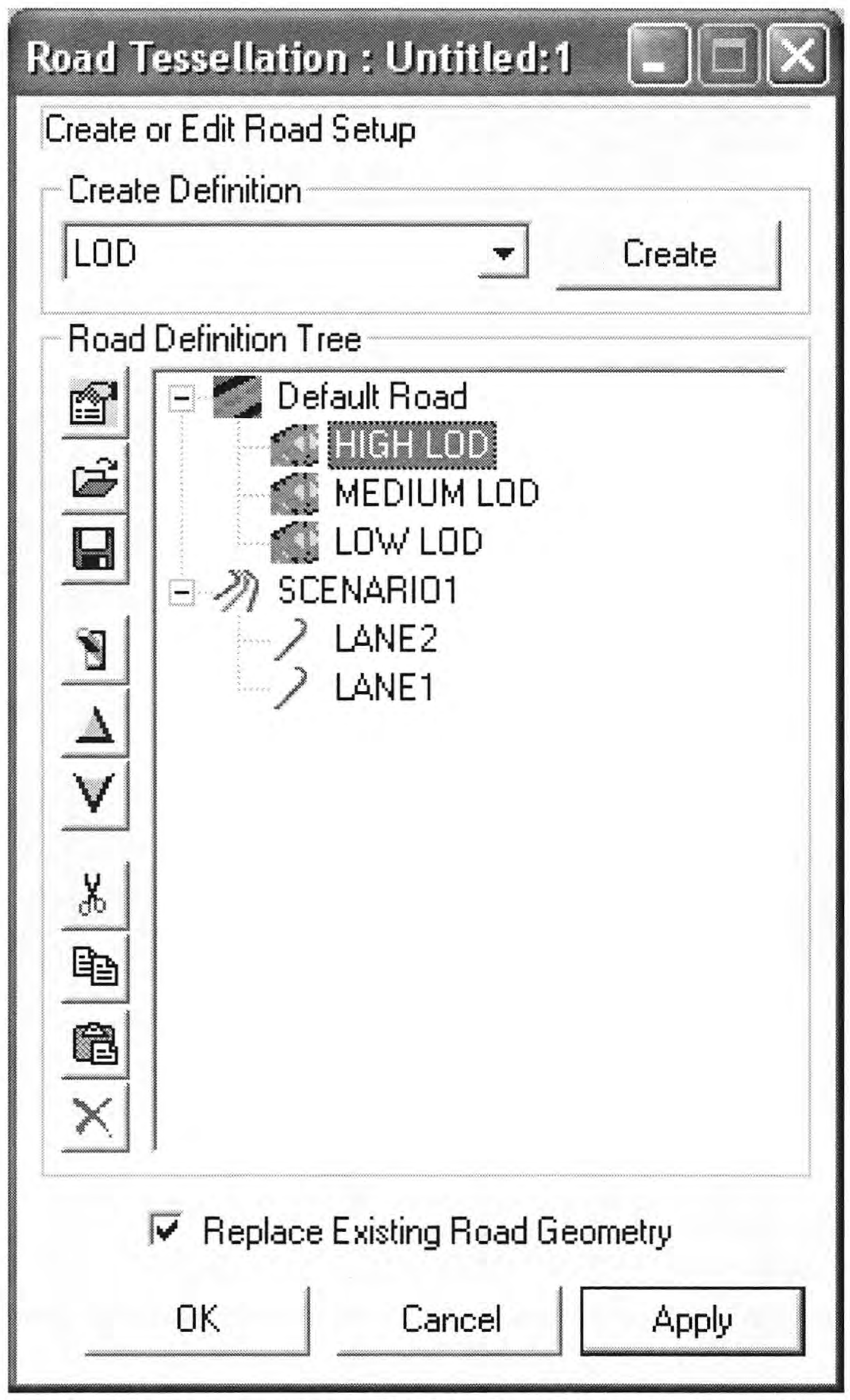


Figure 12. Presagis Creator Road Tools: Road Tessellation dialog box.

The first step of road tessellation is defining the road style. Road Bed attributes define the general characteristics of the road (Figure 13). These settings are overridden by the settings in Road LOD. The definitions of these parameters *(18)* are listed in Table 2.

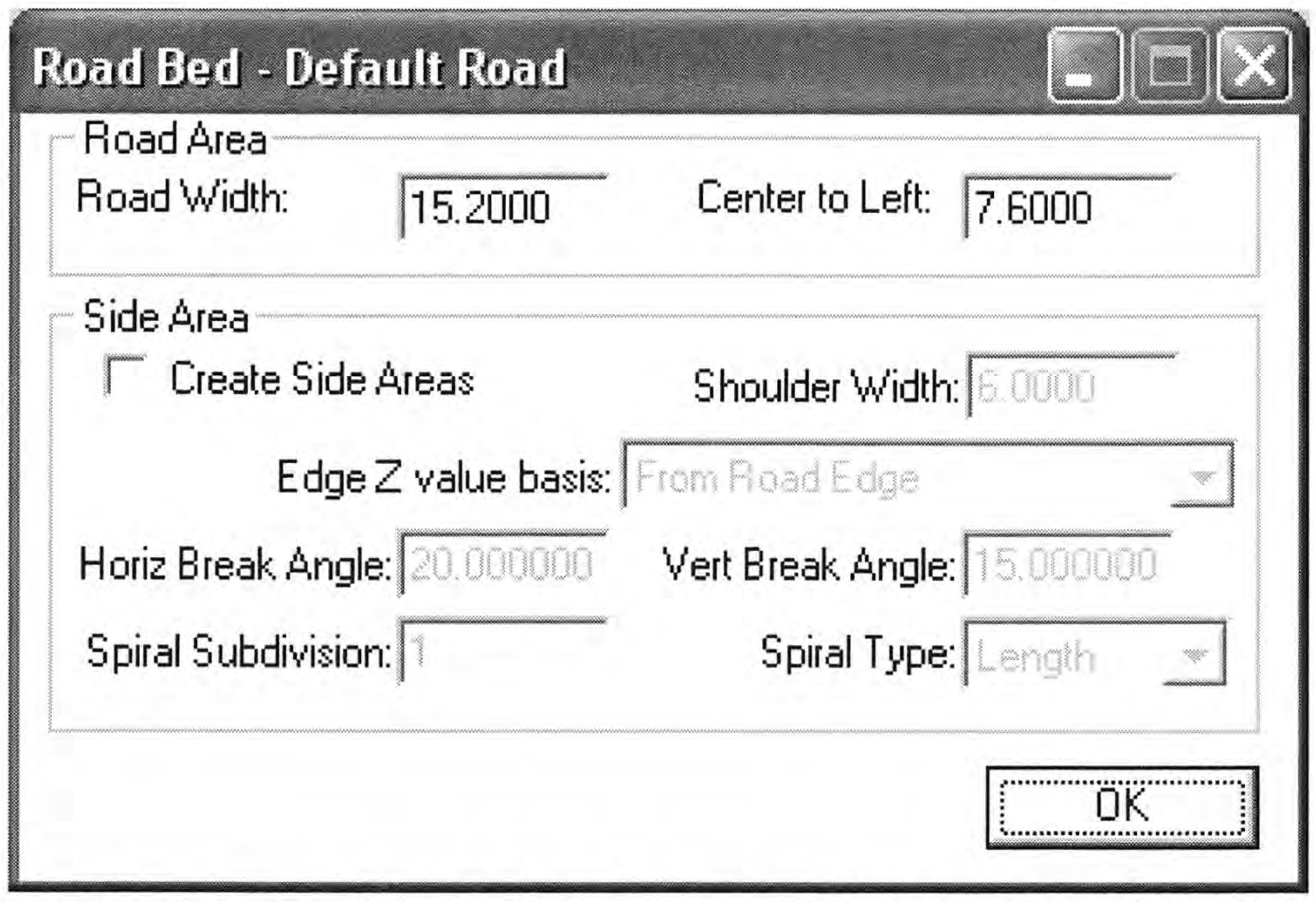


Figure 13. Presagis Creator Road Tools: Road Tessellation: Road Bed dialog box.

Table 2. Definitions of the parameters in the Road Bed dialog box.

Parameter	**Definition**
Road Width	Defines the width of the road in database units.
Center to Left	Defines the distance from the center of the road to the left edge in database units.
Create Side Areas	Specifies whether the road has a shoulder on each side.
Shoulder Width	Defines the width of the shoulder area. This field is not available if Create Side Areas is not set.
Edge Z Value Basis	On superelevated roads, determines whether the side areas will be banked and how the banking will occur: • No Change (Level with Road)—If the road is banked (superelevated), the side areas will be banked to remain at the same angle as the road. • From Road Edge—The side areas extend out flat from the edge of the road. • From Centerline—The far edges of the side areas are set to the z value of the road's centerline.
Horizontal Break Angle and Vertical Break Angle	Controls the construction of the polygons that form the surfaces of horizontal and vertical curves in the side area. The smaller the angle specified (in degrees), the larger the number of polygons produced.
Spiral Subdivision	Smoothes the transition through the spiral section of a curve by dividing the side area into smaller polygons. To create smaller polygons within the spiral segment, the Horizontal Break Angle and Vertical Break Angle values are divided by the Spiral Subdivision value.
Spiral Type	Specifies how the spiral segments are divided into polygons (this has no effect on the basic geometry of the spiral, but only on how polygons are used to surface that geometry): • Angle—Creates polygons according to the Horizontal Break Angle curvature restraint. This generates more polygons for areas of high curvature and fewer polygons for areas of low curvature. • Distance—Uses the same length along the centerline for each polygon in the spiral segment. This length is equal to the length of the polygons in the curved arc segment.

The second step is defining road levels of detail, road properties, and side area properties, as shown in Figure 14. The explanations of these parameters *(18)* are listed in Table 3.

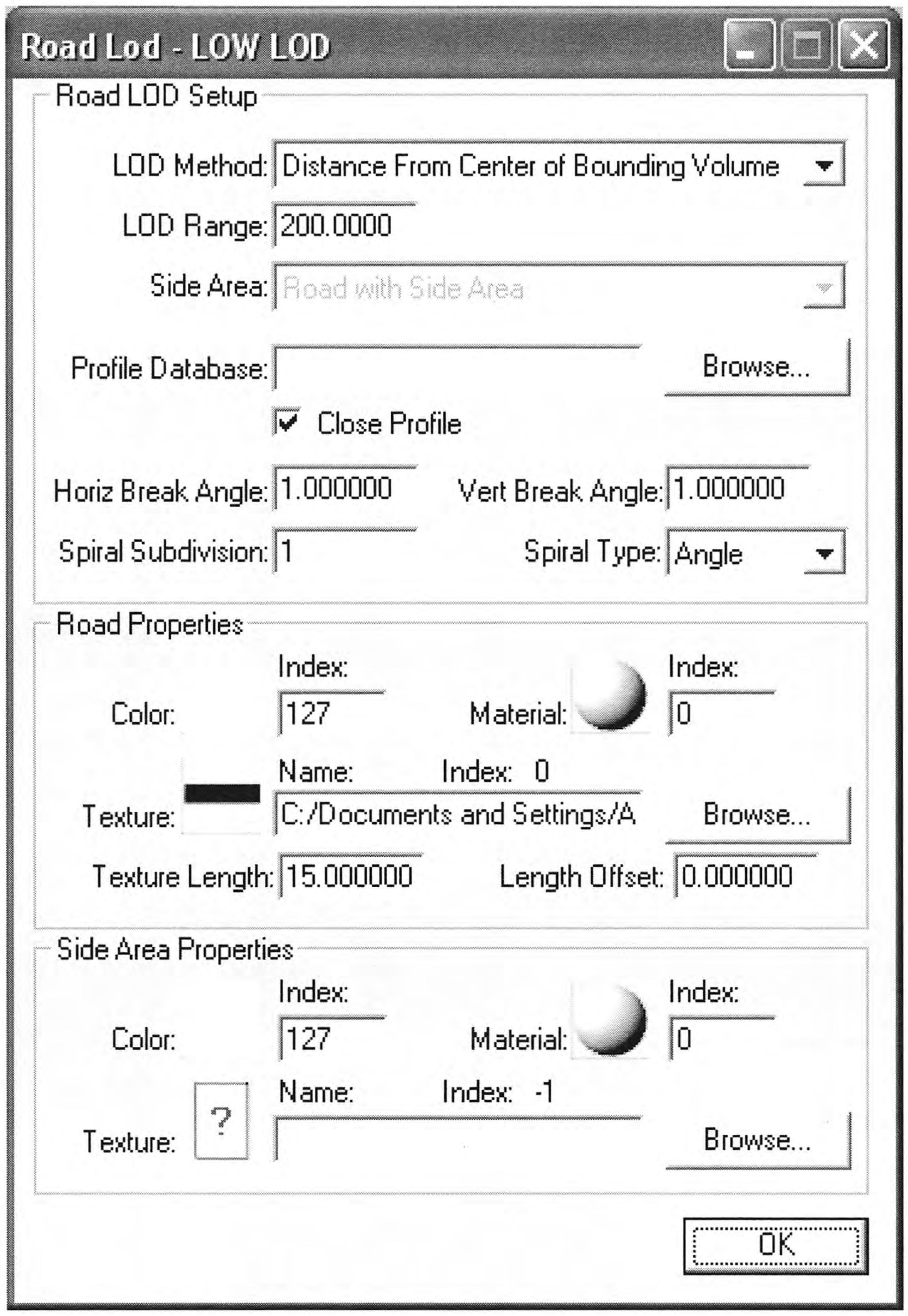


Figure 14. Presagis Creator Road Tools: Road Tessellation: Road LOD dialog box.

Table 3. Definitions of the parameters in the Road LOD dialog box.

Parameter	Definition
LOD Method	Specifies how the level of detail will be switched in: • Distance from Center of Bounding Volume—Switches in at the LOD Range setting by measuring the distance from the eye point to the center of the bounding volume. This is the normal switch-in method for levels of detail. • Multiple of Bounding Volume Size—Multiplies the size of the bounding volume by the LOD Range value. • Distance from Edge of Bounding Volume—Calculates the distance from the eye point to the edge of each bounding volume.
LOD Range	Used together with the LOD Method, calculates the farthest distance from the eye point at which the level of detail is displayed (switch-in distance).
Side Area	Defines how the road and side areas will appear when the LOD switches in: • Road with Side Area—Appears as a road with side areas. • Road Extends into Side Area—The road's width includes the side areas. It uses the Road Bed Side Area definition for tessellation. Any LOD tessellation definitions for side areas are ignored. • Side Areas Only—Only the side areas are visible. This option is very useful when multiple road beds have been defined.
Profile Database	Can be used to loft the road, generating a road with a cross section rather than a flat surface. However, this produces more polygons.
Close Profile	When set, all edges of the polygons in the Profile Database are used to loft the road surface. When cleared, the last edge is not used to loft the road. This is desirable when the road is built on the ground, where the bottom edge is not visible.
Horizontal Break Angle and Vertical Break Angle	Controls the construction of the polygons that form the surfaces of horizontal and vertical curves. The smaller the angle specified (in degrees), the larger the number of polygons produced. Maximizing rendering performance requires some subjective judgments by the modeler. The Horizontal Break Angle and Vertical Break Angle values should be set as high as possible to maintain acceptable visual quality when switching between levels of detail, while minimizing the number of polygons. A few visual tests on the target Realtime software and display system are recommended to optimize these settings for the user's application.
Spiral Subdivision	Allows a smoother transition through the spiral sections of curves by dividing road surface polygons into smaller polygons. For example, when a high rate of twist (rapid superelevation) is modeled, large polygons must cover surfaces that ascend or descend in steep angles. The abrupt angle changes between the polygons produce rough spots (bumps) in the visual model of the road. To create smaller polygons within the spiral segment, the Horizontal Break Angle and Vertical Break Angle values are divided by the subdivide value. For example, a subdivide value of 2 would double the number of polygons used to represent the spiral section for that level of detail. While this results in a smoother driving surface, the extra polygons will impact system performance.
Spiral Type	Specifies how the spiral segments are divided into polygons. This has no effect on the basic geometry of the spiral, but only on how polygons are used to surface that geometry.
Color	Specifies the color and its palette index number that will be applied to the road surface.
Material	Specifies the material property and its palette index number that will be applied to the road surface. Material properties can only be viewed if Calculate Shading is performed.

Table 3. Definitions of the parameters in the Road LOD dialog box (continued).

Parameter	Definition
Texture	Specifies the texture pattern that will be applied to the road surface. The user can specify either a texture pattern name or its index number. Every texture pattern must be loaded in the database's Texture palette before it can be applied to a surface. When the user specifies a Road Texture file or a Side Area Texture file, the Texture palette is checked for the texture pattern. If it is not present, the pattern is automatically loaded into the Texture palette, if possible: • If the user types in a texture name without specifying a directory path, the Texture palette is searched, and the pattern is applied if a match is found. If the specified texture pattern has not been loaded in the Texture palette, the current texture is applied. • If the user types in a texture name without a full directory path and multiple textures in the palette have the same name but different paths, the matching texture with the lowest index is used. To ensure that the correct texture file is used, it is best to specify a full path. • If the texture pattern cannot be found, an error message is generated. The user should check the location of the texture pattern file and enter the correct path.
Texture Length	Defines the length of one copy of the pattern. If this is less than the length of the road section, the pattern repeats, provided the Wrap Method attribute is set to Repeat. (See the "Texture Development" section for more information on texture attributes.)
Length Offset	Defines the distance before the start of the road that the texture pattern starts. If this is not the first section of the road, Length Offset is automatically set to match the texture applied to the previous section.
Side Area Properties: Texture	Specifies the texture file to be applied to the side areas of the road if Create Side Areas is enabled in the Road Bed dialog box.
Side Area Properties: Color	Specifies the color and the palette index number that will be applied to the road's side area.
Side Area Properties: Material	Specifies the material property and the palette index number that will be applied to the road's side area.

Levels of detail permit a real-time system to switch between more or less complex versions of a model for the same object, depending on the distance from the model to the eye point. The user should pay special attention to the profile database that defines and loads the external cross-section model in the dialog box; however, since the official tiles released by Realtime Technologies, Inc., ignore the cross section, it is reasonable to ignore the cross section as well if the purpose of the model permits.

Road properties include the color, material, and texture of the road surface and the texture-mapping strategies. The texture-mapping strategies are correlated to texture development mentioned above.

Roadside area properties include the color, material, and texture of the road surface and the texture-mapping strategies. The library tiles model the travel lane and shoulder together. To design a new tile compatible with an existing tile, the travel lane and shoulder are supposed to be designed by adjusting the texture and texture mapping strategies.

The third step is adding features to a road definition. Features are isolated 3D object models, such as street lights and signs that can be added to the tessellation tree at the root or LOD level. Features can be saved as geometry in the current database or as external references. In this project, the post process software, SimVista™, provides the function to add features into the environment. Consequently, the researchers recommend applying features in later steps with SimVista™.

PATH DEFINITION AND GENERATION

The Write Paths function writes path information in an ASCII text dataset for use by vehicle dynamics and traffic simulation in a driving simulator. There are two datasets for roadway centerline and scenario lanes, which are correlated spatially with the roadway models in the visual database, as shown in Figure 15. Both are isolated to the visible geometry, and the path dataset has its own value source for their specific functions.

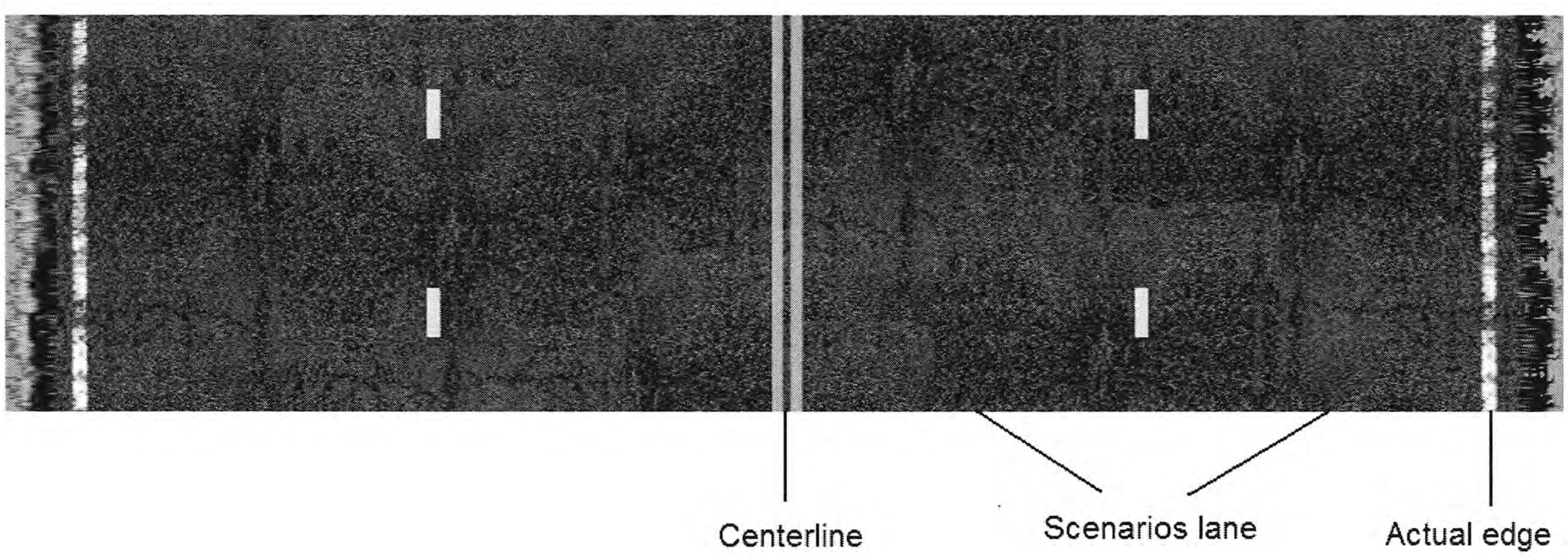


Figure 15. Centerline and scenario lane.

To generate the path files, the first step is defining a Road Scenario in the Road Definition Tree. Road Scenarios contain the information for roadway centerline, such as the profile database shown in Figure 16. Information in the Road Scenario dialog box does not affect the appearance of the road. Scenario information is stored in the path node in the hierarchy database. The path node saves the data about cross section and lane definitions.

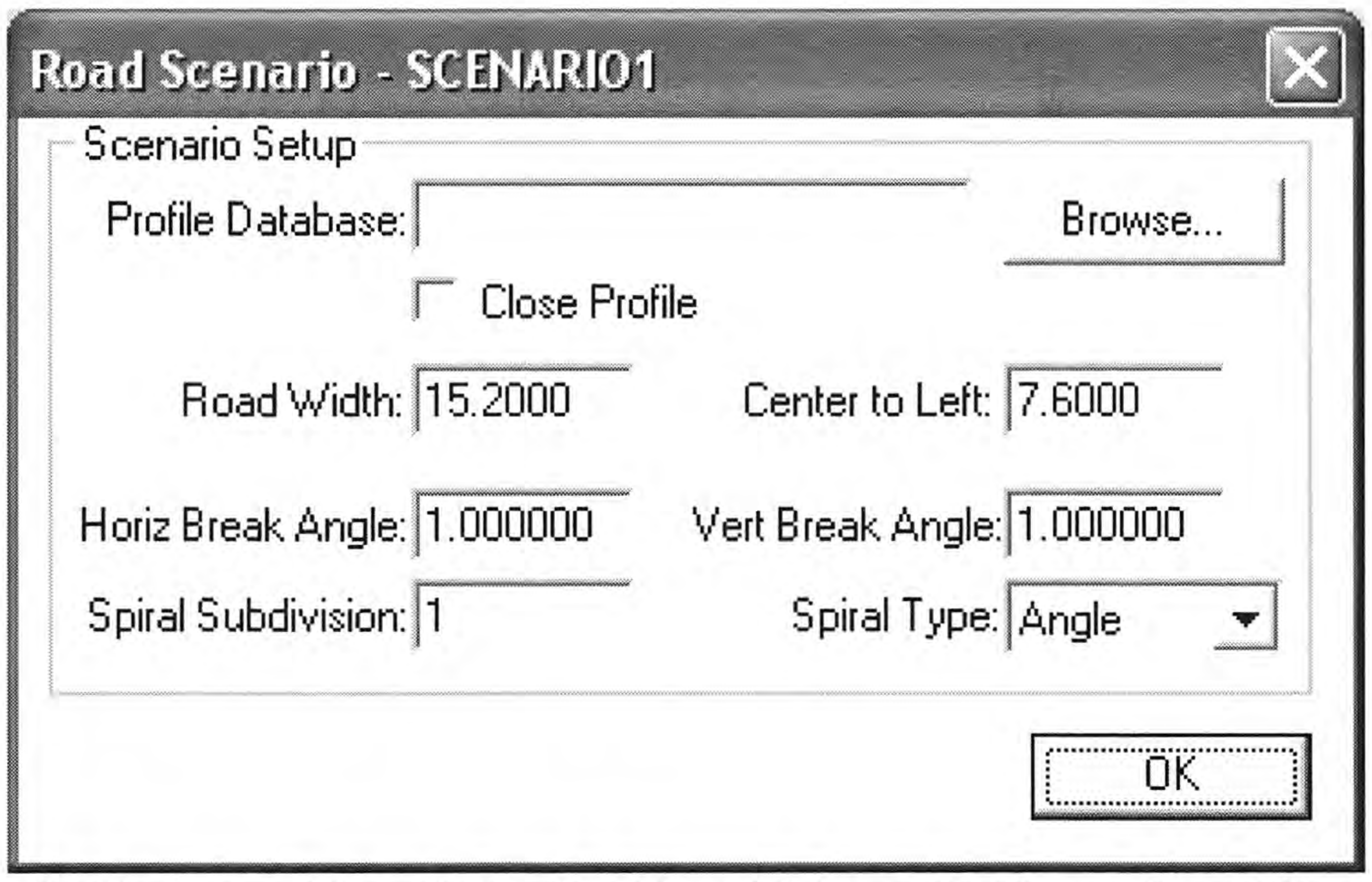


Figure 16. Presagis Creator Road Tools: Road Tessellation: Road Scenario dialog box.

The next step is defining scenario lanes that provide additional scenario paths for traffic simulation software. The centerline is the initial path of the entire paved surface that is drivable, and any number of lanes can exist on that paved surface. Scenario lanes contain lane information for traffic simulations including road design parameters, cross-section data, or other surface information for vehicle dynamics processing.

In the Tessellation Tool's Road Definition Tree, lanes are children nodes of a Road Nodes. The default lane is defined as the road centerline and is also called the Center Lane. Lanes are defined as lateral offsets from the road centerline, as shown in Figure 17. The definitions of these parameters *(18)* are listed in Table 4. In the Hierarchy view, scenario lane information is stored in the face node under the scenario path node. As shown in Figure 18, the green nodes are face nodes (shown in white in the Hierarchy view) for roadway centerline, and the blue nodes are face nodes for scenario lanes. Each lane created generates a new scenario path node and its child face node.

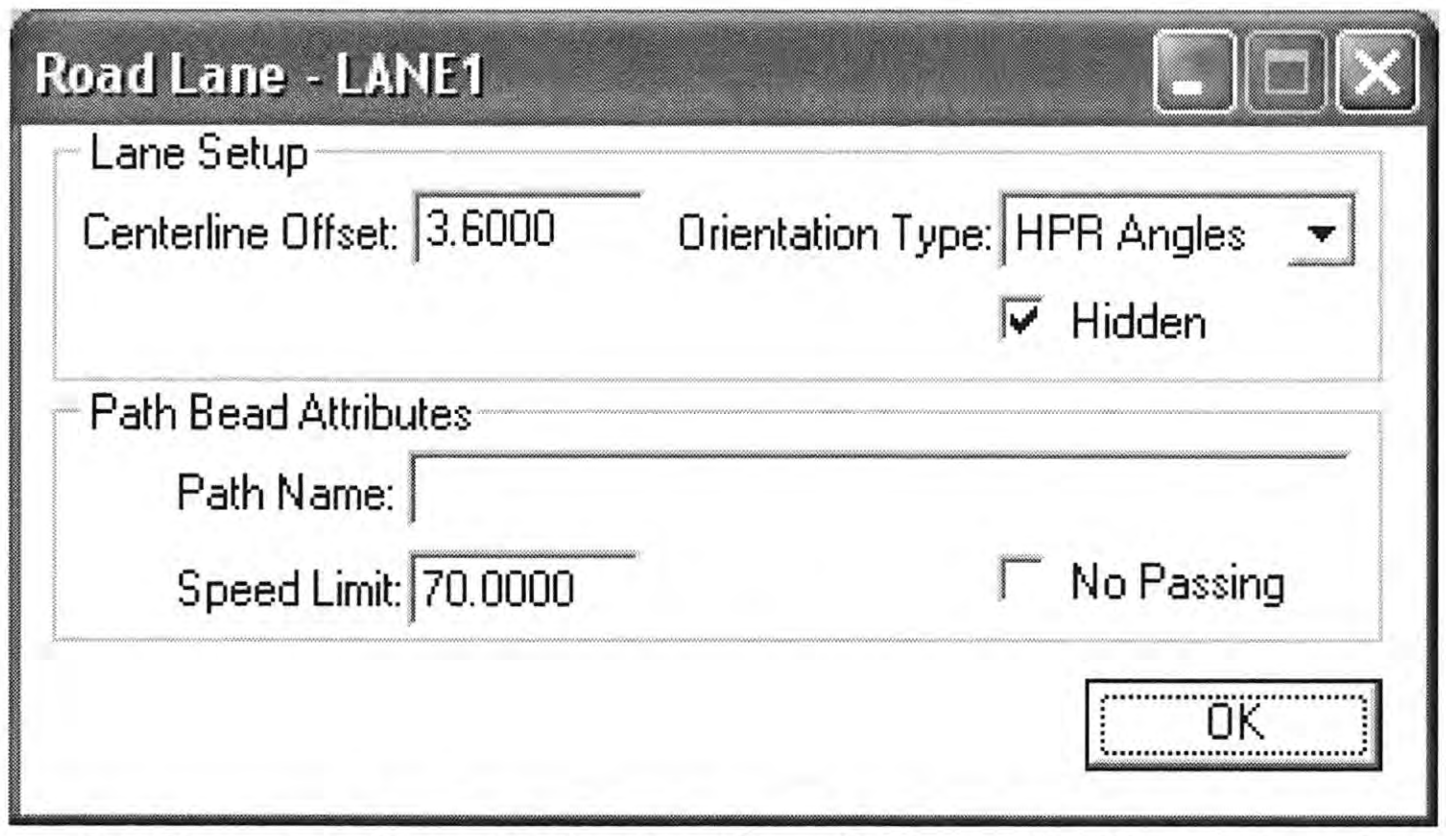


Figure 17. Presagis Creator Road Tools: Road Tessellation: Road Lane dialog box.

Table 4. Definitions of the parameters in the Road Lane dialog box.

Parameter	Definition
Centerline Offset	Specifies the distance from the center of the lane to the center of the road. When the offset is a non-zero value, the centerline appears in the Graphics view as a blue line. The first lane with a zero offset defines the road's centerline.
Orientation Type	Defines the road's surface along the points of the centerline: • Up Vector displays a normalized vector at each point along the centerline. • HPR Angle specifies Heading, Pitch, and Roll at each point.
Hidden	When set, creates the lane centerline as hidden faces that can be displayed by choosing View>Hidden Faces>Draw Normal. When this checkbox is cleared, the centerline is always visible in the Graphics view.
Path Name	Specifies the name of the Path.
Speed Limit	Specifies the speed at which the road will be driven in the Realtime system or simulator.
No Passing	When set, initiates a flag in the Realtime system.

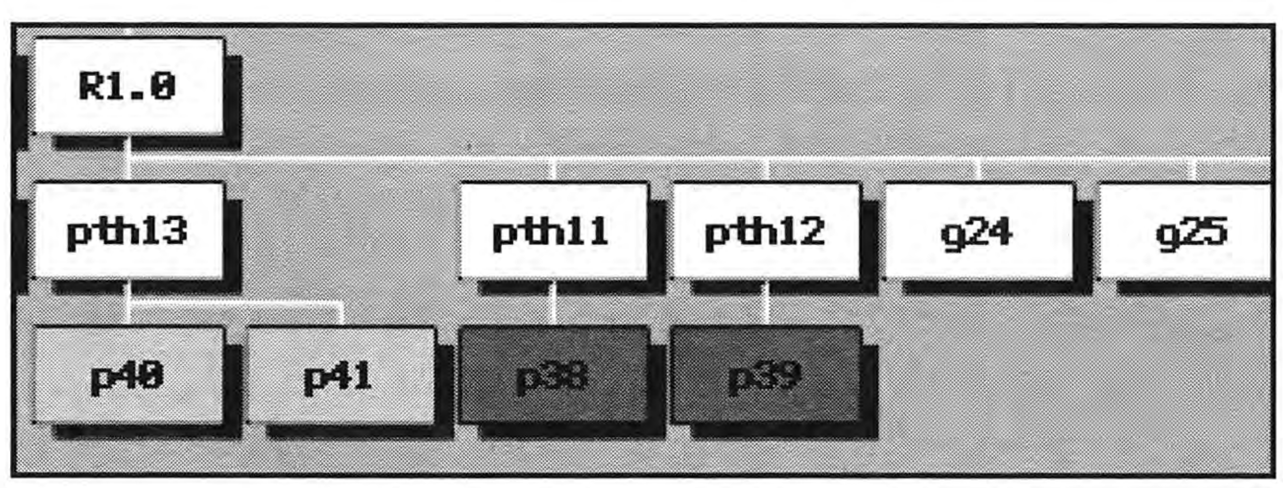


Figure 18. Presagis Creator: Road Node in the hierarchy database.

The Path File Generate function provided in Road Tools can write either the road centerline or scenario lanes. Because this project's goal was to develop a methodology that works with SimVista™, which only needs the road centerline and lane offset parameters for the ISA VRML file, the path file for the road centerline is sufficient for further simulation. The method to write the centerline path file is to select the road section in road mode by selecting the road node or the road section's first path node in the Hierarchy view. Then the user executes the Write Path function in Road Tools.

CONVERSION OF OPENFLIGHT FILE TO STANDARD VRML FILE

As Figure 8 illustrates, once a visual 3D model has been created in Presagis Creator and saved as an OpenFlight file, it needs a series of conversions to be prepared for use in SimVista™.

The next step in the process is to convert the file in OpenFlight format to standard VRML format. Realtime Technologies, Inc., developed *xlate.exe* to carry out this conversion. The *xlate.exe* program can convert the data in OpenFlight format into the standard VRML format, which is also a widely accepted format for 3D modeling. The textures referenced in the OpenFlight file are also converted into the PNG format. The *xlate.exe* program can be run from the command line for the individual file, or a batch file can convert all files in the folder. It also provides a graphical user interface for the user. The command line flags of the program are shown in Figure 19.

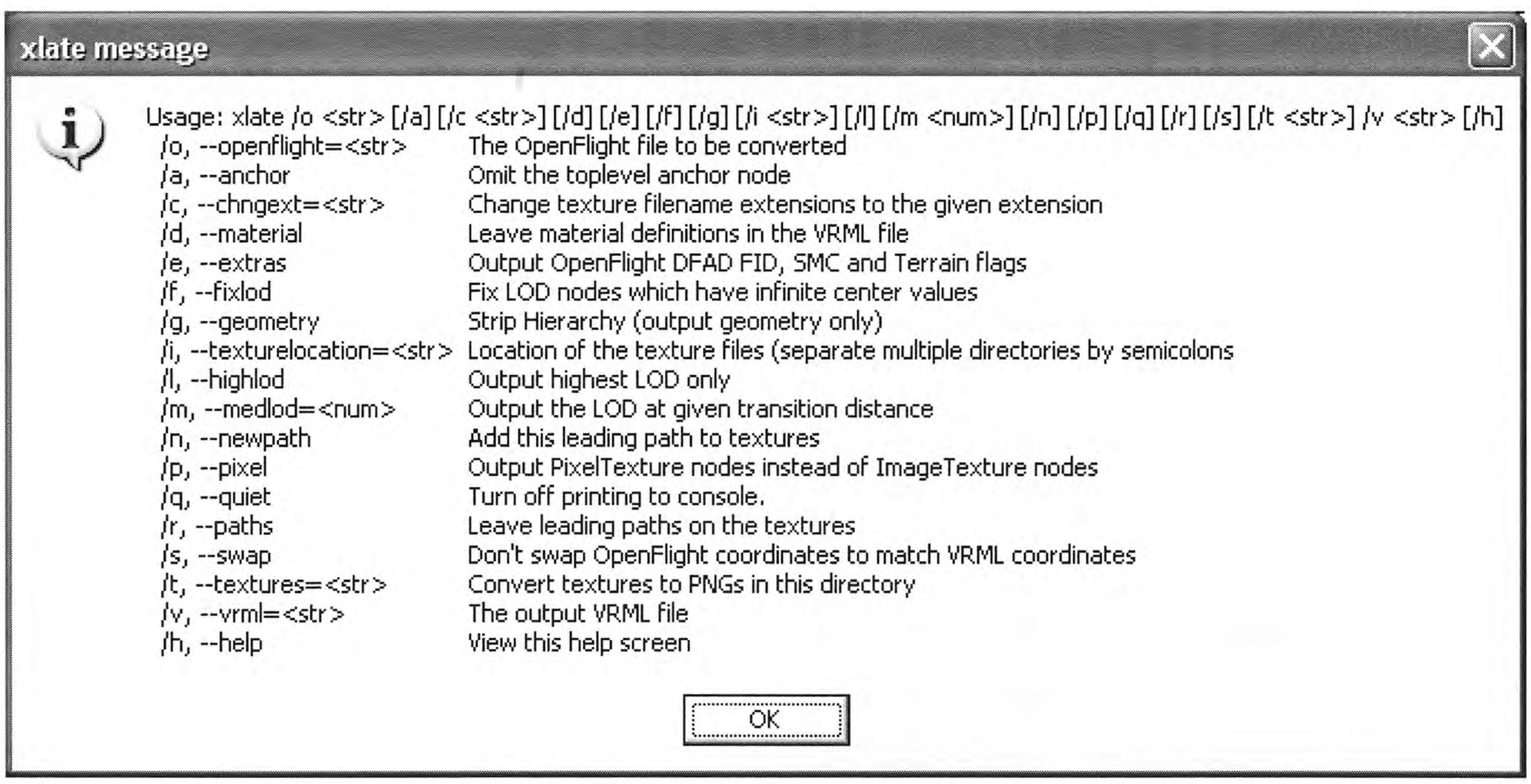


Figure 19. Command line flags of *xlate.exe*.

Before executing the conversion, the external reference and translated nodes in the OpenFlight hierarchy dataset have to be transformed by Made Geometry function in Creator since the program cannot recognize these nodes correctly.

CONVERSION OF STANDARD DRML FILE TO ISA VRML FILE

The next step of the process is to convert the standard VRML file generated in the previous step into a customized VRML format ISA can open. The conversion in this step adds specific information into the standard VRML file. A header with the version and exposed fields information is attached. ISA can detect the version information and determine if the tile needs updating or not. After conversion, the new customized VRML object can support specific data elements such as texture mapping. The *vrmlConvert.exe* program, which is also developed by Realtime Technologies, Inc., executes the conversion from standard VRML format to customized ISA VRML format. The new VRML file does not contain any path information.

CONVERSION OF PATH FILE TO VRML FILE

The path file generated by Presagis Creator Road Tools only contains the information about the route along the roadway geometry indicating autonomous vehicles controlled by the simulator driving route. In driving behavior and human factor studies, the autonomous vehicles are significant to the environment and interaction with the driver. Depending on the complexity of the scenario, it could be very tedious and time consuming to input vehicle control information manually in ISA. The generated path file contains a list of points that define and control the location and orientation of the routes in the scenarios. To utilize the path file in ISA, it must be converted into a VRML file containing all path control points in the order that traffic should flow.

The path conversion tool, *pathConvert.exe*, developed by Realtime Technologies, Inc., accesses the path file and generates a new VRML file containing the path information that can be used by ISA.

COMBINATION OF ISA VRML FILE AND VRML PATH FILE

As a result of the previous two steps, there are two VRML files: one is in a customized VRML format specifically designed for ISA describing the appearance of the visual model, and the other is a VRML file containing information about the paths and routes, along with the roadway, in the model. These two VRML files can now be combined to be opened in ISA as a single file. The specific tool provided by Realtime Technologies, Inc., *tileCombine.exe,* combines these files.

ADJUSTMENT OF DATA IN COMBINED ISA VRML FILE

In previous steps, a series conversion of the model file from the OpenFlight format to the customized VRML format with path information specialized for ISA was performed. Once the final VRML file with ISA is open, the user usually finds that some elements need adjustment because of errors introduced during conversion.

First, the final VRML file loses the directories of textures during conversion. There are two kinds of texture directory modes supported in the OpenFlight texture palette: relative and absolute. Texture directory information will be lost in either of the modes, and therefore manual editing is unavoidable.

Second, the value of coordinates is slightly changed during conversion. Possibly caused by the algorithm of the conversion, the values of coordinates of control points slightly change compared to those of the coordinates in the path file. The errors are acceptable because they are less than a thousandth of a millimeter. Since ISA does not provide an efficient method to access the coordinate information of the visible object, visible polygons may or may not be affected by conversion.

Third, road spline parameters need adjustment since some information is not provided in the path file. Numerous parameters are associated with each road spline, which controls the routes of autonomous vehicles in traffic flow. These parameters are based on Presagis Creator Road Tools, and SimVista™ only utilizes some of the parameters in the simulator currently. Although the only parameters currently used by the simulator are required in the tile, researchers recommend inputting valid values for parameters that may potentially be used in the future. The properties *(19)* are defined as shown in the Table 5. As mentioned in previous sections, ISA only supports road centerline and has its own scenario lane definition. The item LANE_OFFSET is the parameter used to define the scenario lane. These parameters need adjustment according to the requirement of the new tile as well as NUM_LANES since they are correlated. Other parameters may also need special consideration due to errors in conversion that are not mentioned in Presagis Creator Road Tools documentation. The errors noted by the research team are included here.

Table 5. Definitions of the properties in the road spline.

Parameter	Definition
ROAD_TYPE = Curve	Not used.
ARC_RADIUS = 0.10000	Not used.
SPIRAL_LEN1 = 0.00000	Not used.
SPIRAL_LEN2 = 0.00000	Not used.
SUPERELEVATION = 0.00000	Not used.
CONTROL_POINT = 0.0000-3.5000	Not used.
VCURVE_LEN = 400.00000	Not used.
VCURVE_MIN = 20.000000	Not used.
SLOPE1 = 0.000000	Not used.
SLOPE2 = 0.000000	Not used.
SPEED = 24.6	Specifies speed that traffic will attempt to drive on the roadway. This is currently interpreted by the Realtime Technologies, Inc. simulator in meters/second (m/s).
NO_PASSING = TRUE	Not used but may be used in the future.
STORE_HPR = FALSE	The following when true means the control points are in X, Y, Z, Heading, Pitch, and Roll. When false, it means that the control points are X, Y, Z, Xnorm, Ynorm, and Znorm. The simulator only supports STORE_HPR: FALSE.
NUM_POINTS = 7	Must match the number of control points that make up the road.
LEFT_SHOULDER = 3.6	Not currently used but should define the width of the left shoulder in meters. The shoulder starts beyond the road WIDTH (in conjunction with CENTER2LEFT).
NUM_LANES = 2	Specifies the number of lanes on the road. It is used to calculate the width of the lanes and also to read the land offset information. There should be one LANE_OFFSET line for each lane.
LANE_OFFSET = −1.8000000	Specifies the offset from the center of the road to the center of the lane in meters. It is assumed that the LANE_OFFSET is always relative to zero for the center of the road. Negative lane offsets are assumed to be in the negative travel direction, and positive lane offsets are assumed to be in the positive travel direction. The smallest positive lane offset is used to determine the position of all positive lanes, and the smallest negative lane offset is used to determine the position of all negative lanes. Each lane center after that is assumed to be the value of first LANE_OFFSET + WIDTH or the value of first LANE_OFFSET - WIDTH
LANE_OFFSET = 1.8000000	Same as above.
SHOULDER = LR	Used to describe whether there is a shoulder on the left and/or right.
UNIDIRECTIONAL = FALSE	Defines one-way roadways.
CENTER2LEFT = 3.600000	Specifies the distance from the centerline to the left-most edge of the road in meters. Currently this is not used in the simulator.
LANE_DISTRIBUTION_ FACTOR = 0.1	Describes the variation in lane position offset. The actual position of a vehicle is the LANE_DISTRIBUTION × lane width × vehicle desired lane position. This is normalized by the lane width. The vehicle desired lane position ranges from −1 to 1.
SURFACE = PAVEMENT_SMOOTH	Not currently used. It could be used in the future by a road-based terrain query to specify the terrain type of the road.
WIDTH = 7.200000	Specifies the total width of the road in meters. This is currently used to calculate the lane width. Currently it is assumed that all lanes have the same width, which is equal to WIDTH/NUM_LANES.

Note: These parameters are in text format.

TRAFFIC LOGIC AND DRIVING BEHAVIOR

In previous sections, the methodologies to develop tiles with visible objects and road splines directing the route of traffic flow were presented. In driving simulation, immersion and veridicality are the dimensions used to evaluate the reality provided by the system. High fidelity of the models including objects and scene can contribute to the realism. On the other hand, interaction also plays a significant role in the driving simulation since the system is expected to present a dynamic environment that contains the elements interacting with the driver and affecting the driving behaviors. These interactive elements include autonomous vehicles and pedestrians controlled by the simulator or the vehicle driven by another user in the case of a distributed system.

The vehicle driving logic and vehicle characteristics, which are the core algorithm in simulation systems, are not contained in the scenario model file. These files only include specific traffic flow logic that tells the vehicles the route, orientation, and direction to drive. The logical elements included in customized VRML files are correlated to the visible geometry and features of the road section, for example, the intersection, merge lane, and ramp. It means the parameters in the scenario file cannot manipulate single vehicles; rather, they just arrange routes with specific rules directing all the vehicles generated. Each vehicle in the traffic flow follows the route and rules defined in the scenario file according to the inner driving logic and behavior model.

ISA provides a number of categories of correlated data that include control points, road splines, road connectors, and semaphores.

Control points are the type of correlated data used to support road splines. At least two control points are needed to create a road. In fact, SimCreator® only recognizes the path defined by three or more control points, which means that even a straight path needs three control points to be defined. Traffic flow driving on the road defined with control points will move consequently from one point to the next one. As shown in Figure 20, defining a path for curves requires more control points than for straight road segments, and more control points lead to a smoother driving path, while the characteristics of the vehicle also affect the vehicle's track.

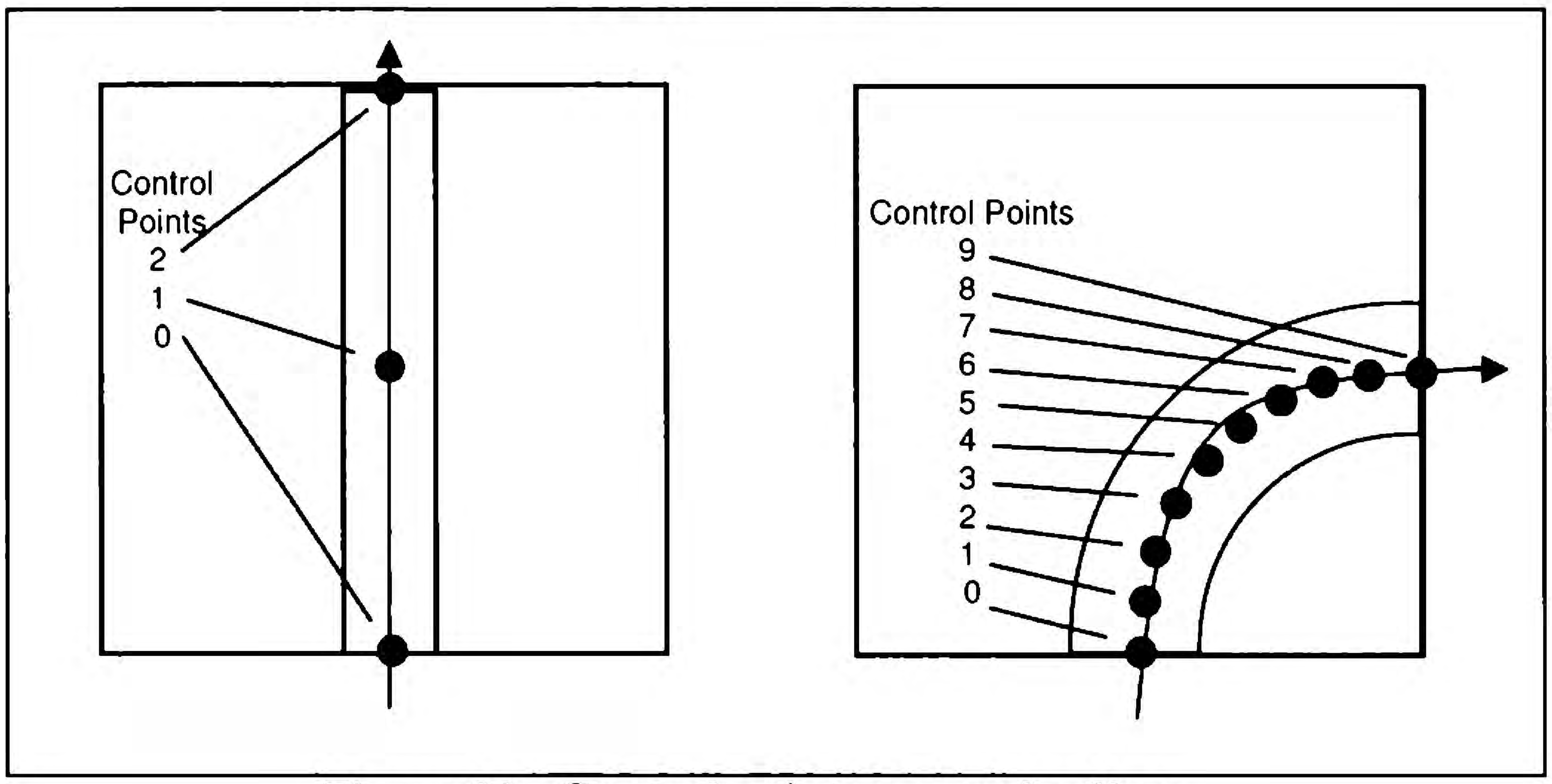


Figure 20. Control points on road splines.

Road splines are a collection of control points. A spline can include as many as 26 parameters associated with it. The roadway splines define the centerline of a roadway on which traffic drives. Every autonomous vehicle has to drive following the road splines and reach the control points on it one by one. Without road splines, the driving behavior logic will instruct autonomous vehicles that there is no road available in the scenario.

Semaphores are the type of correlated data that controls the traffic flow logic at intersections. A semaphore can work at signalized intersections or unsignalized intersections with or without stop signs. The definitions of the properties in a semaphore are shown in Table 6 for a signalized intersection and in Table 7 for an unsignalized intersection *(19).*

Road connectors are used to define the route on which the vehicles move from one road spline to another. Without road connectors, even if the road splines are physically connected, autonomous vehicles will stop at the end of the spline. Where two roadways interact, a set of connectors needs to be created for each possible route the traffic can flow.

Table 6. Definitions of the properties in the signal semaphore.

Parameter	Definition
Name	Specifies the name of the semaphore. The user may use whatever he or she wishes.
Owner	Specifies the parent object that the semaphore is placed in. This is usually the tile name.
Abilities	Not used.
direction0greenTime	Specifies the duration of the green-light phase for the 0 direction in seconds.
direction0yellowTime	Specifies the duration of the yellow-light phase for the 0 direction in seconds.
direction1greenTime	Specifies the duration of the green-light phase for the 1 direction in seconds.
direction1yellowTime	Specifies the duration of the yellow-light phase for the 1 direction in seconds.
type	Specifies the type of semaphore. For signal semaphore, it should be SIGNAL (stop light).

Table 7. Definitions of the properties in the stop semaphore.

Parameter	Definition
Name	Specifies the name of the semaphore. The user may use whatever he or she wishes.
Owner	Specifies the parent object that the semaphore is placed in. This is usually the tile name.
Abilities	Not used.
type	Specifies the type of semaphore. For signal semaphore, it should be STOP (stop sign).

ISA provides Correlated Data Tools in the Object Gallery where the control point, road spline, and road connector objects are stored. Similar to other objects in the gallery, the placement method is the same drag, release, and edit. Except for the control point and semaphore, other data types are invisible.

Generally, autonomous vehicles drive following the control points on the road spline except when they are moving from one road spline to another at the location of an intersection, lane merge, or ramp. Road connectors are used to connect two road splines and guide vehicles to drive on the correct route. The definitions of the properties *(19)* in the road connector are shown in Table 8. According to the properties setting in ISA, the road connector can indicate the vehicle at a specific control point on the source road spline to drive to another specific control point on the next road spline. As shown in Figure 21, there are three significant properties to define the road connector (a detailed explanation is in Table 8): roadOneEndControlPoint, roadOneStartControlPoint, and roadTwoControlPoint. These properties direct the simulation system concerning how to move vehicles from one lane on one road to one lane on another road. One road connector only works with two corresponding lanes.

Table 8. Definitions of the properties in the road connector.

Parameter	Definition
Name	Specifies the name the user has given the road connector.
Owner	Specifies the name of the group this Road Connector belongs to in Scene Tree.
Abilities	Usually none.
changeProbability	Specifies the inverse change probability or the inverse likelihood that a vehicle will take a given connector. 0.0 = Always, and 1.1 = Never take the connector.
direction	Specifies an integer that will control when the light switches if there is a semaphore. All connectors assigned to the direction number will respond to the same light phasing. When 0 (horizontal road [road A]) is active, 1, 2, 3...n are stopped. When 1 (vertical road [road B]) is active, 0,2,3,4 ...n are stopped. Connectors with common traffic flow sequences should be assigned a common direction value. With a semaphore type of SIGN, traffic flowing on the connector defined as direction 1 will stop at the intersection, and traffic flowing on connectors defined as direction 0 will not stop. This is how the user defines stop sign intersections where not all legs have signs assigned to them.
roadOneEndControlPoint	Specifies the last control point of the source road. It should be in the middle of the semaphore. See Figure 21 for an example of where to locate this point. If no semaphore is used (i.e., there is no intersection), this is the last point on road 1 before road 2.
roadOneEndLane	Specifies to which lane the Road Connector connect at the last control point of the source road. Lanes are numbered from 1 to n from the center of the road out. They are either positive or negative based on whether they follow the flow direction of the road relative to the control points. This is the lane definition for the segment of the source road.
roadOneStartControlPoint	Specifies where models start to evaluate the status of the semaphore if they need to break or change speed to enter the turn. See Figure 21 for an example location. If no semaphore is used (i.e., there is no intersection), this is the point where the user would like road 1 to be evaluated. This point must be far enough back from the semaphore so that the entity has time and distance to stop. $d = v*v/2a$, where d = distance (meters), v = velocity (m/s), and a = acceleration (m/s/s).
roadOneStartLane	Normally 1/−1 unless there are multiple turn lanes.
roadTwoControlPoint	The model will drive (if it decides to turn) from roadOneEndControlPoint to this control point. This point should be inside the semaphore, usually a symmetrical distance from the center of the semaphore. See Figure 21 for an example of where this control point would be located. If no semaphore is used (i.e., there is no intersection), this is the point where the user would like road 1 to join road 2. The user must specify the road 2 connectors such that the new road cannot be confused with any other road created around that point.
roadTwoLane	Usually 1/−1. The sign is determined by the direction of the control points on the destination road.
semaphore	Specifies the name of the semaphore that was added earlier to control this connector.
stopPosition	Specifies the distance from the point between the start and end position where the vehicles should stop in a stop-flow condition. Normally this is matched to the location of the stopping stripe for the intersection. This number is in meters. A positive (+) sign means the vehicle stops closer to the end control point, and a negative (−) sign means the vehicle stops closer to the start control point.
turnSpeed	Specifies the speed (m/s) that vehicle models should drive the turn connector.

Table 8. Definitions of the properties in the road connector dialog box.

Parameter	Definition
type	Specifies the type of connector. The value has implications for what type of roadway connection is being made and how scenario control will evaluate the list of connectors for each inflow lane(s). Vehicles can only be in one connector at a time. Therefore, proximity of connectors to ensure the vehicle has exited the previous connector is required. See the following for a more detailed explanation: • INTER—Standard intersection connector. Vehicles entering the connector will either attempt to turn or continue straight. Vehicles that attempt to turn left will time out if the light turns yellow or if the vehicle must wait too long. • INTERFORCED—Forced intersection connector. When a vehicle enters an INTERFORCED connector, if the vehicle decides to continue straight, the vehicle is immediately removed from the connector, and additional connectors at the road control point are searched. If the vehicle decides to turn, it will always turn and never time out. The following are two examples when INTERFORCED is needed: At a T-intersection, the intersecting road must have an INTERFORCED connector for a left turn to guarantee that the left turn is completed. On a two-lane road where a single lane can turn left or right, the INTERFORCED connector is used for the first connector (for the right turn). If a right turn is not chosen, the vehicle will leave this connector and search the second connector (in this case the second connector would be a left turn with type INTER). It is important to have the second connector on the same road control point. • CONNECT—Used to wire two roads together. If road2lane is zero, then the vehicle uses its current LANE. Once road 1 ends, the connector is complete. • RIGHTMERGE—Onramp merge or merge when a lane is dropped from a roadway. The vehicle is placed in both roads so that room is made for it on road2. It is assumed that road1 will cross into road2 at some point. The connector is complete when the vehicle driving on road1 ends up inside road2lane on road2. The vehicle then starts driving on road2. It can also be used for left merges. • STOP—Stop sign on current road. The vehicle will come to a stop and then go with no cross traffic. All road2 data are ignored. There is no semaphore (connector set) data. • ENDROAD—The vehicle that reaches this connector will be reset into road2road, road2segment, and road2lane. • RIGHTSPLIT—Exit ramp from a freeway. Can also be used for left splits. • ALLSTOP—Currently not implemented.

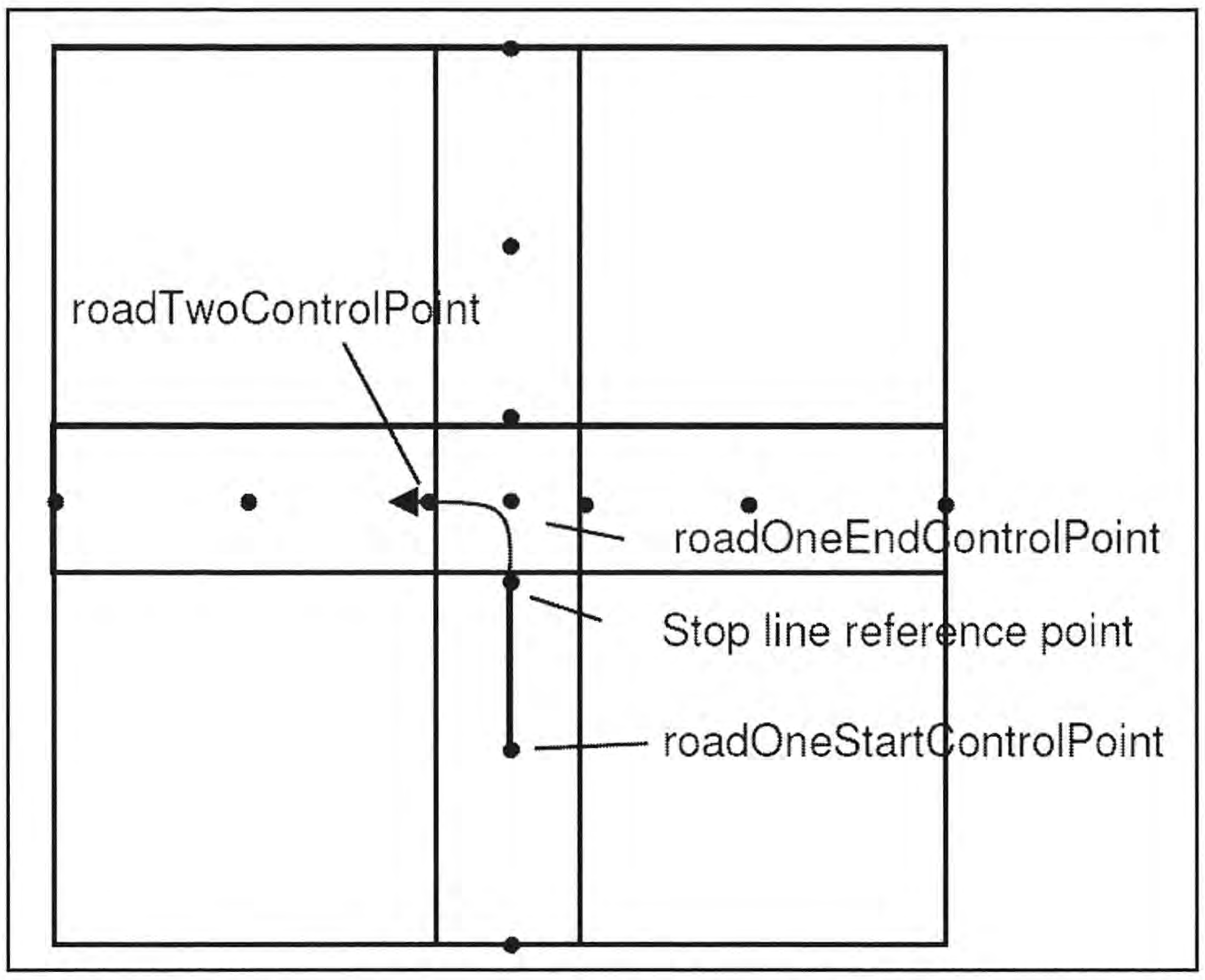


Figure 21. Road connector and correlated control points.

PUBLISHING OBJECT OR SCENE

Once the previous steps have been completed, the new model is prepared to be used by ISA; however, the model is not a tile in the ISA Object Gallery. To become an item in the gallery as a released model and scene, it has to be published.

Before publishing the new tile, it should be saved as a SaveAsObject version of the object since the published tile in the gallery cannot be modified anymore. The SaveAsObject version of the model can be loaded again and modified later if necessary. This also allows loading prepublished modifications to the original data and republishing in the gallery if required.

The final step in the procedure is to publish the tile for use in the ISA Object Gallery. This step optimizes the VRML structure of the tile for use in SimCreator®. Publishing the version prevents any modification of individual elements or correlated data.

RESULTS

Based on the methodologies developed in this project, TTI duplicated a number of official tiles in the ISA gallery released by Realtime Technologies, Inc., for testing. Comparison of the copies to the originals shows that the duplicated scenarios and objects are consistent in appearance and traffic logic. These copies can be used in ISA in the same way as the original tiles in the gallery or instead of the gallery tiles, as shown in Figure 22.

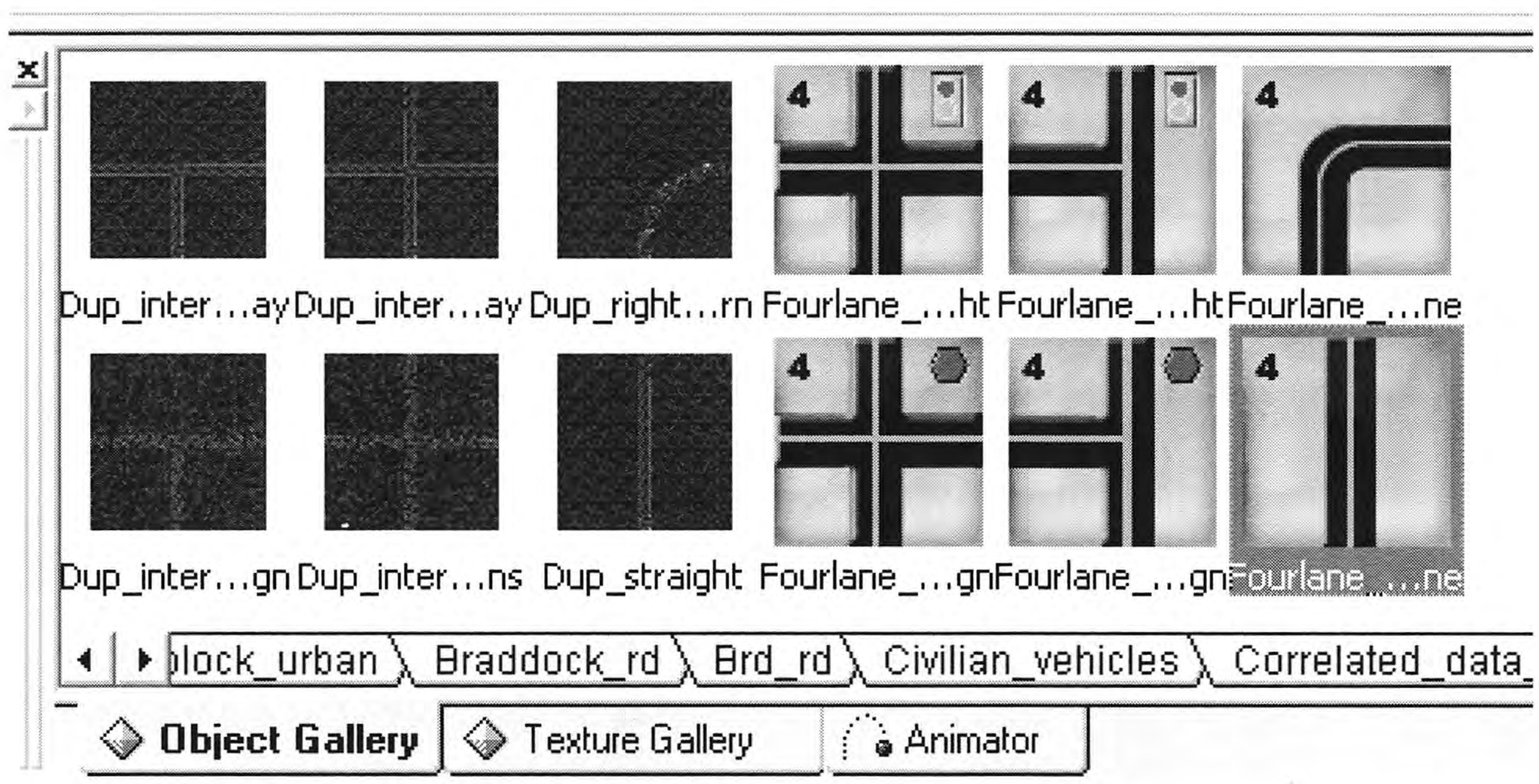


Figure 22. Original tiles and duplicated tiles.

The new tiles, designed to be compatible with existing tiles, present consistent fidelity with the existing tiles and can be used to construct a wide range of driving environments in combination with the existing tiles. As shown in Figure 23, the roadway connection between new tiles and existing tiles is seamless.

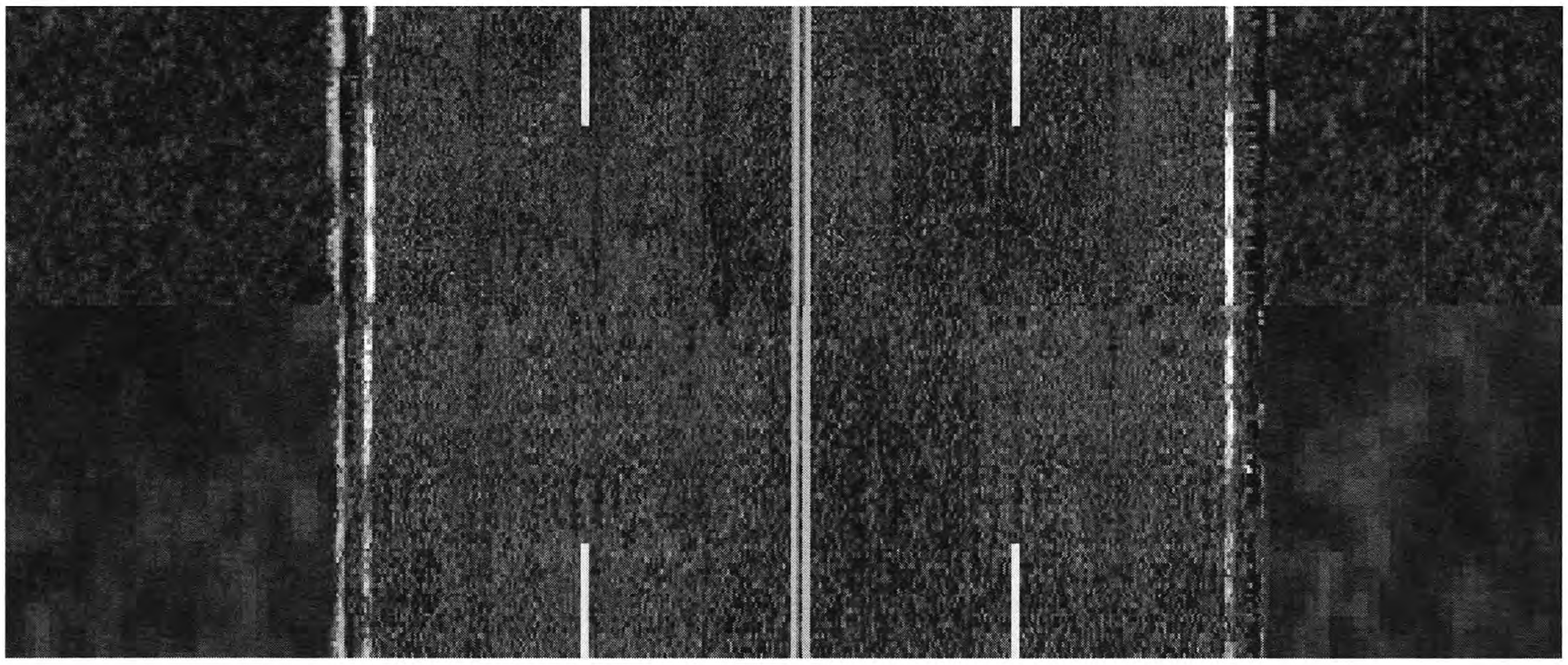


Figure 23. Connection between new tile and existing tile.

The driving environments, consisting of new tiles developed with the methodology, operated correctly in the driving simulation system, as shown in Figure 24. The traffic logic of autonomous vehicles executed correctly in the scenarios as researchers required them to. The driving behavior information can be recorded correctly in the simulation system when the simulation runs.

Figure 24. Driving environments consisting of new tiles.

CONCLUSIONS

In this project, a methodology was developed to design 3D virtual scenario models that are able to be simulated based on roadway design data for specific research. The virtual driving environment generated based on this methodology can provide acceptable fidelity and immersion. The traffic logic included in the scenario models was proven to correctly control the traffic flow in the virtual world at run time. Given the diverse requirements and purposes of the scenarios models, the methodology can also be utilized to design various road sections for corresponding requirements in specific research that uses the driving simulator.

HIGHLIGHTS

Highlights of the methodology are listed below:

- The 3D models generated can satisfy various requirements since the methodology is flexible.
- The 3D models generated are practical and executable in the simulation system since they are reasonably simplified to reduce resource requirements.
- The methodology is easy for researchers to implement with limited knowledge of transportation engineering and programming skills.
- The procedure of the methodology only involves two commercial packages, which means the cost of the software will be low.
- The traffic logic data in the model files do not depend on the geometry feature, which means complex traffic flow control is achievable.
- The methodology can generate models compatible with existing official models in the gallery.

LIMITATIONS

The limitations of the methodology are listed below:

- Using the methodology is time consuming when designing complex scenario models because of manual editing.
- Adjustment of the properties in the final VRML file is necessary.

- Because of simplification, the current methodology cannot provide fidelity as high as that in the original methodology presented in the project proposal. It also cannot utilize the road data designed in AutoCAD® Civil 3D®.

LESSONS LEARNED

The lessons learned during this project are listed below:

- High-fidelity models and high-resolution textures are not practical for TTI's driving simulation system because of the high requirements of hardware resources.
- Significant diversity exists in various model formats for different systems and data-processing and storage methods for different simulation systems.
- If the model can satisfy the requirement and purpose of the research applying the scenarios, the model can be simplified to enhance the featured characteristics.
- A comprehensive study of accessing data in DWG format will overload a one-year project unless it is based on commercial and non-free source codes.

THE NEXT STEP

The methodology developed suits TTI's needs and the level of programming skill of its staff. Since most of TTI's work in the simulator is focused on driving behavior, the minor errors in the roadway tiles are not critical in these types of experiments. The simulator, however, can also be used to provide visualization of new roadway designs. For these types of applications, it is more critical that the exact geometry be rendered accurately. The next step in this research is to continue working on the original methodology mentioned in the project proposal, even though it is doubtful that the expected resulting model is suitable for specific topics studied using driving simulation. Developing the algorithm and application to automatically model roadways based on design data in AutoCAD® Civil 3D® is meaningful for roadway design evaluation and visualization.

REFERENCES

(*1*) Chrysler, S.T., J Wright, and A.A. Williams. *3D Visualization as a Tool to Evaluate Sign Comprehension.* Publication SWUTC/04/167721-1. Southwest University Transportation Center, Texas Transportation Institute, 2004.

(*2*) Chrysler, S.T., A.A. Williams, and E. Park. *Traffic Engineering Applications of Driving Simulation.* Publication SWUTC/06/167142-1. Texas Department of Transportation, 2006.

(*3*) Ullman, G.L., B.R. Ullman, C.L. Dudek, A.A. Williams, and G. Pesti. *Advanced Notification Messages and Use of Sequential Portable Changeable Message Signs in Work Zones.* Publication FHWA/TX-05/0-4748-1. Texas Department of Transportation, 2005.

(*4*) Ullman, G.L., M.D. Finley, A.M. Pike, K.K. Knapp, P. Songchitruksa, and A.A. Williams. *Studies to Improve Temporary Traffic Control at Urban Freeway Interchanges and Pavement Marking Material Selection in Work Zones.* Publication FHWA/TX-08/0-5238-2. Texas Department of Transportation, 2008.

(*5*) Klee, H. *Microscopic Car Modeling for Intelligent Traffic and Scenario Generation in the UCF Driving Simulator Year 1.* University of Central Florida, School of Electrical Engineering and Computing Science, 2000.

(*6*) Laberge. J., C. Scialfa, C. White, and J. Caird. Effects of Passenger and Cellular Phone Conversations on Driver Distraction. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 1899, Transportation Research Board of the National Academies, Washington, D.C., 2004, pp. 109-116.

(*7*) Andersen, G.J., C. Sauer, and A. Saidpour. Visual Information for Car Following by Drivers: Role of Scene Information. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 1899, Transportation Research Board of the National Academies, Washington, D.C., 2004, pp. 104-108.

(*8*) Jenkins, J.M., and L.R. Rilett. Application of Distributed Traffic Simulation for Passing Behavior Study. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 1899, Transportation Research Board of the National Academies, Washington, D.C., 2004, pp. 11-18.

(*9*) Kelly, M.J., S. Lassacher, and Z. Shipstead. *A High Fidelity Driving Simulator as a Tool for Design and Evaluation of Highway Infrastructure Upgrades.* Publication FHWA/MT-07-005/8117-33. Montana Department of Transportation, 2007.

(*10*) Olstam, J.J., J. Lundgren, M. Adlers, and P. Matstoms. A Framework for Simulation of Surrounding Vehicles in Driving Simulators. *ACM Transactions on Modeling and Computer Simulation*, Vol. 18, No. 3, 2008, pp. 9:1-9:24.

(*11*) Cha, M., J. Yang, and S. Han. A Hybrid Driving Simulator with Dynamics-Driven Motion and Data-Driven Motion. *SIMULATION*, Vol. 84, No. 7, 2008, pp. 359-371.

(*12*) M Cha, M., J. Yang, and S. Han. An Interactive Data-Driven Driving Simulator Using Motion Blending. *Computers in Industry*, Vol. 59, No. 5, 2008, pp. 520-531.

(*13*) Inman, V.W., G.W. Davis, I. El-Shawarby, and H. Rakha. *Test Track and Driving Simulator Evaluations of Warnings to Prevent Right-Angle Crashes at Signalized Intersections.* Publication FHWA-HRT-08-070. Federal Highway Administration, Turner Fairbank Highway Research Center, 2008.

(*14*) Yan, X., M. Abdel-Aty, E. Radwan, X. Wang, and P. Chilakapati. Validating a Driving Simulator Using Surrogate Safety Measures. *Accident Analysis and Prevention*, Vol. 40, No. 1, 2008, pp. 274-288.

(*15*) Dutta, A., D.L. Fisher, and D.A. Noyce. Use of a Driving Simulator to Evaluate and Optimize Factors Affecting Understandability of Variable Message Signs. *Transportation Research Part F: Traffic Psychology and Behaviour,* Vol 7, No. 4-5, 2004, pp. 209-227.

(*16*) *A Policy on Geometric Design of Highways and Streets*. American Association of State Highway and Transportation Officials, Washington, D.C., 2001.

(*17*) *Manual on Uniform Traffic Control Devices,* U.S. Department of Transportation, Federal Highway Administration, Washington, D.C., 2003.

(*18*) *Creator User's Guide*. Presagis, Montreal, Quebec, Canada, 2009, pp. 117-1987.

(*19*) *SimVista Tile and Object Development and Maintenance Document.* Realtime Technologies, Royal Oak, Michigan, 2008, pp. 3-48.