WP #2002-2

Introduction to Creating HVE® Environments with Rhinoceros®

James P. Sneddon
Northwestern University

Abstract

Since HVE®'s introduction, users have employed third party software to create environments for import. Numerous computer aided drafting and 3-D modeling programs are available, and the user's choice is influenced by many factors. Some of the more significant factors concerning HVE users are:

- Surface tools
- Seams between surfaces
- Surface normals
- File compatibility

Modeling an HVE environment requires three dimensional point data that is typically acquired from a site survey.
was intended to import surfaces constructed from this data, using either computer aided drafting or 3-D modeling software. Creating surfaces that accurately match the point data is largely dependant upon the software's surface tools. Similarly, ensuring "watertight" seams between adjacent surfaces is a function of the software's capability.

Additionally, the direction of surface normals is critical to HVE calculation models. Surface normal directions are established by the CAD or 3-D modeling software. The method by which normals are oriented vary among programs and file formats, and have caused problems for HVE users. Finally, the software must be able to export a file format supported by HVE. If not, a translator must be available, or another piece of software purchased solely for this purpose.

This whitepaper provides an introduction to using Rhinoceros® (a.k.a. Rhino) as it applies to these issues.

Introduction

Rhinoceros is a three dimensional NURBS modeling program. NURBS (Non-Uniform Rational B-Splines) geometry is a mathematical representation that can accurately define any shape, line or surface. The NURBS geometry provides increased flexibility when editing or modifying NURBS objects. Rhino objects include:

- Points
- Curves
- Surfaces
- Polysurfaces
- Solids
- Polygon Meshes

Absent from this list are lines. Since Rhino defines curves by degree, a line is considered a first degree (linear) curve. A circle or arc is a second degree (quadratic) curve. Interpolated curves, commonly referred to as splines by other programs, are third degree (cubic), or higher, curves.

Surface objects (or entities) in Rhino refer to NURBS surfaces. An accurate mathematical representation of a free form smooth surface is possible using NURBS geometry. Additionally, NURBS surfaces can be joined and split. This functionality greatly increases the simplicity of constructing details such as lane lines, drives, and sidewalks, and enables tight seams between adjacent surfaces. Ultimately, polygon meshes are created from the NURBS surfaces, and imported into HVE.

There are numerous surface and editing tools available in Rhino. As a result, the user can construct and modify surfaces by several means.

The methods discussed herein are not the sole means by which an environment can be created with Rhino. However, they have been found to work well with HVE.

Discussing in detail the procedures used to create an environment with Rhino would require a lengthy text. Instead, this paper introduces the user to the usefulness of Rhino in creating HVE environments.

Surface Tools

The ease with which accurate surfaces are created depends upon the surface commands and tools available. These tools also affect the number and type of points collected during a site survey. Rhino has numerous surface commands used to construct, modify, and edit surfaces. Some of the most useful commands for creating HVE environments are introduced here and listed in Table 1.

Accurately modeling the grade and cross slope of a rolling surface can be easily modeled in Rhino. Using the *NetworkSrf* command, an accurate model of a roadway surface can be constructed from a curve network.

A roadway skeleton is constructed using interpolated curves as depicted in Figure 1. The *InterpCrv* command creates an interpolated curve, or spline, that will pass through the survey points.

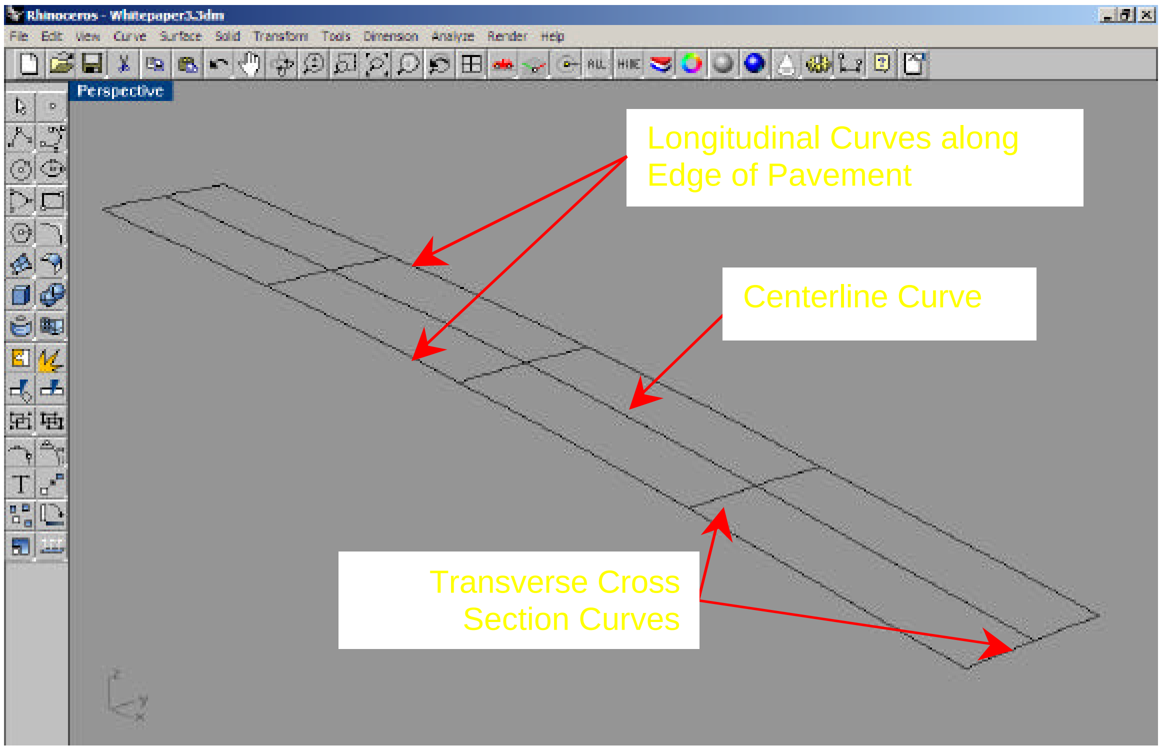

Rhinoceros - Whitepaper3.3dm
File Edit View Curve Surface Solid Transform Tools Dimension Analyze Render Help
Perspective
Longitudinal Curves along Edge of Pavement
Centerline Curve
Transverse Cross Section Curves

Figure 1. *A roadway skeleton is constructed from curves forming the centerline, pavement edges, and cross sections. The curve network contains curves in two directions, longitudinal and transverse.*

COMMAND NAME	ICON	FUNCTION
Dir		Displays direction of NURBS surface normals; flips surface normals.
EdgeSurf		Creates a surface from 2, 3, or 4 edge curves. End points must be coincident or cross.
Explode		Explodes a polysurface into separate surfaces; polylines into line segments; and curves into curve segments.
Extrude		Extrudes a curve into a surface.
Flip		Flips the direction of mesh normals.
InterpCrv		Draws a spline curve through specified points.
Join		Joins objects.
JoinEdge		Joins edges of surfaces that are out of tolerance.
Mesh		Creates a mesh from NURBS objects.
NetworkSrf		Creates a surface from a curve network. Curves are sorted into two orthogonal directions.
Patch		Fits a surface through curves and point objects.
Plane		Creates a rectangular planar surface from specified corner points.
Point	▫	Draws a point object.
Rotate3d		Rotates objects about an axis.
ShowNakedEdges		Displays naked edges of a surface or polysurface.
Split		Splits curves and surfaces with cutting curves, surfaces, and points.
Sweep1		Creates a surface from cross section curves swept along one rail curve.
Sweep2		Creates a surface from cross section curves swept along two rail curves.
TextObject	T	Create text-shaped objects from curves, surfaces, or solids based upon TrueType fonts.
Trim		Trims objects with cutting objects.
Untrim		Untrims a trimmed surface, restoring it to its original shape.

Table 1. *The above table consists of Rhino surface commands and editing tools utilized in creating HVE environments.*

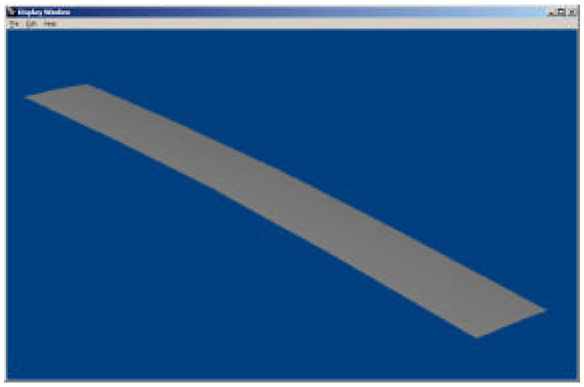


Figure 2. *A surface created with the NetworkSrf command is smooth in both directions.*

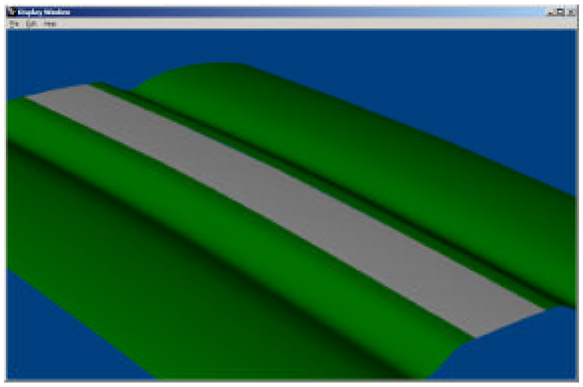


Figure 3. *The pavement edge was used to construct the adjacent surface.*

The *NetworkSrf* command creates a surface from these curves. The resulting roadway surface is smooth in both directions. The curve network must contain curves running in two directions. In the case of the roadway skeleton, one set of curves are longitudinal and the second transverse. All curves in one direction must cross the curves in the other direction. Adjacent surfaces can be constructed using the edge of the first surface as one of the curves in the network as shown in Figure 3.

The *NetworkSrf* command is ideal for sites with no intersecting roadways. Whereas it can be used at intersections, the *Patch* command is often more useful, and may be used for non-intersection sites as well. The *Patch* command fits a surface through the selected points and curves.

If point data is acquired from a total station survey, the output file from the data recorder may already have point objects. If not, points can be quickly created with the *Point* command, and snapping to the objects in the imported file. Curves are required along the edges of the pavement, drives, or any detail that will need to be split from the patch surface. Figure 4 depicts point objects and curves from which a patch surface can be constructed.

The *Patch* command constructs a single surface that includes the intersecting roadway and roadside as shown in Figure 5. The patch surface will pass through the curves and points from which the surface was constructed, creating a smooth surface throughout the environment. Distortions may occur along the perimeter of the patch surface. These areas can be trimmed if they are a problem. However, to ensure an accurate environment model, the surveyor should collect point data beyond the area included in the simulation.

Individual surfaces can be split from the patch surface. The *Split* command is used to split the patch surface into two or more separate surfaces as shown in Figure 6.

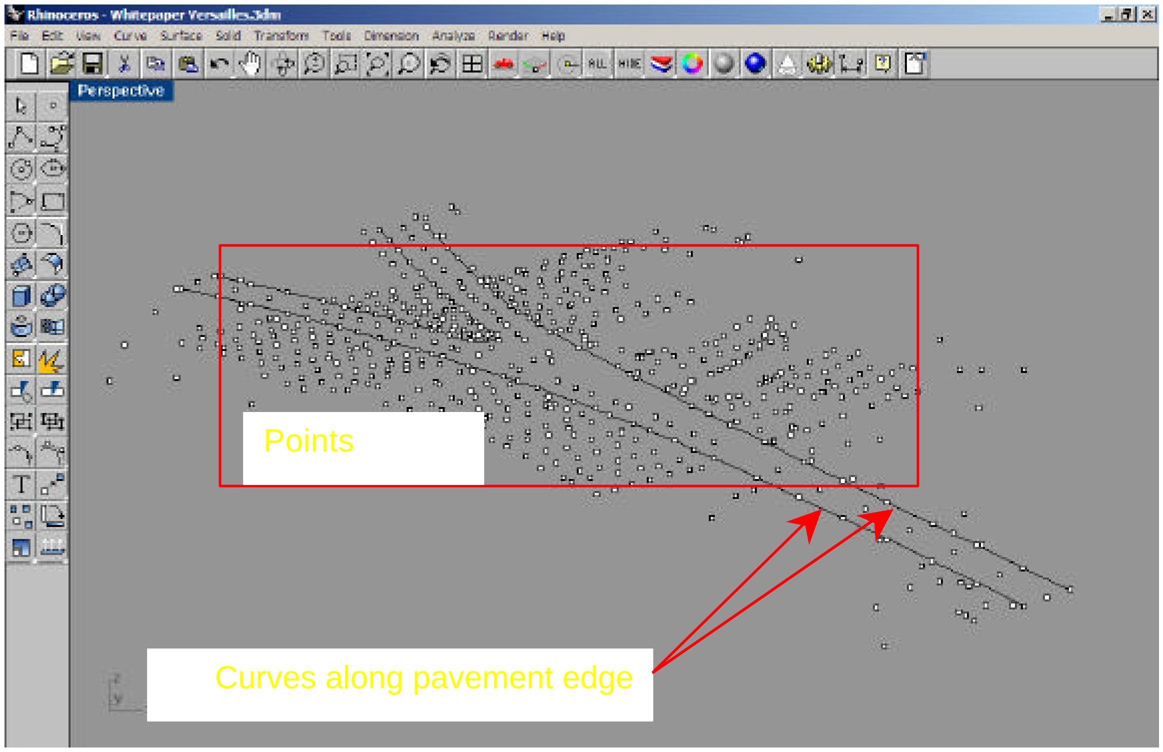

Rhinoceros - Whitepaper Versailles.3dm
File Edit View Curve Surface Solid Transform Tools Dimension Analyze Render Help
Perspective
Points
Curves along pavement edge

Figure 4. *Intersecting roadways are often more easily modeled with the patch command. The above site includes curves along the pavement edges, and point objects from which the patch surface is constructed.*

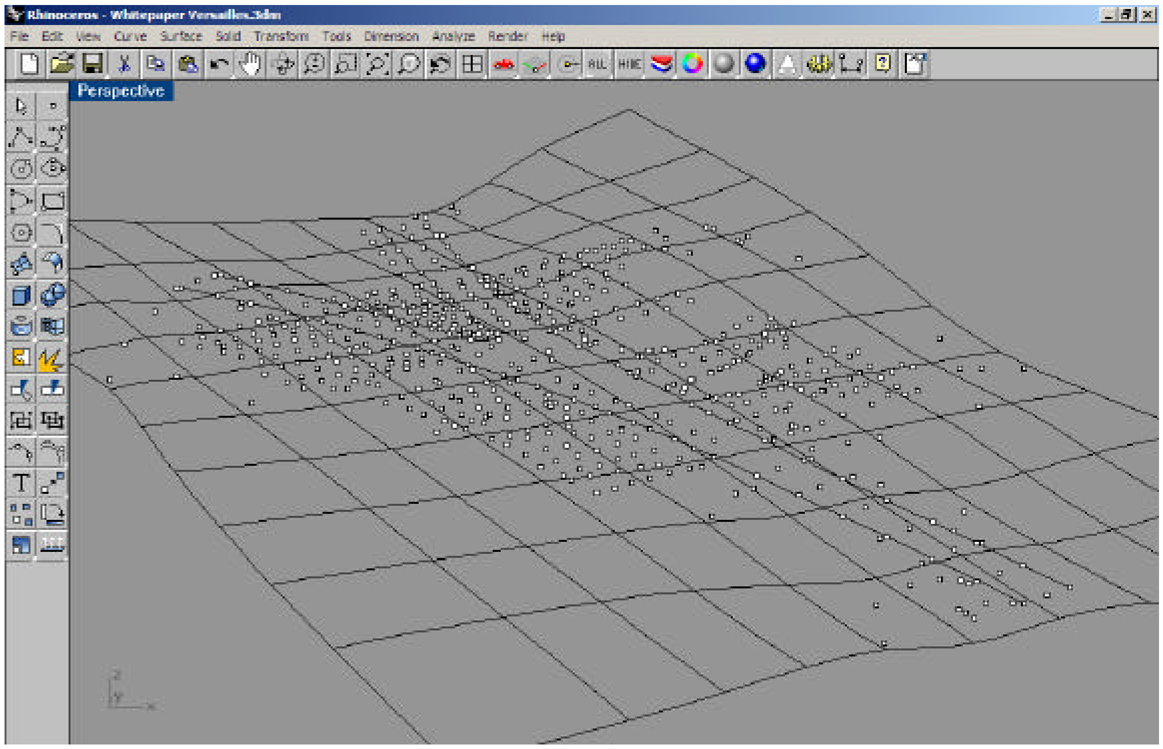

Rhinoceros - Whitepaper Versailles.3dm
File Edit View Curve Surface Solid Transform Tools Dimension Analyze Render Help
Perspective

Figure 5. *The Patch command was used to construct a surface from the points and curves displayed above. The surface can extend beyond the limits of the points.*

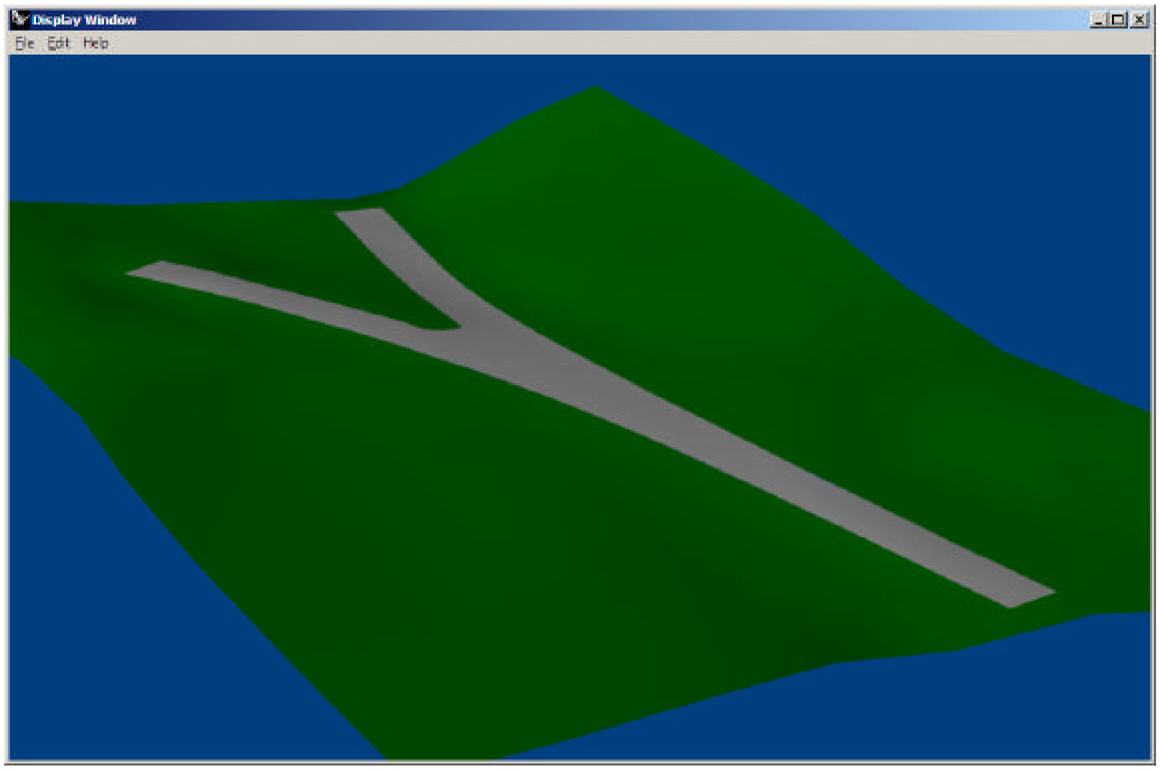


Figure 6. *The patch surface was split into two surfaces, the pavement and roadside.*

If the environment has a rolling terrain with gradual changes in elevation, the entire environment can be split from a single patch surface. However, sharp changes in elevation or vertical faces such as curbs are better modeled with multiple surfaces. The intersection bounded by a curb shown in Figure 7, illustrates a situation where the environment is more accurately modeled from multiple surfaces.

A patch surface is constructed from the curves along the bottom of the curb, and points within the pavement. This surface is trimmed using the *Trim* command so that no part of the surface extends beyond the bottom of curb. A second patch surface is constructed from the top of the curb and roadside points and curves. This surface is trimmed by the top of the curb so that there is no surface over the pavement.

The face of the curb is constructed using the *EdgeSrf* command, which fills the gap between the top and bottom of curb. The *EdgeSrf* command creates a surface between two, three, or four edge curves.

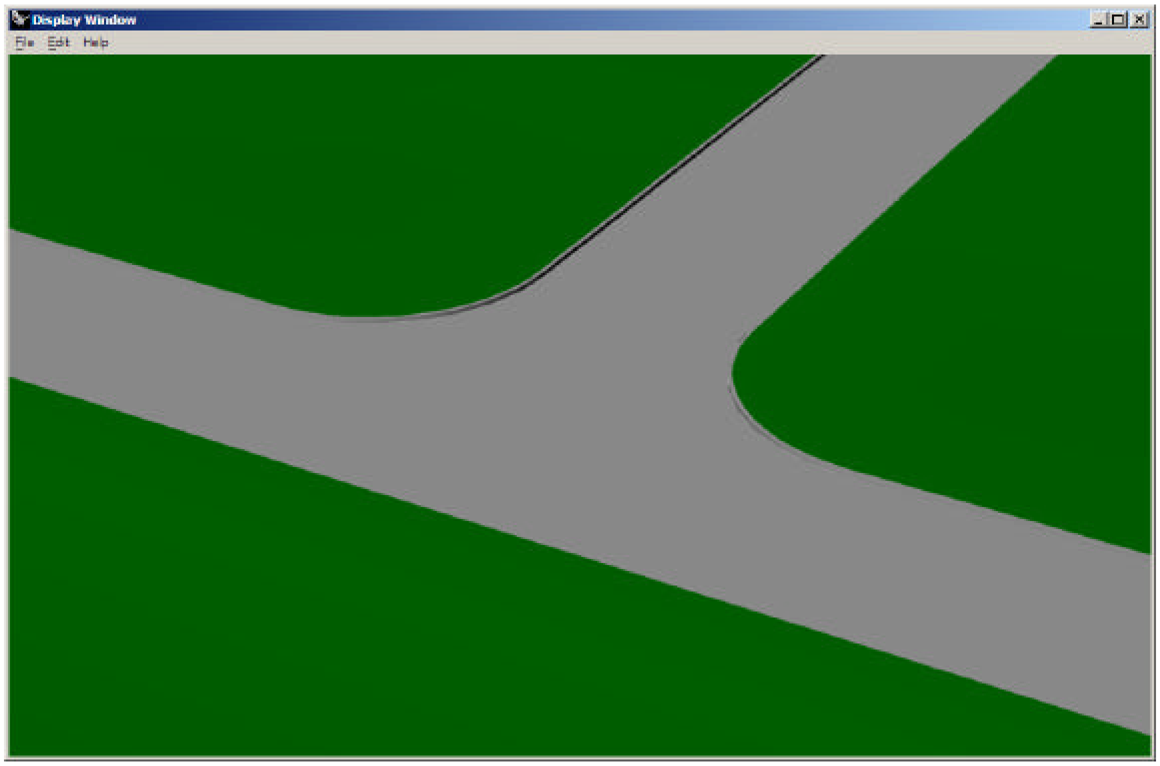


Figure 7. *Two patch surfaces were used to model the above site. One patch surface forms the pavement and bottom of the curb. The second patch surface contains the top of curb and roadside. The vertical face of the curb was constructed with the EdgeSrf command.*

The *Patch* command requires a sufficient number of points or curves to maintain the stability of the surface. Large distances between input points and curves may produce random undulations with large geometries, such as a typical HVE environment. Using curves to define the edges of drives, sidewalks, and other improvements will prevent unwanted undulations.

The *Split* command can be also used to separate drives, sidewalks and other roadside improvements. The cutting edges can be curves, surfaces or polysurfaces. The driveway and sidewalk in Figure 8 were split from the roadside surface.

Lane lines and other pavement markings can also be constructed with the *Split* command. The cutting edges need not be in the same plane as the surface. The edges of lane lines can be constructed from closed polylines in the *xy* plane. The lane lines are split from the pavement surface using the polylines as a cutting edge. The lane lines, stop bar and STOP in Figure 9 were split from the pavement surface.

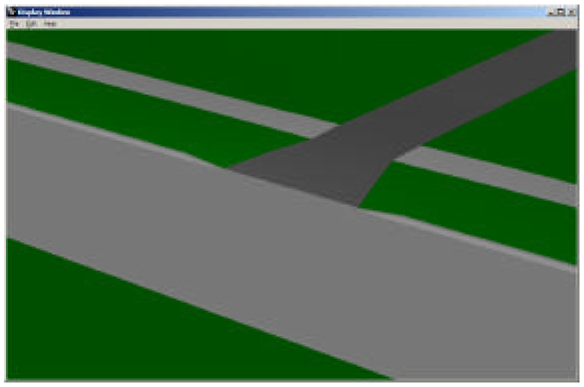


Figure 8. *The driveway and sidewalk was split from a single roadside surface.*

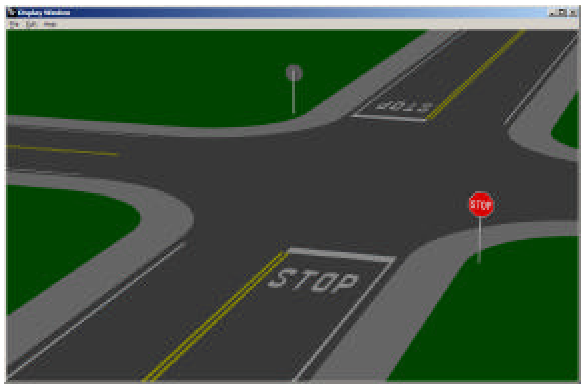


Figure 9. *The yellow barrier lines, fog lines, stop bar and STOP were split from the pavement surface.*

Similarly, roadway evidence such as tiremarks can be split from the underlying surface. If desired, the original surface can be restored with the *Untrim* command. The split surface is retained, and is at the same elevation as the original surface from which it was split. The split surface must be raised so that it is above the original surface, else it will not be visible when rendered.

The ability to split surfaces makes constructing environment details easier, and often reduces the number of points required. This is not the only means by which these details can be modeled. However, ensuring a tight seam between adjacent surfaces may involve additional steps.

A variety of surface tools can be used to create objects such as traffic signals, signs, guard rails, etc. The guardrail in Figure 10 was created with the *Sweep1* command. A cross section curve of the guardrail's profile is constructed. The longitudinal path of the guardrail is established by a rail curve. The curved end is easily modeled in this manner.

Primitive solids can be trimmed to construct objects, or surfaces constructed from edge curves. The *Extrude* command extrudes a surface from an edge curve, and the *Plane* command can construct a surface from corner points. The traffic signal in Figure 11 was constructed with several surface and solid tools. The pole was constructed of a solid cylinder, and the backplate was trimmed from a plane surface. Edge curves were extruded to form the signal head, and the visors were trimmed from cylinders.

Text on objects such as signs must be constructed from surfaces, else it will not be visible in HVE. The *TextObject* command provides the user with the option of creating text objects constructed from curves, solids or surfaces. Figure 12 depicts text surfaces on a speed limit sign. Text objects are based on True Type fonts.

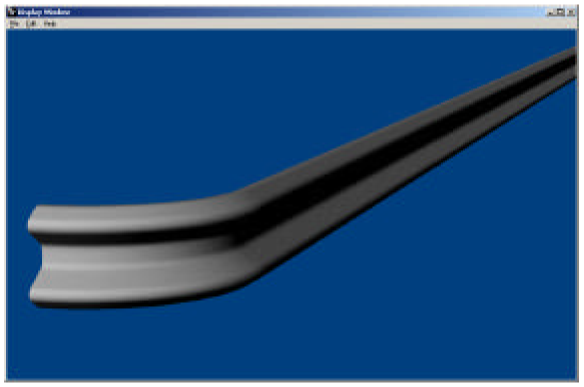


Figure 10. *The guardrail was constructed with the Sweep1 command. A rail curve establishes the curved path of the guardrail.*

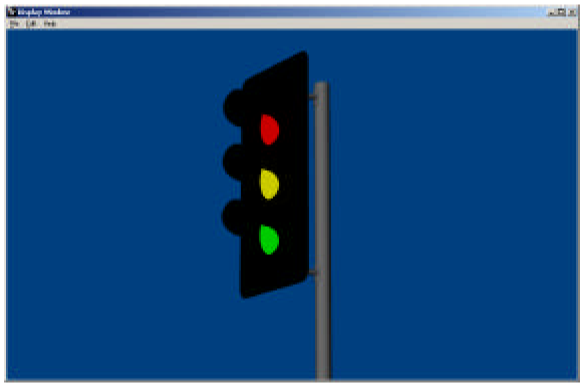


Figure 11. *This traffic signal was constructed using multiple surface, solid and editing commands.*

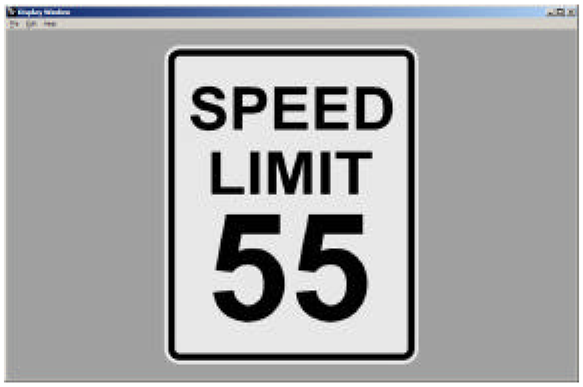


Figure 12. *The text on this speed limit sign are surfaces created with the TextObject command. They are based on the Arial Bold font.*

Seams Between Surfaces

The seam between adjacent surfaces must be "watertight." Gaps between surfaces will be visible and may cause an HVE event to terminate with excessive tire deflection. Eliminating gaps requires that mesh vertices between adjacent surfaces be coincident. Using the same curves to construct adjacent surfaces will not guarantee a tight seam. Figure 13 illustrates a gap between adjacent surfaces, despite using the same curve to construct each surface. A watertight seam can be created by joining the surfaces, a feature of Rhino's NURBS geometry.

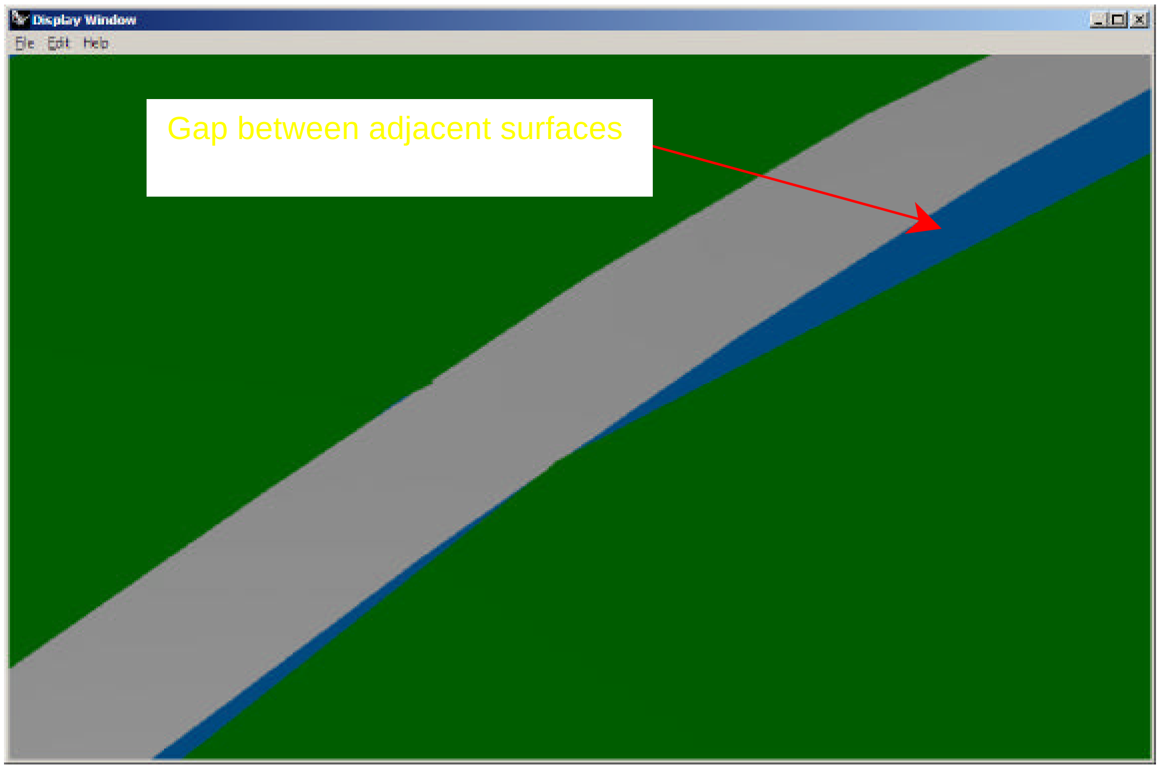


Figure 13. *The blue background shows through the gap between adjacent surfaces. The gap appears despite using the same curve to construct both surfaces.*

The *Join* command can be used to join two or more surfaces. Joining "glues" adjacent surfaces into one polysurface so that meshing crosses the seams with no gaps. The surfaces must touch by naked edges to be joined. A naked edge is the edge of a surface or polysurface that is not connected to another edge. If the naked edges do not match within tolerance, the *Join* command will fail. The tolerance is set in *Document Properties*, and defined by three parameters, absolute, relative and angular. The default tolerances are 0.01 units, 1%, and 3° respectively.

In these instances, the *JoinEdge* command can be used to join two naked edges that do not match exactly. This command overrides the tolerance that prevents the *Join* command from successfully joining the surfaces. A dialog informs the user of the tolerance required to join the edges.

The *Join* command may successfully join several surfaces into one polysurface, yet still permit a gap within the polysurface. This gap occurs when a naked edge of a surface does not match the adjacent surface within tolerance. If the other naked edges match, the *Join* command will be successful. The *ShowNakedEdges* command can be used to check for gaps. Naked edges appearing other than on the perimeter of the polysurface will indicate gaps. The *JoinEdge* command can be used to close these gaps. Additional editing tools are available in Rhino that *Match*, *Blend* and *Merge* surfaces. However, *Join* and *JoinEdge* have shown to be the most useful in creating environments.

HVE environments do not use NURBS surfaces. Instead, polygon meshes must be created from NURBS surfaces prior to exporting the file to HVE. To eliminate gaps between adjacent surfaces, the mesh vertices must be coincident. This is accomplished by meshing polysurfaces.

After all surfaces have been constructed, join them into one polysurface. Objects such as building, signs, utility poles, etc. need not be included with this polysurface. Instead, consider burying building walls, sign posts and poles below ground. Construct a polygon mesh from the polysurface using the *Mesh* command. A polygon mesh is a faceted approximation of the NURBS surface. The accuracy of the mesh is proportional to the mesh density, or polygon count. The higher the polygon count, the more closely the mesh approximates the surface. The mesh density can be set by the user, and can be reduced later.

Like the polysurface, the polygon mesh is one object. It must be exploded into individual meshes that correspond to the original surfaces. The mesh vertices will be coincident at the seams as depicted in Figure 14.

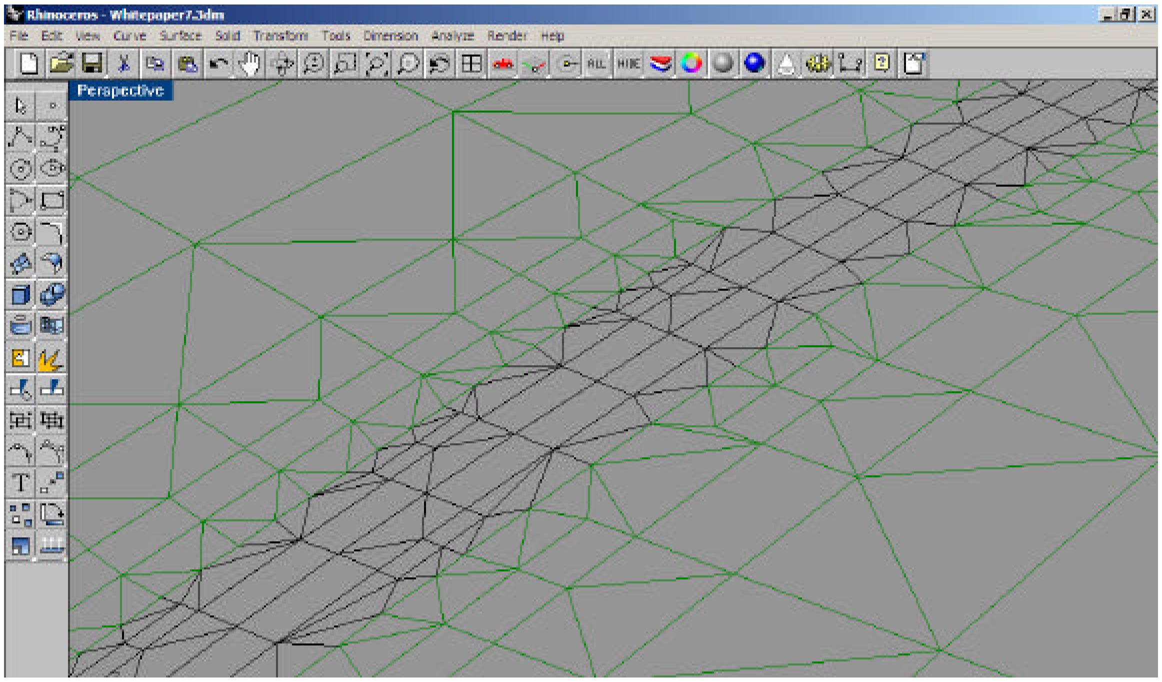

Rhinoceros - Whitepaper7.3dm
File Edit View Curve Surface Solid Transform Tools Dimension Analyze Render Help
Perspective

Figure 14. *The polygon mesh is exploded into individual meshes that correspond to the original surfaces. The mesh vertices are coincident.*

Rhino objects are defined by material properties. The polygon mesh will retain the properties of the polysurface from which it was constructed. These can be changed as desired. The material color property sets the diffuse color in HVE, and the reflective finish color sets the ambient color. To avoid the "washed out" look to surfaces, set the reflective finish to black. The material properties of the meshes in Figure 15 have been changed to the desired colors.

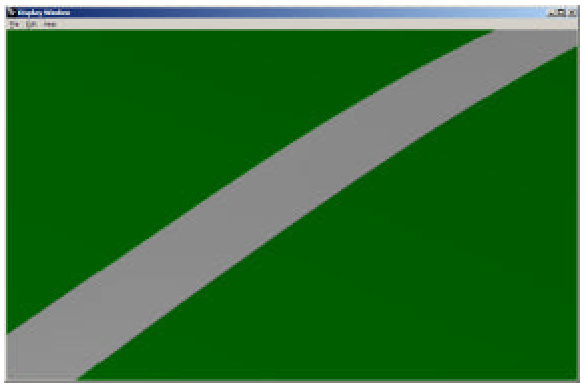


Figure 15. *The material properties of the mesh determine the rendered colors Rhino and HVE.*

Surface Normals

The surface normal defines which side of the surface is up. Since HVE calculation methods interact with only the positive side of the surface, it is crucial that the surface normals for all surfaces over which any vehicle travels are oriented properly. Additionally, only the positive side is illuminated, and skidmarks are drawn on this side.

Problems with surface normal orientation vary among CAD programs and file formats. The most notorious file format is the drawing exchange format (.dxf), in which the normals of polygons within the same mesh may be reversed. This problem does not occur with the VRML file format. However, care must be taken to ensure that the normals are in the proper direction before the file is exported.

The *Dir* command makes it easy to check the direction of the normals of a NURBS surface, and flip them if necessary. Figure 16 shows the normals displayed for the selected surface. If the surface normals are pointed down, simply invoke the *FlipNormal* option of the *Dir* command to reverse their direction.

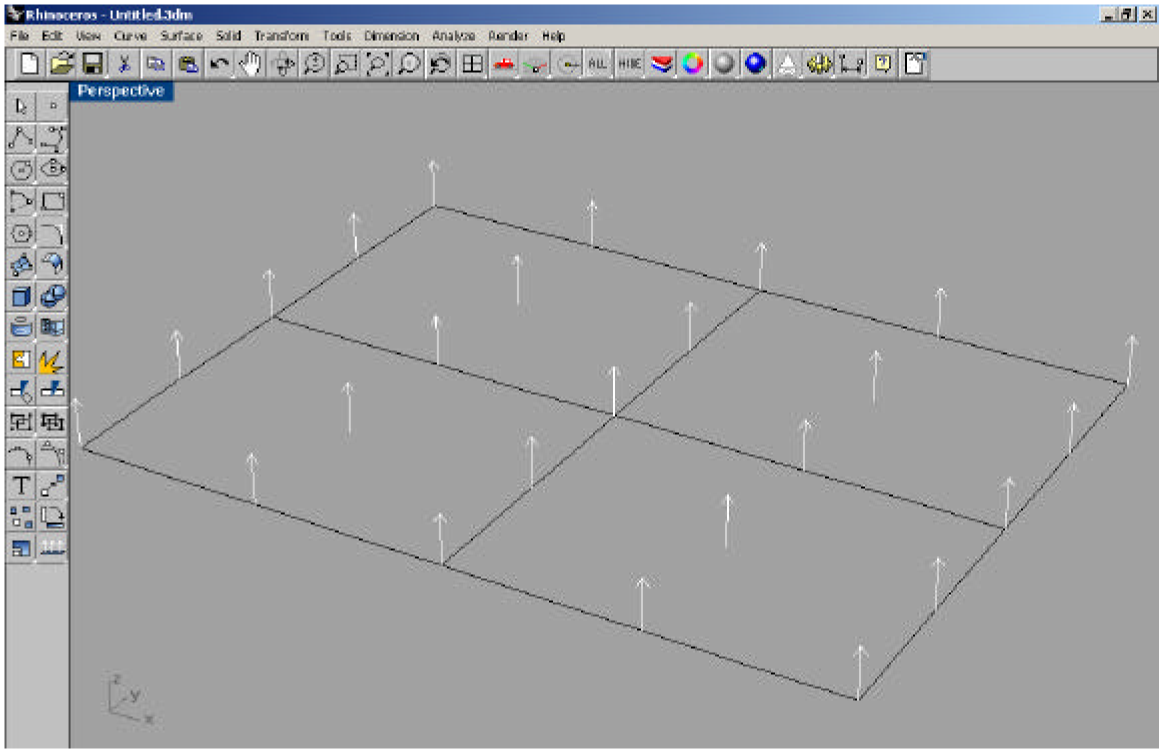
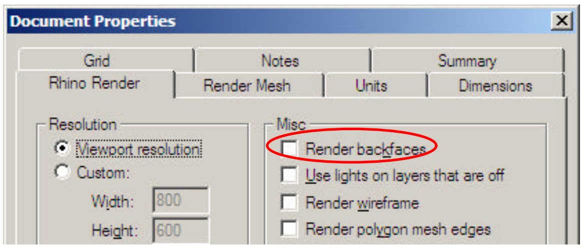

Rhinoceros - Untitled.3dm
File Edit View Curve Surface Solid Transform Tools Dimension Analyze Render Help
Perspective

Figure 16. *The direction of normals of the selected surface is indicated by the DIR command.*

Rhino provides a quick means of checking the normals of all surfaces or meshes visible in the current viewport. In *Document Properties*, uncheck the *Render Backfaces* box on the Rhino Render tab as shown in Figure 17. After the dialog is closed, only the positive side of a surface or mesh will appear in the render or shade views.

With render backfaces disabled, reversed normals are readily identifiable. Figure 18 shows the display window of a rendered viewport. The driveway surface does not appear since the normals are reversed. The viewport background color shows through.


Document Properties

Grid | Notes | Summary
Rhino Render | Render Mesh | Units | Dimensions

Resolution
[✓ Viewport resolution]
Custom:
Width: 800
Height: 600

Misc
☐ Render backfaces
☐ Use lights on layers that are off
☐ Render wireframe
☐ Render polygon mesh edges

Figure 17. *The render backfaces box is located on the Rhino Render tab of the Document Properties dialog.*

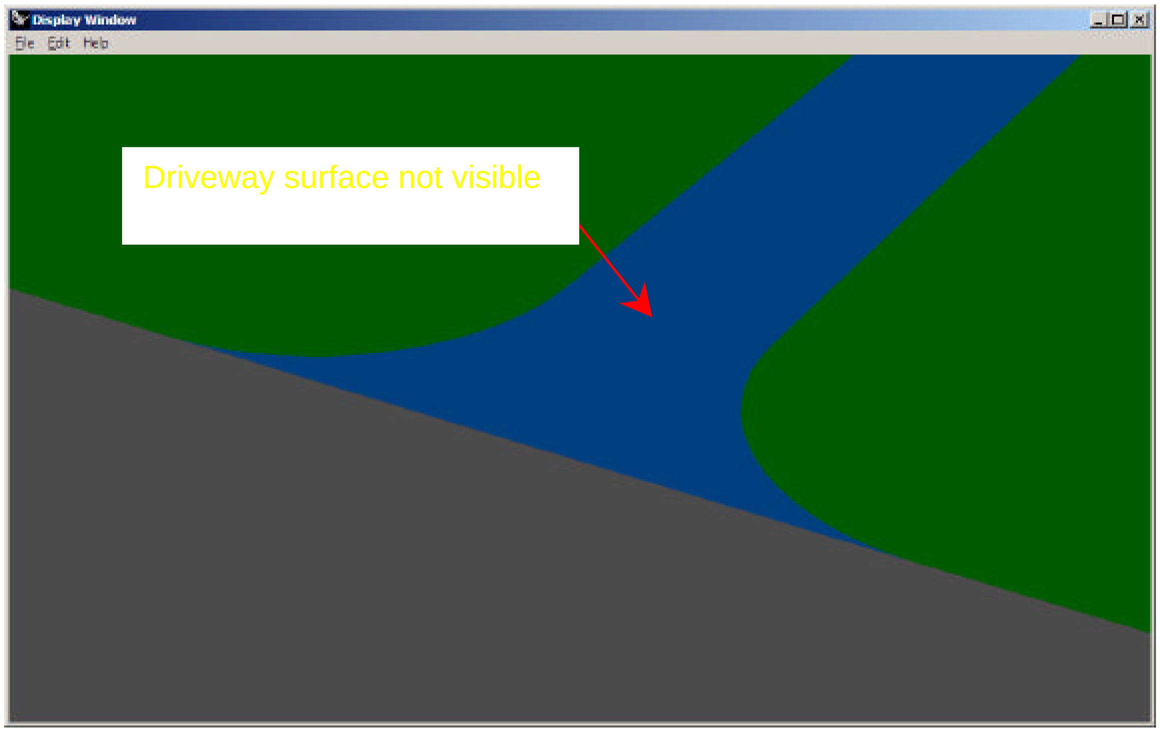


Figure 18. *The surface normals of the driveway are reversed. With render backfaces disabled, the drive is not visible when rendered. The render background color shows through the surface.*

The normals of polygon meshes created from NURBS surfaces retain the direction of the original object. Checking the normal direction before meshing a NURBS surface will ensure that the mesh normals are in the proper direction. However, the normals of meshes can also be flipped using the *Flip* command.

Using the *JoinEdge* command to join surfaces can result in reversed normals as depicted in Figure 19. This can be corrected after the surface is meshed. Once the mesh is exploded, the user can reverse the normals as needed with the *Flip* command. This problem can be avoided by using the *Join* command.

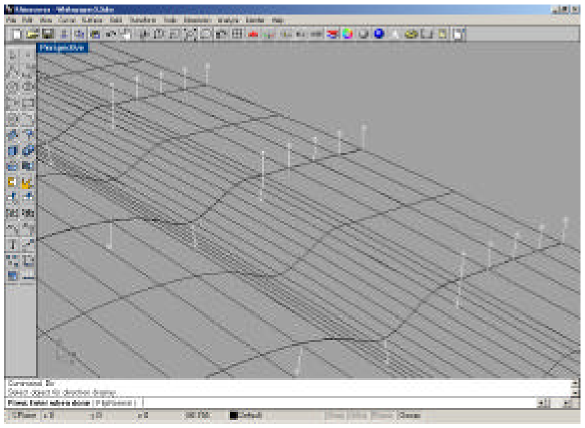


Figure 19. *The above polysurface was joined from two NURBS surfaces using the JoinEdge command. The normals of the nearer surface are reversed.*

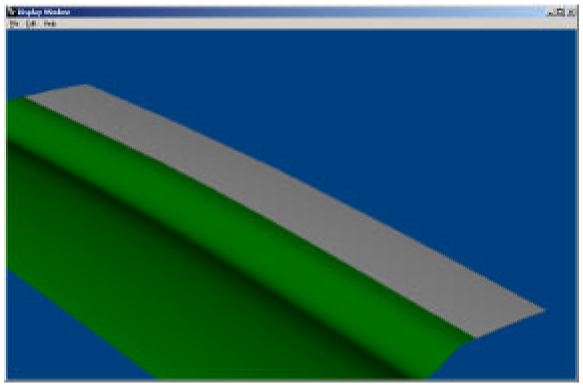


Figure 20. *A polygon mesh was created from the polysurface in Figure 21. After exploding the mesh, the normals were reversed using the Flip command. The corrected normals are now in the proper orientation to import into HVE.*

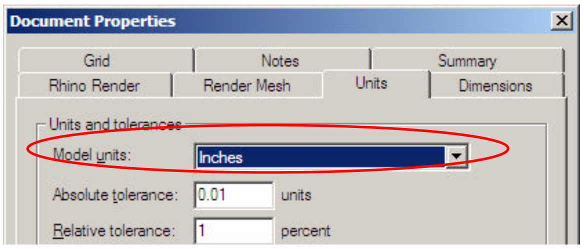
File Compatibility

The preferred file format for HVE environment files is VRML (.wrl) Version 1.0. Rhino supports this format for export. NURBS surfaces are not exported in the VRML format. Instead, polygon meshes are created from the surfaces for export. Only the polygon meshes should be exported in the VRML file.

The individual properties of polygon meshes are retained when exported as VRML files. This is irrespective of the layer on which the mesh resided. This increases the ease with which the environment file can be edited in HVE's 3-D Editor.

Rhino layers do not import into HVE. All objects appear on the default, *Untitled*, layer. However, objects can be moved to new layers created in HVE's 3-D Editor. The object attributes can be changed with the 3-D Editor in the same manner as with any imported file.

Preparing the file for export requires three steps. In *Document Properties* change the model units to inches as shown in Figure 21. Rhino prompts the user whether to scale the model by the appropriate factor. The environment is now scaled with the proper units.



Model units: Inches
Absolute tolerance: 0.01 units
Relative tolerance: 1 percent

Figure 21. *The entire environment can be scaled to the proper units by changing model units to inches on the Units tab of Document Properties.*

The environment must be rotated about the *x* axis to match the earth fixed coordinate system in HVE. Use the *Rotate3D* command, and rotate the entire environment 180 degrees about the *x* axis. The file is now ready for export.

The exported file should include only polygon meshes. The *Export* command creates a new file with only selected objects. Select only the polygon meshes, and change the *Save as type:* to VRML (.wrl). Select Version 1.0 in the VRML Export Options dialog. The VRML file can be opened in HVE, and modified in the 3-D Editor. Additionally, Rhino imports and exports common AutoDesk® formats such as .dwg, .dxf. and 3ds, among others.

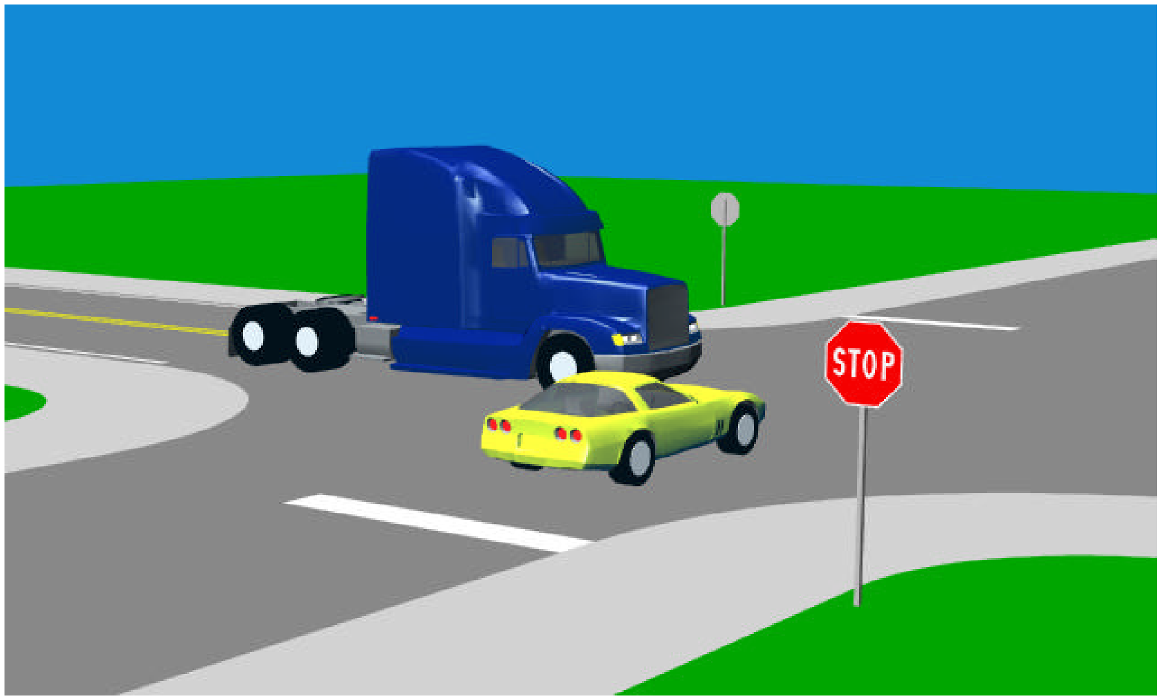


Figure 22. *VRML (.wrl) Version 1.0 files import seamlessly into HVE.*

Conclusion

Rhinoceros offers many advantages to the user when creating HVE environments. Its functionality makes creating environments easier and more accurate, and issues common to other programs are easily handled in Rhino. This whitepaper is intended as an introduction, and is not inclusive of all the advantages of Rhino to HVE users.

References

Rhinoceros Users Guide, Version 2.0, Robert McNeel & Associates, 2001.

Cheng, Ron K.C., *Inside Rhinoceros*, 2002, Onword Press, Albany, NY.

HVE Operations Manual, Engineering Dynamics Corporation, Beaverton, OR, 2000.

Trademarks