Mechanical Simulation
755 Phoenix Drive, Ann Arbor MI, 48108, USA
Phone: 734 668-2930 • Fax: 734 668-2877 • Email: info@carsim.com

Road Surface Visualization

Besides representing roads in the vehicle math model, BikeSim, CarSim, and TruckSim generate 3D shapes to help visualize the road surfaces. This document describes the screens and tools used to define the visual appearance of a road surface as viewed in VS Visualizer.

Note Roads and paths in BikeSim, CarSim, and TruckSim are intended for use by two levels of users. *Model Users* are engineers and designers who will be simulating vehicles in previously defined scenarios and conditions. If the simulation has just one road and mainly involves the evaluation of the vehicle behavior, it is usually not necessary to assemble an environment with multiple road surfaces; you can just use the existing road datasets or make copies and adjust the properties.

The other level of user is the *Model Builder*. A Model Builder is a more advanced user who can assemble paths, roads, and other simulation components to build new scenarios with greater complexity. The road and path concepts in VehicleSim products support the creation and simulation of complicated situations. Further, the road surface information is used to automatically generate 3D shapes used by VS Visualizer. Considering these combinations can also require some thought and planning.

Assembly of a Road Surface

In most simulations, the vehicle is controlled to follow paths of interest, and tires only contact the ground near those paths of interest. In the VehicleSim math model, the concept of a *road surface* is mainly a representation of the ground properties (geometry and friction) in a form that is well-defined where the vehicle tires are likely to travel, and sparse or nonexistent where the tires are unlikely to be. To do this, VS Roads represent the road surface using a coordinate system based on a 2D VS Reference Path — a continuous line that exists in a horizontal plane with continuity in position and gradient.

The path coordinates are station S (distance along the path) and lateral coordinate L (distance a point is from the path, measured on a line that intersects the point and the path, and is perpendicular to the path at the point of intersection (Figure 1).

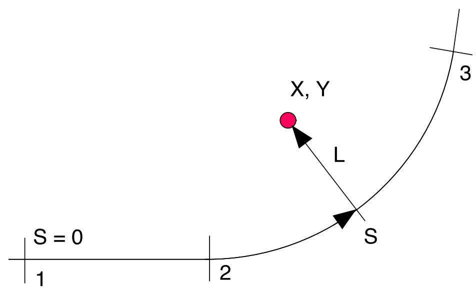


Figure 1. S and L are the coordinates of a reference path.

A VS Road Surface is an assembly of datasets that use the same S-L coordinate system of a VS Reference Path. The datasets that make up a VS Road Surface are linked together using a dataset in the **Road: 3D Surface (All Properties)** library (Figure 2), described in the **Help** menu document *Paths and Roads*. The controls that are numbered in the figure involve the visualization of the road surface.

1 Link to VS Reference Path that defines the S-L coordinate system for the road. Depending on the library where the dataset is located, it might also include elevation information for the reference line, and even animator assets. This is the only link on the screen that is mandatory.

2 Link(s) to dZ datasets that define a component of elevation as a 2D function of S and L. One of the libraries that may be used here is **Road: Off-Center Elevation Map, Variable Width**, which defines edges of lanes that may vary with S. Datasets from this library support elevation details for variable-width road surfaces, and support visualization of variable-width lanes. In Figure 2, for example, the concrete lane on the right side of the road narrows, as does the asphalt and grass lanes on the left side to the road. In addition, the travelled part of the road widens with station (these properties are seen more clearly in Figure 9, page 8).

3 Button to have the Browser generate polygon shapes. This uses the 3D road data from the links on the left side of the screen (1 and 2), in combination with a linked dataset containing animator shape definitions 4.

4 Link to a dataset from the **Road: Animator Surface Shapes** library, described in the following section.

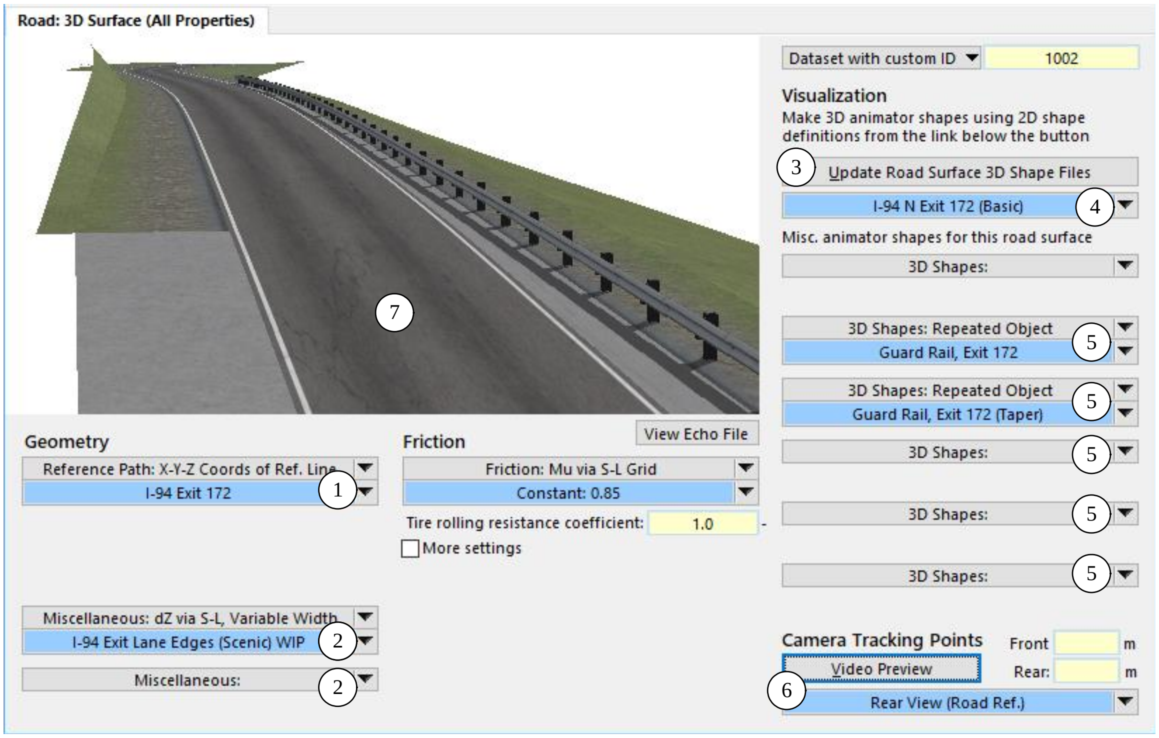


Figure 2. Road: 3D Surfaces (All Properties) dataset assembles datasets for a road surface.

5 Link(s) to animator datasets, such as the sky and background, as well as objects whose locations are set using S-L coordinates using the **Road: Animator Repeated Object** library, which is described in a later section (page 31).

6 Button to launch VS Visualizer and view the road. The view may be copied and pasted into the graphic area 7 to provide a convenient visual reference for the dataset.

7 Visual reference image. This is typically obtained by clicking the Video button 6 to launch VS Visualizer to show nothing but the road surface and related animator shapes.

How Road Visualization Works

BikeSim, CarSim, and TruckSim use a VS Solver called VS Road Calculator to calculate X-Y-Z coordinates of triangles assembled into shapes for VS Visualizer. The software includes many 2D material types (textures defined with images and support files for lighting, reflectiveness, blending, etc.) to render realistic looking surfaces for the generated shapes (Figure 2 and Figure 3).

Each road surface for the math model is shown during visualization using sets of computer-generated shapes based on a range of S and L that each define a “visual strip” with distinct graphical properties. For example, Figure 4 shows the triangle shapes underlying the rendering seen in Figure 3.

Figure 3. Appearance of a road surface in VS Visualizer.

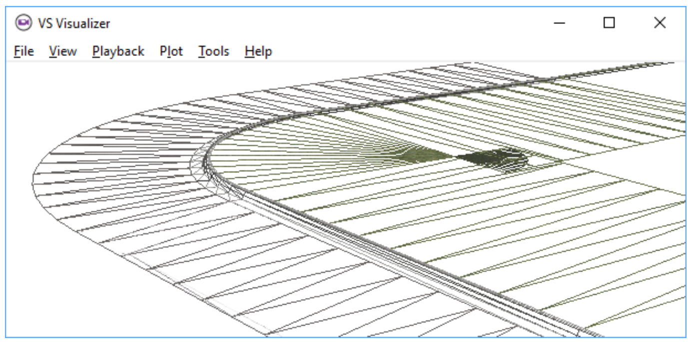


Figure 4. Computer generated shapes for an example road surface.

The 3D shapes that make up a strip use the OBJ file format. If material textures are used, then a corresponding material MTL file and one or more 2D image files (photographs, textures) can be mapped over the 3D road surface. If the image is smaller than the area of the strip, it is repeated as needed to cover the strip entirely. The repeated image copies are mapped over the rectangles that define the strip, to give the effect of a "skin" covering the 3D road shape.

The OBJ and MTL file formats, originally developed by Wavefront, are described and discussed in the **VS Visualizer Reference Manual.** If you are interested in adding new custom textures to the road shapes, then please refer to the sections in that manual that describe the MTL file format.

The Road: Animator Surface Shapes Screen

You will use the **Road: Animator Surface Shapes** screen (Figure 5) to construct and arrange up to 50 visual strips, which will define a 3D road surface for viewing. The strips can be connected or alternated using this screen to create stripes and grids.

The datasets associated with this screen are used to update the visualization whenever you make a change to a dataset in the **Road: 3D Surface** library or click the **Update Road Surface 3D Shapes** button on the **Road: 3D Surface** screen.

The columns in the table of strips appear in several groups. The first (1 - 6) specify how an image is repeated to show the strip. The next columns (7 - 12) specify the range of L coordinates and the interval used to generate shapes that follow the road surface, and columns (13 - 16) specify the range of S coordinates and interval. The last columns (17 and 18) specify an incremental elevation and a method for adding detail to the textures of image copies.

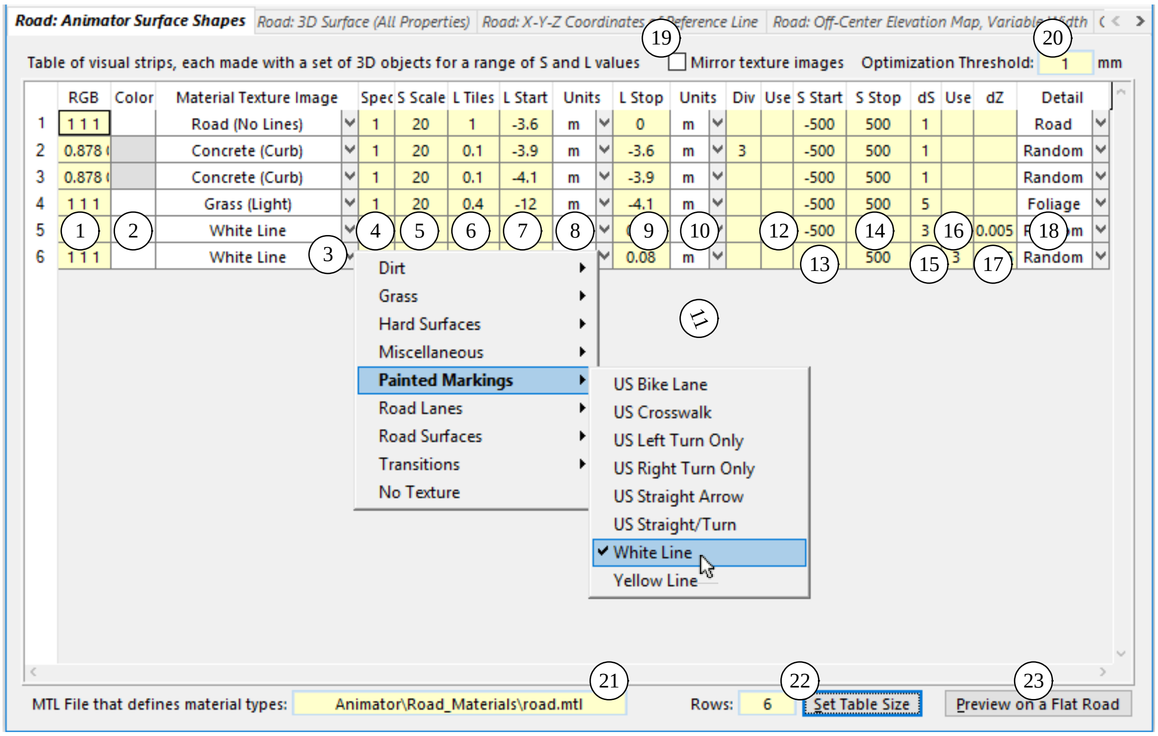


Figure 5. The Road: Animator Surface Shapes screen.

Properties of the Texture Image

(1) **RGB** color component (red, green, blue). Color is saved in the dataset with RGB numbers separated by a space. The color defined by the RGB values is shown in the adjacent **Color** column (2).

(2) **Color**. Each cell shows the color defined by the adjacent RGB values (1). Each cell is also a button; click a cell to view a window with a color selector (Figure 6).

In this pop-up, you can view and edit the color for each strip in the table. Select a new color with the control in the lower-left corner of the pop-up window. When you are done, click **OK**. The RGB numbers for the color are immediately shown in the adjacent field (1).

Be aware that the color selected here interacts with colors that might be in the image used for texture associated with the material type. Any color other than white (RGB = 1 1 1) will darken the texture.

(3) **Material Texture Image**. Each drop-down control in this column provides a menu to select a texture image (Figure 5). Each selection includes a baseline texture image, plus other files that enhance lighting, reflecting, blending, etc. There is also an option **No Texture** that will simply use a solid color that will be applied directly to the machine-generated objects.

Guidelines for selecting a material type and setting other properties are provided in the next section (page 12).

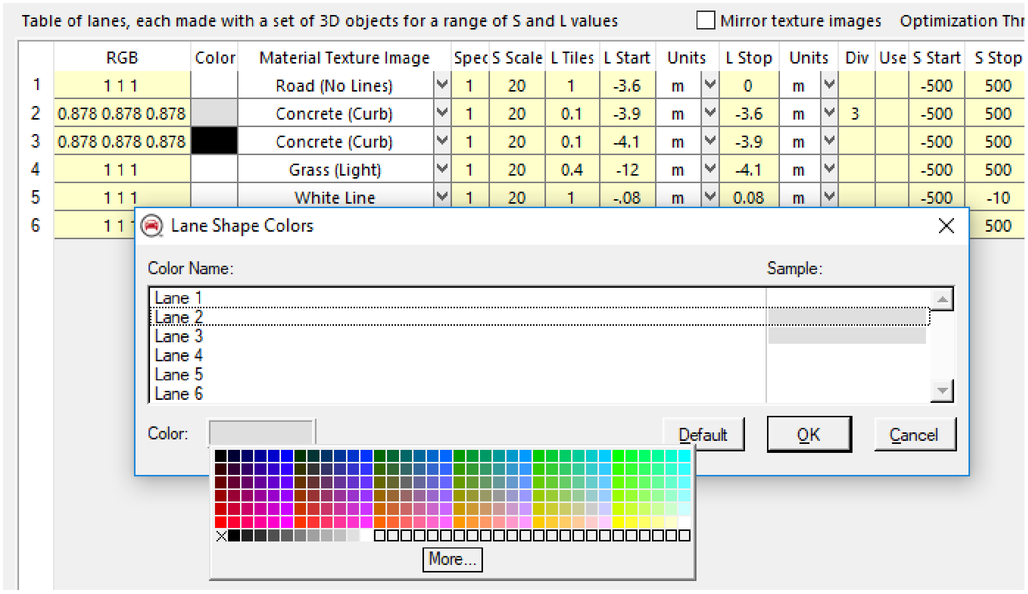


Figure 6. Pop-up color selector.

4 Specular value **Spec** (shininess). Shiny surfaces reflect light in a specific direction, while dull surfaces reflect the light equally in all directions. The specular value for a surface determines how directionally light is reflected, relative to a viewing angle. A high value means the surface is very shiny, like a mirror. A value of 0 means the surface is dull.

All of the material texture images have built-in design reflective properties, and are intended to be used with a specular value of 1.

5 **S Scale** of texture graphic. When a material image is specified for a strip, then the image will be rendered using this value as the dimension in S. Typically, a road is longer than the image, so the image will be repeated as needed to fill the shape of the strip, treating the image as a "tile" with this length. For example, a value of 20 would cause the texture image to be repeated every 20 m. Figure 7 shows the effects of two different values for the longitudinal scale.

S Scale values are recommended for each of the installed material types in the next subsection (page 12).

Figure 7. Longitudinal (S) scale.

6 **L Tiles** is the number of times the texture image will repeat in the lateral direction. When a material image is specified for a strip, then this value defines how many times the image will be repeated to fill the strip in the lateral direction. For example, if the strip describes the road surface, a value of 1 causes the texture of the road to be stretched to fill the strip (Figure 8).

Figure 8. Lateral repeats of texture.

If the material type is represented with a square 2D image file, then the L Tiles value is often set by dividing the width of a strip (the L Stop value 9 minus the L Start value 7) by the S Scale length 5. If the material type is not intended to repeat (e.g., a transition from pavement to grass), the L Tiles field is typically set to 1.

If the image is asymmetric, you can set flip the image with respect to L by setting a negative value of L Tiles. For example, some of the material types show a transition from pavement to grass. For the left side of the road (increasing L) the L Tiles value should be 1; for the right side (decreasing L) the same material type is used, but the L Tiles value should be –1.

Lateral Dimensions of Machine-Generated Polygons

Strips widths are defined by lower and upper limits in the lateral coordinate L. Two options are available for specifying lateral positions:

1. L can be specified directly with units of meters. This mode is the simplest choice if your strip has a constant width. To use this mode, set the **Units** (8 and 10) to **m** as shown for all strips in Figure 5.
2. L can be specified by identifying edges. This mode allows for a strip to vary in width. To use this mode, set the **Units** (8 and 10) to **edge** as shown for all strips in Figure 10 and refer to the subsection below.

Specifying L directly

When defining a strip which has a constant width, the width can be defined simply with starting and stopping L values.

7 **L Start**. If the Units 8 are for **L Start** are set to **m**, then this value defines one of the bounding values of L.

9 **L Stop**. If the Units 10 are for **L Stop** are set to **m**, then this value defines the bounding value of L. The strip will cover the range of **L Start** to **L Stop**. It does not matter whether **L Stop** is greater than **L Start**, as the software will reverse the values if necessary. However, **L Stop** must not be equal to **L Start**.

Specifying L Using Edges

When defining a strip which has variable width, lateral boundaries can be specified using edges. For example, Figure 9 shows a highway exit road that starts with one traffic lane and then widens to two lanes. A concrete lane on the right side starts with a full width at the start (not visible in the figure) and eventually narrows to almost nothing. This is the same road surface that was shown earlier (Figure 2, page 3), but stripped down - in this case only the basic road geometry is described.

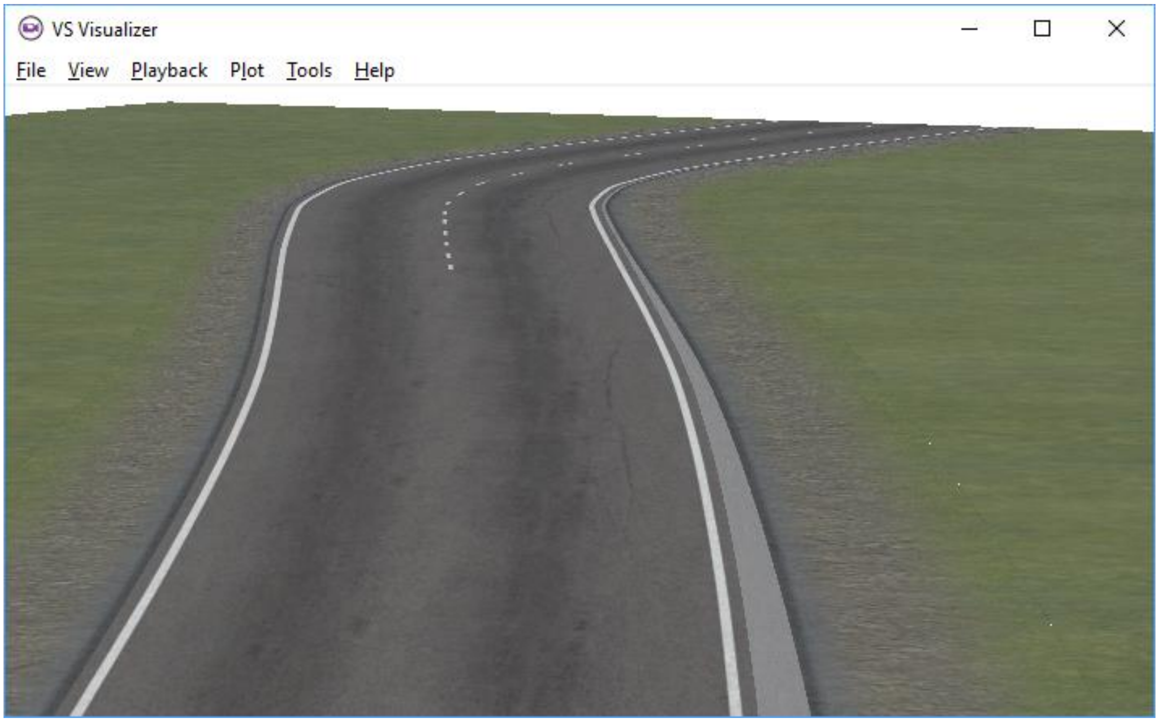
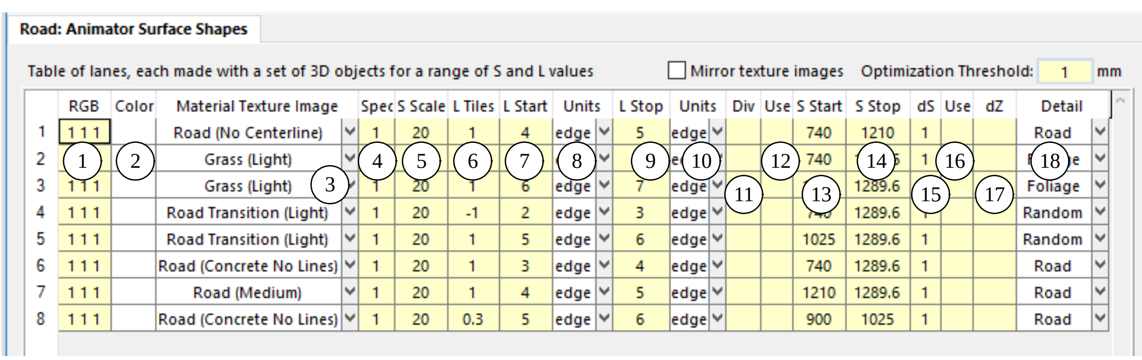


Figure 9. Highway exit road with variable-width lanes.

Figure 10 shows the **Road: Animator Surface Shapes** dataset for this road surface. Notice that all Units for **L Start** and **L Stop** are set to **edge** (8 and 10). The edges themselves are defined in another dataset using the **Road: Off-Center Elevation Map, Variable Width** screen (Figure 11), which can provide incremental elevation using edges whose L coordinate may vary with station.



Road: Animator Surface Shapes

Table of lanes, each made with a set of 3D objects for a range of S and L values   Mirror texture images   Optimization Threshold: 1 mm

	RGB	Color	Material Texture Image	Spec	S Scale	L Tiles	L Start	Units	L Stop	Units	Div	Use	S Start	S Stop	dS	Use	dZ	Detail
1	1 1 1		Road (No Centerline)	1	20	1	4	edge	5	edge			740	1210	1			Road
2	[illegible]		Grass (Light)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]			740	[illegible]	1			[illegible]
3	1 1 1		Grass (Light)	1	20	1	6	edge	7	edge			[illegible]	1289.6	[illegible]			Foliage
4	1 1 1		Road Transition (Light)	1	20	-1	2	edge	3	edge			[illegible]	1289.6	[illegible]			Random
5	1 1 1		Road Transition (Light)	1	20	1	5	edge	6	edge			1025	1289.6	1			Random
6	1 1 1		Road (Concrete No Lines)	1	20	1	3	edge	4	edge			740	1289.6	1			Road
7	1 1 1		Road (Medium)	1	20	1	4	edge	5	edge			1210	1289.6	1			Road
8	1 1 1		Road (Concrete No Lines)	1	20	0.3	5	edge	6	edge			900	1025	1			Road

Figure 10. Road Animator Surface Shapes based on edges for L.

A **Road: Off-Center Elevation Map, Variable Width** dataset serves two purposes:

3. It defines edges to make animator shapes for strips that are not necessarily parallel to the road reference path.

4. It defines incremental elevation dZ using a variable-width grid. As described in the **Help** menu document *Paths and Roads*, it can greatly reduce the amount of data needed to define simple geometric features that are not parallel or perpendicular to the road reference path.

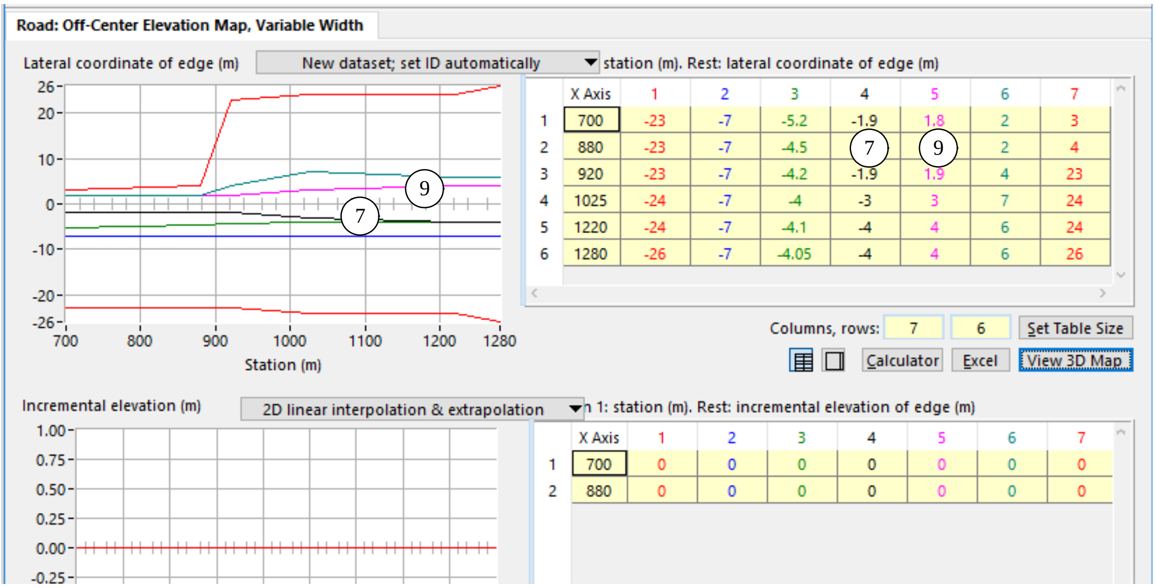


Figure 11. Variable Width Elevation Map used to define edges.

In the example shown in Figure 9 and Figure 11, the variable-width dataset has no incremental elevation data; it exists solely to define edges that can be used to make animator shapes.

When a **Units** drop-down control (8 and 10) is set to **edge**, the preceding value is an integer that identifies a column in the 2D table from a dataset from the linked **Road: Off-Center Elevation Map, Variable Width** library. For example, the first row of the table in Figure 10 sets **L Start** to edge 4 7 and **L Stop** to edge 5 9. The plots of Lateral coordinate for the edges in Figure 11 show how the strip between these two edges (4 and 5) widens as Station increases.

Keep in mind, to use edges for defining strips a **Road: 3D Surface (All Properties)** dataset must link to both a **Road: Off-Center Elevation Map, Variable Width** dataset, (Figure 2, page 3) that defines the edges 1 and a **Road: Animator Surface Shapes** dataset that uses them 4.

Number of Divisions and Usage

In addition to defining the boundaries of the strip, it is possible to customize the level of detail in the strip geometry, and create alternating patterns using the **Div** and **Use** parameters.

11 Number of divisions laterally (**Div**). Each strip shape is made up of a series of triangles paired to create rectangles (Figure 4, page 4). By default, a single rectangle will cover the lateral extent of the strip. If the strip has significant transverse elevation changes, this field allows you to add extra sub-divisions to add more detail in the geometry (Figure 12). However, more

sub-divisions mean the shape files will be larger and will take longer to load. If not specified, a single rectangle will be sized to match the strip width. In the first example (Figure 5, page 5), extra divisions are specified in row 2 of the table to show the curb in detail (Figure 3 and Figure 4).

Div = 1 *Div = 3*

Figure 12. Lateral number of divisions.

(12) Lateral skip counter (**Use**), which allows some divisions to be skipped. If not specified, a count of 1 is applied, indicating that every strip division be shown. A count of 2 means every other division is shown, a count of 3 means every third, and so on. This option is included to make it efficient to generate checkerboard or stripe patterns (Figure 13).

Use = 2 *Use = 4*

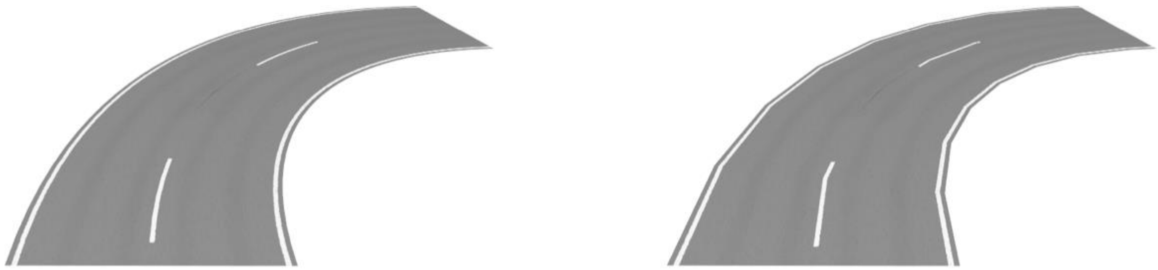
Figure 13. Lateral "Use" setting to skip divisions.

Longitudinal Dimensions of Machine-Generated Polygons

Strips lengths are defined by lower and upper limits in the longitudinal coordinate S.

(13) **S Start** limit and **S Stop** (14) limit. Shapes will be generated for the strip following along the path from the Start limit to the Stop limit.

(15) Longitudinal interval **dS**. This is similar to the **Div** parameter, but for length – it allows you to define the interval at which new rectangles are generated going over the longitudinal range of the strip. If the road is straight and flat, the interval can be as large as the range covered (**S Stop – S Start**). For roads with curves or vertical changes, the interval should be short enough to show the surface shape changing (Figure 14). Be aware that a smaller interval means the shape files are larger and take longer to load. If not specified, a default value of 2m is used.



dS = 1 *dS = 4*

Figure 14. Longitudinal interval dS.

The screen includes an optimization threshold (20). If this is nonzero, the software will try to increase the interval if doing so will not affect the visible detail. Therefore, long straight sections may have just a few shapes, if doing so keeps the corners of the potential polygons within the specified threshold.

(16) Lateral skip counter **Use**, which allows some divisions to be skipped. If not specified, a count of 1 is applied, indicating that every **dS** increment be shown. A count of 2 means every second increment is shown, a count of 3 means every third, and so on (Figure 15). This option is included to make it efficient to generate checkerboard or stripe patterns.

Use = 2 *Use = 4*

Figure 15. Effect of longitudinal "Use" setting to skip intervals.

Adjustments in Elevation and Detail

(17) Vertical offset **dZ**. This value is added to the elevation of the road when the road shape generator defines the corners of the shapes made for the strip. These values can be used to make sure layered shapes appear as intended. For example, stripes on the road can be set up with a slight positive offset such as 0.01 m to prevent them from being submerged in road bumps. If not specified, a value of zero is used.

(18) **Detail**. This drop-down control is used to select a type of texture detailing that is applied when the scene is rendered with the **Standard** render mode of the VS Visualizer. Note that VS Visualizer in **Compatibility** render mode is not affected by this setting. This texture detailing is used to add realism by breaking up the tiling patterns associated with these types of repeated textured geometry.

Random is the default value assigned to a new row, and uses a generic noise texture projected top down in large sections onto the generated geometry. **Foliage** looks best with off-road sections. **Road** should be used with long sections of contiguous road with no lateral tiling. **Wrapped** is best selected with smaller sections with tight texture tiling, as it re-uses the geometry's texture wrapping coordinates, loosening them for the detail texture. **Off** will omit the detail texture, which can provide some speed benefit, particularly on older hardware.

Miscellaneous Settings for the Dataset

(19) Checkbox to mirror texture images. The 2D images used to build the texture of the road surface can be repeated exactly as they are originally shown, or mirrored to maintain continuity. Images that are designed to be used as textures usually appear best with the mirror option not checked; images that are not carefully designed will often look better with the mirror option checked.

If using only the materials provided with the software, this should be unchecked.

(20) **Optimization Threshold**. When displaying very detailed shapes, the animation may run slowly on some computers if the longitudinal interval is small or the lateral number of divisions is large. In order to reduce the size of the files, the shape generator can look for flat areas to simplify if there is insignificant loss of detail. If the vertices between two adjacent rectangles can be removed and cause less error than this optimization threshold, the rectangles will be combined to improve performance.

If you specify zero or a negative value, then no optimization is done, and all rectangles are generated.

(21) Pathname for MTL material file that defines materials that can be used to define road surfaces for the animator. As described in the **VS Visualizer Reference Manual**, the animator supports Wavefront OBJ files that in turn reference MTL material files that define one or more materials. The materials are defined in terms of lighting properties and optionally textures defined in 2D image files stored in familiar formats such as BMP, JPG, etc.

According to the OBJ specification, the pathname for the MTL file should be defined relative to the folder containing the OBJ files. The OBJ files are all created in the folder for the 3D road datasets (`Roads\3D_Road`). The material file installed in BikeSim, CarSim, and TruckSim is located in the folder `Roads\Material`.

When given a relative pathname, the VS Browser looks in two locations for the parent folder:

5. The current database folder (e.g., `CarSim_Data`)
6. A folder in the `Resources` folder for the product (e.g., `CarSim_Prog\Resources`). The `Resources` folder contains image and 3D shape files that are available for all databases used with a product.

In most cases, the MTL file from `Resources` is used. (The folder `Resources\Animator\Road_Materials` is the default location for the MTL files used with all of the examples installed with BikeSim, CarSim, and TruckSim.)

(22) Controls to set the table size and number of strips. To change the number of rows in the table, edit the number in the yellow field and click the adjacent **Set Table Size** button. A maximum of 50 rows can be set for the table.

(23) Click the **Preview on a Flat Road** button to generate a straight flat road using the strip definitions current in display and show the surface in VS Visualizer (Figure 16).

Guidelines for Using Materials

BikeSim, CarSim, and TruckSim include over 50 material types. This section shows all of them and provides guidelines for setting **S Scale** (5), **L Tiles** (6), and **Detail** (18).

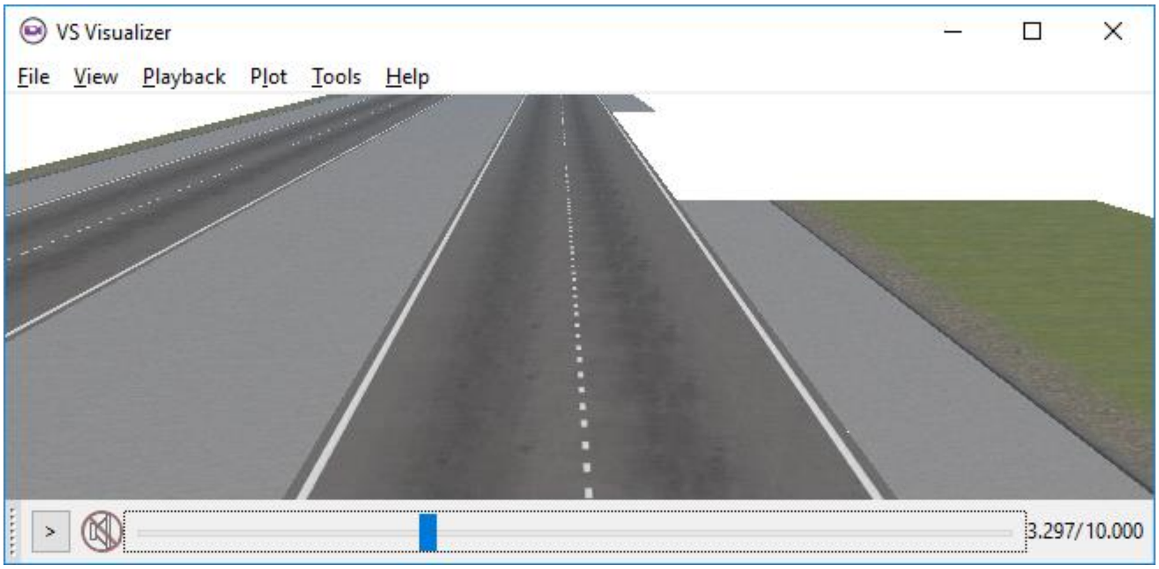


Figure 16. Flat straight road generated and viewed using the Preview on a Flat Road button.

Dirt

The dirt material types use square images. For these surfaces, the L Tiles value should be approximately the width of the strip divided by the value in the S Scale column. This will keep the material from distorting.

The examples below are using S Scale = 20, L Tiles = 1 and Detail= “Foliage” on a strip with width = 20m (Figure 17 and Figure 18).

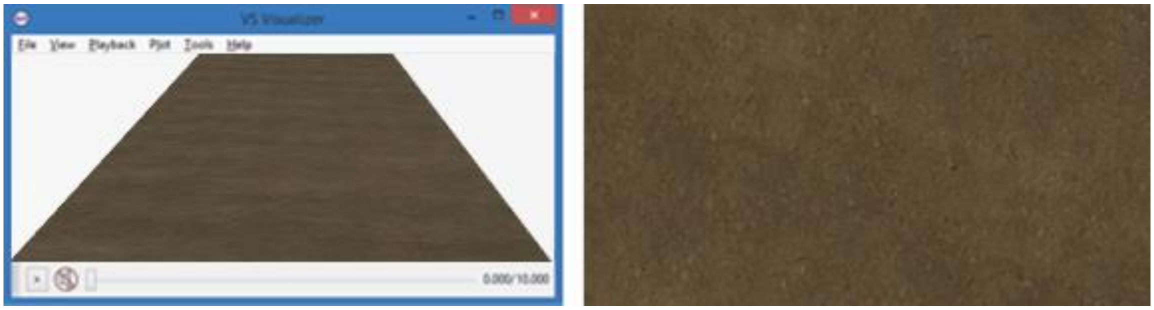


Figure 17. Dirt (Dark).

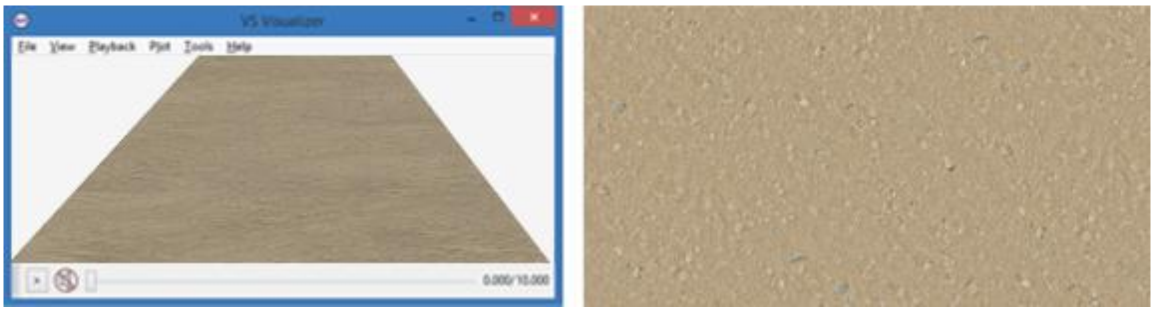


Figure 18. Dirt (Light).

Grass

The grass material types use square images. For these surfaces, the L Tiles value should be approximately the width of the strip divided by the value in the S Scale column. This will keep the material from distorting.

The examples below are using S Scale = 20, L Tiles = 1 and Detail= “Foliage” on a strip with width = 20m (Figure 19 and Figure 20).

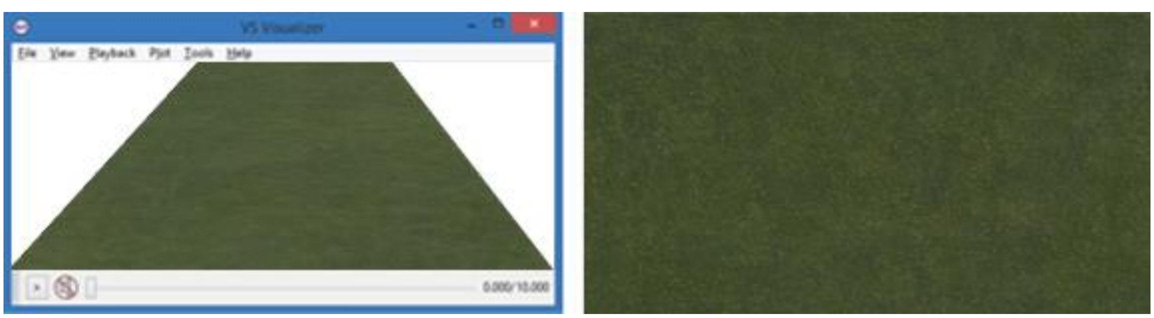


Figure 19. Grass (Dark).

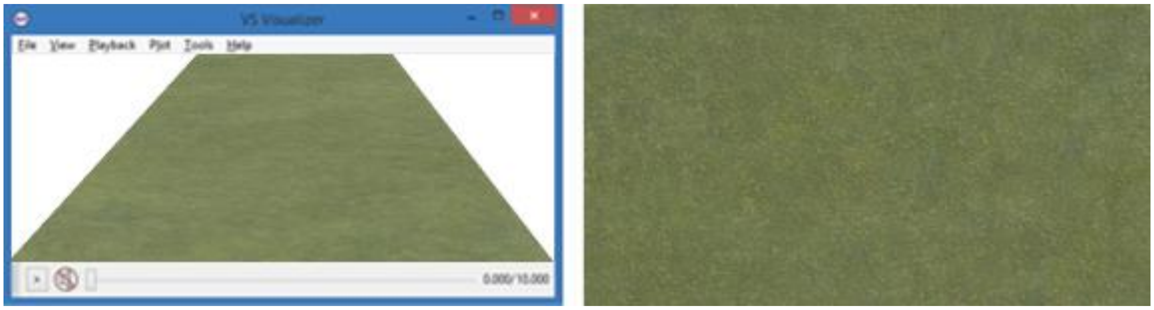


Figure 20. Grass (Light).

Hard Surfaces

The following material types use square images and have no lines. For these surfaces, the L Tiles value should be approximately the width of the strip divided by the value in the S Scale column. This will keep the material from distorting.

The examples below are using = S Scale of 20, L Tiles = 1 and Detail= "Off" on a strip with width = 20m surface (Figure 21 to Figure 38).

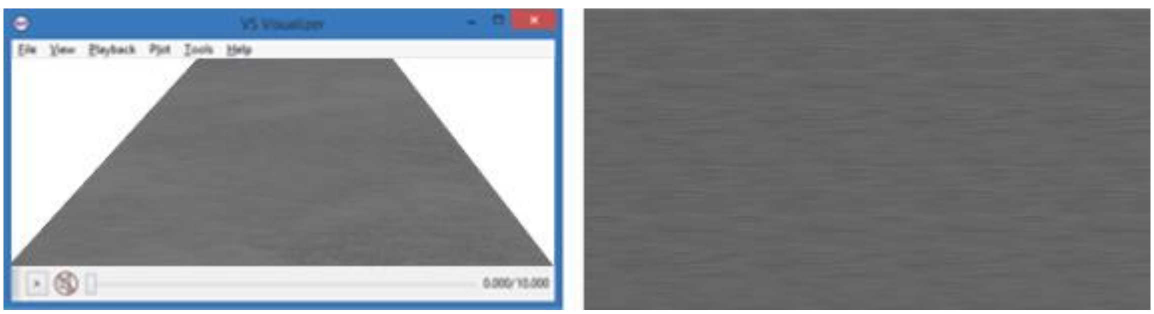


Figure 21. Brushed Metal.

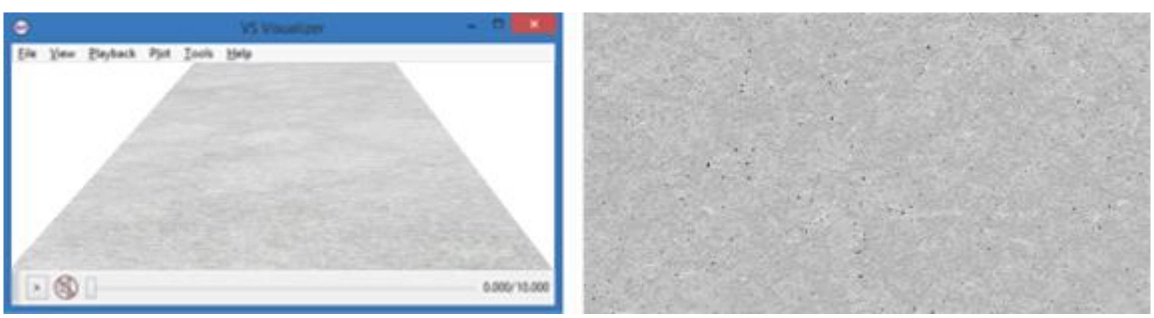


Figure 22. Cement (Pitted).

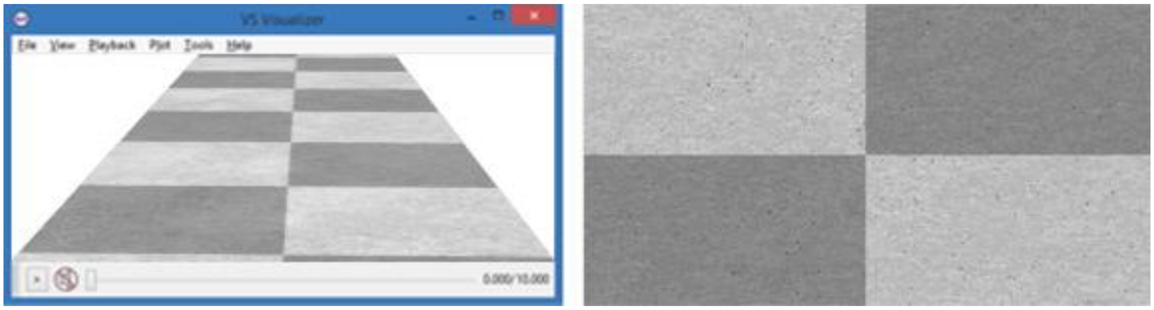


Figure 23. Checkerboard.

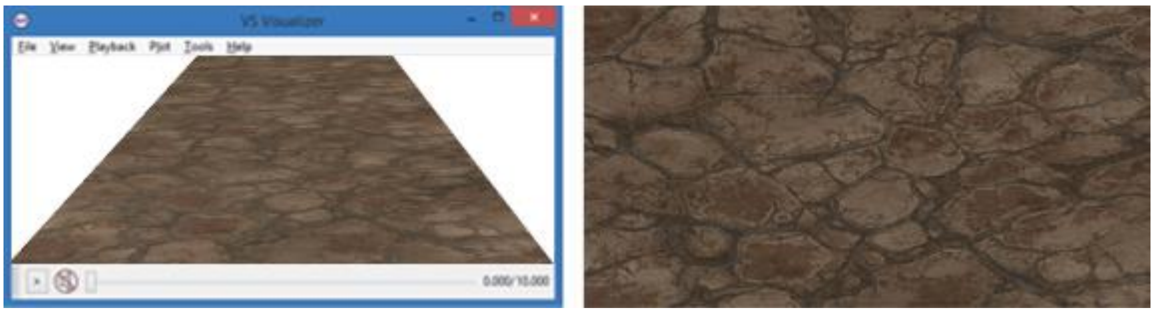


Figure 24. Clay (Cracked).

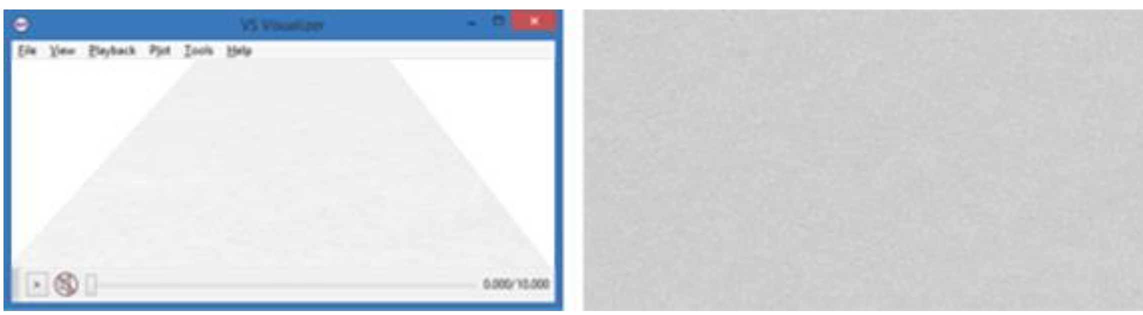


Figure 25. Concrete.

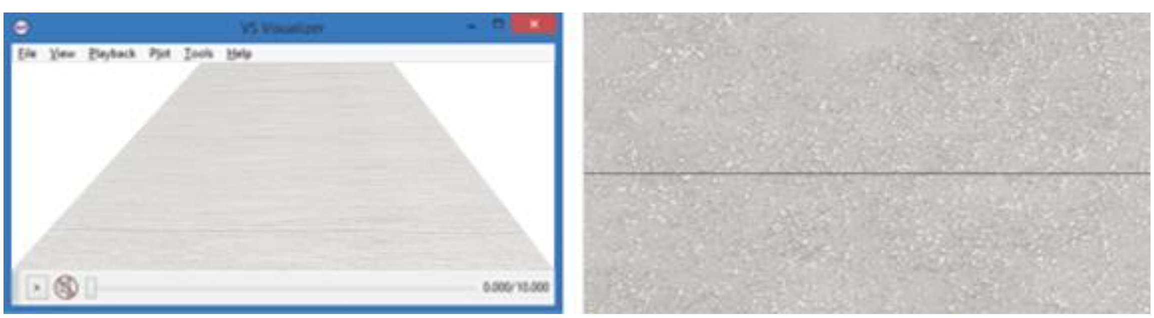


Figure 26. Concrete (Curb).

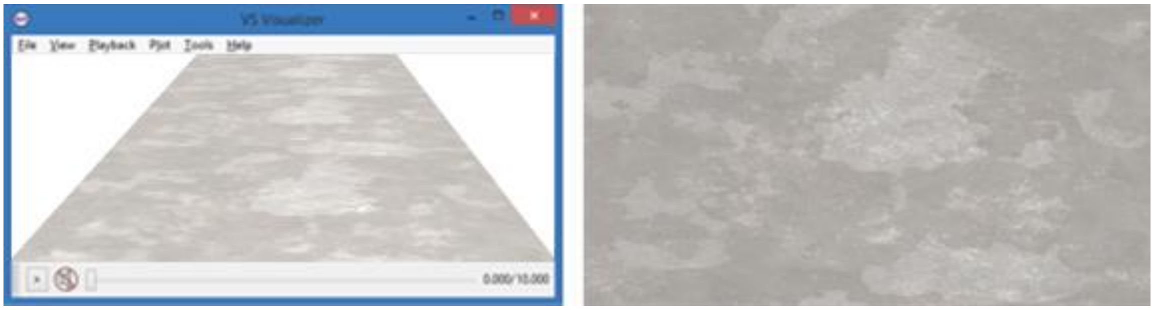


Figure 27. Concrete (Worn).

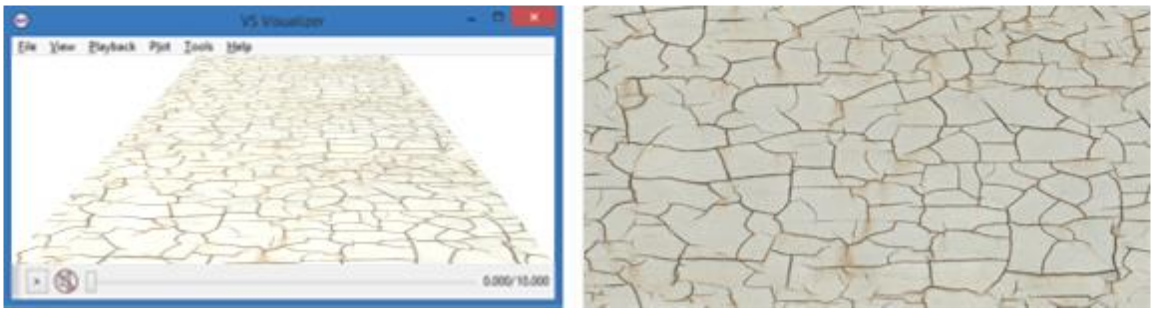


Figure 28. Cracked Paint.

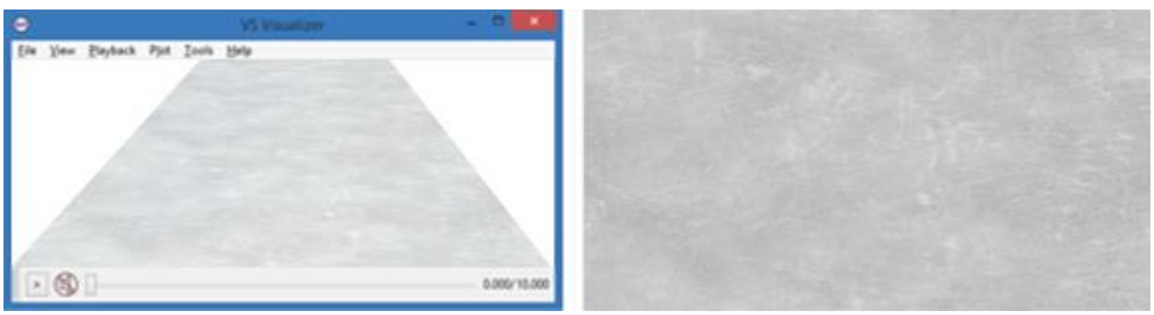


Figure 29. Ice.

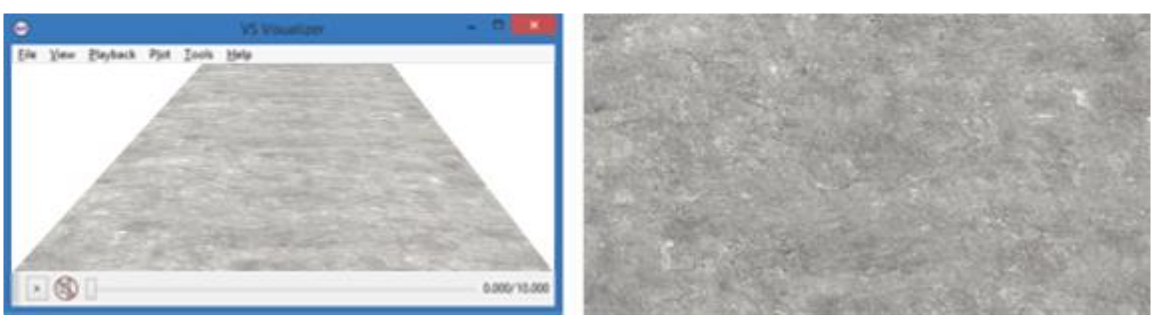


Figure 30. Slate (White).

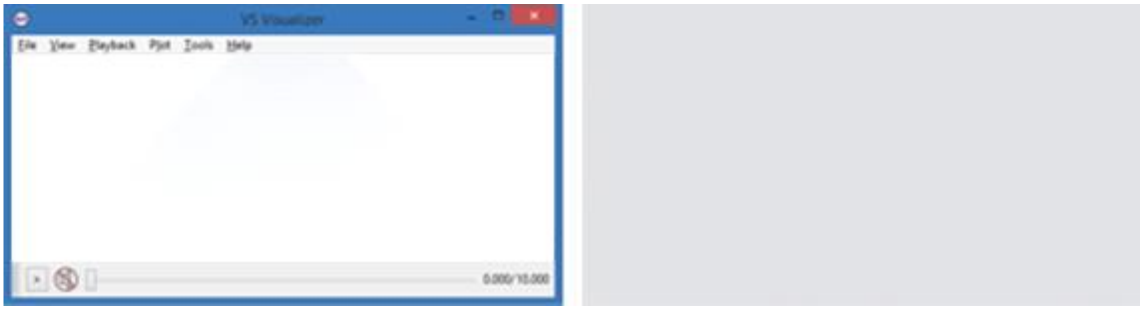


Figure 31. Snow.

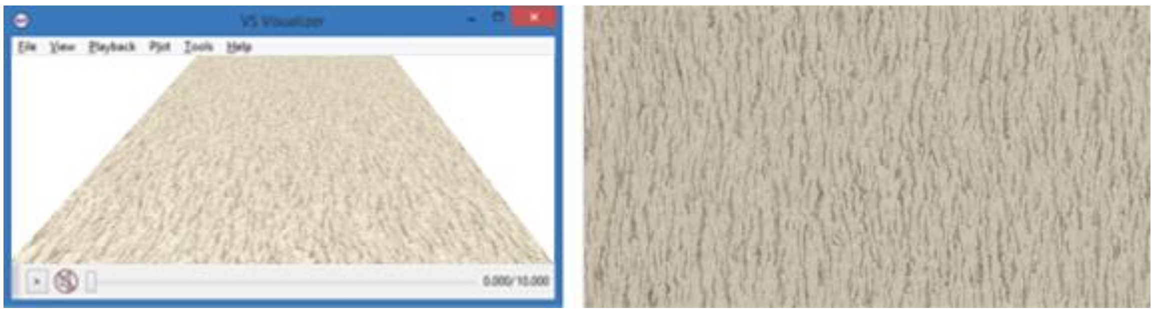


Figure 32. Stucco (Extreme).

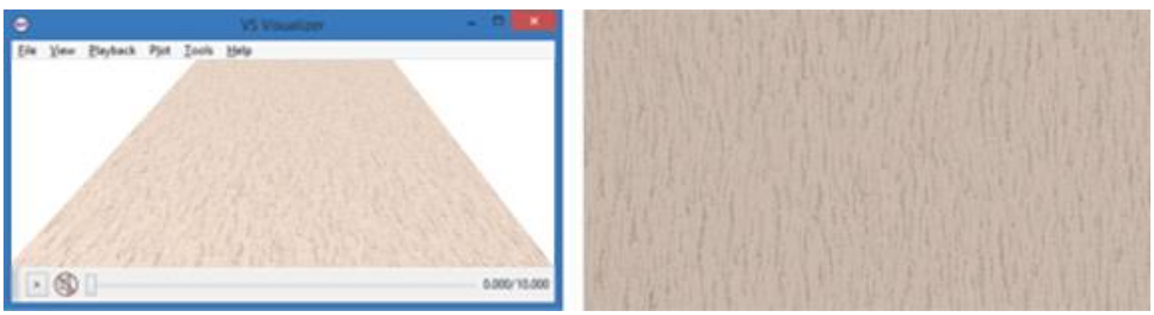


Figure 33. Stucco (Heavy).

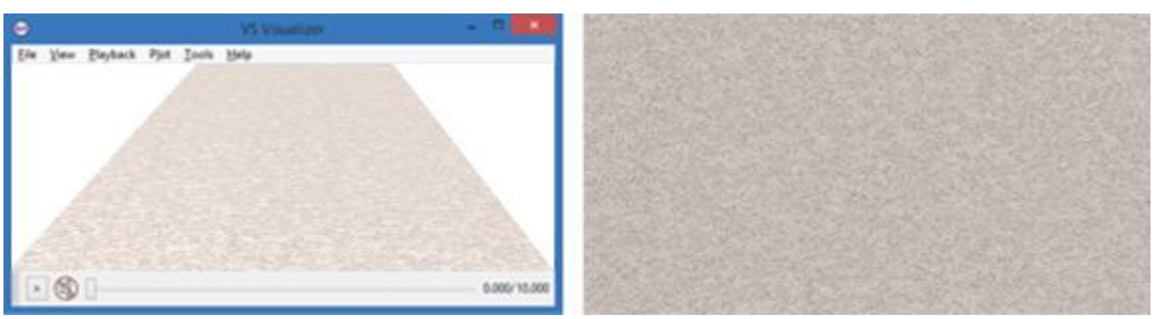


Figure 34. Stucco (Medium).

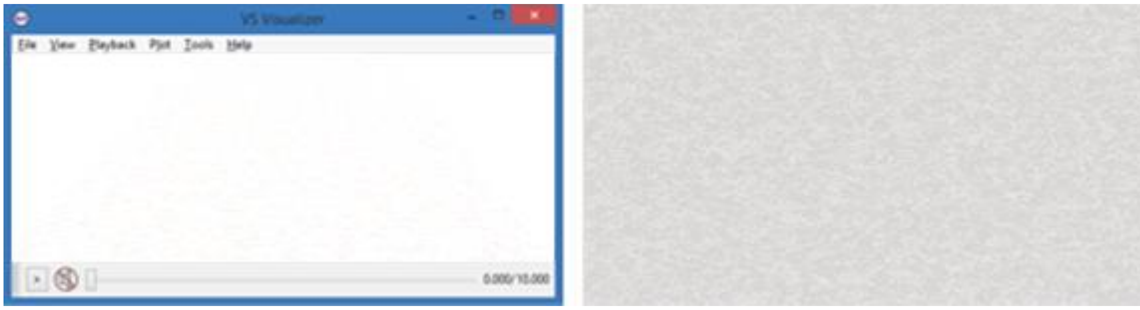


Figure 35. Stucco (Smooth).

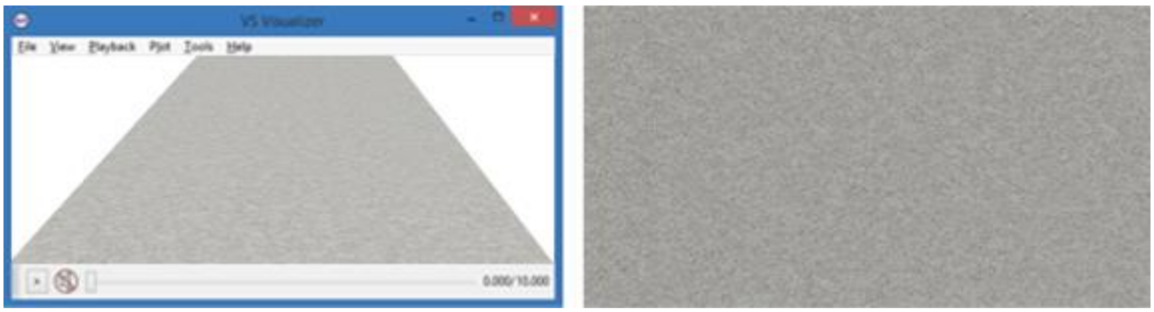


Figure 36. Texture Gray.

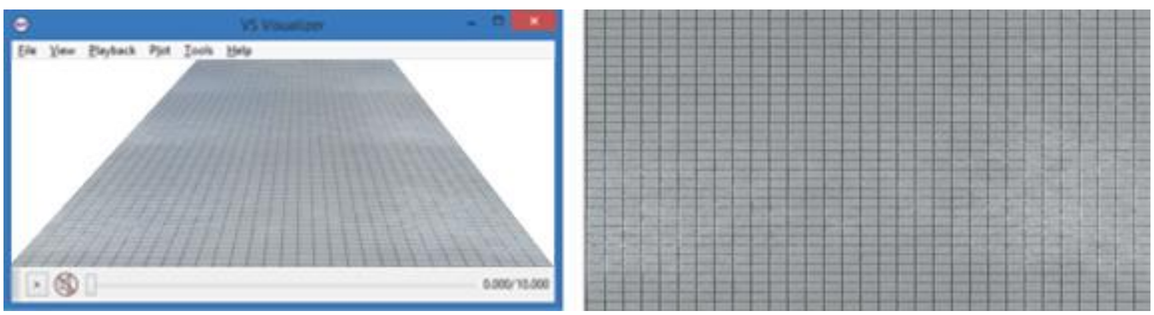


Figure 37. Tile Ice.

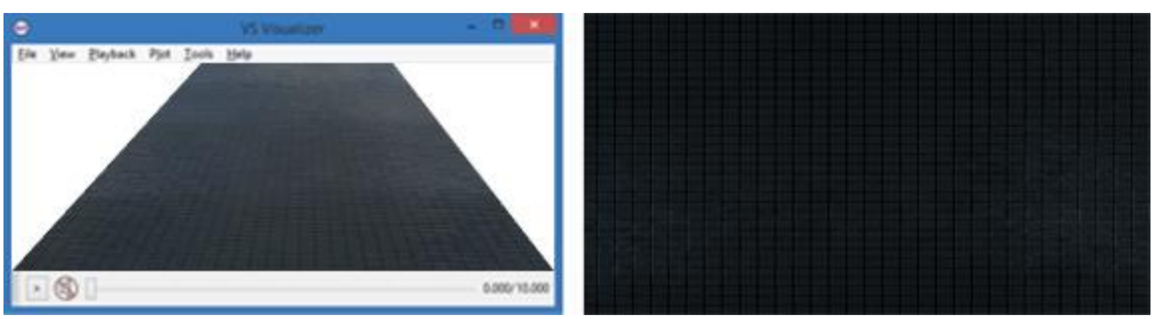


Figure 38. Tile Wet.

Miscellaneous

These materials are used to add miscellaneous details. For these surfaces, you will want to set the S Scale value to 4 and the L Tiles value to 1. This will repeat the material (texture) once every 4 meters and tiles once across the surface.

The examples below are using S Scale = 20, L Tiles = 1 and Detail= "Foliage" on a strip with width = 20m (Figure 39 to Figure 41).

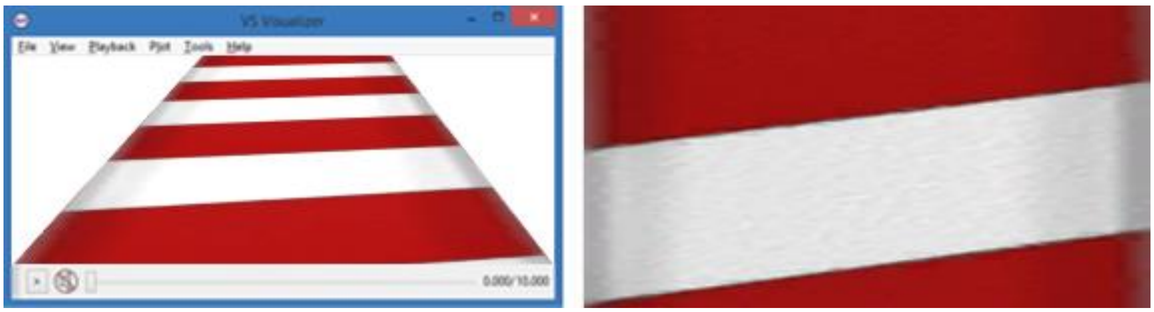


Figure 39. Apex Curbing.

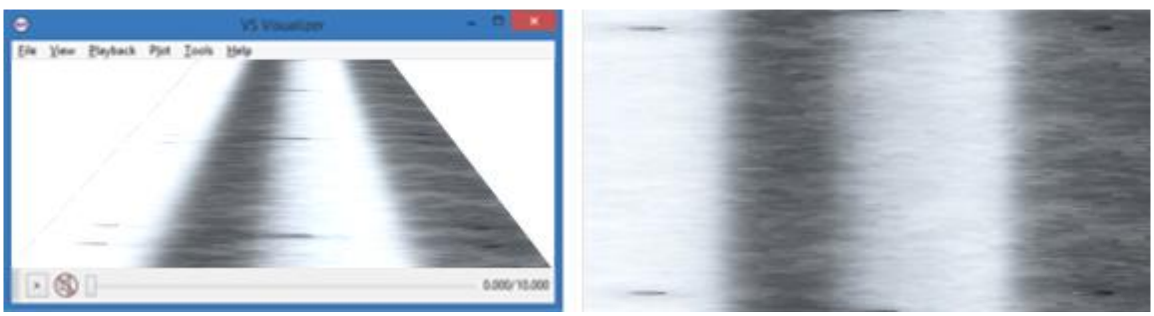


Figure 40. Guardrail.

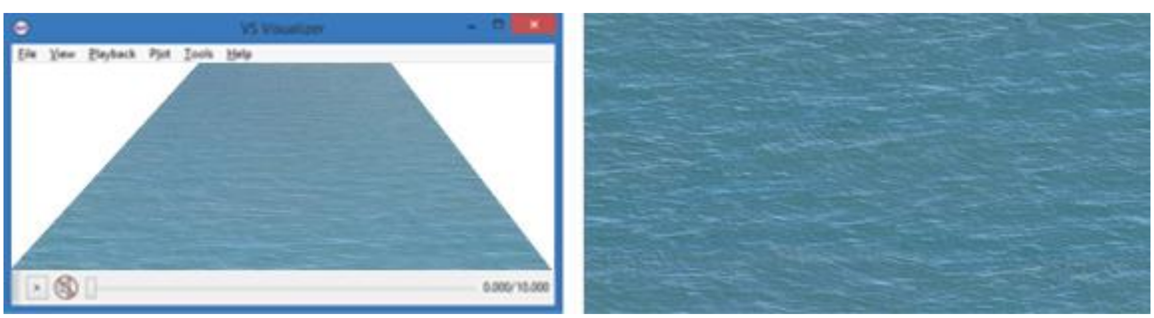


Figure 41. Water.

Painted Markings

These materials are used to add miscellaneous painted road markings. For these surfaces, you will want to set the S Scale value to match the length, L Tiles value to 1 and a dZ value of 0.05. This will give you a tile repeat of 1 across the surface and will "float" the image slightly off the road surface for proper rendering (Figure 42 to Figure 47).

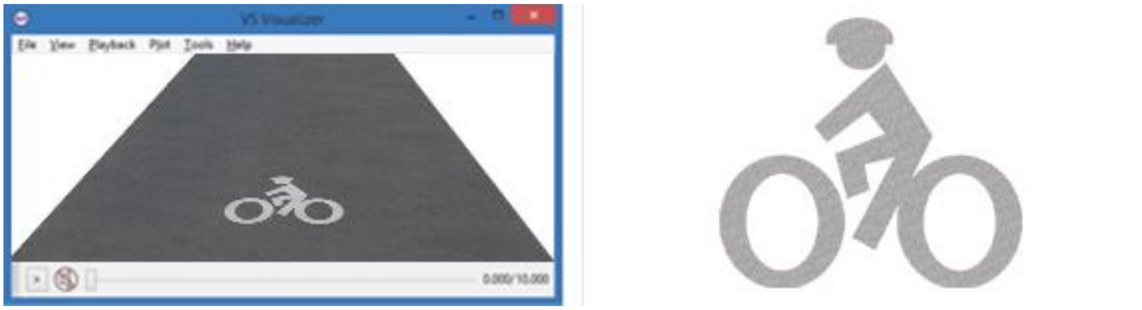


Figure 42. US Bike Lane.

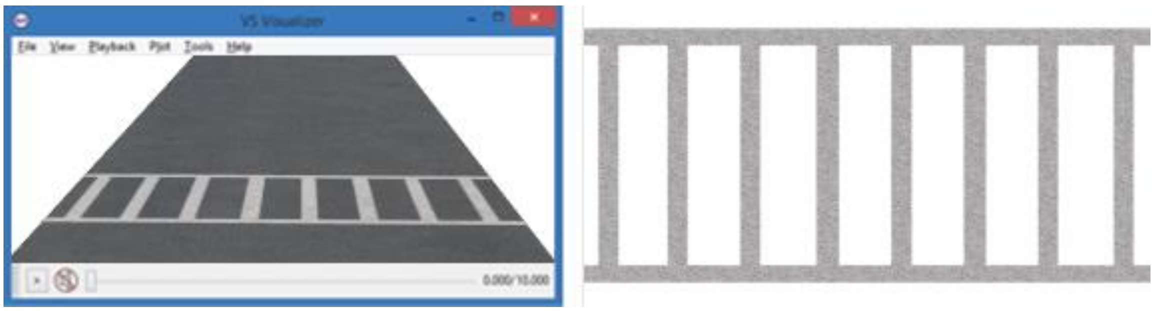


Figure 43. US Crosswalk.

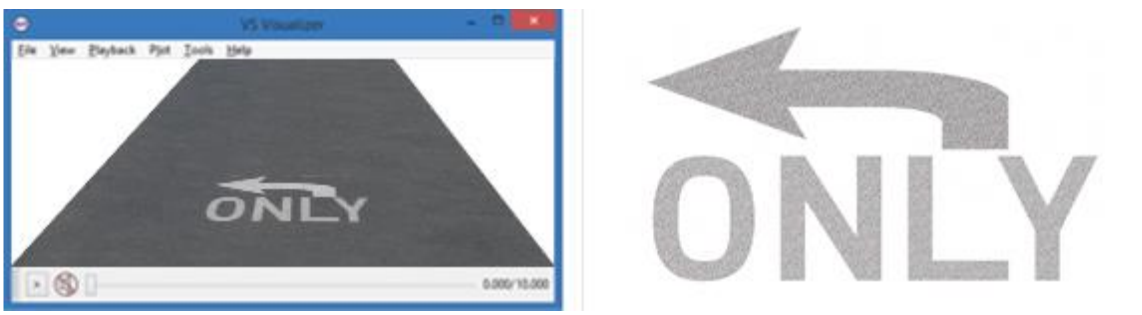


Figure 44. US Left Turn Only.

Figure 45. US Right Turn Only.

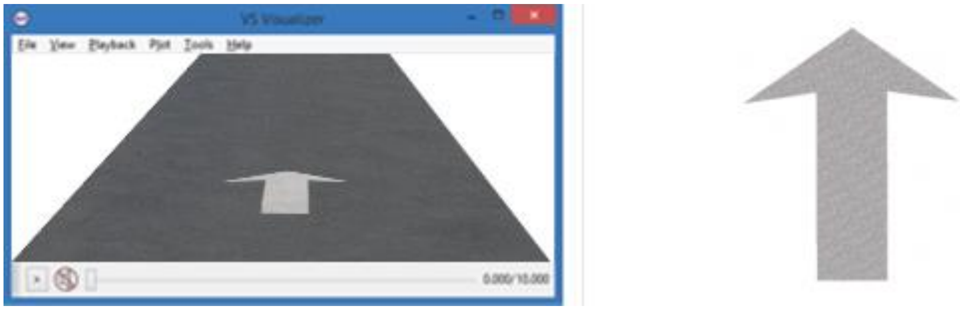


Figure 46. US Straight Arrow.

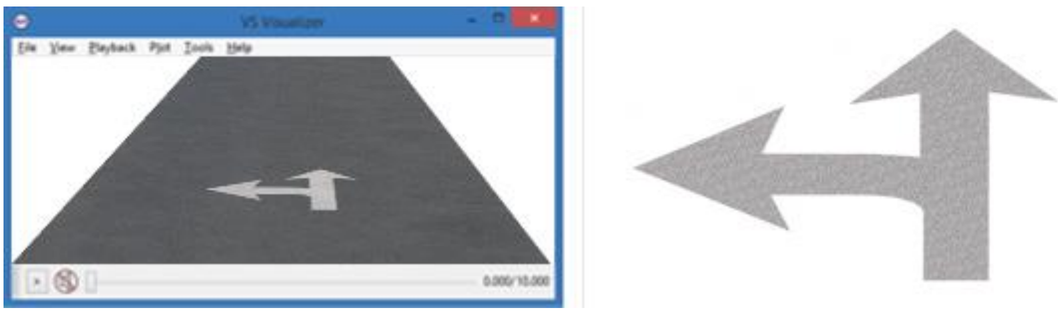


Figure 47. US Straight/Turn.

Painted Markings: Lines

These materials are used to add painted road lines. For these surfaces, you will want to set the S Scale value to 4, the L Tiles value to 1 and a dZ value of 0.05. This will give you a road line that repeats once every 4 meters and tiles once across the surface and will “float” the painted line slightly off the road surface for proper rendering.

The examples below are using S Scale = 4, L Tiles = 1 and Detail= “Road” on a 0.3m strip in the center of the road (Figure 48 and Figure 49).

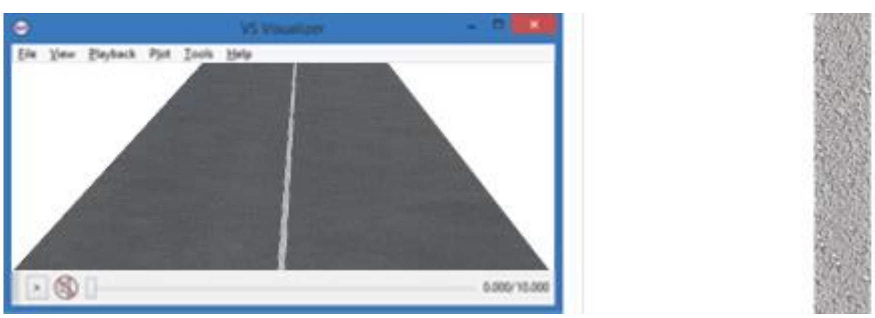


Figure 48. White Line.

Figure 49. Yellow Line.

Road Lanes

These material types have asphalt with painted lines embedded in the graphic. For the majority of road lanes, you will want to set the S Scale value to 20 and an L Tiles value to 1. This will give you a road material (texture) that repeats once every 20 meters longitudinally and tiles once laterally across the road surface.

The examples below are using an S Scale = 20, L Tiles = 1 and an Detail= “Road” on a strip with width = 20m surface (Figure 50 to Figure 56).

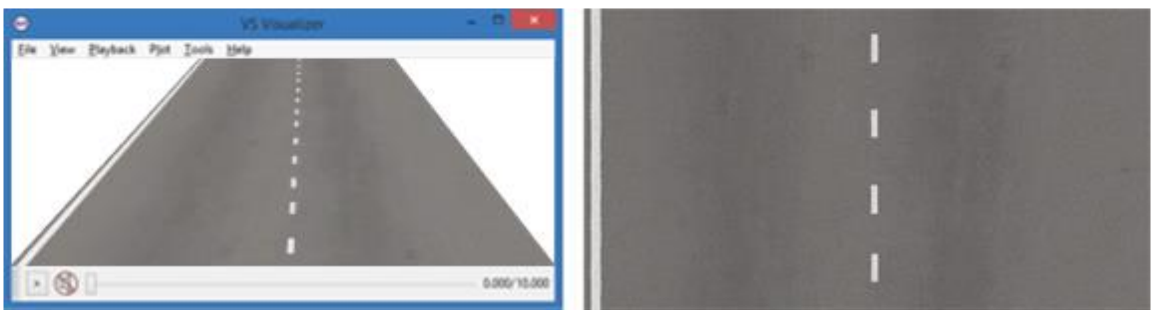


Figure 50. 4 Strip Left Lane.

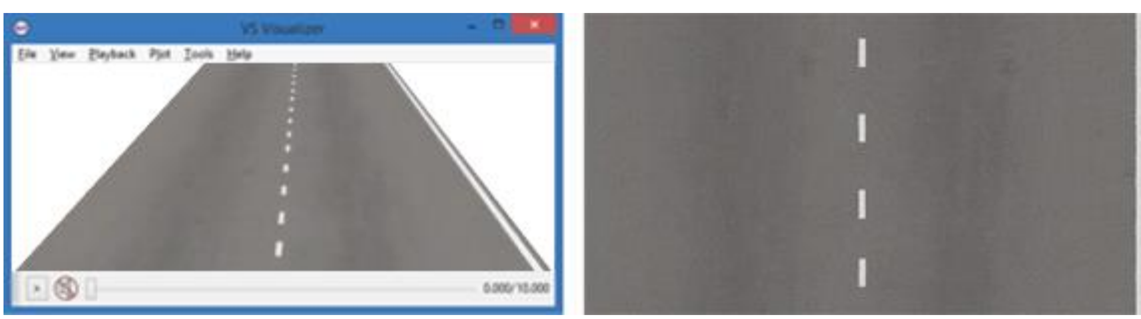


Figure 51. 4 Lane Right Lane.

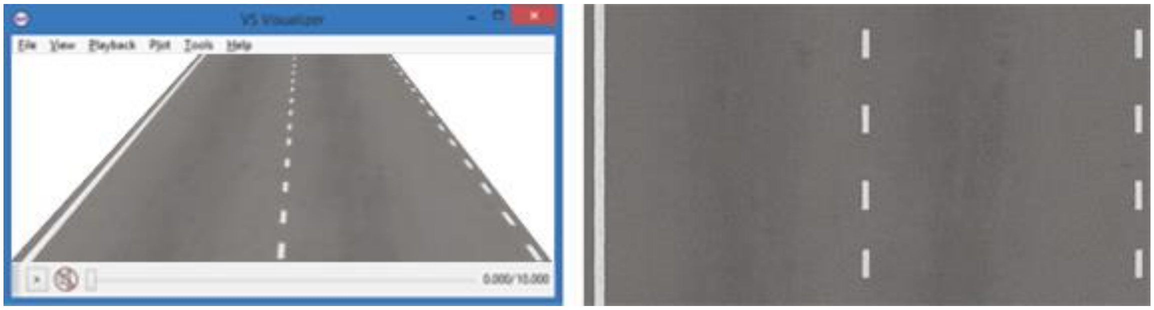


Figure 52. 5 Lane Left Lane.

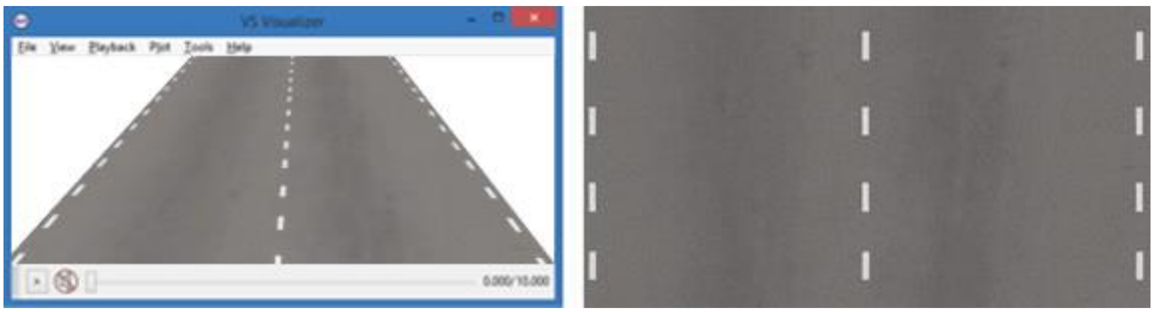


Figure 53. 5 Strip Middle Lane.

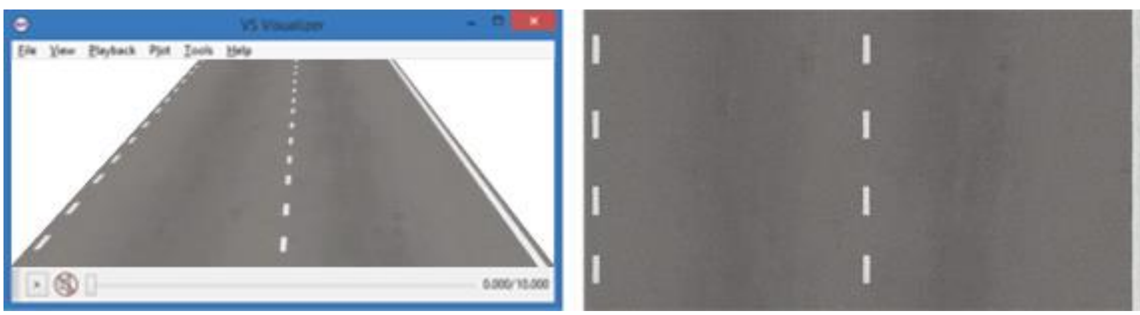


Figure 54. 5 Strip Right Lane.

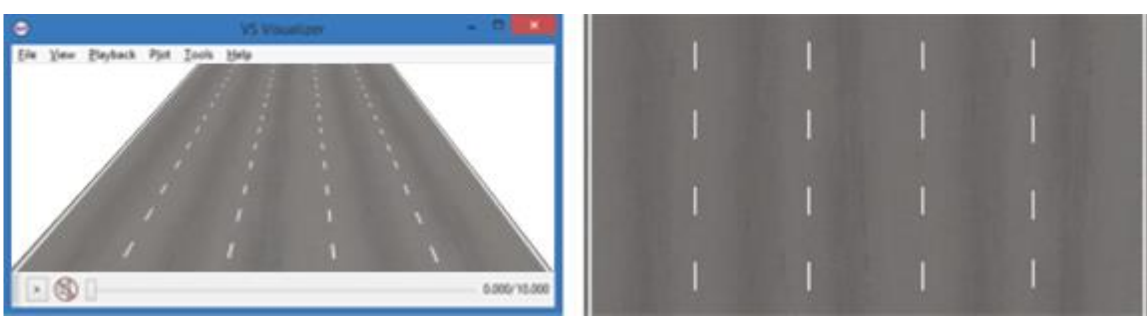


Figure 55. 5 Lane Combined.

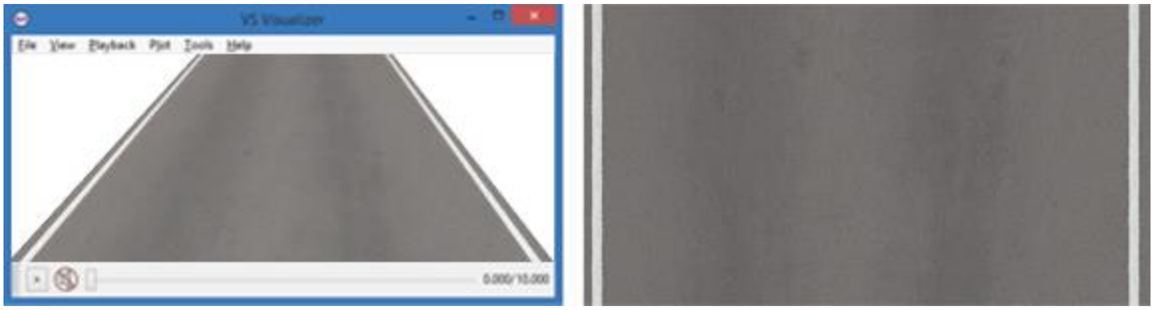


Figure 56. 5 Road (One Lane).

Road Surfaces

These material types have asphalt or concrete surfaces, often with painted lines embedded in the graphic. For the majority of road surfaces you will want to set the S Scale value to 20 and an L Tiles value to 1. This will give you a road material (texture) that repeats once every 20 meters longitudinally and tiles once laterally across the road surface.

The examples below are using an S Scale = 20, L Tiles = 1 and an Detail= “Road” on a strip with width = 20m (Figure 57 to Figure 70).

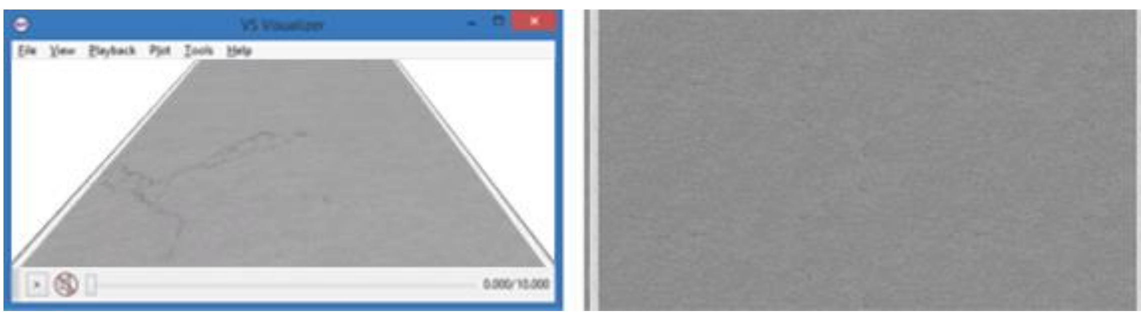


Figure 57. Asphalt (Fine One Strip).

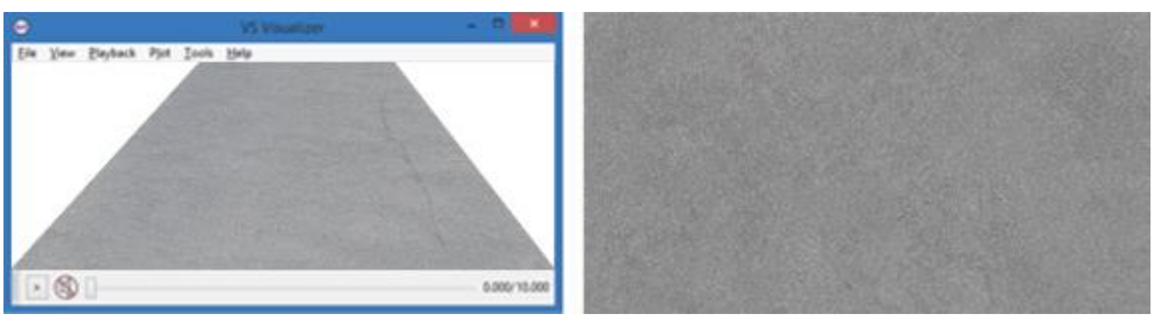


Figure 58. Asphalt (Fine).

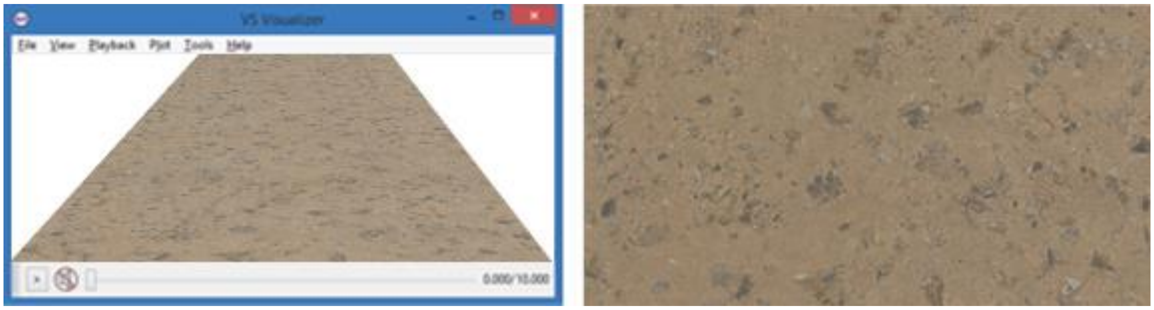


Figure 59. Dirt.

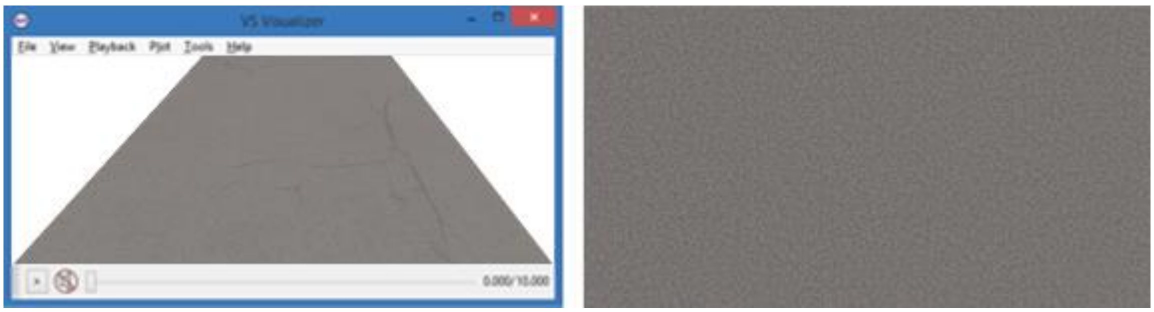


Figure 60. Pavement.

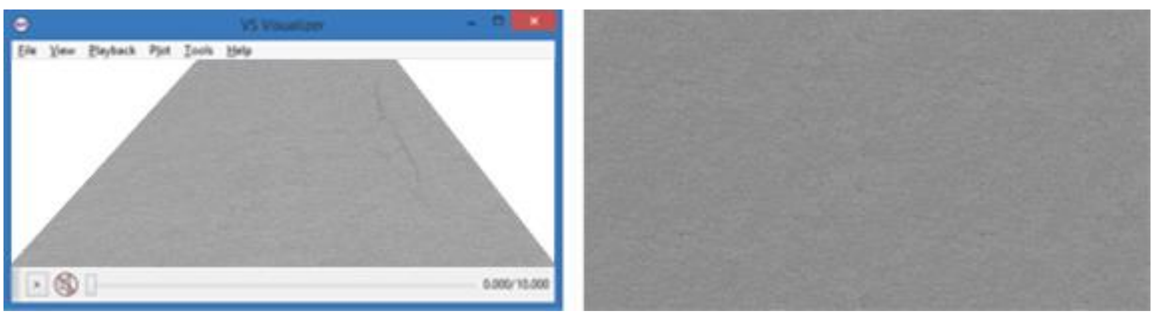


Figure 61. Road (Concrete No Lines).

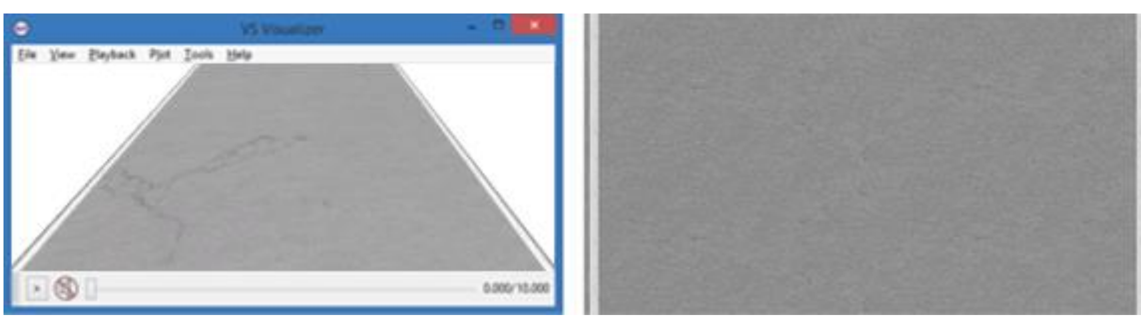


Figure 62. Road (Concrete).

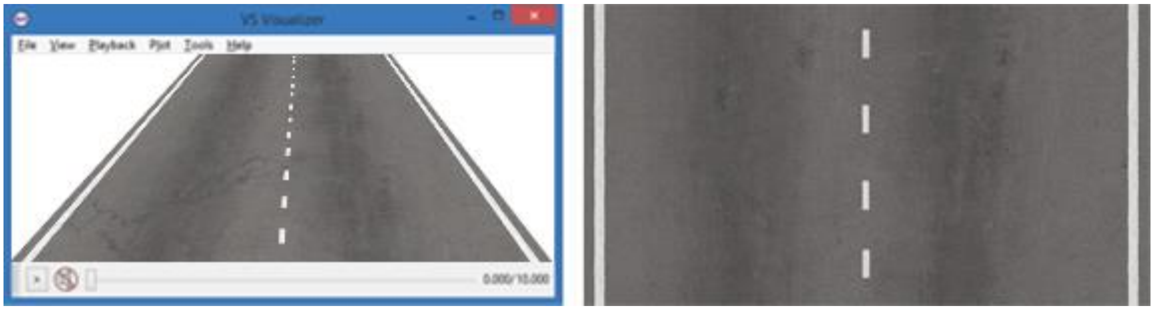


Figure 63. Road (Medium).

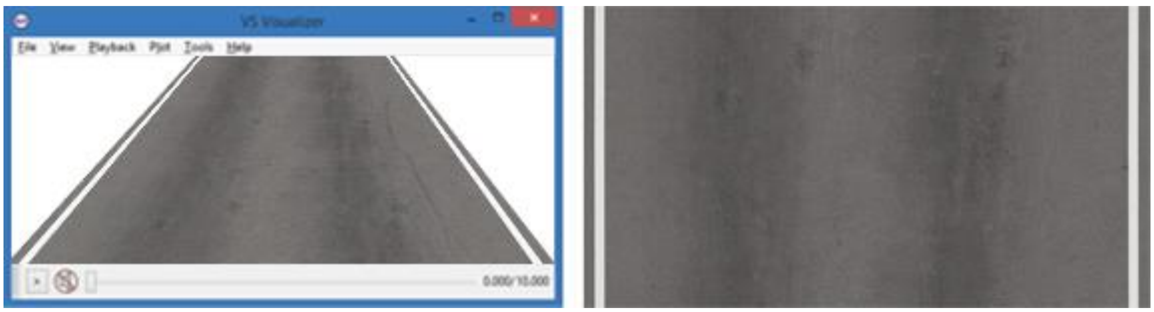


Figure 64. Road (No Centerline).

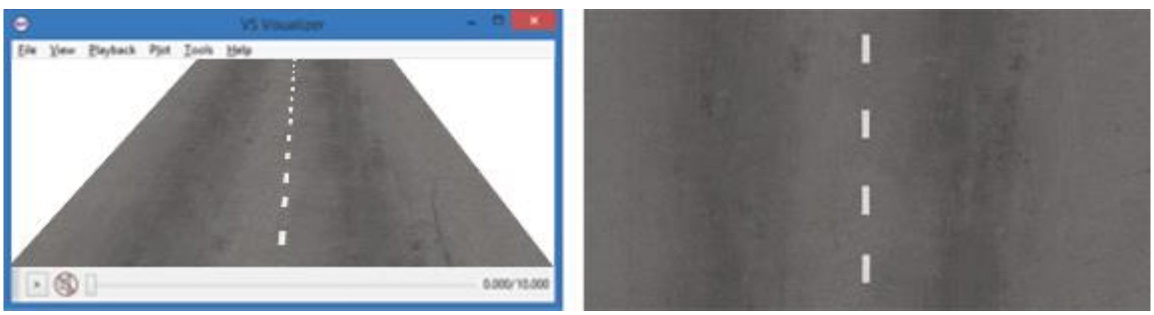


Figure 65. Road (No Edge Lines).

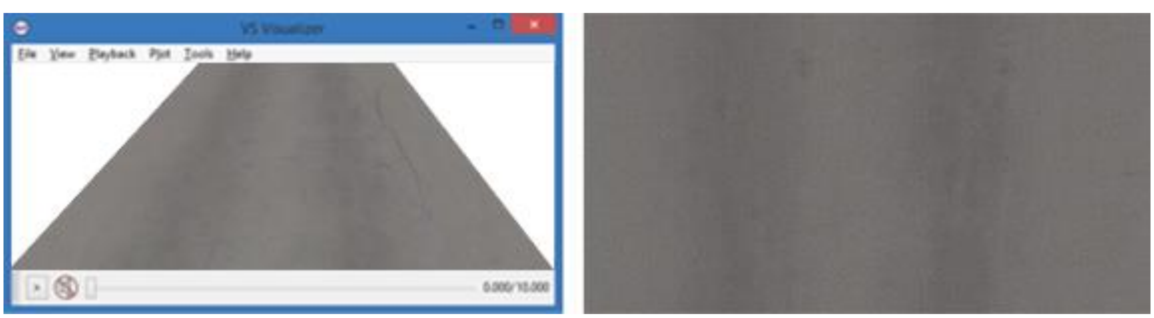


Figure 66. Road (No Lines).

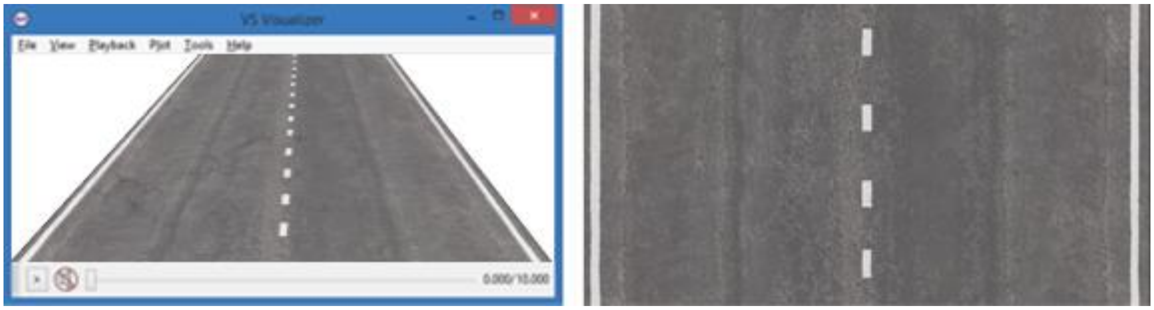


Figure 67. Road (Rough).

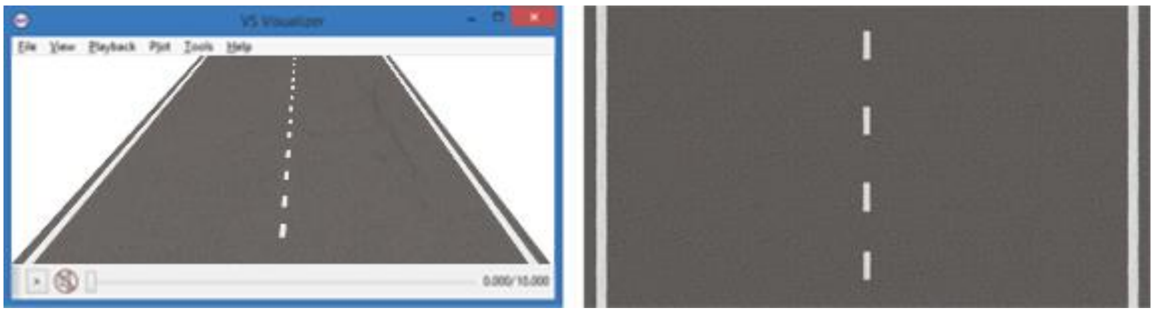


Figure 68. Road (Smooth).

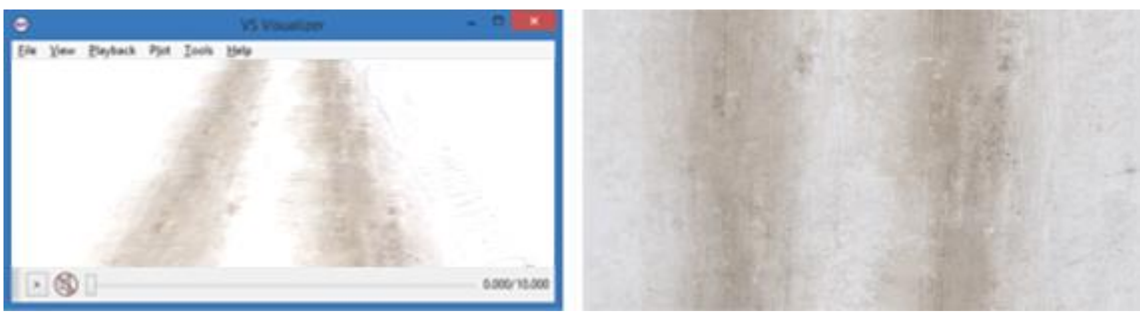


Figure 69. Road (Snow).

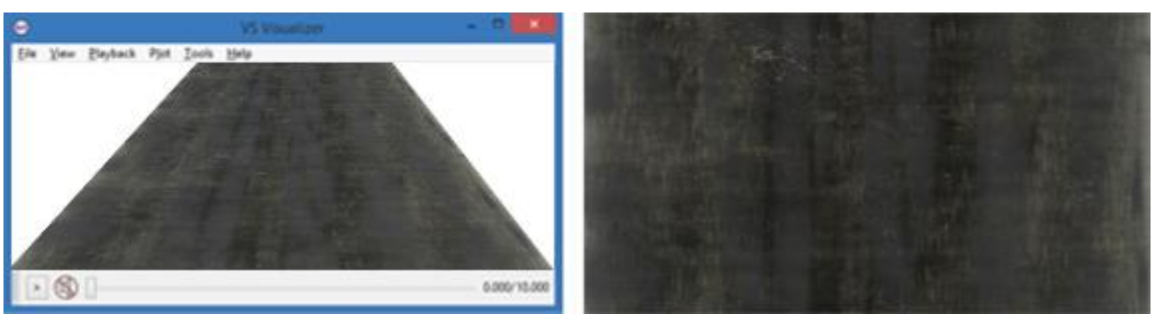


Figure 70. Road (Tarmac).

Transitions

These material types are used to transition from road surfaces to grass, snow and dirt areas. For the majority of transition surfaces, you will want to set the S Scale value to 20 and an L Tiles value to 1. This will give you a transition material (texture) that repeats once every 20 meters and tiles once across the surface.

The examples below are using S Scale = 20, L Tiles = 1 and Detail= “Foliage” on a strip with width = 20m (Figure 71 to Figure 77).

Figure 71. Dirt Transition.

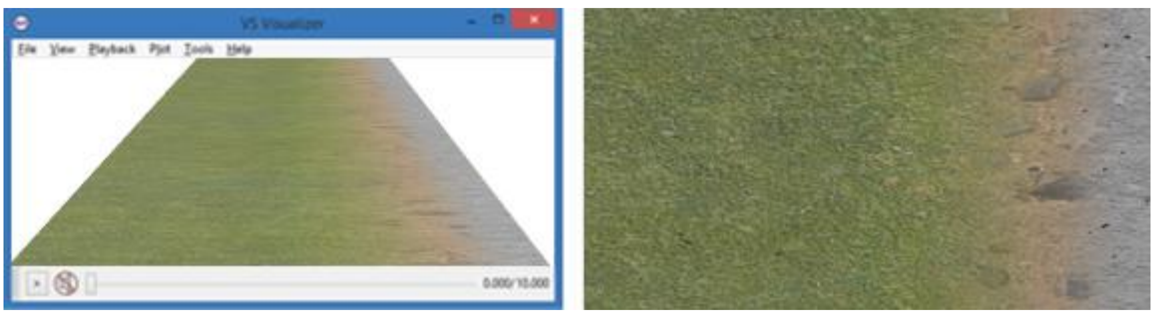


Figure 72. Road Transition (Concrete).

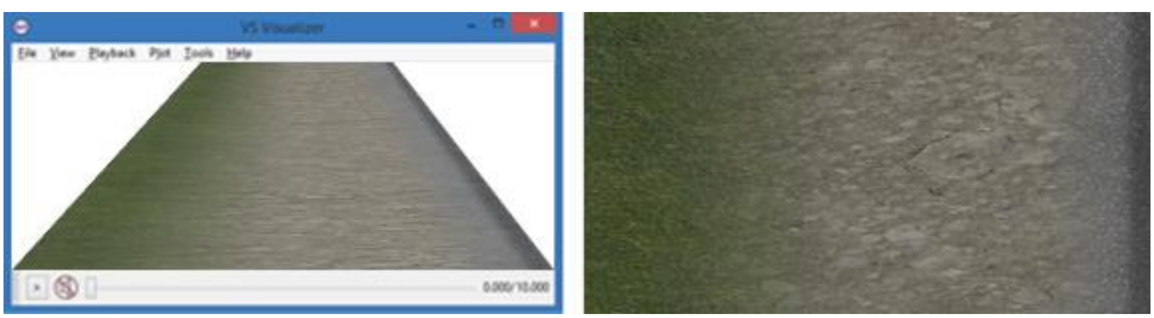


Figure 73. Road Transition (Dark).

Figure 74. Road Transition (Light).

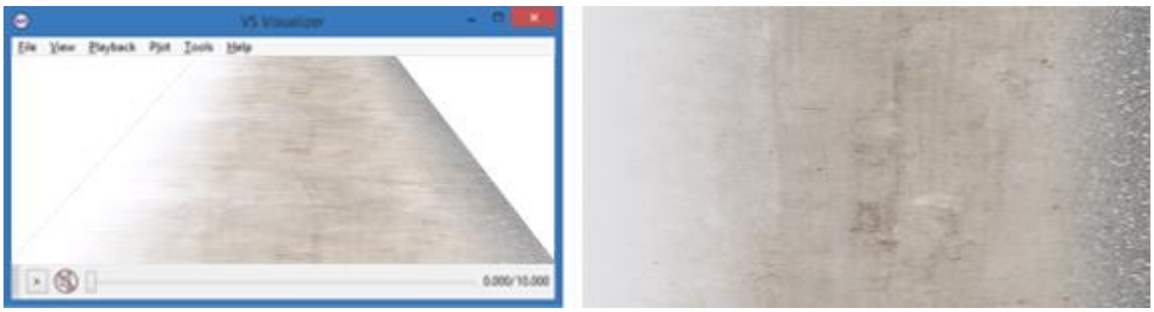


Figure 75. Road Transition (Snow).

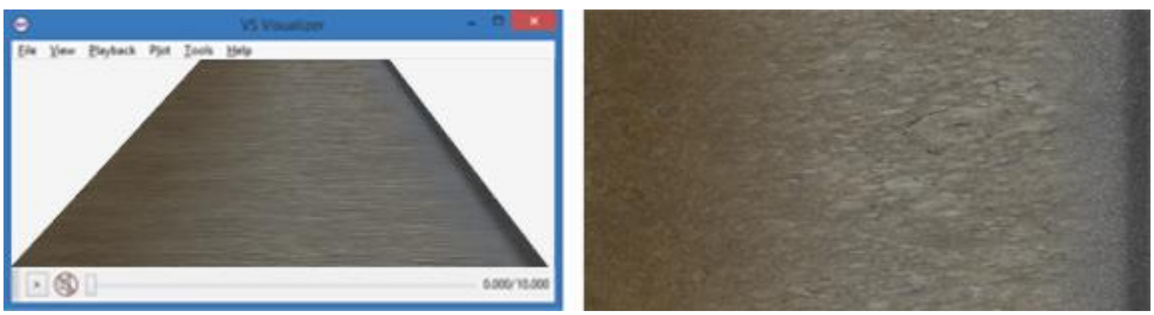


Figure 76. Transition (Dirt Dark).

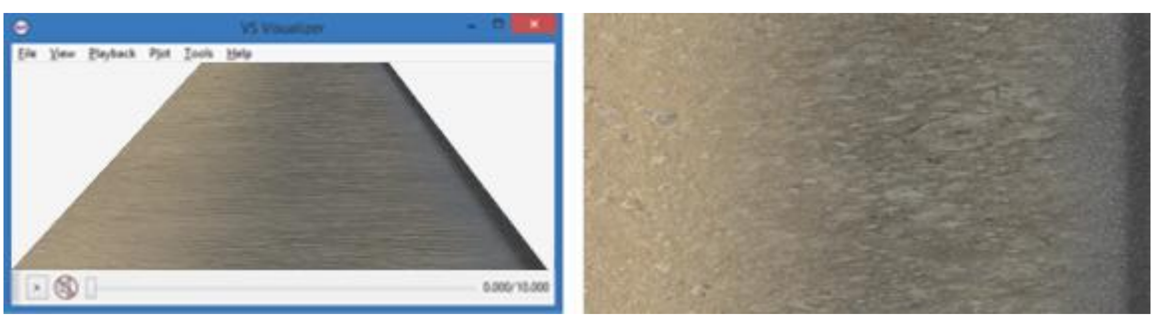


Figure 77. Transition (Dirt Light).

The Road: Animator Repeated Object Screen

Use this screen to assemble a group of objects with the same shape along the road using either road S-L coordinates or global X-Y coordinates (Figure 78). Links to datasets from this library are always made from the **Roads: 3D Surface** screen, 5 in Figure 2 (page 3).

The purpose of this screen is to allow a common type of object such as a traffic cone or roadside tree to be copied many times and located using either road S-L coordinates or global X-Y coordinates.

By itself, the information on this screen is not sufficient to define the locations of the repeated object. However, when combined with the horizontal and vertical geometry in linked datasets from the **Roads: 3D Surface** screen, there is enough information to transform the coordinates.

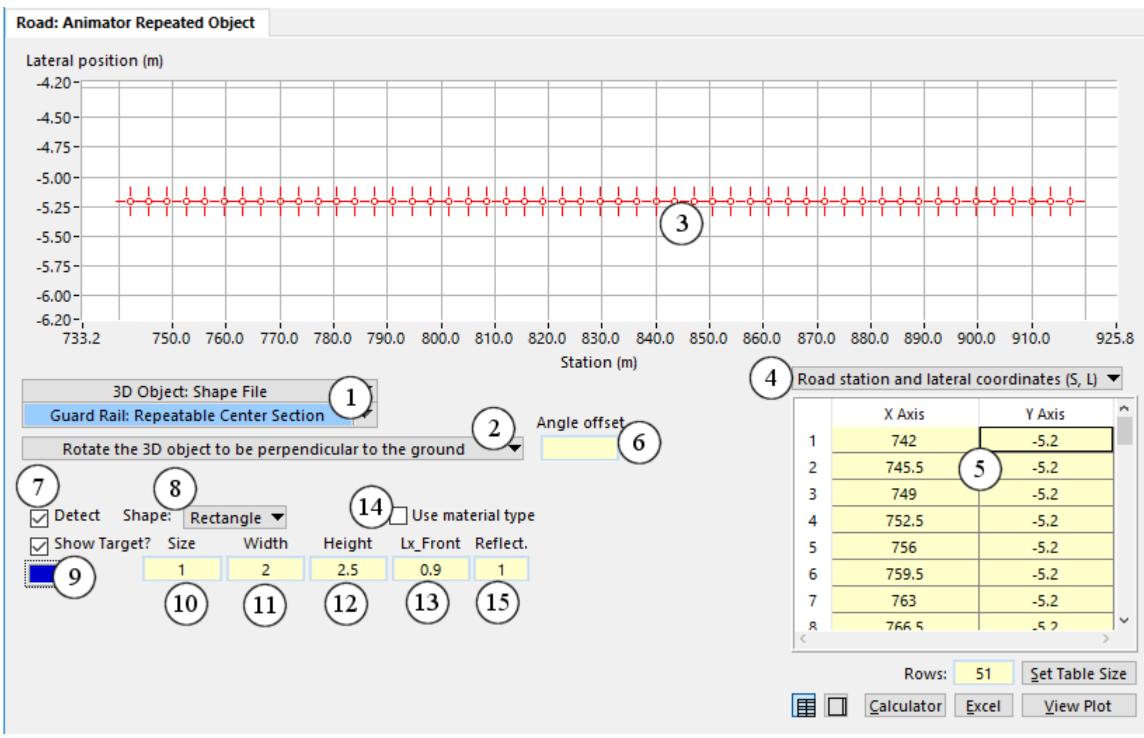


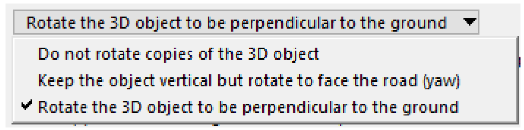
Figure 78. The Road: Animator Repeated Object screen.

The transformations are done by the same routines used in the VS Solvers when the **Roads: 3D Surface** screen is visible. Because the transformation depends on several datasets, including the one defined with this screen, the transformations should be updated whenever a change is made to any of the linked datasets, including this one.

1 Link to an animator object or group. This object or group will be shown by the animator at every location defined in the table 5.

2 Drop-down list for specifying rotation options for the repeated object. Besides calculating X-Y-Z coordinates for the copies of the animator object, the **Roads: 3D Surface** screen display the objects in rotated positions using the three options available in this control. The second option rotates the object to face the road and works well with buildings and trees. The last

option, to keep the object perpendicular to the ground, is recommended for traffic cones and other objects whose bases should be parallel to the ground or road surface.

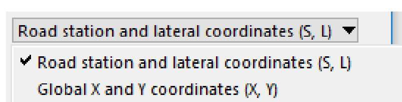


3 Plot of the locations of the repeated object. The plot shows the contents of the table 5 graphically. If the S-L coordinates are selected in the drop-down list 4, lateral position appears on the vertical axis and station on the horizontal axis. When applied by the animator, the line of zero lateral position will follow the road design path as it curves through 3D space. If the X-Y coordinates are selected in the drop-down list 4, the Y coordinate appears on the vertical axis and the X coordinate on the horizontal axis. The plot is scaled automatically to fill the available space.

Since adding objects such as trees may be done better visually than by coordinates of points, this plot has a unique set of controls. These controls are listed below and can be accessed by right mouse button clicking on the plot.

1. **Add Point**. Ctrl + left mouse button.
2. **Select Point**. Left mouse button on point.
3. **Move Nearest Point**. Left mouse button on empty area of graph.
4. **Move Selected Point**. Left mouse button drag selected point.
5. **Delete Selected Point**. Delete or Backspace.

4 Drop-down list for specifying the type of coordinates. Options are S-L coordinates or X-Y coordinates.



5 Table of locations. Each row in this table will result in a copy being shown of the animator object or group 1. Each row has two numbers: either S and L or X and Y depending on the selection of the drop-down list 4. The conversion of these values to global X-Y-Z values is done from the **Roads: 3D Surface** screen, using the horizontal and vertical geometry datasets that are also linked to that screen.

The values are shown and edited using the same spreadsheet or text field used for other tables, as described in *VehicleSim Browser Reference Manual*, and the same buttons and controls are provided under the data.

6 Angle offset will rotate each instance of the object by the specified amount in degrees. This offset will apply to the object based on the orientation selected by 2 option.

7 The detect checkbox will enable object detection, if an ADAS sensor is placed on the ego vehicle. This checkbox unhides several controls that allow the user to specify shape properties of the object.

8 Drop-down list of shape types. **Cylinder**, **Rectangle**, and **Segment** all have a common behavior of creating one detectable shape per data point in the table.

If **Polygon** is selected, then a single object will be created for the entire table 5. The data points in the table will be interpreted as a single shape, connected by a linear polygon strip.

9 Checkbox to enable the bounding volume in the visualizer. Once enabled, the VsVisualizer will display the detectable volume of the object with a transparent solid color shape. The color can be specified by the color picker.

10 Primary dimension of object as detected by a sensor. If the object is detected as a rectangle 8, then this is the length (keyword = LENGTH_OBJ). If the object is detected as a cylinder, then this is the diameter (keyword = DIAMETER_OBJ).

11 Width of rectangular object as detected by a sensor (keyword = WIDTH_OBJ). This field is hidden if the object is a cylinder. When the object is a segment, the field becomes an input for the visibility angle. The visibility angle limits how far from the front heading of the object the viewing sensor can be located before the object is considered undetected.

12 Height of the point detected by a sensor (keyword = H_OBJ). This value is used for two purposes. First, it is used to determine the pitch angle connecting the sensor point to the object, and possibly, to reduce the magnitude of the sensor output based on sensitivity to elevation angle. Second, it is used to scale the animation of the sensor target if the checkbox 9 is checked to show a bounding volume.

The VS Math Models support two methods of determining the height of the detection point. The choice between the two methods is made using the sign of the height parameter. If a positive value is given, then it locates the target point above the ground by the specified distance. If a negative value is given, then the math model sets the target height to match the nominal height of the sensor above the ground. This causes the pitch angle connecting the sensor point to the object to be approximately parallel to the ground, and usually indicates a line to a point on the target at a height that is closest to the sensor.

If the height is given a positive value, and the target shape is animated, the animated shape is scaled vertically with twice the height, such that the target will be defined at half the height of the object. If the height is given a negative value to indicate that the target point depends on the sensor, and the target shape is animated, then the absolute value of the number is used to scale the shape vertically.

13 Distance from object reference point to the front of a rectangle (keyword = `LX_FRONT_OBJ`). The point associated with the motion of the object is not necessarily in the center. For example, many of the animation datasets for vehicles are set up relative to a reference point at the center of the front axle. In these cases, this distance would go from the front axle to the front surface of the vehicle that would be detected by a sensor. This field is hidden if the object is not a rectangle.

14 Checkbox to specify how reflectiveness is defined for each object. Each object has an integer parameter `MATERIAL_TYPE_OBJ` that is interpreted as a material type if the value is positive. In that case, the material type is used in a table together with a sensor type to

determine the reflectiveness of the object for that type of sensor (see the section **The Sensor Reflectiveness Map Screen**, page **Error! Bookmark not defined.**).

If the box is unchecked, then `MATERIAL_TYPE_OBJ` is set to 0 to indicate that a single reflectiveness value for the object applies for all types of sensors.

The material type or reflectiveness of each object is given in a yellow field ⑮. If this box is checked, the values in the fields are material types; if the box is not checked, the values in the fields are reflectiveness values.

⑮ Material type or reflectiveness for the object.

If the box ⑭ is not checked, then a single reflectiveness value is used for the object regardless of the type of sensor. In this case, the keyword for the data field is `REFLECT_OBJ`, and a line is automatically written into the parsfile `MATERIAL_TYPE_OBJ 0` to indicate that the sensor type is not considered.

The value for this field is a floating-point number between 0.0 and 1.0. A value of 0 indicates the object will be invisible to sensors: it cannot be detected, nor can it occlude other objects that are detectable.

A value of -1 will make the object visible to sensors even through occluding objects.

If the box ⑭ is checked, then the keyword for the data field is `MATERIAL_TYPE_OBJ`, with an integer value that indicates the type of material. The reflectiveness is then obtained from a table that has reflectiveness values for combinations of material type and sensor type.

Further details on how the objects will behave can be found in the *ADAS Sensors and Target Objects* document.