

**PROCEDURAL GENERATION OF
BRIDGES AND TUNNELS**

MASTERS THESIS

CHAITANYA KAPU

N.C.C.A BOURNEMOUTH UNIVERSITY

20 August, 2010

LIST OF CONTENTS

1. Introduction	8
2. Research and Related Work	9
3. Technical Background	11
4. Road Tools Digital Assets	12
4.1 Terrain Asset	12
4.2 Controls Asset	13
4.3 Path Calculations asset	13
4.3.1 Full Path Generation	13
4.3.2 Bridge Path Generation	14
4.3.3 Tunnel Path Generation	15
4.3.4 Level Path Generation	16
4.3.5 Primary Calculations	17
4.3.6 Bridge Path Calculations	18
4.3.7 Secondary Calculations	25
4.3.8 Height Calculations	26
4.3.9 Small or Large Primitives	26
4.3.10 Second Offset Calculations	27
4.3.11 Bridge Support Calculations	27
4.4 Road Asset	28
4.4.1 Level Sharp Corners	29
4.5 Detailed Bridge Asset	31
4.5.1 Cable Curvature Calculations	31
4.5.2 Bridge Orientation Calculations	32

4.6 Tunnel Asset	33
4.7 Level Terrain Asset	34
4.7.1 Profile for levelling	34
4.8 Procedural Texture Asset	36
5. Conclusion	38
6. Renders	39

LIST OF FIGURES

Fig1: Bridge that is taken as reference for cables of large bridge design(Braces are made differently)	9
Fig2: Rectangular base tunnel profile that is taken as reference for design	9
Fig3: Showing procedural tree generated(Kelly and McCabe, 2006)	10
Fig4: Photo realistic scenery with procedural geometry and textures by M.Giuli ,Terragen artist (Kelly and McCabe , 2006)	10
Fig5: Road tunnel in Paris, France(FHWA, 2006)	11
Fig6: Calculations made in a vopsop for generating a random terrain	12
Fig7: Full calculated path generated for the input curve	13
Fig8: Road generated on the calculated path	13
Fig9: Calculated area for bridges	14
Fig10: Bridges generated on the calculated path	14
Fig11: Calculated area for tunnels for the input curve	15
Fig12: Tunnels generated on the calculated path	15
Fig13: Calculated area for levelling of the input curve	16
Fig14: Wireframe mode showing level area on terrain	16
Fig15: Calculations made in a vopsop for finding the primary bridge and tunnel paths	17
Fig16: Input curve given	18
Fig17: Black indicates tunnel area and red indicates bridge area	18
Fig18: Part of the bridge area to be straightened.	19
Fig19:Delete all points except ends in order to make it straight	19
Fig20:Select one of the two points to move	19

Fig21:Path formed after moving the selected point in such a way that it has the same y position as the other point	19
Fig22: Area with arc length greater than divide once range partitioned once	20
Fig23: Resultant after making the points of a primitive have same Y pos	20
Fig24:Disconnected Curve after bridge path straightening	20
Fig25:Bridge path at start of the curve to be divided once(4 points)	21
Fig26:Bridge path at start of the curve divided once(4 points)	21
Fig27:Bridge path at end of the curve to be divided once(4 points)	21
Fig28:Bridge path at end of the curve divided once(4 points)	21
Fig29:Bridge path in between the curve to be divided once(5 points)	21
Fig30:Bridge path in between the curve divided once(5 points)	21
Fig31:Bridge path at start of the curve to be divided twice(5 points)	22
Fig32:Bridge path at start of the curve divided twice(5 points)	22
Fig33:Bridge path at end of the curve to be divided twice(5 points)	22
Fig34:Bridge path at end of the curve divided twice(5 or 4 points)	22
Fig35:Bridge path in between the curve to be divided twice(6 points)	22
Fig36:Bridge path in between the curve to be divided twice(6 points)	22
Fig37: Network showing usage of Foreach Sop for joining the curve	23
Fig38: Connected curve after calculations in Foreach Sop	24
Fig39: Colour values transformed in Y axis	24
Fig40: Curve partitioned on colour info	24
Fig41: Part of input curve for secondary calculations	25
Fig42: Resultant after secondary calculations	25
Fig43: Resultant after height calculations	25
Fig44: The red points depicted in fig. above changed to green which will later be used for levelling	25

Fig45: Height calculations network	26
Fig46 :Part of bridge area	27
Fig47 :Points of bridge part transformed on to terrain	27
Fig48 :Supports generated between the above 2 curves	27
Fig49 :After deleting some supports	27
Fig50 :Vopsop calculations in order to delete the closer points	28
Fig51 :Closest point has yellow colour	28
Fig52 :Deleted closer points	28
Fig53: Network showing the process of deleting the sharp corner points	29
Fig54: Showing the sharp corner points	30
Fig55: Result after levelling the sharp points	30
Fig56: Road generated after levelling the corner points	30
Fig57: Input given to the detailed bridge asset	31
Fig58: Bridge generated on the input provided	31
Fig59: Vopsop used for calculating the curvature of the cables	31
Fig60: Input given for curvature	32
Fig61: Curvature obtained after the above calculations	32
Fig62: Showing the input part where bridge has to be proposed	33
Fig63: Cables placed on either sides of the input curve	33
Fig64: Tunnel profile	34
Fig65: Path of curve for tunnel	34

Fig66: Tunnel along with texture generated on the input curve	34
Fig67: Input profile	35
Fig68: Profile formed after doing some calculations	35
Fig69 :Resultant after sweeping the profile on to the parts of input path	35
Fig70 : Output formed after reducing the polygons	36
Fig71: Input path to be textured	37
Fig72 : Input Profile	37
Fig73 : Output after texturing	37

Chapter 1

Introduction:

Most of the contemporary computer games, animated films need a lot of content which is procedurally created rather than static as it provides re usability, saves a lot of time and memory. Procedural content ensures that a wide variety of content can be generated from a few parameters. Procedural generation usually involves programming. Most of the 3D software packages like Maya, Houdini provide a great platform for the procedural content generation through scripts; plug-ins. Procedural techniques provide a great amount of flexibility and control which in turn provide the designer a platform for artistic freedom and experimentation.

This thesis discusses the procedural modelling and texturing of bridges and tunnels, generated on a curve given by the user. The system calculates where to place a bridge and where to place a tunnel and where to level the area based on some procedural calculations. A single digital asset file which contains 8 assets is created and is given to the end user with lots of parameters to control the look of tunnels and bridges. Side Effects Houdini was used throughout the project since its procedural work flow helps in troubleshooting the problems easily compared to other software.

The main aim of this project is to provide the user a tool which can build bridges and tunnels on a terrain with just a few clicks. About eighty percent of the project is done in the Sop level of Houdini and **Vopsops**, Foreach sops are used to do most of the important calculations. The asset is totally independent and the texture files are also added on to the asset. The texturing is done procedurally. A single asset called as Procedural Texture is used for texturing road, bridges, tunnels and terrain. The user is provided to choose the texture he requires. Lighting is not included in the tool.

Chapter 2

Research and related work:

Studying and creating procedurally generated assets is the motto of this thesis. Most of the research was done on how to achieve procedural assets in Side Effects Houdini. Side Effects tutorials, 3d Buzz Peter Claes Bridge Tutorial, Elevator Tutorial, Cmixfx Procedural generation of roads tutorial were watched in order to explore most of the powerful nodes and H-script expressions and vex builder networks. Several queries have been asked in Side Effects Houdini, odforce forums in order to have a basic idea of the approach to be taken and for solving some of the problems. Good amount of time is spent on how to create a digital asset in a most powerful way and how to make it independent.

Since most of the calculations used in the project are mainly done using position, colour attributes, mathematics behind them is studied. Time is given to study how attributes work. Lots of references have been taken to design bridges and tunnels.



Fig1:Bridge that is taken as reference for cables of large bridge design(Braces are made differently) (Edinburgh 2009).



Fig2:Rectangular base tunnel profile that is taken as reference for design(Anonymous 2010)

Several procedural methods have been proposed till date in order to generate geometry, textures. In a survey by Kelly and McCabe (2006) about procedural techniques used in city generation, several methods have been discussed. Procedural techniques were used to create random parameters for a 3d geometry, noise for textures. Recursive algorithms which are more complex such as L-systems, fractals, Perlin noise are also used in order to generate procedural content. L-systems are being used in the generation of procedural trees, Perlin noise is being used in generating procedural textures.



Fig3:Showing procedurally tree generated(Kelly and McCabe, 2006)

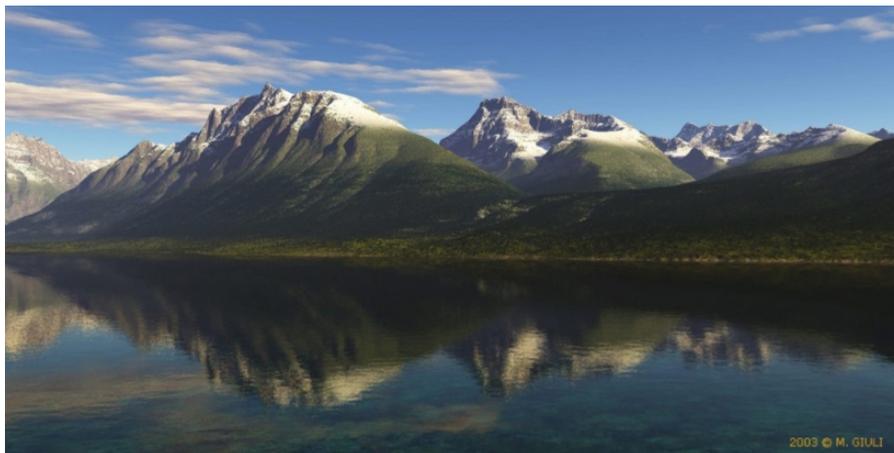


Fig4:Photorealistic scenery with procedural geometry and textures by M.Giuli ,Terragen artist (Kelly and McCabe , 2006)

Chapter 3

Technical Background:

Side Effects Houdini has been used from modelling till rendering. It is a high end 3d software package which is used extensively for creating effects and physics based simulations in feature films and games. Right from the beginning of the project, concentration is given to the procedural generation of various road elements and the mathematics, physics behind them. Good amount of flexibility is provided to the user in order to create procedural bridges and tunnels along with textures from just a curve path. All the calculations done are simple and easy to understand.

Several interesting facts about the construction of real bridges and tunnels have been studied. Technical manual for design and construction of road tunnels (FWCA, 2006) provided a deep knowledge on how exactly the tunnels are made, types of tunnels, cross section of the tunnels etc,. The following picture shows various parts and factors of a tunnel. Although much detail is not provided to the tunnel which is designed for this asset, but the study really helped in generating a basic visually good looking tunnel.

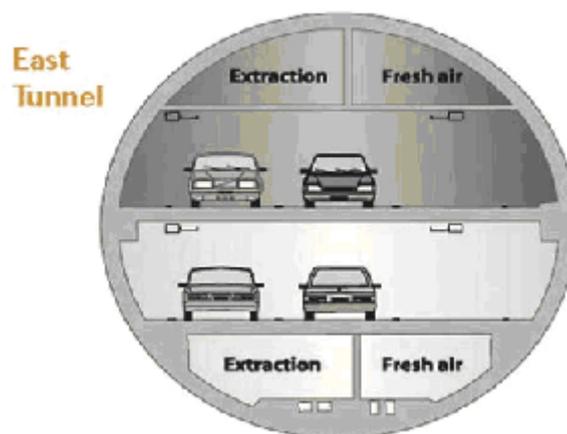


Fig5:Road tunnel in Paris, France(FHWA, 2006)

Connections: Middle click on the inputs of the node for the details about their respective connections.

Inputs: Controls should be connected to the first input and **Terrain** should be connected to the second input.

Output: Output should be connected to **Road** asset.

4.3.2. Bridge Path Generation:

In this mode, path calculations asset generates bridge path which will be helpful in generating bridges on it.

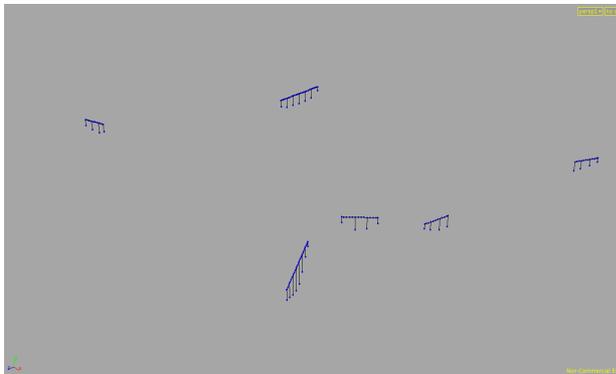


Fig9: Calculated area for bridges

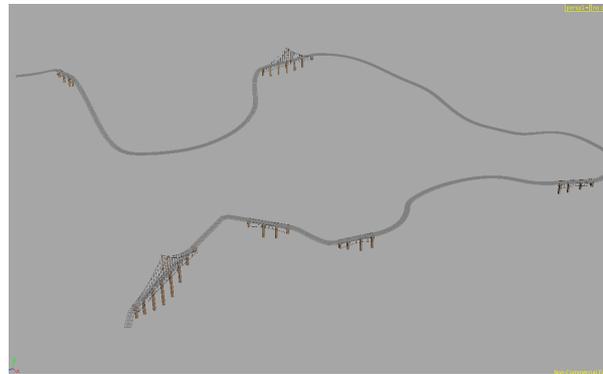


Fig10: Bridges generated on the calculated path

Steps to be followed for the proper generation of bridge path:

- * Uncheck full path parameter (as it is checked by default).
- * Uncheck colour parameter.
- * Select Bridge area.
- * Uncheck Level Area.
- * Check on Second Offset.
- * Give a second offset value and check on keep inside parameter and uncheck keep outside parameter. Outside part is used for levelling.
- * Keep only large primitives.
- * Check on bridge support parameter.

Connections: Middle click on the inputs of the node for the details about their respective connections.

Inputs: Controls should be connected to the first input and **Terrain** should be connected to the second input.

Output: Output should be connected to **Detailed Bridge** asset.

4.3.3. Tunnel Path Generation:

In this mode, path calculations asset generates tunnel path which will be helpful in generating tunnels on it.

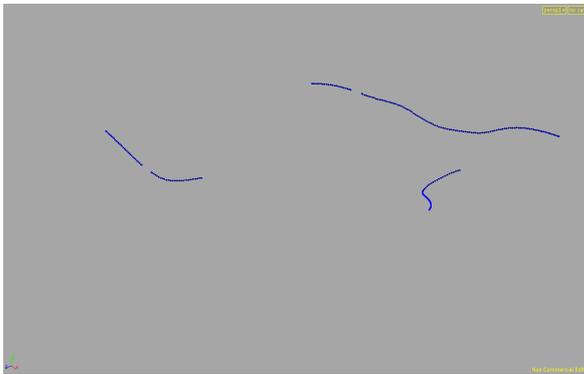


Fig11: Calculated area for tunnels for the input curve

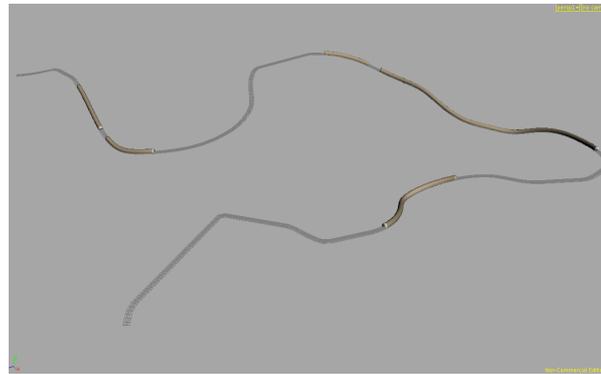


Fig12: Tunnels generated on the calculated path

Steps to be followed for the proper generation of tunnel path:

- * Uncheck full path parameter (as it is checked by default).

- * Uncheck colour parameter.

- * Select Tunnel area.

- * Uncheck Level Area.

- * Check on Second Offset.

- * Give a second offset value and check on keep inside parameter and uncheck keep outside parameter. Outside part is used for levelling.

- * Keep only large primitives.

Connections: Middle click on the inputs of the node for the details about their respective connections.

Inputs: Controls should be connected to the first input and **Terrain** should be connected to the second input.

Output: Output should be connected to **Tunnel** asset.

4.3.4. Level Path Generation:

In this mode, path calculations asset generates level path which will be helpful in levelling.

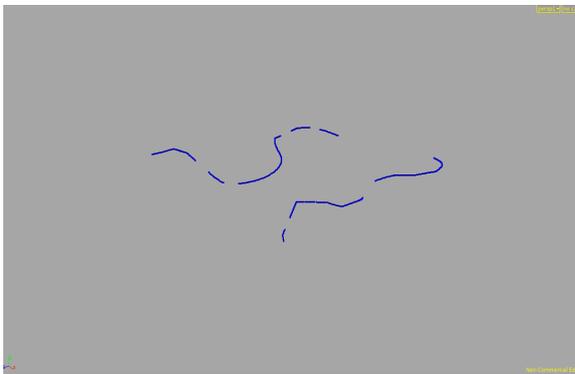


Fig13: Calculated area for levelling of the input curve

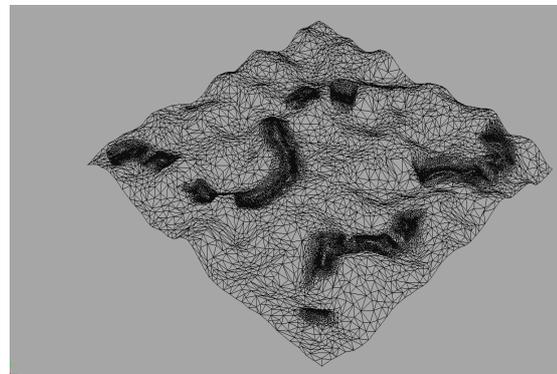


Fig14: Wireframe mode showing level area on terrain

Steps to be followed for the proper generation of level path:

- * Uncheck full path parameter (as it is checked by default).

- * Uncheck colour parameter.

- * Select both the areas (Both from the drop down menu of Select Area parameter) as levelling should be done for both the areas.

- * Check on Level Area.

- * Check on Second Offset.

- * Give a second offset value and check on keep outside parameter and uncheck keep inside parameter. Inside part is used for generation of bridge or tunnel. Value between 3 and 4 would be an ideal value.

- * Keep small and large primitives.

Connections: Middle click on the inputs of the node for the details about their respective connections.

Inputs: Controls should be connected to the first input and **Terrain** should be connected to the second input.

Output: Output should be connected to **Level Terrain** asset.

4.3.5 Primary Calculations:

Vopsop is used for calculation of bridge path and tunnel path.

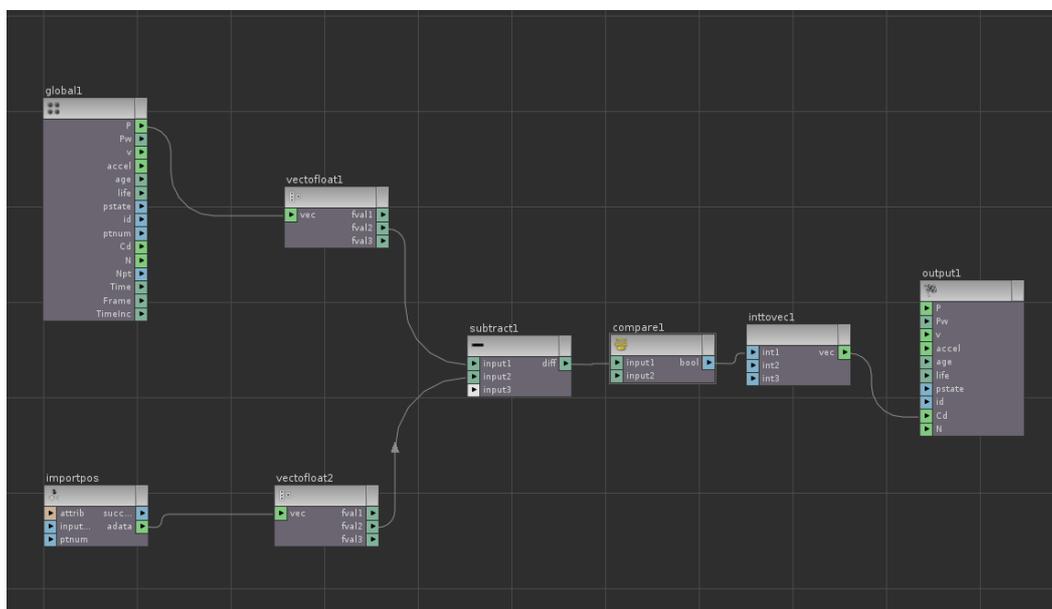


Fig15: Calculations made in a vopsop for finding the primary bridge and tunnel paths.

At first Y position of all the points on the given input curve are transformed on to the terrain using a Ray SOP. Then in a vopsop, the y-position of every point of the input curve is subtracted from the y-position of every point of the transformed curve. If the difference is less than 0, then the path is under the terrain so a tunnel has to be proposed along that area, and if the difference is greater than 0, then the path is above the terrain so a bridge has to be proposed along that area. The following pictures depict the area representation after primary calculations.

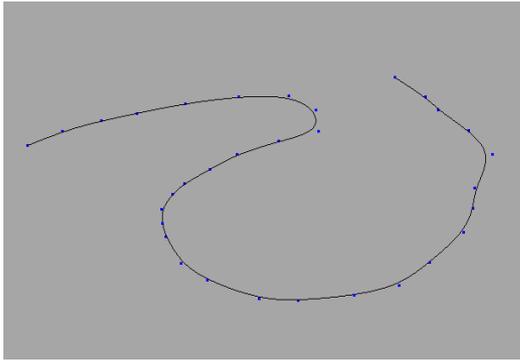


Fig16:Input curve given

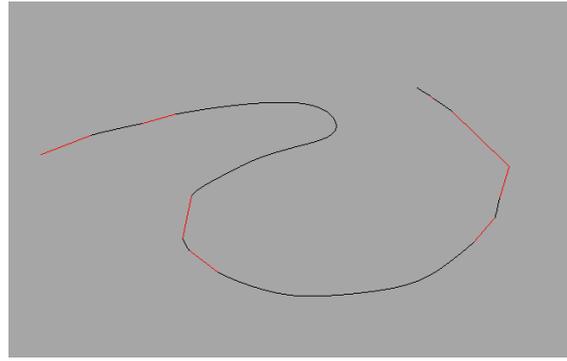


Fig17:Black indicates tunnel area and red indicates bridge area

Primary calculations provide an area for the bridges and tunnels in an initial stage. Division of the primitives is mainly done by intersecting the input curve and the curve formed by transforming the points of input curve on to the terrain using Ray SOP. This process provides a lot of flexibility to do calculations on individual areas. Different set of calculations have to be done on bridge part and different set of calculations have to be done on tunnel part. Colour information of points is used throughout this asset and it helped in doing a lot of calculations in partitioning the areas. Throughout this asset, 3 colours have been used for distinction between different areas.

Red colour indicates Bridge area.

Black colour indicates Tunnel area.

Green colour indicates Level area (This will be discussed in the later part of the thesis).

4.3.6 Bridge Path Calculations:

The basic idea which strikes the mind when constructing bridges is that the path on which a bridge is to be proposed should be straight. At first, part of the bridge area is selected and all the points on it except the ends are deleted. In order to make the points in the area to have the same y position, one of the two points should be moved such

that it has the same y position as the other. This provides a straight path with both the points on the same level.

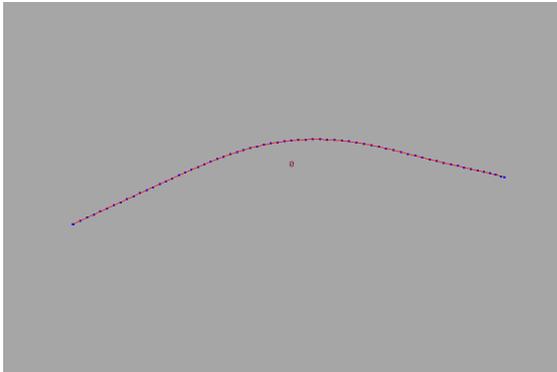


Fig18: Part of the bridge area to be straightened.

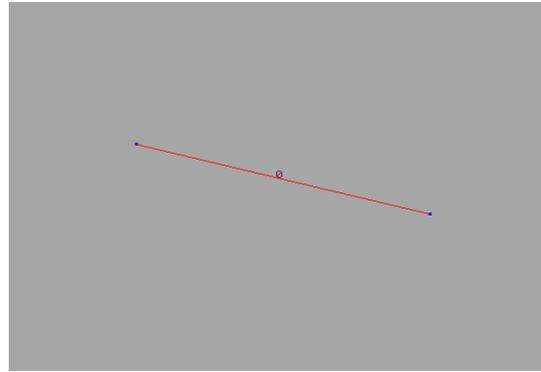


Fig19: Delete all points except ends in order to make it straight

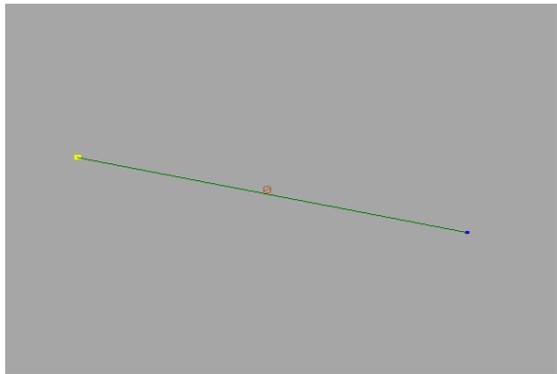


Fig20: Select one of the two points to move

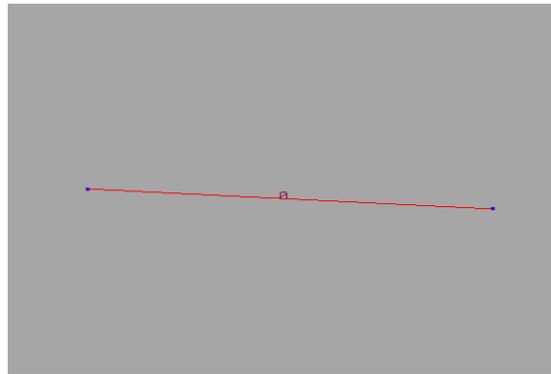


Fig21: Path formed after moving the selected point in such a way that it has the same y position as the other point

Calculate the arc length of the primitives with points of red colour. If the arc length is greater than Divide Once Range which is a parameter on Path Calculations asset, then partition the primitive once. If the arc length is greater than Divide Twice Range which is also a parameter on Path Calculations asset, then divide the curve twice. This is to ensure that the calculated path somehow follows the original input curve even after deleting the points.

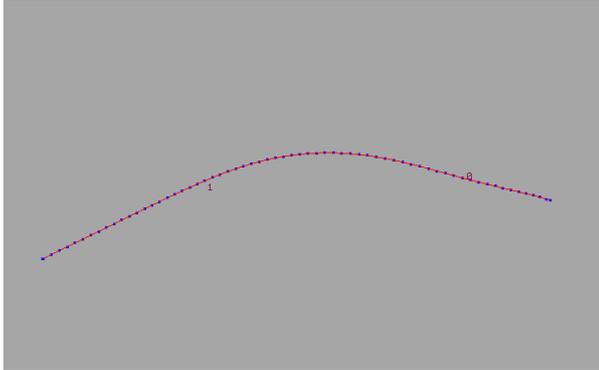


Fig22: Area with arc length greater than divide once range partitioned once.

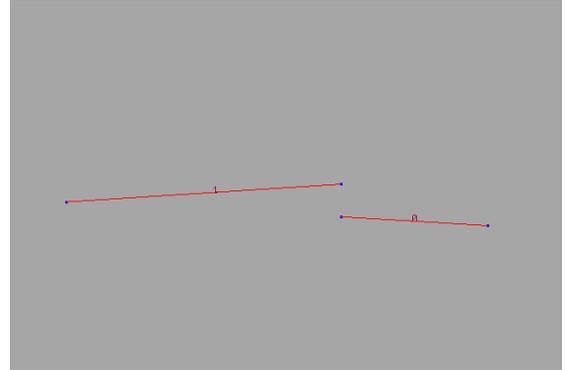


Fig23: Resultant after making the points of a primitive have same Y pos

Once the above process is done, the curve is disconnected as shown in (Fig: give number). The next step is to make it connected again. In order to achieve this Foreach Sop is used.

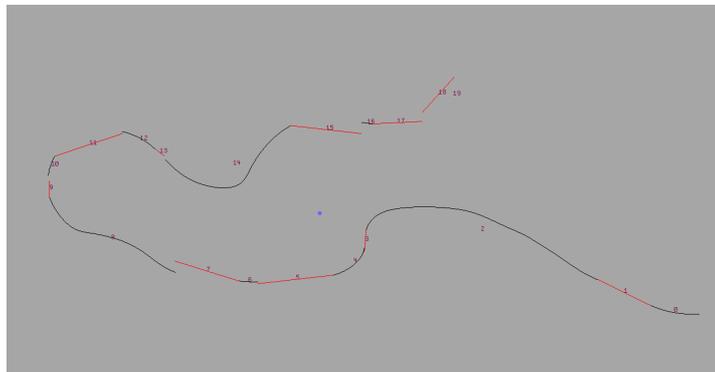


Fig24:Disconnected Curve after bridge path straightening

Cases for partitioning bridge path whose arc length greater than divide once range (2 very near points as shown in every case are formed while intersecting the original curve for respective paths).

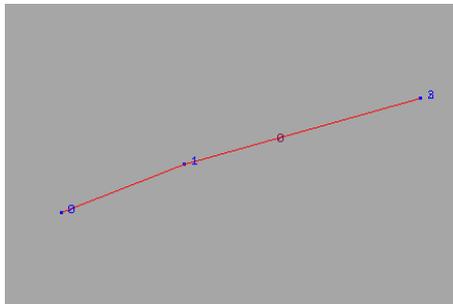


Fig25:Bridge path at start of the curve to be divided once(4 points)

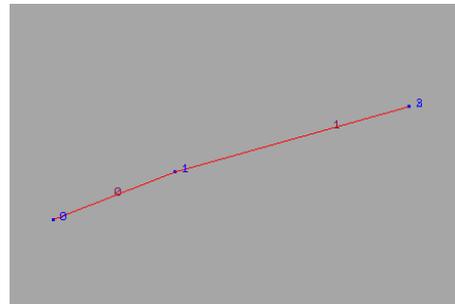


Fig26:Bridge path at start of the curve divided once(4 points)

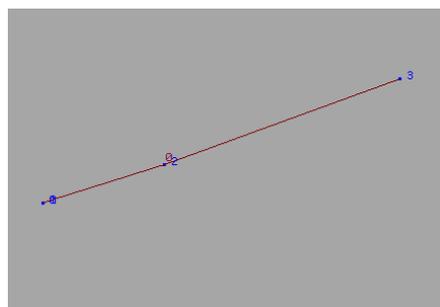


Fig27:Bridge path at end of the curve to be divided once(4 points)

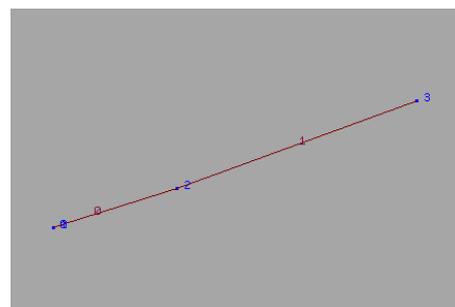


Fig28:Bridge path at end of the curve divided once(4 points)

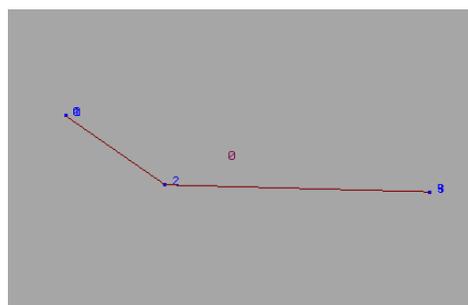


Fig29:Bridge path in between the curve to be divided once(5 points)

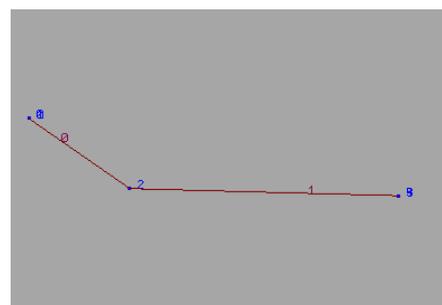


Fig30:Bridge path in between the curve divided once(5 points)

Cases for partitioning bridge path whose arc length greater than divide twice range (2 very near points as shown in every case are formed while intersecting the original curve for respective paths).

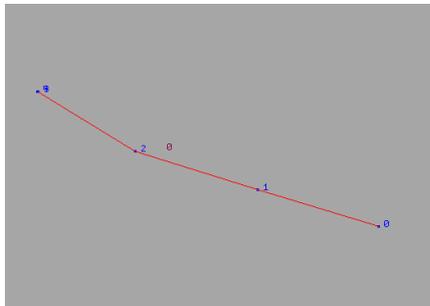


Fig31: Bridge path at start of the curve to be divided twice (5 points)

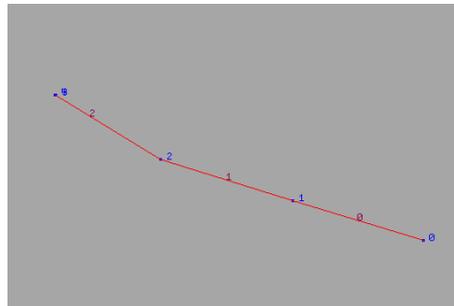


Fig32: Bridge path at start of the curve divided twice (5 points)

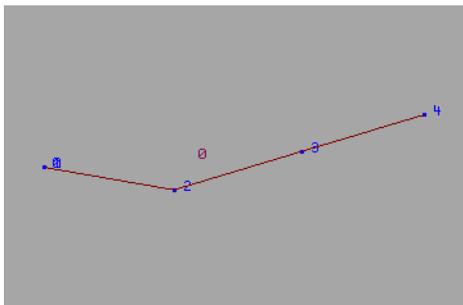


Fig33: Bridge path at end of the curve to be divided twice (5 points)

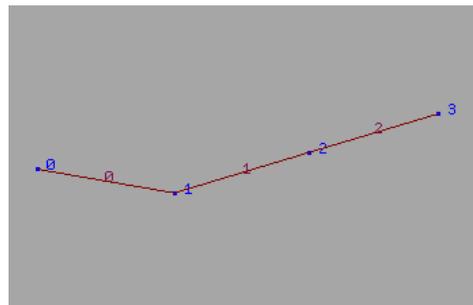


Fig34: Bridge path at end of the curve divided twice (5 or 4 points)

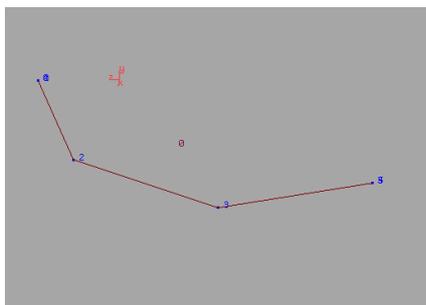


Fig35: Bridge path in between the curve to be divided twice (6 points)

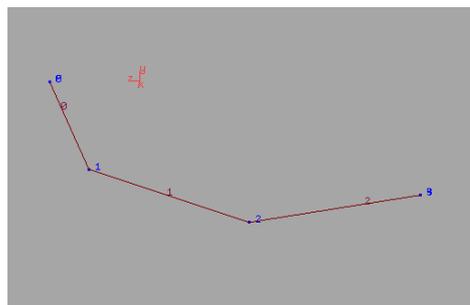


Fig36: Bridge path in between the curve to be divided twice (6 points)

Usage of Foreach SOP for joining the curve again:

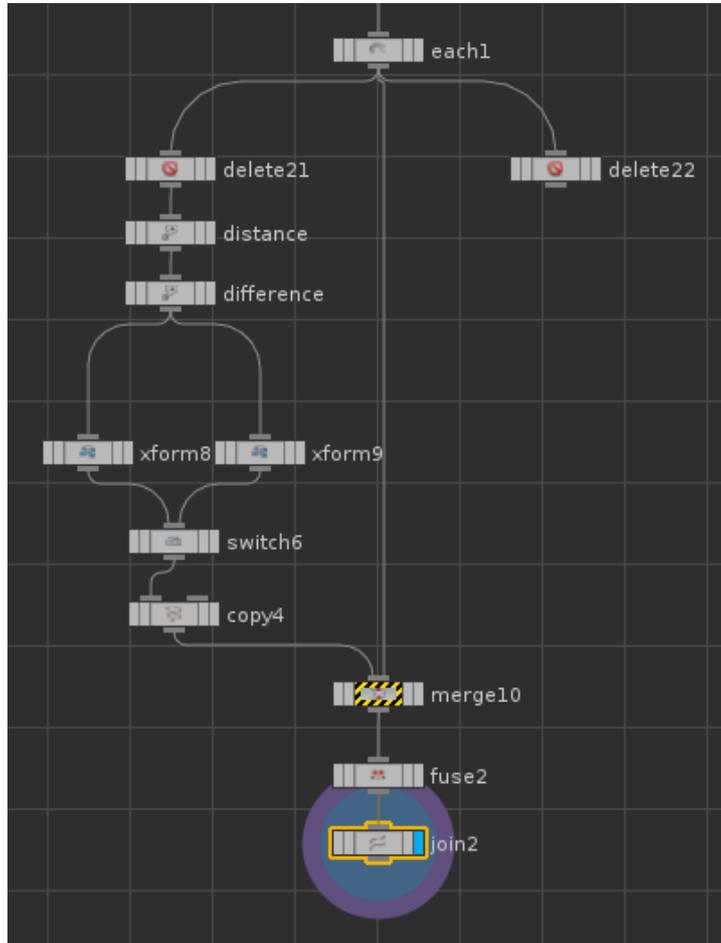


Fig37: Network showing usage of ForEach Sop for joining the curve.

In the Foreach sop, in the first iteration only first two primitives are used and the rest are deleted. Distance, difference in Y positions between the last point of first primitive and the first point of the second primitive are found. Based on the difference calculations, the primitive is moved up or down the distance calculated. Later on the moved primitive is merged with the original curve. This process should be done (number of primitives of the curve) times. The resultant curve would be as shown in the figure below.

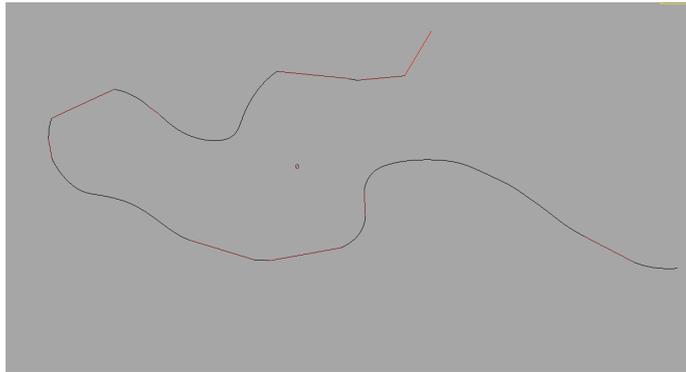


Fig38: Connected curve after calculations in Foreach Sop

As a next step the curve has to be partitioned in to individual parts for bridges and tunnels. This is done by using the colour information of every point of the curve. The colour value of every point is made such that red points have red value (\$CR) as 1 and black points have red value (\$CR) as 0. Then the points are translated in Y based on their colour information and then partitioned.

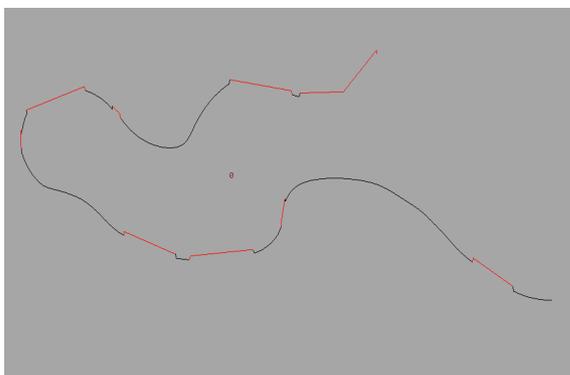


Fig39: Colour values transformed in Y axis

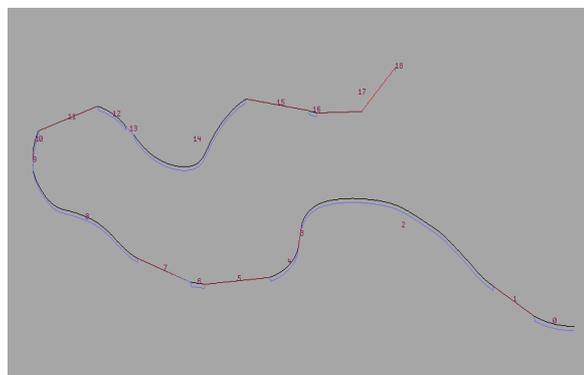


Fig40: Curve partitioned on colour info

4.3.7 Secondary Calculations:

After the bridge path calculations, every point's coordinates change. So calculations have to be done again for proper bridge and tunnel paths. The following figures depict the curve belong to tunnel area undergone secondary calculations. Some points have their Y position greater than the Y position of the terrain. Those points are given green value and used later on for levelling. The following figures depict secondary calculation on a part of input curve.

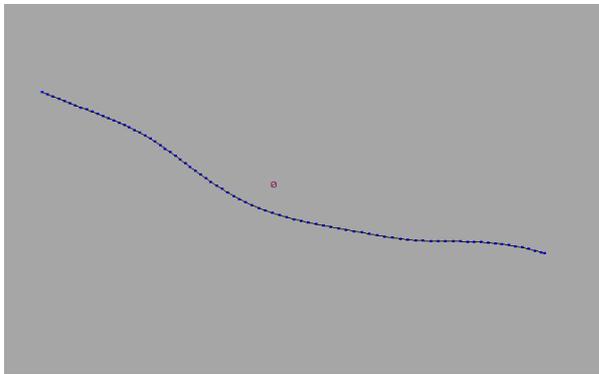


Fig41: Part of input curve for secondary calculations

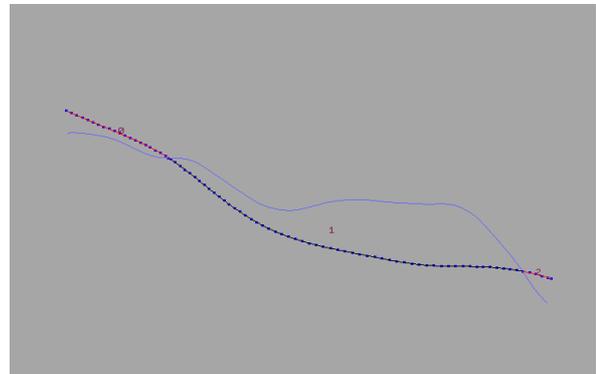


Fig42: Resultant after secondary calculations

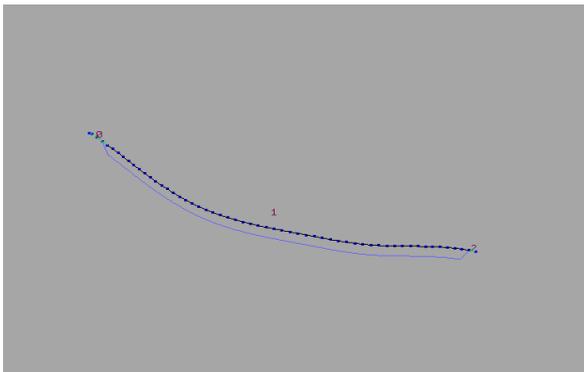


Fig43: Resultant after height calculations

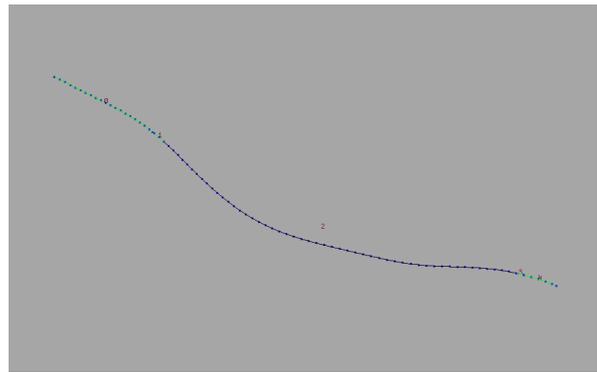


Fig44: The red points depicted in fig. above changed to green which will later be used for levelling

4.3.8 Height Calculations:

Distance between every point on the part of the input curve and that part of curve transformed on to terrain using Ray Sop is found. If the distance is less than Height Thres (Bridge or Tunnel) parameters, then green value is given to those points (depicted in fig:31). Those points later on will be used for levelling. This process is done to have a nice tunnel shape at the start and end of tunnel and also to make sure that the tunnel part won't collide with terrain from inside.

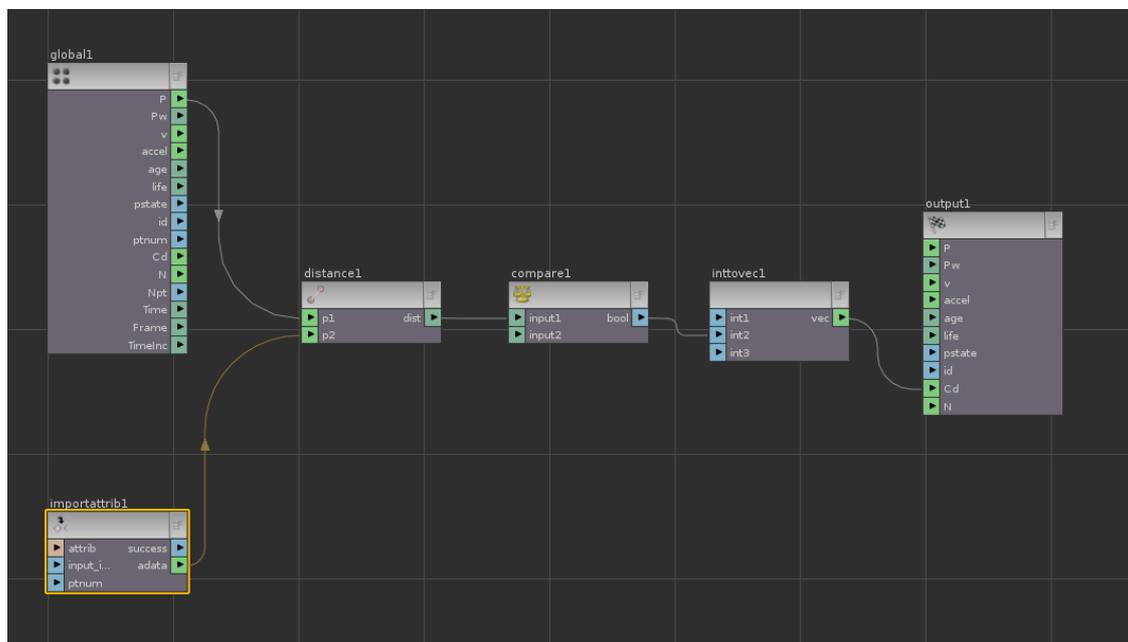


Fig45: Height calculations network

4.3.9 Small or Large Primitives:

Flexibility is provided to the user to select either small primitives or large primitives in certain modes of road generation (E.g. It won't be visually good to have bridges on very small primitives, so they can be disabled in bridge generation mode and can be used in level path mode for levelling.)

4.3.10 Second Offset Calculations:

A control is given such that the user can offset the individual parts of the curve from both the sides. Either inside part or outside part or both can be selected. For bridges and tunnels inside part has to be used and for levelling outside part should be used.

4.3.11 Bridge Support Calculations:

Bridge supports are the important things to be generated for a bridge. The following pictures show the process of generating the bridge supports. Bridge supports are generated between the actual path and the terrain path (points of the actual path transformed on to terrain). In order to achieve a good result some supports are deleted. 2 parameters Del. Supp. Lower and Del. Supp. Higher are responsible for deleting the supports.

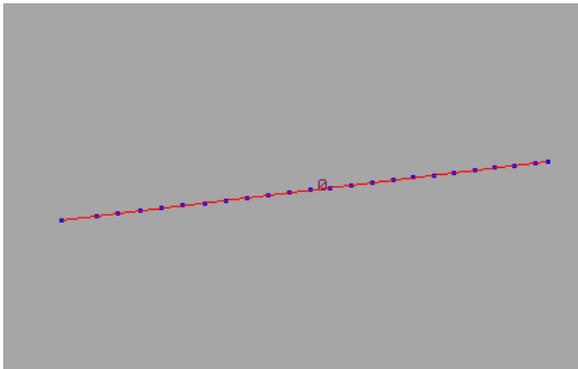


Fig46 :Part of bridge area

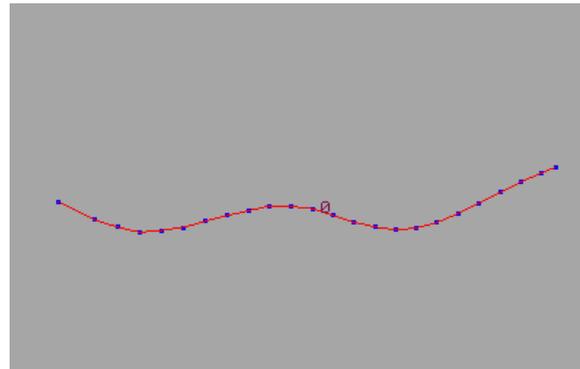


Fig47 :Points of bridge part transformed on to terrain

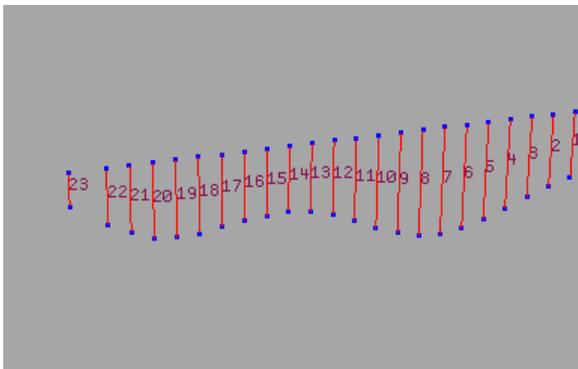


Fig48 :Supports generated between the above 2 curves

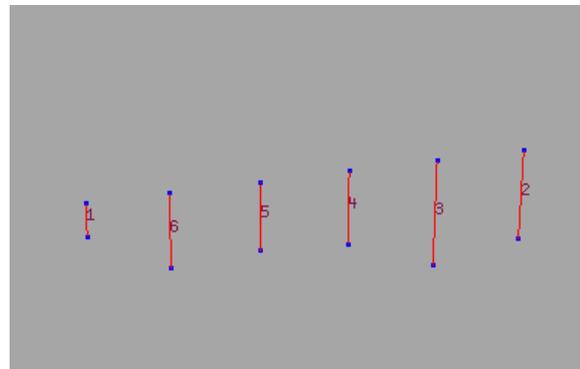


Fig49 :After deleting some supports

4.4 Road Asset:

Full path calculated using path calculations asset is fed in to Road asset in order to generate road on it. The primary step taken in the asset is to delete the closest points so that the road generated will have good detail on it .In order to do the calculations vopsop is used.

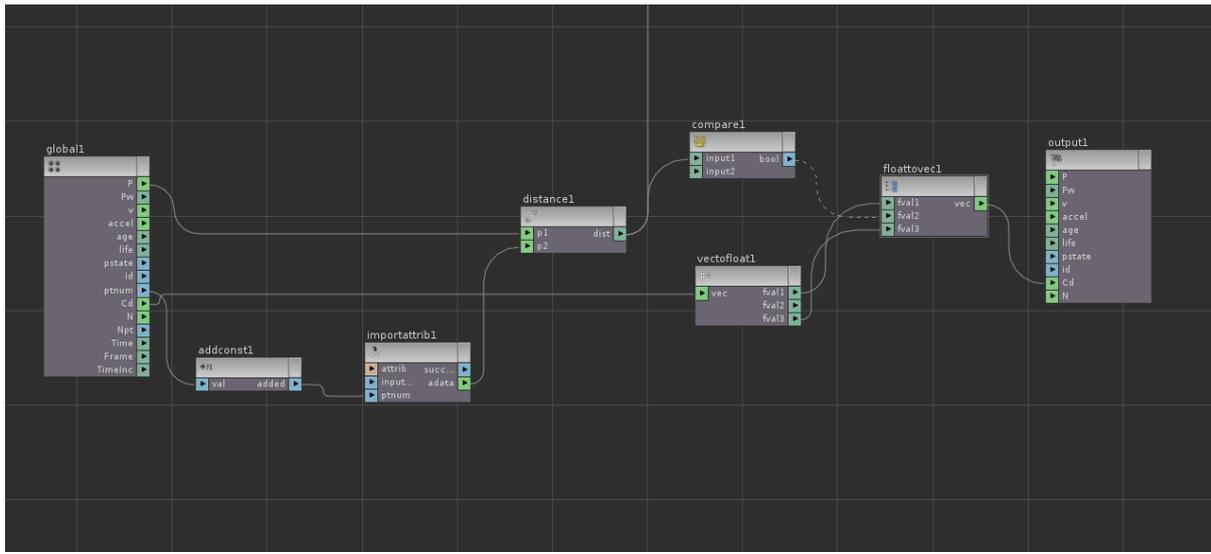


Fig50 :Vopsop calculations in order to delete the closer points

Distance between every point is calculated and if the calculated distance is smaller than some threshold, then the point is given colour information. Point that has the particular colour information is deleted.

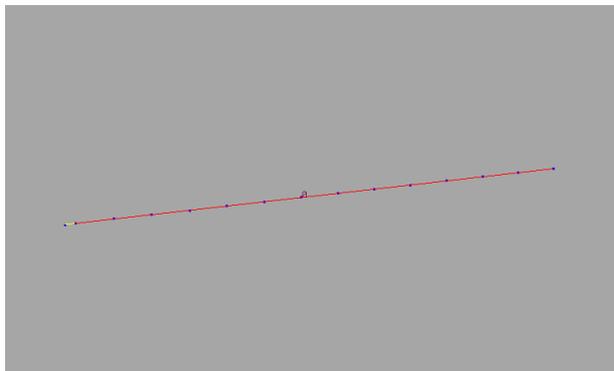


Fig51 :Closest point has yellow colour

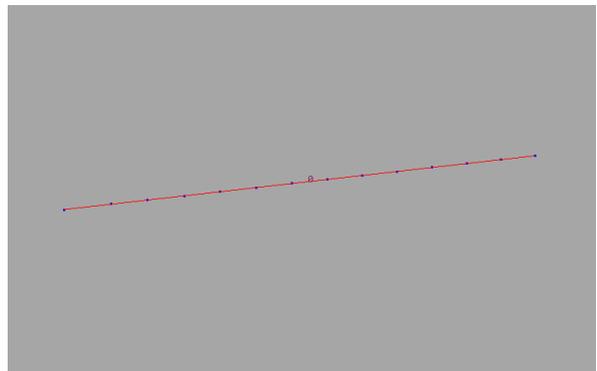


Fig52 :Deleted closer points

4.4.1 Level Sharp Corners:

As a next step, sharp corner points are deleted. This provides a smooth curve for the road to be generated on. The process is done by the use of Foreach Sop.

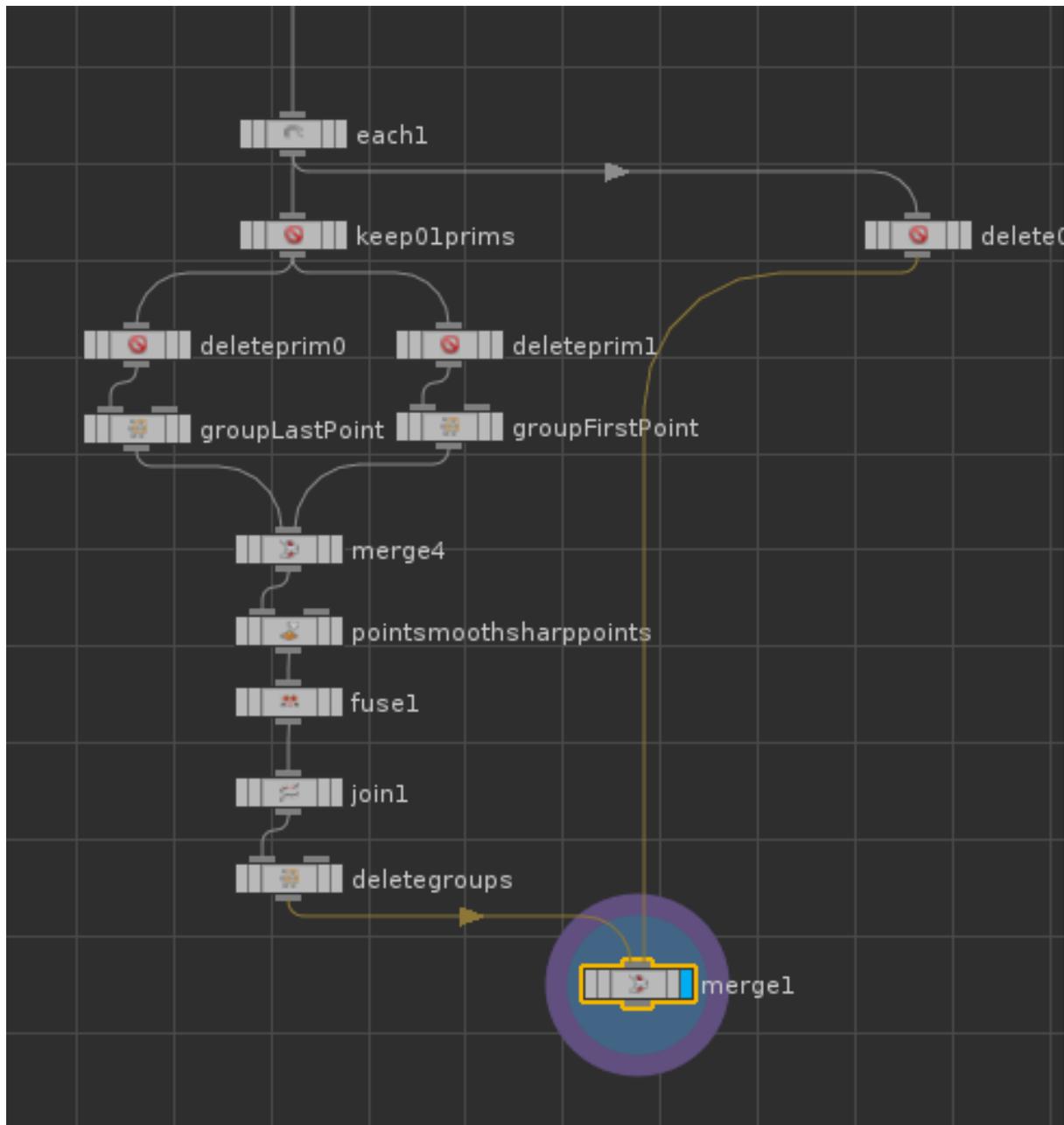


Fig53: Network showing the process of deleting the sharp corner points

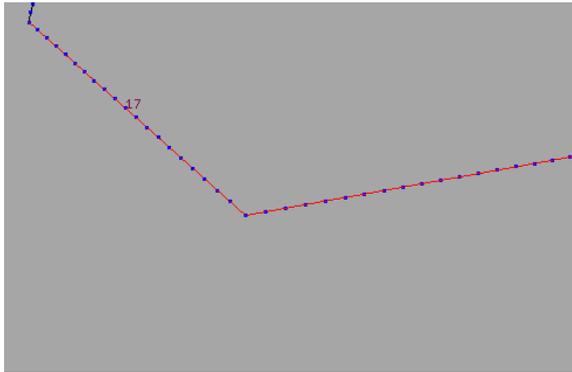


Fig54: Showing the sharp corner points

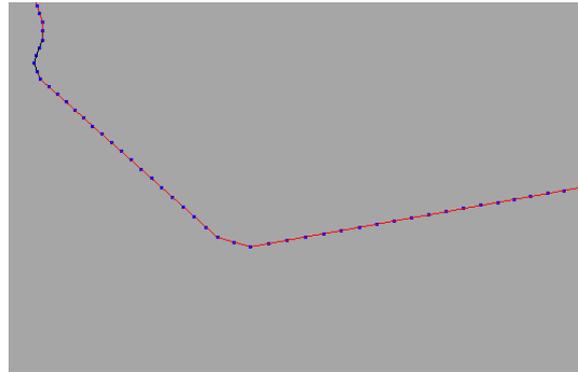


Fig55: Result after levelling the sharp points

In a Foreach Sop, in the first iteration only first 2 primitives are selected. Last point of the first primitive and First point of the second primitive are selected using a Group Sop. The primitives are then merged together and the selected points are moved towards the average position of last but one point of first primitive and second point of second primitive. It forms a smooth curve at the corners as shown in the right hand side figure. The two primitives are then joined in to a single primitive and then merged with the other primitives. The process should be done (number of primitives of curve) times.

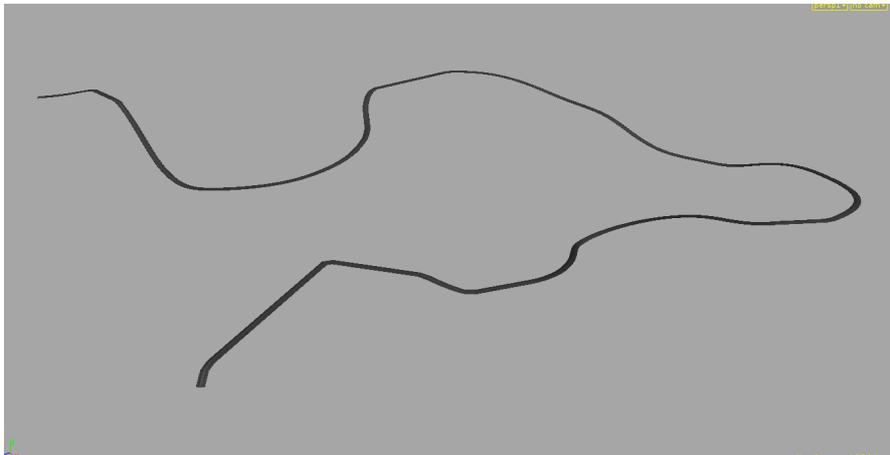


Fig56: Road generated after levelling the corner points

4.5 Detailed Bridge Asset:

This asset is responsible for generating bridges on the bridge path calculated using path calculations asset. It generates small or large bridges based on the perimeter of the input given. Bridge path along with supports should be given as input. The asset generates bridge on the actual path and supports on the bridge support path. It is depicted in the fig. below.

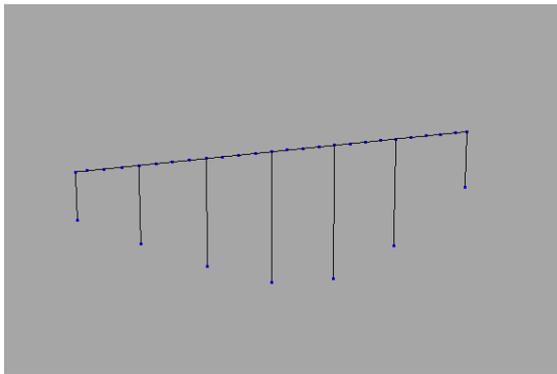


Fig57: Input given to the detailed bridge asset



Fig58: Bridge generated on the input provided

4.5.1 Cable Curvature Calculations:

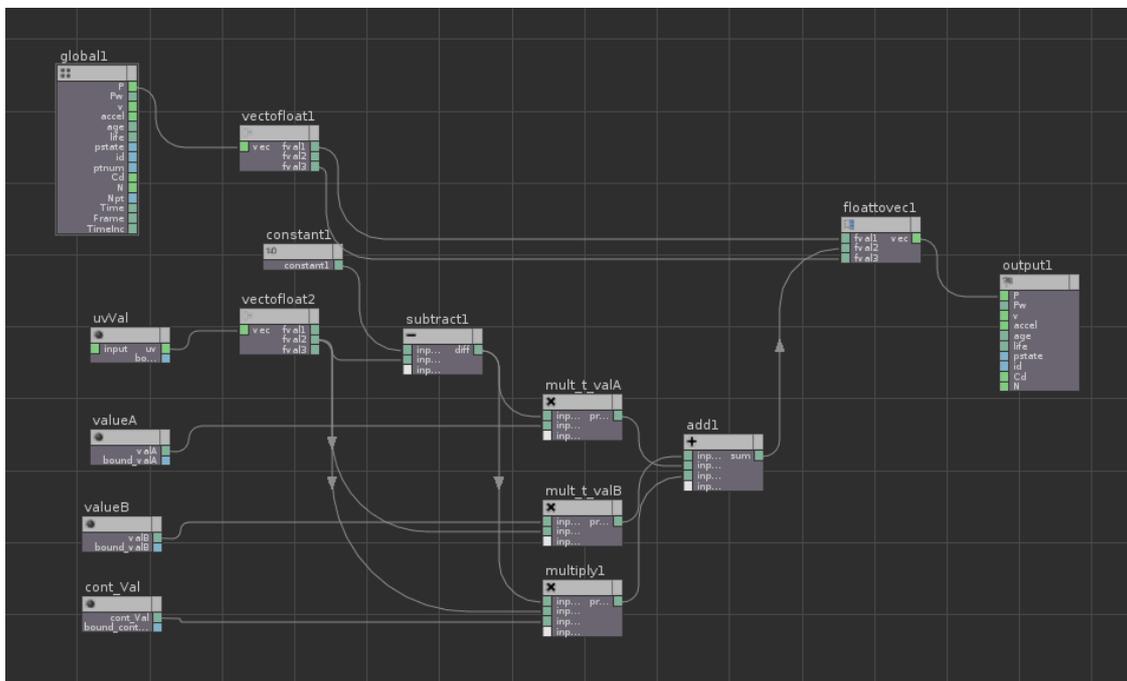


Fig59: Vopsop used for calculating the curvature of the cables

The cable curvature calculations are mainly based on a formula,

$$Va * t + Vb * (1-t) + Vc * t * (1-t)$$

Here Va, Vb and Vc are the control values given.

At first a line pointing in Y direction is taken as input. Then by using a vertex sop, the Y coordinate of a texture parameter is changed to the Y position of the input line. This information is fed in to the part of the input curve where bridge has to be proposed. All this information is fed in to vopsop .Inside vopsop 4 parameters are created in order to get the information of Va, Vb ,Vc and t (Y position of the texture coordinate). Then the calculations are done according to the above mentioned formula. The resultant is shown in the figures below.

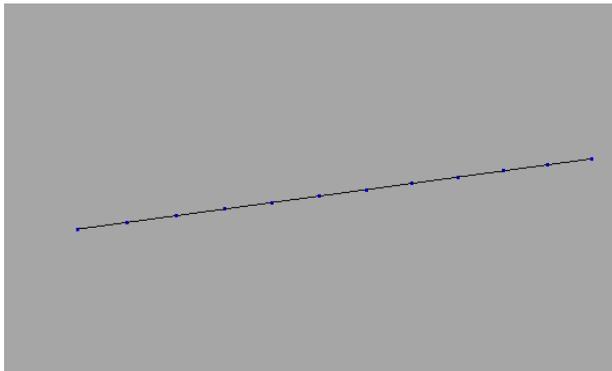


Fig60: Input given for curvature

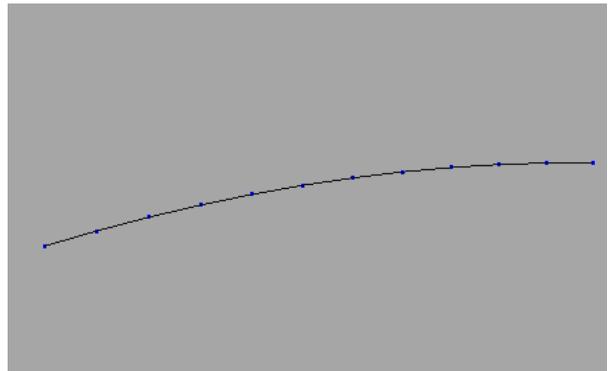


Fig61: Curvature obtained after the above calculations

4.5.2 Bridge Orientation Calculations:

After obtaining the cable curvature, the major problem faced is the orientation of bridges. The bridges have to be oriented according to the input curves given. The orientation of every input curve is different. The part of the input curve where a bridge should be proposed is used as a line passing through the centre of bridge and cables have to be placed on either sides of this line. Calculations have been made according to that. Clear depiction is given in the pictures below.

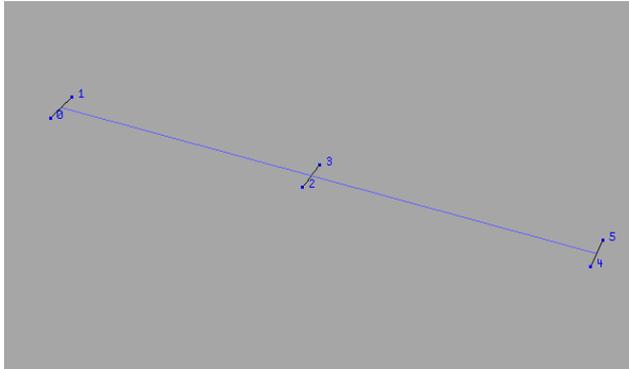


Fig62: Showing the input part where bridge has to be proposed

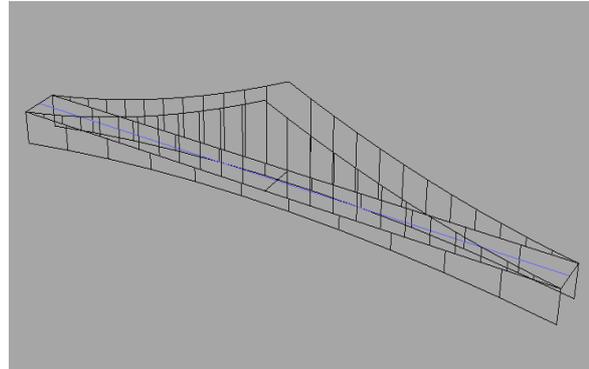


Fig63: Cables placed on either sides of the input curve

At first a line of unit length is copied on to the points of input curve (which has 3 points at an equal distance). The next step is to place the cables on both the sides of this input curve. In order to orient the sides of the bridge correctly in x, z directions, point positions of 2, 3 points on the left hand side figure are used. Cables are translated to the left by the use of x, z positions of point 2 by subtracting the respective centroids from the positions, to the right by the use of x, z positions of point 3 by subtracting the respective centroids from the positions. The resultant is shown in the right hand side figure. This process is done for every part of the input curve where a bridge has to be proposed.

4.6 Tunnel Asset:

This asset generates tunnel on the given path. At first profile of the tunnel shape is made and then it is swept and skinned on to the given input path using Procedural Texture Asset. This produces the tunnel along with texture on it. The process followed in the Procedural Texture asset is discussed later. The resultant is as shown in the figure below. The idea of this asset is adapted and improved from a video tutorial *Houdini Procedural road creation 3*.

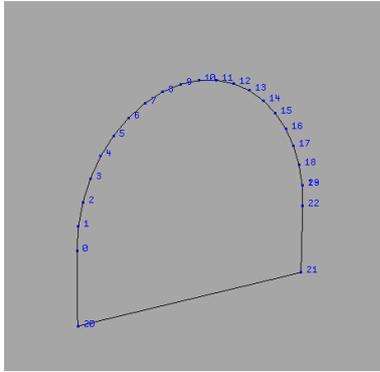


Fig64: Tunnel profile

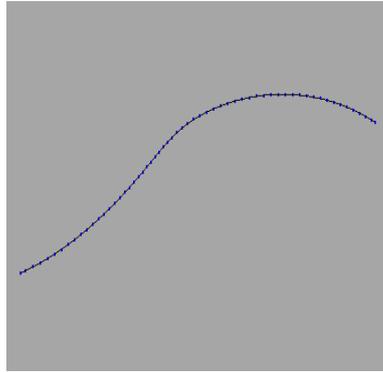


Fig65: Path of curve for tunnel

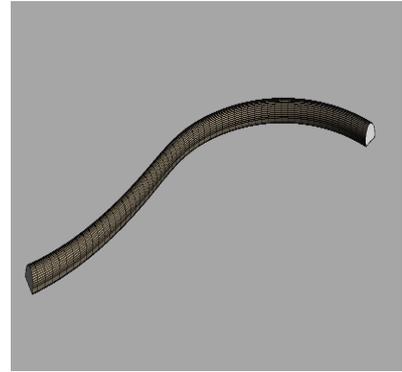


Fig66: Tunnel along with texture generated on the input curve.

4.7 Level Terrain Asset:

This asset plays a very important role in levelling the terrain at the beginning and end of tunnels and bridges thus by increasing the visual output. The process followed is adapted and improved from a video tutorial *Houdini Procedural Road Creation3*. This asset needs 3 inputs. Tunnel should be connected to the first input, terrain should be connected to the second input and calculated path for level area should be connected to the third input. Please middle click on the input ports for information about the connections.

4.7.1 Profile for levelling:

At first a profile is created for levelling. The process starts with the generation of a line oriented in X and its centre at origin. The distance between the first and last points of the line is the total span of the area which is to be levelled. The calculations are made in such a way that the centre points will have the maximum colour values when compared to the ends (start and end). It is clearly depicted in the pictures below.

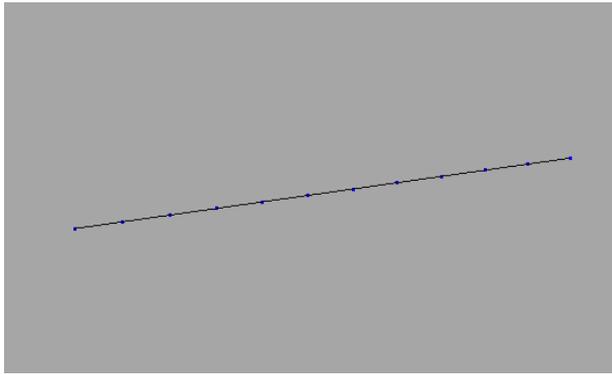


Fig67: Input profile

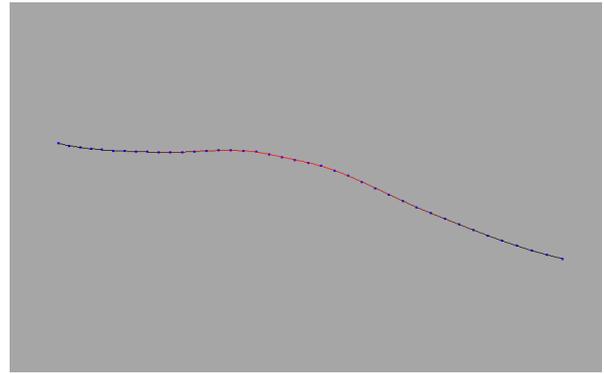


Fig68: Profile formed after doing some calculations

As shown in the right hand side picture, the red colour (\$CR) gradually increases, reaches to maximum at centre and then gradually decreases. The colour information of the points is added to their Y positions. This forms a very good profile which later on can be used for levelling. The profile is swept and skinned on to the parts of the input curve which needs to be levelled (calculated using Path Calculations asset in Level Area Mode) using procedural Texture asset.

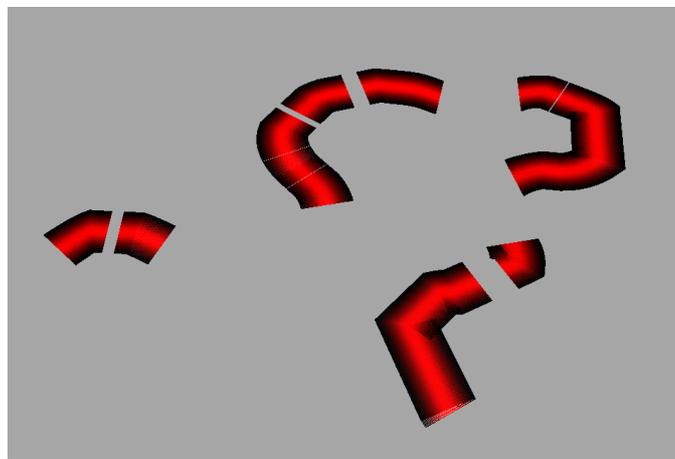


Fig69 :Resultant after sweeping the profile on to the parts of input path

The point position of every point is fed in to the green channel of the colour. The points with y position greater than zero will have green colour on them. The colour information is then transferred to the terrain using Ray Sop. Then the geometry which has the colour information on the

terrain is subdivided thrice. This forms a good area for the road to be merged on later. The attribute name of colour is changed in to weight and then used in Poly reduce Sop to reduce the polygons. The output is shown in the figure below.

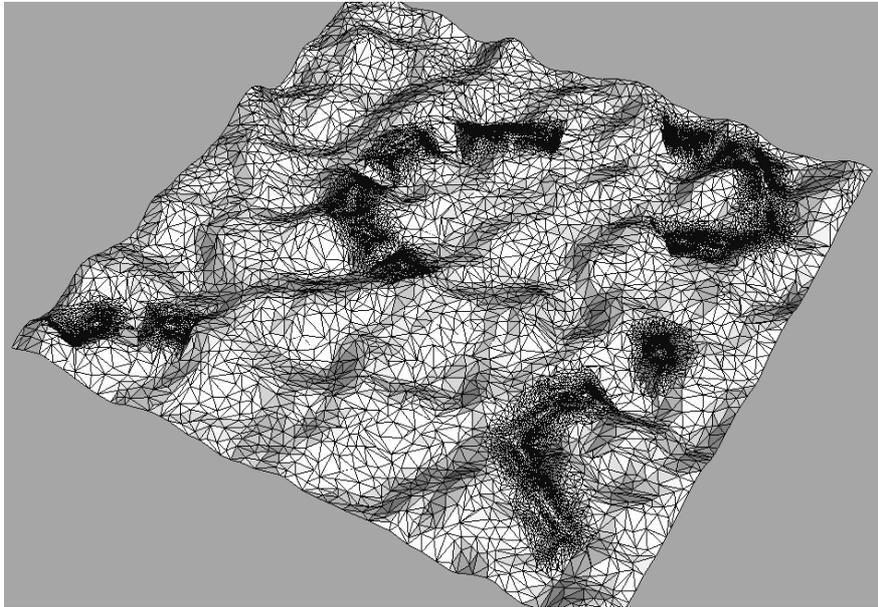


Fig70 :Output formed after reducing the polygons

4.8 Procedural Texture Asset:

This asset is used in road, detailed bridge, and tunnel and level Terrain assets in order to texture them. Profile should be connected to the first input and path should be connected to the second input. Materials are created inside a shop network exclusive for procedural texture asset in order to make it independent. Different materials are created for road, bridges, bridge supports, tunnels, towers, braces and cables.

The following pictures depict the inputs and output of the procedural texture asset.

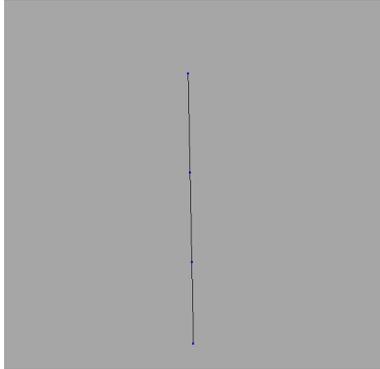


Fig71: Input path

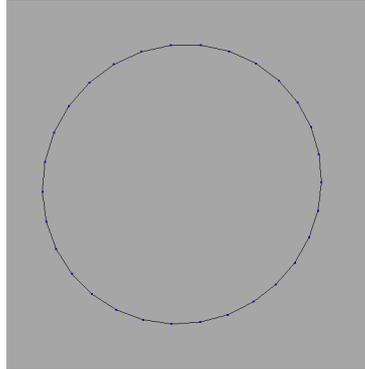


Fig72 :Input Profile



Fig73 :Output

Chapter 5

Conclusion:

This tool can be used in games for generation of different levels. Given any input curve the system generates bridges and tunnels accordingly. Most of the things are done automatically by the system and the user can change the parameters to get a desired look. All the calculations are encapsulated in to the digital asset and the user needs a bit of understanding how the tool works and mainly the network connections. A preset of the network connections is depicted in User guide. The user has to follow the connections in order to achieve the complete set up. Description on how to draw a curve is also provided in the user guide. At present the tools generates the procedural geometry and textures but in future some more controls will be added (such as lighting and shading) which will be helpful in generating the whole scene with just an input curve.

Renders :



Bridge View



Tunnel and level Area View

References :

- 1.Kelly, G. and McCabe, H., 2006. A survey of procedural techniques for city generation. In ITB Journal, Issue 14. Available from: <http://www.gamesitb.com/SurveyProcedural.pdf> [Accessed 10 August 2010].
- 2.U.S.Department of transportation Federal Highway Administration, 2010. *Technical Manual for design and construction of road tunnels-Civil Elements*. U.S.A. Available from:<http://www.fhwa.dot.gov/bridge/tunnel/pubs/nhi09010/index.cfm> [Accessed 20th July]
- 3.2010. *Houdini Documentation*. Available from: www.sidefx.com

Video Tutorials:

4. *Getting started with digital assets*, video tutorial .Available from www.digitaltutors.com [Accessed 10 June 2010].
5. *H9 Bridge Asset*, video tutorial .Available from: www.3dbuzz.com [Accessed 10 June 2010].
6. *H9 Elevator Asset*, video tutorial. Available from: www.3dbuzz.com [Accessed 10 June 2010].
7. *Houdini Procedural Road Creation1*, video tutorial .Available from: www.cmivfx.com [Accessed 20 June 2010].
8. *Houdini Procedural Road Creation2*, video tutorial .Available from: www.cmivfx.com [Accessed 10 July 2010].
9. *Houdini Procedural Road Creation3*, video tutorial .Available from: www.cmivfx.com [Accessed 1 August 2010].
- 10.Side Effects Tutorials, video tutorial. Available from: www.sidefx.com [Accessed 10 August 2010].

Images :

- Fig1: Edinburgh, Fife and East Scotland, 2009. *Forth Bridges* [photograph].Edinburgh. Available from: http://www.bbc.co.uk/edinburghandeastscotland/content/image_galleries/forth_bridges_gallery.shtml?6 [Accessed 10th August]

Fig2: Anonymous. 2010. Available from <http://www.mesalek.com/colo/picts/col19bouldercyn-tnl.JPG>.
[Accessed 10 August 2010]

Fig3: Kelly, G. and McCabe, H., 2006. A survey of procedural techniques for city generation. In ITB Journal, Issue 14. Available from: <http://www.gamesitb.com/SurveyProcedural.pdf> [Accessed 10 August 2010].

Fig4: Kelly, G. and McCabe, H., 2006. A survey of procedural techniques for city generation. In ITB Journal, Issue 14. Available from: <http://www.gamesitb.com/SurveyProcedural.pdf> [Accessed 10 August 2010].

Fig5: U.S.Department of transportation Federal Highway Administration, 2010. *Technical Manual for design and construction of road tunnels-Civil Elements*. U.S.A. Available from:<http://www.fhwa.dot.gov/bridge/tunnel/pubs/nhi09010/index.cfm> [Accessed 20th July]

Textures:

Available from: www.cgtextures.com [Accessed 25th July].

Forums:

Odforce Forums, 2010. Available from: www.odforce.net .
[Accessed 18th August]

Side Effects Forums, 2010. Available from: www.sidefx.com .
[Accessed 18th August]