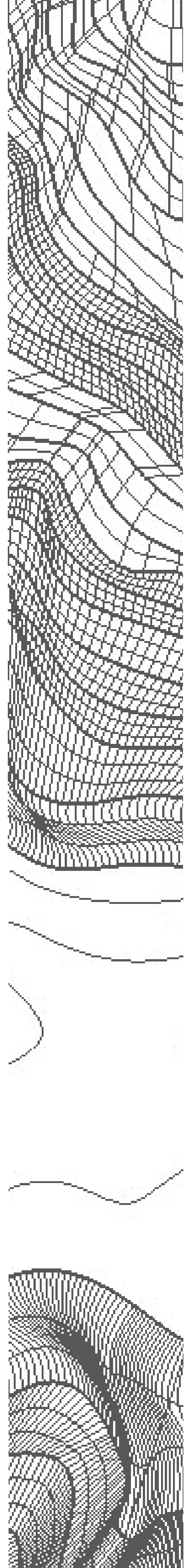


LAR 651 GRAPHICS WORKSHOP

University of Virginia
School of Architecture
Fall 2008
R Dripps

INTRODUCTION

This book is intended to support ALAR 651, Graphics Workshop, offered in the Fall of 2008 at University of Virginia's School of Architecture as a study in the relationship between ideas and their representation. The course operates as a workshop exploring sketch diagramming, rapid physical modeling in paper, and 3D computer modeling in Rhino. These modes of thought and action are used interchangeably to build more complex accounts of human settlement.

Navigating

These pages are filled with tips from students who use Rhino for architecture and landscape architecture. The first section contains Topic sheets, which serve as more technical guides for useful Rhino commands. When referring to these, you should look beyond the immediate actions they produce to think about the larger ideas that they might sponsor. The second section is made of Examples, which take the reader through applying ideas and commands relevant to the assignments in the course. Topics and Examples reference each other and should be used simultaneously.

The best and most exciting use of this primer would be for you to test out the content, fearlessly explore alternate possibilities, and then add your own discoveries to a growing body of knowledge. Keep this for your later reference!

Words in **bold** are typed **Commands**, words in *italics* are menu paths, and words in [BRACKETS] are references to other Topic and Example pages.

About the Authors

Robin Dripps drew the dots and lines, while **Lucia Phinney** filled them in with ground and picked late blueberries. **Gennifer Munoz** busied herself with petting the cats, **Mary Becica** arrived just to dig in the garden, and **Dave Malda** added his rational from a mountaintop.

TABLE OF CONTENTS

Examples

NAVIGATION | Making Rhino appear and making things appear in Rhino

Rhino is a modeling software ready for exploration. It began as a tool for boat builders and is now an excellent interface for architects and landscape architects because of its flexibility between other modeling software and mediums. It is friendly enough to be a sketch tool and precise enough to produce documents for CNC fabrication. All of its user settings are customizable and easy to learn once you start experimenting. Don't be afraid to get lost and then found by the Rhino Help Menu and Glossary, or just ask the local experts in school for a how-to.

Open and Enter Rhinoceros

Rhino at UVA

In the architecture school, Rhino can be found under *Start Menu>All Programs>Architecture and 3D Modeling>Rhinoceros 4.0>Rhinoceros 4.0*. Rhino versions on school computers will prompt for a license number before the software opens, which only requires pressing okay. When Rhino is being heavily used on school computers (typically in the hours before studio and especially leading up to charette), UVA will run out of available licenses, so it is always safe to have a Rhino-ready personal computer. Running Rhino on UVA's Mac computers using Parallels Desktop is not recommended, since it is usually too slow to be efficient.

Note that user preferences within Rhino will not be saved on UVA's computers but will save on your personal laptop.

Rhino on your computer

When geometries get complex or renders get large, Rhino consumes a lot of memory on your computer. Look for pointers throughout these Topic pages for tips on saving memory. If your computer is still too slow to handle the software, consider buying more RAM for your computer (an affordable fix).

If you need further assistance setting up your laptop, licensing issues, or your user preferences on UVA computers, take advantage of UVa's Computing Staff located at Room 304 in Campbell Hall or at http://www.arch.virginia.edu/computing.

Entering the 3D modeling world : Coordinate Systems

3D objects in modeling applications are described by points. These points are the basis of lines, spheres, boxes, or any other object you might model.

In order to create or modify these points and the related objects, modeling applications use a coordinate system. *Rhino Help>Coordinate Entry*

The default coordinate system for most applications is Cartesian Coordinates. This describes the model "world" with three axes: x, y, and z. Within this system every point is assigned a value along each axis. Values are described relative to the coordinate system's *origin* where the assigned value is (0,0,0). Values can be positive or negative.
http://en.wikipedia.org/wiki/Cartesian_coordinate_system

A model can have multiple coordinate systems. Sometimes it is useful to set up an auxiliary coordinate system with axes aligned to another system. Auxiliary coordinate systems do not overwrite the world coordinate system, they are defined relative to it. An example of this might be in modeling two portions of a city with different grid orientations. Switching between coordinate systems allows you to use drawing aids like **ortho** that lock to the coordinate system's axes.

coordinate system 01 coordinate system 02

Viewports
In Rhino, you view your model through four viewports. By default they are Top, Front, Right, and Perspective. At the bottom left of each viewport you will see World Axis Icon orienting you to the base coordinate system. The Top, Front, and Right viewports have two perpendicular axes; they are orthogonal to the base coordinate system. Perspective has three axes and imitates a perspective view.

Rhino and other 3D modeling programs are based on a camera system; all views are captured in the viewports by cameras. When you navigate a viewport, you literally are moving, panning, or zooming with a hidden camera. If you make the camera visible that is capturing the perspective image (**Camera**), for example, the camera will appear in the Top, Front, and Right viewports and can be manipulated from there. Manipulating cameras and views will come in handy later [EXPORTING], but their controls can be unfriendly to the beginner, so if you get lost keep in mind that you can always return to default views. See *Rhino Help>Camera* for more information.

Construction Planes
The construction plane is the active 2D plane on which new points are drawn. By default the construction plane is based on the X and Y axes and passes through Z coordinate 0. Changing the construction plane offers some of the same potential as changing the coordinate system. It is particularly useful for drawing in the perspective view.

Drawing in two-dimensional views (front, side, top, etc) automatically relates to a different construction plane that corresponds with the axes of the view (i.e. top = x,y).

The command **CPlane** brings up a variety of options for changing the construction plane. *Rhino Help>CPlane.*

Notice the option to modify the cplane's elevation. This is particularly useful when drawing floor plans at multiple levels or some distance above the origin.

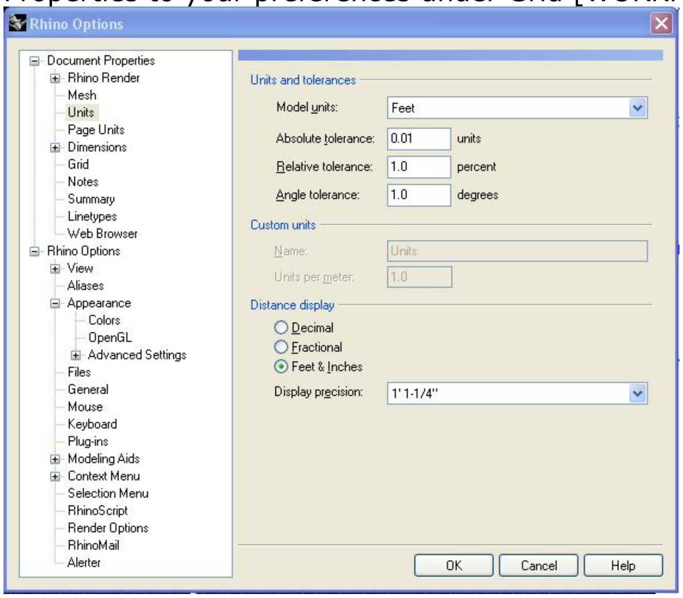
Units and Options
3D modeling gives you the power to work at any scale or unit, so it is imperative that you set up your units and precision before you start working. The units assigned to the model (i.e. feet) are stored within the model. A model that was drawn with values in feet will change scale if it is opened as a model with values in meters [EXPORTING].

Manipulate Rhino Options under *File>Properties>Document Properties* or Command Prompt:

Options. Explore and customize!

Set the units, distance display and display precision under Units. Keep in mind the difference between the Units of the model and the Dimensions of the model; Dimensions is just a display option, and does not toggle the actual built units. Page Units refers to Printing options.

Having a visible grid can be helpful to remind you of your scale while you are modeling, so set Grid Properties to your preferences under Grid [WORKPLACE].



Zoom and Navigate
3D modeling demands simultaneous attention to four viewports, the command line, toolbars, the keyboard, drop-down menus, etc. It is very active! Drawing in the Perspective view often leads to disorientation if you do not actively pay attention to where you are in the coordinate system, so watch the parallel viewports while you are drawing in the Perspective view. Set up your views before you start drawing to maximize your productivity.

If you get lost in your viewports, it is easy to get back to your model or the defaut view. Some handy tools include:

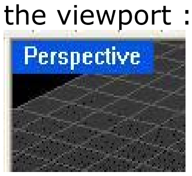
UndoView or  returns to the previous view.

Zoom and then typing **Extents** or  zooms to the extent of the objects. In other words you see everything you have modeled.

Often the camera will slow down or not continue to zoom in. This is either because the camera has approached the picture plane, or that your model is too complex and the viewports are refreshing slowly. To fix this, redefine the picture plane using:

Zoom or  prompts for a zoom window. Click and drag to redefine the picture plane.

To return to the default viewport view, right click the blue name button on the upper left corner of the viewport :



Then click *Set View* and then select the view you would like to return to, or select a new view. For more, see [WORKPLACE, EXPORTING] or *Rhino Help>Viewport>Navigation*.

Shortcuts

See Options>Rhino Options>keyboard for additional information or to edit shortcut keys
See the "Select command" page on the help menu index.

Mouse Works: Clicking and Scrolling
When not active in a command:

Key	In parallel viewports (top, front, right)	In perspective viewports
right click + drag	pan	rotate
shift, right click + drag	--	pan
ctrl, shift, right click + drag	rotate	rotate
ctrl + right click + drag	smooth zoom	smooth zoom
scroll	zoom	zoom
right click	repeat	repeat

When active in a command:

Key	Command
shift	Toggles Ortho on or off
right click	enter
tab	direction lock
ctrl	elevator mode (in perspective) *see Rhino Help search : elevator mode*

Selecting
When not active in a command :
Selection box : left click + drag allows selection of multiple objects. Dragging the box from left to right creates a solid outline and only objects entirely encompassed in the box will be selected. Dragging the box from right to left creates a dotted outline, and all objects that overlap into the box will be selected.

When active in a command :
shift + left click adds objects to selected group
ctrl + left click removes objects to selected group

Useful selection commands :
SelLast : Selects the last changed objects
SelAll : Selects all objects
Explore other selection commands specific to your model, such as **SelPt**, **SelPolyline**, etc.

Other Shortcuts
When using Rhino on a MAC, some keyboard shortcuts differ from a PC. Shortcuts can be edited or created on your personal computer under *tools/options/rhino options/keyboard*

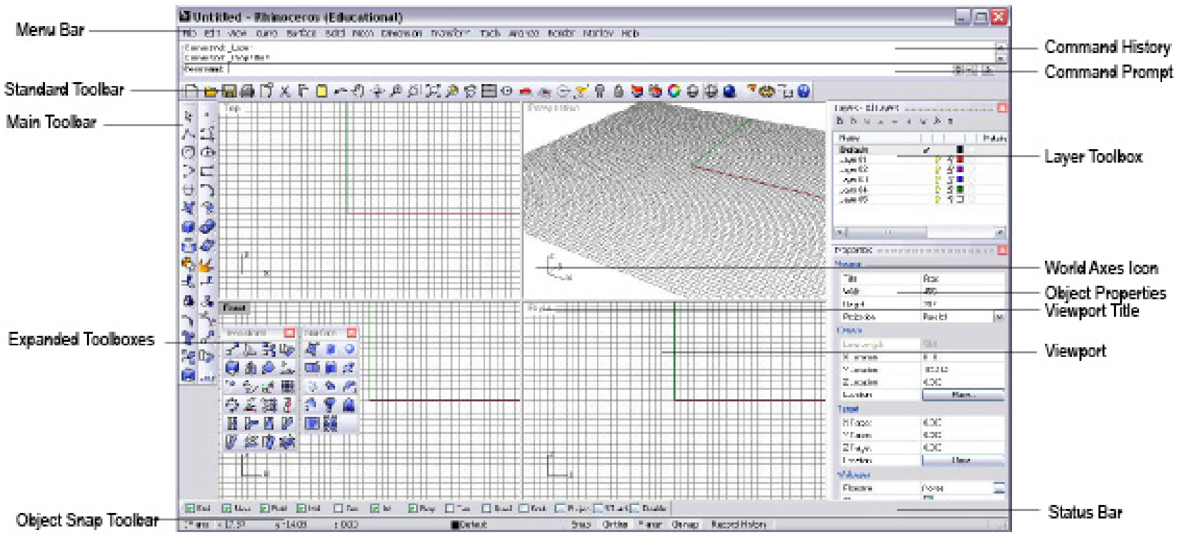
Key	Command
ctrl + s	save
ctrl + d	delete
ctrl + z	undo
ctrl + j	join

WORKPLACE | Setting up the work space

It is important to set up a workplace in Rhino that you are comfortable with and one that allows ease of use with fluid motion from operation to operation. The workplace is customizable, and can be set up for personal use. Object properties and viewport properties can be altered to display your digital model in a way that can be conducive to exciting modeling ideas and inventions. Once you feel at home in Rhino, its use as a 3D design tool can be evocative and rewarding.

Navigating the Workplace

There are areas of the Rhino window that give and take information, and there are areas that display tool buttons that are linked to commands. There are multiple ways of navigating through Rhino that achieve the same results including the Menu Bar, Toolbars, and the Command Prompt.



Your personal workplace may be different, and this is an example of a workplace that is useful to one student. The workplace must be set up each time you sit down at a computer because Rhino retains the settings of the previous session. If you are on your own laptop the settings will remain intact for the next session of Rhino.

<u>Saving</u>
Saving is important to your progression through a project, and it will give you easy access to your thought process, diagrams of your ideas, and the ability to create multiple iterations at moments of clarity.

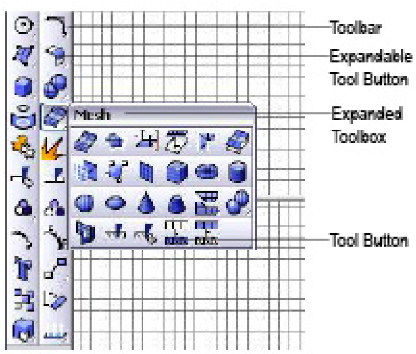
Iterations of a .3dm file can be saved and organized by a simple system that will help you retreive it later. Save your files as something meaningful. Think ahead to a structured naming system that can result in a sequence of iterations that are easy to reference later. (PaintField_1, PaintField_2...)
(Mondrian_091308, Mondrian_092208)

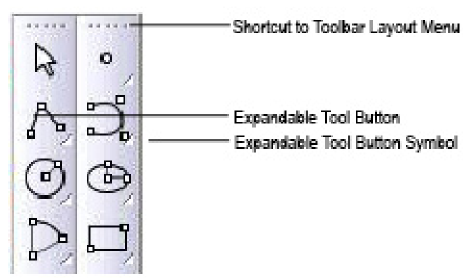
In the Rhino **Options** menu, under *Files* you can set up an autosave system to avoid losing information.

Remember, SAVE OFTEN. SAVE USEFUL IMAGES. SAVE ITERATIONS THAT YOU WANT TO KEEP.

Menu Bar
The Menu Bar is one of a few of the modes of navigation within Rhino. From here you can access commands, options, and the help menu. When a choice is made from the drop down menus, the command is displayed in the Command History box as **Command**. For future reference, this can be typed without the underscore to achieve the same action.

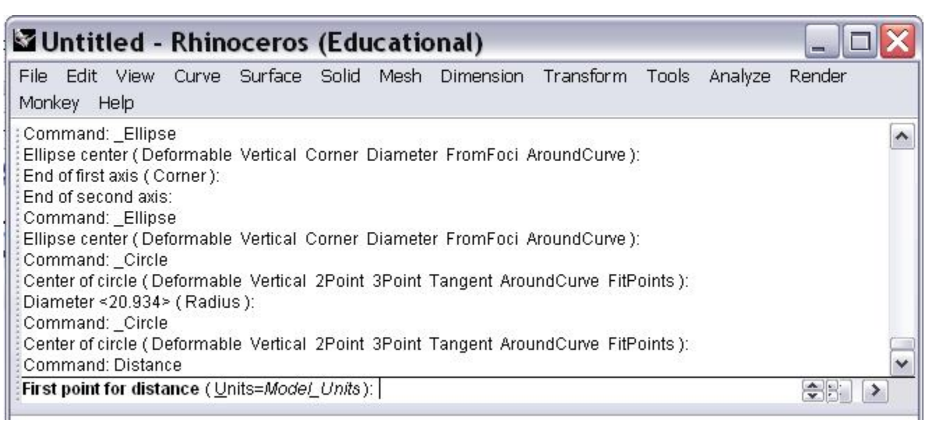
Toolbars
Toolbars are customizable, and they can be adjusted, added, or removed to increase the fluidity and productivity of your digital modeling endeavors. Toolbars can be moved by clicking and dragging on the dotted line at the end of the bar. The Toolbar Layout Menu can be found in the Menu Bar under *Tools>Toolbar Layout.* It can also be found by right-clicking the dotted line at the top of the main toolbar. Additional information about opening, positioning, saving workspaces, and toolbar options can be found in the help menu's index.
Help>Help Topics>Index>Workspace.





Tool Boxes
Toolboxes can be found by holding down icons with triangles in the bottom right corner and they can be dragged into the workspace by their blue title bars. In order to keep the box, it must be moved away from its initial location, otherwise it will disappear when it is no longer active.

Command Bar
The Command Bar is vital to your awareness of what is going on in Rhino. It is also one of the methods of choosing operations in which you type commands. The Command Prompt is where you type commands, and the Command History is a display of the current active command and past commands used. Command history can be expanded to easily trace what you have been up to. It is important that the Command Bar is wide enough to display this vital information. The bar can be adjusted by grabbing its bottom edge and pulling down.

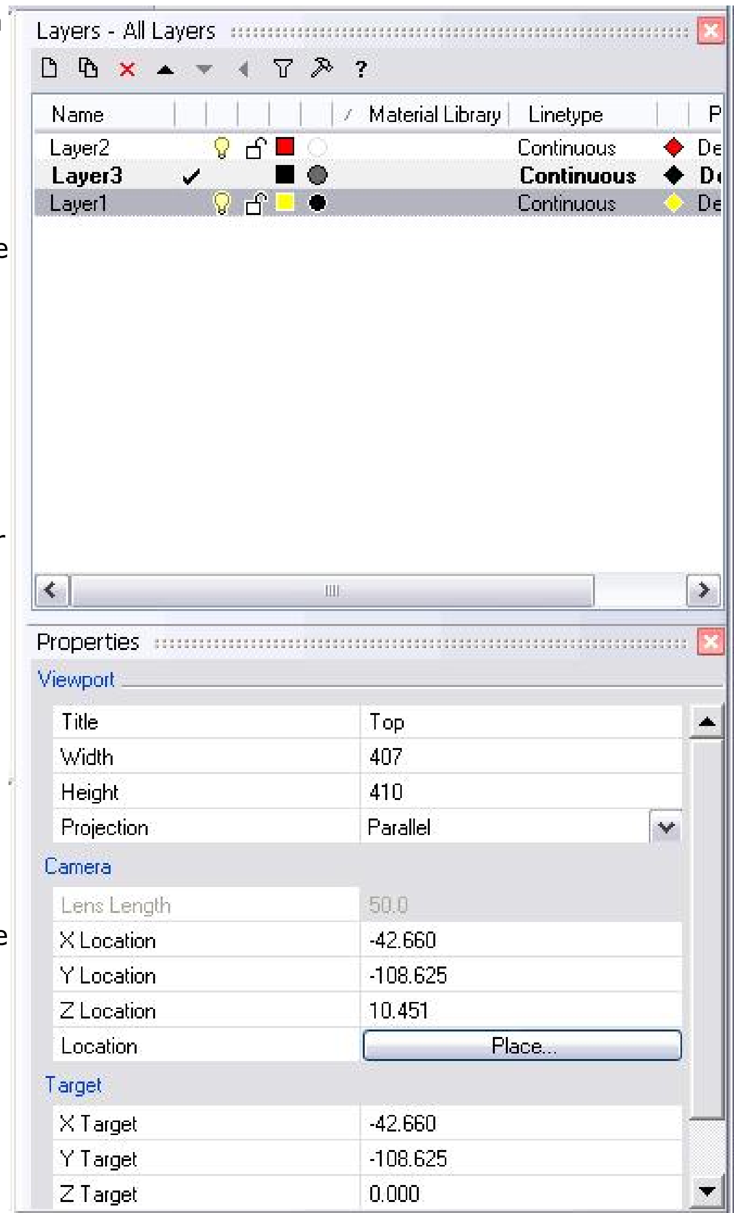


***Distance** is the active command.*

If you need a tool, but cannot find it in the toolbar, or you would like to see other commands that branch from that tool, begin to type the keyword, (Extru___, Arr____, Sel___) and see the drop down menu for choices of various tools you may be looking for. As far as locating commands within Rhino, the various methods of doing so (menu bar, toolbars, command prompt, viewport title, and other various short cuts) leave the style of working up to you. It is important to be familiar with Rhino and it's abilities, but as far as speed of work, this is left up to your preference as far as clicking or typing, or both. A combination of both typing and clicking is ideal, so again, one hand on the mouse, one hand on the keyboard!

<u>Layering</u>

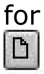
The Layer Toolbox is important to maintaining an organized 3D model in Rhino and layers are important to organizing your ideas. Having the box visible while you work will allow you to navigate between layers as you build.



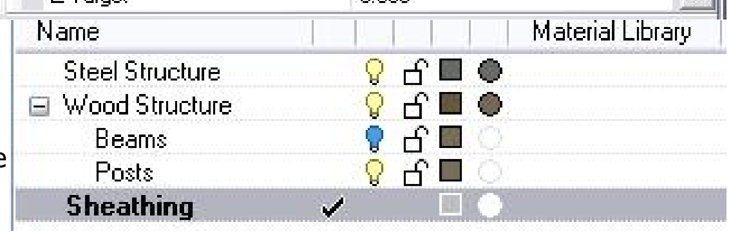
Layers can be imagined as transparent overlays with each layer carrying a particular aspect of the model. Layers work well when the information on any layer is a coherent whole. The power of layers is their ability to organize your work and to turn on a specific layer, to hide a specific layer, to select a specific layer, or to look at any number of specific layers at one time. Layers can also be a useful way to test out options and set up hierarchy within the structure of a model. Since you are likely to have a substantial number of layers, it is important to have an effective labeling strategy and related layer colors.

<u>Layer Controls</u>

 **Layer** opens the Layers Toolbox. This is good to have on screen and docked at all times for organization and awareness.

 Creates a new Layer.

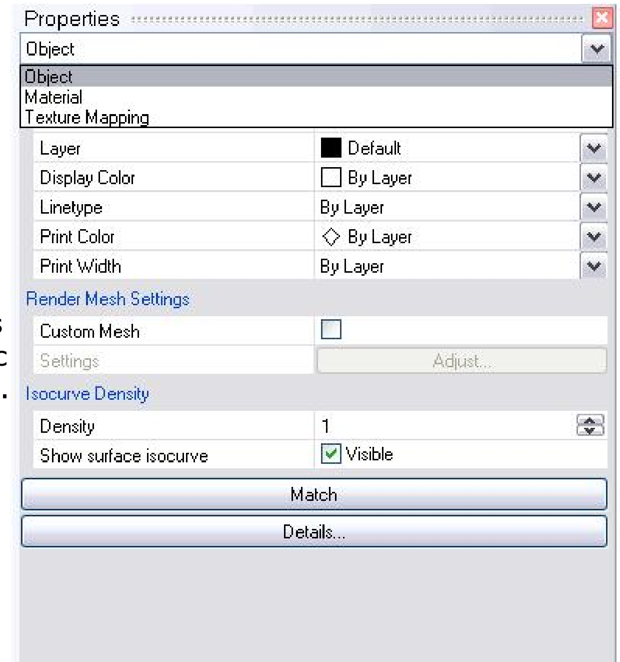
 Creates a new Sublayer. To change the name of a layer, double click the name. To change the properties of multiple layers (color, on/off, or locked) highlight a list using shift and then make changes. You cannot make these changes if a command is active, but you can create a new Layer or Sublayer while in a command.



Notice what the active layer is when using a command; objects will be drawn in the layer that is active. The active layer will be indicated in the layer properties box (with a bold name and check mark) and appears at all times at the bottom of the screen. To make a layer active, click the white space to the left of the light bulb, which toggles the layer on. Only one layer can be active at once.

Object Properties

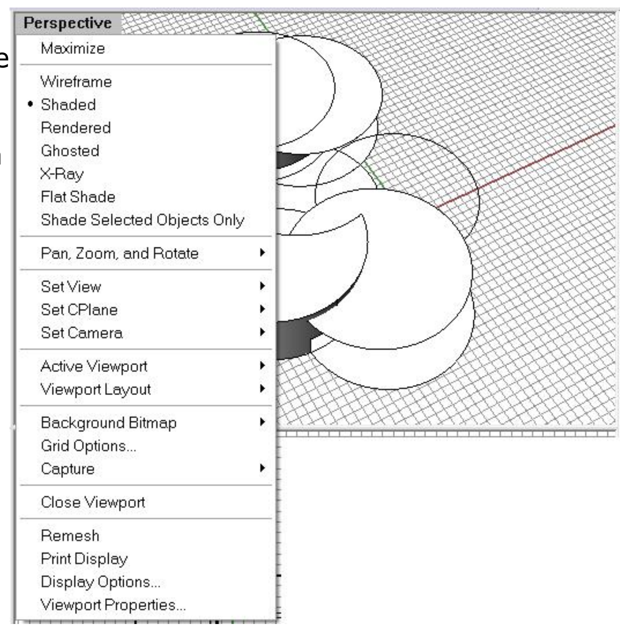
 **Properties** opens the Properties box where you can edit the object, material, and texture mapping properties of one or multiple objects at a time.



If your viewport is set to render mode, the object properties will affect the objects in the viewport. This will allow you to format the views while you are working. [EXPORTING] will describe in further detail the operations behind setting up an aesthetically pleasing and a strategic viewport window and how to capture views of your model.

Viewports

The viewport windows in Rhino allow you to display different views of your model at once. This is crucial to operating in a 3-D modeling program. Not only will it give you the chance to design in plan, perspective, elevation, and section simultaneously, but it will also expand your spatial understanding and the implications of an action in plan on the model's section.



The viewport layout can be arranged in various ways, as a grid of four windows - plan, perspective, front, left views - or as one window that fills the Rhino workspace. It can be changed to meet the needs of your modeling endeavors.

A shortcut way to get to some of the viewport layout options and modes is by right clicking the viewport title box. Here you will find some shortcuts to things like rendering mode, view (where you can save views), cameras, and bitmaps [EXPORTING].

Standard viewport options found in the viewport title menu:

- Wireframe (**wireframeViewport**) : displays isocurves and wireframes

- Shaded (**shadedViewport**) : displays solids, surfaces, and wireframes with materials defined by layer properties

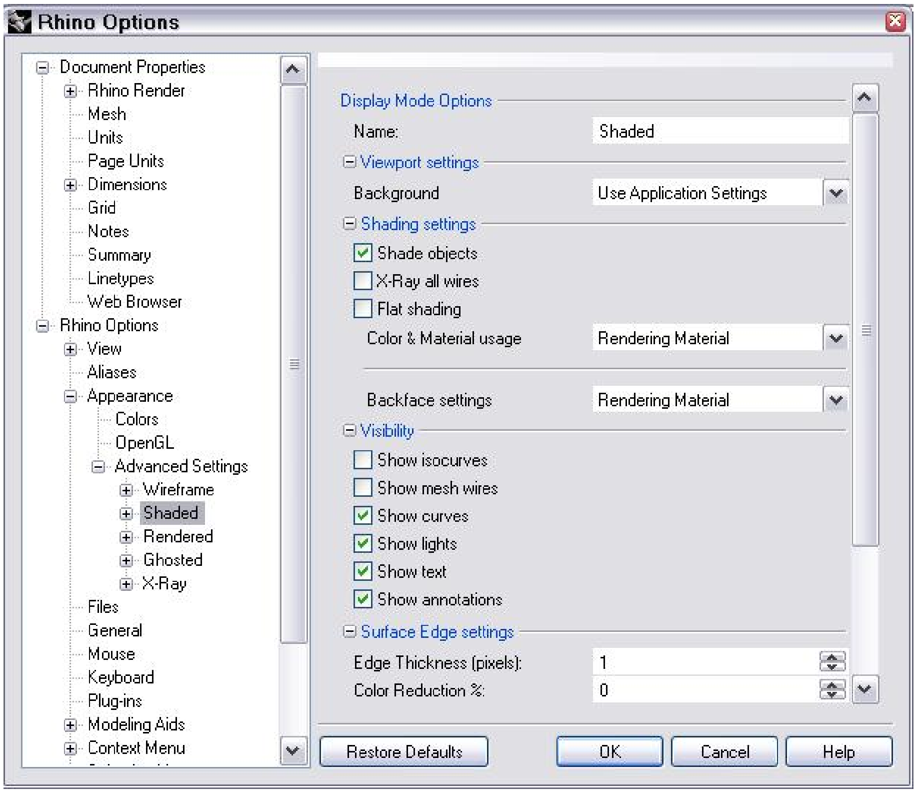
- Ghosted (**ghostedViewport**) : displays solids, surfaces and wireframes with materials defined by layer properties with a higher transparency

- Render (**renderedViewport**) : displays solids and surfaces with materials defined by material properties

Included in this menu is 'Viewport Properties,' which can also be entered as a command

ViewportProperties, which is explained in [EXPORTING]. *>View>Viewport Layout>*

'Display Options' can be found in this menu as well, which gives you an opportunity to change appearance of the grid and background colors. This is all in the main **Options** menu.

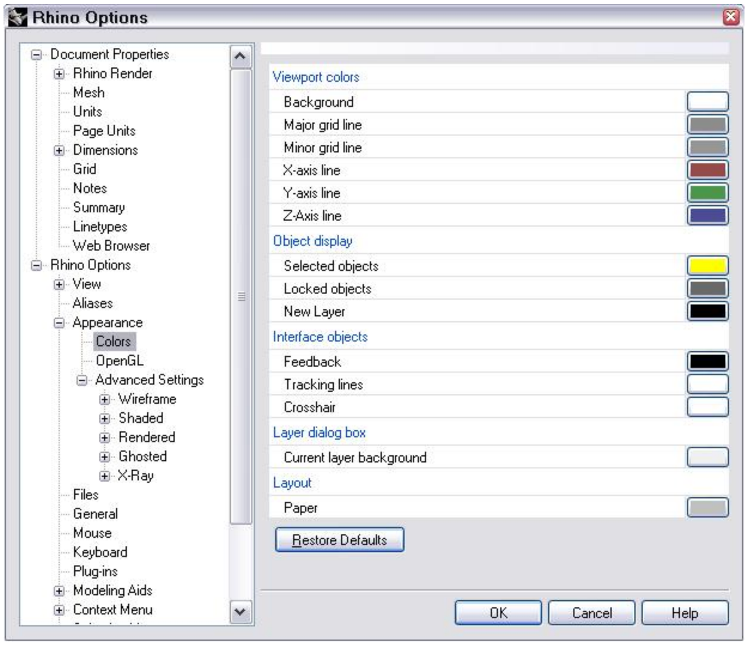


The "Display Options" box will bring up the appearance options for the current rendering style you are using. In the box above, the 'Shaded' mode is active, so that will be the mode that is altered if any changes are made in the options dialog box. Some useful settings in this options box are: Shading settings, Visibility, and Surface Edge settings.

Shading settings : Determines how objects are displayed. Can be Object's Color, Single Color, Rendering Material, or Custom Material. This option is useful in setting up a workplace with a meaningful color palette.

Visibility : Allows you to turn on and off certain aspects of the display: isocurves, mesh wires, curves, lights, text, annotations. Isocurves are lines on surfaces that help display it's shape. Mesh wires work similarly for mesh surfaces. It is often a good thing to turn these lines on to show a more intricate construction of the model. They can also help you develop new ideas about your project.

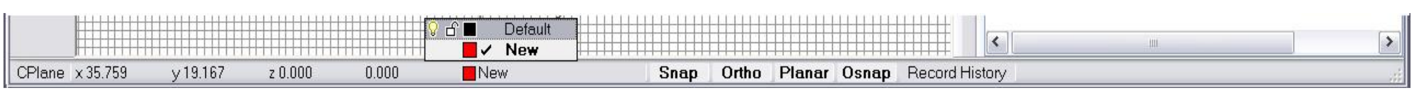
Surface Edge : This will determine the appearance of surface edges, allowing you to adjust the thickness and color saturation.



The appearance of your viewports is crucial to the work you produce. If your viewport is full of lurid clashing colors that are not useful to your ideas or hard to decipher, the model will have trouble progressing in the right direction. In addition, the appearance of your screen is important because you will frequently save time by using screenshots rather than renders [EXPORTING]. Screenshots can be just as evocative and a lot quicker than rendering. If your viewport is set up in an appealing way, then capturing diagrammatic images and moments of clarity can be simple.

'Grid' properties specifically can come in handy in viewport appearance. Keep the grid colors close to the background color and set them to a unit that is helpful for your modeling. If you do not like to use a grid, it is handy to set the grid spacing to a human scale, such as 6 feet, so you have a constant reminder of how large your model is [NAVIGATION].

<u>Status Bar (Snaps etc.)</u>
The status bar is located at the bottom of the Rhino window, and it will help guide your existence in 3-D modeling. Here you will find the current layer (which can be chosen to toggle another layer active), the state of snap modes, and the cursor's location within the coordinate system.



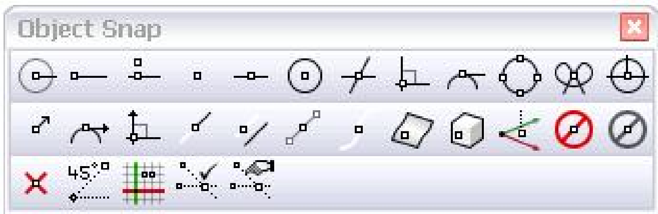
In Rhino, to snap is a state of being. Various snaps and settings can be turned on or off in order to work in different modes including Snap, Ortho, Planar, multiple OSnaps, and Record History. They are indicated by the status bar below located at the bottom of the Rhino window. To snap or not to snap is a very good question. The following will describe the usefulness of these different modes.

Snap : See "Snap command" in the Help index for additional information including the hierarchy of snapping modes. In snap mode the drawing cursor sticks to grid snap points. The spacing of the grid snap can be modified in the Options menu or with **SnapSize**. If you are using this mode, it is most likely useful if you set the size to be the same spacing as the grid itself.

Ortho : See "Ortho command" in the Help index. Ortho mode restricts the movement of the cursor from the last point created to multiples of an angle specified in the **OrthoAngle** command. The default angle in ortho mode is 90 degrees, which would draw horizontal and vertical lines.

Planar : See "Planar command" in the Help index. Planar mode limits picked points in a command to the same construction plane elevation. This can be useful in setting up contour levels at specific heights, and it can also help when working in perspective mode which can be optically deceptive.

OSnap : OSnaps or object snaps are used to add objects to one another or create connections between objects with a specific predetermined geometrical relationship. See "Osnap" in the Help index for an explanation of each.
- The Object snap toolbox can be found under the Layer Options Toolbar. Here, the "one-pick" Osnaps or "persistent" Osnaps can be chosen. "One-pick" snaps are useful as an exception to the mode in place, if one click needs to take on a different snap than the rest. "One-pick" snaps must be chosen while active in a command, and deactivate after one click.



- OSnap brings up a list of OSnaps that can be turned on and off by typing the underlined letter and also indicates what is on and off.
- The OSnap toolbar can be turned on by choosing OSnap in the Status Bar. A series of check boxes display the possible object snaps that can be turned on.
- DisableOSnap brings up the OSnap toggle options.
- Project forces the object snaps to the construction plane so that no matter what is snapped to, the drawing will occur directly above or below that point on the active construction plane. See "ProjectOsnap" in the Help Index.
- STrack, or Smart Track, is a function listed in the Object Snap box. When enabled, it allows the use of temporary reference lines and points to create relationships between objects. See "SmartTrack" in the Help index.

Record History: See "history" in the Help index. Records the history of objects and updates 'history aware' objects. **History** brings up a list of Record History commands.

IMPORTING | The information you need

It is important to remember that 3-D modeling is not a stand-alone mode of design. The designer should be constantly moving from drawing to digital to physical models, printing screen shots and tracing over them, then bringing them back in to Rhino as a bitmap reference for continuing a model, importing photographs or sketches as a texture map or a reference for geometry. This is how ideas come together, not through one medium, but rather through the integration of many diverse and nearly simultaneous operations.

Bitmap Commands

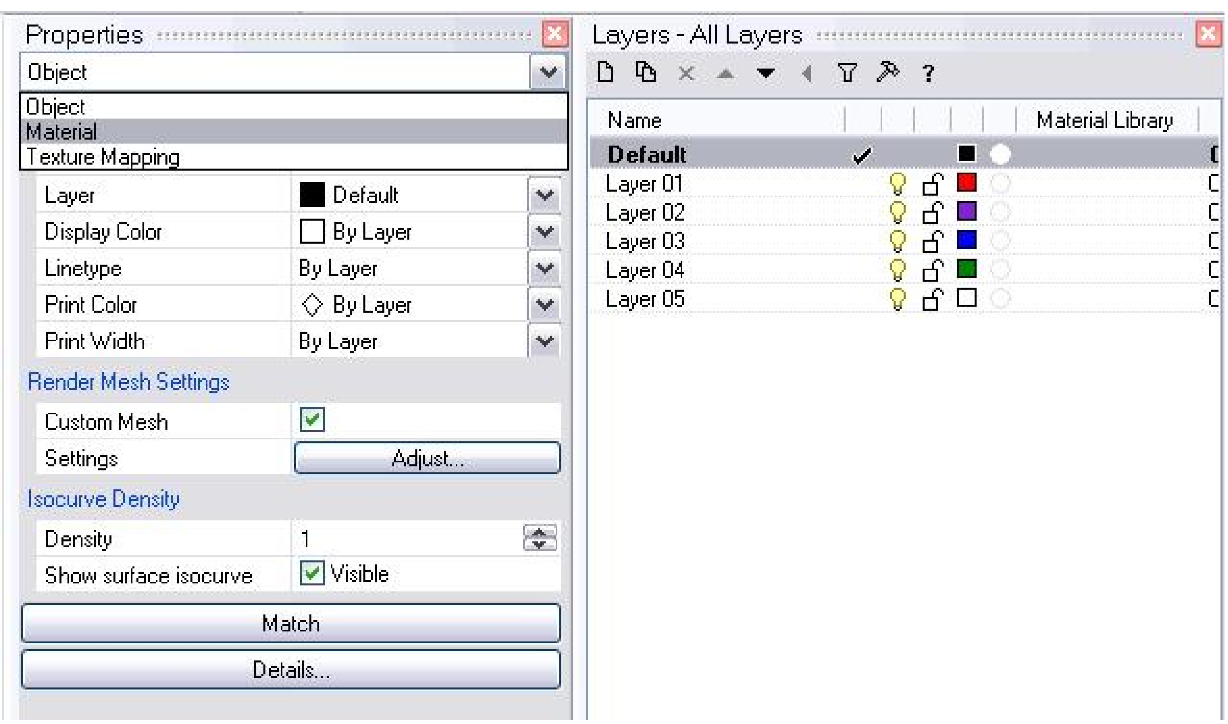
BackgroundBitmap : Places an image in the background of the active viewport. Background bitmaps cannot be transformed like objects; scaling, moving, and other changes must be done through the **BackgroundBitmap** command. This can also be found by right clicking the blue viewport name box, or going to *View>BackgroundBitmap*. Only one bitmap may be placed in a viewport at a time, and bitmaps will not appear in render unless specified in render options.

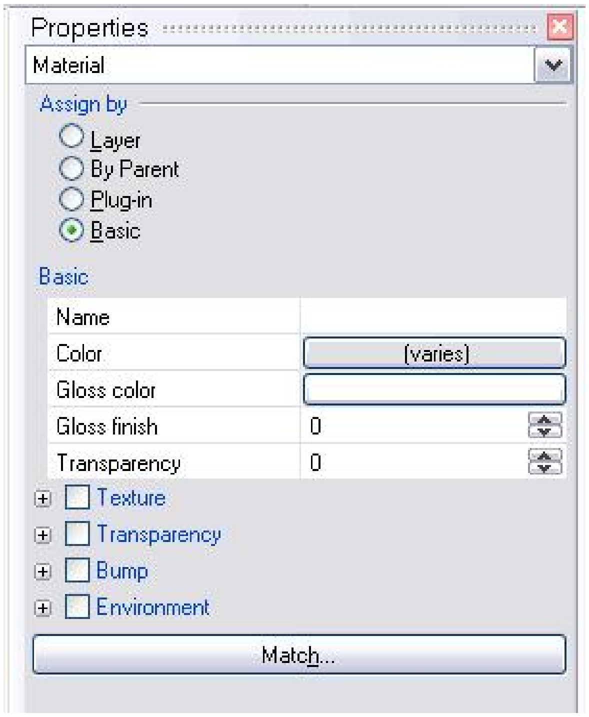
PictureFrame : Allows you to open an image file and create a plane constrained to the size of the image that it is mapped on. This functions similarly to a background bitmap, except the picture frame object is fully manipulable, whereas a background bitmap is locked in place. This way, the picture frame can be put anywhere in 3-D space, rather than on the construction plane like a background bitmap.

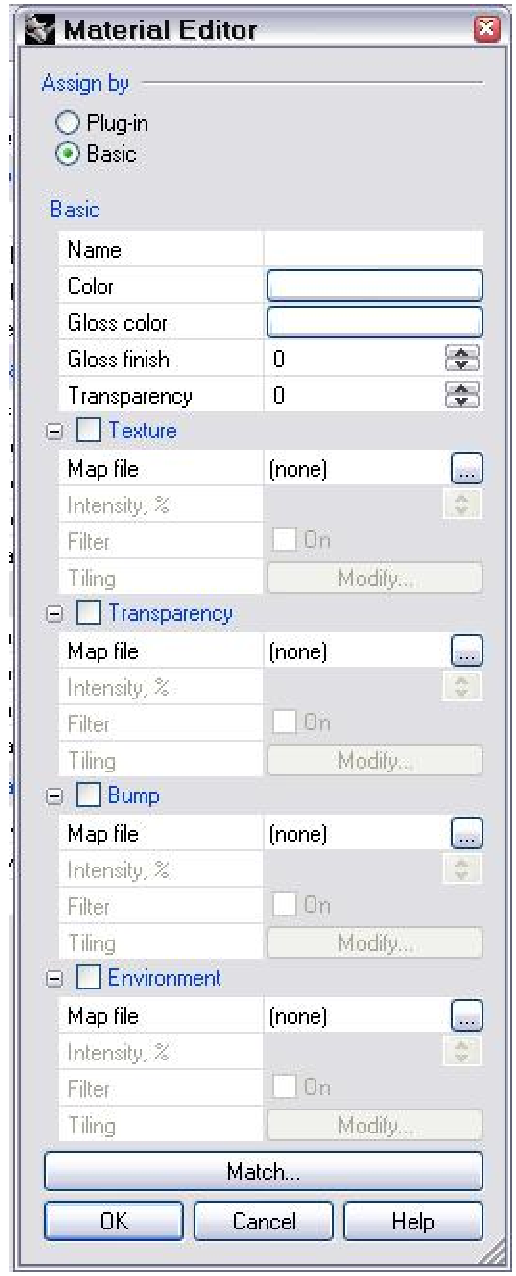
[PLACING A BITMAP] shows step by step use of **BackgroundBitmap** and **PictureFrame**. It also describes the issue of map scale vs model scale.

Material Mapping

Materials can be mapped to all the objects in a layer, or just to a single object. In order to change the material mapping of all objects in a layer, click the circle next to that layer in "Layers" to get to "Material Editor." If you would like to change a single object's material, choose the object and then "Material" from the drop down menu in the Properties box.







In both of these material editors, there is the option to set a material color, gloss (how much and what color of light reflects off of an object), transparency, and also mapping opportunities: texture, transparency, bump, and environment. These settings will map an image or bmp file over an object. The transparency mapping is especially intriguing, as objects with different textures can be layered, hidden, or revealed to display different conditions of ground. The 'Match' option allows you to choose an existing object on the screen to adopt material mapping properties from.

Other File Types

The movement between modeling programs is also crucial to the design process because one program cannot handle all the operations possible in the current digital realm. The transfer of one file type to another program can be difficult, but with some research, or usually using the program's "Export" or "Save As" function, it is usually possible. Some other programs you may be interested in collaborating with Rhino include GIS (Landform data), AutoCAD, Digital Project (Frank Gehry parametric modeling program), Microstation (3D modeling), Generative Components (Parametric modeling). Graphics programs such as Adobe Photoshop and Adobe Illustrator can be a useful way of 'finishing' a drawing or creating a collage of images, screen shots, wireframe views, or renderings [EXPORTING].

OBJECTS AND OPERATIONS IN 2D | Drawing and Connecting

Drawing in 2D seems like basic geometry, but staying (or returning) to two dimensions can be useful for sketching and publishing ideas. Having control and an understanding of the simplest geometries will give you further control over the complex ones later. These tools should become second nature.

Note that all objects in Rhino have a direction except points. This will be critical to surfacing and solids later [3D OPERATIONS]. Try to be organized with your base geometries, so you can return to them later and they will still be clear and useful.

Objects

A **Point** is an object without direction or length. They are useful for creating grids (**PointGrid**) and endpoints for lines or other objects.

Line draws one straight line, while **PolyLine** draws a series of connected lines. When a **PolyLine** starts and ends at the same point, Rhino closes the line [ISSUES].

Other objects include **Circle**, **Rectangle**, **Arc** (Arc: Start, End, Radius is a useful one). For curves and other more complex objects, see [CURVES].

Connections

These commands become efficient when good selection techniques are used. Note that you can **Trim** or **Extend** multiple objects by using a selection box. It may be easier to select multiple objects before you start the **Join** or **Group** command, since then you have the opportunity to use **SelLast** in case you mis-click.

Group : Allows multiple objects to be selected as one. A good organizational tool if Layers are not appropriate.
Trim : Pay careful attention to the command line when you trim; the first object selected is the cutting object and the second object is deleted depending on where you click. Be careful that the objects are overlapping in all appropriate dimensions.
Extend : Extends one object to meet another. If Extend fails it usually means the objects will not intersect anywhere in space. Check that both objects are on the same plane. Rhino is also sensitive to which end of the object you click to extend; if possible, a double click will usually extend both ends.

> Dynamic : Extend is not always dependent on having two objects; you can extend to a point or length you specify. To access Extend Dynamic press Enter while **Extend** is active.

Split : Works just like trim but it is necessary to manually delete the part that you wish deleted. Useful for when Trim refuses to work.
Join : Makes one object from two or more connecting objects. Joining is sensitive to selection order and has a few logic-based rules. See more at *Rhino Help>Join*.
Explode : Explode essentially is the opposite of Join; it makes multiple, more basic objects out of one object.

<u>Control Points</u>
Control points are useful for editing existing objects. For example, if a line is drawn and the end does not reach the desired point, the control point for that line can be moved as follows: 1) Turn on control points for object. 2) Select point (or points) to be adjusted. Note that if multiple are selected they will move as a group based on the point selected as the origin. 3) Drag points or use **Move** command to reposition point(s).
PointsOn (F10 key)
PointsOff (Or in order to turn off all control points, use ESC key or the F11 key.)

Note: Objects cannot be selected for other operations when their control points are on. Only their control points can be selected until turned off.

TRANSFORMATIONS | Tools that will change your life

Transformations are precise, rule based alterations. You should be aware of what inputs you are asked to provide and to consider the implications in terms of your intentions.

All of these transformations require attention to the coordinate system [NAVIGATION].

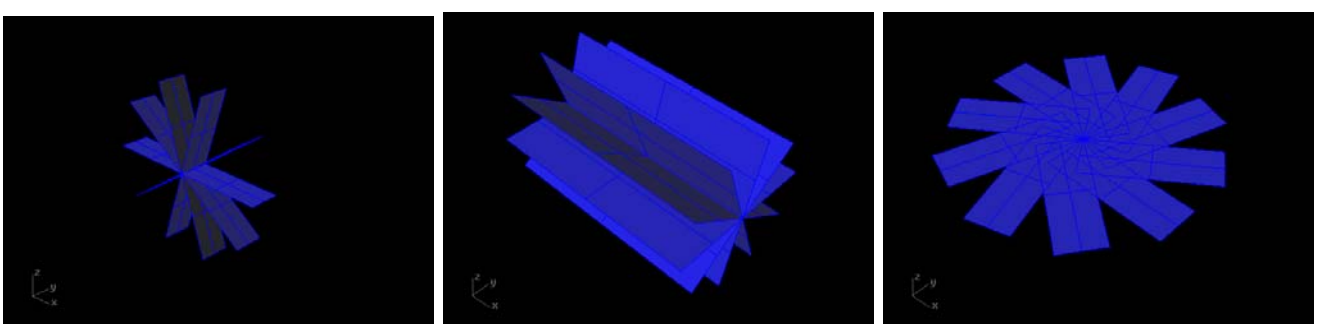
Transform

Move
Rotate : Only rotates the object perpendicular to the active construction plane. For rotating in other directions, use **Rotate3D**. **Rotate3D** can get tricky if you are not actively watching all Viewports.
Scale : Scales in all dimensions, unlike **Scale1D** or **Scale2D**. If you do not know your "scale factor" draw origin and destination points before you use the command.
Mirror : Reflects the objects across a mirror line you define.
Offset : Copies all points of an object to a distance you specify (type distance while in the command). Where you click in the viewport decides which side of the object the copy is offset to.

Multiply

Copy : Can be "InPlace" to add a copy of an object to another layer, etc.
Array : Sets up a regularly spaced "grid" of modules. (Can be spaced differently in each axis)
ArrayCrv : Arrays a module along a path curve controlled by distance between or number of modules desired.
ArrayPolar : Arrays in Z direction around chosen point.
The command prompts are as follows:

-Select object.
-Select origin (Must be chosen in the viewport you wish to rotate around the z axis. Use the World View Icon to help you determine the correct orientation.)
-Number of items.
-Angle.
Play around with this tool because it can be tricky.



Array Polar

Divide : Evenly divides a line by number of divisions or length of divisions and places points at those divisions.

PRINT AND FABRICATE | Experience Your Model with all 5 Senses

Creating and manipulating physical forms of a model produced in Rhino is essential to most design processes. Fabrication should influence how you build your digital model and vice versa. Consider scale and materials critically!

Print

Print makes the "Print Setup" pop up. The preview on the right will be a capture of the active viewport.

For "Destination" assistance, including setting up UVA printers on your personal computer, plot settings, and the like, consult the Computing Help Desk in room 304, or visit http://www.arch.virginia.edu/computing/printing/. Pay careful attention to the "Printing Guidelines" page.

Often it is difficult to control the lineweights when printing directly from Rhino (usually they are too heavy). Lineweight is decided by material properties; if you click **Properties** you can change the "Print Width" without changing the appearance of your model on-screen ("hairline" width usually creates a cleaner print). This goes for the "print color" as well.

For printing in programs outside of Rhino, see [EXPORTING].

Fabricate

Fabrication is exploration! How you fabricate should further your ideas and lead you to new discoveries. UVA has several different machines for Computer Numerically Controlled (CNC) Fabrication: the 3D printer, router, and the laser cutter. Each machine utilizes a different fabrication technique; the 3D printer is additive and slowly prints a small model with spools of plastic, the router is subtractive and voids areas of solid material, and the laser cutter is also subtractive, cutting outlines into sheets of material. Each technique requires the selected geometry of your model to be in a different form.

You cannot just send your Rhino model to the laser cutter; fabrication requires planning and design steps to get your model from digital to physical. Keep in mind material properties, machine tolerances (and exactness), tectonics of the pieces after you've fabricated, and prototyping. More on fabrication will be covered in class.

CURVES | Check them out

"Since most fields are orthogonal, a curvilinear figure with no shared properties with the field will be more distinct than a rectilinear one." from *A Primer on Composition*, R. Dripps.

The apparently unruly, promiscuous behavior of the curve can bring to it an unwarranted distrust and even fear. This is a mistake. Curves have precise structure, a substantial mathematical foundation and a formal logic that makes them exemplary characters within any construct. Once this is understood, the curve becomes a useful friend. Even better news is the ease with which the underlying geometrical structure of curves is made explicit and thereby easily and understandably manipulated by the commands within 3D modeling programs.

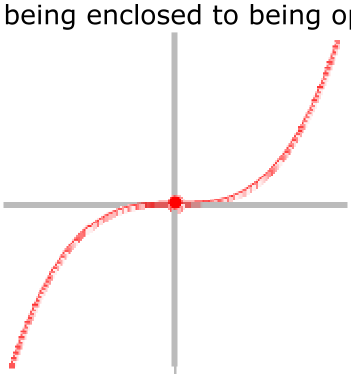
There are a series of named curve structures in 3D modeling programs. These include circles, conic sections, and freeform curves such as Bezier, B-Spline, and NURBS. If you are interested in further elucidation of curve geometry, see *Architectural Geometry* (Pottmann, et al.): "Curves and Surfaces" (211) "Freeform Curves" (253).   An online resource is:
http://www.mathworld.wolfram.com

Curve Basics

Curves can be derived in numerous ways - from simple pure geometries to multi variable mathematical equations. The nature of a curve can be as a *distinct undeformed arc* that is differentiable from land form, or as an *incorporated deformed arc* whose shape responds to given conditions.

The logic of curves informs how Rhino tools draw and edit those curves, and so understanding the following concepts leads to understanding the curves in your model:

Inflection Points
An inflection point is where the curve changes concavity from positive to negative. This is useful in determining the construction of curves. It is also a means by which space or activity changes from being enclosed to being open to the world.



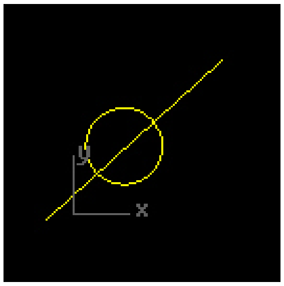
Inflection Point indicated as the red dot.

Direction
Curves have direction that is established by their construction, or the order in which they are constructed. The direction is important to functions involving curves such as **loft** and **sweep**, where a consistent direction is expected. To determine a curve's direction use the **Dir** command [ANALYSIS].

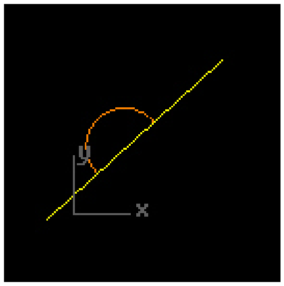
Control Points

Control points are the means by which a curve's shape is manipulated. They are critical to explaining the different types of curves found in Rhino.

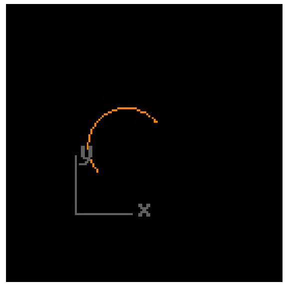
The following examples distinguish a *distinct undeformed arc* and an *incorporated deformed arc* (mentioned above). A *distinct undeformed arc* can be thought of as curves derived from circles and conic sections. Control point locations are decided from the inherent logic in circles, not from your clicking:



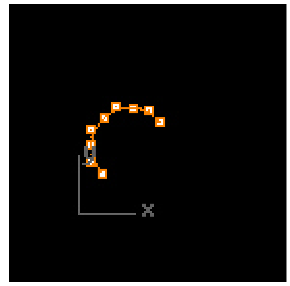
Circle



Trimmed circle with trim line.

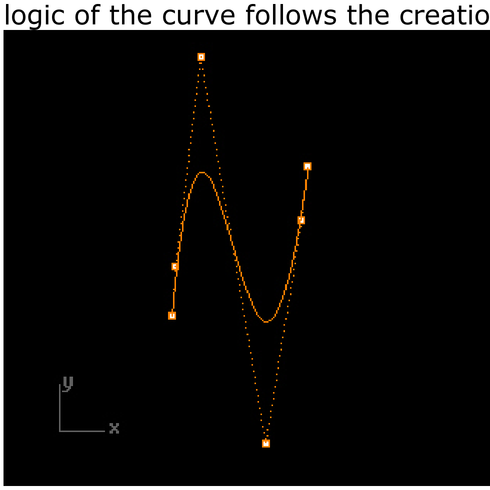


Curve remains.



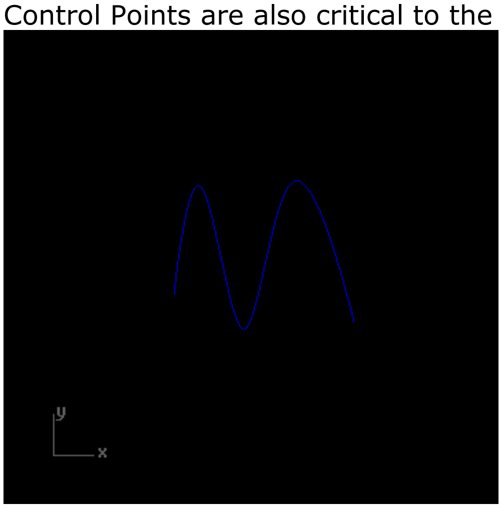
Control Points.

And an *incorporated deformed arc* are more freeform curves produced from control points, and the logic of the curve follows the creation of the points:

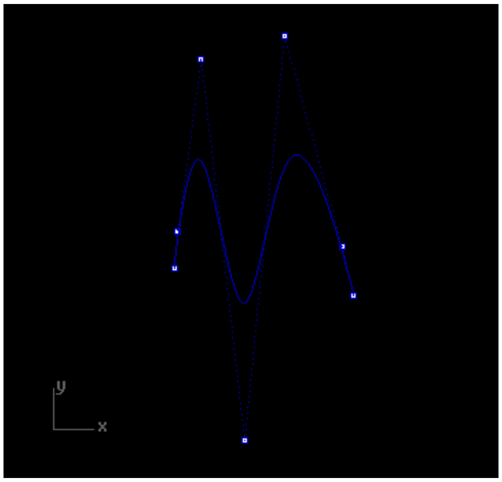


A curve with 6 control points.

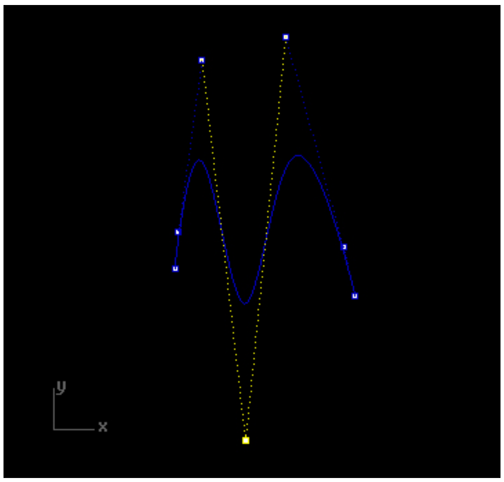
Control Points are also critical to the manipulation of curves:



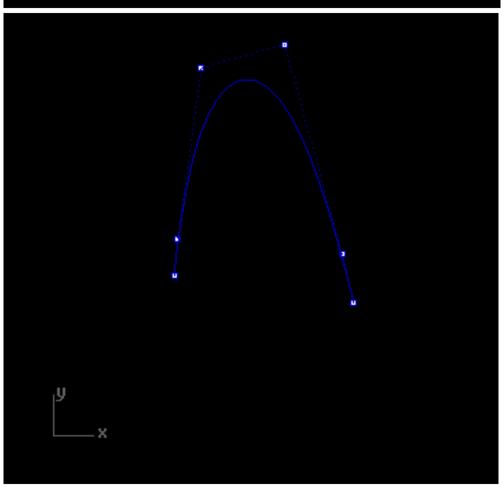
Curve.



Control points activated.



Control point selected.



Control point deleted.

Control points can be manipulated using the command prompt bar or simply dragged using the mouse by clicking on the point (or points) and holding while you drag. When dragging control points to meet other objects, the object snaps (OSnaps) are useful.

<u>Degree</u>

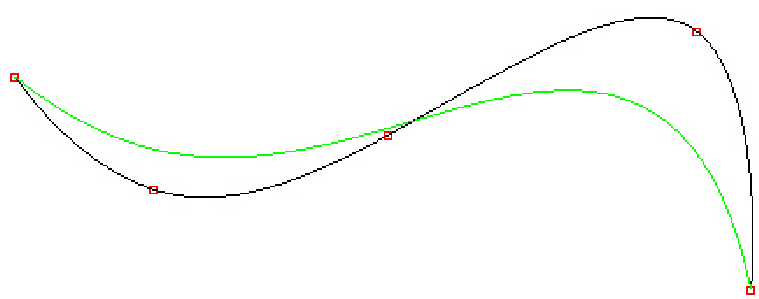
Degree is determined by and determines the number of bends in a curve. This comes from the mathematical equation (rational polynomial) that derives the form. The degree of a curve is one more than the number of bends.

Degree is an option within the Rhino **Curve** command. Rhino will draw a curve to the degree you request if you draw at least one more control point that the degree you have entered.

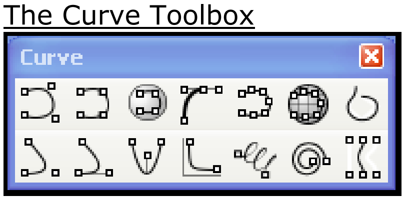
Curve Creation

In Rhino, there are various types of curves that differ in their behavior - control point curves, interpolated curves, etc.

Here, two basic curves are constructed using red points. An interpolated curve in black, and control point curve in green:



The Curve Toolbox



Curve Sets up a curve from control points.

CurveThroughPt Curve through points sets up a curve based on a defined point set. The points are selected, then the curve is created.

- When selecting the points, order matters, unless you use a selection box AT ALL (even to select one point.) The line will be drawn connecting the points in the order you select them. Otherwise, the command chooses a logical way of connecting them.
- This command has the option of using an interpolated curve or a control point curve.

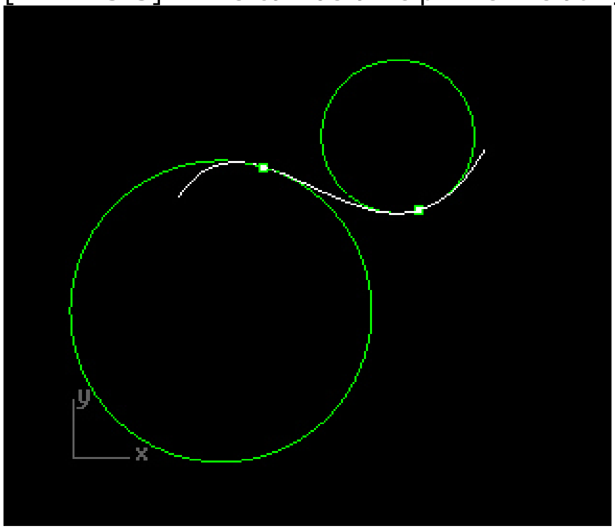
InterpCrv Useful in tracing contours or other existing curves because the curve follows through the points created.

InterpcrvOnSrf Allows you to draw a curve on an existing surface. (Use with **Fin** tool.)

The toolbox also includes conics, parabolas, and hyperbolas which are derived from mathmatical equations. Try all the curve tools to get a feel for how they work.

Transform
Curves can be made by deforming or adjusting existing geometries using the transform tools (**Trim**, **Split**, **Extend**, etc), using the curve tools in the curve pop-out toolbox, or using the arc toolbar [TRANSFORMATIONS]. Curve tangents can be a way to draw together curvilinear and rectilinear forms.

Tools not found on Curves Toolbox
Curvature This is a curve measurement tool that allows you to determine the radius of a curve at any point on that curve. The radius can be retained or just printed in the command box [ANALYSIS]. This can be a help when relating curves to one another and to rectilinear geometries.



Curve's radii produced by **Curvature** tool. (MarkCurvature option set to Yes.)

DimRadius [ANALYSIS] can also be used to measure the radius at any point along a curve or circle.

Length [ANALYSIS]

SoftEditCrv (Curve tools) Manipulates an existing curve to move closer to the curvature desired. By choosing a point and dragging it to another, the curve's control points are modulated based on the movement inputted. The end points are left in their original positions. This tool is very useful for contour tracing, as you can go back and edit the lines you create to make them closer to the existing contour.

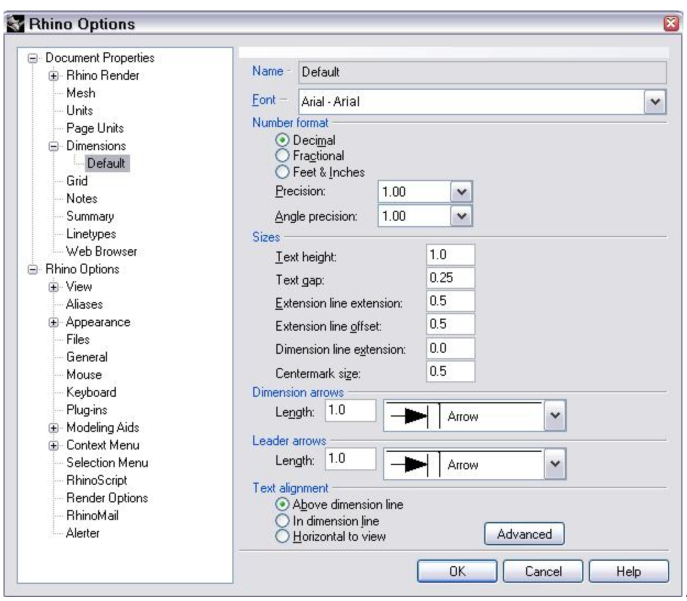
ANALYSIS| Figuring out what you've done

The analysis tools can be used on existing geometries. The Analyze toolbox can be opened from the main toolbar, from the drop down menu, or by right clicking the dotted line on the main toolbar.

Analyze Tools

Dir : Shows an object's (surface, curve) normal direction and allows it to be changed using the flip command.
Flip : Reverses the direction of a curve or surface. Useful when lofting curves or extruding surfaces. To build a surface, the direction of curves should be the same, so if the surface isn't working, try analyzing the directions and flipping where necessary.
Length : Determines the length of a curve.
Distance : Determines the straight distance between two points.
Angle : Measures the angle between two lines. (Can be set up along surface edges.)
Radius : Measures the radius of a curve.
Curvature : This is a curve measurement tool that allows you to determine the radius of a curve at any point on that curve. The radius can be retained or just printed in the command box.
Area : Measures the area of an enclosure. (Closed polyline, surface) If you use this tool on a polysurface or a solid, it will return the sum of all the surfaces' areas.
Dim : Dimensions a vertical or horizontal distance. Note, this dimension is affected by the units precision determined in the Rhino Options menu.
DimRotated : Dimensions a specified rotated distance.
SelBadObjects : Selects objects that are bad geometries or incompletely defined. These objects should be deleted or rebuilt. This command is useful if you are having problems with rendering or navigation. If something seems wrong with an object, this command will tell you [ISSUES].

Dimension Styles



In the Rhino Options Menu

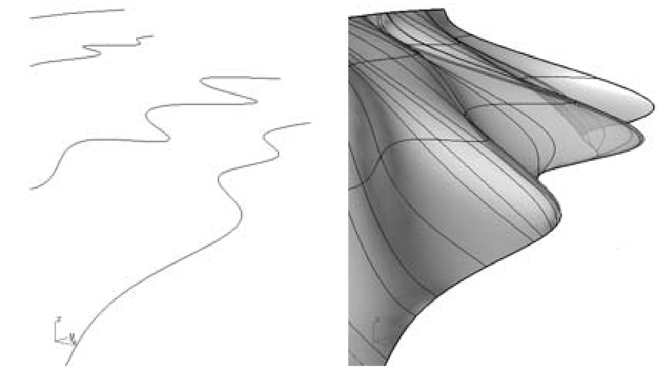
Multiple new dimension styles can be created and modified in the Rhino Options menu. To turn a style on or off, use the **Dim** command then choose "Style."

3D Operations | Building Surface and Depth

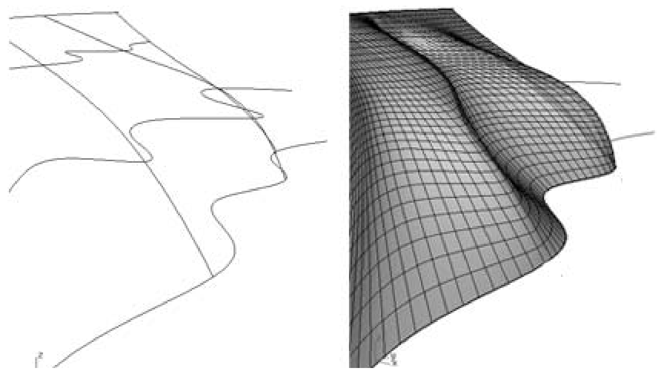
There are a few different ways to explore the 3rd dimension. Turning curves into freeform surfaces results in a skin defined by malleable control points, while solid tools have more rigid dimensional controls. Both are appropriate in different situations.

Surface

Try out all surface buttons!
SrfPt : simple surface but 3-4 corner points is often too little
EdgeSrf : useful because the edge curves can be out of plane**.** Note that you can select the edges of surfaces in addition to lines or curves, but only while the command is active. Order of selection matters!
PlanarSrf : creates a simple surface but does not work if curves are not within the same plane
Plane : hold toolbar button down to get further options or type **Deformable** to get a more useful rectangular plane**.**
Loft : stretches surface between a series of curves. Explore options for further control, and if the surface is still too complex, use **NetworkSrf**



Contours lofted create strange horizontal bumps.

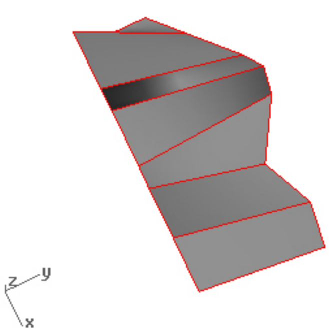


The same contours have regulating perpenduclar lines added to them, NetworkSrf is used, and there are no bumps.

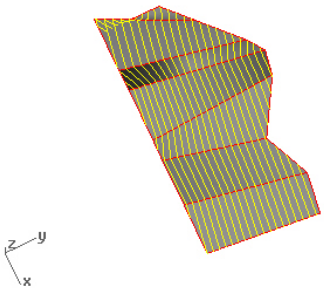
Patch : creates a gridded surface that includes selected curves or points. Use the options box (preview is helpful) to chose the number of grid divisions. Creates clean surfaces that are easy to manipulate once the Control Points are on (**PointsOn**).
Sweep1 : Extrudes a section along a line.
Sweep2 : Uses multiple sections and two edge lines to extrude a changing section shape.

Contouring

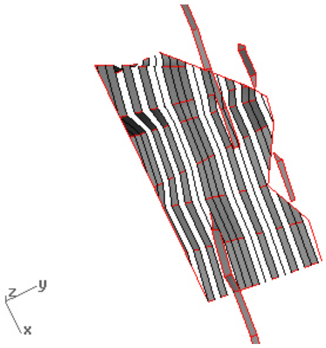
Contour : divides objects at an even distance perpendicularly along a chosen axis. This command is very useful in breaking up curves, surfaces, meshes, and/or landforms to manipulate them as smaller components of a larger whole.



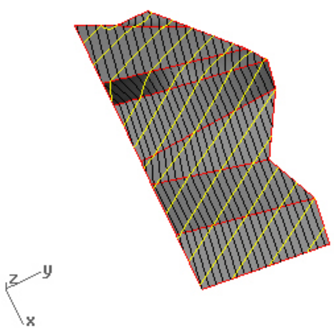
A surface in three dimensions (viewed in perspective) that is created by lofting two polylines.



Contour is used, axis is chosen along bottom edge of surface, and the distance between elements is entered.



This can be used to split the surface and then manipulate the individual elements as needed.



Contour can be used along multiple axes with different spacing.

Multi-step Geometry

Tube : makes a closed cylinder with a concentric hole.
Pipe : makes a circular surface around a curve.
ExtrudeCrv : creates a surface by pulling a curve in the direction perpendicular to the construction plane. Gives you the option of choosing the direction, or extruding both sides.
ExtrudeSrf : creates a solid by pulling a surface in a straight line, gives you the option of choosing the direction, or extruding both sides, or capping the ends of the extruded surface.
Fin : A curve must be drawn or projected onto a surface for this tool to work; **Fin** extrudes a curve normal to that surface, similar to a fish fin extruding from its dorsal.
Ribbon : Offsets a curve and then surfaces between the original curve and offset curve.

Intermediate Transformations

Project : Projects curves or points onto a surface. Only will project parallel to the current construction plane, so sometimes the construction plane will have to be temporarily redefined [SEE TOPIC : NAVIGATION]. *Rhino Help : Project*
Cap : Cap creates a surface over holes in a polysurface. Works best when capping trimmed solids,

extruded surfaces, or other clear capping opportunities. Will not work if the cap surface is too complex.
MakeHole : Similar to **Project**, a closed polyline is projected onto a surface, however, **MakeHole** cuts a hole in the surface in the projection of the polyline and also surfaces the original polyline and the extruded sides between the polyline and the surface.
CutPlane : an extremely useful tool that draws a Plane that overlaps the selected objects in the direction you specify (but be careful to actively watch all Viewports). Makes trimming and splitting easy.

Mesh Tools

Mesh surfaces differ from surfaces because they are able to be controlled easily and smoothly. A mesh can be triangulated or quadrangulated into a series of interlocking surface planes that are controllable with control points. Meshes can be manipulated further by converting them to polysurfaces and then back into Meshes again.

All meshes must be manipulated using specific mesh tools, i.e. **MeshSplit** or **MeshTrim**. They act just like their counterpart commands.

Mesh : Creates a mesh from Polysurface or a surface. Note that it keeps a copy of the original surface, and that meshes will not always line up perfectly with the original.
ReduceMesh : Useful for simplifying the amount of polygons on a mesh surface. This will change the way a surface looks when rendered so try a few tests.

EXPORTING | Get out of Rhino right

Rhino's strengths should be taken advantage of as much as possible, but understanding its shortcomings is just as helpful in the design process. Sometimes it is advantageous to get your work out of Rhino and work with it in different programs (Photoshop, Illustrator) or mediums (physical modeling from printouts or in the CNC lab).
[CELLAR THEATER]

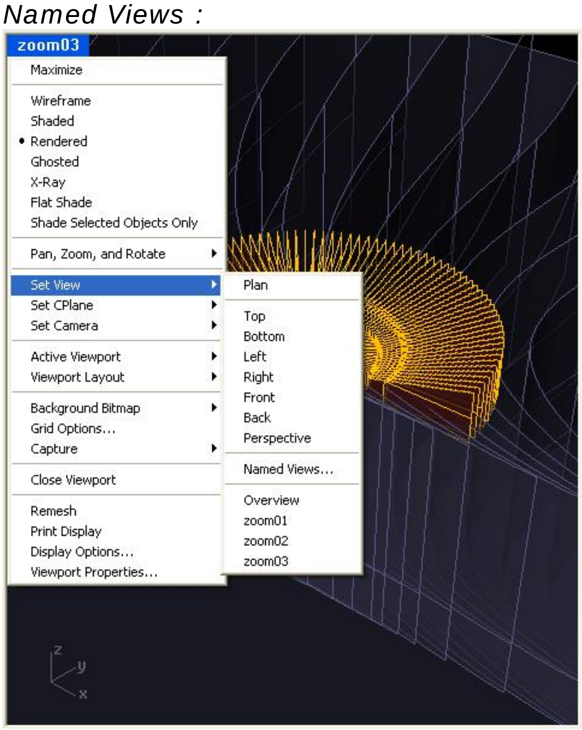
3D modeling software and Illustrator are all Vector based, in other words, line based. Photoshop and other image editors are Raster based, in other words, pixel based.

Raster files (.jpg, .bmp, .tiff, etc)

Note : .tiff saves more information, and is valuable if you plan on doing a lot of editing in Photoshop, plotting a huge render, or creating your final presentation materials. .jpg is great for everything else, and .bmp usually is unmanageable and should be avoided. Think ahead to your eventual dreaded archiving process and what kind of files will have the longest digital life. For the file-type obsessed, see Kirk Martini.

<u>Saving Views</u>
Saving your views means you can return to that view at any time. It's great for documenting your process! It is most useful when your Viewport settings lead to clear diagrams [WORKPLACE].

namedView is found by right clicking the blue box of the active viewport and click *Set View > Named Views :*

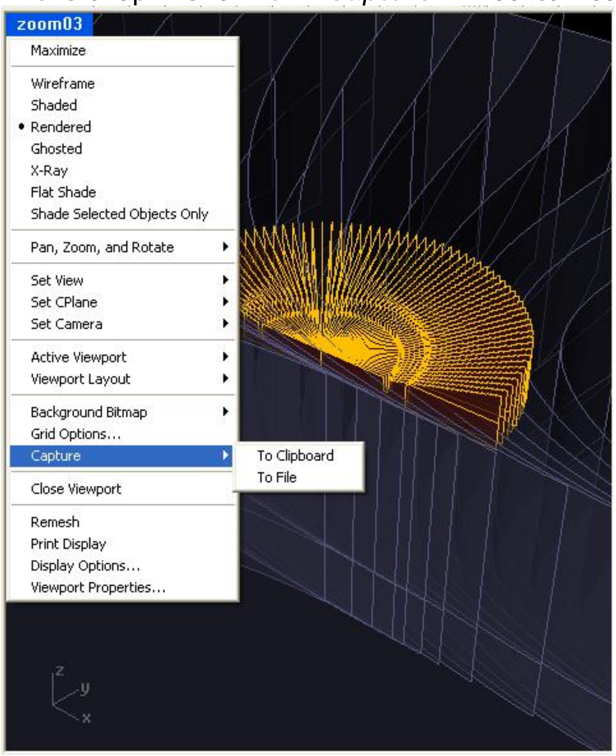


In the Named View popup box, click *Save...* to save the current view and hit Close. Note that the name of your viewport (in the blue box) has changed to the name you just created, and if you right click the blue box again and click *Set View*, the new view is on the bottom of the list. You can also go back to the Named View popup box to edit your saved views.

Capturing Views
There are a few different ways to save your view instead of rendering, and they can be much more useful to your process then rendering.

Print to Image File : Under Print dialogue box select *Destination>ImageFile*. Proceed with print options, choose print. Select file type, location, and name.

ViewCaptureToFile : Image capture can be used on either operating system, and it can be found in the drop menu *View>Capture>*. You can save the image to a file or to the clipboard.



Running Rhino on a PC, you can simply use the "PrintScreen" key on the keyboard. On a MAC, this operation is not as easily achieved, but can be found on the "On Screen Keyboard" that is located under *Start Menu>Programs>Accessories>Accessibility>On-Screen Keyboard*. The On-Screen Keyboard shows up in the image capture, so it needs to be placed strategically on the screen. Once the PrintScreen command has been used in either case, you can simply Paste the screen image into Photoshop.

Rendering
See Rhino Help Topics>Render
A render is just another technique of explaining your idea and should lead to discussion of those ideas. Think about rendering as a tool to describe the performance of your model, not the literal materials.

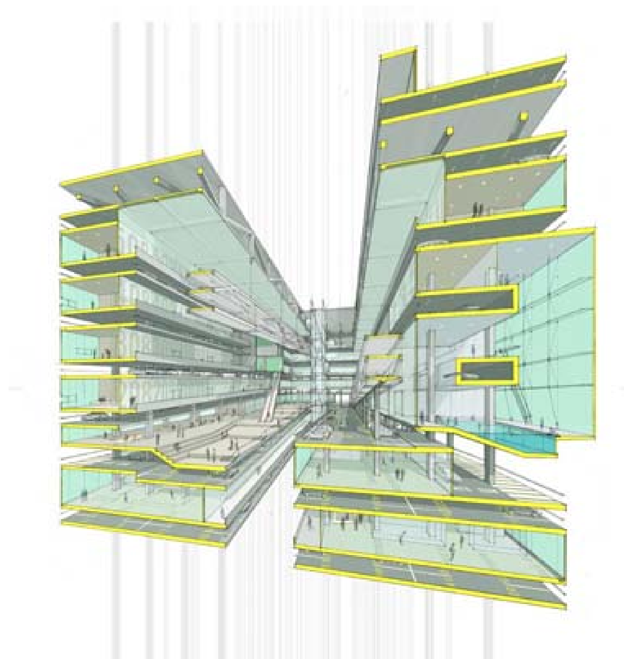
Rendering creates an image of the viewport in a different window with material properties and light sources that you can then save out of Rhino as an image file. There are different programs that produce renderings; the default in Rhino is Rhino Render, which provides basic lighting effects and can be highly effective. You can change the render program at *Render>Current Renderer*. Other renderers require licences, which UVA has, but a personal copy of Rhino may not. Flamingo Raytrace is recommended as an alternative to Rhino Render for renderings where more realistic lighting is deemed necessary.

Most rendering work and adjustment goes toward creating a provocative image in a *timely* *fashion*. While you are setting your material and light properties, choose low-res render settings to preview them, and when you think you've got it right, then make a larger, slower render. Or, if you would like to avoid the hassles of rendering, choose your render and material settings, set your Viewport to **RenderViewport** and then follow the Capturing Views instructions above.

Rendering takes memory, so as your renders get larger it helps to close other programs and possibly walk away [see Topic : ISSUES]. If you plan on rendering for hours make sure the image will be worth all the time and memory!

Always ask yourself, what constitutes a "final" render? A render is usually not "finished" right out of the Rhino oven. Consider the following examples :

22 Front Street, Philadelphia
Hans P. Stein Architects



Park Tower, Venice
Lewis Tsurumaki Lewis

Knowing what you want out of a render saves you a lot of time doing unprovocative modeling. Keeping it abstract may lead to better drawings and design solutions later!

Rhino Render Options
Options>Rhino Render
Resolution and DPI should be increased when images will be enlarged or edited, but doing so will greatly increase render time. 72 DPI is usually too small for any purpose other than previews.

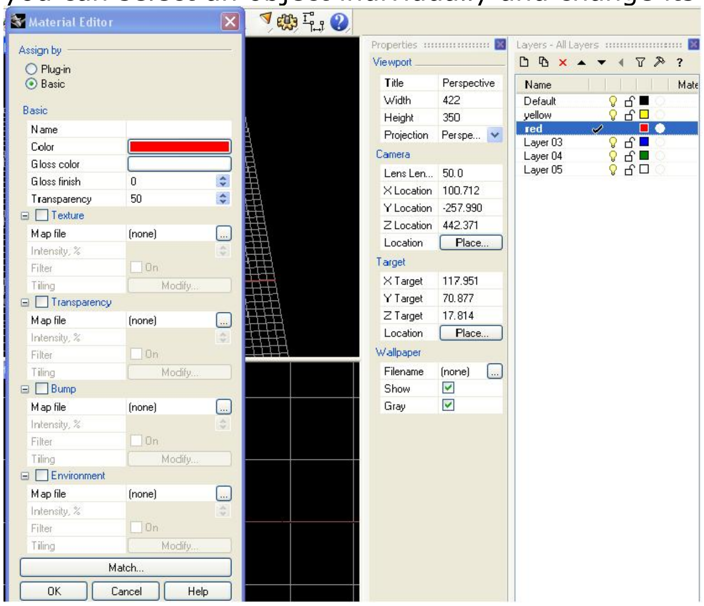
Anti-aliasing minimizes pixel distortion ("aliasing" means distorting, or jittering). The higher you make Anti-aliasing, the longer your render will take, but for complicated surfaces or perspectives, it may be worth it. For the passionate, consult wikipedia : http://en.wikipedia.org/wiki/Anti-aliasing

Background Color : Try out black or white instead of gray, or choose based on how your render will be used or where it will appear. For plotting and photoshop manipulation or layering, a white background makes work easier. Similarly, a black background works best for digital presentations with a black background (recommended for presentations on the wide screen in room C!)

Under "Miscellaneous", play with toggling Render Curves and Render Surface Edges and Isocurves.

Flamingo Raytrace Options
Flamingo is different from Rhino Render in a few ways: Flamingo is better at reflecting light off of surfaces for multiple bounces, it has more advanced material settings, and it has a setting that stimulates the sun (you can set the time and place of your model). For all of these reasons, Flamingo renders take longer. Consult the local experts if you'd like to get into Flamingo; for this Topic sheet only Rhino Render specifics will be discussed.

Materials Editor
Click the circle in "Layers" to get to "Material Editor". Materials usually are assigned by layer, but you can select an object individually and change its material settings in the "Properties" box.



Beginnings:
Rhino render can be extremely effective with thoughtful manipulation a few parameters, namely color and transparency. Keep colors desaturated unless there is specific information that needs highlighting. Colors should be chosen critically and should have meaning and communicate the material's qualities that are significant to the overall composition. Transparency should be taken advantage of as much as possible; although sometimes abstract, transparency can articulate the layering and connections of structure.

Middles:
The next level of complexity involves sources outside of Rhino to define the texture of the material. In the *Edit>Object Properties>Material* command, Rhino applies and alters (to your specifications) image files to the material. Choose texture, transparancy, bump (the surface appears bumpy without changing the structure's geometry), or environment according to your intentions in describing the material. Texture and Transparency are particularly useful (dark portions of Transparency are transparent, allowing for shadows to be cast). Experiment with Tiling, Intensity, and Match.

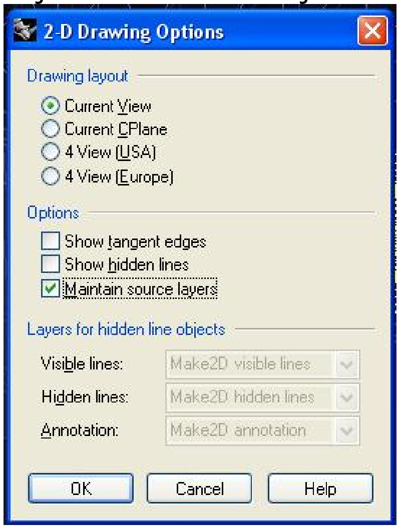
Start a folder with image files of textures and bumps that you are interested in using. Fill it with images found online (best to first manipulate them in Photoshop to be more effective), scanned sketches, or scans and photos of physical models. A collection of these files will continue to be useful in future models, so label and organize clearly!

Ends:
lights! flamingo! custom images! shine! large files! (You'll get there sooner than you think, and whether or not that is a good thing for your process is up to you).

Vector files (.ai, Rhino)

<u>To 3D modeling programs</u>
Export : Exports selected objects to another vector file, including .3dm (Rhino), .dwg (autoCAD), .kml (GoogleEarth) or .ai (Illustrator). This command is also found by right clicking the Save button on the toolbar, or going to *File>Export Selected...* or *File>Export with Origin....* Remember that exporting is relative to the Coordinate System; an exported file will also come with information on where the origin is relative to the objects.

<u>To Illustrator</u>
Exporting wireframes to Illustrator is extremely useful because Illustrator will import your Rhino layers and properties. However, Illustrator is not on the scale of Rhino, so extra steps are needed.
Make2D : Creates a 2D projection of selected objects. Rhino will create new layers for these 2D objects and the 2D objects will appear at the origin of the Top view. This box will pop up :



Show <u>h</u>idden lines : puts the hidden lines on a seperate layer named : your layer name + "hidden"

<u>M</u>aintain source layers : puts visible lines on a seperate layer named : your layer name + "visible"

Another layer will appear named : your layer name + "Annotation" which will include an origin annotation as well as any text or dimensioning that has been selected.

Before clicking **Export** to Illustrator, scale the new 2D object appropriately, and pay attention to where the 2D object is in relationship to the origin. If it is too far away (further than the page extents length in Illustrator, maybe 40 inches), it will not appear in the exported .ai file.

When you Export to an .ai, an options box will pop up asking about scale. A "snapshot of current view" will be at the scale of the viewport at that moment. "Preserving model scale" will scale the model appropriately, but it is a better guarantee that the objects will be scaled correctly in Illustrator if you scale them first manually and then "Preserve model scale" at 1'=12".

[ISSUES and CELLAR THEATER]

Print to .pdf is another way to get the current view wireframe into Illustrator, but scaling and moving must be done in the Print Dialogue box.

ISSUES | We all have them

Wacky Lofts
Lofts get wacky when the direction of the curves to loft are different. Align the directions and the surface may work itself out [SEE TOPIC : ANALYSIS]. Often a Loft will insist on being wacky anyway. In this case, stop lofting, **NetworkSrf** or **Sweep2** for more control. Question why you need to loft at all (try making an implied surface instead of an explicit one).

Zoom
Sometimes when zooming in on an area of the model, the viewport will 'stick.' This means it has reached the front of its picture plane. To solve this problem, you can use the **Zoom** command to drag a zoom window around the area you are attempting to reach. The zoom window you drag becomes the new picture plane, and once you reach that, you'll have to drag a new one again.

RotateView
If you are using the **RotateView** command, and the viewport seems to be reacting in the opposite direction you expect, it is likely your model has turned itself upside down. Once you reorient yourself, it should be fine.

Removing Excess Data
Sometimes you have to manually cut down on the amount of information you're putting into your model, either in raw form (imported contour information) or cooked form (you've spent hours on hours modeling every square inch). Excess data can lead to instant critic confusion and poor design decisions. Be critical of your obsessive compulsive thoroughness!

Organization
Be rigorous in the design of your workspace, especially with layers and filenames! Define layers by the object's materials and purpose. Try to imagine coming back to your file in five years. Will anything make sense? The chances of model clarity improves with organizational clarity.

Open Geometries (Curves and Surfaces)
Rhino defines geometries as "open" or "closed." "Open" means the object has exposed edges or ends. "Closed" means that Rhino understands the object as complete. This will become important only when you are trying to manipulate an object and the fact that it is either "open" or "closed" is preventing said manipulation. **Explode** works well to make a "closed" object into a series of "open" objects. Going the other way, piecing together "open" objects and using **join** to make them into "closed" objects, is much more difficult [SEE EXAMPLE : PIPES]. For curves, turn the Edit Points on and drag the end point to another end to close. You can select either "open" or "closed" objects by using commands such as **SelClosedSrf** or **SelOpenCrv**.

The Incorrectness of Perspective and other Viewport Failures
If you are a perspective guru, you may have already noticed that "Perspective" in Rhino is not the type of perspective we experience in the world. It is perspective derived from linear algebra and so is therefore more mathematically "perfect" than our own sight. Rhino does not have peripheral vision, so often spaces in Rhino perspective aren't really how one would experience those spaces in real life. It is another reason why it is good to get your models out of Rhino and play with them using the imperfection of the human hand.

Rendering
Rendering sometimes fails because of Bad Objects in your model [SEE TOPIC : ANALYSIS]. Rendering sometimes takes far too long and therefore, fails [SEE TOPIC : EXPORTING], or a render makes Rhino crash. See below for Crashing tips.

Crashing
Rhino crashes when Windows runs out of memory, or because Rhino doesn't like your model. If something doesn't work, try doing it in a few steps... for example, splitting part of the objects in a few groups. Or, make more layers with less information on each one, and then turn off more layers while doing commands that take up a lot of memory.

Execute memory-heavy commands in wireframe mode and close other memory-heavy applications that may be simultaneously running.

See your memory use in the task manager. Buy RAM, increase your total memory. Learn from the experts which UVA computers have more memory and then horde them late into the night to make yourself look like an expert too.

If you are still crashing Rhino unexpectedly, walk away. Get some sleep. Treat yourself to hot food. See your loved ones.

EXAMPLE : PLACING A BITMAP

[IMPORTING] [2D OBJECTS] [CURVES]

The topography map is a tool that comes to use with nearly every architectural endeavor, and this 2D image can be manipulated in the 3D digital realm to produce a structured and integrated land form with architecture. Using a contour map as a background bitmap will allow you to easily translate and expand land form data into something more provocative than flat or layered contour lines.

- Place background bitmap.

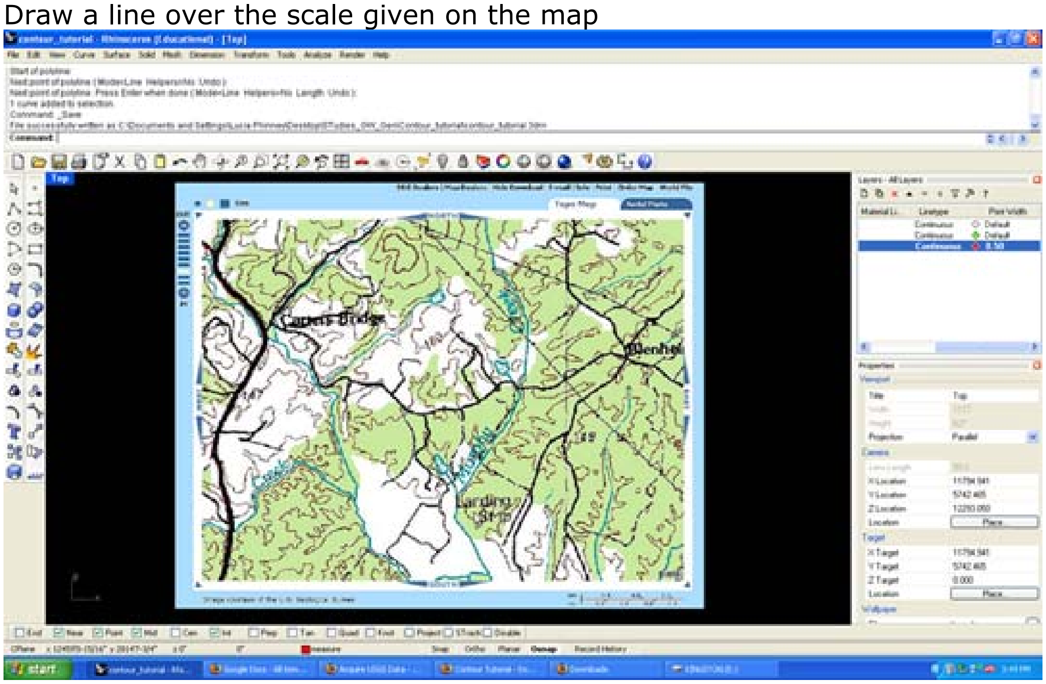
When using a map, it is important that you are aware of the rhino model units and scale as they compare to the scale of the map. When you place the bitmap you have the choice to scale the image, and this is the best time to do that - before you begin to place additional objects. You will need to use the map scale to measure the length of the image and use that length when you place the bitmap. The scale can be true, as in 1 mile = 1 mile, or scaled in the z-axis to exaggerate elevation changes.

-Scaling a bitmap map

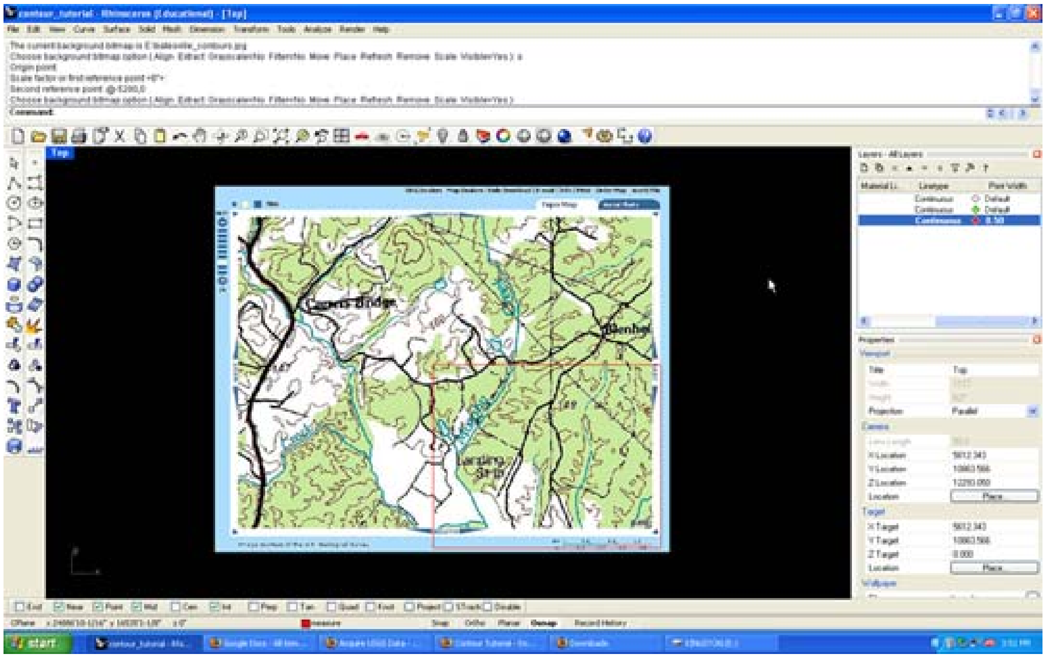
When placing a bitmap in Rhino, it is incredibly useful to retain the map's key scale so you will be able to use the **Scale** command in Rhino rather than tedious mathematical calculations.

Using **BackgroundBitmap**, **Place** the bitmap in Rhino with an arbitrary scale.

Draw a line over the scale given on the map



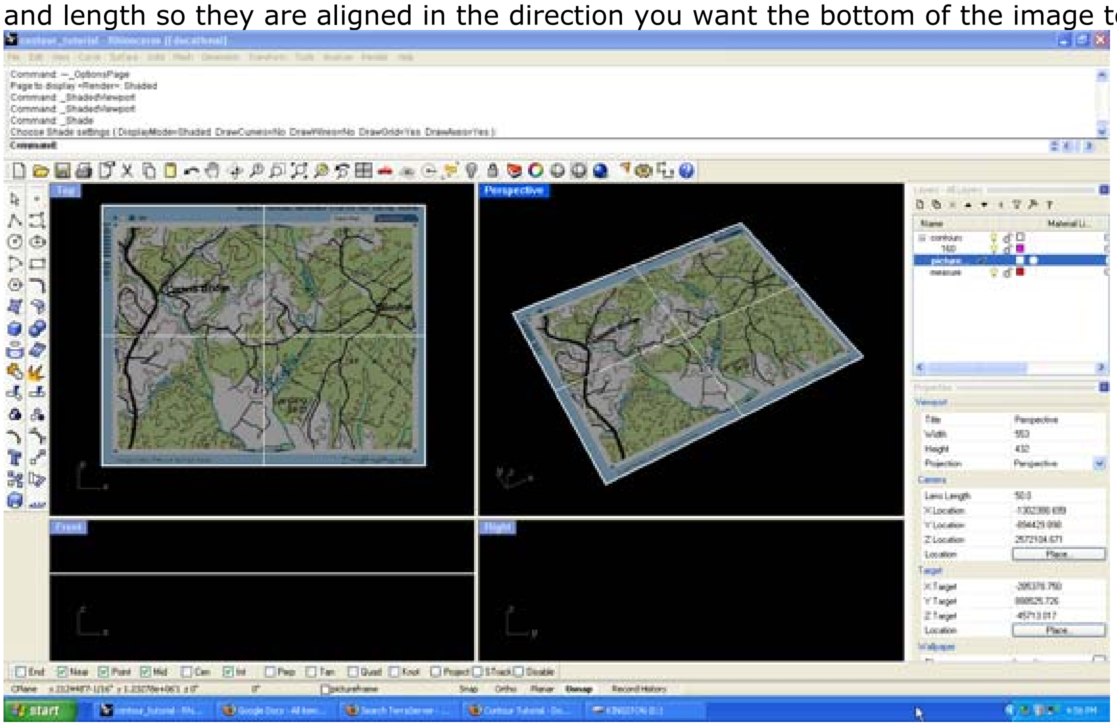
Use the **BackgroundBitmap** options and select **Scale**.
\Select either end of the line.
\Enter the second reference point. (I.E. If your scale is 1 mile, enter @-5280',0)
\The map should be true to scale now. If you want to set up the file to be half-scale, adjust accordingly, but make sure to adjust dimensions on all three axes and keep track of them.



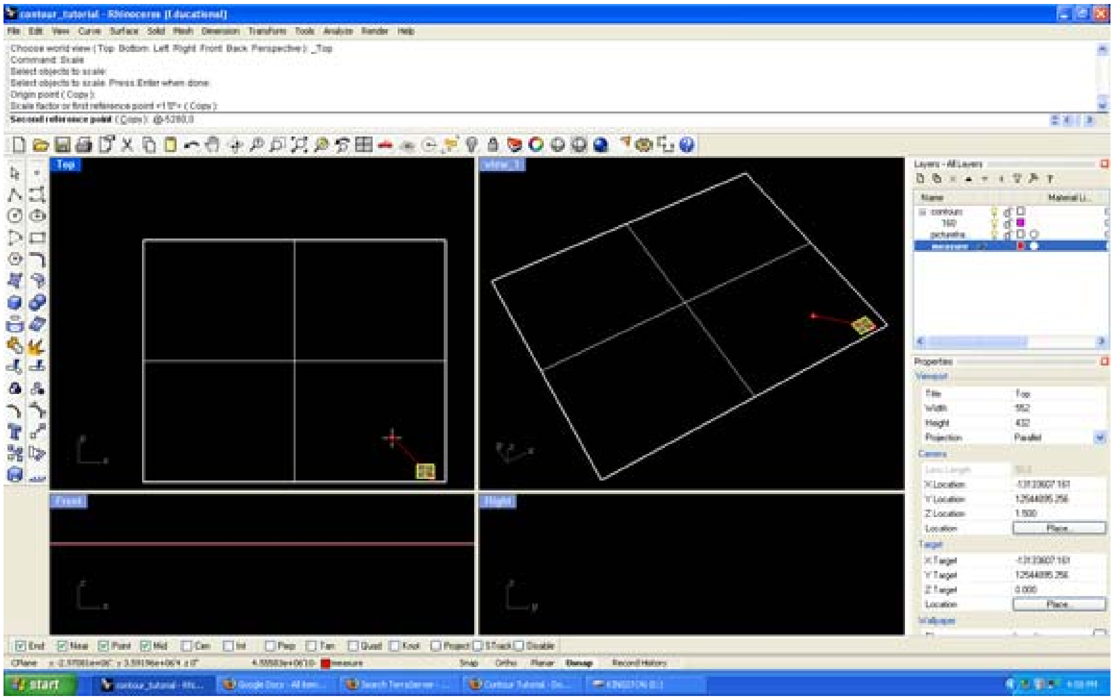
Note the rectangle indicating the previous size of the bitmap.
The background bitmap is only one form of contour information that can be used to create a digital land form model. Digitized contours can be obtained from GIS information and other sources.

PictureFrame can be used instead of a **BackgroundBitmap** to place a contour map into a digital file. This allows you more freedom in changing the transparency of the image to allow objects drawn on top of it to show up, also functions as an actual object that is selectable and visable in all viewports. In order to use this tool most effectively, the **PictureFrame** should be placed on its own layer that is locked when it is not being edited to avoid selecting the object inadvertently.

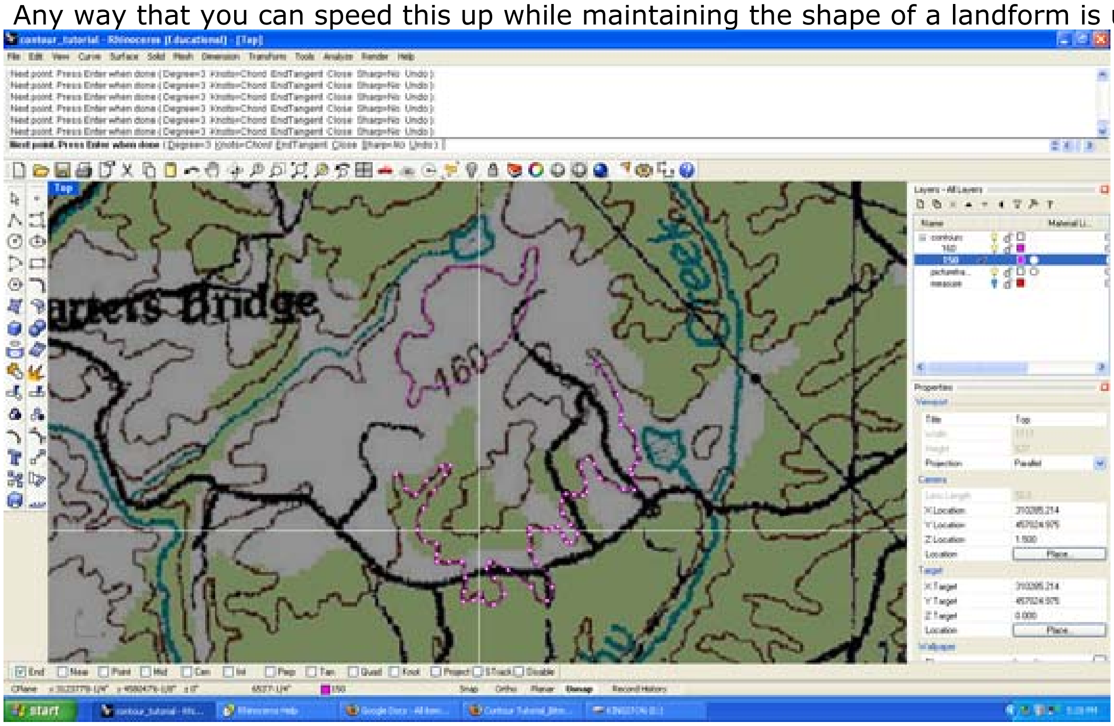
Use the **PictureFrame** command to create the frame. Choose the map file. Choose the start point and length so they are aligned in the direction you want the bottom of the image to follow.



Scale the frame using the same strategy described above, but use the **Scale** command.



Use the **InterpCrv** command to begin to trace contour lines. But avoid overworking it.
Contour lines in a digital model do not have to follow the exact bends of a contour line. It is not always necessary to include all contour lines. The most important thing is that you are getting the appropriate contour lines to represent YOUR IDEA OF THE LAND.
Close observation will allow one to understand the geographic features that are being depicted by contour lines. For instance, an experienced reader of contours can easily distinguish between contours that reveal a valley and contours that reveal a promontory. Once you achieve landform literacy, you might decide to abstract contours to emphasize particular features. Also, it is not always necessary to draw every contour. Digitizing contours is a slow and often repetitive process. Any way that you can speed this up while maintaining the shape of a landform is useful!



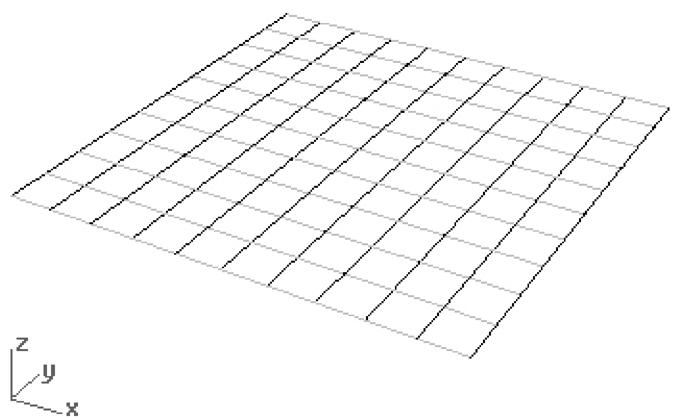
Use layers to save contour lines into groups that will be easily selected and moved to their proper elevation later.

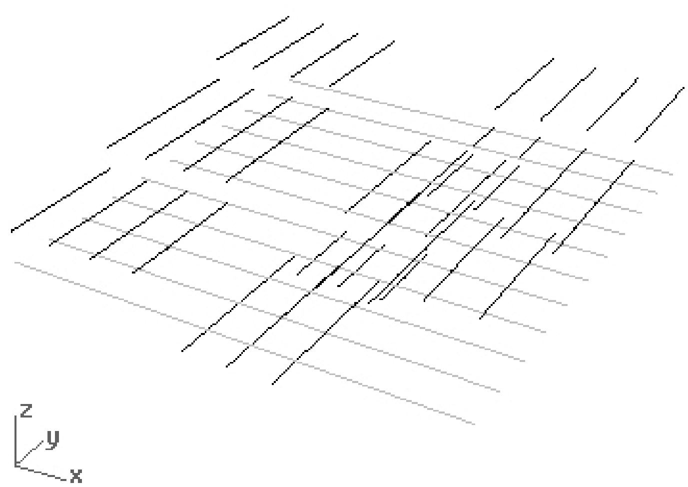
EXAMPLE : GRID

Valley Landform From a "Regular" Grid

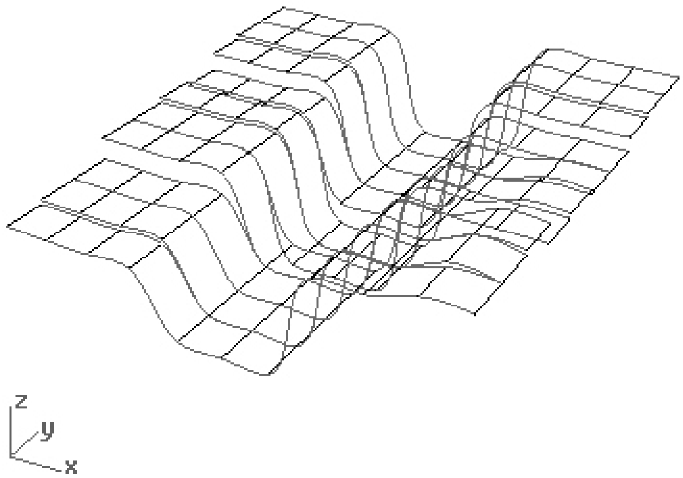
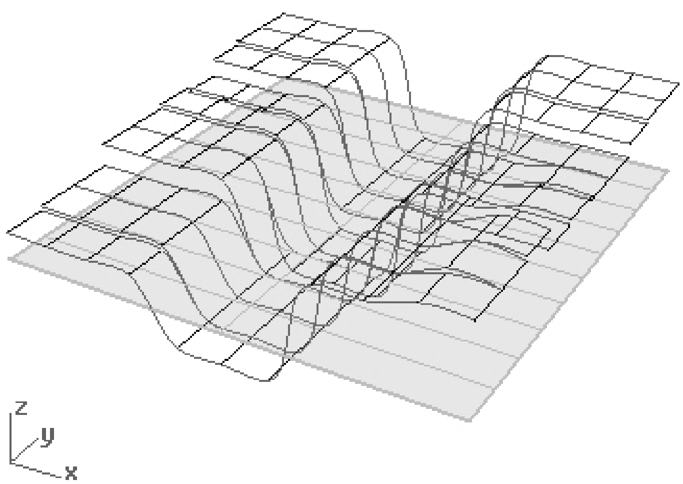
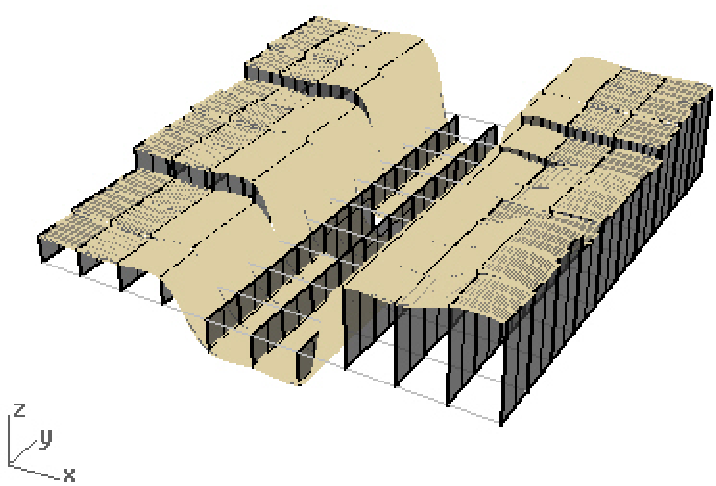
[2D OBJECTS] [TRANSFORMATIONS] [3D OPERATIONS]

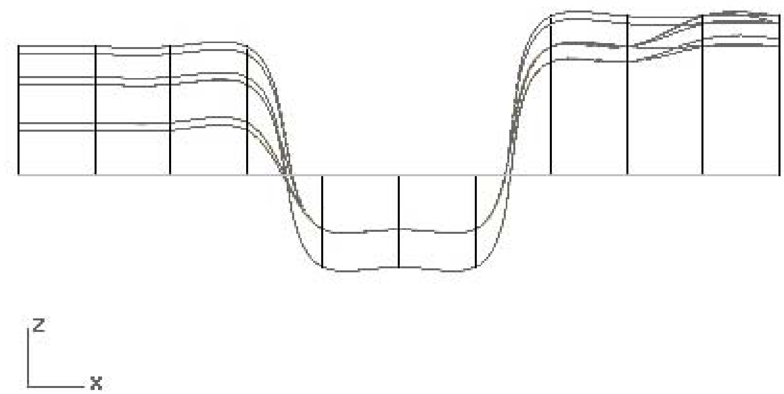
The intent of this study is to transform a very regularized series of intersecting edges into a functional structured landform resembling a canyon or steep valley. The structure of the landform is meant to inform architectural intervention and also imply space within the land itself. This form is not based on a contour map, but rather meant to instruct methods that can be applied to contour maps in other situations.

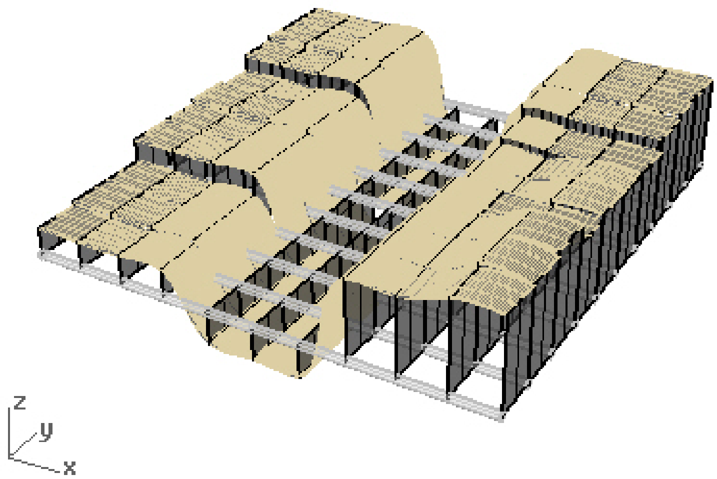
<table>
<tr>
<td>
</td>
<td>Line
Array
>New Layers created, assigned to sets of lines.

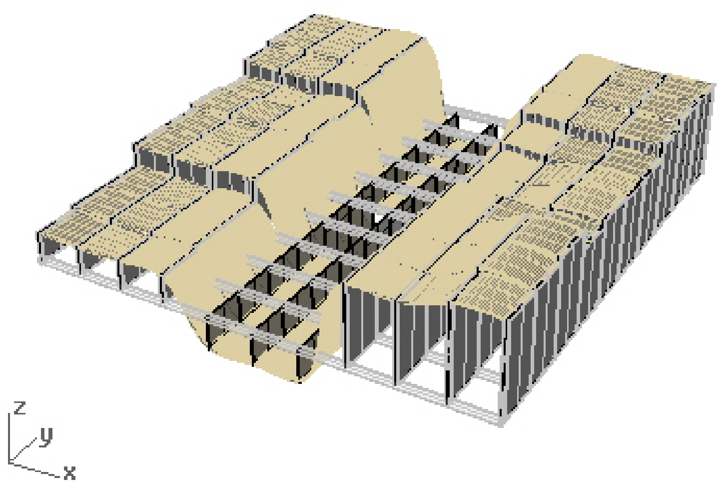
A 10 x 10 "regular grid" is set up in the plan viewport. Layers organize the two sets of line directions.</td>
</tr>
<tr>
<td>
</td>
<td>Split
Move

One direction of grid lines are split by the others and elevated to give the model height.</td>
</tr>
</table>

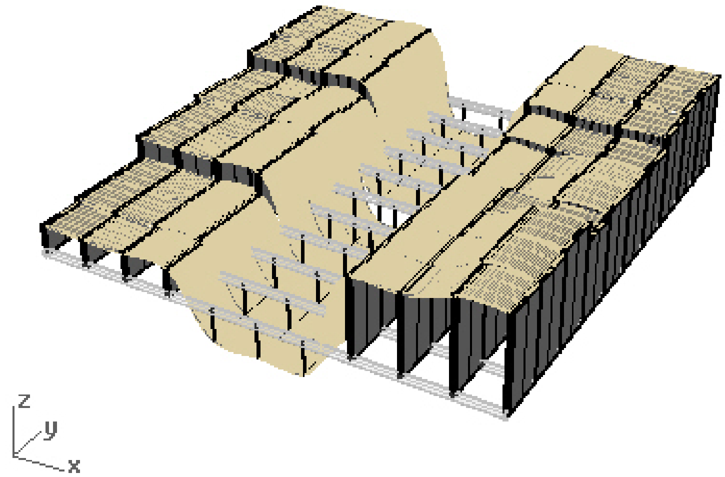
	**InterpCrv** **Copy** *>Toggle Layer control.* The split and elevated grid lines are connected using **InterpCrv**. This type of curve was chosen because it will draw lines through the selected end points rather than using them as reference for control points.
	*>Toggle Layer control.* **SrfPt** The gray layer is turned on to indicate the initial plane of the grid as a possible base for the model.
	**NetworkSrf** **Extrude** Using **NetworkSrf**, each set of curves and the curves that cross them form surfaces that appear as ribbons sweeping from the top of the hills to the valley. The split and elevated grid lines are extruded to the inital grid plane to act as structure.

<table>
<tr><td>
</td><td>>Toggle Side View

Side view with structure incorporated.</td></tr>
<tr><td>
</td><td>Pipe

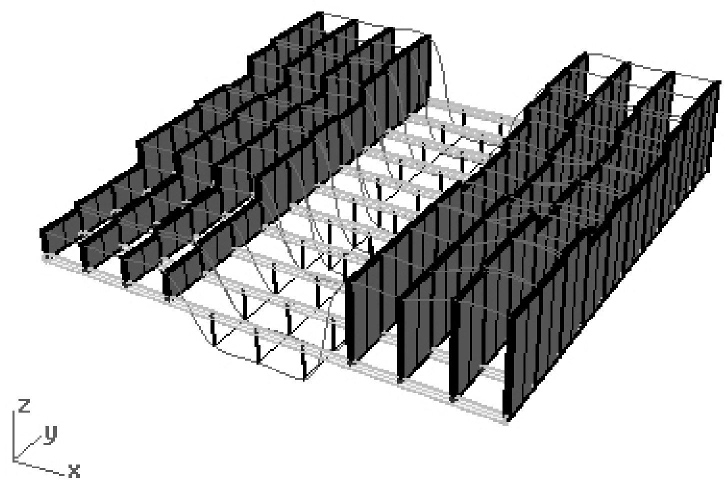
A series of pipes are created around the inital grid lines running parallel to one another. They are envisioned to be the base of the model that structure is attached to.</td></tr>
<tr><td>
</td><td>ExtrudeSrf

The ribs are thickened to be structure for the surfaces.</td></tr>
</table>



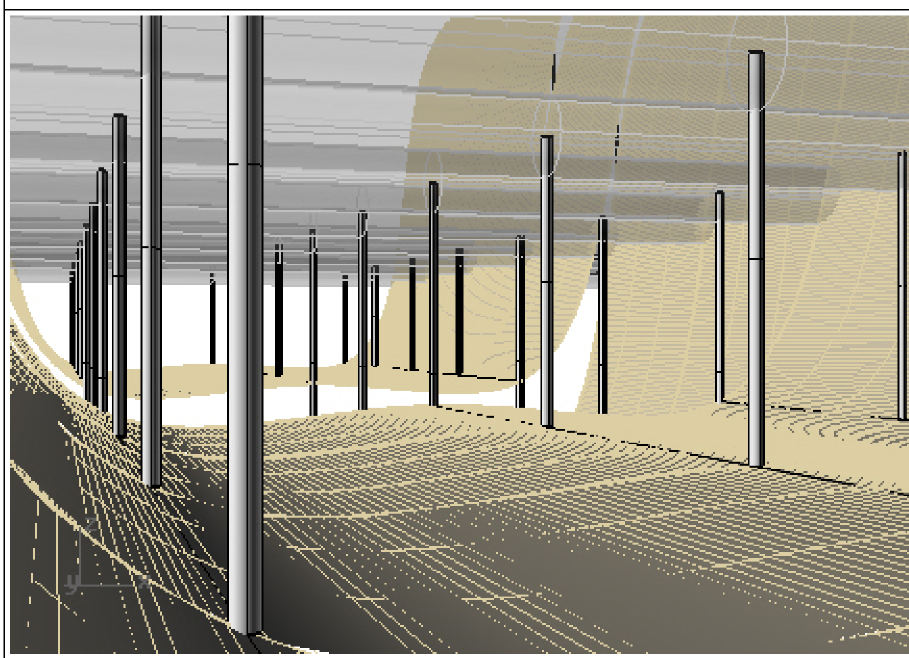
Delete
Pipe

The structural pieces that hang beneath the model base are deleted and turned into a grid of pipes meant to represent wires in tension. This sets up a duality of compression members (above the base beams) and tension members (suspended from the beams.)

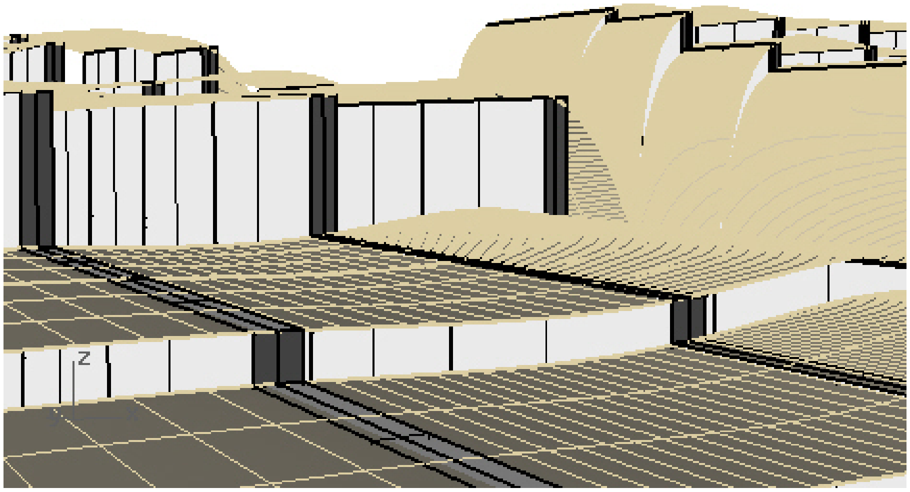
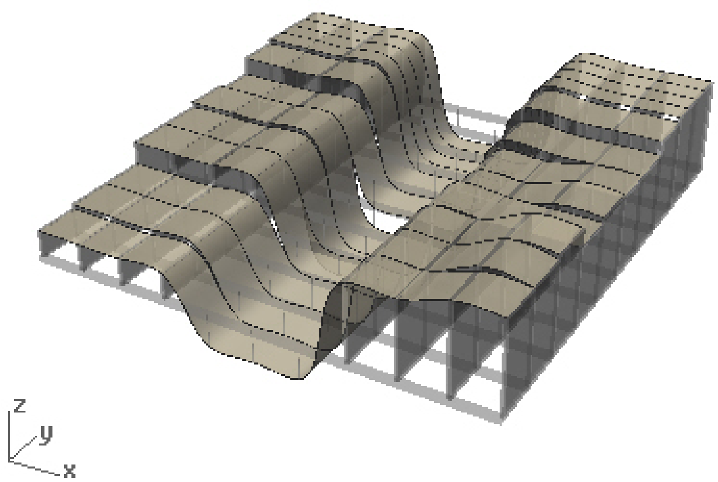


>Toggle Surface layer off.

The surface is turned off to reveal the structural aspects of the model.



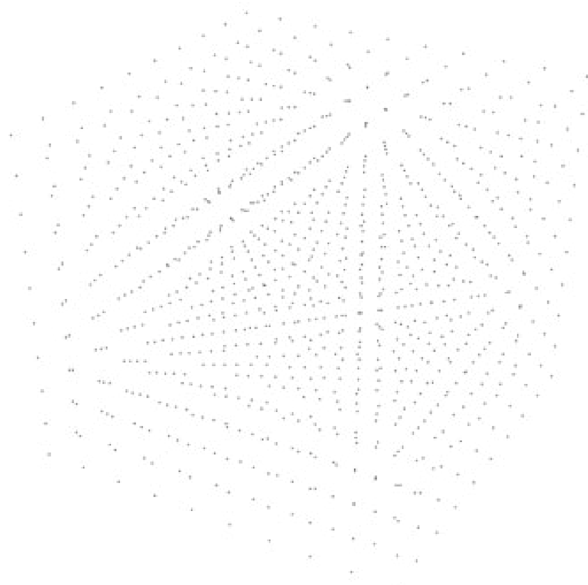
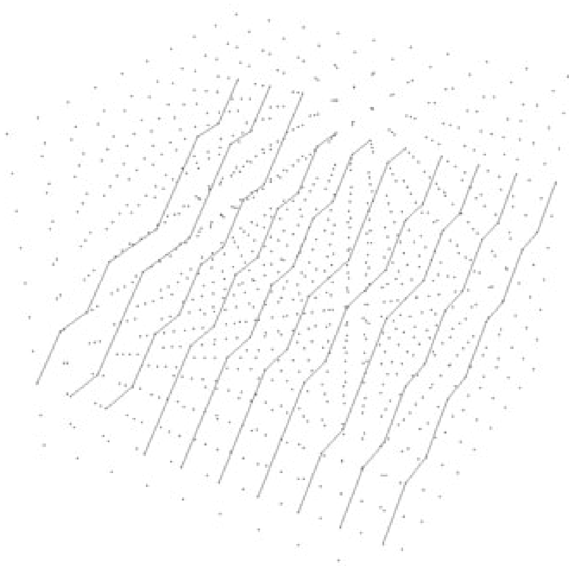
A view of the underground system.

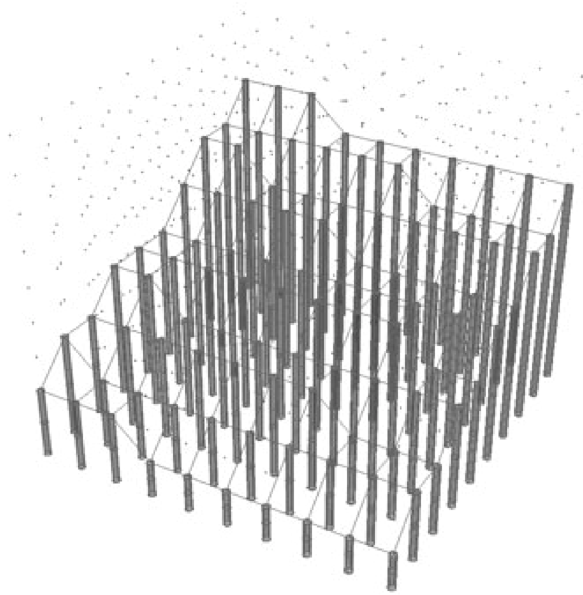
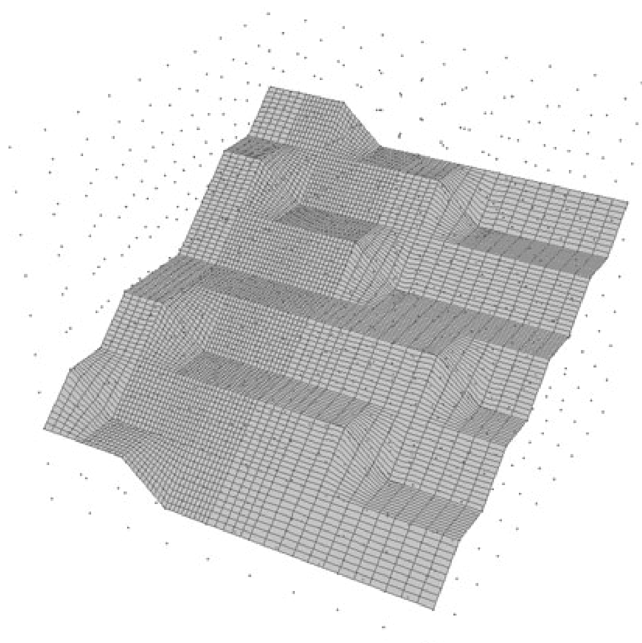
	A view of the structural reveal between surface edges.
	Model with adjusted material properties.

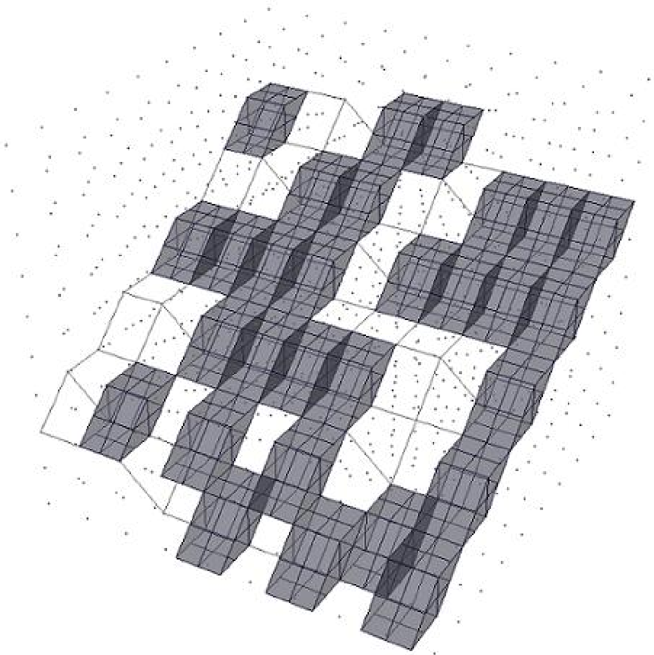
EXAMPLE : POINTCLOUD
Hill Landform From an Array of Points

[2D OBJECTS] [TRANSFORMATIONS] [3D OPERATIONS]

There are unending possibilities for landform when you start with points. Keeping the grid of points (especially in 3 dimensions) visible throughout the process is an easy way to keep the dimensions consistent, which leads to field-making. This particular landform deals with water systems, and all geometries are related to potential application in the physical world.

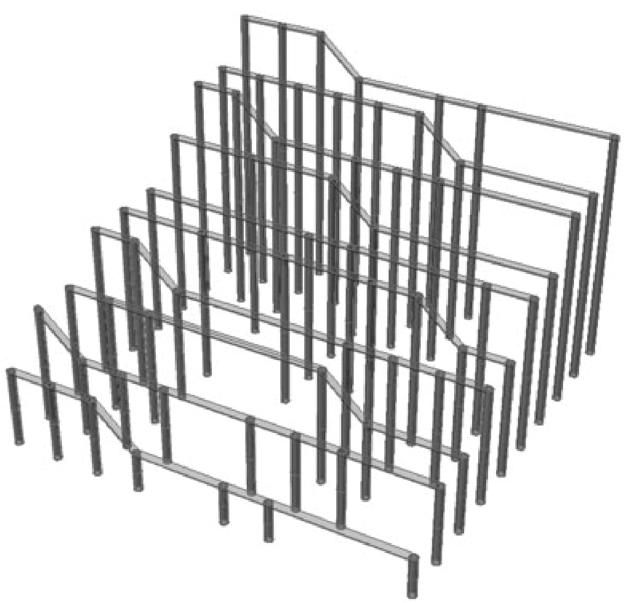
	**PointCloud** **Array** (0 x-direction, 0 y-direction, 10 z-direction) A 10'x10' PointCloud is made with a point every foot. It is then Arrayed 10' in the z-direction, making a 10x10x10 cube of points.
	**PolyLine** **Copy** **EditPtOn** A PolyLine is drawn snapped to the points and then copied 10 times along the cube, implying a surface. The lines are then edited to create a changing slope.

	**PolyLine** **Pipe** **Copy** Polylines are drawn perpendicular to complete the gridded surface. At each intersection a Pipe is drawn, implying a tree-like system of vertical water movement.
	**NetworkSrf** To get more of an idea of the landform, a surface is drawn from the network of curves.

<table>
<tr>
<td></td>
<td>PolyLine
Extrude
Cap
Copy

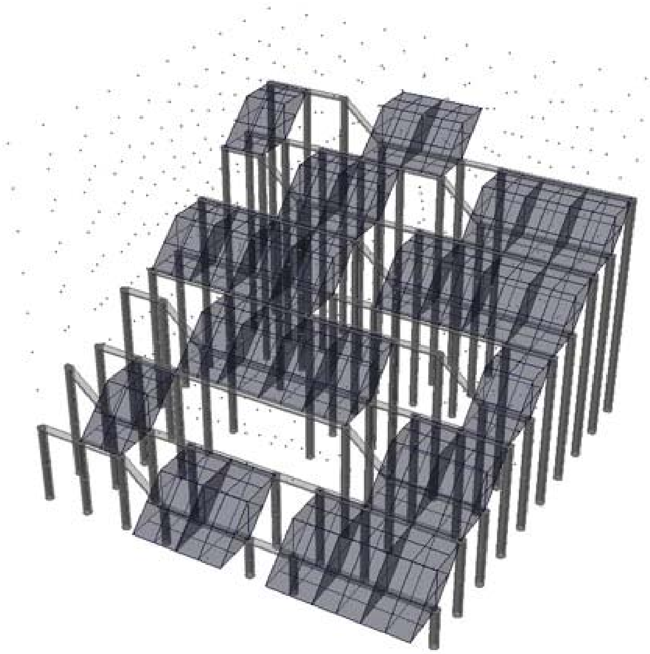
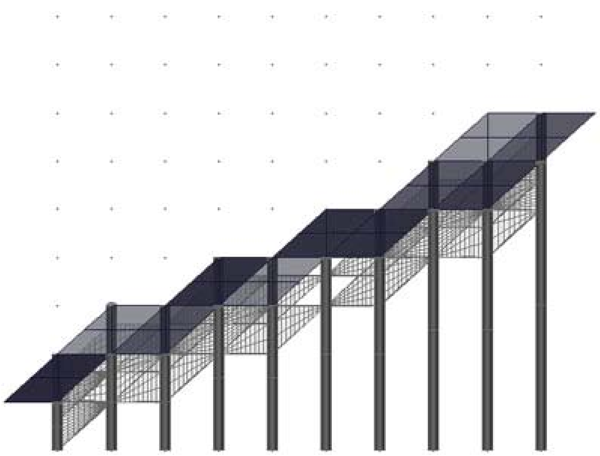
These skewed cube modulars are made from drawing a parallelogram, extruding the polyline, and then capping the geometry.

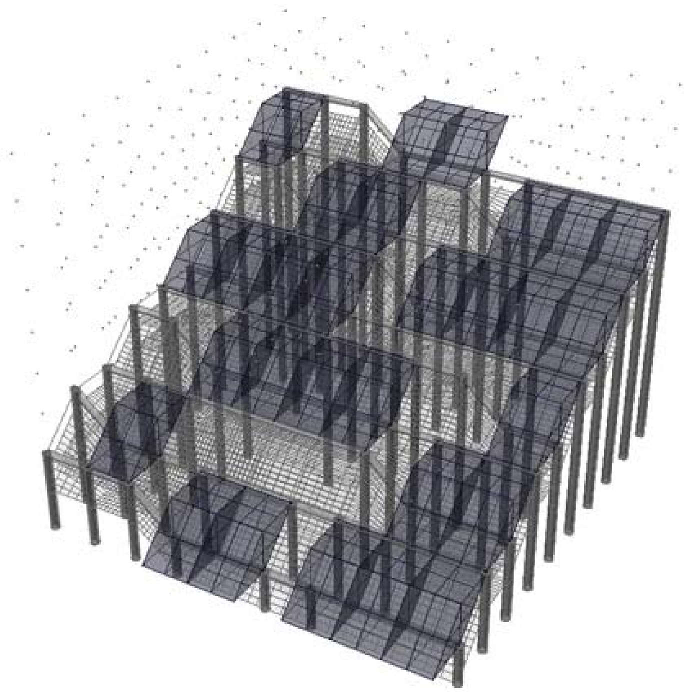
They've been copied to where they fit on the surface, leaving the horizontal changes in slope exposed.

These are thought of as potential impervious water collectors, and so demand more connection to the overall system.</td>
</tr>
<tr>
<td></td>
<td>Pipe
Extrude (bothsides)
Trim

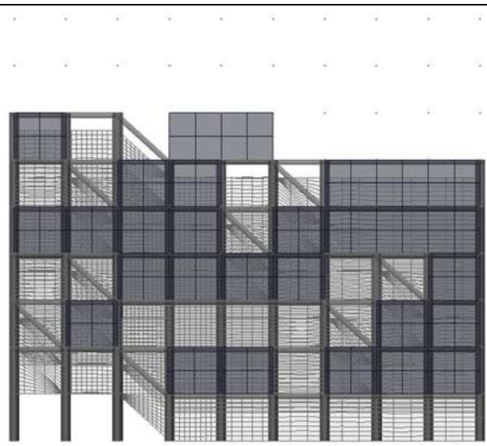
To connect the water collectors and vertical water movers, horizontal pipes are drawn. The pipes will act as trenches, so the top of them needs to be trimmed off. Extruding the horizontal line (on both sides) that the pipes are made from creates a cut plane that changes with the pipe's geometry, and then it is easy to trim.

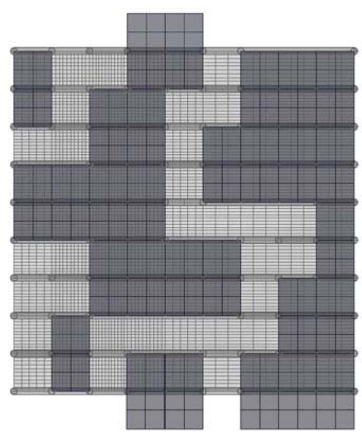
Some vertical pipes are deleted according to where the modulars are; now verticals only exist where they would be structurally necessary. This further describes the volume of water in different locations.</td>
</tr>
</table>

	The pipe and modular system are put together, but the porous surfaces (now empty) need more definition.
	**Copy** **Move** The surface from a network of curves reappears (layer is turned on). Moved down and copied gives the surface depth and the model more structural stability.

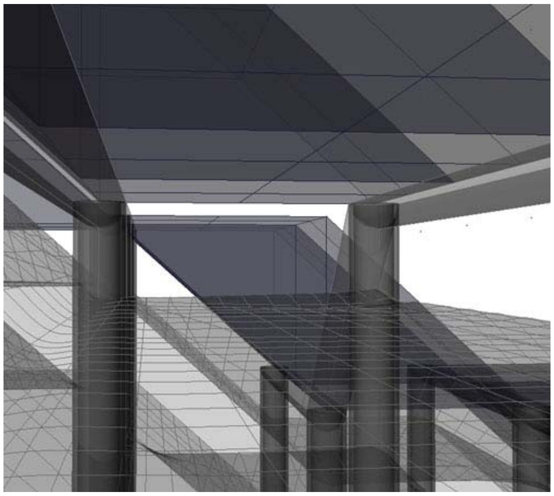
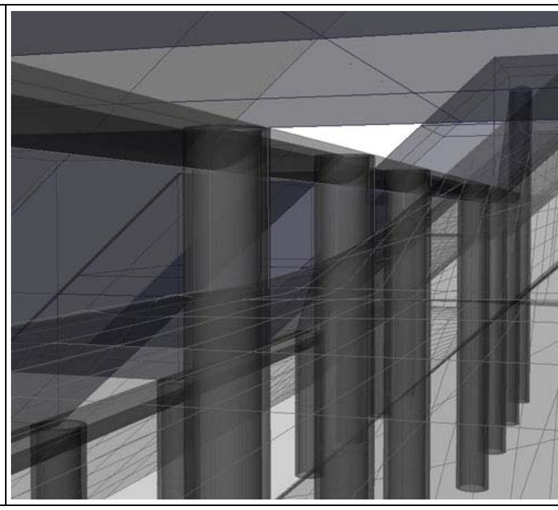


The layers are all turned on. The surface is kept very transparent and is futher implied by keeping the isocurves and edgecurves on for rendering.





The front and side views explain how the water might flow and connect (or fail to flow and connect).

	
Within the model, there is potential for habitation between the layers of surface and modular.	

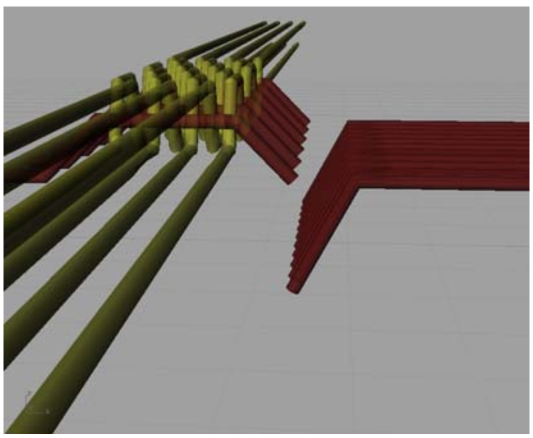
EXAMPLE : PIPES
3D Solids Deconstruction and Reconstruction

[TRANSFORMATIONS] [3D OPERATIONS]

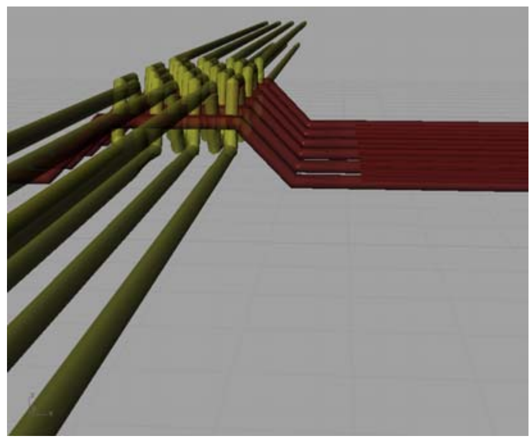
Polysurfaces (**pipe**, **tube**, **box**, etc) are classified as *solids* in Rhino, and operate under their own set of rules within the intelligence of their geometries. Sometimes it is beneficial to keep that intelligence, but sometimes it is better to **explode** those polysurfaces into simpler surfaces, manipulate, and then **join** those surfaces back together. There are many different approaches to this process, but some certainly take longer than others; if you find yourself doing hours of repetitive work, Rhino probably has a smarter way!

Within this example, we want to connect a series of red pipes together.

Given:

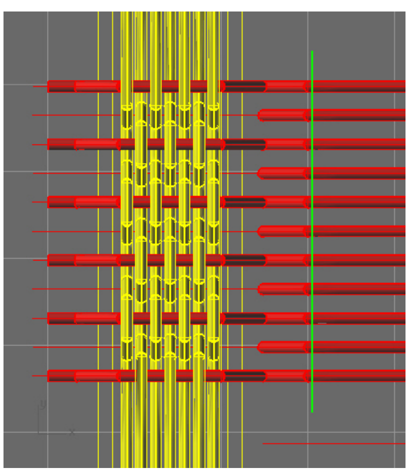


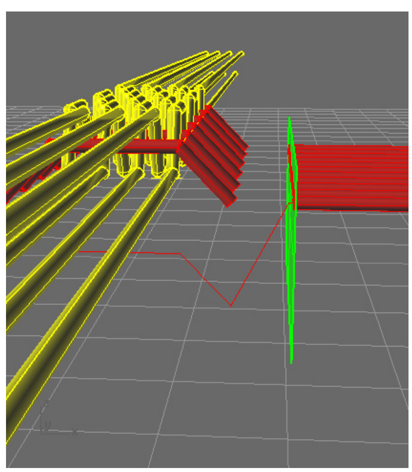
Solution:



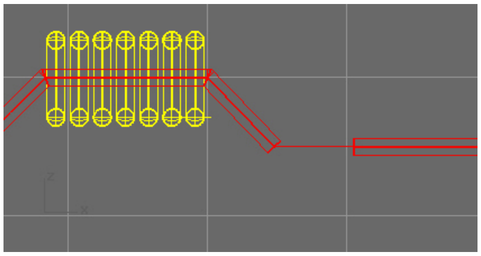
The pipes are polysurfaces created from polylines. By rule they cannot be joined like surfaces, so this procedure requires changing the polyline geometry and then building back up to polysurfaces.

The first step is to draw a **CutPlane**, which draws a surface that encompasses the selected geometries to be trimmed. Drawn in plan, the cutplane is then used to **Trim** the appropriate pipes.

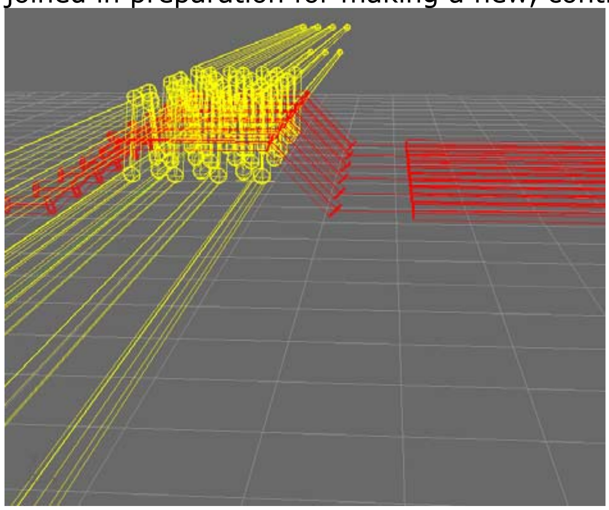




The polyline is also trimmed, and then the cutplane is deleted.



Then a polyline is drawn between the endpoints of the existing polylines on either side of it and joined in preparation for making a new, continuous pipe.

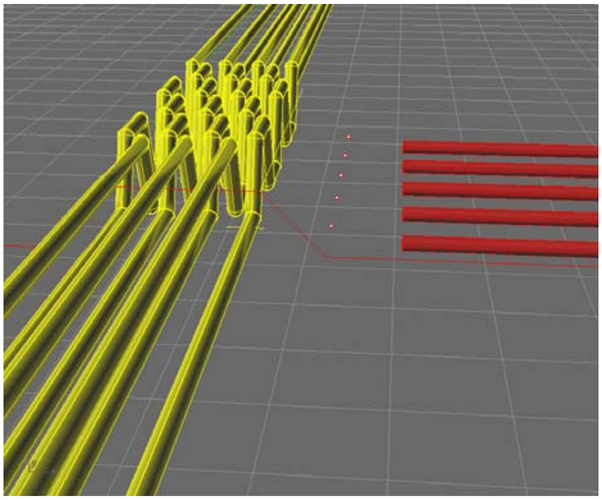


Depending on your preferences and your painting, you may want to copy the base polyline geometry along the series of pipes.

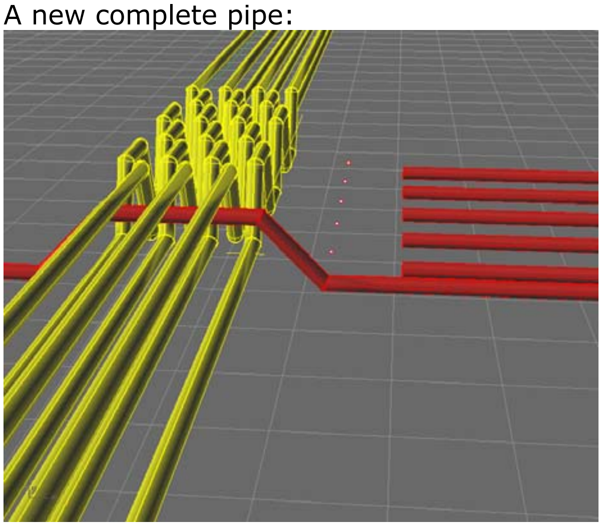
At this point, it is seemingly logical that a new short pipe can be made between the pieces and then joined, but once again, polysurfaces do not allow joining. It also seems logical to explode the polysurface into surfaces, delete the caps of the pipes, and then attempt to join the surfaces into a new polysurface, but this fails in two ways : it is a long, more tedious process, and at certain joints the surfaces cannot be joined and require the creation of new, more complex surfaces. Do not fall into this trap! In the end it is a waste of time and does not create a geometry with built-in intelligence.

The easiest way is to delete the old short pipes completely and replace them with new, complete polysurfaces derived from the new polyline. This gives you clean, intelligent geometry without having to skip around between surfacing and polysurfacing.

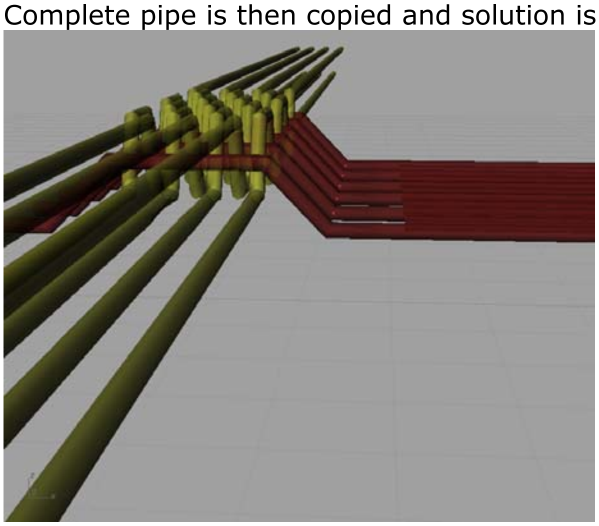
After deletion:



A new complete pipe:



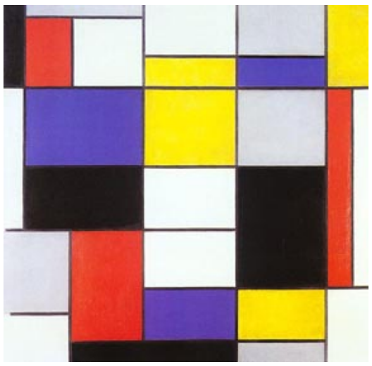
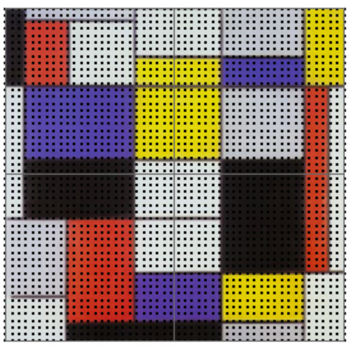
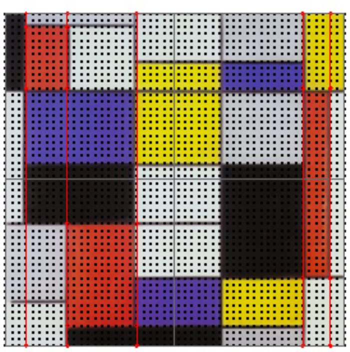
Complete pipe is then copied and solution is found.

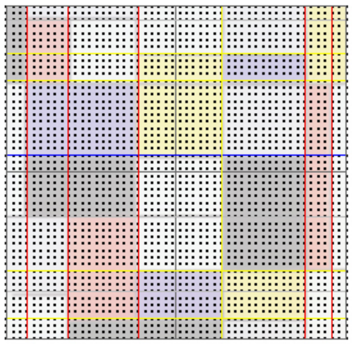


EXAMPLE : EXTRUSION
3D Field from a System of Rules

[IMPORTING] [2D OBJECTS] [3D OPERATIONS] [TRANSFORMATIONS]

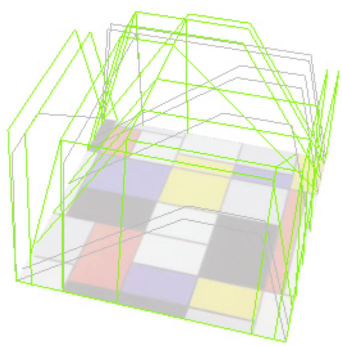
It can be advantageous in field-design to set up a system of rules that drive the manipulation of the model. This set of rules will help you make decisions where the answer is not always apparent, or to break the rules where a wonderful discovery is made. A simple set of rules can create something surprisingly evocative! Furthermore, these simple operational rules can also be understood metaphorically and therefore suggest relationships between your model and ideas of order and structure within ecological, cultural, and even political intentions. While making decisions, then, think about the larger implications that these abstractions might connect to.

	**PictureFrame** Piet Mondrian
	**PointGrid** The **PointGrid** command is used to create a 50x50 grid overtop of the painting. This will serve as a standard guide for the extrusion of the painting. Note, 50x50 was chosen because the scale of the painting is 100' x 100', so the points are spaced at 2'.
	**Polyline** Note: While in the polyline command, if you miss-click a point, you can use the **Undo** command or the Ctrl-Z shortcut to remove the last point. Orthogonal polylines of colors specific to the painting are used to connect the overlaid grid. Points of the polyline are clicked at the corners of each color block in the painting, anchored to the edge of the painting, and the orientation of the line is chosen to be in the direction of the long edge of the rectangle. Two straight lines are drawn along the long edge for each color block, and anywhere the line intersects a block of the same color, a point of the polyline is clicked at that intersection.



>Layer Control
Each color group is set to its own layer, so that while adding each series of color-tuned lines, the others can be toggled off to avoid confusion. Notice the picture frame material setting has been set to a higher transparency so the model is more visible. This is an advantage of the **PictureFrame** command over **Background Bitmap** because the latter cannot be manipulated.

Here comes into play the breaking of rules - or the modification - in the case of a square color block, the figure is given 4 lines - one on each side. It is hoped that later this will give this special case a more prominent figural position within the model.

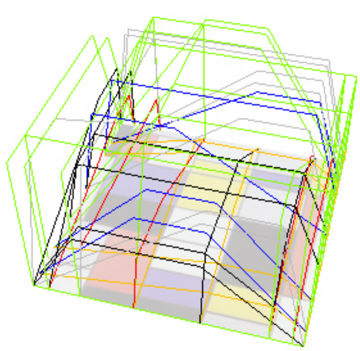


PointsOn
Move

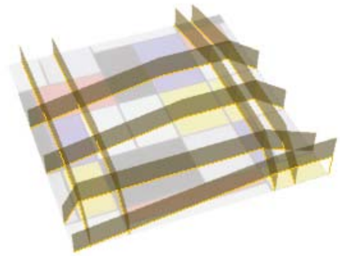
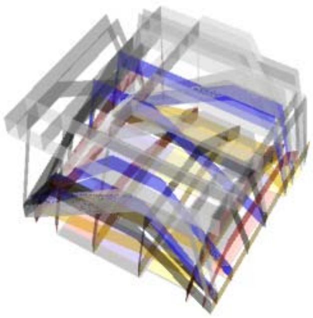
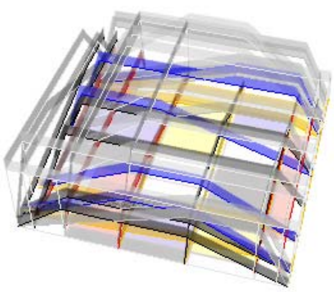
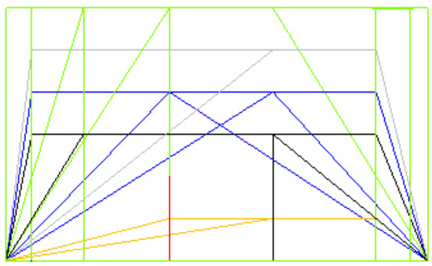
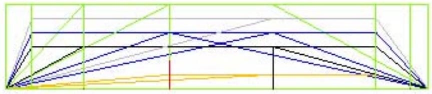
Now, a set of rules for each color is made concerning the elevation of control points. The scale of the painting is kept in mind - 100'x100'. The more intense colors will be elevated from the construction plane less, while the less intense will come forward. (Intensity based on personal judgment - the squinting test.)

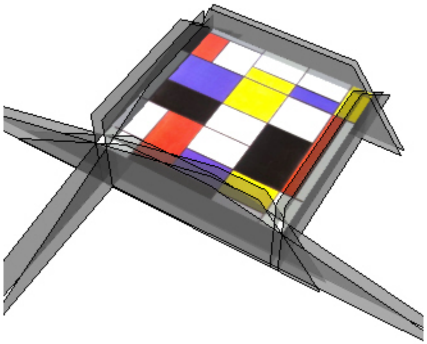
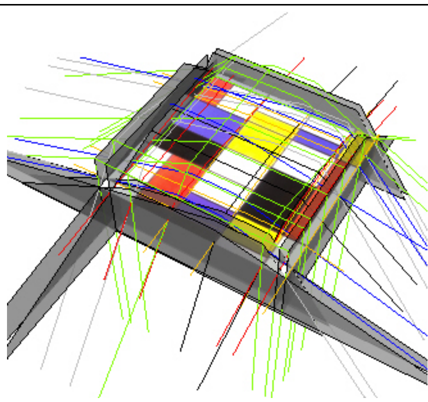
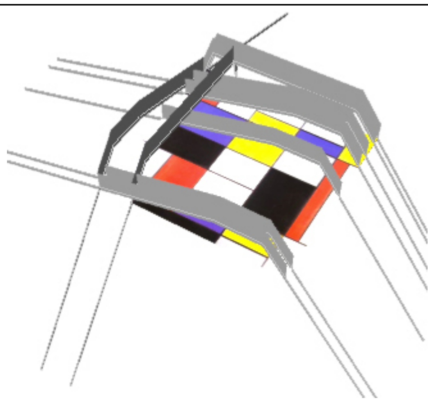
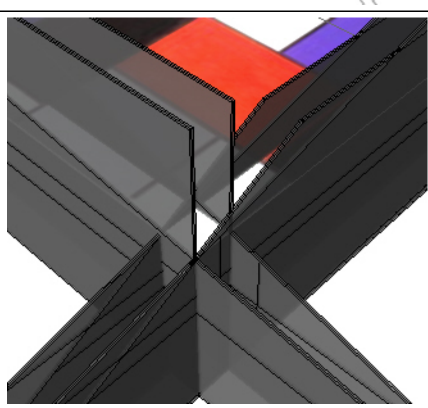
White - 60' (Displayed in Green)
Gray - 50'
Blue - 40'
Black - 30'
Red - 20'
Yellow - 10'

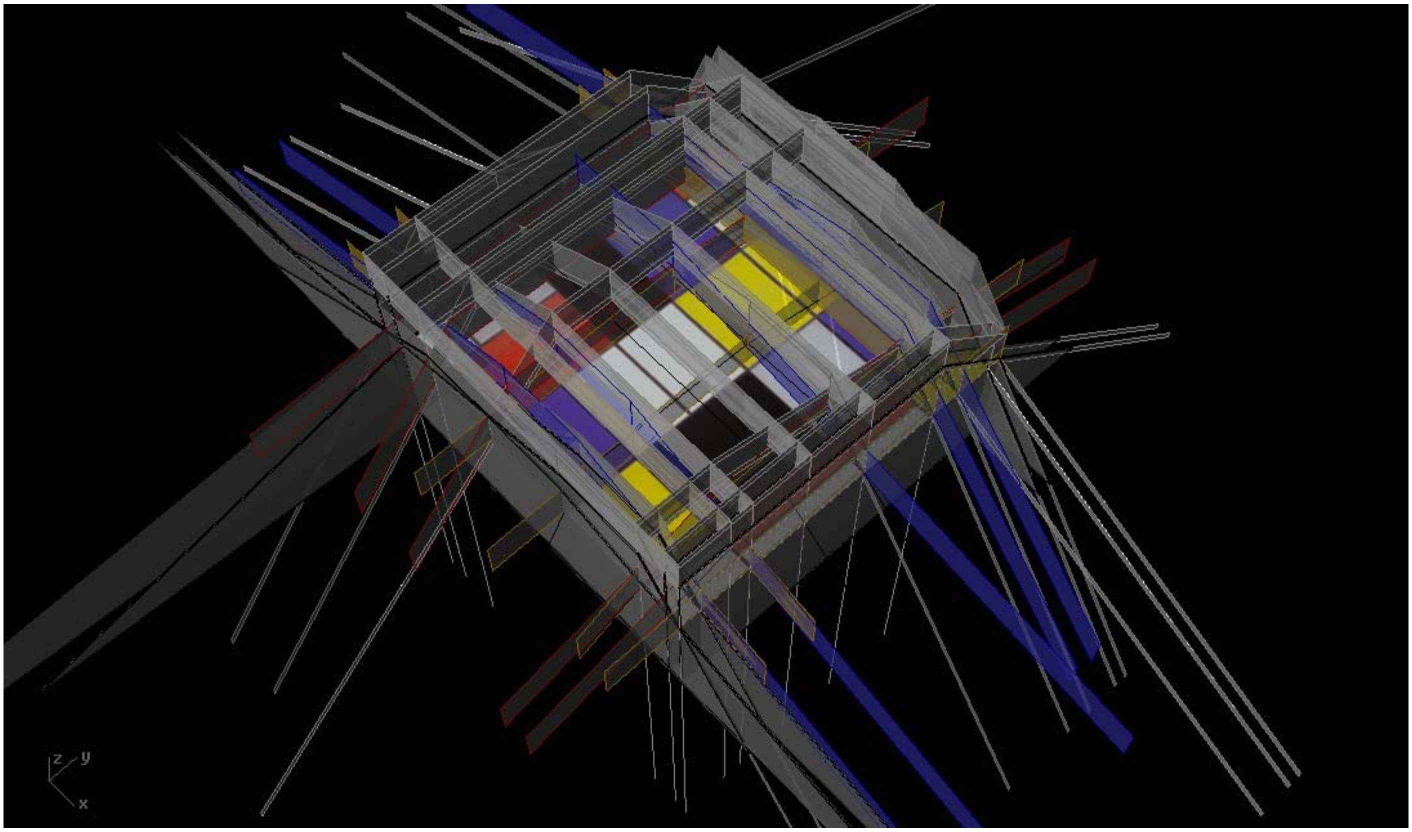
If a block's long edge is starts at the painting's edge, a line is drawn straight up from the point to the appropriate elevation to create a 90 degree angle.



The complete set of elevated color-coded lines.

	**ExtrudeCrv** Each set of color-coded lines is extruded to a thickness of 10'. Note that when extruding a curve, the direction Rhino automatically chooses is not always the direction you need. To change this, while in the command, choose Direction, and then draw a reference line in the direction you need. It is especially helpful to use the Front or Side view to create a reference line if you are trying to extrude vertically.
	The full set of extruded lines.
	**Scale1D** In thinking of this painting as an inhabitable landform, the model seems to be too steep for the scale of its area, or for the opportunity of occupation. **Scale1D** was used to bring the height down. This command allows you to scale an object, or in this case, many objects, in only one axis.
	Before **Scale1D** - Model height is 60'.
	After - Model height is 20', approximately 2 stories.

	**PlanarSrf** **Extend** **Join** Taking another approach to this model, at the point after your critic approaches you with 'just a thought.' This is a good time to save a copy of your model - some screen shots, or another iteration with a new file name. The new approach will attempt to extend the field beyond the edges of Mondrian's painting. The particularly long rectangular blocks along edges of the painting are chosen as placement for structural ribs. These curves are extended and surfaced to serve as a frame for the model. The curves have to be closed and planar (within the same plane) to use **PlanarSrf**.
	**Extend** The color specific lines are extended based, once again, on the initial rules set up. They are extended 2x their initial elevation height.
	**ExtrudeCrv** **Split** The extended curves are extruded once into a thin band, then the curve is split using the **PictureFrame** as a cutting object. Next, the part directly over the painting is extruded into a thicker band. Shown here is only one of the colors.
	**ExtrudeSrf** The rib structures are extruded to a thickened piece that has more potential to be physically realized than a flimsy surface. While the surface object magically stands up on its own in Rhinoland, the real world will have none of this nonsense. Thinking about the structural integrity of your 3D model and choosing material qualities to compliment these thoughts will boost the believability factor of your idea by quite a bit.

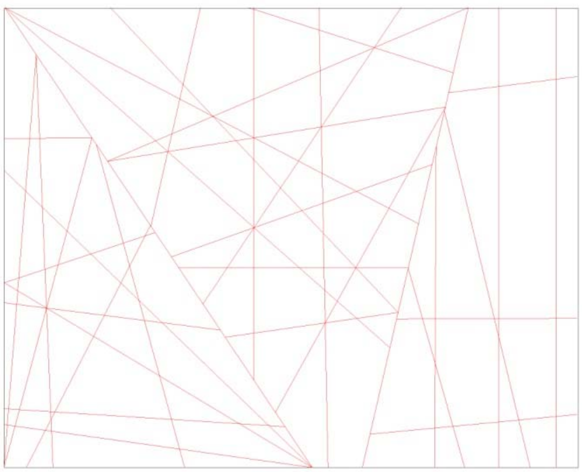
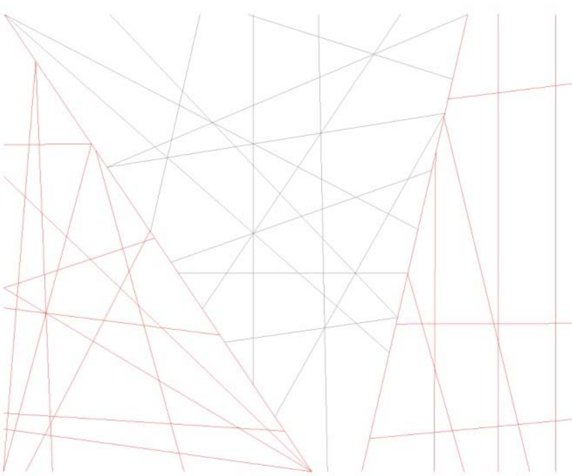


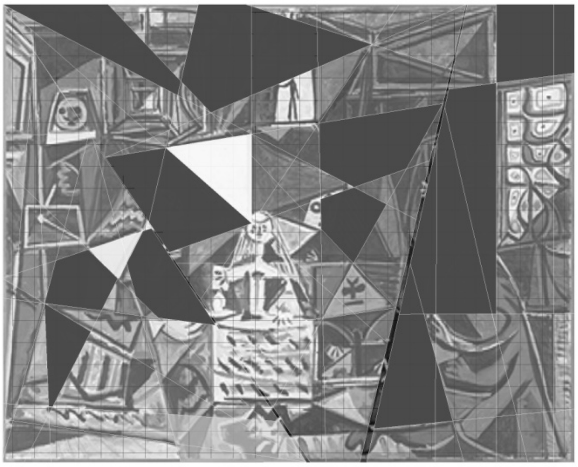
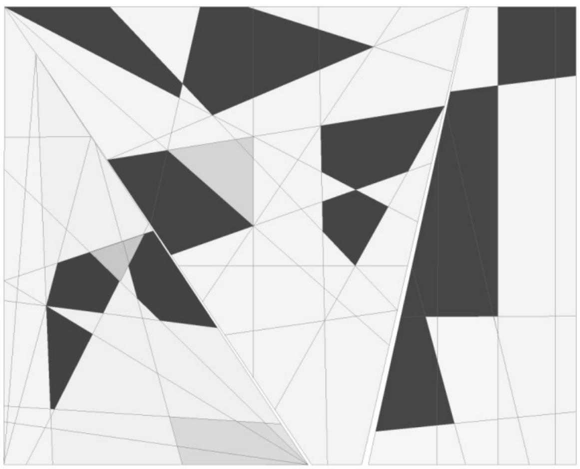
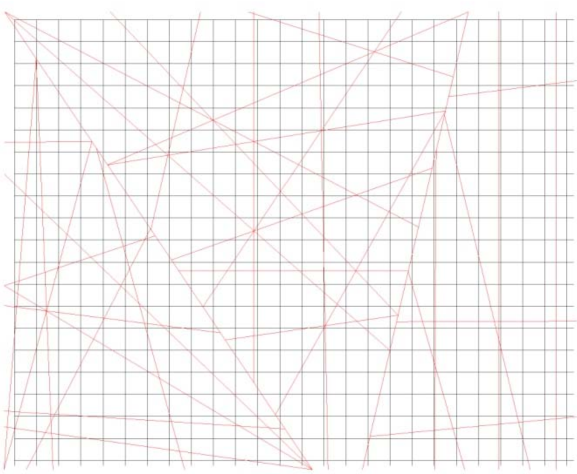
Voila! …I wonder how Piet would feel about this one…

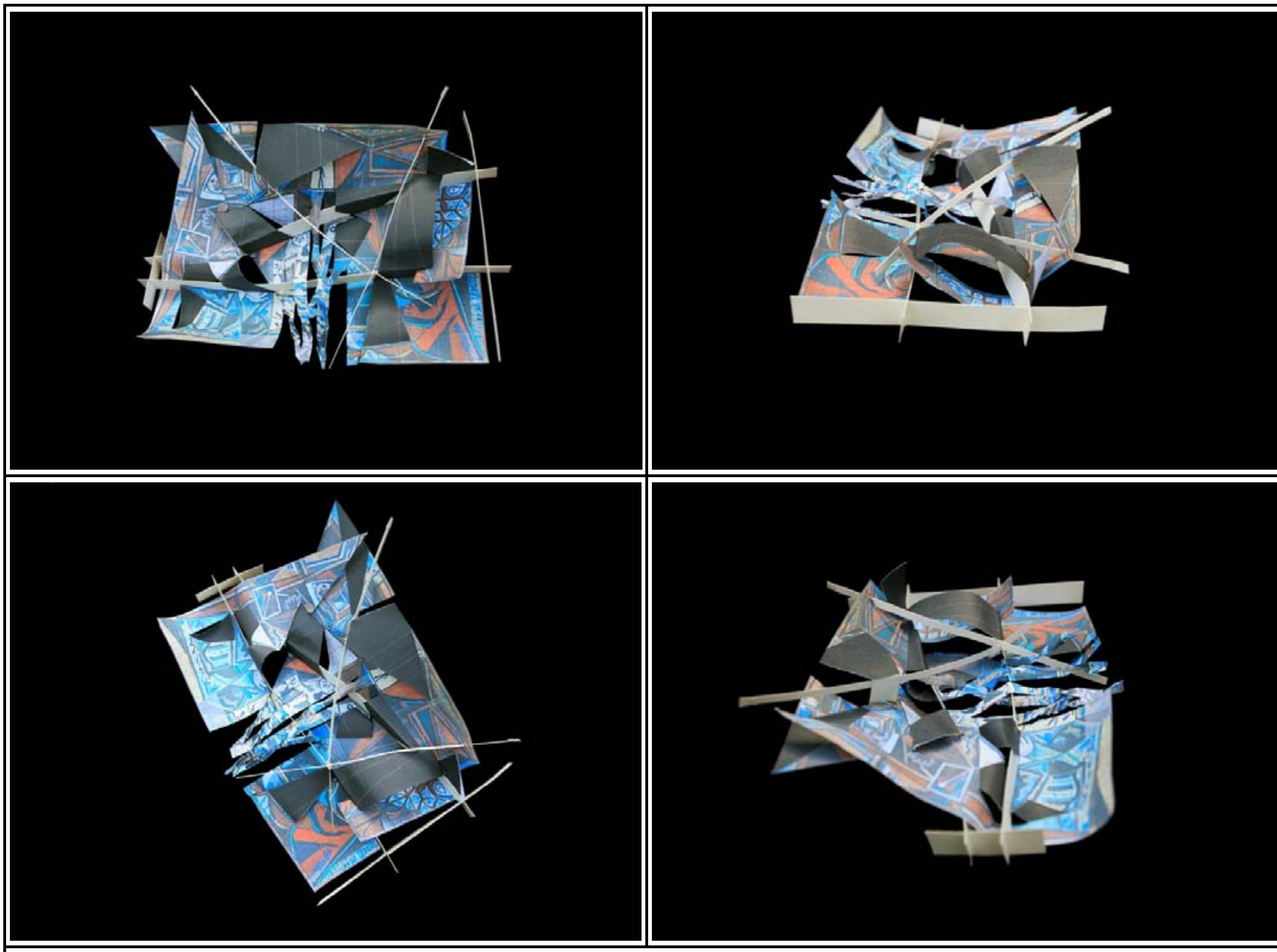
This has been a very simple example in extrusions, brought to you by Piet Mondrian. Some other commands to try: **Ribbon, Fin, Slab, ExtrudeCrvToPoint, ExtrudeSrfToPoint, ExtrudeCrvAlongCrv, Slab**... and many more! Use your imagination!

EXAMPLE : PHYSICAL MODEL
From Painting to Landform

[IMPORT][2D OBJECTS] [PRINT AND FABRICATE]

	Start with a Picasso.
	A field of lines is derived from the triangular geometries of the painting.
	A direction is specified within the field of triangular geometries based on the movement within the painting.

	Clues within the painting allow a series of surfaces to shade and present depth using the field as a guide.
	Implied depth of painting using triangular field.
	An orthogonal grid is laid over the triangular field



The 2D field is printed and manipulated – cut, woven, notched, folded, crumpled.

The painting becomes landform - a gently rolling valley where water is converging to a rushing stream. The landform holds a sturdy primary structure that is woven below and above ground, revealing itself at the surface to describe its formation and hierarchy. A secondary structure bridges the watery divide to bring together a neighborhood of habitations – pockets extruded from the surface and trenches folded under.

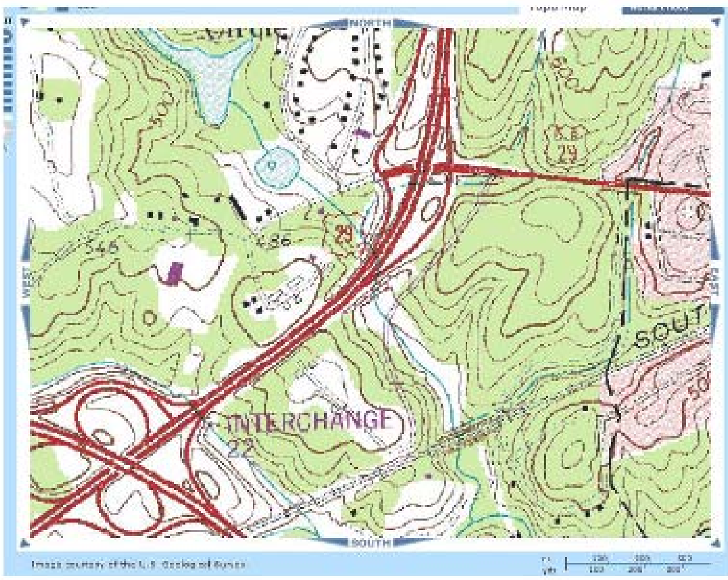
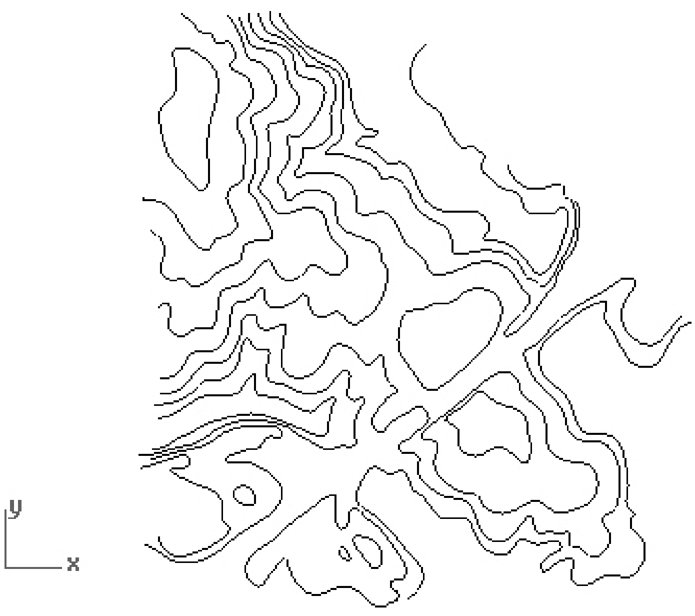
Dwelling is an act inherent to the structure of this landform.

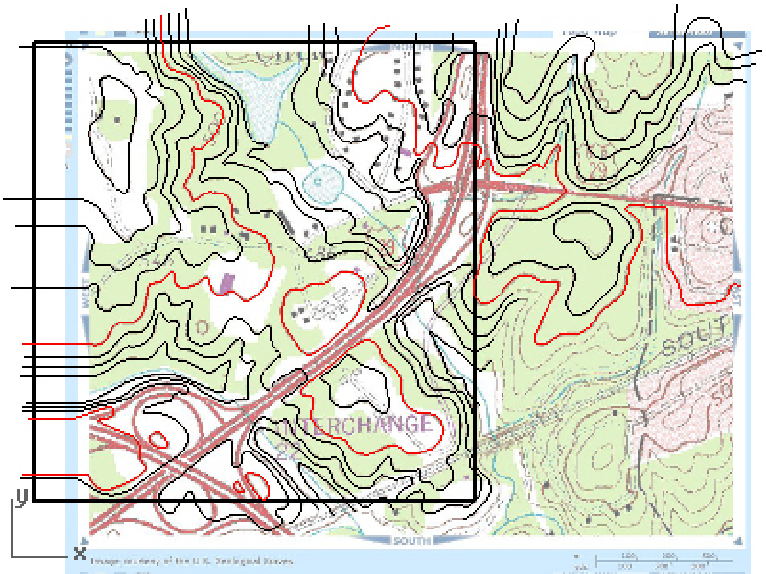
EXAMPLE : PANCAKE LANDFORM
Setting Up and Using Imported Contours

[IMPORTING] [2D OBJECTS] [TRANSFORMATION]

The "Pancake Method" of modeling landform is one that is used often and achieved in various ways. However, this method often leads to unresolved relationships between building and ground, and can be a big problem for both you and your critics. Presented here is one way of achieving this effect.

/map provided by www.terraserver-us.com, all images from view capture, no rendering required.

	Contour Map inserted as a picture frame. Note that the scale key is intact, so the map may be scaled correctly.
	**InterpCrv** Contour lines are traced and therefore digitized.

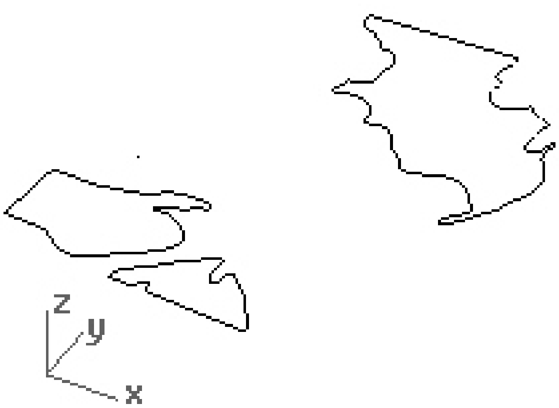


Trim

Something to trim about. A rectangle is extruded around the contour lines to trim their extents. The surface must intersect the curves in order to serve as a 'cutting object.'

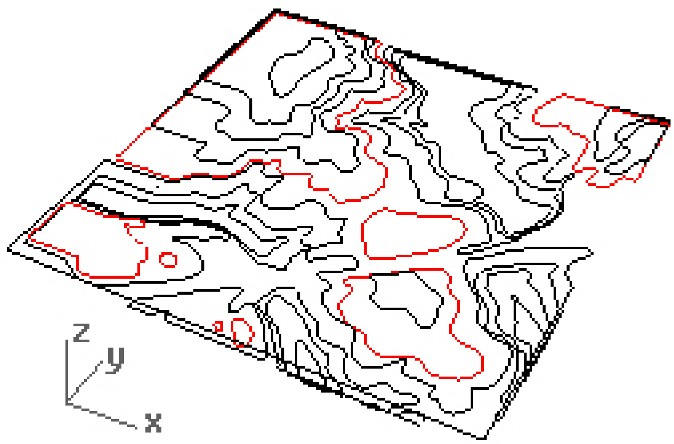
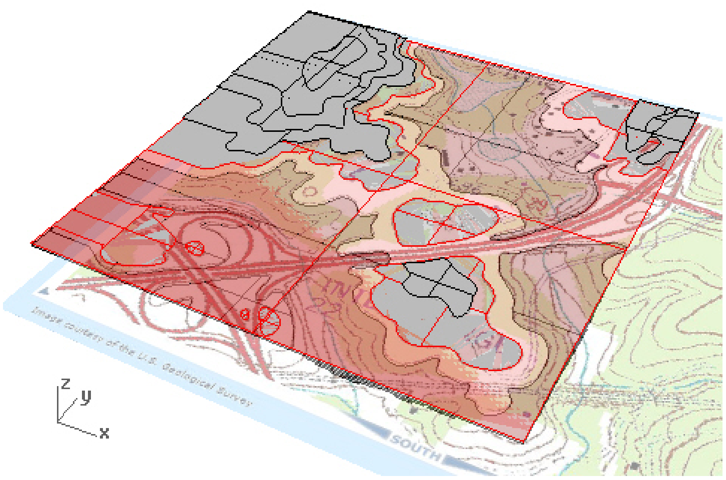
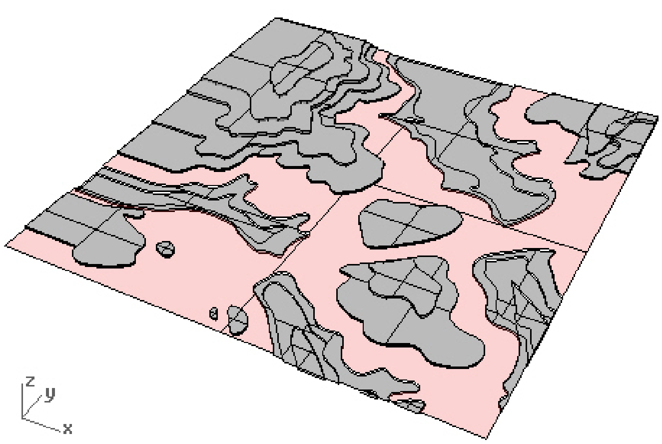
Project
Move

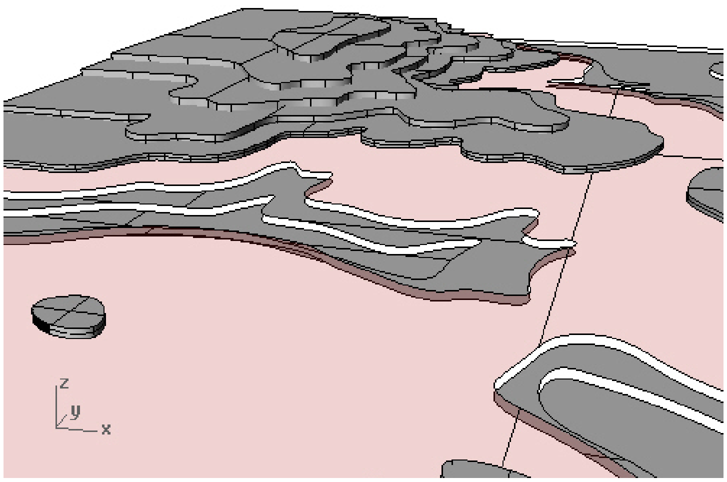
A surface is drawn at the 'zero plane' and contour lines are elevated from this plane. Usually the zero plane will serve as an idea of the model's base, but it can be imagined in different ways. In this case, the **Project** command was used. First the surface plane was elevated then the contours were projected to it at each appropriate level. This was a way of maintaining the original contours in one plane.



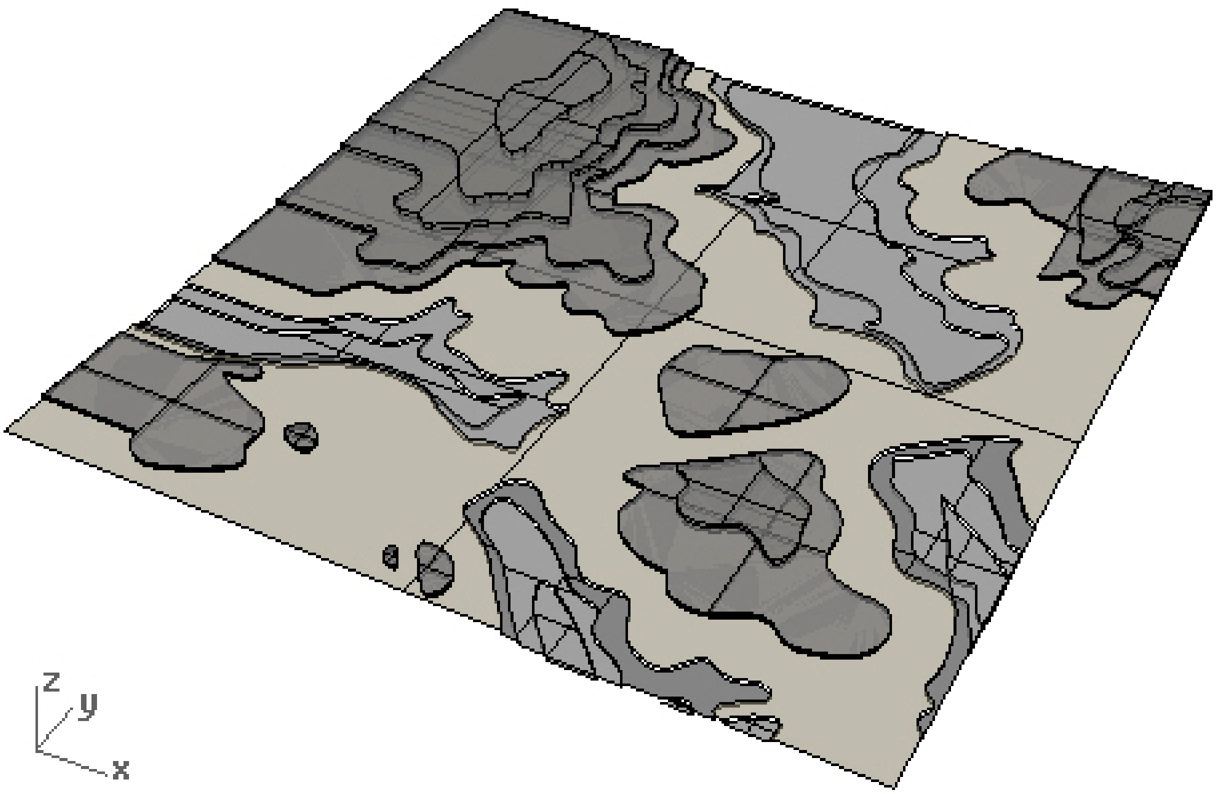
>Layer Control

A set of contour lines at one elevation. Notice how the curves have been closed so that each contour, where it meets the surface edge (see above) becomes a closed figure. Each level of elevation (indicated by the map as 20' contours) is given its own layer so they can be easily located, selected, and elevated.

	The completed set of closed curve contours.
	**PlanarSrf** Each contour level is surfaced in order to make their height more apparent. The model base, or 'zero plane' is indicated in red. Note that it is between layers that are elevated and pushed down.
	**ExtrudeSrf** **Split** The contour levels where the elevation gets taller than the reference plane are extruded. The lower levels are left as thin surfaces that are visible through cuts created using **Split** in the model base.



A closer look at the model reveals the thickened hills and the thin recessed layers.



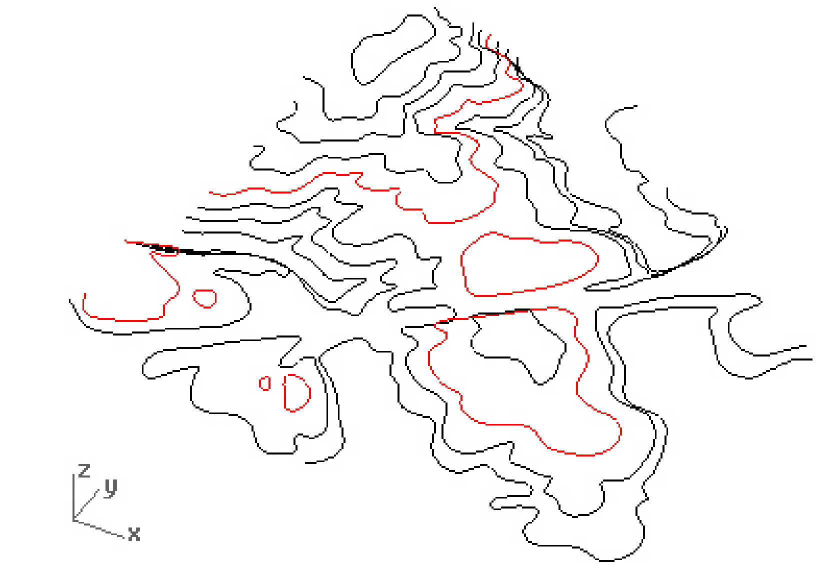
A final view of the model shows material properties that have been adjusted for the model's performance and to provide hints of depth in the view.

The pancake mode of landform modeling can be quite difficult. It is only one way to model landform, and in its most standard form, it is not conducive to locking the architecture into the structure of the land. However, it can be one component of a more complex and intriguing inhabitable landform model.

EXAMPLE : RIBBONS
Landform From a Contour Map Using Ribs

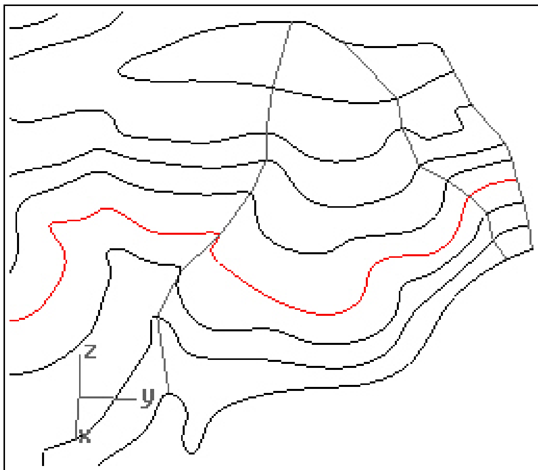
[2D OBJECTS] [3D OPERATIONS] [TRANSFORMATIONS]

/map provided by www.terraserver-us.com



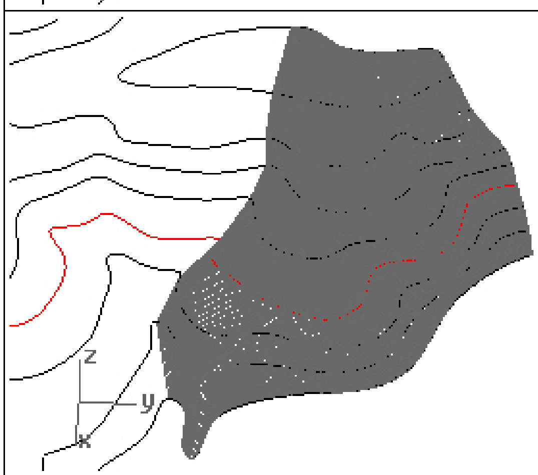
>File>SaveAs

Digitized contours taken from map used in Pancake method and saved as another iteration. Iterations of a model are an important part of documenting your work, and will come in handy when setting up diagrams of your ideas!



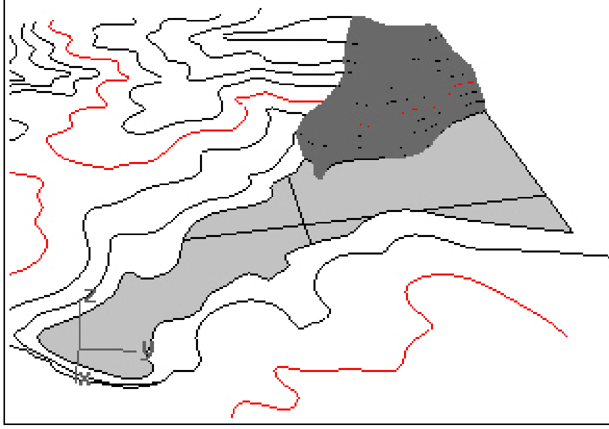
Line

A portion of the contour map, in perspective. The perpendicular object snap was used to create the guide lines through contours for the network surface. Contour curves are divided at inflection points in order to create an accurate and smooth surface from a network of curves.



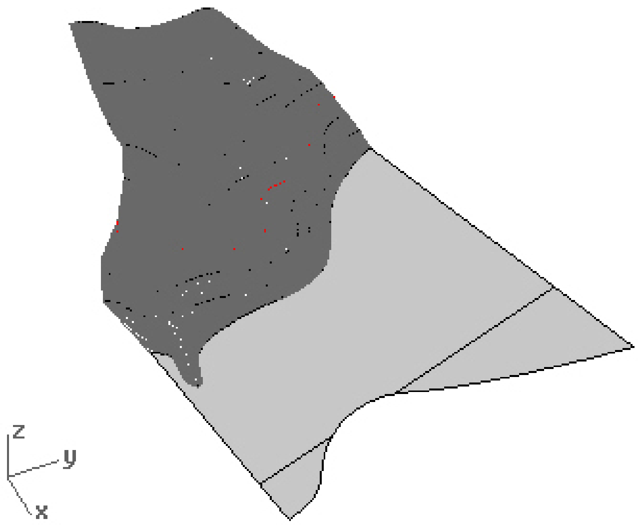
NetworkSrf

A network surface is created from the contours and guide lines. It is important to split the contours at the edges of the desired surface prior to creating the network surface. Curves must be continuous and intersect. If the surface isn't turning out correctly, the problem could be the complexity of curvature between perpendiculars, and you may need to add more of these guide lines.



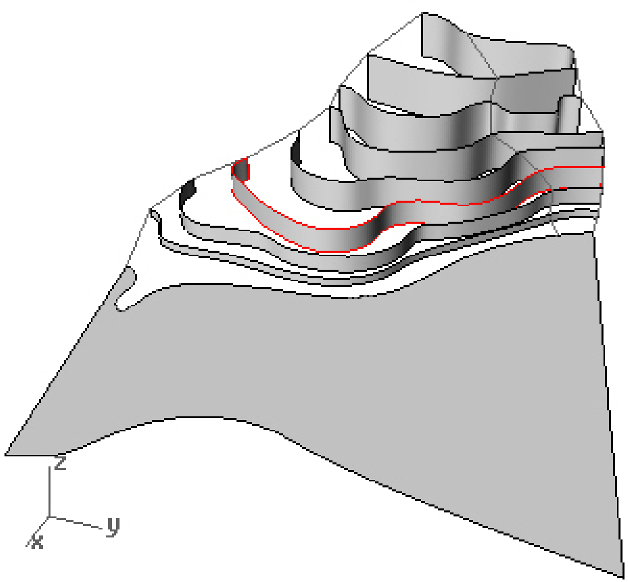
PlanarSrf
>Planar

The lowest topography line is used to represent a water's edge, perhaps a cove. The contour line is closed and joined at the right edge, and then **PlanarSrf** is used to create a surface of this geometry. To use this command, it is necessary that the curve is closed and planar. >*Planar* mode can be toggled to aid in this process.



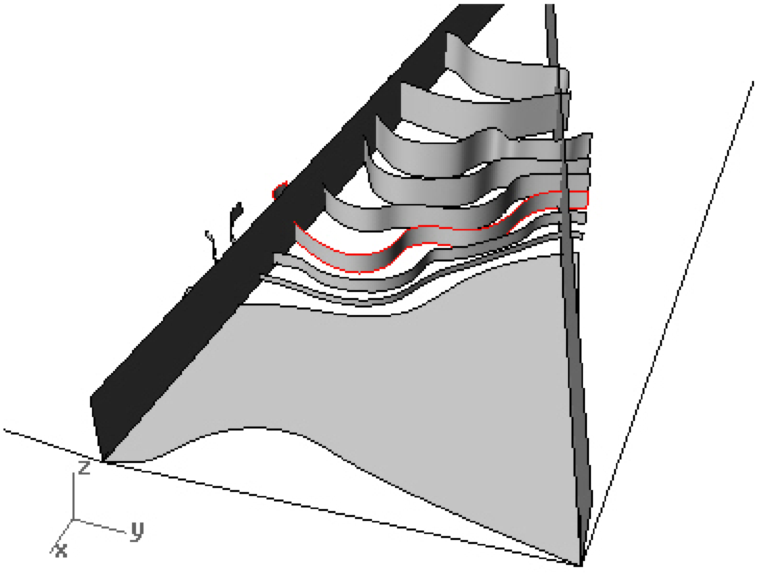
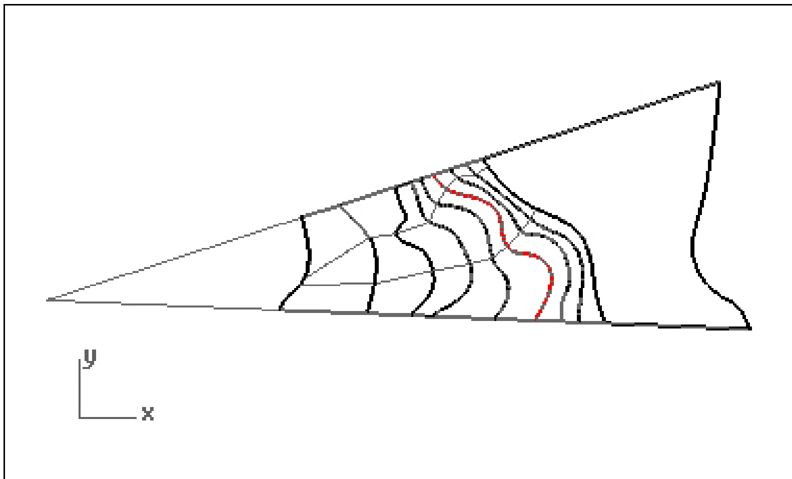
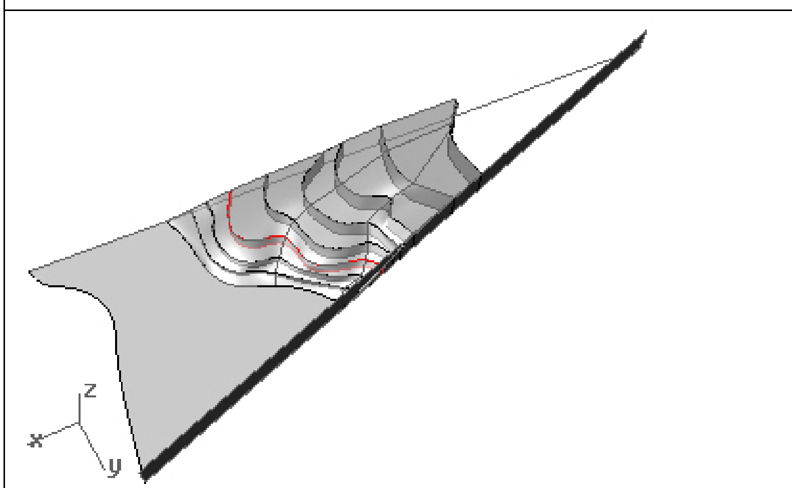
Split
>Layer Control

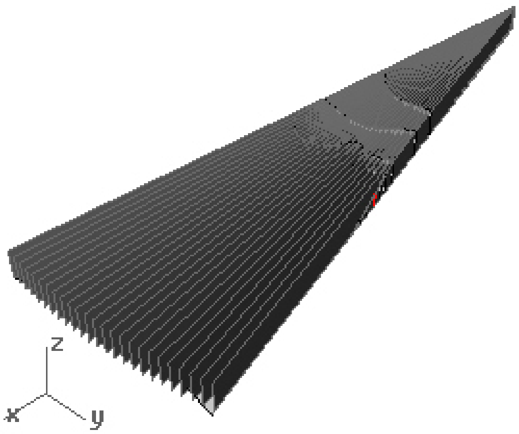
The contour lines outside the designated model boundaries are split from the edge of the **NetworkSrf**, and moved to a hidden layer to save for later reference.



ExtrudeCrv

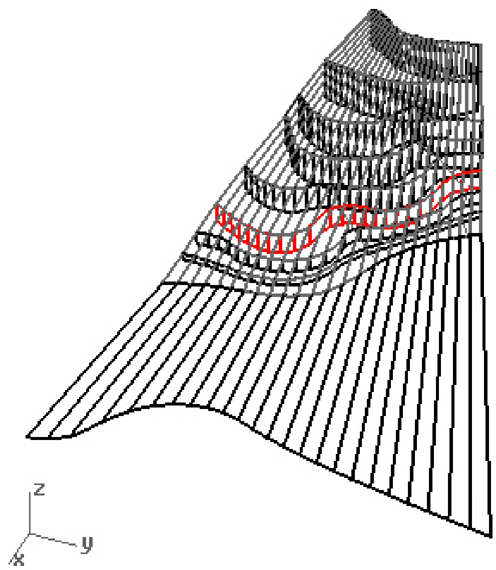
The surface layer is toggled off in order to reveal the underlying curvature, which is then extruded to provide structure. Each contour line is extruded individually in order to reach the same "zero point" which is the water level. This serves as a base for the model, which is necessary to imagine the model as a built form.

	**Line** **Extend** **ExtrudeCrv** The edges of the model are cleaned up to create a distinct edge. (This is not a necessary idea to a digital model. Sometimes it is important the model implies extension outward, to infinity, especially in landscape design.) Lines are drawn at the angle of the water plane and extended to meet one another. These lines are trimmed to form a point and then extruded.
	**Trim** The cleaned up model edges resembles a wedge of pizza. The trim command was used to select the extruded edges and trim the contour ribs.
	Model prior to contour ribs with surface visible.



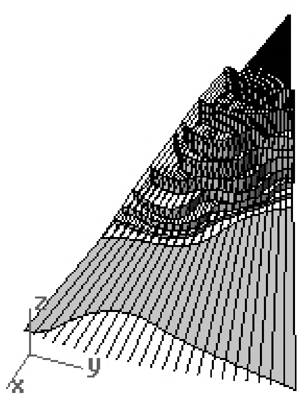
ArrayPolar

A polar array of surfaces is created using one of the extruded edges, anchored at the intersection point mentioned earlier. This creates a series of cut planes through the entire model.

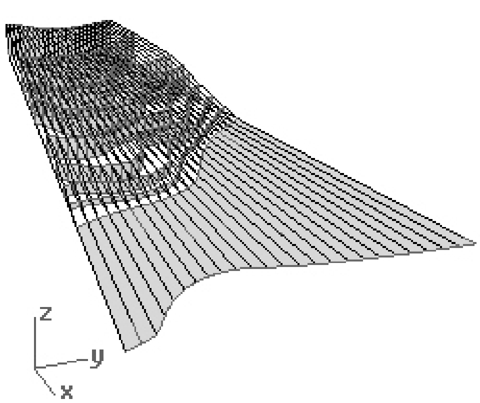


Split
Delete

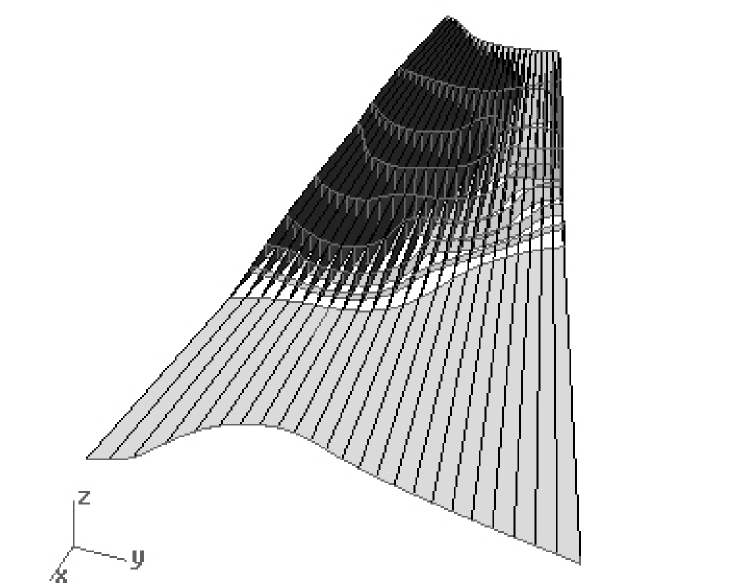
The cut planes are used to split the contour ribs and network surface. However, the complexity of the surface did not permit Rhino to operate in shade mode, and the file got several error messages regarding memory space. A solution had to be decided.



A series of lines were drawn in order to imply the complex surface. Note the rotation axis of the polar array behind the model.

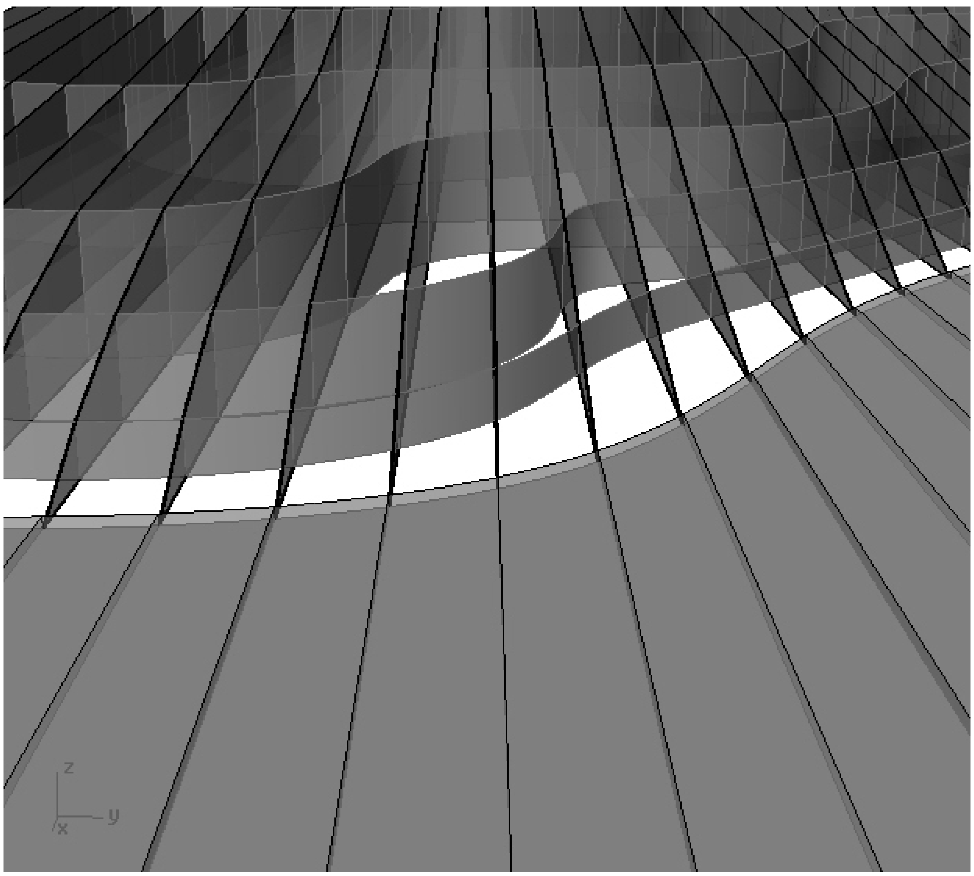


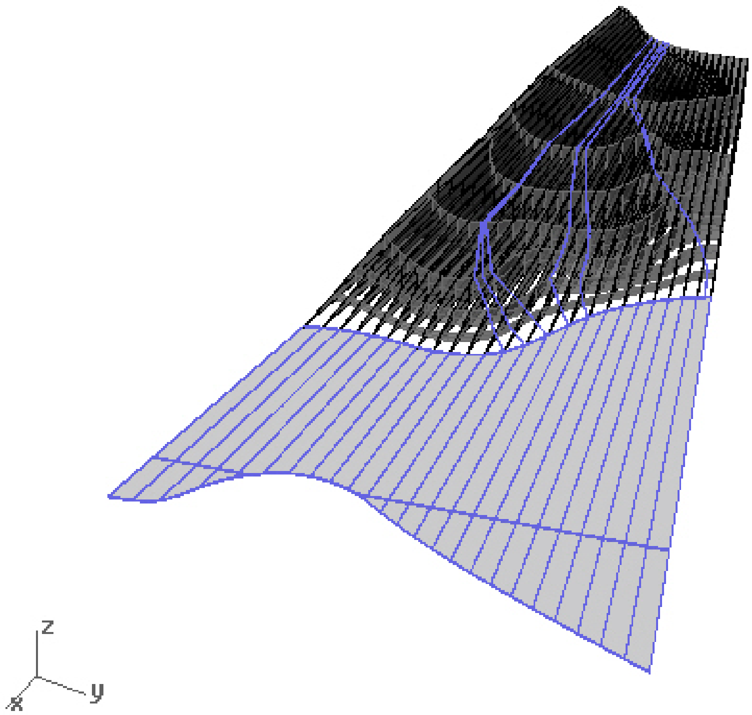
The completed and trimmed implied surface.



PlanarSrf

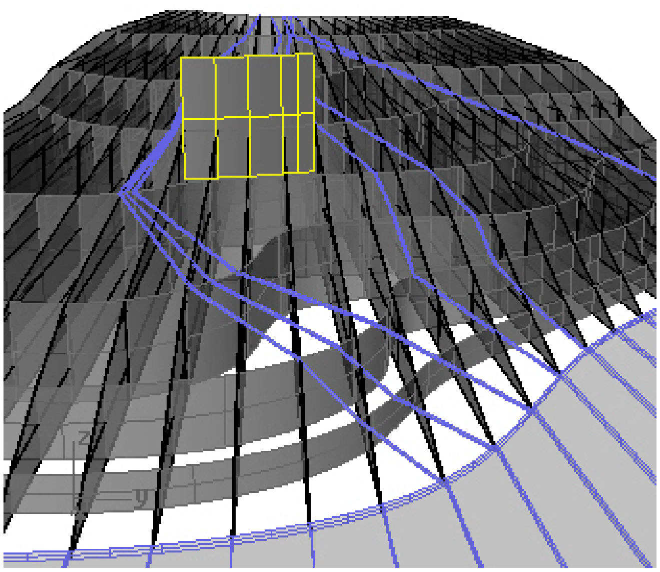
The implied surface was transformed into a series of structural ribs intersecting the system of contour ribs. This is seen as the secondary structure.





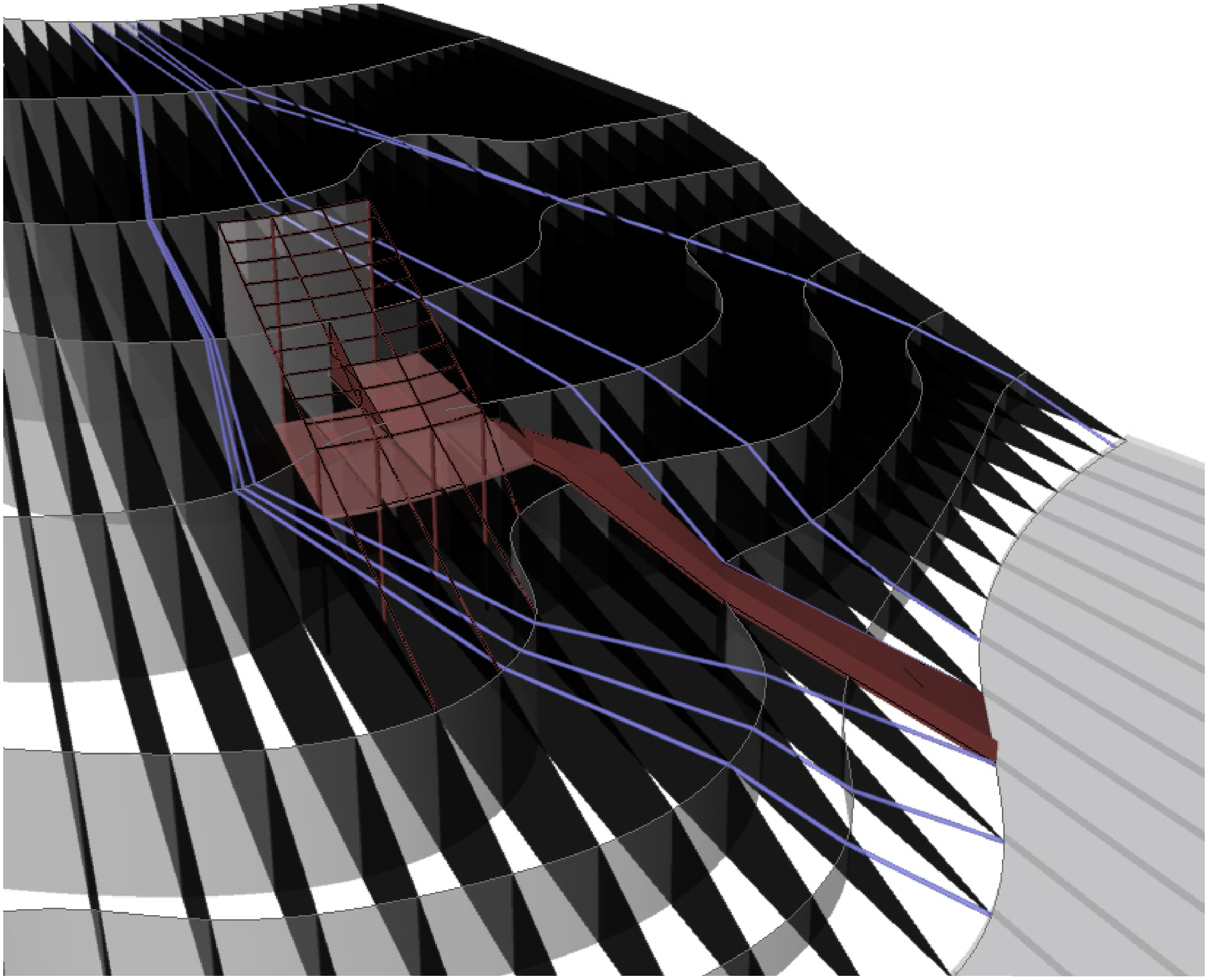
Line
>Perp
Pipe

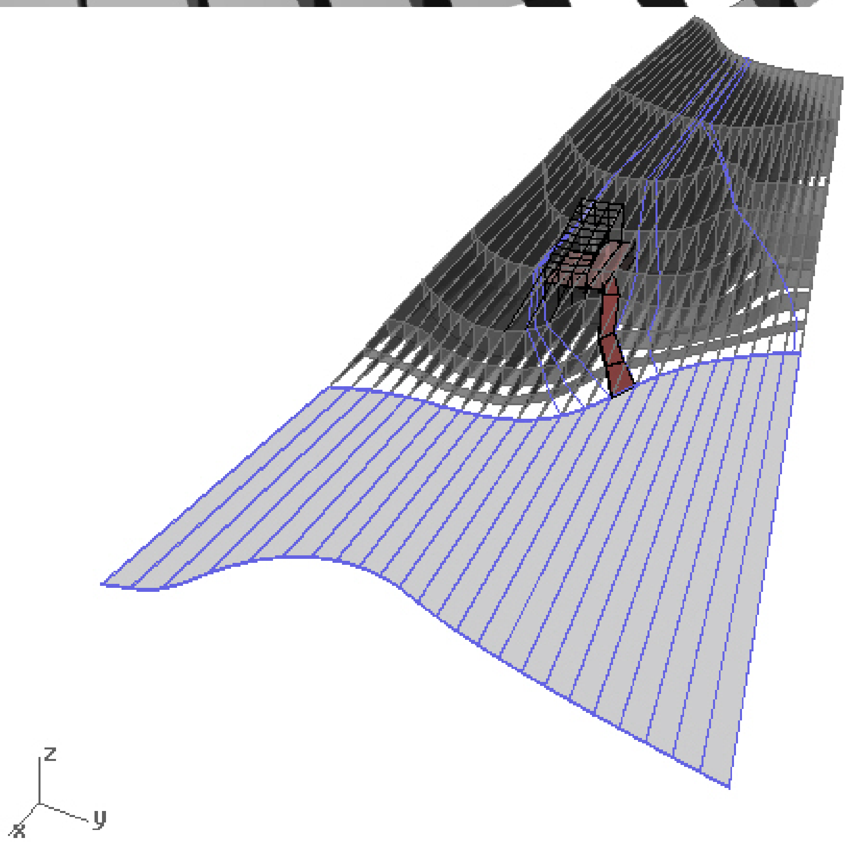
In thinking about an intervention into this location, the focus on water came to the forefront. The water flow down this hillside was analyzed and simulated using polylines that run perpendicular to the contours, as drops of water naturally would. The polylines were piped to emphasize the line weight and flow.

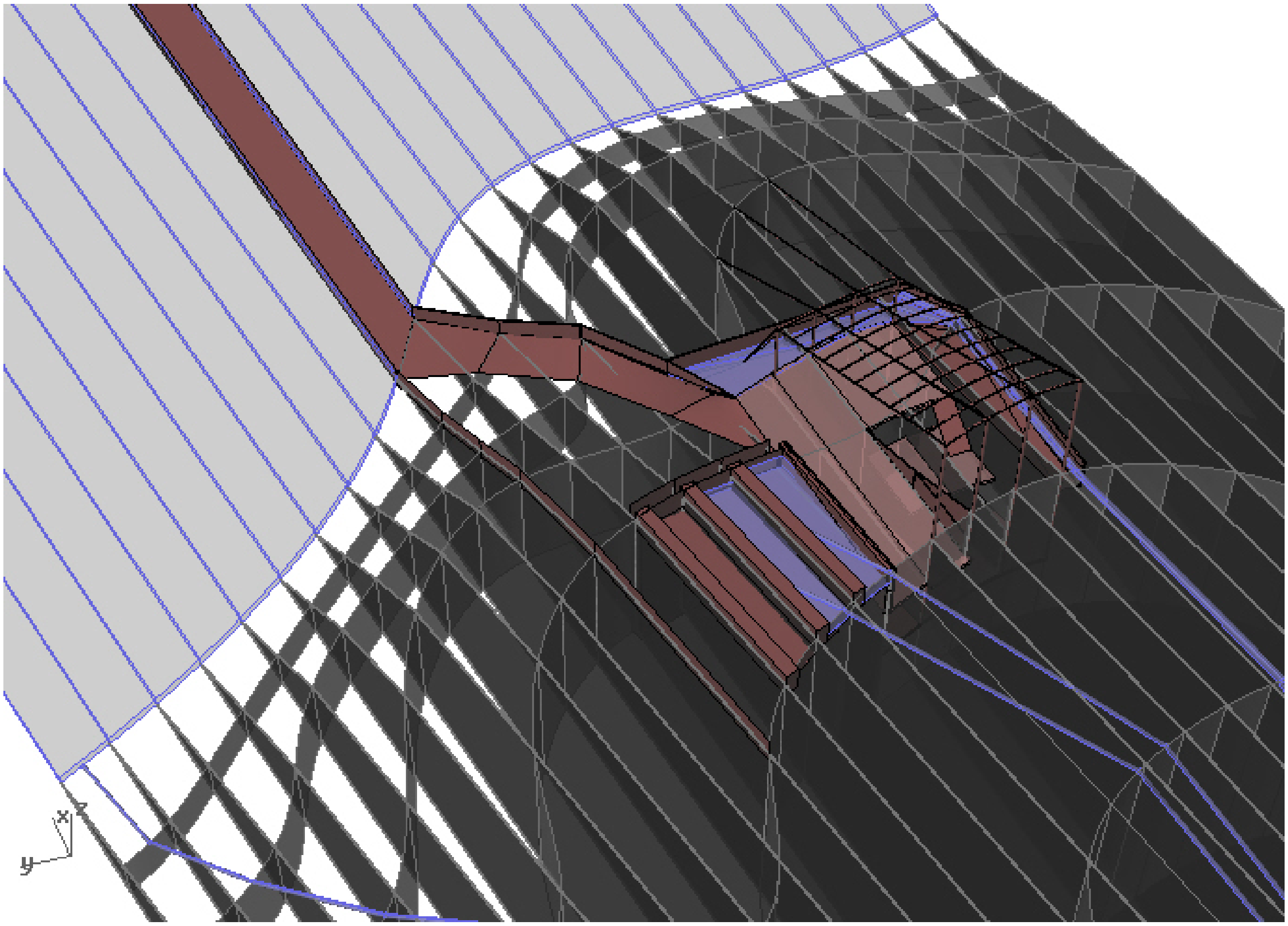


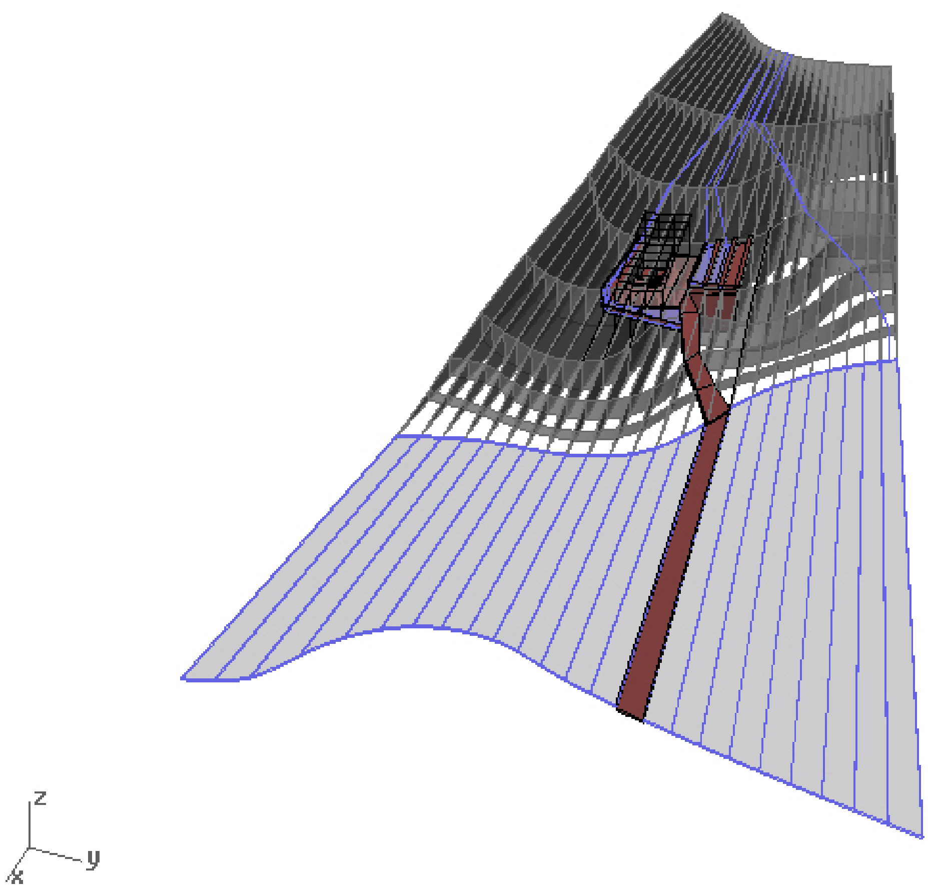
ExtrudeCrv

A site was located based on the flow of water around an area. The contour rib is extruded as an act of initial structure being implemented, and as a mode of integrating architecture and structured landform.



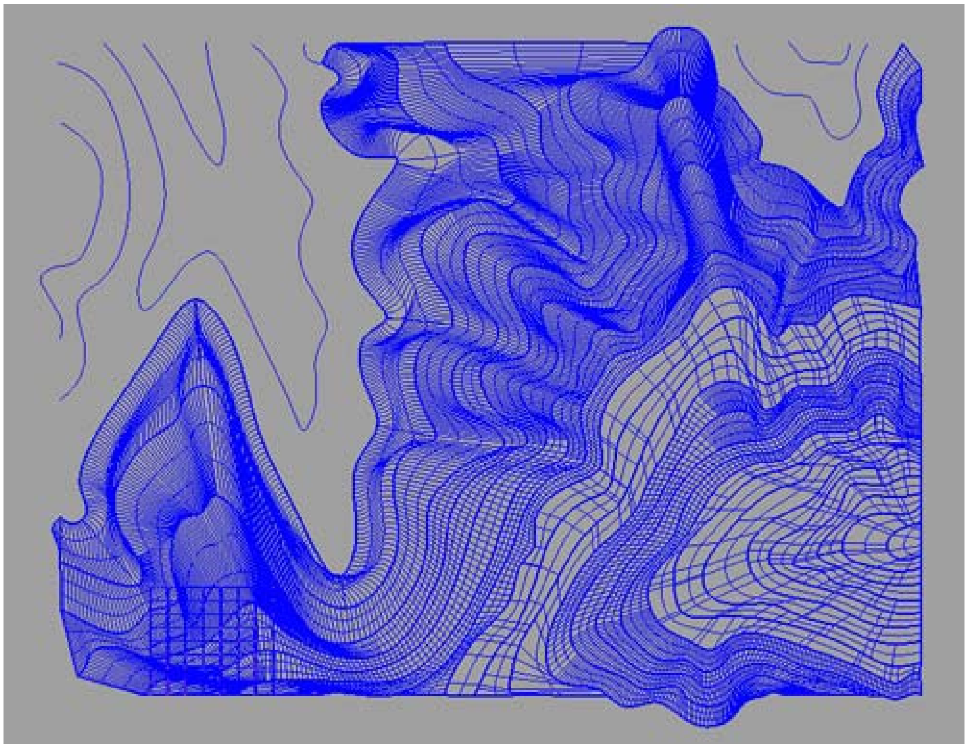

z
y
x



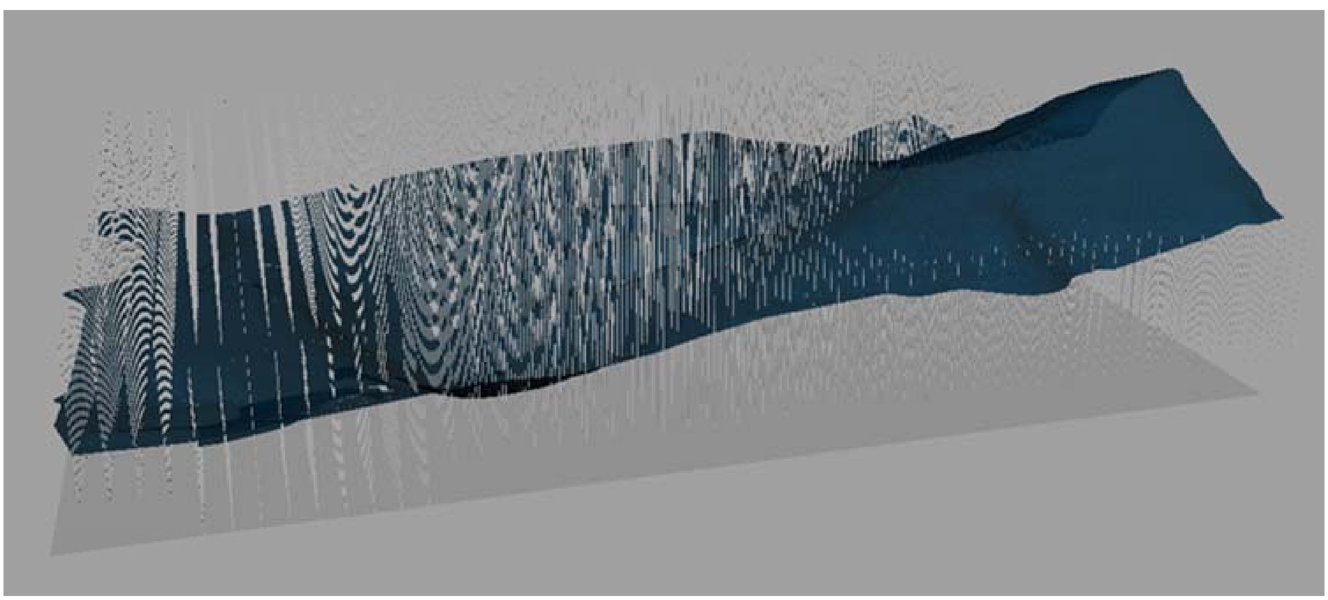


The proposed intervention converses with the structured land and the flow of its watershed.

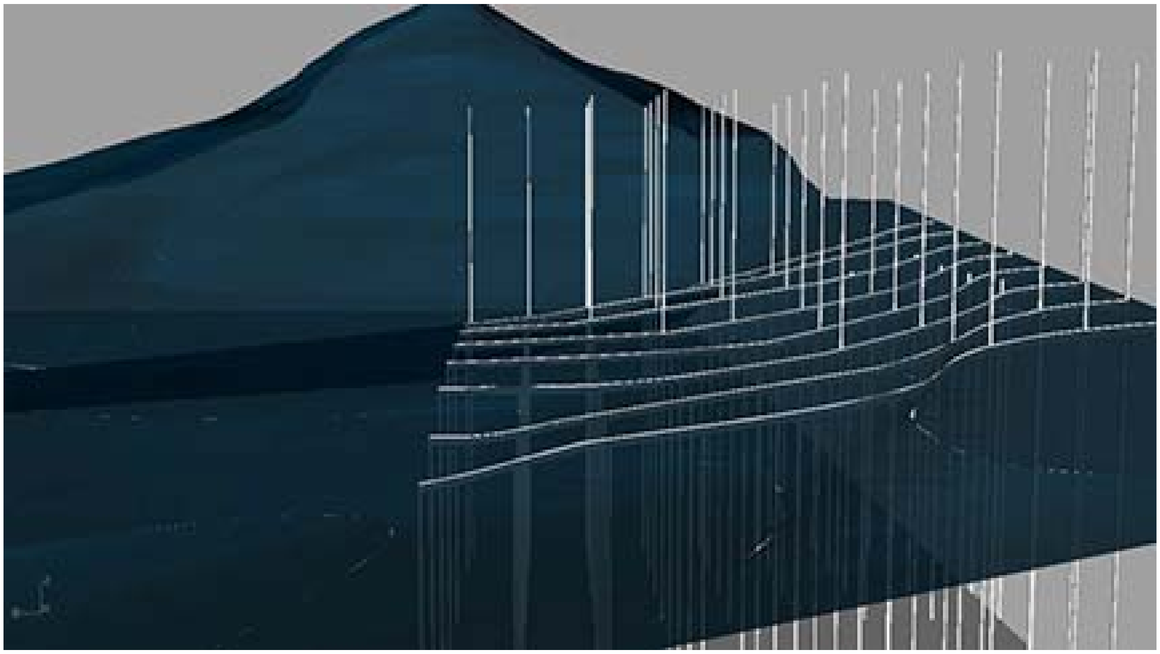
EXAMPLE : BATESVILLE VALLEY GRID
A Grid/Contour Landform
[2D OBJECTS] [TRANSFORMATIONS] [3D OPERATIONS] [



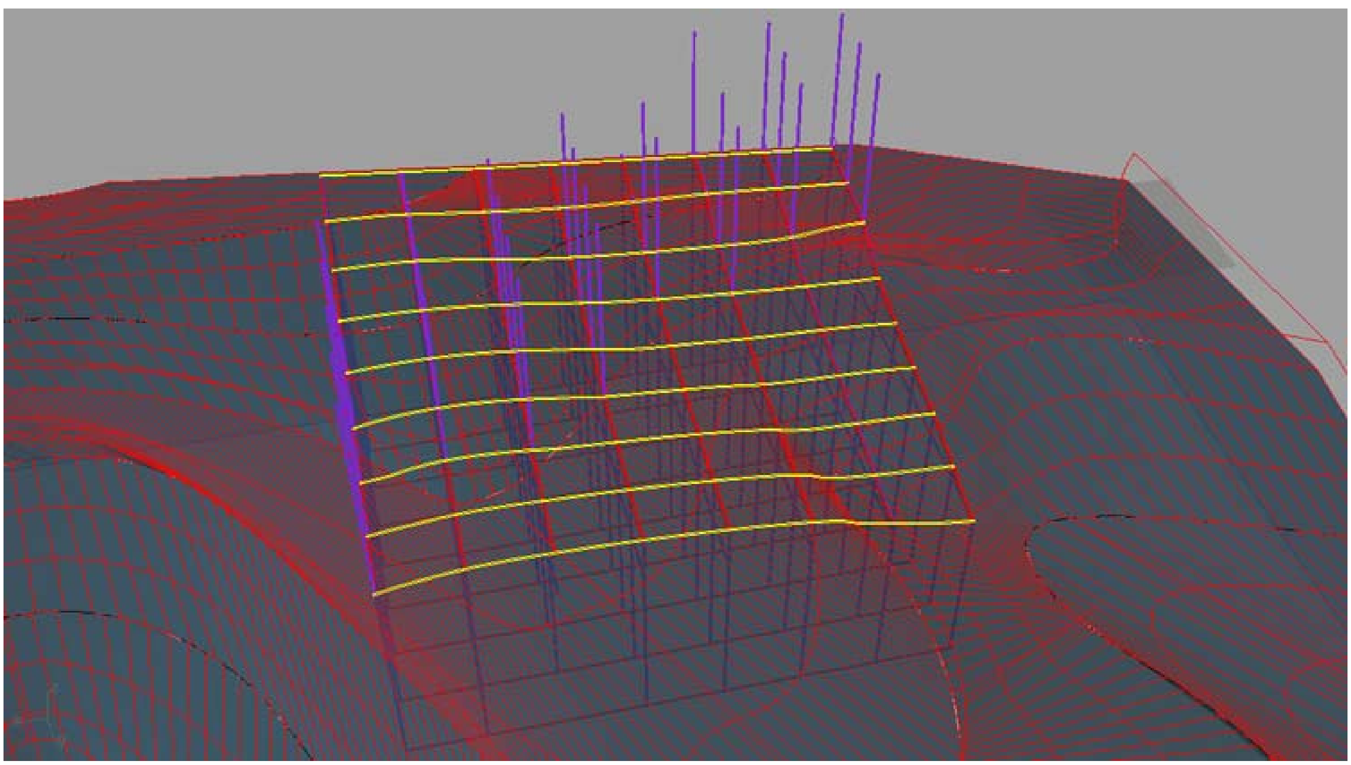
The landform was made by digitizing contours from a USGS map, then creating a surface using surface tools: **EdgeSrf** and **NetworkSrf**.



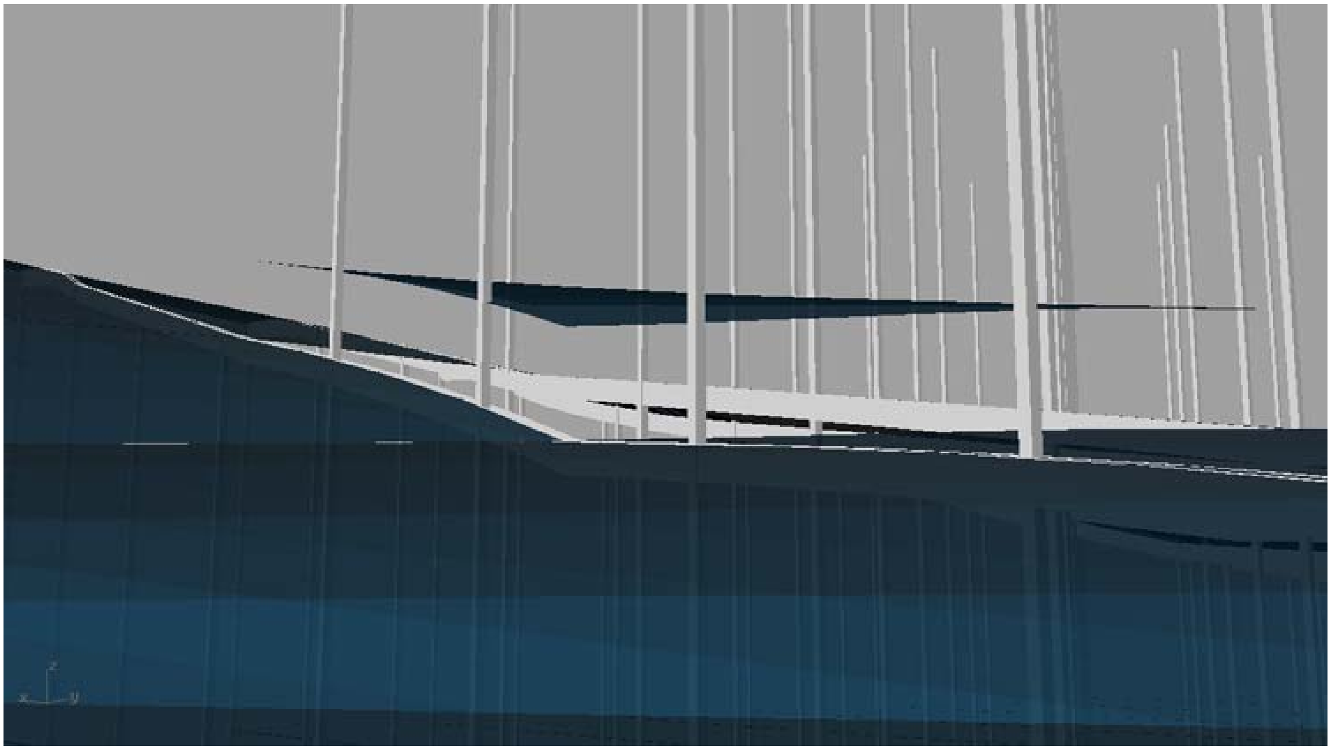
From a single column created from the **Box** command, thousands of columns were arrayed using **Array** in the Transform menu. It quickly became apparent that this was an unworkable mass of geometry, so all but 72 are hidden.



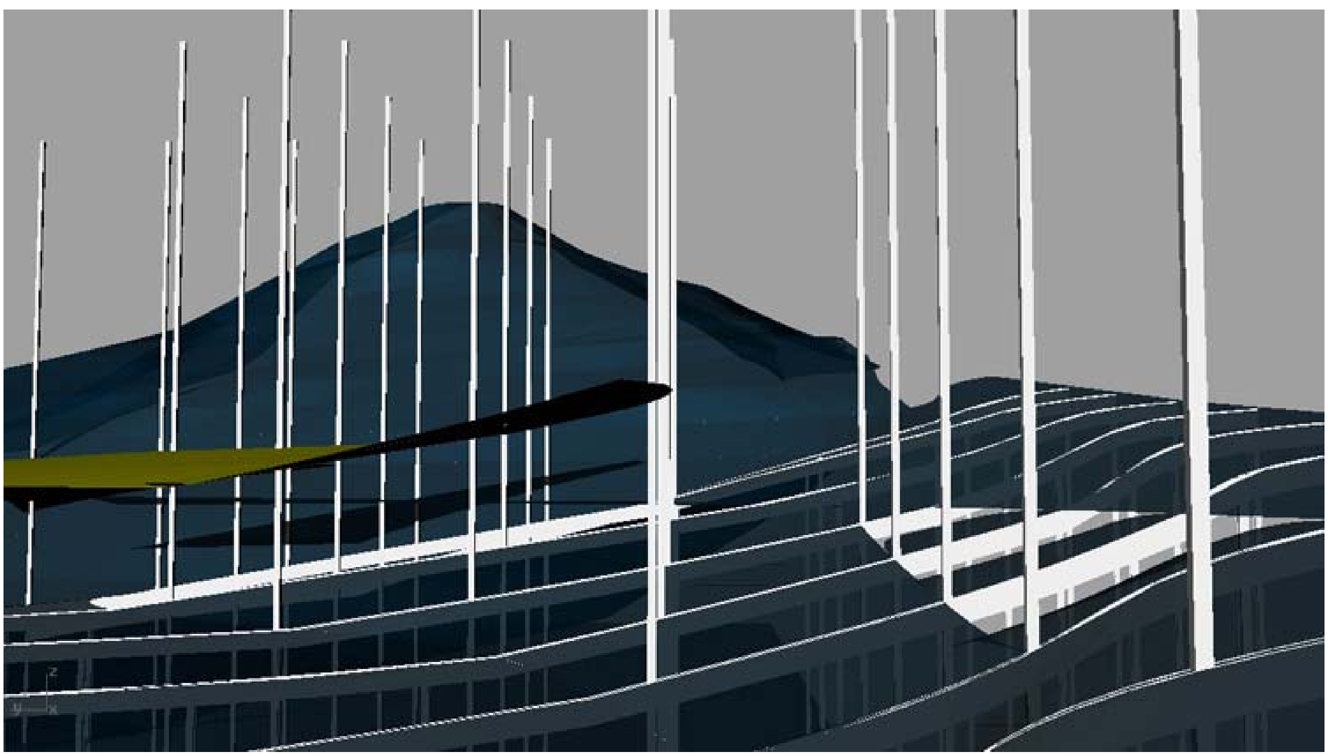
Beams, connecting column to column, and following the shape of the landform, were inserted using **Plane**, **Project** (curve from objects) and **ExtrudeCrv**. The landform surface is rendered as 40% transparent to reveal the construction. Notice that some of the columns extend above the ground plane while others have been cut off using **Trim** (the landform surface is the cutting object).



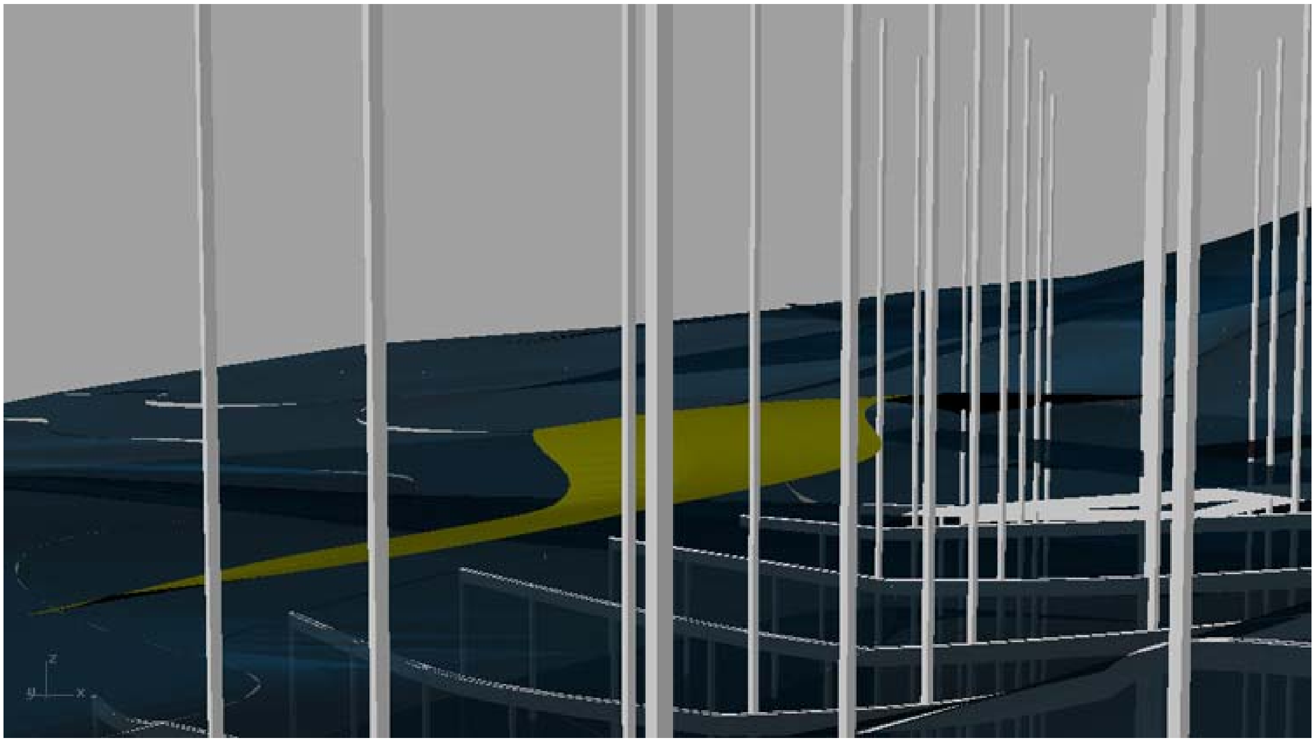
A secondary system created at 90º with the same process is shown in the yellow selected color. This becomes the basis to split the landform surfaces into smaller parts using **Split**.



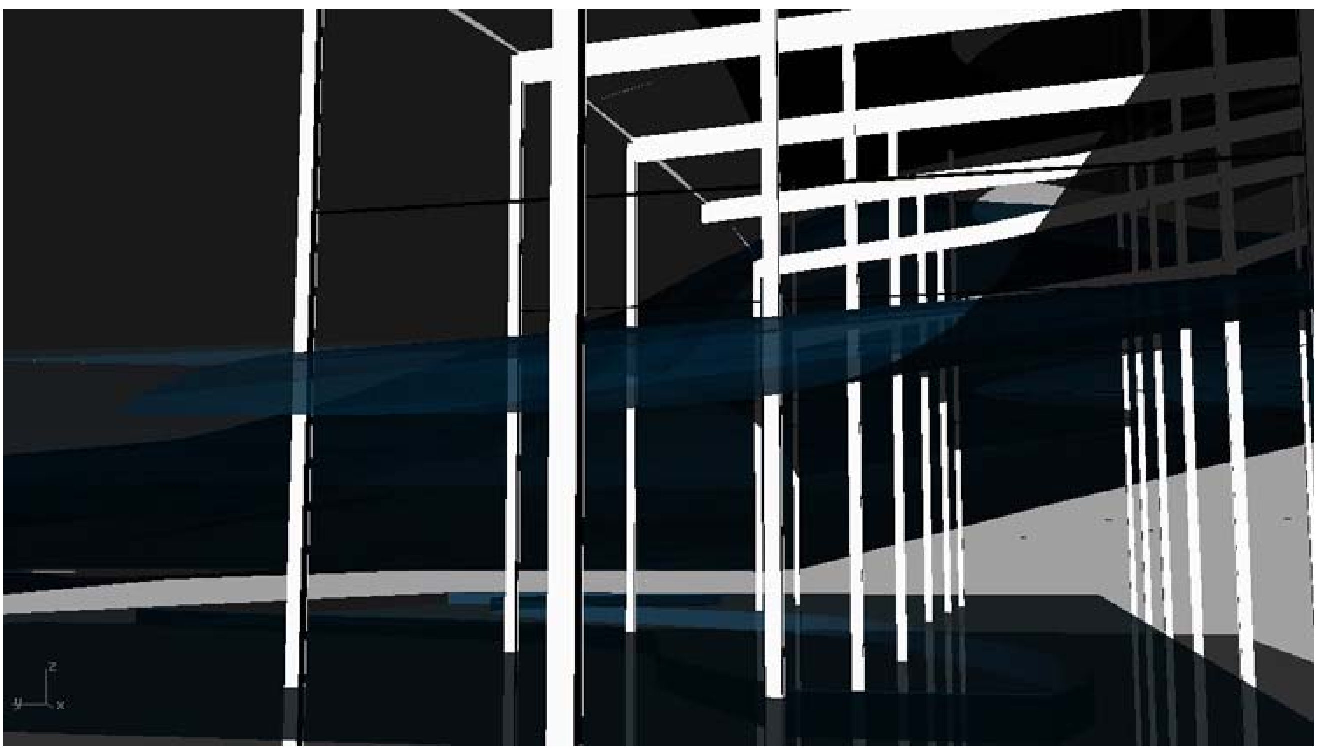
The split surface can then be manipulated using transform and surface tools. Here, a section of the landform is moved up a specified distance.



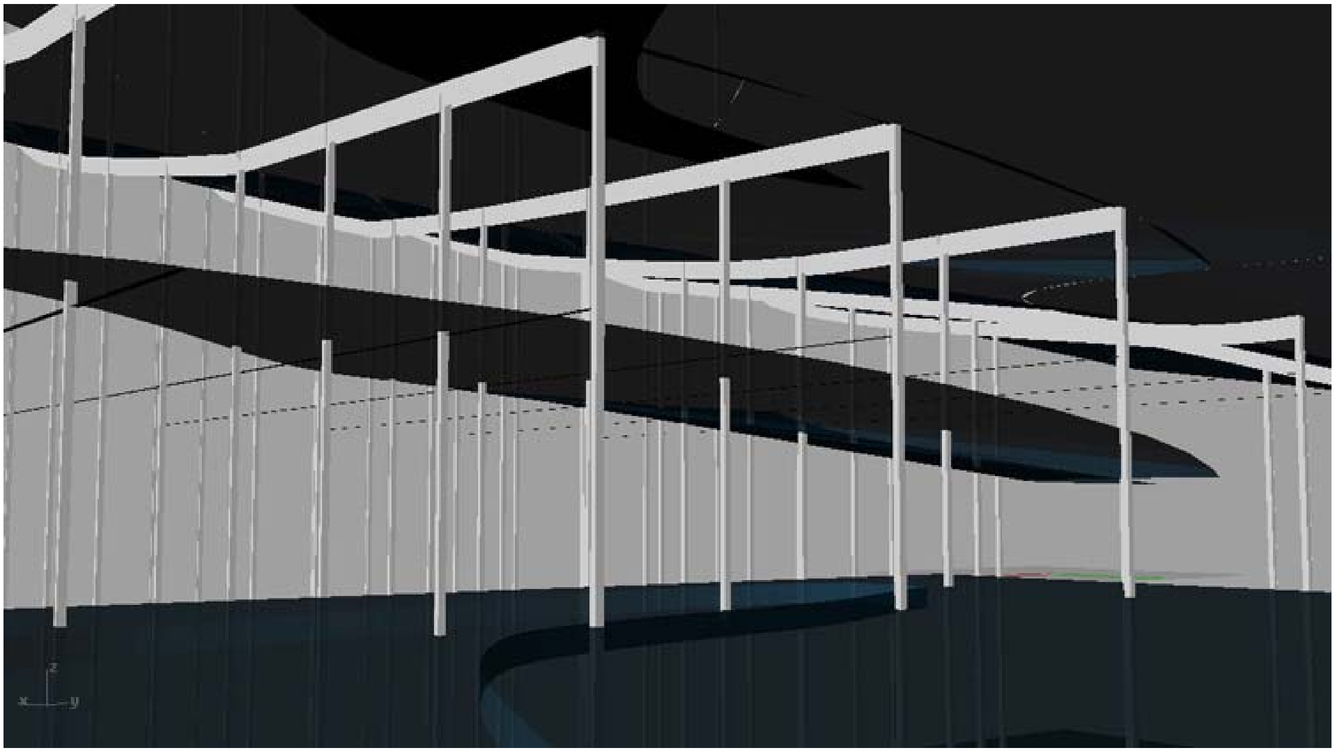
Other surfaces are shifted or duplicated using **Rotate** and **Bend** from the transform menu, and **OffsetSrf** from the surface menu.



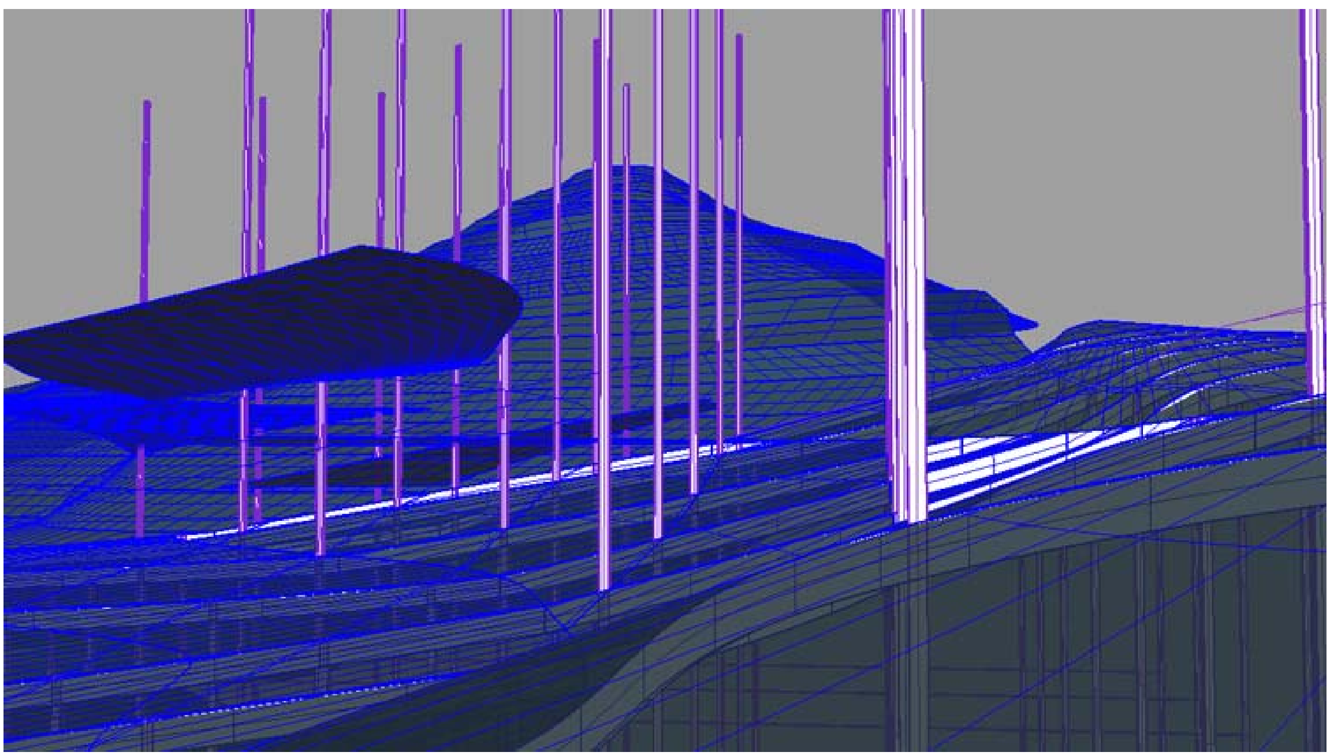
The green canopy is a sector of landform surface, split from the larger contour section, then copied and lifted using **OffsetSrf.**



Canopies are placed under the landform surface using **OffsetSrf**.



Contour-shaped mezzanines can be constructed by projecting the contour shape onto an orthogonal surface (**Project**), then trimming the results of a **MakeHole** operation.



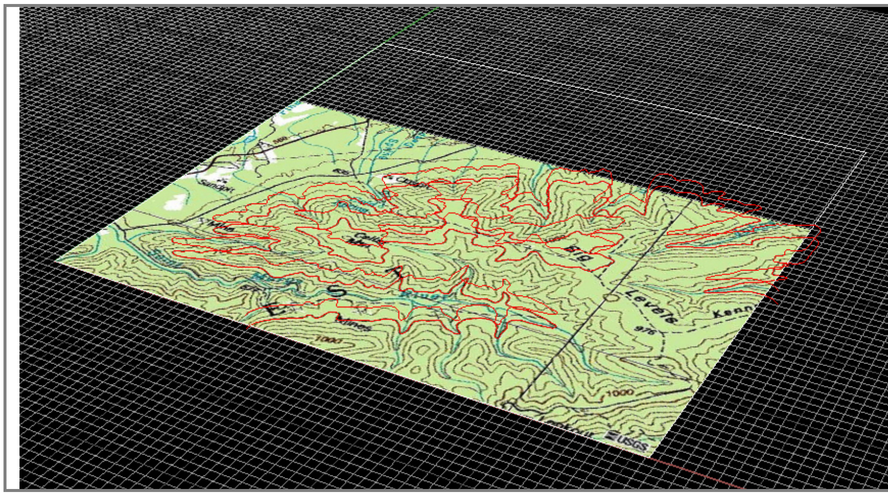
The aesthetic of this quick model is highly dependent on the tools chosen for its construction. The selection of the orthogonal geometry, the landform curves, and the transform tools already suggests a set of open and active surfaces that define a place within the larger landscape. The decision to explore both upper and undersurfaces of the landform reinforce the idea of the active surface. (Note that while certain structure-like shapes are present, many "floors" and "roofs" appear to float in thickened air).

EXAMPLE : CELLAR MOUNTAIN THEATER
Views, Exporting, and Presentation

[WORKPLACE] [TRANSFORMATIONS] [CURVES] [3D OPERATIONS] [EXPORTING]

Rarely is there a time when it is valuable to show your critic your digital model by zooming through it! While you are building a digital model, you should be critical about how you will be presenting it (what views are valuable to the development of your design) and therefore, what parts of the model are worth building to greater precision. There is rarely any value in completely developing an entire model! The nature of 3d modeling is such that it is possible to model every nut and bolt in a structure or every bench in a city, but usually it is more powerful to leave some things unresolved and it often takes more self control to do so.

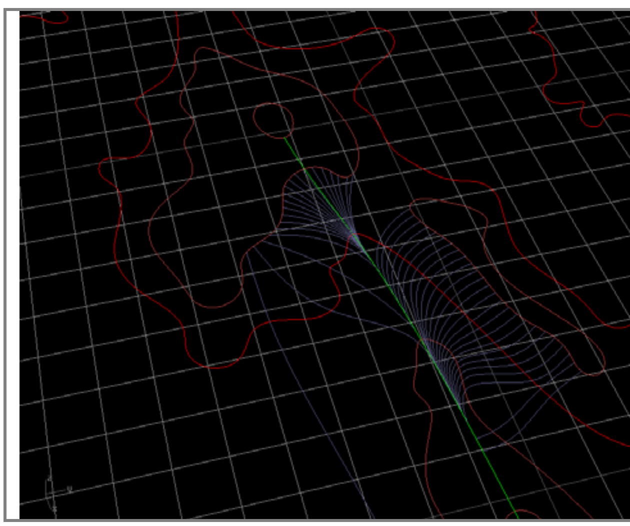
When you are building your model, it is beneficial to document your process, especially if you work through multiple iterations. See [EXPORTING] for Saving and Capturing views, Rendering, and other related commands throughout this Example.



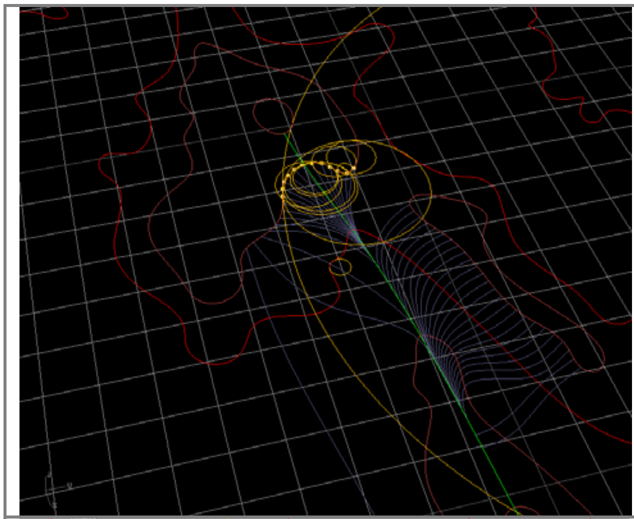
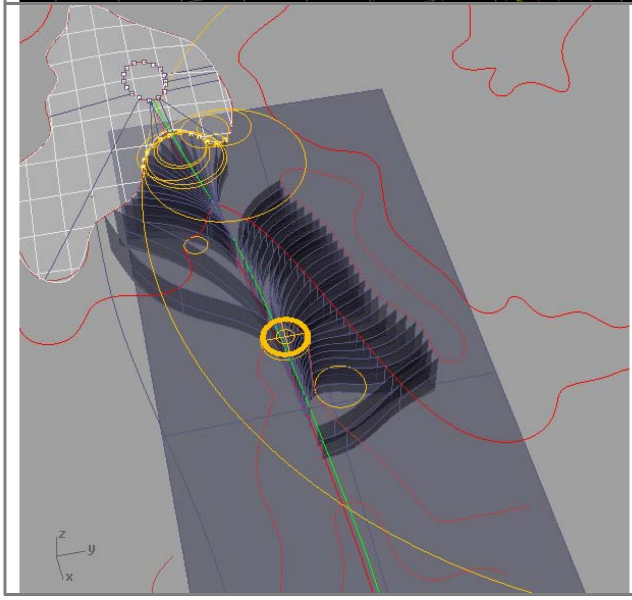
To start: a contour map of Cellar Mountain, down near good old Batesville, Virginia. A few important contours are drawn and offset to proper height [EXAMPLE : PANCAKES].

This view is a captured view of the viewport (**ViewCaptureToFile**), not a render.

This next series are captured views documenting a starting process :



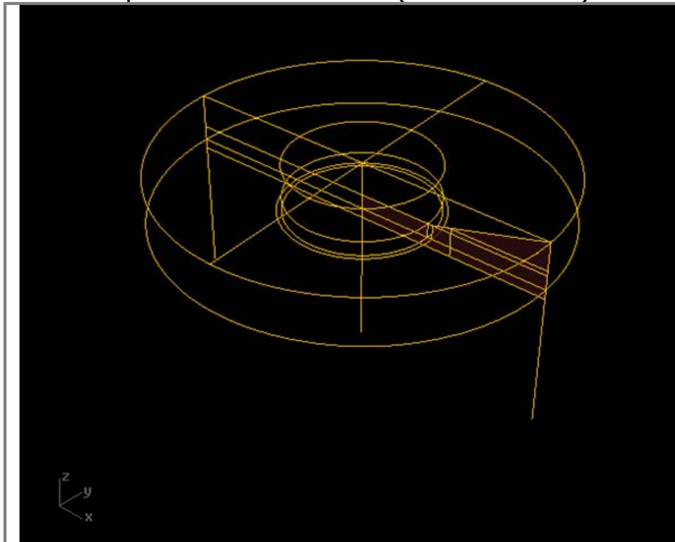
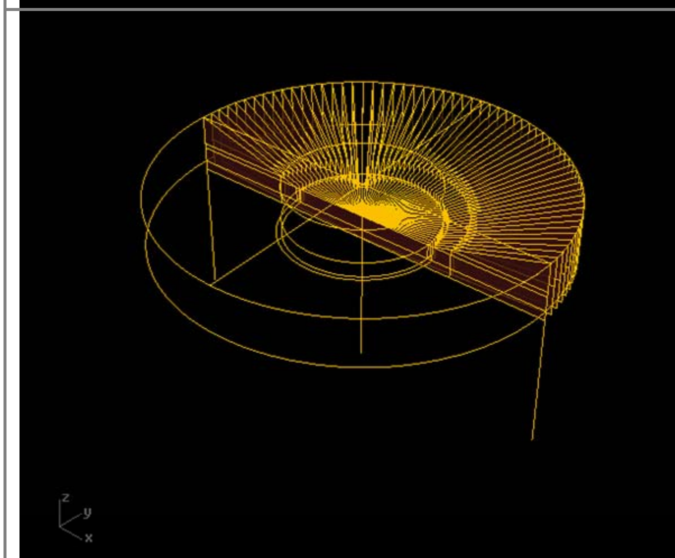
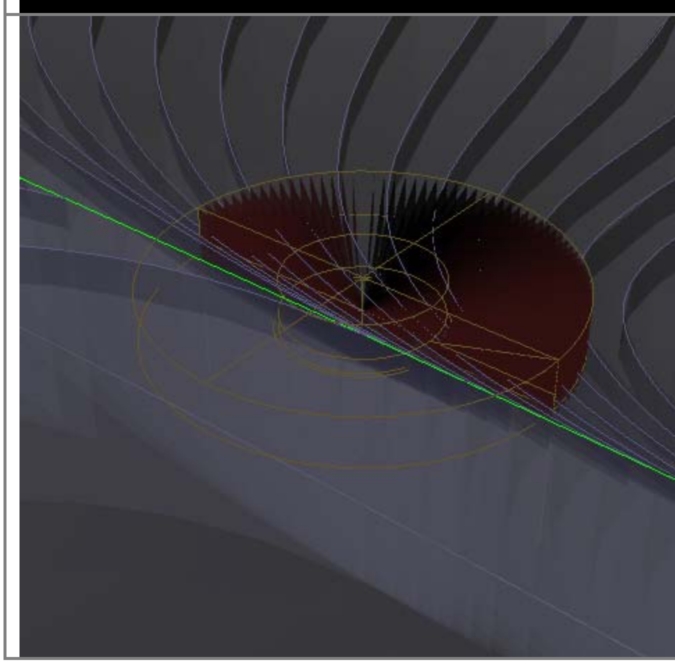
Drawing perpendicular lines to the contours in an area a theater could potentially occupy. These lines represent potential water flow, as seen in [Example : RIBBONS].

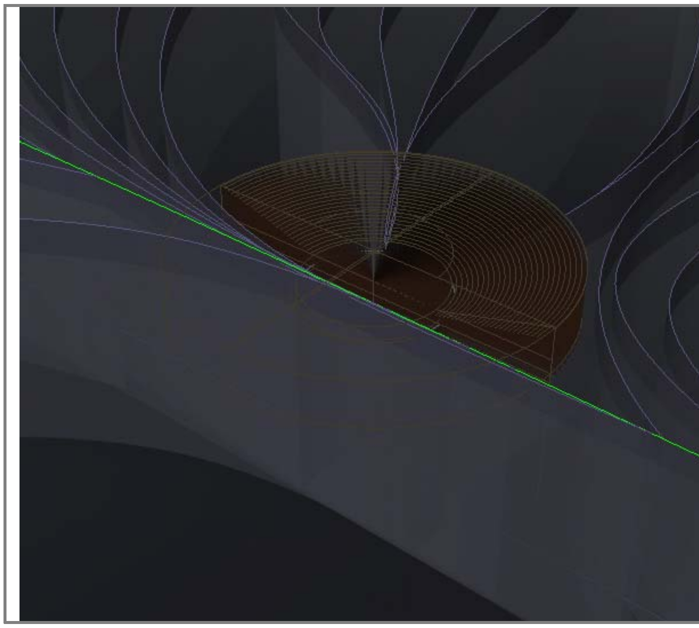
	Drawing potential sites and scales for the theater using circles with tangents to the contours.
	Extruding the water flow lines implies a surface. The theater site is selected and the theater begins to be manipulated.

Create and toggle layers to isolate objects, but then remember to put them back into context!

Returning repeatedly and capturing a saved **namedView** calls attention to the objects that have changed within that view. Had the view changed, the viewer would have had to work on reorienting themselves instead of working on discussing the model.

More captured saved views (**namedView**) of an isolated process:

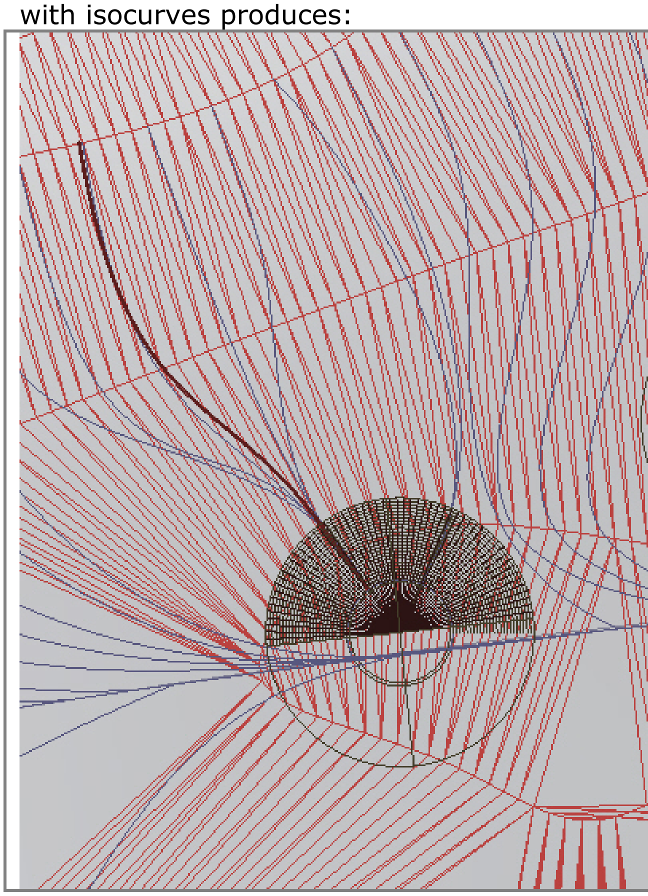
	**PlanarSrf** of the section through stage, trench, and audience seating
	**ArrayPolar** that surface implies seating without making a solid.
	Toggle the water layer back on. The built theater interrupts the landscape and reroutes the water below the aisles in the seating before it joins the stream behind the stage (modeled in green), so these existing water lines must be moved. These captured views explain how the architecture affects the landscape, before building:



and after.

Be critical of these "flying" views! How useful is a view to the discussion of your design? Is it more important to have a view in a random point in space, or to have a view on a human scale from within the model?

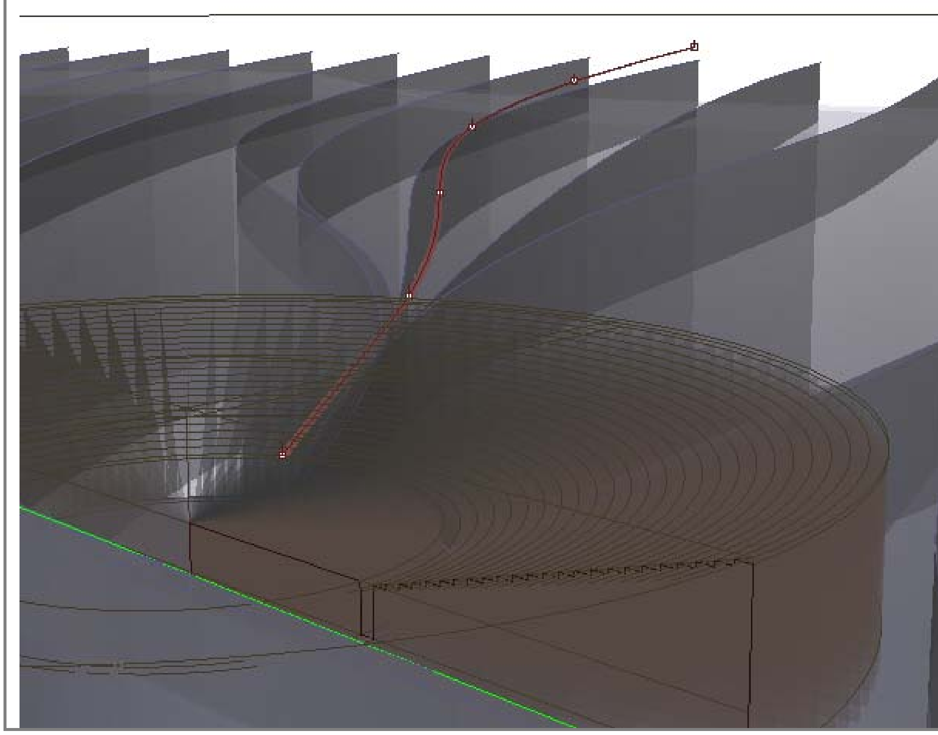
A basic path of an audience member is created following the water lines. Rendering in plan view with isocurves produces:



The path was created using Offset of a water path line and using **ExtrudeCrv** (**bothsides**). To give the path thickness, **ExtrudeSrf** is used for 6" below the surface, and then the path surface is copied to complete the polysurface.

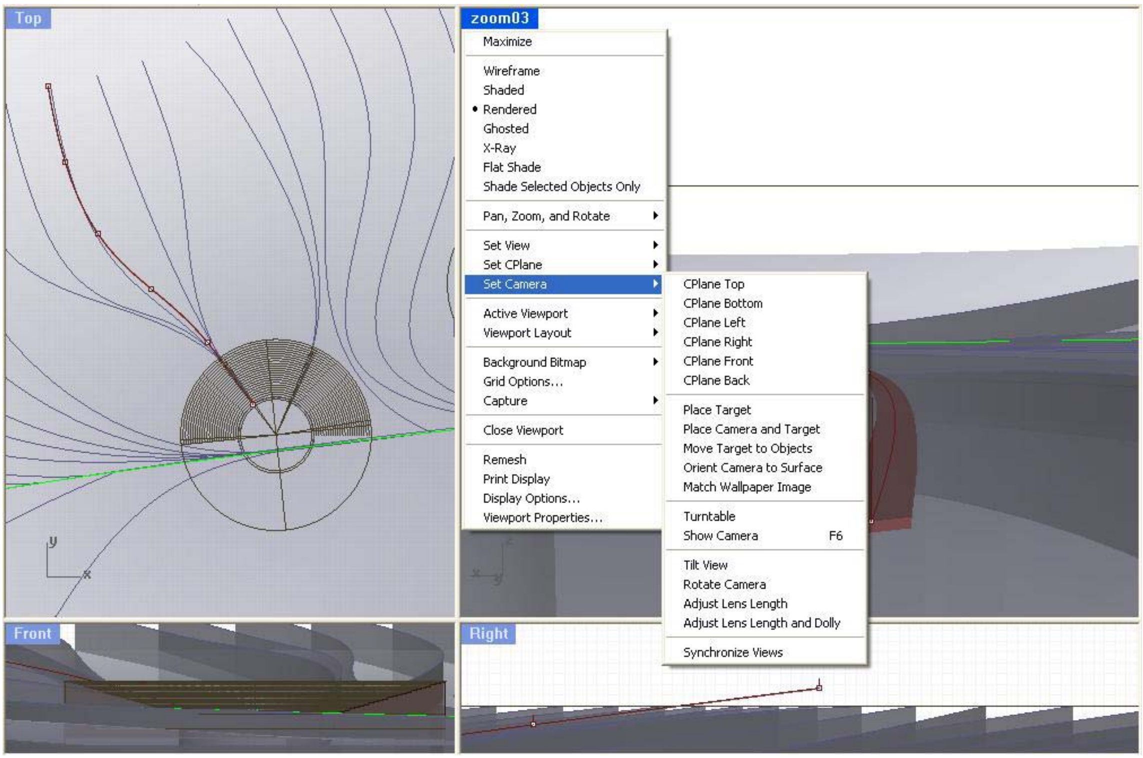
It would be advantageous in this case to show a series of views along the path. This can be useful to explain entry sequences and other experiential moments.

This process requires the manual manipulation of the **Camera**, and with that comes frustration and fear. However, if you prepare some basic lines and points (namely, the origin point of the camera and the focal point of the camera) before camera manipulation, it becomes very easy.

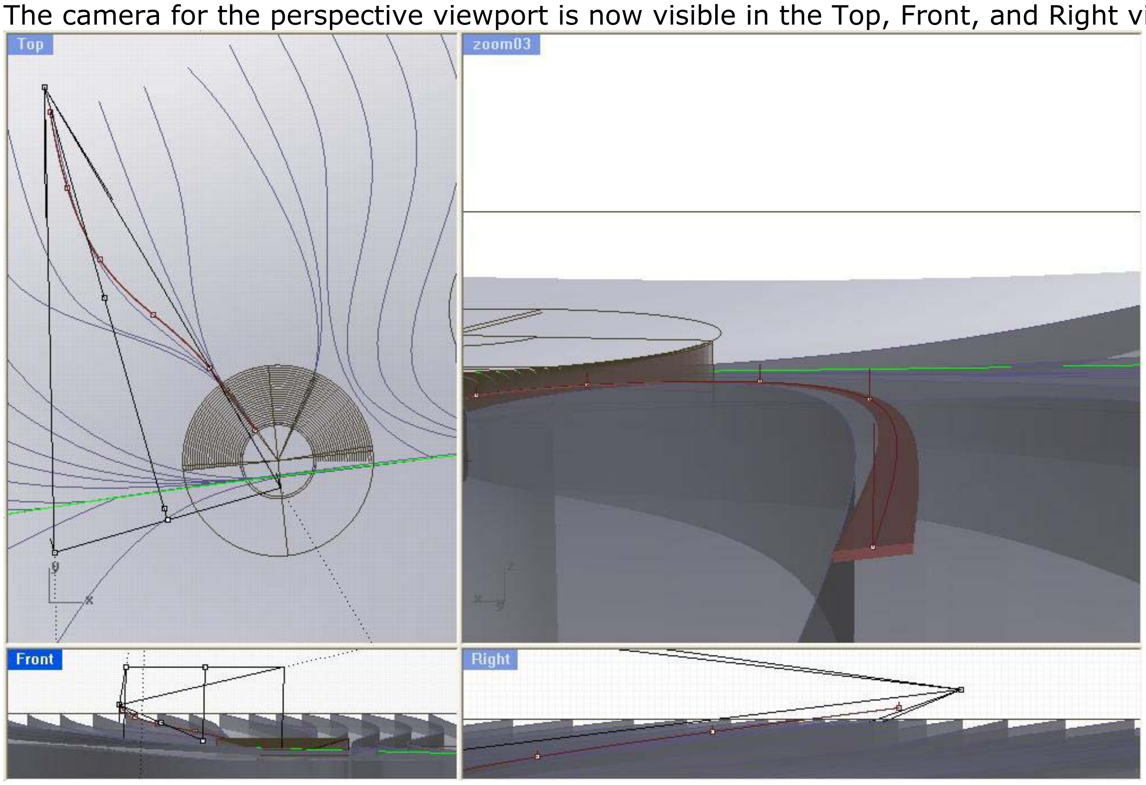


To set up a series of views, use **divide** to create equally spaced points along the center line of the path. Then draw a 5'6" line vertically from those points (or whatever dimension your eyeheight may be). The top of these lines will provide anchors for the camera.

To increase control of your view, the camera can be turned on and manipulated. For every Viewport in Rhino there is an active camera capturing that view, they are just hidden by default so you have not seen them. The Top, Front, and Right viewports have cameras with "parallel" views to the model, but the Perspective camera is flying around in space to wherever your mouse moves it. The Perspective camera is what will be used to capture views as you walk down the path. To "Show" the camera, right click on the Viewport name (blue box*)>Set Camera>Show Camera* or type **Camera** then show:

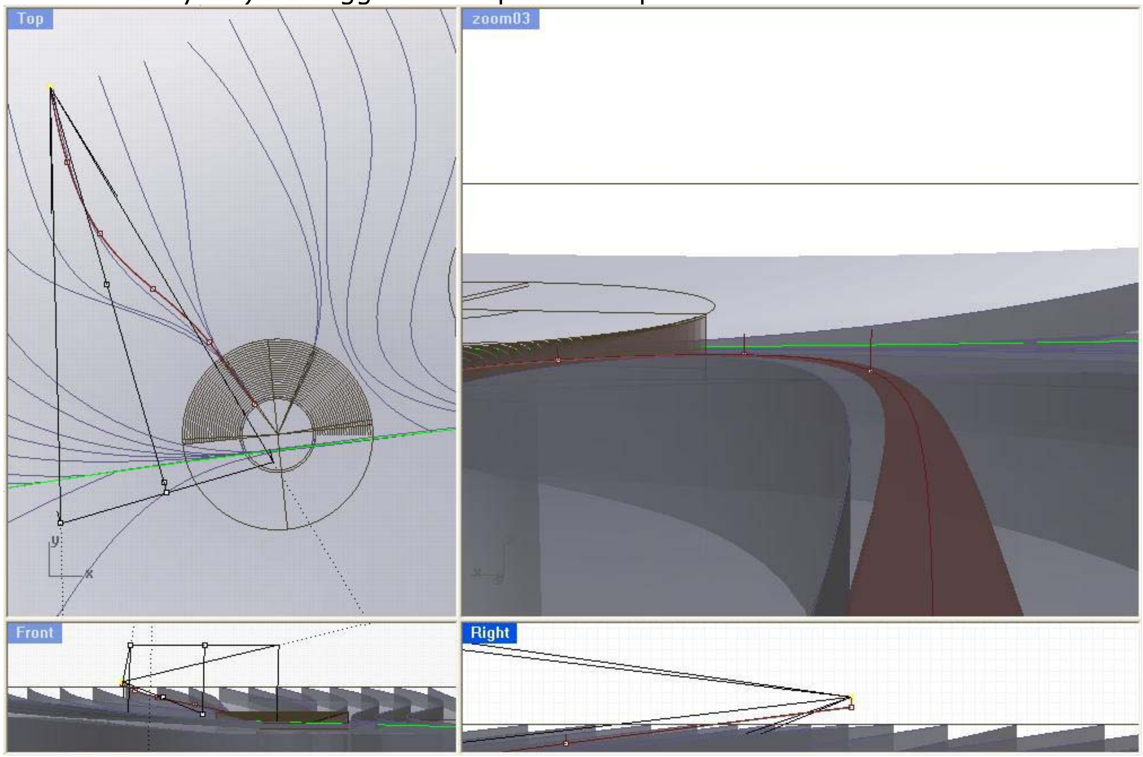


The camera for the perspective viewport is now visible in the Top, Front, and Right viewports:



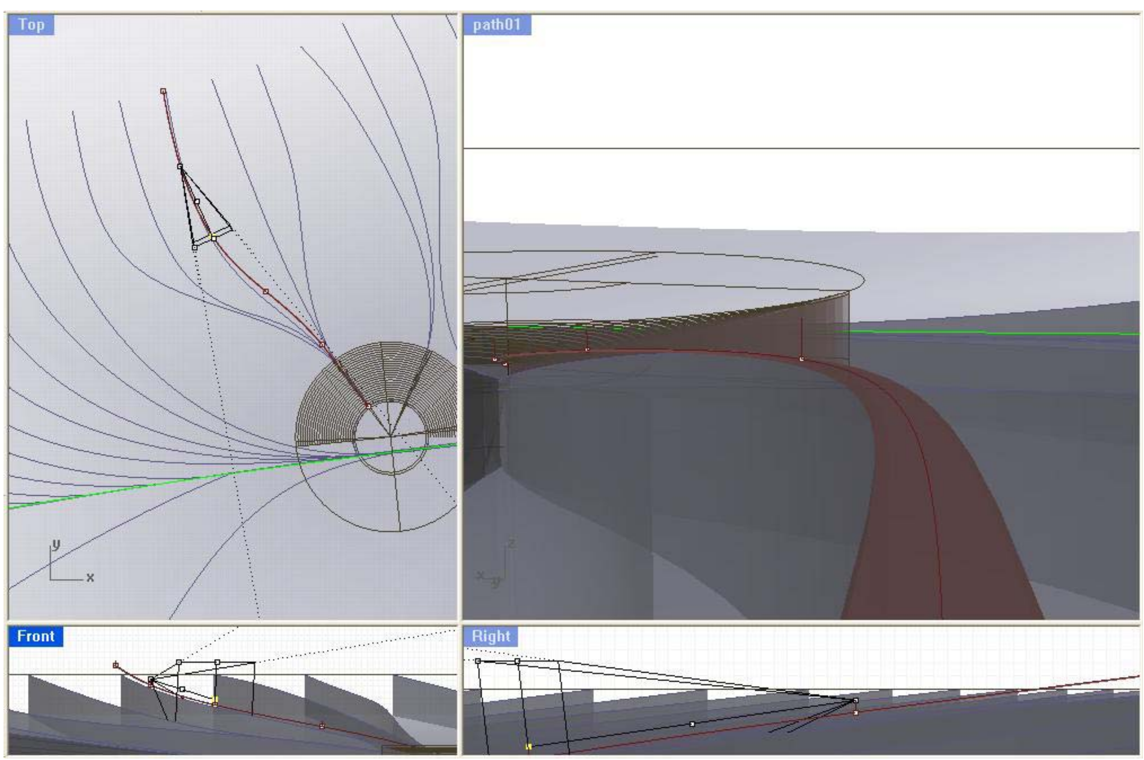
Notes: if you zoom, pan or rotate in the Perspective viewport, the camera moves accordingly in the other Viewports. The camera is not on a layer, and you can only move it by selecting its control points. There are 5 different control points on a camera, and each affects different actions (pan, zoom, etc). You can read all about that on *Rhino Help>Camera Show*, and in the meantime, be patient! Experiment with the different points and then return to a saved view to undo. Even with multiple set-ups the camera can be unpredictable. Save your work now!

For the first view along the path, the camera viewpoint (the point at the tip of the triangle, where the camera eye is) is dragged to snap to the top of the first line.

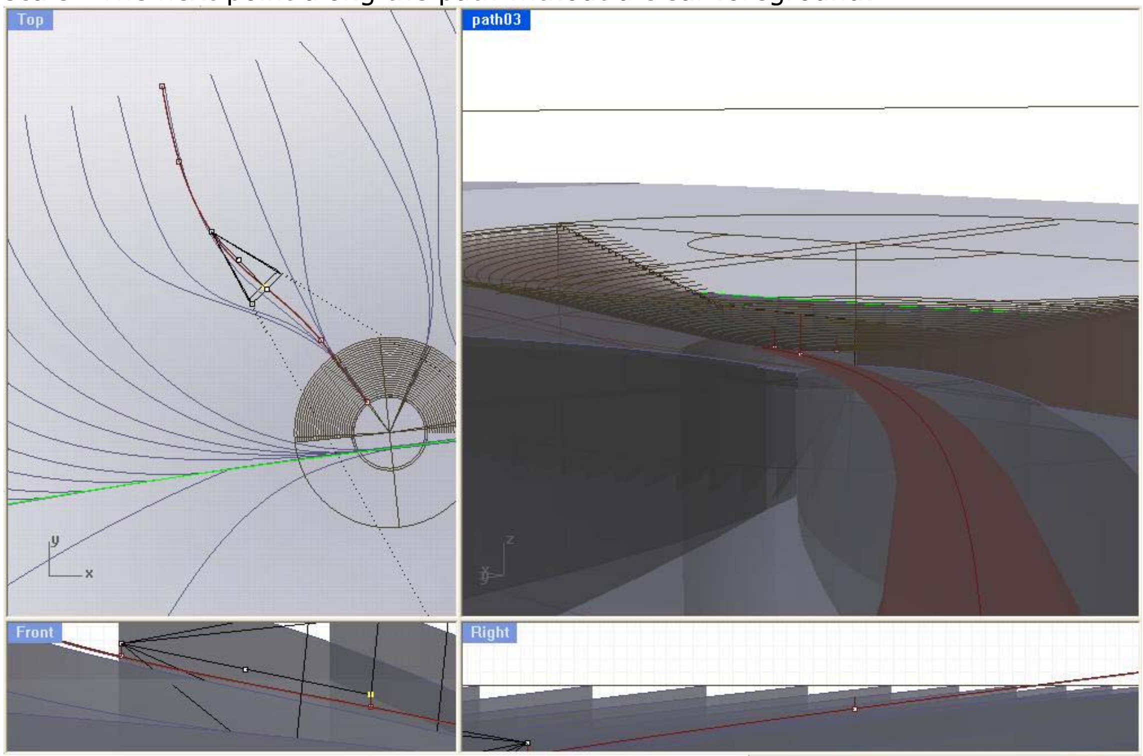


From there you can **Render** or **ViewCaptureToFile** [EXPORTING], and then move on to the next point along the line. If you would like to remove those vertical construction lines, put them on a different layer and then toggle that layer off before you capture the view.

The second view's origin point is moved the same way. To control the camera's direction, the "Target point", or the point in the middle of the base of the view triangle (there is a line through the center of the triangle connecting the origin to the target) is snapped to the next line.

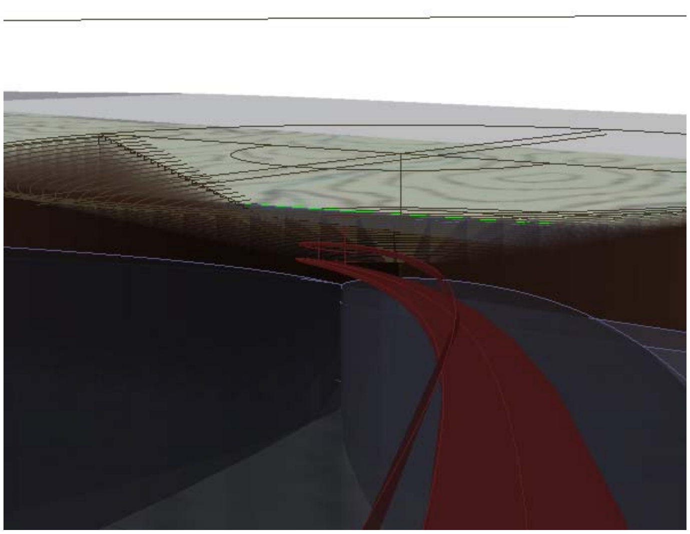


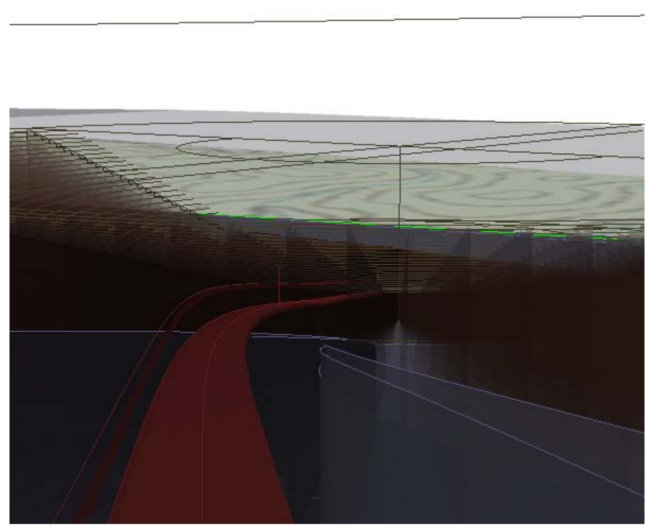
To improve a perspective view, it is recommended to have a foreground, middle ground, and background. These views lack a foreground, which would give the viewer a better sense of human scale. The next point along the path without a clear foreground:



A handrail is an easy addition, but think of other objects that could appear within an arm's reach or a hand's grasp. Tree branches? Bicycle handlebars? Other theater-goers? Many of these things are better added in other programs [EXPORTING] but keep a foreground in mind when you are planning your views.

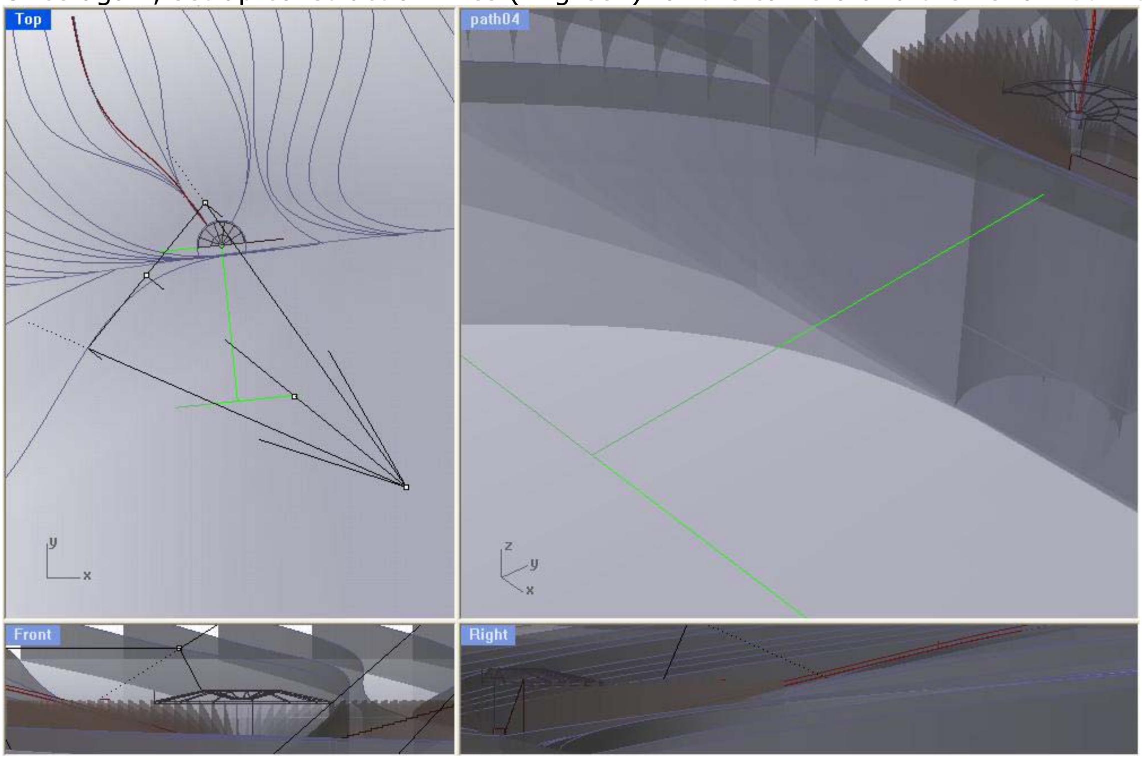
Rendered views of the 3rd and 4th points along the path, with handrail:



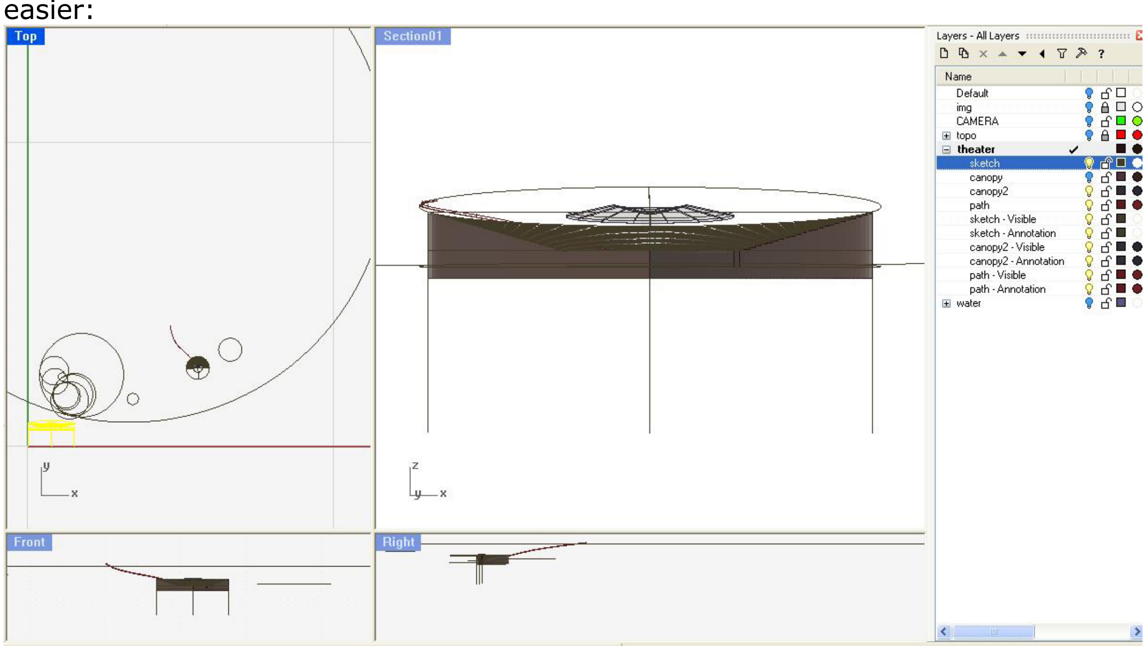


A canopy is added to the theater, as a water collector, shader, and structure for theater lights. A section would be helpful to describe the relationship of water flow. This requires getting the camera parallel to the section line, which can be done in a few different ways.

Once again, set up construction lines (in green) for the camera and then show **Camera**:



The construction lines are on plane and parallel with where the section will be cut. If they are out of plane, the camera will be tilted. Other layers are turned off to make camera manipulation easier:



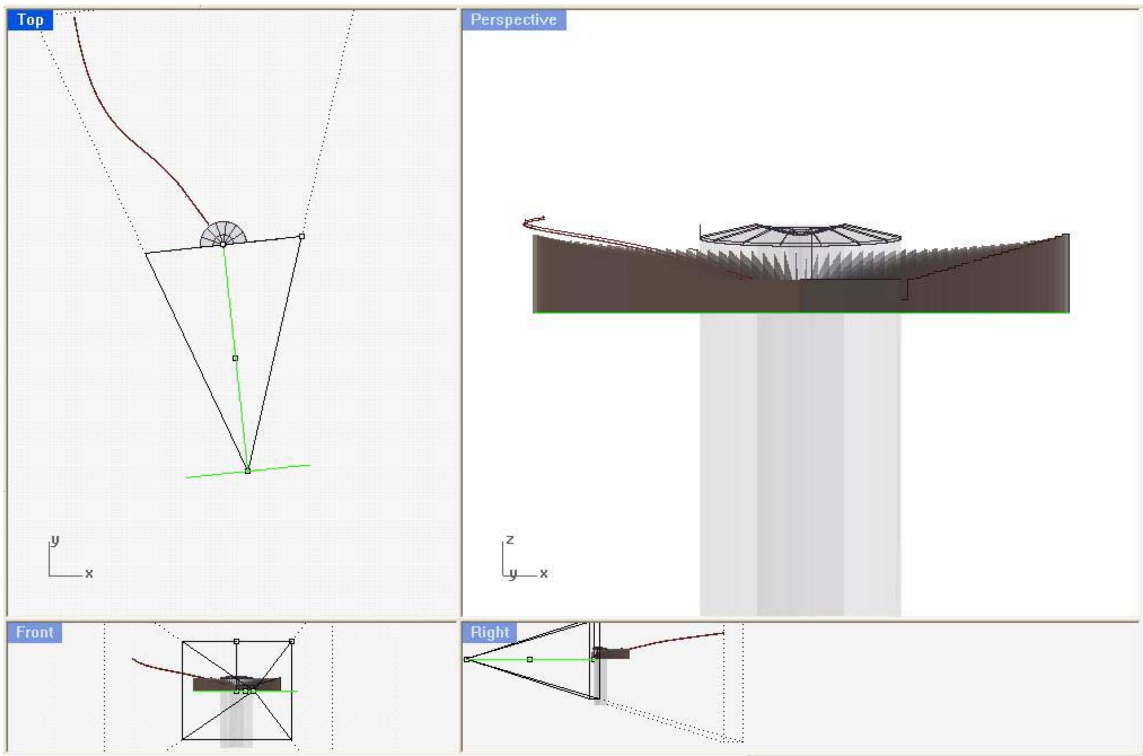
Save the view and hide the camera to continue.

[Had the model been already parallel to the construction plane, we would not have to manipulate a camera like this, since the preset Front, Back, Right, or Left views would have provided a parallel camera. The alternative to getting a measured drawing requires redefining the construction plane instead of the camera.]

To get the wireframe section lines, select what you want to section and use **Make2D**. Make sure all objects that you want to export are unlocked and visible, and make sure you type in **Make2D** while the appropriate Viewport is active (in this case, the newly saved Perspective). The wireframe will appear in the Top view near the origin on new layers.

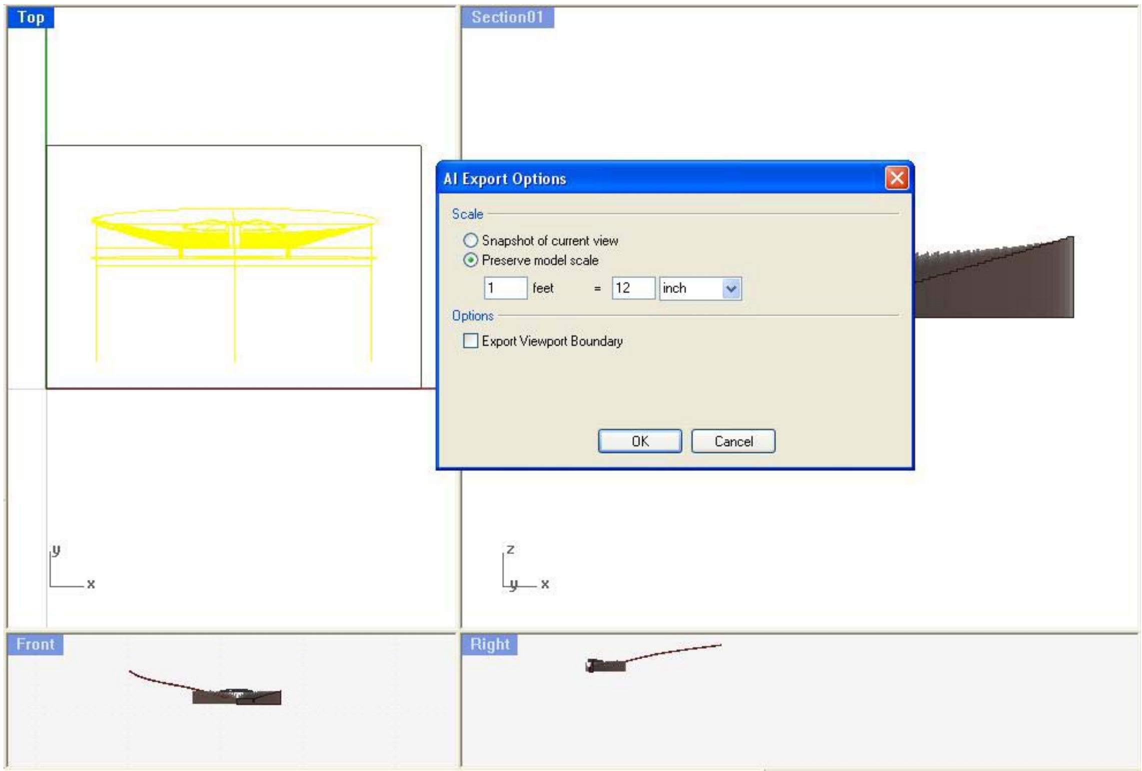
In this case, hidden lines are not shown, and layer information is kept. [EXPORTING for further details on Make2D].

Note where the new objects have appeared in the top view (in yellow), and the 6 additional sublayers added to the "theater" layer. These additional layers can get very messy if you are doing a lot of Make2D, so keep an eye on them.



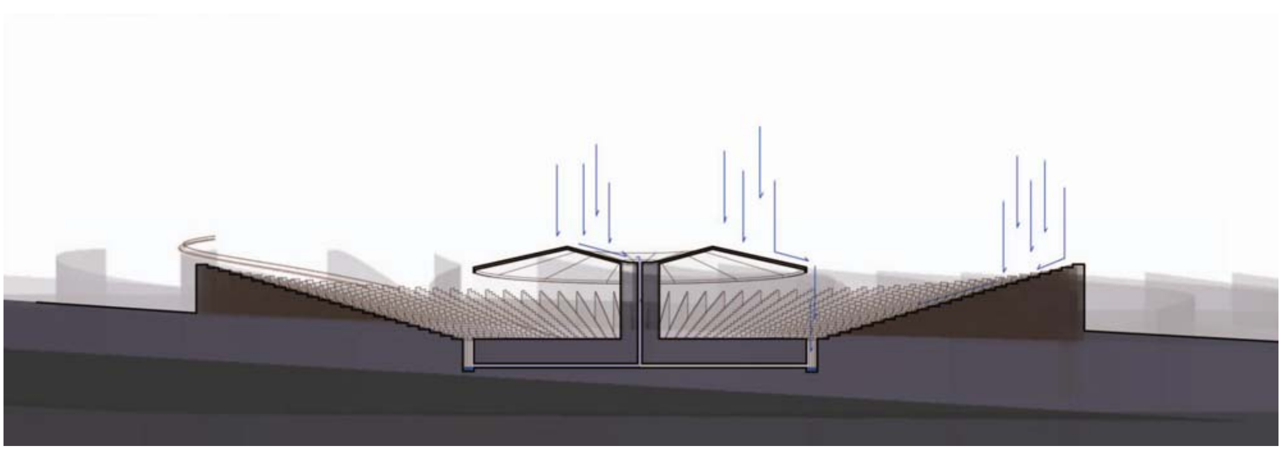
There are numerous ways to manipulate this 2D drawing now. It can be moved within this same Rhino file, edited for clarity and scale, and then printed right from Rhino. Or it can be exported to a new Rhino file or other 3D modeling software (say, for creating construction documents or setting up for CNC fabrication). In this case, however, Illustrator will be used so an image can be layered on top of the wireframes.

The 2D drawing must be scaled before **Export**. The final drawing will be on an 11"x17", so the objects are scaled to 1' = 1/64" scale. To make exporting easier, an 11"x17" rectangle is drawn at the origin and the scaled drawing is placed within it.



The 2D objects are selected for Export (shown in yellow above). Since the model was scaled manually, when the "AI Export Options" box pops up, the "Preserve model scale" can be kept at 1'=12".

Before going into Illustrator, a **Render** is taken of the current Perspective Viewport with more layers on. While in Illustrator, the wireframe lineweights are edited and the render transparency is adjusted so that the section line is clear. To make the drawing more useful, a simple rain diagram is added in Illustrator.



Once the model is properly exported out of Rhino, more information can be added in many different ways. This example only uses outside programs minimally and does not incorporate any hand-drawn work. **Make2D** is also extremely useful if the raw wireframes are printed (either for a complex perspective or a measured drawing) and then drafted over. Then the hand-drawn wor kcan be scanned in and again layered over more information. Photoshop can be used to layer in background images, people, textures, and atmospheres. Look at examples to get ideas of what reads well and what describes your project best.

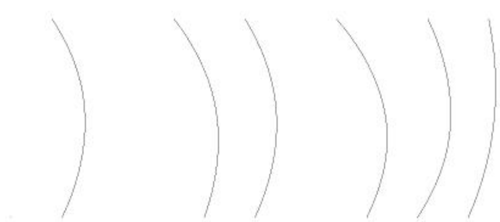
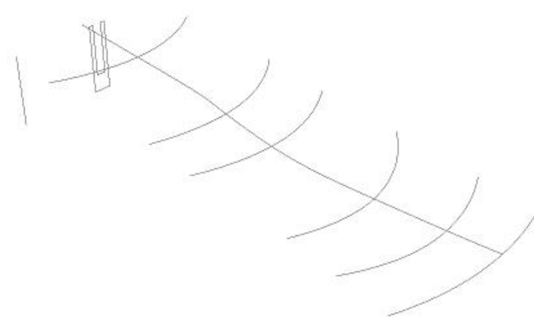
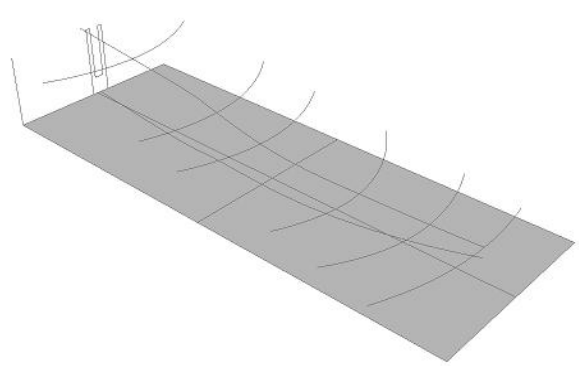
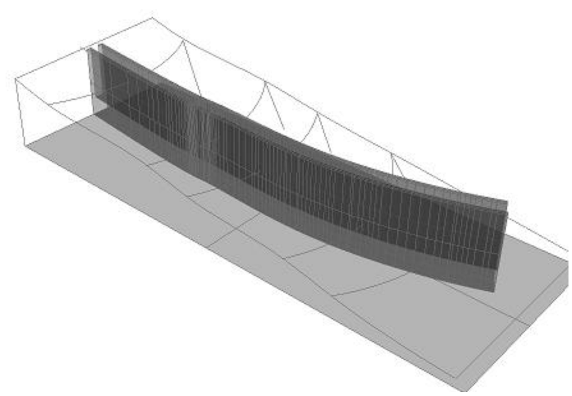
EXAMPLE : URBAN LEAF
A Contoured Slope as an Urban Leaf Fabric

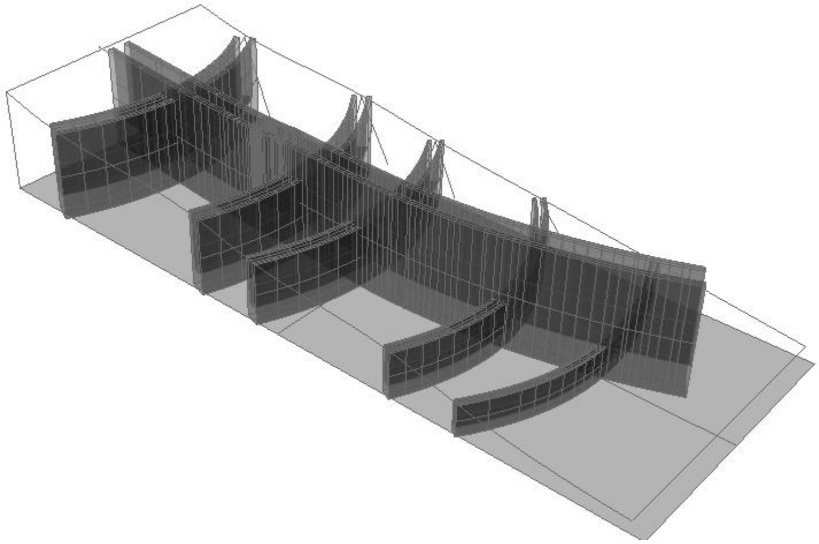
[2D OBJECTS] [3D OPERATIONS] [TRANSFORMATIONS]

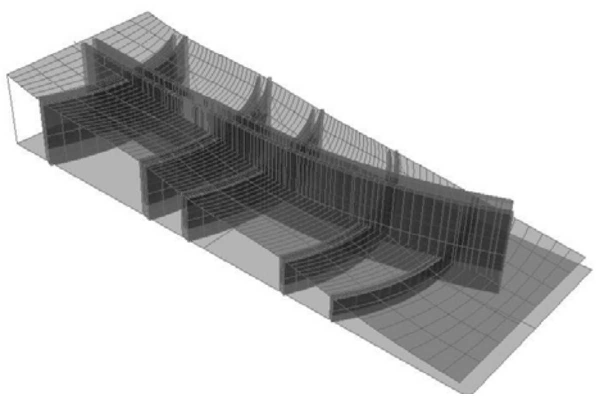
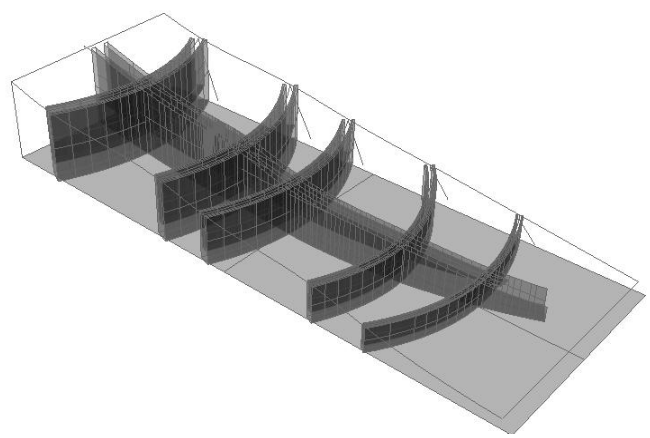
Starting from a small scale contour map, the landform is given a structure inspired by leaf systems. In this case, the hierarchy of structure found in a leaf lends itself well to the hierarchy of an urban structure.

In Rhino technical terms, precision of the model is upheld by construction surfaces that are not seen in the final result. Throughout this process most of the geometry is trimmed by other geometries, which requires surfaces to extend other surfaces. This can quickly become imprecise and unmanageable if the more "finalized" objects are being used as trimming tools. In order to keep this in check, surfaces are duplicated, extended, and then put onto an inactive layer after they are used. This way they can be used again (keeping the model precise) and the edges of the model do not move arbitrarily.

Don't be afraid to model something and decide that it is not working; just put it on a different layer so it can be used again in the future.

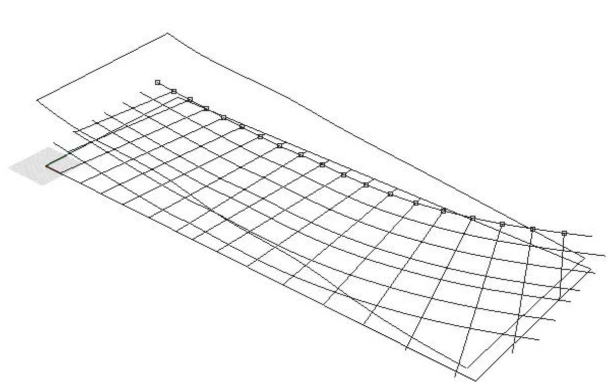
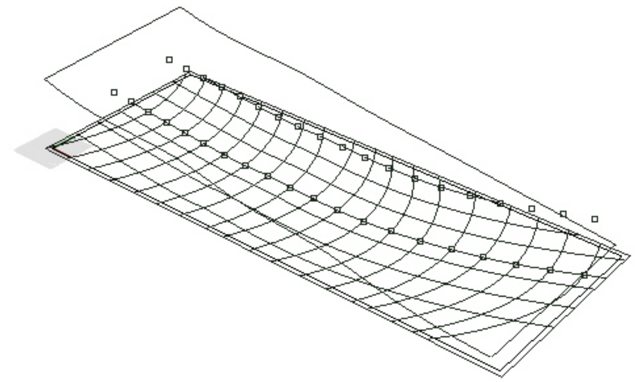
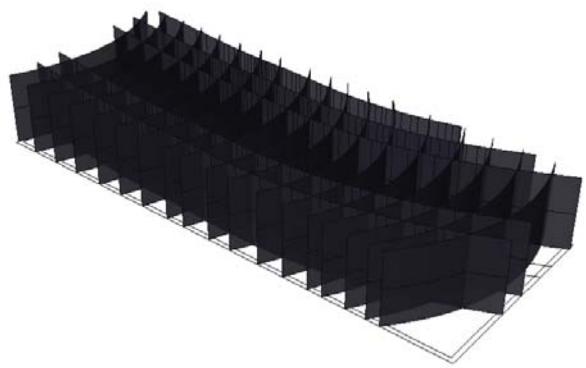
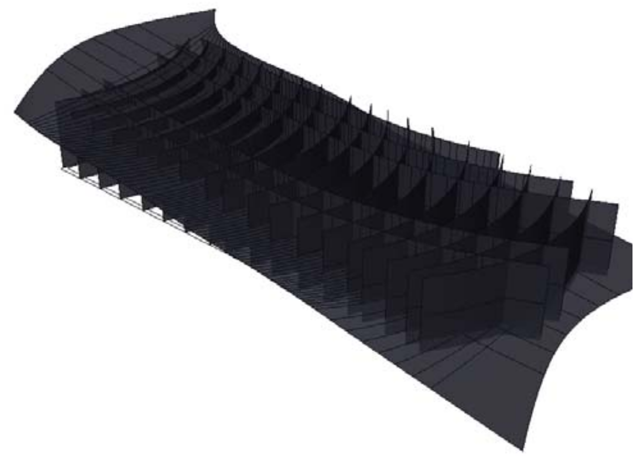
	
Beginning contour lines.	**Move** contours vertically [see Ex: PANCAKE] and draw a curve perpendicular to them. At the start of this line, a section is drawn of the large "vein" figure, which is the outline of a trench.
	
In order to properly extrude the figure along the perpendicular line to make the "vein", the line must be projected onto a base plane [see Ex : PANCAKE]. If the "vein" section were to be swept along the existing line, it would change vertically along the line, which is undesirable.	**Sweep1** is used because the section will be rotated perpendicular along the line as it is swept along it. Extruding would keep the section parallel to its original direction.

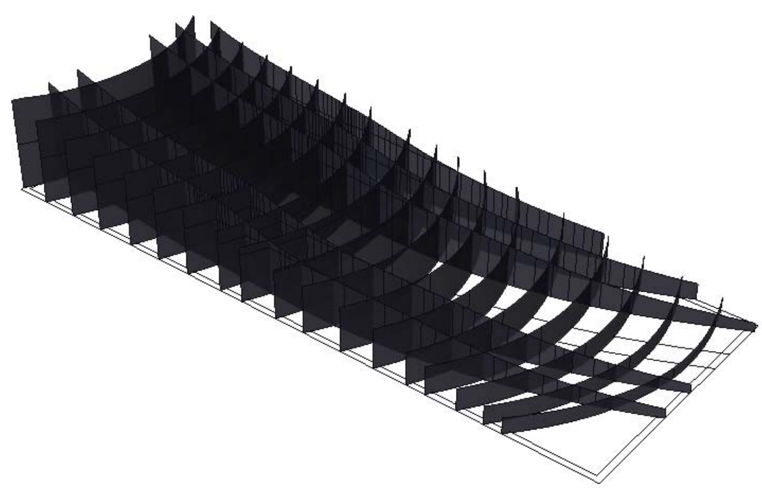
	The same "vein" section is copied, scaled and rotated to align with the contour lines, and then the same **Sweep1** technique is applied.

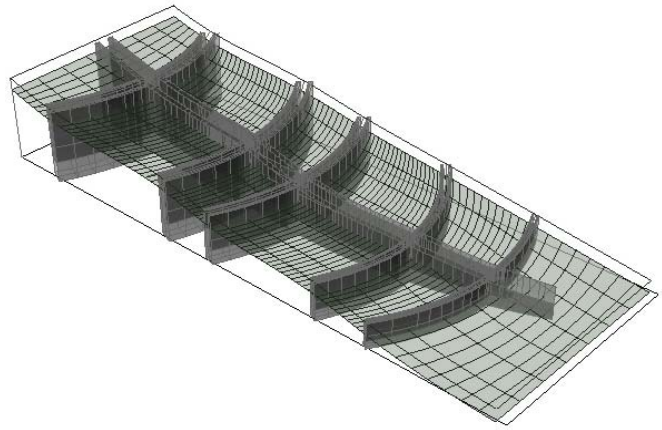
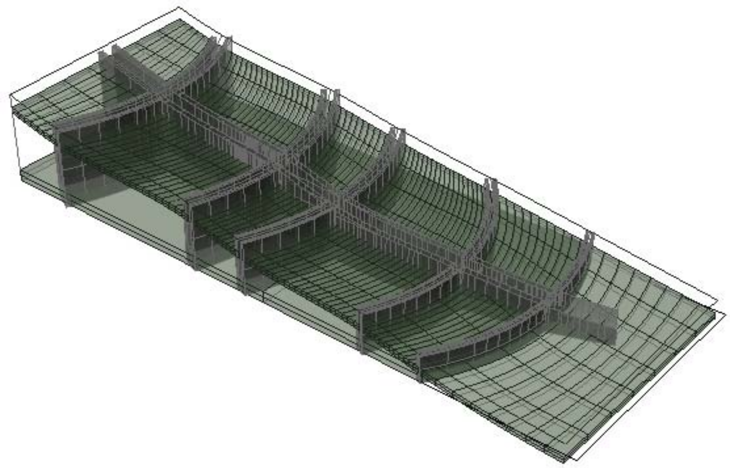
	
NetworkSrf creates a literal representation of the surface.	**Trim**. These veins will be the heaviest figure and will work to structure the urban fabric between them.

The next step is a 2d preparation for inserting a grid between the veins. The longer curves are **offset** from the large vein's curve. Points are drawn onto the end line using **divide** (divided by the same Length as the offset dimension).

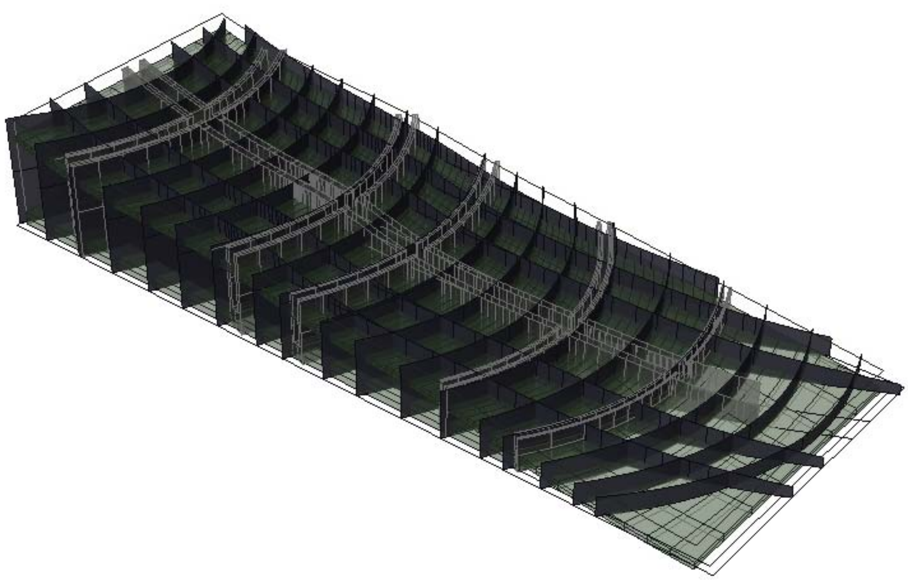
This is an exploration of two different ways the grid could be laid out:

	
Here the short lines are snapped perpendicular, but when they are extruded they do not relate to the smaller veins.	Here the short curves are derived from the smaller veins, copied, and rotated to average out nicely. This is preferred, so the curves are extended to a uniform edge in preparation for extrusion.
	
ExtrudeCrv	**ExtendSrf** (the same surface used to trim the veins).

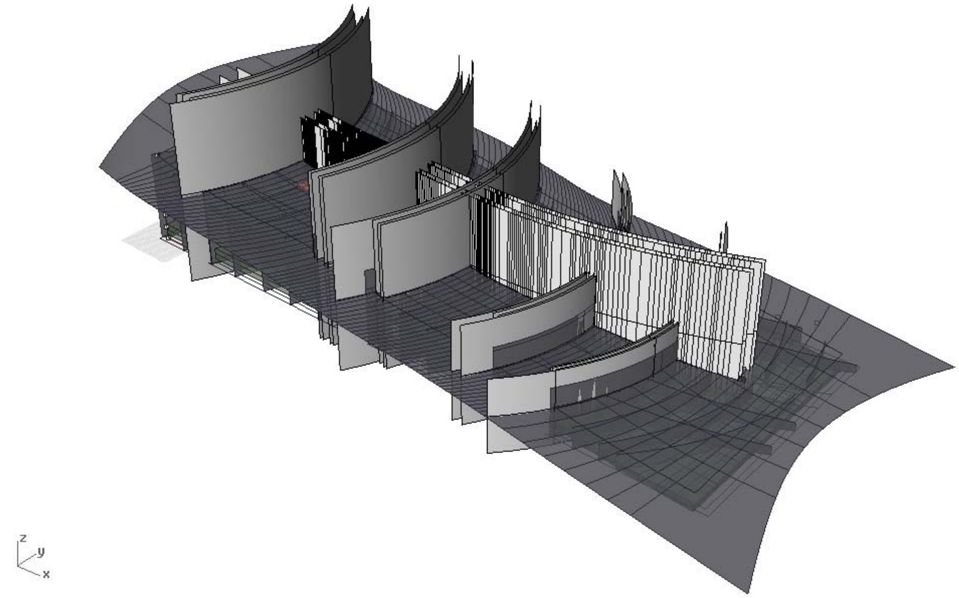
	**Trim**

	The next structural system made is the thick surfaces running between the grid. The original **NetworkSrf** is copied, moved, rotated and **ExtrudeSrf** is used to thicken.
	The two thick surfaces are joined to create polysurfaces and trimmed so they do not run within the vein trenches.

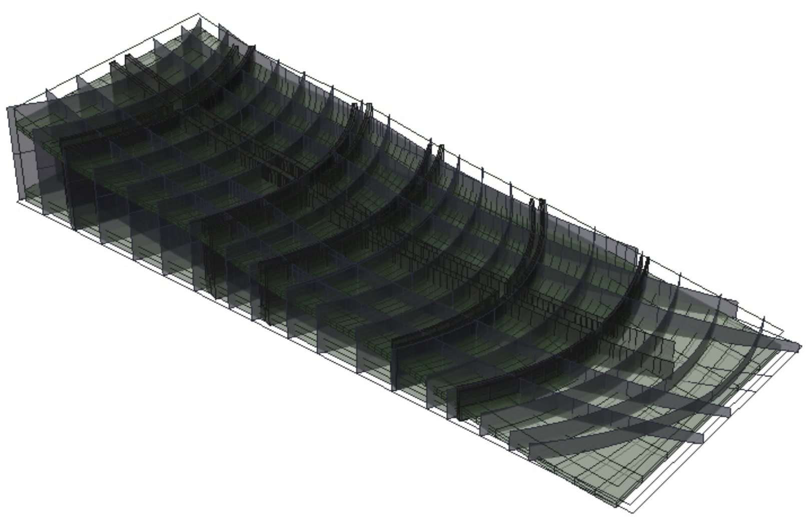
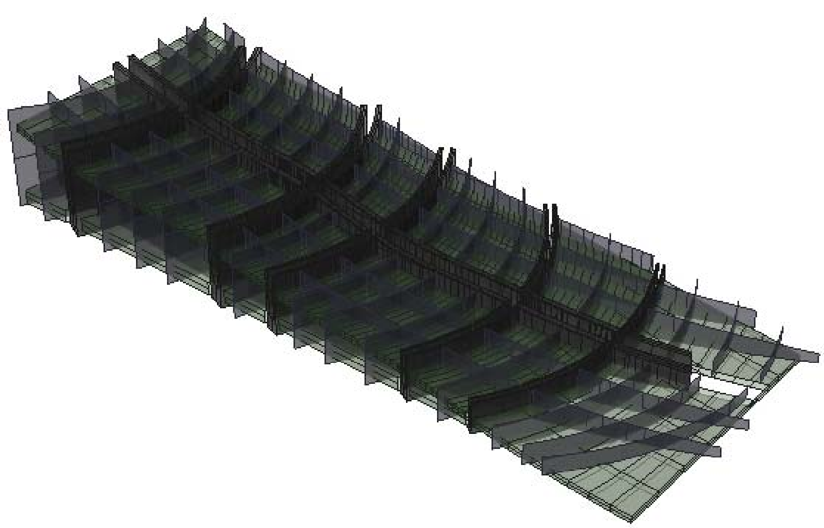
The layers are turned on to reveal the overlapping systems:



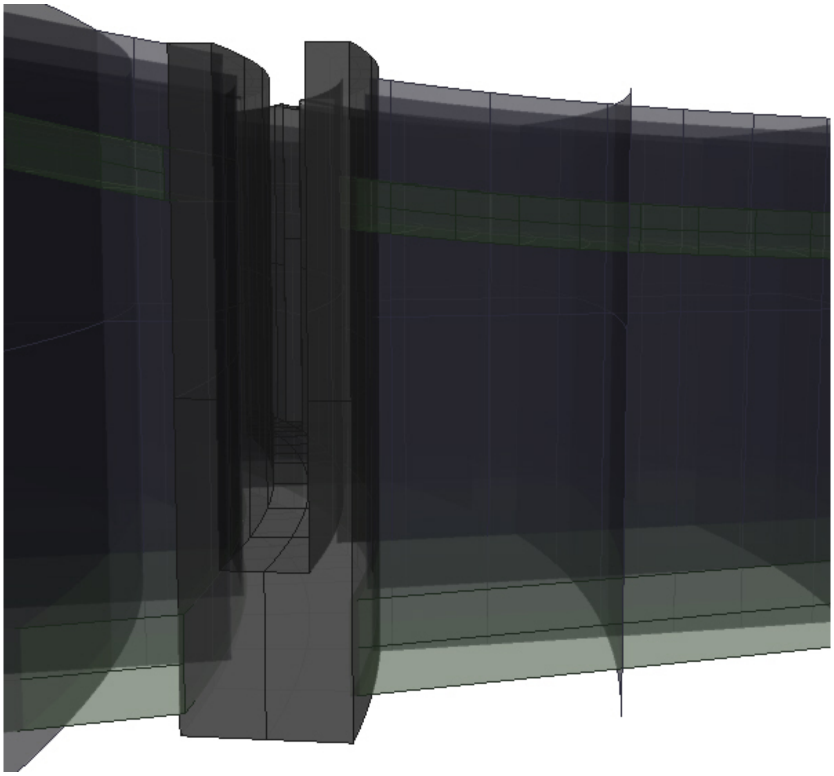
Meanwhile, if all of the construction planes, surfaces, lines and points are revealed:



But those are hidden again and the following edits are made:

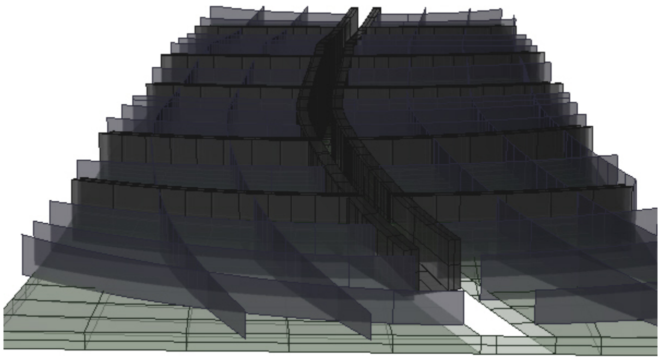
	Layer material and color information is changed to better describe the aforementioned hierarchy.
	All surfaces are trimmed and capped as appropriate, which makes a large impact on a smaller scale:

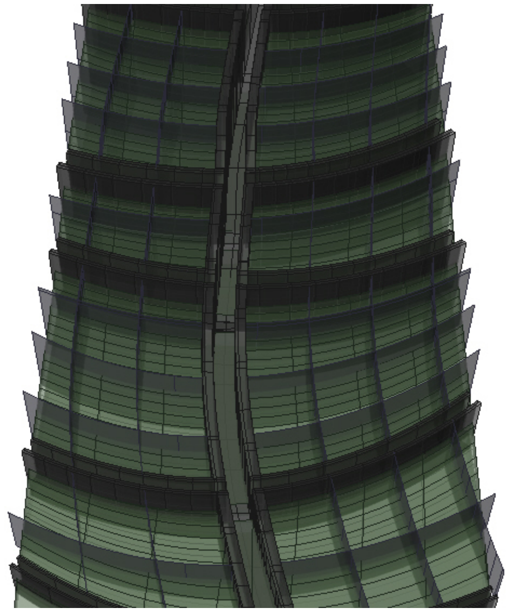
The thick landform allows interpretation for where the ground starts and where the city begins:



Within the veins of an underground water system, or above on the urban street:

Depending on the angle, the grid of planes closes or opens to light:





In an urban scene, perhaps the gridded planes are structural bearing walls between units, and perhaps they are supporting the heavier veins. A simple surface insertion.