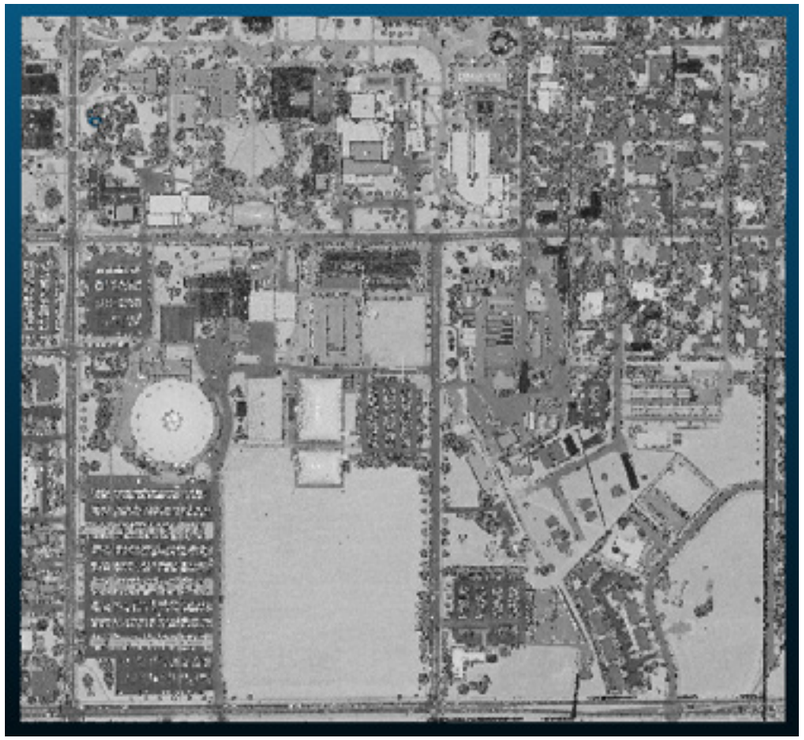
POINT CLOUD INTRO

Step 1

Find point cloud data from LIDAR/GIS website. (To be honest I forgot where I found this one. Most likely Montana LIDAR project). Cloudcompare uses .laz files.

Step 2

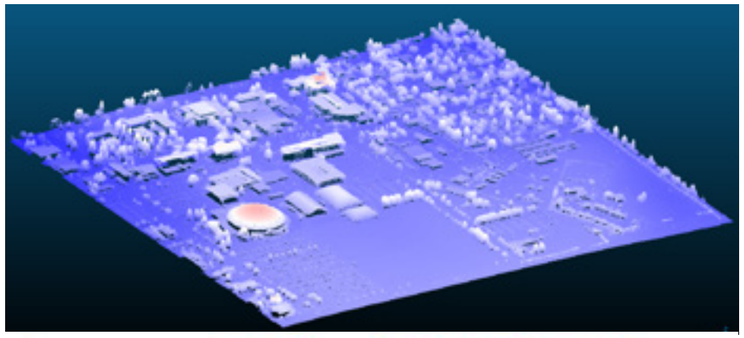
Import .laz file into cloud compare



Step 3 (optional)

Change colors based on height

- Tools > projection > export coordinates to SFs
- Edit > colors > height ramp



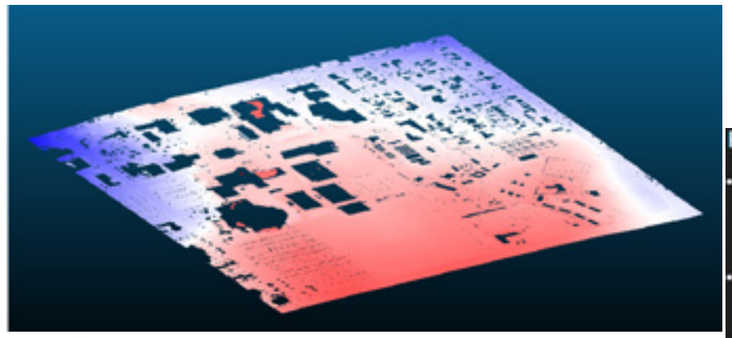
Step 4

Separate buildings and ground

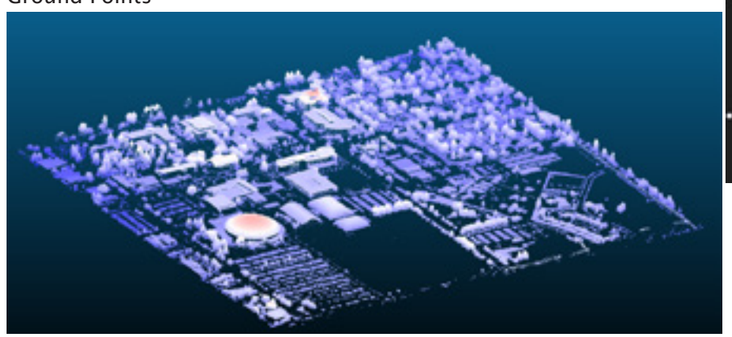
- Plugins > CSF filter
- Play around with advanced parameter settings

Export selection

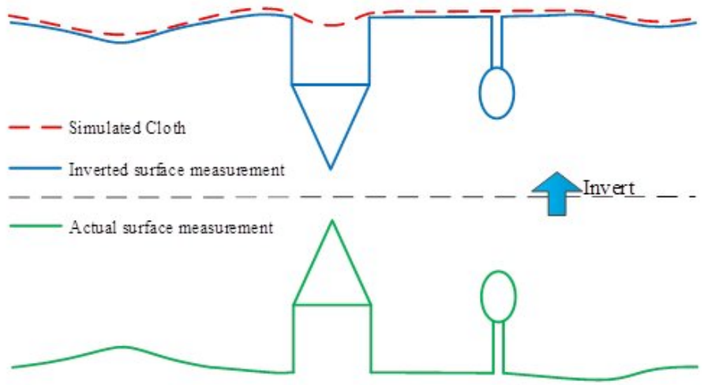
- Save current entity - .obj file most likely



Ground Points



Off-Ground Points



Cloth Simulation Filter Visual Description

How it works with the CSF Filter:

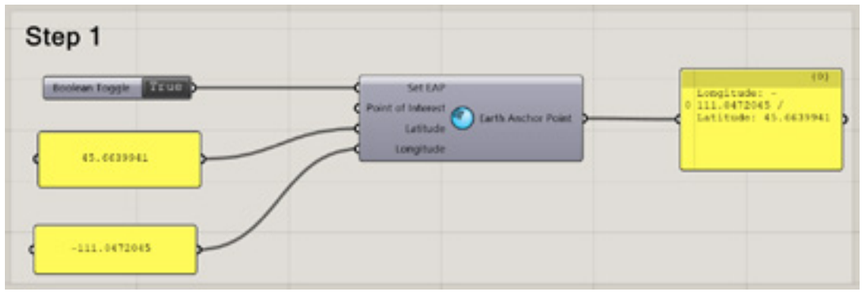
- **Ground Filtering**: The CSF simulates a virtual cloth draped over the point cloud. Points where the cloth touches are ground, while others (objects, steep slopes) are off-ground.
- **Steep Slope Handling**: For steep areas, the cloth might float above the actual surface, misclassifying ground points as off-ground.
- **Enabling Slope Processing**: Activating this option in the CSF settings makes the algorithm better account for these steep slopes, preventing it from discarding valid ground points on inclines, ensuring more accurate ground separation and detailed feature preservation (like ditches).
- **Adjusting Parameters**: You can fine-tune it with "Cloth resolution" (smaller = higher resolution, more detail, but slower) and "Classification threshold" (controls how steep a point must be to be classified as off-ground).

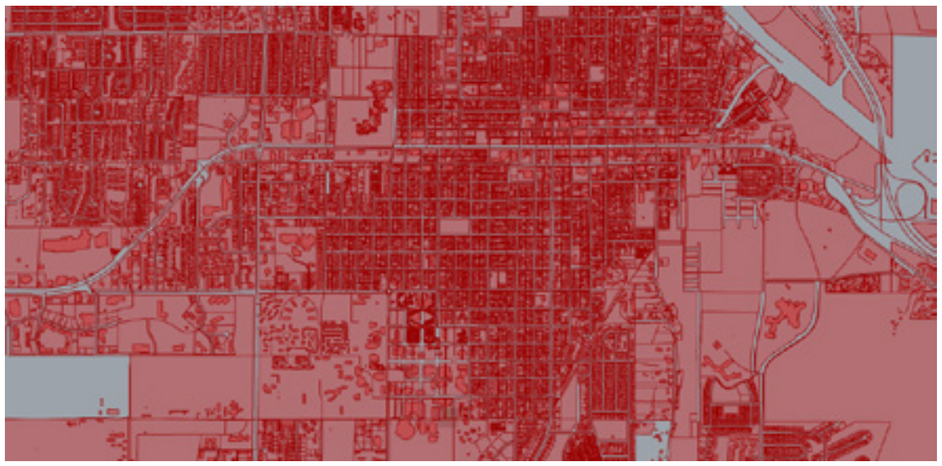
Slope Processing Description

GIS Site Formation

Step 1

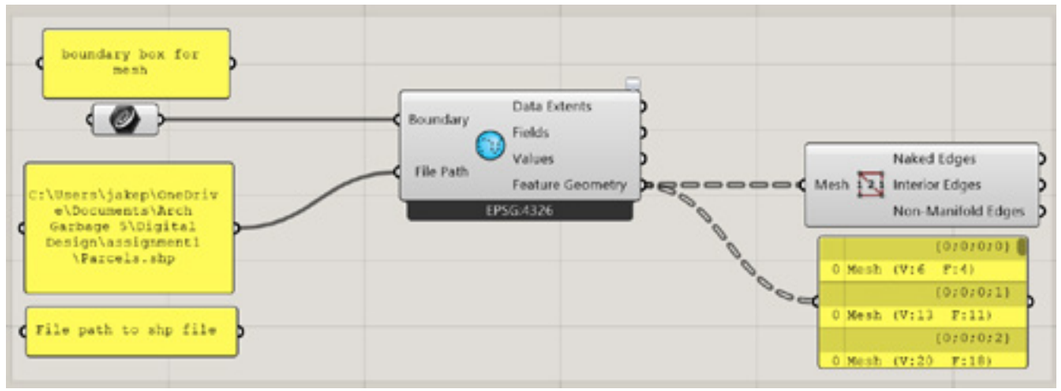
Use Heron to set an anchor point. Grab site coordinates from Google and plug them in



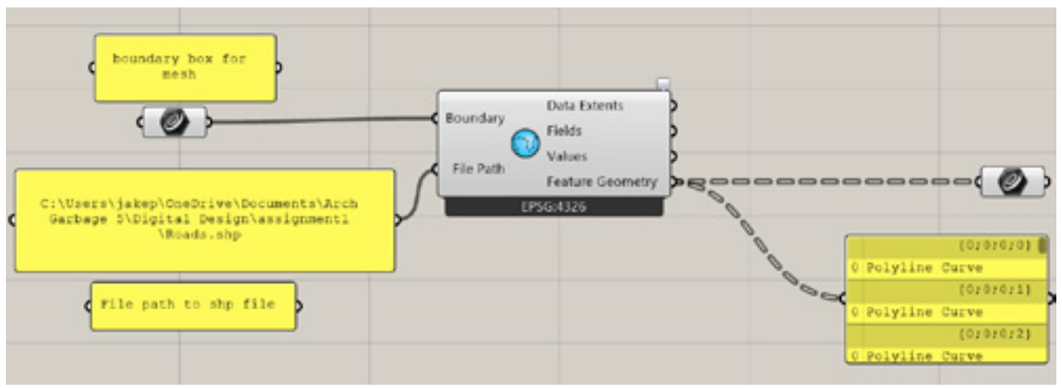


Step 2

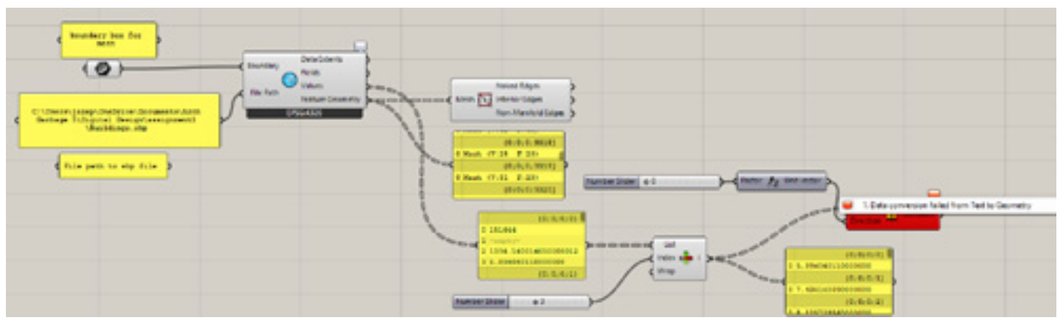
Graph GIS data from the web and save it to a file path. Use that file path to link the info to a Heron node. .shp files are accepted



Parcel Data



Road Data



Building Data + Extrusion Script I never Finished

DRAWING 1

TIME CHANGE

As time increased, more details about the branching nature were revealed. The location of bulbs and relations between them became more apparent in the later drawings

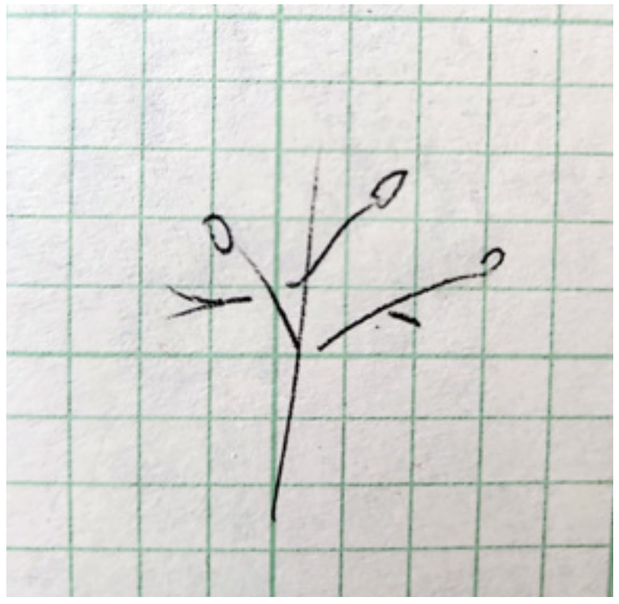
RESILIENT FEATURES

This branching structure was the main parametric feature. It is displayed in each drawing and becomes more intricate through the increase of time

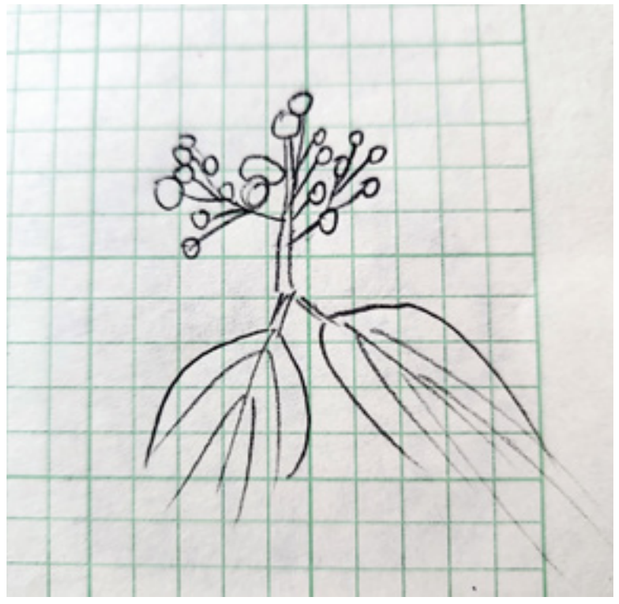
PARAMETRIC THINKING

This example may relate itself to growth over time. Something that changes over time introduces a 4D set of parameters

10 SECONDS

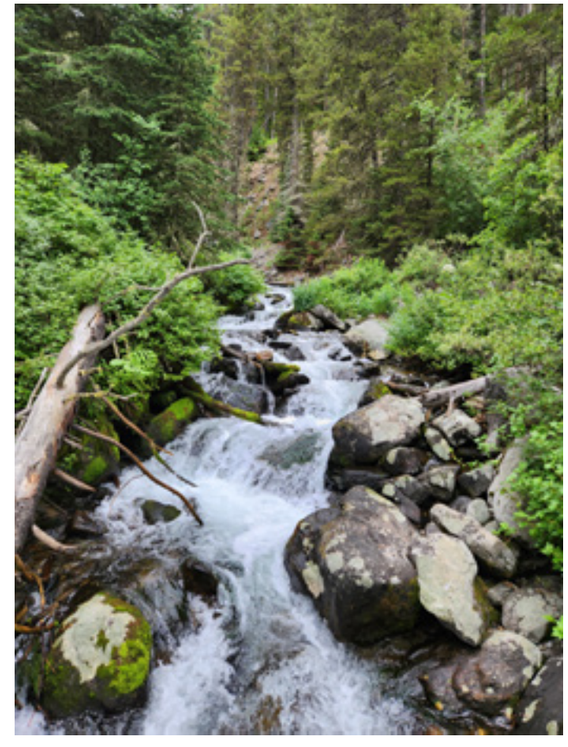


1 MINUTE



10 MINUTES

DRAWING 2



TIME CHANGE

As the time increased, I was able to include more detail. The parametric features don't necessarily show up in the first two drawings.

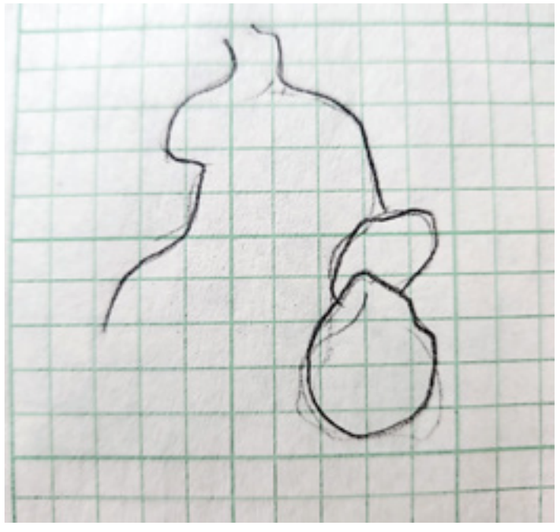
RESILIENT FEATURES

The flowing motion of the water exists in all drawings. The first two drawings show the outline of the stream, suggesting flow, but it becomes more apparent (hopefully) in the third drawing

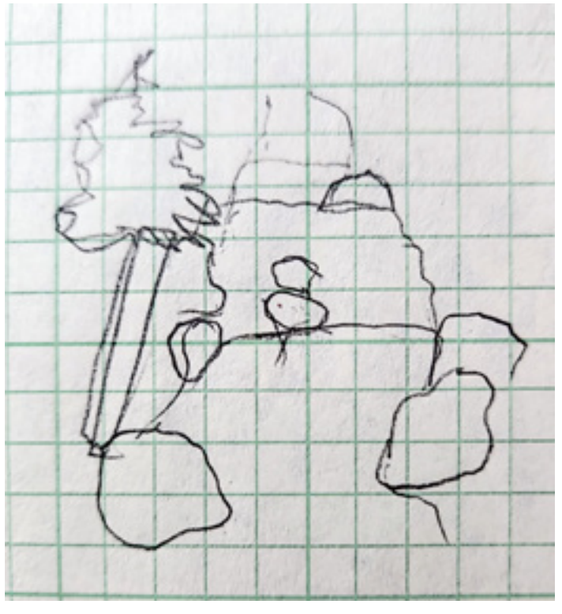
PARAMETRIC THINKING

Water flows according to physical rules. This manifests itself in a stream, which while not the first thing many people think about when discussing parametric objects, is almost the epitome of parametricism

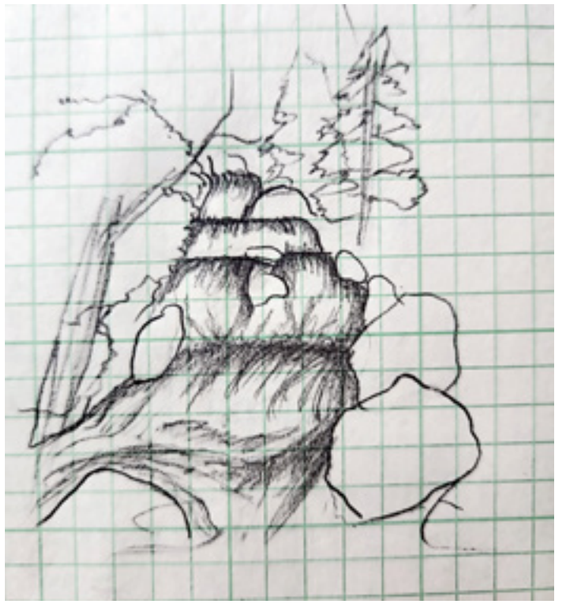
10 SECONDS



1 MINUTE



10 MINUTES



DRAWING 3

TIME CHANGE

Through the time change, I was able to include finer details of the parametric system. The pines on the cactus, for example, are another parameter that was secondary to the main divisions I included in each drawing

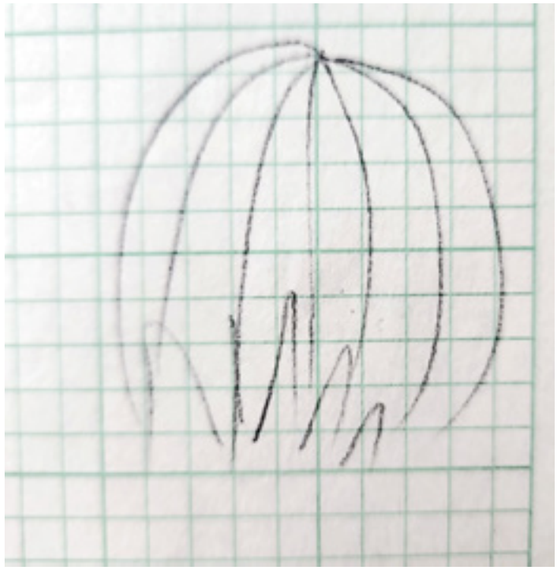
RESILIENT FEATURES

Throughout all the drawings, the features that were constant were the radial “fins” of the cactus

PARAMETRIC THINKING

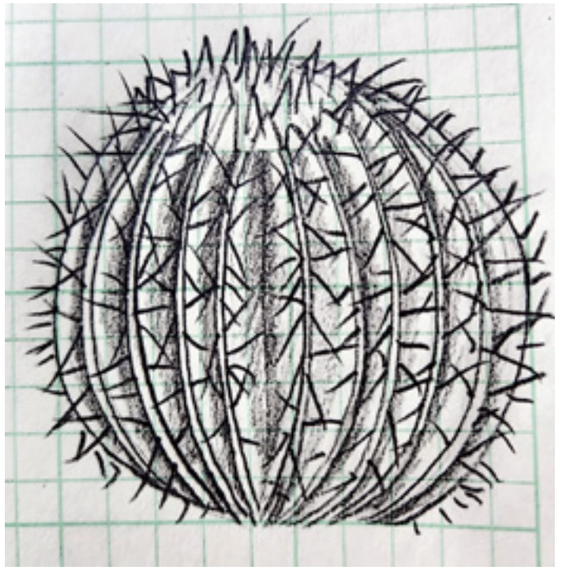
This may introduce the idea that there are primary and secondary features that our buildings represent. In this parametric sense, a relationship is formed between systems.

10 SECONDS



1 MINUTE

10 MINUTES



ASSIGNMENT 1 REFLECTION

Assignment 1 helped me realize that parametricism isn't an artificial construct; it can be seen everywhere in life and is the basis for many things in nature. Sitting down and looking over what I had seen on my hike (and the cactus I included) helped me notice the intricacies of the world around us

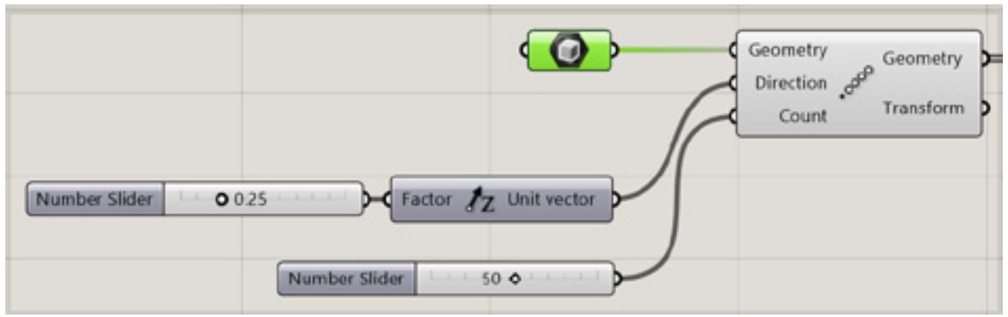
Taking a break from the chaos of school and daily life allowed me to think clearly about what I was observing, and I became more appreciative of the intricacies in world around me.

ASSIGNMENT 2

In this assignment, I wanted to test out different facade types that could elevate my building's appearance. This script in particular was an attempt at making a wave brick pattern.

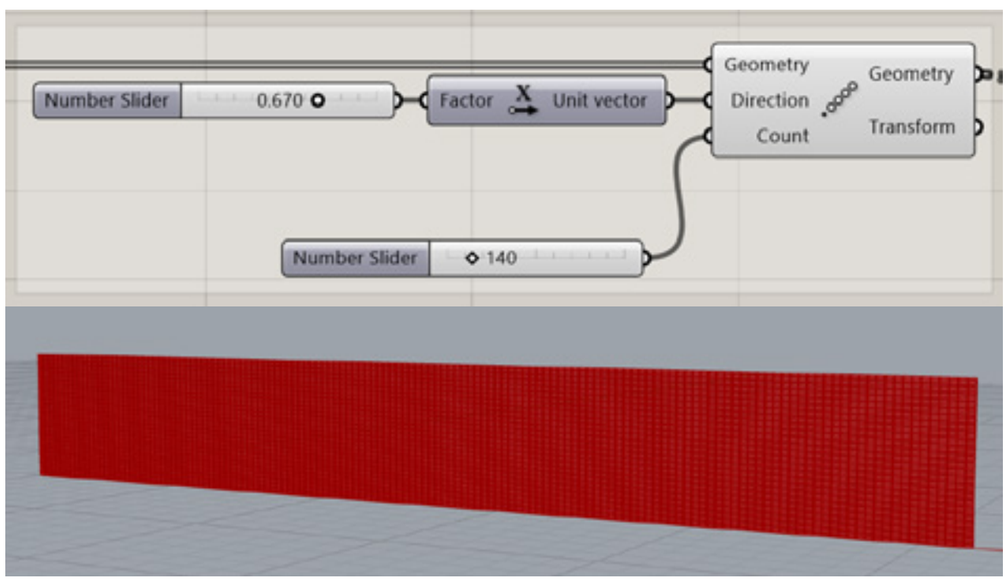
STEP 1

The first step was to create a singular brick (standard dimensions) and array it vertically, spacing each unit by a slight margin. This creates the vertical component of the wall.



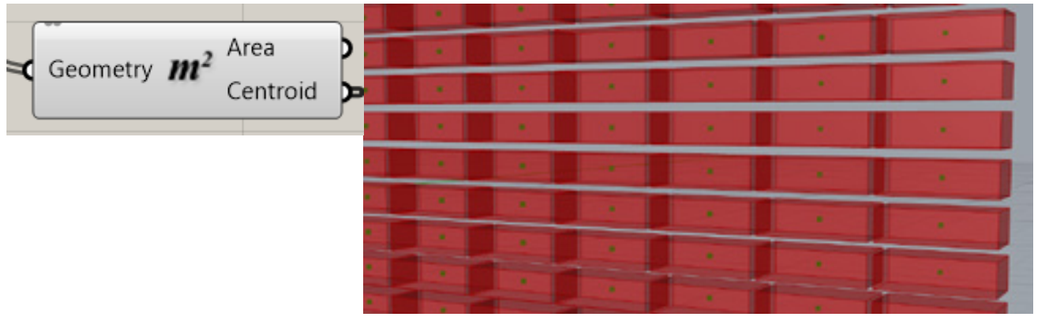
STEP 2

The second step was simply arraying the vertical portion horizontally to create the base wall



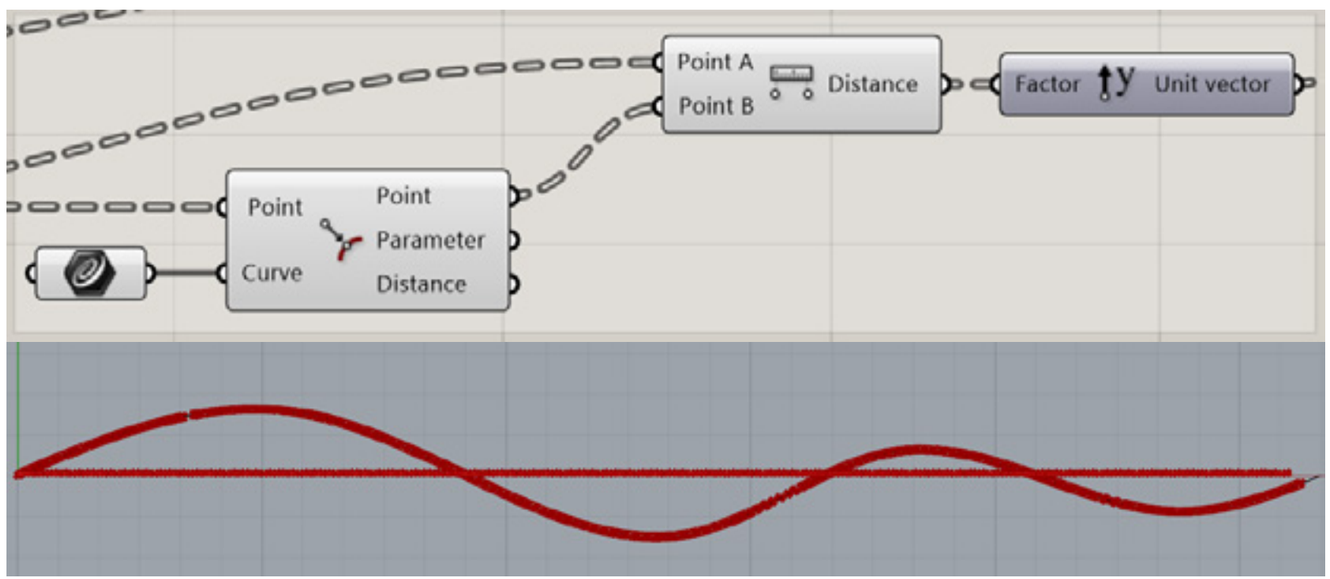
STEP 3

Excluded from the other groups, this node grabbed the centroids from each brick so that there was a point from which to move them.



STEP 4

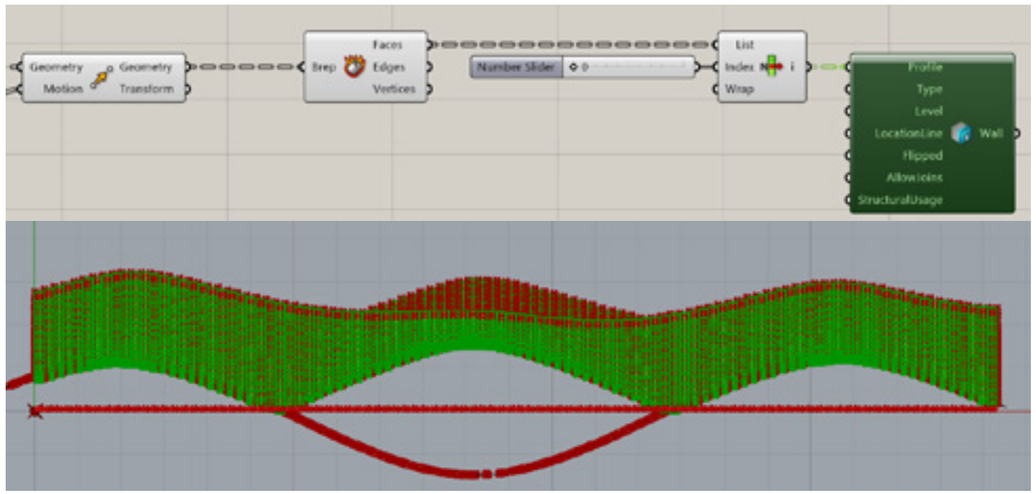
Step 4 takes a curve and uses the points on it to offset the bricks, from their centroids, according to the distance they are from the closest point on the curve



ASSIGNMENT 2

STEP 5

Step 5 takes the offset bricks, compiles them in a list, and then takes the outer face of each item in that list. That face is then used for the profile of a wall component in Revit, successfully transferring the elements into native Revit items

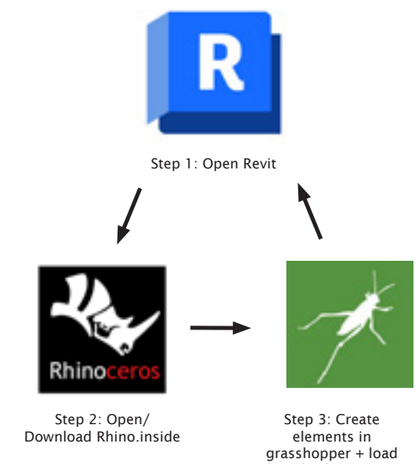


SUMMARY

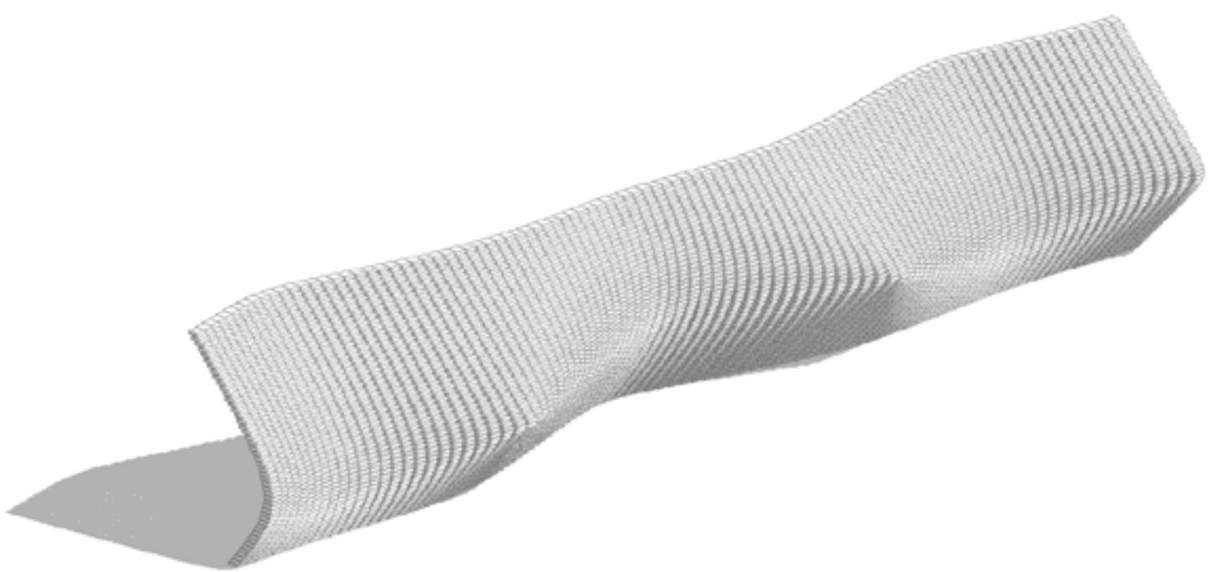
One of the biggest design choices I've had this semester is finding a means to turn my southern facade, which currently all glass, into something that is more thermally insulating, but remains visually appealing (i.e. not punching holes in the wall for windows). One of the suggestions that I have received was to use brick and create a pattern with how it is laid. This was a conceptual approach on how I might do that.

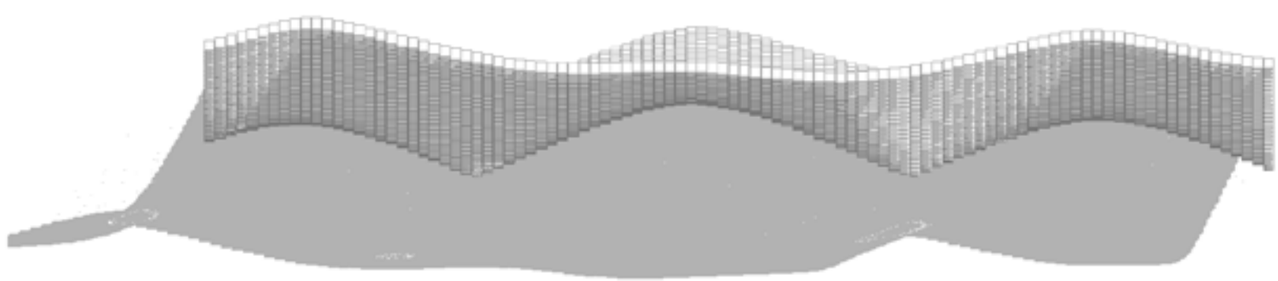
Using Rhino.Inside, I learned how one can modify Revit components once they understand the parameters that are required. This will allow me to find solutions that I run into later regarding Revit's strict design features.

Workflow diagram



FINAL PRODUCT





ASSIGNMENT 2 REFLECTION

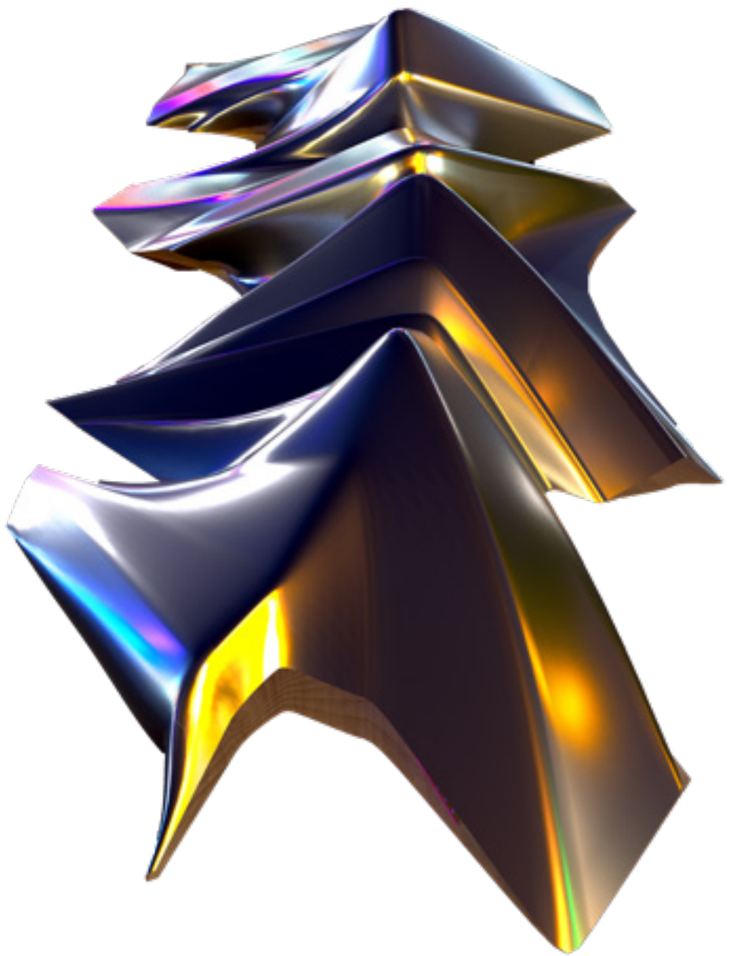
Assignment 2 showed me how I can overcome the roadblocks in software, especially in Revit. While the interaction between Rhino and Revit was the predominant one, this assignment gave me an understanding of how components from each program work in others. These interactions create a map that I can follow and use to transfer work between software, playing to the strengths in each.

ASSIGNMENT 3

DIGITAL MODEL

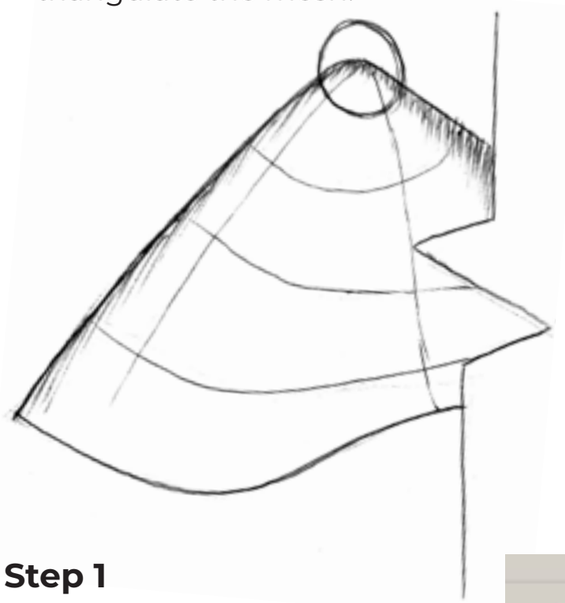
Project Selection

A project in my portfolio that has always bothered me was one from a third year studio. While I like the shape, the design itself lacks any thought about structural elements. For assignment 3, I wanted to change that. By utilizing what I know now about digital modeling and grasshopper, solutions made much more sense that before. Since the building was spanning long distances below this form, a space frame design was ideal to work with the unusual shape and large spans.



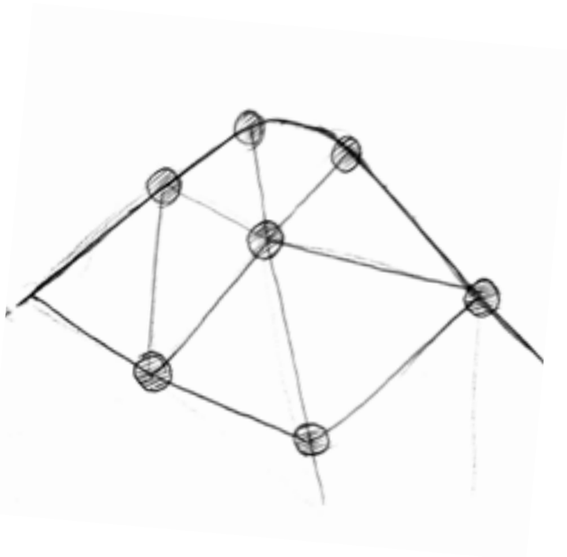
Initial Dilemma

The main problem with attaching a calculated structure to a shape that is so non-calculated is that the two don't work together. A space frame needs points to connect to, but this mesh seemingly had none. When bringing this question up, it was suggested to me that I triangulate the mesh.



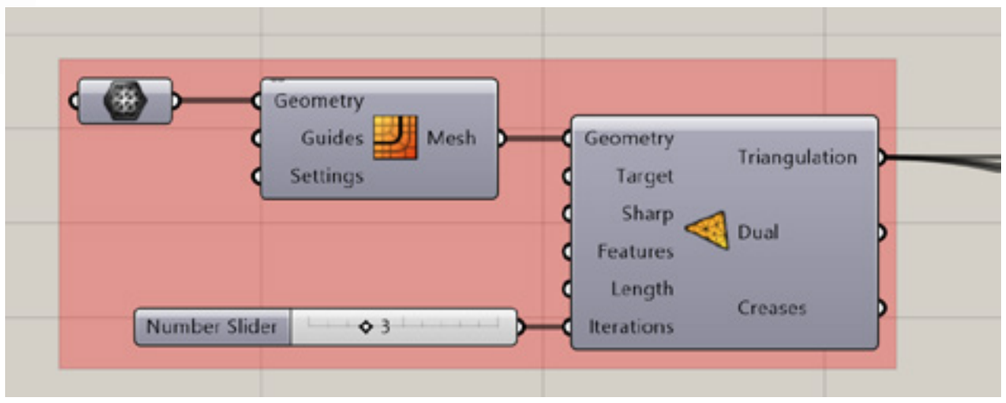
Initial Solution

By triangulating the mesh, it would be turned into a series of triangles. These surfaces have geometry that rhino can work with and were the base of my solution.



Step 1

The first portion of the grasshopper file takes an input mesh and reformats it into a triangulated mesh. This creates the base geometry for the rest of the program

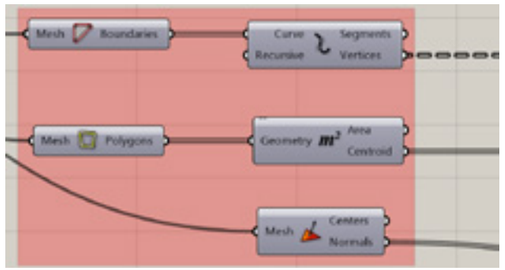


ASSIGNMENT 3

SCRIPT BREAKDOWN

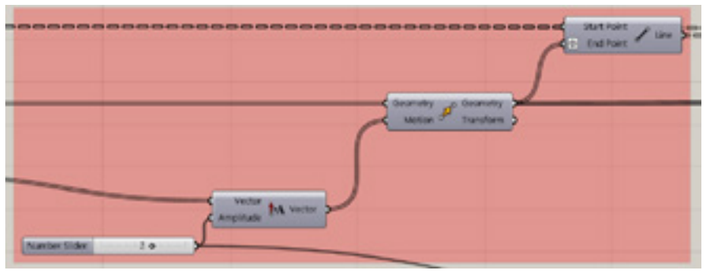
Step 2

The mesh produces boundaries and polygons, which are turned into the spanning structure and centroids respectively. These centroids are then offset in a normal vector



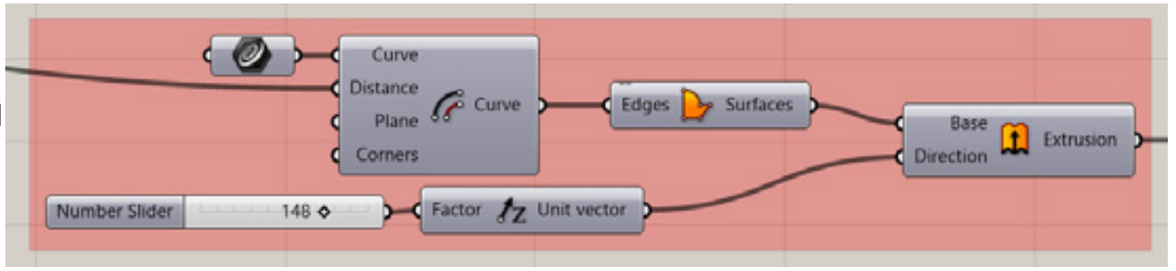
Step 3

The mesh produces boundaries and polygons, which are turned into the spanning structure and centroids respectively. These centroids are then offset in a normal vector



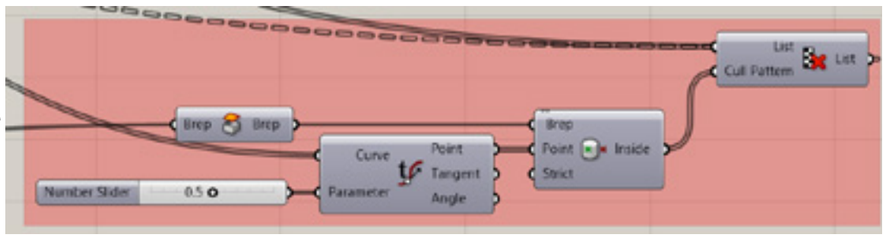
Step 4

To cull any unwanted lines, a curve is created from a flattened version of the mesh. This is offset by the same distance as the centroids are. It is extruded to surround the shape



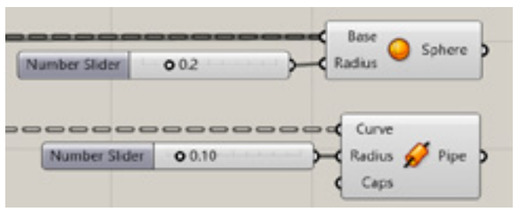
Step 5

By sorting the list of lines and culling the ones that are outside of the extrusion, all unwanted lines are left out of the final structure

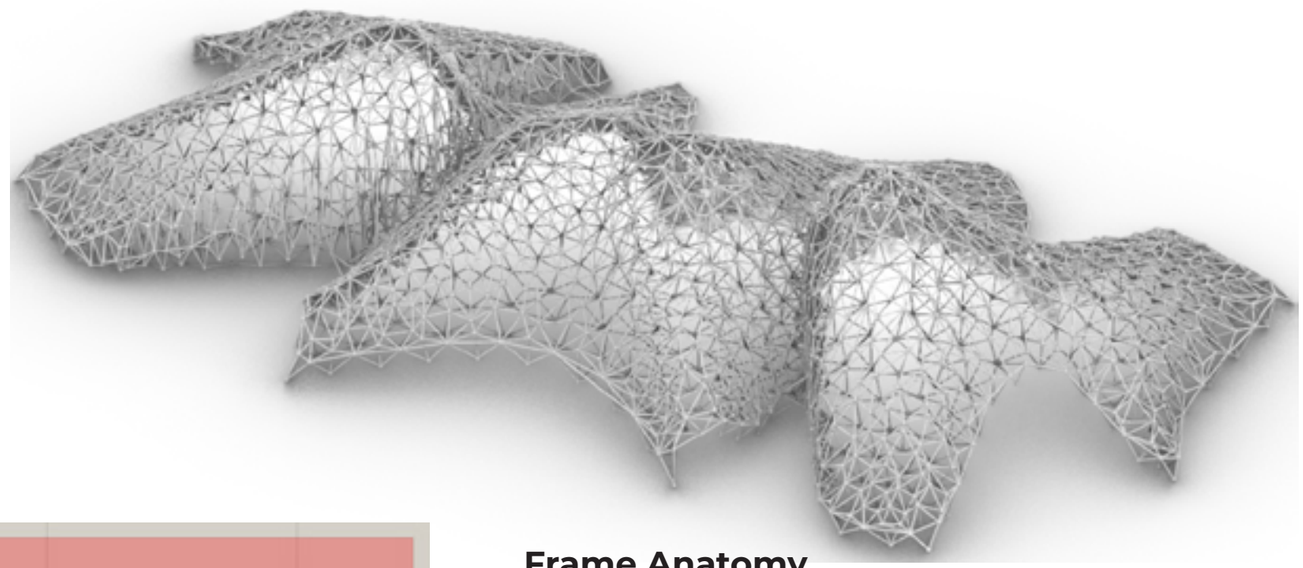


Step 6

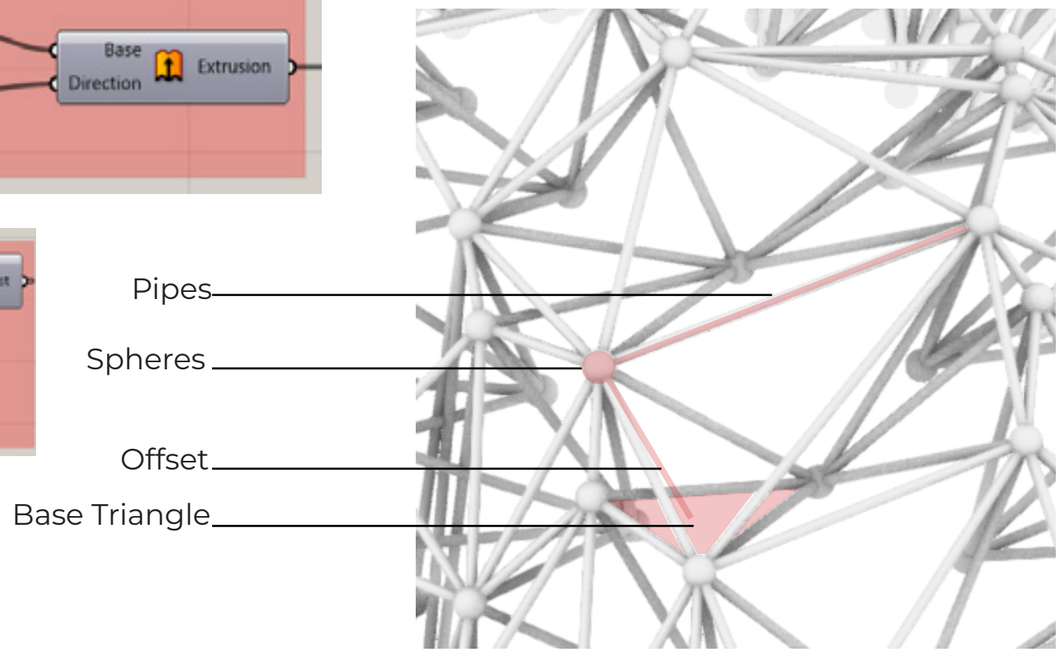
The final output are spheres, placed at the intersections of the pipes for connections, and the pipes themselves



Base Result



Frame Anatomy

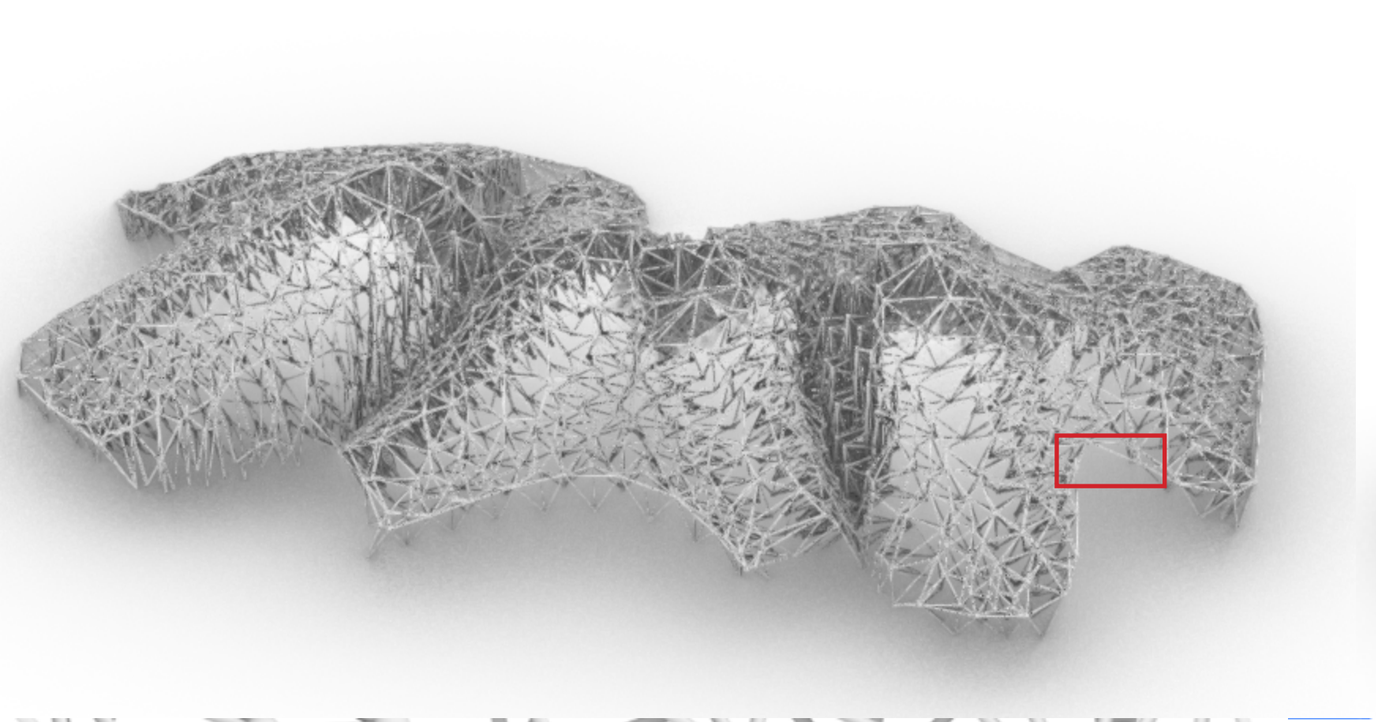


ASSIGNMENT 3

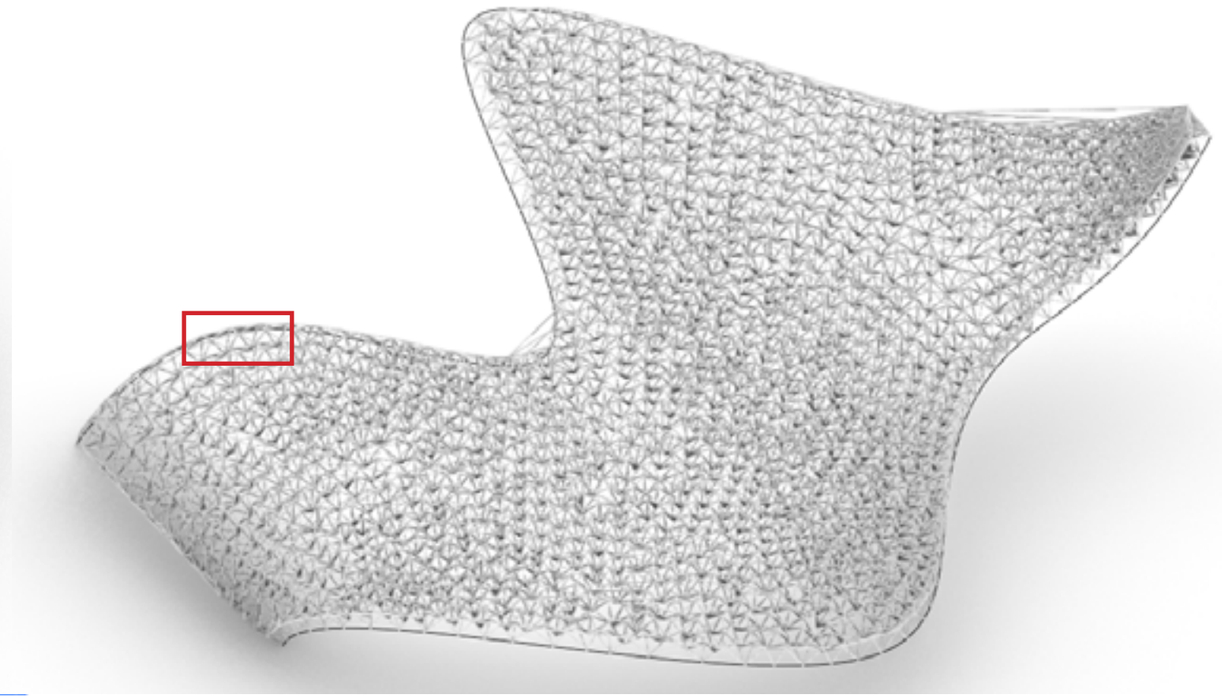
TAKEAWAYS + ADDITIONAL EXPERIMENTS

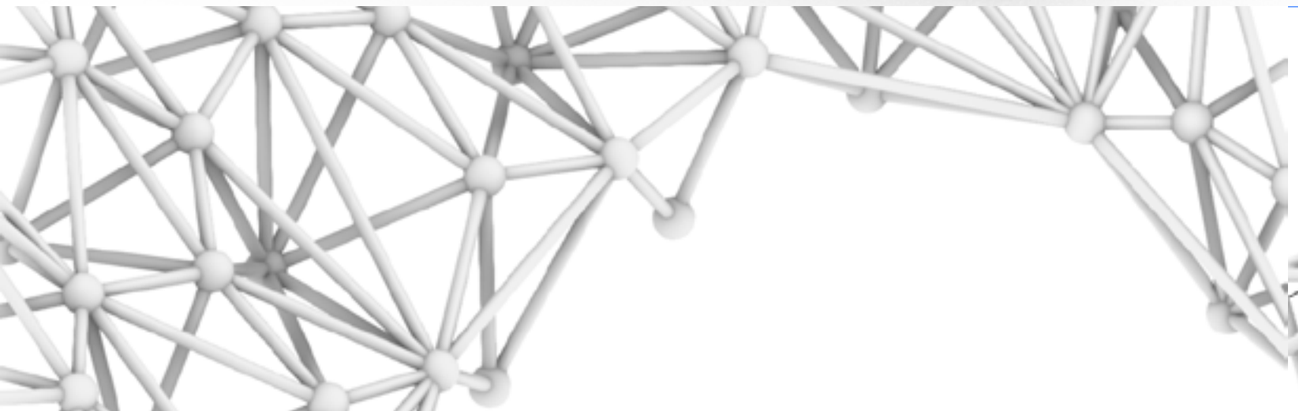
This project showed me the importance of establishing variables in old projects in order to create parametric designs. Making a parametric system for a design which was meant to have parametric features added to it works well. When retrofitting an old project with new parametric features, it is important to understand what variables you have to work with. I thought this mesh was incompatible at first, but after having conversations, I realized that there were in fact variables (vertices, faces, and edges) hiding in the mesh that I had to look for more closely. One can create something so long as they have basic components to work with.

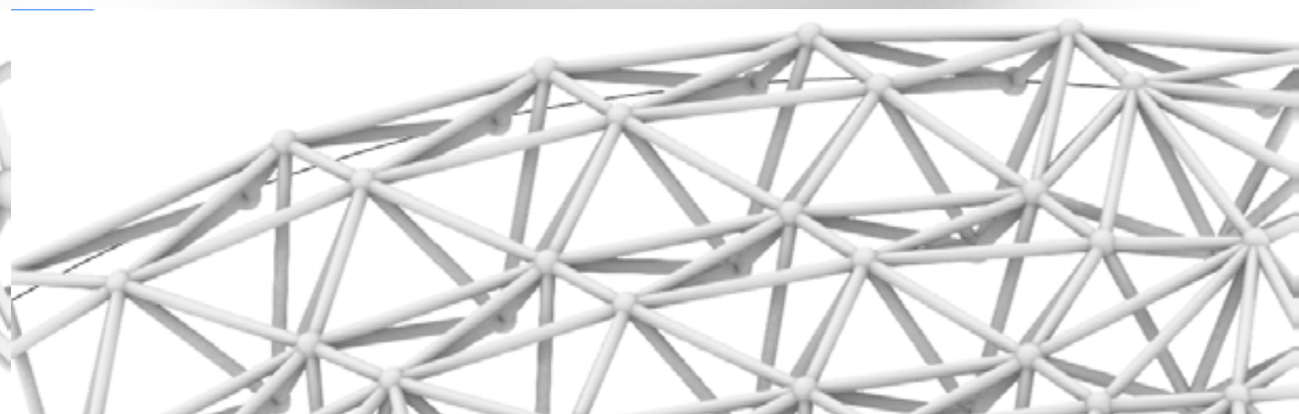
Increasing the Offset Distance (Too Much)



Substituting Other Meshes in Script







ASSIGNMENT 3 REFLECTION

Assignment 3 showed me in particular how to retrofit a project with new information. When I made the original form in the assignment, I had no idea how the interaction between softwares worked. Again, by finding the shared qualities between them (Maya and Rhino in specific) I was able to conjure a script that bridged the gap between the two.

Creating a parametric design only requires you to understand the variables that you are working with. From there, you are able to make anything that utilizes those variable.

ASSIGNMENT 4 REFLECTION

Assignment 4 helped me better understand how to use assets available to me to create engaging imagery. TwinMotion in particular was very useful, as I hadn't been able to find a render engine that both offered me assets, created a vivid rendering at the end and was free. Tweaking the settings to adjust the mood and feeling of each rendering helped me better understand how each variable added to the final composition.

RESOURCES

RHINO

- Cloudcompare - free + open source pointcloud software
- McNeel Europe - Youtube - https://www.youtube.com/@McNeelEurope
- Chris' Youtube channel - https://www.youtube.com/@motus406
- Rhino API - for coding - https://developer.rhino3d.com/api/rhinocommon/

REVIT

- BimPure - website - https://www.bimpure.com/
- Gediminas Kirdeikis - Youtube tutorials - https://www.youtube.com/@DesignGoBrr
- Parametric Monkey - software collaboration - https://parametricmonkey.com/

RENDERING

- Adobe 3d assets - program/assets
- Epic asset library - assets
- Cesium for Unreal - plug-in
- Gediminas Kirdeikis - Youtube tutorials - https://www.youtube.com/@DesignGoBrr

GRASSHOPPER

- Generative landscapes - website
- David Rutten - Grasshopper developer - https://ieatbugsforbreakfast.wordpress.com/
- Daniel Piker - Kangaroo developer - https://spacesymmetrystructure.wordpress.com/
- Hydrashare
- Ladybug fourums - https://discourse.ladybug.tools/