# BRIDGE THE GAP

7ZM7M0 | Parametric Design


M. Geraedts – 1380303
M. Goudswaard – 1337149
V.G.R. van Oss – 1363360

# Contents

# 1. Introduction

This report gives insight into the design of a structure that enhances an event that takes place in Eindhoven on four different squares (Stationsplein, Lichtplein, Ketelhuisplein and Augustijnenkerkplein). The main goal of this assignment is to parametrically generate design options that can be altered for each different location with just a few adjustments. The parametric design of the components is done in Grasshopper and by combining Rhino, the structures are visualized.

The idea is to design a temporary structure that can facilitate this event and make sure it will create a nice ambience during the entire event. With that, the event needs to be resistant to the Dutch weather, which includes rain.

Next to these requirements, there are a few standards that the structure will need to suffice. The first one is that it does not need any demolition on the squares, like for example buildings and street furniture. With that, the structure can only occupy a maximum of 50% of the area. And to support the performers, a backstage area with sanitary facilities needs to be added to the solution.

To make sure the event will be visited, the focus is on the mobility of both pedestrians and cars. For motorized mobility, the goal is to keep this running. Meaning that there will be zero to limited interference with the roads. For unmotorized mobility, like pedestrians and cyclists, the idea is to intrigue them to visit the event. This can be done by for example ‘blocking’ standard routes or introducing new and exciting ways of travelling through the squares.

Since the main theme of the design is mobility, an analysis is conducted on the four different squares. This analysis contains the space that we can use with the degree of freedom, as well as where the main routing is on these plots. It can be seen that for Stationsplein and Augstijnenkerkplein mobility plays a key role, see Figure 1. Two main roads cross the plots. These roads are also main roads that will lead traffic further into the city center and are used by public transport. This introduces an extra requirement for the design, the busses will need to be able to still use the roads and thus a height requirement for the bridge is introduced. A normal bus is around 3.5 meters high (Wikipedia-bijdragers, 2023), and to give some free space between the bridge and the busses to avoid any accident happening, the bottom of the bridge needs to be at a height of 4 meters.

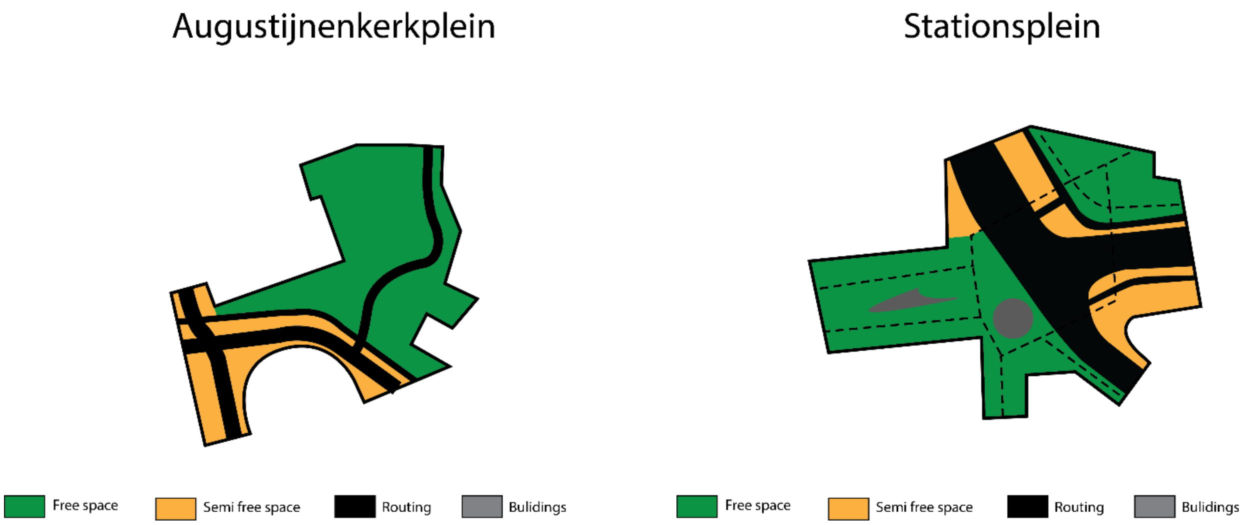



**Figure 1: Square analysis of Augustijnenkerkplein and Stationsplein**

The other two squares have less dominant routing through them. Lichtplein, see Figure 2, is only used by pedestrians, since on the yellow part of the plot, a parking garage is situated. For Ketelhuisplein, see Figure 2, some routes going through the plot that is used by traffic that needs to be in the area.

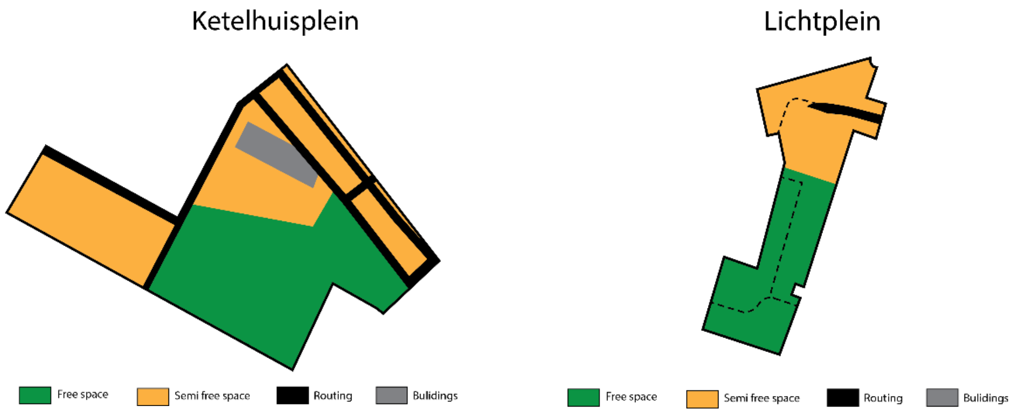



**Figure 2: Square analysis for Ketelhuisplein and Lichtplein**

With the use of the analysis, requirements and focus on mobility, the idea of using a bridge as a way to promote mobility, as well as seating for the event, is created. This bridge will allow keeping the current traffic flows and simultaneously creates an interesting way of transportation for pedestrians. Also, this will create a way of walking that is not obstructed by crossing, but will keep the traffic flowing. To promote pedestrian traffic throughout the city for environmental and health purposes, the bridge is covered to allow the event to take place in all-weather circumstances. The bridge will be covered using a tent-like structure. This tent-like structure also keeps the event structure temporary and demountable. However, to keep the traffic flow going, only the seating area on the bridge will be covered, and the top part of the bridge will remain partly uncovered.

The first step in the design process is determining where the bridge is placed, together with the dimensions of the bridge. After deciding on the placement and scale of the bridge, tent and stage, the exact points of the location can be chosen, this is explained in Chapter 2. Afterwards, Speckle is used to 'send' the points of placement of the structure to the script that generates the bridge. The design of the bridge is explained in Chapter 3. When the design of the bridge is finished, the tent structure can be placed. This tent structure depends on the anchor points on the bridge, shared with the use of Speckle, as well as the placement of the stage, which is also shared with Speckle. The design of the tent structure can be found in Chapter 4.

When all of this is done, the complete design is placed within point clouds of the squares, which were provided to us. For the square that fits the compass of the design best, the design is more detailed and a complete list of components is given to show that the bridge can be shipped to the site and manufactured.

# 2. Placement of structure

As explained, the main component of the design is the bridge. This bridge has to be placed strategically to avoid clashes with the roads and current buildings and greenery. With the help of the square analysis done in the Introduction, the bridges can be placed strategically. If the bridges are placed correctly, they can promote both pedestrian movement as well as the event. If a bridge is not placed correctly it will look like a random object on a square. It will help the event probably a bit, however, when placed in the best spot, for example on a route used by pedestrians, it will allow more people to visit the event.

Since the event is a people-focused place, with music and a nice ambience, more people will decide on walking through the city and visiting each spot, instead of taking the car everywhere (Lambert, 2019). This also allows the creation of a walking route like Glow. However, in this case, instead of using a light show to intrigue the public to start walking through the city, temporary bridges can be placed in key locations in the city.

The two squares that have the most straightforward placement of the bridge are Augustijnenkerkplein and Stationsplein, see Figure 3. This placement is based on the roads crossing the plots and the placement degree of free space on the plot. It becomes clear that for both plots the structure is placed with the idea that it spans a road and it stays within the free public space.

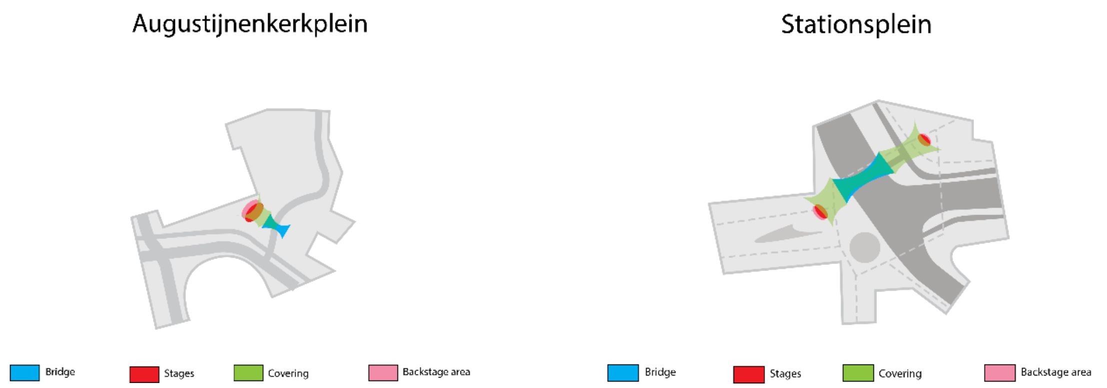



**Figure 3: Placement structure on Augustijnenkerkplein and Stationsplein**

For Stationsplein, two stages are placed on either side of the road, this was mainly done because of the freedom of the space and to demonstrate that all types of configurations are possible when looking at the bridge as a center point. Now, depending on the size of the event, the same kind of structure that is now placed on Stationsplein can also be placed on Augustijnenkerkplein. This way, there will be two stages present in the plot. With that, there is another option of promoting both walking and the event on this plot. This can be done by adding a bridge on the lower right side of the plot, see Figure 4.

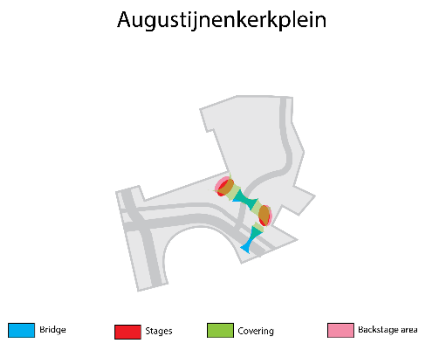



**Figure 4: Extra option for placement structure Augustijnenkerkplein**

For Ketelhuisplein, there is quite a big area that can be freely used, see Figure 2. And with that, there is a small road crossing the square. This meant that the placement of one bridge would be pretty obvious, this can be seen in Figure 5. The other bridge is now placed on a semi-free square and placed on a spot where it is more likely that people will enter the square. Another way of placing the bridge would be to focus on the view on top of the bridge, however, it is decided not to focus on this parameter now.

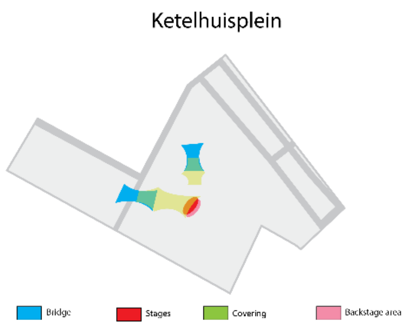



**Figure 5: Placement structure on Ketelhuisplein**

The last square is Lichtplein, for this one the bridge cannot be used for crossing main roads. However, it can be used to 'block' the main routing in the square and force people to use the bridge and simultaneously enter the event, see Figure 6. Also, only one bridge can be logically placed on the bottom of the plot in the free space, see Figure 2. This is because of the parking garage underneath the upper part of the plot. This parking garage is already incorporated as an incline with seating placements on it. This means that this area can be used for seating when the weather allows it. If the organizers of the event want this seating to also be incorporated into a tent structure, this is could be possible.

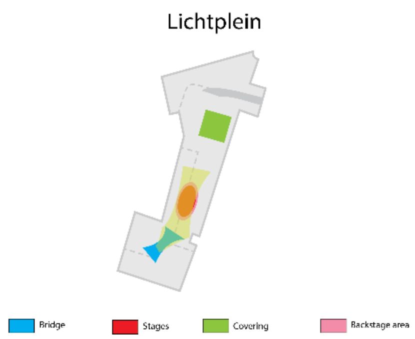



**Figure 6: Placement of structure on Lichtplein**

# Workflow & script structure

With the rough placement of the bridges and stages, the final placement of the plot can be determined. This is done with the help of Grasshopper and Rhino. The rough steps of the design can be found in the flowchart in Figure 7. The first step was creating the plot shapes in Rhino so that the correct data can be extracted. This was done by importing the plot shapes drawn in AutoCAD, which include the important roads on the plot and the buildings on squares where needed.

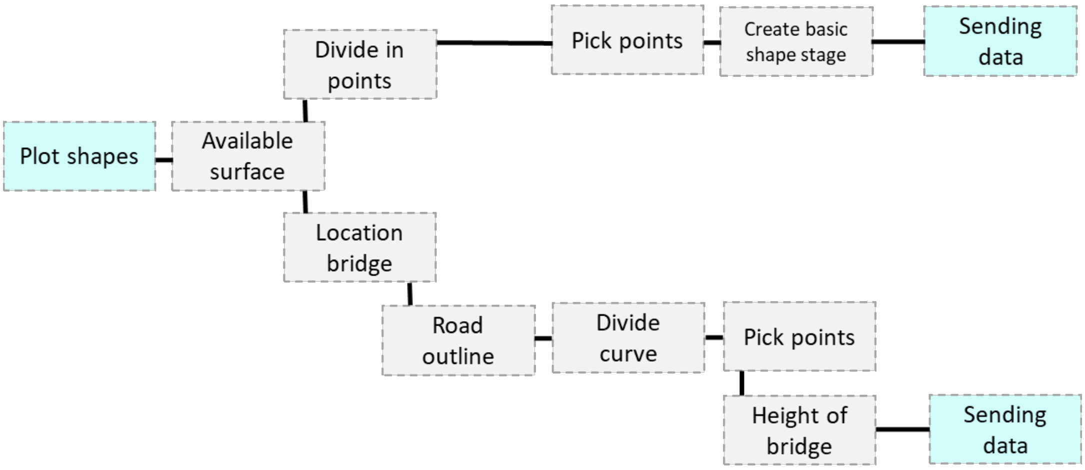



Figure 7: Flowchart of the process of placing the bridge in Grasshopper

When this entire process is followed in Grasshopper, the result looks something like this, see Figure 8. In this figure, the placement of the structure on Stationsplein. The only problem with the script right now is that the pick points in the flow chart are manually picked, with the use of lists.

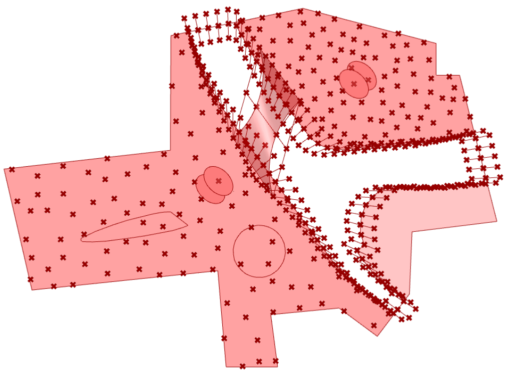


Figure 8: Final output Grasshopper placement structure

Specific points of bridge, location and stage are shared with Speckle. This helped improve the collaboration between all the scripts and to create one final look. For the bridge, the base points and the middle points that determine the height were sent, because these points are needed to create the bridge structure. Furthermore, the stages were sent so that the tent can be placed.

# 3. Bridge structure

## Concept

For designing the shape and structure of the bridge, a few parameters have to be taken into account. First of all, the free height underneath the bridge must be at least four meters, to allow public transport to pass underneath the bridge. Secondly, the idea is that the event lasts for about two weeks, meaning that the bridge needs to be demountable. And lastly, the bridge must be accessible for pedestrians and accommodate seating.

For the structure, some reference projects were found and the cardboard bridge designed by the architect Shigeru Ban is mainly used for the overall shape, especially the supporting structure, see Figure 9. The main idea behind using the truss in an arch is to make a relatively light bridge structure, in this case meaning a more slender and 'transparent', but still keep structural integrity.

**Figure 9: Shigeru Ban's Cardboard Bridge across the Gardon River France (Yoneda, 2014)**

In this case, instead of using cardboard, timber is used to create the entire bridge, except for the connections. Timber is chosen to create a more natural-looking bridge, making it more appealing to the public, instead of having an industrial-looking steel structure. In addition, round timber beams will be used to create the truss, this has a few advantages regarding sustainable construction and overall structural integrity. One of the advantages of round timber beams is that there is less waste when cutting these beams, and a smaller truss can be used, resulting in more efficient use of trees. For keeping the truss slender and aesthetically pleasing, round beams are also better than regular rectangular-shaped beams. Due to the way the round timber beam is cut from the tree, the wood fibers are not severed during milling (Roberts, 2018). This will result in more slender elements making the bridge look relatively light. However, in case a more permanent or slender design is needed, a steel truss can be considered.

For the stairs, a bit more consideration is needed to create a comfortable walking path for the pedestrians, but also make a comfortable seating area for visitors. This means that alternated steps are used, this is explained more in detail in this chapter. However, the idea is that in between the high steps designated for seating, smaller steps are placed. The height of the steps is in line with the regulations set by the government (Bouwbesluit). This means that the steps have a height of 200 mm, whereas the Bouwbesluit states that the rise of the step is a maximum of 210 mm, (Rijksoverheid, n.d.).

## Workflow & script structure

The bridge is built based on the points provided by a different script, these points are shared through Speckle and accessed at the start of the script.

All the files are in principle pre-loaded, and only connections must be made to the coordinate system. These points are used as the base curve for the bridge. From there the script splits into two main paths; the truss structure and the stair/seating structure, see Figure 10Figure 11. The two main directions are further explained in this part of the report. After the generation of the geometries, different parts can be sent back to speckle; such as the connection points for the tent structure.

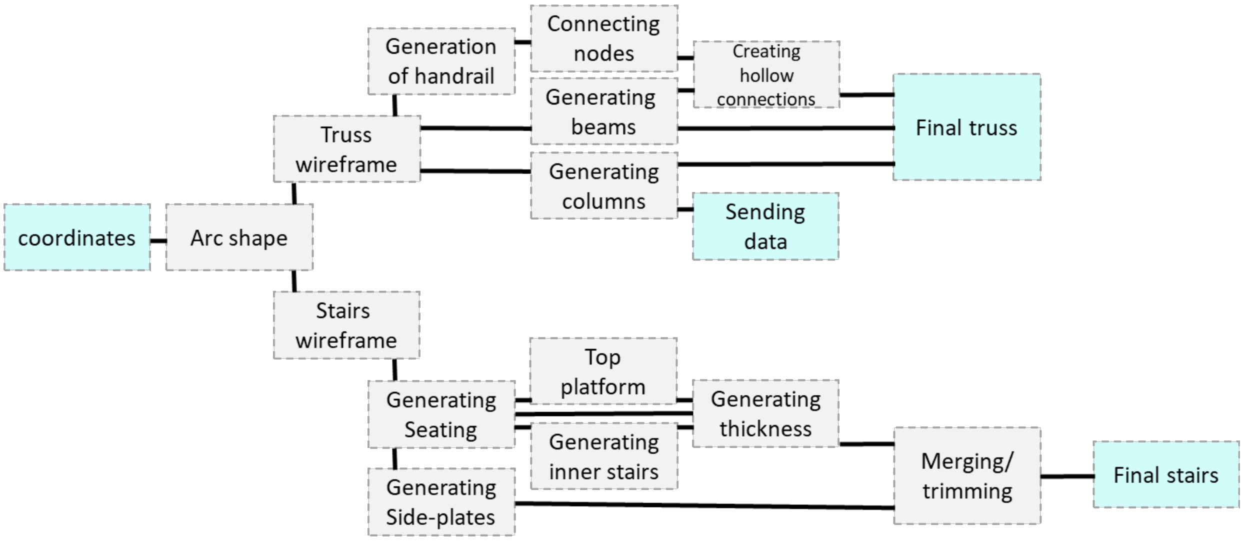



**Figure 10: Flowchart of design bridge structure**

## Truss structure

### *Process*

Initially, the truss was made using the plug-in lunchbox, a pipe along the curve was converted into a truss. This worked fine in some coordinate systems, however in some cases the truss rotation relative to the pipe it was projected on flipped, see Figure 11. And this happened randomly and was not influenced by code at all. This resulted in an unreliable script, which led to the second approach of manually making the truss structure.

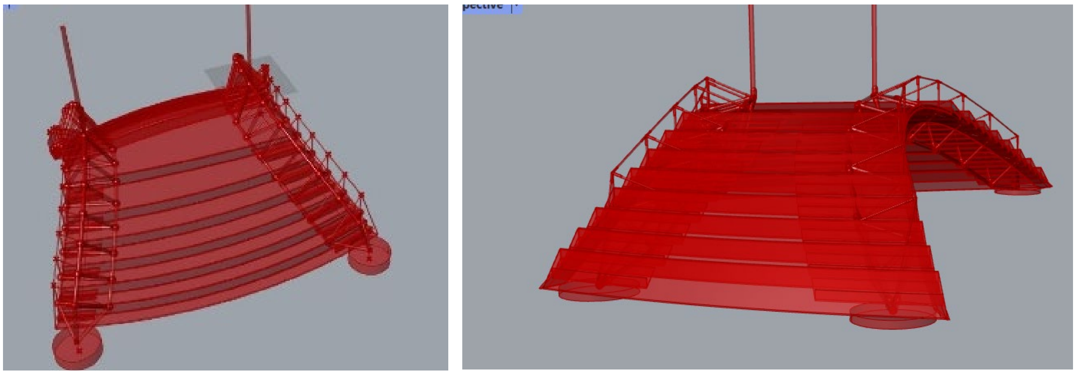


**Figure 11: First design of bridge using lunchbox**

### *Manually creating the truss*

First, the arch generated before is split into the number of desired nodes, after which a set of lines is calculated straight up with the diameter of the truss structure. Secondly, a plane is created on the arch perpendicular to it and in line with the previously create geometry. After this, a rotation axis is made through which the line is rotated 75 and 135 degrees creating the triangle truss structure, see Figure 12. From there, the rest of the truss is created by making the lines, thickness and connectors. This structure proved to be most robust, however when points are selected in the wrong order it can still 'flip', and thus when selecting the points for the base geometry sometimes the order needs to be flipped to generate the desired.

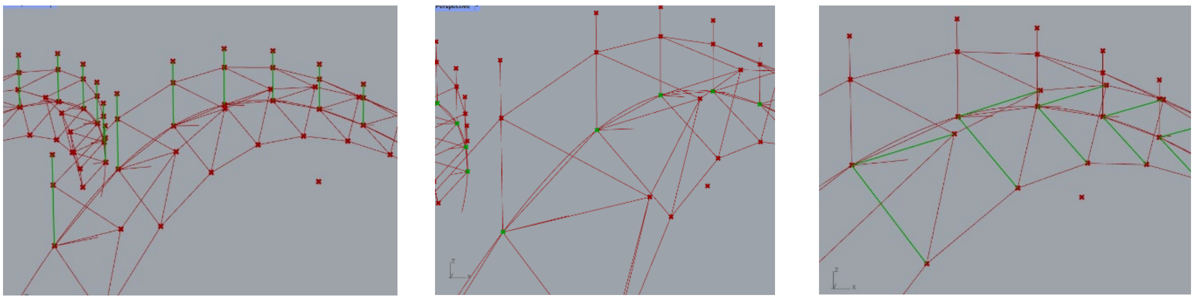


**Figure 12: Visualization of the manual cration of the truss**

Secondly, a handrail was made by offsetting the points on the base arc again upward with 1000mm. These were the new nodes for the rail, they were then connected to the two nodes on the truss and off course to each other. The data tree for this second truss was also way easier to work with than the lunchbox truss, because this tree was designed to create connections and was way more straightforward.

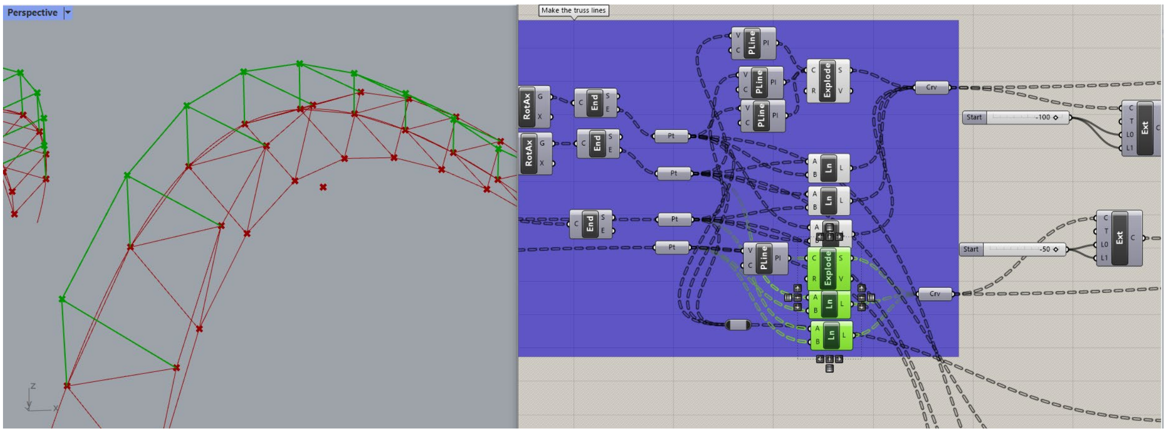



Figure 13: Design of the handrail on the bridge

## *Building the truss*

This wireframe then needed to be calculated to volumes, the lines between the nodes formed the base of the pipe structures. And the endpoints formed the base of the connecting nodes. The beams were created by offsetting the lines negatively to make sure no intersections between different beams were generated, after which the main beams and handrails were made as pipes.

The connection nodes went from the endpoints of the wireframe lines and were projected inwards. These were then also made as pipes with a larger diameter than the beams. Afterwards, the beams were subtracted from these volume pipes; resulting in the hollow connecting nodes.

This results in a pin-hole connection which is shown in Figure 14. The final beams are shown and are 75mm wooden poles, they should be cut to length straight and slot into the connectors, they can then be screwed in place. The handrails are made from 25mm stainless steel tubing and are also assembled in the same way as the truss.

The connectors are more complex, as they are created from a metal tube with inner diameters the same as the beams. All the angles are different and thus jigs must be made for every tube. We could aid this process by creating cut files for the short tube sections, after which they easily can be placed against each other and welded.

Time did not allow for a more detailed structural analysis of the structural integrity of the system. There is the option to use additional tension cables to create more triangles within the truss, these can then handle the tension loads as our pin-hole connection is not very effective in handling tension. These cables can also be seen in the project by Shigeru Ban, and are probably very necessary for our structure as well.

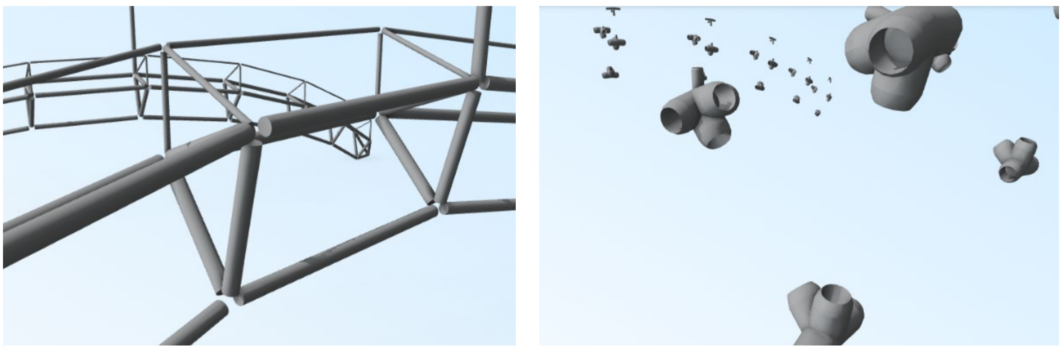


Figure 14: Connection design of the truss

## Stair structure

### *Process*

Like the truss, the structure of the stairs was not done in one go, the initial structure of the stairs consisted of multiple straight paths, where the seating area and the stairs were all on different angels, this approach, however, did create some issues with connecting these different surfaces and looks a bit ‘messy’. As well as this it also resulted in very strange geometry when the seating area became smaller. Thus, the new approach was a geometry where the stairs were part of the larger seating steps. The advantages of this are easier assembly, a cleaner look and mostly very little strange small geometry in edge cases. Thus, making the whole script more robust to different changes in the requirements.

**Figure 15: First stair design in comparison to the final design**

### *Creating the stair structure*

The stair structure was created from the same base curve as the truss structure, the base structure starts from a tween curve between the base curves, is extended a bit to create the curved stairs and from the highest point a line is drawn to z=0, this line is divided and planes are added at the height of the sitting platforms. These planes are then intersected with the three curves and these points create a set of arcs. These arches can then be translated down into z level by the seat height. Then with some smart selection, these lines can be connected with lofts to create the surface stair structure. These steps can be seen in Figure 16.

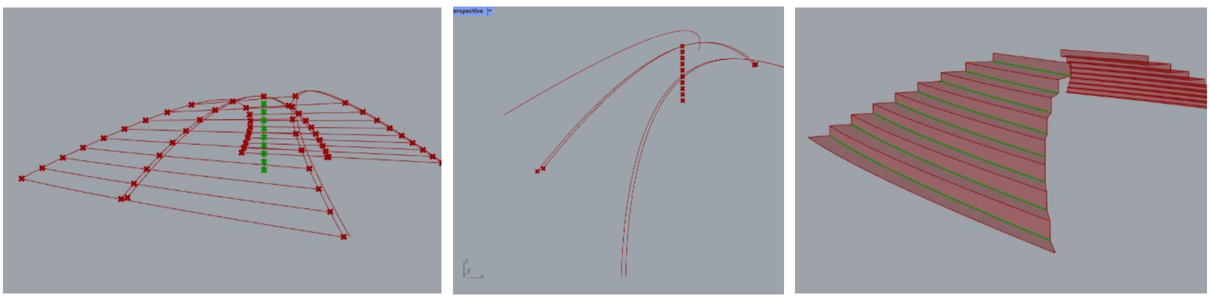


**Figure 16: Design process of the steps**

After this the intermediary stairs are created in a very similar fashion, shattering the base curves and translating them down, outward and down again. Then we can loft these new lines together again, to create the surface model of the stairs/seating. After this, only the side plates and the top surface are created with small separate functions.

The surface model of the stairs is then offset and lofted again to create the 3D geometry. Some trimming is done to prevent overlapping plates. And the side plates are used to cut all the access off the stair sides (These are generated with loft functions and thus the sides are straight lines unlike the bridge should be).

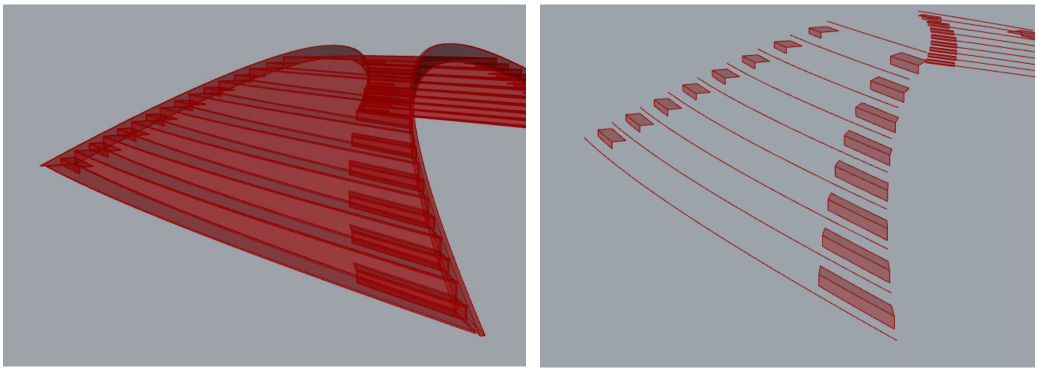


**Figure 17: The different component of the stairs**

## *Building the stairs*

The stairs are to be manufactured from our cutting patterns, the main stairs are all flat, and of course have different dimensions and sides. However the fronts of the stairs as well as the big side plates are curved, these curves are very gentle and supported by the rest of the structure, we expect the stair fronts to bend well to the gentle curves. The big side plates however are a very important structural element as they form the connection between the truss and the stairs. Moreover, it is way thicker at 75mm, they can perhaps be manufactured by steaming large beams and bending them along the stair structure.

In the end, there was no time to make cutting patterns for all the separate stairs, however, all the separate planks were exported in an assembled shape, ready for further processing.

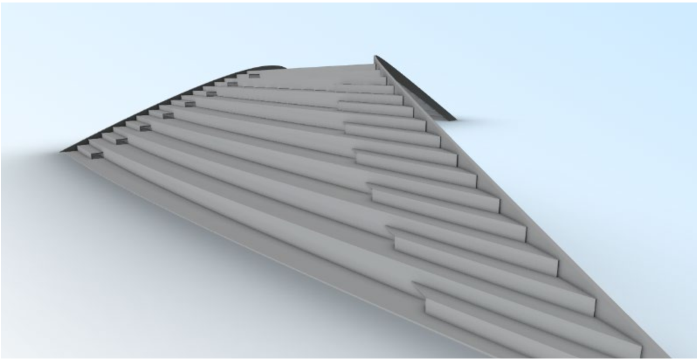


**Figure 18: The final look of a staircase**

## Combined result

The resulting script can take the coordinates provided by the previous script and generate the bridge for a lot of different design parameters, to show what is possible a few extreme cases have been generated with the script, see Figure 19. It is safe to say that the script at this moment does not account for structural stability, and thus a lot of variations generated are not viable to be built.

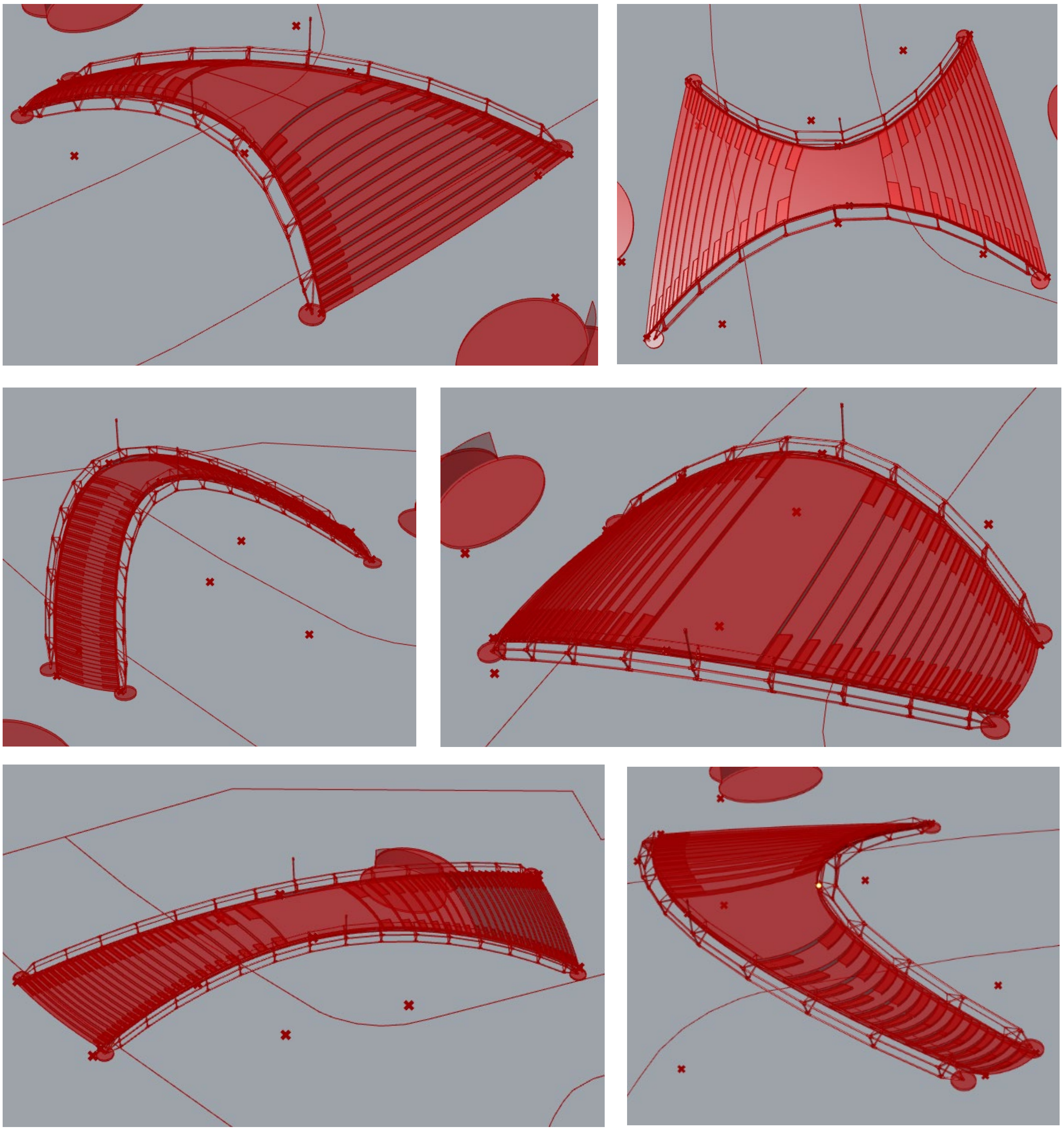


**Figure 19: A representation of possible bridge designs**

# 4. Tent structure

## Concept

As explained, to allow the event to happen during all types of weather, some kind of cover system has to be designed. The idea is to keep this cover structure as easy as possible since the construction of the bridge is already complicated enough. And thus a tensile membrane structure is created. The idea is that a precut pattern is delivered to the site, and with the use of columns, the tent can be constructed on-site. This design also leaves a free floor plan underneath the covered places, meaning that all types of configurations for the stage are possible.

Another advantage of this membrane is that it uses less material than conventional cover structures and if the correct material for the tent is chosen, it can be completely recycled. In this case, that means that PVC coated polyester fabric is used, this means that if the event will only happen once, the membranes can be completely recycled into for example road cones (The advantage of tensile structure, n.d.).

## Workflow & script structure

### Workflow

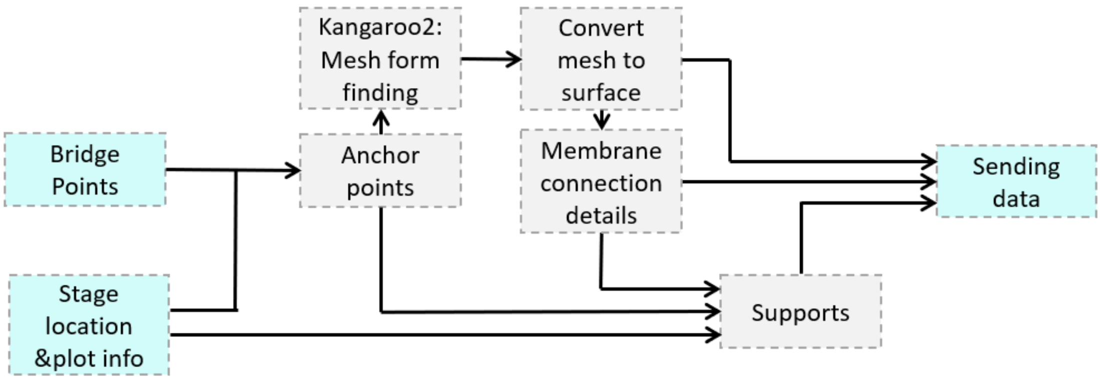



**Figure 20 Roof script structure**

The script to generate the roof takes some selected outputs of the location and bridge scripts as inputs through Speckle. The script generates three types of elements;

- A membrane surface loaded purely in tension.
- The supports of this membrane.
- The connection of this membrane to said supports.

These geometries are then sent to the Speckle server. The script can generate a roof structure on either side of the bridge. To create a roof structure for two bridges the script must be run twice using different Speckle inputs.

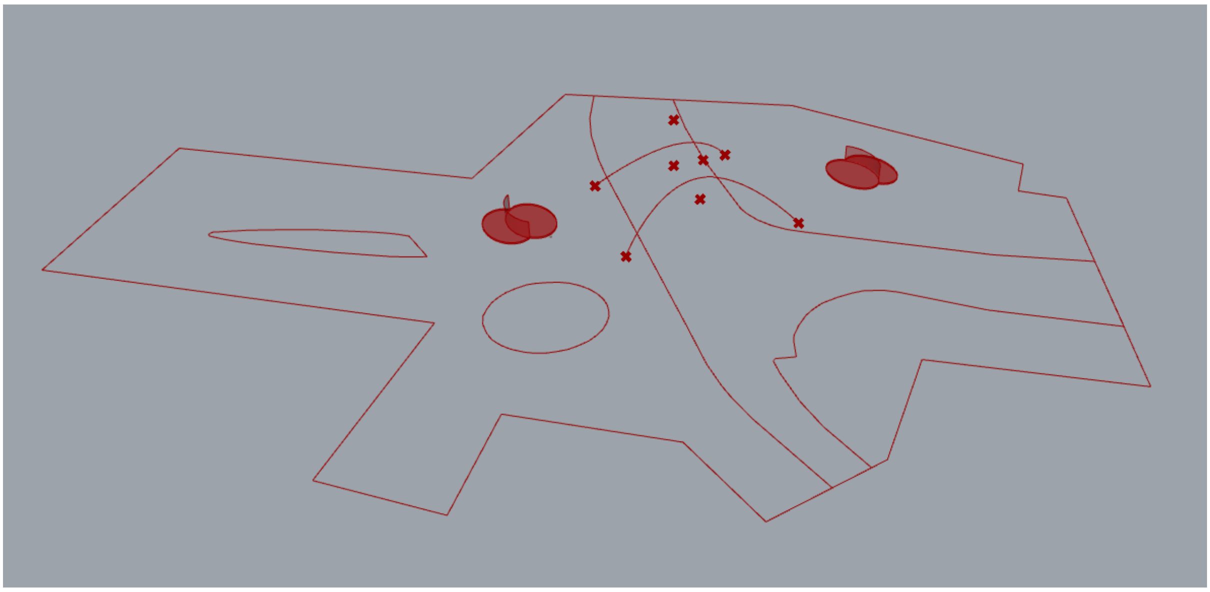


Figure 21 Speckle input to generate the roof.

## Anchor points

Using some points from the bridge (like the tip of the column in the middle) and location info (like the location of the stage) six anchor points are defined. Four points on the edge of the roof surface and two points in the middle lower than those four. If all anchor points are at the same height the double curvature which gives tensile structures their stiffness is not created. Two anchor points coordinates are the points to which the membrane connects to the top of the bridge and are set in stone. The other four can be translated using sliders which are the main inputs for this script.

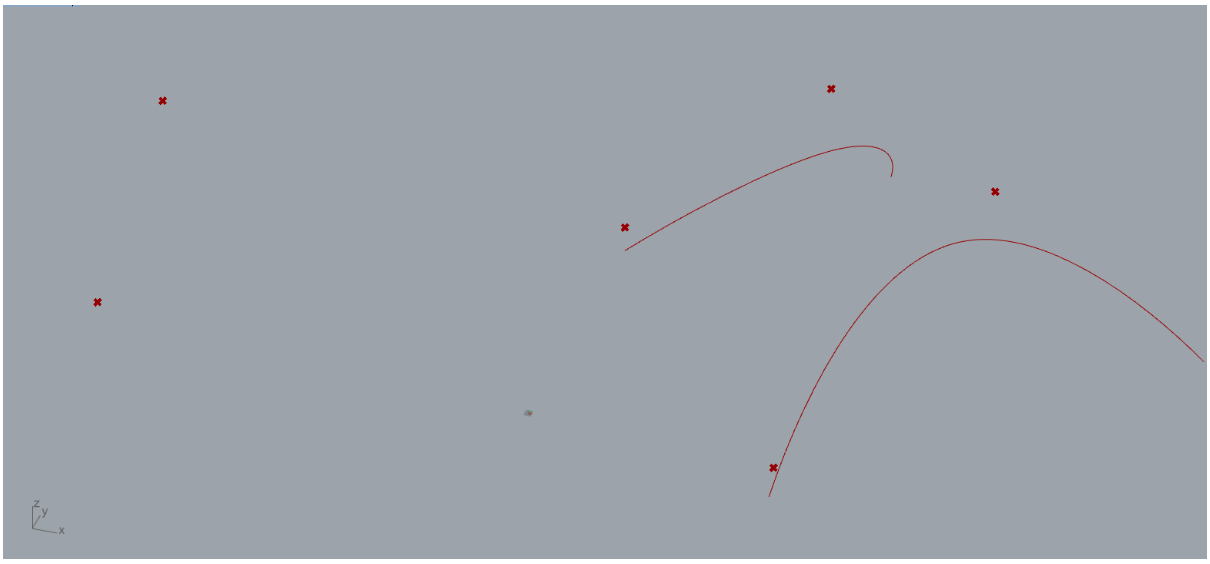


Figure 22 Anchor points and bridge outline.

## Form finding and surface fitting

These anchor points are the points to which a flat mesh is stretched to create the membrane surface. The boundary of this mesh is defined by a closed polyline running through the four highest anchor points. The actual form finding of a tensile structure is mathematically complex and performed by a dedicated plug-in; Kangaroo2. This plug-in does the heavy lifting of the iterative computation. Care must be taken that the Kangaroo module is not in a cluster, as this stops the calculation after only a few iterations. Additionally, the threshold cannot be the default value, this means too many iterations which slows down the script massively. The output of Kangaroo is a series of points, to which a surface is fitted.

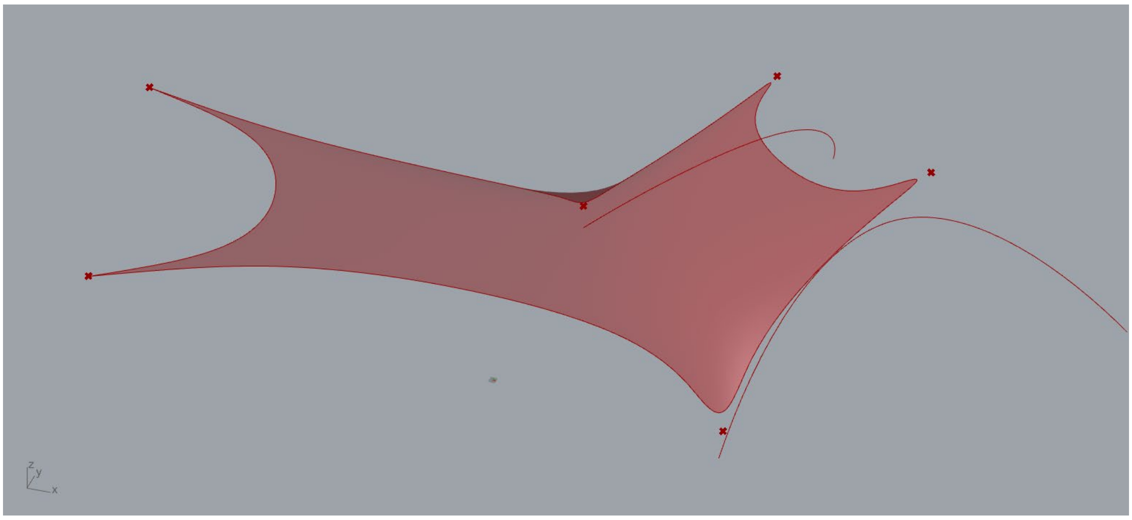


Figure 23 Surface created by Kangaroo2.

## Supports

Each roof structure requires at least two columns and an optional two to increase the height of the middle anchor points. Additionally, these columns must in turn be supported by at least two cables each (except the two optional columns due to their small length can take some bending moments). In real practice angles between these support members are determined using calculations to assure that only normal forces occur, and equilibrium is maintained. The script does not do a structural calculation so an eyeball engineering judgment is made in combination with common sense. For example, if the tension in two wires supporting a column is the same, then their angle to the membrane wire must be the same to maintain equilibrium. This is included in the script. The inputs for the generation of the supports are the slant of the columns and support lines and the angle of the support wires to each other. Slanted columns with near-vertical support wires occupy the least amount of space.

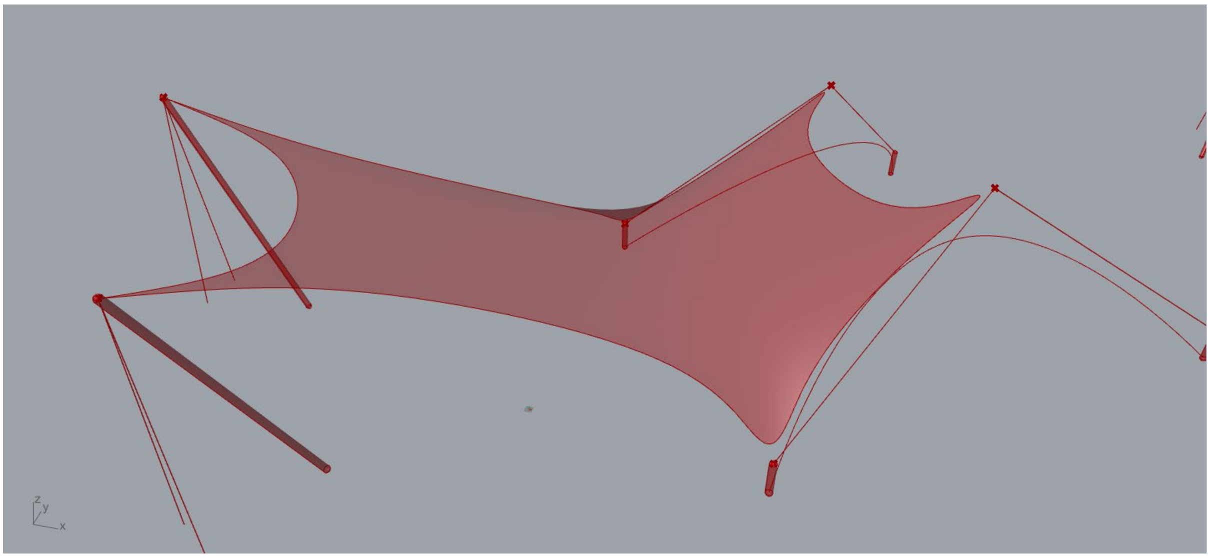


Figure 24 Tent with support members.

## Detailing

The Kangaroo output pulls a point on the mesh to an anchor point in space. For our theoretical mesh in Grasshopper, this means that the surface converges toward and is supported by an infinitesimally small point. This cannot be created in the reality of course, as this creates a force on an infinitesimally small surface, meaning infinite stresses. To reduce these peak stresses, membrane structures require specific steel support details.

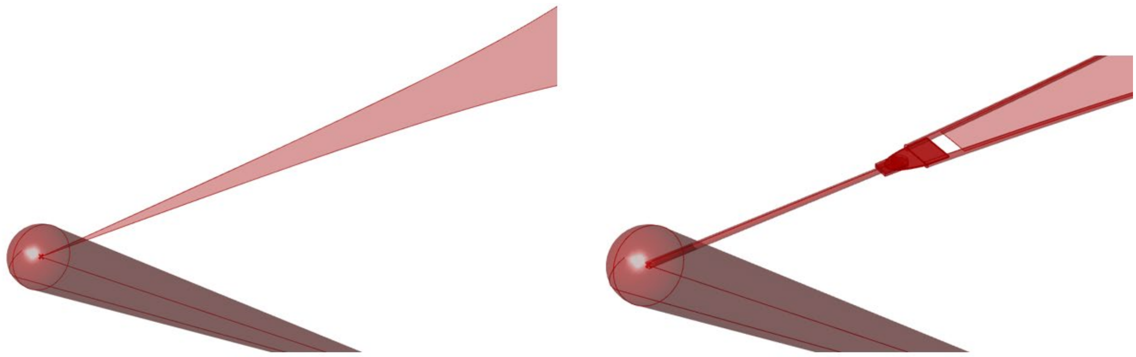


**Figure 25 Input and output detailing module.**

The script creates these details by a combination of trimming the surface, creating lines and extrusions. The main inputs are the length of the line running from the column to the beginning of the steel strip and the length between the column and the beginning of the membrane surface.

## Combined result

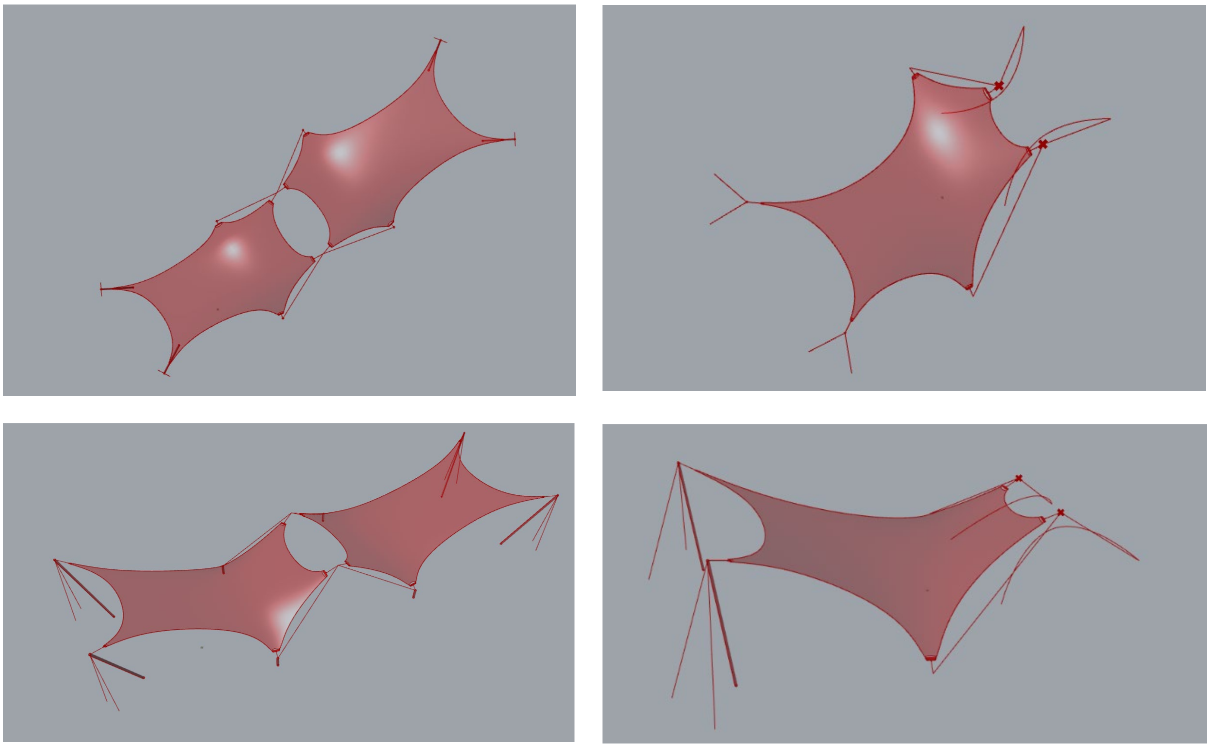


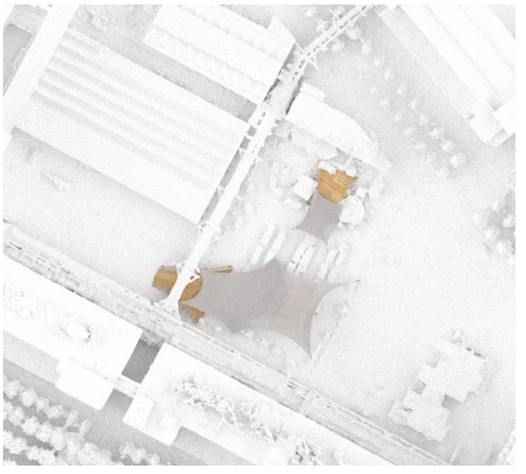
**Figure 26 Combined script results.** **Variant 1** **Variant 2**

# 5. Overall design

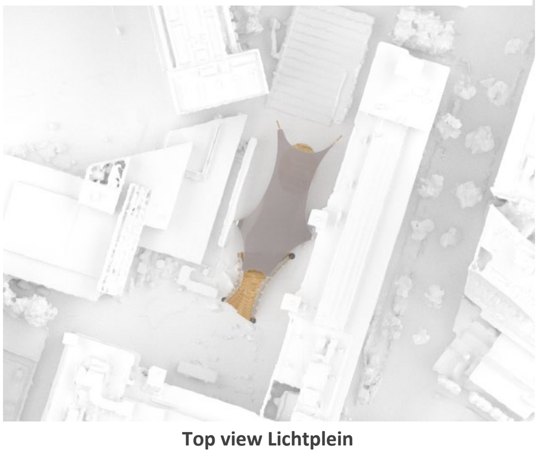
After making all the separate components for the entire design, everything can be combined into the point cloud files. It becomes clear that Stationsplein fits the theme best and makes the most compelling design of the event in theme mobility. However, this does not mean that the other squares do not have any potential to become just as grand of a design as Stationsplein. As said, one of the extra design inputs could be placing the bridge in such a place and height that it gives a nice view over the city. Below, a summary of all renders can be found, these show the completed design and how the structure is eventually placed on the squares.

## Final placement on squares

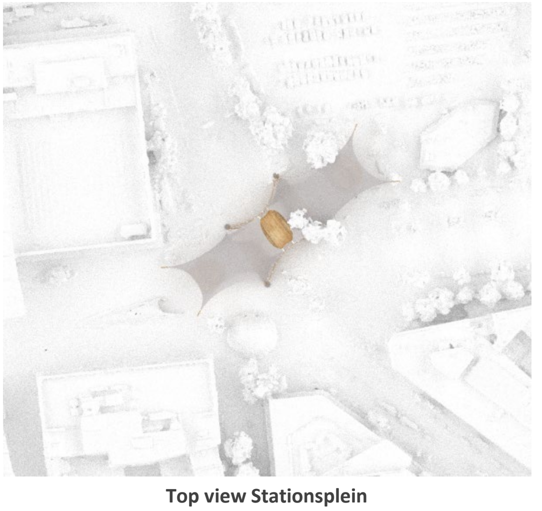
**Top view Augustijnenkerkplein**



**Top view Ketelhuisplein**

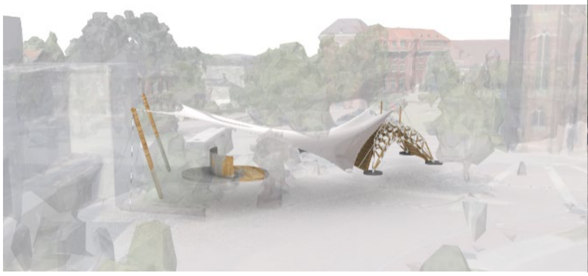


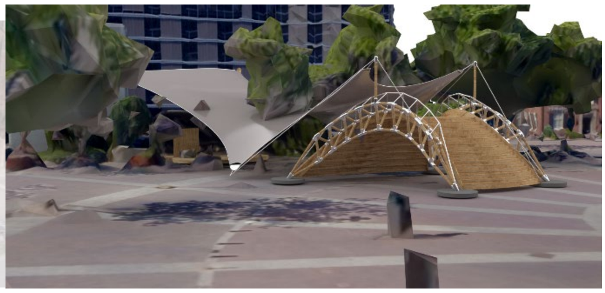
**Top view Lichtplein**

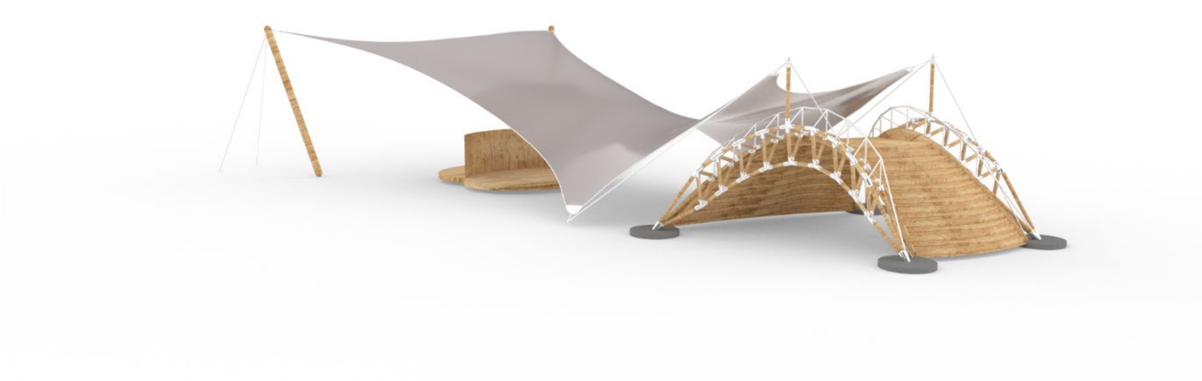


**Top view Stationsplein**

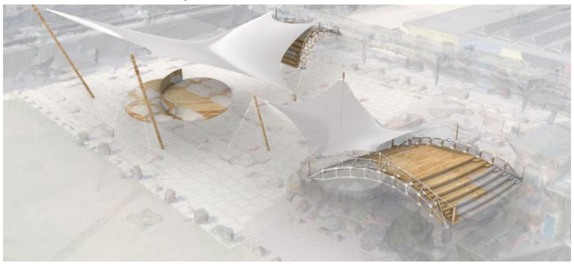
# Augustijnenkerkplein

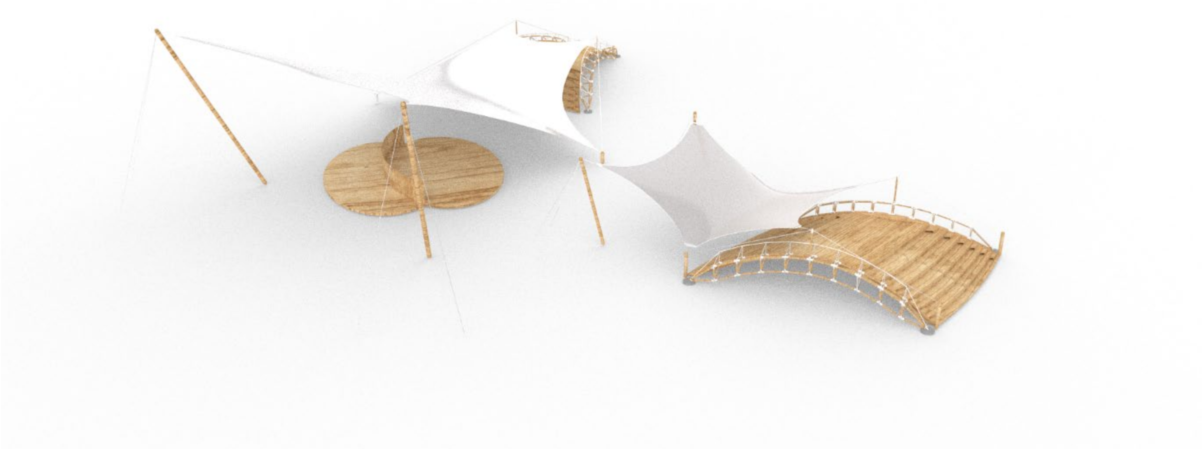




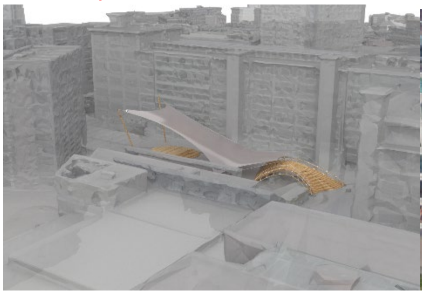


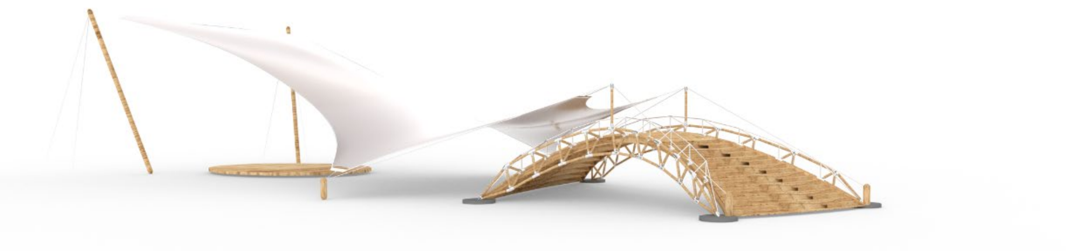
# Ketelhuisplein



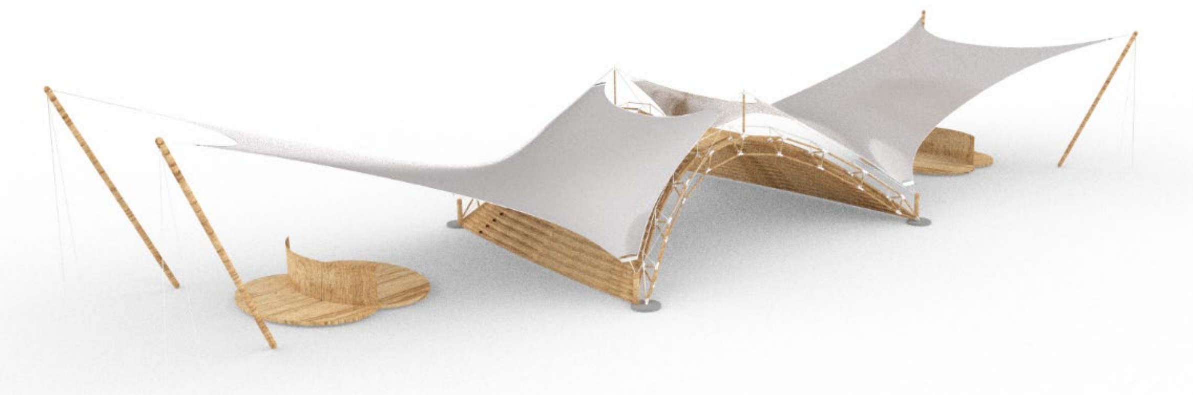


# Lichtplein



# Stationsplein

From these renders, the type of material is visible as well as the connecting pieces used to combine the truss, as explained in Chapter 3. It can also be seen that the use of round timber elements ensures a slender and lightweight-looking bridge, which was one of the requirements set at the beginning of the design process. For the variant placed on Stationsplein, all components are listed separately, however, due to time constraints, we were not able to finish the complete list of materials with their lengths and correct dimensions to ensure the structural integrity of the bridge.

This could be done using the Karamba plug-in in Grasshopper together with the use of Eurocodes 3 and 5, these are for steel and timber. Also, the foundation of the bridge needs to be worked out further. Right now, the idea is to use 4 big concrete footings to hold down the bridge. These concrete footings will create a bit of counterweight when for example large wind gusts happen during the event. Together with analyzing the bridge, the tent structure needs to be analyzed, to see whether PVC can be used for the structure that is now designed.

But looking at the entire structure and the fact that the maximum transportable length of an element is around 12 meters, the separate components should not be a problem. This is mainly because the truss can be completely assembled on sight, thus ensuring the possibility of transportation to the squares.

The current elements are separated, and with a few adjustments to the entire script and structural analysis, a complete list can be made and used to assemble the bridge. These components can be found in the Speckle workflow used during this entire design process, and are listed in Chapter 6.

# 6. References


Lambert, R. B. (2019, September). *How to create walking friendly cities*. Retrieved from Natural walking cities: https://naturalwalkingcities.com/how-to-make-cities-walking-friendly/#:~:text=Cities%20and%20places%20that%20have,the%20levels%20that%20people%20walk.

Rijksoverheid. (n.d.). *Afdeling 2.5. Trap*. Retrieved from Bouwbesluit Online 2012: https://rijksoverheid.bouwbesluit.com/Inhoud/docs/wet/bb2012/hfd2/afd2-5

Roberts, T. (2018, March 28). *The Benefits of Roundwood Timber Framing*. Retrieved from Rise: https://www.buildwithrise.com/stories/the-benefits-of-roundwood-timber-framing

*The advantage of tensile structure*. (n.d.). Retrieved from J&J Carter: https://www.jjcarter.com/news/the-advantages-of-tensile-structure#:~:text=Quick%20Installation,and%20build%20out%20new%20walls.

Wikipedia-bijdragers. (2023, Febuary 17). *Van Hool ExquiCity*. Retrieved from Wikipedia: https://nl.wikipedia.org/wiki/Van_Hool_ExquiCity

Yoneda, Y. (2014, March 25). *Shigeru Ban's Incredible Cardboard Bridge Can Hold 20 People at a Time!* Retrieved from Inhabitat: https://inhabitat.com/shigeru-bans-cardboard-bridge/

Beams: https://speckle.xyz/streams/9b497ce083/commits/41cb44f52b

Connectors: https://speckle.xyz/streams/9b497ce083/commits/07b52d8836

Wood parts: https://speckle.xyz/streams/9b497ce083/commits/1005e9bab0