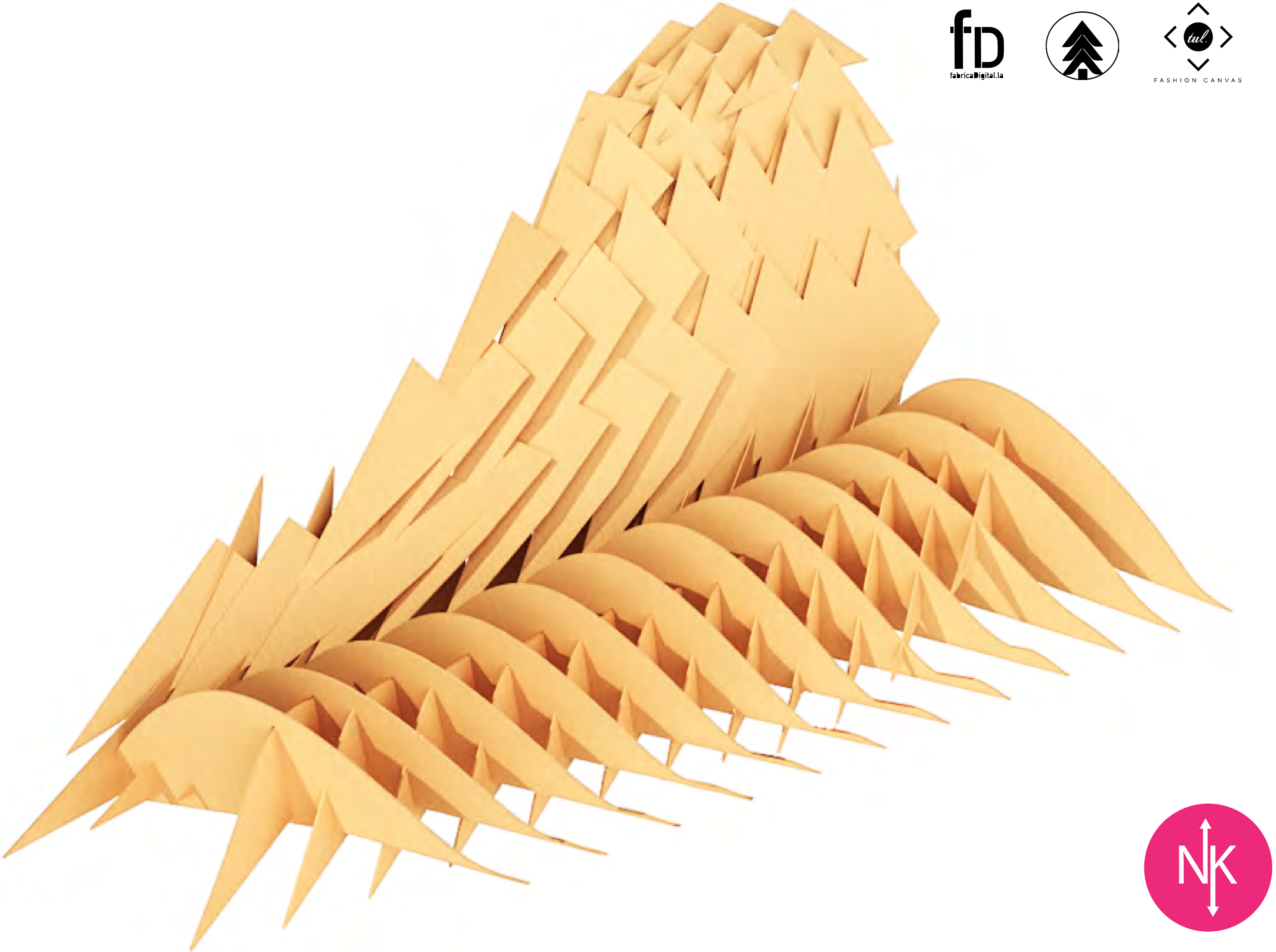
ARCHITECTURAL ASSOCIATION
SCHOOL OF ARCHITECTURE

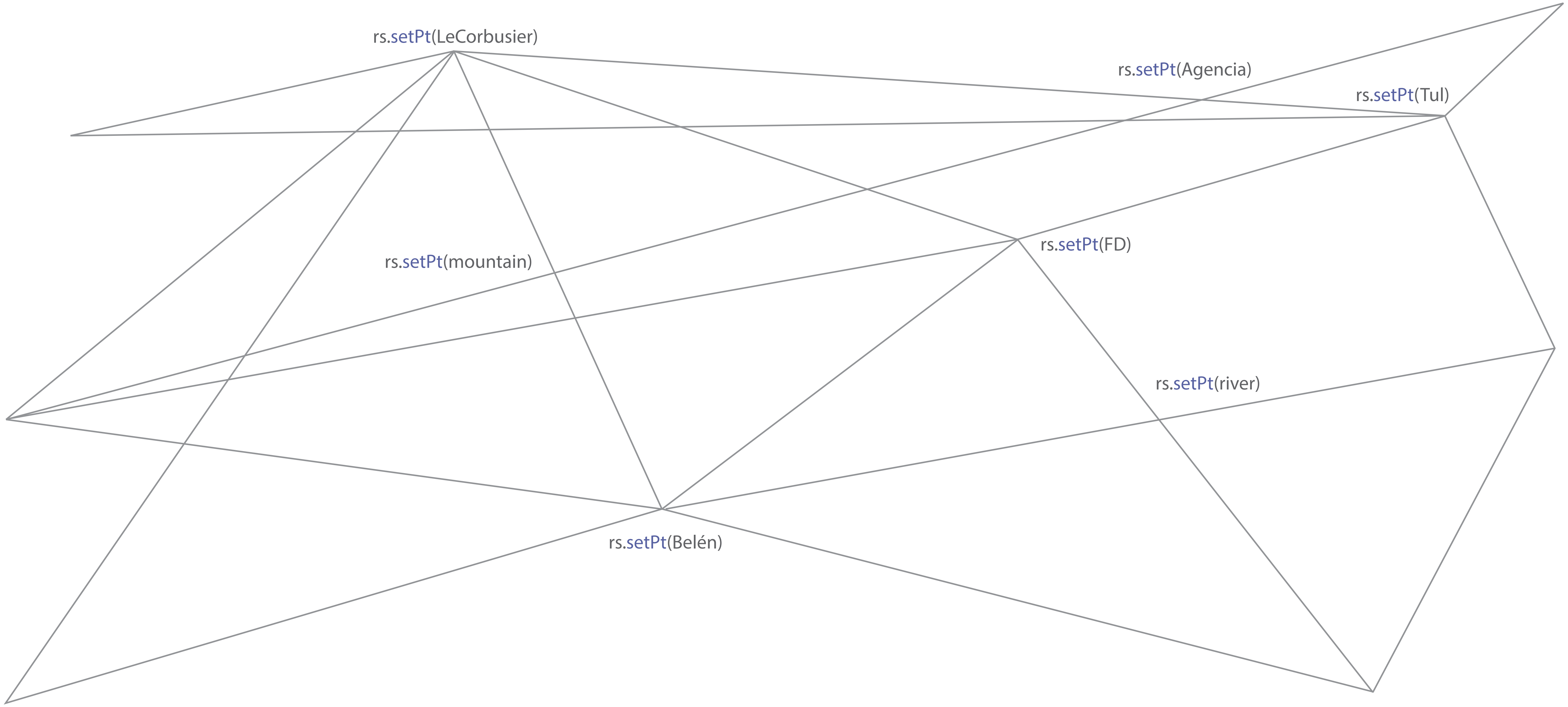


Pink Team

Arq. Eduardo Cardenas P.
Arq. Nicolas Díaz B.
Arq. Laura Alejandra García
Arq. Nadia Mendez
Dis. Ricardo Rambal Fattori
Arq. Daniela Vega
Arq. Rodrigo Vargas

"Up the river, Up the mountain"

AA VISITING SCHOOL BOGOTÁ 2013



rs.setPt(LeCorbusier)

rs.setPt(Agencia)

rs.setPt(Tul)

rs.setPt(FD)

rs.setPt(mountain)

rs.setPt(river)

rs.setPt(Belén)

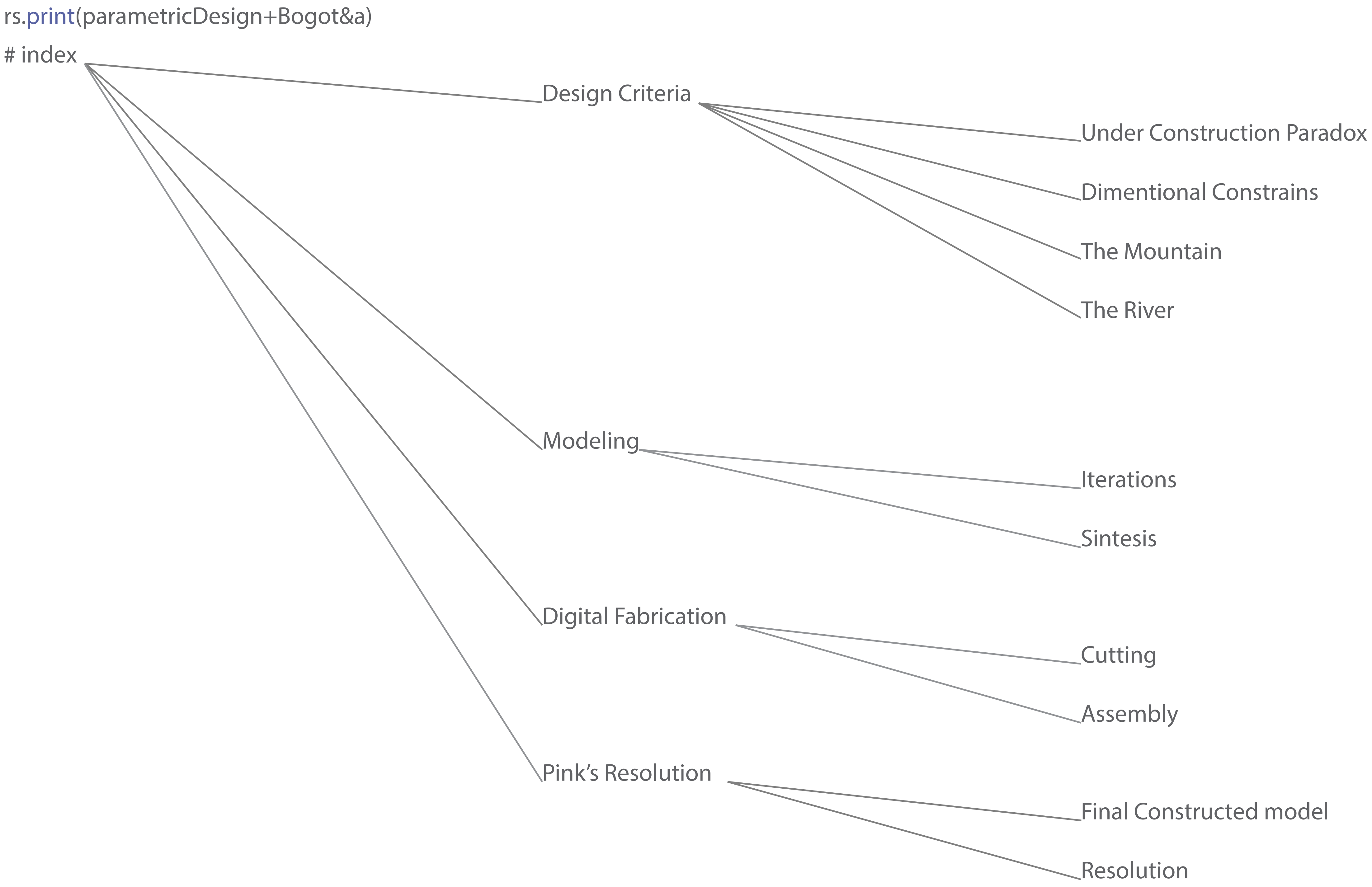
rs.print(parametricDesign+Bogot&a)

//////

We will deploy small-scale paradoxical actions, this appropriation of a structure in the city will allow us to interrogate one of Bogotá's most striking features: its bizarre ad infinitum phase construction.

//////

rs.print(parametricDesign+Bogot&a)



# index

Dailly stresses can drag anyone down, making escapism and getting away from it all an important part of getting through life.

The tendency to seek distraction and relief from unpleasant realities, esp. by seeking entertainment or engaging in fantasy.

Normal object in unusual places = unusual object in normal places, dreamlike scenery.

We will deploy small-scale paradoxical actions, this appropriation of a structure in the city will allow us to interrogate one of Bogotá's most striking features: its bizarre and infinitum phase construction.

Surrealism

Design Criteria

Under Construction Paradox

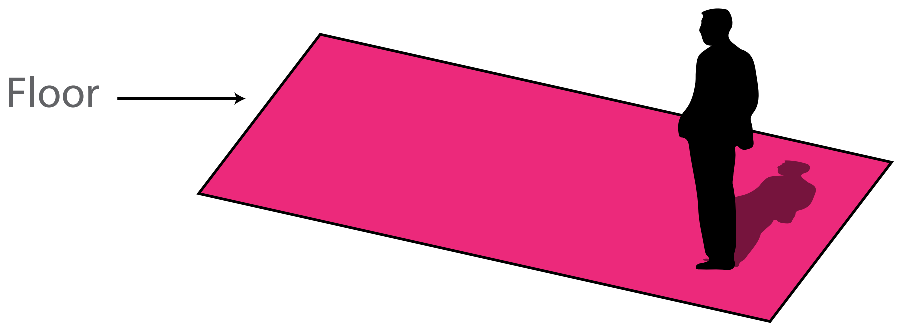
Dimentional Constrains

The Mountain

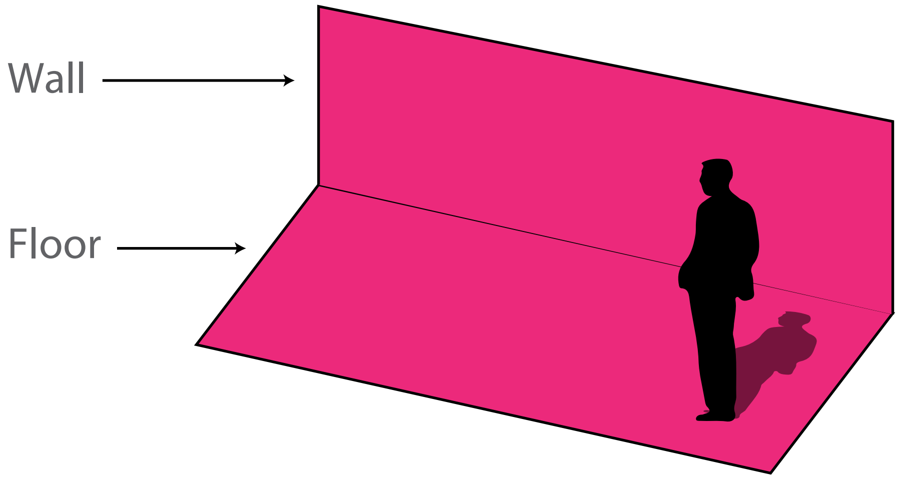
The River

```
rs. print(designCriteria)
  """
  to give “form” a sense of being besides itselft, a need
  for conceptualizing  ideas in a frame of needs and
desires .
"""
```

Situation 1



Situation 2



rs. print(Dimentional Constrains)

"""""

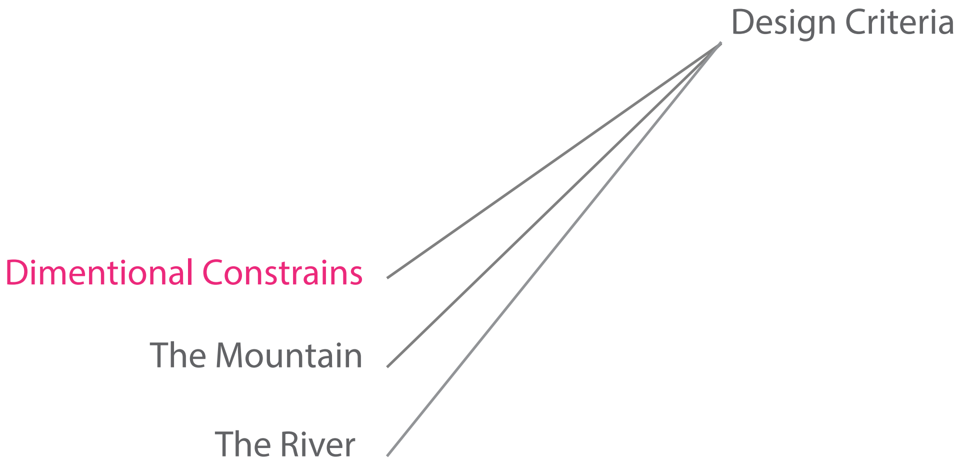
The object must respond to diferent places each once having its own unique constrain.

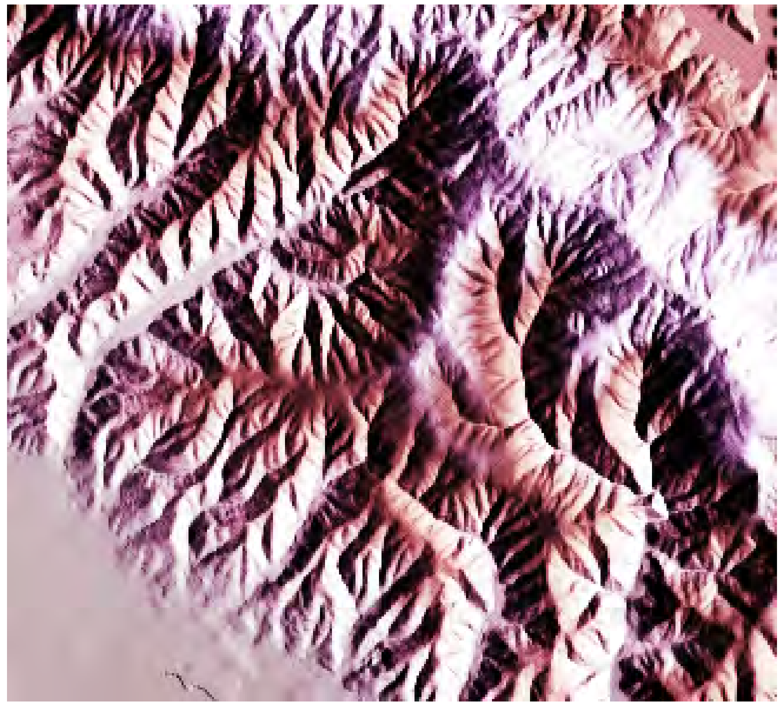
"""""

Tul Workshop

La Agencia Rooftop

Fabrica Digital



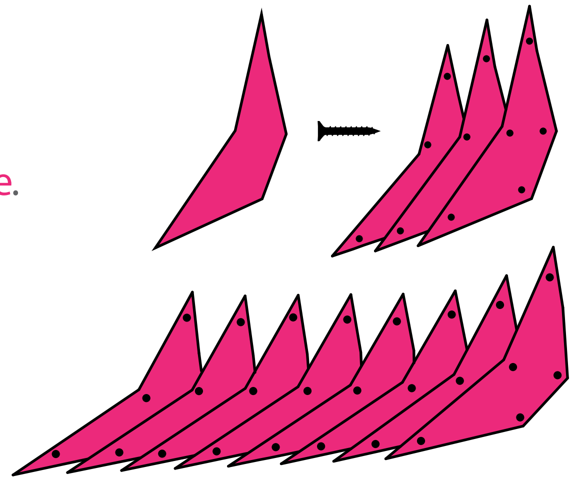


rs. print(The Mountain)

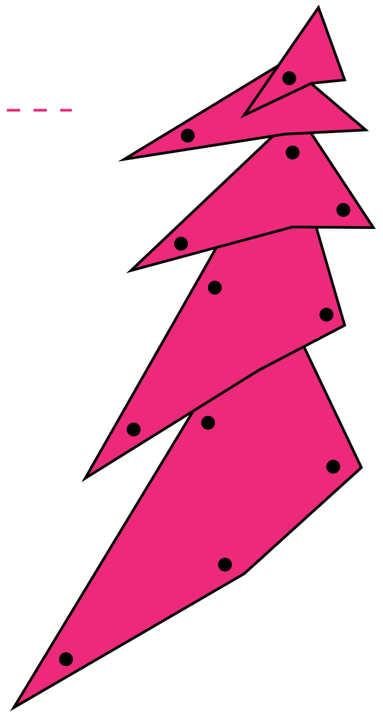
A large natural elevation of the earth's surface rising abruptly from the surrounding level.

```
for i in range(1, len(curves)-10):
.....for j in range(0, len(div)-5):

.....if i<20 :
............crv3Temp = rs.AddInterpCurve (((divPTCoor[i+2][j], divPTCoor[i+3][j+4], divPTCoor[i+1][j+5])),1)
```

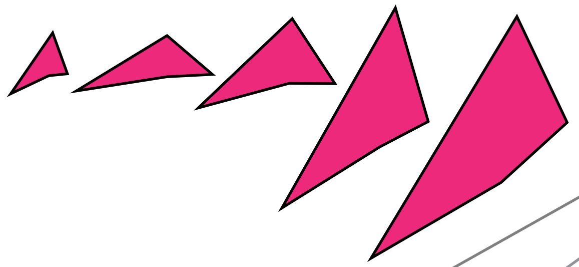
This parameter defines the curvature of the lines, we chose the lowest degree to mimic the mountain texture.



```
for i in range(1, len(curves)-10):
.....for j in range(0, len(div)-5):

.....if i<20 :
............crv3Temp = rs.AddInterpCurve (((divPTCoor[i+2][j], divPTCoor[i+3][j+4], divPTCoor[i+1][j+5])),1)
```



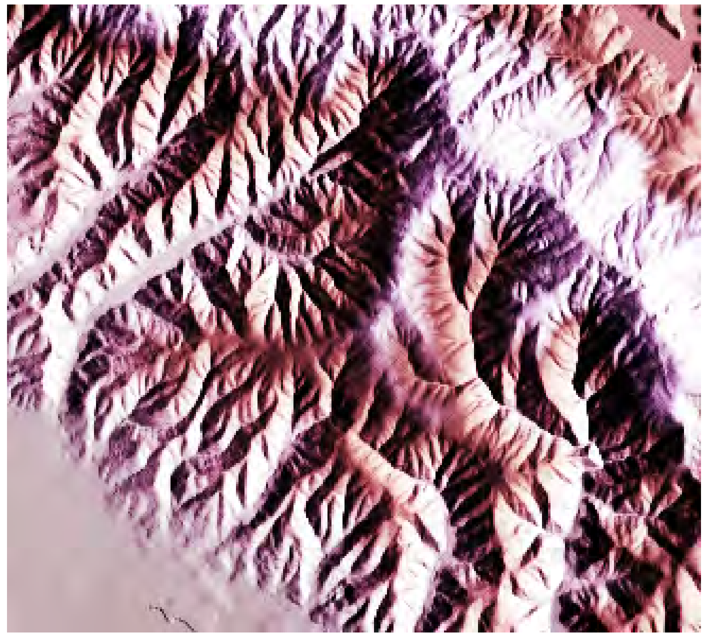
This parameter defines how the surfaces mutate through the shapes in order to achieve the growth of the mountain. At the bottom the pieces are big whereas as the top we wanted to represent the peak of the mountain.

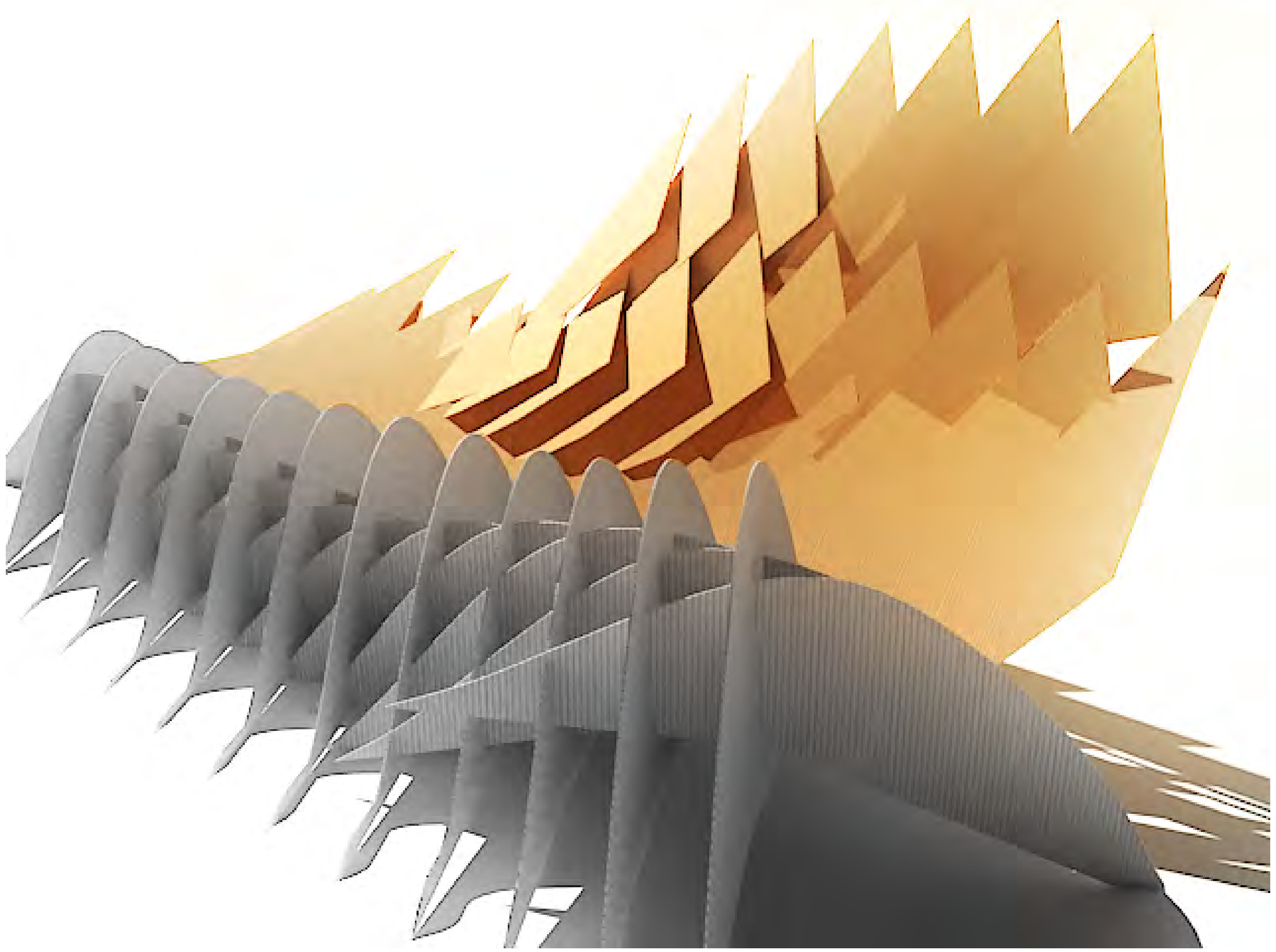


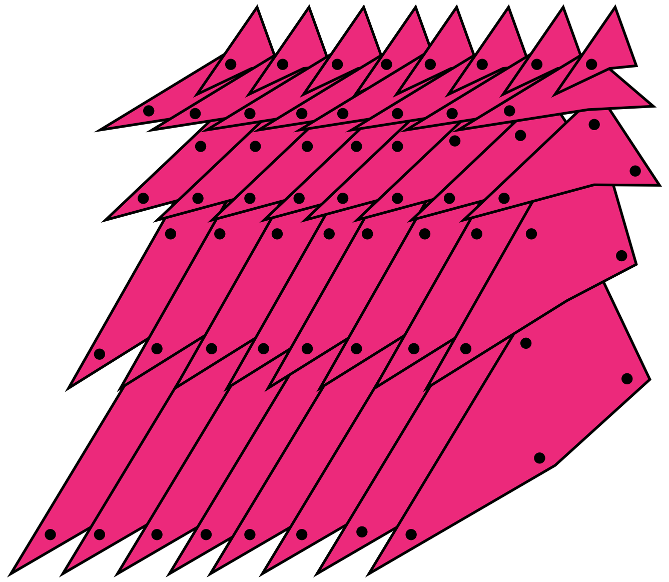
Design Criteria

The Mountain

The River







Design Criteria

The Mountain

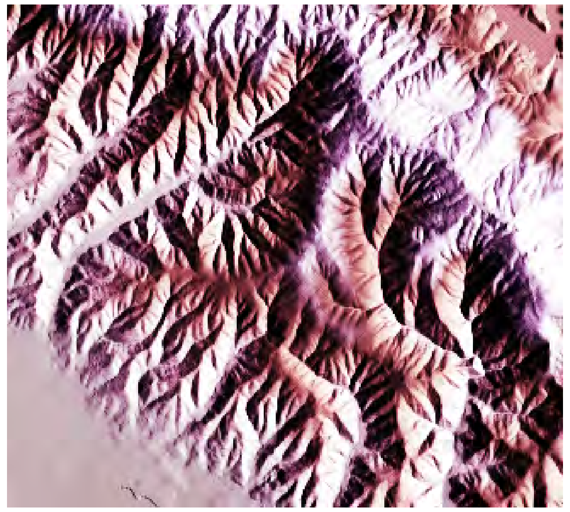
The River

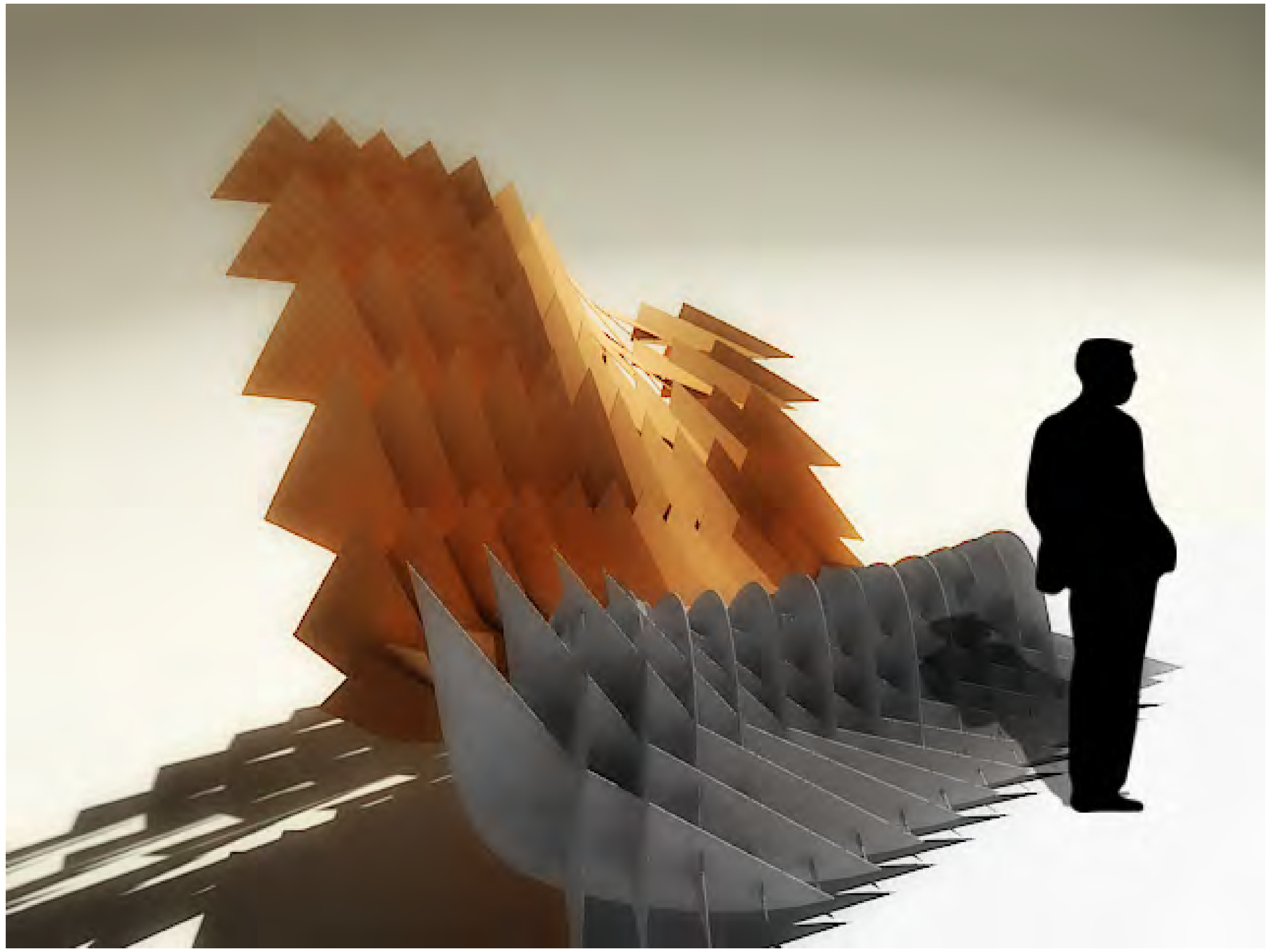
rs. print

//////

A large natural elevation of the earth's surface
rising abruptly *from the surrounding level.*

//////



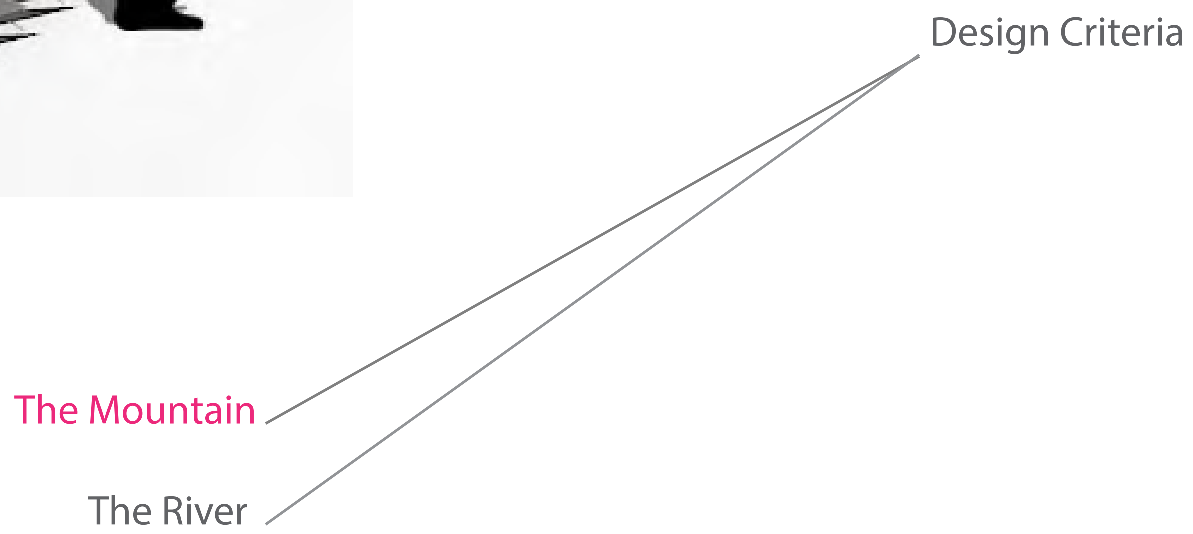


rs. print

/////

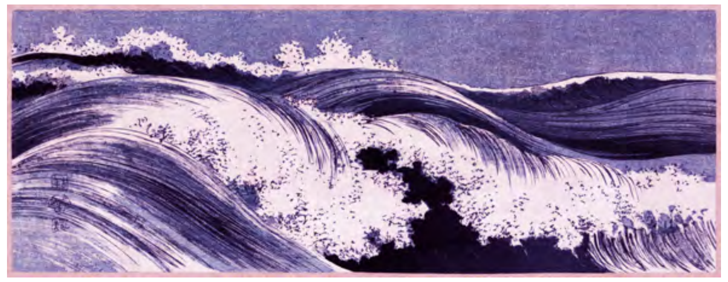
A large natural elevation of the earth's surface rising abruptly from the surrounding level.

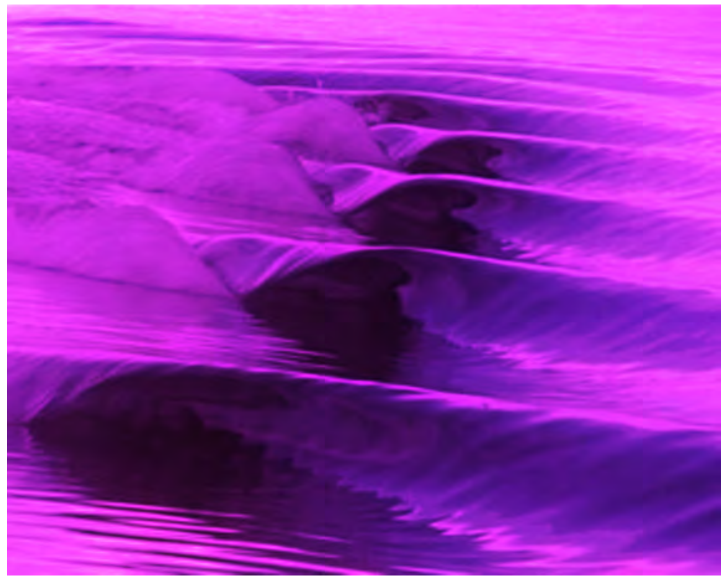
/////


Design Criteria

The Mountain

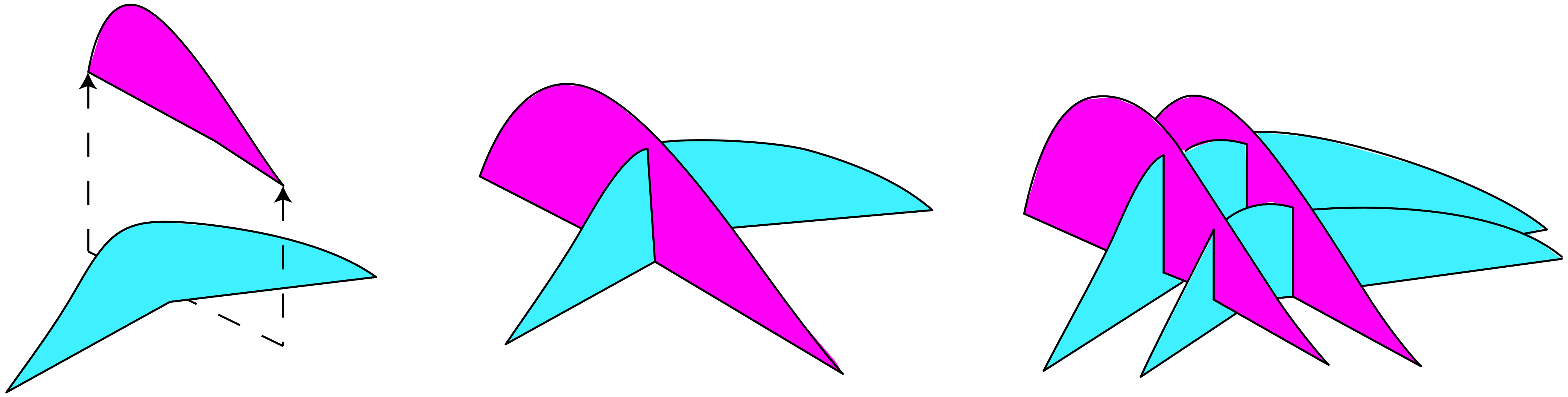
The River



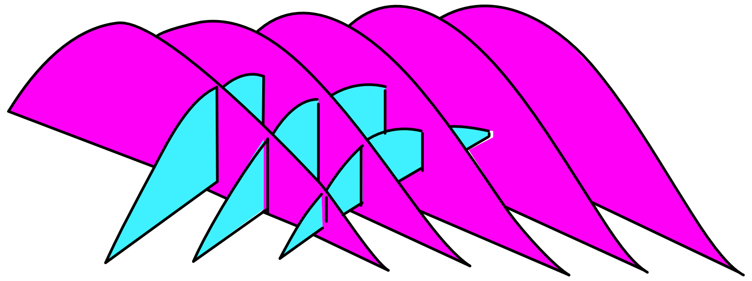


```
for i in range(1, len(curves)-10):
.....for j in range(0, len(div)-5):

.....if i<20 :
............crv3Temp = rs.AddInterpCurve (((divPTCoor[i+2][j], divPTCoor[i+3][j+4], divPTCoor[i+1][j+5])),3)
```

This parameter defines the curvature of the lines, we chose the highest degree to mimic the water waves.



To attain the waffle union between the waves, we applied the code to a surface in order to modify the shape the way we wanted to.



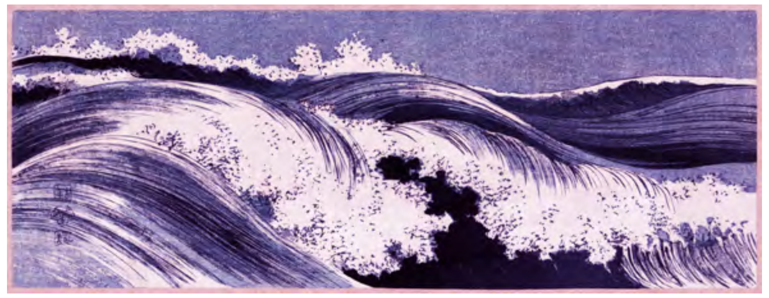
rs. print

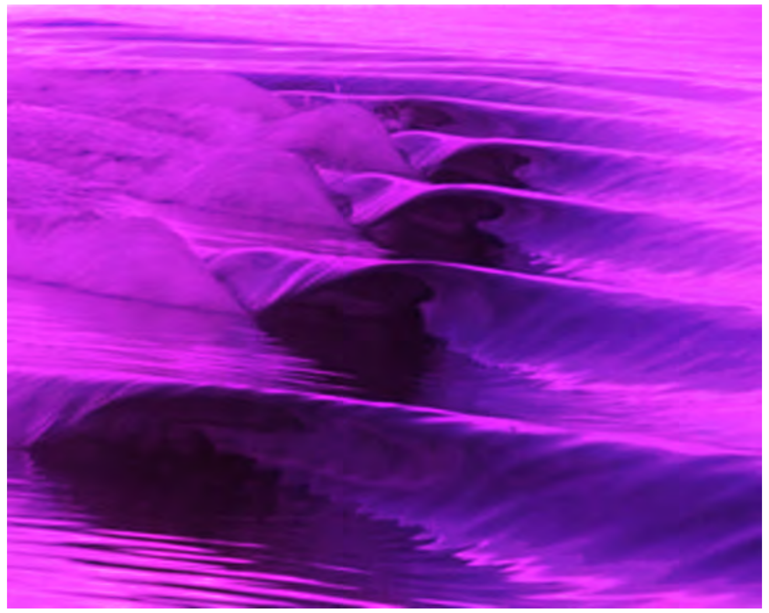
"""""

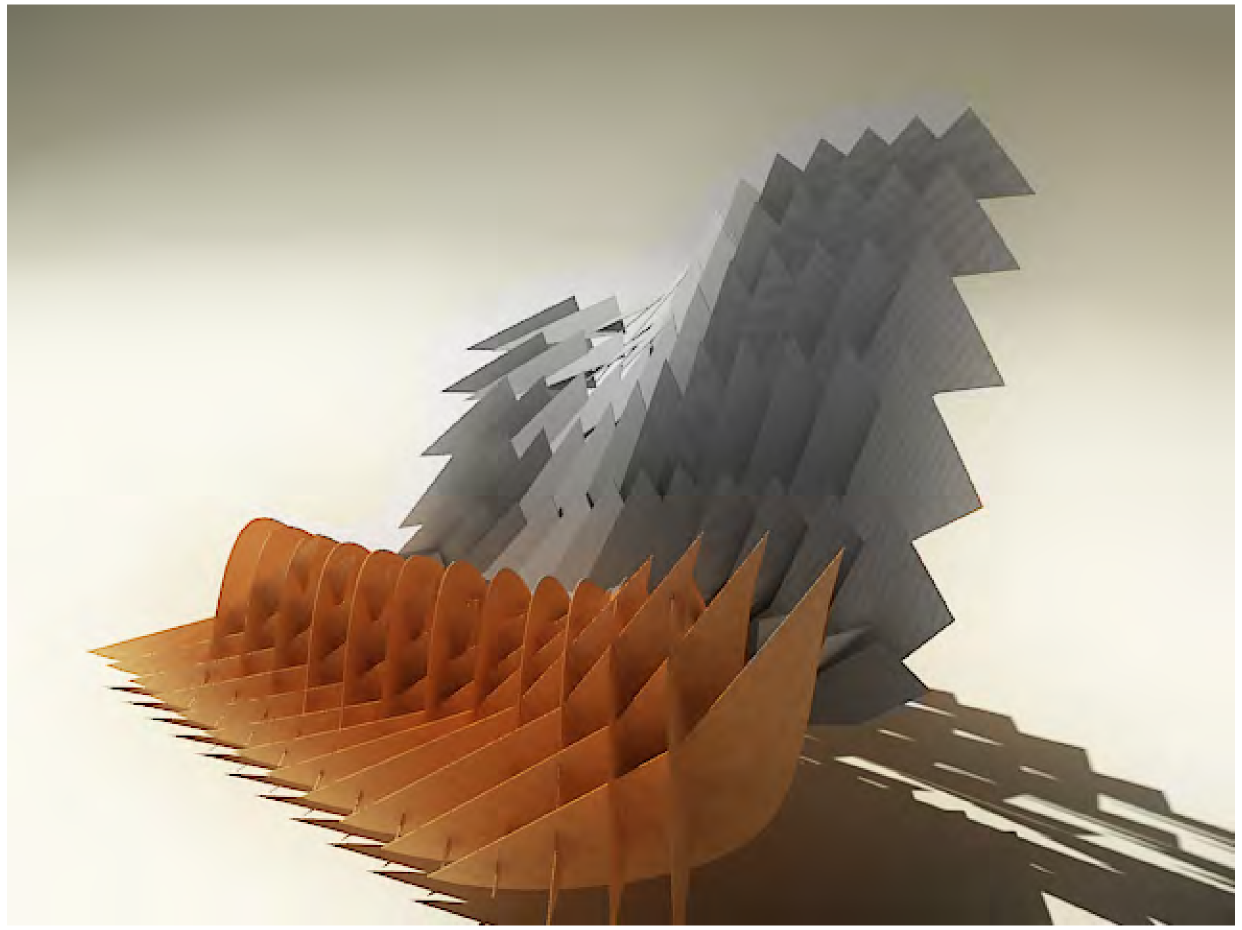
A large quantity of a flowing substance.

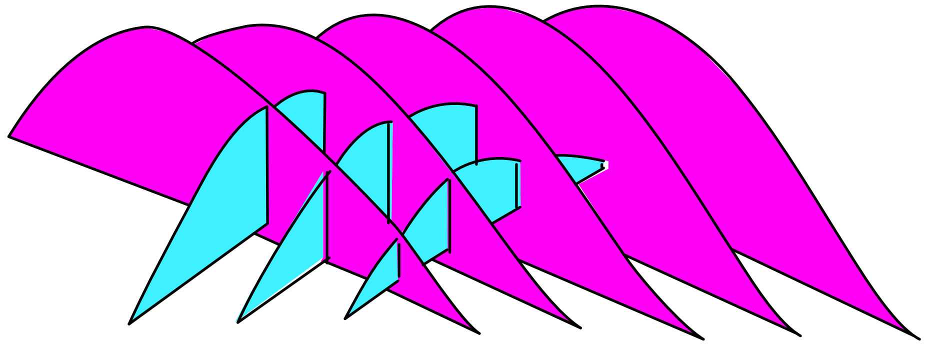
"""""

Design Criteria
The River









Design Criteria

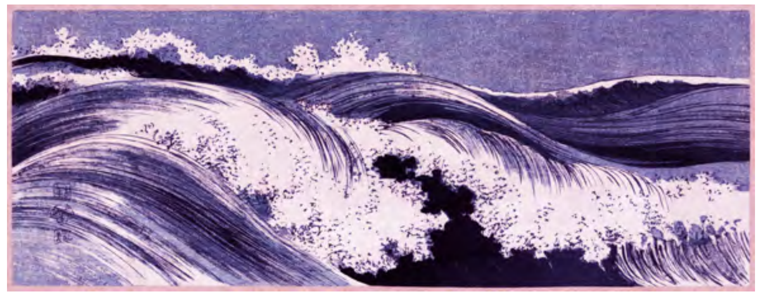
rs. print

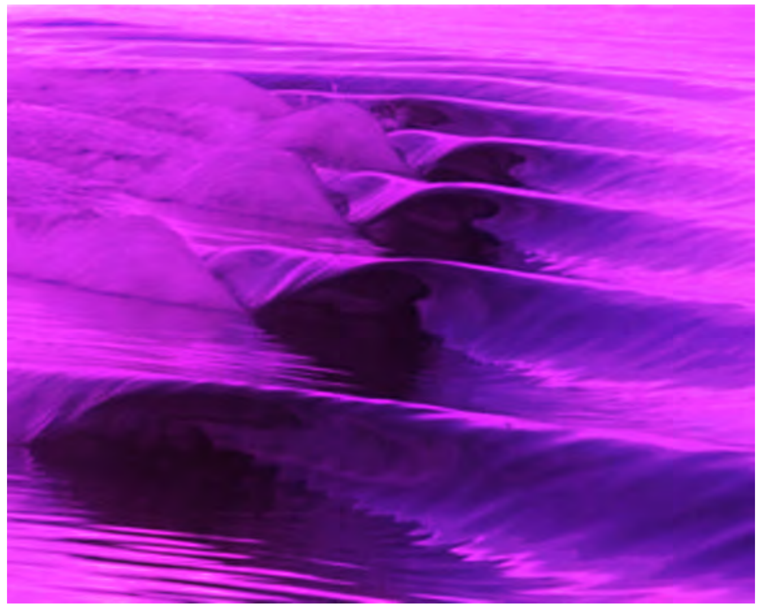
//////

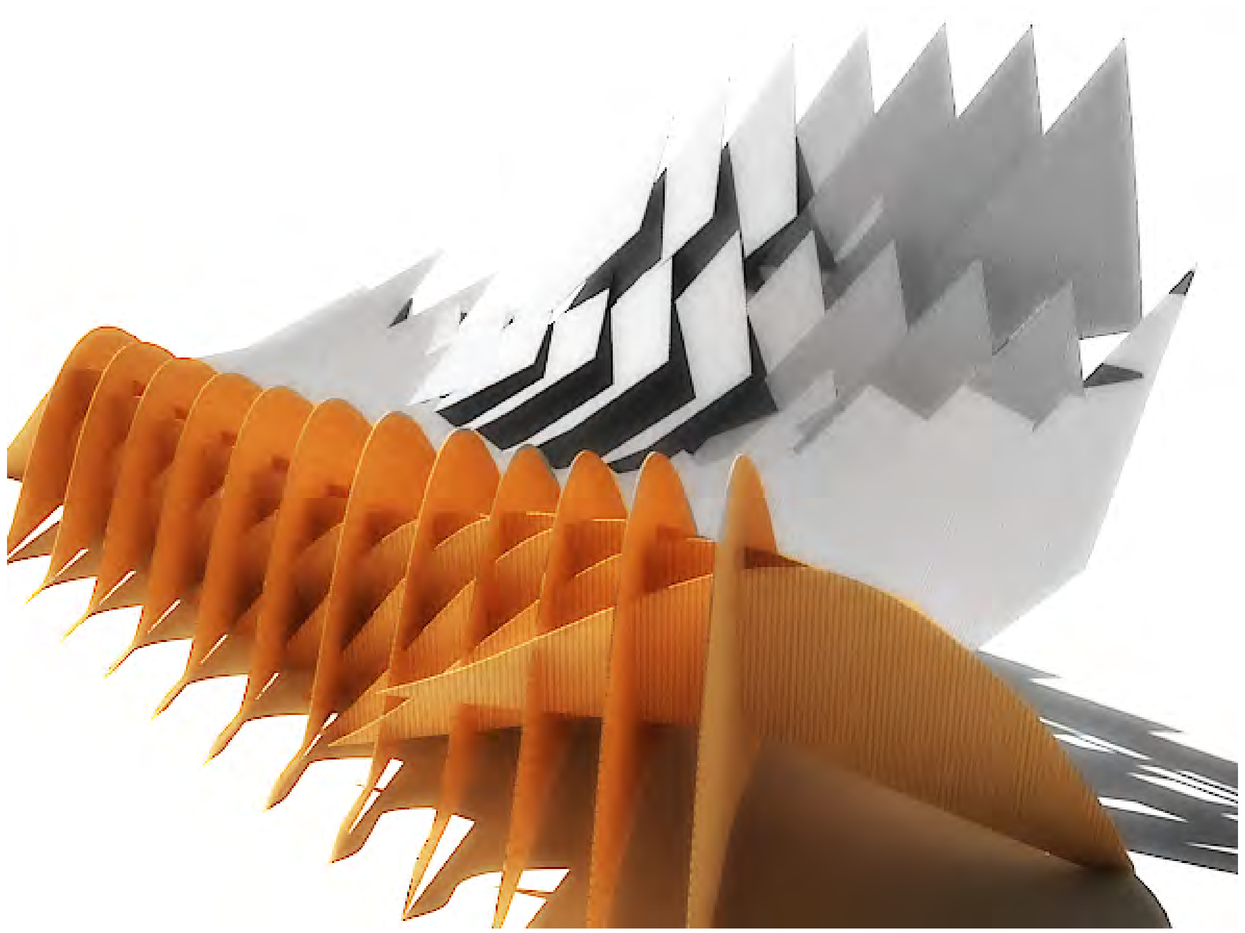
A large quantity of a flowing substance.

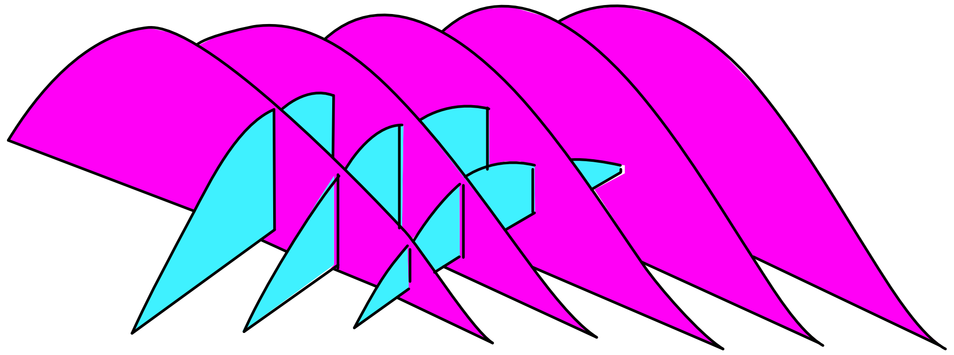
//////

The River









Design Criteria

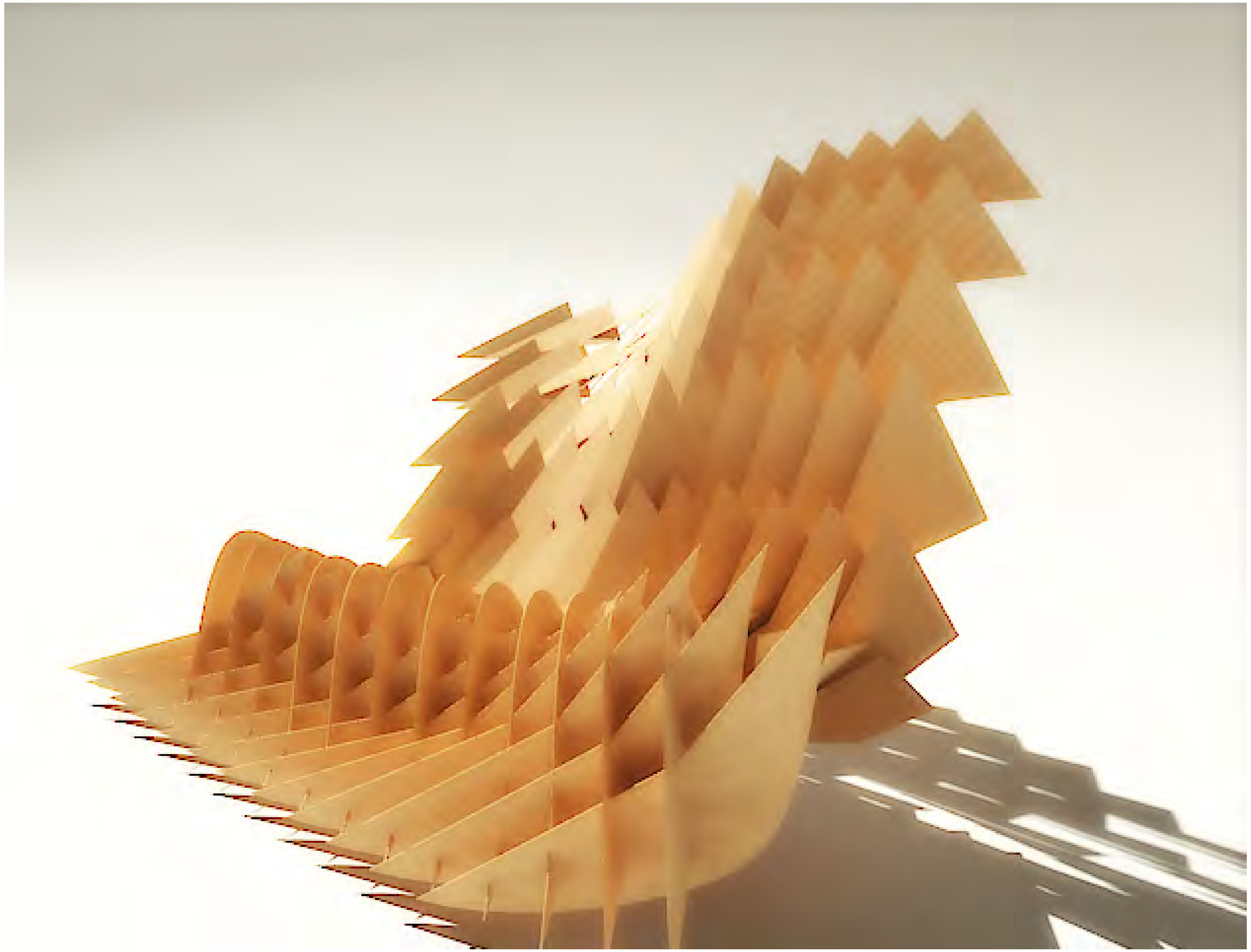
rs. print

//////

A large quantity of a flowing substance.

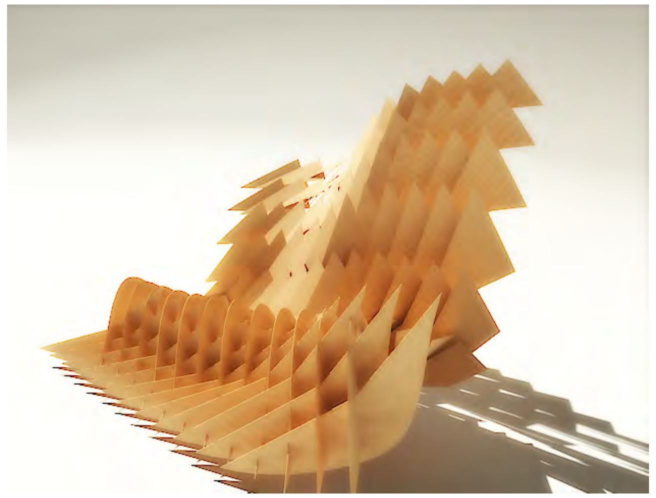
//////

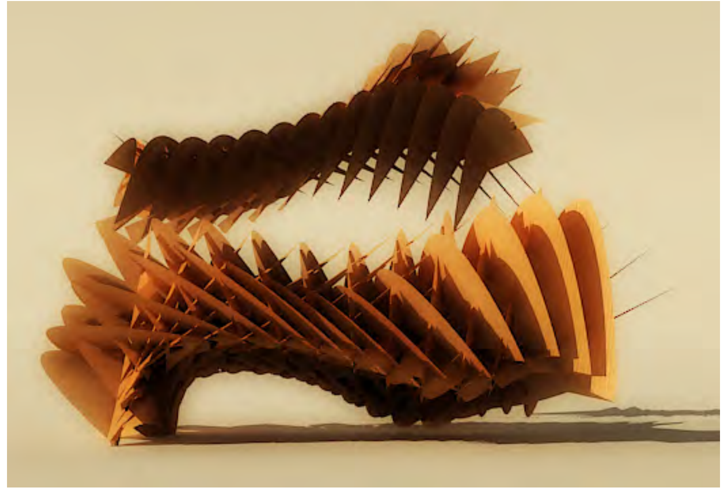
The River

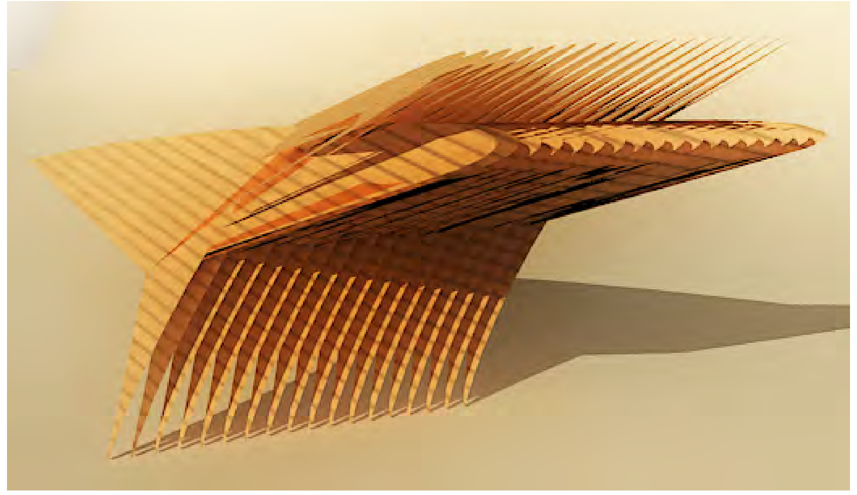


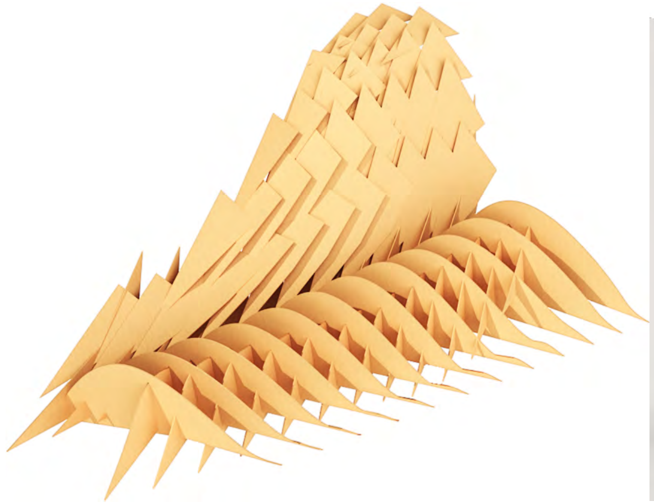
"Up the river, Up the mountain"

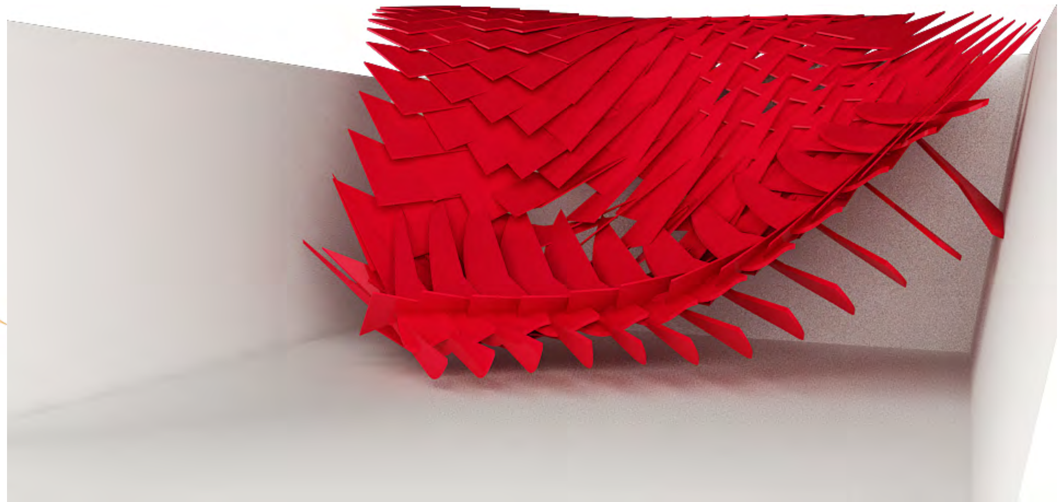
AA VISITING SCHOOL BOGOTÁ 2013

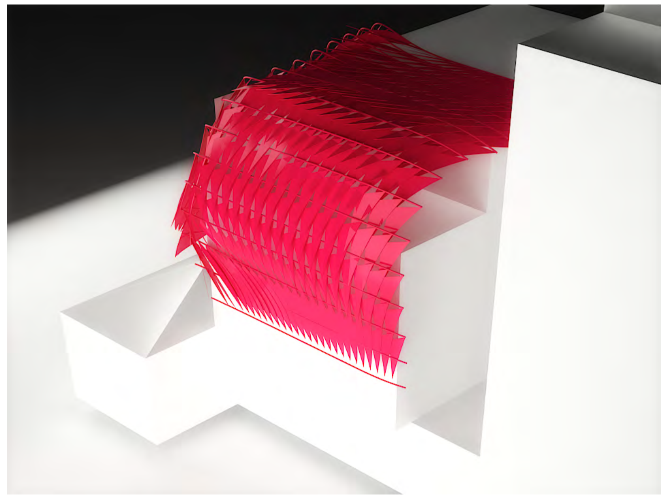


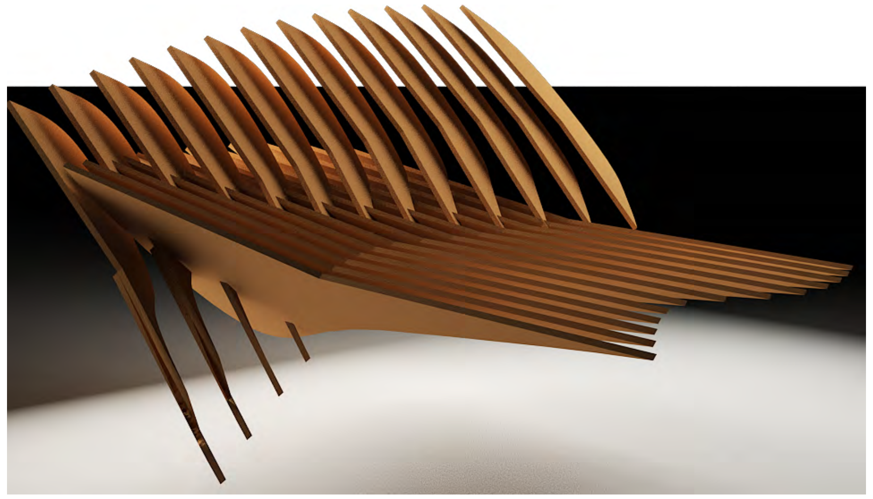


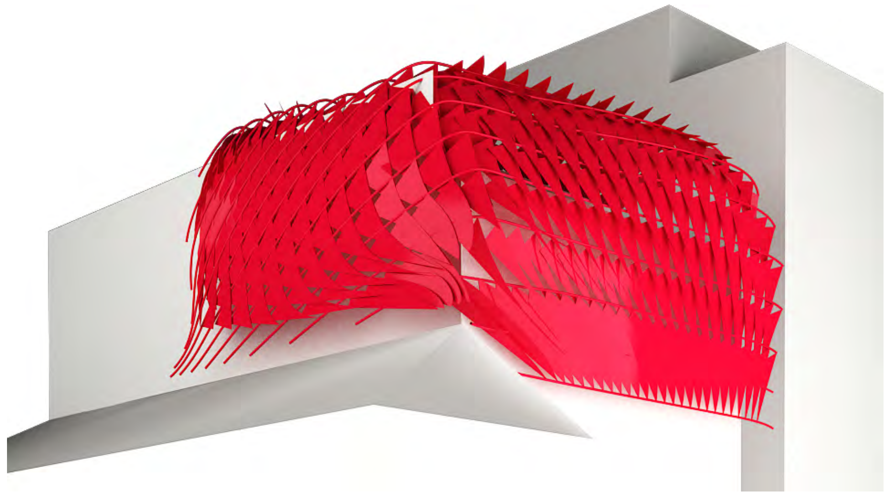


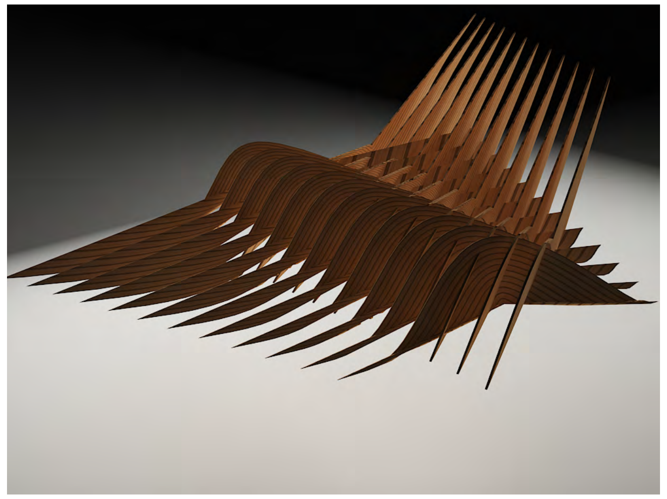


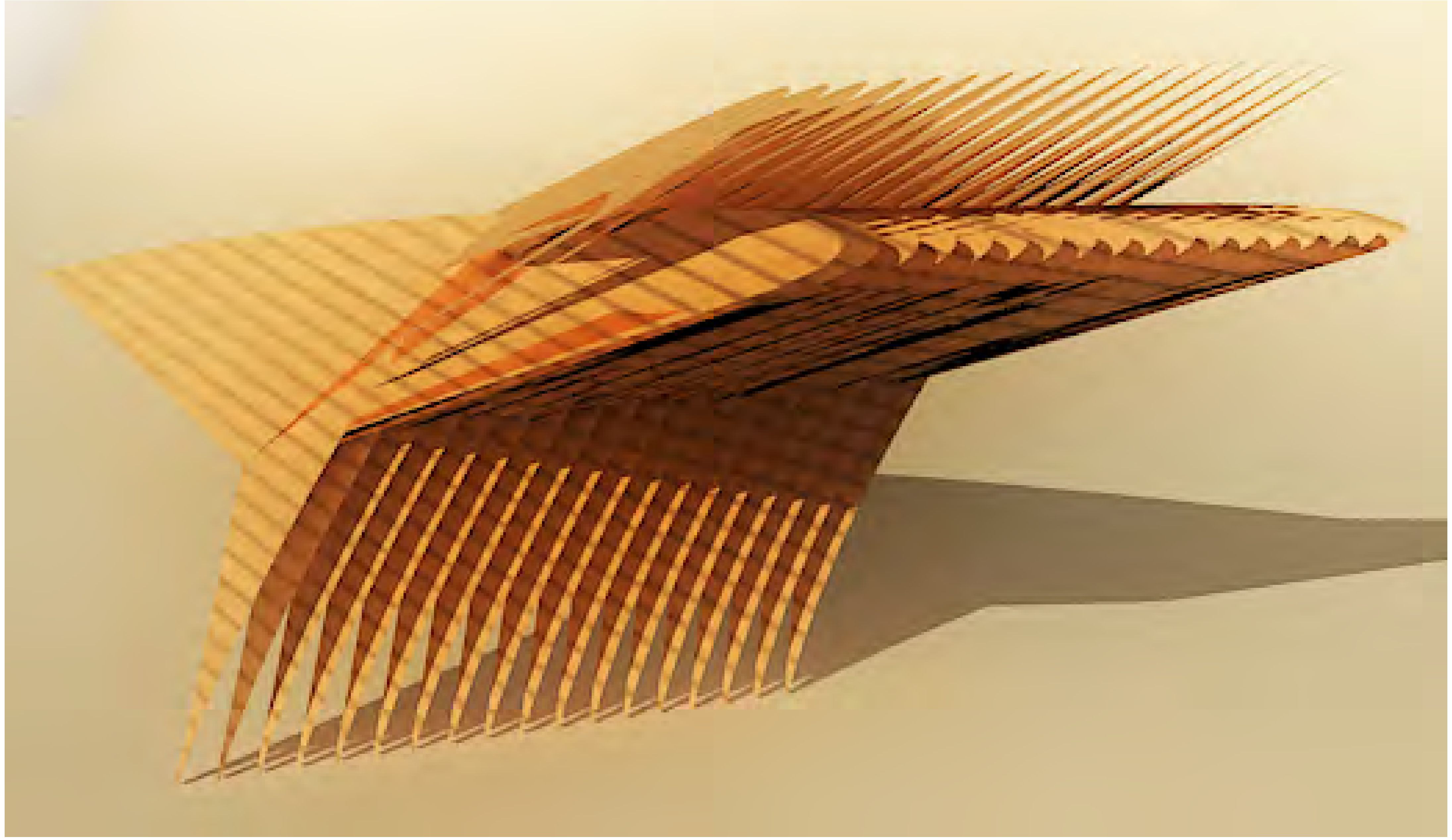










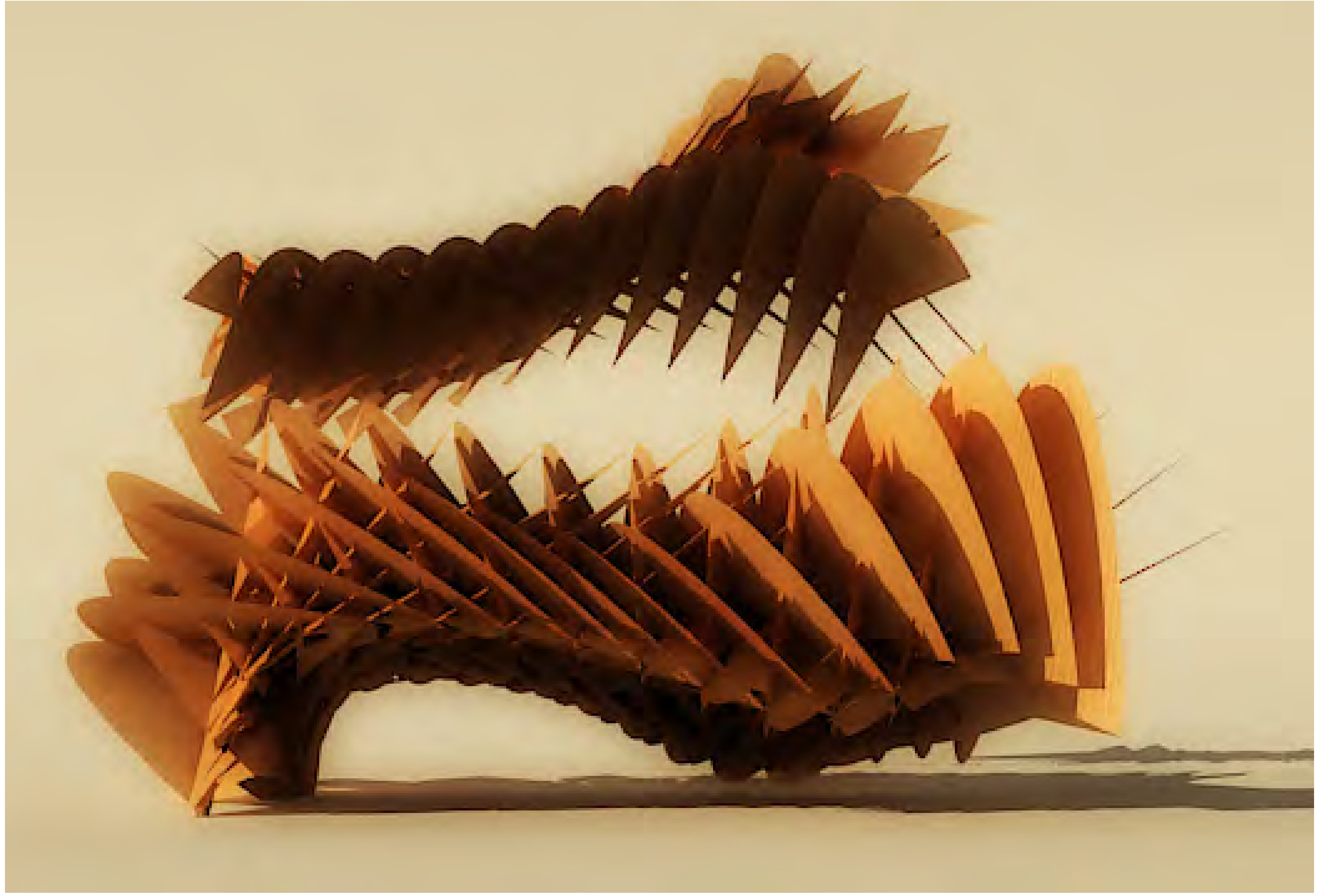


rs. print

//////

Arq. Eduardo Cardenas P.

//////

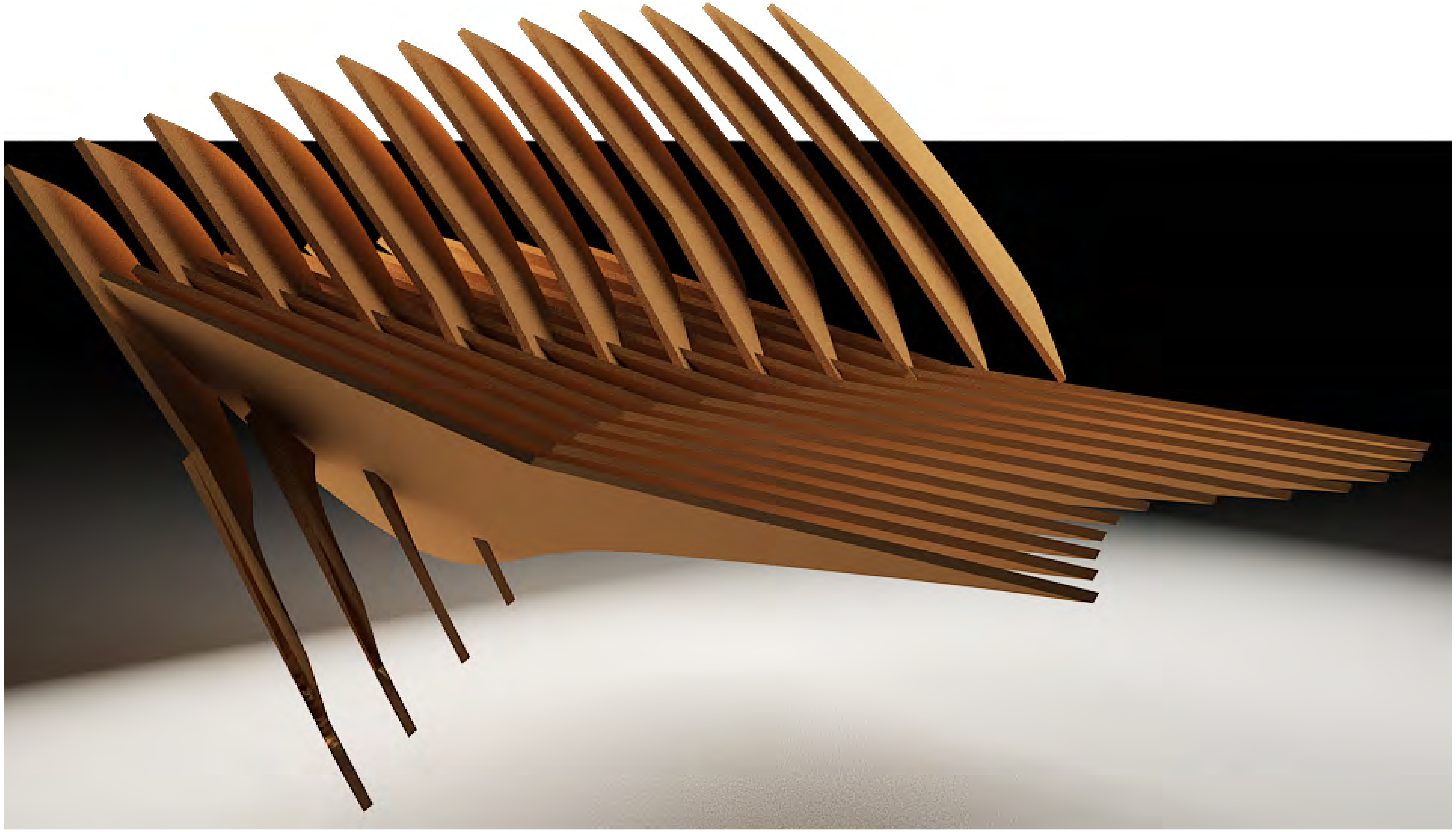


rs. print

//////

Arq. Nadia Mendez

//////

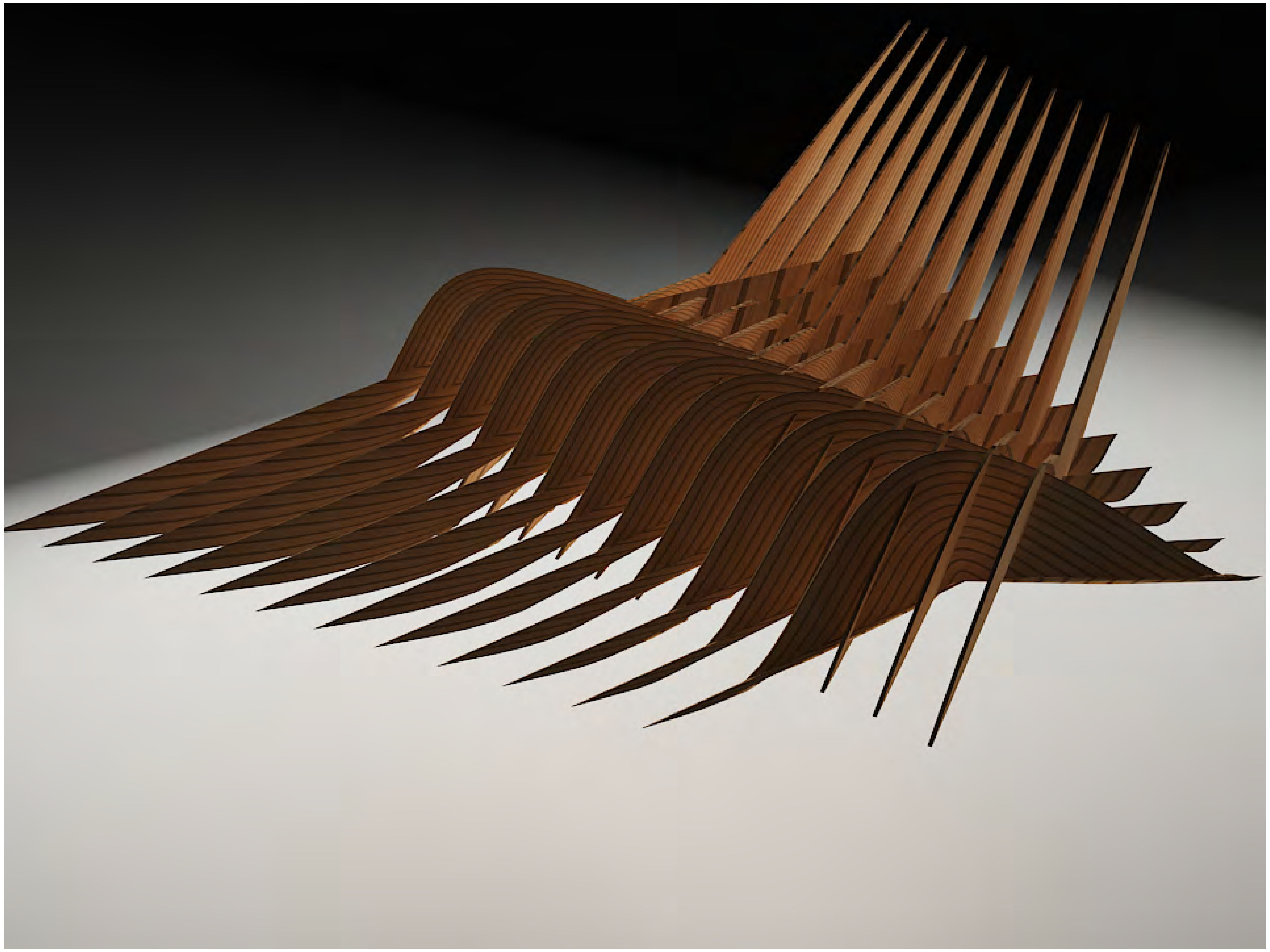


rs. print

//////

Arq. Daniela Vega

//////

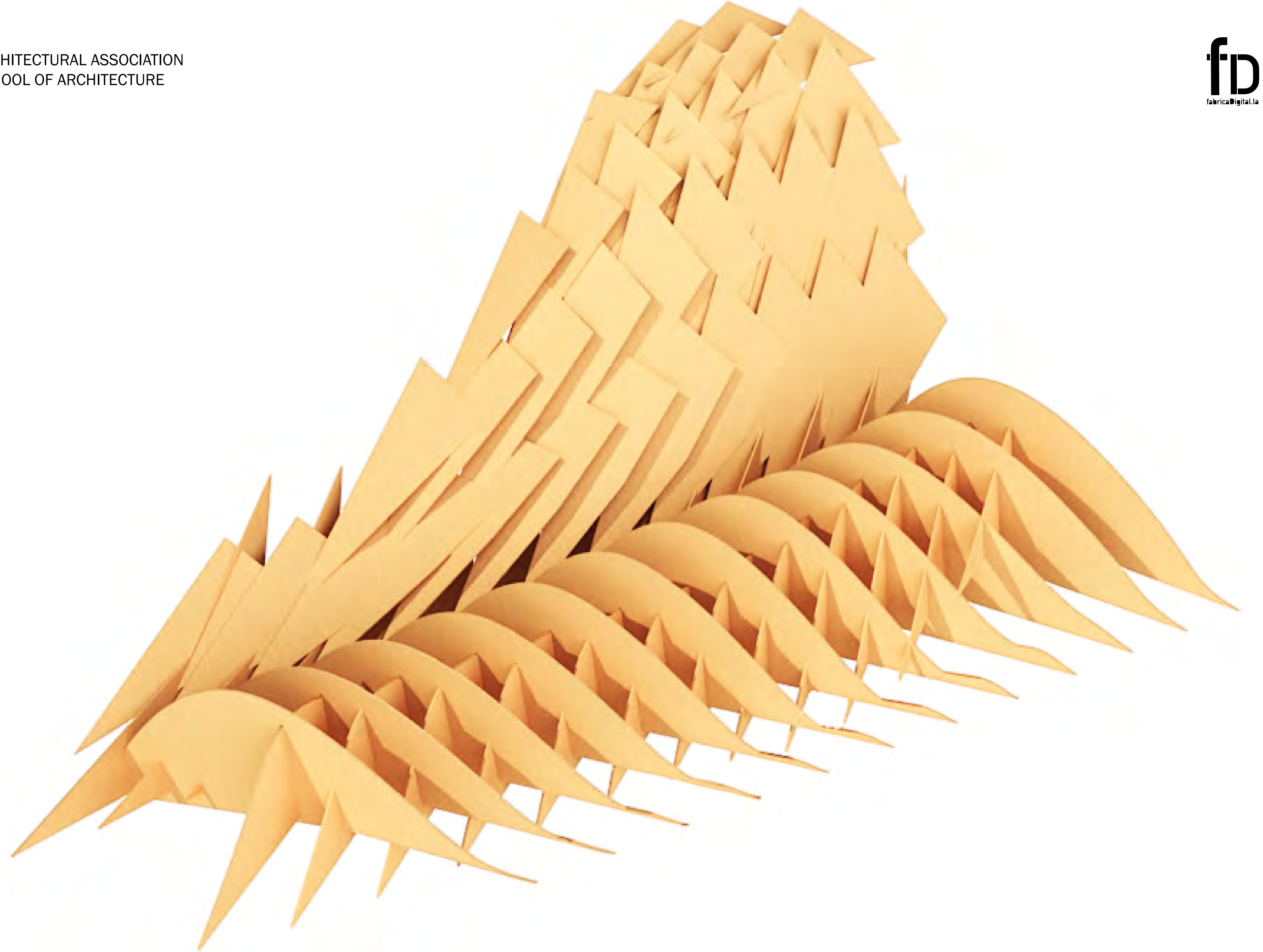


rs. print

//////

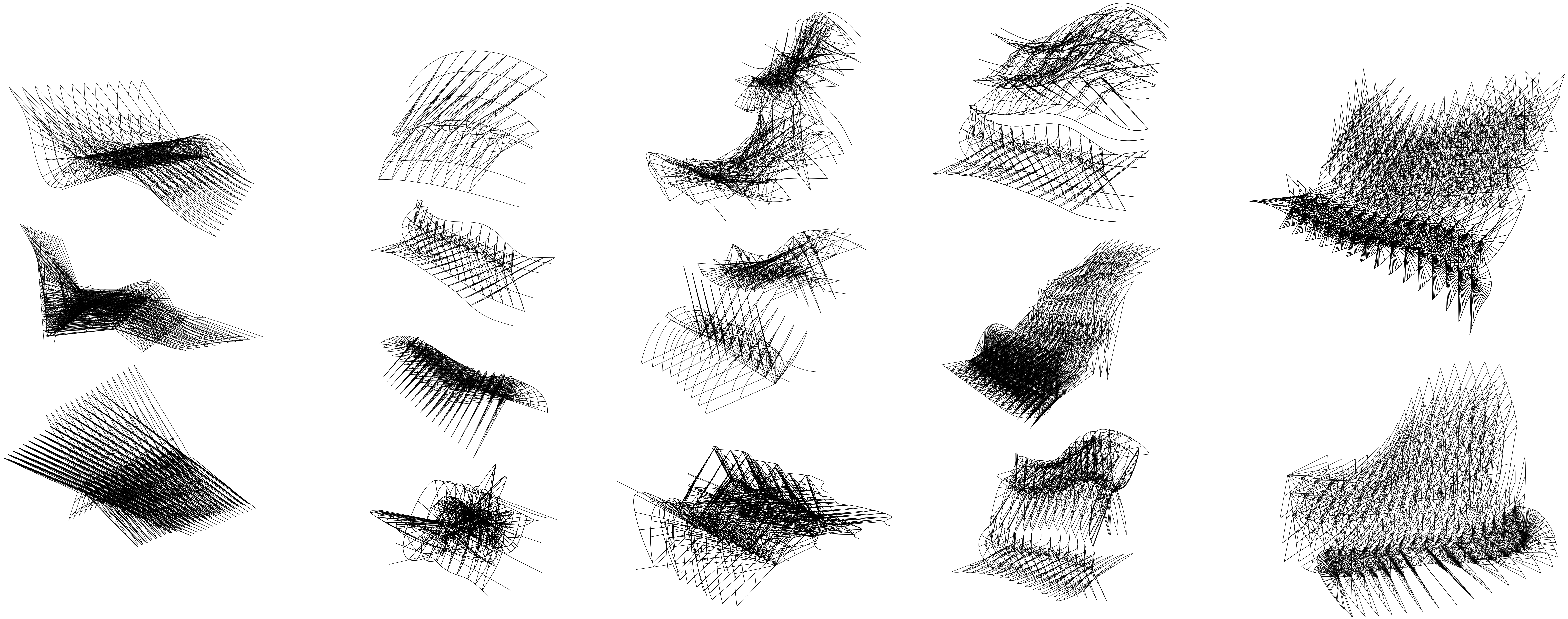
Arq. Nicolas Díaz B.

//////



"Up the river, Up the mountain"

AA VISITING SCHOOL BOGOTÁ 2013

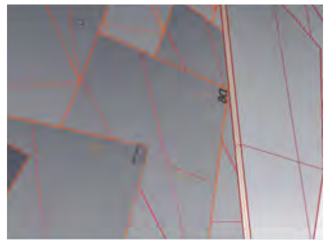


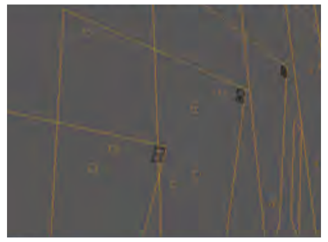
rs. print

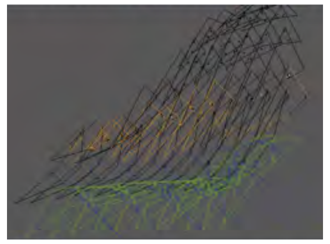
//////

Catalogue

//////







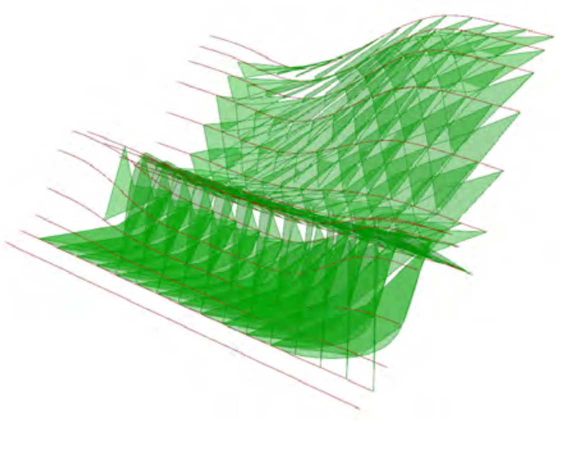
rs. print

/////

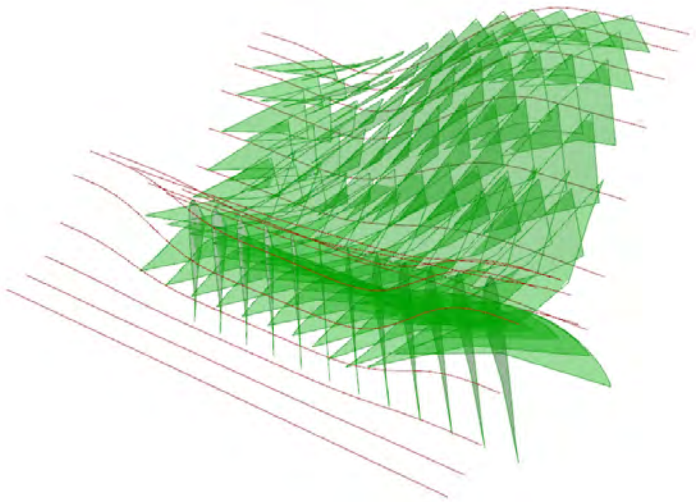
Rhino Phase

/////

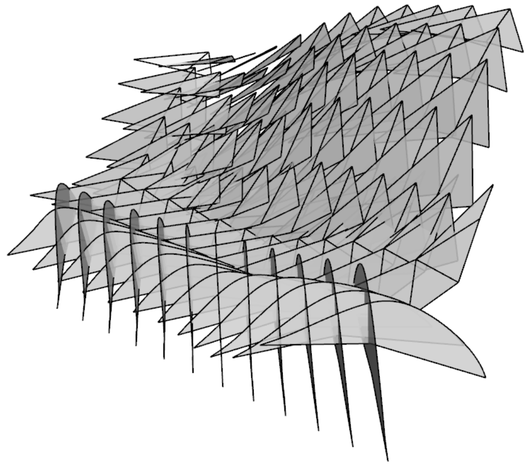
Original code output



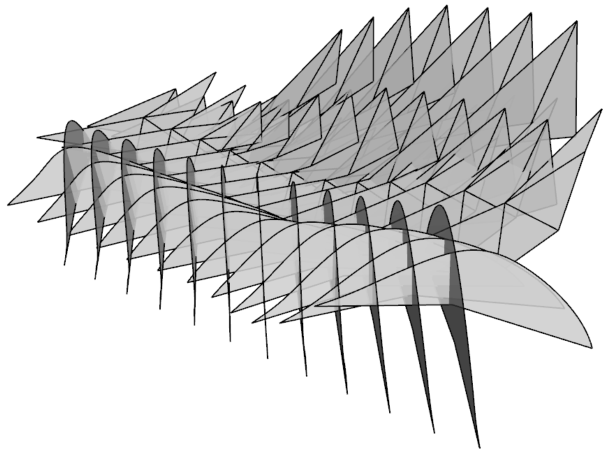
Model after manipulation in Grasshopper

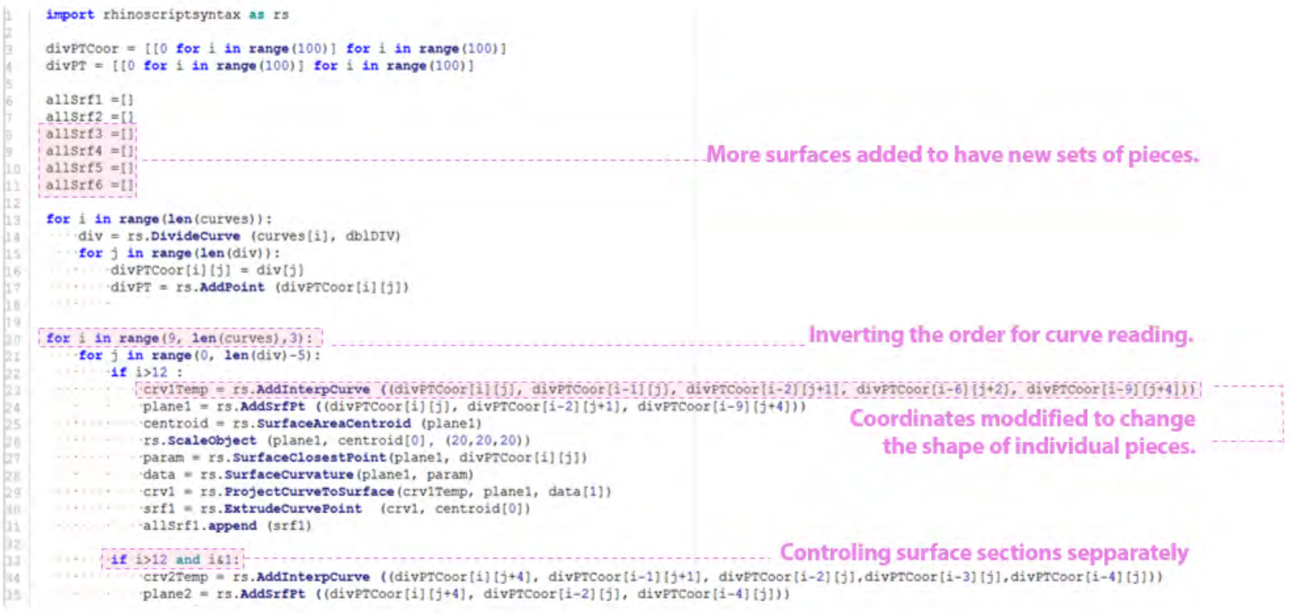


"Baked" model



Final Rhino model





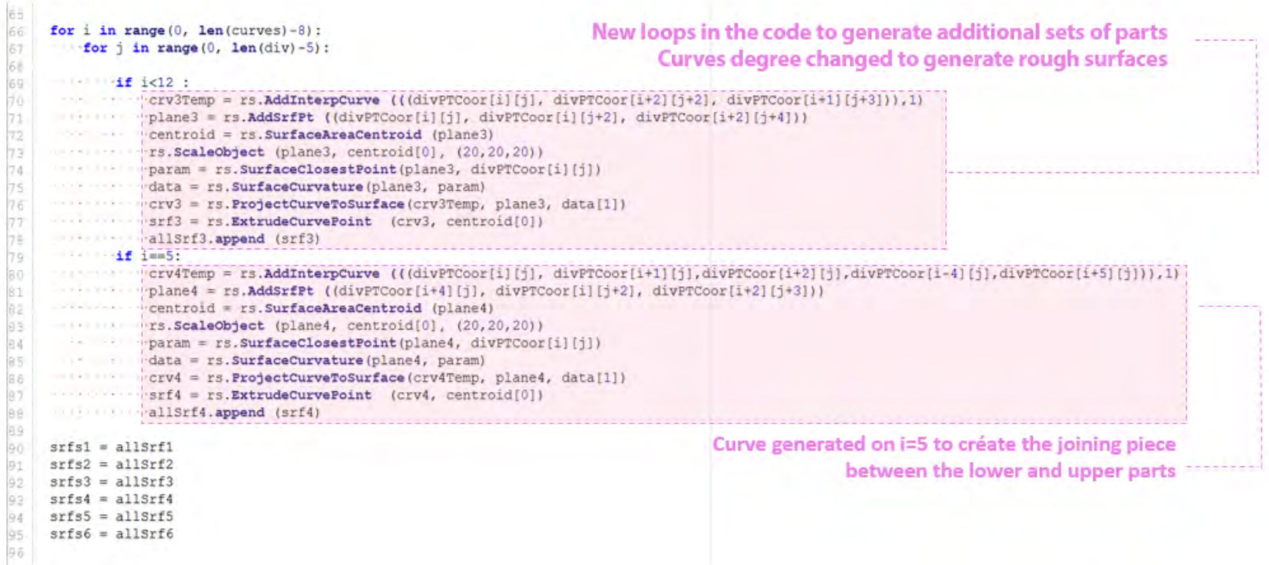
```python
import rhinoscriptsyntax as rs

divPTCoor = [[0 for i in range(100)] for i in range(100)]
divPT = [[0 for i in range(100)] for i in range(100)]

allSrf1 =[]
allSrf2 =[]
allSrf3 =[]
allSrf4 =[]
allSrf5 =[]
allSrf6 =[]

for i in range(len(curves)):
    div = rs.DivideCurve (curves[i], dblDIV)
    for j in range(len(div)):
        divPTCoor[i][j] = div[j]
        divPT = rs.AddPoint (divPTCoor[i][j])


for i in range(9, len(curves),3):
    for j in range(0, len(div)-5):
        if i>12 :
            crv1Temp = rs.AddInterpCurve ((divPTCoor[i][j], divPTCoor[i-1][j], divPTCoor[i-2][j+1], divPTCoor[i-6][j+2], divPTCoor[i-9][j+4]))
            plane1 = rs.AddSrfPt ((divPTCoor[i][j], divPTCoor[i-2][j+1], divPTCoor[i-9][j+4]))
            centroid = rs.SurfaceAreaCentroid (plane1)
            rs.ScaleObject (plane1, centroid[0], (20,20,20))
            param = rs.SurfaceClosestPoint(plane1, divPTCoor[i][j])
            data = rs.SurfaceCurvature(plane1, param)
            crv1 = rs.ProjectCurveToSurface(crv1Temp, plane1, data[1])
            srf1 = rs.ExtrudeCurvePoint  (crv1, centroid[0])
            allSrf1.append (srf1)

        if i>12 and i&1:
            crv2Temp = rs.AddInterpCurve ((divPTCoor[i][j+4], divPTCoor[i-1][j+1], divPTCoor[i-2][j],divPTCoor[i-3][j],divPTCoor[i-4][j]))
            plane2 = rs.AddSrfPt ((divPTCoor[i][j+4], divPTCoor[i-2][j], divPTCoor[i-4][j]))
```

More surfaces added to have new sets of pieces.

Inverting the order for curve reading.

Coordinates moddified to change the shape of individual pieces.

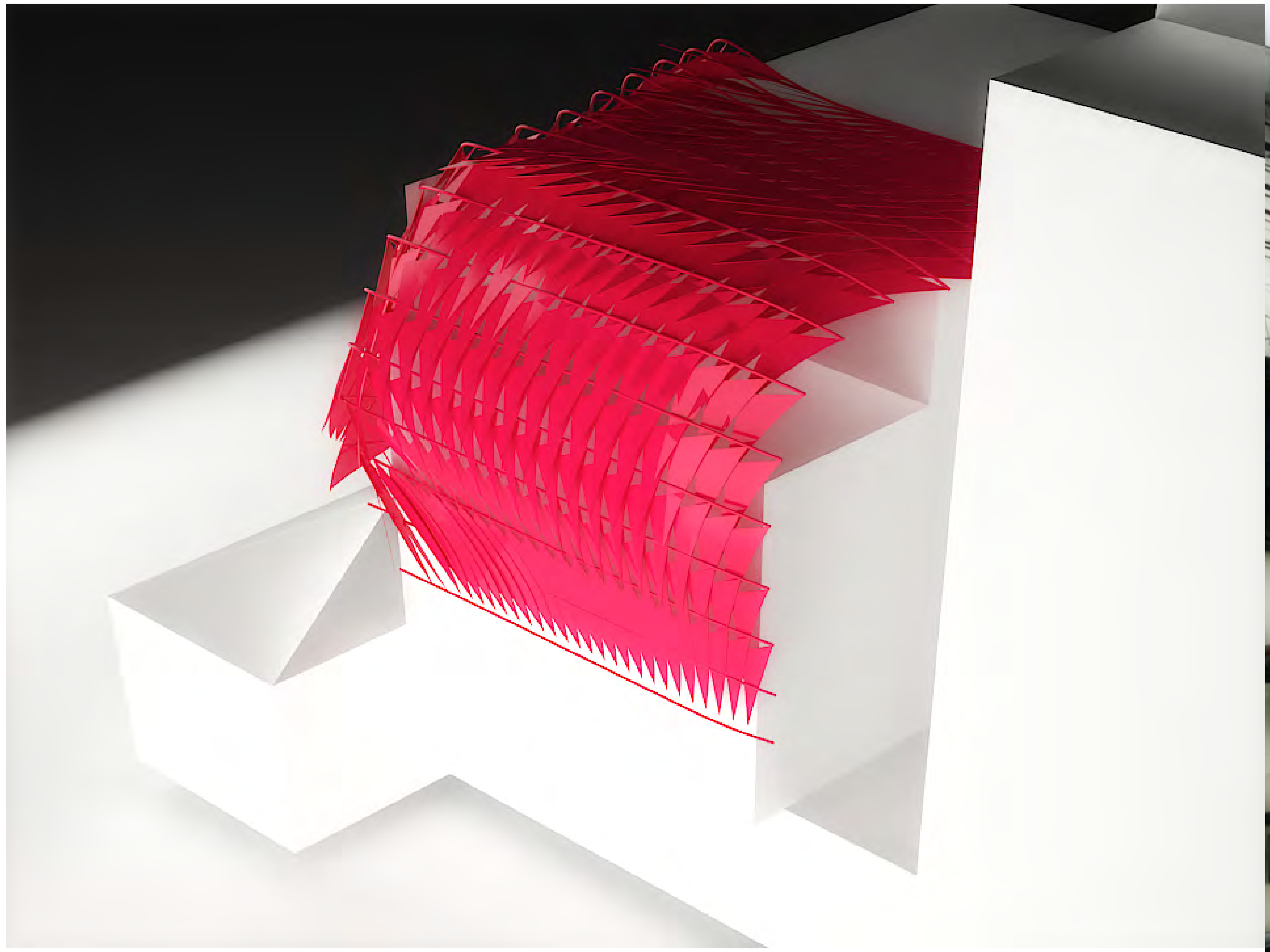
Controling surface sections sepparately

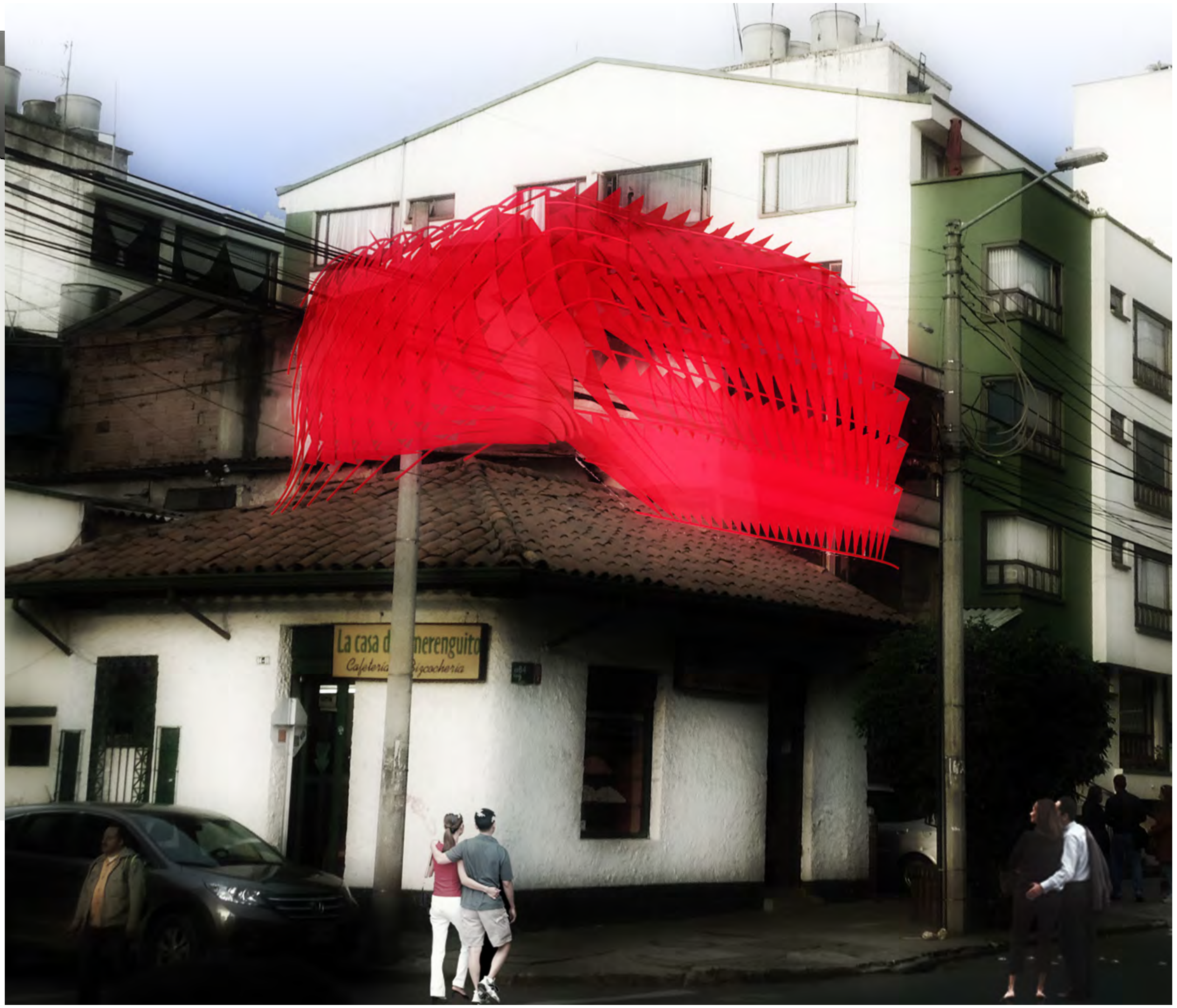


```python
for i in range(0, len(curves)-8):
    for j in range(0, len(div)-5):

        if i<12 :
            crv3Temp = rs.AddInterpCurve (((divPTCoor[i][j], divPTCoor[i+2][j+2], divPTCoor[i+1][j+3])),1)
            plane3 = rs.AddSrfPt ((divPTCoor[i][j], divPTCoor[i][j+2], divPTCoor[i+2][j+4]))
            centroid = rs.SurfaceAreaCentroid (plane3)
            rs.ScaleObject (plane3, centroid[0], (20,20,20))
            param = rs.SurfaceClosestPoint(plane3, divPTCoor[i][j])
            data = rs.SurfaceCurvature(plane3, param)
            crv3 = rs.ProjectCurveToSurface(crv3Temp, plane3, data[1])
            srf3 = rs.ExtrudeCurvePoint  (crv3, centroid[0])
            allSrf3.append (srf3)
        if i==5:
            crv4Temp = rs.AddInterpCurve (((divPTCoor[i][j], divPTCoor[i+1][j],divPTCoor[i+2][j],divPTCoor[i-4][j],divPTCoor[i+5][j])),1)
            plane4 = rs.AddSrfPt ((divPTCoor[i+4][j], divPTCoor[i][j+2], divPTCoor[i+2][j+3]))
            centroid = rs.SurfaceAreaCentroid (plane4)
            rs.ScaleObject (plane4, centroid[0], (20,20,20))
            param = rs.SurfaceClosestPoint(plane4, divPTCoor[i][j])
            data = rs.SurfaceCurvature(plane4, param)
            crv4 = rs.ProjectCurveToSurface(crv4Temp, plane4, data[1])
            srf4 = rs.ExtrudeCurvePoint  (crv4, centroid[0])
            allSrf4.append (srf4)

srfs1 = allSrf1
srfs2 = allSrf2
srfs3 = allSrf3
srfs4 = allSrf4
srfs5 = allSrf5
srfs6 = allSrf6
```

New loops in the code to generate additional sets of parts
Curves degree changed to generate rough surfaces

Curve generated on i=5 to créate the joining piece between the lower and upper parts



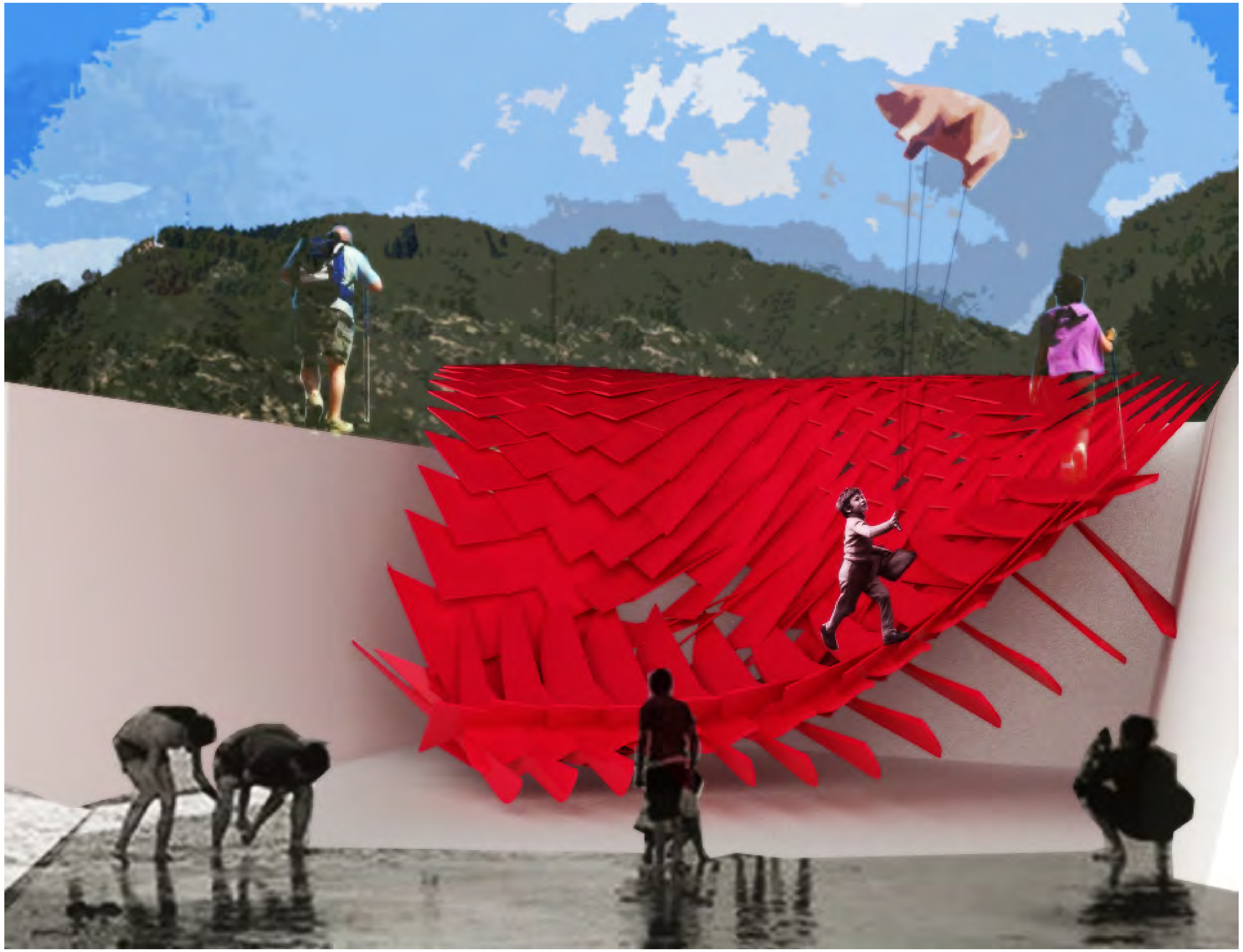


rs. print

/////

La Agencia Proposal

/////

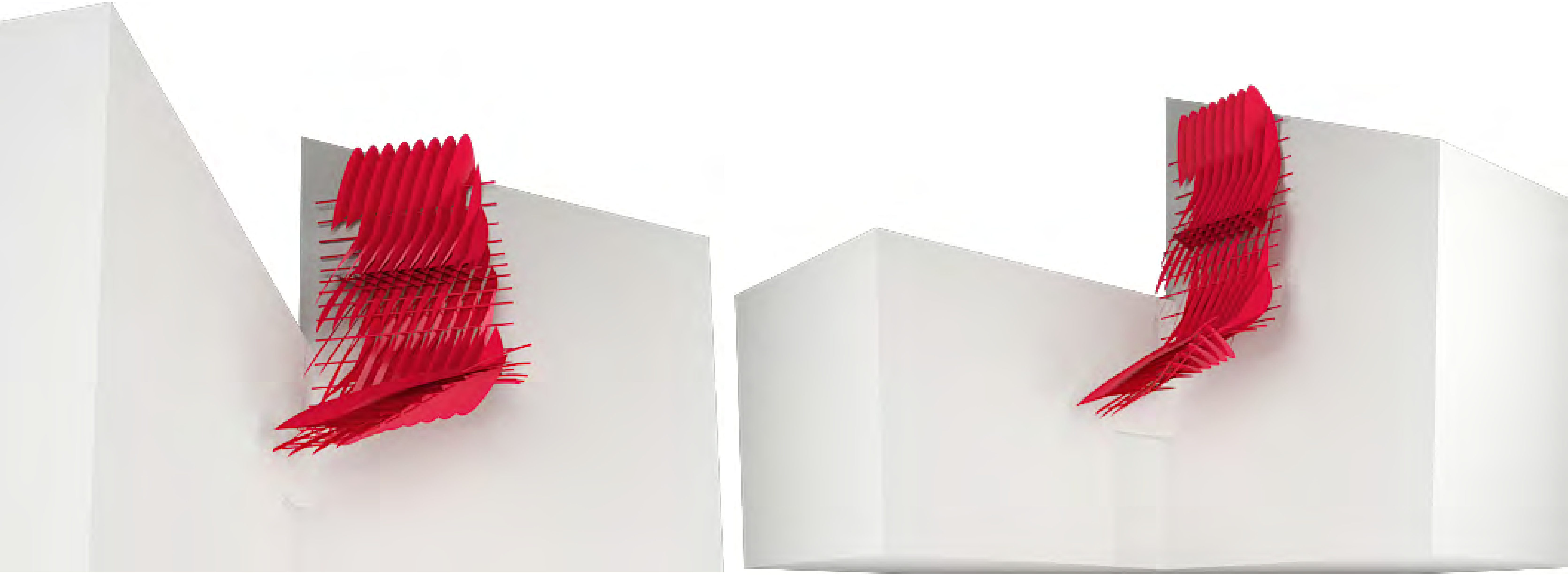


rs. print

//////

Belen Patio House proposal

//////



rs. print

//////

Fabrica Digital proposal

//////