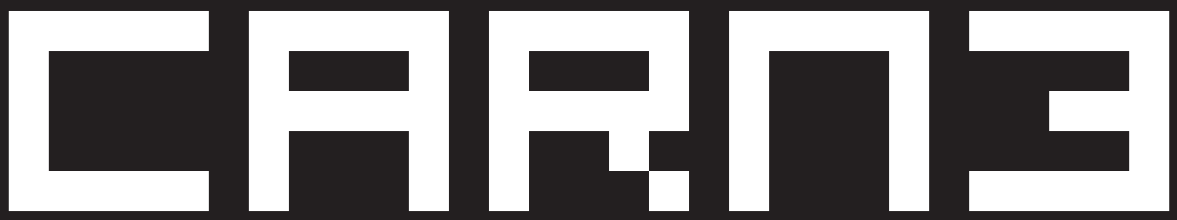

CARN3

CARN3 racconta il percorso formativo dello studio OFL architecture.

CARN3 rivela le interiora e i muscoli dell'architettura, come un composto di pezzi triturati in una promiscuità d'arte.
Immagini visionarie e modelli architettonici accompagnano lo spettatore in un percorso unico nel suo genere grazie ad un allestimento sterile che ne esalta le capacità evocative dei progetti.

CARN3 indaga in maniera originale sulle nuove possibilità abitative strizzando un occhio alle nuove tendenze ecoenergetiche con lo scopo di sovvertirle per raggiungere nuovi limiti costruttivi.
Forme fluide e visionarie, poi, saranno lo stimolo per una riflessione sul futuro prossimo dell'essere umano.

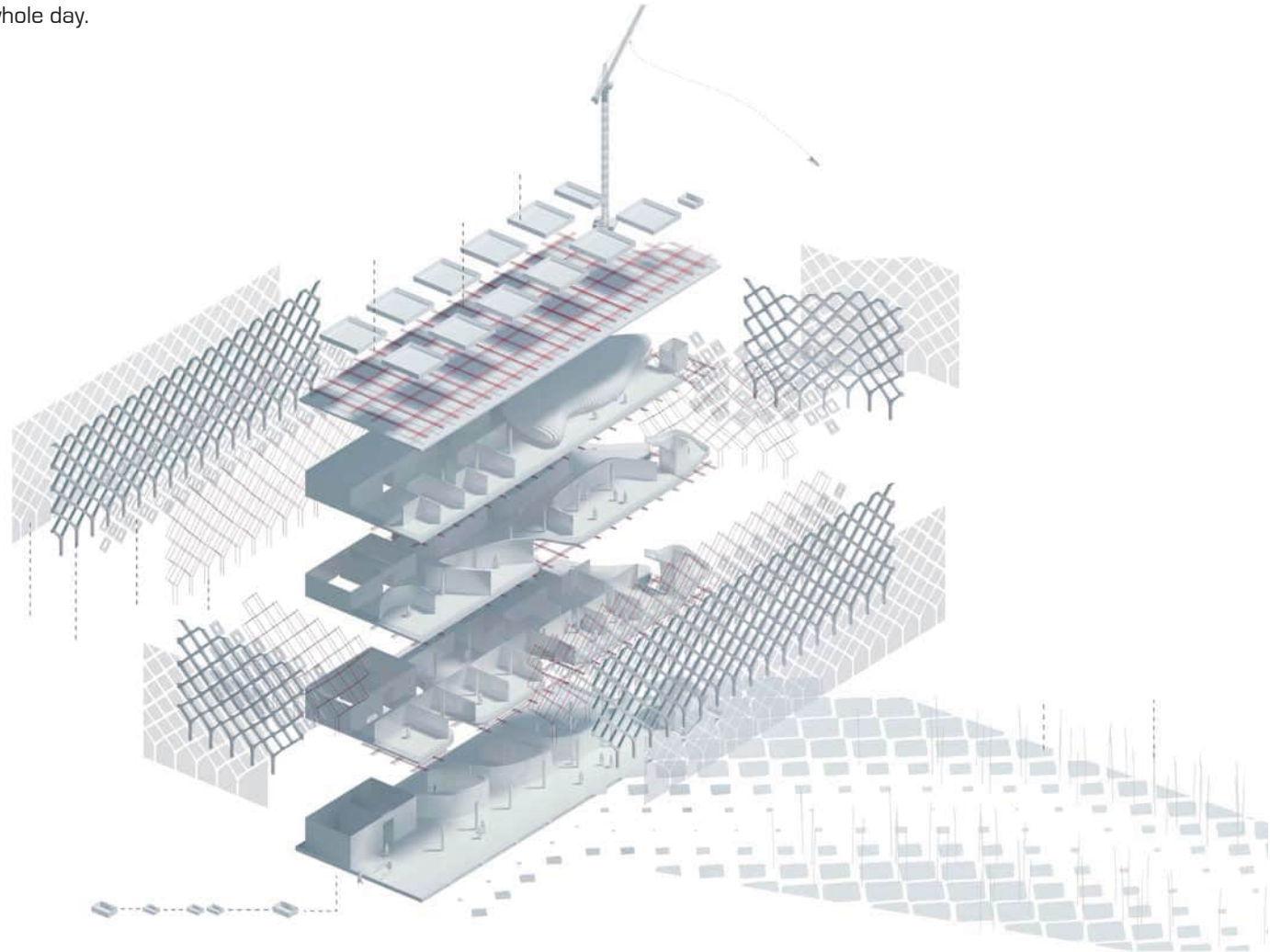
CENTRO EDILE ANDREA PALLADIO

The new Andrea Palladio Building Center is made of a structural skin evoking nature, like tree branches that by weaving themselves create a dynamic façade made of steel and glass. The arches on the ground floor evoke the architecture of the Renaissance, and suggest the idea of an ethical and emotional architecture. Steel and glass together, combined as something organic and dynamic, allow the building to be in harmony with the context. The shielding front is made of mirror glass panels with a thickness of 10 mm. They have the double function of reflecting the natural environment and protecting the building from the sun. The glazed turbots, generated by the external structure, and the plastic tubes that pervade the entire building have been designed in order to enable the sectors to obtain an adequate level of lighting comfort.

This building generates a metamorphosis with its surroundings because it reflects and emphasizes his beauty merging it with the context.

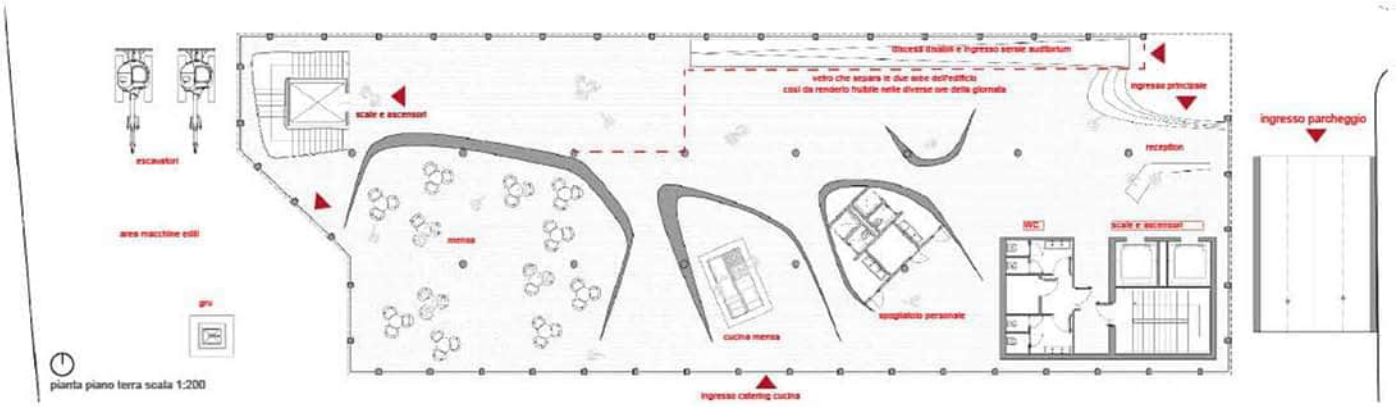
A first layer of “smart pipes” leads the air from outside to inside the building while a second layer of “smart pipes” uses rainwater collection tanks positioned on the roof of the building and spread, as roots of a tree, in the whole building. Both layers help to reduce the heat load of the front and to adjust the light input through different geometries of shielding for the whole day.

CREDITS
Plan’s title: Nuova Sede Centro Edile Andrea Palladio
Location: Vicenza
Project Design: Francesco Lipari
Project team: Francesco Lipari, Vanessa Todaro, Pierpaolo Pugliano, Gianni Esposito
Timing: design 2009
Web information: www.francescolipari.it

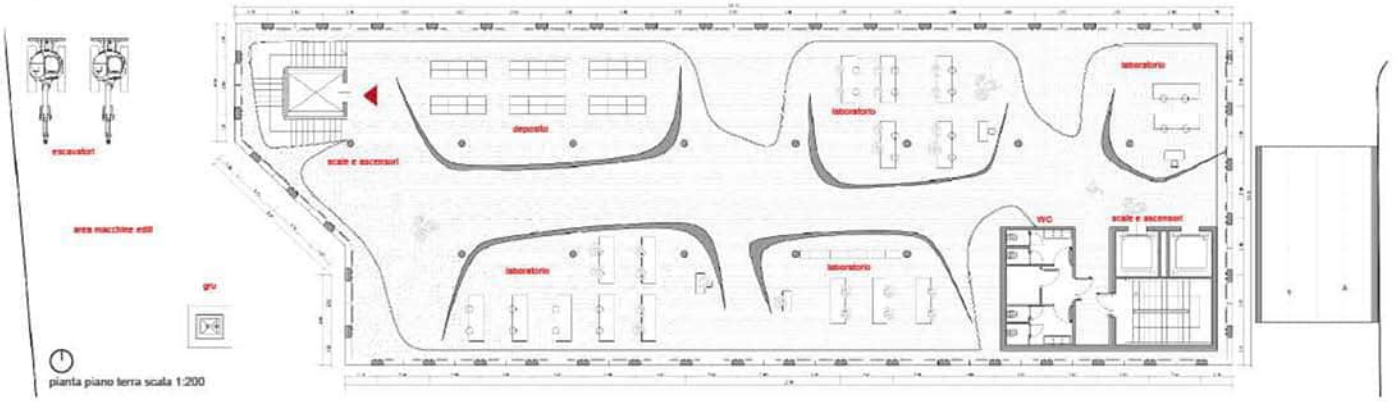



LABORATORY 1

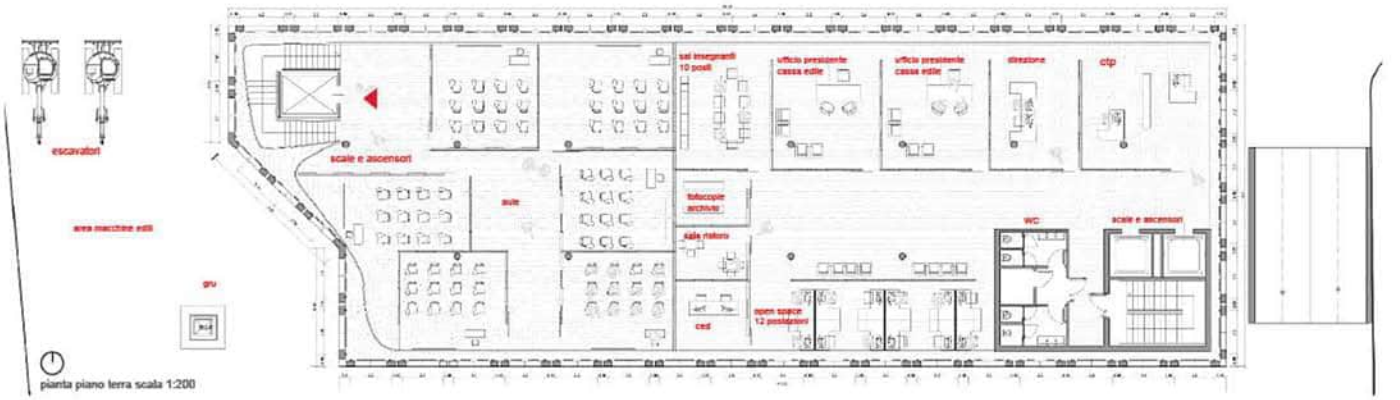

CENTRO EDILE
ANDREA PALLADIO



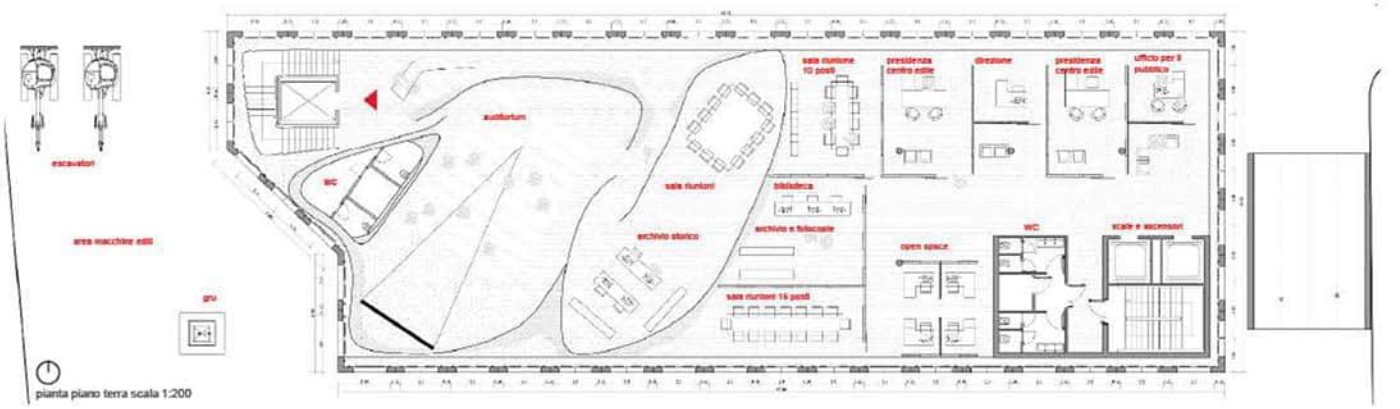
pianta piano terra scala 1:200



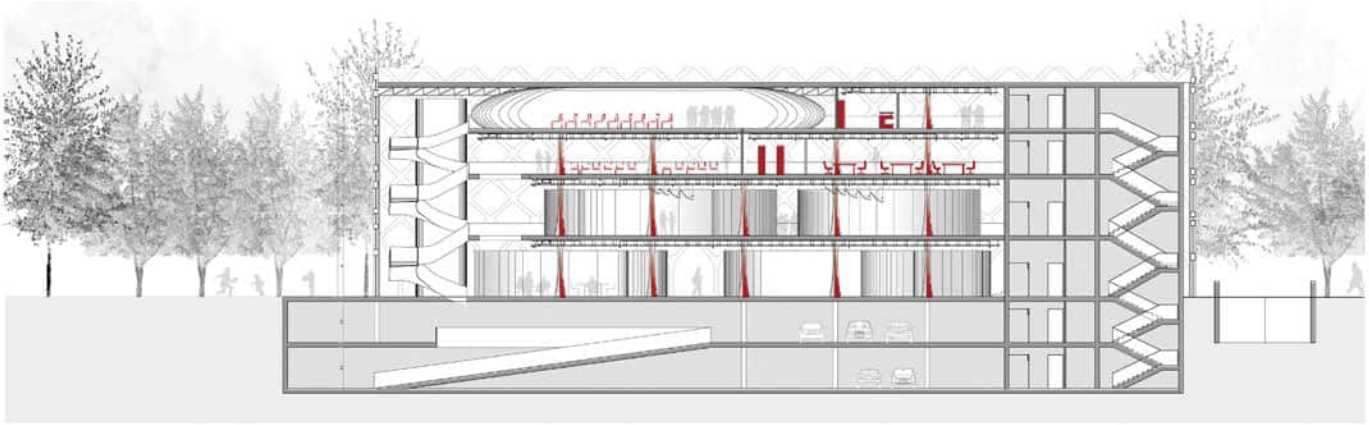
pianta piano terra scala 1:200

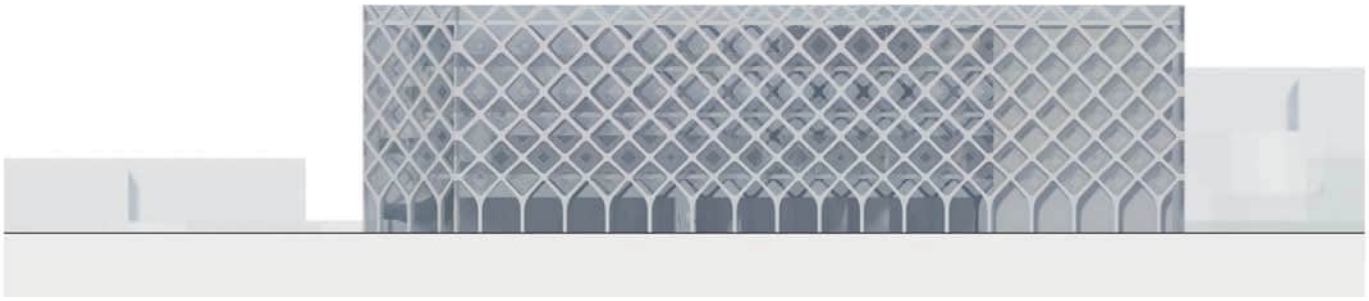


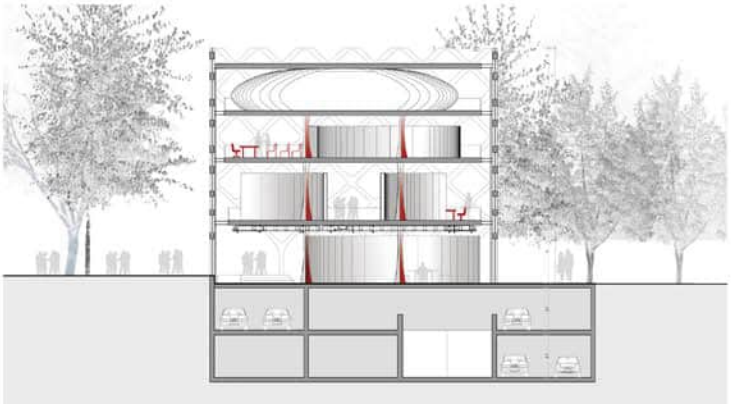
pianta piano terra scala 1:200

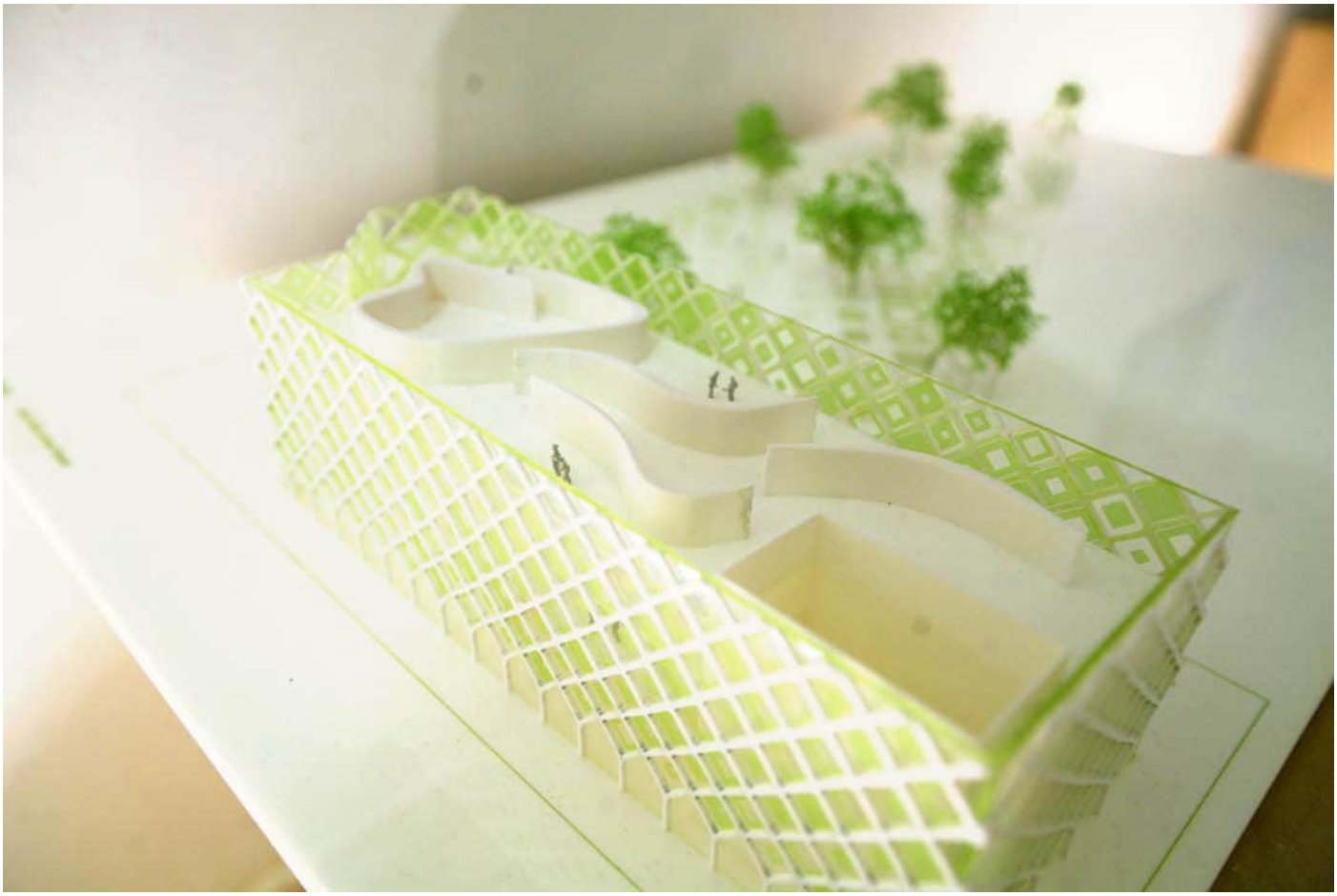


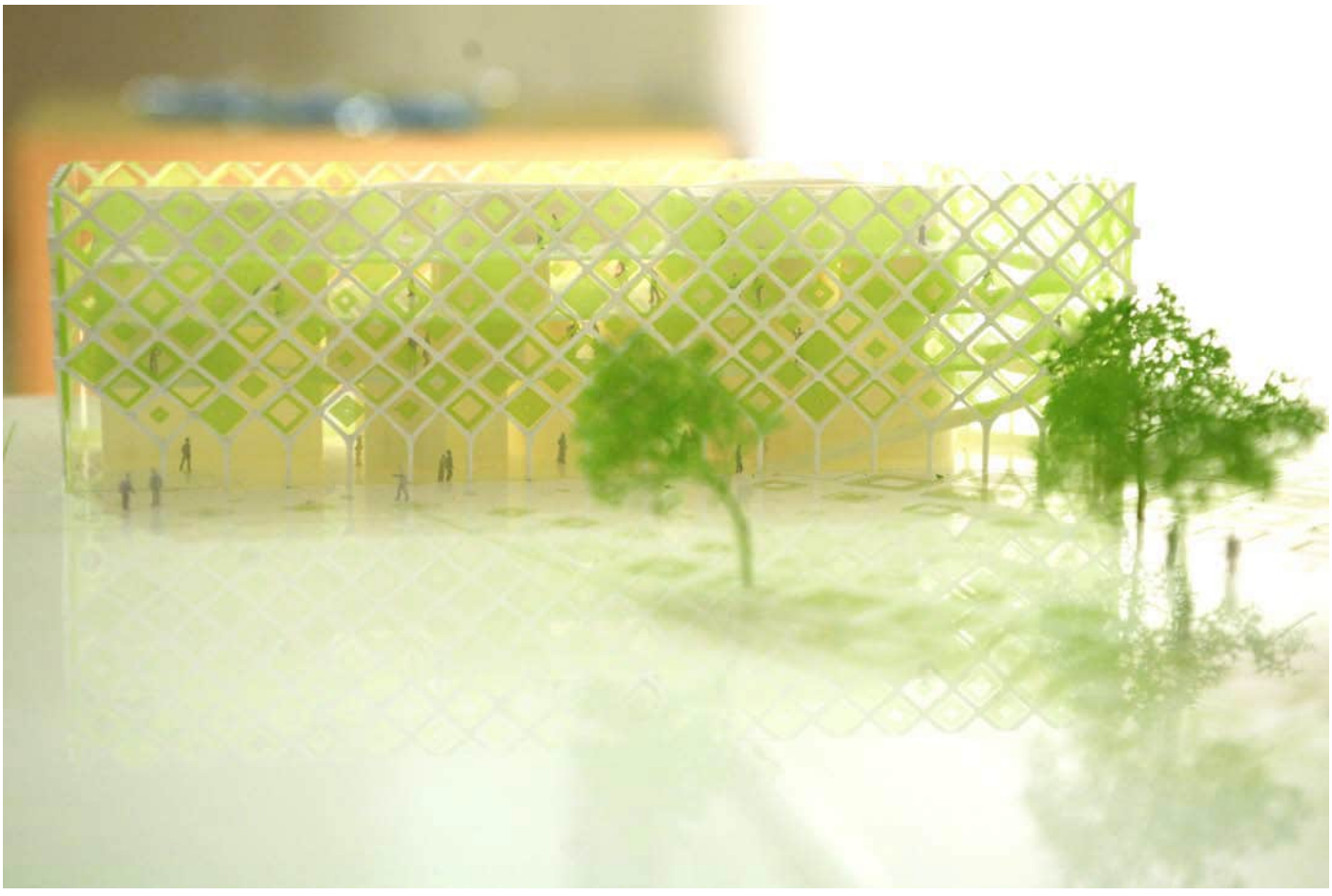
pianta piano terra scala 1:200

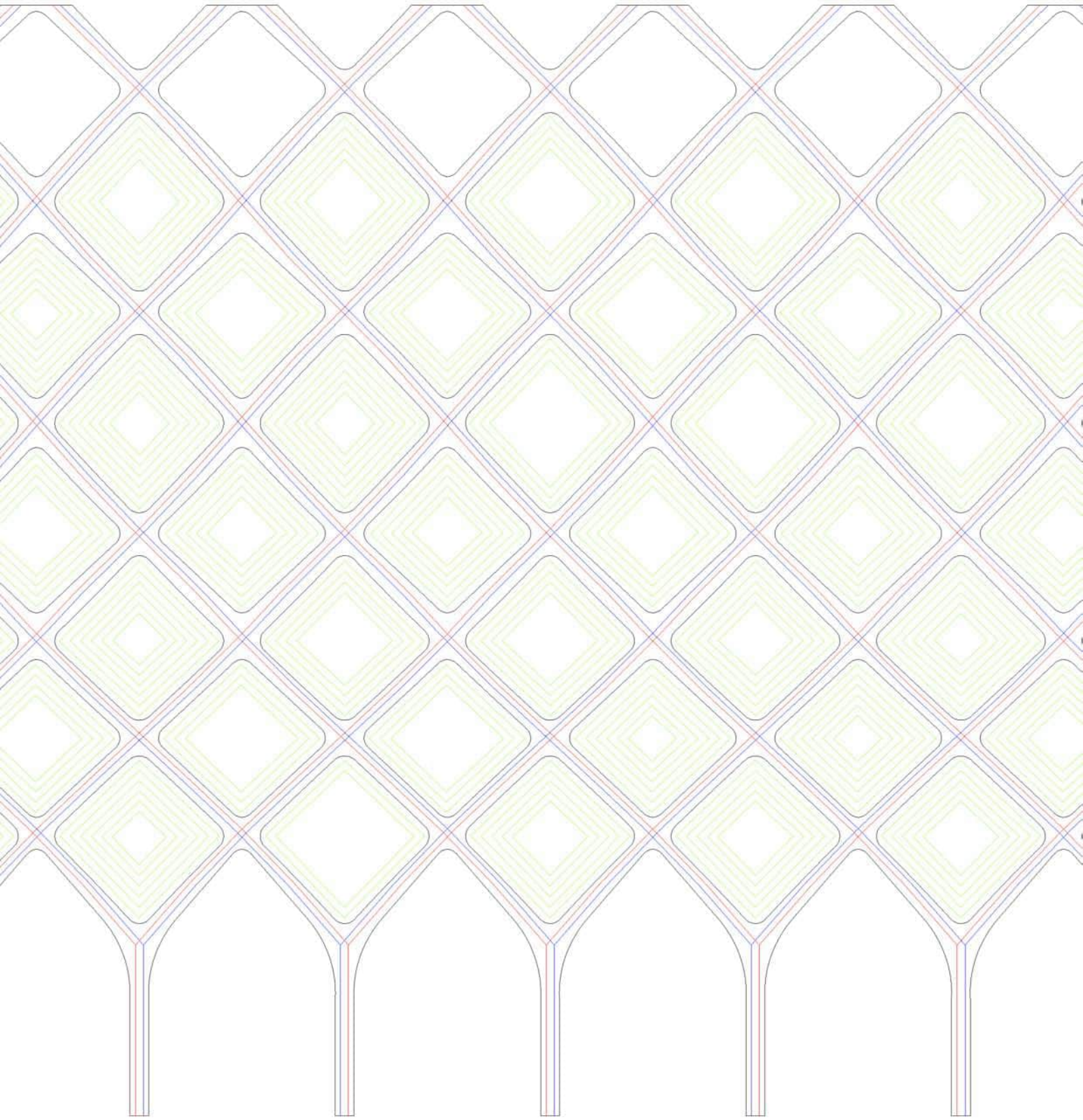


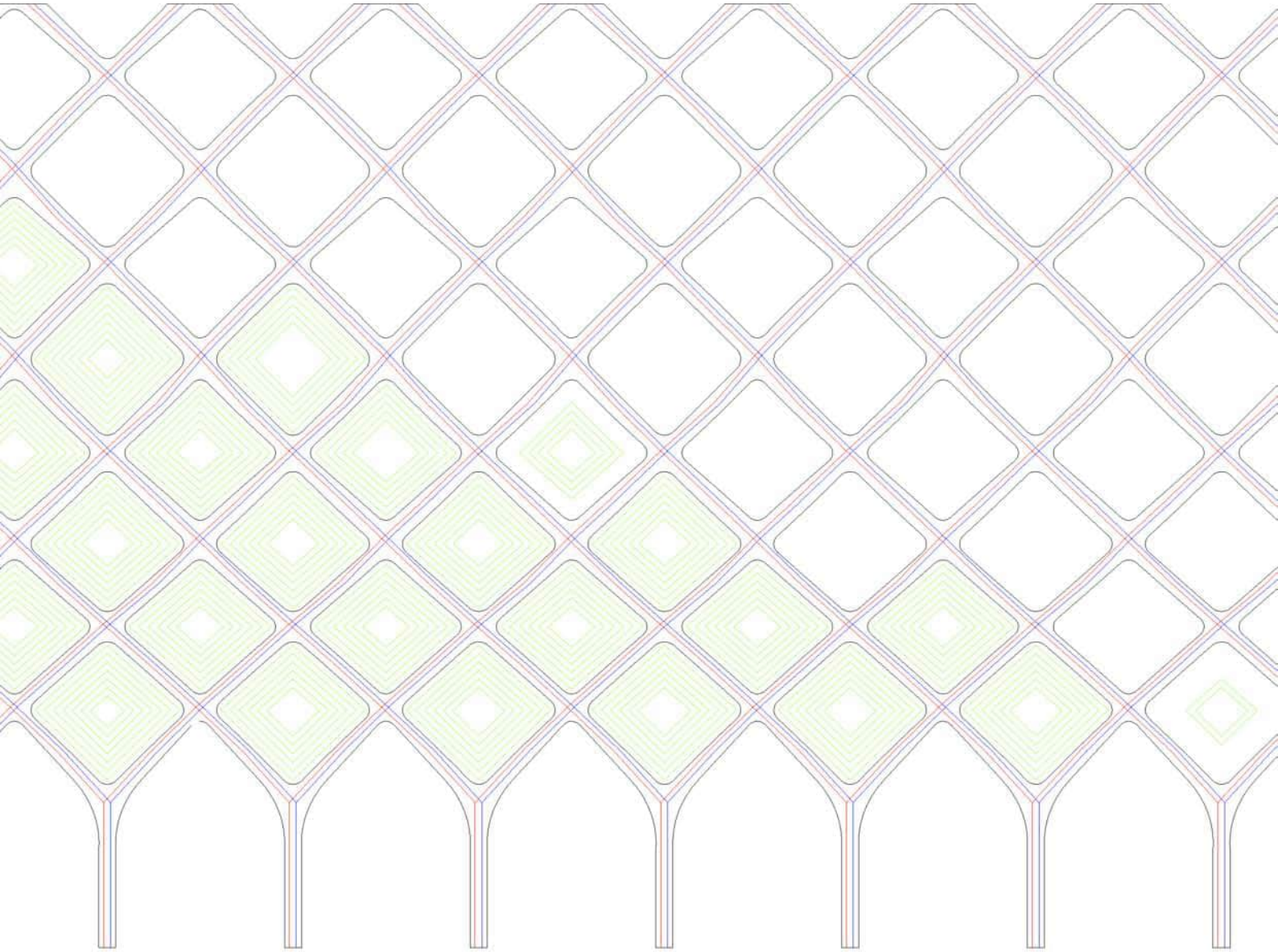


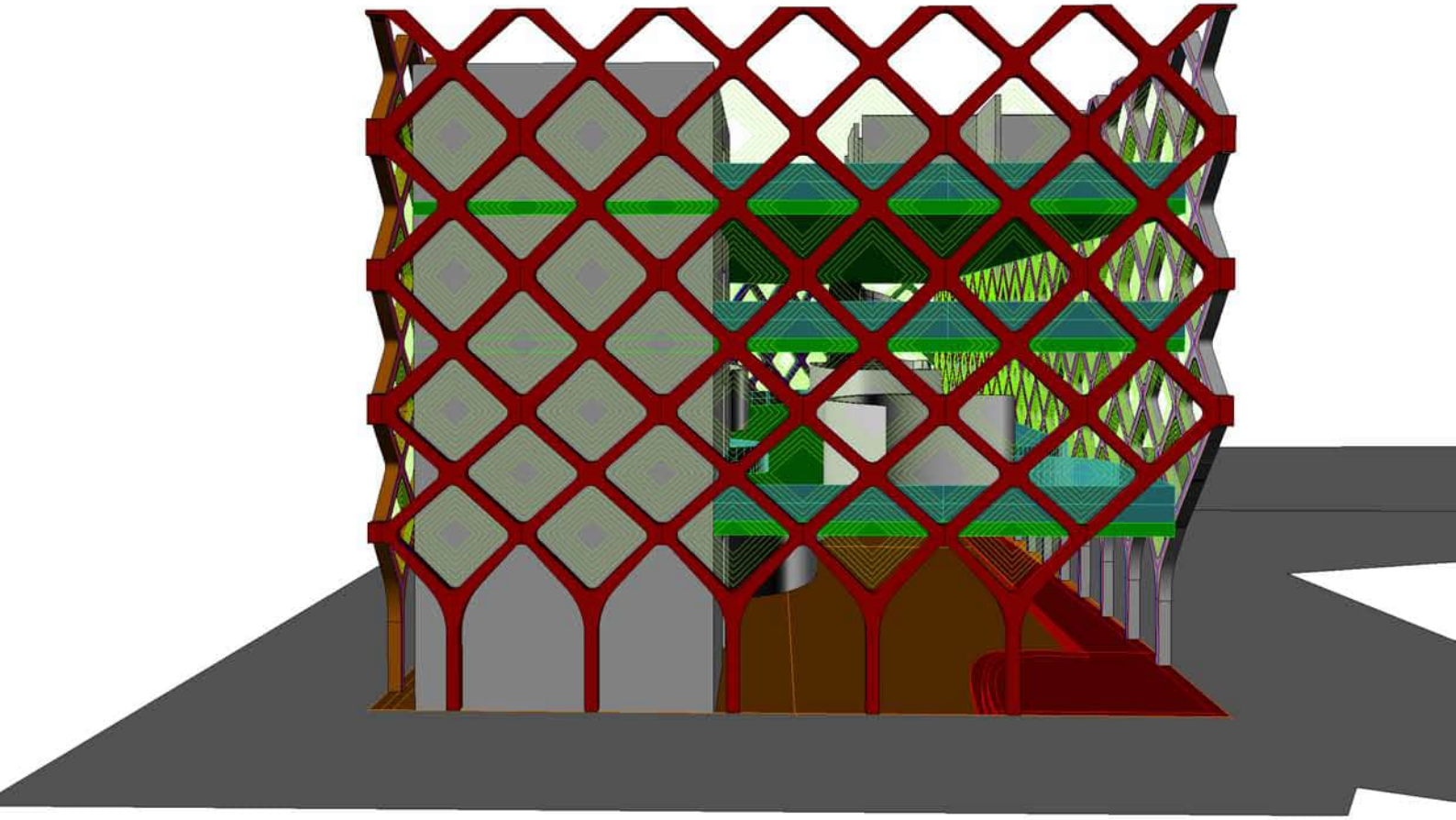


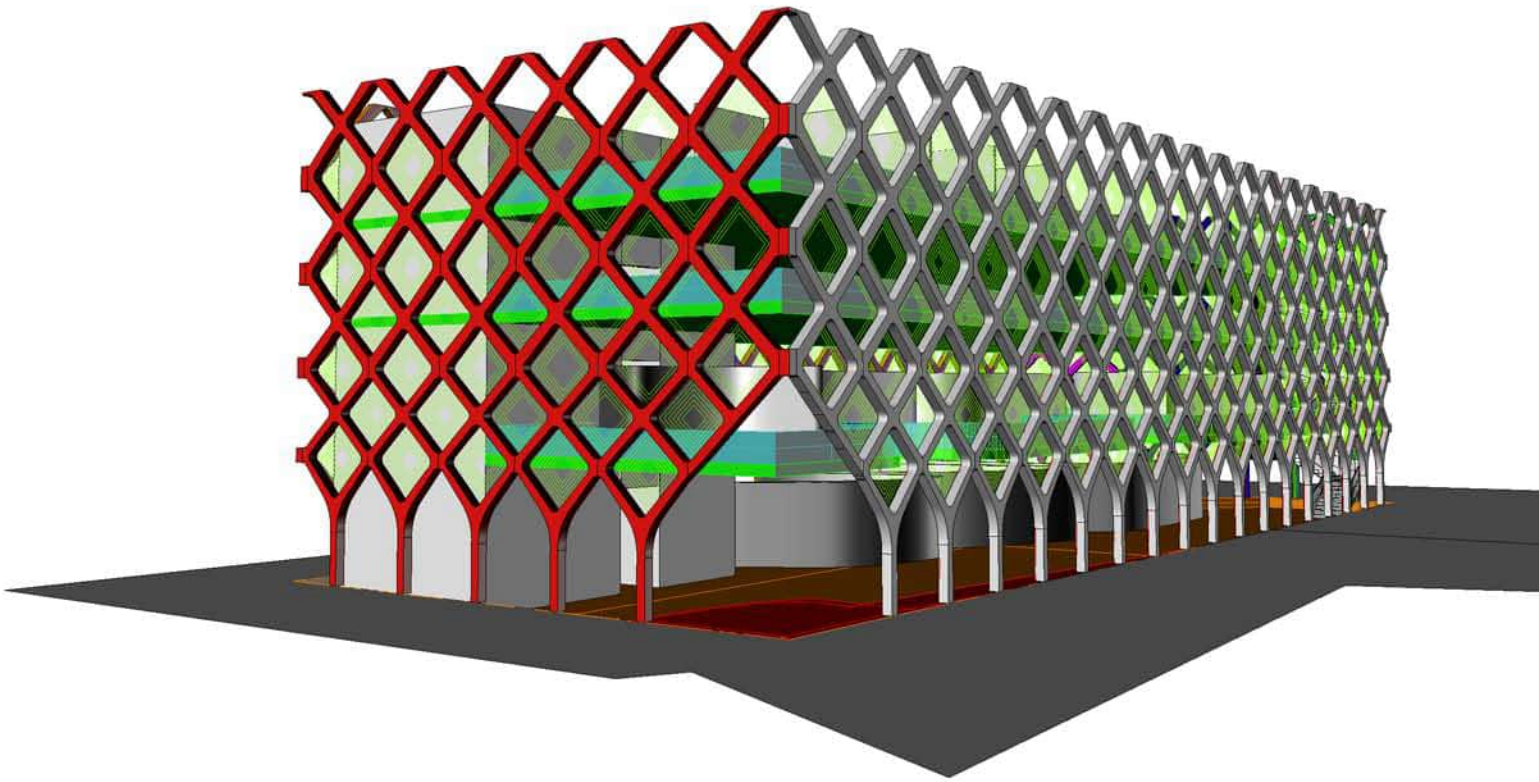


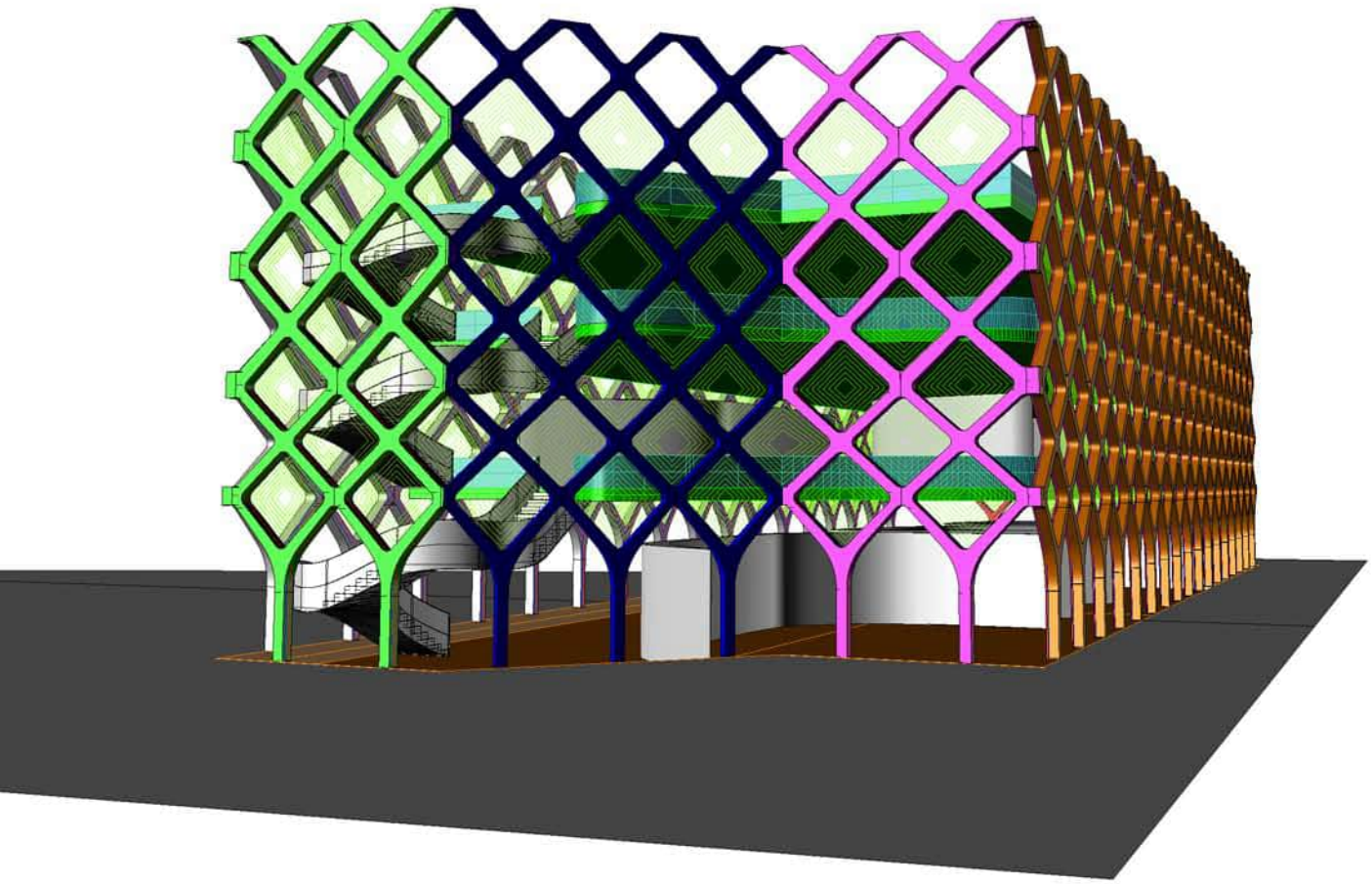


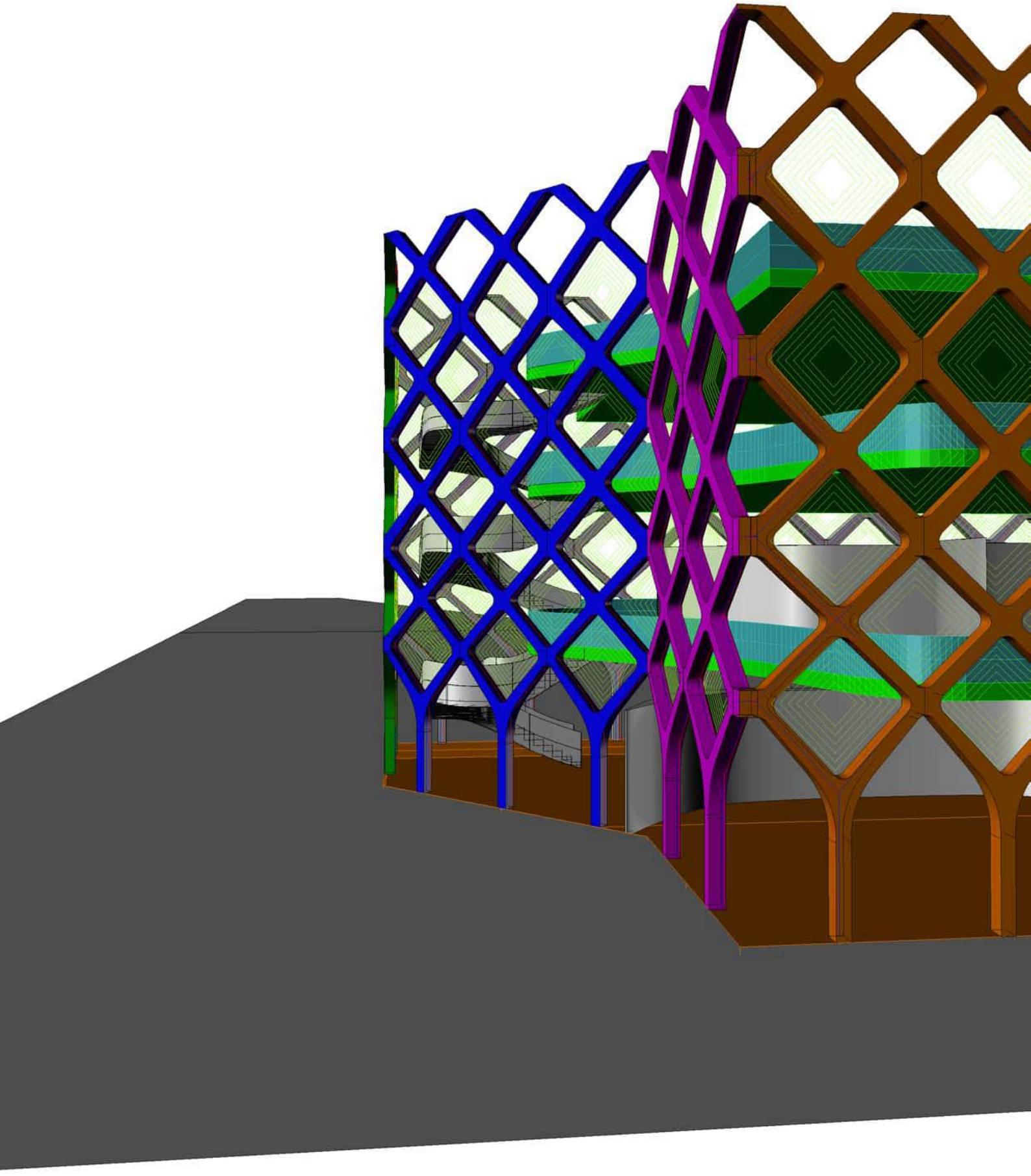


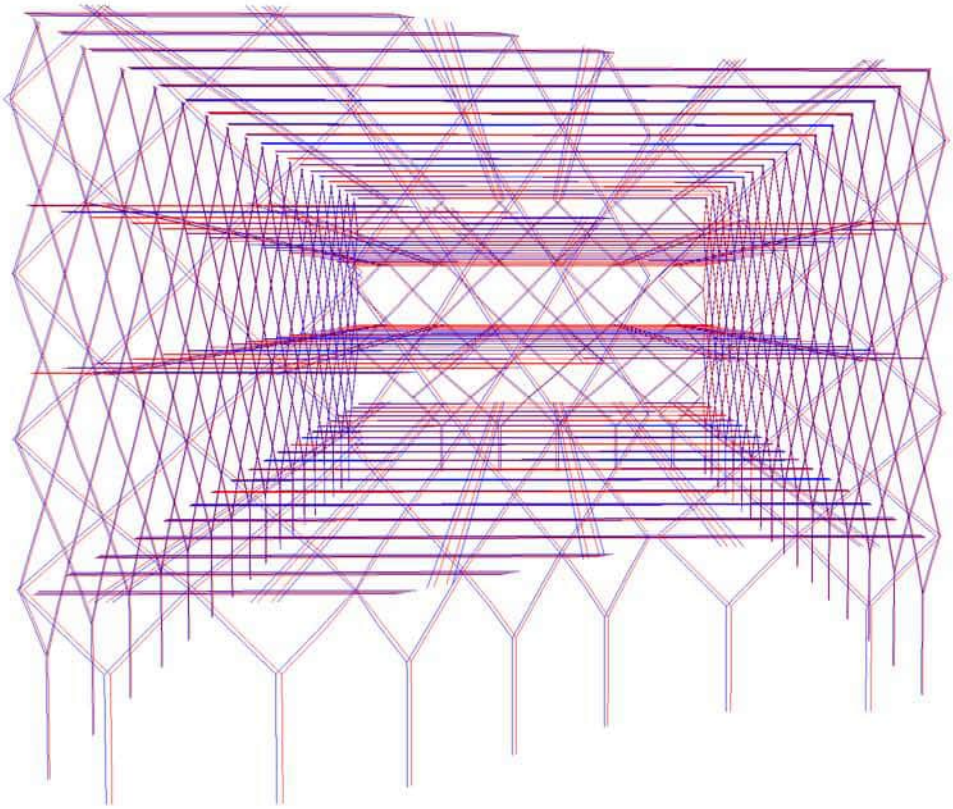


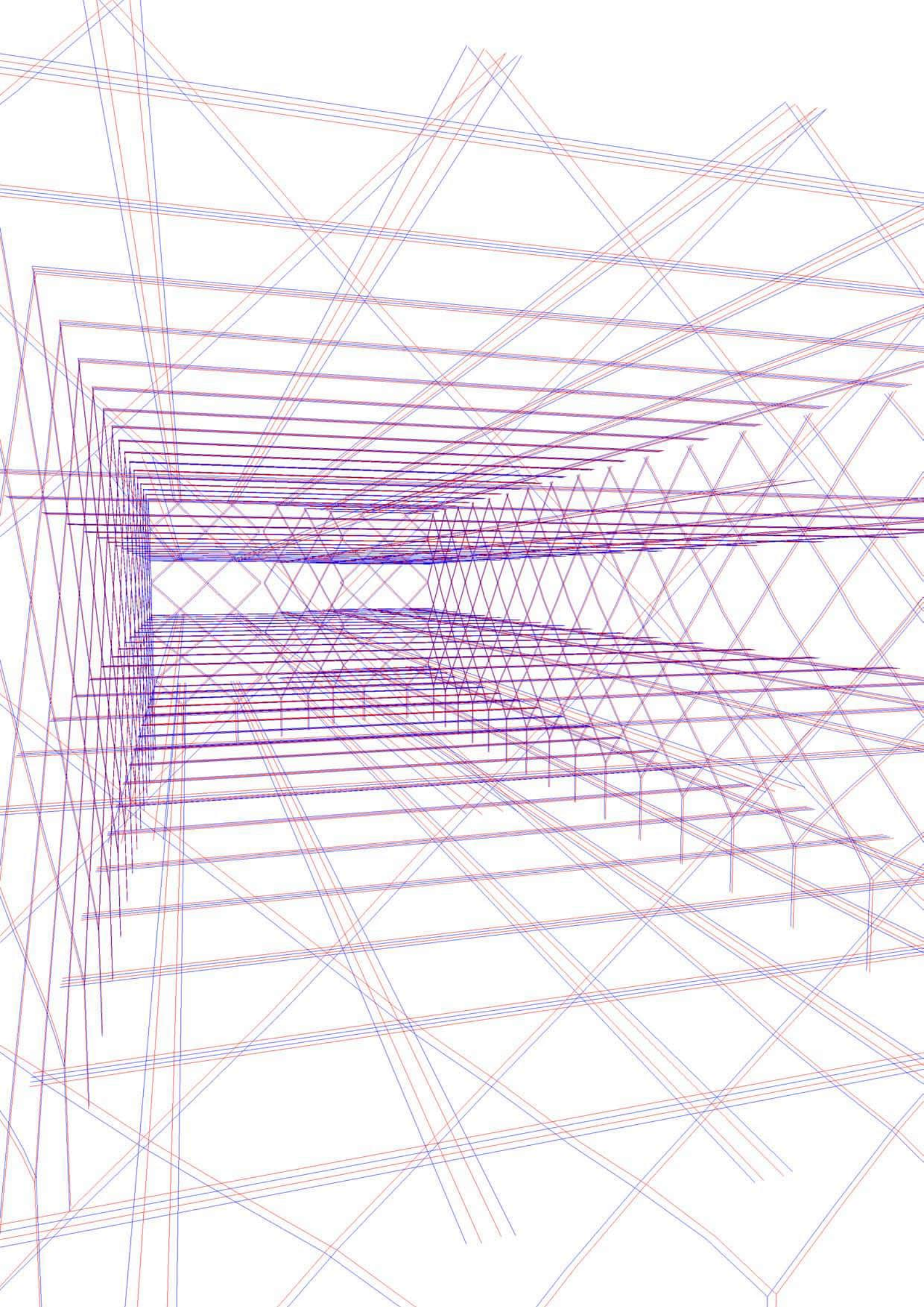


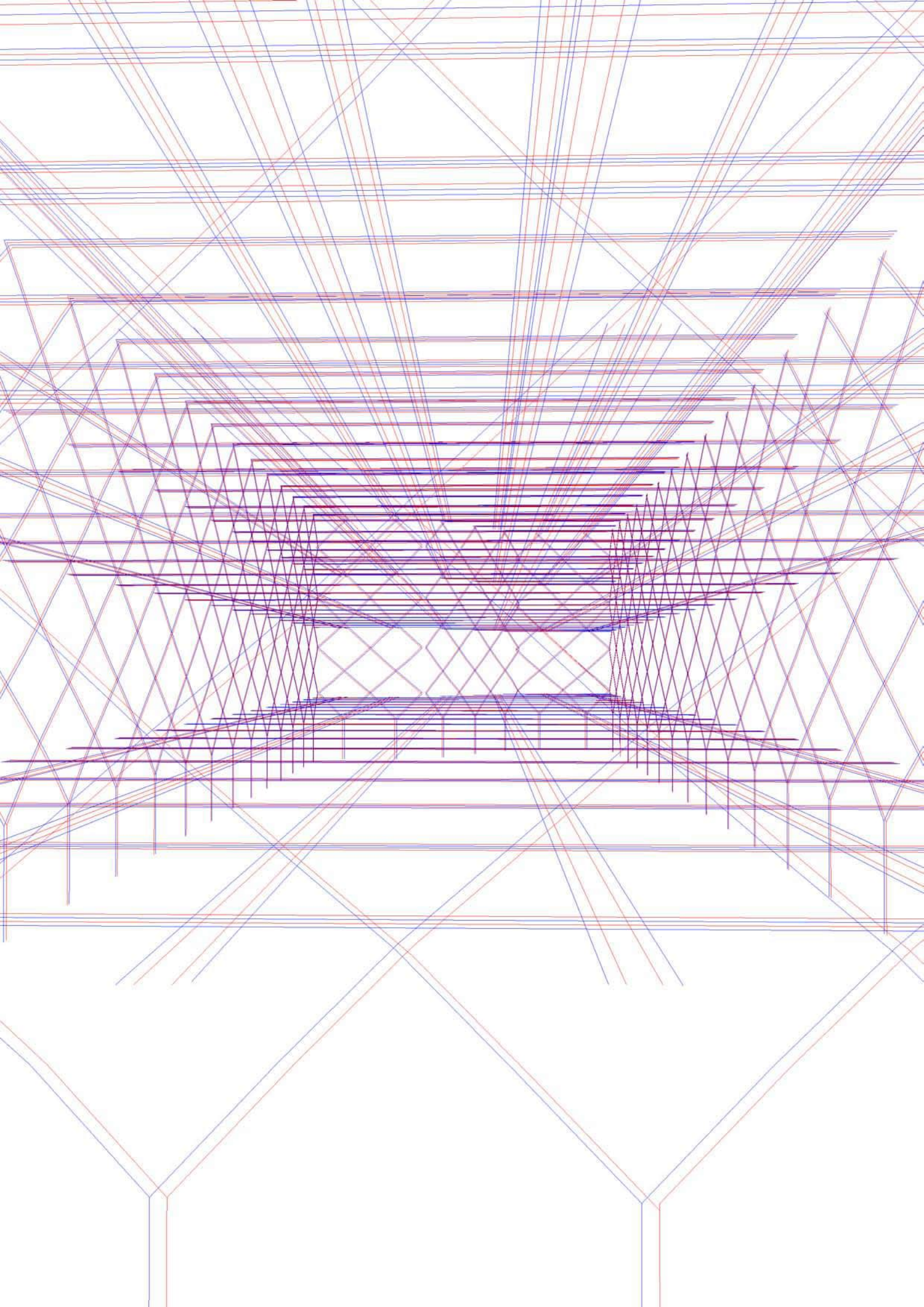


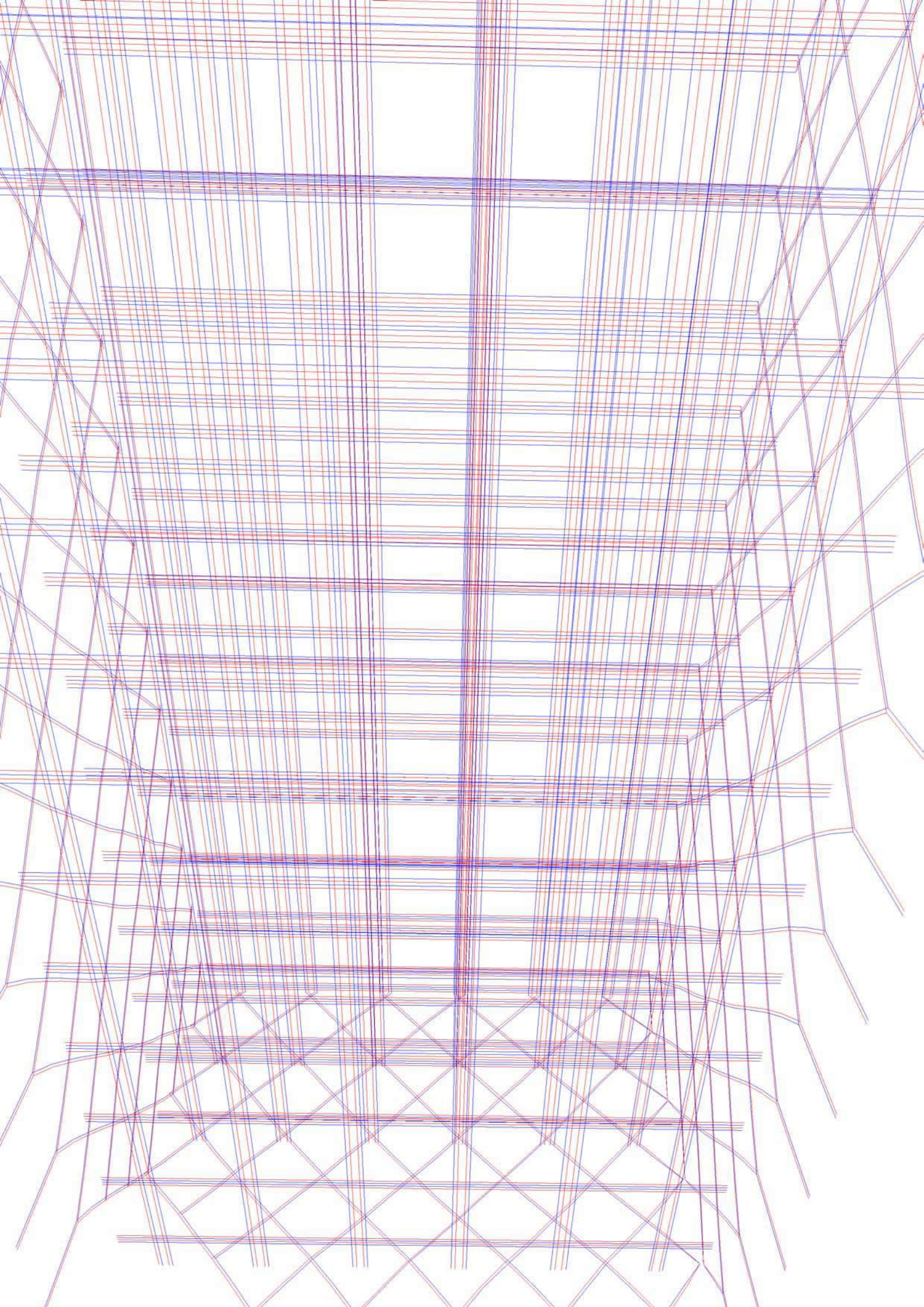












```
Option Explicit
'Script written by andrea bugli
' http://andbug.blogspot.com
'Script version venerdì 6 febbraio 2009 15.11.11
'
'I've seen an interesting surface tessellation on http://www.opensys-log.com/ and I tried to do something different

Call Main()
Sub Main()

Call Rhino.Enableredraw(False)

Dim idsrf: idsrf = Rhino.GetObject ("select surface", 8, True, True)
If IsNull(idsrf) Then Exit Sub

Dim udomain: udomain = Rhino.surfacedomain(idsrf, 0)
Dim vdomain: vdomain = Rhino.SurfaceDomain(idsrf, 1)

Dim u0: u0 = udomain(0)
Dim u1: u1 = udomain(1)
Dim v0: v0 = vdomain(0)
Dim v1: v1 = vdomain(1)

Dim A: A = Rhino.EvaluateSurface(idsrf, Array(u0,v0))
Dim B: B = Rhino.evaluatesurface(idsrf, Array(u1,v0))
Dim C: C = Rhino.evaluatesurface(idsrf, Array(u1,v1))
Dim D: D = Rhino.EvaluateSurface(idsrf, Array(u0,v1))

Call recursivetriangle(idsrf, A, B, D)
Call recursivetriangle(idsrf, B, C, D)

Call Rhino.EnableRedraw(True)

End Sub

Sub recursivetriangle(ByVal idsrf, ByVal A, ByVal B, ByVal D)

Dim distAB: distAB = Rhino.distance(A, B)
Dim distBD: distBD = Rhino.Distance(B, D)
Dim distAD: distAD = Rhino.Distance(A, D)
Dim arrdist: arrdist = Array(distAB, distBD, distAD)
arrdist = Rhino.sortnumbers(arrdist,True)

Dim H,K,J

If distAB = arrdist(2) Then
H = A
K = B
J = D
End If
If distBD = arrdist(2) Then
H = B
K = D
J = A
End If
If distAD = arrdist(2) Then
H = A
K = D
J = B
End If

Dim Z(2)
Z(0) = (H(0)+K(0))/2
Z(1) = (H(1)+K(1))/2
```

```
Z(2) = (H(2)+K(2))/2

Dim Zuv: Zuv = Rhino.surfaceclosestpoint(idsrf, Z)
Dim Zp: Zp = Rhino.evaluatesurface (idsrf, Zuv)

Dim distcurv: distcurv = Rhino.distance (Zp, Z)
Dim distang: distang = Rhino.Distance (H, K)

If (distcurv <>
Call extrudedomain (idsrf, A, B, D)
Else
Call recursivetriangle (idsrf, H, J, Zp)
Call recursivetriangle (idsrf, K, J, Zp)
End If

End Sub

Sub extrudedomain (ByVal idsrf, ByVal T, ByVal S, ByVal P)

Dim Tuv: Tuv = Rhino.surfaceclosestpoint(idsrf,T)
Dim Suv: Suv = Rhino.surfaceclosestpoint(idsrf,S)
Dim Puv: Puv = Rhino.surfaceclosestpoint(idsrf,P)

Dim nT: nT = Rhino.SurfaceNormal(idsrf, Tuv)
Dim nS: nS = Rhino.SurfaceNormal(idsrf, Suv)
Dim nP: nP = Rhino.SurfaceNormal(idsrf, Puv)

Dim E: E = Rhino.vectoradd(T, nT)
Dim F: F = Rhino.vectoradd(S, nS)
Dim G: G = Rhino.vectoradd(P, nP)

Dim idbase: idbase = Rhino.addcurve(Array(T,S,P,T), 2)
Dim idtop: idtop = Rhino.AddCurve(Array(E,F,G,E), 2)
‘changing last numbers give different types of curve

Dim centrebase(2)
centrebase (0) = (T(0)+S(0)+P(0))/3
centrebase (1) = (T(1)+S(1)+P(1))/3
centrebase (2) = (T(2)+S(2)+P(2))/3

Dim centretop(2)
centretop (0) = (E(0)+F(0)+G(0))/3
centretop (1) = (E(1)+F(1)+G(1))/3
centretop (2) = (E(2)+F(2)+G(2))/3

Dim idtopscal: idtopscal = Rhino.scaleobject(idtop, centretop, Array(0.5,0.5,0.5),True)
Dim idbasescal: idbasescal = Rhino.scaleobject(idbase, centrebase, Array(0.8,0.8,0.8),True)
‘create other two curves that are lofted to obtain a new surface

Call Rhino.AddLoftSrf(Array(idbase,idbasescal,idtopscal,idtop,idbase, , , , , ,True))
Call Rhino.deleteobject(idbase)
Call Rhino.DeleteObject(idtop)
Call Rhino.DeleteObject(idtopscal)
Call Rhino.DeleteObject(idbasescal)

End Sub
```

```
Option Explicit

Call Main()
Sub Main()
Dim idSrf : idSrf = Rhino.GetObject("Surface to recurse", 8, True, True)
If IsNull(idSrf) Then Exit Sub

Dim uDomain : uDomain = Rhino.SurfaceDomain(idSrf, 0)
Dim vDomain : vDomain = Rhino.SurfaceDomain(idSrf, 1)

Call RecurseSurface(idSrf, uDomain(0), uDomain(1), vDomain(0), vDomain(1))
End Sub

Function RecurseSurface(ByVal idSrf, ByVal u0, ByVal u1, ByVal v0, ByVal v1)
Dim um : um = u0 + 0.5 * (u1-u0)
Dim vm : vm = v0 + 0.5 * (v1-v0)

Dim A : A = Rhino.EvaluateSurface(idSrf, Array(u0, v0))
Dim B : B = Rhino.EvaluateSurface(idSrf, Array(u1, v0))
Dim C : C = Rhino.EvaluateSurface(idSrf, Array(u1, v1))
Dim D : D = Rhino.EvaluateSurface(idSrf, Array(u0, v1))
Dim E : E = Rhino.EvaluateSurface(idSrf, Array(um, vm))
Dim M(2)
m(0) = (A(0) + C(0))/2
m(1) = (A(1) + C(1))/2
m(2) = (A(2) + C(2))/2
Dim N(2)
n(0) = (D(0) + B(0))/2
n(1) = (D(1) + B(1))/2
n(2) = (D(2) + B(2))/2

If (Rhino.Distance(E, M) < 0.5) And (Rhino.Distance(E, N) < 0.5) Then
Call ExtrudeDomain (idSrf, u0, u1, v0, v1)
Else
Call RecurseSurface(idSrf, u0, um, v0, vm)
Call RecurseSurface(idSrf, um, u1, v0, vm)
Call RecurseSurface(idSrf, u0, um, vm, v1)
Call RecurseSurface(idSrf, um, u1, vm, v1)
End If
End Function

Sub ExtrudeDomain(ByVal idSrf, ByVal u0, ByVal u1, ByVal v0, ByVal v1)
Dim A : A = Rhino.EvaluateSurface(idSrf, Array(u0, v0))
Dim B : B = Rhino.EvaluateSurface(idSrf, Array(u1, v0))
Dim C : C = Rhino.EvaluateSurface(idSrf, Array(u1, v1))
Dim D : D = Rhino.EvaluateSurface(idSrf, Array(u0, v1))

Dim Na : Na = Rhino.SurfaceNormal(idSrf, Array(u0, v0))
Dim Nb : Nb = Rhino.SurfaceNormal(idSrf, Array(u1, v0))
Dim Nc : Nc = Rhino.SurfaceNormal(idSrf, Array(u1, v1))
Dim Nd : Nd = Rhino.SurfaceNormal(idSrf, Array(u0, v1))

Dim E : E = Rhino.vectoradd(A, Na)
```

```
Dim F : F = Rhino.vectoradd(B, Nb)
Dim G : G = Rhino.vectoradd(C, Nc)
Dim H : H = Rhino.vectoradd(D, Nd)

Dim idBase, idTop
idBase = Rhino.AddCurve(Array(A, B, C, D, A), 1)
idTop = Rhino.AddCurve(Array(E, F, G, H, E), 1)

Call Rhino.AddLoftSrf(Array(idBase, idTop))
Call Rhino.DeleteObject (idBase)
Call Rhino.DeleteObject (idTop)
End Sub
```

VENUS TOWER

Venus. The concept design for the master plan and city tower of the City of Incheon comes from a single moment, an inspiration, hence, a re-discovery and re-interpretation of a master piece: Botticelli's Birth of Venus. We strongly believe that contemporary architecture, today strongly attached with time and politics and socio-political issues, needs to also form a dialogue with the past. A dialogue not only understood as literal interpretation but subjective definition, a metamorphosis of form and meaning. We propose to base our design concept on the Birth of Venus, reinterpreting the act of birth, the birth of a new City that rises from a large body of water and a rich blend of cultural diversity: the cosmopolitan city of past, present and future.

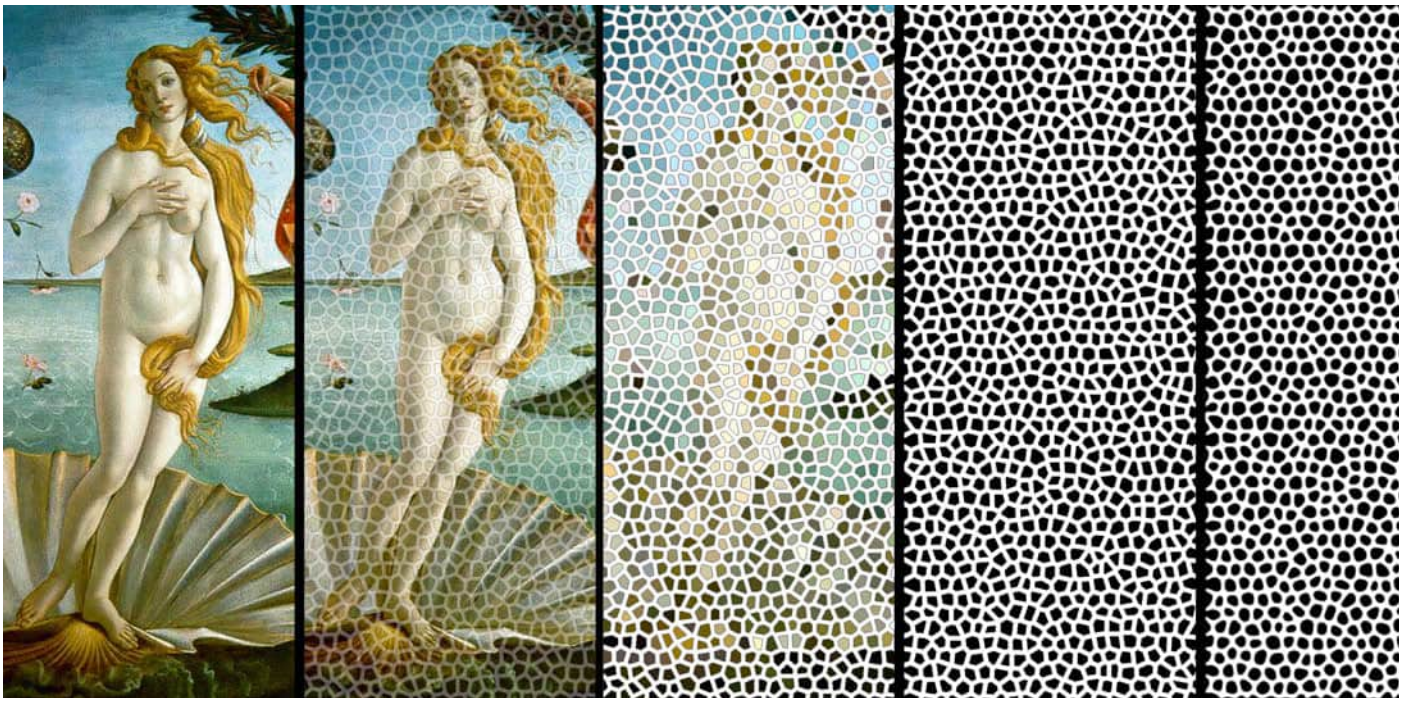
Building Geometry

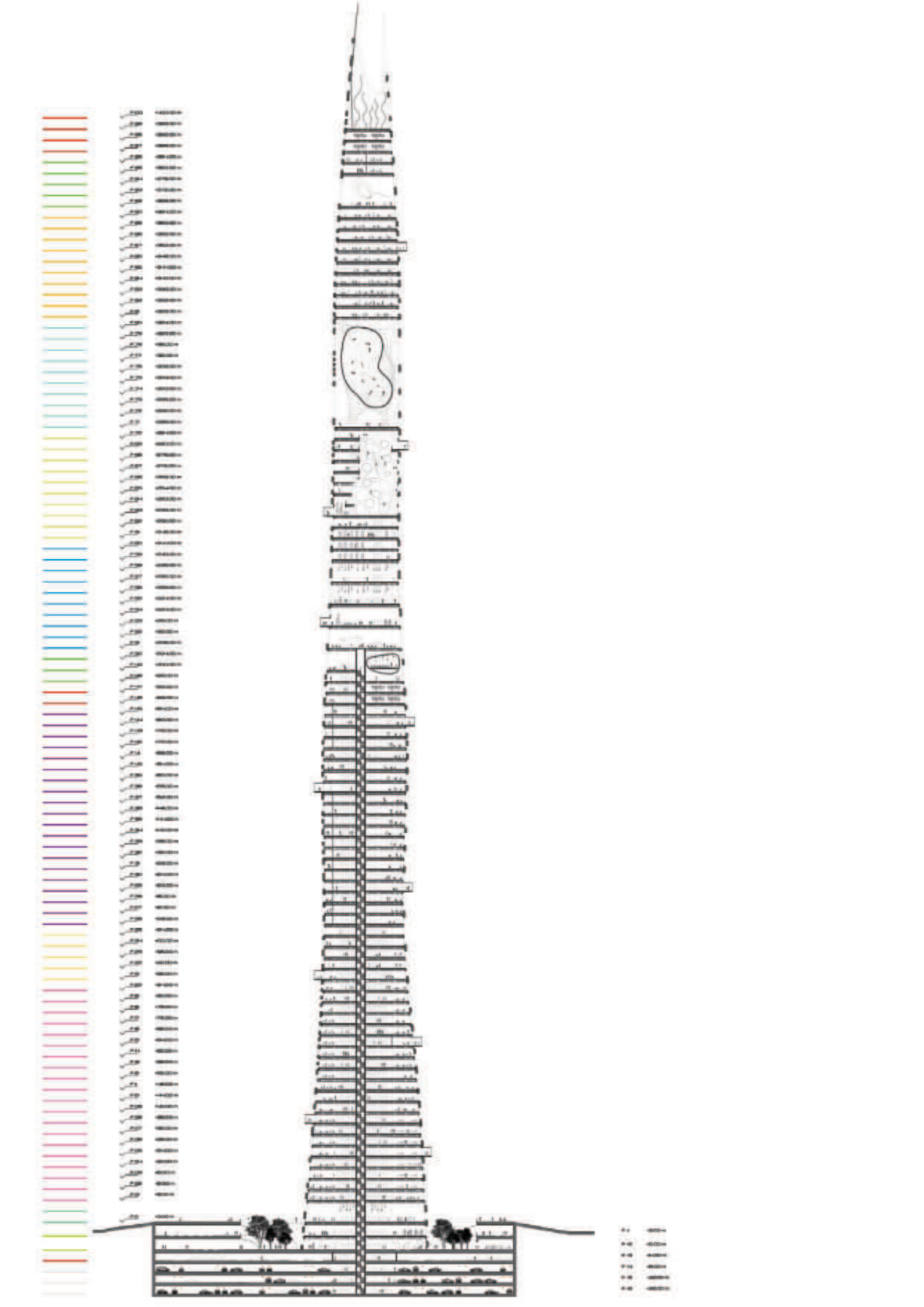
The geometric form of the City Tower is regulated by the actual operations and intersections of its systems: the lake, the different infrastructures, the land and the building, providing a flexible envelope open to continues change. The reorganization of the site topography would be responding to internal dynamics connected to programmatic imperatives and to the external interaction of the two topographies, those being the natural and the artificial. The resulting artificial landscape is characterized by a well defined distinctive skyline, a landmark which will become a pole of identity for the city of Incheon and for the Korean cultural heritage in general.

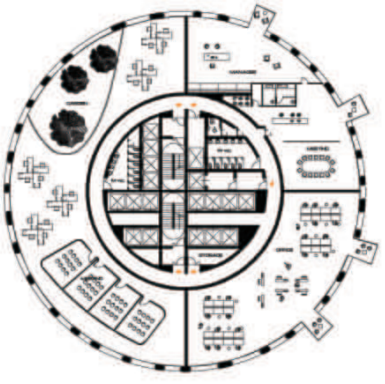
Over many years now we have developed and refined techniques to translate complex shapes into optimized building geometries. Today state of the art computer tools allow us to do this as a largely automated process. So called Parametric design tools allow us to handle huge and complex geometries in order to establish a continuous data-flow from early design stages to construction drawings and eventually building production.

CREDITS

Title: Venus Tower: a zero gravity observatory
Location: Incheon, South Korea
Client: Korea Land Corporation
Data: 120.000 sqm (tower)
Cost: € 150 M
Materials: steel, green, concrete, glass
Project Design: OFL architecture + FUERALAB
Project Team: Francesco Lipari, Stefano Rocchetti, Gabriel Belli Butler
Timing: design 2008
Web information: www.francescolipari.it

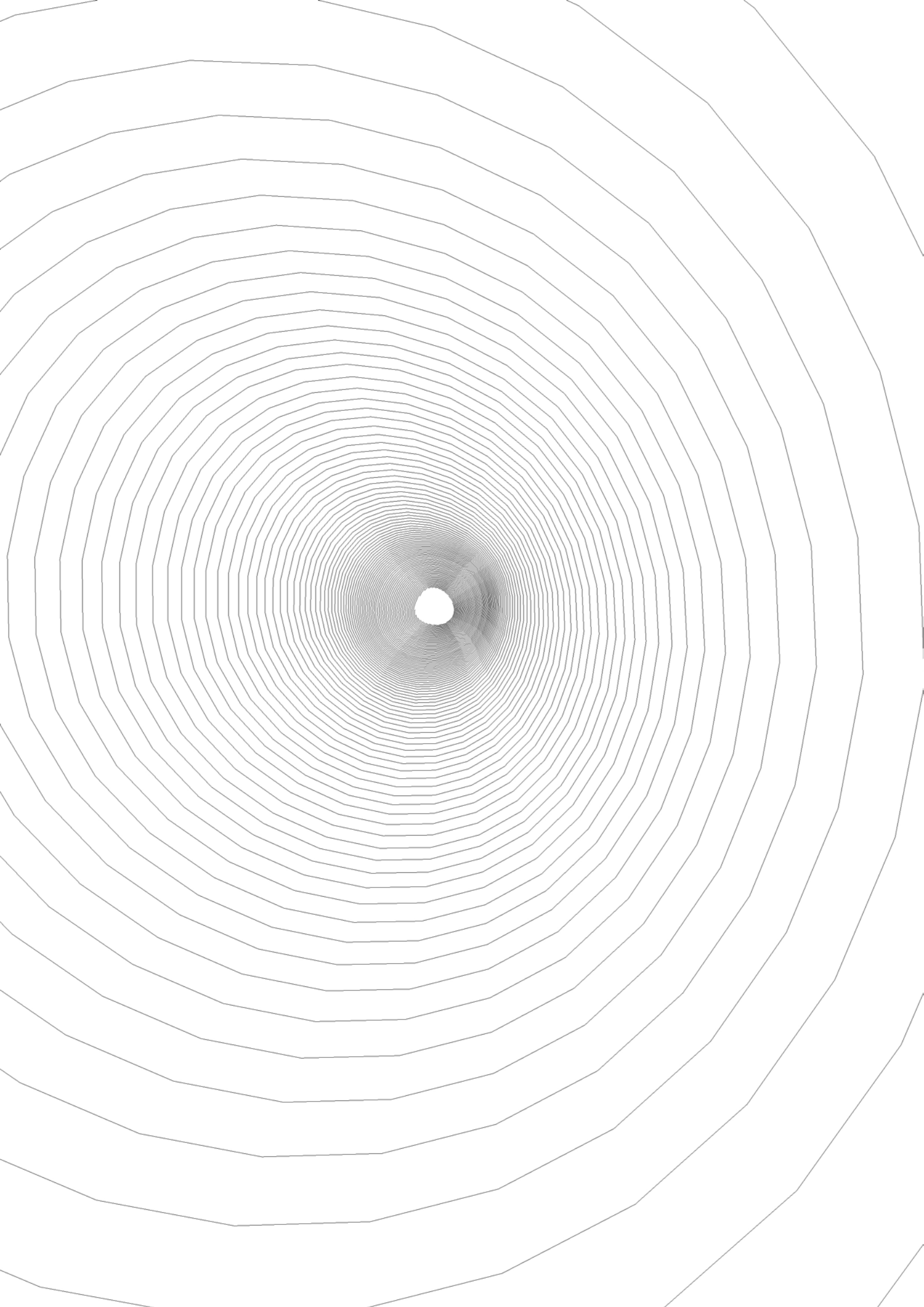







VALENTINO


Fan Coil
Controlled Natural Ventilation System
Mirrored Projection
Exhausted
Exhausted
Fresh Air
Fresh Air
Fan Coil
Controlled Natural Ventilation System
Mirrored Projection
Exhausted
Exhausted
Privé-Lite, or similar Glazing System
Ductal, or Similar (SCC)Self Compacting Concrete with Steel Fibers facade skeleton
Fresh Air
Fresh Air
Observatory Deck
Exhausted
Exhausted



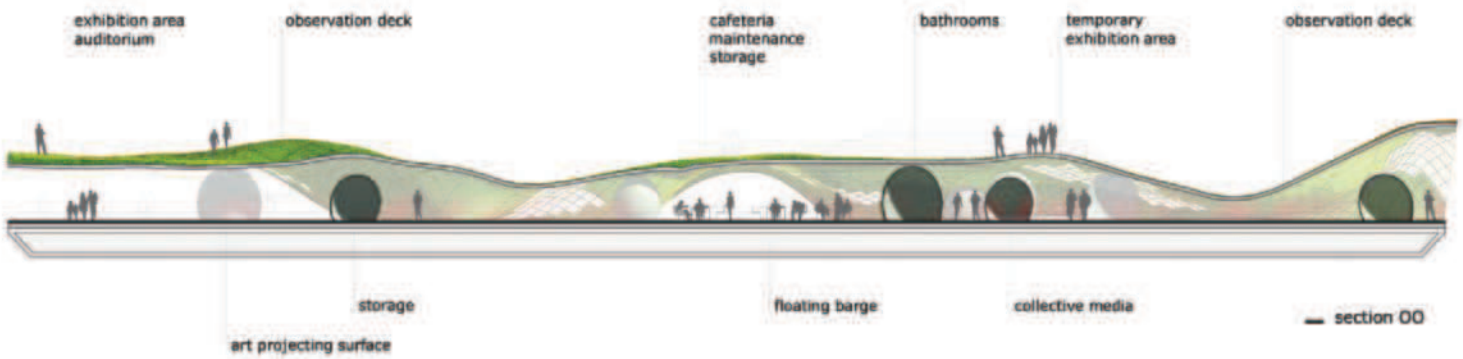
```
Option Explicit
‘ script written by roland snooks | 2007 | www.kokkugia.com

Call attractorToolsScale

Sub attractorToolsScale

Dim i, j, arrObjects, arrAttract, thresholdDist, scaleFactor, arrBBox, arrCntrPt, counter, arrAttPt
Dim dblAttDistTest, dblClosestAttDist, dblClosestAtt, adjAmount,dblParam

‘ input
arrObjects = Rhino.GetObjects(“select objects”)
arrAttract = Rhino.GetObjects(“select line attractors”, 0)
thresholdDist = Rhino.GetReal(“distance threshold”, 10, 0)
scaleFactor = Rhino.GetReal(“scale factor”, 1, 0)

‘ loop through each object and get the closest attractor
For i = 0 To UBound(arrObjects)

‘ get object bounding box centerpoint
arrBBox = Rhino.BoundingBox(arrObjects(i))
arrCntrPt = Array((((arrBBox(2)(0)) + (arrBBox(0)(0))) / 2), (((arrBBox(2)(1)) + (arrBBox(0)(1))) / 2),
(((arrBBox(0)(2)) + (arrBBox(4)(2))) / 2))

‘ loop through attractors to find the distance to the closest
counter = 0
For j = 0 To UBound(arrAttract)

dblParam = Rhino.CurveClosestPoint(arrAttract(j), arrCntrPt)
arrAttPt = Rhino.EvaluateCurve(arrAttract(j), dblParam)

‘ get distance
dblAttDistTest = Rhino.Distance(arrAttPt, arrCntrPt)

‘ is it closer
If counter < 1 Then
dblClosestAttDist = dblAttDistTest
dblClosestAtt = counter
Else
If dblAttDistTest < thresholdDist Then
If dblAttDistTest < dblClosestAttDist Then
dblClosestAttDist = dblAttDistTest
dblClosestAtt = counter
End If
End If
End If
counter = counter + 1

Next

‘ if the object is within the threshold then operate on it
If dblClosestAttDist < thresholdDist Then
```

```
‘ caculate the adjustment amount
‘ decrease infinite amount
adjAmount = 1 - ((thresholdDist - dblClosestAttDist)/thresholdDist)
‘ increase by factor of up to 100% x scaleFactor
‘adjAmount = ((thresholdDist - dblClosestAttDist)/thresholdDist) * scaleFactor + 1

‘ based on attractor proximity do something to the object (eg scale)

‘ scale
Rhino.ScaleObject arrObjects(i), arrCntrPt, Array(adjAmount,adjAmount,adjAmount)
at_ChangeColor arrObjects(i),thresholdDist,dblClosestAttDist,scaleFactor
End If

Next

End Sub

Function at_ChangeColor(obj,tDist,aDist,sFactor)
Dim objColor, newColor
‘ caculate color
newColor = 255 - ((1 - ((tDist - aDist)/tDist)) * 255)
‘ change color
objColor = Rhino.ObjectColor(obj, newColor)

End Function
```

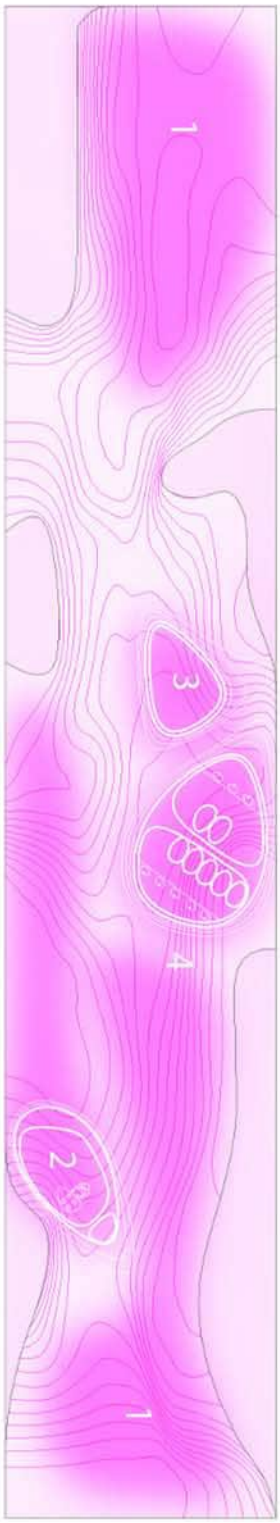
LONDON FLOATING GALLERY

The concept design for the floating gallery on the River Thames comes from a single moment, an inspiration, hence, a re-discovery and re-interpretation of a landscape. We strongly believe that contemporary architecture, clearly influenced by time and politics and socio-political issues, needs to also maintain a direct dialogue with nature. A dialogue not only understood as literal interpretation but subjective definition, a metamorphosis of form and meaning. We propose to base our design concept on a topography, a cut, a landscape; reinterpreting the idea of fluidity, whether in a liquid stage or a fixed 'frozen' surface. Water is our inspiration. Water is life, breath, power, and above all things, beauty. The floating gallery takes all these elements, proposing a fluid exhibition space for recreational uses and contemplation; a city landmark... a landscape. Landscape Across an intensive study of the vegetation and its colouring, we have based our exterior exhibition space under the large possibility of varying the colour four times a year. Four different typologies of flowers, actually, will colour the entire landscape cyclically, every four months, hence revealing the current season: red tulips during spring, yellow freesia during summer, blue iris during the fall, and orange narcissus during winter. The VESSEL envelope has been first conceived as a sculptural artifact. To translate the free (fluid) form into a defined geometry that can be build with today's construction methods and within a reasonable budget is one of the key challenges for a unique project like this one. The building envelope embraces the floating gallery vessel like an organic woven piece of cloth with variable roof heights. There is no differentiation between façade and structure. The whole building envelope has been merged into one continuous sculptural element composed of two different systems: an external glass and vegetation plane and an internal organic wood truss mesh. This double skin envelop has been carefully crafted to give the gallery an almost fluid appearance of soft transition from concave and convex, from transparent to opaque. The two façade systems converge and reject one another in a subtle way, offering ever changing perspectives for the beholder both inside and outside the vessel. The structure that holds together these two façade layers functions as a homogeneous mesh of wood trusses and glass panes, giving structural unity and coherence throughout the concept design. Direct Light The exterior glass facade and interior wood truss structure are arranged in a way to prevent direct sun-light penetrating the exhibition spaces. Furthermore, the outer skin glazing has a relative high reflectivity. The outer glazing design has been conceived as random openings placed strategically among the green vegetation, allowing for a certain amount of light inside the various spaces of the vessel, such as the cafeteria, common rest areas, and observation deck.

CREDITS
Title: London floating exhibition gallery
Location: London - UK
Organizer: AA
Data: 1.000 mq
Cost: € 2 M
Materials: steel, green, concrete, glass
Project Design: Francesco Lipari, Stefano Rocchetti, Gabriel Belli Butler (FUERALAB)
Project Team: Francesco Lipari, Stefano Rocchetti, Gabriel Belli Butler (FUERALAB)
Timing: design 2008
Web information: www.francescolipari.it



OUTSIDE LANDSCAPE



INSIDE LANDSCAPE

1. EXHIBITION GALLERY-AUDITOIUM

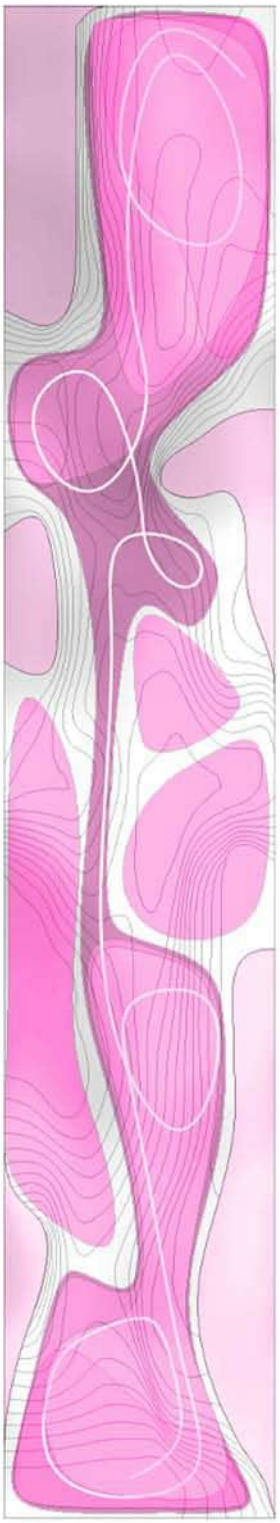
2. CAFE'

3. STORAGE

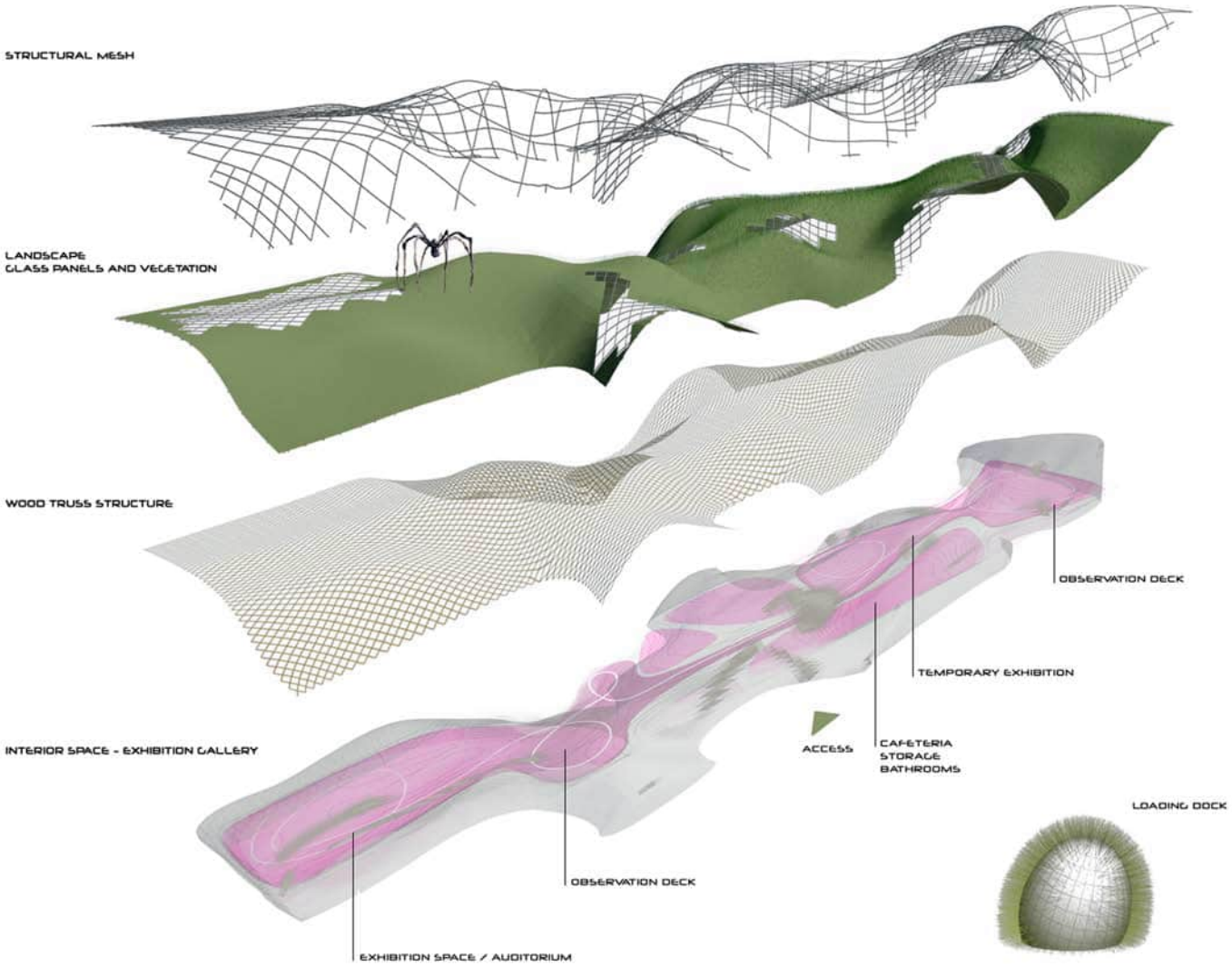
3. TOILETTES

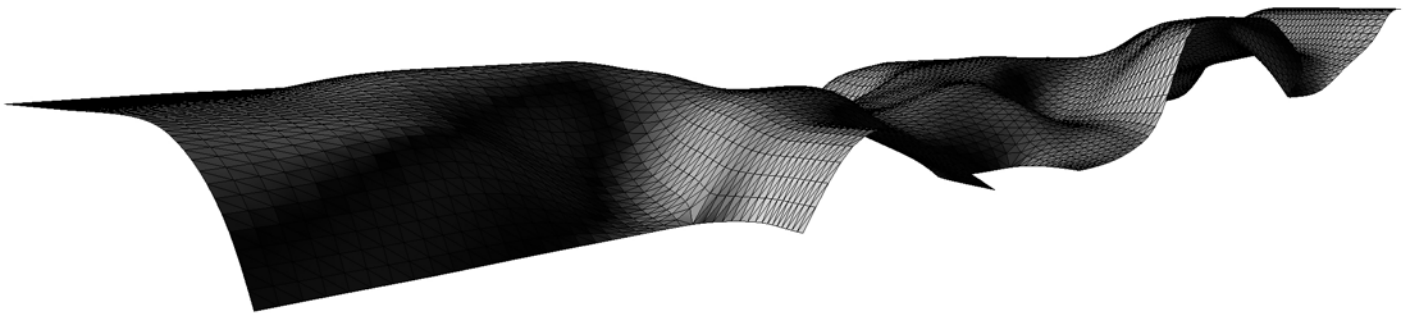
FLOATING BARGE PLAN

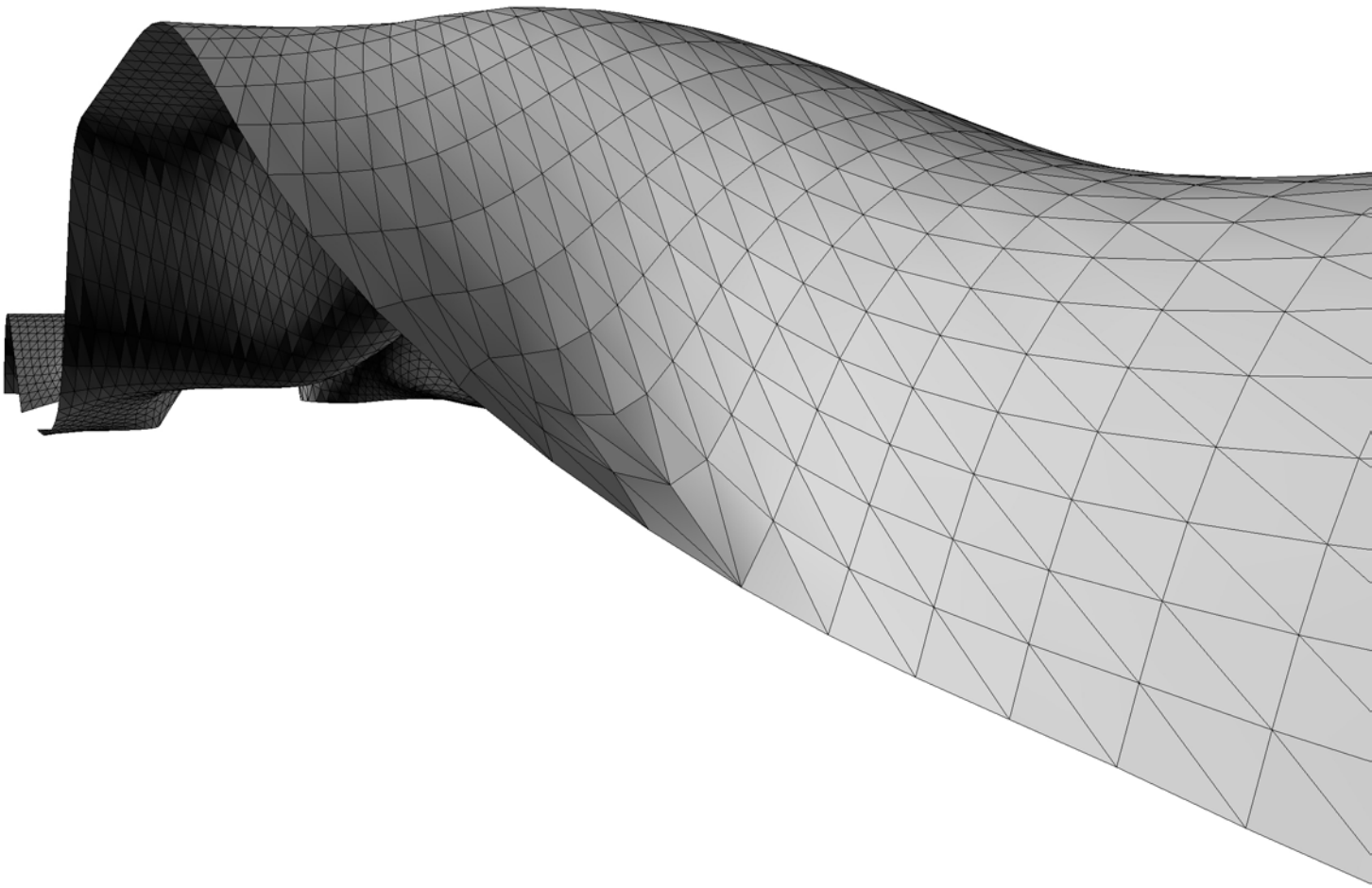
EXHIBITION SPACE

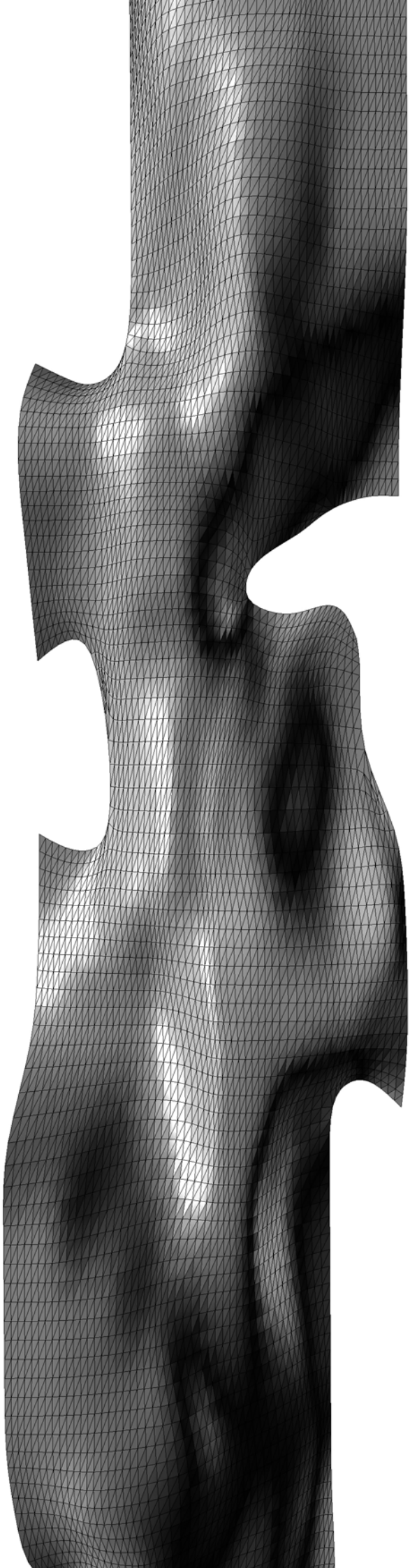


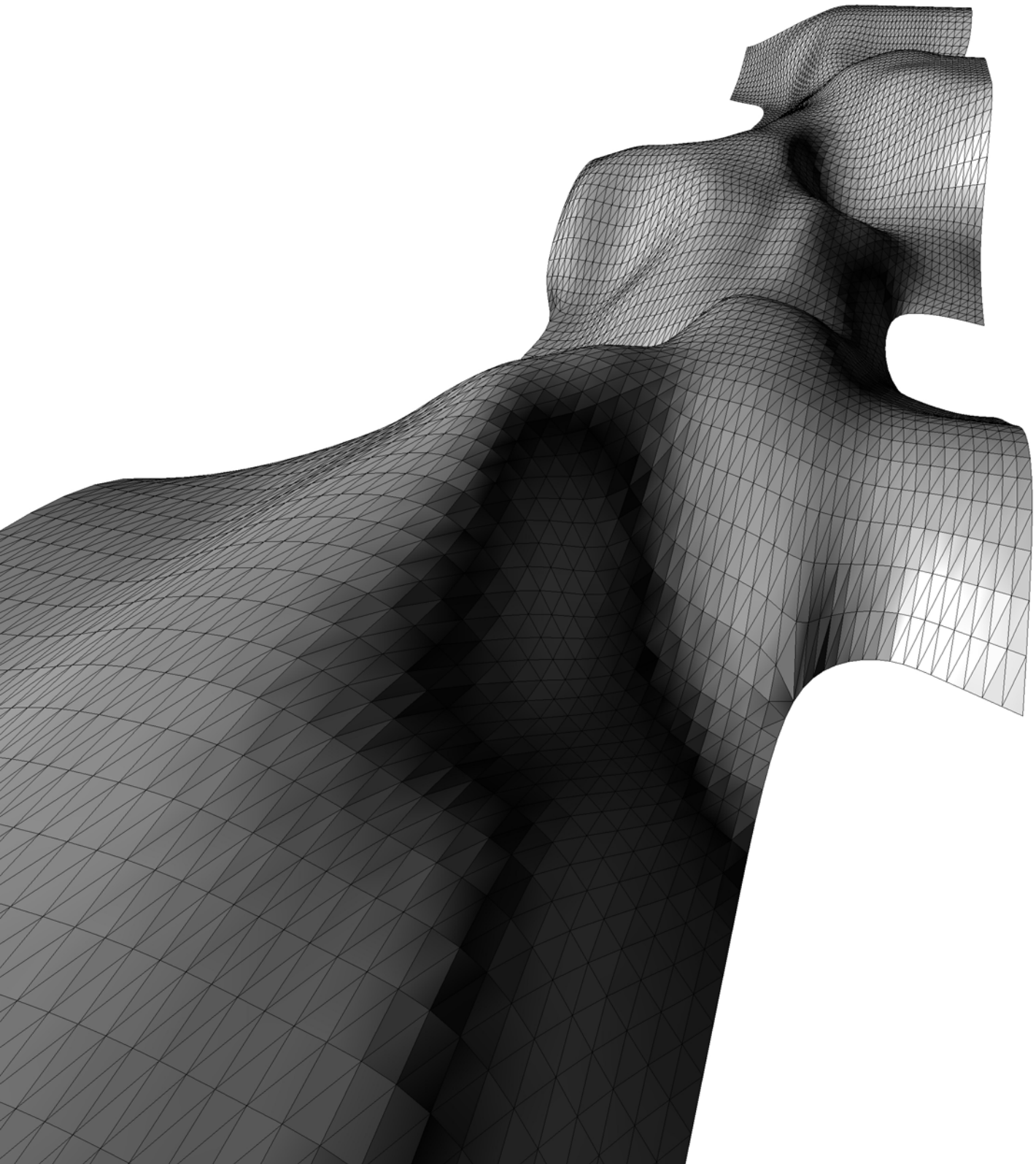
SPACES AND MOVEMENTS

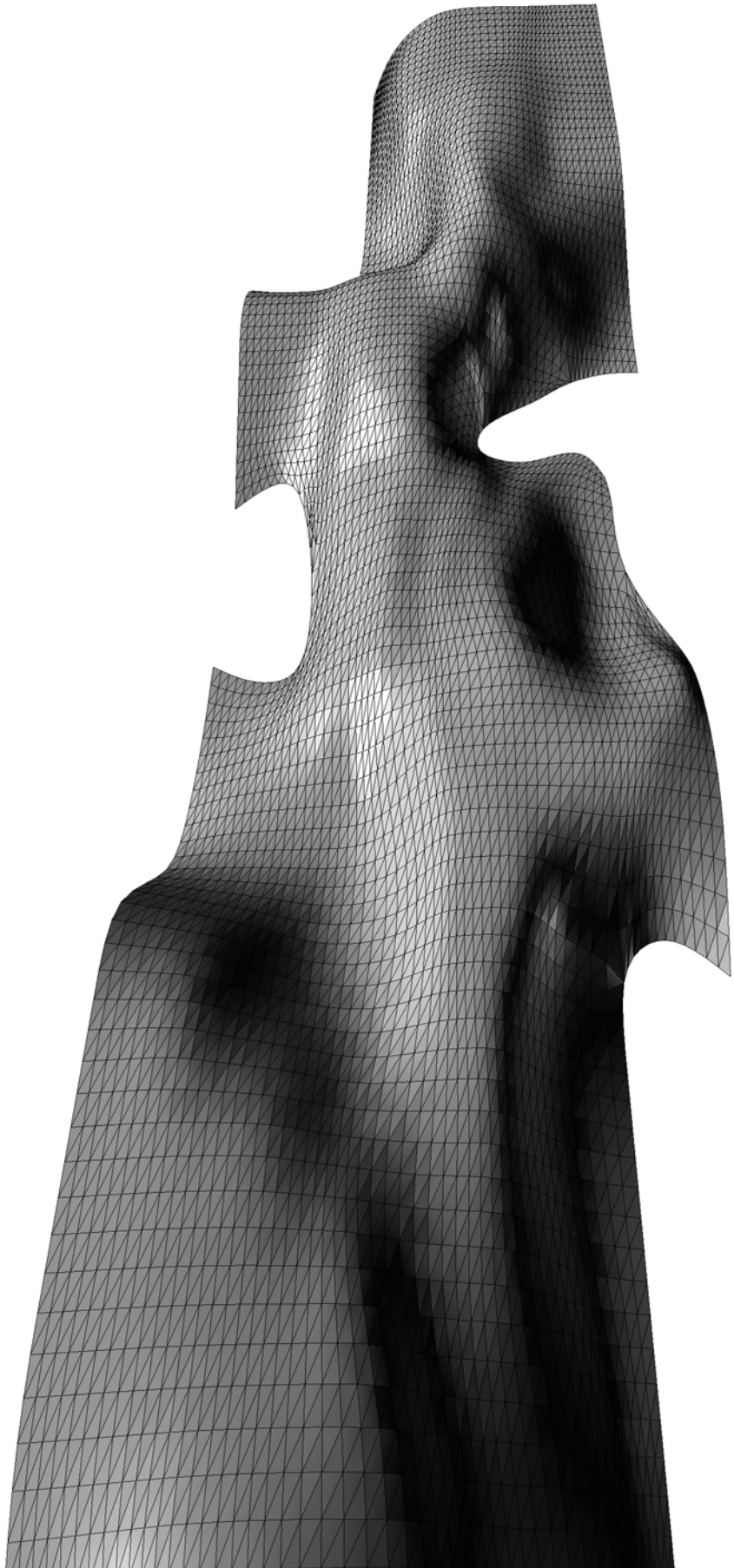

STRUCTURAL MESH
LANDSCAPE
GLASS PANELS AND VEGETATION
WOOD TRUSS STRUCTURE
OBSERVATION DECK
TEMPORARY EXHIBITION
INTERIOR SPACE - EXHIBITION GALLERY
ACCESS
CAFETERIA
STORAGE
BATHROOMS
LOADING DOCK
OBSERVATION DECK
EXHIBITION SPACE / AUDITORIUM

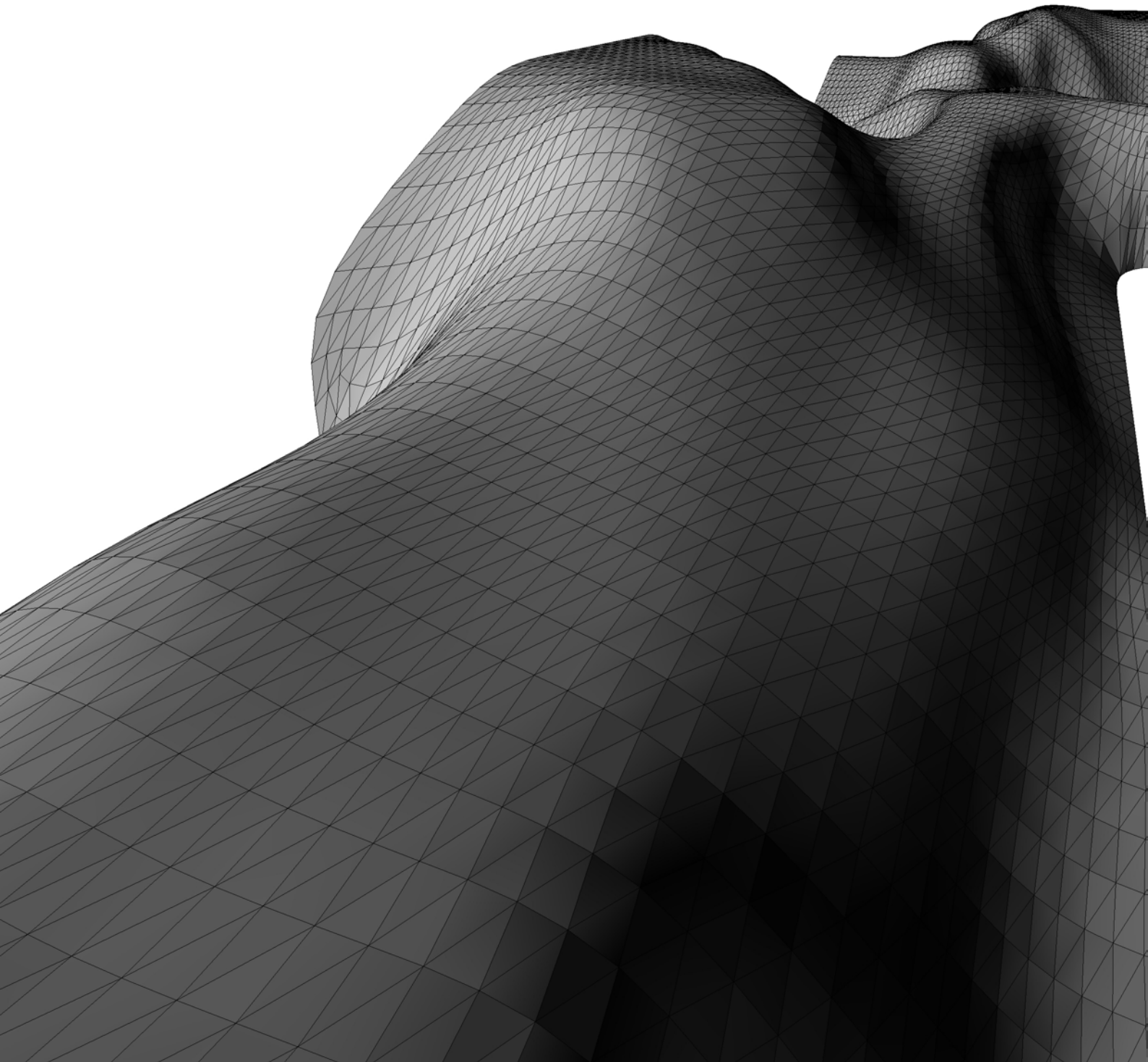










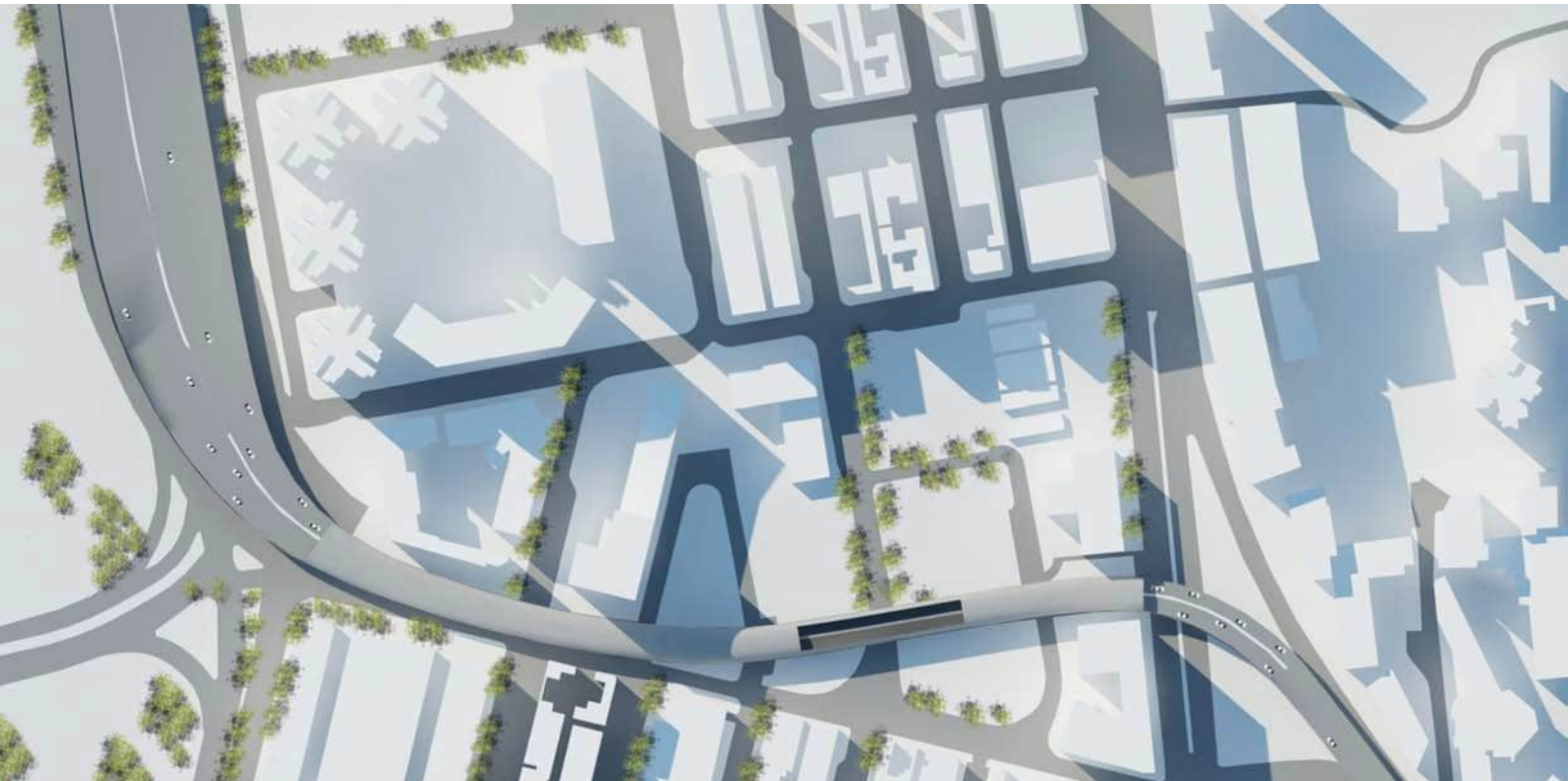


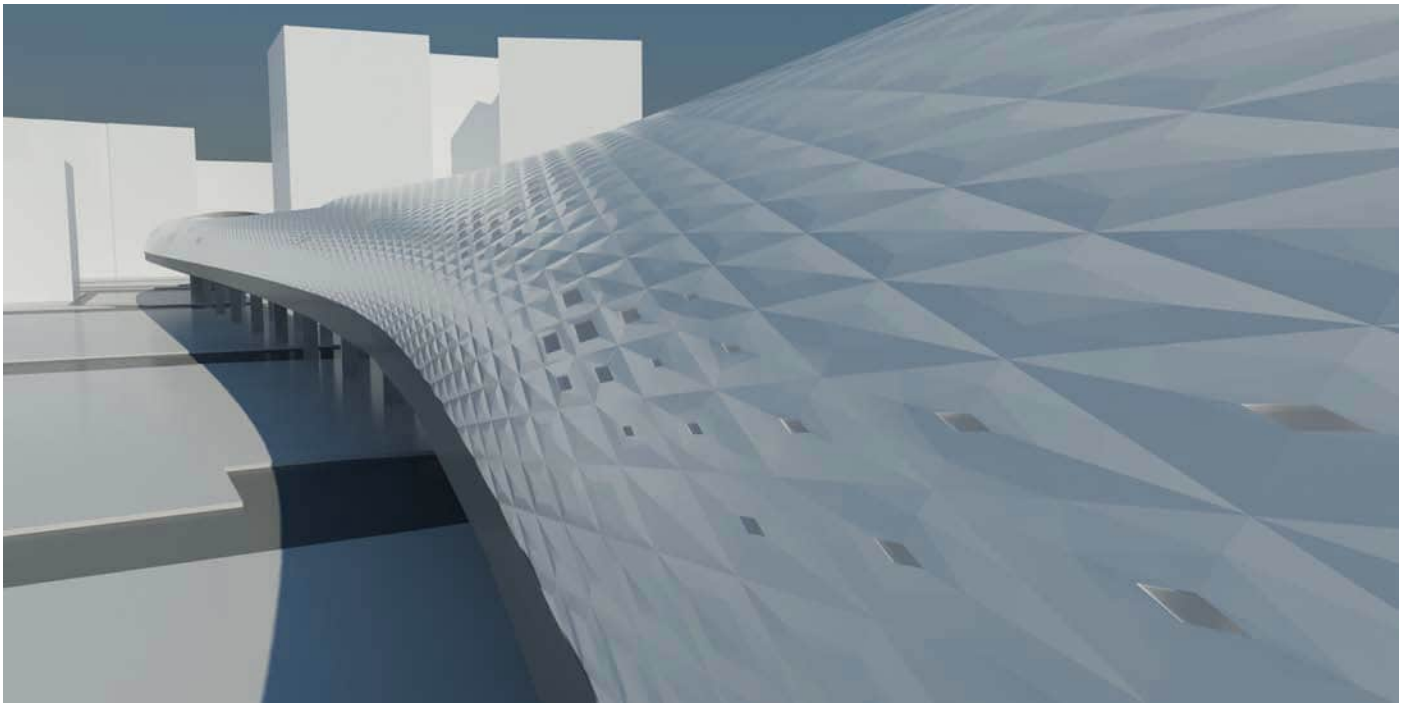
HONG KONG NOISE BARRIER

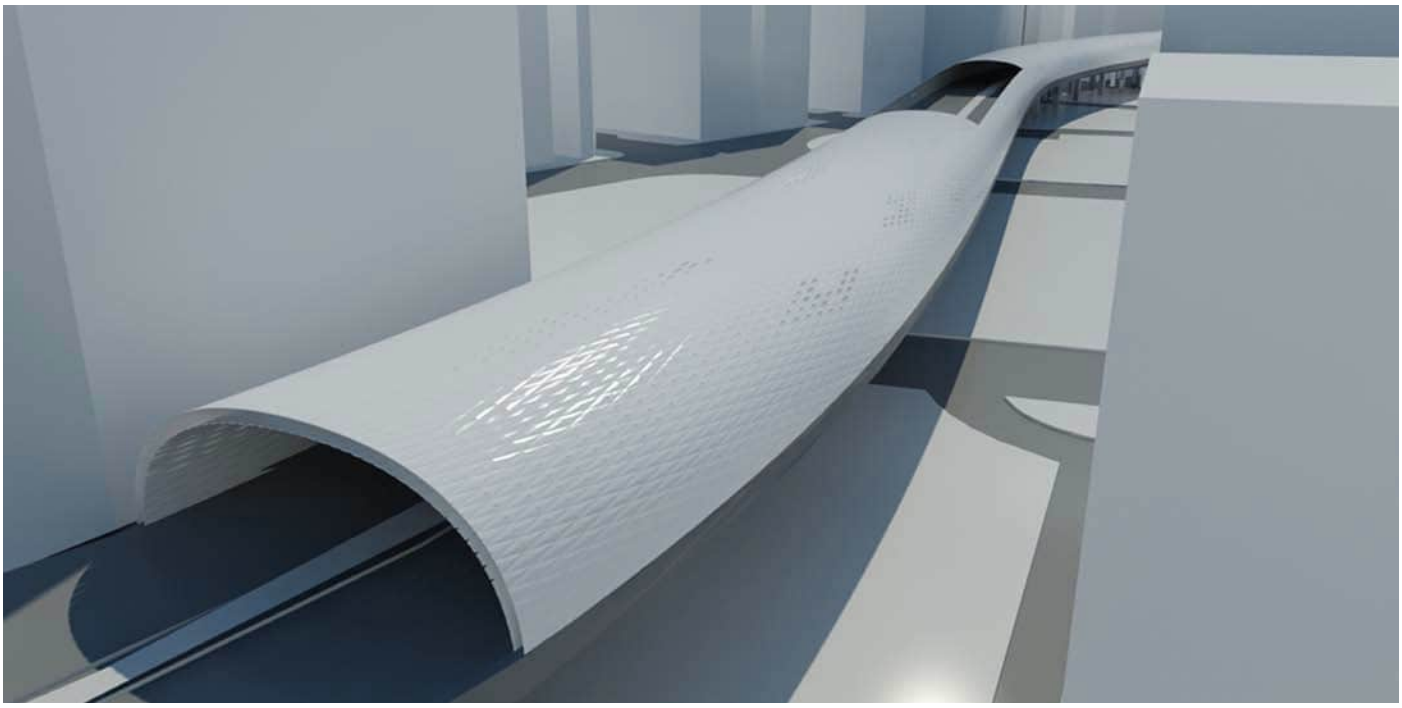
The noise barrier project consists of 4 pieces located in the city of Hong Kong along the streets of Tai Po Tai Wo Road e Gascoigne road.
A cantilevered barrier of 170 metres long and 6 metres height, a vertical barrier of 140 metres long and a variable wide from 23 to 20 metres, a vertical barrier 70 metres long and 4,5 metres height and a semi-full-cantilivered noise barrier of 500 metres long and 6 metres height. The space frame structure of the noise barrier has been constructed from hot-dip galvanized 50 mm tubes for the main structure and 30 mm tubes for the secondary structure. The entire structure is covered with translucent polycarbonate panels, with a different range of apertures, each one with a dimensions of 3 m x 0.65m. The panels change color with the daylight and react to the impulse of the shaft located at the intersection of every panel. During the day, the new noise barriers forms an harmonious unity with the surrounding landscape. They also integrate an innovative green system. The design is developed incorporating the natural growth characteristics of the grass. What makes the design unique is that the construction itself, as a sound barrier is a living, green unit. Light and shadow on the white panels establish a suggestive relationship with the surrounding landscape. After sunset, the artificial light illuminates the architectural body from inside, turning the shapes into a illuminated sculptures.

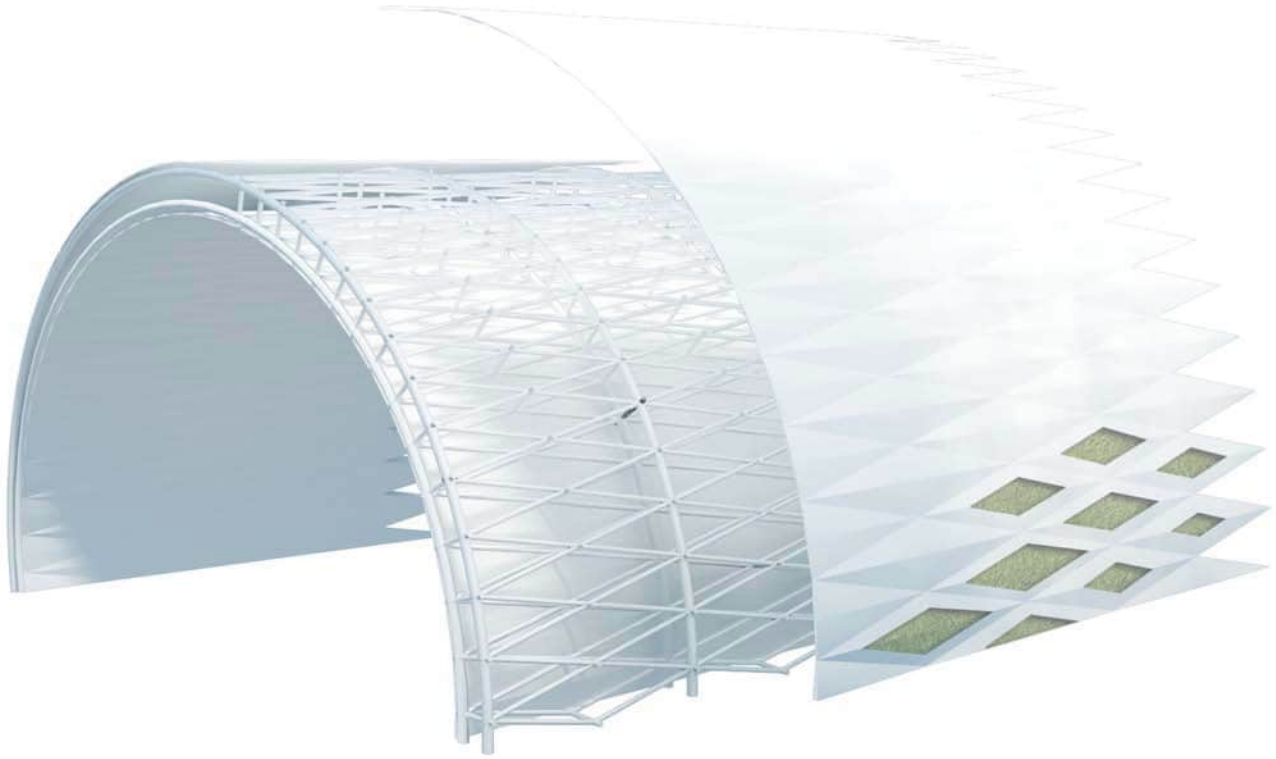
HIGH TECH BLADE RICCIO ©
The focal point of the whole system is the blade named RICCIO © It's an innovative technological system that transform the air movement in energy and then, thanks to a generator, in electricity. The blade dimensions are: a = 1 cm; b = 1 cm; l = 40 cm; Weight = 150gr
During the vehicle transit close by the noise barrier it produces a pressure wave with a kinetic energy and potential. This energy is transferred to the blade orthogonally to the wave front. Then the blade stimulated by the wave, begin swinging in analogy to a damped harmonic oscillator.
Considering the kinetic energy E = ½ *m*□2A2 = ½*m*K/m*A2 = ½ *(a*b3)/(4h3)*□*A2, the blade dimensions (a, b, l), the material (□) and the extent of the first oscillation, the energy produced by the blade (per second) is: E ≈ 1 Joule or W ≈ 1 Watt. The electricity produced will be stored by appropriate capacitors. The electricity will be used to manage the noise barriers and to provide clean energy for the districts.

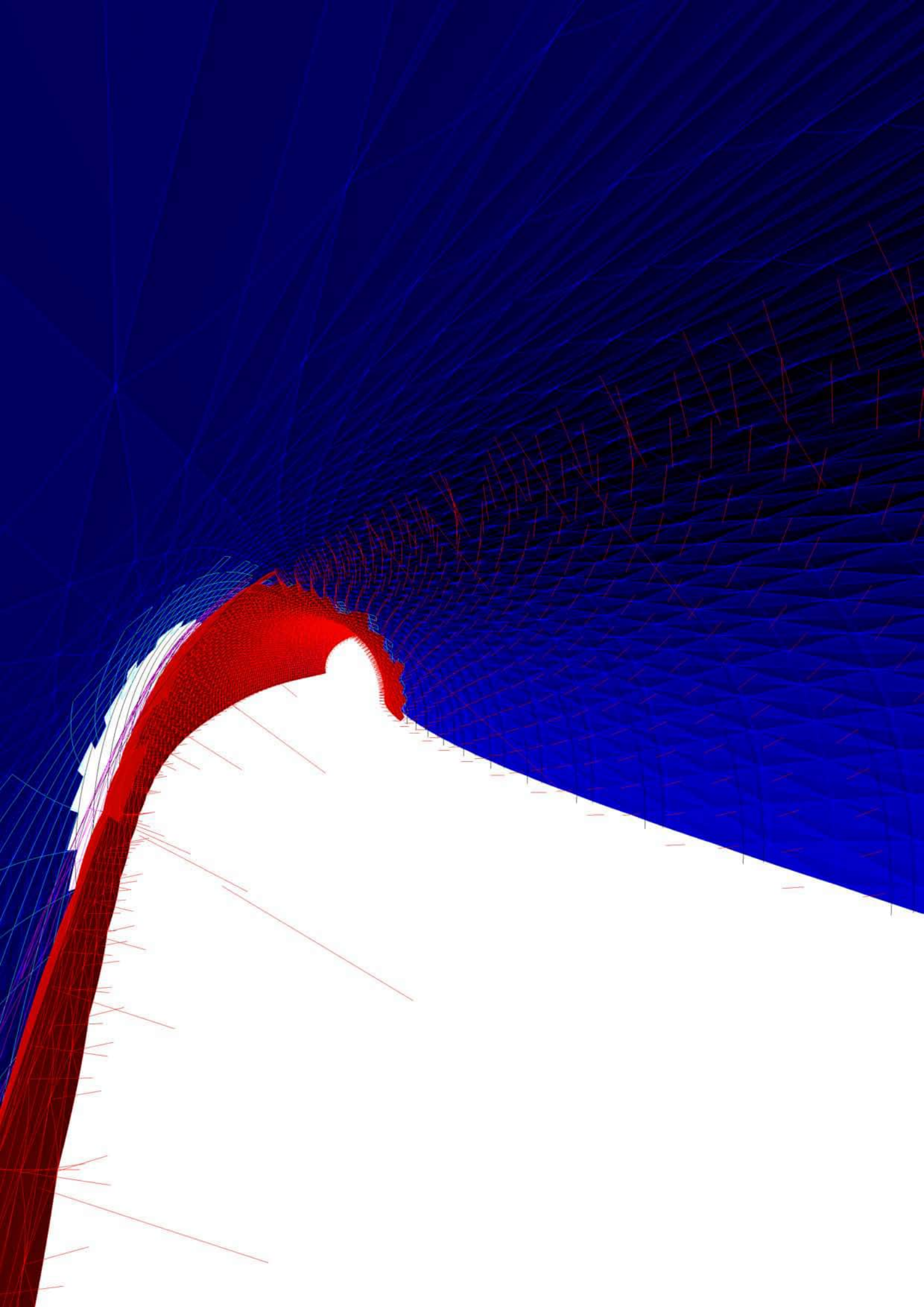
CREDITS
Title: Hong Kong Noise barrier
Location: Hong Kong
Organizer: Government of the Hong Kong Special Administrative Region
Cost: € 2 M
Materials: lenght 500m, height 6m, Steel 900t, 9000 polycarbonate panels
Project Design: Francesco Lipari
Project Team: Vanessa Todaro, Pierpaolo Pugliano
Timing: design 2009
Web information: www.francescolipari.it

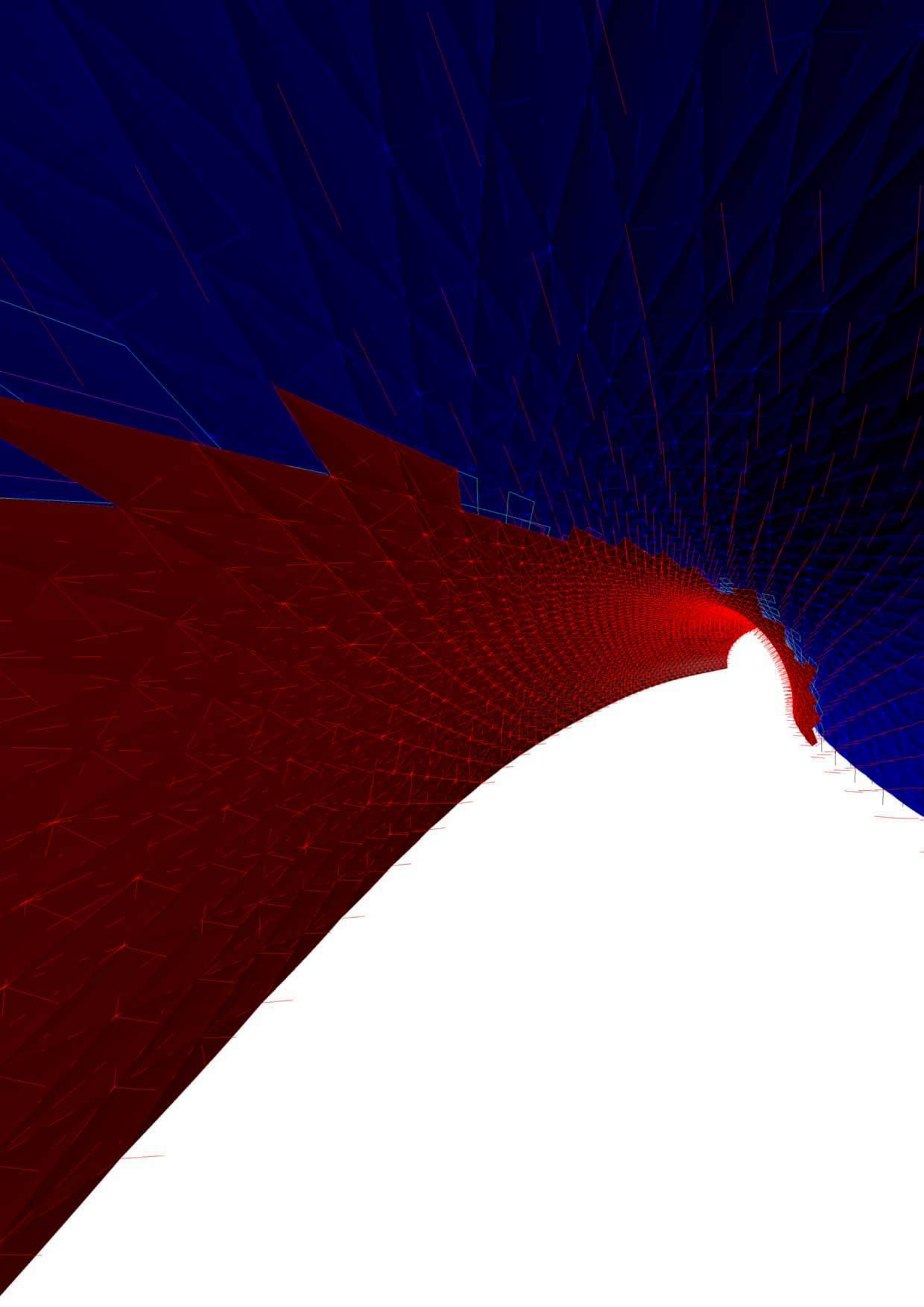


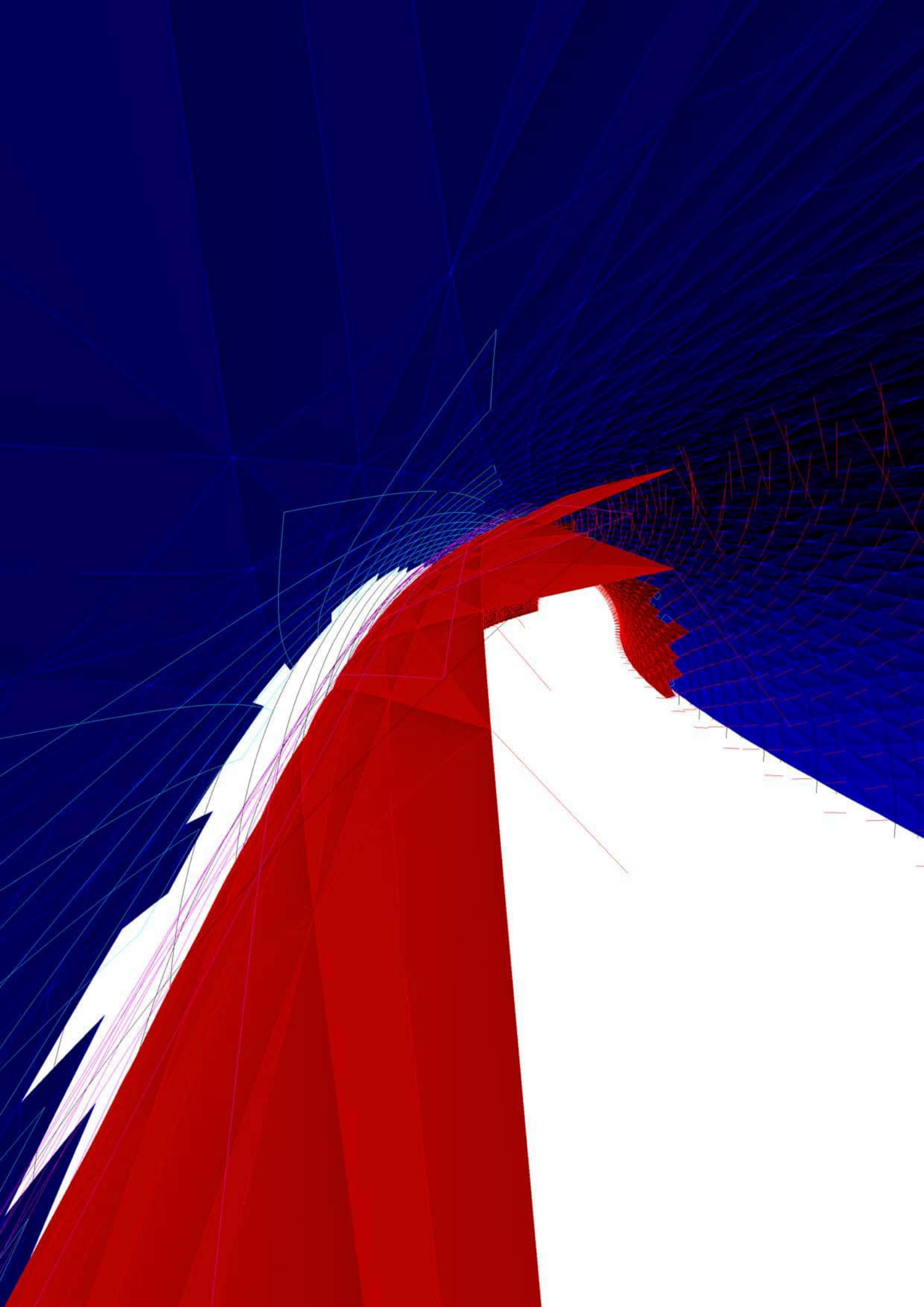


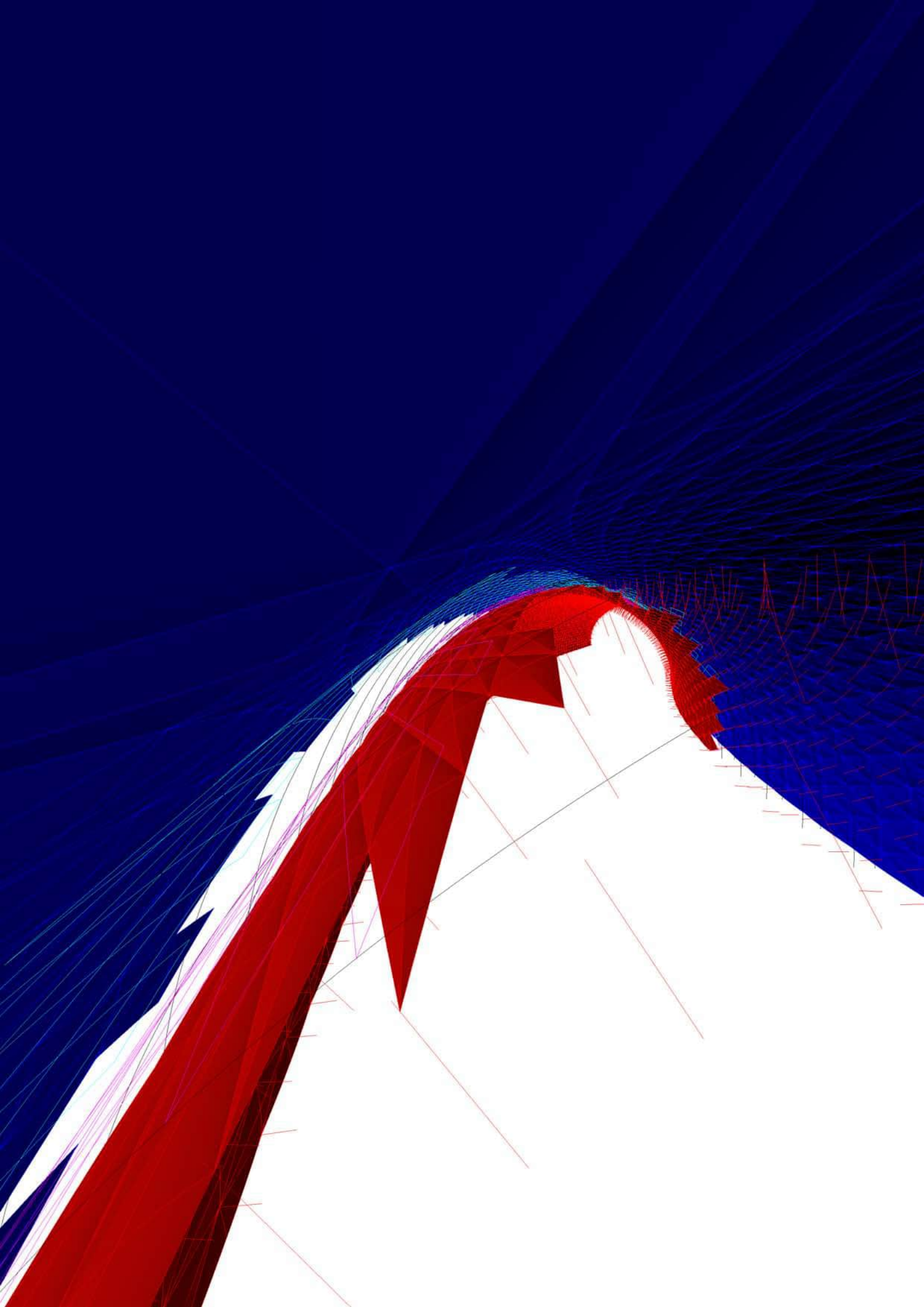


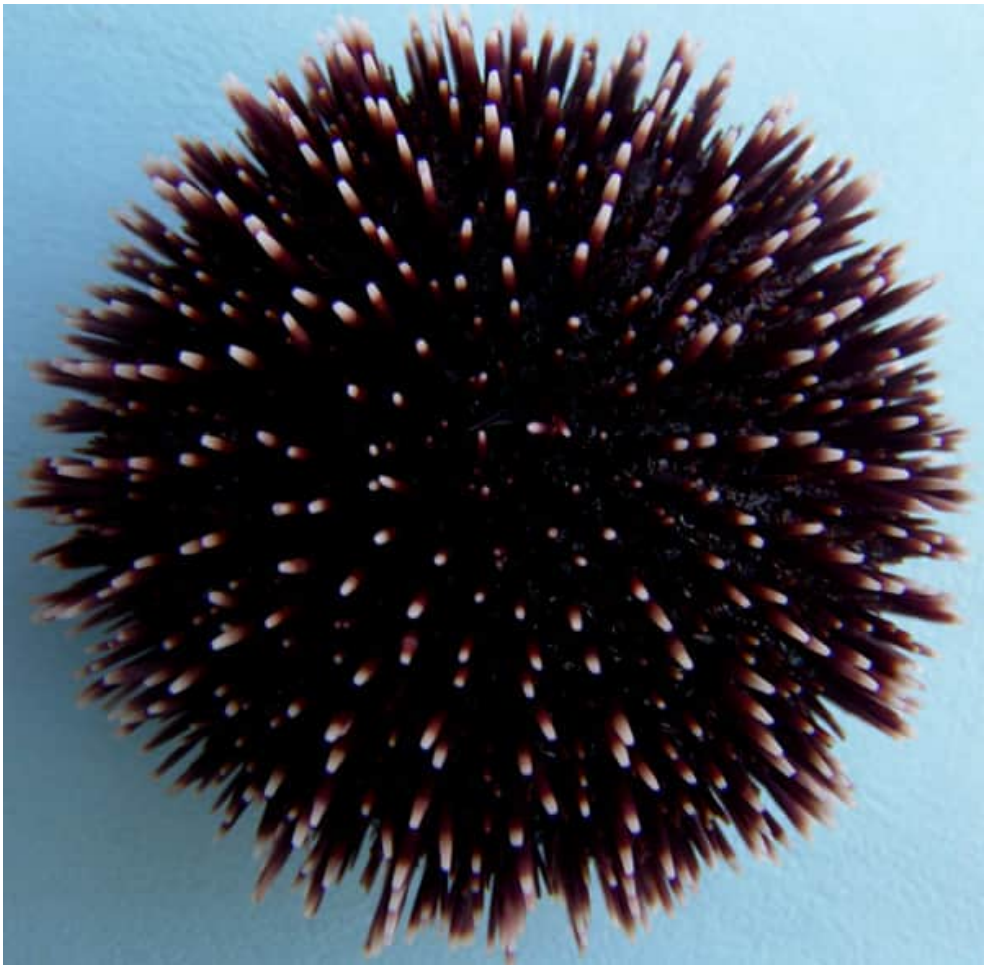


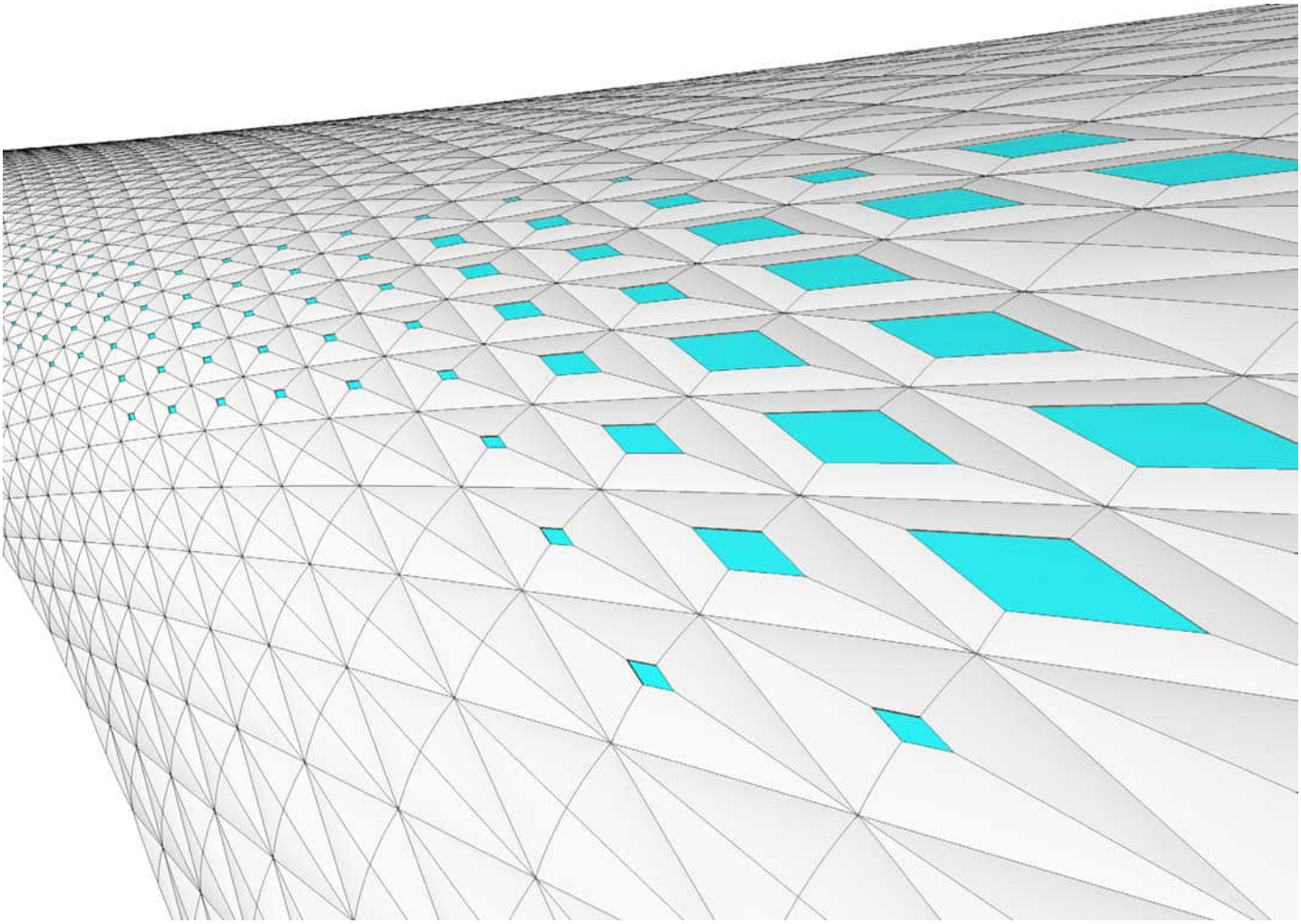


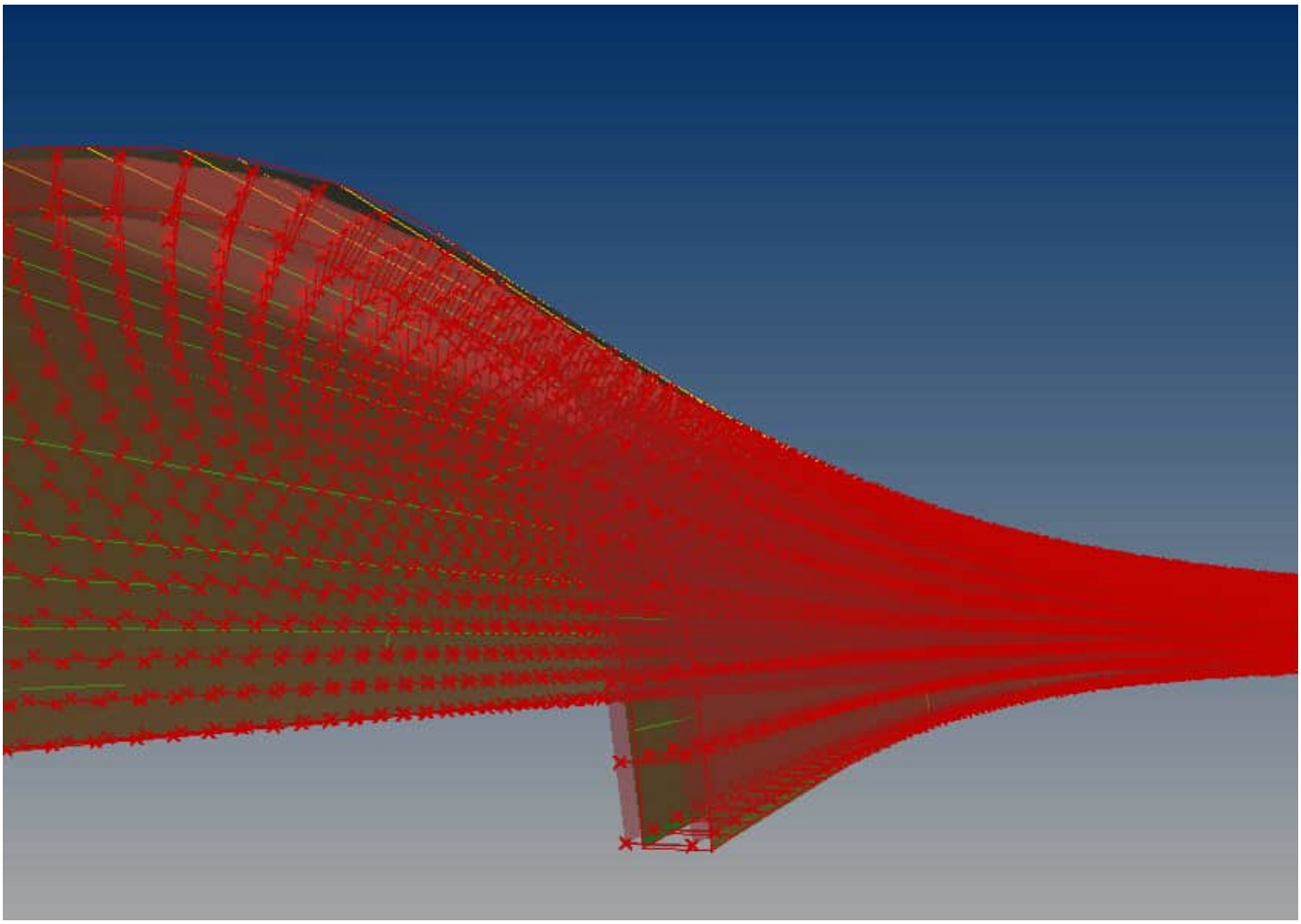


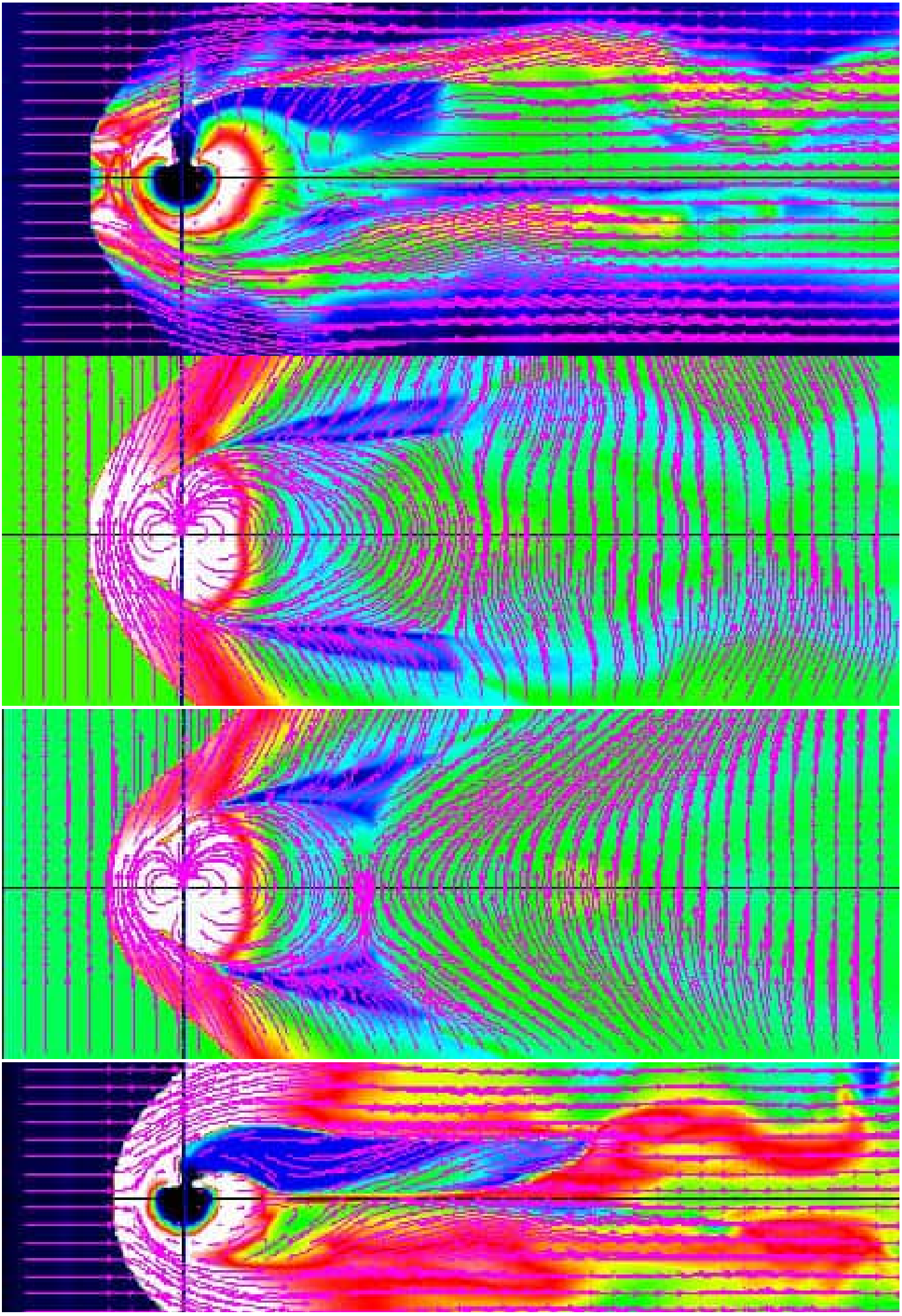


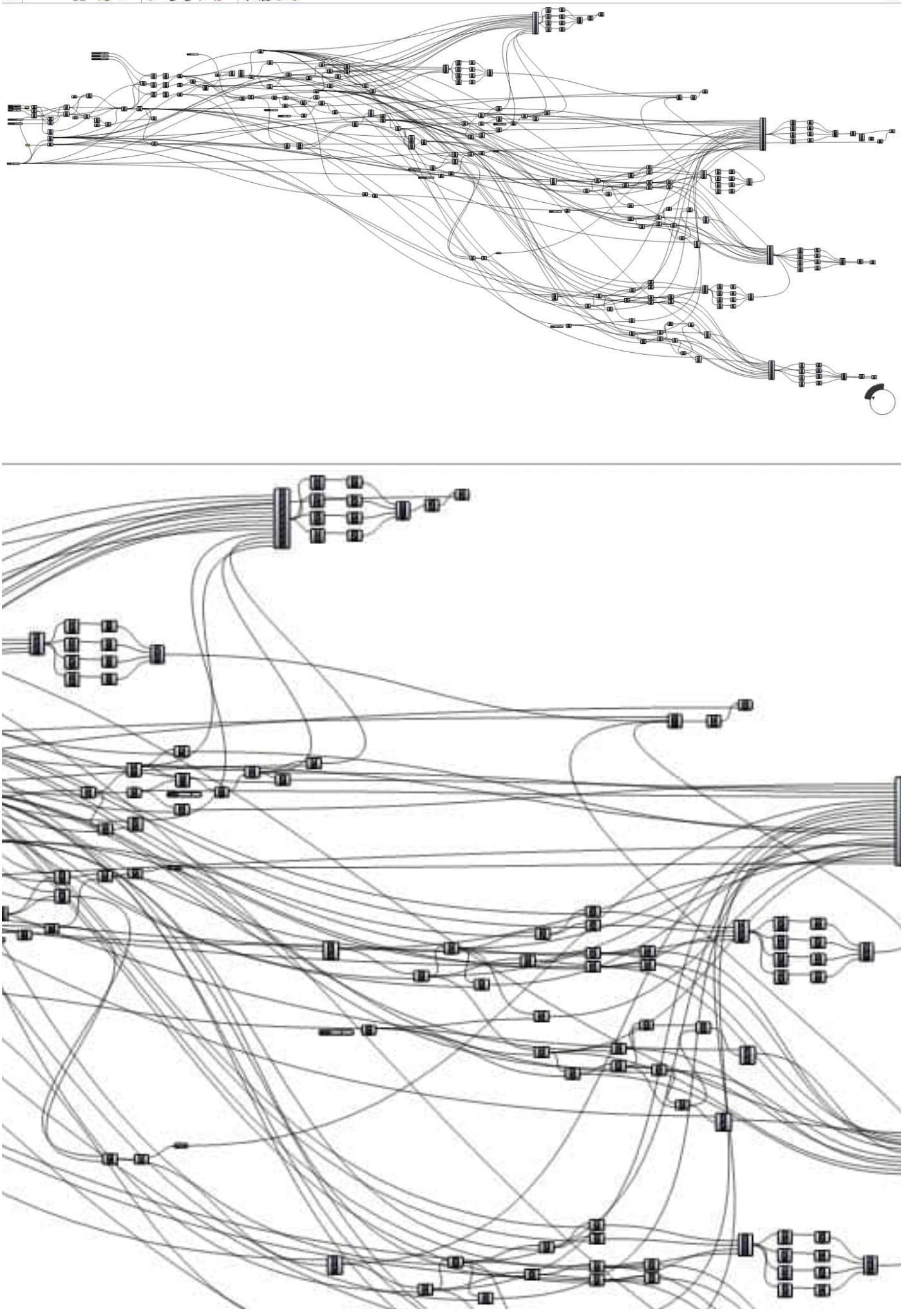












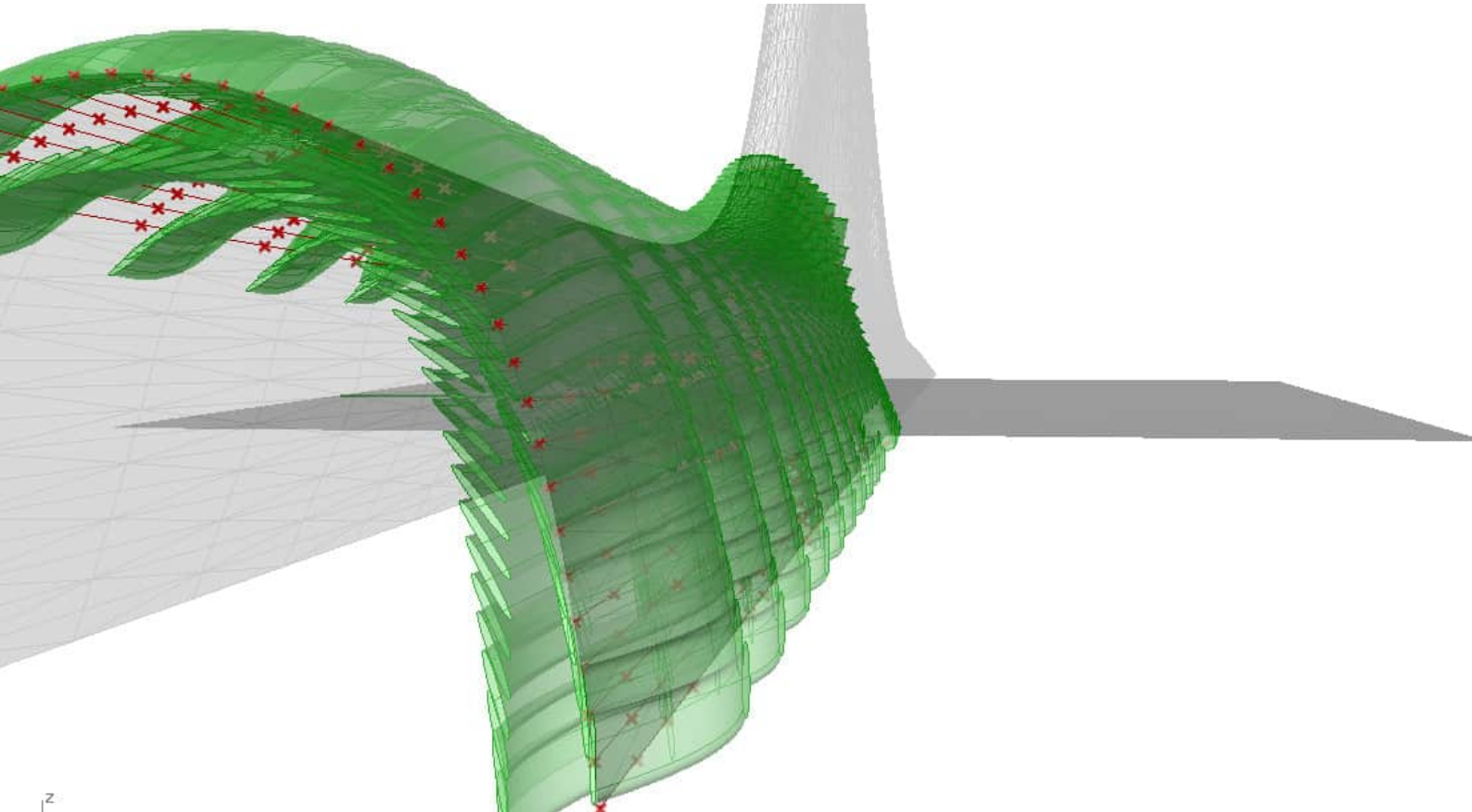
SILK ROAD MAP EVOLUTION

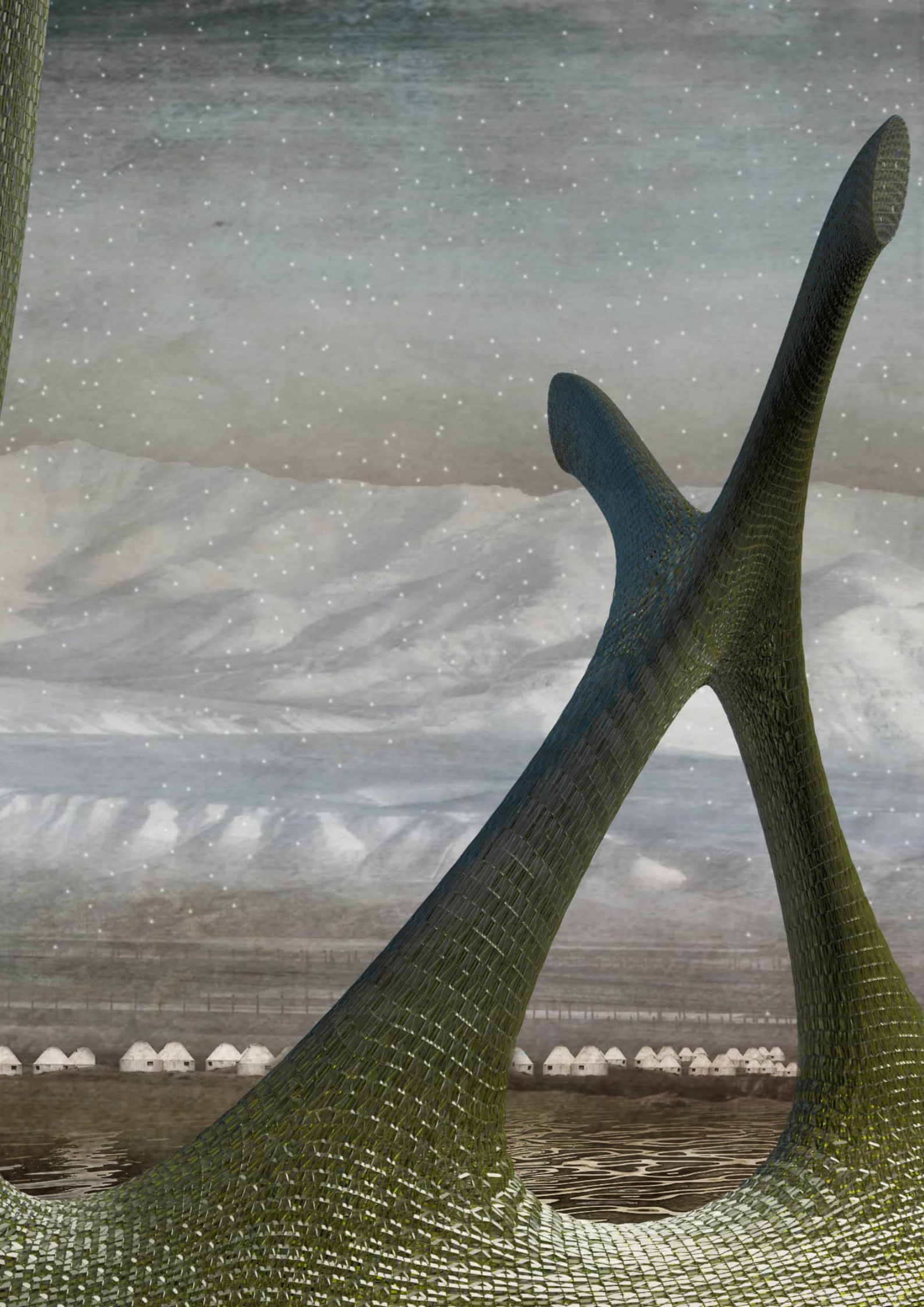
SRME. Silk Road Map Evolution is a project born out of the will to revive and regenerate the current layout of the silk road. This is to be accomplished by means of a social, economic, political and architectonic redevelopment of the historic stretch of the road that once belonged to Marco Polo.
The project deeply integrates infrastructure with architecture and by means of a new railway system functioning on gravitational platforms follows the trail from Venice to Xian, Shanghai and Tokyo, extending its "arms" to create new infrastructures, commercial services and residences.

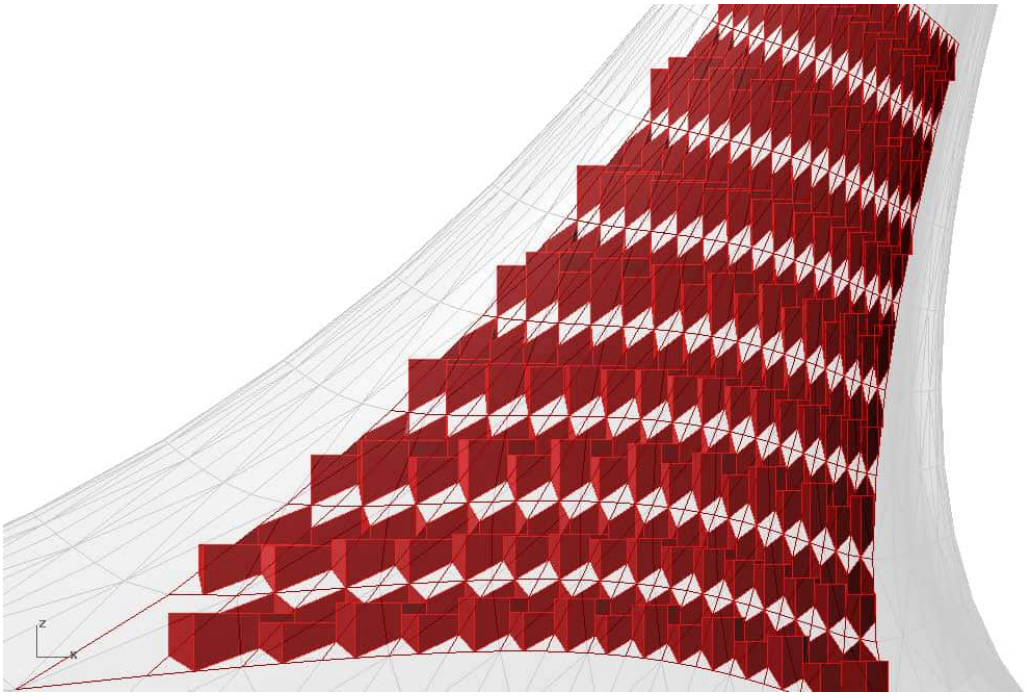
A wiry MOTOR CITY extends itself to help out urban realities and struggling economies. The (linearly) diffused city runs into other micro-cities in such a way that the greater entity hooks onto the smaller ones to help them survive and, like an economic pump, extends life from the greater nodes to the smaller and poorer extremities. The 15,000 km of the silk road shall be broken up by bionic towers which will represent the centers of new urban sprawls. The new silk road line will also serve as the GENERATOR of other paths that will branch off of the main course of the road to develop a larger economic armature.

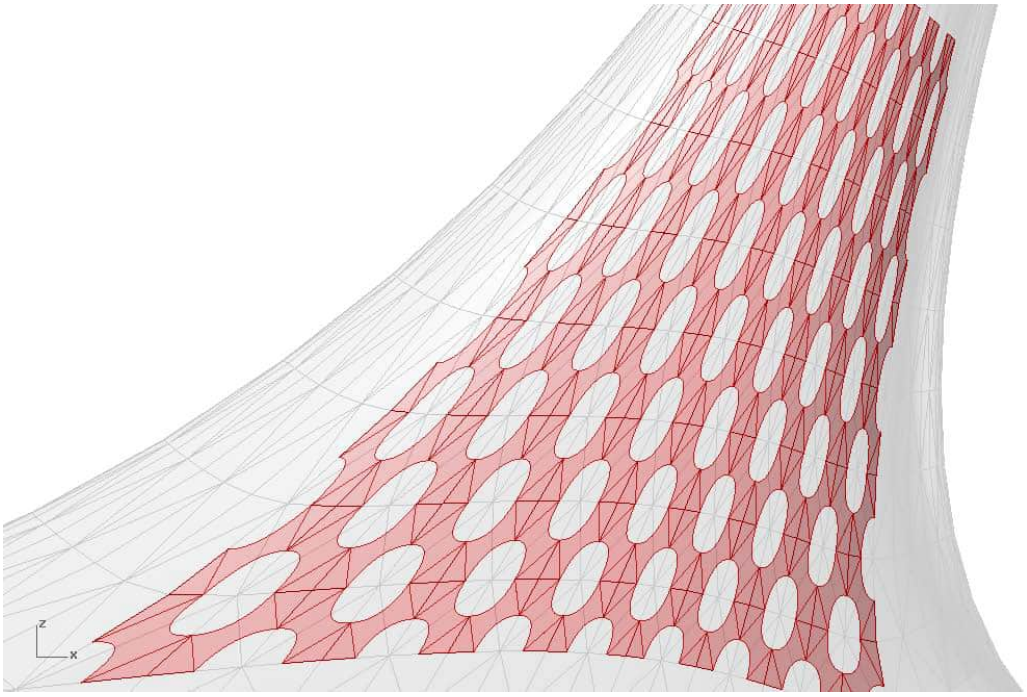
The studio driven towards the development of the SRME was created following an attentive analysis of the actual condition of the city on a global level. The common problem of large contemporary cities is that of congestion of circulation. This is a natural consequence of demographic expansion, a phenomenon that can be defined as explosive considering the exponential trend that manifested in the 20th century. The "classic" urban structure of cities developed concentrically around a central nucleus is no longer capable of resolving the problematic issues of traffic and pollution. As a result, the model of an ideal city will evolve, placing emphasis on MOBILITY and creating an environment in which displacements are reduced to a minimum and sustainable means of transport are implemented. The key issues of the innovative project of mobility are therefore the growing mechanization of displacements and the abandon of personal vehicles.

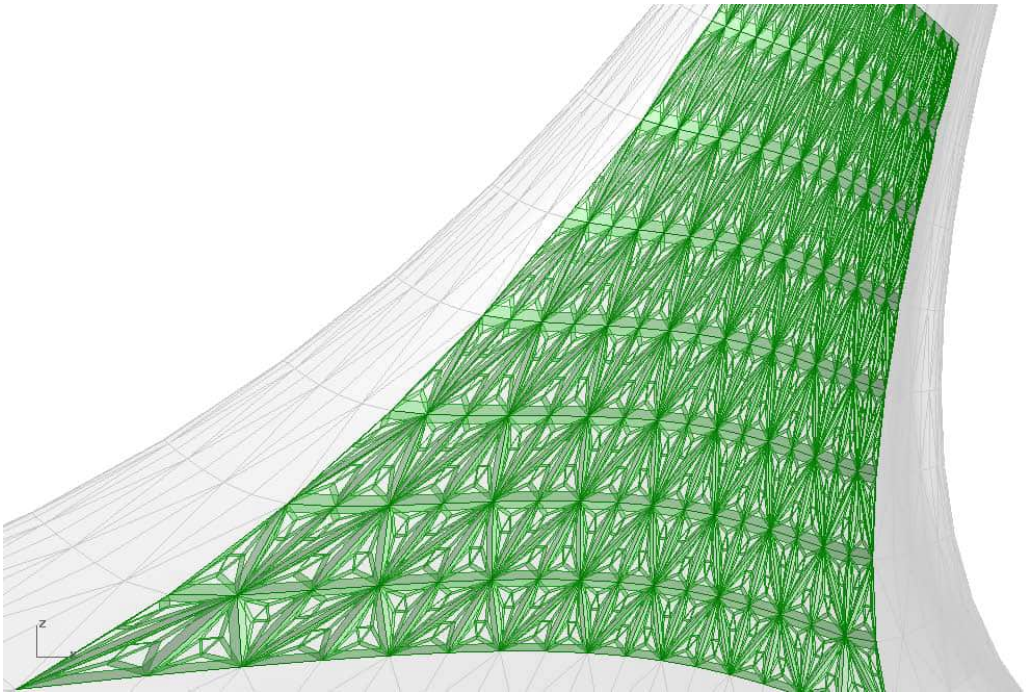
CREDITS
Design: OFL architecture
Site area: Silk Road Line
Construction area: 2000 sqm
Competition name: Silk Road Map International Competition
Organizers: New Italian Blood
Phase: International competition | first place
Construction budget: unknown
Year: 2010
Design team: Francesco Lipari, Vanessa Todaro, Andrea Debilio, Alejandro Liu Cheng (parametric design consultant)

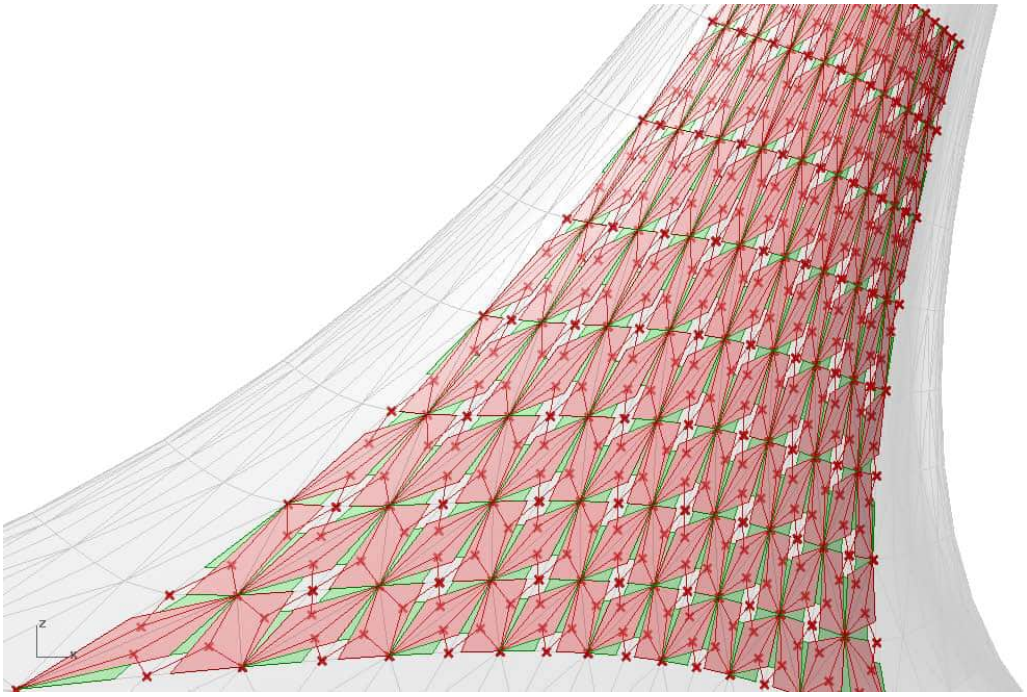


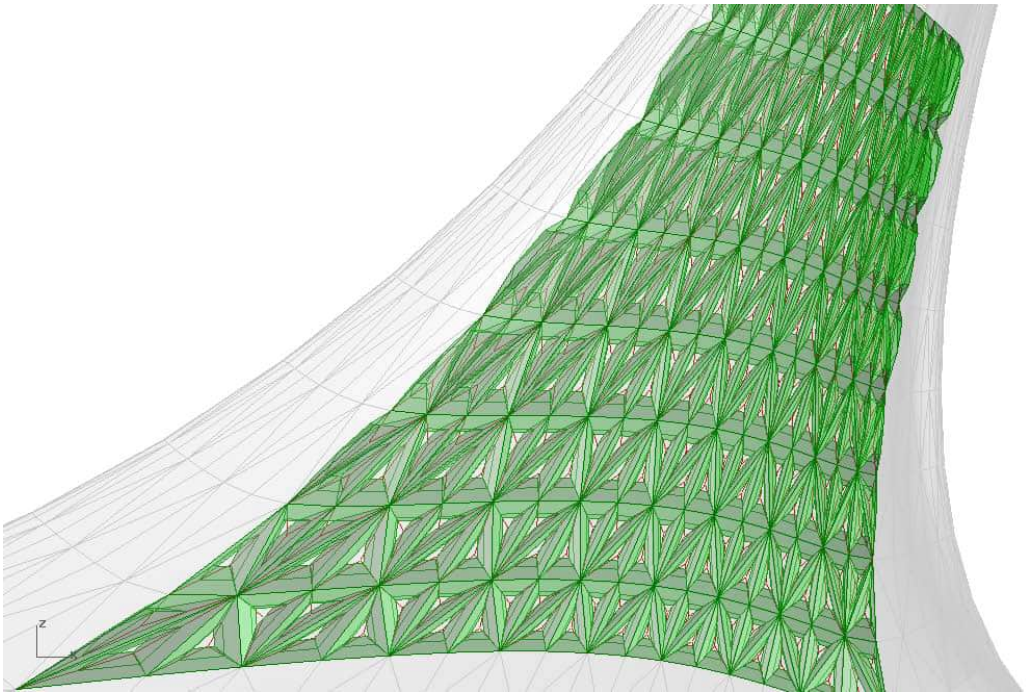




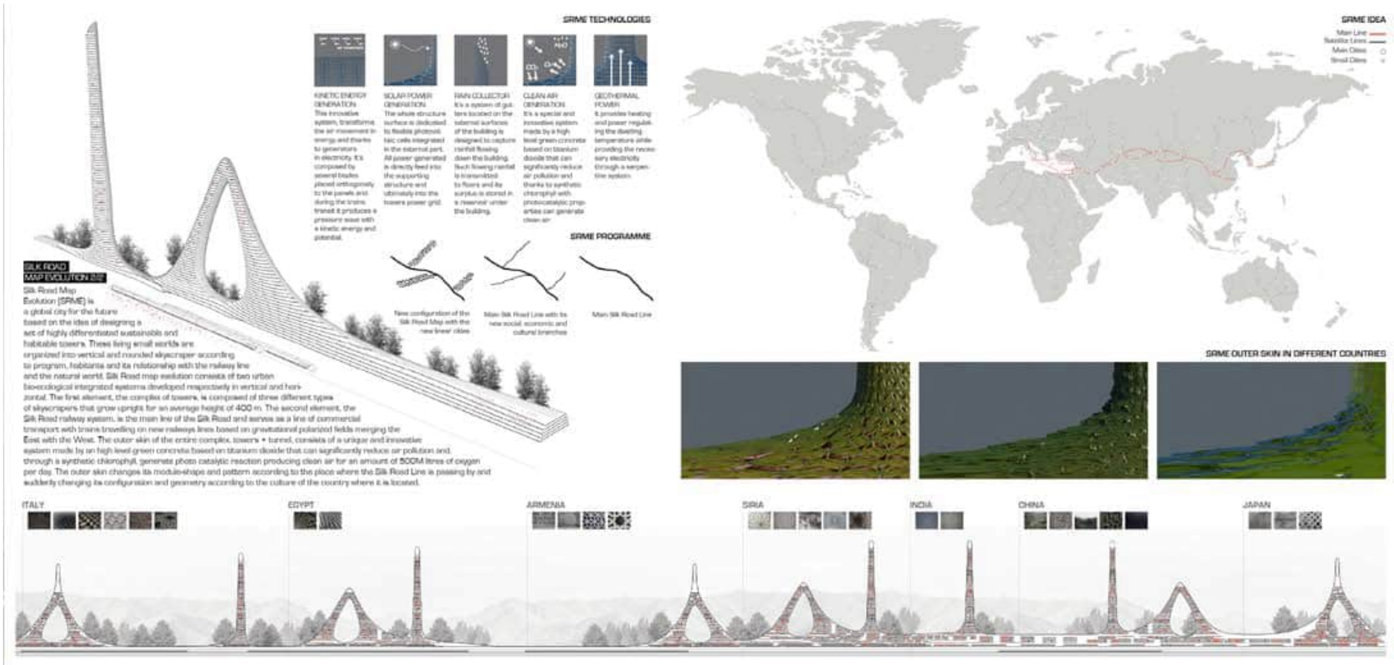


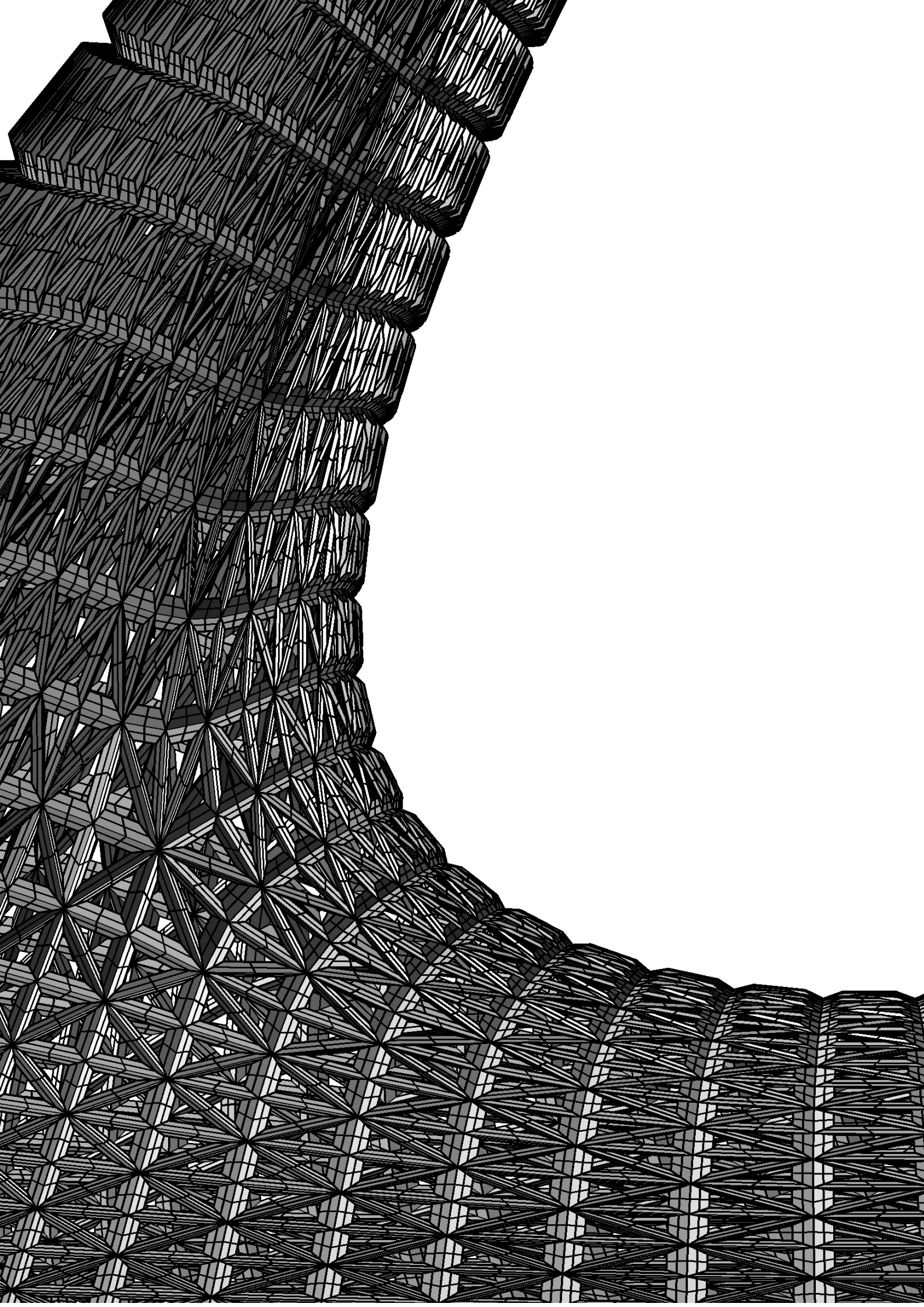


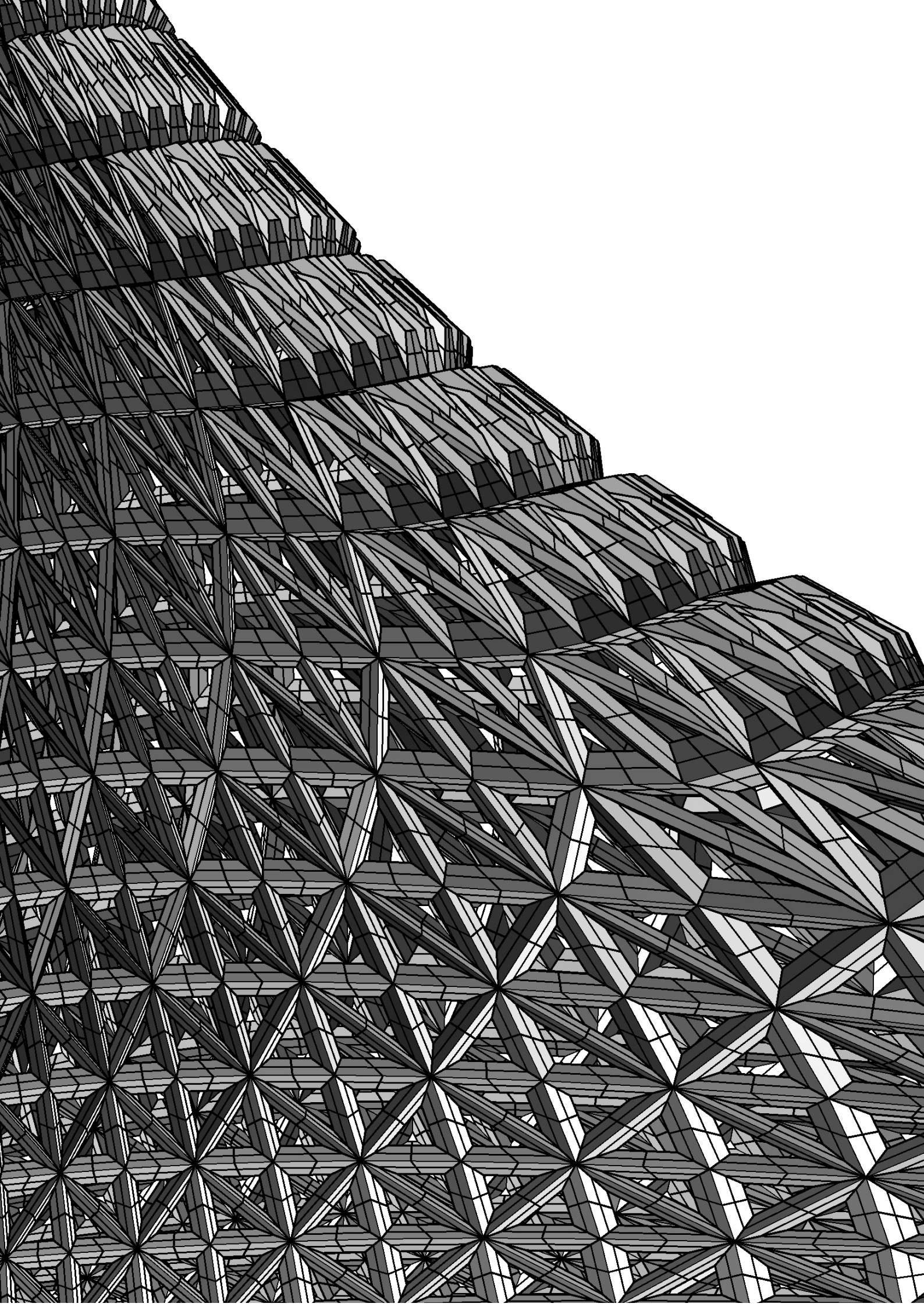


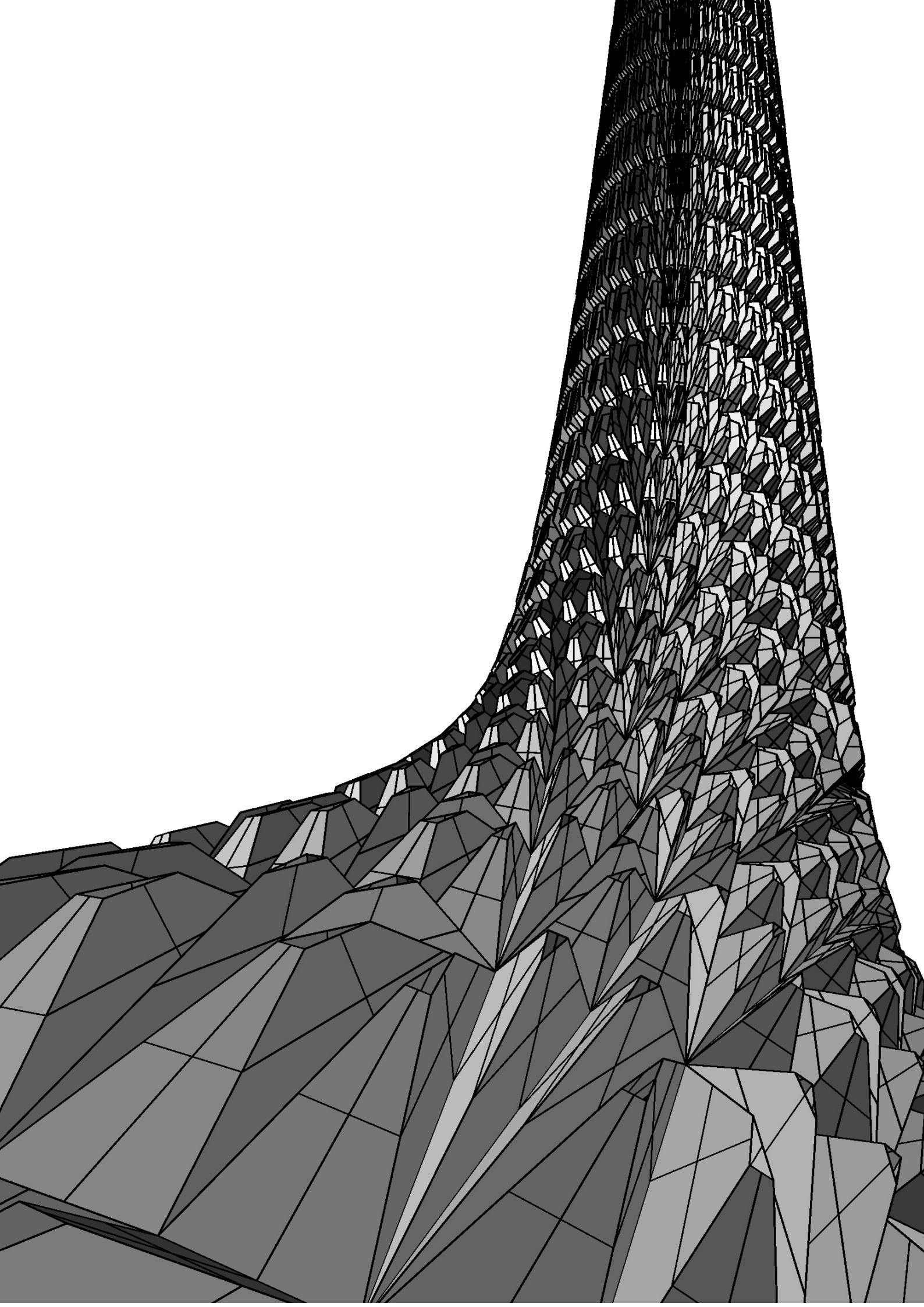


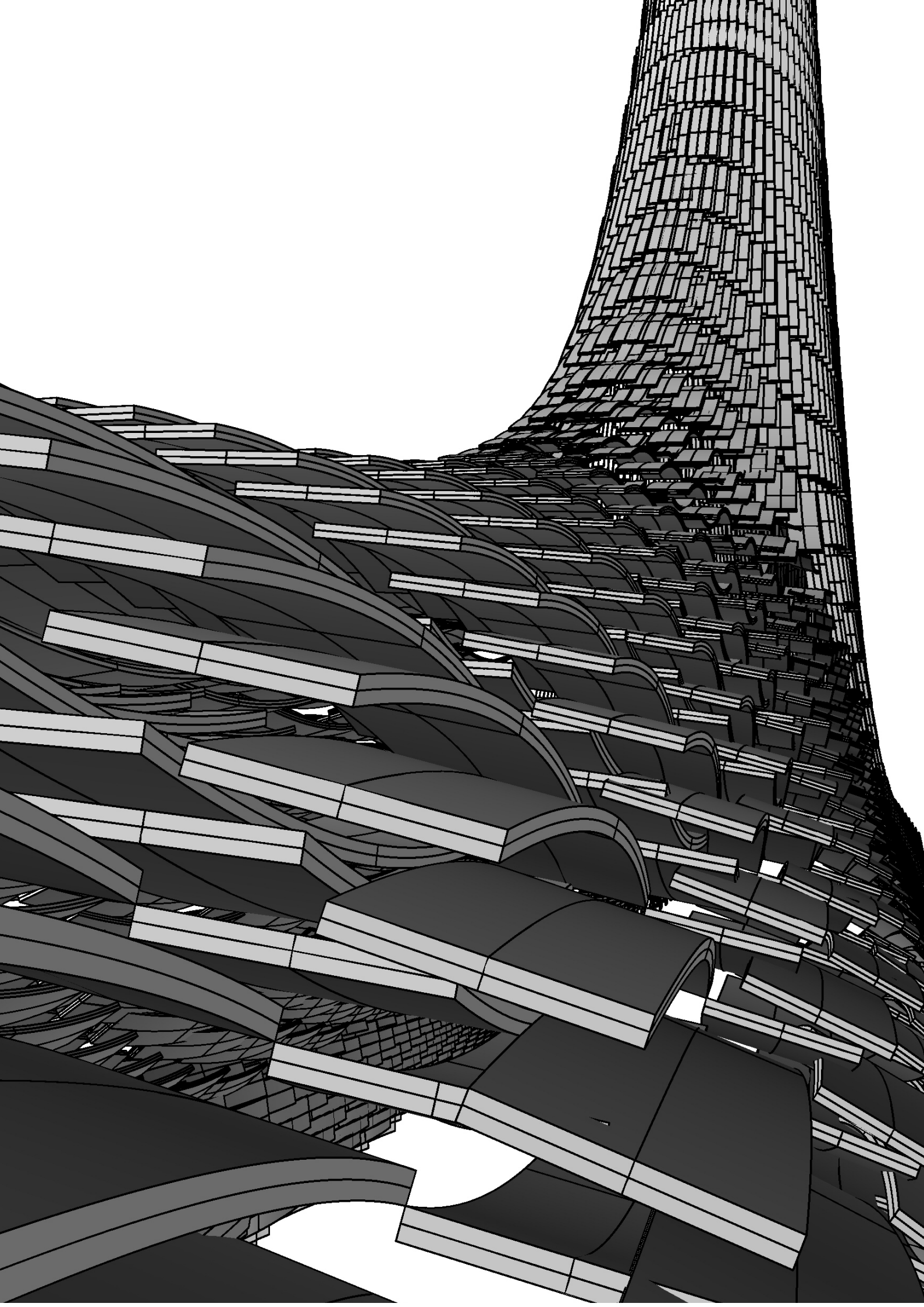

SILK ROAD MAP EVOLUTION 1/2
COORDINATES 32.129105,114.093018

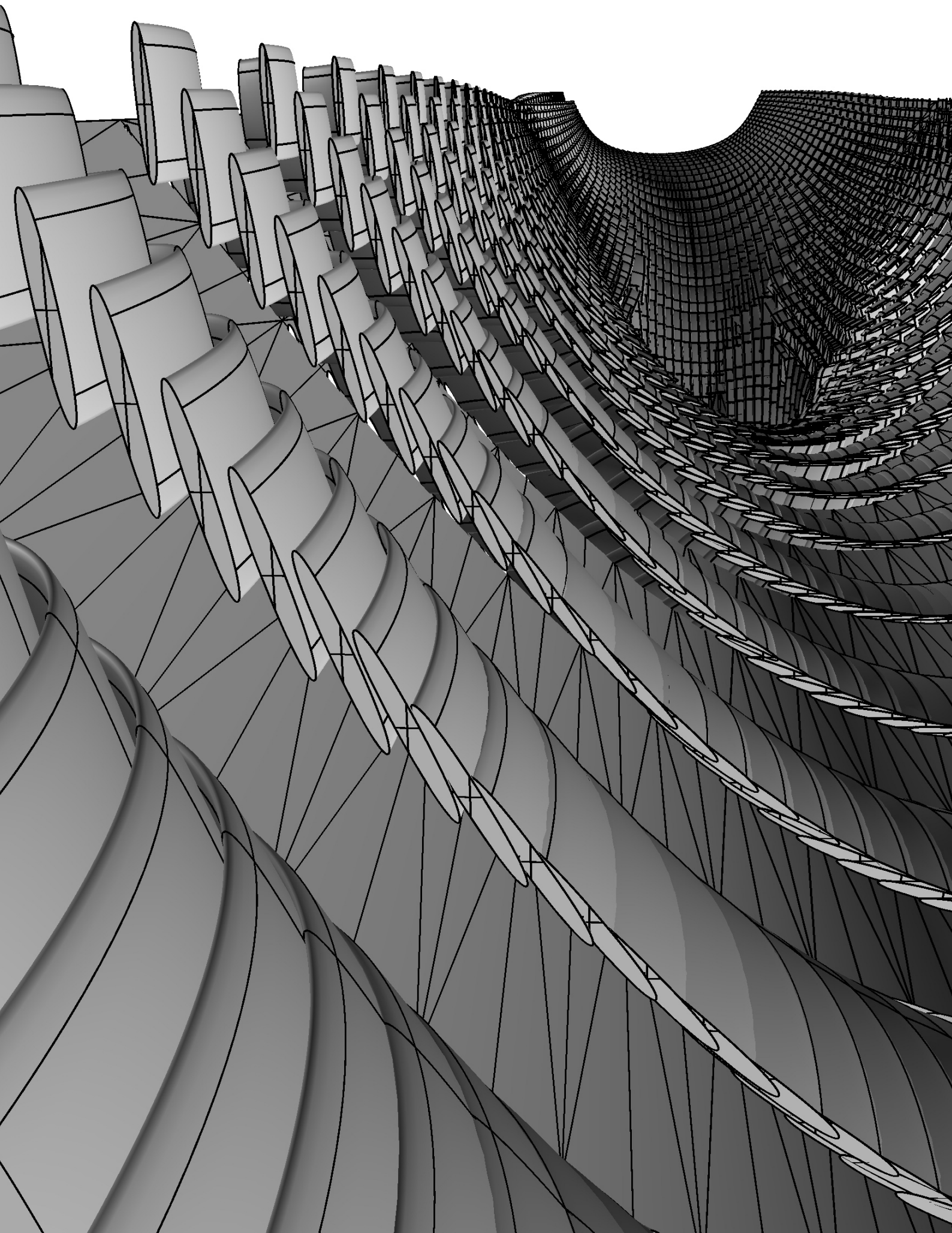

SRME TECHNOLOGIES
KINETIC ENERGY GENERATION
This innovative system, transforms the air movement in energy and thanks to generators in electricity. It's composed by several blades placed orthogonaly to the panels and during the trains transit it produces a pressure wave with a kinetic energy and potential.
SOLAR POWER GENERATION
The whole structure surface is dedicated to flexible photovoltaic cells integrated in the external part. All power generated is directly feed into the supporting structure and ultimately into the towers power grid.
RAIN COLLECTOR
It's a system of gutters located on the external surfaces of the building is designed to capture rainfall flowing down the building. Such flowing rainfall is transmitted to floors and its surplus is stored in a reservoir under the building.
CLEAN AIR GENERATION
It's a special and innovative system made by a high level green concrete based on titanium dioxide that can significantly reduce air pollution and thanks to synthetic chlorophyll with photocatalytic properties can generate clean air.
GEOTHERMAL POWER
It provides heating and power regulating the dwelling temperature while providing the necessary electricity through a serpentine system.
SRME PROGRAMME
New configuration of the Silk Road Map with the new linear cities
Main Silk Road Line with its new social, economic and cultural branches
Main Silk Road Line
SILK ROAD MAP EVOLUTION 2/2
Silk Road Map Evolution (SRME) is a global city for the future based on the idea of designing a set of highly differentiated sustainable and habitable towers. These living small worlds are organized into vertical and rounded skyscraper according to program, habitants and its relationship with the railway line and the natural world. Silk Road map evolution consists of two urban bio-ecological integrated systems developed respectively in vertical and horizontal. The first element, the complex of towers, is composed of three different types of skyscrapers that grow upright for an average height of 400 m. The second element, the Silk Road railway system, is the main line of the Silk Road and serves as a line of commercial transport with trains travelling on new railways lines based on gravitational polarized fields merging the East with the West. The outer skin of the entire complex, towers + tunnel, consists of a unique and innovative system made by an high level green concrete based on titanium dioxide that can significantly reduce air pollution and, through a synthetic chlorophyll, generate photo catalytic reaction producing clean air for an amount of 500M litres of oxygen per day. The outer skin changes its module-shape and pattern according to the place where the Silk Road Line is passing by and suddenly changing its configuration and geometry according to the culture of the country where it is located.
SRME IDEA
Main Line
Satellite Lines
Main Cities
Small Cities
SRME OUTER SKIN IN DIFFERENT COUNTRIES
ITALY
EGYPT
ARMENIA
SIRIA
INDIA
CHINA
JAPAN

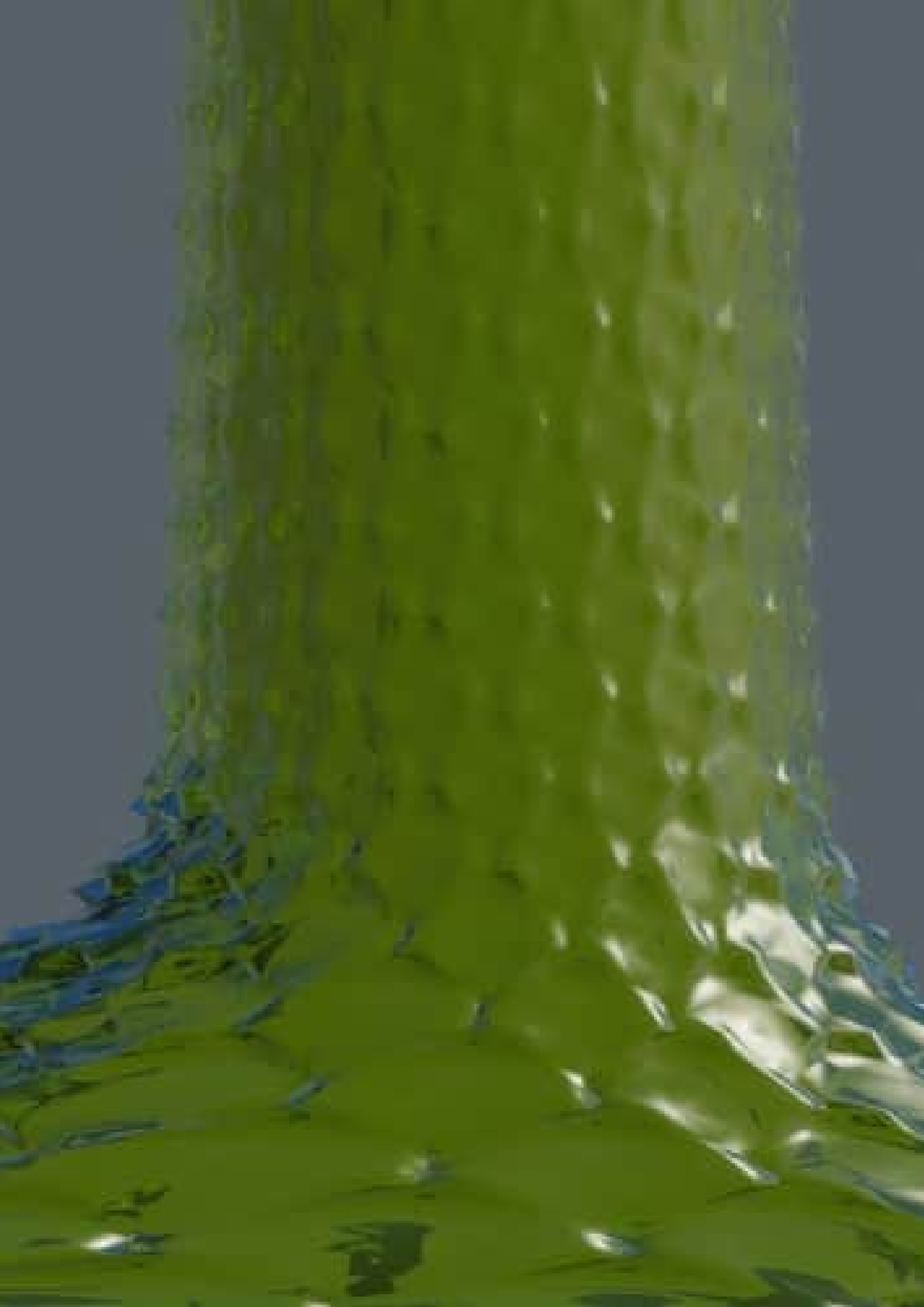


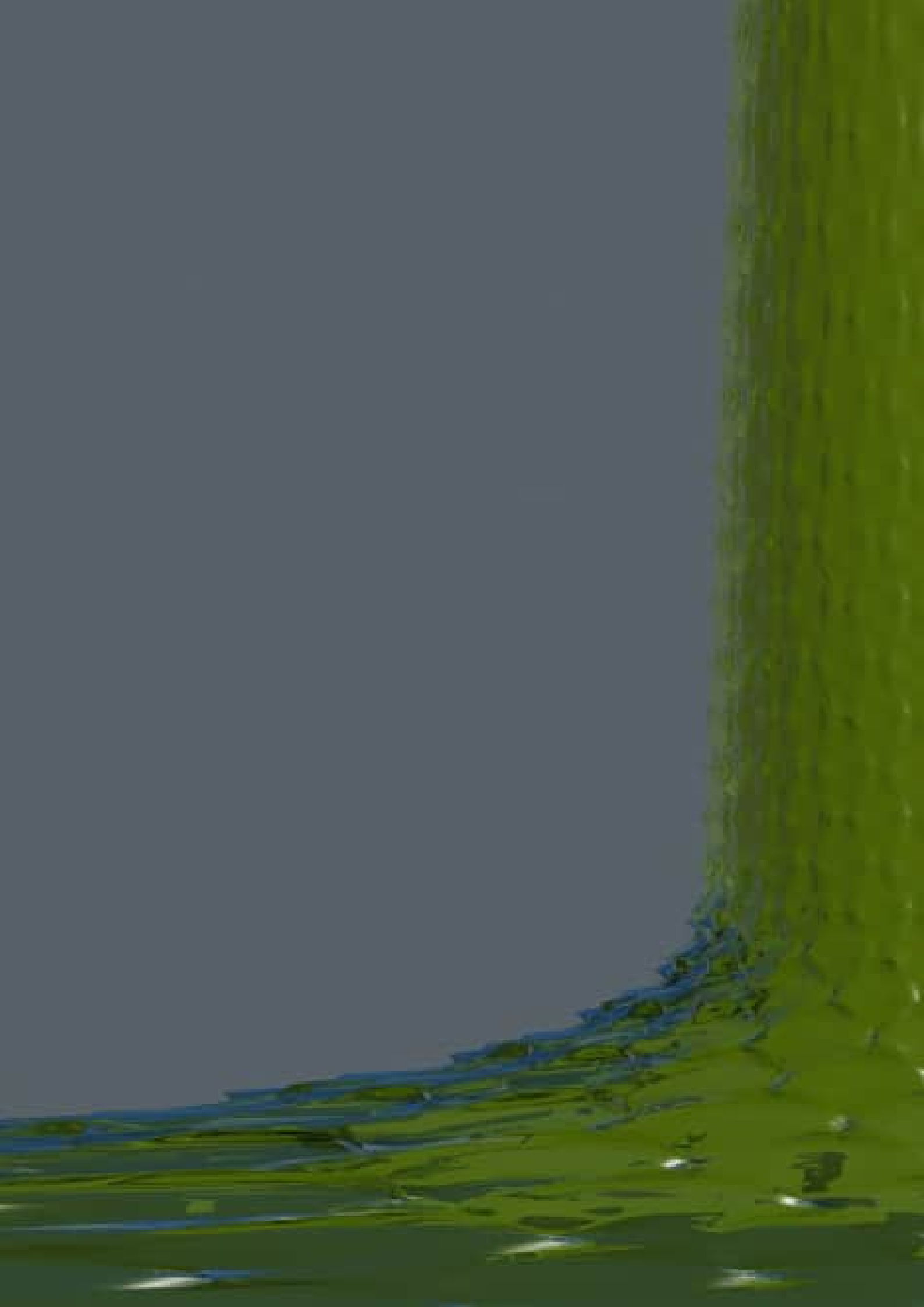


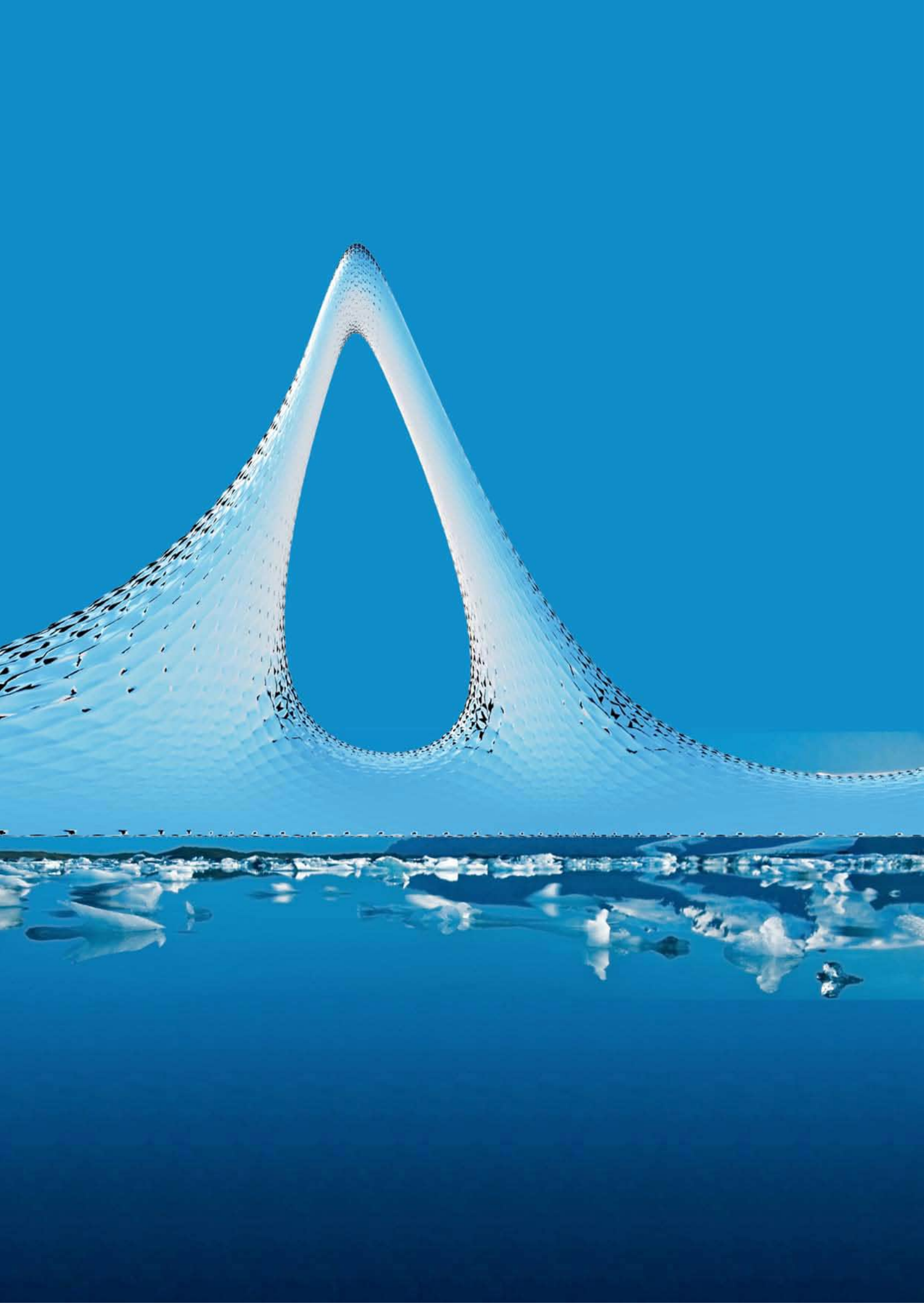


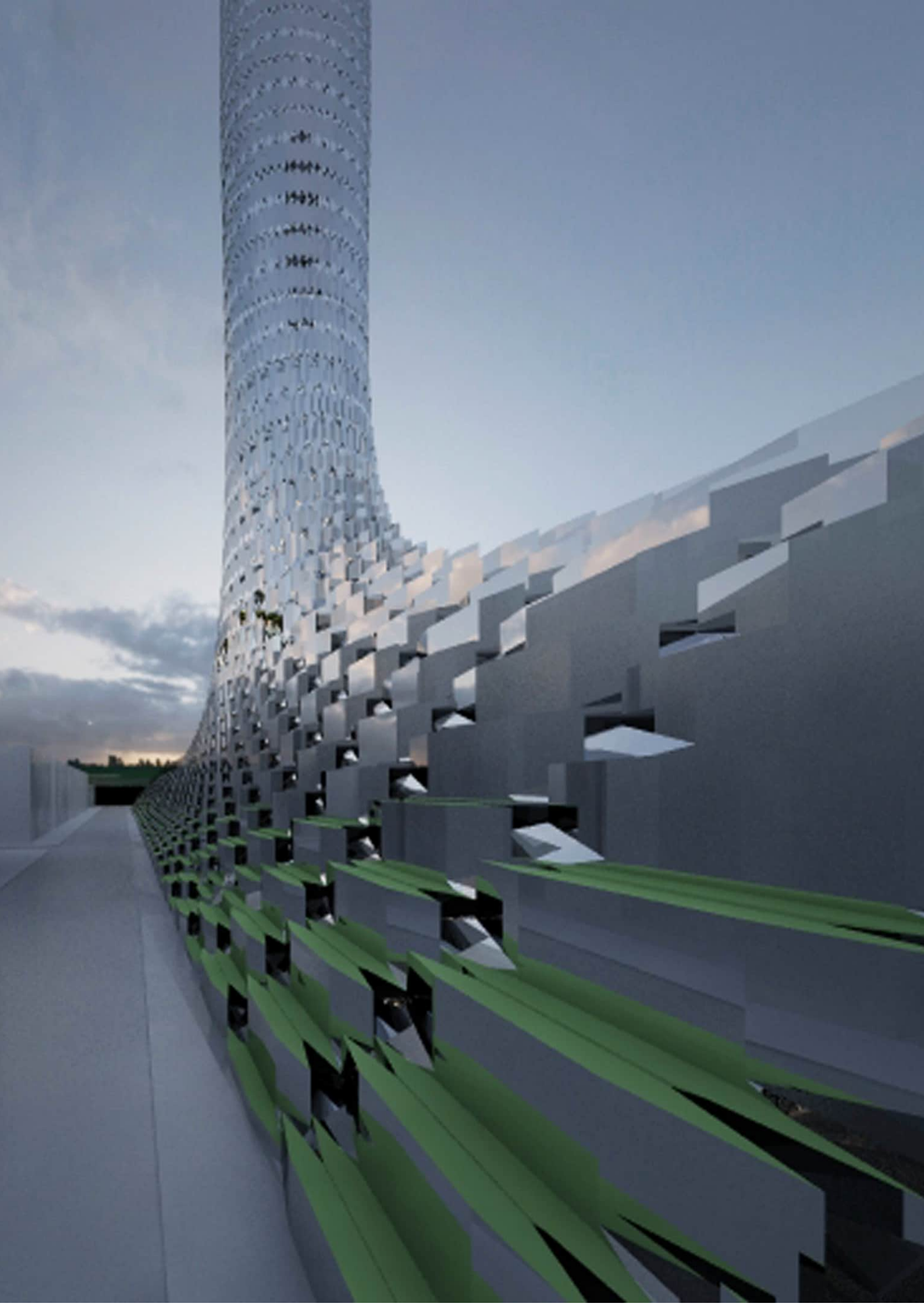










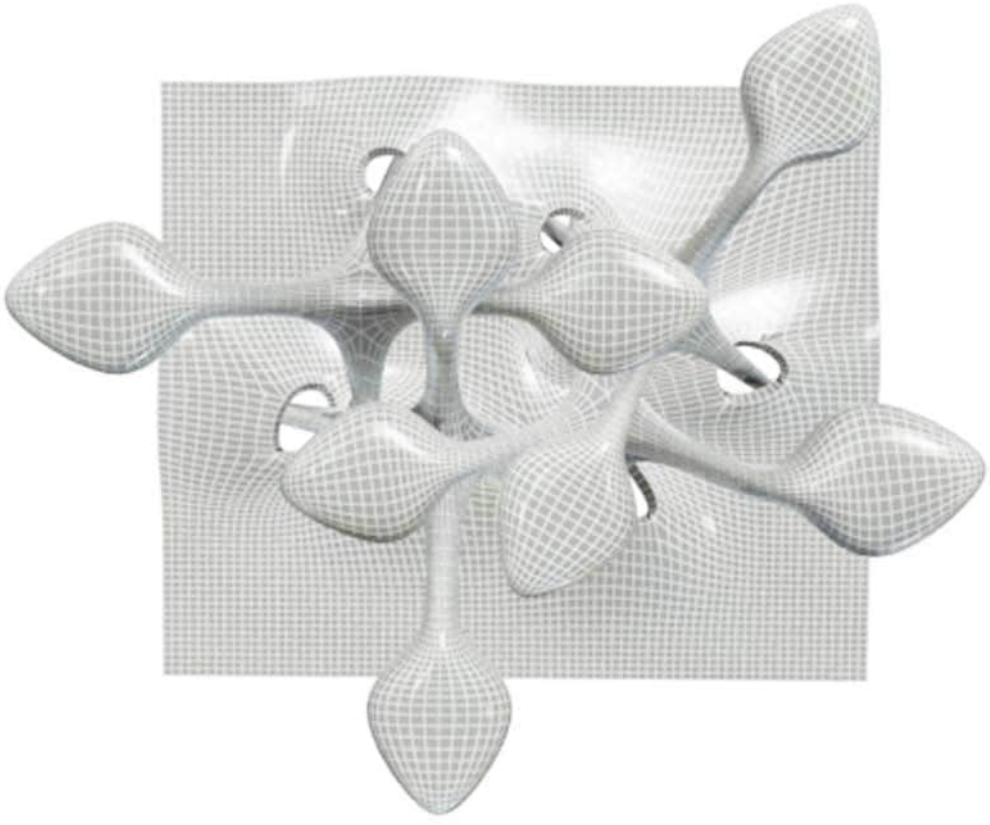


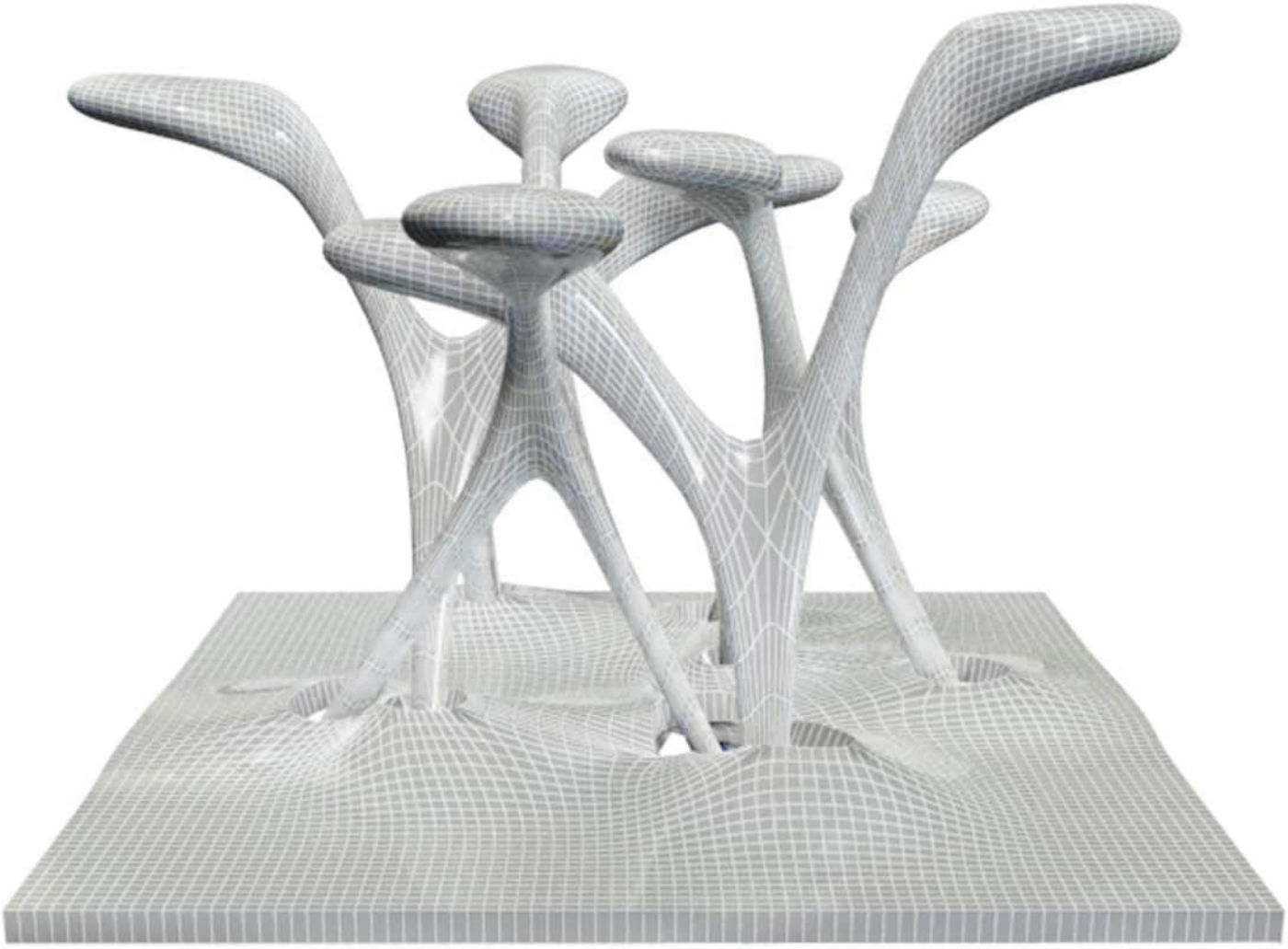
```
Option Explicit

Call Main()
Sub Main()
Dim idSrf : idSrf = Rhino.GetObject("Surface to recurse", 8, True, True)
If IsNull(idSrf) Then Exit Sub

Dim uDomain : uDomain = Rhino.SurfaceDomain(idSrf, 0)
Dim vDomain : vDomain = Rhino.SurfaceDomain(idSrf, 1)

Call RecurseSurface(idSrf, uDomain(0), uDomain(1), vDomain(0), vDomain(1))
End Sub

Function RecurseSurface(ByVal idSrf, ByVal u0, ByVal u1, ByVal v0, ByVal v1)
Dim um : um = u0 + 0.5 * (u1-u0)
Dim vm : vm = v0 + 0.5 * (v1-v0)

Dim A : A = Rhino.EvaluateSurface(idSrf, Array(u0, v0))
Dim B : B = Rhino.EvaluateSurface(idSrf, Array(u1, v0))
Dim C : C = Rhino.EvaluateSurface(idSrf, Array(u1, v1))
Dim D : D = Rhino.EvaluateSurface(idSrf, Array(u0, v1))
Dim E : E = Rhino.EvaluateSurface(idSrf, Array(um, vm))
Dim M(2)
m(0) = (A(0) + C(0))/2
m(1) = (A(1) + C(1))/2
m(2) = (A(2) + C(2))/2
Dim N(2)
n(0) = (D(0) + B(0))/2
n(1) = (D(1) + B(1))/2
n(2) = (D(2) + B(2))/2

If (rhino.Distance(E, M) < 0.5) And (rhino.Distance(E, N) < 0.5) Then
Call ExtrudeDomain (idSrf, u0, u1, v0, v1)
Else
Call RecurseSurface(idSrf, u0, um, v0, vm)
Call RecurseSurface(idSrf, um, u1, v0, vm)
Call RecurseSurface(idSrf, u0, um, vm, v1)
Call RecurseSurface(idSrf, um, u1, vm, v1)
End If
End Function

Sub ExtrudeDomain(ByVal idSrf, ByVal u0, ByVal u1, ByVal v0, ByVal v1)
Dim A : A = Rhino.EvaluateSurface(idSrf, Array(u0, v0))
Dim B : B = Rhino.EvaluateSurface(idSrf, Array(u1, v0))
Dim C : C = Rhino.EvaluateSurface(idSrf, Array(u1, v1))
Dim D : D = Rhino.EvaluateSurface(idSrf, Array(u0, v1))

Dim Na : Na = Rhino.SurfaceNormal(idSrf, Array(u0, v0))
Dim Nb : Nb = Rhino.SurfaceNormal(idSrf, Array(u1, v0))
Dim Nc : Nc = Rhino.SurfaceNormal(idSrf, Array(u1, v1))
Dim Nd : Nd = Rhino.SurfaceNormal(idSrf, Array(u0, v1))

Dim E : E = Rhino.vectoradd(A, Na)
Dim F : F = Rhino.vectoradd(B, Nb)
Dim G : G = Rhino.vectoradd(C, Nc)
Dim H : H = Rhino.vectoradd(D, Nd)

Dim idBase, idTop
idBase = Rhino.AddCurve(Array(A, B, C, D, A), 1)
idTop = Rhino.AddCurve(Array(E, F, G, H, E), 1)

Call Rhino.AddLoftSrf(Array(idBase, idTop))
Call Rhino.DeleteObject (idBase)
Call Rhino.DeleteObject (idTop)
```

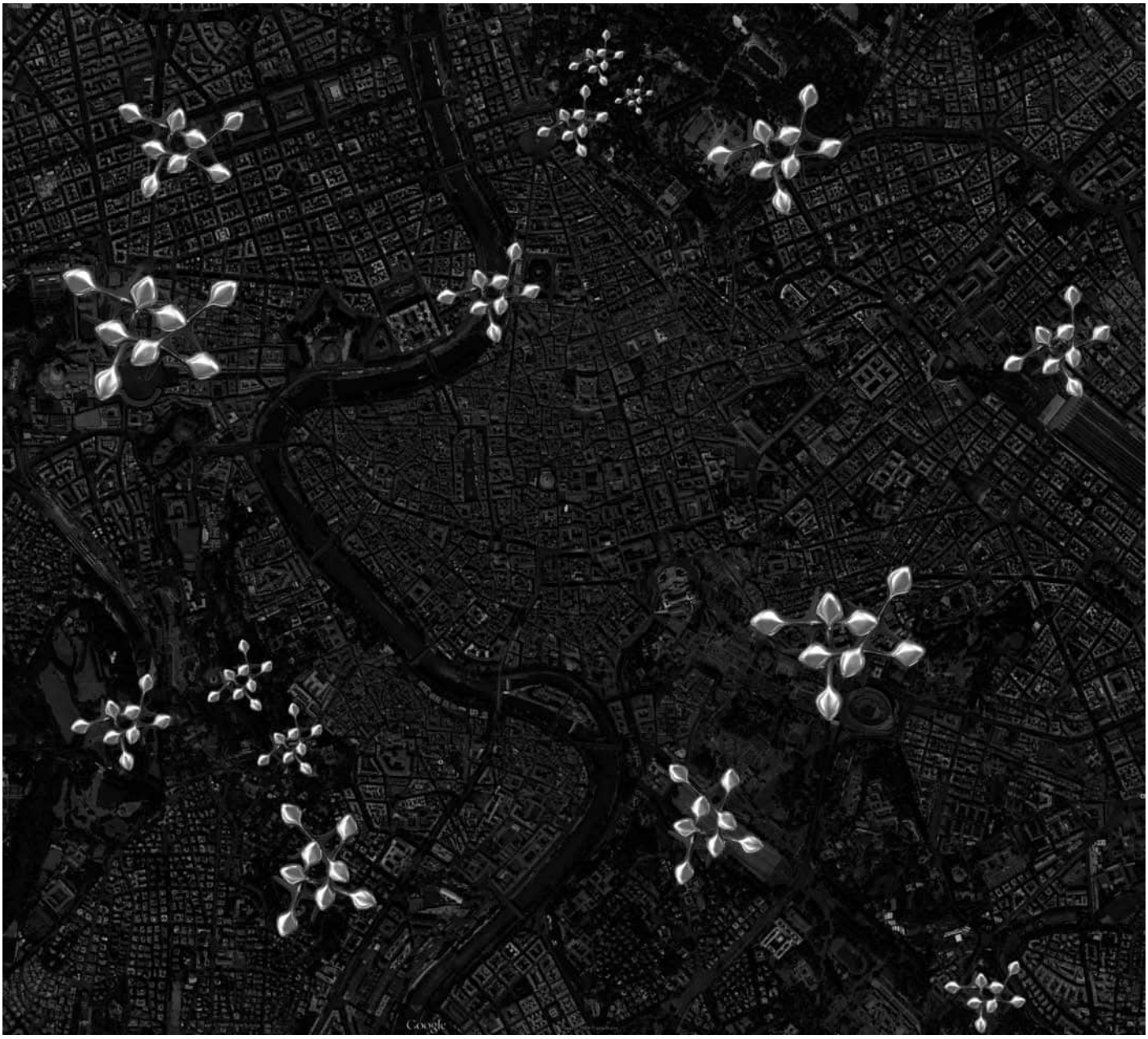
ENOKI ROME ECOCITY

Enoki rome ecocity is a project born from the desire of investigate the possibilities for future housing. The achievements in material science, energy conservation, aerodynamic and environmental solutions allow designers to experience new housing typologies that can be housed in self sufficient and highly innovative building envelopes. Paying homage to the new millennium and to the city of Rome with the intention of stimulating and supporting the contemporary city. The Enoki project is installed above the actual city of Rome and it integrates itself with the historical part of the city. It draws its' lifeblood from parks, green areas and water courses to push itself upward. They are small self-contained cities with residences, commercial spaces, green areas, spaces for community activities, sport and cultural activities. The outer skin, made with steel diamond-shaped panels, appropriately follows the main cellular structure of the enoki made from steel and glass and having a molecular shape. A thermoformed glass surface provides a lucid and static resistance with the possibility of infinite curvatures with lower costs.
The Enoki project consists of 150 stories. The classic lifts are replaced by flying shuttles that run outside the enoki bringing inhabitants to and from different levels of the building and the old city. Enoki can accommodate up to 6,000 residents with an area of 240,000 square meters and 300,000 square meters for recreational activities for the "new city".

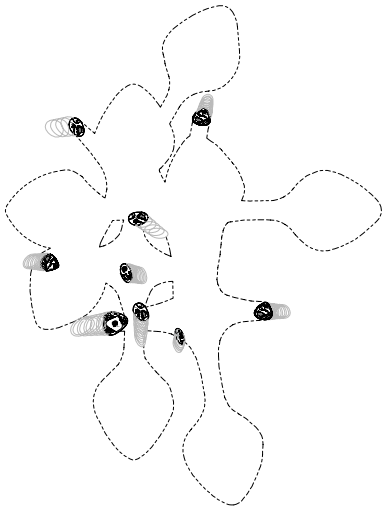
CREDITS
Plan's title: Enoki Rome Ecocity
Location: Rome
Project Design: Francesco Lipari
Timing: design 2010
Web information: www.francescolipari.it



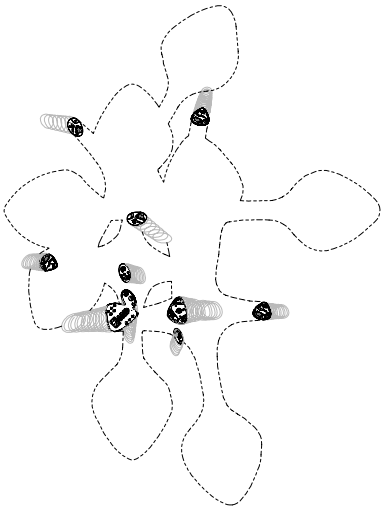



Google

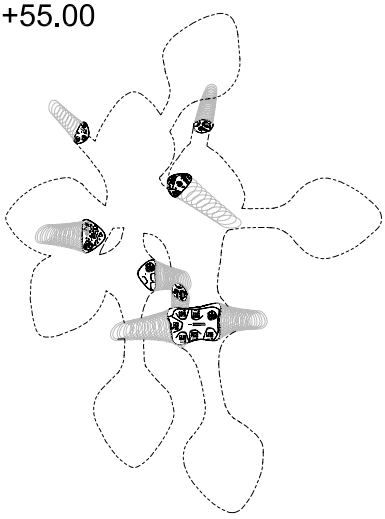
+30.00



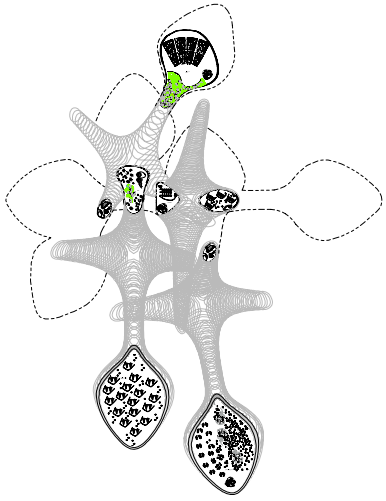
+35.00



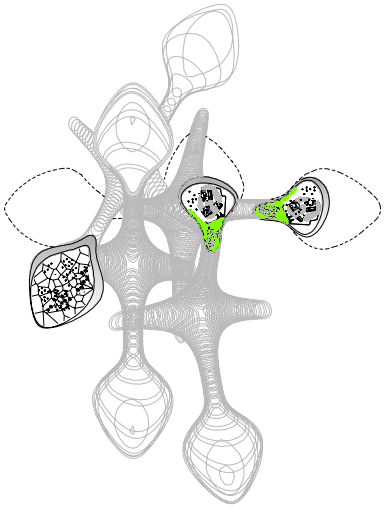
+55.00



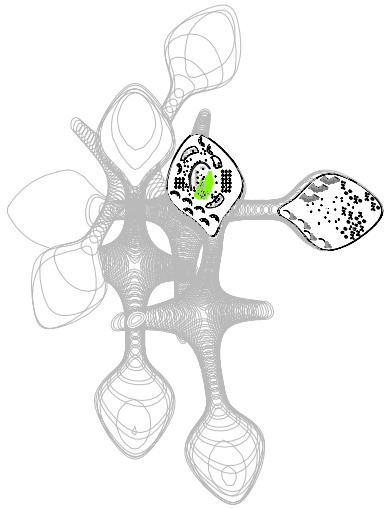
+170.00



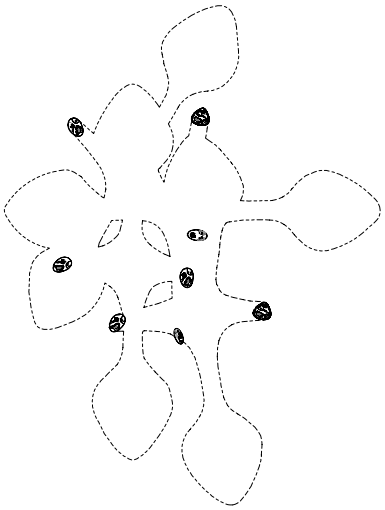
+200.00



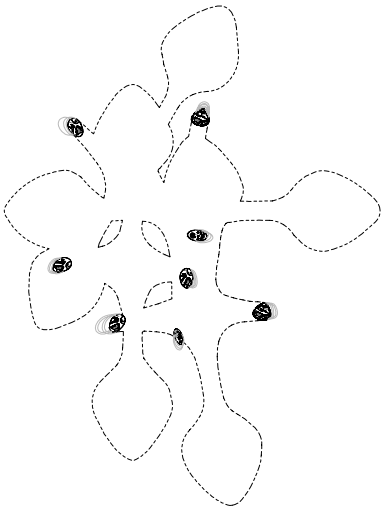
+215.00



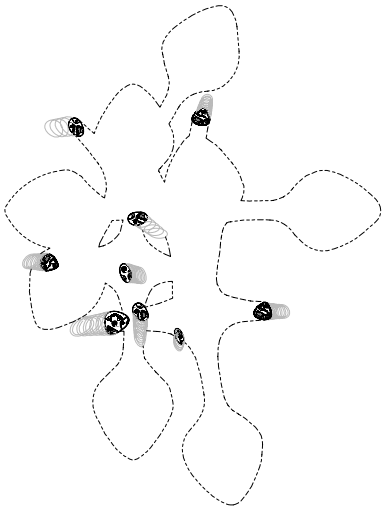
0.00



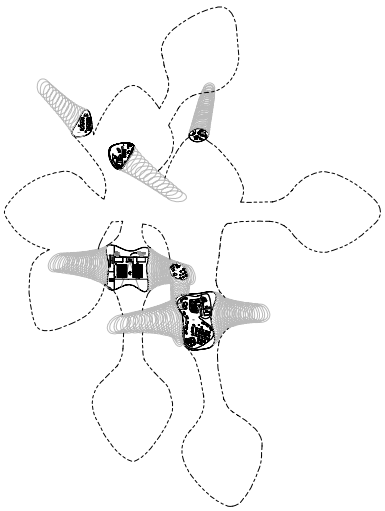
+10.00



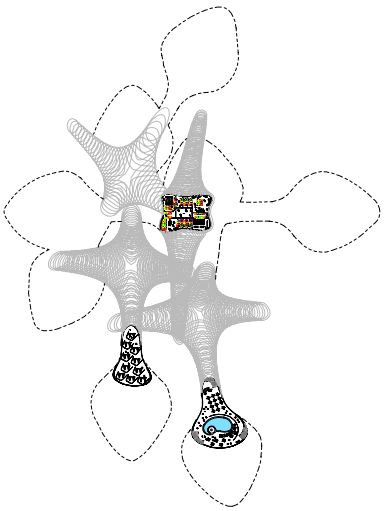
+25.00



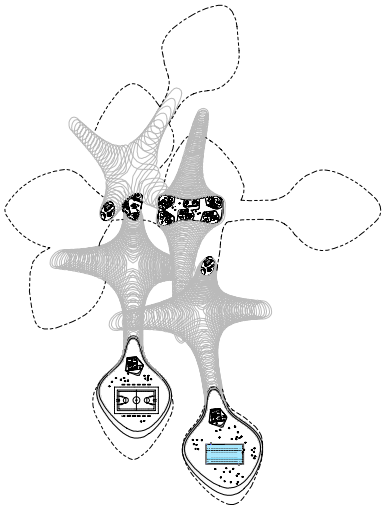
+70.00



+135.00



+145.00

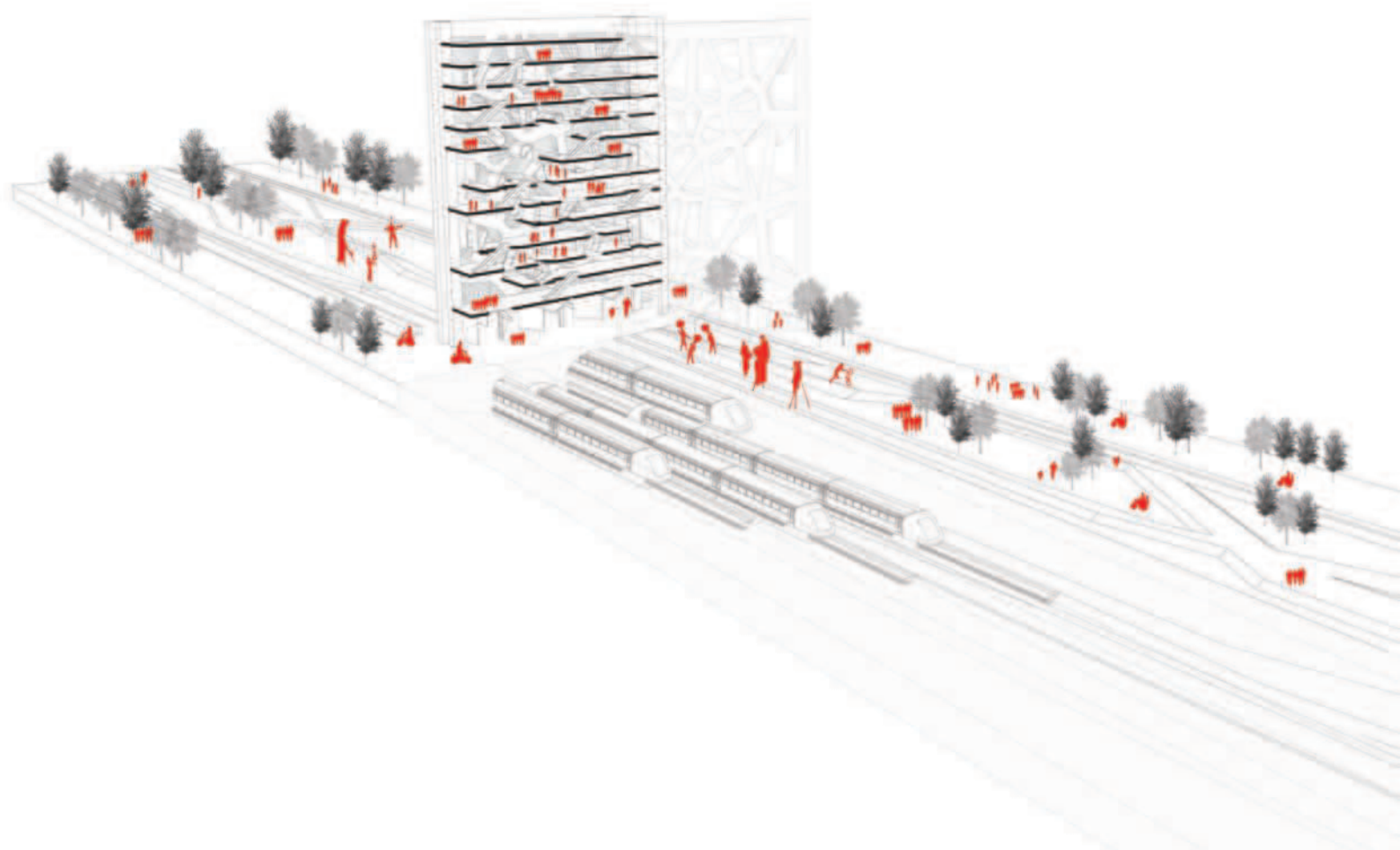


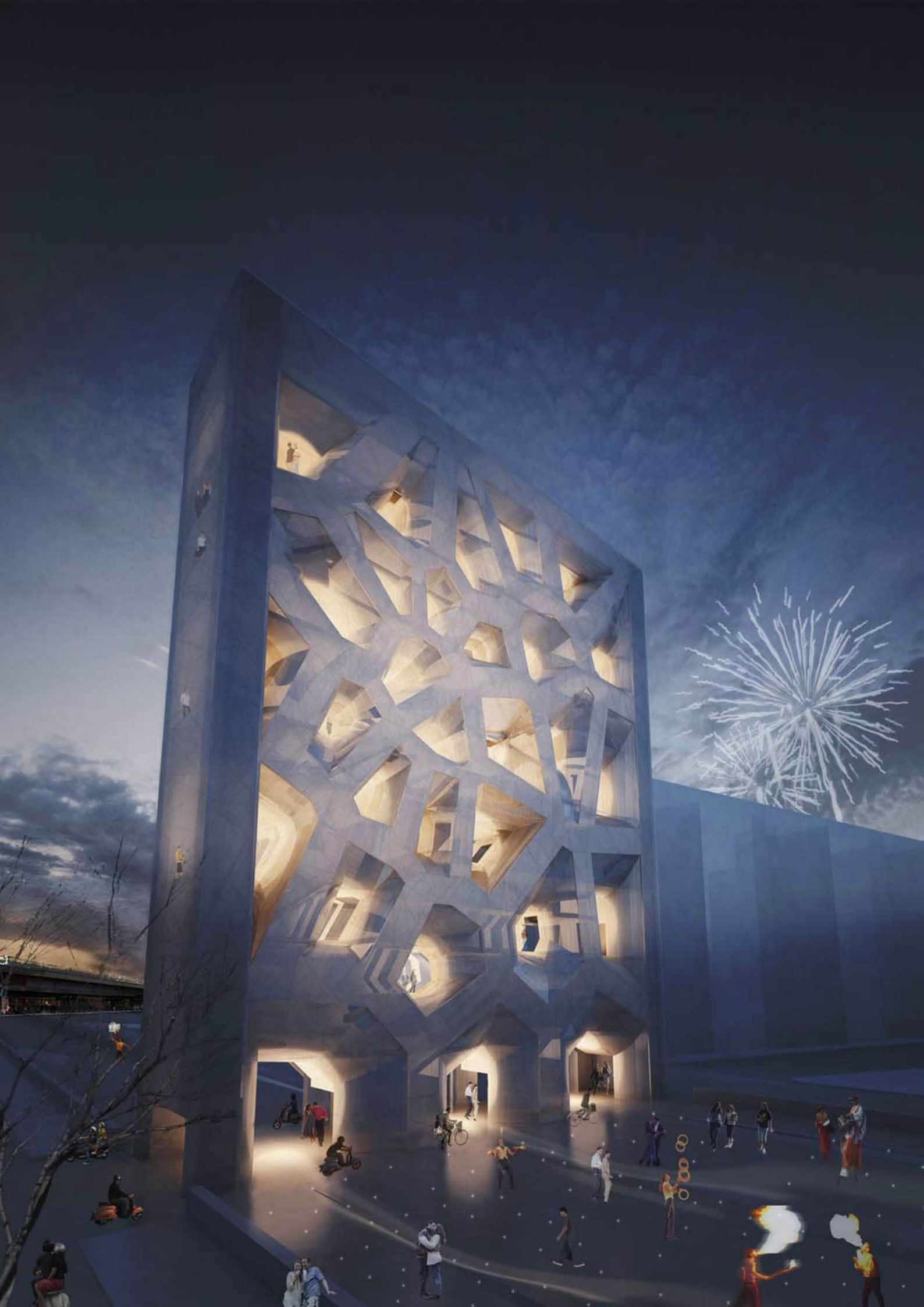
```
Option Explicit

CirclesRnd

Sub CirclesRnd()

Dim x,y,R, arrPoint, UB,LB, R1,R2,arrPlane
Dim arrCircle, strCircle, intIndex, arrPObj

For x=0 To 100 Step 2

  For y=0 To 10 Step 2

   UB=3*x/30 : LB=-3*x/30 : R=Rnd*x/20
   R1=(UB-LB+1)*Rnd+LB : R2=(UB-LB+1)*Rnd+LB

   arrPoint = Array (x+R1,y+R2,R1)
   arrPlane = Array (arrPoint, Array(1,0,0), Array(0,1,0), Array(0,0,1) )

   If Not(R=0) Then
     strCircle = Rhino.AddCircle (arrPlane, R)
     arrCircle=Array(strCircle)
     arrPObj = Rhino.AddPlanarSrf ( arrCircle )
     intIndex = Rhino.ObjectMaterialIndex(arrPObj(0))
     intIndex = Rhino.addMaterialToObject(arrPObj(0))
    If (intIndex > -1) Then
     Rhino.MaterialColor intIndex, Rhino.ColorHLSToRGB( Array((X+10)*1.5*Rnd,
125, 255) )
     Rhino.MaterialTransparency intIndex, 0.6
    End If
   End If
   Next
Next

End Sub
```

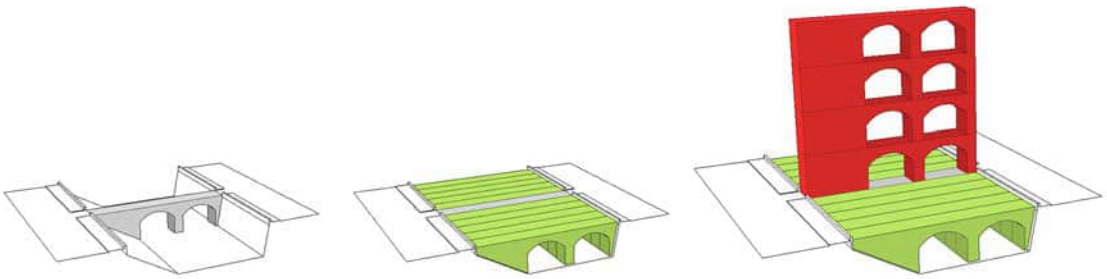
INSTANT BRIDGE

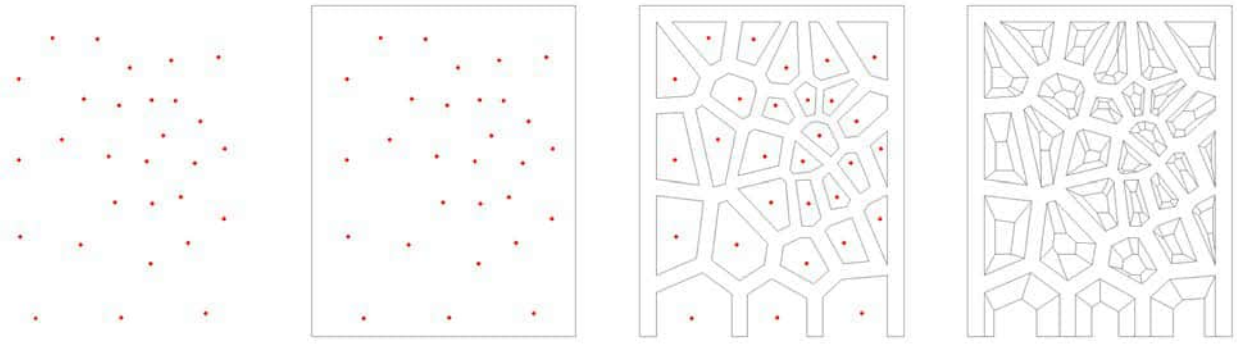
The majority of city bridges, in the past, were typically inhabited and they were perfectly blended with the surrounding housing style. This characteristic can be found in the historical Ponte Vecchio in Florence which survives unchanged until the present with its direct connection between the bridge's architecture and the river.
The "Instant Bridge" born from the idea of subverting the normal canons of a pedestrian bridge starting from a sociological analysis of the neighbourhood. It is an allegorical and a reflective transformation of a pedestrian bridge. It is something enjoyable for all the inhabitants with the goal of transmitting the metamorphosis of a brick structure into a cultural building.
The Pigneto district had a notable evolution and transformation from a working-class neighbourhood to a cultural and artistic one. Nevertheless there is a strong disjunction of these artistic scenes, relegated to small existing shops; these elements qualify the district as an artistic underdeveloped neighbourhood. Inside there are several functions distributed over 14 stories for a total height of 45 metres. These functions are reached by glass lifts located at the extreme sides of the building and across an internal system of escalators.
Every level is planned for a different purpose with materials and colours which accentuate the spaces. The materials used are low cost and recyclable (e.g. non toxic paint).
The steel structure is produced using recycled rails procured by remittances of disused railways.
The main structure, the façade and the interior spatial elements merge into a single three-dimensional form. It's a very efficient and seismic proof structural system that requires 30% less steel than a structure consisting of pillars and beams. This orthotropic structure spends more than 15 meters.
The building is also largely day lit with carefully controlled natural ventilation systems.

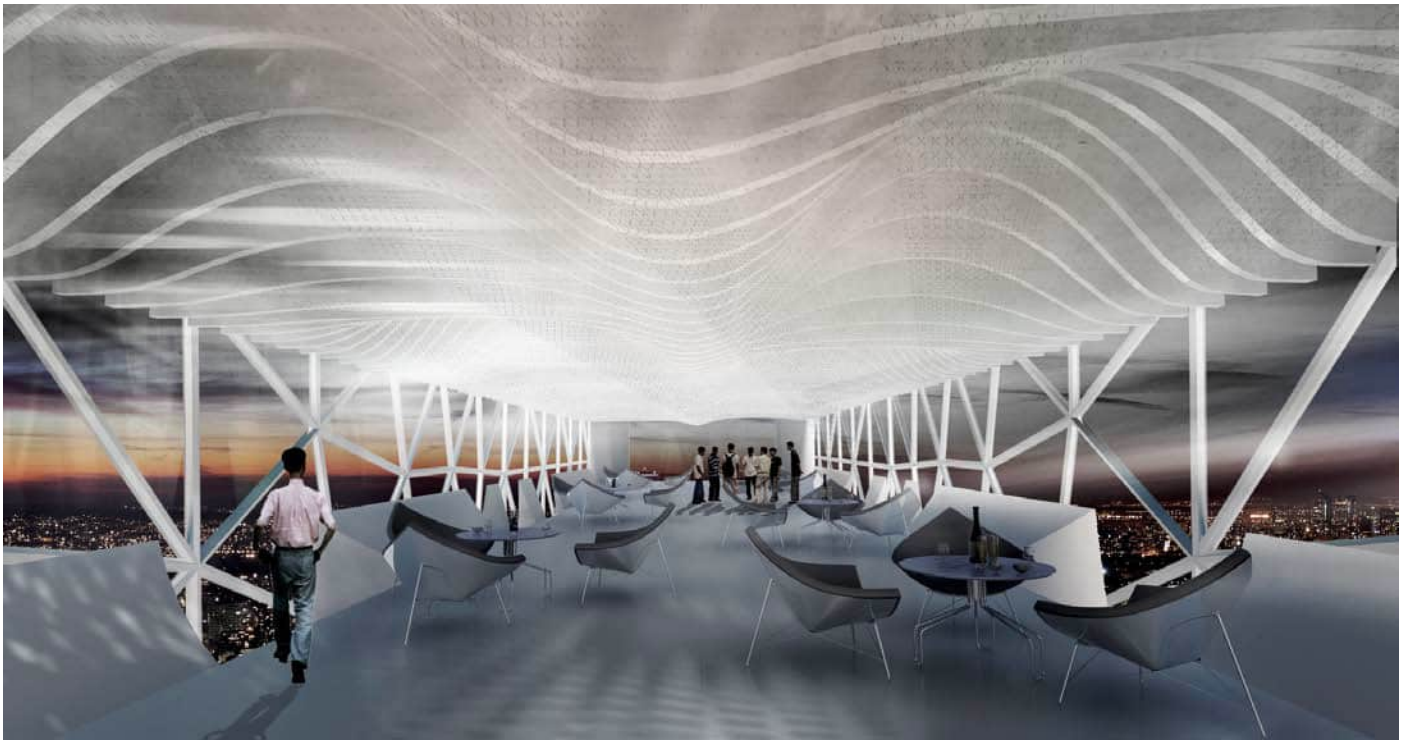
CREDITS
Plan's title: PontèPigneto_Instant Bridge
Location: Roma
Organizer: Pigmenti
Sqm: 2000
Cost: € 2 M
Materials: Steel (facades), Glass, Concrete
Project Design: Francesco Lipari
Project Team: Vanessa Todaro, Pierpaolo Pugliano
Timing: design 2010
Web information: www.francescolipari.it

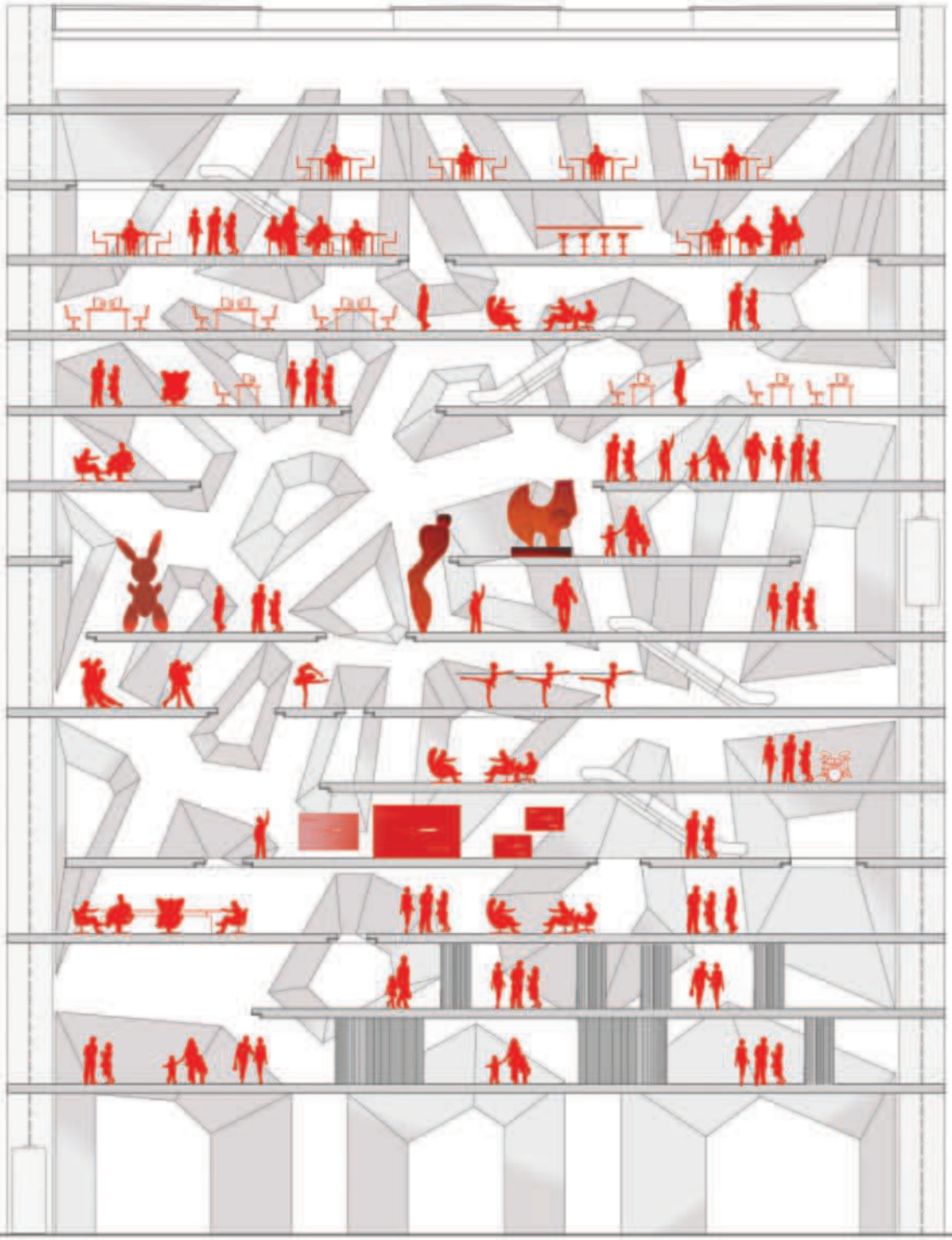


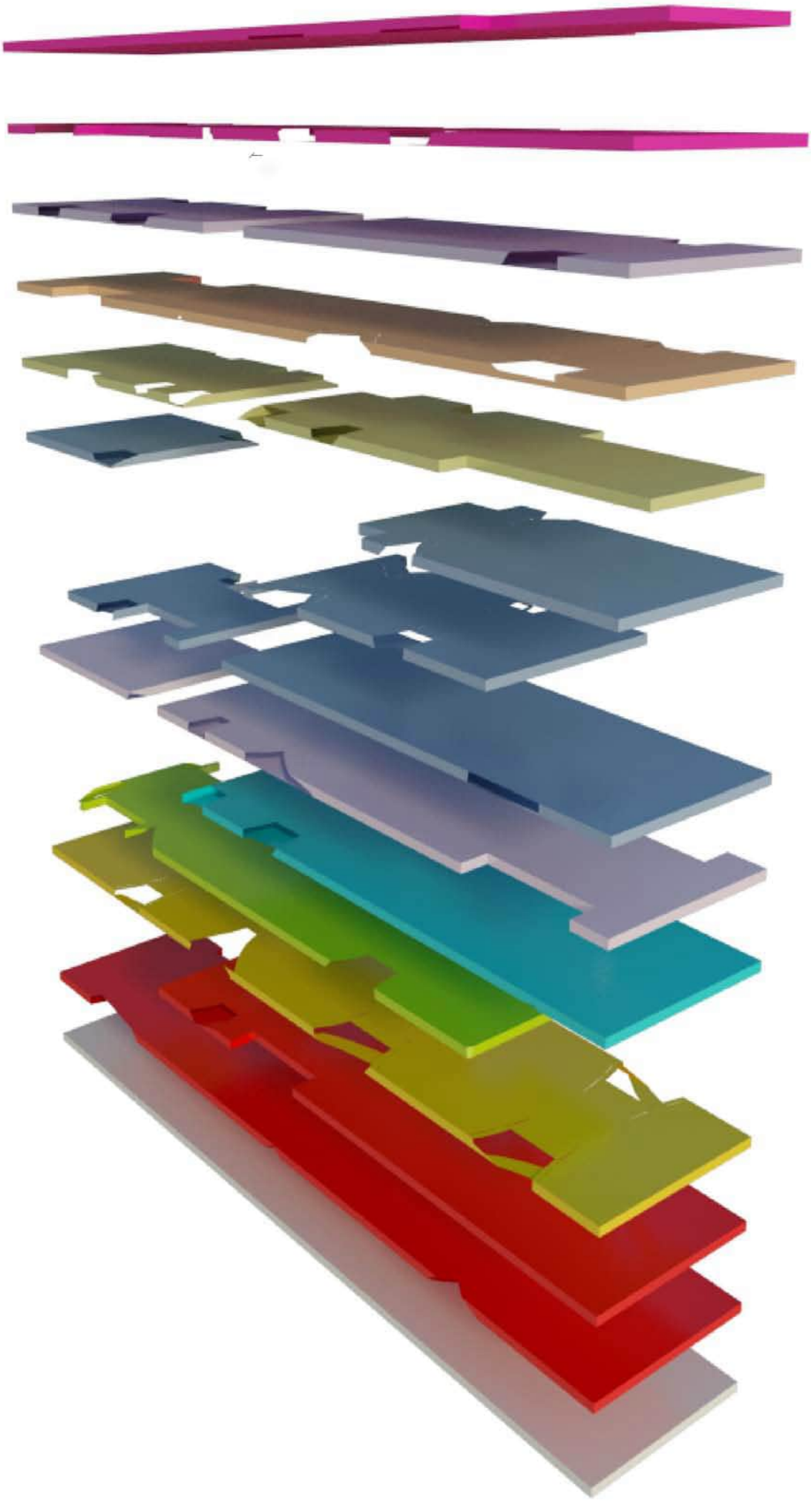









sky bar
restaurant
workshop
wi-fi point
cinemateque
EXHIBITION SPACE
EXHIBITION SPACE
EXHIBITION SPACE
dance workshop
music workshop
art workshop
experimental theater
art & crafts
art & crafts
ingresso e passaggio
ponte pedonale



SKY BAR

FUSION RESTAURANT

WORKSHOP

WIFI POINT

CINEMATEQUE

EXHIBITION

DANCE WORKSHOP

MUSIC WORKSHOP

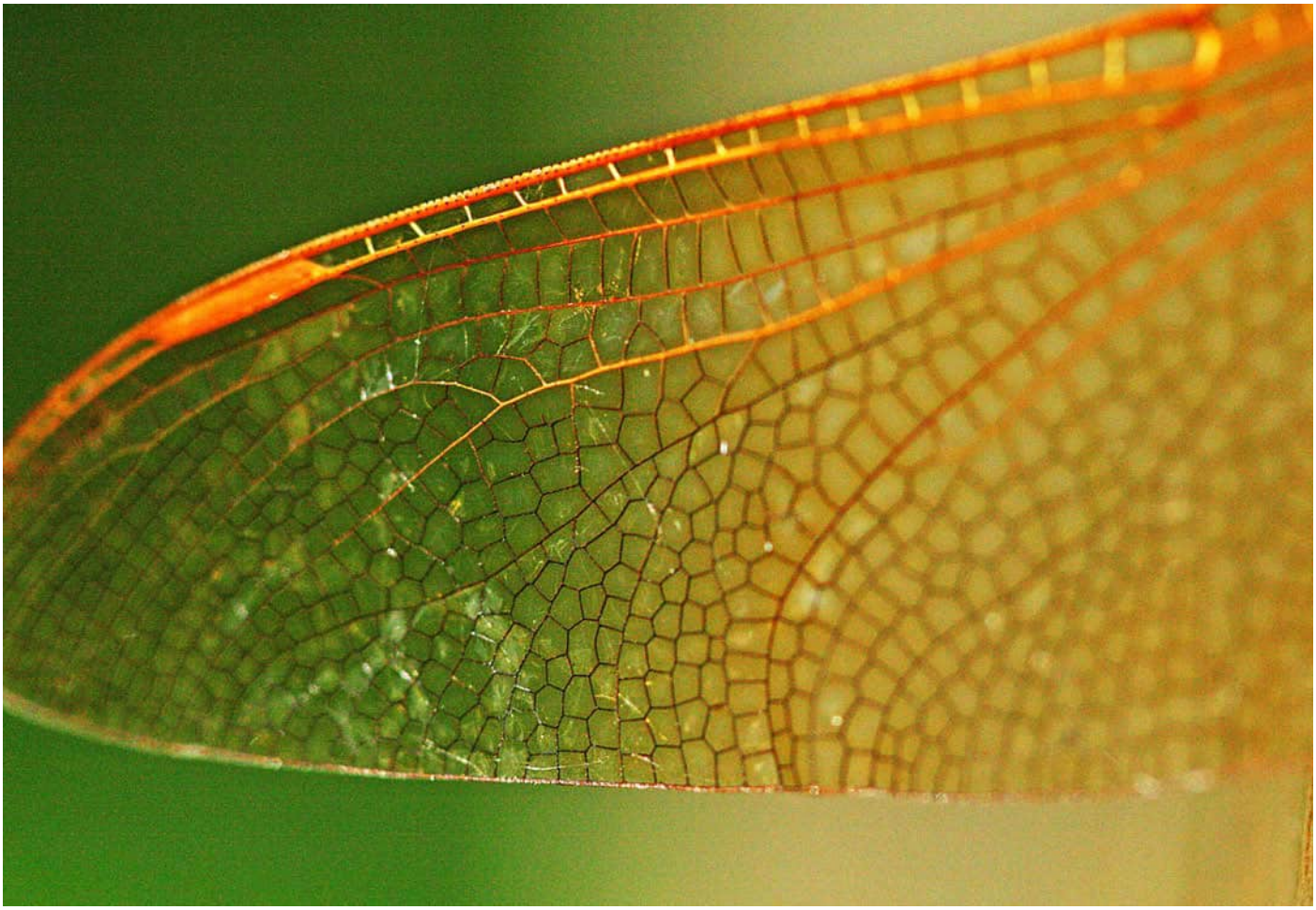
ART WORKSHOP

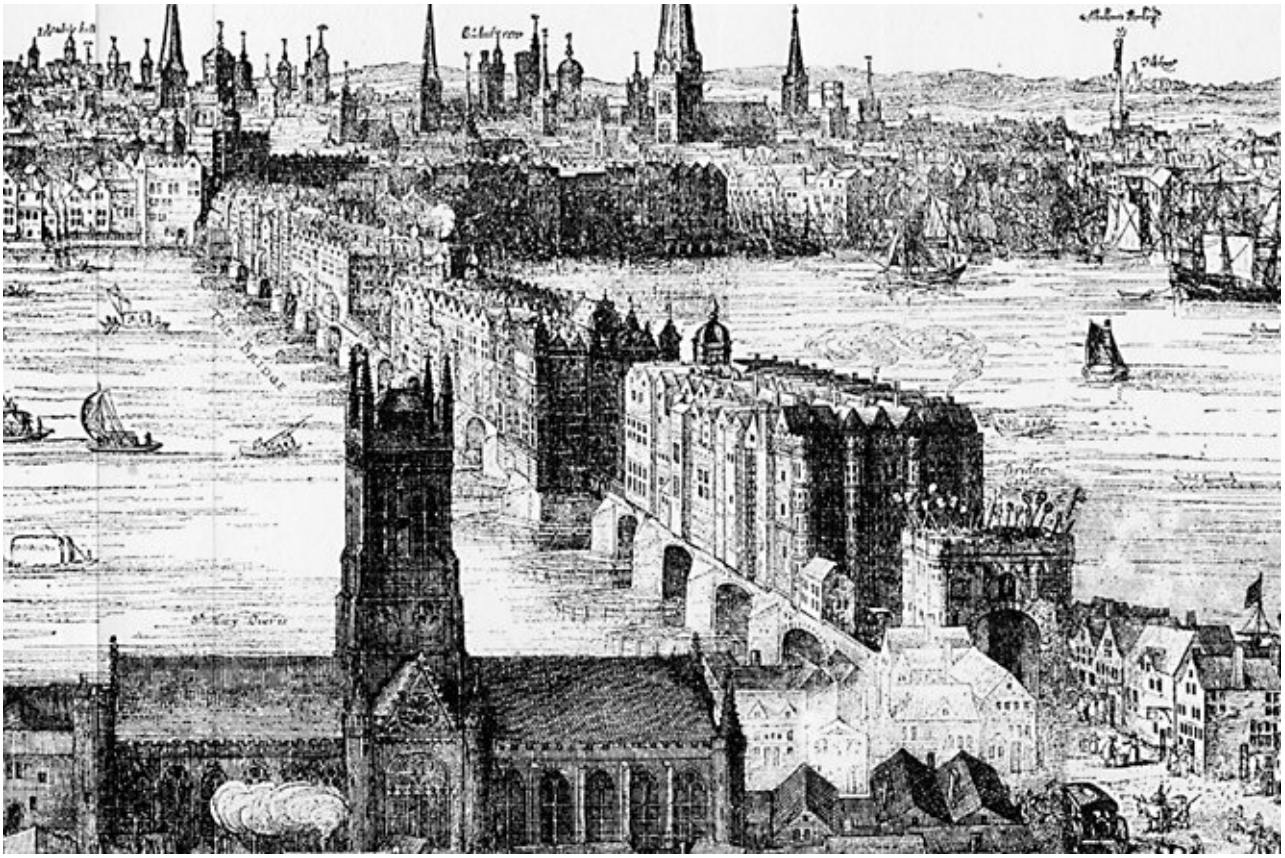
ARTS & CRAFTS

TEATRO SPERIMENTALE

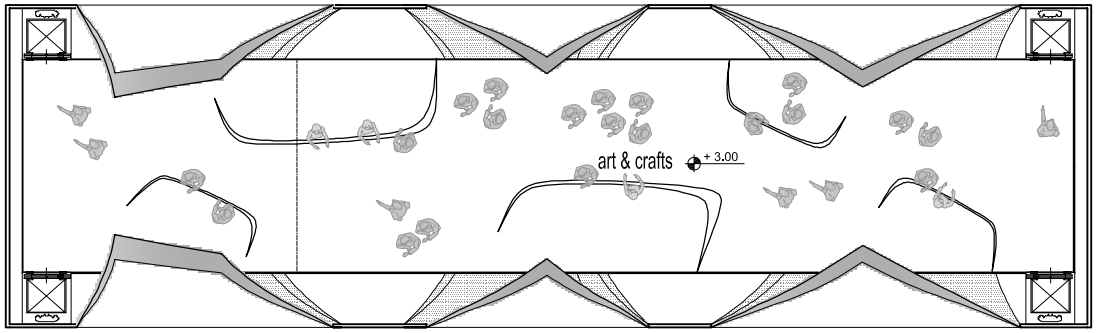
INGRESSO E PASSAGGIO
PONTE PEDONALE

SCHEMA DISTRIBUTIVO

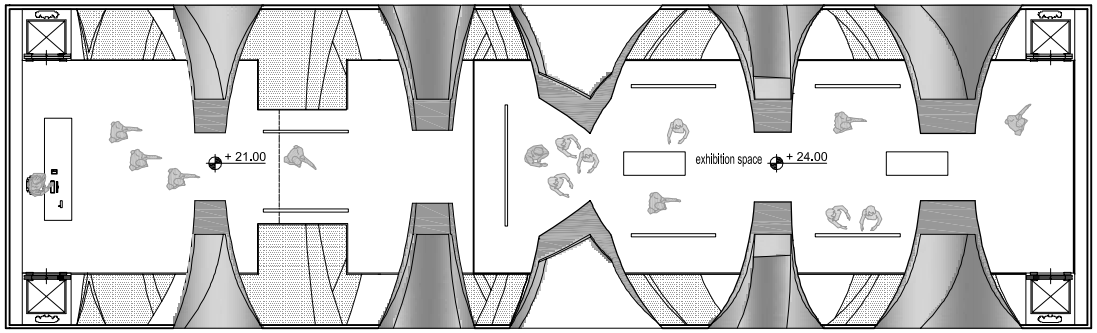




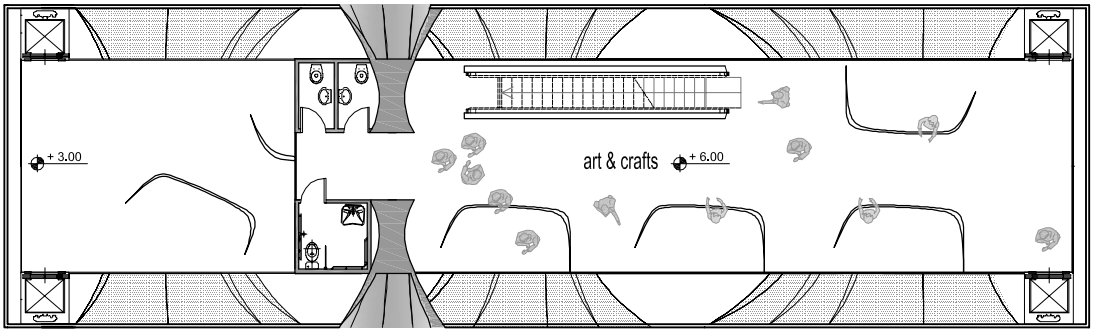
+3.00



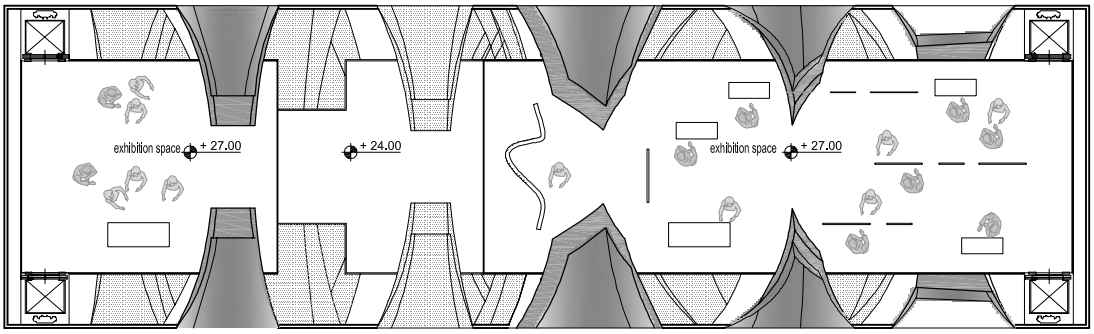
+24.00



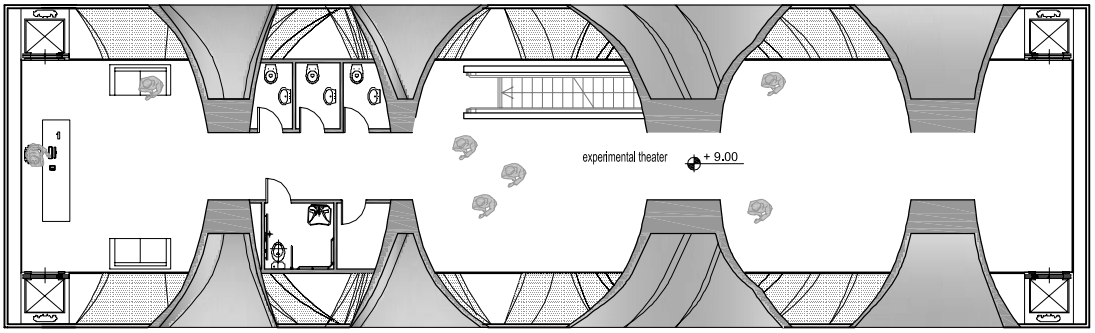
+6.00



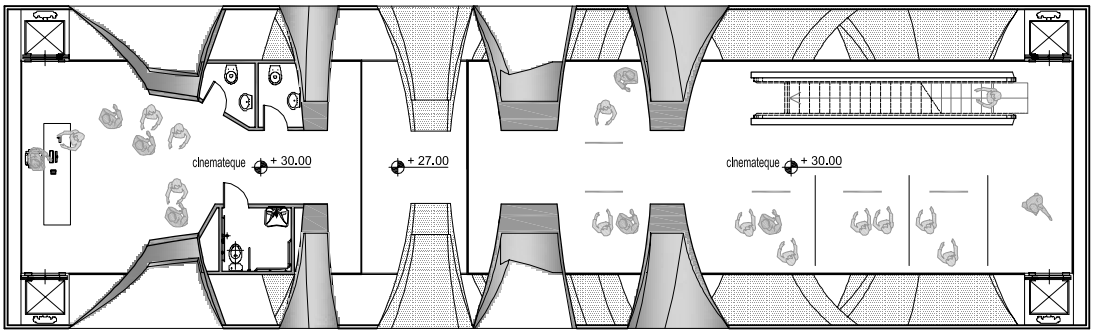
+27.00



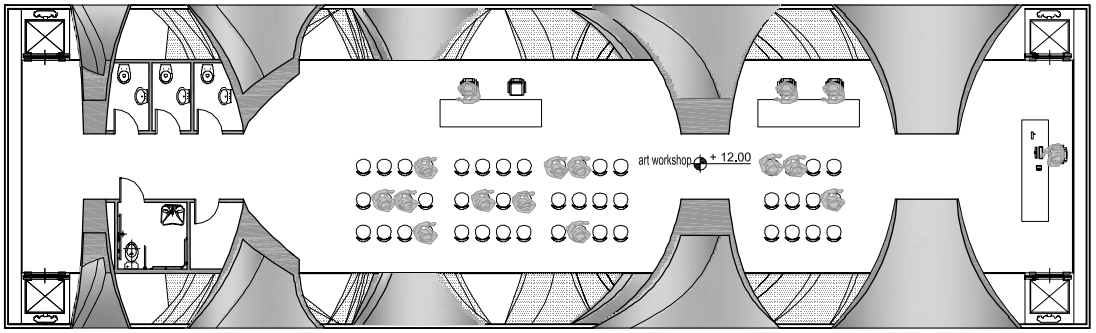
+9.00



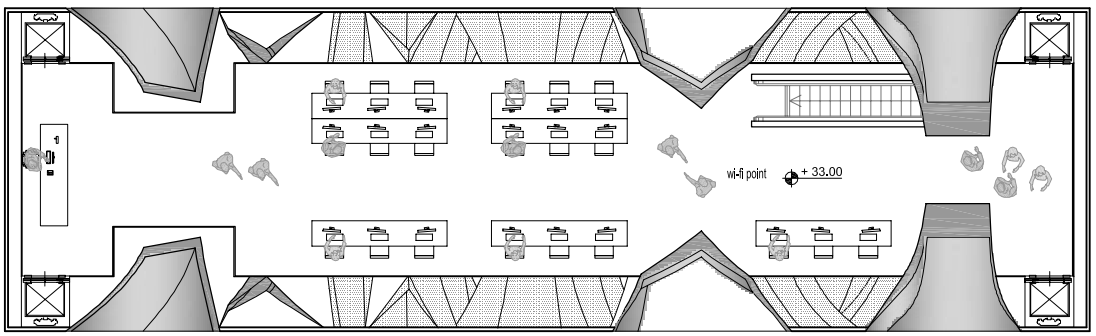
+30.00



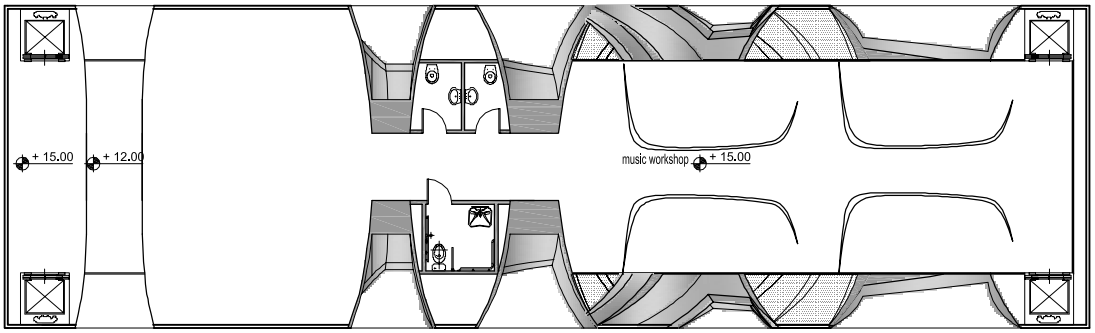
+12.00



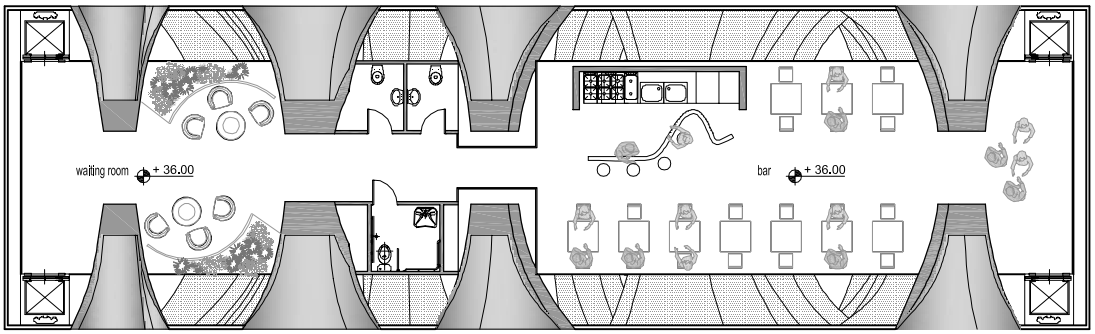
+33.00



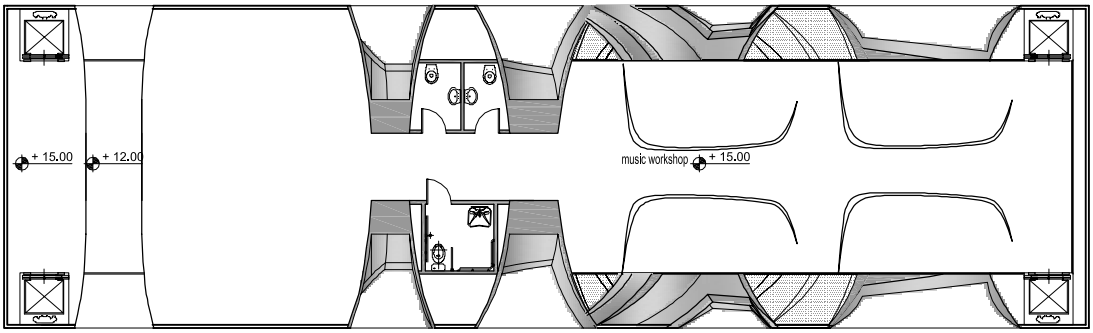
+15.00



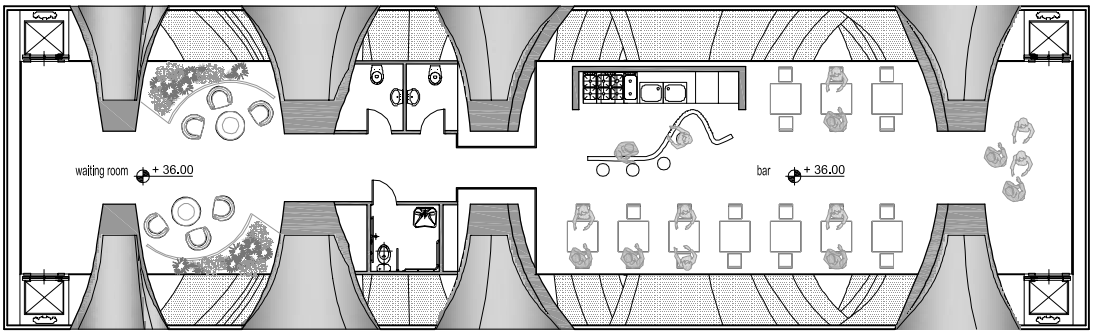
+36.00



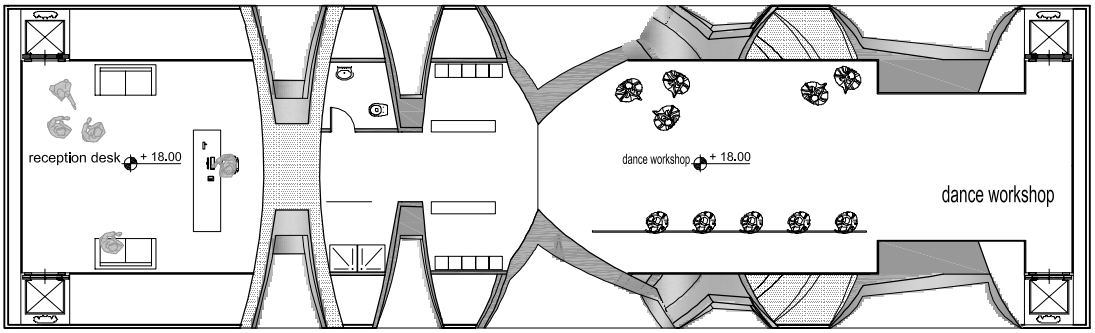
+15.00



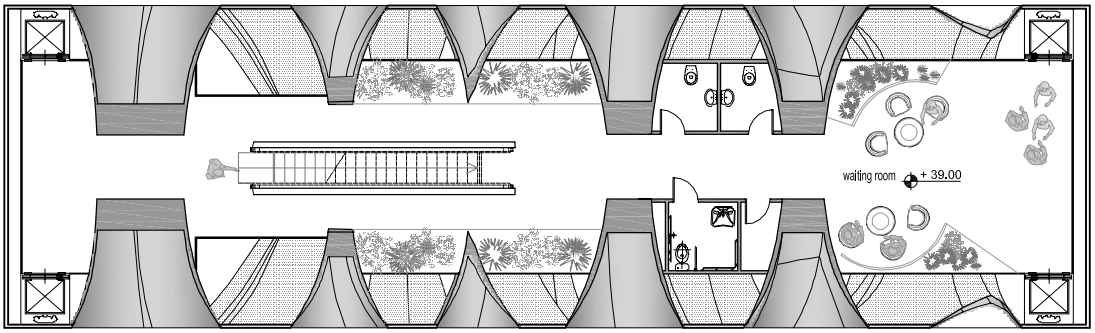
+36.00



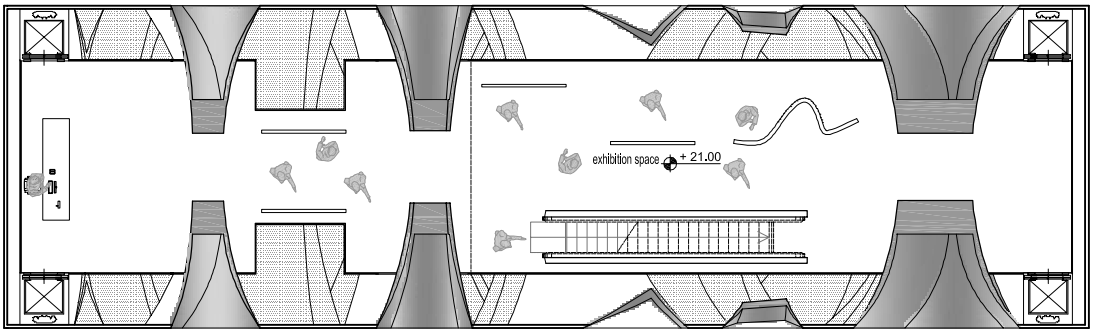
+18.00



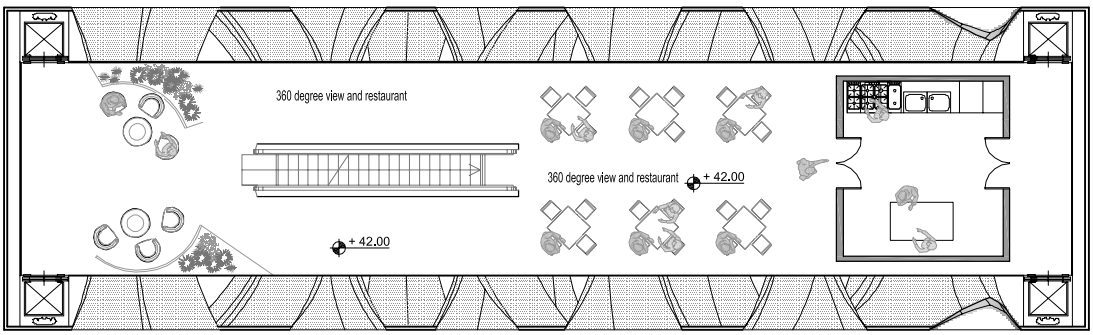
+39.00

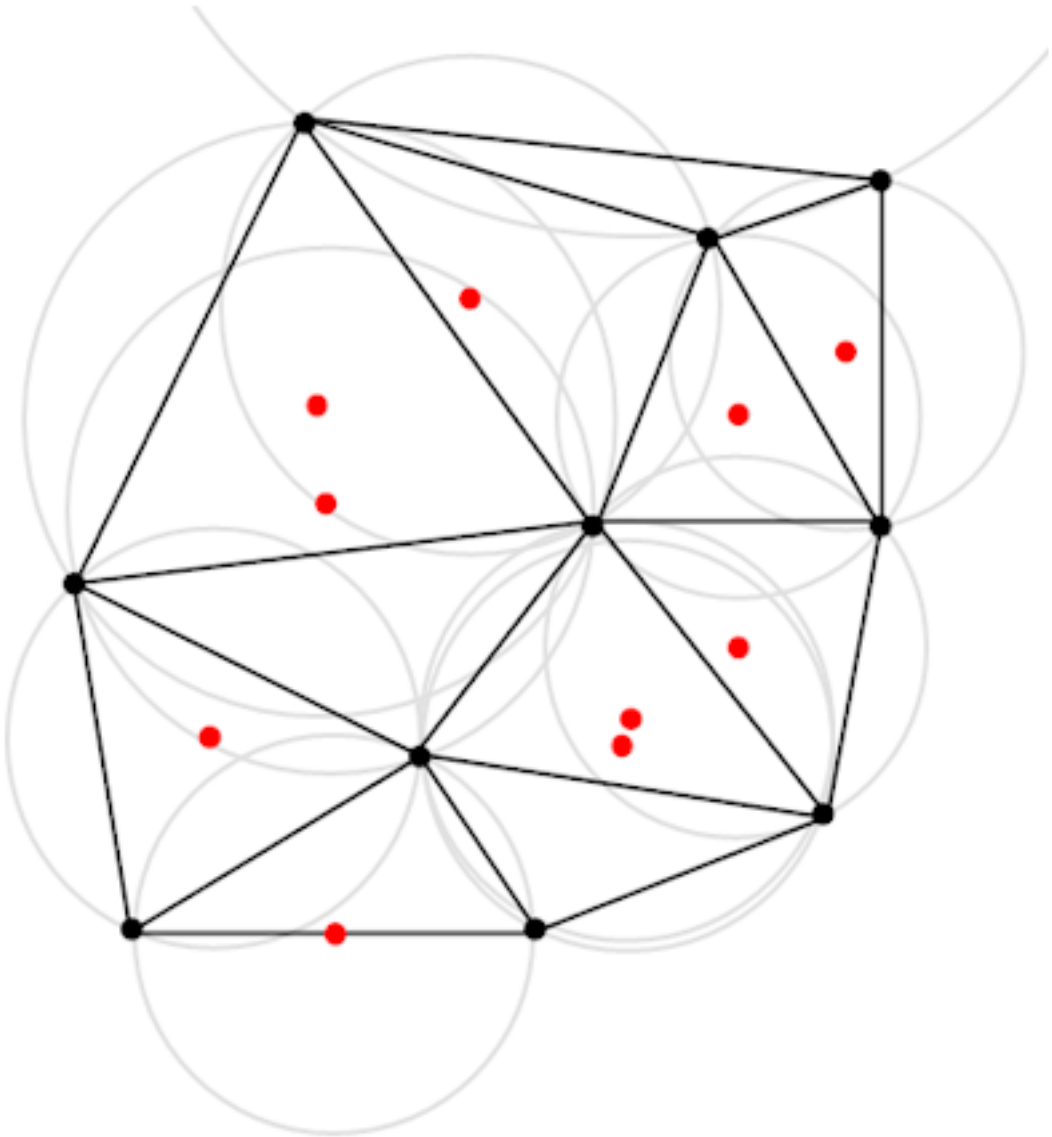


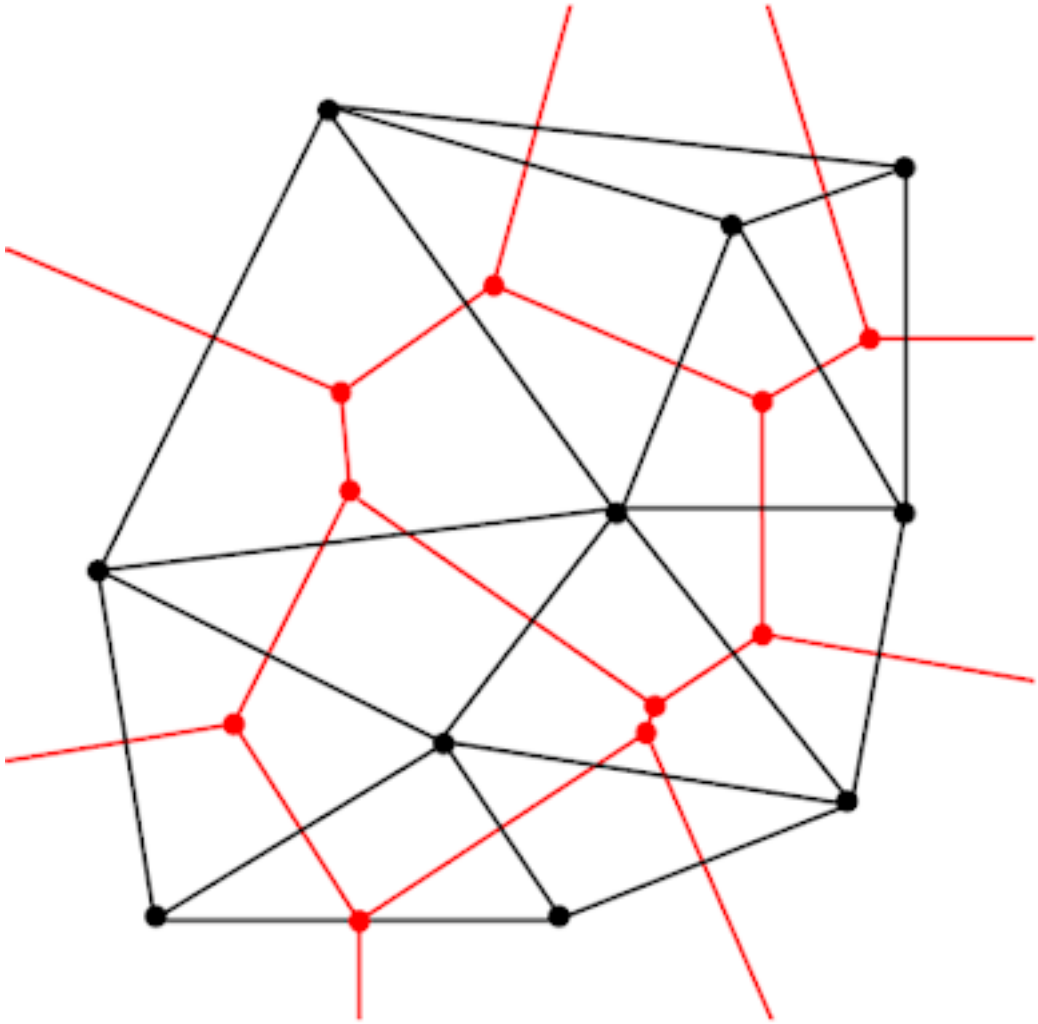
+21.00



+42.00







```
Option Explicit

' 3d Voronoi AKA Project Cell
' (c) Gabe Smedresman 2005
' All Rights Reserved.


"' global naming variables

Dim XX: XX = 0
Dim YY: YY = 1
Dim ZZ: ZZ = 2

GenerateVoronoiCells

Sub GenerateVoronoiCells()
  Randomize

        Dim arrBoundingVolume ' array of polysurfaces indicating bounding volume
        Dim arrPoints ' array of string references to voronoi points

        Dim i,j

  Dim startTime : startTime = Now

        ' select objects

  arrBoundingVolume = Rhino.GetObjects("Select a Bounding Volume",16+8,vbTrue,vbTrue)
  If IsNull(arrBoundingVolume) Then Exit Sub
  HideObjects arrBoundingVolume

        arrPoints = Rhino.GetObjects("Select Cell Points",1,vbFalse,vbFalse)

        ShowObjects arrBoundingVolume

        If IsNull(arrPoints) Then Exit Sub

        ' go through each point in the list, and create a cell, name it, color it, and update the progress report.

        Dim strCell, strName
        Rhino.Print "Beginning cell divisions: " & UBound(arrPoints)+1 & " cells total."
        For i = 0 To UBound(arrPoints) ' for each point to make a cell around  ' one point at a time

          strCell = GenerateCell(arrPoints(i),arrPoints,arrBoundingVolume)
          Dim value : value = Int(Rnd()*255)
          Rhino.objectColor strCell, RGB(255,255,value)
          strName = "Cell #" & i
          Rhino.objectname strCell, strName

    Dim strTime : strTime = GetTimeDescription(startTime, (i+1) * 1.0 / (UBound(arrPoints)+1) )

          Rhino.Print i+1 & " of " & UBound(arrPoints)+1 & " cells (" & Int((i+1) * 100 / (UBound(arrPoints)+1)) & "%) completed. " & strTime

        Next 'i

        ' hide source geometry to reveal the cells
        HideObjects arrPoints
        HideObjects arrBoundingVolume


End Sub
```

```
" ---------------------------------------------------------------------
" given the center, an array of points, and the bounding volume,
" perform the operation, make sure all intermediate geometries
" have been deleted, then return the voronoi cell.
" ---------------------------------------------------------------------
Function GenerateCell(centerPoint,arrPoints,arrBoundingVolume)
  Dim arrBlocks
            arrBlocks = CreateBlocks(centerPoint, arrPoints)

            Dim strCell
  strCell = IntersectBlocks(arrBlocks,arrBoundingVolume)

  Dim m: For m = 0 To UBound(arrBlocks)
    If IsPolySurface(arrBlocks(m)) Then DeleteObject(arrBlocks(m))
  Next

  GenerateCell = strCell
End Function


" ---------------------------------------------------------------------
" using the array of point strings, and one point reference for the center
" create a group of blocks, which, when intersected, will result in the
" voronoi volume.
"
" this is done by, for each point, generating a plane perpendicular to the
" line between the center and the test point, at the midpoint of that line.
" this plane faces the center point and is equidistant from the two points
" across the entire surface.
"
" then extrude the plane towards the center point, theoretically an infinite
" amount but really to the end of the point cloud. meaning, this extrusion
" is closer to the center than it is to the test point.
"
" intersect all of these volumes and you will have the area closest
" to the center.
" ---------------------------------------------------------------------
Function CreateBlocks(centerPoint, arrPoints)

  Dim arrBlocks
  ReDim arrBlocks(UBound(arrPoints))

            Dim newPlane
            Dim numBlocks : numBlocks = 0
            Dim i
            Dim midpoint, normal

            Dim greatestdiagonalspread
            greatestdiagonalspread = FindGreatestDiagonalSpread(arrPoints)

            Dim centercoords,pointcoords
  centercoords = Rhino.PointCoordinates(centerPoint)

            ' ---- CREATE BOUNDING PLANES
            For i = 0 To UBound(arrPoints) ' for each point to consider boundary to

    If Not centerPoint = arrPoints(i) Then ' as long as the point isn't the central point

      pointcoords = Rhino.PointCoordinates(arrPoints(i))
      Dim strPlane
      strPlane = CreateBisectingPlane(centercoords, pointcoords, greatestdiagonalspread)

      If(Rhino.IsSurface(strPlane)) Then 'if it's a valid object, add to the array
```

```
            'newPlane has been created

            midpoint = VectorMidpoint(centercoords, pointcoords)

            normal = VectorUnitize(VectorSubtract(centercoords, pointcoords))

            'strPath is now created

            Dim strPath
            strPath = Rhino.AddLine(midpoint, VectorAdd(midpoint,VectorScale(normal, greatestdiagonalspread)))

            Dim strExtrusion
            strExtrusion = Rhino.ExtrudeSurface( strPlane, strPath )
            Rhino.objectColor strExtrusion, RGB(128,128,128)

      arrBlocks(numBlocks) = strExtrusion
            numBlocks = numBlocks + 1

            If(Rhino.IsObject(strPath)) Then Rhino.DeleteObject(strPath)
            If(Rhino.IsObject(strPlane)) Then Rhino.DeleteObject(strPlane)
          End If 'end if it's a surface
        End If 'end if considering different points
      Next 'i

      ReDim Preserve arrBlocks(numBlocks-1)
      'Rhino.Print(numBlocks & " blocks created.")

      CreateBlocks = arrBlocks
End Function



" ---------------------------------------------------------------------
" Takes two point coordinate arrays, finds the midpoint
" makes a coordinate system based on the line between these two points
" and uses this coordinate system to make the plane that marks the 3d bisector
" of that line.
" IE this plane is the halfway boundary betweensc the two points
" ---------------------------------------------------------------------

Function CreateBisectingPlane(arrPtOne, arrPtTwo, reach)
  Dim i
  Dim center
  center = VectorMidpoint(arrPtOne, arrPtTwo)

  ' make new coordinate system p,r,s: p is the line between 1 and 2, r is to the side, s is up(ish) from the line
  Dim p :  p  = VectorSubtract(arrPtTwo,arrPtOne) : p = VectorUnitize(p)
  Dim up : up = Array(0,0,1)
  Dim r : r  = VectorCrossProduct(p,up) : If IsVectorZero(r) Then r = Array(0,1,0)
  r = VectorUnitize(r) " points to the right
  Dim s :  s  = VectorCrossProduct(p,r) : s = VectorUnitize(s) " points to perpendicular to p (forward) and r (side)

  ' now find four points, 1 up 2 left 3 down 4 right (looking from one to two)
  Dim arrCorners(3)
  arrCorners(0) = VectorAdd(center,VectorScale(s,reach))
  arrCorners(1) = VectorAdd(center,VectorScale(r,reach * -1))
  arrCorners(2) = VectorAdd(center,VectorScale(s,reach * -1))
  arrCorners(3) = VectorAdd(center,VectorScale(r,reach))

  Dim strPlane
  strPlane = Rhino.AddSrfPt( arrCorners )

  CreateBisectingPlane = strPlane
End Function
```

```
" ---------------------------------------------------------------------
" given the array of generated blocks, perform a boolean intersection
" with the bounding volume and each block, one by one.
" delete each source as you iterate. some blocks remain: if
" the intersection fails (no areas intersect), the sources
" will not be deleted.
" ---------------------------------------------------------------------
Function IntersectBlocks(arrBlocks, arrBoundingVolume)
  Dim i

  IntersectBlocks = Null
  i = 0
  Dim results, pendingresults
  Dim strBlock
  Dim group2

  Do
    strBlock = arrBlocks(i)
    group2 = Array(strBlock)
    pendingresults = Rhino.BooleanIntersection(arrBoundingVolume,group2,vbFalse)
    i = i + 1
  Loop While Not IsArray(pendingresults)

  results = pendingresults

  Dim j
  For j = i To UBound(arrBlocks)
    strBlock = arrBlocks(j)
    group2 = Array(strBlock)
    pendingresults = Rhino.BooleanIntersection(results,group2)
    If IsArray(pendingresults) Then results = pendingresults

  Next

  IntersectBlocks = results(0)
End Function


" ---------------------------------------------------------------------
" find the maximum 3d diagonal length between one extreme corner
" of a point cloud and the other
" ---------------------------------------------------------------------
Function FindGreatestDiagonalSpread(arrPoints)
  If(Not IsArray(arrPoints)) Then FindGreatestDiagonalSpread = 0

  Dim max : max = Rhino.PointCoordinates(arrPoints(0))
  Dim min : min = Rhino.PointCoordinates(arrPoints(0))
  Dim i, pt, spread

  For i = 0 To UBound(arrPoints)
    pt = Rhino.PointCoordinates(arrPoints(i))
    If(max(XX) < pt(XX)) Then max(XX) = pt(XX)
    If(max(YY) < pt(YY)) Then max(YY) = pt(YY)
    If(max(ZZ) < pt(ZZ)) Then max(ZZ) = pt(ZZ)
    If(min(XX) > pt(XX)) Then min(XX) = pt(XX)
    If(min(YY) > pt(YY)) Then min(YY) = pt(YY)
    If(min(ZZ) > pt(ZZ)) Then min(ZZ) = pt(ZZ)
  Next 'i

  FindGreatestDiagonalSpread = VectorLength(VectorSubtract(max,min))
End Function
```

```
" ---------------------------------------------------------------------
" given the starting time and fraction complete, estimate time to
" completion and then generate a sentence of the format
" "3 minutes 4 seconds elapsed: should be finished at 04:00 PM today."
" ---------------------------------------------------------------------
Function GetTimeDescription(startTime, fractioncomplete)
        Dim strDescription
        strDescription = ""
          Dim elapsedseconds : elapsedseconds = DateDiff("s",startTime,Now)
          Dim m,h,s : s = elapsedseconds
          m = Int(s/60) : s = s - m * 60
          h = Int(m/60) : m = m - h * 60
          If h = 1 Then strDescription = strDescription & " " & h & " hour"
          If h > 1 Then strDescription = strDescription & " " & h & " hours"
          If m = 1 Then strDescription = strDescription & " " & m & " minute"
          If m > 1 Then strDescription = strDescription & " " & m & " minutes"
          If s = 1 Then strDescription = strDescription & " " & s & " second"
          If s > 1 Then strDescription = strDescription & " " & s & " seconds"
          strDescription = strDescription & " elapsed:"

          Dim secondstogo : secondstogo = elapsedseconds / fractioncomplete * (1-fractioncomplete)
          Dim ETA : ETA = DateAdd("s",secondstogo,Now)

          If(fractioncomplete < 1) Then

            strDescription = strDescription & " should be finished at " & FormatDateTime(ETA,4)
            If(DatePart("d",ETA) = DatePart("d",Now)) Then
            strDescription = strDescription & " today."
            Else
            strDescription = strDescription & " in " & DateDiff("d",Now,ETA) & " day(s)."
            End If
          Else
            strDescription = strDescription & " finished " & Now & "!"
          End If

          GetTimeDescription = strDescription

End Function



''' ************************************************************************
''' Rhinoscript Vector Functions
''' ************************************************************************

''' ---------------------------------------------------------------------
''' Make a vector from two 3D points
''' ---------------------------------------------------------------------
Public Function VectorCreate(p1, p2)
  VectorCreate = Array(p2(0) - p1(0), p2(1) - p1(1), p2(2) - p1(2))
End Function

''' ---------------------------------------------------------------------
''' Unitize a 3D vector
''' ---------------------------------------------------------------------
Public Function VectorUnitize(v)

  VectorUnitize = Null
  Dim dist, x, y, z, x2, y2, z2

  x = v(XX) : y = v(YY) : z = v(ZZ)
  x2 = x * x : y2 = y * y : z2 = z * z
```

```
  dist = x2 + y2 + z2
  If dist <= 0.0 Then Exit Function
  dist = Sqr(dist)

  x = x / dist
  y = y / dist
  z = z / dist

  VectorUnitize = Array(x, y, z)
End Function

''' ------------------------------------------------------------------------
''' Return the length of a 3D vector
''' ------------------------------------------------------------------------
Public Function VectorLength(v)

  VectorLength = Null

  Dim dist, x, y, z, x2, y2, z2
  x = v(XX) : y = v(YY) : z = v(ZZ)
  x2 = x * x : y2 = y * y : z2 = z * z
  dist = x2 + y2 + z2

  VectorLength = Sqr(dist)

End Function

''' ------------------------------------------------------------------------
''' Return the dot product of two 3D vectors
''' ------------------------------------------------------------------------
Public Function VectorDotProduct(v1, v2)
  VectorDotProduct = v1(XX) * v2(XX) + v1(YY) * v2(YY) + v1(ZZ) * v2(ZZ)
End Function

''' ------------------------------------------------------------------------
''' Return the cross product of two 3D vectors
''' ------------------------------------------------------------------------
Public Function VectorCrossProduct(v1, v2)

  VectorCrossProduct = Null
  Dim x, y, z
  x = v1(YY) * v2(ZZ) - v1(ZZ) * v2(YY)
  y = v1(ZZ) * v2(XX) - v1(XX) * v2(ZZ)
  z = v1(XX) * v2(YY) - v1(YY) * v2(XX)
  VectorCrossProduct = Array(x, y, z)

End Function


''' ------------------------------------------------------------------------
''' Add two 3D vectors
''' ------------------------------------------------------------------------
Public Function VectorAdd(v1, v2)
  VectorAdd = Null
  VectorAdd = Array(v1(XX) + v2(XX), v1(YY) + v2(YY), v1(ZZ) + v2(ZZ))
End Function

''' ------------------------------------------------------------------------
''' Subtract two 3D vectors
''' ------------------------------------------------------------------------
Public Function VectorSubtract(v1, v2)
  VectorSubtract = Null
  VectorSubtract = Array(v1(XX) - v2(XX), v1(YY) - v2(YY), v1(ZZ) - v2(ZZ))
End Function
```

```
''' ------------------------------------------------------------------------
''' Multiply two 3D vectors
''' ------------------------------------------------------------------------
Public Function VectorMultiply(v1, v2)
  VectorMultiply = Null
  VectorMultiply = Array(v1(XX) * v2(XX), v1(YY) * v2(YY), v1(ZZ) * v2(ZZ))
End Function

''' ------------------------------------------------------------------------
''' Scale a 3D vectors by a value
''' ------------------------------------------------------------------------
Public Function VectorScale(v, d)
  VectorScale = Null
  VectorScale = Array(v(XX) * d, v(YY) * d, v(ZZ) * d)
End Function

''' ------------------------------------------------------------------------
''' Compare two 3D vectors for equality
''' ------------------------------------------------------------------------

Public Function VectorCompare(v1, v2)
  VectorCompare = vbFalse

  If v1(XX) = v2(XX) And v1(YY) = v2(YY) And v1(ZZ) = v2(ZZ) Then
    VectorCompare = vbTrue
  End If

End Function

''' ------------------------------------------------------------------------
''' Negate a 3D vector
''' ------------------------------------------------------------------------
Public Function VectorNegate(v)
  VectorNegate = Null
  VectorNegate = Array(-v(XX), -v(YY), -v(ZZ))
End Function

''' ------------------------------------------------------------------------
''' Tiny vector test
''' ------------------------------------------------------------------------

Public Function IsVectorTiny(v)

  IsVectorTiny = vbFalse
  Dim tol :  tol = 1.0e-12 ' ON_ZERO_TOLERANCE

  If (Abs(v(XX)) <= tol) And (Abs(v(YY)) <= tol) And (Abs(v(ZZ)) <= tol) Then
    IsVectorTiny = vbTrue
  End If

End Function

''' ------------------------------------------------------------------------
''' Zero vector test
''' ------------------------------------------------------------------------
Public Function IsVectorZero(v)

  IsVectorZero = vbFalse
  If (v(XX) = 0.0) And (v(YY) = 0.0) And (v(ZZ) = 0.0) Then IsVectorZero = vbTrue

End Function

''' My more specialized functions
```

CARNE EXHIBITION

CARN3 è una mostra d' architettura che racconta il percorso formativo dello studio OFL architecture.
CARN3 rivela le interiora e i muscoli dell'architettura, come un composto di pezzi triturati in una promiscuità d'arte.
Immagini visionarie e modelli d'architettura accompagnano lo spettatore in un percorso unico nel suo genere grazie ad un allestimento sterile che ne esalta le capacità evocative dei progetti.
La mostra indaga in maniera originale sulle nuove possibilità abitative strizzando un occhio alle nuove tendenze ecoenergetiche con lo scopo di sovvertirle per raggiungere nuovi limiti costruttivi.
Forme fluide e visionarie, poi, saranno lo stimolo per una riflessione sul futuro prossimo dell'essere umano.

luogo e data
23 dicembre 2009 - 30 dicembre 2009
ore 17.00 - 21.00

Biblioteca comunale di Campofranco (CL)
Piazza Vittorio Veneto

Inaugurazione mercoledi 23 dicembre ore 19.00
www.francescolipari.it/carn3

Proiezione del video JOCU realizzato da Emanuele Capponi per OFL_architecture

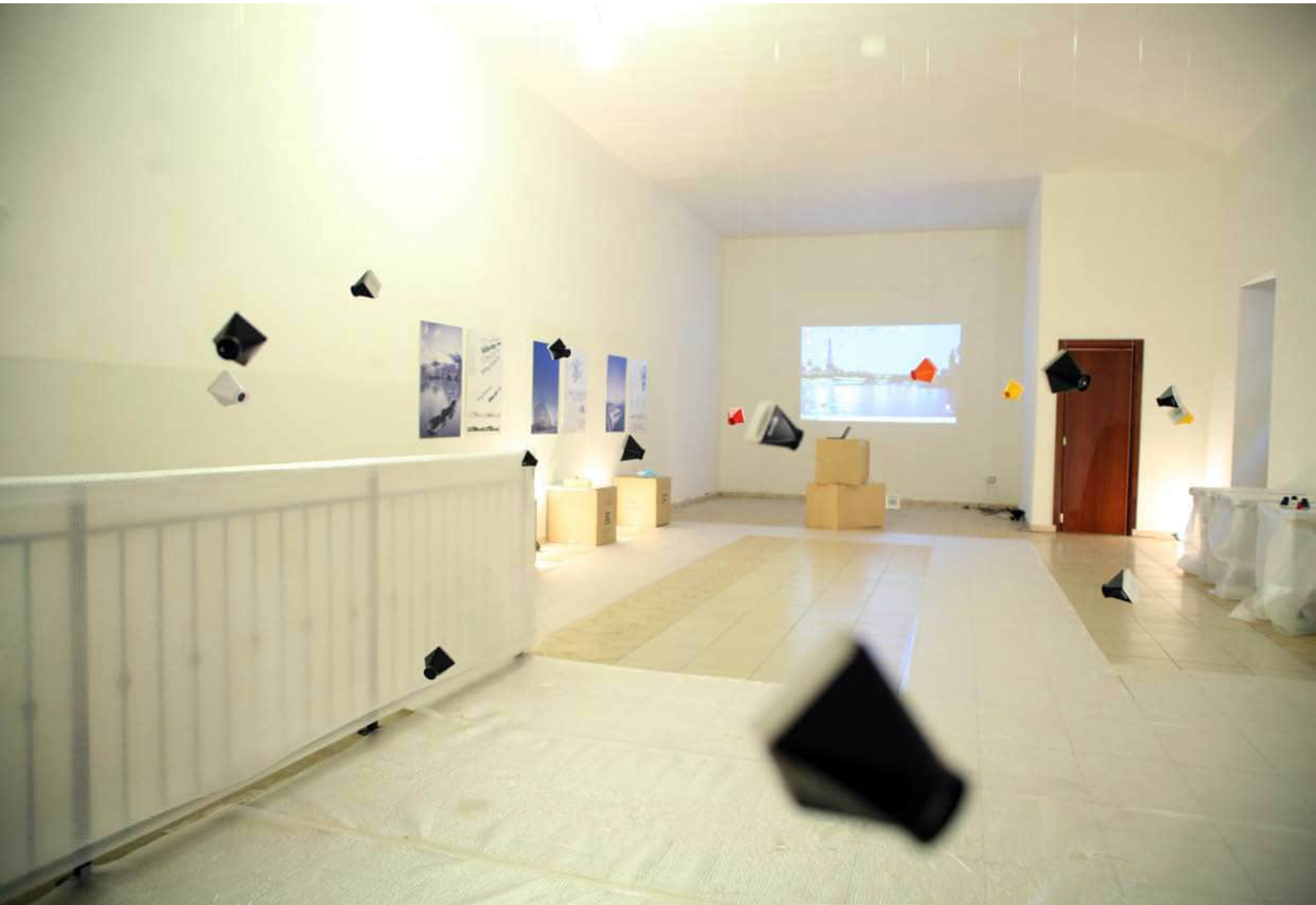
CREDITS
Curatori
Francesco Lipari, Vanessa Todaro

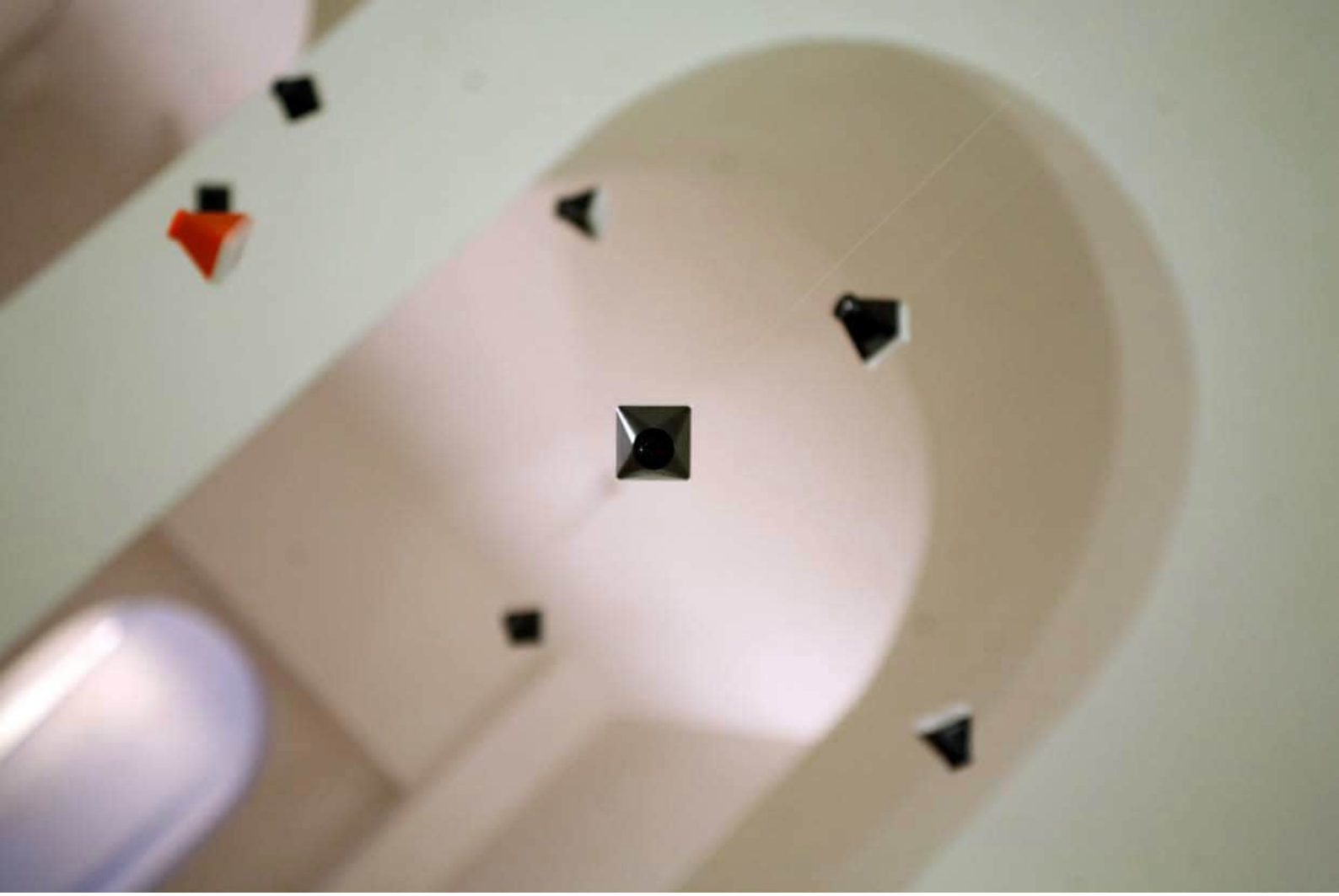
Modelli
Modelab, Vanessa Todaro

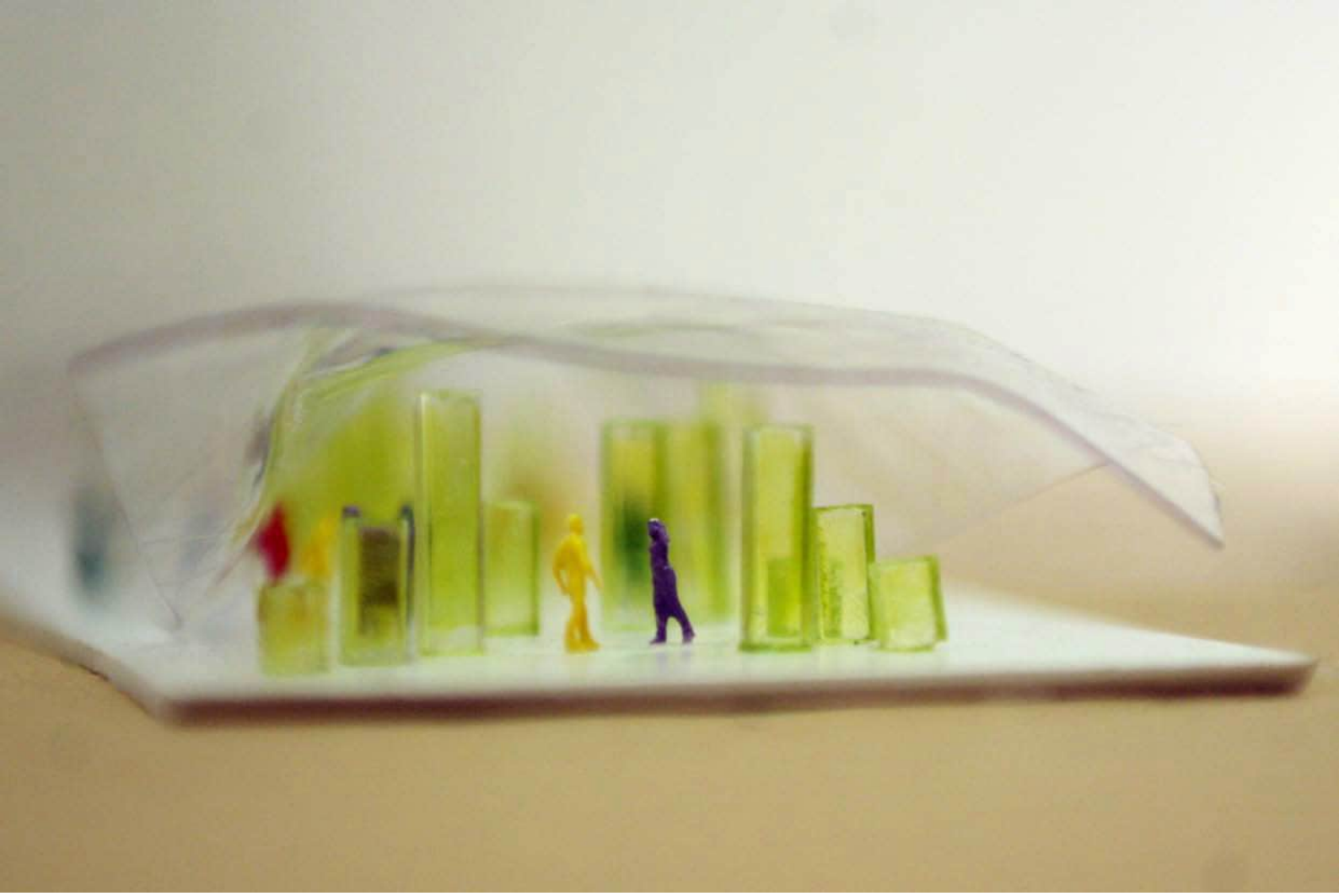
Comunicazione
Rossella Catalano, Enza Baldone, Anna Maria Di Gesù

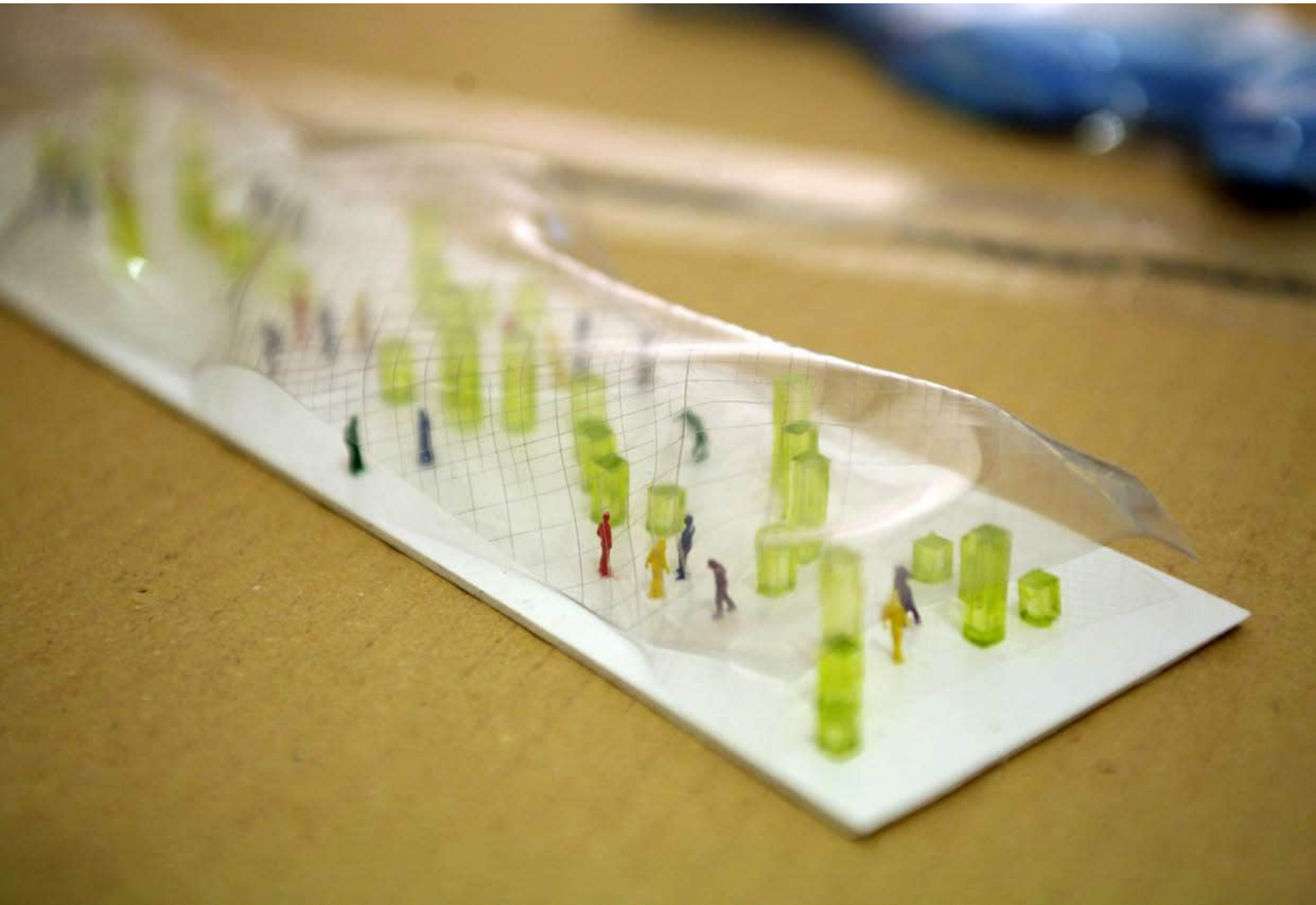
Realizzazione allestimento
Marco Modica, Alessio Lipari, Enrico Lipari

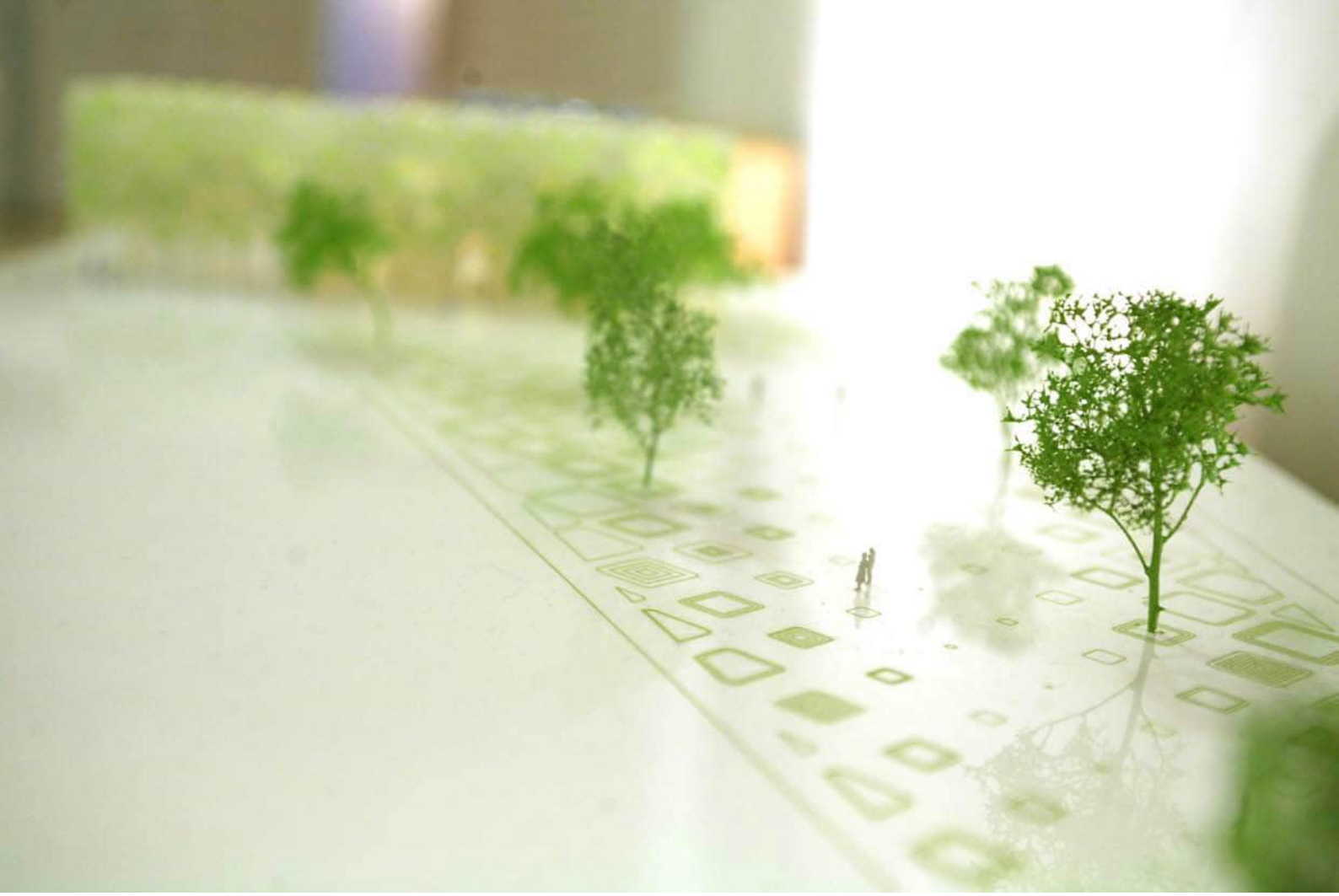
Coordinamento organizzativo
Vincenzo D'anna, Calogero Favata, Calogero Vitellaro

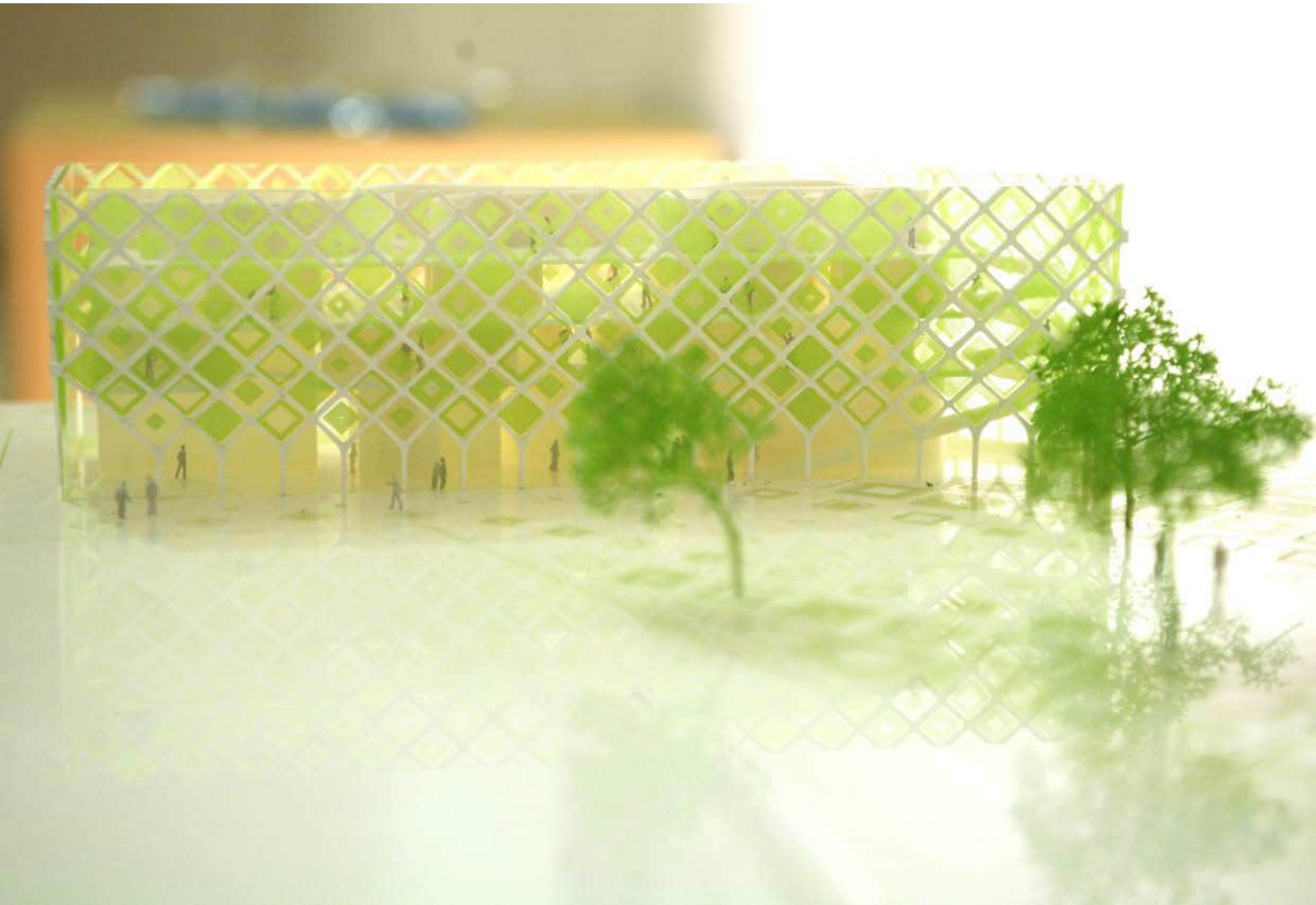


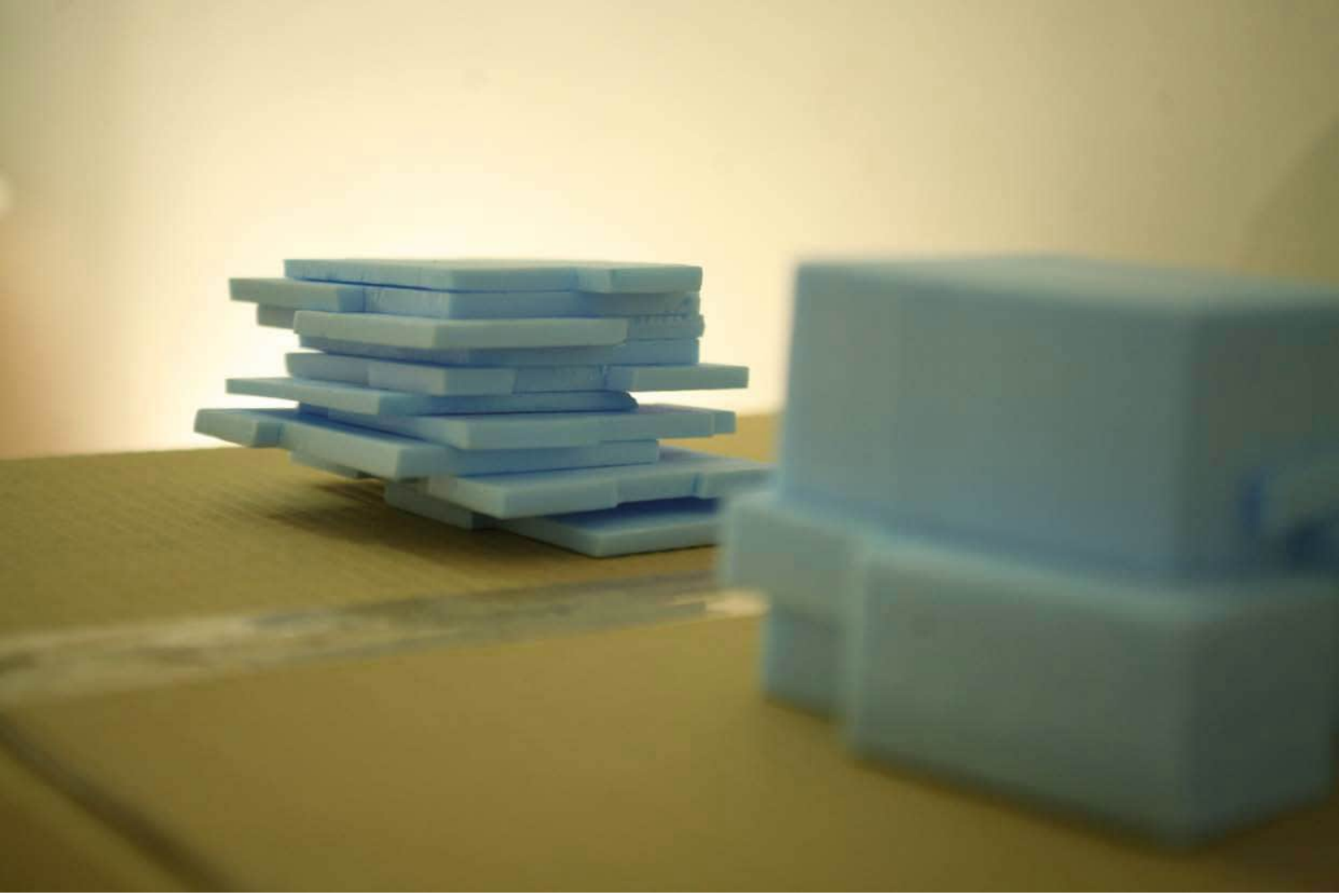


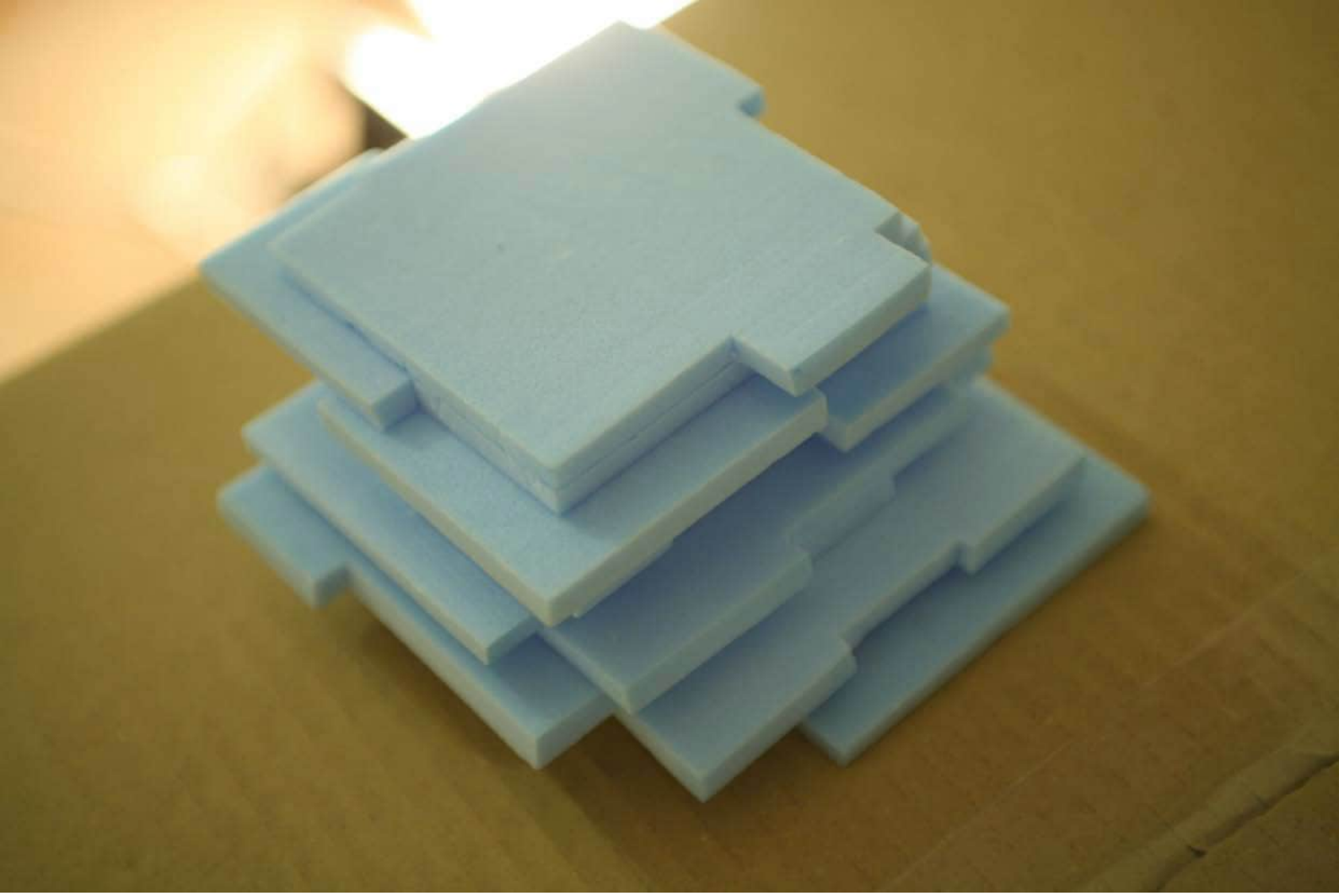


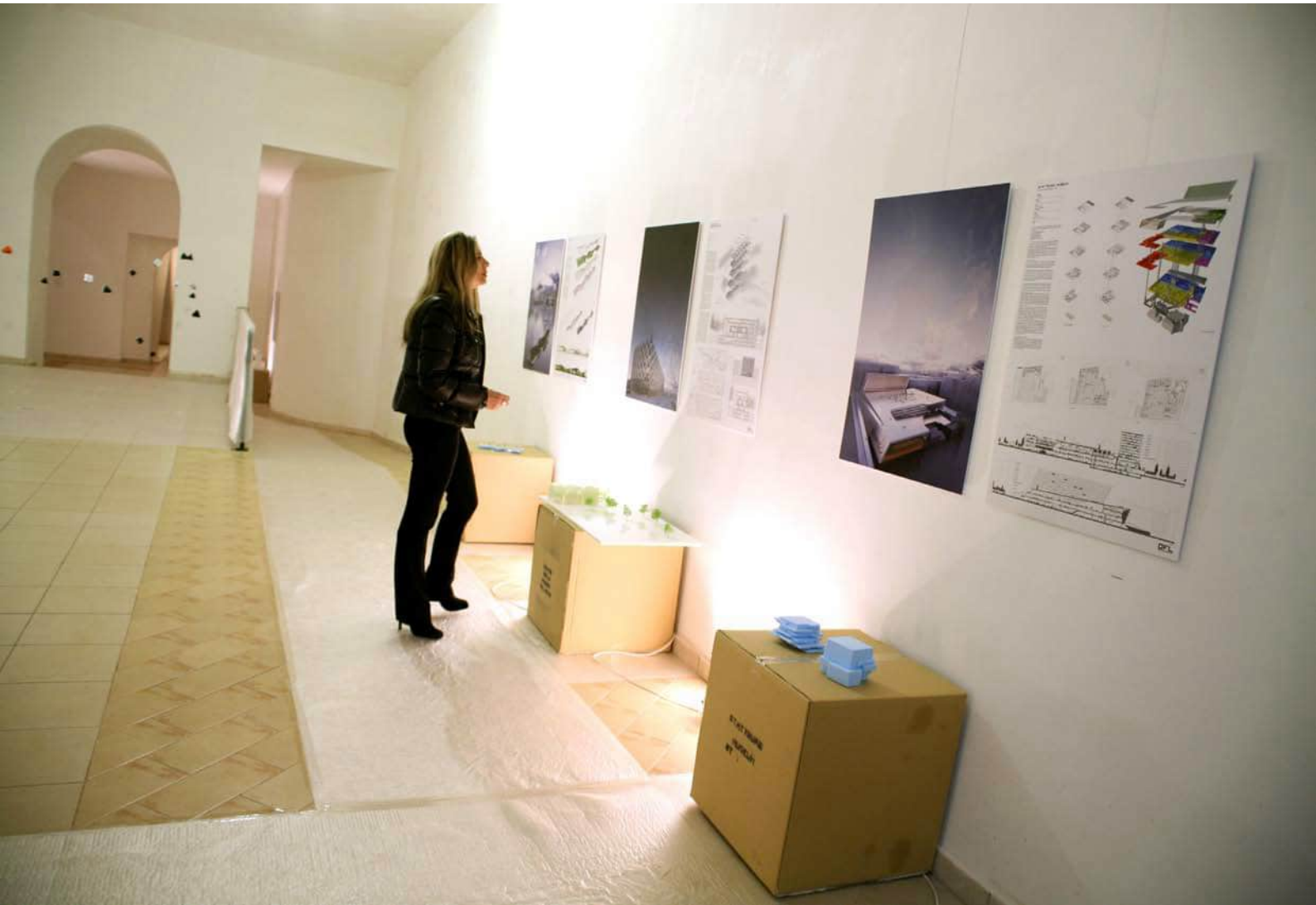


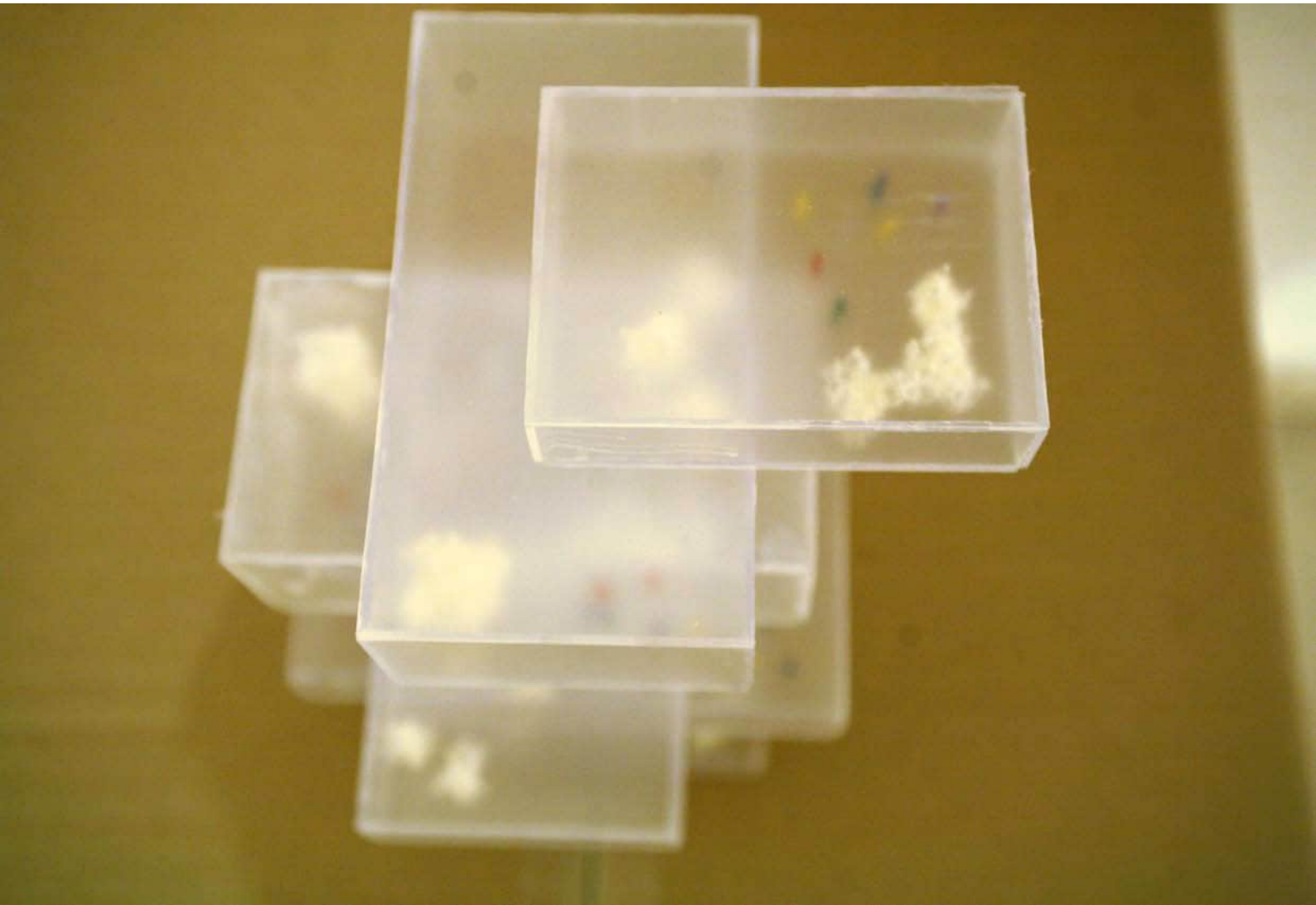













HOUSE OF
ARTS AND
CULTURES

OFL ARCHITECTURE

OFL architecture is an interdisciplinary architectural research laboratory interested in redefining potential relationships within the contemporary city and its existing urban conditions. Francesco established OFL architecture in 2008 with the intention of developing a new design methodology that integrates architecture with parallel disciplines: Art, Sculpture, Biology, Cinema. In 2009 and 2010 the office receive several prizes: the honourable mention at the Hong Kong Noise barrier competiton with the Riccio project, the third prize at the PontèPigneto competiton with the project Instant Bridge and the first prize for the Silk Road Map International Competition. In december 2009 OFL has its first solo exhibition in Sicily. The exhibition named CARNE provoked the attention of the critic thanks to the provocative installation and projects. His works have been exhibited and published in Rome Venice, Milan, Seville, Seoul, Turin, Stockholm, Bangkok, Shanghai, UK, Hong Kong.

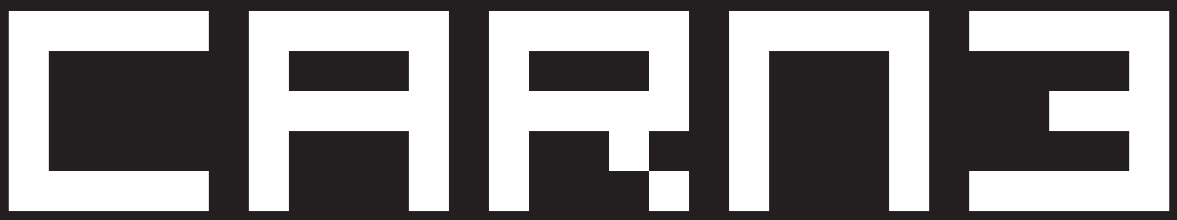
works
10 SRME Silk Road Map Evolution - Shanghai
10 Pass Ater Tiburtino III international competition with Transsolar
10 FARM Favara Cultural Park - Favara (AG)
10 PontèPigneto - Roma 3° premio
09 Statsbygg national museum of art, architecture and design - Oslo
09 Nuova sede centro edile Andrea Palladio - Vicenza, Italy
09 International Noise Barrier competition design - Hong Kong
09 House of arts and cultures - Beyrut
09 Guyang masterplan - Huaxi, china
09 Conrad hotel - Beijing, china
08 London Floating Exhibition Gallery - London
08 International Idea Competition for Cheongna City Tower design in Incheon - Korea
08 The Hermitage Guggenheim Vilnius Museum - Vilnius
08 Shopping Mall - Bratislava
08 International Idea Competition for Cheongna City Tower - Incheon, Korea
07 Evolo International Housing competition - New York
07 International School competition - Treviso, Italy
07 Europan 9 - Reggio Emilia, Italy
07 International Competition for Public Administration Town - Seoul, Korea

exhibitions
10 Silk Road Map Exhibition - Shanghai WTC
10 27/37 Rassegna Internazionale di Giovani Architetti Italiani - Shanghai Italian Pavillion
10 PontèPigneto - Rome
09 CARNE - OFL solo exhibition - Campofranco, (CL)
09 Hong Kong Noise Barrier Design exhibition - Hong Kong
09 27/37 Rassegna internazionale giovani architetti romani - Seville, Rome
09 aast / Advanced Architecture Settimo Torinese
06 10th Venice biennale

lectures
10 Casa dell'architettura - Progetto innovativo: il caso hong kong noise barrier - Cesarch, Rome

awards
10 Silk Road Map international competition. First Prize
10 Active Graniti Fiandre award. Special Mention
09 International Noise Barrier competition design. Honorable Mention
07 International competition TORINO GEODESIGN. Winner
07 Workshop WinP_07 and Lions città di Salerno. First prize

press
10 RUM / Stockholm
10 FRAME / UK
10 MARK / UK
10 B1 Magazine / Bangkok
10 Settecortili / Favara (AG)
10 ecologic structures / Milano
10 young italian blood / Rome
09 cityproject / Hong Kong Noise Barrier
09 talents mag / Rome
09 peruarki / Hong Kong Noise Barrier
09 storm from the east / Hong Kong Noise Barrier
09 insideffeders / Hong Kong Noise Barrier
09 archicentral / Hong Kong Noise Barrier
09 archiportale / Hong Kong Noise Barrier
09 architetto.info / Hong Kong Noise Barrier
09 professione architetto / Hong Kong Noise Barrier
08 concept 115.09 / London Floating Gallery
08 concept 13/09 / Venus Tower
08 citytower International competition book / Venus Tower
08 europaconcorsi / Venus Tower


CARN3