CAAD Professur Hovestadt
ETH Zürich

ETH Zurich
Prof. Dr. Ludger Hovestadt
Chair for CAAD
Institute for Technology in Architecture (ITA)
HPZ Floor F
Schafmattstr. 32
CH-8093 Zurich

T +41 44 633 34 89
F +41 44 633 10 50

http://www.caad.arch.ethz.ch/
http://www.mas.caad.arch.ethz.ch/

CREDITS

Professor
Prof. Dr. Ludger Hovestadt

MAS Coordinator
Michael Hansmayer

CAAD Chair

Markus Braach
Benjamin Dillenburger
Philipp Dohmen
Pia Fricker
Michael Hansmeyer
Martin Jann
Steffen Lemmerzahl
Karsten Droste
Tom Pawlofsky
Sibylla Spycher
Christoph Wartmann
Oskar Zieta
Vera Bühlmann
Mathias Bernhard
Georg Vrachliotis

MAS 08/09

Mathias Bernhard
Katerina Bouziana
Kent Brockmann
Günes Direk
Aphrodite Stavropoulou
Jasmin Zarali

INDEX

PARAMETRIC **CARDBOARD**

The first module of the MAS was about practicing on the concept of digital chain.
The digital chain is a seamless process of concept, design, development, fabrication, and construction - all using the same architect-produced and coordinated data. By using portable-scale computing, and designing completely in 3D, data can be sent directly to facilities which provide modular cutting services. The data is quickly converted to finished and highly customized components.
MAS students did their first experiments with RhinoScript by programming 3D geometries for lamp shades, which was later exhibited at Extra Bar in Zurich.
Next step was a challenge to program and construct a pavilion, a dome built according to the construction principles of a traditional igloo.

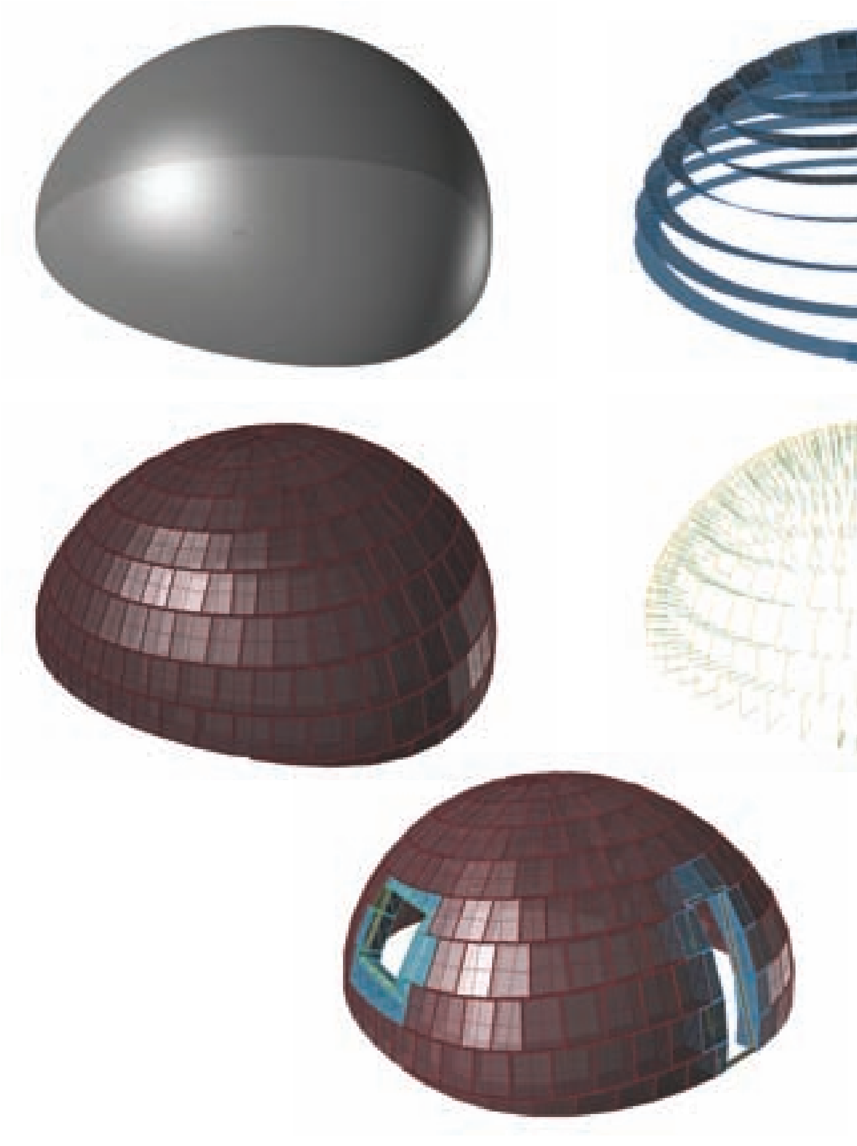
SCRIPTING

1
The overall shape can be any free-form surface. The only condition is that the curvature is in the same direction everywhere on the surface. A change of curvature (convex) would cause the igloo-bricks to “fall out”.

2
A spiral is drawn on the surface manually with a regular u-v distribution.
This curve is reworked by hand in order to prevent to narrow pieces. The final spiral-curve is subdivided with a second script into segments of 20mm. These subdivision points are triangulated with a shortest edge algorithm to have the lines as vertical as possible. Intuitive manual adjustments can be made for poorly fitting proposals.

3
Another script helps finding the joints by proposing the maximal distant line. The surface is divided up through vertical lines, according the spiral. The final position of the blocks is defined by manual selection of the lines.
The wall thickness of 400mm was then added to these surface lines by adding lines pointing to a “focal point” at the center of the igloo’s floor.

4
By selecting five of these “C-shapes” per brick (two borders plus the neighbours), a script automatically generated the “M-Shape” - the edges of each of the 311 individual pieces.
A next script generated all the surfaces of a folded brick (with all the overlapping for assembly) which could then be unfolded and numbered to generate the machine-files for the production.



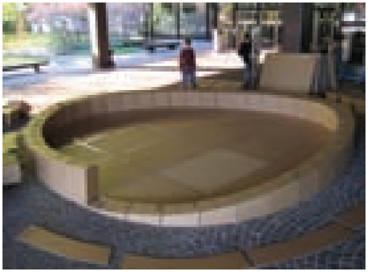
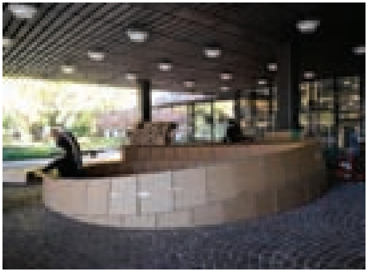
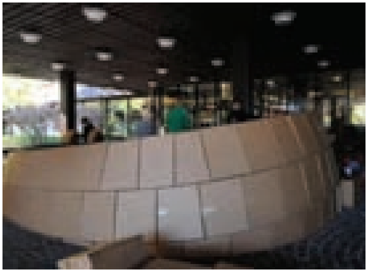
PRODUCTION

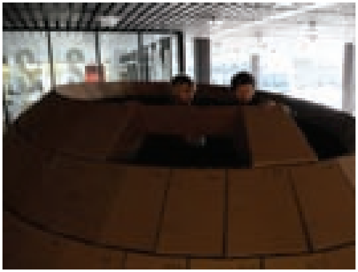
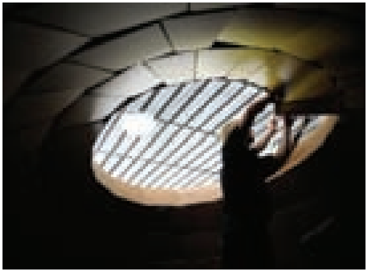
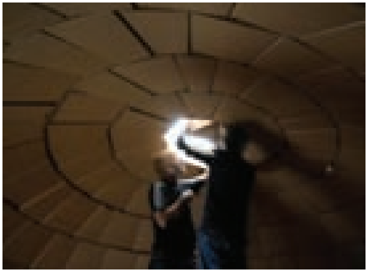
Unfolded and numbered pieces are produced by the cutting plotter, ready to be folded and assembled to create 311 unique blocks. After sourcing a window and door from a retired passenger railcar, frame elements could be built. Frame locations were then selected in the wall and calculated. Corresponding blocks were then recalculated using the new frame information. New cutting patterns were produced and the new pieces were substituted for the originals. The floor was used as a guide to place the first circle of 46 boxes. A series of cardboard strips were used to laterally connect boxes together in groups of 4.

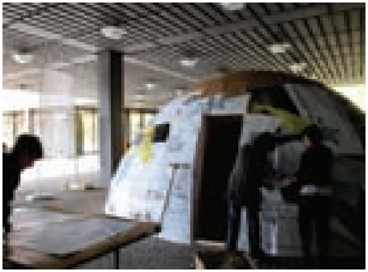
ASSEMBLY

The strips form a long spiral band which provides momentary stability during assembly. It also serves as a unit placement guide for ease of construction. Gravity and compression ultimately hold the pavilion together vertically, as every pass of blocks is placed freely on the layer below. Commercial poster paper was donated by a local promotions firm for the outer membrane. The skin provides an elementary level of protection from both wind and water. It is assembled arbitrarily as pieces are glued in spontaneous arrangements. By gluing sheets across numerous block units, additional stability is provided with simple means. By concealing the spiral of blocks on the outside, visitors are introduced to the uncovered units which dramatically present the construction process.



CUTTING

In order for the pavilion to be moved, it needed to be divided into smaller pieces. For those parts which were at top center, the pieces were made small enough for one person to handle overhead - 4 pieces with between 5 and 6 units joined together covered enough area that they could be placed from the bottom by hand. This left the surrounding “wall” material, which were divided into 7 tall thin sections, similar looking to cake slices. These were small enough so that a minimum of two persons could easily transport them between the site and truck.
The units were joined on the interior by stapling strips of cardboard along the bordering lines of separation. Then cuts were made through with a small saw to pierce both the continuous spiral ring, and covering paper to detach the pieces from one another.

RE-ASSEMBLY

After 4 trips with a small transport truck, all of the pieces were transferred to a nearby park with a nice overlook of the valley and city of Zurich. Once the pieces were moved to the assembly location, they were laid out around the floor radius to verify the order. The side pieces were lifted and joined together, beginning with the door and window sections, since they would be more vulnerable to the sudden movements necessary when adding new pieces. Overall, actual reassembly took less than one hour, and the pavilion remained at its new home from one afternoon, to the next morning, in all, about 16 hours.

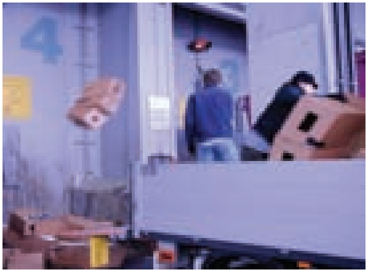




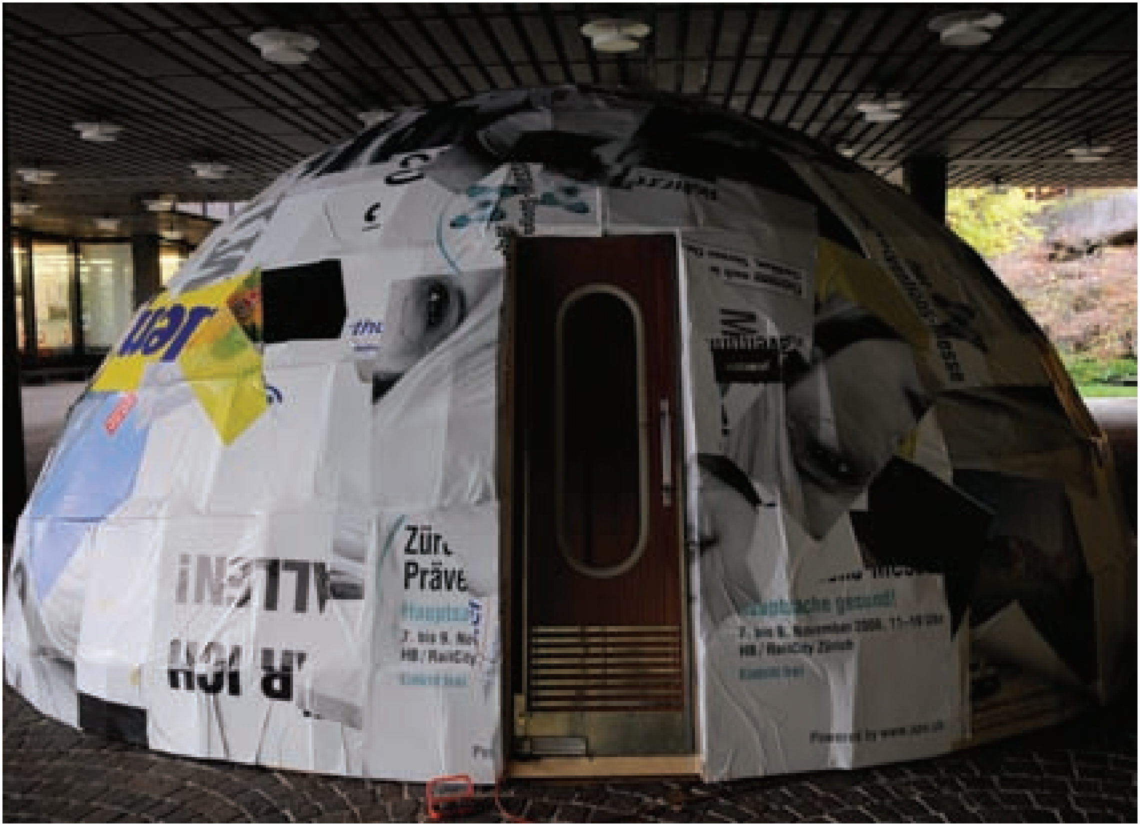






Poster for "Simple Systems - Complex Capacities" competition, 6th place in 115 entries.

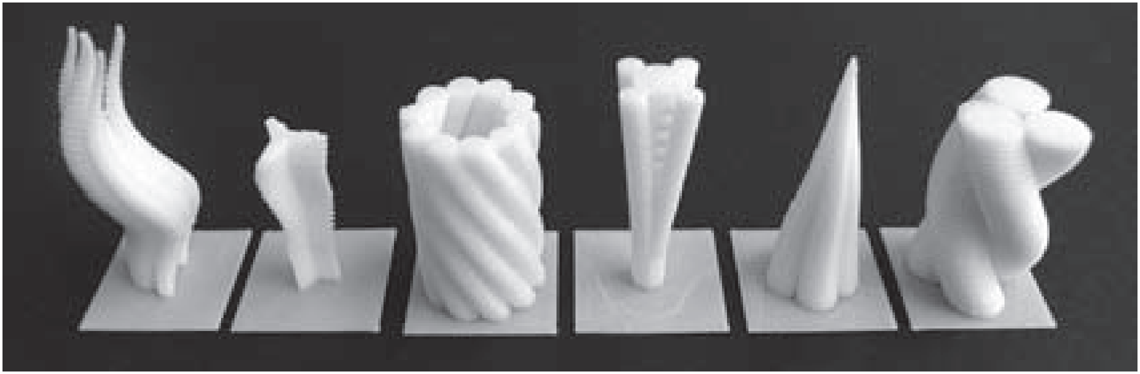


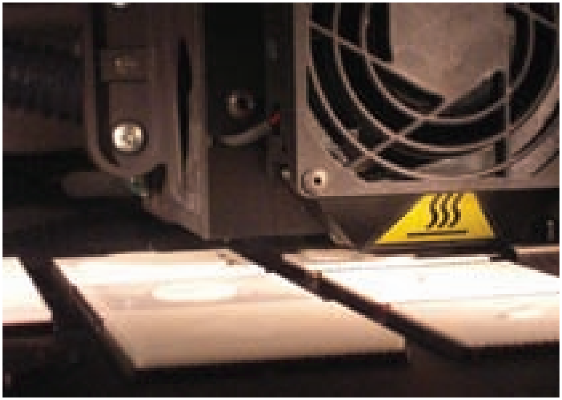
Dimensions: 6x7x3 m
Floor area: 30 m^2
Number of units: 310
Cost /m^2: 18 €
Cutting time: 12 h
Folding: 14 h *
Assembling : 10 h *

* 5 person crew

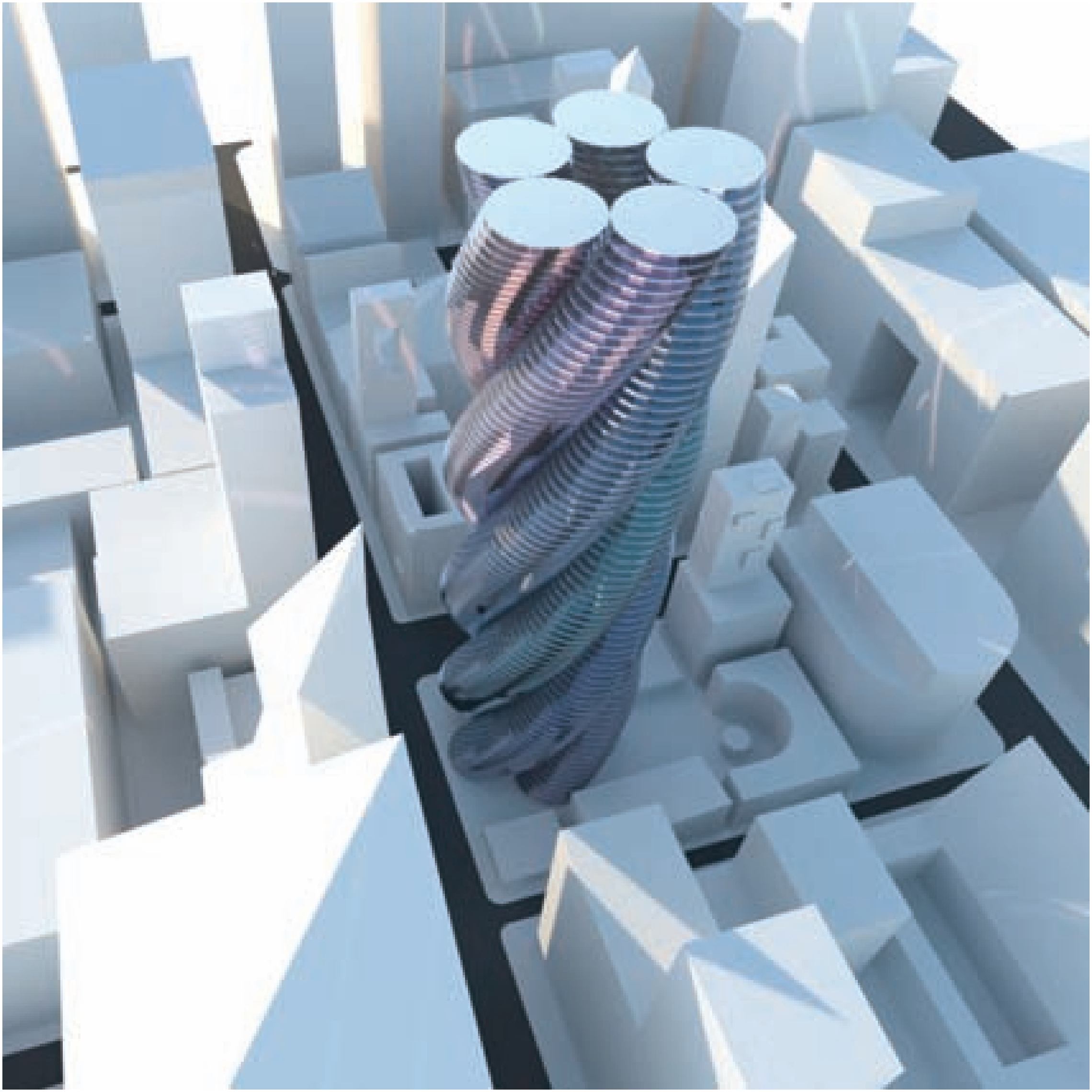
FRAMEWORKS

The MAS project “Frameworks” ,under the supervision of Marcus Braach and Benjamin Dillenburger, was the first step into object-oriented programming (OOP), using Processing and Java.
The first part of the Processing course was the previous workshop from Christoph Wartmann, “Parametric Skyscrapers” , in which the students designed skyscrapers based on parametric algorithms in Processing and prepared their 3D models for printing on the 3D plastic printer. With a few lines of code in Processing, it was possible to make complex geometric objects, resembling modern skyscrapers, which are then exported to .stl file format, and used as an input to the 3D printer.

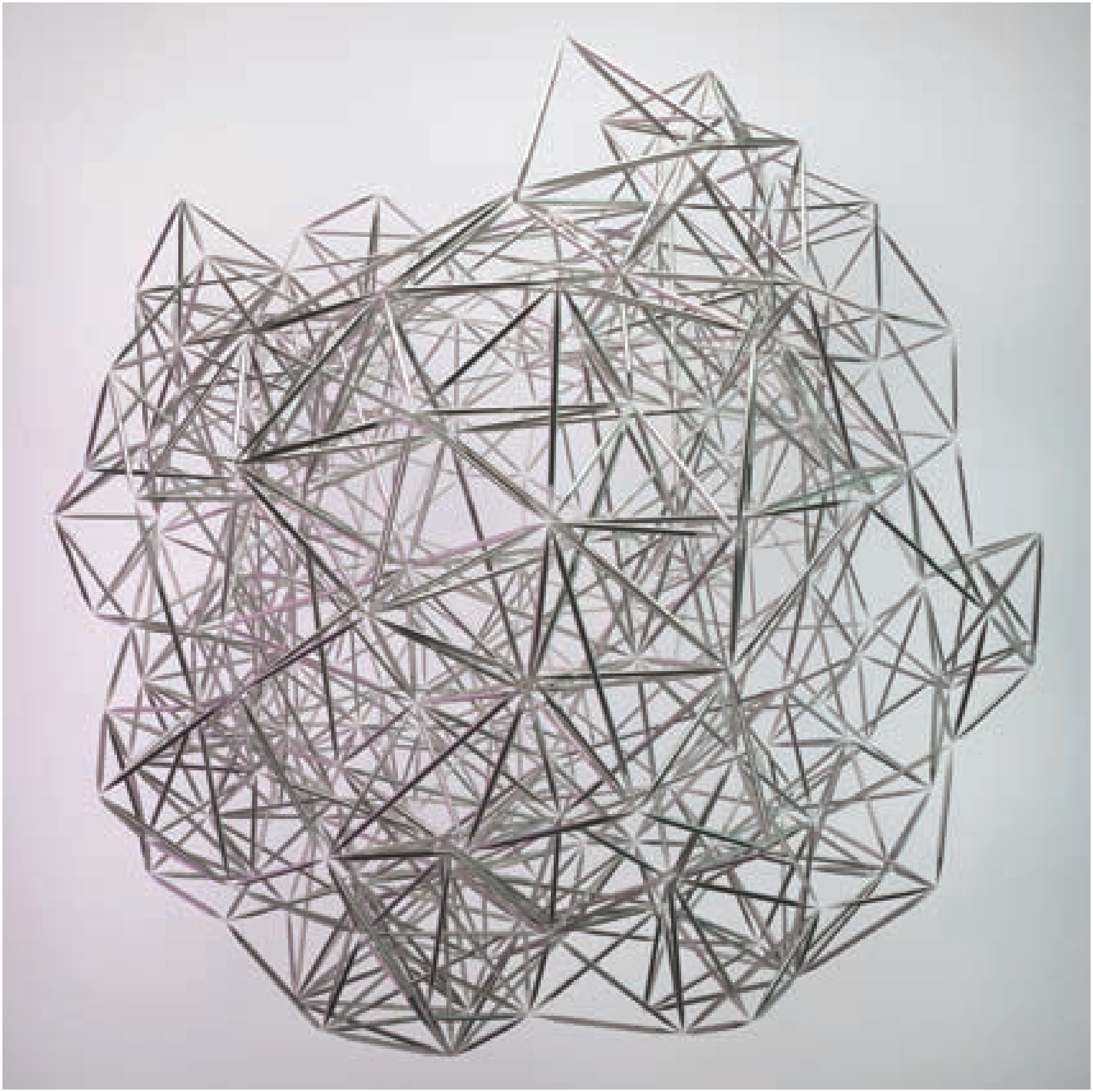


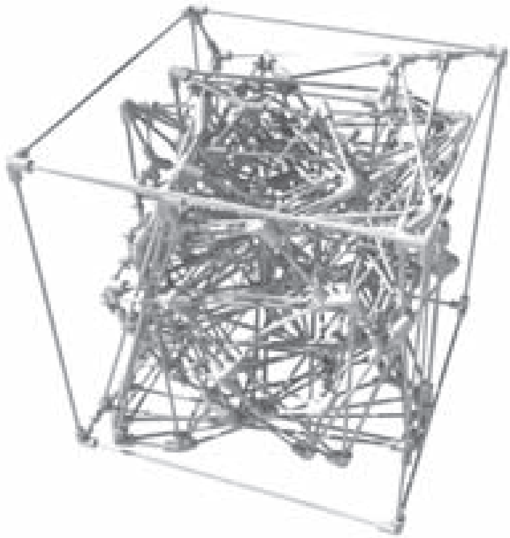


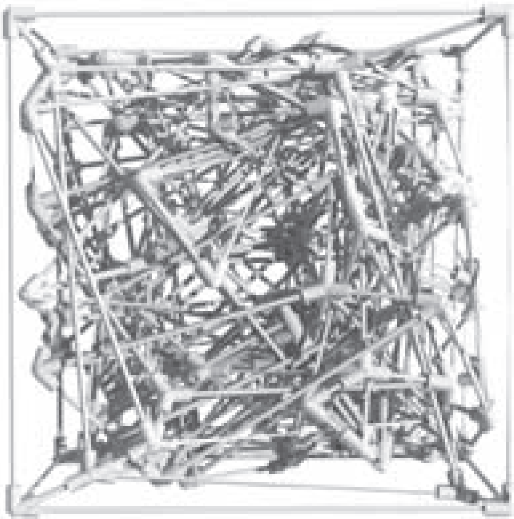
3D printer in during the printing process and the final results.

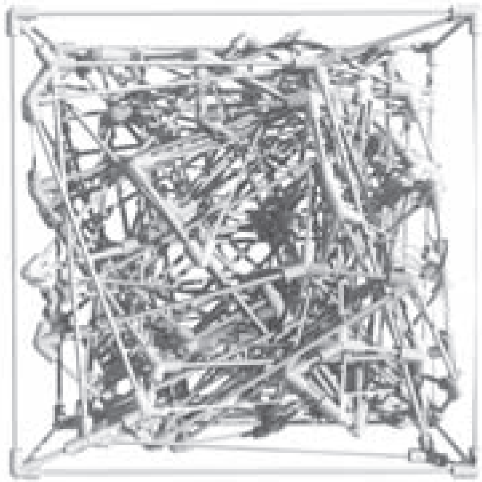


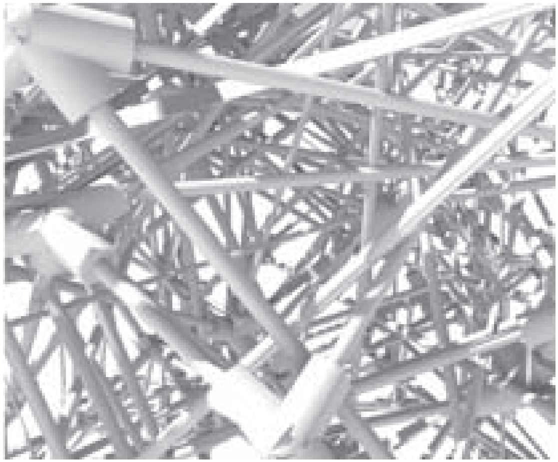
Frameworks module takes this one step further and introduces the concept of object-oriented programming. This is done by creating classes for connections and nodes. Nodes interact according to the forces present on the connections, thus creating a dynamic and complex structure, which tries to organize itself and maintain a balance of forces.
The structure generated with Processing is then exported to a 3D geometry and used as an input for the code written in RhinoScript, which in turn created various types of nodes and connections.

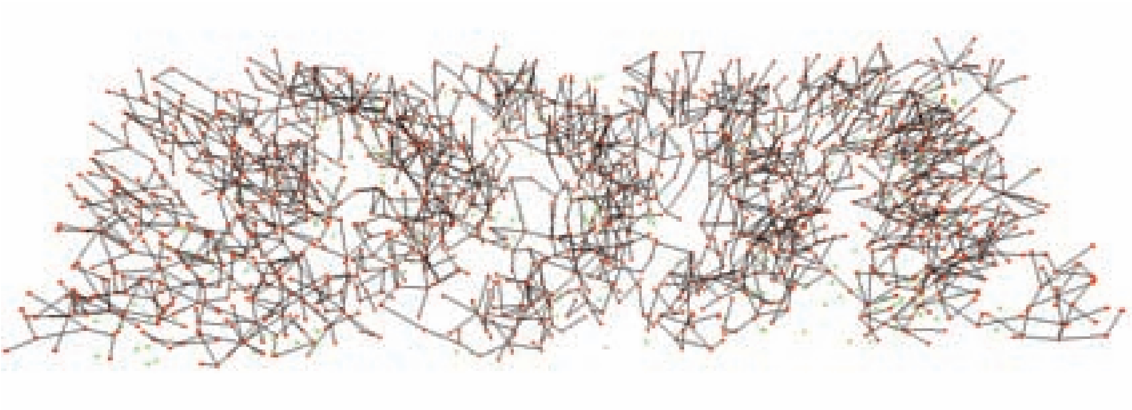


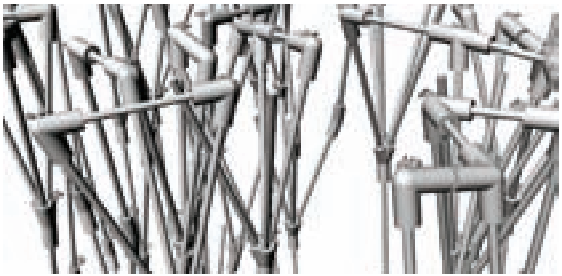




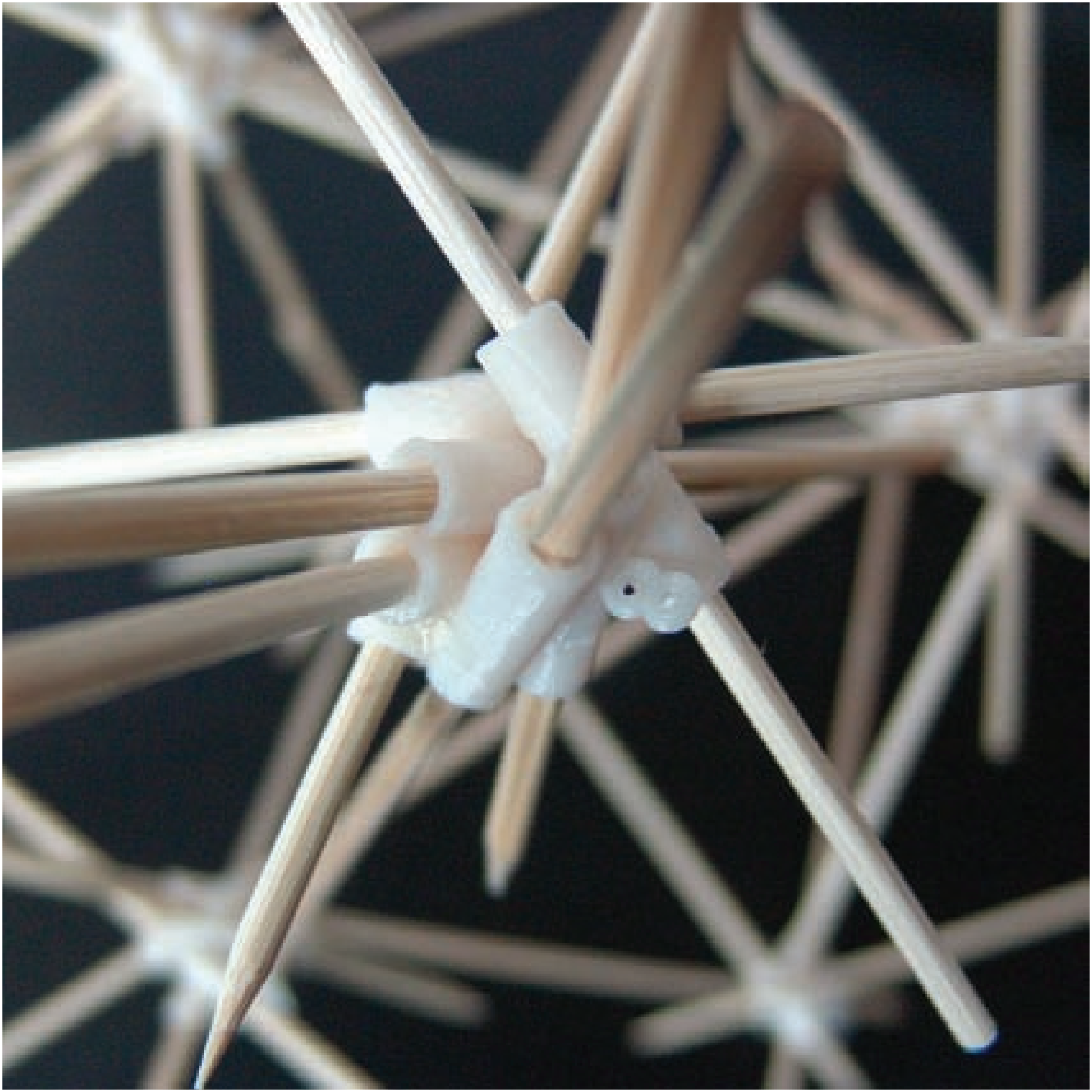


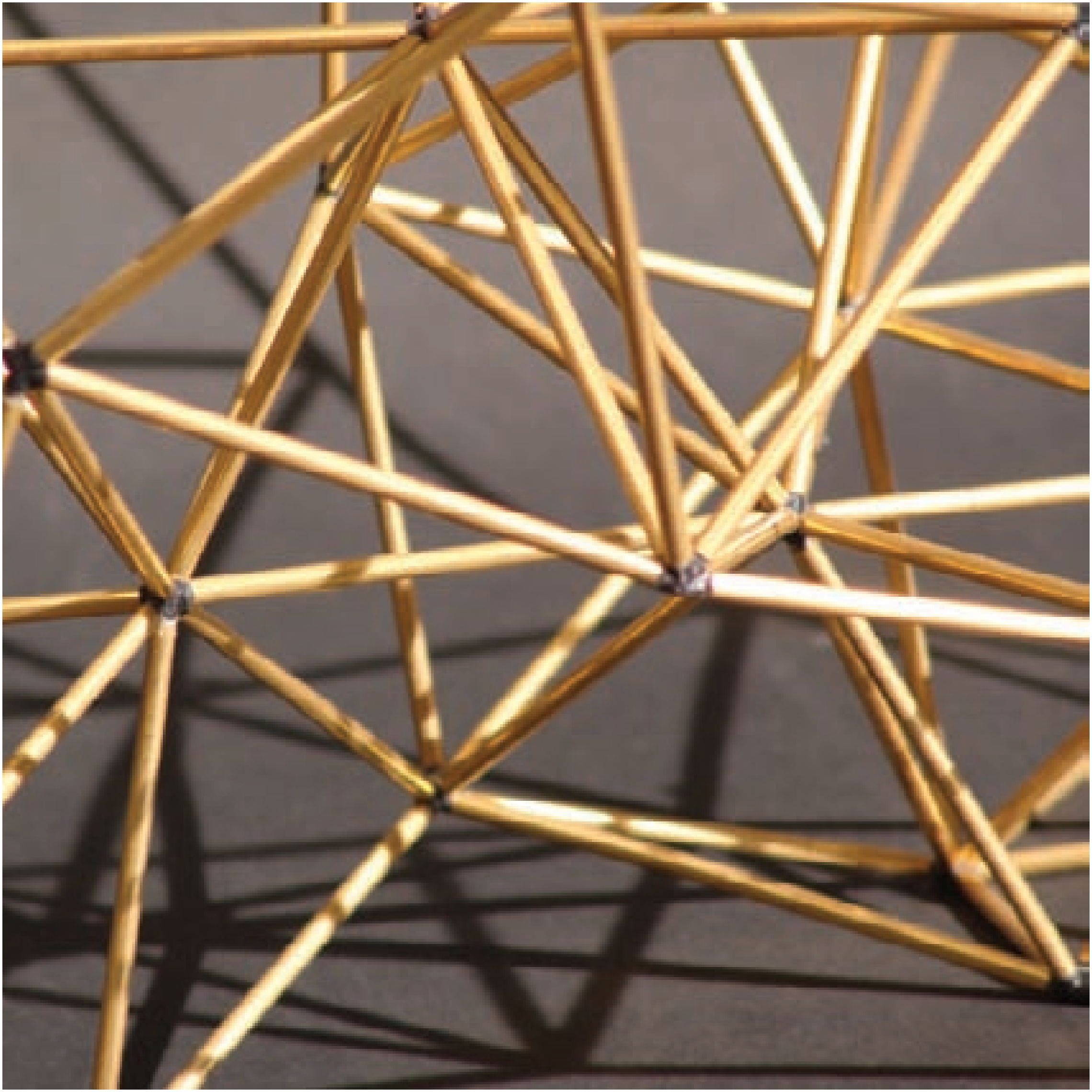


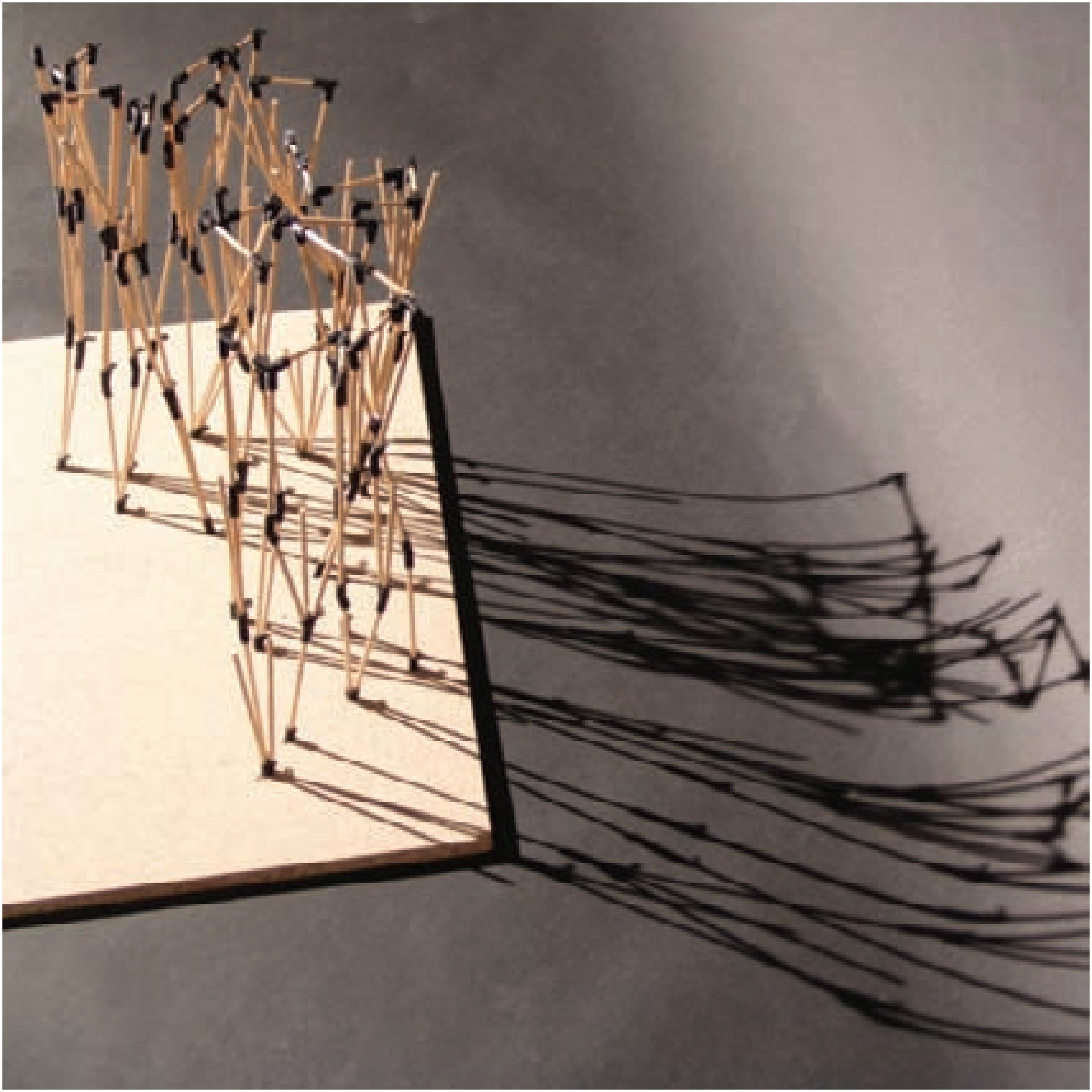




Printed nodes covered with support material from the 3D printer.





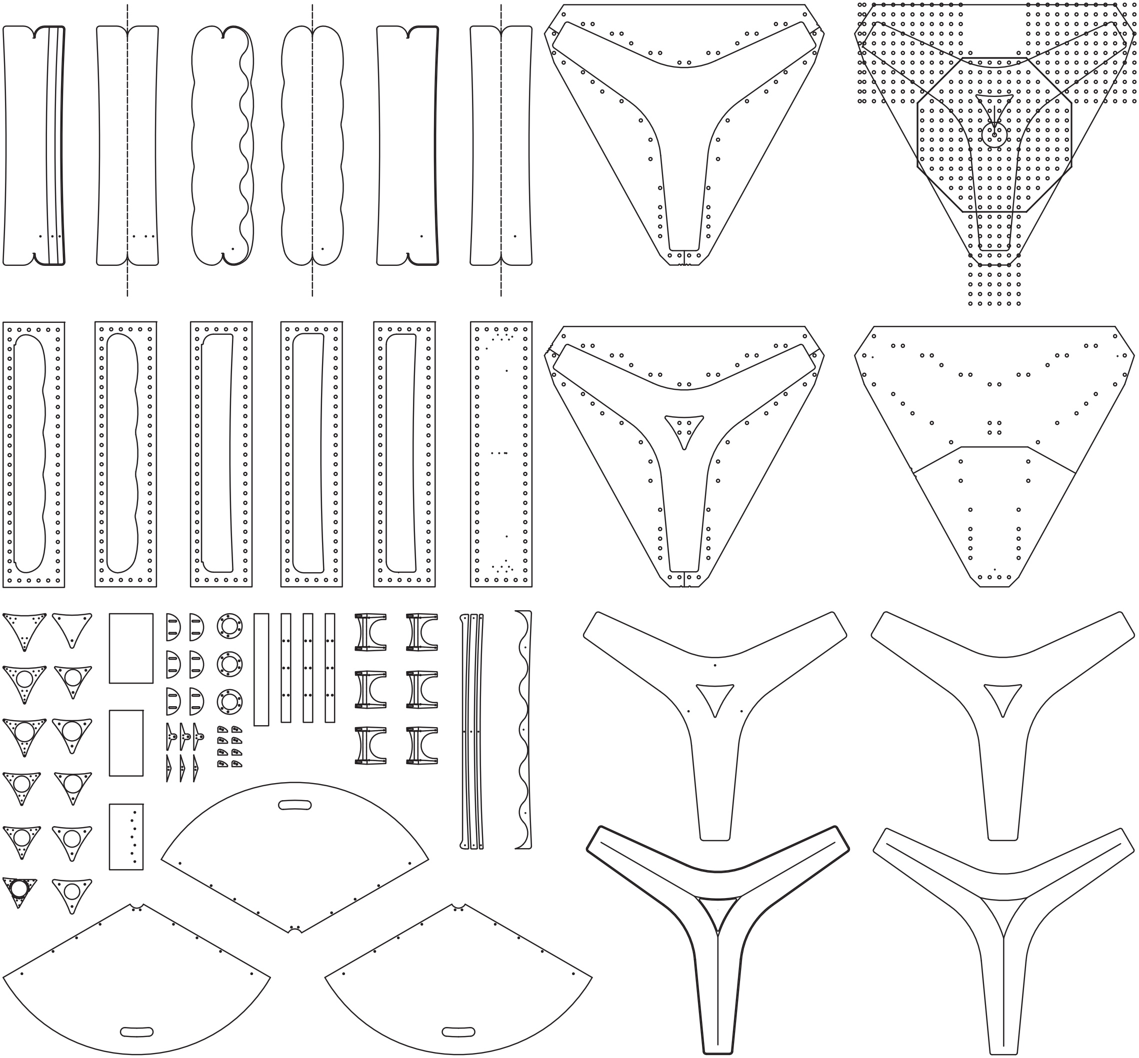


COEUS **FIDU-ROTOR**

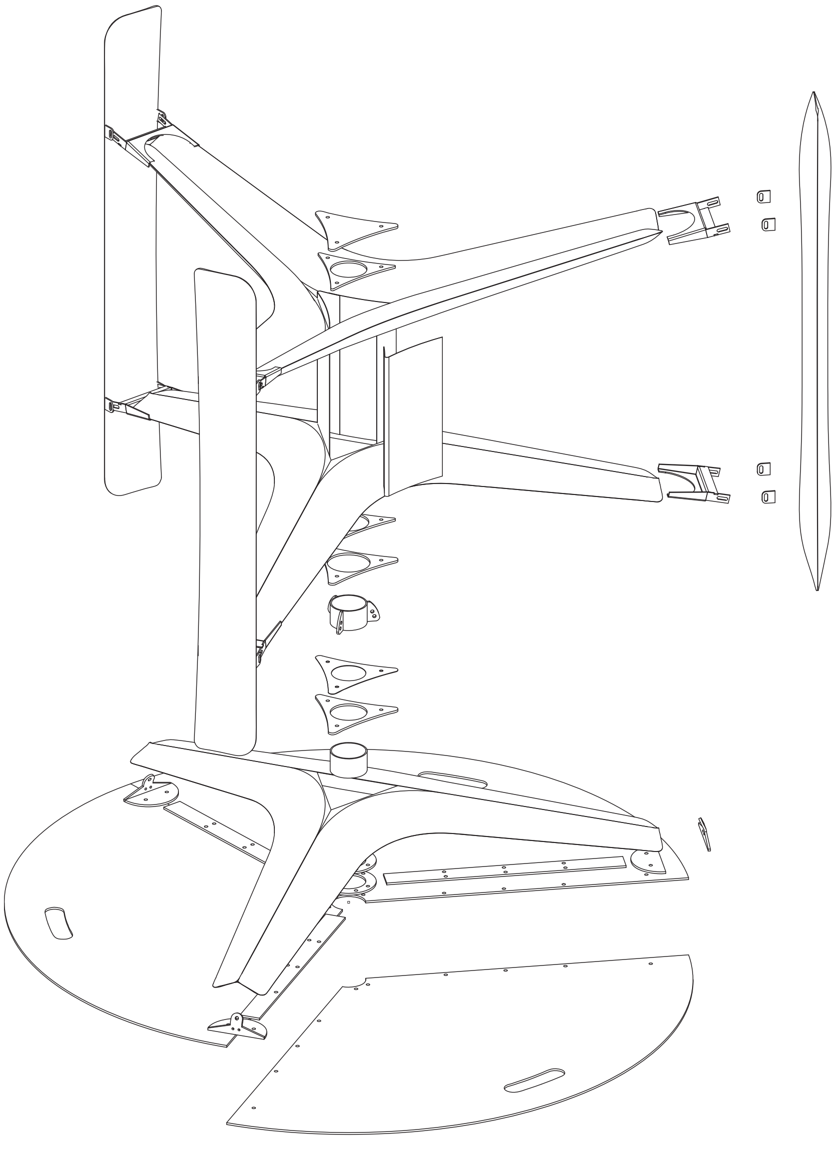
FiDU, ‘Freien Innen Druck Umformung’, is a recent development of Oskar Zieta and Philipp Dohmnen. This kind of forming has the benefit that it processes the metal completely without any tools. The MAS module FiDU tries to explore more on the possibilities of this technique.
First tries were made at the workshop at Hönggerberg campus ETH, where students learned basics about welding. Through collaborations with metal processing companies laser cutting technologies are used in the further experiments of the module.
The main project of this module focuses on designing a wind turbine using FiDU technology.

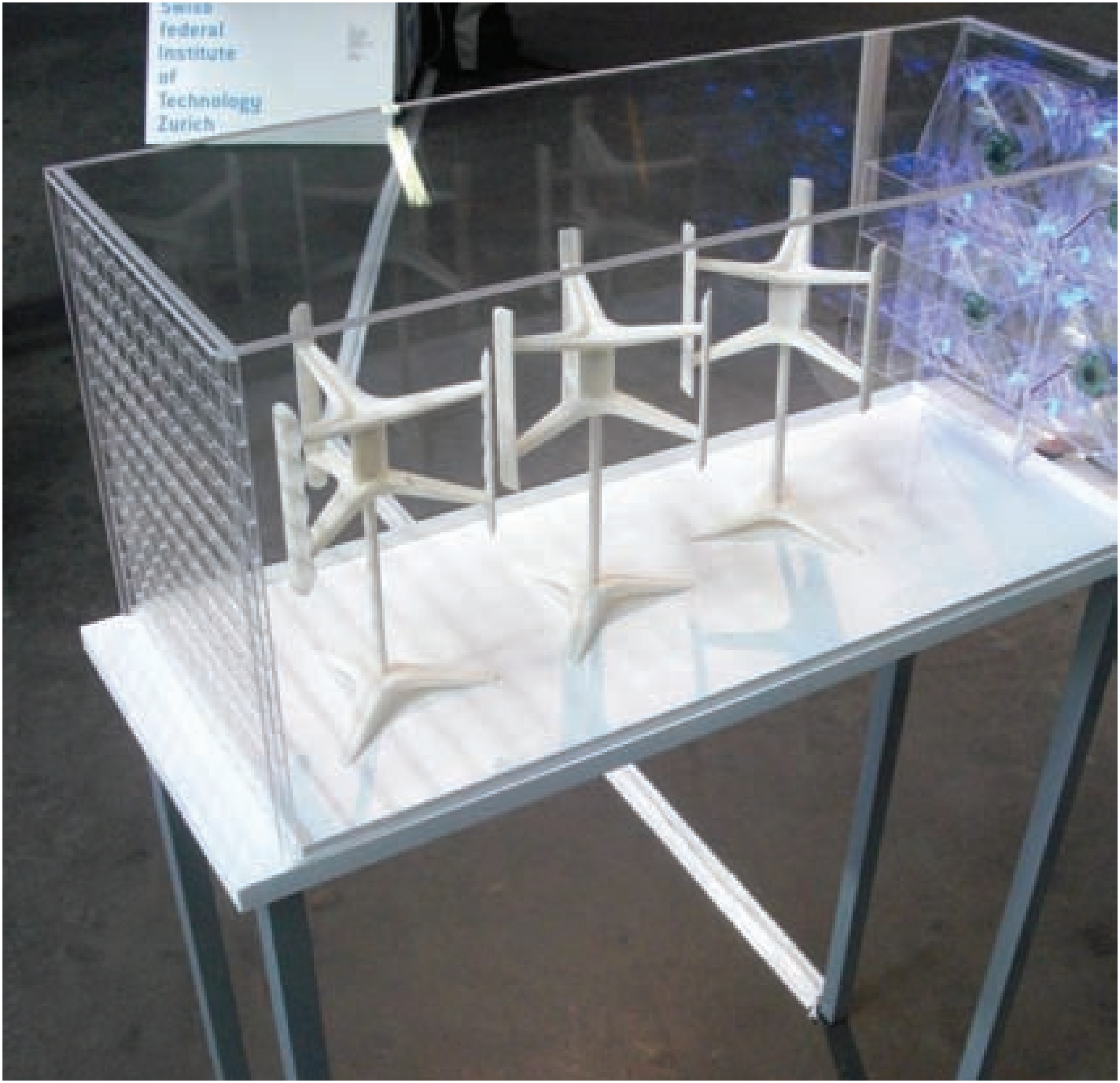
In the search for possibilities concerning renewable energies, a wind turbine is demonstrated which can be produced using FiDU technology. A Vertical Axis Wind Turbine (VAWT) format was chosen since they can operate efficiently without sophisticated and expensive equipment to orient them into the wind. They are often more flexible in their design and have a wider range of applications than horizontal turbines with regards to the potential to decentralize yet upscale the capacity for energy production.

The dimensions and proportions were based on existing studies of free-standing wind turbines. Models were used to test which forms and contours resulted in the desired profiles. Three separate wing designs were subsequently advanced and developed: a single-chamber, triple-chamber and a waveform wing. These designs lead to different approaches regarding the connection of the three-armed support. The production of the primary components of the wind turbine was accompanied by a search for suitable assembly methods. The biggest assembly challenge was combining the deformed FiDU elements in a precise and stable manner.

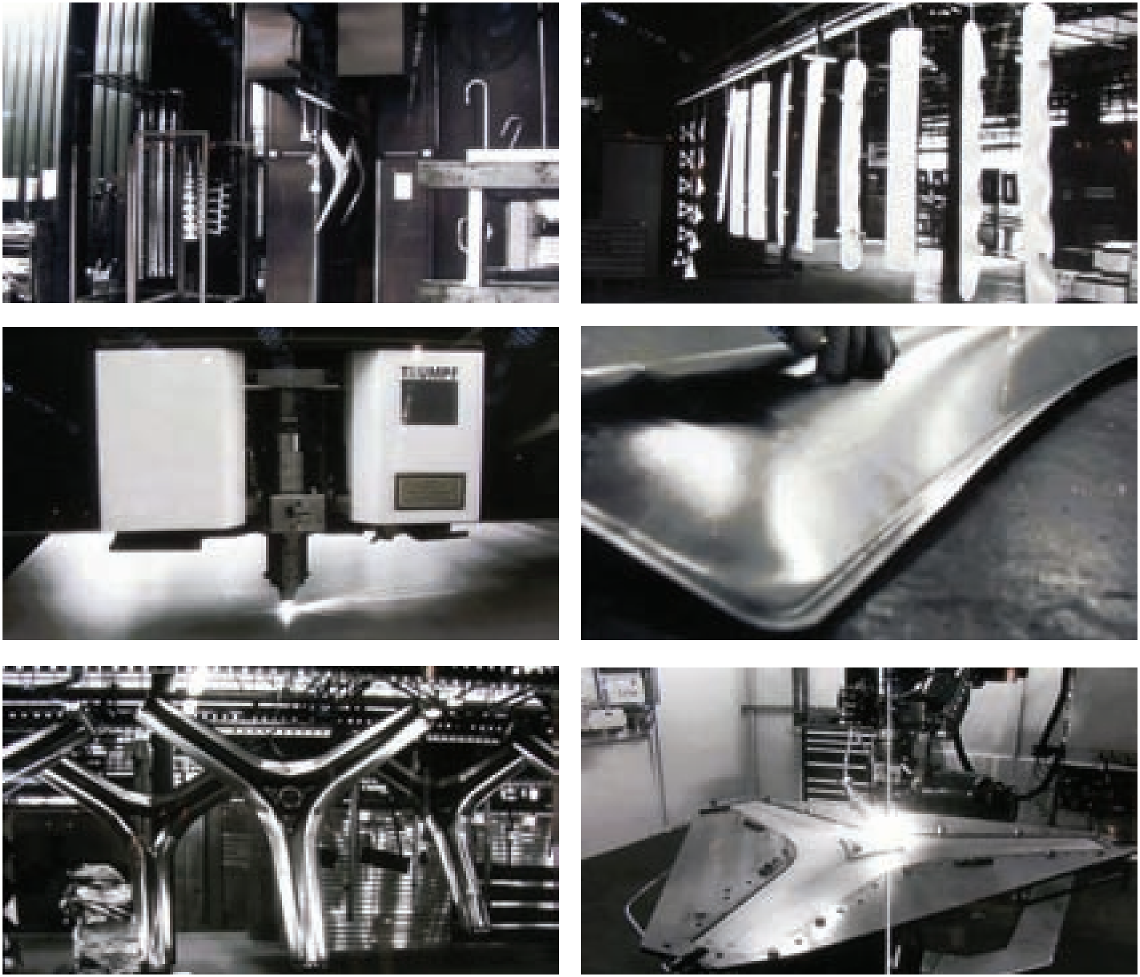


All parts are cut with an industrial laser cutter.





Wind box and 1/10 scale 3D printed wind turbines



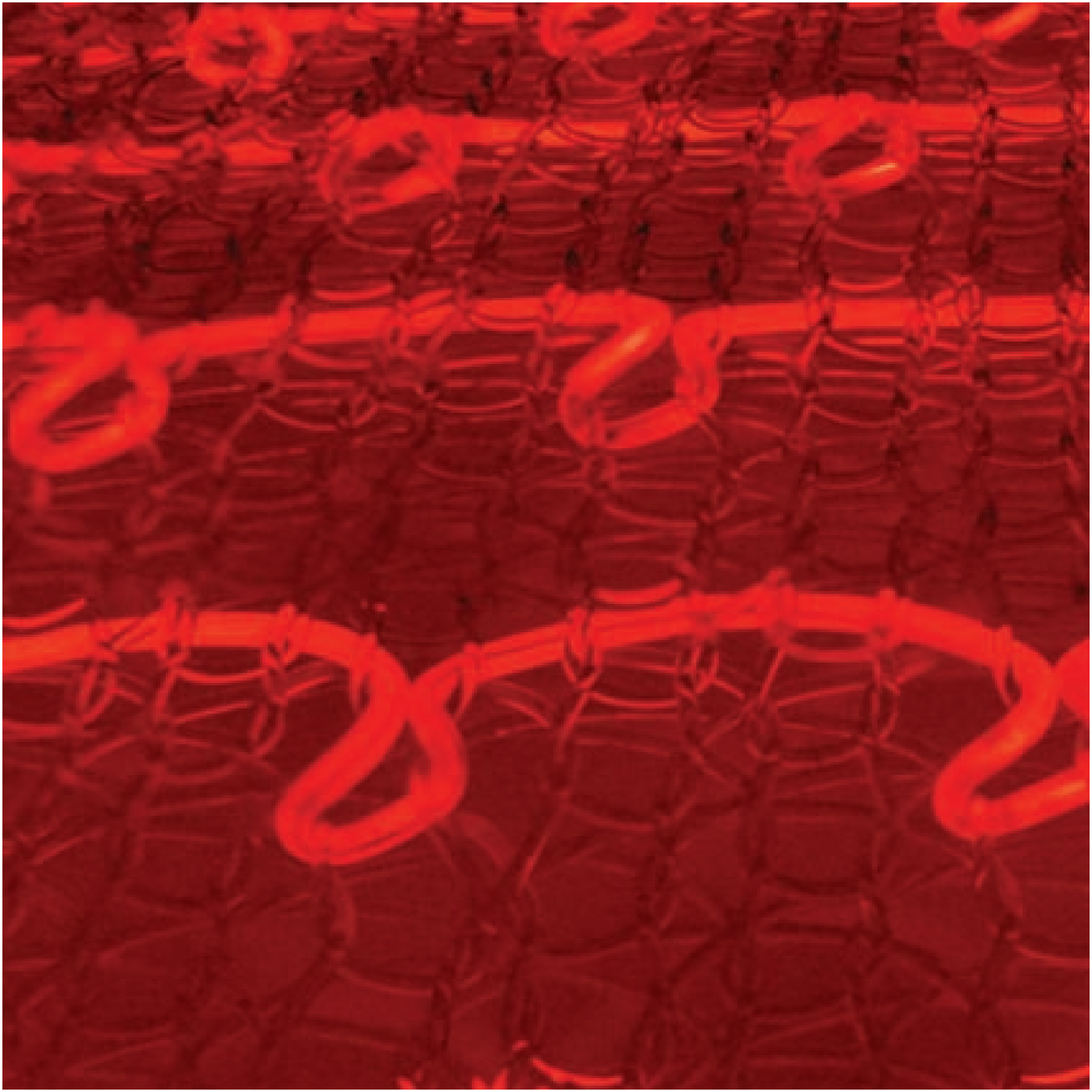
Stills from the short documentation movie, presented at the exhibition

Preparation for assembly at DMY Berlin

PUBLICATIONS

http://www.designboom.com/weblog/cat/16/view/6694/oskar-zieta-and-caad-fidu-rotor-at-dmy-berlin-design-week-09.html
http://www.dailytonic.com/fidu-rotor-by-oskar-zieta-eth-zurich-at-dmy/

Coeus at the DMY Design Festival Berlin, Youngsters Exhibition



INTERACTIVE URBAN **FABRIC**

ABSTRACT

This years final thesis is about experimenting on interactive architecture. With the use of thin electroluminescent wire, the MAS students searched for possibilities to bring visual interaction based on movements in the environment together with tensile structures , creating a small scale architectural space to show the potential in interactive textiles in an urban environment.

CONCEPT

With the rapid evolution of electronic components, complex computations are possible in small dimensions and for low costs. With microcontrollers for analog to digital conversion and computations, and sensors to communicate with the environment, it is possible nowadays, to build simple interactions with a little effort. This has resulted in the growth of a new field in design and architecture.
Interaction is a growing concept in the architecture world. Many examples of the recent years have proven that architectural space can have a different role then what it used to have. Walls and surfaces can become more than a structural or a spatial element, a tool to communicate, visualize or simply respond to its environment. They can detect changes in their surrounding and respond with predefined patterns. Therefore, interaction design on a spatial scale brings a new feeling to architecture, and a challenge at the same time, to bring together different disciplines together and design a frame to make them work together.

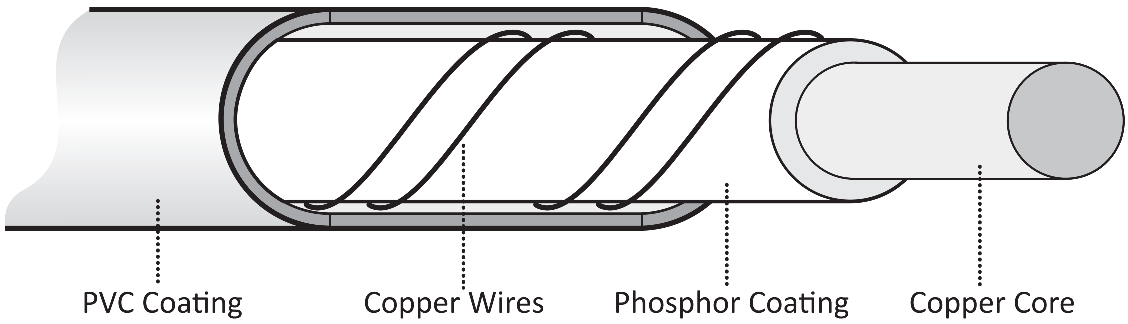
On the implementation side, thousands of web pages, documentations and examples are available online to be used for any kind of application.

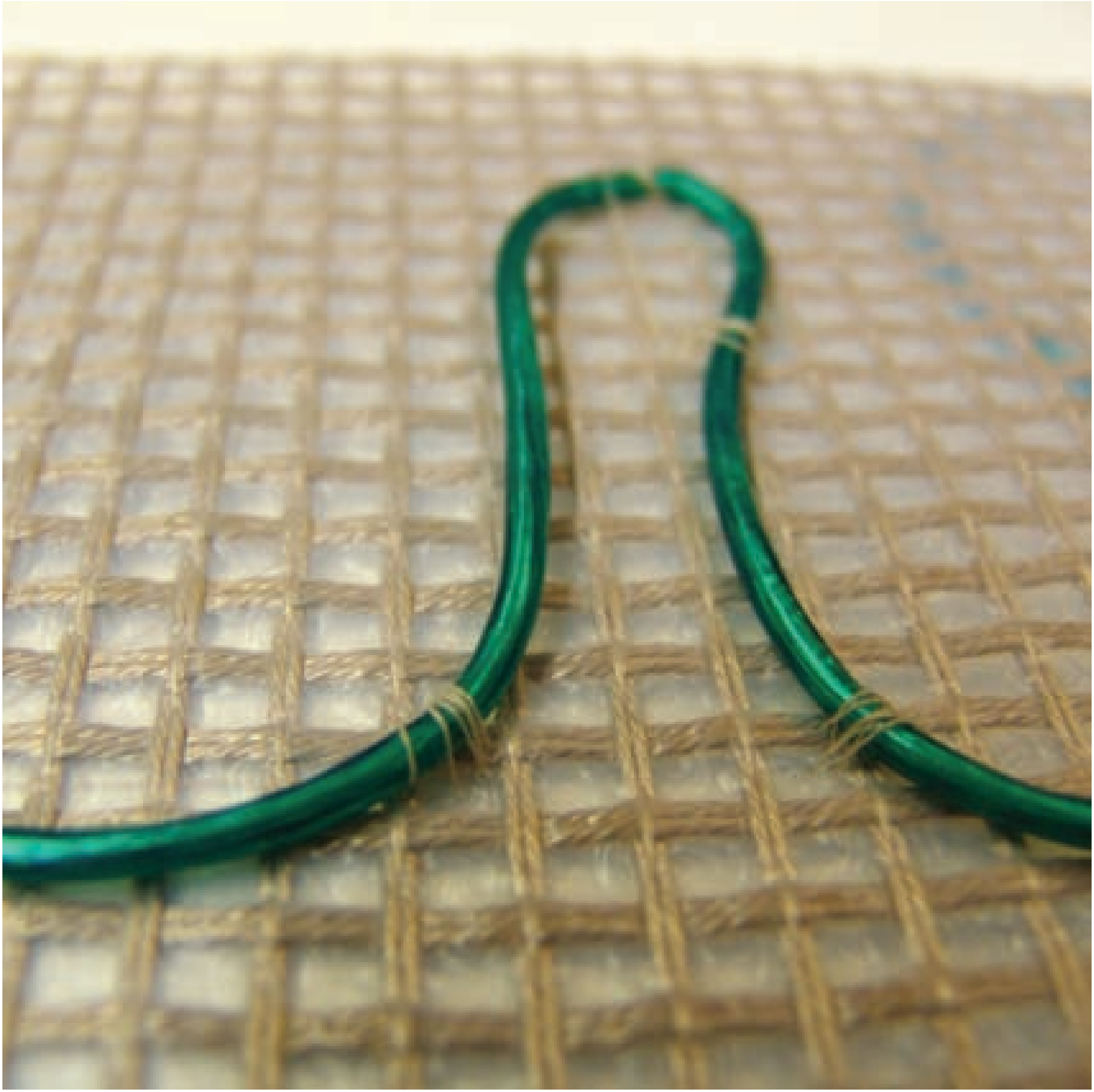
INTERACTIVE TEXTILES
From fashion design to media, the expressive side of textiles have always intrigued designers to create new forms and try out new materials in textile design. Also in architecture, textiles as flexible surfaces have always had a great potential. It expresses fluidity, deformability and movement in architectural spaces.
With the use of interactivity, the research for textiles had a new challenge, new technologies in material science, developments in production techniques and experiments from designers have contributed to this field.

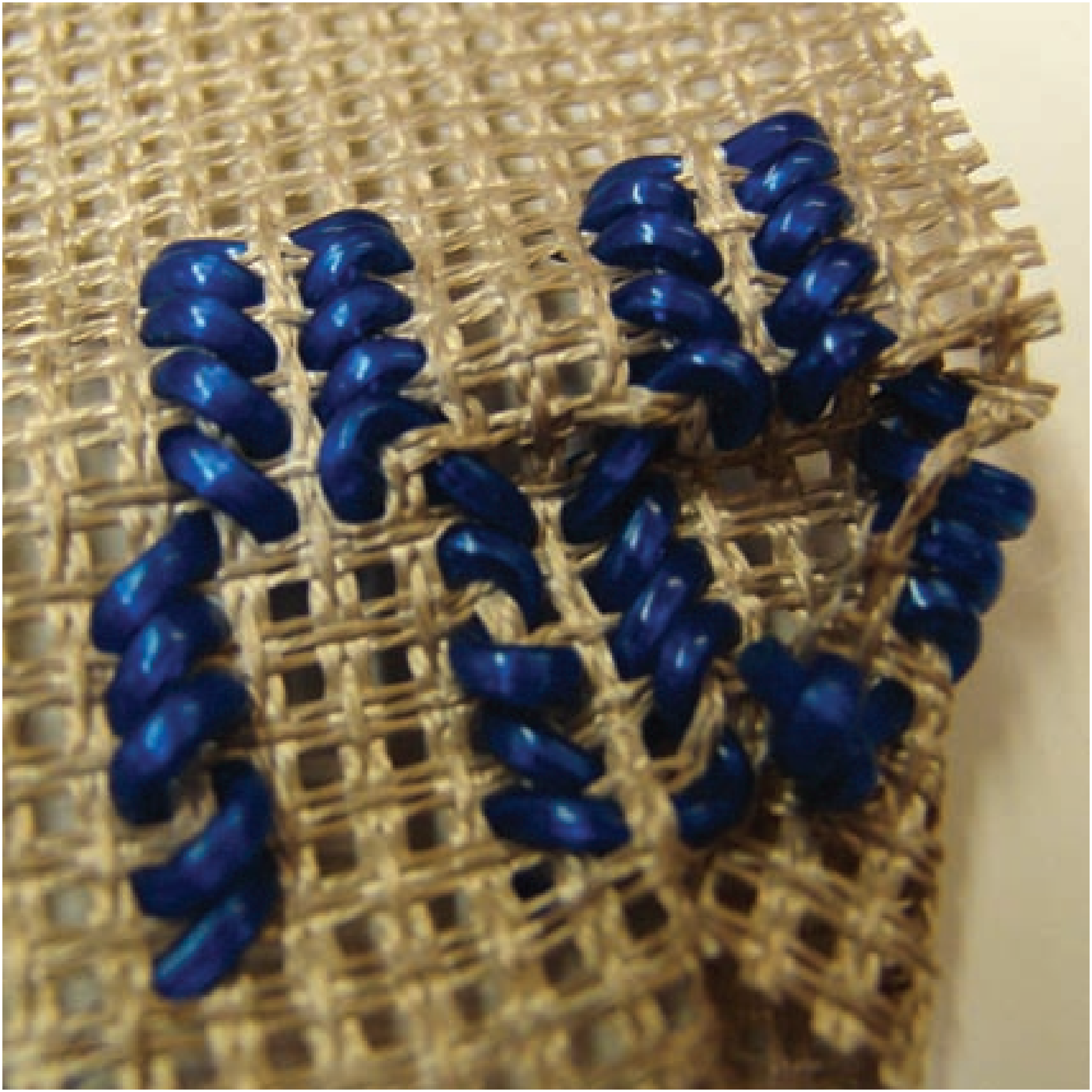
Using this potential to contribute to the exploration of urban use of interactive textiles was the choice for this years MAS students, to combine and take further their experiences in structures, programming and electronics.

“Electroluminescence (EL) is an optical phenomenon and electrical phenomenon in which a material emits light in response to an electric current passed through it, or to a strong electric field.”

At the end of 90’s the company Lytec developed a product that uses EL inside a coaxial cable. When connected from only one end to a AC (alternating current) power source , the wire emits light. The wire can have different thicknesses and recently it was also made available in 0.9 mm fibers, which makes it possibly suitable for using it inside textiles. Therefore EL wire is chosen as a light source for the visual output of the interaction.



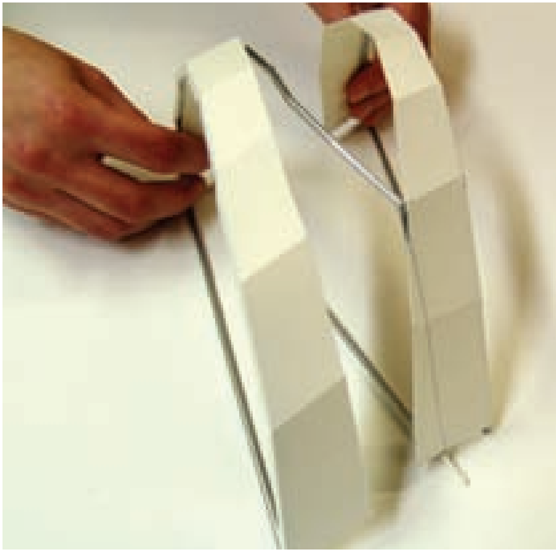




FORM

The process of form finding was realized after a long consideration of which physical properties were being exhibited by the surface material. Since there had been no prior works incorporating EL wire in such an integrated way - as a necessary component of the surface construction - the project matured through a process of form consideration and constant discovery of the evolving conditions and possibilities which were realized in the surface itself.

Since the function of public shelter or pavilion was a strong desire for the final project, a certain amount of freedom was maintained since the greatest amount of information to determine the form would also be derived from the capabilities of the EL wire / substrate weave combination. Ideas ranging from tension structures to free-standing frameworks were investigated, and due to certain conditions at the proposed site, a free-standing concept was developed which was able to use the material as a critical system component.



What resulted was a composite between a tension structure and a frame structure- an arched passageway using the knitted surface for lateral support.

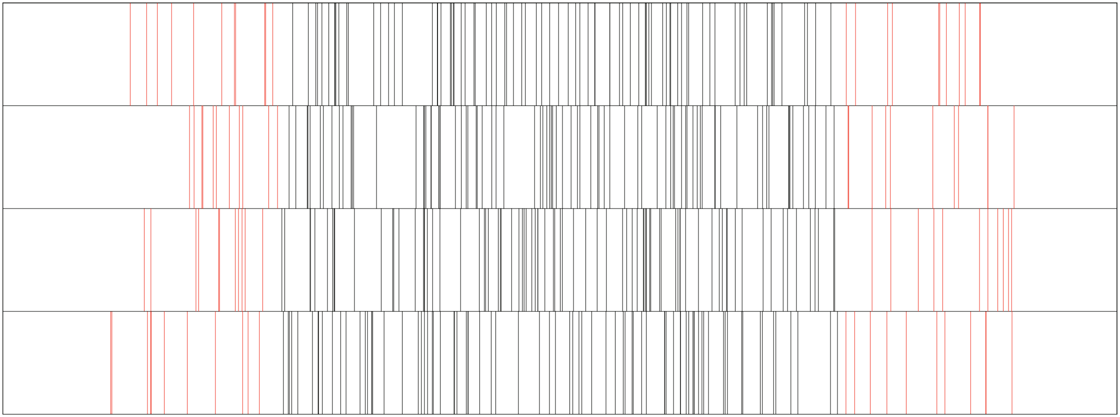
After many steps of investigation and collaboration for the surface construction, certain opportunities (and some minor limitations) presented themselves.

At many moments, considering the novel concept of the surface construction, and after many prototypes, we learned that certain surface panel shapes were not lending themselves well to the production machinery. This helped us to develop a surface system comprised of straight rectangular strips which still afforded us wide possibilities for patterning and programming the active surface, while speeding production.

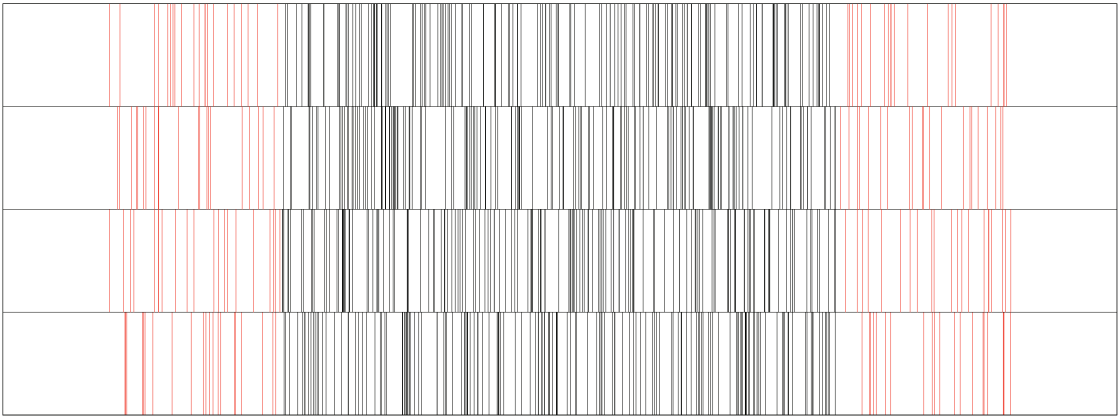
```
Call Main()
Function GetSurfaceControlPoints(strSurface)

            '----create border polyline and points at this border line----
            Dim strPolyline
            Dim arrPoints
            Dim arrBorder

            arrBorder= Rhino.DuplicateSurfaceBorder(strSurface)
            strPolyline= arrBorder(0)
            arrPoints= Rhino.CurvePoints (strPolyline)

            GetSurfaceControlPoints= arrPoints
End Function

Function DrawBorderLines(arrPoints, intSides)
            '----draws lines between border edge points----
            Dim arrLine
            Dim i
            ReDim arrLine(intSides)
            For i=0 To intSides-1

                        arrLine(i)= Rhino.AddLine(arrPoints(i), arrPoints(i+1))
            Next

            DrawBorderLines= arrLine
End Function

Function DrawStripes(arrCurve, intSeg)

            Dim arrVec
            Dim arrBack
            Dim arrsBack
            Dim i, j
            Dim arrStartPt, arrEndPt
            Dim arrVecStep
            Dim dblDistance
            Dim intCount
            Dim strCurve
            Dim arrDivLines
            ReDim arrDivLines(intSeg)
            ReDim arrBack(intSeg)
            ReDim arrsBack(2)

            '----pick the border lines to create the stripes----
            arrCurve= Rhino.GetObjects("Pick the nicest lines to draw the segments: ", 4)
            If IsArray (arrCurve) Then
                        '----count the number of stripes----

                        For Each strCurve In arrCurve
                                    intCount= intCount+1
                                    Rhino.Print("intCount: " & intCount)
                        Next
            End If

            For i=0 To intCount-1

                        arrStartPt= Rhino.CurveStartPoint(arrCurve(i))

'                       If IsArray (arrStartPt) Then

'                                   Rhino.Print("hello! i'm start point ")

                        '                       End If
                        arrEndPt= Rhino.CurveEndPoint(arrCurve(i))
'                       If IsArray (arrEndPt) Then
                        '                       End If

                        arrVec= Rhino.VectorSubtract(arrEndPt, arrStartPt)

'                       If IsArray (arrVec) Then
'                                   Rhino.Print("I'm an array ")
                        '                       End If

            'dblDistance= Rhino.Distance(arrEndPt, arrStartPt)

            'Rhino.Print("my distance: " & dblDistance)
                        arrVec= Rhino.VectorDivide(arrVec, intSeg)
                        '                       If IsArray (arrVec) Then

                        '                                   Rhino.Print ("i am a beautifull array...")

                        '                       End If

                        For j=0 To intSeg
                                    arrVecStep= Rhino.VectorScale(arrVec, j)

                                    arrBack(j)= Rhino.PointAdd(arrVecStep,arrStartPt)
```

PROGRAMMING THE PATTERN

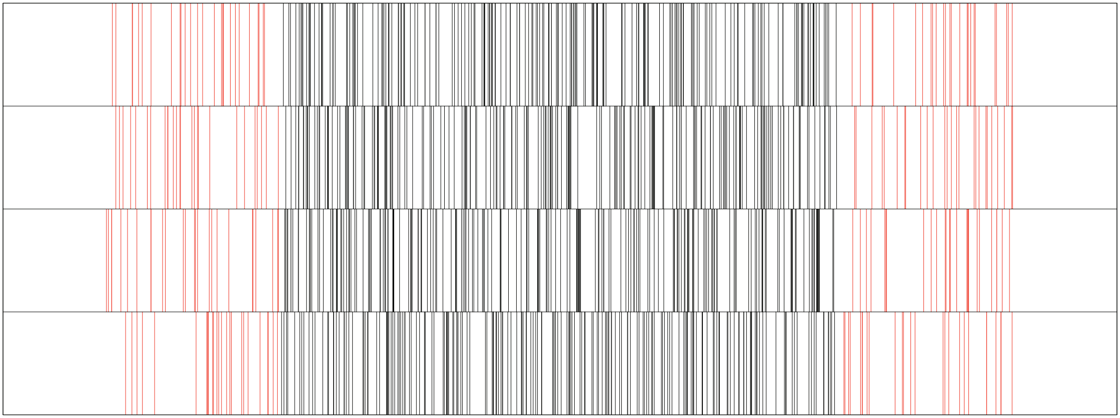
A simple code, in Rhino script, that gives the light pattern. It divides the general form in areas with different colors and creates a pattern with random lines in different density. The programmed pattern is an interesting and simple effect that results from the machine’s limitations and possibilities. Through the simplicity of the light pattern, accent is given to the interaction and the form. The code, also, calculates the length and the cost of the EL wires.



Total lines: 400 ,Meters of El-wire: 520



Total lines: 800 ,Meters of El-wire: 1040



Total lines: 1000 ,Meters of El-wire: 1300

Total lines: 1200 ,Meters of El-wire: 1560

KNITTING

The topology of a knit fabric is relatively complex. Unlike woven fabrics, where strands usually run straight horizontally and vertically, threads that have been knit follow a loopy path along their rows.

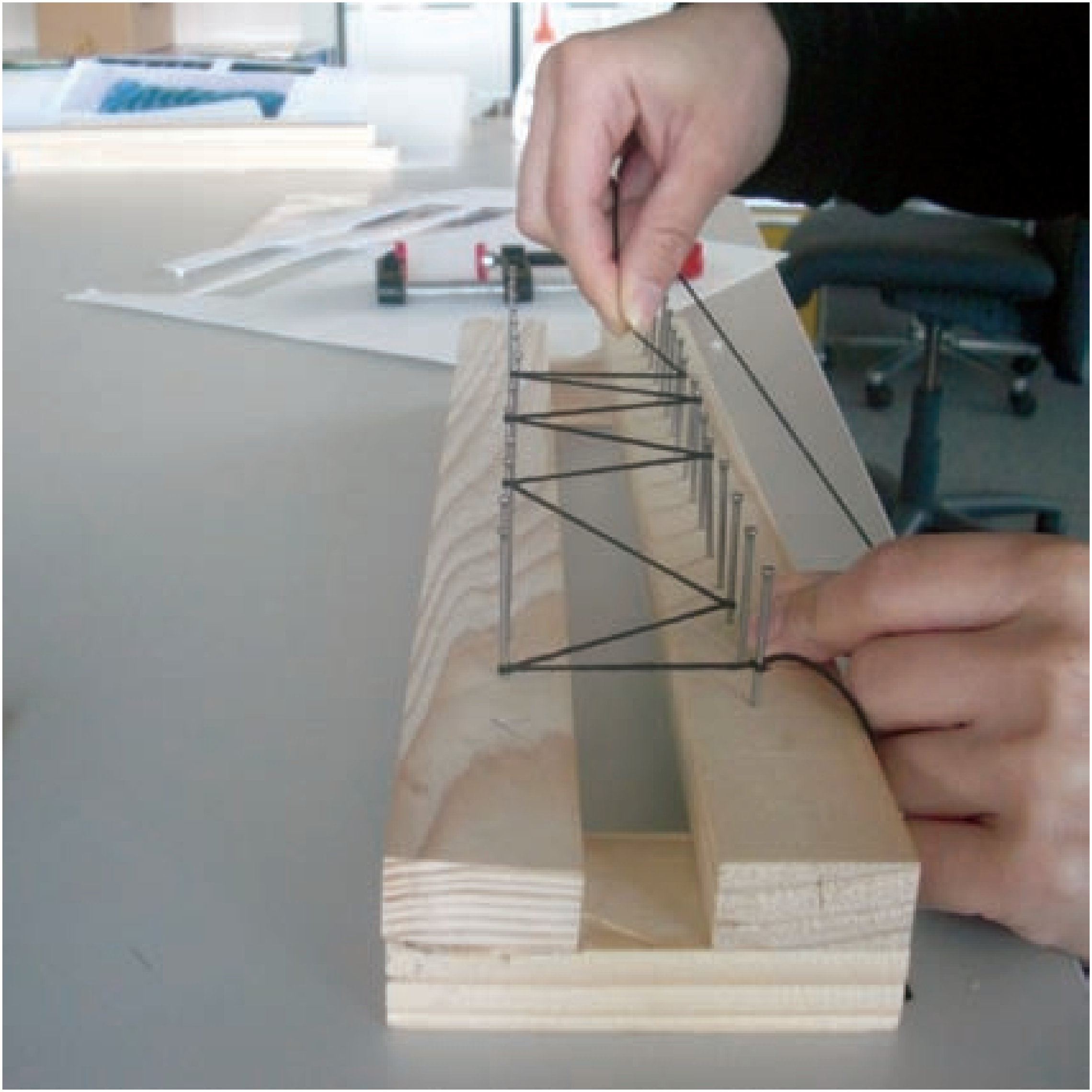
Because there is no single straight line of yarn anywhere in the pattern, a knit piece will be stretchy in all directions (some more than others, depending on the specific yarn and the specific pattern). This stretchiness is unavailable in woven fabrics.

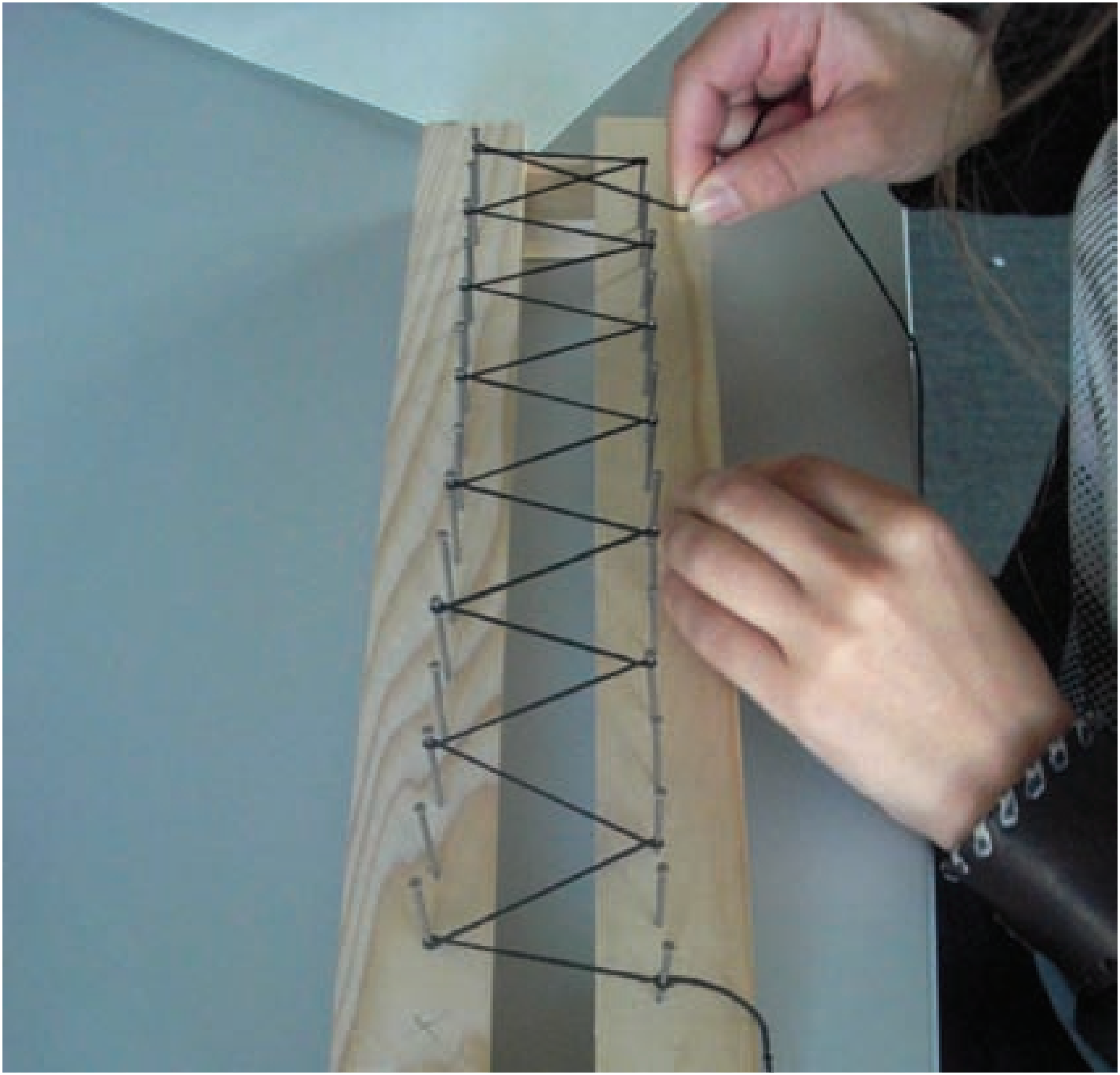
A piece of knitting begins with the process of “casting on”, which involves the initial creation of the stitches on the needle. The body of a knitted piece may include plain stitches or a number of color and textured patterns. The number of active stitches remains the same as when cast on unless stitches are added (an increase) or removed (a decrease) to shape the item. Once the knitted piece is finished, the remaining live stitches are “cast off”, where the stitches are made across each other so they can be removed from the needle without unravelling the item.

Typically, the final knitted garment will be made of several knitted pieces, with individual sections of the garment knit separately and then sewn together once all the pieces have been completed. Seamless knitting, where a whole garment is knit as a single piece is also possible.

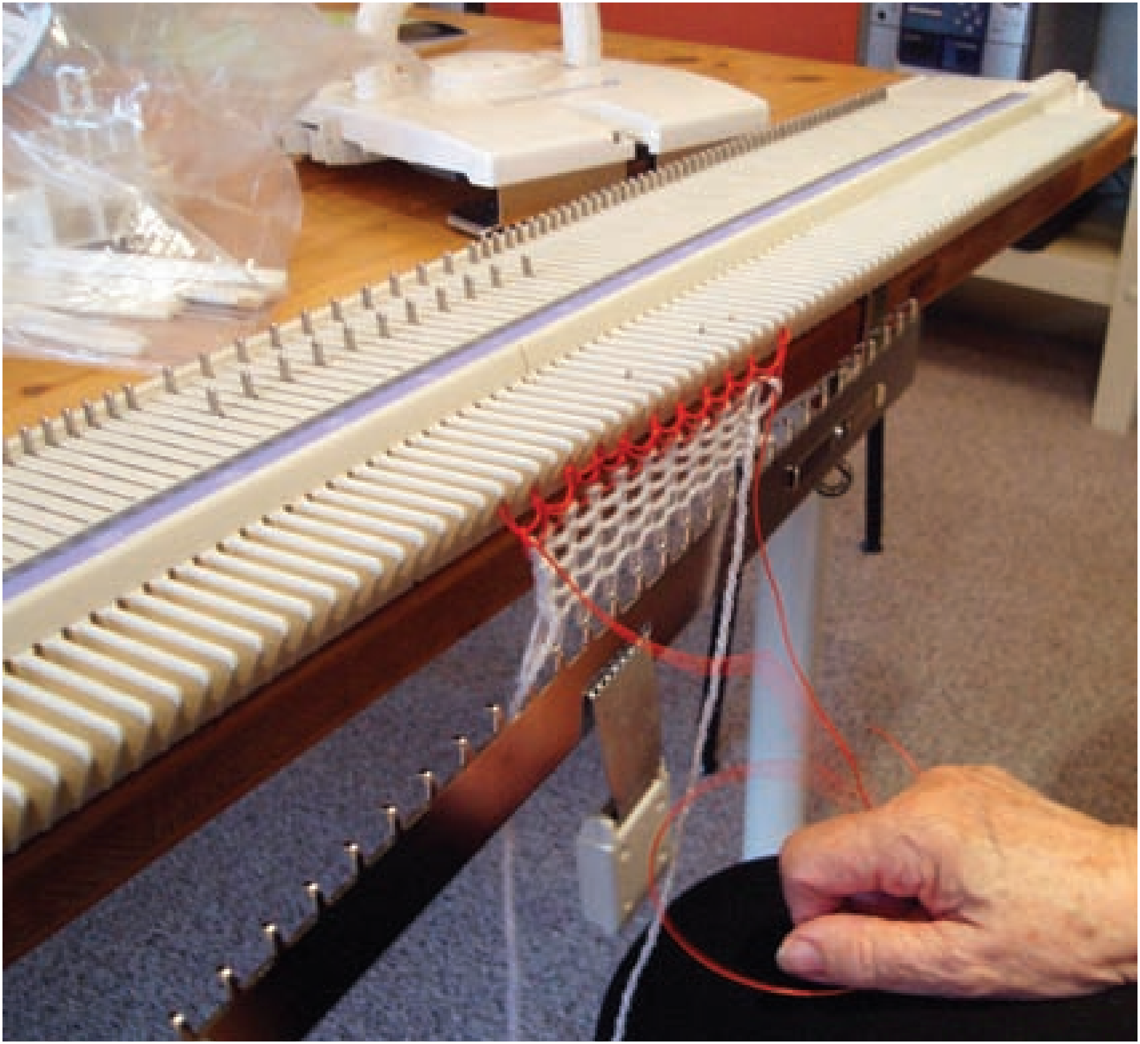
For our purposes, we were interested to incorporate a digital production method to knitting, whether by only through the generation of the pattern, or ultimately to feed data to a production-scale industrial knitting machine with full computer controls.

In our experimentation with various methods, we created knitting boards for experimenting with knitting and to gain knowledge of the basic properties of our EL material in this context. We also met with local shop owners who operate home-scale table top knitting machines, to gather information and benefit from the expertise of people who have worked with machine-based knitting and to gauge the interest of these people in a technical project such as ours to judge its potential.





ANALOG TABLE TOP KNITTING MACHINE




CMS 530 HP
STOLL

MACHINE KNITTING

A few simple devices permit knitting without needles for toy or hobby purposes. The simplest of these is spool knitting, followed by “Knitting Boards” or “Knitting Looms”, which consist of two rows of pins mounted in two parallel rows approximately half an inch apart. Yarn is wound around the pins; various patterns of winding produce different textured knitting. A needle or special tool is then used to transfer the loops of yarn from around the pins, either off the pins or to other pins, to produce the knitting. Knitting Boards can produce complex designs.

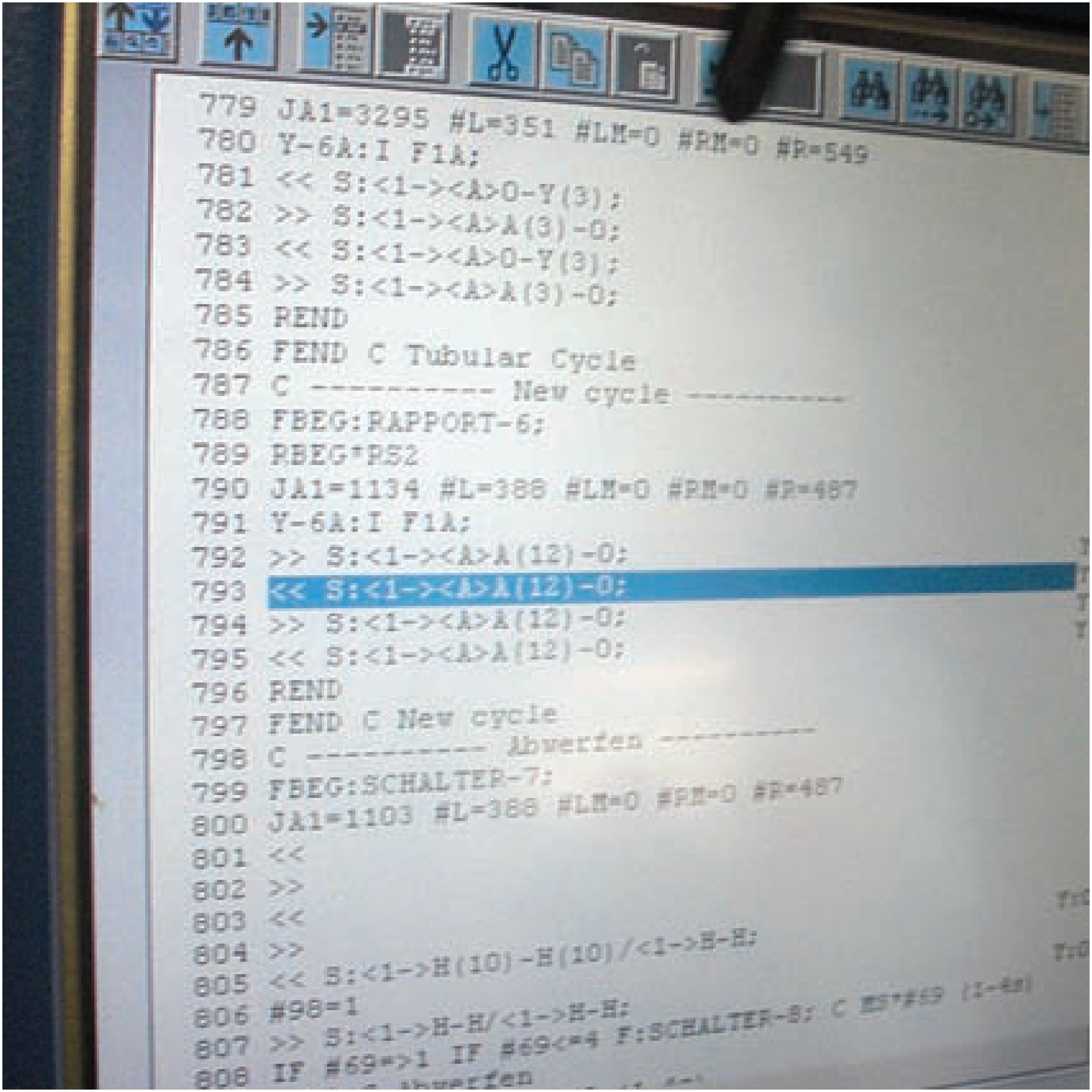
To produce larger and more complex knitted items, such as garments, domestic and industrial machines, with either flat or circular beds that produce rectangular or tubular fabrics are needed. Double bed machines have two flat beds facing each other, in order to produce purl and plain rib fabrics plus a variety of multi patterns. Ribbing attachments can be added to single bed machines to achieve a similar result.

Domestic or studio models typically use up to 200 latch hook needles. A carriage or cam box is passed across the bed of needles causing the needle movements required to produce each next stitch. By means of various selection methods, e.g. punch cards, particular needles can be caused to travel by alternate pathways through the cam box. In the most modern and industrial-scaled machines, punch cards have been replaced by full computer control.

We were fortunate that we made an early contact with Stoll GmbH and Company in Reutlingen, Germany. Their company is a world leader in machine knitting technology, especially in the context of full computer controlled garment and textile production.

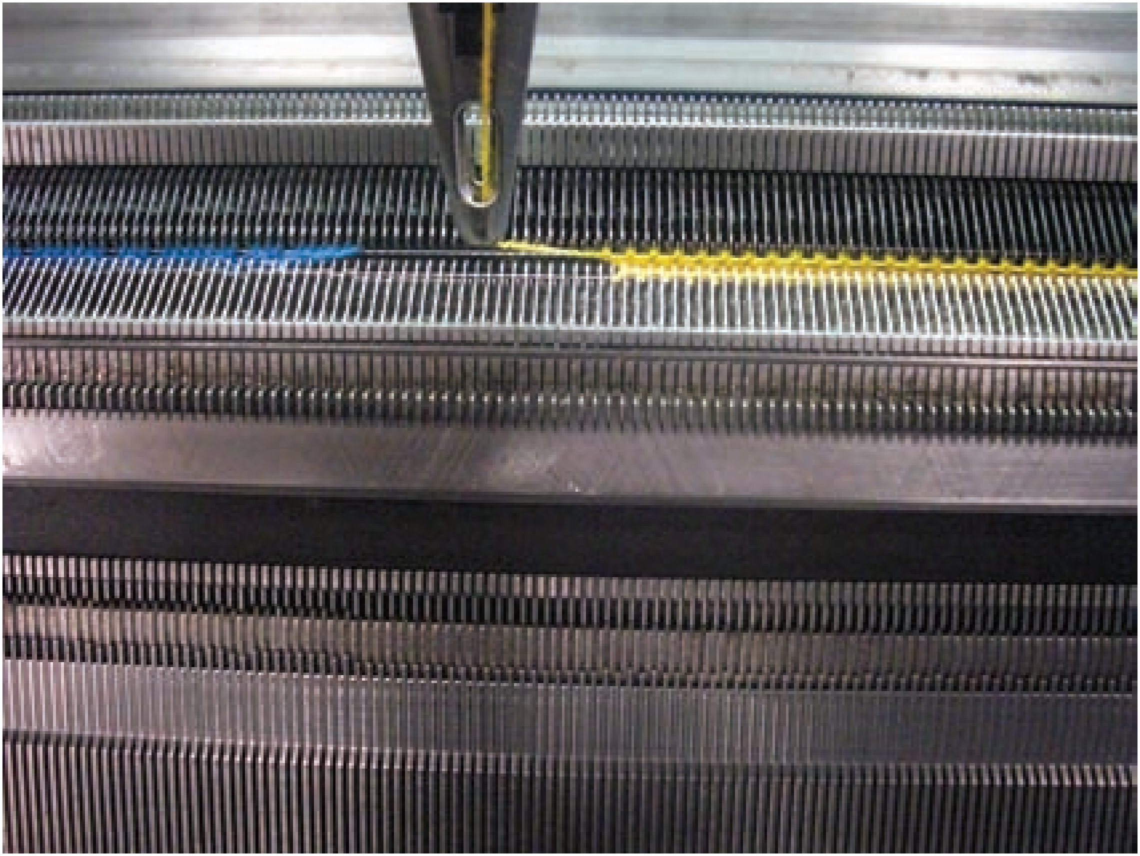
After many meetings and collaborations, Stoll agreed to assist us with the production and preliminary testing with our EL wire and their CNC knitting machines. Testing was extremely informative and beneficial, leading us to a practical and interesting range of solutions for creating our EL wire surface.

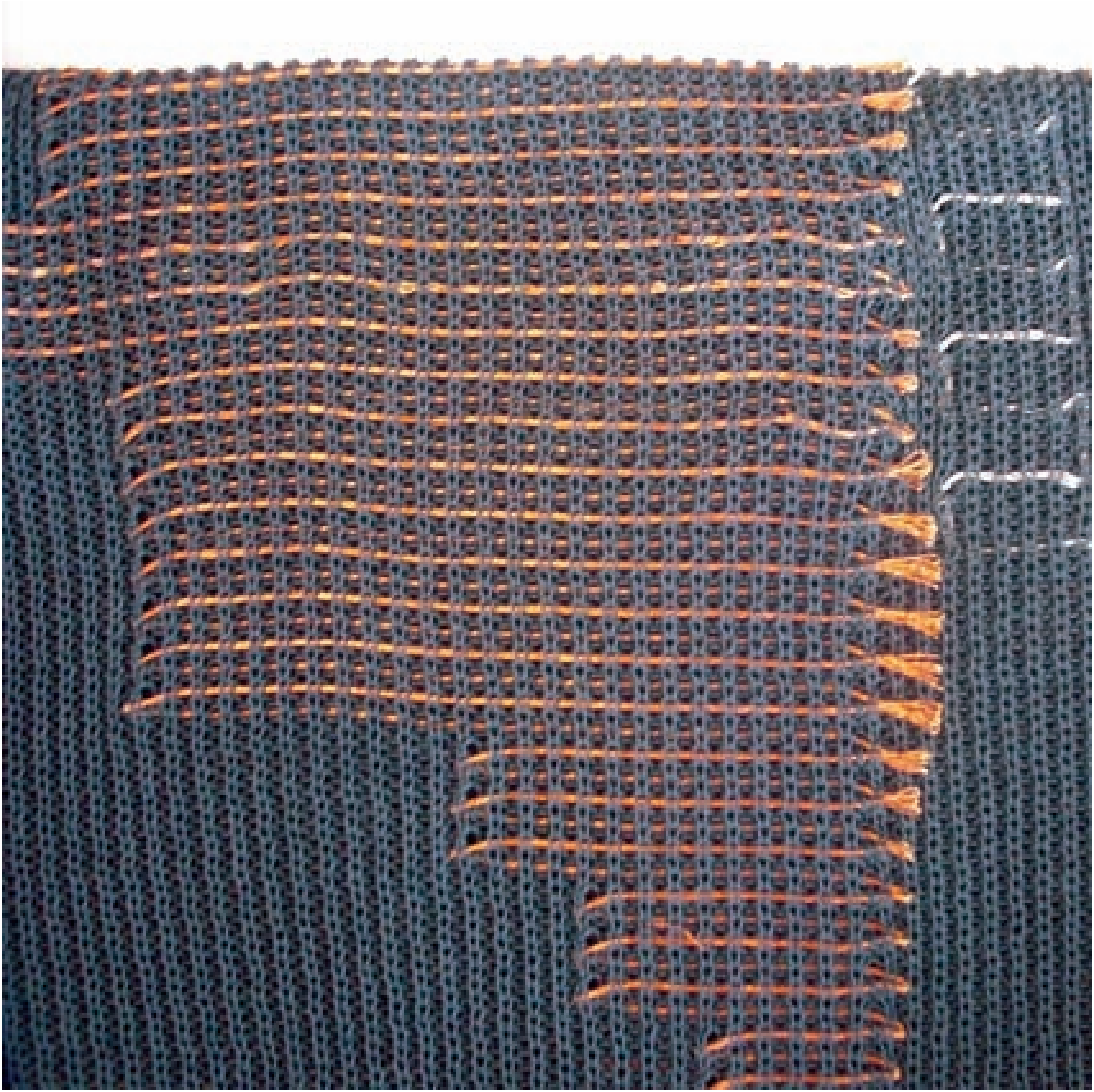
Stoll was instrumental in testing and suggesting various materials for a substrate or background knit matrix, based on our design goals, and after a short integration period, was able to begin producing our patterned pieces of EL textile from our pattern data which was produced by computer scripting developed in our group, and translated into Stoll’s machine data by experienced members of their Technical Fabrics Division.

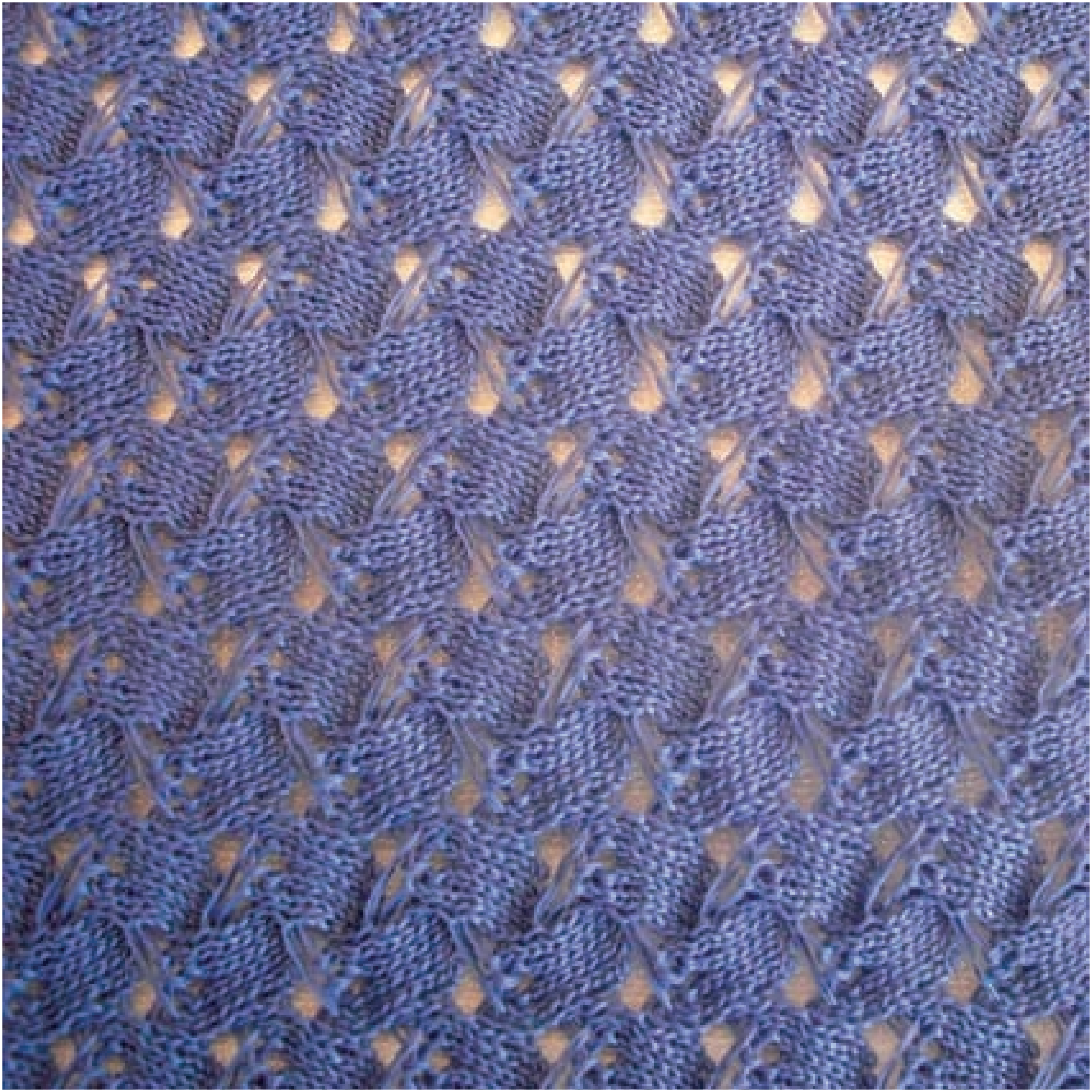

779 JA1=3295 #L=351 #LM=0 #RM=0 #R=549
780 Y-6A:I F1A;
781 << S:<1-><A>0-Y(3);
782 >> S:<1-><A>A(3)-0;
783 << S:<1-><A>0-Y(3);
784 >> S:<1-><A>A(3)-0;
785 REND
786 FEND C Tubular Cycle
787 C ---------- New cycle ----------
788 FBEG:RAPPORT-6;
789 RBEG*RS2
790 JA1=1134 #L=388 #LM=0 #RM=0 #R=487
791 Y-6A:I F1A;
792 >> S:<1-><A>A(12)-0;
793 << S:<1-><A>A(12)-0;
794 >> S:<1-><A>A(12)-0;
795 << S:<1-><A>A(12)-0;
796 REND
797 FEND C New cycle
798 C ---------- Abwerfen ----------
799 FBEG:SCHALTER-7;
800 JA1=1103 #L=388 #LM=0 #RM=0 #R=487
801 <<
802 >>
803 <<
804 >>
805 << S:<1->H(10)-H(10)/<1->H-H;
806 #98=1
807 >> S:<1->H-H/<1->H-H;
808 IF #69=>1 IF #69<=4 F:SCHALTER-8; C MS*#69 (1-4)

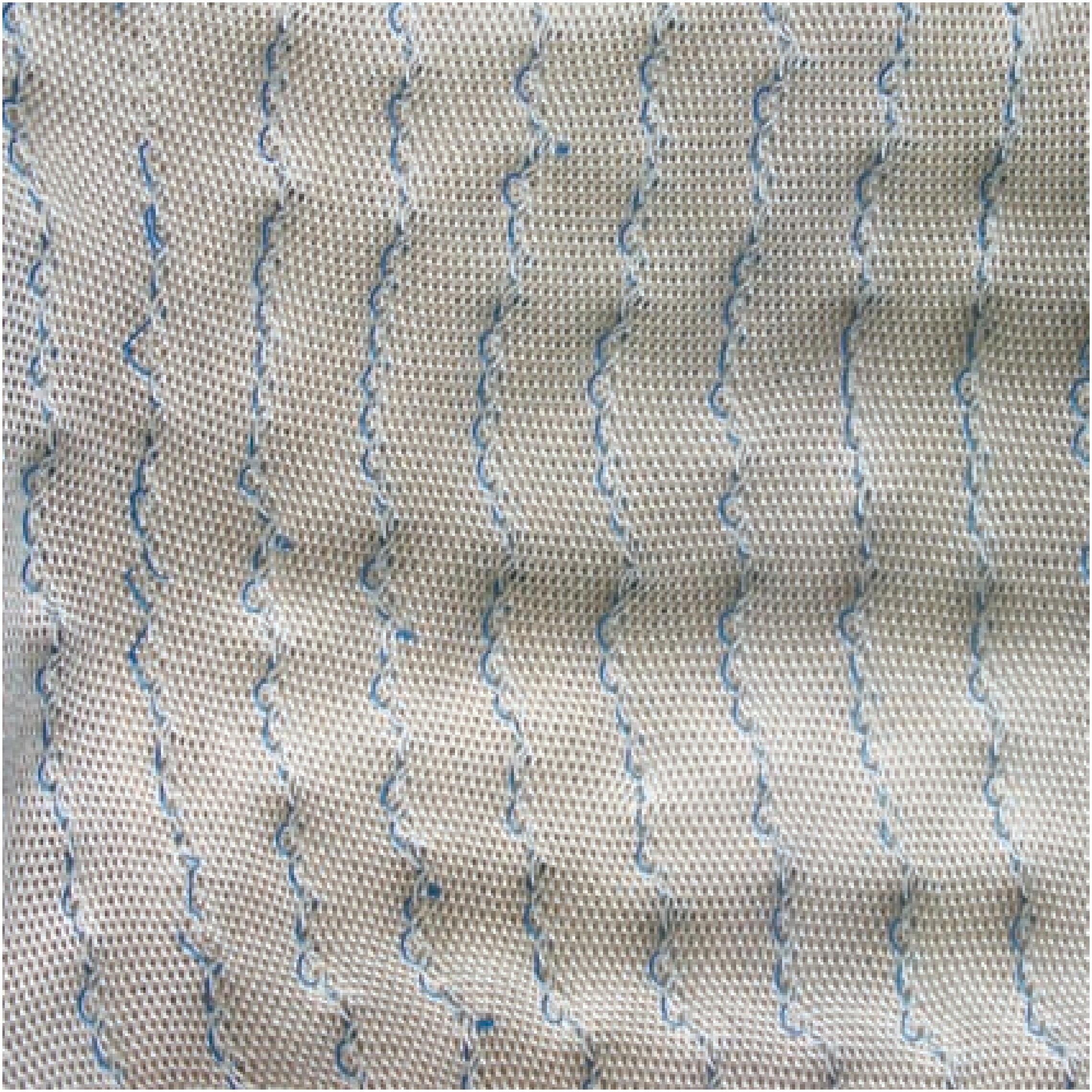
The Stoll machine uses a double-bed configuration, where controllers pass threads (and our EL wire) to needle beds on either side of the knitting process, producing a single sheet. All fibers are incorporated during a single pass of the machine down and back across the needle bed.

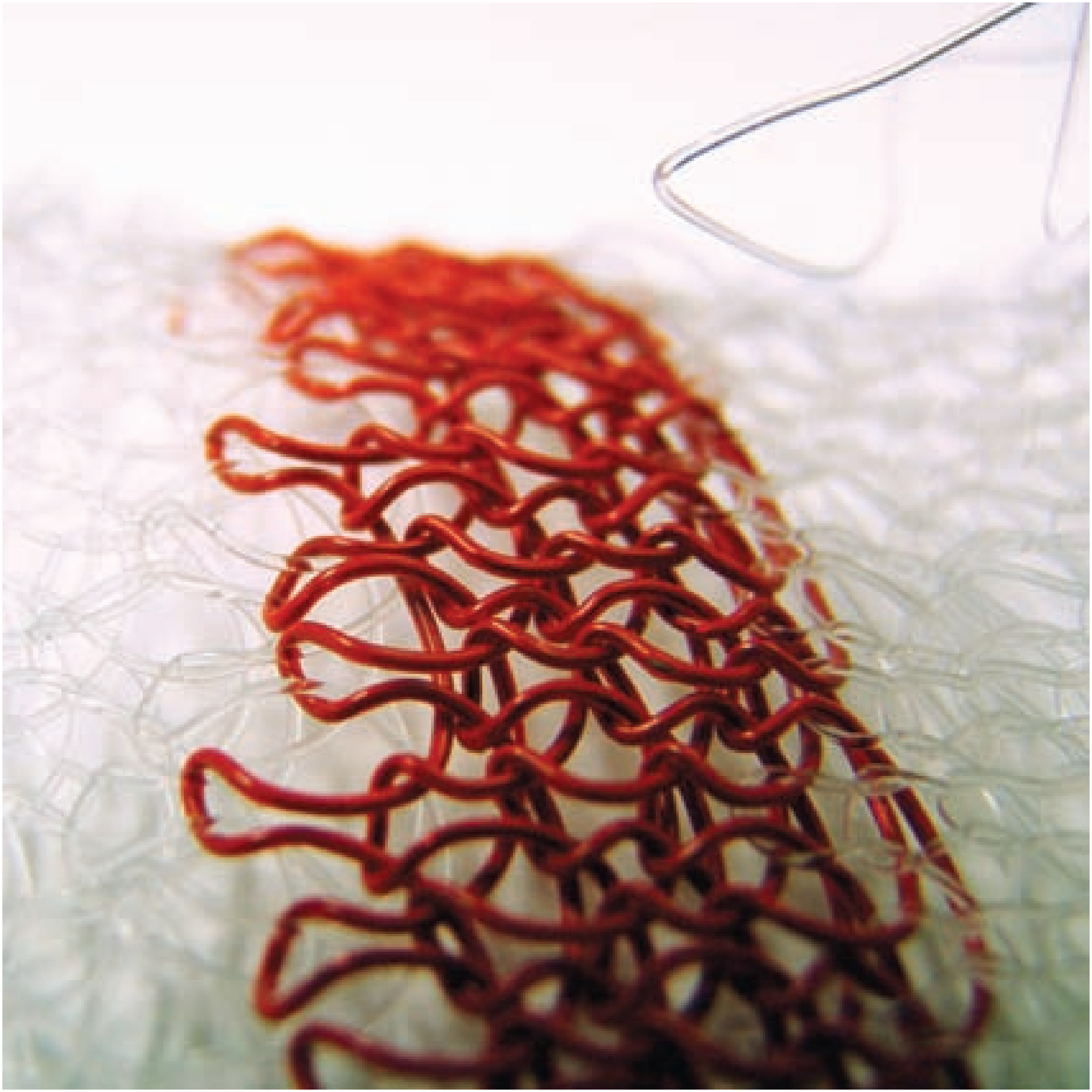
While considering special conditions with our EL material, namely its inherent need to be treated delicately while being bent, the Stoll team was able to develop a very rapid and careful process which was very well suited for our material, design goal, and delivery schedule. It is the hopes of our team that future research will be carried out by Stoll to further develop this technique, and foster continued collaborations between Industrial and Educational partners.

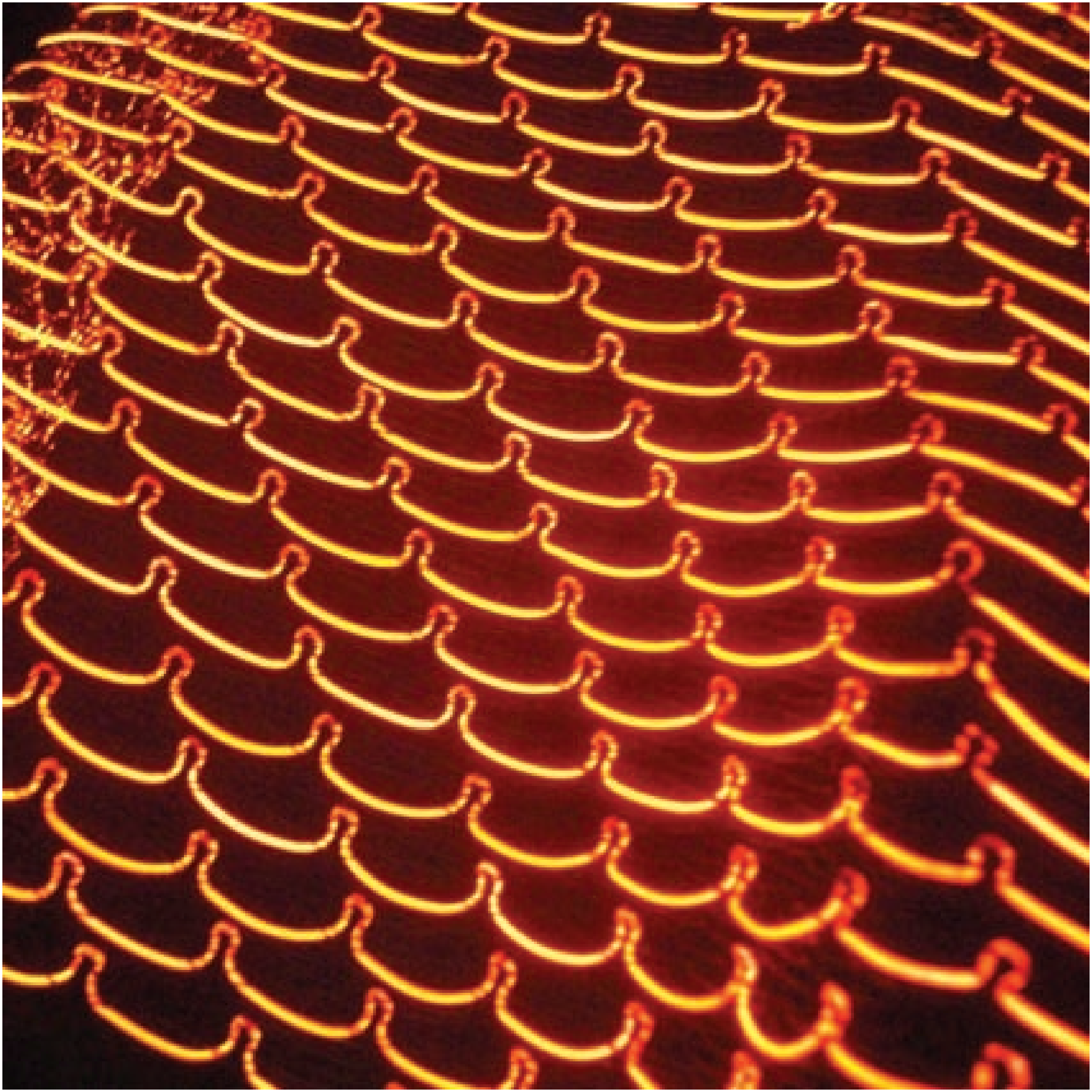


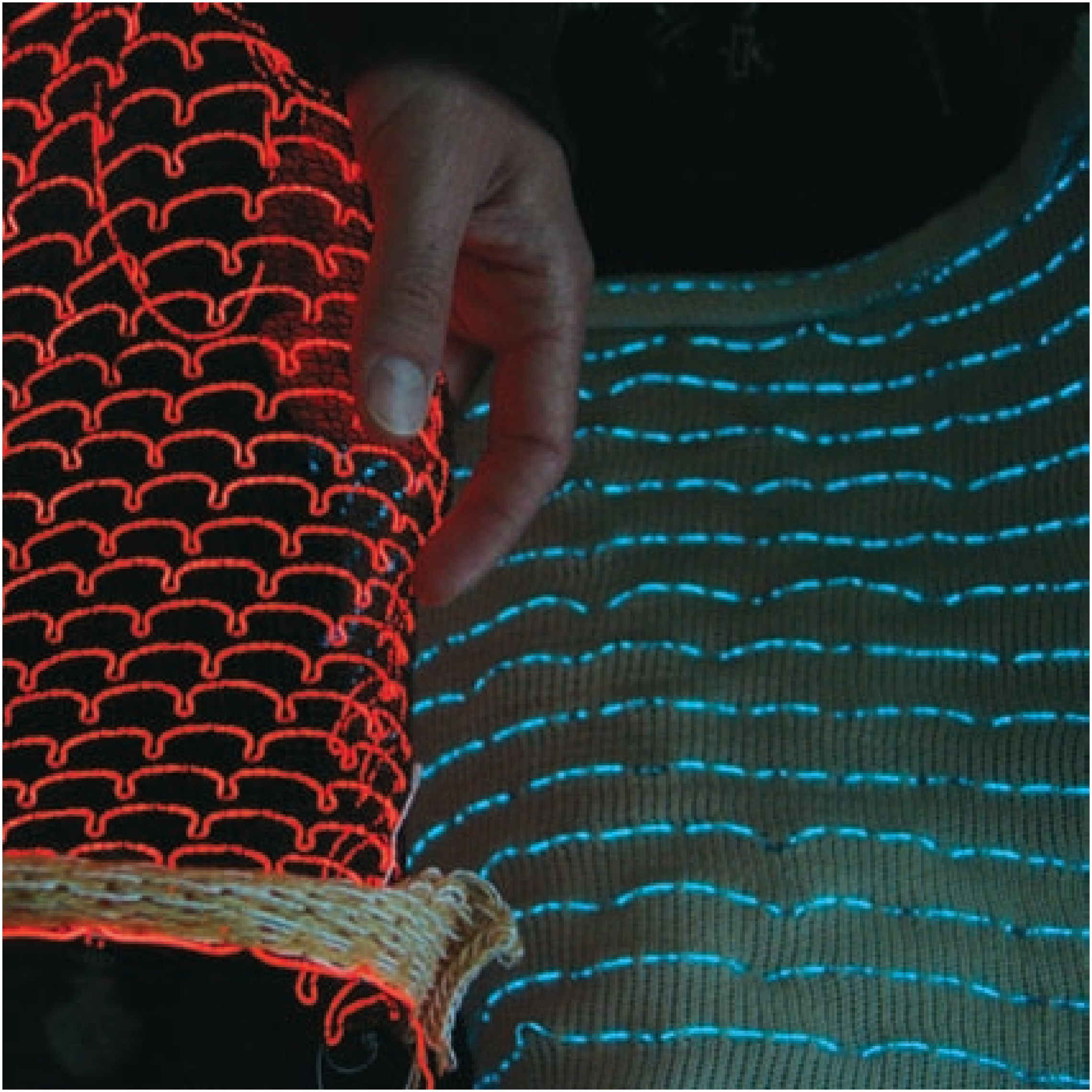


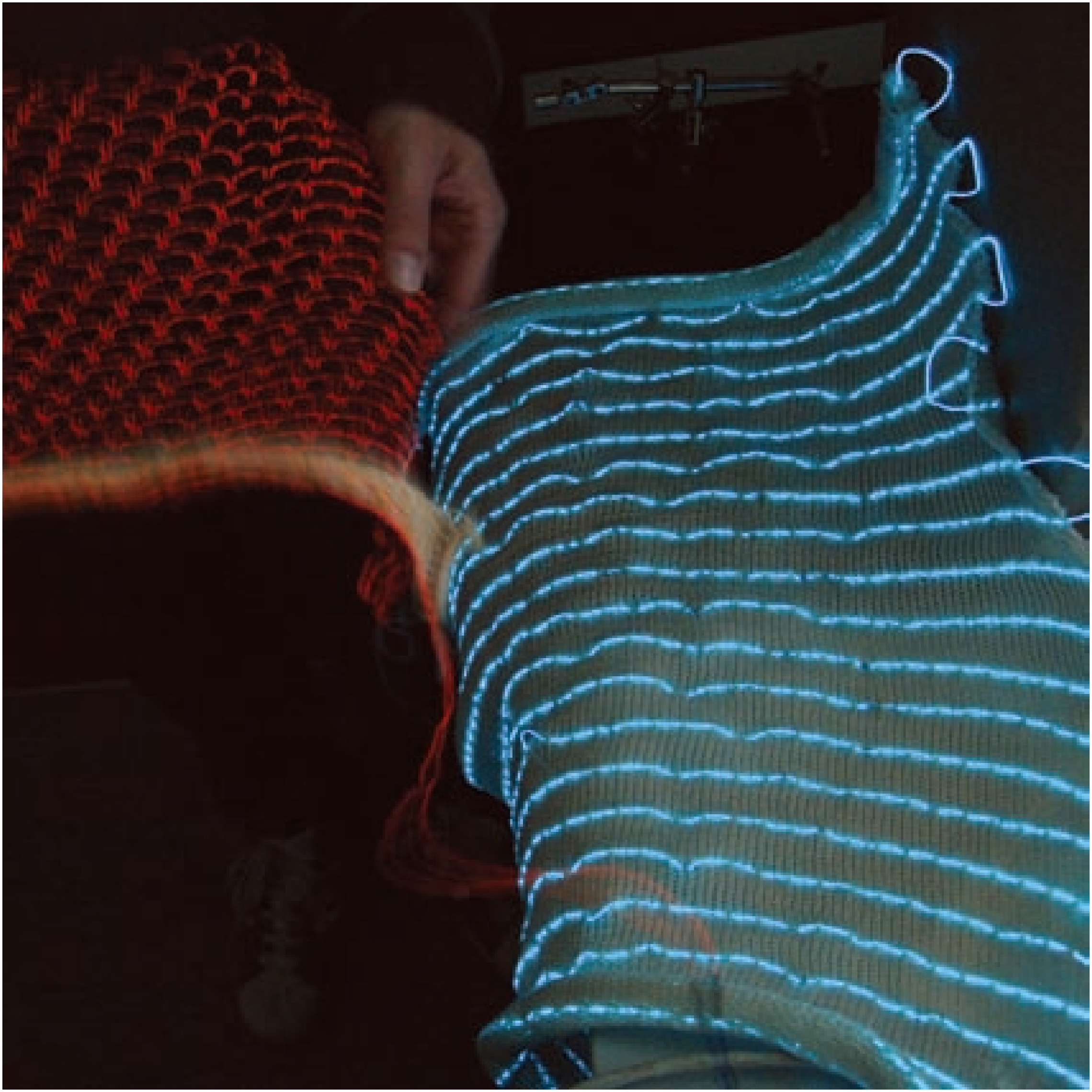


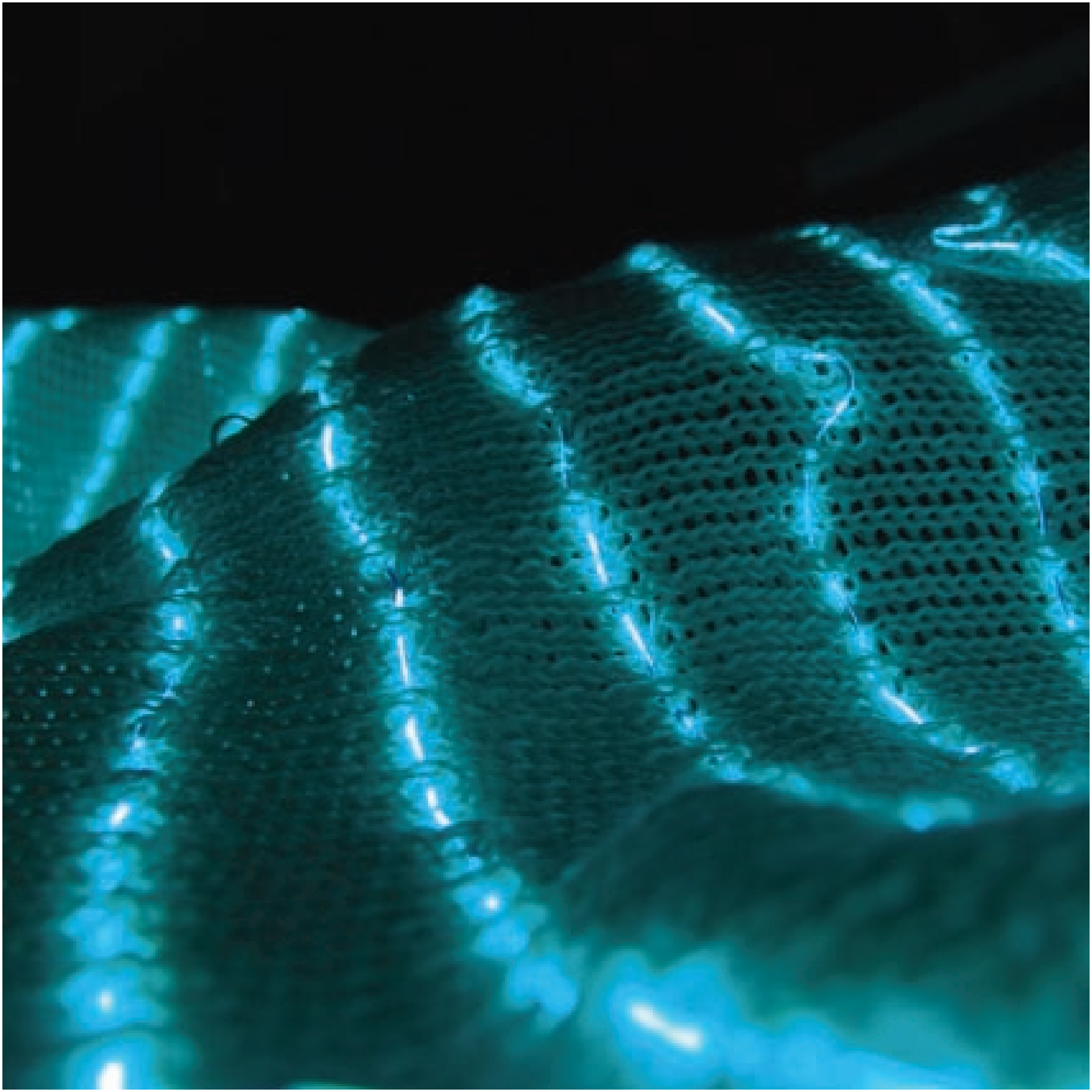


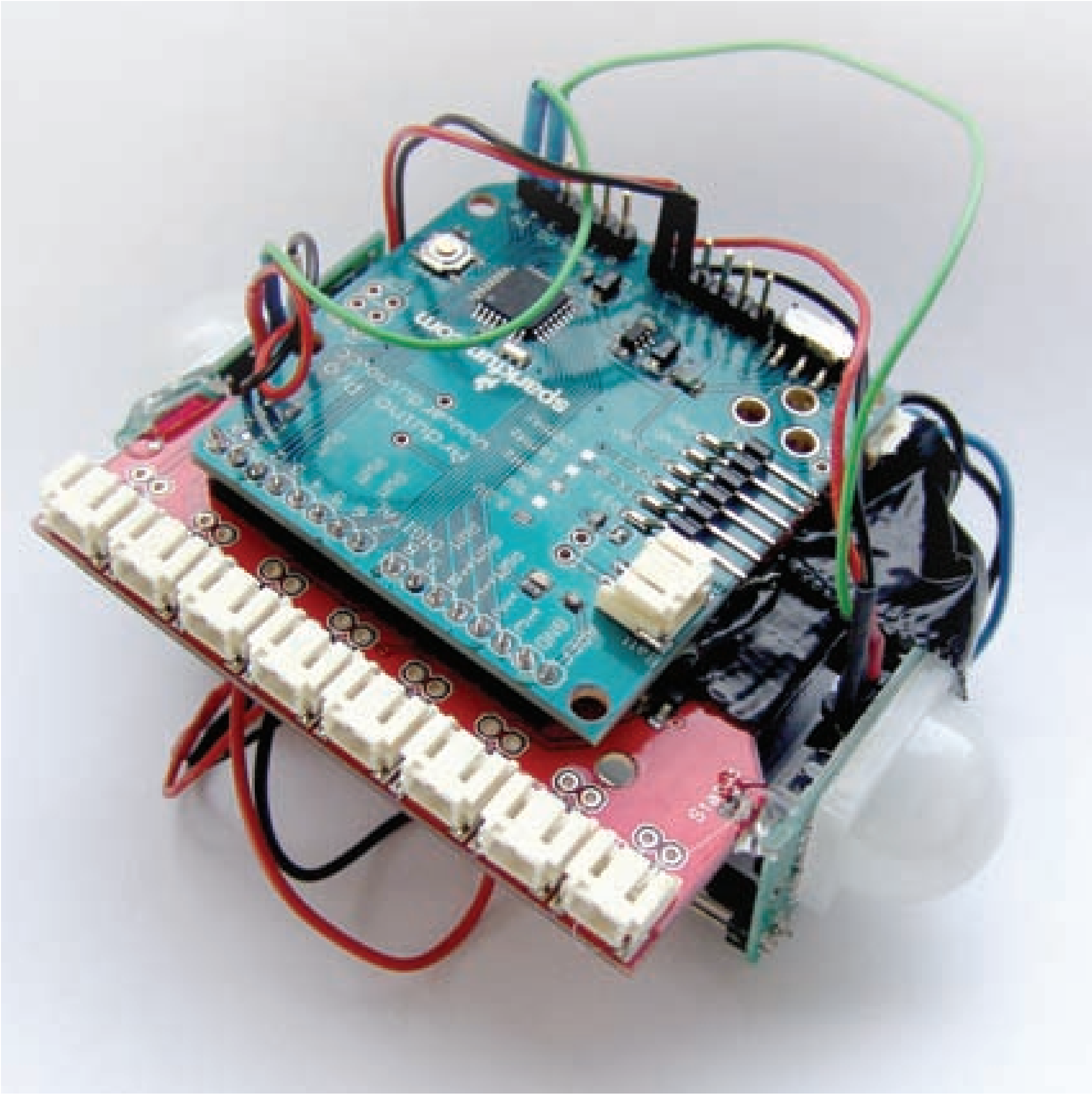








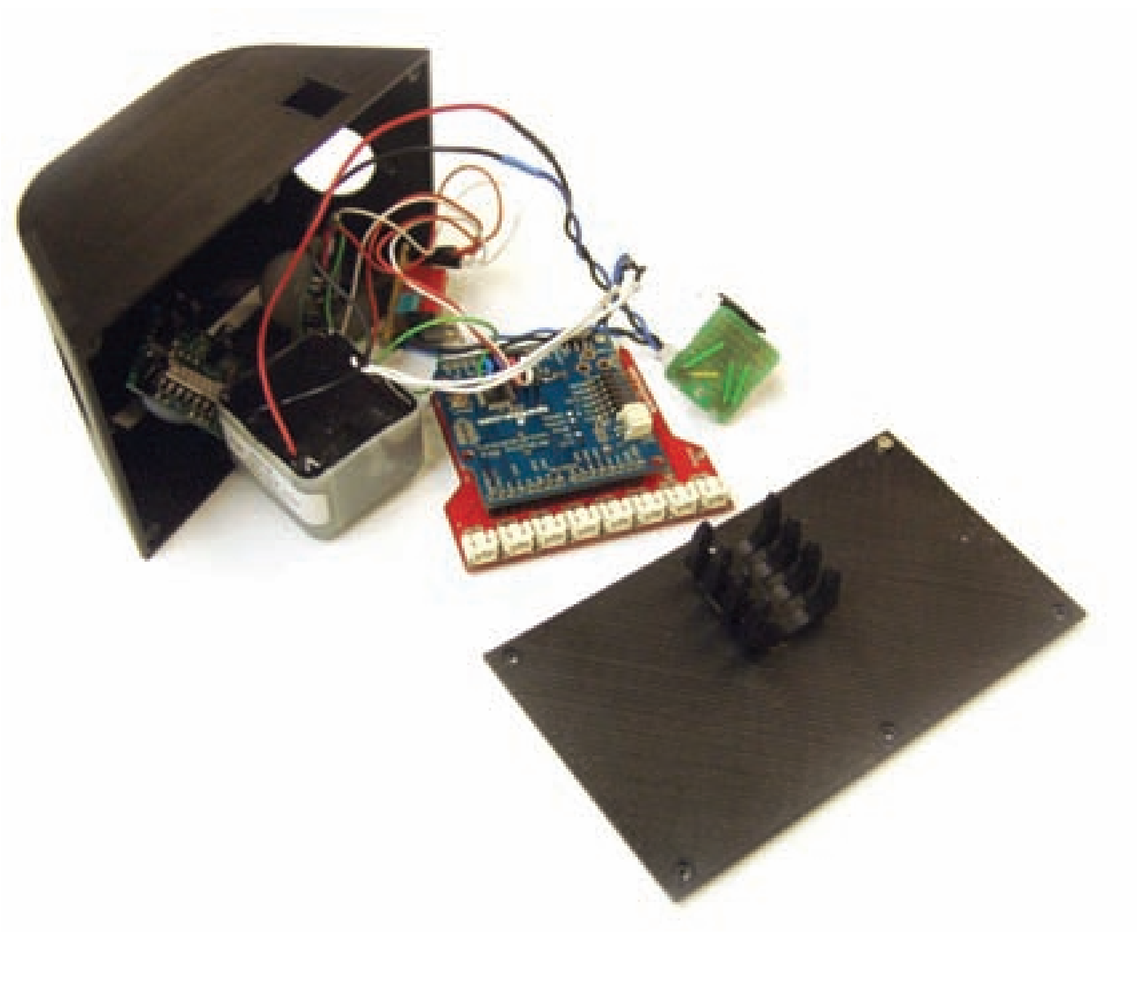




PHYSICAL COMPUTING

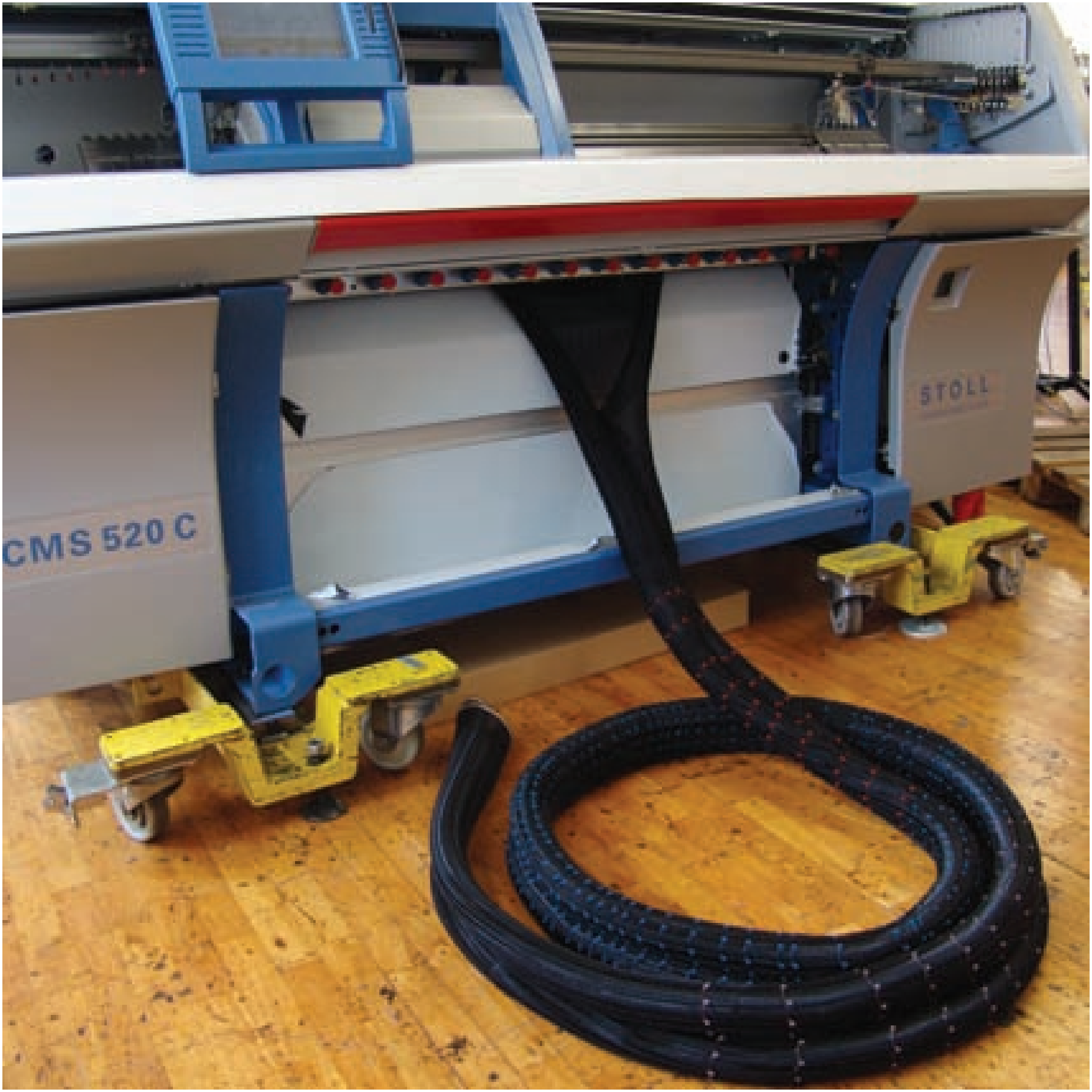
One of the most important parts in the project is the physical computing part, which includes building the electronic infrastructure, designing a box and programming the microcontroller, i.e. the lighting sequence of the lights based on the environmental input.
To achieve this, two PCBs (Printed Circuit Board) and other components are used to process the information from PIR motion detectors, sequence the lights according to given pattern and to power the instruments.
1.Arduino Pro (Sparkfun); This microcontroller has a Atmega 128 chip inside and has multiple input and outputs. The digital information received from two PIR sensors are processed here, to give output to the sequencer board.
2.EL Escudo , sequencer (Sparkfun) ; This module is recently developed by Sparkfun industries to make it easy to interface Arduino with EL wire, which uses AC (alternating current) voltage to run. The main problem with sequencing EL wire is that the Arduino board uses DC (direct current) power to switch on and off the digital outputs, so there is a need to have a device in between to trigger on and off the AC input of the EL wire. This is the component called TRIAC (triode for alternating current), which is embedded in the design of the EL Escudo board.
3.Inverter is a component to convert DC to AC, this is used only to light up the EL wire. It lights up while the current is passing through TRIACs , when the gate of the TRIAC is triggered by the digital output from Arduino microcontroller.
The inverter used here functions with 12V DC power and converts it into 120 V AC.
4.PIR motion sensors; two sensors are used to detect movement from two opposite sides. There is 4 different cases for each box: left on, right off; left off, right on; both off; both on. There is different light patterns for every different case.

“PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don’t wear out. For that reason they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, “Passive Infrared”, “Pyroelectric”, or “IR motion” sensors.”



SENSOR BOX

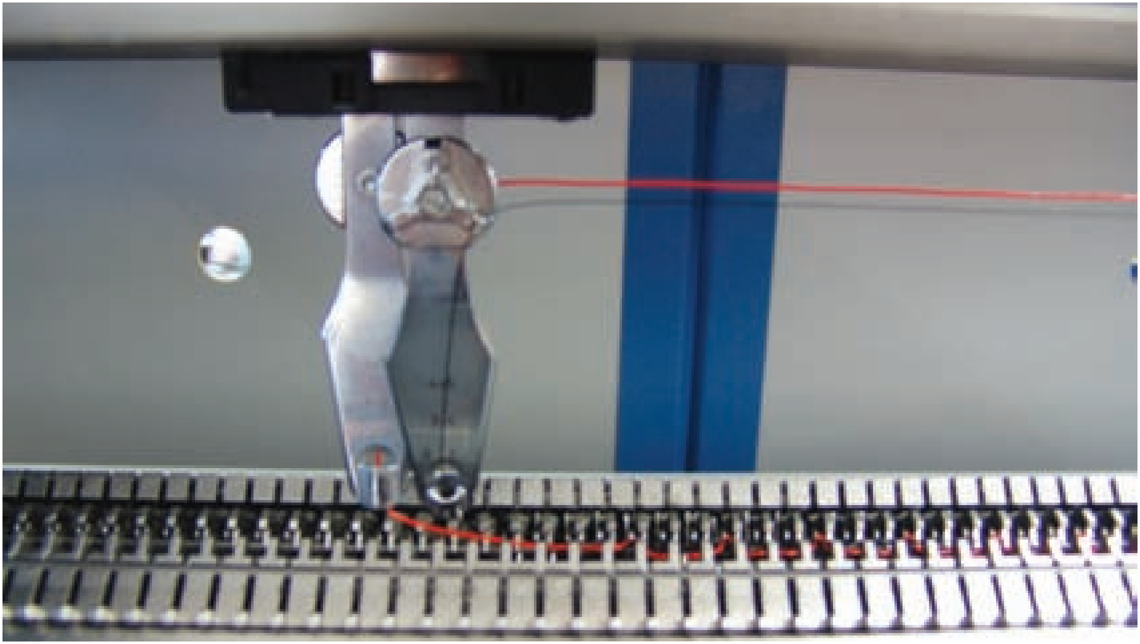
Boxes are designed and printed with the 3D printer, they provide protection against physical effects from the environment. Additionally the grips on the backside of the box make it easy to attach the box to the structure. It is made in two separate parts, the lid and the body, which are screwed together through the screw holes included in the design.

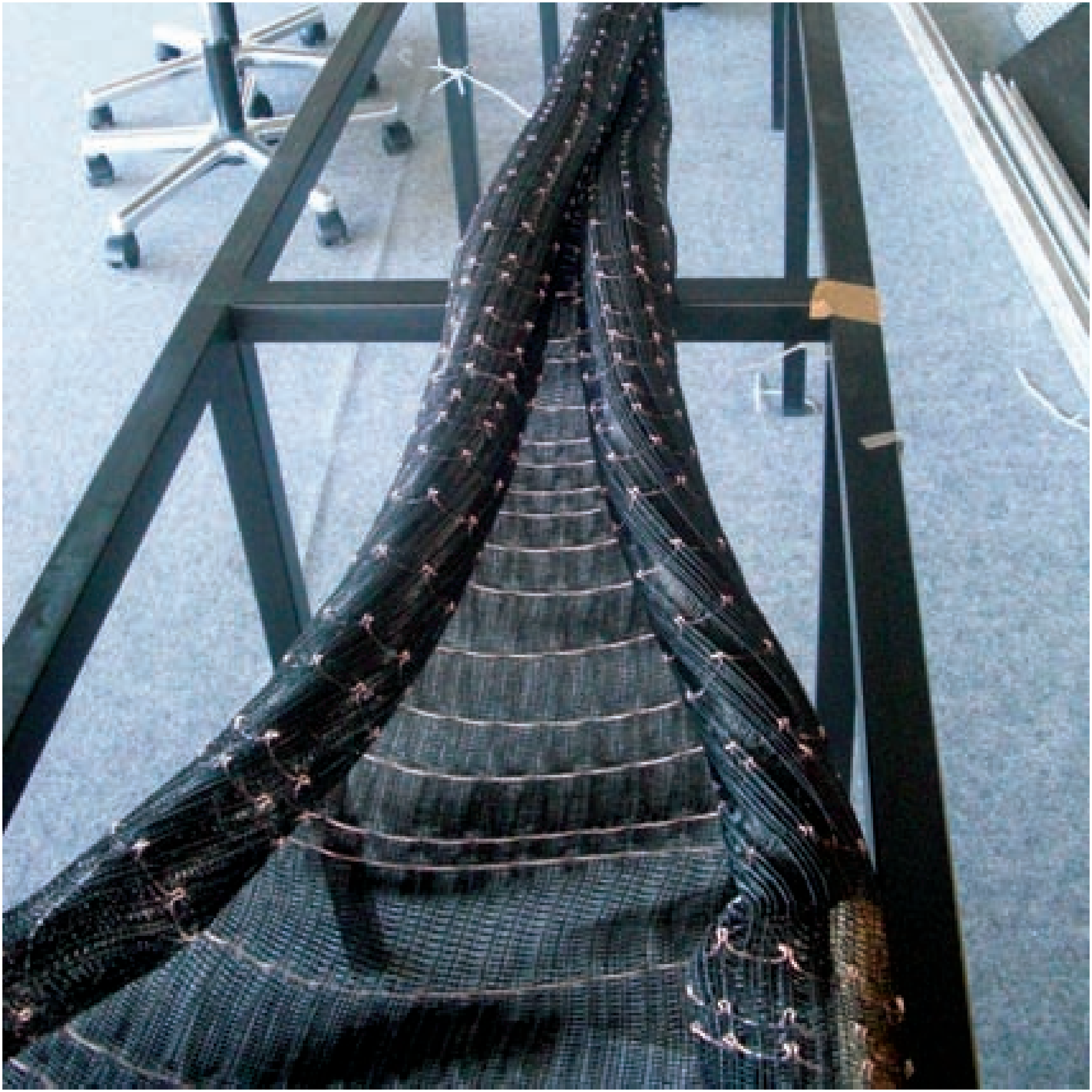

CMS 520 C
STOLL

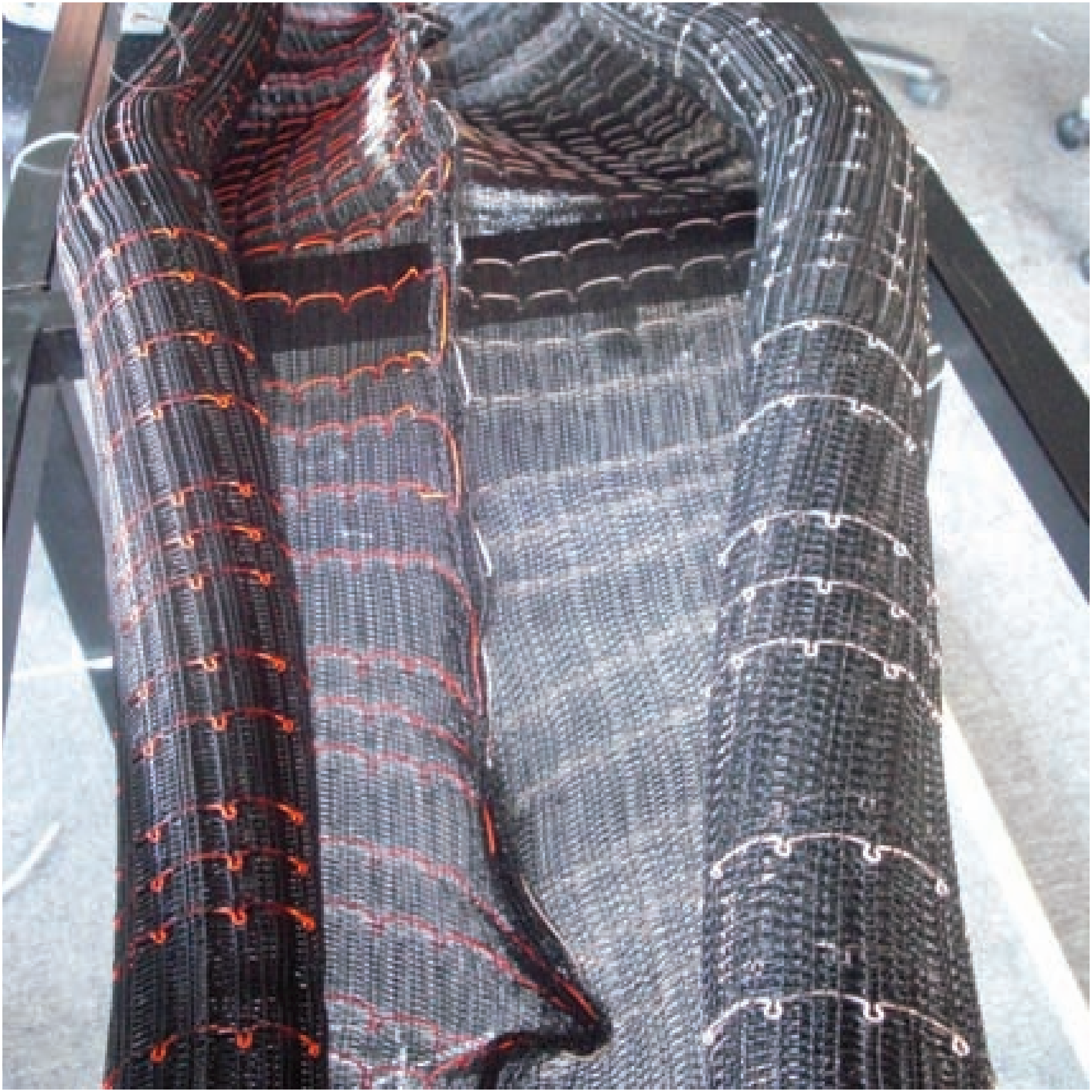
PRODUCTION

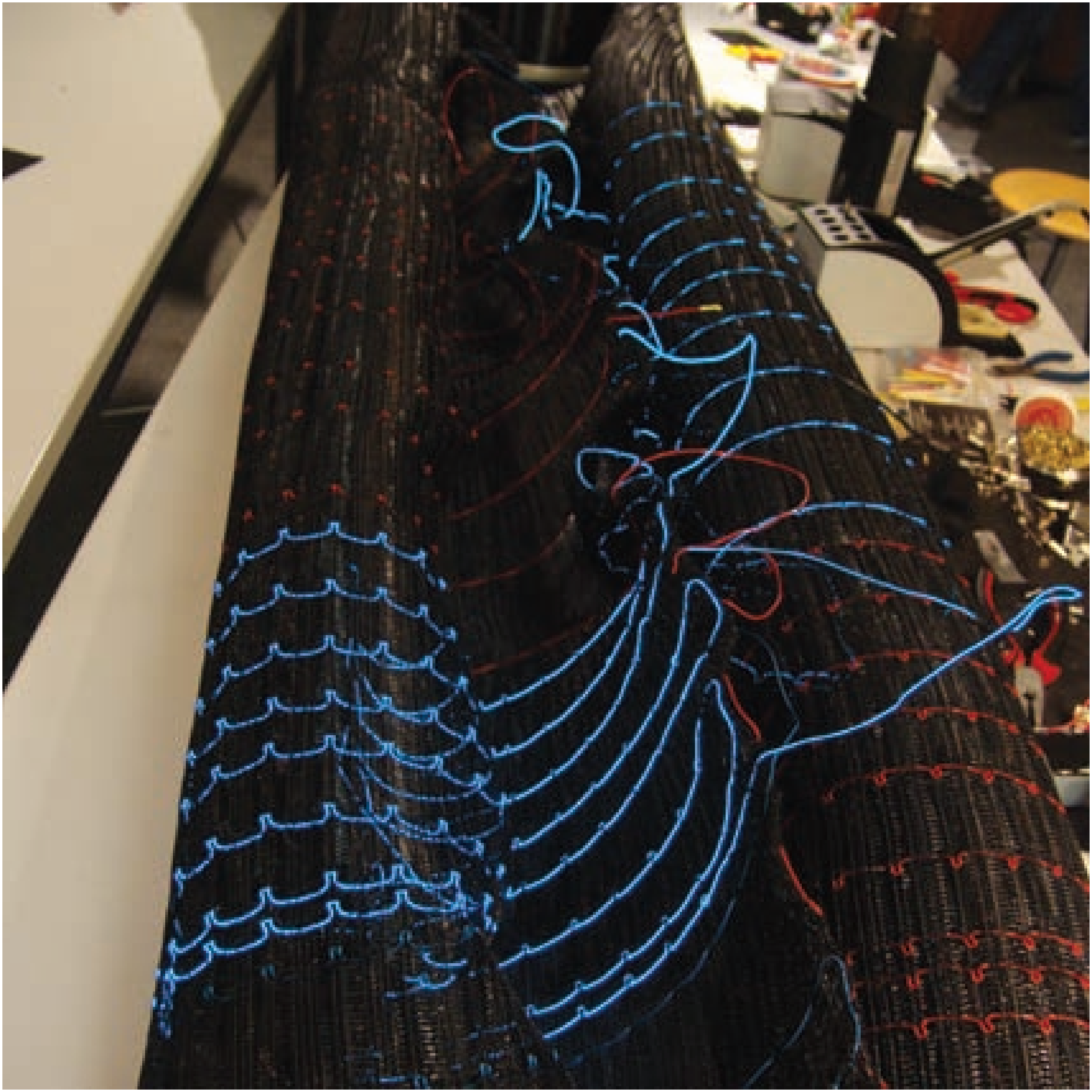
Stoll Company played an important role in the production process by proposing and introducing new materials and techniques. Monofilament of 4mm was chosen as the support material owing to its technical specifications. Similar to EL wire, Monofilament has an exemplary performance in terms of elasticity and strength. Concerning the knitting method, two different techniques were proposed; the first one provides strength along the fabric length while the other one along its width. After a thorough research of each technique's competence to the supporting structural frame, the one that provided maximum strength along the length was chosen.

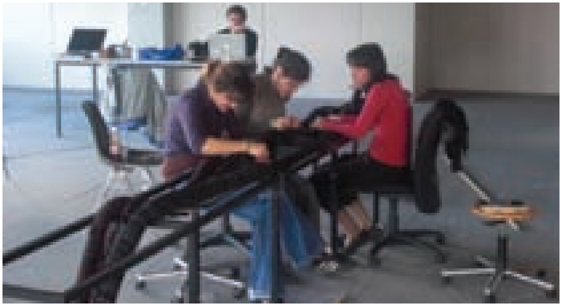
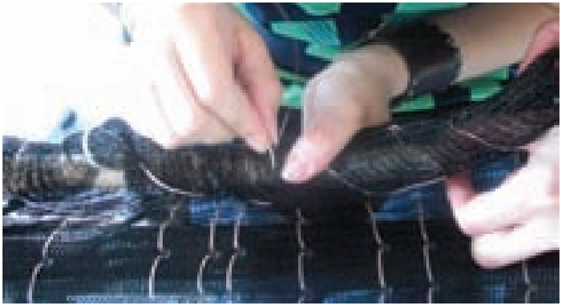
Spatz Company, specializing upon lightweight constructions, played also a key role in the realization process. With respect to the company's experience upon various projects and structures, we were able to make the most of their valuable advice on finding the most successful combination between structure and fabric.

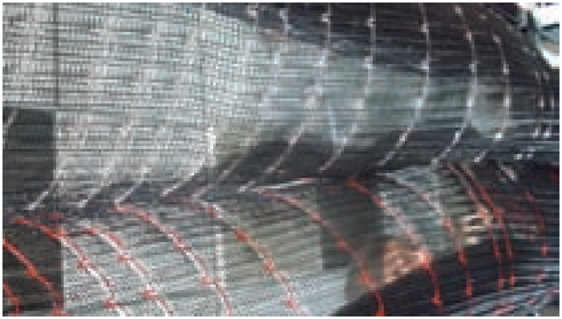
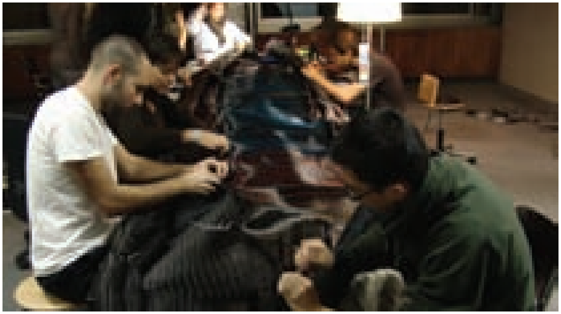
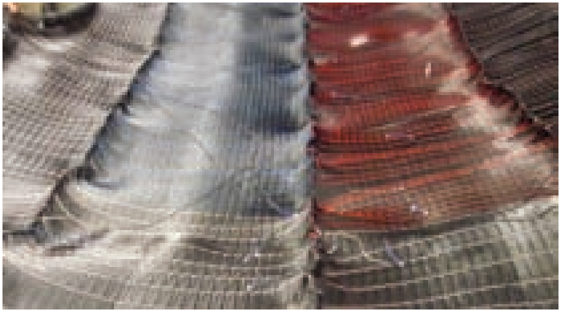


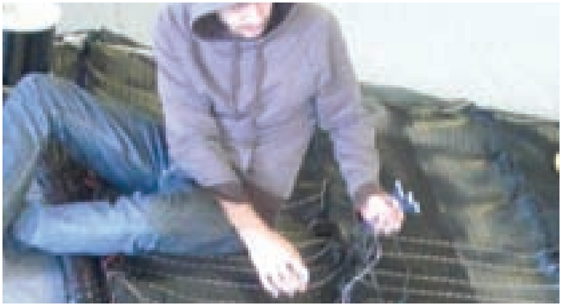
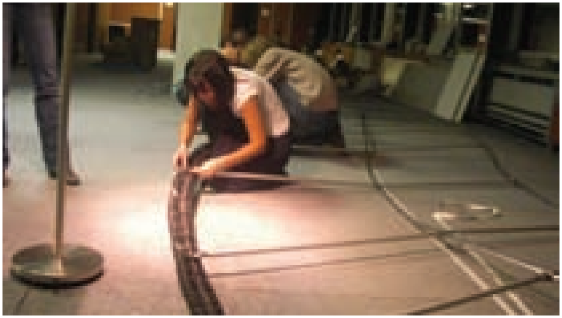


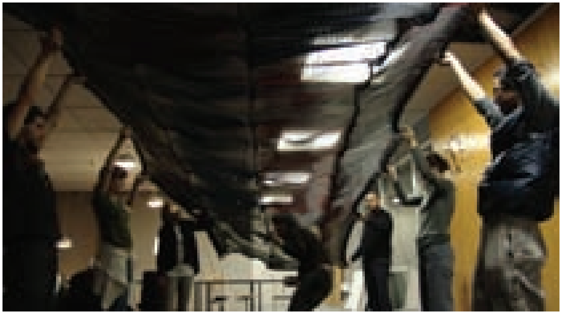


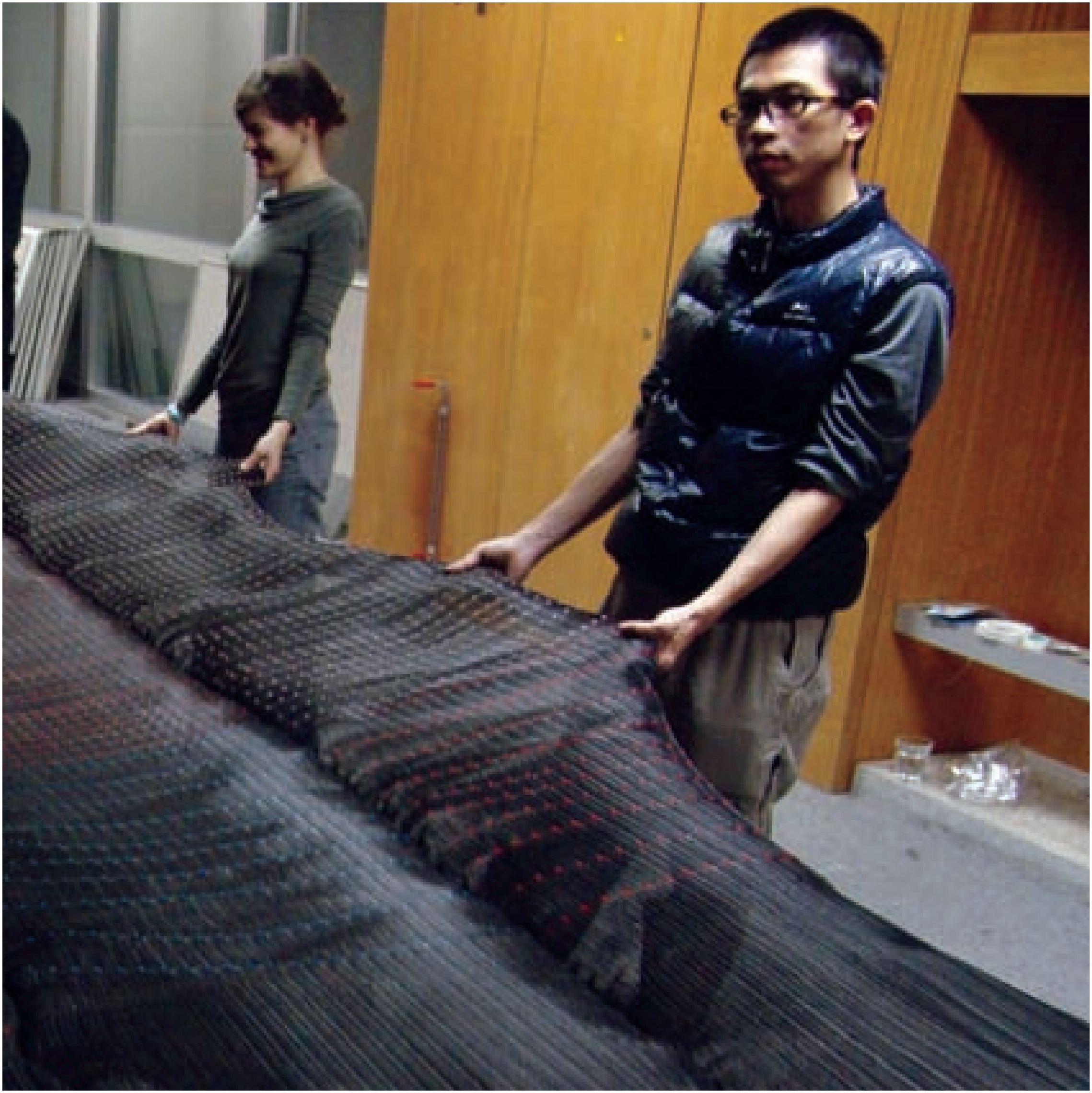


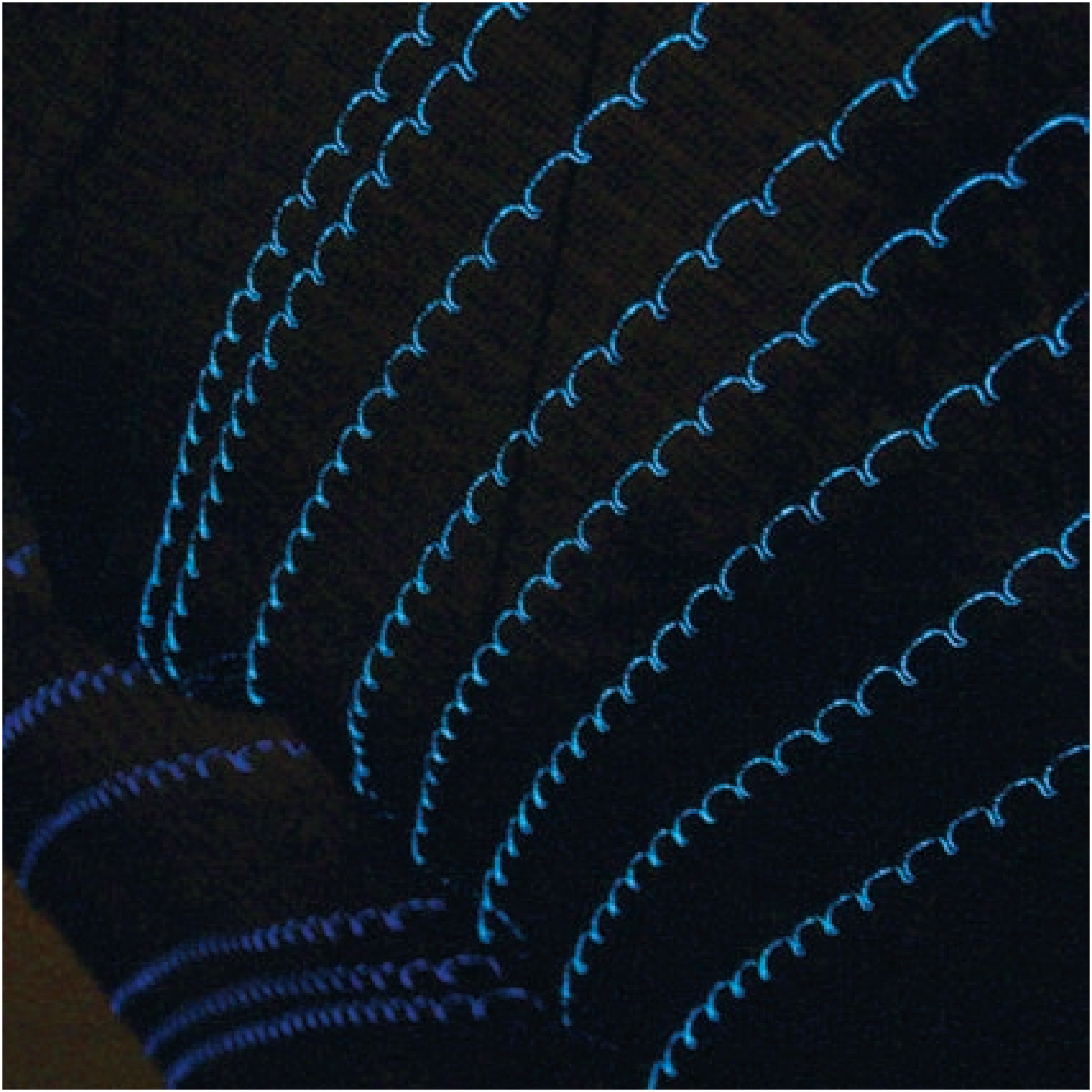



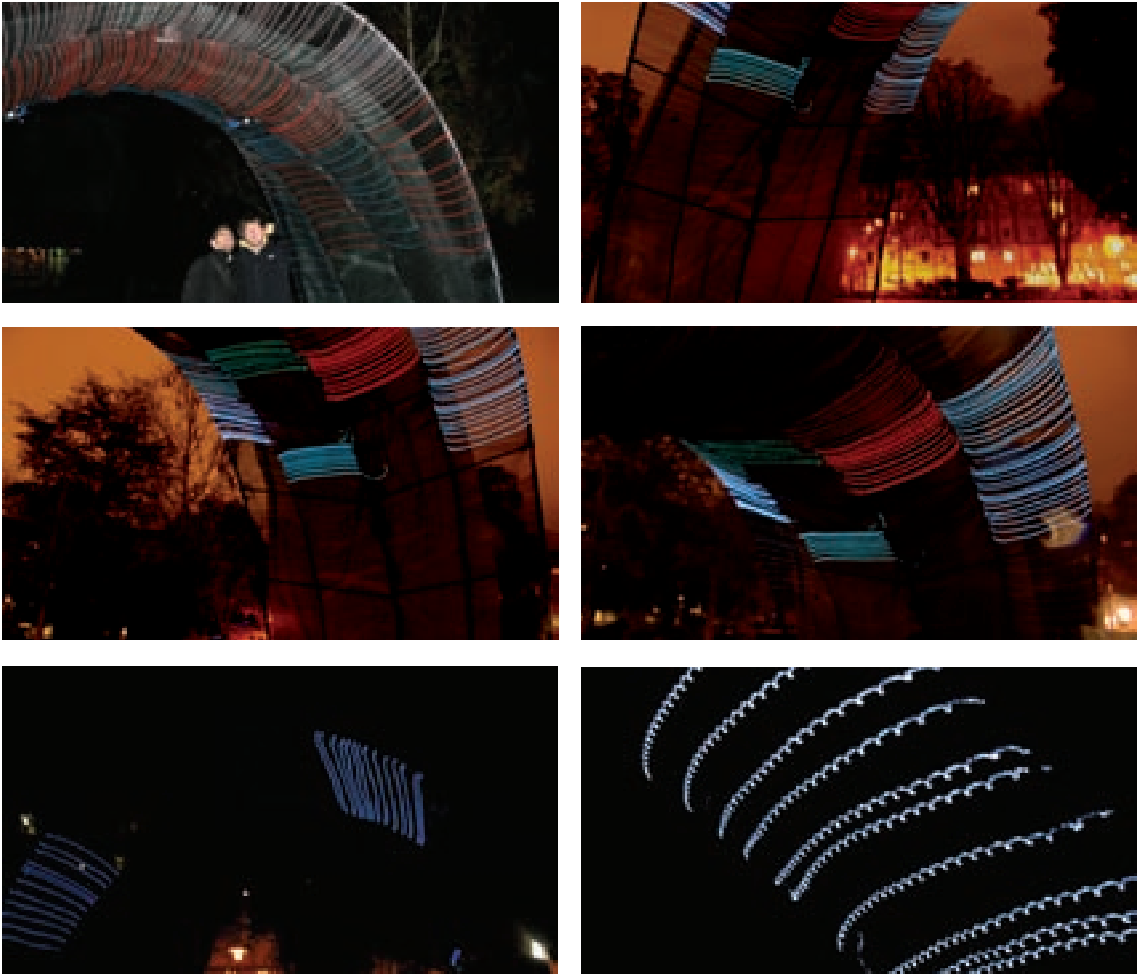




VERNISSAGE

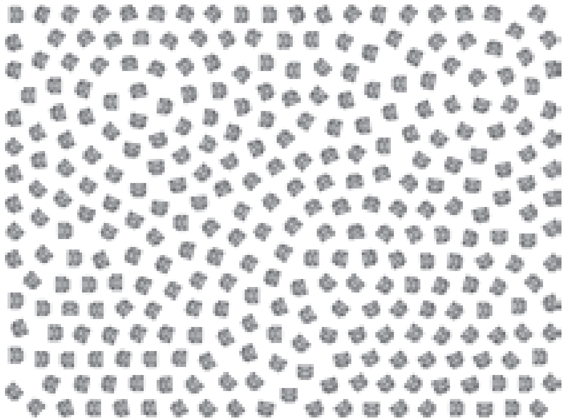
The community center of Bäckeranlage Park is a famous place to the residents of Zurich where residents of all ages, interests and backgrounds meet. The Bäckeranlage community supports various activities, initiatives and events proposed by individuals or community groups. Their goal is to promote coexistence, exchange of ideas and enhancement of the social network.
Furthermore, the place is an important node both for the pedestrian circulation, and public transportation. Its location by a main street in the center of Zurich ensures visibility and the potential for activities that take place in the park to act as attraction poles.
In order to offer achieve maximum public participation we choose the Bäckeranlage for the location of our project. Our goal is to provide the opportunity of a true interaction with the public. In that sense our structure is awaiting to be interpreted by various individuals-actors. Only through this active interpretation does the INTERACTIVE URBAN FABRIC gain its full meaning as a concrete urban stage.

Assembling at Bäckeranlage Park



During the Vernissage

INDIVIDUAL **THESES**



EINFLUSS DER TOPOGRAFIE AUF COMPUTERGENERIERTE SIEDLUNGSSTRUKTUREN

THESIS BY MATHIAS BERNHARD

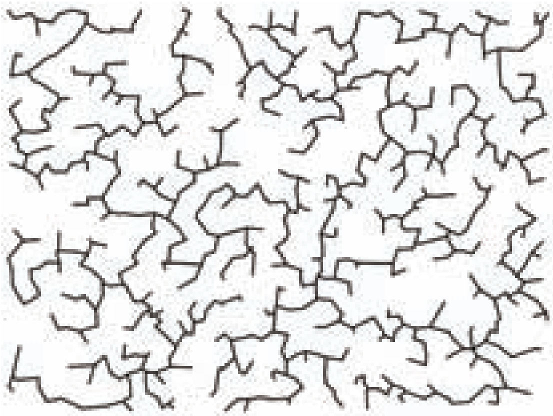
The following pages are a short summary of the MAS final thesis by Mathias Bernhard. It was conceived and is printed also as a stand-alone book. Therefore it is written in German and there might be some references to chapters, that are not part of this extract.

The work deals with questions of distribution and connection of different elements of a city. The authors mentor was Markus Braach of the research group KAISERSROT, who brought a lot of experience in the field of computer generated architecture and urbanism to the chair of CAAD.

One main focus is on the influence of topography on the generated networks and on how the topology changes at under weightings of height differences between nodes.

BÄUME

Die kürzeste Gesamtlänge aller Kanten in einem Graphen erhält man bei einem sogenannten Minimalen Spannbaum (Minimal Spanning Tree, MST). Ein Spannbaum enthält alle Knoten eines Graphen und es gibt für jeden Knoten genau einen möglichen Weg zu jedem anderen Knoten des Graphen - entweder direkt über eine die beiden Knoten verbindende Kante oder indirekt via andere Knoten.



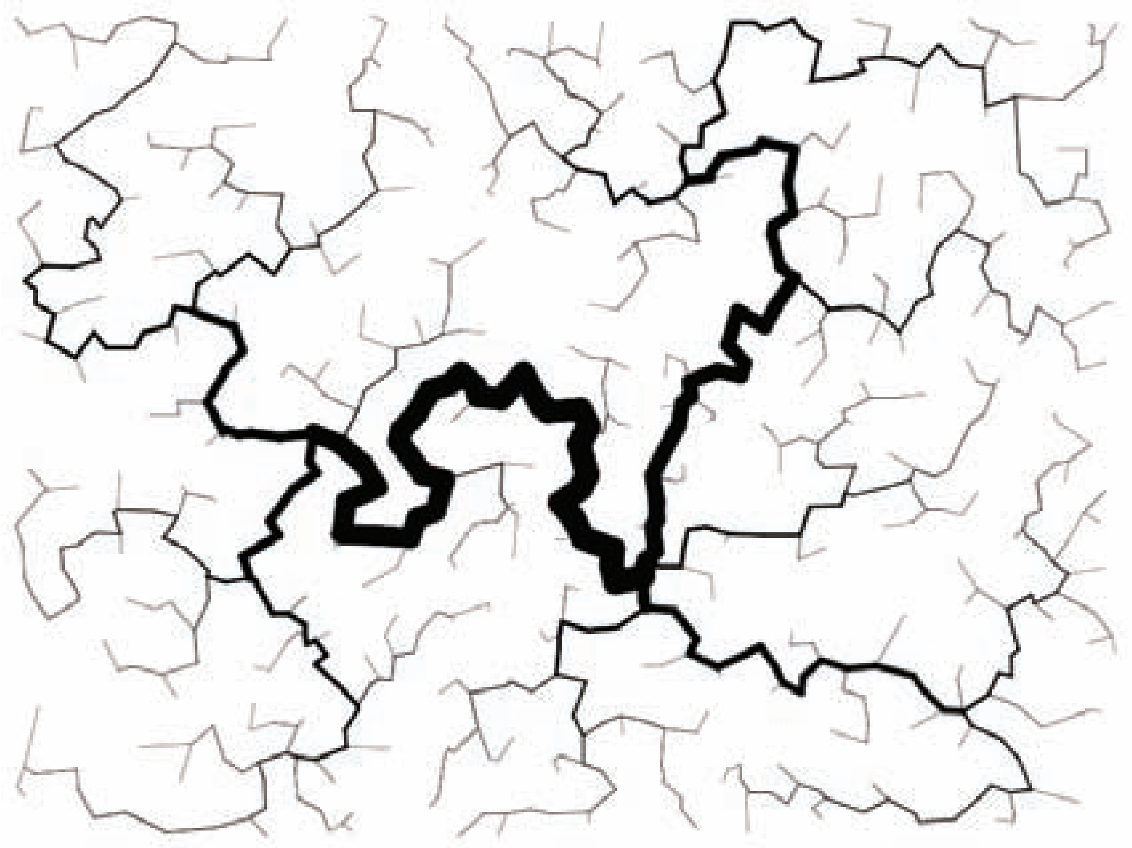
Minimal Spanning Tree (MST)

```
Liste unbesuchteKnoten = alle Knoten
Knoten k = unbesuchteKnoten[0]
Liste besuchteKnoten = leere Liste
füge k zu besuchteKnoten
entferne k aus unbesuchteKnoten
Während (unbesuchteKnoten nicht leer)
  kürzesterAbstand = 999999999
  für jeden Knoten kt in besuchteKnoten
    kn = suche nächsten Nachbarn von kt
    a = Abstand(kt,kn)
    Wenn (a < kürzesterAbstand) dann
      kürzesterAbstand = a
      startKnoten = kt
      endKnoten = kn
    Ende(Wenn)
  Nächster
  neuerKnoten = Kante(startKnoten,endKnoten)
  entferne kn aus unbesuchteKnoten
  füge kn zu besuchteKnoten
Weiter
```

Eine noch kürzere Gesamtlänge aller Kanten würde man durch Berechnung eines sogenannten Steiner-Baumes erhalten. Dabei werden an gewissen Stellen zwischen jeweils 3 Knoten zusätzliche, sogenannte Steinerknoten eingeführt, so dass die drei davon ausgehenden Kanten jeweils einen Winkel von genau 120° aufspannen. Solche Strukturen können auch mit zwischen Nägeln aufgespannten Seifenhäuten beobachtet werden.

GEWICHTETER BAUM

Im obigen Beispiel sind alle Kanten als gleichwertig dargestellt. Einige sind jedoch zentraler und wichtiger als andere. Um dieser Tatsache Rechnung zu tragen kann der Graph gewichtet werden. Dabei wird bei jeder einzelnen Kante die Grösse des Baumes (Anzahl Kanten) auf jeweils beiden Seiten gezählt. Der kleinere von beiden entspricht dann dem Gewicht der jeweiligen Kante.



MST, gewichtet nach Zentralität

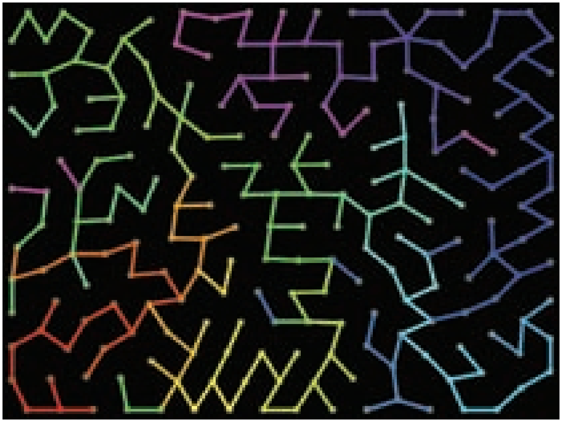
Diese Grösse variiert folglich zwischen 0 (Sackgassen, Knoten mit nur einer Kante) und N/2 (N=Anz. Knoten, zentralste Kante, auf beiden Seiten je die Hälfte des Graphen). Um diese Gewichtung darzustellen wurden in der Grafik die Liniendicken gemäss dem Gewicht der Kante verändert. Die Knoten, an die dicksten Kanten anschliessen, wären in einer klassischen Baumdarsellung dann die obersten.

Bezogen auf z.B. ein Strassenverkehrsnetz wären dies die am stärksten befahrenen Strassen, wenn alle Knoten alle anderen besuchen würden. Eine andere Grösse, die die Zentralität einer Kante ausdrückt, ist der längst-nötige Weg zu allen anderen Knoten/Kanten im Graphen.

„A CITY IS NOT A TREE"

Christopher Alexander schreibt in *A Pattern Language* den berühmten Text „A City is not a Tree". Er meint damit, dass eine Stadt oder auch eine Gesellschaft oder allgemein eine Organisation aus vielen einzelnen Teilen mit dieser hierarchischen Art von Abhängigkeiten nicht funktionieren kann. Zu Beginn des Textes macht er die Unterscheidung zwischen „natural" (natürlich gewachsene) und „artificial" (künstliche, auf dem Reissbrett entworfene) Städte, um dann aufzuzeigen, dass diese stärkere Art von Vernetzung der natürlichen Städte das sei, was vielen künstlichen fehle.

Ein minimaler Spannbaum ist ein sogenannt zusammenhängender Graph, bei dem alle Knoten mit jedem anderen durch genau eine Folge von Kanten verbunden ist. Es würde also reichen, eine Kante zu unterbrechen, um das ganze System in zwei unabhängige Teilsysteme zu unterteilen. Im Hinblick wiederum auf die Interpretation als Strassennetz würde dies heissen, dass eine Baustelle ein Teil der Stadt unzugänglich für andere Teile macht und auch nicht umfahren werden kann (*genau* eine Folge von Kanten und nicht *mindestens* eine Folge).



Minimaler Spannbaum mit Entstehungsalter

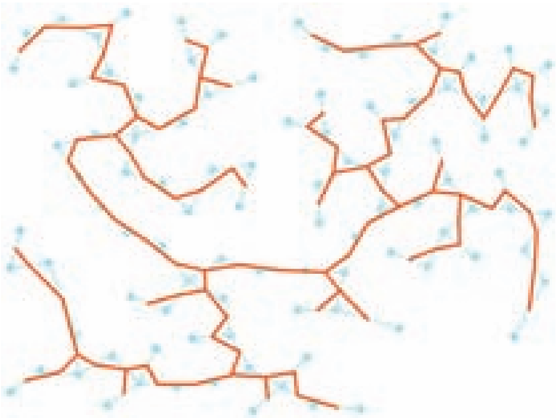
Spannbaum mit geschlossenen Loops

Bei diesen beiden Graphen wurde erst ein normaler Spannbaum berechnet. Die Farben geben hierbei Auskunft über die Entstehungsreihenfolge der jeweiligen Kante (HSB: rot-orange-gelb-grün-cyan-blau-lila).

In einem zweiten Schritt wurden dann alle offenen Enden (Knoten ersten Grades mit nur einer Kante, „Sackgassen“) mit dem jeweils zweitnächsten Knoten verbunden (der nächste ist die bestehende Verbindung). Bei diesem Schritt könnte die minimal (und maximal) zulässige Grösse der Schleifen (entstehende geschlossene Polygone) festgelegt und kontrolliert werden. Hier wurde dieses Minimum auf vier Knoten gesetzt.

BAUM ZWEITEN GRADES

Sollen die Kanten des Graphen als eine Erschliessung der Knoten (Strassen und Häuser) betrachtet werden, macht eine Berechnung auf die Knotenmittelpunkte wenig Sinn. Vielmehr sollen die Knoten die Kanten „säumen“. Bei diesem Beispiel eines Baumes zweiten Grades wird



blau: MST 1. Grades, rot: MST 2. Grades

1 zuerst ein Spannbaum ersten Grades mit den ursprünglichen Knoten gerechnet

2 dann von jeder Kante der Mittelpunkt als Knoten zweiten Grades extrahiert

3 und auf diesen Knoten dann wieder ein Spannbaum gerechnet.

So können pro Kante mehrere Knoten links und rechts erschlossen werden und starke Zick-zack-Verläufe werden begradigt. Die Gesamtzahl der Kanten E bei einer Anzahl Knoten N wird auf diese Weise nur unwesentlich reduziert (`MST: E = N-1, MST2: E=N-2`).

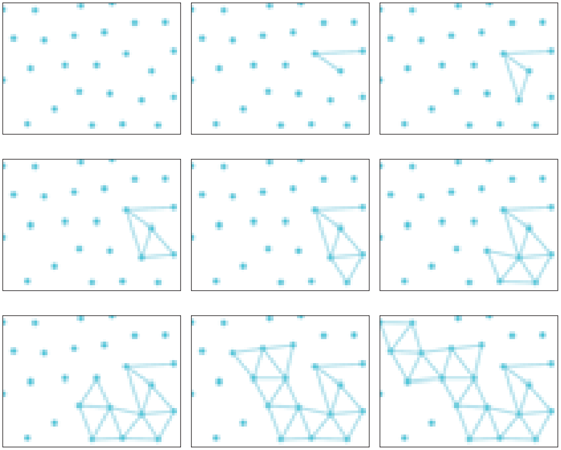
TRIANGULATION

Bereits in einem vorangehenden Kapitel (*Baum zweiten Grades*) wurde ein möglicher Weg beschrieben, mit den Kanten nicht direkt die Knoten zu verbinden, sondern sie so zu erschliessen, dass die Knoten die Kanten säumen (wie Häuser entlang der Strasse). In diesem Abschnitt soll noch ein weiteres Werkzeug eingeführt werden, das diese Strategie verfolgt.

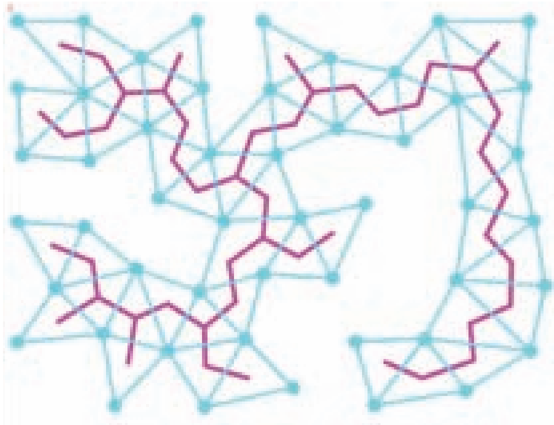
Durch das Ermitteln von Nachbarschaftsverhältnissen (Abstände der Knoten untereinander) wird in einem ersten Schritt eine Art Clusterung[1] vorgenommen. Diese wird ausgedrückt, indem immer drei naheliegende

1 unter Cluster wird hier eine Gruppe von Knoten (eine Teilmenge mit 1 - N Elementen des Graphen G mit N Elementen) verstanden, die zueinander näher sind als zu allen anderen Knoten.

Knoten untereinander ein Dreieick aufspannen. Nachfolgend eine Illustration, wie dies Schritt für Schritt konstruiert wird.



Triangulation Schritt für Schritt



cyan: Triangulation, magenta: MST auf Centroiden

1 Die Menge aller Knoten im Graphen mit einem gewissen Mindestabstand
2 Die kürzeste Kante zwischen zwei Knoten wird gesucht
3 Der Knoten, der mit der Kante (aus 2) das Dreieck mit dem kürzesten Umfang bildet wird eingebunden
4 Von jedem Dreieck aus einem noch nicht besuchten Knoten und einer existierenden Kante wird der Umfang gemessen. Das kleinste wird hinzugefügt
5 Schritt 4 wird solange wiederholt, bis alle Knoten als „besucht“ markiert sind

Die so entstehenden Dreiecke bilden die Ausgangslage für den nächsten Schritt. Dabei wird von allen Dreiecken das Zentrum ermittelt `[P((x1+x2+x3)/3, (y1+y2+y3)/3)]`. Diese Punkte dienen dann wiederum als Knoten für einen minimalen Spannbaum.

Mit dieser Methode wird automatisch der dichtere Raum zwischen den Knoten zur Erschliessungsfläche, die weiteren Räume bleiben unberührt. Alle Knoten haben so einen mittleren Abstand zur nächst gelegenen Kante, oder eben Häuser zur Strasse.

Eine solch homogene Triangulierung entsteht nur bei einer relativ gleichmässigen, sprich hexagonalen Verteilung der Punkte. Liegen einzelne weit von den anderen entfernt, entstehen flache spitze Dreiecke und der Verbindungspfad kann sehr nah an den Knoten liegen.

Darstellung desselben Graphen als Luftbild

Kombiniert man dieses Resultat nun noch mit dem im vorangehenden Kapitel vorgestellten Werkzeug (geschlossene Loops), rundet die Knicke etwas aus und wählt eine andere Repräsentation hat man schon fast ein Dorf vor sich.

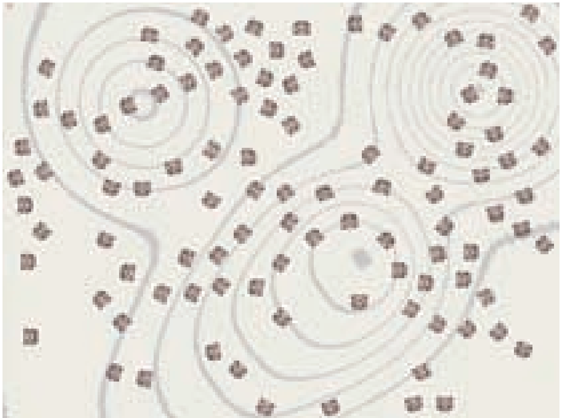
REAKTION AUF DIE TOPOGRAPHIE

Die in den vorangehenden Kapiteln *Verteilen* und *Verbinden* vorgestellten Werkzeuge behalten auch in den folgenden Untersuchungen uneingeschränkt ihre Gültigkeit und werden vorausgesetzt. Als zusätzlicher Input, worauf die Objekte unterschiedlich reagieren (können) kommt ein Gelände hinzu.

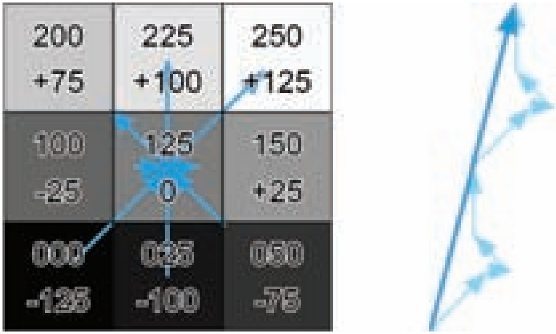
DREHUNG GEMÄSS NEIGUNGSRICHTUNG

Im Kapitel *4.2 Reaktion auf den Untergrund* wurde beschrieben, wie einzelne Objekte spezifisch auf örtlich variierende Parameter wie zum Beispiel Bodenpreise reagieren können. Auch der absolute Wert „Höhe“ (Z-Komponente des Standorts in der Landschaft), der relative Wert „Höhenunterschied“ zweier Objekte oder der Vektor „Hangrichtung“ kann die Konfiguration eines Objektes steuern.

In diesem Beispiel wird dieser Vektor, also die Verlaufsrichtung des Gefälles am Standort eines Hausobjektes, ermittelt um die Längsseite des Hauses quasi-parallel zu den Höhenlinien auszurichten. Die Balkone sind somit alle talwärts ausgerichtet. Dadurch wird der Aushub minimiert und die Aussicht maximiert.



Ausrichtung parallel zu den Höhenlinien



Berechnung der Hangrichtung

Die Richtung soll also ein talwärts zeigender Vektor sein, der Rotation des Hauses um seine eigene Z-Achse bestimmt. Um die Ausrichtung eines Hauses zu ermitteln, wird

1 an dem Standort in 8 Richtungen (or, r, ur, u, ul, l, ol, o) der Höhenunterschied gemessen,

2 die 8 Richtungsvektoren (`1/1, 1/0, 1/-1, 0/-1`, etc) mit dem jeweiligen Höhenunterschied skaliert und schliesslich

3 alle 8 skalierten Vektoren addiert.

GEWICHTUNG DES HÖHENUNTERSCHIEDES

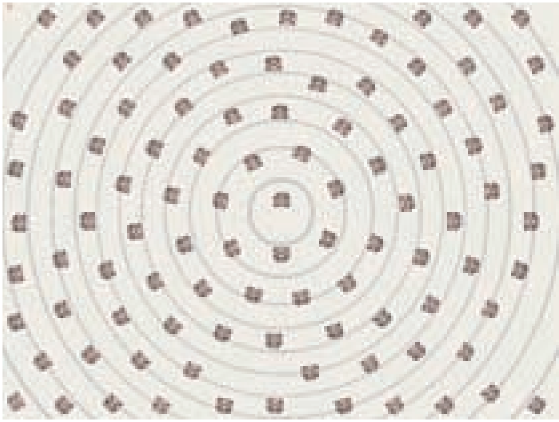
Bis jetzt wurde bei der Generierung der Graphen als Abstand zweier Punkte nur die 2D-Distanz in der XY-Ebene berechnet. Liegen die Knoten A und B jedoch auf unterschiedlicher Höhe, muss der benötigte Aufwand um von A nach B zu gelangen ungleich höher bewertet werden. Dabei genügt es wohl nicht, nur die tatsächliche Länge der Verbindung (`Sqrt(x² + y² + z²)`) zu kalkulieren, sondern es soll eine Methode gefunden werden, wie der Höhenunterschied (`Abs(A.z - B.z)`) in die Kalkulation einfliessen muss, um dem wahren Mehraufwand Rechnung zu tragen.

Beispiel: Angenommen eine Strasse zwischen A und B, beide Punkte in der Ebene z0 mit Abstand 100m. Verschiebt man nun B um 20m nach oben (20% Steigung), wird die Verbindung nur knapp 2m länger, der Aufwand (Kraft, Energie, Zeit, Geld) nimmt aber viel mehr zu.

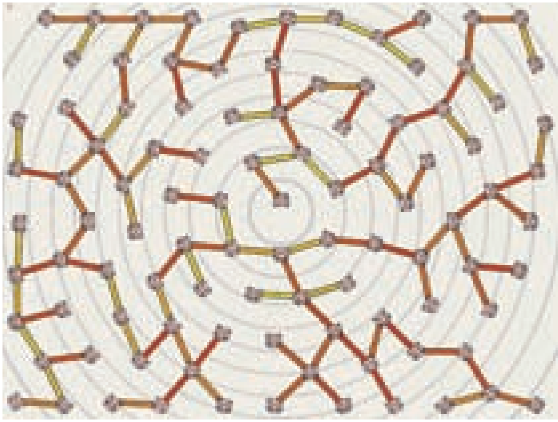
Testtopographie, ol: Displacement Map, or: Höhenlinien, u: 3D-Mesh

Im Folgenden wird in einer „Testtopographie“ mit einem zentralen, runden Hügel die Entwicklung eines Spannbaumes gezeigt, bei dem der Höhenunterschied jeweils N mal zum 2D-Abstand dazugezählt wird.

Zuerst wird eine Initialpopulation von 100 Häusern zufällig auf dem Areal verteilt und dynamisch mit einem gewissen Mindestabstand zueinander ausgerichtet. Die Häuser alinieren sich ebenfalls parallel zu den Höhenlinien, wie dies im vorangehenden Kapitel beschrieben wurde.



Initialpopulation, 100 Häuser



Minimaler Spannbaum, nur 2D-Distanz

Die Verteilung, also die Position der Knoten wird in den folgenden Darstellungen nicht mehr verändert. Es werden vorerst unterschiedliche Arten untersucht, die Knoten miteinander zu verbinden.

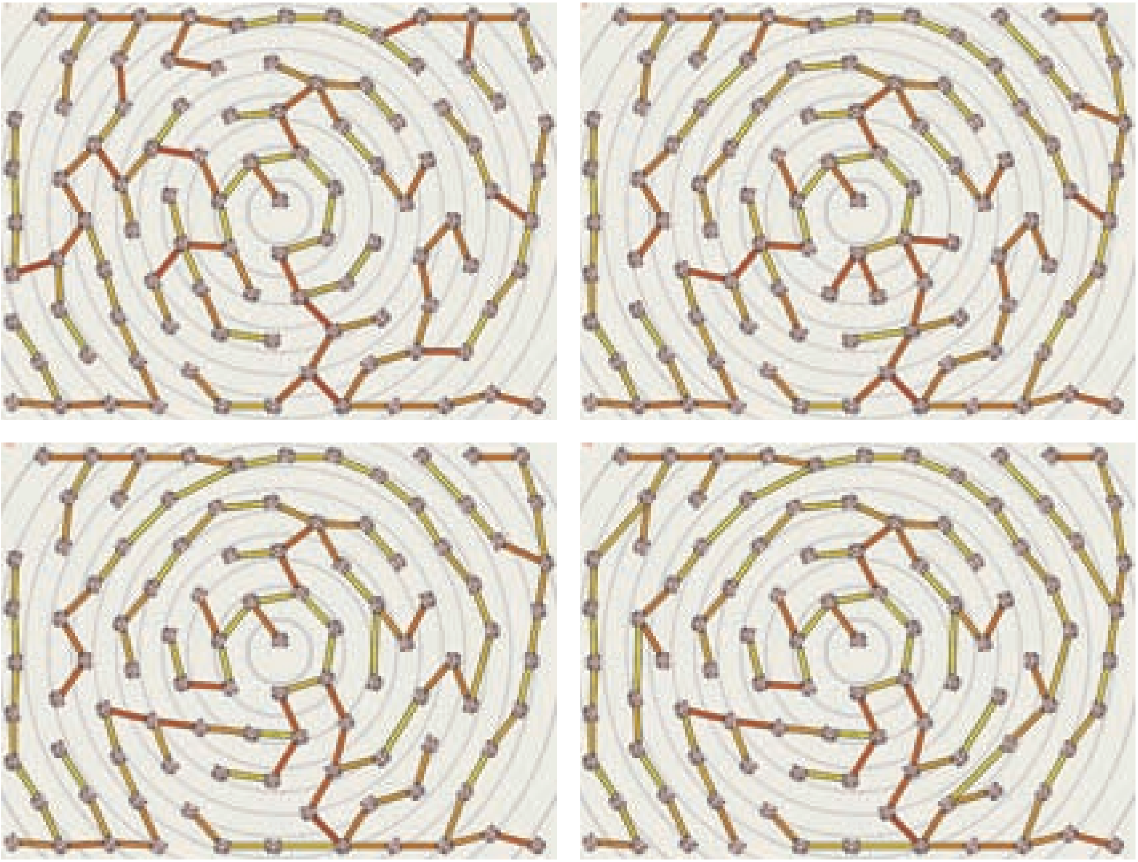
Als zweites wird ein minimaler Spannbaum ohne Berücksichtigung der Höhenunterschiede, also bloss in 2D, generiert.

Die farbliche Codierung der Strassen von gelb nach rot repräsentiert deren Steigung (gelb entspricht einem kleinen, rot einem grossen Höhenunterschied). Der Rot-Anteil ist bei dieser Lösung insgesamt sehr hoch. Wird nun Schritt für Schritt die Gewichtung N des Höhenunterschieds zwischen zwei Knoten erhöht, beginnt sich dieser Anteil in Richtung Gelb zu verschieben.

Beispiel:

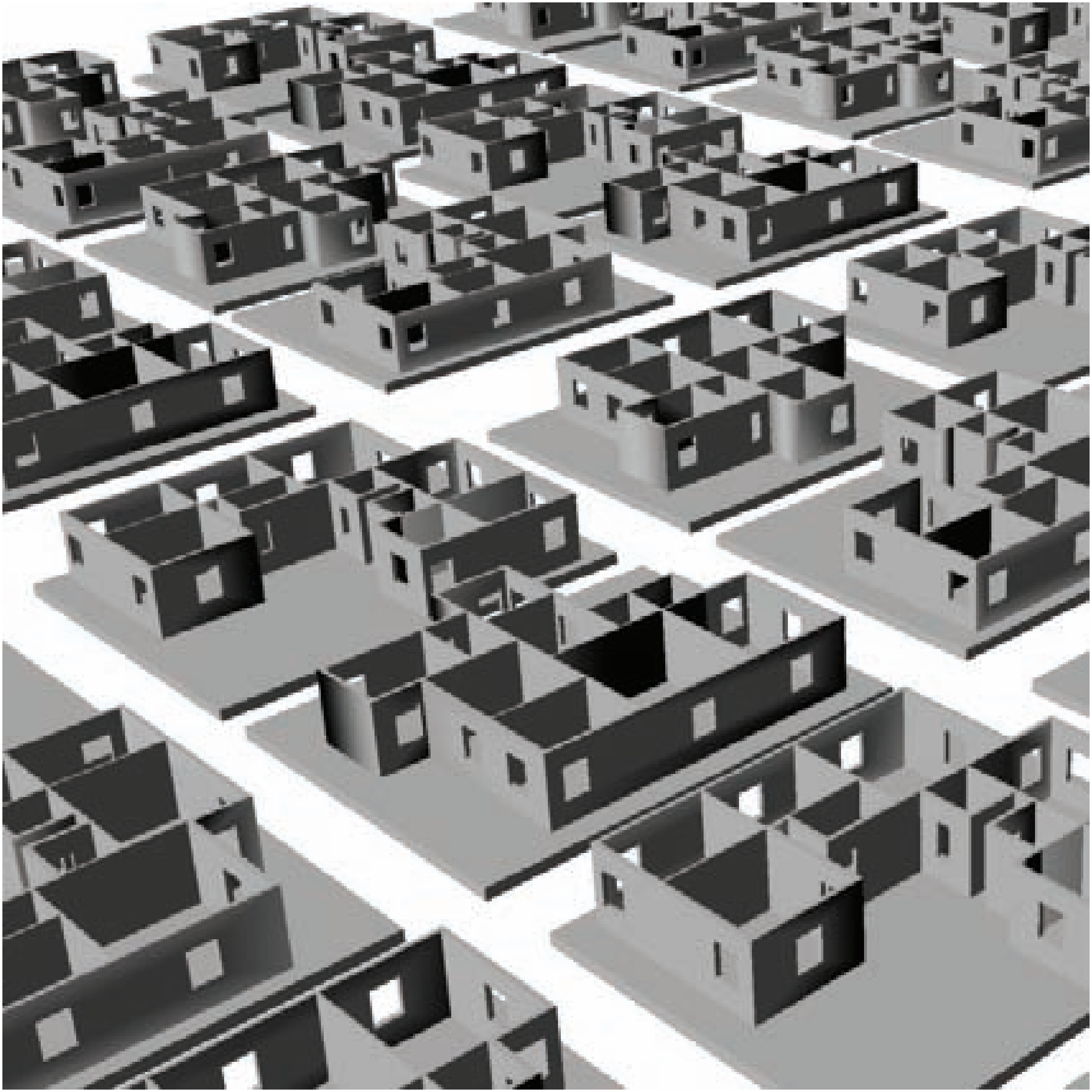
```
P1 = (0,0,0), P2 = (4,3,2) > 2D-Distanz d = 5
deltaZ = 2
Länge L = d + N * deltaZ
N = 0 > L = 5 + 0 = 5
N = 1 > L = 5 + 2 = 7
N = 2 > L = 5 + 4 = 9
...
N = 5 > L = 5 + 10 = 15
etc...
```

Welche Gewichtung die „Richtige“ ist kann nicht abschliessend beurteilt werden. Ob eine Strasse mit 40% Steigung zu passieren den ein-einhalb-fachen, den doppelten oder den dreifachen Aufwand bedeutet hängt wohl auch massgeblich vom verwendeten Verkehrsmittel ab. Auf eine physikalisch korrekte, energetische Berechnung wird an dieser Stelle verzichtet.



Unterschiedliche Gewichtung von deltaZ, ol: N = 1, or: N = 5, ul: N = 10, ur: N = 15

In diesen Darstellungen ist deutlich zu erkennen, wie sich bei gleichbleibender Anordnung der Häuser die Strassen mehr und mehr zu konzentrischen Ringen formieren. Die durchschnittliche Steigung der Strassen nimmt kontinuierlich ab (mehr gelbe, weniger rote).



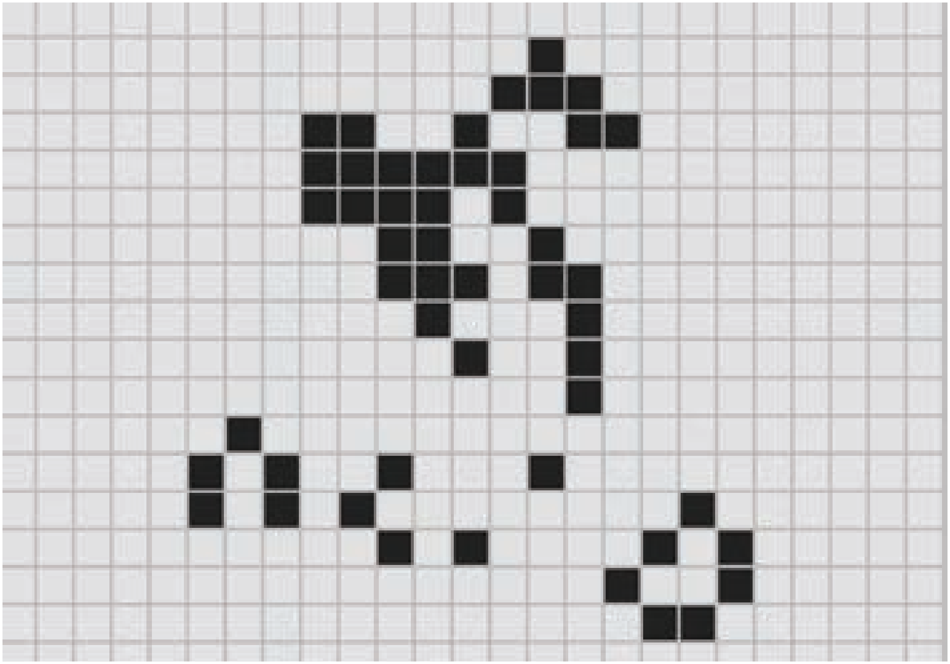
THE CLIMATE HOUSE
THESIS BY KATERINA BOUZIANA

House is basic need for human beings. From nomads to the richest people, everybody tries to build a house to stay, to protect him and gain some privacy.
Evidence is gathering that human activities are changing the climate. This 'climate change' could have a huge impact on our lives. As contemporary architects we have the obligation to propose solutions in order to create safe and viable houses.
By combining these two facts, the title of the project arouses: "The Climate-House". It is a first try to approach and create a program that filters contemporary dwelling through the vital problem of climate denaturation and changing of our planet. It's an attempt to predict and design the best possible floor plan of a house and how it changes over the time. It's also an attempt to create a flow of information from nature to residence and vice versa, in real time.
It is a project that started with pretensions and can constitute an important tool for contemporary architect, when refined.

HOW IT WORKS

An application in Java-based programming language Processing Beta 0135 that produces possible floor plans for houses, depending on the climate of the area that it is going to be built in. The result every time is evaluated from factors such as average temperature, noise, light and the house is optimized until it finds the best position for the rooms. The program is based on cellular automaton theory.

Cellular automaton:
A cellular automaton evolves in discrete time steps, with the value of a variable at one site (t+1) being affected by the values of variables at sites in its “neighborhood” on the previous time step (t).
The theory has four distinct characteristics:
1. cells
2. state of the cell
3. neighborhood of a cell
4. transition rules



Cellular automaton

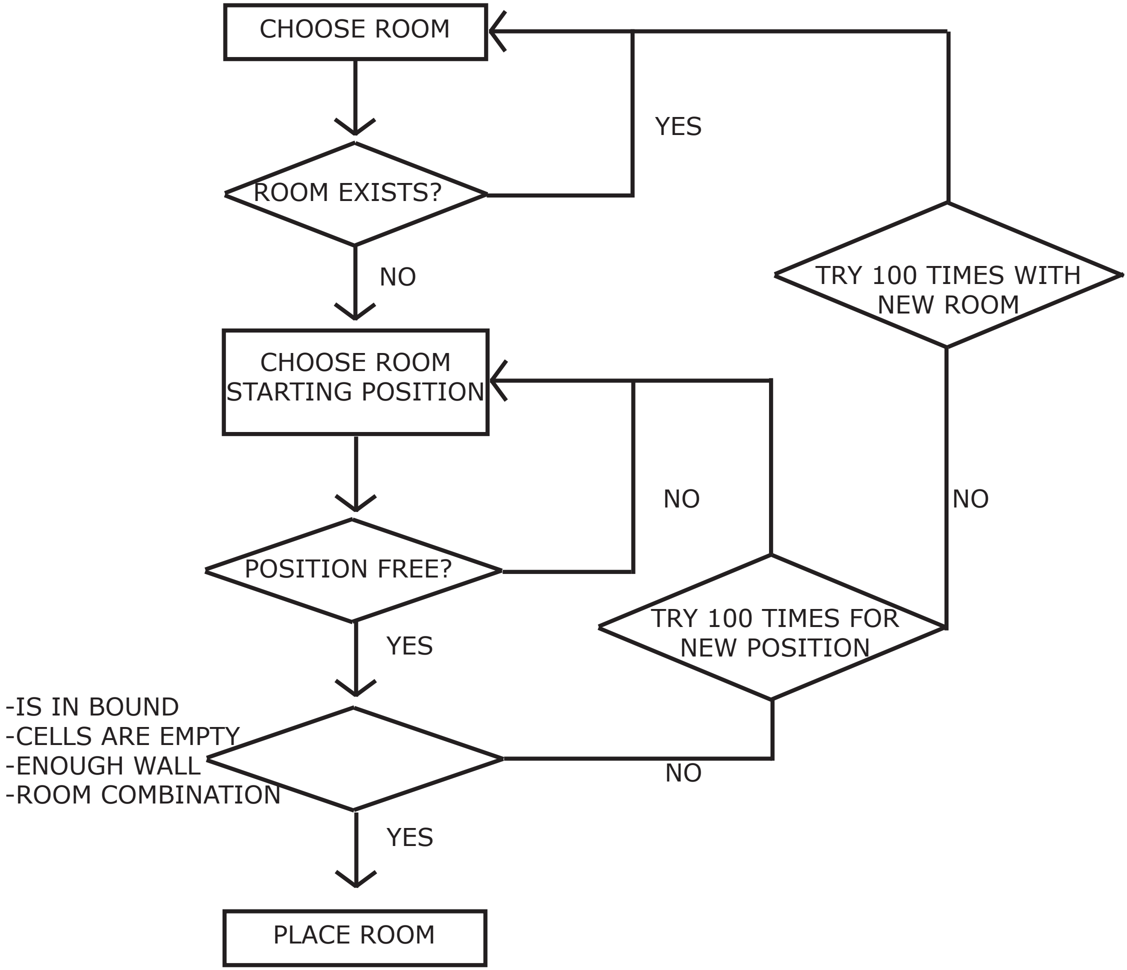
The process is non linear but logical. On one hand, the monad, the perfect, the non-subdivided object that is reproduced, multiplied, and on the other hand, the continuous matter that is subdivided, seeks for and creates relations among its points, its parts, with the aim of creatively distorting them.
Although the above criteria have been described in terms of optimizing functions, the aim is not to produce global optimum solutions but rather to direct the evolutionary process to produce populations of good solutions either as components for higher levels or at the final level itself. So that, even though the global optimum solution for the shape of a house using the above criteria, may be known, this may not be the optimum solution at the zone and city levels. By selecting other non-optimal but good solutions, according to the given criteria, good unexpected results may be achieved for the overall design.

adjoining room combinations

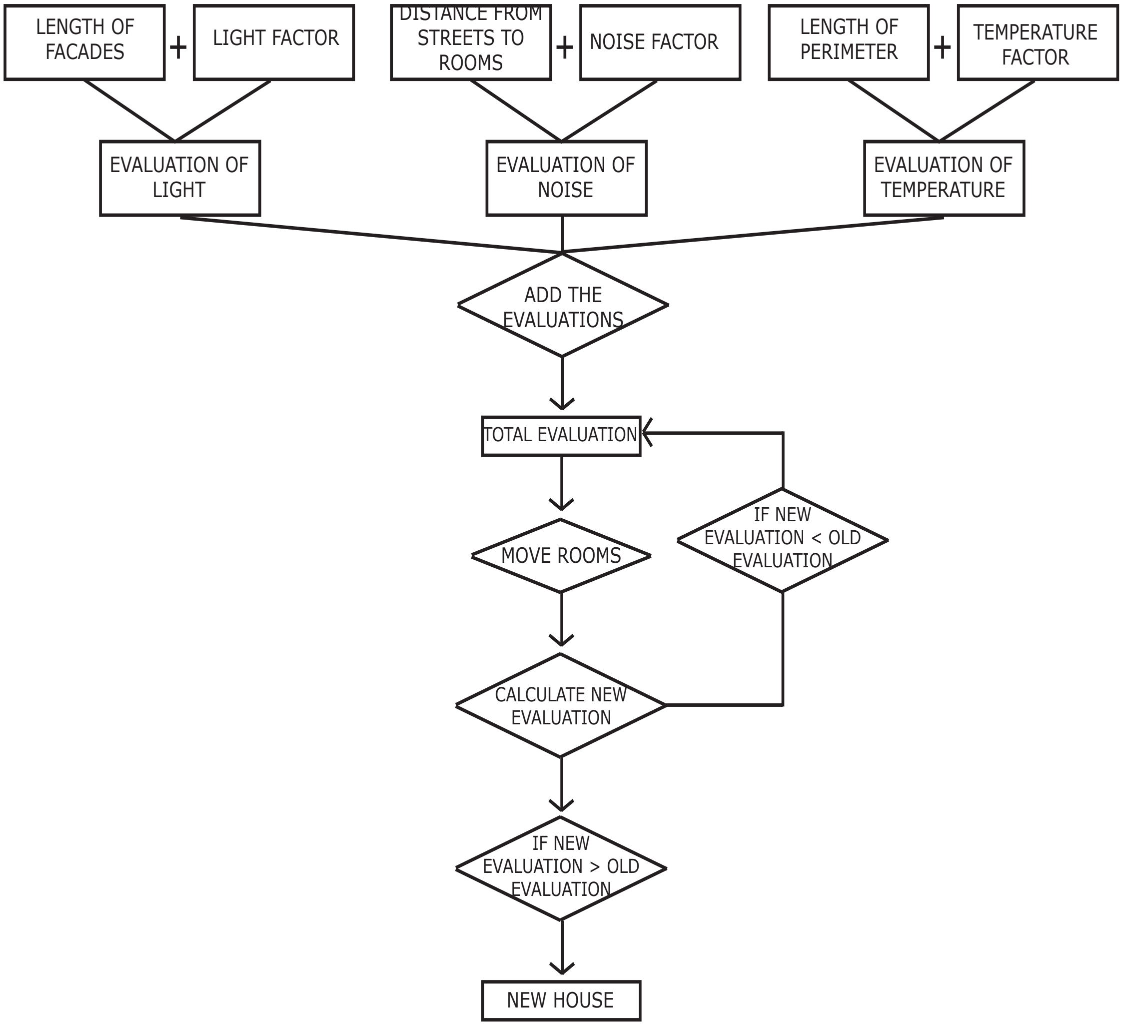
		entrance	living room	kitchen	bedroom	bedroom	bedroom	bathroom	wc	guestroom	office
	room id	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
entrance	**1**	2	1	1	0	0	0	0	1	0	1
living room	**2**	1	2	1	1	1	1	0	1	0	1
kitchen	**3**	1	1	2	0	0	0	0	1	0	1
bedroom	**4**	0	1	0	2	1	1	1	0	1	1
bedroom	**5**	0	1	0	1	2	1	1	0	1	1
bedroom	**6**	0	1	0	1	1	2	1	0	1	1
bathroom	**7**	0	0	0	1	1	1	2	0	0	1
wc	**8**	1	1	1	0	0	0	0	2	1	1
guestroom	**9**	0	0	0	1	1	1	0	1	2	0
office	**10**	1	1	1	1	1	1	1	1	0	2

0 = room combination can not be next to each other
1 = room combination can be next to each other

Allowed room combinations



Rules for creating the house



Evaluation of the house

1 Room: Number 1 = entrance
2 Room: Number 3 = kitchen
3 Room: Number 8 = wc
4 Room: Number 2 = livingRoom
5 Room: Number 10 = office
6 Room: Number 6 = bedRoom 3
7 Room: Number 5 = bedRoom 2
8 Room: Number 7 = bathRoom
9 Room: Number 4 = bedRoom 1
10 Room: Number 9 = guestRoom

Perimeter: 844
North Facade: 224
South Facade: 224
East Facade: 198
West Facade: 198

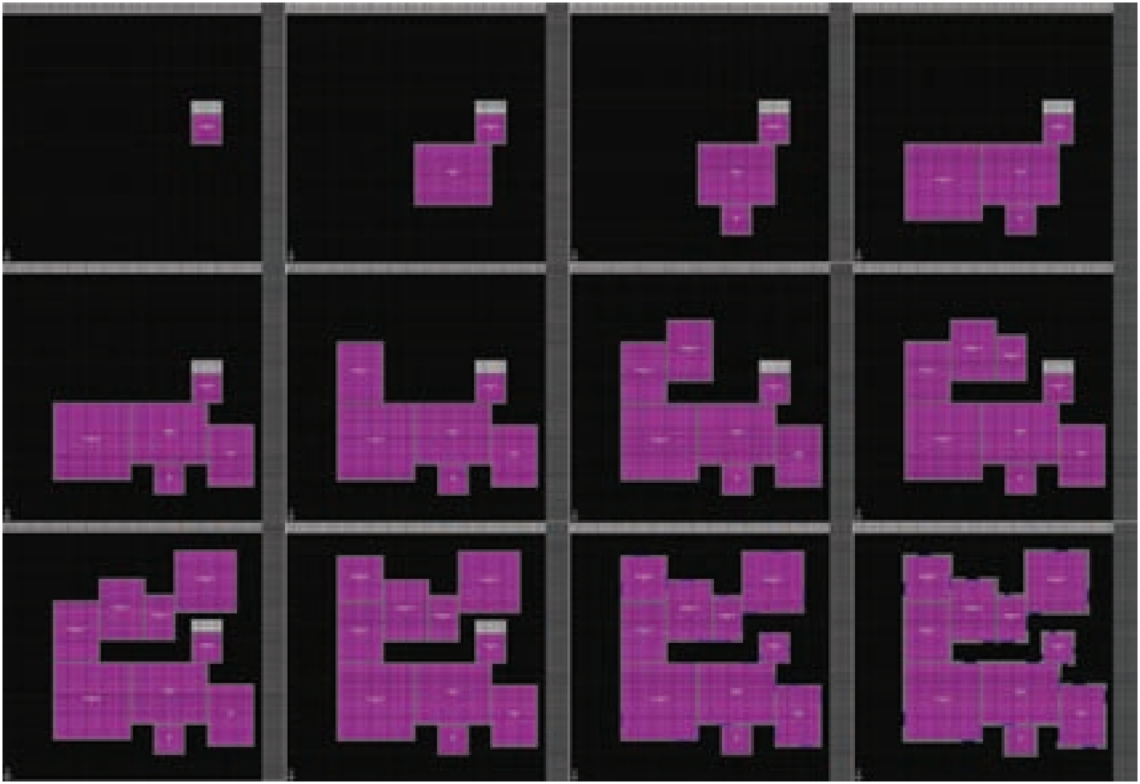
Temperature: 35 C
Evaluation of Temperature: 240

	DistL	DistD	Noise	Light	Up	Down	Left	Right
entrance:	74,000	35,000	288	31	0	0	0	0
kitchen:	103,000	59,000	293	36	0	0	0	0
wc:	132,000	59,000	293	28	0	0	1	1
livingRoom:	108,000	108,000	292	75	0	0	1	0
office:	117,000	20,000	249	60	1	1	0	0
bedRoom 3:	64,000	118,000	282	45	0	0	1	0
bedRoom 2:	50,000	89,000	273	69	1	1	0	0
bathRoom:	55,000	65,000	291	25	1	1	0	0
bedRoom 1:	32,000	36,000	249	111	1	1	0	0
guestRoom:	30,000	118,000	276	78	0	0	0	0
total:			2786	558	4	4	3	1

Total Evaluation Of House: 3584.0

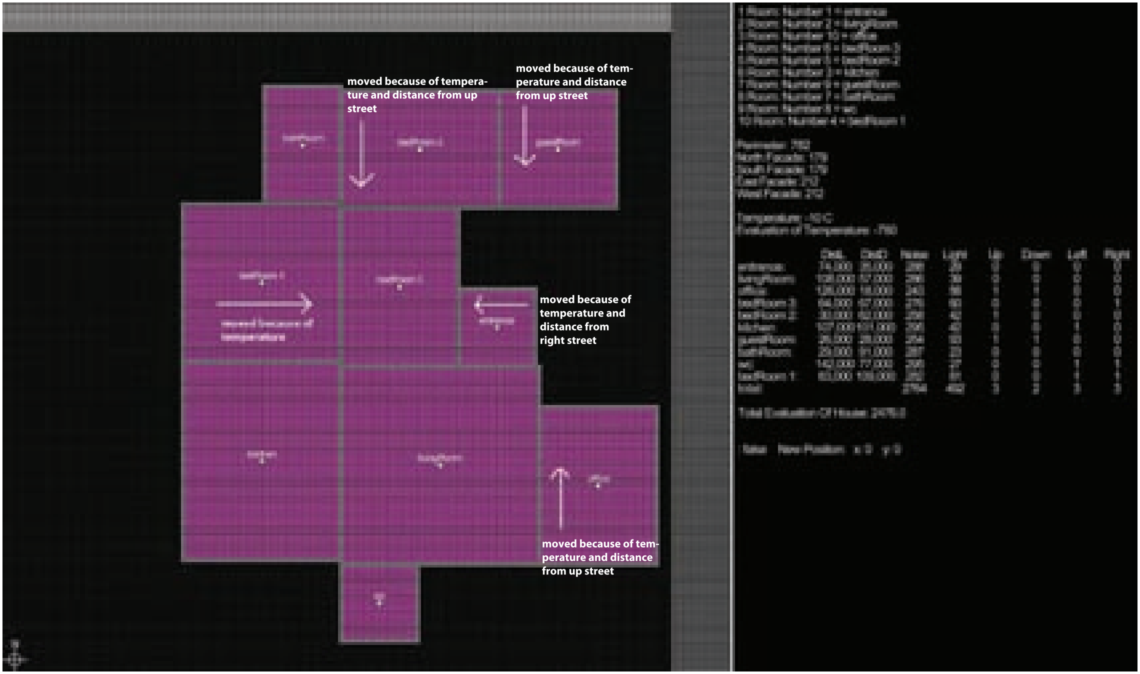
Information Table of the house (order of placed rooms, length of perimeter and facades, distances from streets, evaluation of noise and light, rooms that can be moved up-down-left-right)

The project results of a clear scripting procedure but it maintains the concept of the human's needs and the viability of the house.
The repetition, the multiplication, the genesis, the reproduction of the elements, they all relate to movement, the size, the history, the memory, the evolution.
Architecture finally becomes truly time-based and climate- based. It is no longer a simulation, not only in the isolated sectors of the design process but in the experience of the space itself.
Space in my project communicates actively with the users of the space in real time: they know each other, they flock together, space, nature- climate and people are becoming linked through a complex series of net-works.

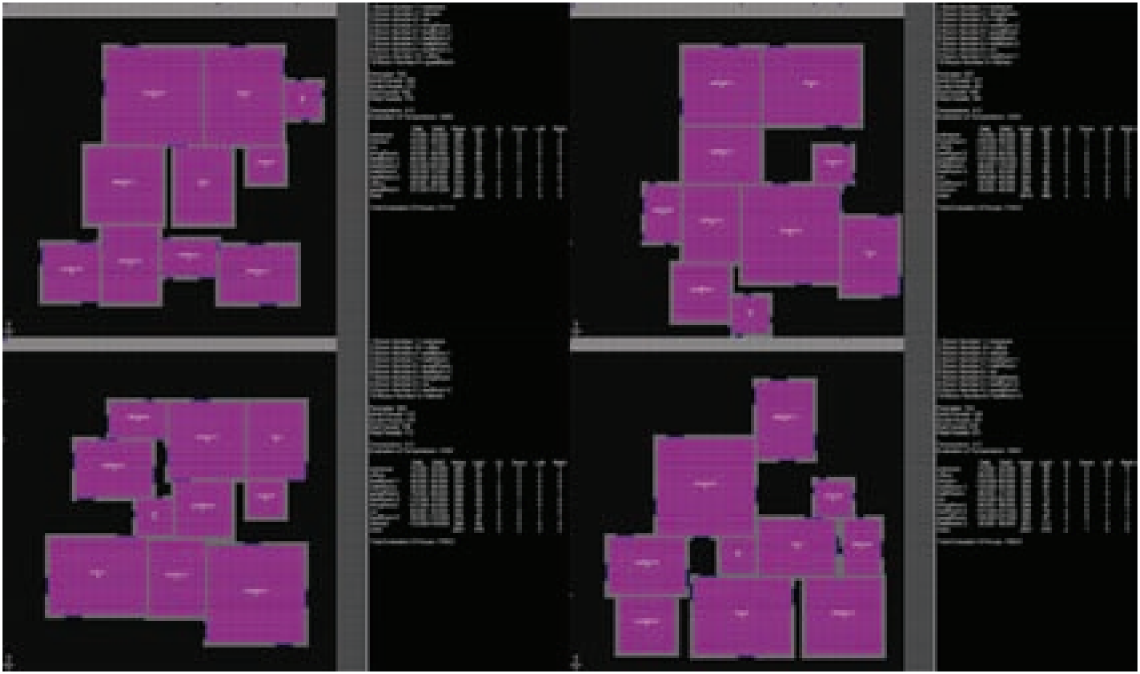


Sequential placing of the rooms, windows and the final house

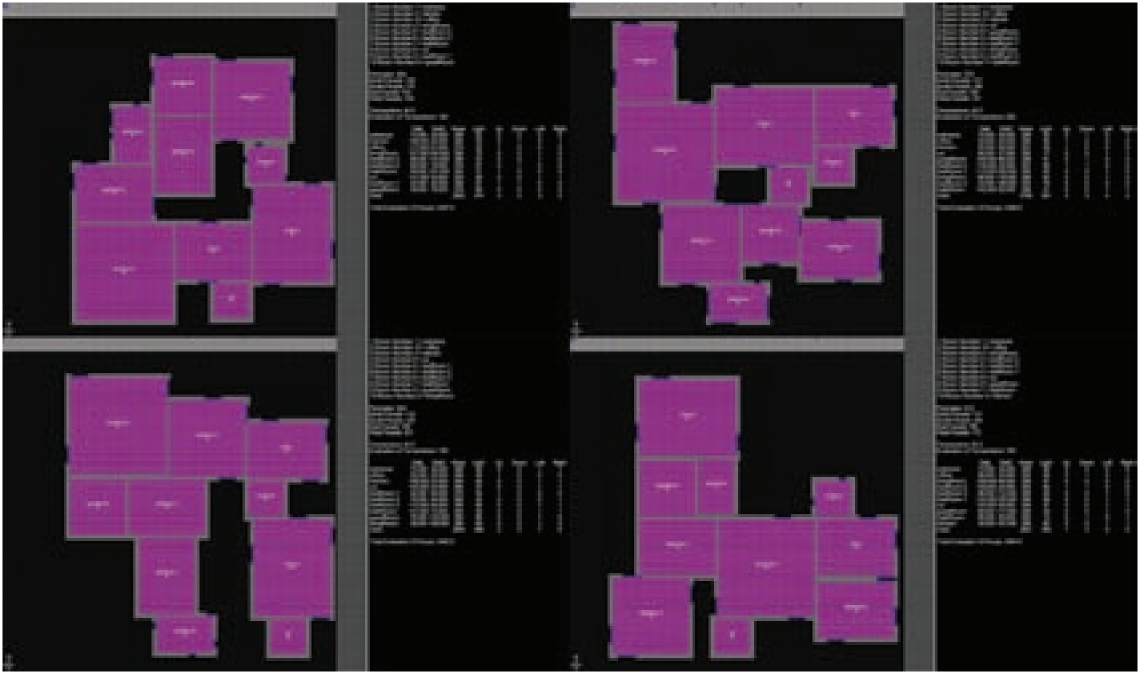
Floor plan before optimization



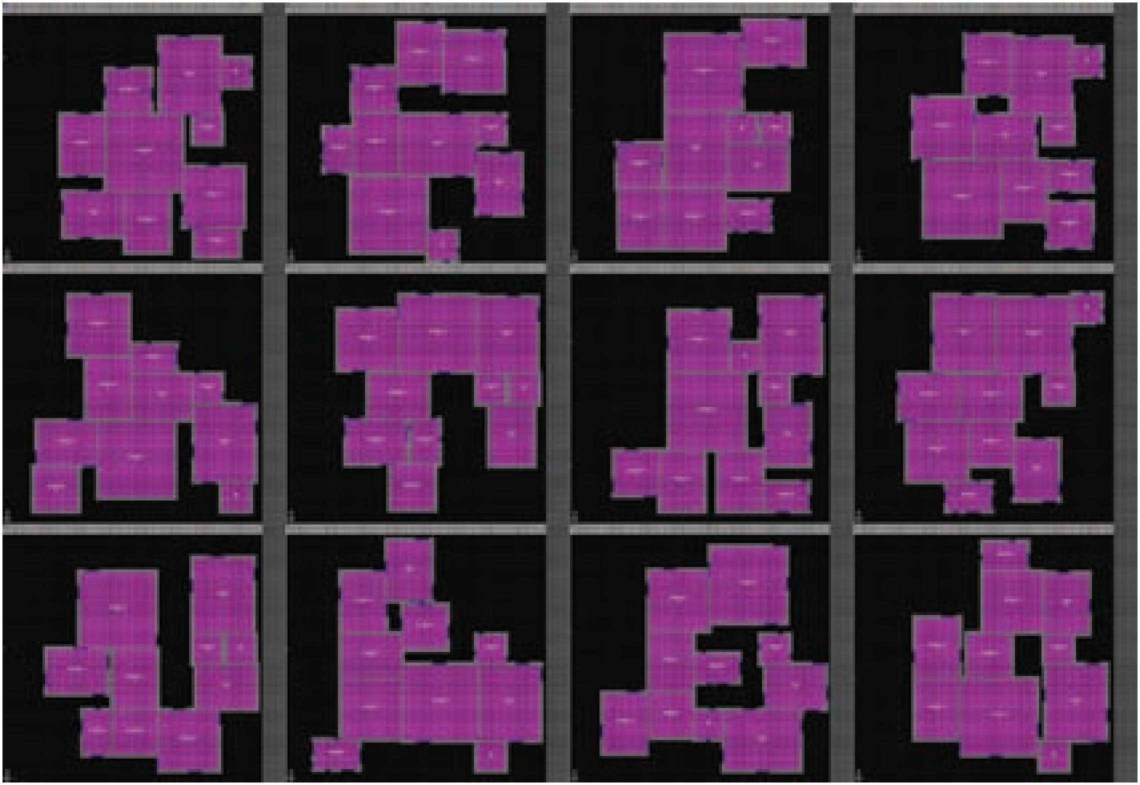
Floor plan after optimization



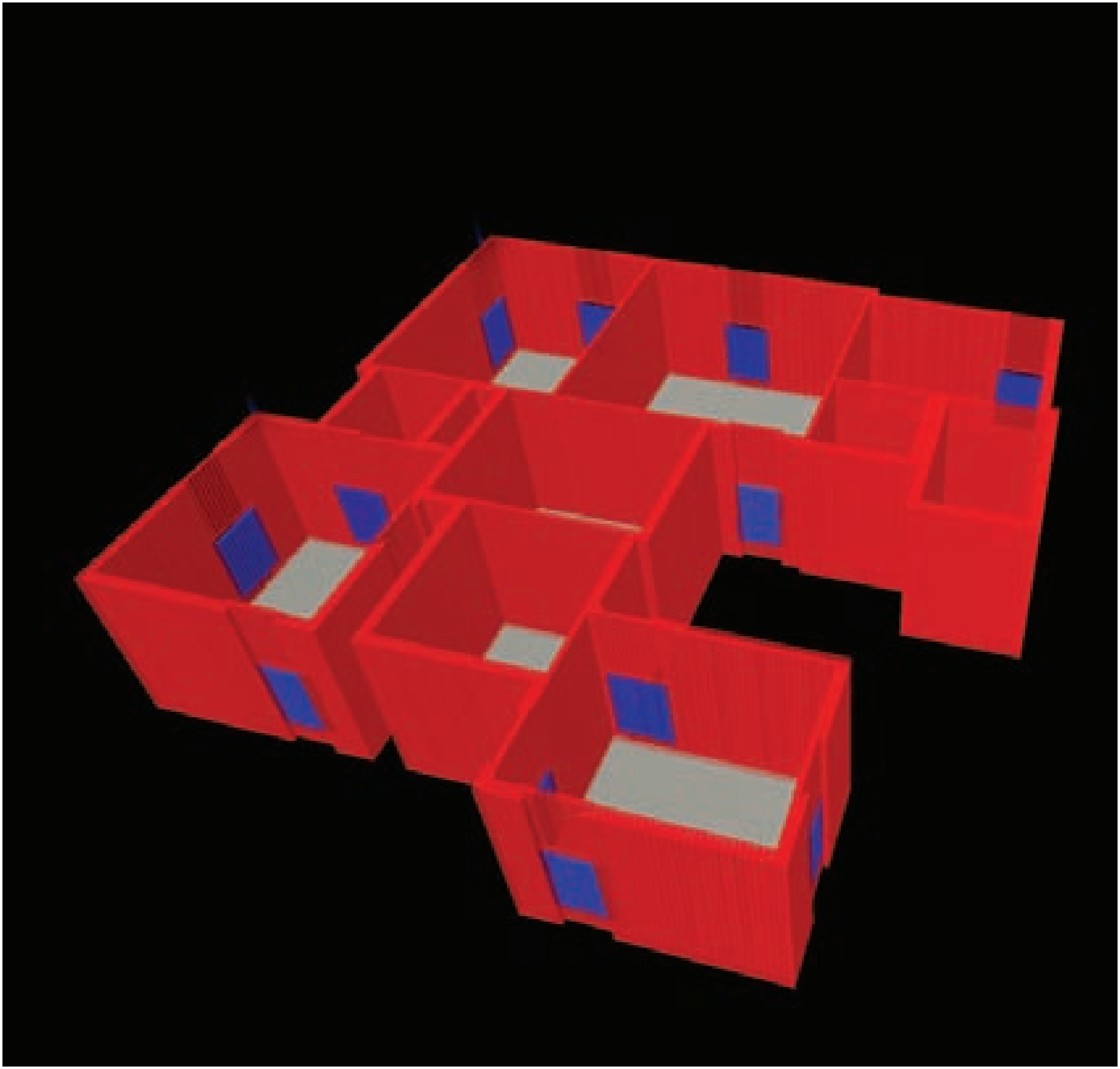
examples in temperature of -5 C



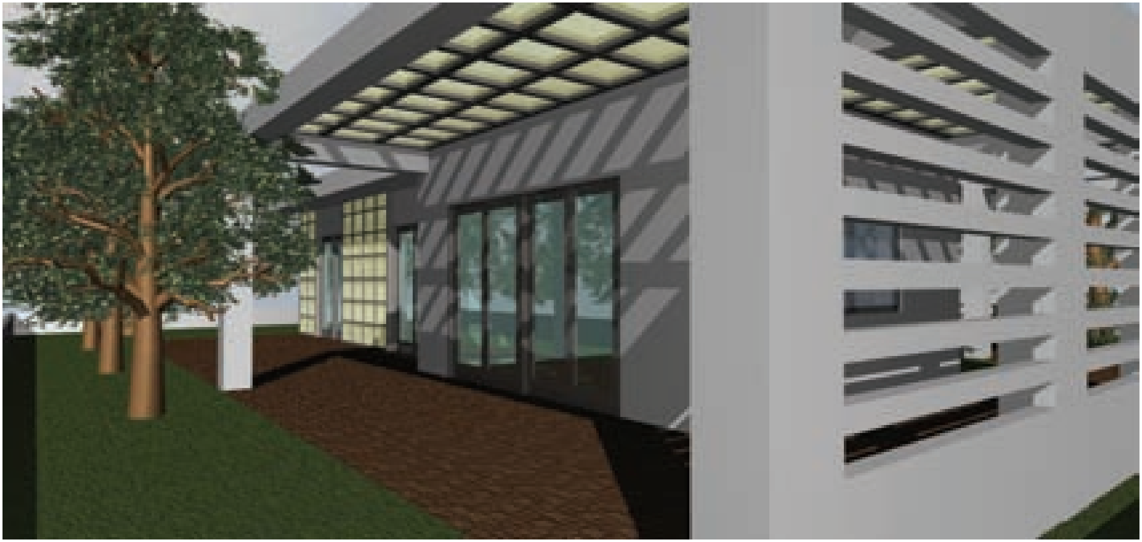
examples in temperature of +40 C



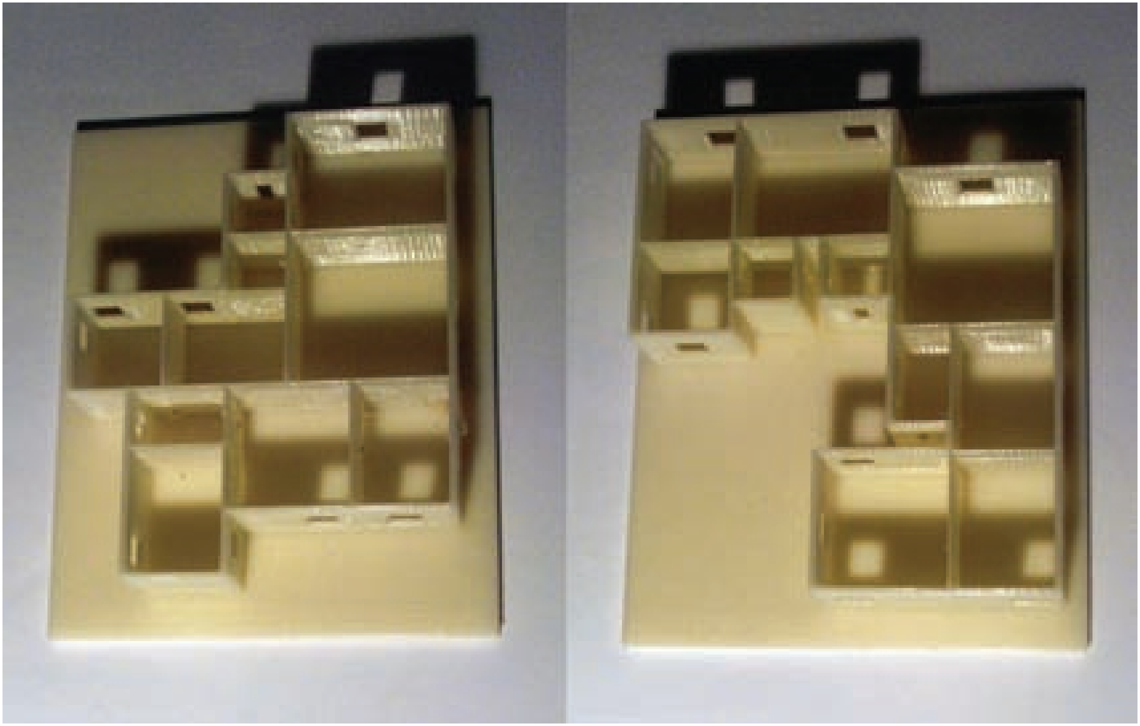
12 floor plans



3D in Processing



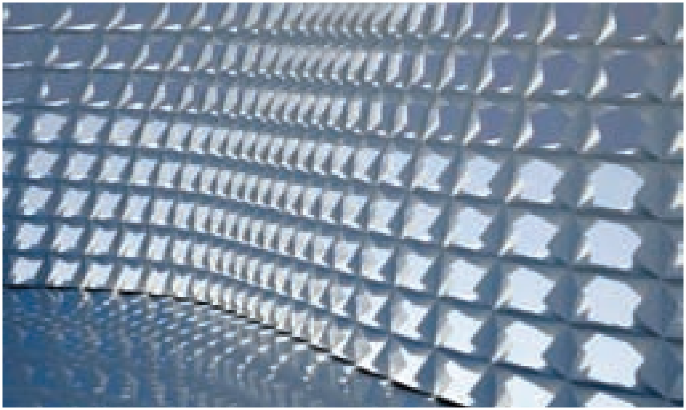
Renderings of the final house

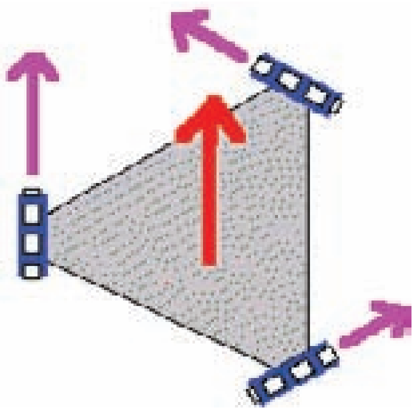


Models printed in the 3D printer

NEXT STEPS: THE VISION

This project serves as a way of visualizing the potential for a house that respects the nature and the inhabitants and has the ability to change during the time. In order to finalize the dwelling it is suggested to apply solar power panels for electric independency. Another important step would be the design of a parametric and movable façade that could control the introversion and the extroversion of the house. Lastly, the application of special systems for moving the rooms in real time would end in a smart and flexible house.





solar panels, parametric façade and moving systems

CONCLUSION: THE BUILDING BECOMES THE INSTALLATION

My project tries to give a solution to the ultimate task of architecture, which is to inform its new born structures in real time. The design task of the information architect is how to keep the process alive and apply meaning to the behavior in real time. How can the designers tunnel a continuous stream of data to and from the built structure and give meaning to the shape and content of the structures changing in real time? These dynamic buildings I regard as running processes, which are continuously informed and which continuously inform other running processes. They are active nodes in a complex adaptive operational network.
The building becomes an active installation where numerous actuators are constantly communicating with other actuators, their users and their environments. We know from practice that in each building there is already a large portion of the budget dedicated to the electromechanical installations up to 30% of the total budget. In the bright future of dynamic buildings the whole structure will be interpreted as an installation. Project the actual trends into the near future it makes sense to regard all constituting components of the built structure as active members of the installation. The building becomes a live organism, it becomes the installation.

Prototype of Architonic Concept Space III, designed by Oskar Zieta and the CAAD Faculty at ETH Zurich

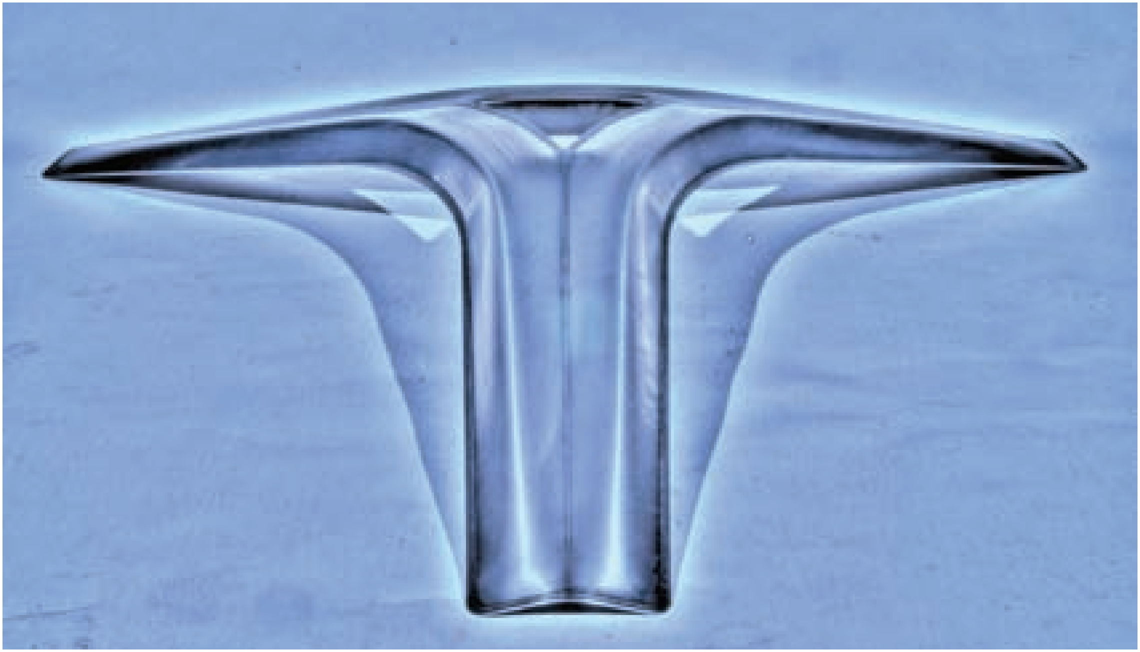
RE-BREAKING THE MOULD:
THESIS BY KENT BROCKMANN

Re-Breaking the Mould:
A Simulation of Pressurized Deformation as a strategy for design development and testing.

The FiDU process of precision-cutting, welding, and inflating of metal sheets, developed by Oskar Zieta and Philipp Dohmen at the Department for CAAD at ETH- Zurich, is an innovative method of producing light weight metal objects and assemblies. My research during the year has focused on many things concerning the FiDU process; new project concepts, concepts for producing specific changes in deformation, attempting to derive 2D shapes from desired 3D forms, learning proper basic welding techniques, acquainting myself with the KUKA Robot and Fronius welding equipment, and helping to establish dependable working conditions at the new SEA Hall, where FiDU operations are being placed.

However, despite gaining experience with all of these important activities, the first - and most valuable - lesson I have learned is that the development of each FiDU project requires numerous trial and error steps entailing hours of prototype production, resulting in potentially vast expenses in time and material usage.

Therefore, what I imagine being most valuable to the efforts of the FiDU Institute, and its mission for developing this technique, is to have the capability to accurately simulate the deformation of metal sheets before any production steps, and then judge accuracy and viability by testing the results of simulation itself against those of existing FiDU prototyping processes.



FiDU - What is it?

FiDU - an acronym of the German terms - Freien Innen Druck Umformung, or Free Pressurized Deformation. This technique transforms flat sheet metal into dimensionally stable elements. Two thin metal sheets are welded together and then simultaneously deformed and stiffened by the application of internal pneumatic pressure.

In previous load tests, the strength of FiDU elements have been partially evaluated. The indication of high stability and strength of these sheet metal elements with such low weight and material consumption shows innovative potential for not only cost savings, but for the nearly infinite forming possibilities. Pound for pound, FiDU elements are much more efficient that comparably performing components and due to their low weight, their transport and assembly are much easier.

The objective of the FiDU Institute at ETH-Zurich is to bring these elements to production level, and the aim of my research is to help examine and develop methods which make this possible.

FiDU - How is it done?

The basic FiDU process has always relied upon the flexible ability to generate nearly any form from a 2D pair of metal sheets which are drawn in various CAD software, and then sent out to a contracting firm and cut by CNC laser equipment.

While the FiDU Institute has gone to great lengths to minimize fully assembled prototypes, and learned a considerable amount of prediction capability through accumulated expertise, in order to fully realize the results of each and every shape attempt, it has been a necessity to do a small series of test productions to create a base of parts which can be further tested.

After production of the metal sheets, the parts must be transported to another location where the joining process can take place.

Most recently, the new SEA Hall facility of CAAD in Schlieren, Switzerland has been opened specifically for this task. Using a computer-controlled industrial robot, fixed with a gas welding head, the geometry of the desired part is fed again into a new set of equipment, which must be referenced manually by the welding operator.

This process has relied upon a large amount of expertise again, and is relatively easy to execute. However for clarity's sake, the prototyping process is now well into its second or third day, if not longer, and the information, parts and material have all travelled a considerable distance.

Once the welding process is complete, the parts are ready to be deformed, which also can take place at the same facility. This is perhaps the least precise part of the entire procedure, thankfully it is also the most simple to perform. Air or liquid is introduced to the space between the sheets, and the expansion begins. Being a completely manual procedure though, the only way to check fitness and usability of finished parts is by carefully monitoring the inflation process, sometimes destroying parts along the way. Also, if a series of parts is needed for a particular prototype, there is yet no guarantee or method that can deliver the inflation medium at a consistent rate for a specified time. This process can also occupy more than one day if certain geometries are less suitable for inflation than predicted, requiring a variety of inflation steps.

Until now, this basic process has been the mainstay of testing and acquiring potential for new geometries, and it has required the organization of many different people at different locations, the physical transport of materials, and the use of testing time, labor, and resources.

The results are always fulfilling, of course, since they are tangible parts which can be examined in person.

Currently, the prototyping process is identical to full production - with exception to the volume of parts, and must be repeated for different geometries if parts are found to be unusable.

My work is not a critique of the FiDU process, as FiDU already represents an innovative and flexible method for producing nearly any size and shape of construction part.

My proposal is for an added step strictly at the prototyping phase to substitute dependable and repeatable visualizations and data for an otherwise time consuming process which should be reserved for verifiably suitable geometries. In essence, **FiDU will now be prototyped!**

But how to go about prototyping such a physically bold and dynamic process?

The Finite Element Method

The benefits of using computer simulations in production and research are indisputable, and include more efficient utilization of resource and cost saving from this increased efficiency. Indeed, in many fields, computer simulation is integral and therefore essential to business and research.

A variety of specializations under the umbrella of the mechanical engineering discipline (such as aeronautical, biomechanical, and automotive industries) commonly use integrated Finite Element Method (FEM) in design and development of their products.

In structural simulations, FEM helps tremendously in producing stiffness and strength visualizations and also in **minimizing weight, materials, and costs**.

FEM allows detailed visualization of where structures bend or twist, and indicates the distribution of stresses and displacements. FEM software provides a wide range of simulation options for controlling the complexity of both modeling and analysis of a system. Similarly, the desired level of accuracy required and associated computational time requirements can be managed simultaneously to address most engineering applications.

FEM allows designs to be constructed, refined, and optimized before the design is manufactured.

This powerful design tool has **significantly improved** both the standard of engineering designs and the methodology of the design process in many industrial applications. The introduction of FEM has substantially **decreased the time to take products from concept to the production line**. It is primarily through improved initial prototype designs using FEM that testing and development have been accelerated. In summary, benefits of FEM include increased **accuracy, enhanced design and better insight** into critical design parameters, virtual prototyping, fewer hardware prototypes, a **faster and less expensive** design cycle, **increased productivity**, and **increased revenue**.

FiDU goes FEM

At ETH-Zurich, FEM software is available to campus-wide users in many fields of Engineering and Mathematics. Previously, for the FiDU Institute, and the Department for CAAD, FiDU visualization was thought only possible through very inappropriate CAD 3Dmodelling programs such as Maya and Rhino. These types of modelling techniques typically produced results which could not be repeated or verified at best, and did not at all produce a characteristic response based on real materials.

In my long search for producing a realistic and reliable method for prediction of deformations, I surmised that FiDU was a prime candidate for simulation in FEM software.

The basic approach is illustrated in a preliminary model developed through the invaluable efforts of the Computational Physics, and Material Testing Departments at ETH-Zurich using Abaqus/CAE software.

Abaqus is a commercial software package for finite element analysis developed by HKS Inc of Rhode Island, USA and now marketed under the SIMULIA brand of Dassault Systemes S.A.

Abaqus/CAE provides an integrated modelling (preprocessing) and visualization (postprocessing) environment for the analysis products. Abaqus uses the open-source scripting language Python for scripting and customization.

Abaqus is used in the automotive, aerospace, and industrial product industries. It is popular with academic and research institutions due to the wide material modeling capability, and the program's ability to be customized. It also provides a good collection of multiphysics capabilities, making it attractive for production-level simulations where multiple fields need to be coupled.

Abaqus was initially designed to address non-linear physical behavior; as a result, the package has an extensive range of material models. Its plastic and elastic material capabilities are particularly noteworthy in the case of FiDU, since it is able to effectively apply and treat the physical properties of many metals.

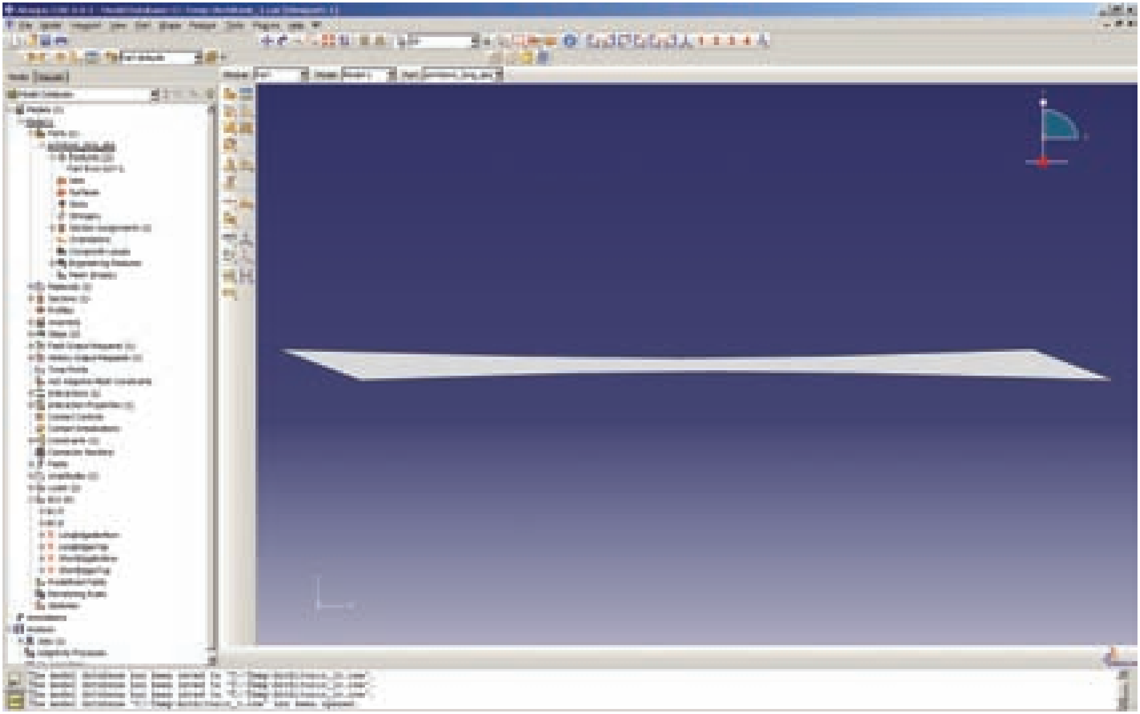
CAD geometries can be imported into Abaqus in any number of formats, for the purposes of this simulation, DXF data was used since it represents the ubiquitous CAD standard.

By creating this “division of labor”, the designer is not burdened to produce special data formats or learn a new software package in order for his or her ideas to be simluated. It also frees the production of geometries from the simulation process, assuring a particular freedom in the design process overall.

First, a geometry is chosen, drawn traditionally in any CAD software program and saved as a DXF. The model demonstrated here is sampled from the Architonic Concept Space III, designed by Oskar Zieta and the CAAD Faculty at ETH Zurich.

Then the geometry is prepared for transfer to Abaqus. If a geometry exhibits certain axes of symmetry, these redundant "mirror" regions can be omitted from the simulation as Abaqus can manage their presence by careful consideration of selected degrees of freedom.

The geometry is imported and used as the basis of the part geometry, in the form of a sketch.



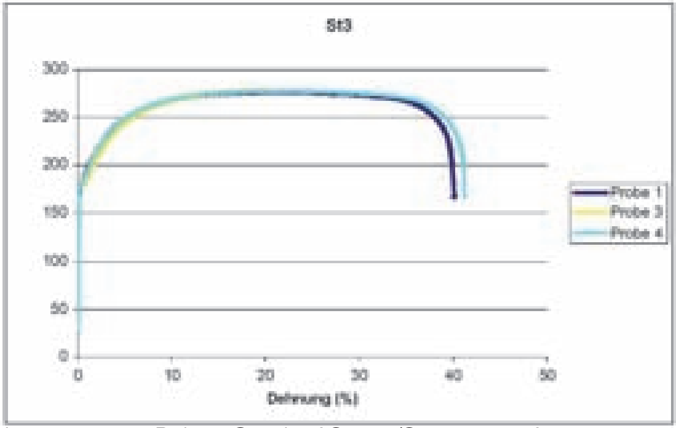
Formation of a sketch object in Abaqus, this model contains two seperate sheet parts for anaylsis.

The sketch is then imbued with dimensional and material characteristics, which are used at the basis of the modelled part. This step is most critical of all, since it is only by the accuracy of the material data that an accurate simulation can be made. Of primary interest are Tension and Cyclic Loading data for a given metal. For this simulation, to date, only Tension test data was available, but it provides an already acceptable demonstration of a deformation.

Zug-Versuche

Probentyp: Flachzugproben (Hundeknochenform) 60 mm parallele Länge, 0.8 mm Dicke, 10 mm Breite

Probe	Geschw. [mm/min]	$Rp_{0.2}$ [MPa]	Rm [MPa]	Ag [%]	Af [%]	Bruchstelle; Bemerkung
St3_1	3.6	175	276.5	18	38.7	Mitte; ungefähre Messwerte
St3_2	3.6	-	-	-	-	Mitte; Extensiometer nicht in Betrieb
St3_3	3.6	181.6	278.1	21.5	41.5	Mitte
St3_4	3.6	178.2	277.7	21.3	41.08	Mitte
Durchschnittswerte		178.3	277.4	20.3	40.4	
RostenderStahl_1	3.6	200	316.3	>17	39	Mitte; ungefähre Messwerte (Extensiometer nicht von Anfang an in Betrieb)
RostenderStahl_2	3.6	240	316.1	<20	38	Mitte; ungefähre Messwerte (Extensiometer nicht von Anfang an in Betrieb)
RostenderStahl_3	3.6	207.1	317.3	21.8	41.33	Mitte
RostenderStahl_4	3.6	189.4	315.9	23.2	42.01	Mitte
Durchschnittswerte		209.1	316.4	20.5	40.1	
RostfreierStahl_1	3.6	281.1	693.2	69.5	74.39	Mitte
RostfreierStahl_2	3.6	279.9	696.5	65	70.15	Mitte
RostfreierStahl_3	3.6	280.8	695.5	60.2	65.19	Mitte
RostfreierStahl_4	3.6	279.7	697.5	59.9	64.87	Mitte
Durchschnittswerte		280.4	695.7	63.7	68.7	



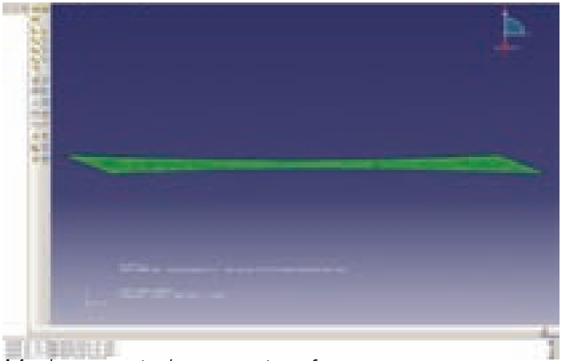
Above: Material Data sheet *Below: Graph of Stress/Strain curve from testing of a steel sample*

Once the material characteristics are loaded into the part model, a mesh is generated over the part. The mesh is critical in terms of modelling accuracy and computational cost, and can be adjusted manually, or by scripting with regards to where the significant effort and accuracy is needed.

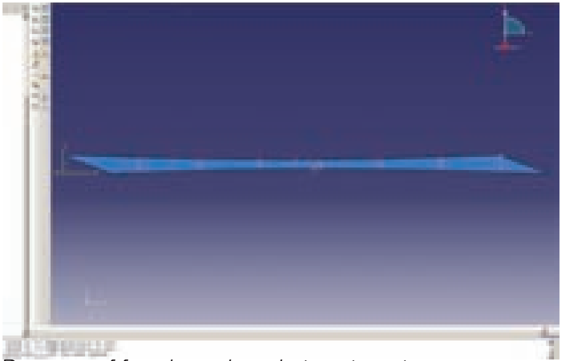
For FiDU the mesh can be made finer in whole regions of complex geometric boundary conditions, where deformation is likely to be large and somewhat unknown, and along edges where the most deformation and material stresses are likely to occur.

After meshing the part, boundary conditions and loads are defined, in this case pneumatic pressure in the appropriate direction to act on the meshed part as air would from within the welded assembly.

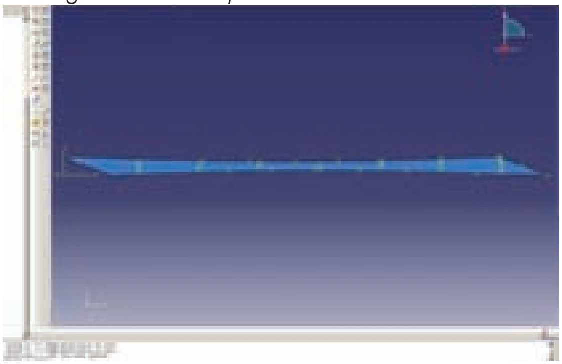
Once the load steps are established, the file can be submitted for processing.



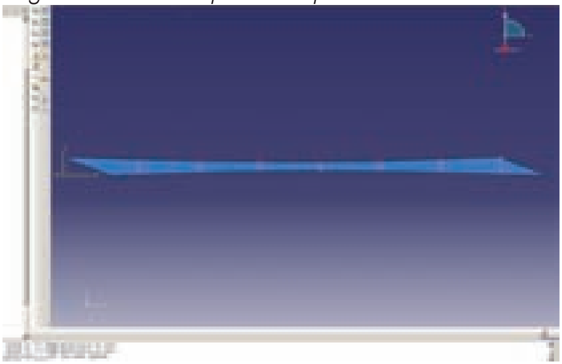
Mesh generated over part surface



Degrees of freedom placed at part center



Areas of interaction are defined across part surface

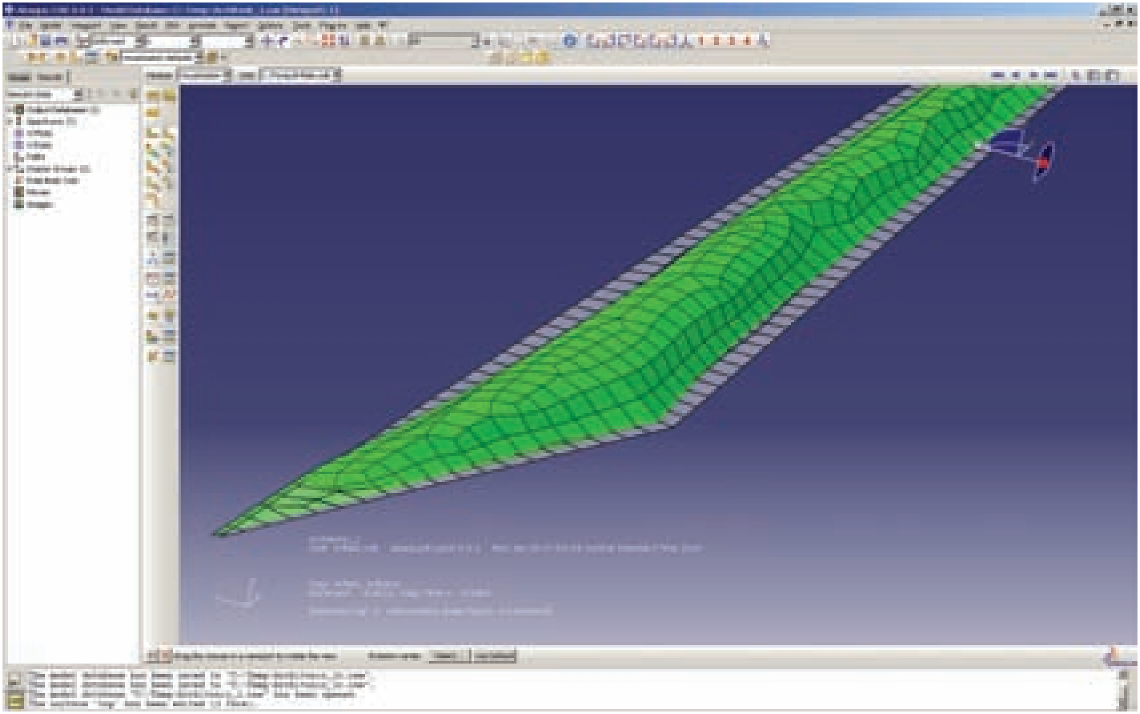


Load distributed across surfaces of part

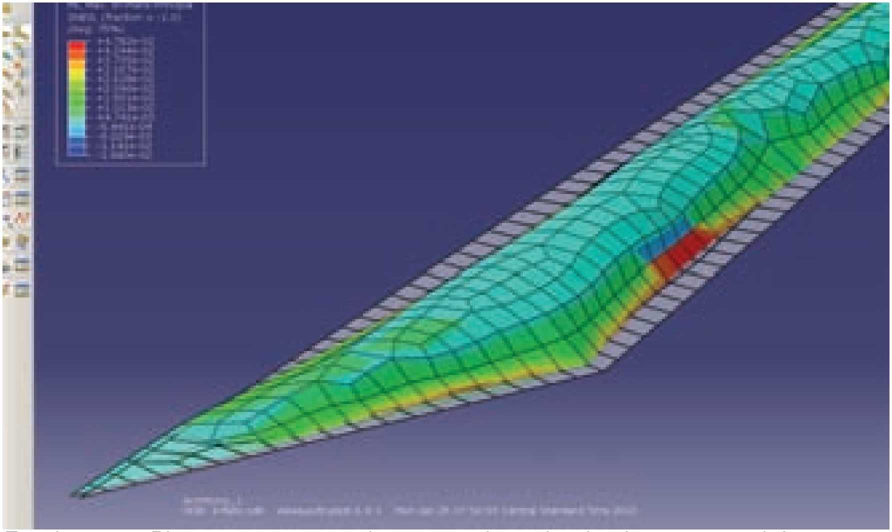
In the case of the first simulation, over 1.1 million iterations were calculated by the software, resulting in a calculation of approximately 4 hours. This is the time range for a typical desktop computer with dual-core processor.

Quantitative data such as Von Mises stress can be displayed and is used to predict yielding of materials under any loading condition from results of simple uniaxial tensile tests. Since it can be presented in 3D in the resulting deformed mesh part, it to helps show where FiDU parts have undergone plastic deformation, and will provide a basis for further research to strengthening parts through work hardening.

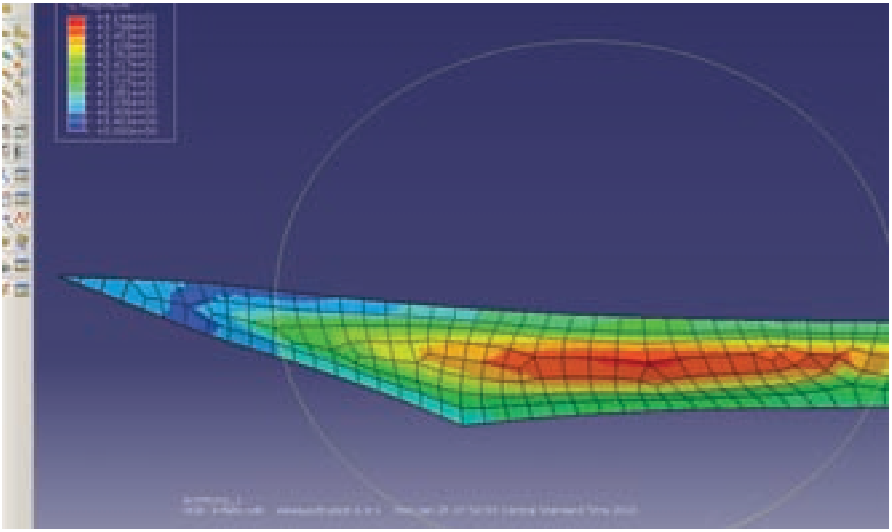
Qualitatively, we can also see the potential for FiDU parts to exhibit movement during the deformation. Whether anticipated or not, the value of this visualization alone is paramount to nonphysical testing as a viable method for FiDU prototyping, since we can either attempt to provide FiDU elements with a “behavioral” nature, or choose to omit these possibilities when necessary. These topics are also the basis for further research.



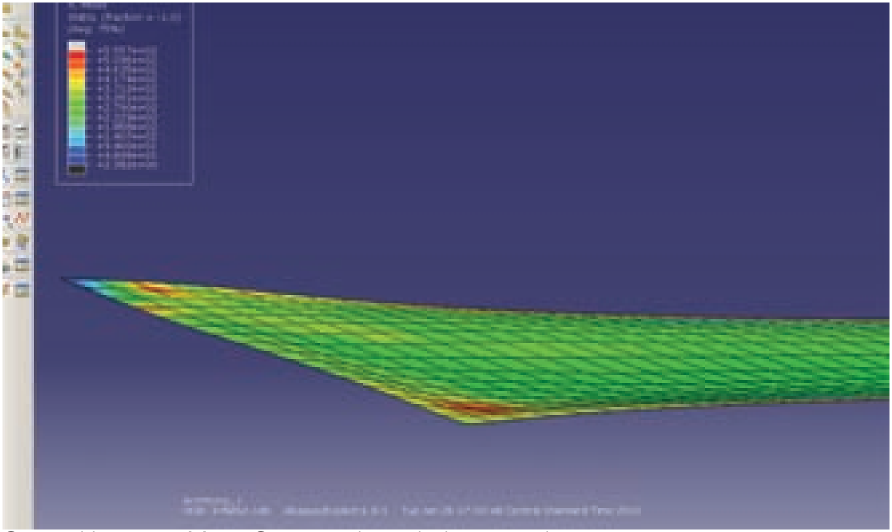
Resulting model after simulation showing deviation from original geometry in the FiDU part.



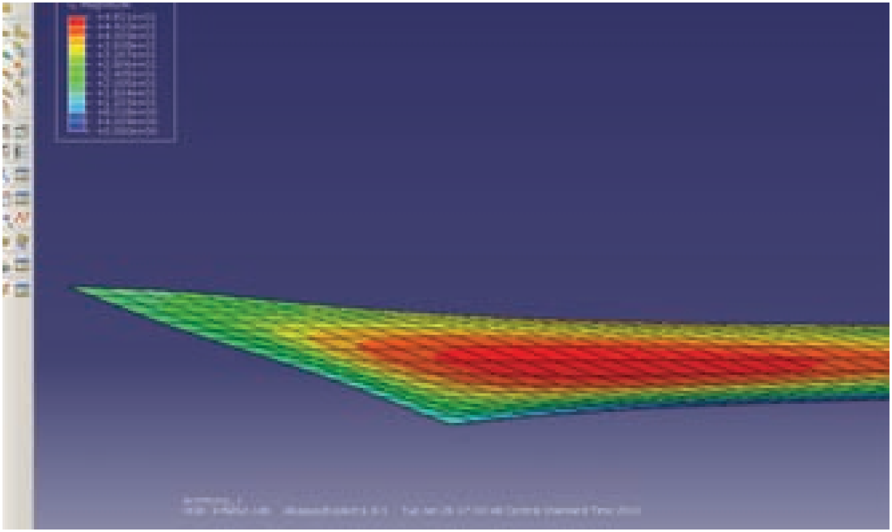
First Iteration: Plastic stresses are shown to indicate levels of permanent deformation



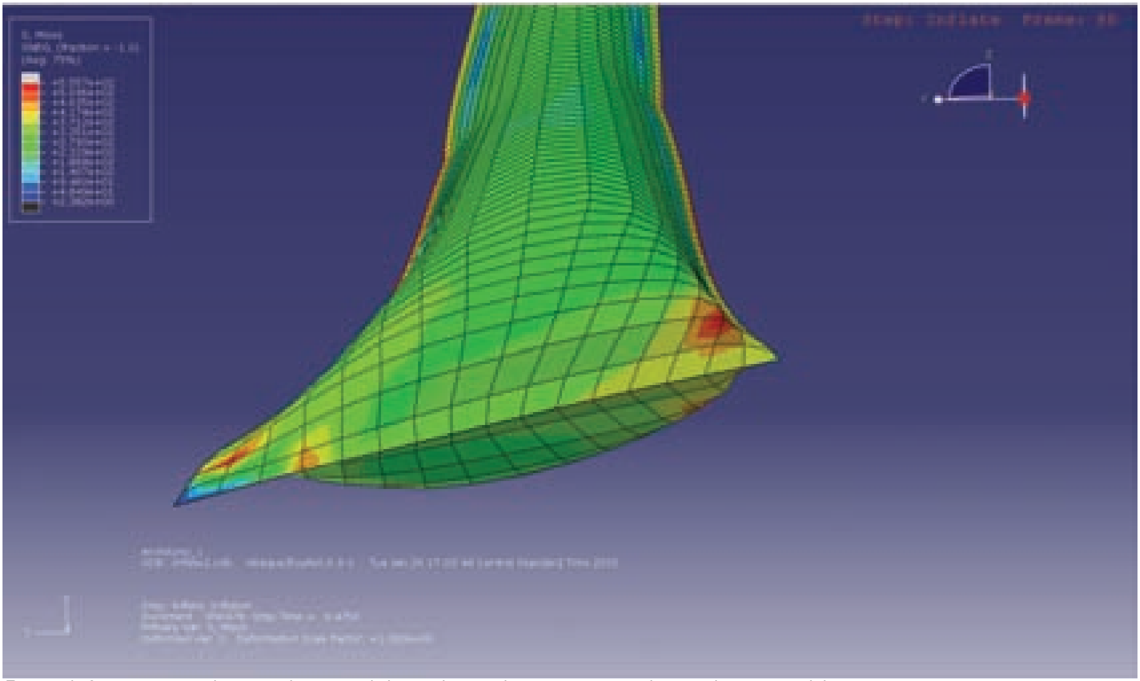
First Iteration: Post-deformation change in height across the part geometry



Second Iteration: Mises Stresses through the revised part



Second Iteration: post-deformation change in height across the part geometry



Post-deformation edge quality, model pinching shows potential to indicate problem regions.

FiDU is simulated

By general comparison, after my 2 month period of introduction to and learning the interface, this simulation took approximately **one to two days of modelling, setup, and calculating**. Once the calculation is finished, it is also possible to export the final data and recompose a 3D mesh file (.stl) to use as a basis for rapid prototyping on a 3D printer using various methods. This can serve to show possible shape or construction issues with multi-part assemblies. These are of course only preliminary steps in the overall goal to accurately simulate FiDU by digital means.

Further research specific to FiDU simulation should address these topics -

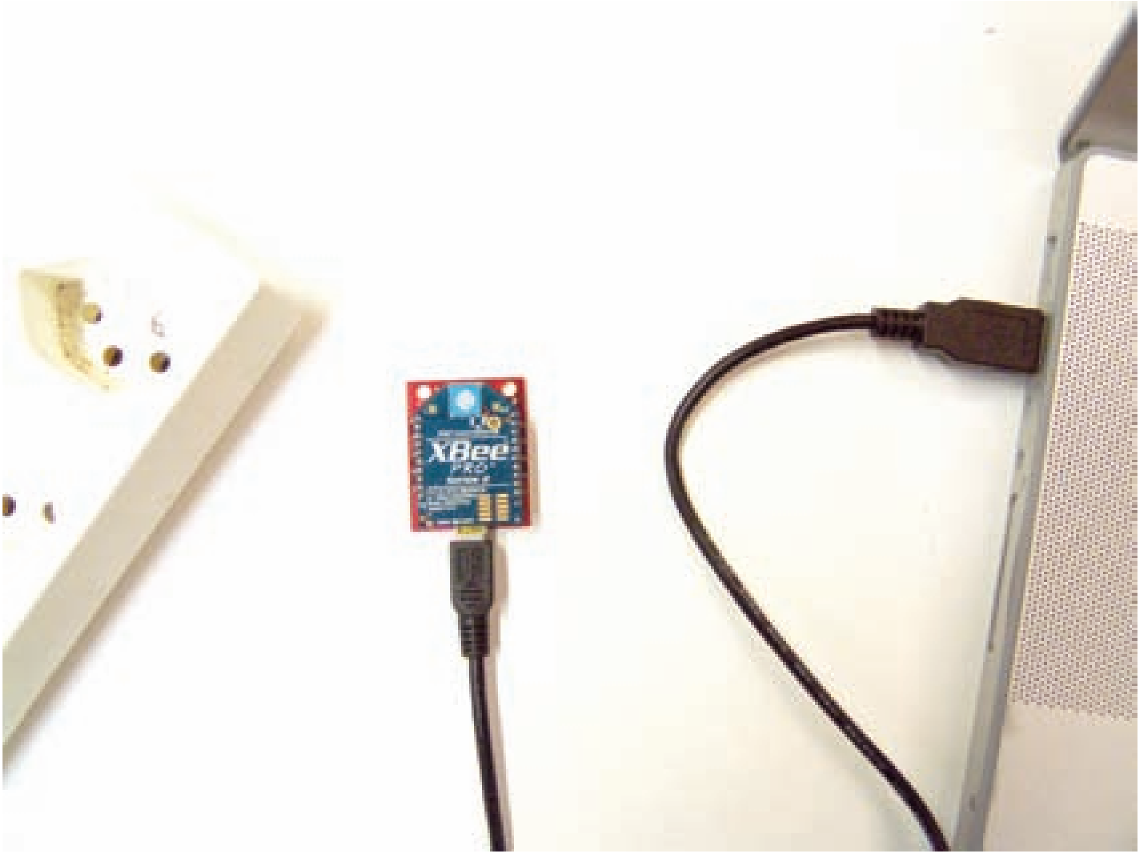
- parametrization of a pool of 2D geometries for optimization of model input data
- more accurate material properties, to include Cyclic Loading data, and of numerous materials
- an introduction of welding and heat effects and their inherent irregularities to multiple piece models
- simulation of a variety of inflating conditions, with multiple steps, at various flow rates and pressures
- simulate secondary manipulations to modify model part shape (bending of parts prior to and during inflation)
- external force simulations to demonstrate structural abilities (simulate destructive testing)
- development of a method to evaluate 3D results and resubmit adaptively from a pool 2D model geometries
- additional topics as they arise (they will)

While the ability to digitally simulate FiDU is in its infancy, it can already inform the designer on the previously mentioned basic quantitative and qualitative aspects. **It has the benefit of maturing into a fully viable and effective prototyping system which can be operated and evaluated economically and quickly.**

Special thanks:

Prof. Dr. Ludger Hovestadt and staff at ETH Department for CAAD:
Oscar Zieta
Philipp Dohmen
Mathias Bernhard

Also the invaluable assistance from these departments at ETH-Zurich
Dr. Falk Wittel - ETH IfB Computational Physics for Building Materials
Prof. Peter J. Uggowitzer - ETH Laboratory of Metal Physics and Technology
Serge Bilgeri - ETH DArch Informatics Support Group

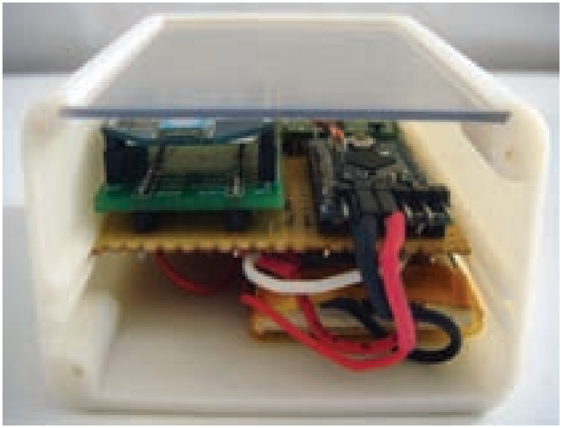

XBee
PRO

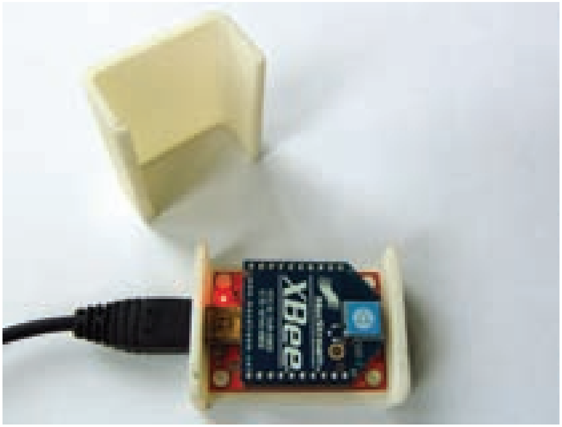
SMART POWER STRIP
THESIS BY GUNES DIREK

In the post-digital world we are living, its extremely important to make use of the information available everywhere. Using the information and interpret it the right way can be a key role to make us more conscious to what is going on around us. In this context, the energy issue, one of the most important problems of the last decade, is worth dealing with. Making public aware of the energy consumption does not have to be with statistics and global rates of data charts, but also by showing people how much energy they consume in the household, by small modifications of household objects, in other words, by “hacking” the existing system of energy distribution. This project follows a simple “hack”, that modifies a common power strip, in order to track power consumption that is distributed from it.
Several examples have been made about this topic, most famously, the “Tweet-a-watt” from ladyada, which hacks the famous energy consumption monitoring device Kill-a-watt, and makes it possible to wirelessly send the data using XBees, and send it over Twitter. This was a hack of a commercially available product, and measuring the power was not a part of the hack.

BACKGROUND

The final thesis was a continuation of previous experiences in embedded electronics. Getting sensor data from a light color sensor and directing it over serial to XBee and capturing the data and using processing to visualize the input was a small project done to explore the possibilities of XBee, wireless communication and the serial data flow over microcontrollers.

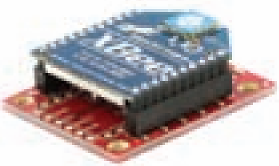




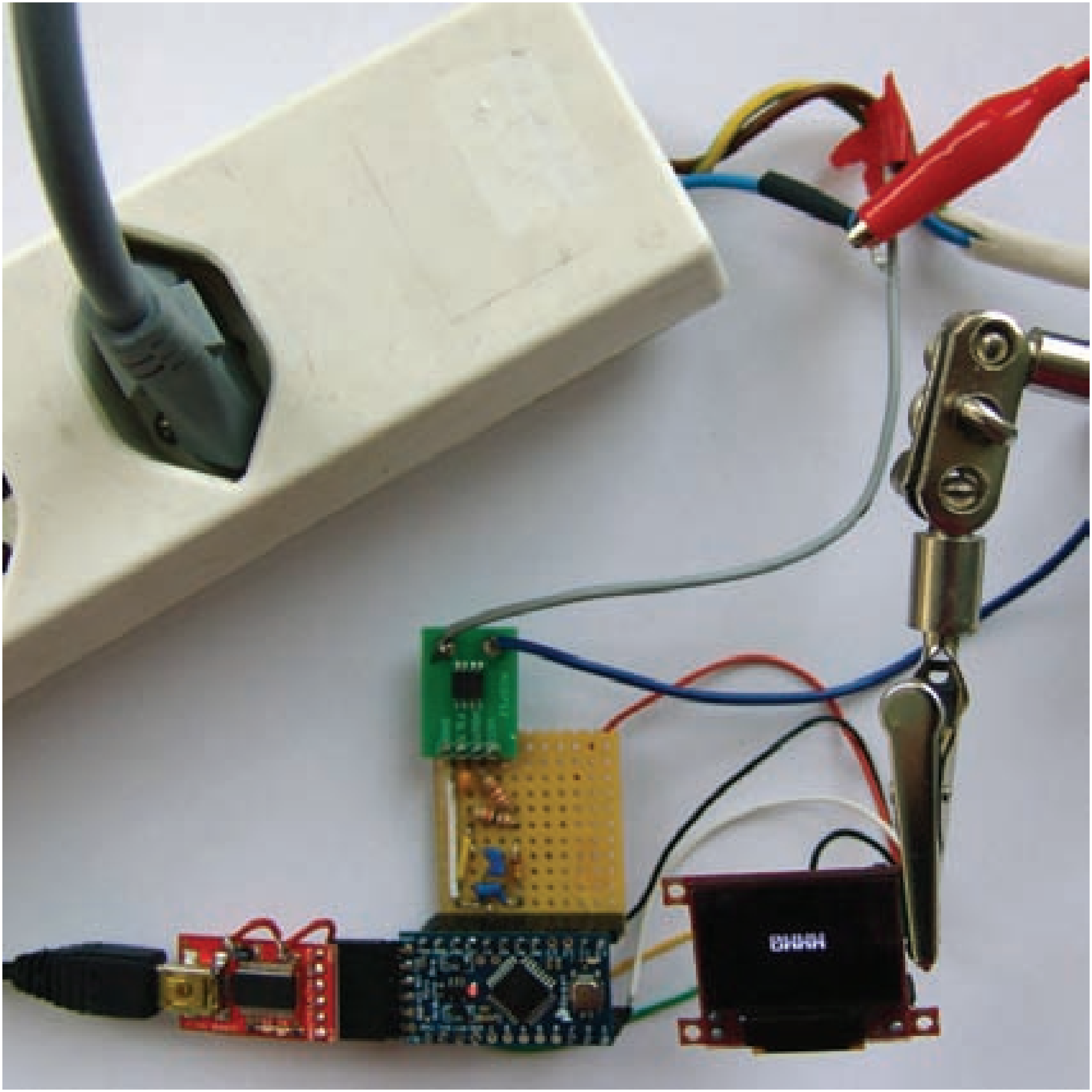
μOLED-96-G1: 0.96" OLED display from 4D Systems. This small organic light-emitting diode color display module has a built in serial application interface, and has 65,536 colors, which makes it easy to show color images, letters, graphics and any embedded visualization needs.

Arduino Pro Mini: The Arduino Pro Mini is a microcontroller board based on the ATmega168. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, an on-board resonator, a reset button, and holes for mounting pin headers. Powered by 5V.

ACS712 Hall effect based current sensor and breakout board from Sparkfun Industries: This current sensor uses hall effect to measure both AC and DC voltage. When the board is powered and connected to the current to be measured, it gives an output voltage , which is proportional to the measured current.

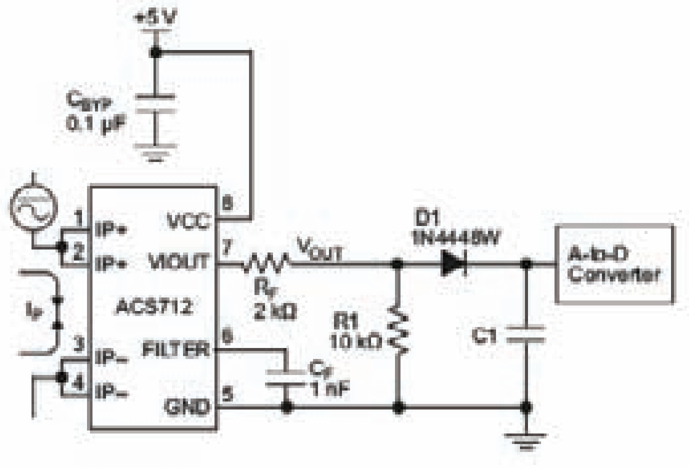


XBee transceiver radio module and XBee explorer board from Sparkfun Industries: This is a very popular radio module that allows wireless communication between two and more transceivers. The board regulates the voltage so the XBee can communicate with 5V systems.



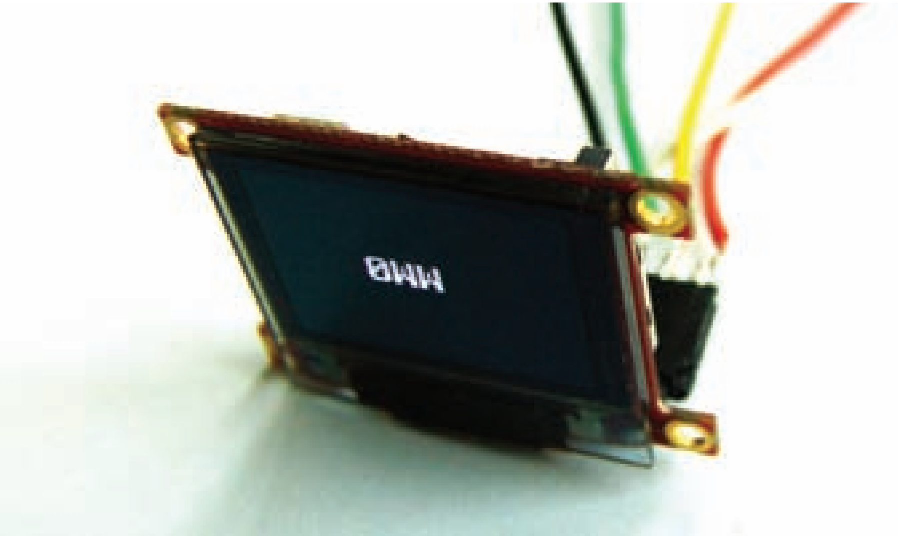
MEASURING CURRENT

First step of the project was to test the current sensor ACS712, which is based on what is known as the "Hall Effect", a well known electromagnetic effect used in many different applications all around the world. As hall effect is based on the effects of a magnetic field, hall effect sensors can be used in any sensing device that incorporates magnetic fields.
"In this generalized sensing device, the Hall sensor senses the field produced by the magnetic system. The magnetic system responds to the physical quantity to be sensed (temperature, pressure, position, etc. through the input interface." (Honeywell Inc. Hall Effect Sensing and Application)
Basically in the ACS712 the flowing current is used to induce a small voltage, which can be read using the output pin and the ground pin. This voltage is then converted in the Atmega128 chip of the Arduino Pro Mini, using a 10-bit analog-to-digital converter. That means the analog voltage -between 2,5 V - 5 V - fed into one of the inputs of the microcontroller is converted to a integer between 0 and 1024 (10-bit, 2 over 10 is 1024).
As a first test an interface, shown below, is built to scale and rectify the output of the sensor. This is necessary to have a successful reading with the microcontroller, because the measured current is AC, that makes the voltage output AC as well.
The reading is just a integer value showing how much voltage is at the output pin of the current sensor. To make this value useful, it has to be converted into watts or any familiar value. It is known that the output voltage is proportional to the current on the line. The regular voltage is 2.5 V, so it has to be between 2.5 and 5 V output on the pin. Mapping the voltage to digital values of arduino shows it is 4.9 mV per unit.
After converting the value into volts, charts on the data sheet of ACS712 help to convert it into amperes. This should be tested experimentally, because of irregularities of the output and the digital conversion.



OLED

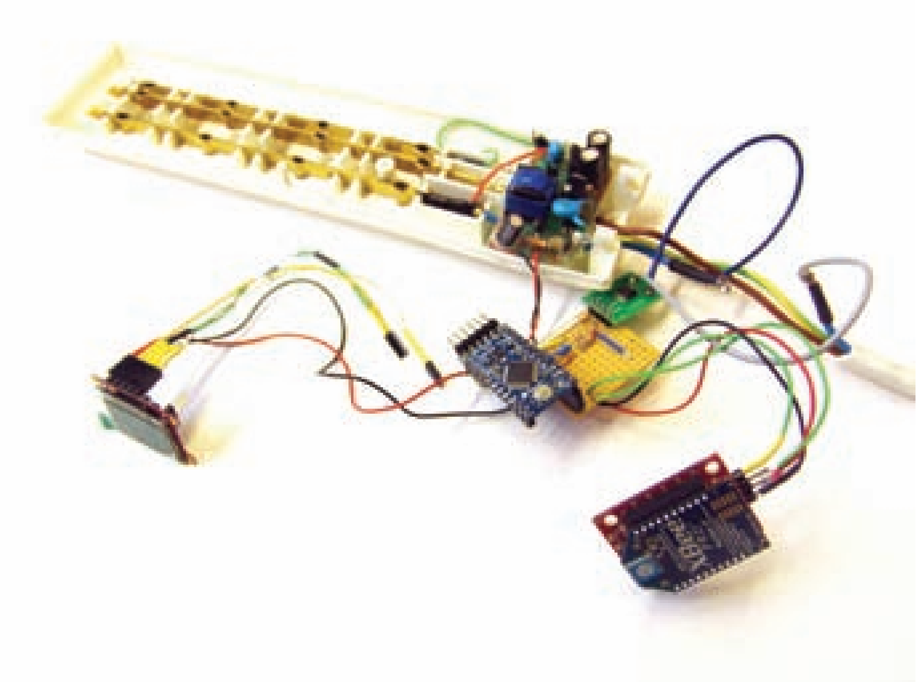
After having a successful reading as a digital output from arduino, the next step is to send the information to the OLED display, using the proper communication procedures used by the microchip embedded inside the display unit. This , at first requires an initialization procedure, in which the display is reset by pulling the reset pin HIGH, a command is sent to make the display auto-detect the serial baud rate, and another command to clear the screen.
In order to make the command set easier to write in arduino, a library is written to make the microcontroller communicate easily with the display unit. The original library from Oscar Gonzalez from 4D Systems is used as a base and the data sheet provided the remaining command sets from the application programming interface (API).
After the initialization, the arduino goes trough a loop, where it clears the display screen, receives an integer from the current sensor that corresponds to the current flowing through the cable, and sends it to the display placed inside a command from the API.



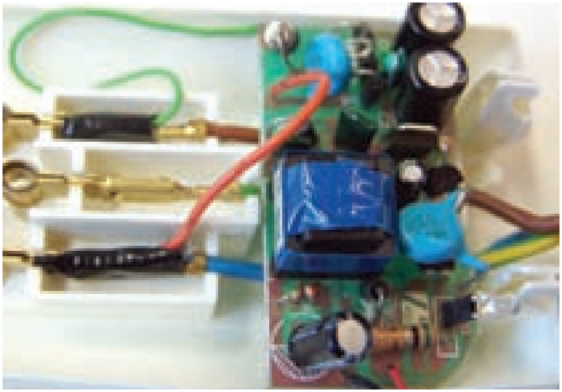
XBEE

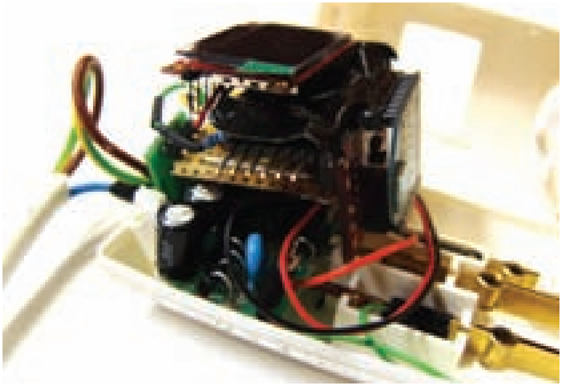
Next step in the development was to implement the wireless radio module XBee. In order to do that the XBee explorer board from sparkfun was used to step down from the main voltage of 5V to 3.3V which is XBees operational voltage. The serial logic levels also have to be regulated according to that voltage.

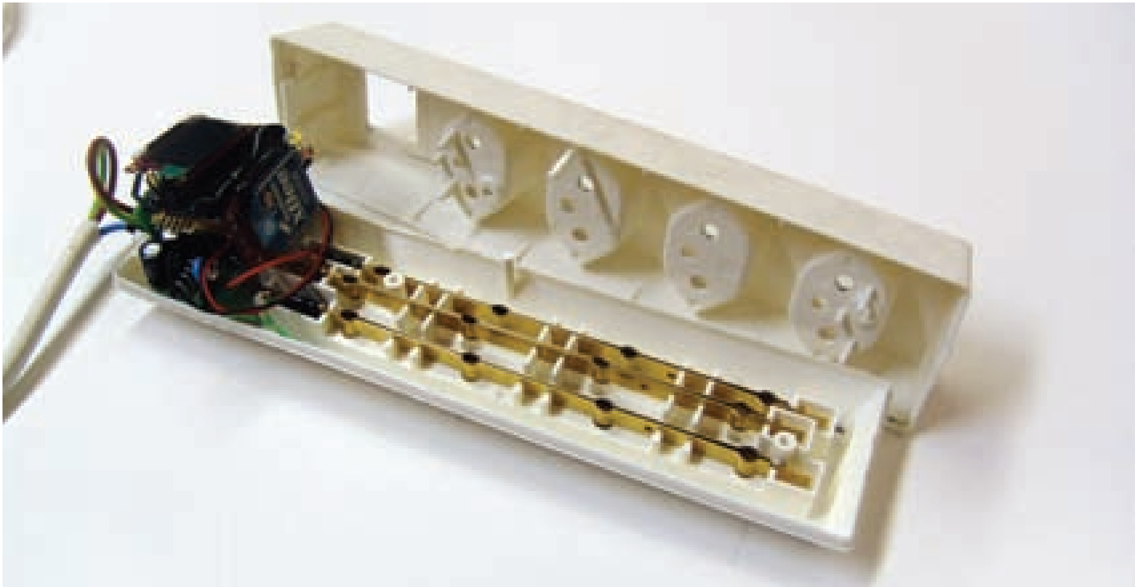
Adding XBee to the system had one problem, Arduino Pro Mini doesn't have another hardware channel to support a second serial communication. The main serial port of the board is digital pins 0 and 1, Rx and Tx respectively. To overcome this problem, NewSoftSerial library is used to create a serial port on any of the other digital pins using only software. It creates a serial object with the given pins. Sending data over XBee is very straightforward, if the transceiver is programmed to send to a specific device (another XBee module) , it is enough to send the data over serial without doing anything else, and it will wirelessly submit that data. With the help of the library it was possible to send the data over XBee, to be captured by another transceiver to track the data over time.



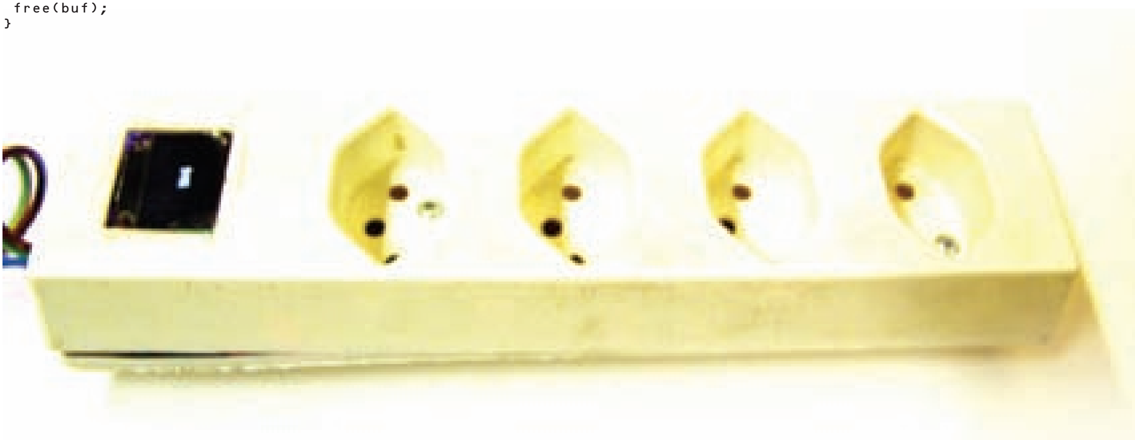
Powering is done by adding a AC DC transformer, which is taken from a small wall adapter. The AC end is soldered into the two ends inside the power strip, DC output is directly fed into the RAW and GND pins of Arduino. The microcontroller takes the unregulated voltage from RAW and supplies other components with regulated voltage from the VCC pin.





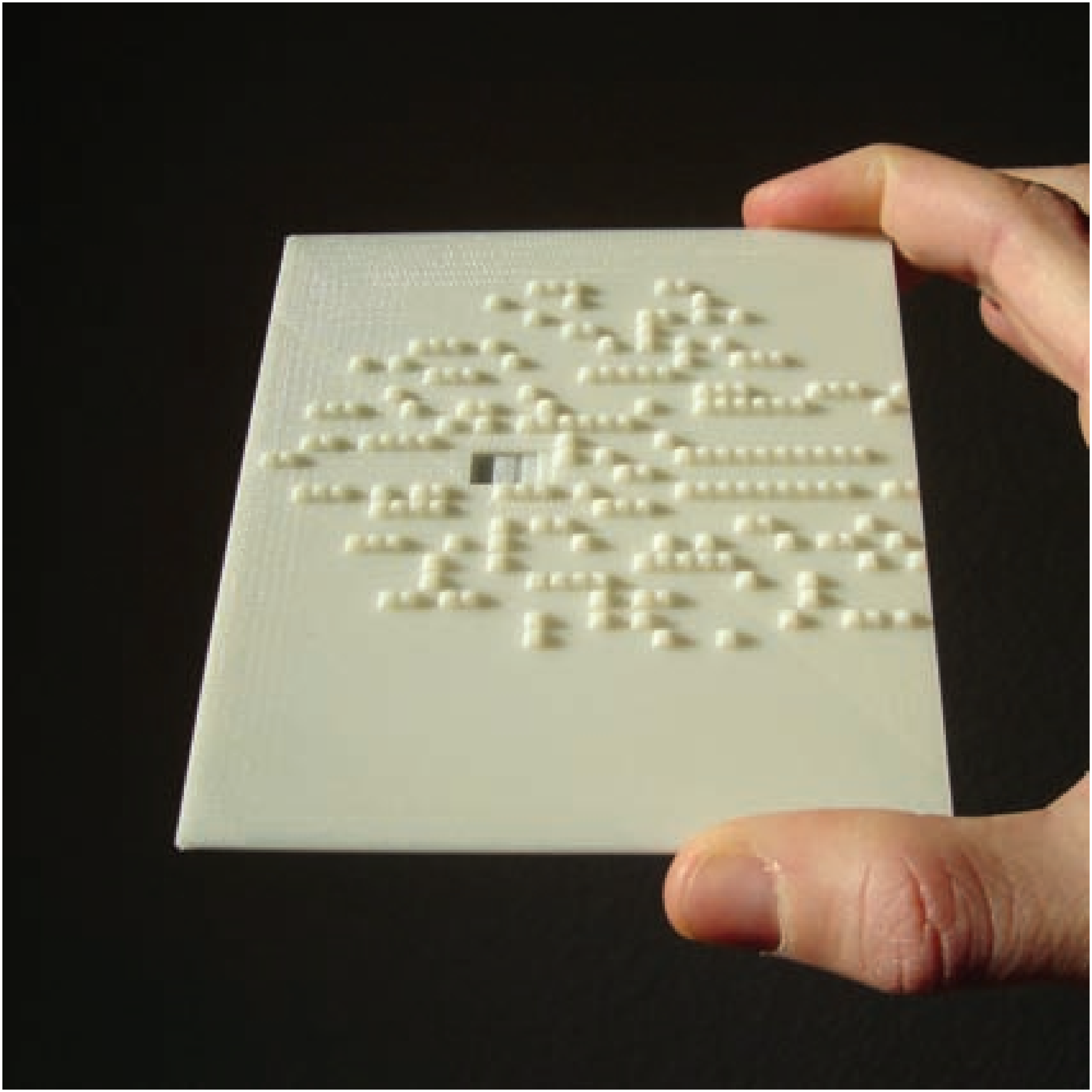


```
#include <stdlib.h> // for itoa() call
#include <NewSoftSerial.h> //for XBee serial
#define readPin 2  //analogread pin for ACS712

float k = 0.06; // factor for converting to current
float w = 0;

char buf[5];
int val =0;

NewSoftSerial xSerial(11,10);

  void setup()
{
  pinMode(2,INPUT);
  Serial.begin(38400);  //OLED
  xSerial.begin(4800);  //xBee

  OLED_Init();
  OLED_Clear();
}

void loop(){
 OLED_Clear();

 val = abs(analogRead(readPin));
 w= (val*k)*220;
 itoa(val,buf,10);  //convert int to string
 buf[strlen(buf)]='W';
 OLED_DrawText(5,2,2,buf,31,31,31);
 xSerial.print(char(w));
 delay(1000);
 free(buf);
}
```



CONCLUSION

Although it can still be developed further, as it is just a first experiment in measuring current, the project shows a great potential in homemade hacks and custom made visualization tools. The next step in the development should be using the wireless data in a wireless network and collect data from multiple XBee nodes. Then implementing a visualization interface on the computer screen or on the OLED displays of single nodes.



KINETOGRAPHIES
AGENT-BASED SELF ORGANISING SETTLEMENT
THESIS BY APHRODITE STAVROPOULOU

The Thesis seeks to an agent-based system that can be used for simulating campers' activity and behavior and to a model that simulates a dynamic process such as self-organize refugee settlement.

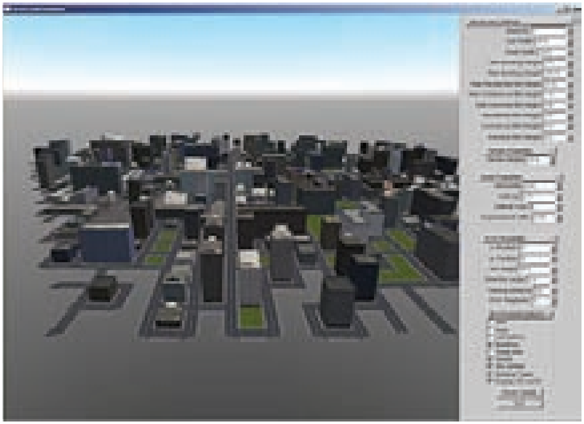
This system is based on cellular automata and agent technology. Agents represent (objects or) people with their own behavior, moving in a simulated space, based on the cellular automata grid. Each iteration of the simulation is based on a parallel update of the agents conforming local rules. Agents have sensors to perceive their local neighborhood and affect their environment. In this manner, autonomous individuals and the interaction between them can be simulated by the system.

The approach will lead to a system that may serve as a tool in the design process to better understand how a design will influence user behavior. Therefore, this method could provide a tool that controls the templates of UNHCR for refugee camps.

DEFINING URBAN ELEMENTS

There is an extended literature concerning urban growth and the formation of cities. In an attempt to find a substantial and specialised vocabulary to describe common urban elements and patterns, the origins of their formation, and their evolution through time, one may find different and contradicting doctrines. Christopher Alexander in his work “A Pattern Language” lists 253 patterns which constitute a “language for building and planning” (Alexander, 1977, p.ix). This approach is echoed in the movement of the New Urbanism, in its declaration of principles for interpreting and designing cities (For more information: www.newurbanism.org). Yet, the socio-spatial expression of the humans’ fundamental need for symbiosis is, throughout history so complex in nature and context that there are “obvious objections to the idea that urban forms evolve according to general laws” (Hillier, 1996, p.262). Moreover, Hanson particularizes by saying that “descriptive typologies are generally speaking too simple to be useful […] or to detailed as to be idiosyncratic” (Hanson, 1989, p.81). However, “determinants” which influence the development of the early rural and later urban shelters do exist, and may be identified in the special topographical and climatological features of the land, the resources on construction materials and the hitherto technological advances in constructional methods, the need for mobility, and economic, political, religious and defensive factors (A.E.J. Morris, 1972). Although the analytical method of Morris provides a great variety of urban form determinants, in an urban simulation one may naturally expect the reduction of the encoded elements that will be included, both in the procedure of the realization of the model and the end product itself. This holds merit, regardless of the fact that in a lower level of analysis, it is the complex, yet fundamental, socio-spatial nature of our existence and our environment which also determines and shapes our cities (Hillier and Hanson, 1984). The reproduction of reality in its whole is both unfeasible and impractical. Therefore, in the simulation of a city there is a certain degree of abstraction in the way elements are represented and associated.

Lynch identified five distinctive elements that compose the city: **paths** (linear movement channels such as streets, trails, and sidewalks), **edges** (linear elements such as walls and shores), **districts** (identifiable city sectors with common features), **nodes** (strategic points of concentration) and **landmarks** (physical objects used for reference) (Lynch, 1960). Thus, in the Virtual Urbanity program, the city is reduced to an elementary street network and core elements of buildings. The differentiation of the generated results is to be found on these basic subsystems of cities, and their classification is done accordingly.



http://www.infinitylab.com.au/research/prototypes.htm
Real-time procedural generation of `pseudo infinite' cities, (Greuter et al., 2003)

RELATED WORK

There is a great number of interrelated research projects in the extended field of computer aided urban simulations. These either focus on simulating 2D urban growth on a land planning level, or concentrate in modelling 3D virtual cities. Most of the latter projects aim at the construction of virtual equivalents of existing real environments. It is obvious that the manual design of these large scale environments would require a huge amount of time and effort, making the whole venture practically impossible. Hence, these projects use automated or semi-automated procedural methods that depend on the utilisation of digital databases. Specifically, the most common procedures include the use of 2D or 3D GIS data (Coelho et al., 2005), elevation and building footprint data extracted from 2D plans (Laycock and Day, 2003), and photogrammetric or laser-scanner (LIDAR) methods on aerial photographs (Brenner, Haala and Fritsch, 2000; Zhang et al., 2003, Deng et al., 2005). However, these projects are not considered purely generative since they don't generate new spatial relations, but rather reproduce existing ones, by rendering 2D data into 3D forms. There are projects that advance towards a more generative approach and produce completely fictitious environments from scratch. Greuter, in (Greuter et al., 2003), proposes a system that bases its function on the combination of preset floor plans and real time interactive generation. This system is using the user's behavior to inform an "on demand" bottom-up generative process, and does not focus on the creation of an environment infused with actual urban properties, but rather reproduces a simple pattern.

Lastly there are projects which are able to reconstruct the patterns of existing cities, or alternatively generate completely new ones (Parish and Muller, 2001; Sun et al., 2002; Gonzaga da Silveira and Musse, 2006). These projects have a better global understanding of urban dynamics and employ global rules in order to simulate them to a higher or lower degree. The first two concentrate on the creation of a hierarchical street network, while Parish and Muller's City Engine further deepens incorporating advanced methods for dividing blocks into lots, assembling volumes for the creation of the buildings, and classifying by assigning the type, the maximum height and the architectural style of them. The last project has a more dynamic approach and proposes a generic framework for the initial generation of urban structures and the continuous update of their status by relating the generative process with the influential semantic forces of the activities of interactive virtual avatars.

From the urban growth point of view, most studies that research city development, utilise Cellular Automata generative methods, multi-agent systems or the combination of both. Such a studies engage in an in depth analysis of pre-manufactured models of cities (usually in 2D) and report on their potential development by examining local and global statistical data, planning regulations and geomorphic particularities.

RESOURCE ABOUT REFUGEES CAMPS AND UNHCR

Herz Manual in his work “Refugee Camps - or - Ideal-Cities in Dust and Dirt” analyzes “UNHCR’s strategies in setting up and planning spatial structures for refugees, how these ‘urban’ strategies – based upon European models – lead to permanence in a situation that by definition should be temporary”.

In this work there is the issue, if the specific local knowledge of the region, the social and political aspects and the characteristics of the refugees are taken into consideration during the planning strategies for refugee camps.

“Refugees are meant to plant vegetable gardens to achieve self-sufficiency...Many of the refugees come from villages of the northern regions of CAR and have previously practiced a craft or ran small shops. Other refugees are nomads of the tribe ‘Buel’ and were raising large cattle herds. They don’t like and don’t eat vegetables, but are now made to grow vegetables. Through a specific act of planning and a simple design move, those village societies and nomads are being homogenized and turned into rural societies”.

Refugee Camp ‘Gondje’, extended camp allotments, photo by Herz Manual

Original plan of Refugee Camp ‘Amboko’, plan by UNHCR

SCENARIO

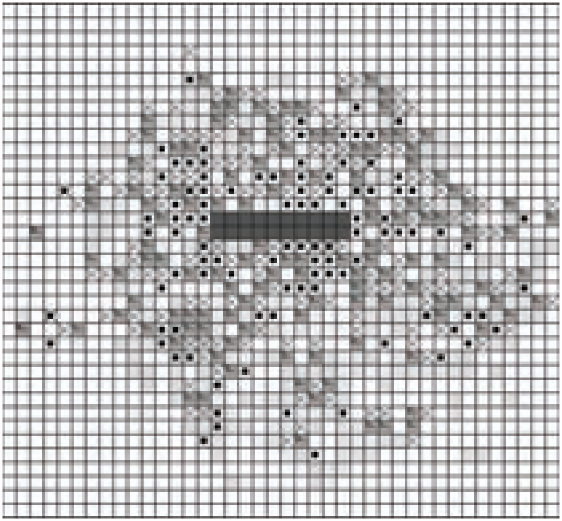
In the specific application, agents represent refugees, moving over a network of the settlement.
Agents can have particular targets, a series of stops, a route of shortest duration or the most attractive route.

Step1: search for position of your tent
walk and check around for an empty place
if you find a position go to step 2.

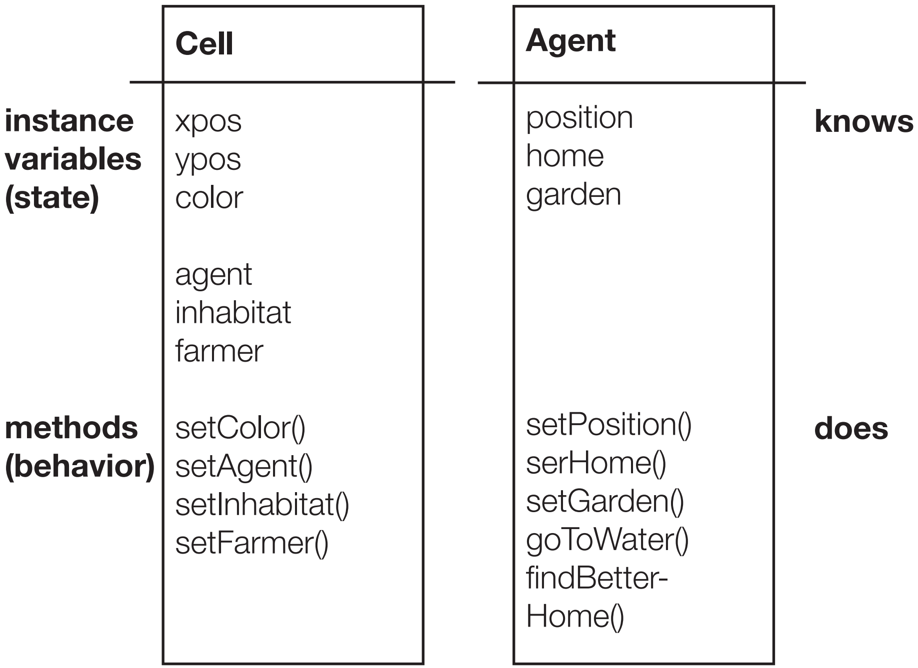
Step2: search the path to supply point
check for better home place
do you find a better home or not , return boolean , true false

If you find a better position
Then go to step 3 Else do Step1

Step3: delete your home and built a new one

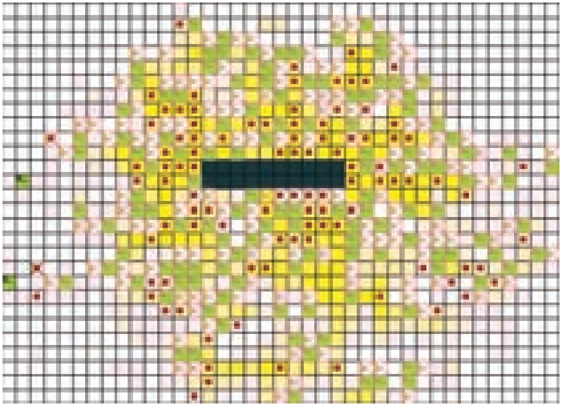


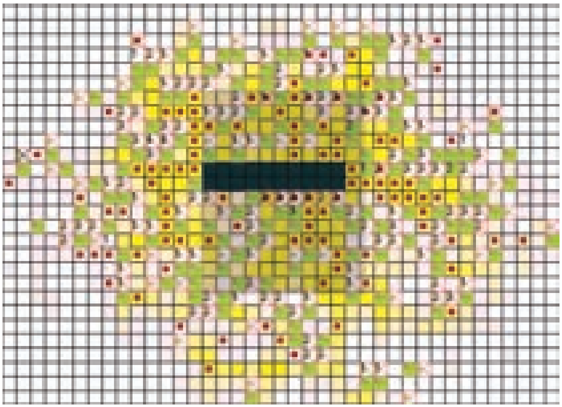
Cellular automata grid

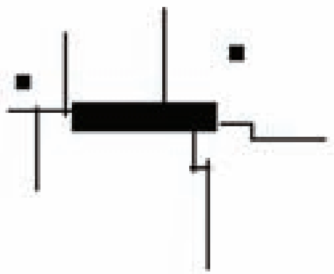


Based on cellular automata and agent technology:

Agents represent objects or people with their own behavior, moving over a network. Each agent is located in a simulated space, based on the cellular automata grid. Each iteration of the simulation is based on a parallel update of the agents conforming local rules. Agents positioned within an environment have sensors to perceive their local neighborhood and affect their environment. In this manner, autonomous individuals and the interaction between them can be simulated by the system.

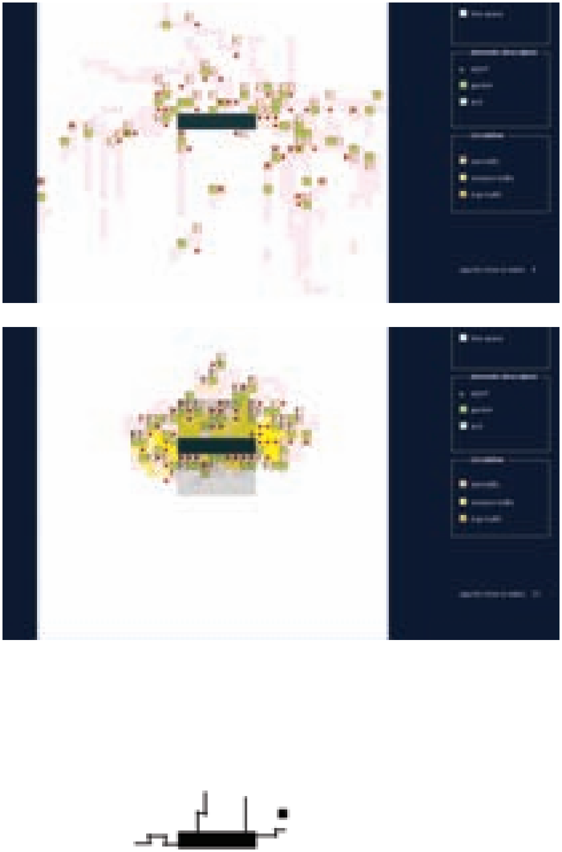


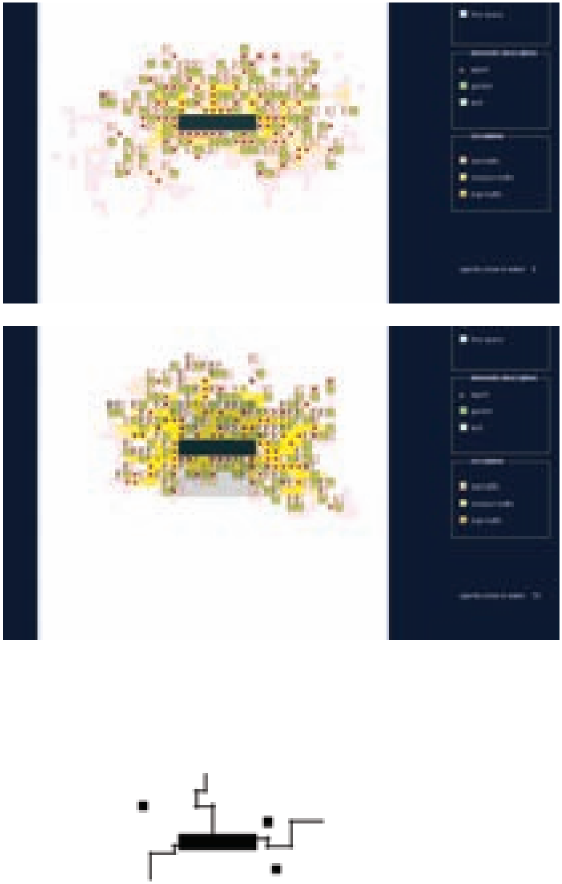




points with density, new centers
main roads

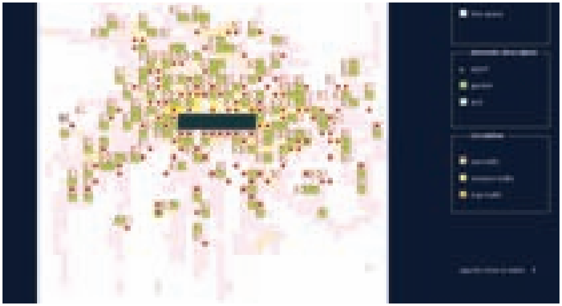
Cases that refugees choose the routes of shortest duration to the water

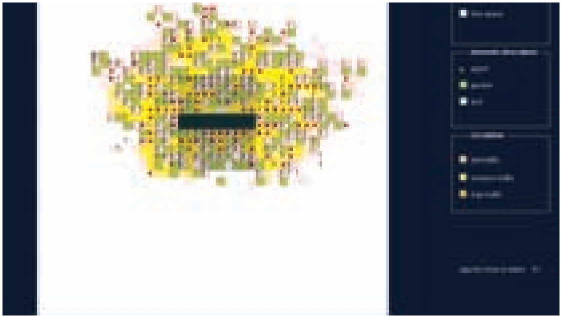


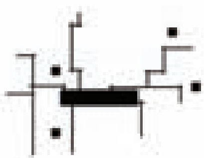


50 Agents

100 Agents

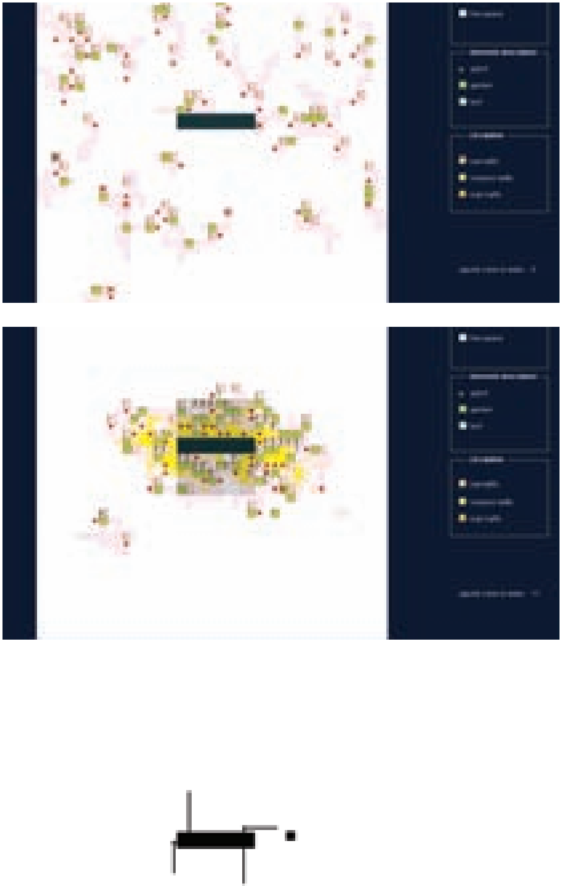




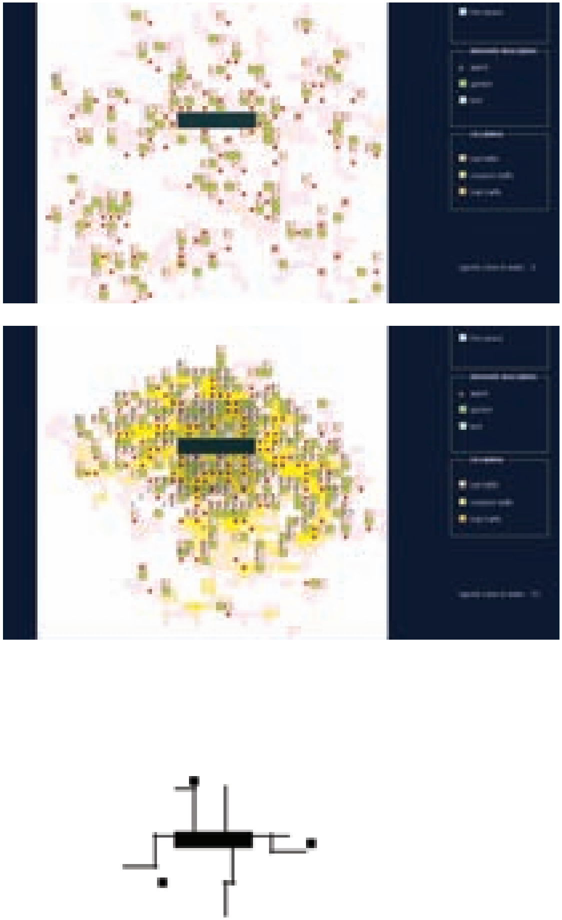


150 Agents

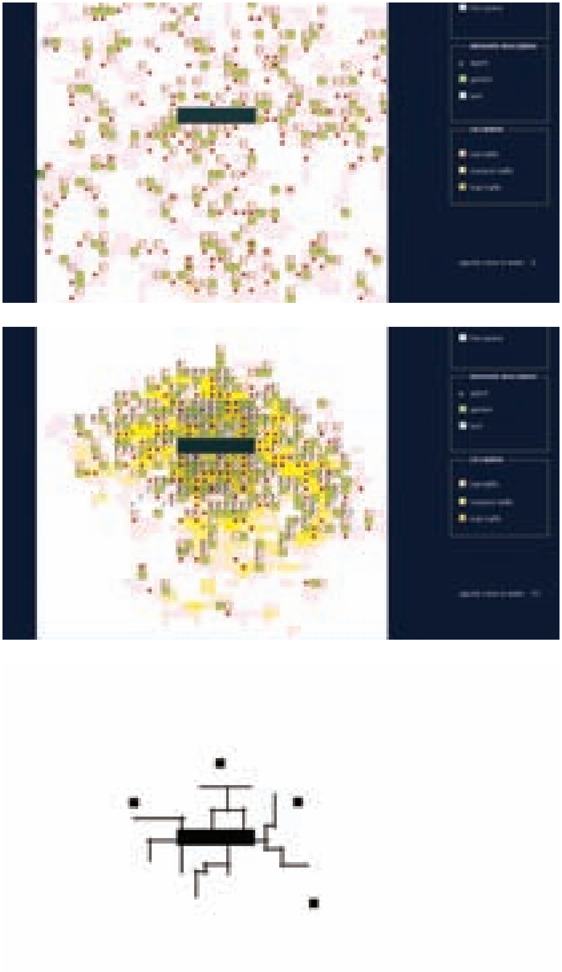
Cases that refugees do not choose the routes of shortest duration to the water



50 Agents



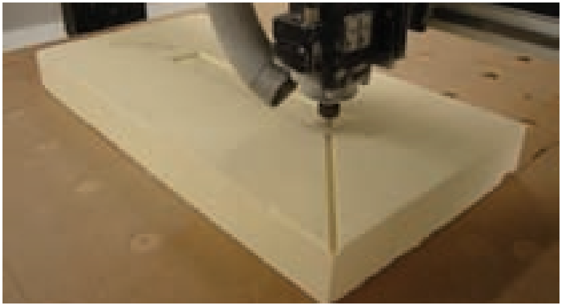
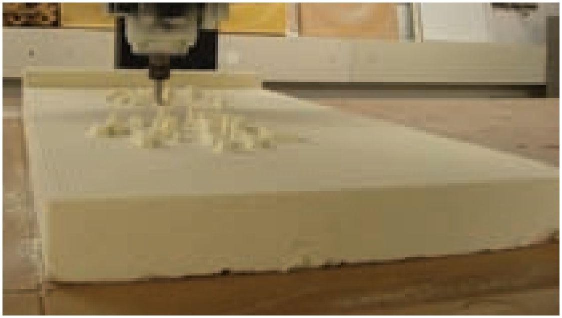
100 Agents

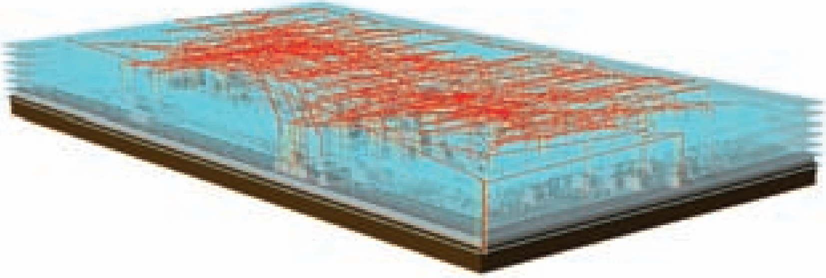


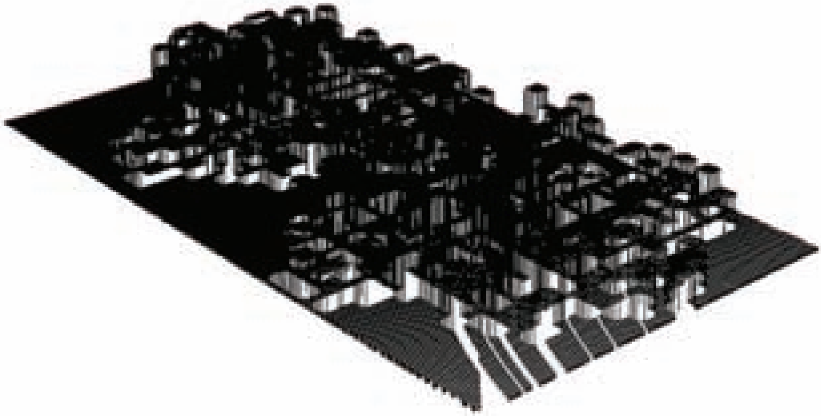
150 Agents

Combining several tools
(Eclipse, Rhino, Rhino grasshopper,
Rhino CAM) a final model presents a urbanistic figure.

The rules are based on vicinity and neighborhoods as well as on the distance from the attractor points.





CONCLUSION

The project shows an approach based in autonomous individuals and the interaction to the environment, in order to come close to a model that simulates a dynamic process such as a self-organize refugee settlement. The dialogue between macro-level of an urban form and the micro-level of each individual trying to achieve their goal was a question of this resource from the beginning.
The vision is a method which could provide a tool that controls the templates of UNHCR for refugee camps based on the characteristics of the user.

Mentor: Benjamin Dillenburger

Keywords: urban growth, parametric, generative, procedural, simulation, navigation, way-finding, legibility, hierarchy, patterns

Alexander, C., Ishikawa, S. and Silverstein, M. (1977) *A Pattern Language: Towns, Buildings, Construction.* Oxford University Press

Brenner, C., Haala, N. and Fritsch, D. (2000) *Towards Fully Automatic Generation of City Models.* In: IAPRS Vol. XXXIII, Part B3/1, Comm. III, ISPRS Congress, Amsterdam. 2000, pp. 85-92

Charitos D. (1997) *Designing Space in Virtual Environments for Aiding Wayfinding Behavior.* Department of Architecture, University of Strathclyde. The 4th UK VR-SIG Conference, November 1, Brunel University

Coelho, A.F. , Augusto de Sousa, A., Ferreira,F.N. (2005) *Modelling urban scenes for LBMS, Proceedings of the 10th international conference on 3D Web technology,* March 29-April 01, 2005, Bangor, United Kingdom

Herz Manual, (2009) *Refugee Camps - or - Ideal-Cities in Dust and Dirt.*

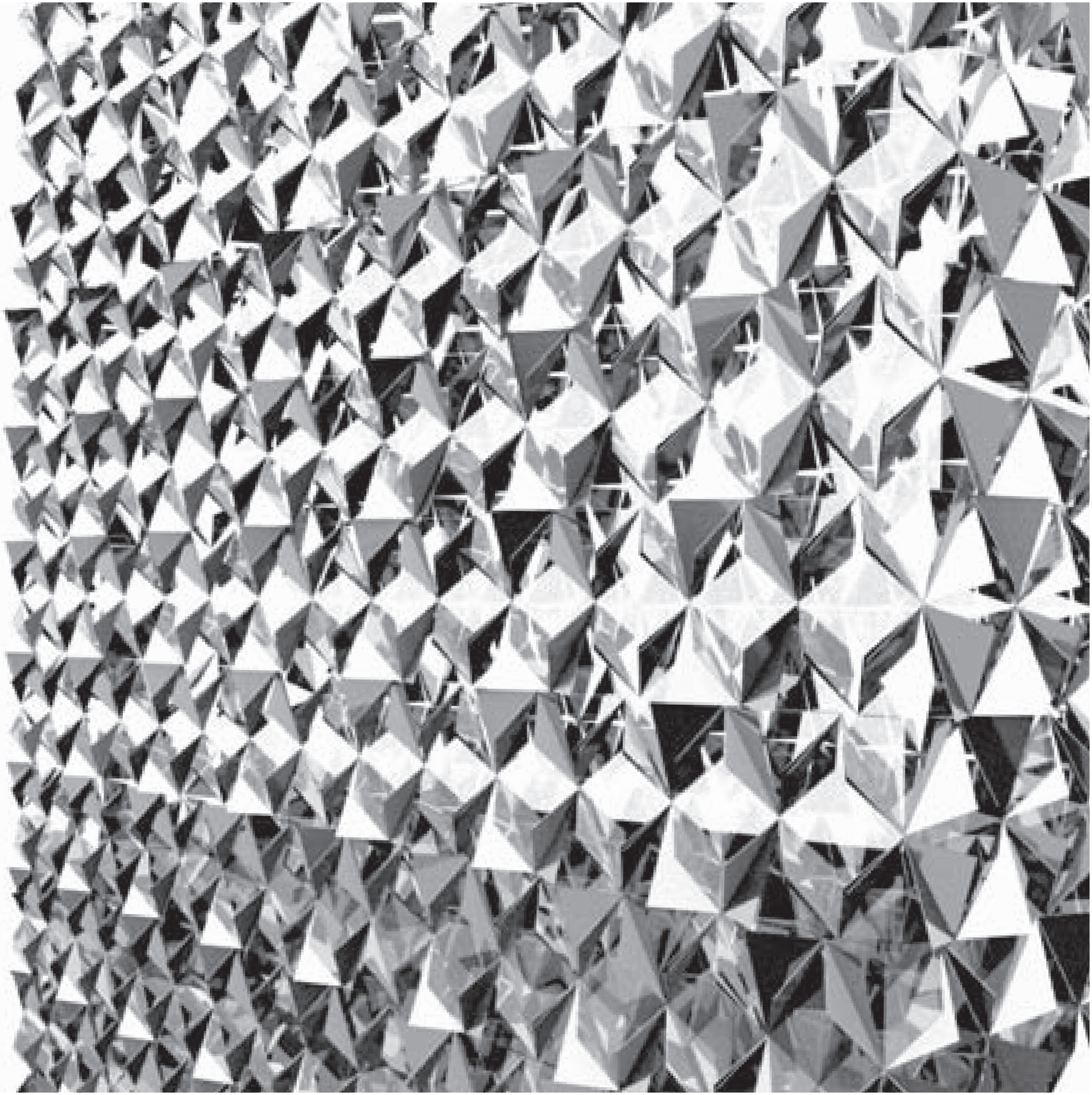
Hillier, B. (1996) *Space is the machine.* Cambridge University Press, Cambridge

Hillier, B. & Hanson, J. (1984) *The Social Logic Of Space,* Cambridge University Press, Cambridge 2005

Lynch, K. (1960) *The Image of the City*, M.I.T. Press, Cambridge, MA, USA

Morris, A.E.J. (1976) *History of Urban Form. Before the Industrial Revolutions.* Third edition, 1994, Longman Scientific & Technical, Essex, United Kingdom

Zhang, Z., Wu, J., Zhang, Y., Zhang, Y., and Zhang, J. (2003) *Multi-view 3D city model generation with image sequences,* Proceedings of the international workshop on Vision Techniques for Digital Architectural and Archaeological Archives, July 1-3, 2003, Portonovo – Ancona , Italy



DIGITAL FAÇADES
INSTRUCTIONS FOR DESIGN PARAMETRIC FAÇADES
THESIS BY JASMIN ZARALI

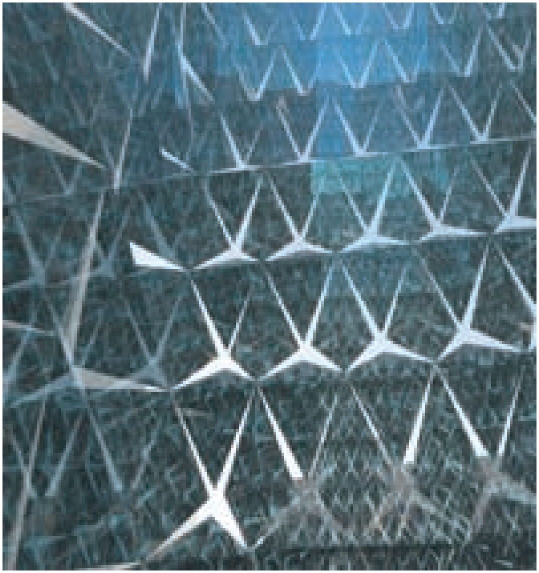
FOREWORD

Mentor: Steffen Lemmerzahl

In the postgraduate program of Master of Advanced Studies in CAAD we examined current information technologies to expand our comprehension of architecture. The usage of digital tools sensibilizes us to seek for new fields in architecture where they can be implemented.
Current computer-based programming provides a range of tools for the efficient design, analysis, and manufacturing of complex shapes. This opens up new horizons for architecture but also challenges the architect to comprehend this developement and to integrate it into work. Nowadays this developement is not yet accepted by the majority of architects as methods to design architecture. Consequently the new possibilities with their benefits are refused.

Around these „problems“ my research project wants to give architects an understanding of digital tools who are not familiar with it by showing the advantages in a certain field of architecture.
The design of a façade is a subject with what an architect is always confronted with. It is a very important and complex aspect in architecture as it expresses and sets the character of the building. The façade should be able to react and communicate individually according to it’s environment and the function of the building. This essay will demonstrate that the scope of creativity can be expanded by these new technologies.

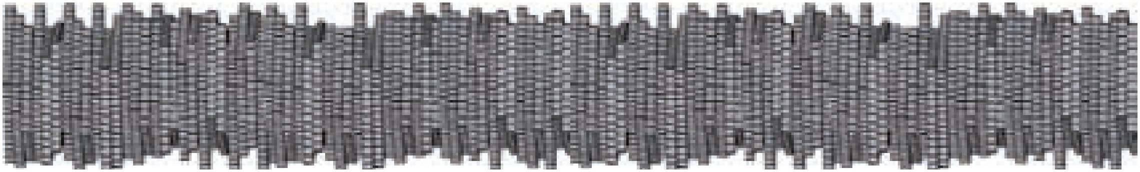
Many research projects already showed that parametrically controlled façades can be handled in a very practical way. It allows the user to define interior and exterior parameters for every geometrical form. This technique is ideal for producing results and variations with the computer in a fast way and further to transfer them into reality by making models, which cannot be realised by conventional methods.

TOOLS

CATIA (Computer Aided Three-dimensional Interactive Application) is a multi-platform CAD/CAM/CAE commercial software suite.
The software was created in the late 1970s and early 1980s to develop Dassault‘s Mirage fighter jet, then was adopted in the aerospace, automotive, shipbuilding, and other industries.

DIGITAL PROJECT is a Computer-aided Design (CAD) software application based on CATIA V5 and developed by Gehry Technologies, a technology company owned by the architect Frank Gehry. Among the changes made by Gehry Technologies to CATIA is a new visual interface suitable for architecture work.
In contrast to some CAD packages used for architecture, engineering and construction, Digital Project (like CATIA) enables information to be sent directly to manufacture [citation needed], rather than needed to be processed separately in preparation for sending out of house.



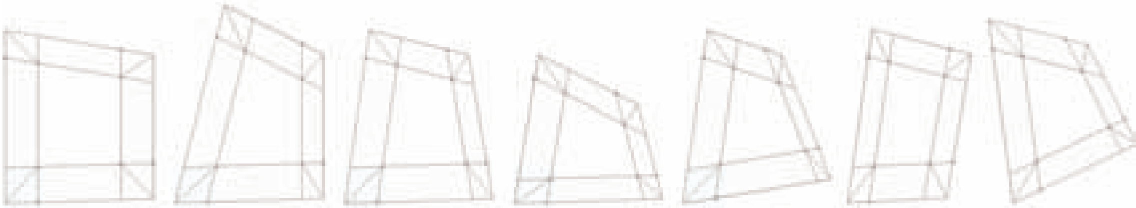
INTRODUCTION

Digital Façades - Instructions for design parametric façades will explore different techniques to develope parametric façades for architects without special programming skills. The digital methods should be considered as further tools to generate architecture.

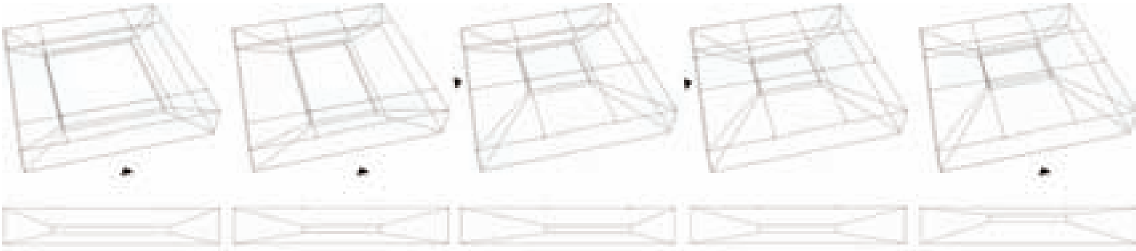
The new technologies offer us the possibility to experiment with new complex creations and new forms of aethetics. The traditional way of a serial process in planning and building a house can be interpreted and organized in a new way. The handling with digital tools to generate architecture help us to react quick to changing requirements and to work paralell in different fields. But the majority of the architectural offices don't have the knowledge to work with these methods. And the offices who are able to implement the new technology in their projects are dependend to their special trained people.

This thesis wants to give an access to the usage of digital tools without learning complicated tedious processes. Therefore the different case studies still deal with traditional ways of designing architecture (making sketches- define constraints,..) by developing and demonstrating them in a new way. These techniques for making façades help us to operate with complex forms and to constitute in a fast way results and alternatives. The definition of parameters prevents the concept to become random. The case studies examine different ways of handling the new tools. The façade elements in these studies are all made with Digital Project/CATIA. I chose this program because it affords the architect to work with a visual interface. A three dimensional model of the façade element is drawn parametrically with this program. Finally the designer can deform every construction element, and this allowable movement can be later on linked to a parameter.
The complementary programs are RhinoScript and Java Eclipse and for demontstration Rhino 3D and MX Studio.

The first case study, developed by my mentor S.Lemmerzahl gives already a good introduction in parametric façades through benefit of user interface and bitmaps- but at least a time consuming process.
The second case study can work with less steps and programs but still has too many limits in designing the form and surface of the bulding.
The last case study shows an easy and clear principle for creating parametric façades with reduced constraints and programs. The embedded triangulation system offers the oppurtunity to design freeformed surfaces.

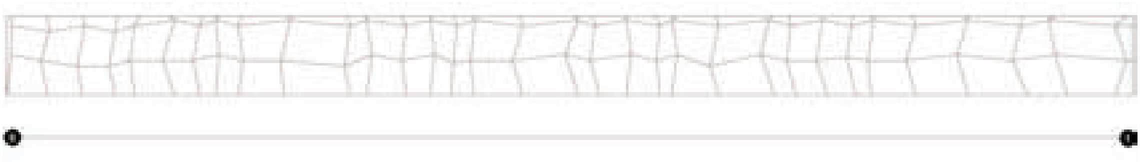


p2y p2y,px2 p2y,px2,px3 p2y,px2,px3,py3 p2y-py4 p2y-px4 px1-px4



param max 1111 param width 1101 param frame below 1001 param frame above 0001 param depth 0000

Digital Project/CATIA: façade elements



unfolded façade

CASE STUDY 1: BITMAP DRIVEN PARAMETRIC FAÇADES
TOOLS: DIGITAL PROJECT/CATIA, RHINO 3D, RHINOSCRIPT, PERL, PHOTOSHOP

Design

The project of an residential housing is situated in Zurich and had been an entry for a competition. The building consists of 200 apartments, a shop and a coffee shop.

The aim was to develope a new façade with Digital Project/CATIA that accommodates to it‘s functional and environmental requirements while considering the urban planning and the layout of the apartments. The final form of the elements is controlled by several parameters like sun position, noise emission and the functional program of the building.

According to these parameters the openings varies in size and depth. The façade becomes a translator between inside and outside.

Technique

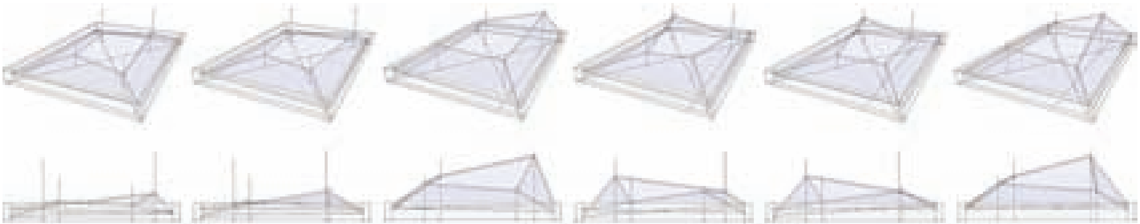
A script in Rhino unfolds the 3D geometry of the building for creating the façade- areas in 2D. The grid of the façade is drawn manually in Rhino3D (*unfolded façade*). With the unfolding procedure the proportions for the bitmaps through which the parameters will be enciphered are generated.

The bitmaps are the tools for steering the parameters. They are created in grayscales (black is the max. for a parameter and white the min.) with a graphic program. Another Rhinoscript makes a XML-File which contains all the coordinates of the façade grid. This file will be read in by a script written in PERL, that orders brightness values according to the coordinates from the region of the bitmap - as parameters to the façade elements dependent on their position on the façade (*Digital Project/CATIA: façade elements*).

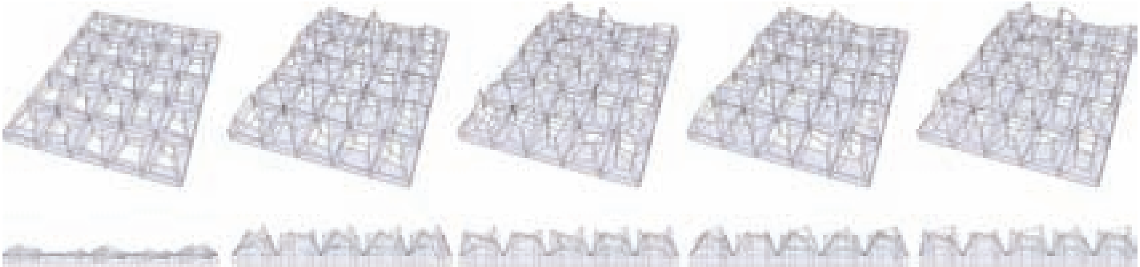
Result

This technique already allows architects to create parametric façade elements without having special skills in programming. This program gives a freedom in designing openings in every size and position.

But this process takes time and needs a lot of steps and programs. According to changes (f.e. the shape of the bulding or the facade grid) a new unfolding of the volume and new bitmaps are required. So I was looking for more simplified systems to develope parametric façades which can react more flexible to changes but still are easy to use.

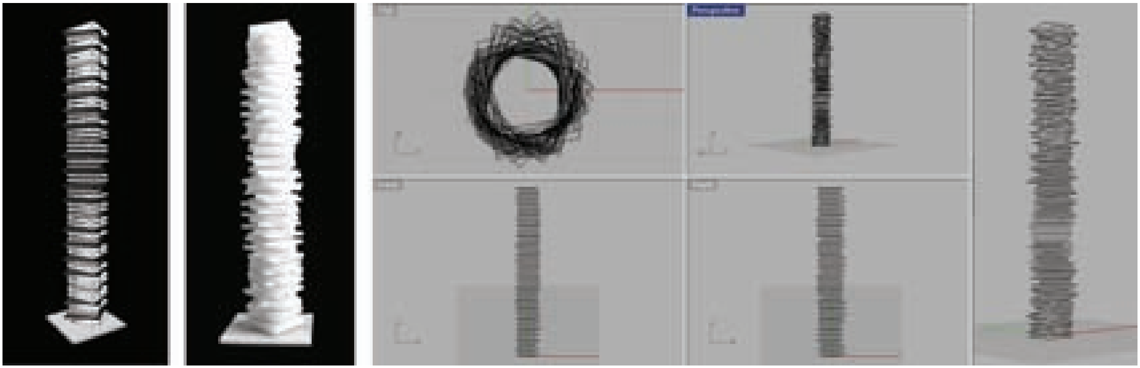


param 000 param 001 param 110 param 010 param 011 param 111

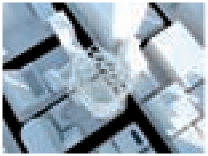


param 000 param 010 param 110 param 011 param 111

Digital Project/CATIA: façade elements



developement: screenshot of Processing, 3D print, screenshots of Rhino3D: output of Rhinoscript



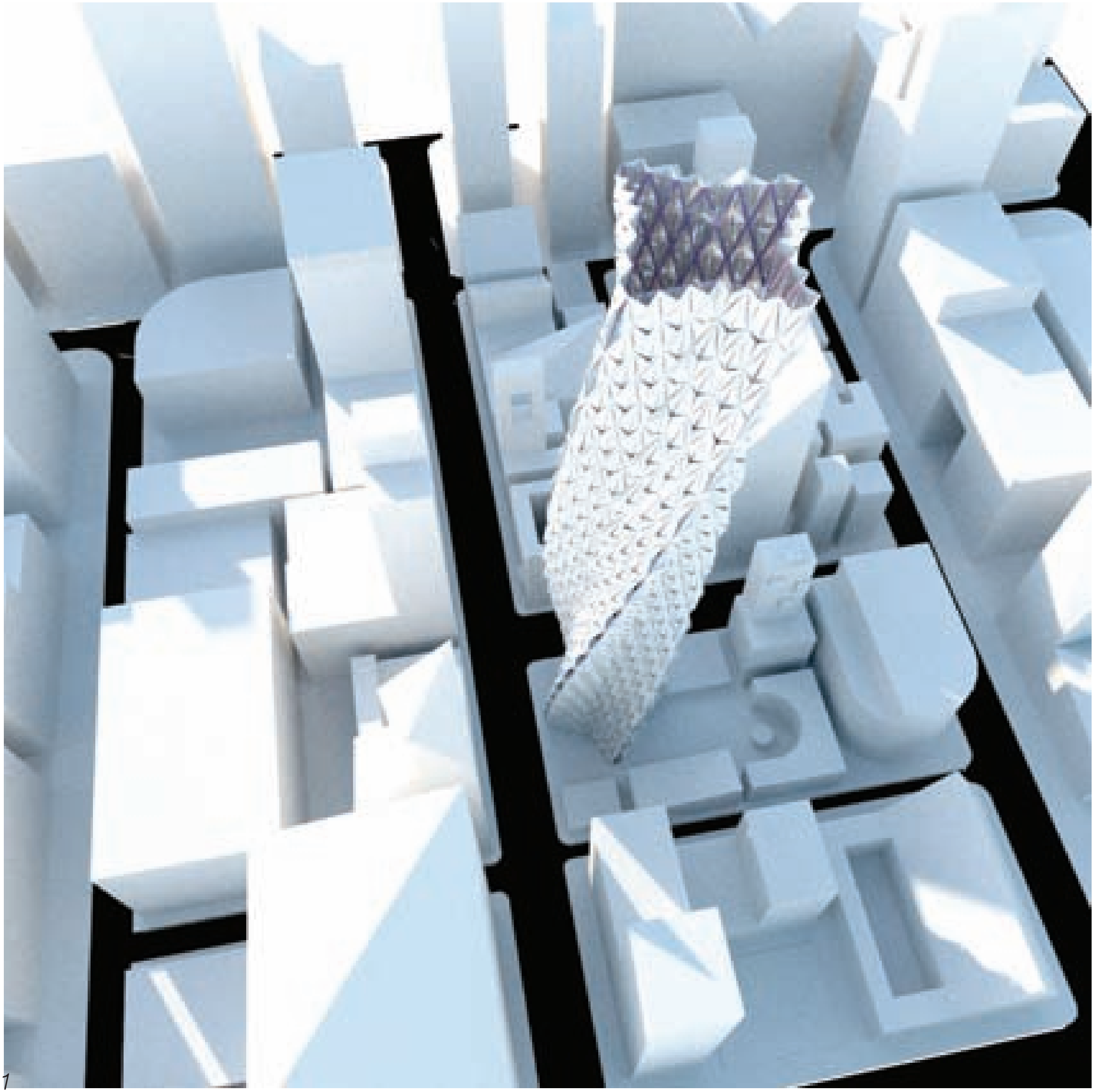


CASE STUDY 2: JAVA STEERED PARAMETRIC FAÇADES
TOOLS: DIGITAL PROJECT/CATIA, RHINOSCRIPT, JAVA

Design
This study focused on another technique and program interfaces to develope parametric façades and to test the design possibilities with Digital Project.
This building has no real basic conditions. The context and the function is free chosen. The skyscraper consists of different rectangular cubes that varies in size and position. Each façade element is based on 5x5 shaped bodies (*Digital Project/CATIA: façade elements*). The façade appears agile through the geometry of the surfaces and their interaction with shadow and light. The three-dimensionality of the elements participates in the creation of the form of the buidling. These parts are controlled by different parameters like orientation, height and opening degree by random.

Technique
The skyscraper was basically programmed with processing. The geometrical assembly was not useful for Digital Project because every single cube consists of triangles, so the script was rewritten with Rhino.
This one creates polygons as the groundfloors for each cube and generates the XML-file with the coordinates. The number of polygons, their rotations and sizes are specified through user-input. Another script made with Java defines the grid for the façade and the parameters. Finally the XML-file and the elements are read in the Javascript for modeling the façade. The skyscraper exists of 240 floors with a floorheight of 3m and a maximum façade length of 25m. The 4499 elements have an average width of 1.75m (*developement*).

Result
As a cooperating program with Digital Project/CATIA I would suggest Rhino. This program fulfills the specific requirements the best. It contains a good scripting language but still works as a common CAD program which is easier to deal with.
This tested method here is faster to use and reacts better to changes according to the façade grid. But still the usage of two different scripting languages keeps it very complex. Also the construction of the elements for the skyscraper had problems. The previous design with the 5x5 bodies was divided into too small sections which requiered an immense power of the processor. The final design is a simplified version. Furthermore the formal design is limited by the straight rectangular forms and the design possibilties of the elements.



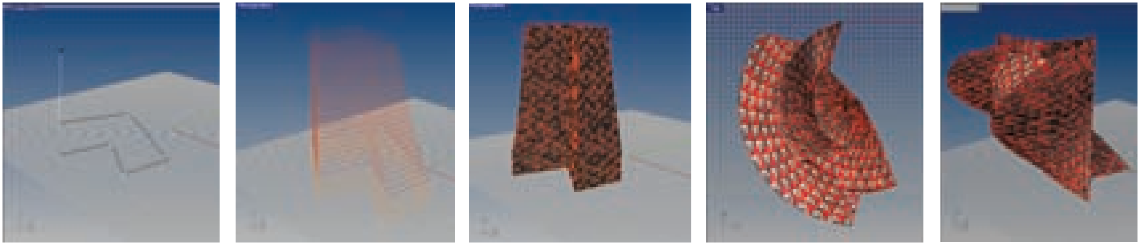
mod.01

CASE STUDY 3: RHINOSCRIPT STEERED PARAMETRIC FACADE
TOOLS: DIGITAL PROJECT/CATIA, RHINOSCRIPT

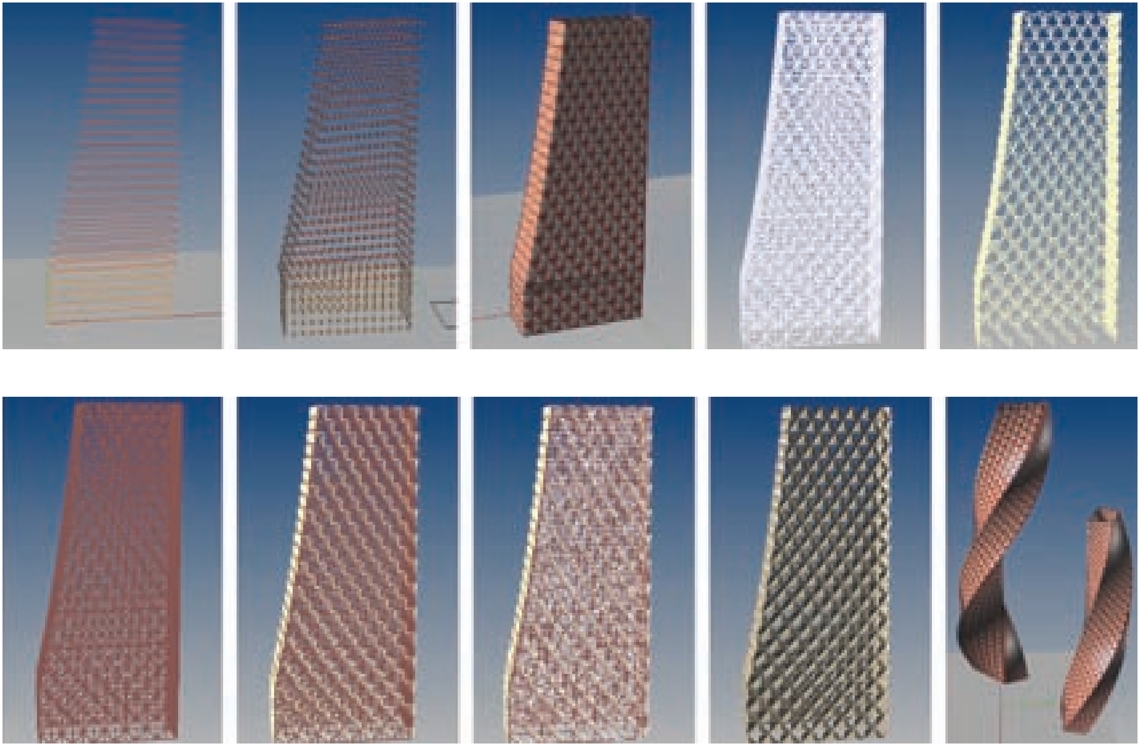
The design limitation of the case study 2 motivates me to experiment with other geometrical forms. The next program for designing parametric façades should not only simplify the process respectively the technique, it should also minimize the constraints. More freedom in creating drawings and transfering them into the computer should be given, so that the architect can act more creative.
The key to generate more freeform buldings was to use the method of triangulation (an old technique to deal with complex geometry)- this also gave me the design principle for the new façade element.

Design
The aim was to create a façade for the new Warteck Tower at the fairground in Basel- a competition that had been won by Morger+Dettli architects. The situation is characterized by large scaled buildings like the fairtower of Morger&Degelo and the new planned bulding for the fair of Herzog & de Meuron.
The new tower consists of 27 floors, which reduces in size upwards. The main functions are for office-, gastronomy- and residential usage. The gastronomy is located in the groundfloor followed by fifteen floors of office use. The last floors are reserved for the apartments.
The idea of segmentation of the skin respectively of an ornamentic façade gives the tower a human scale for the urban space in contrast to the large dimension of other buildings. Consequently it can react better to it's internal conditions like the small scaled residential use. The elements are controlled by parameters like: height-function and orientation.

The elements for the façade are divided up in triangulated forms so that they can match on the form of the building. Each element consists of two layers. The first layer is the window with glass and frame. The second layer is the panel for sunprotection that can be fold away. Through the development of the technique a lot of possibilities in generating different forms and façade elements in other environments had been tested to improve the script until it leads to the real project of the tower in Basel.

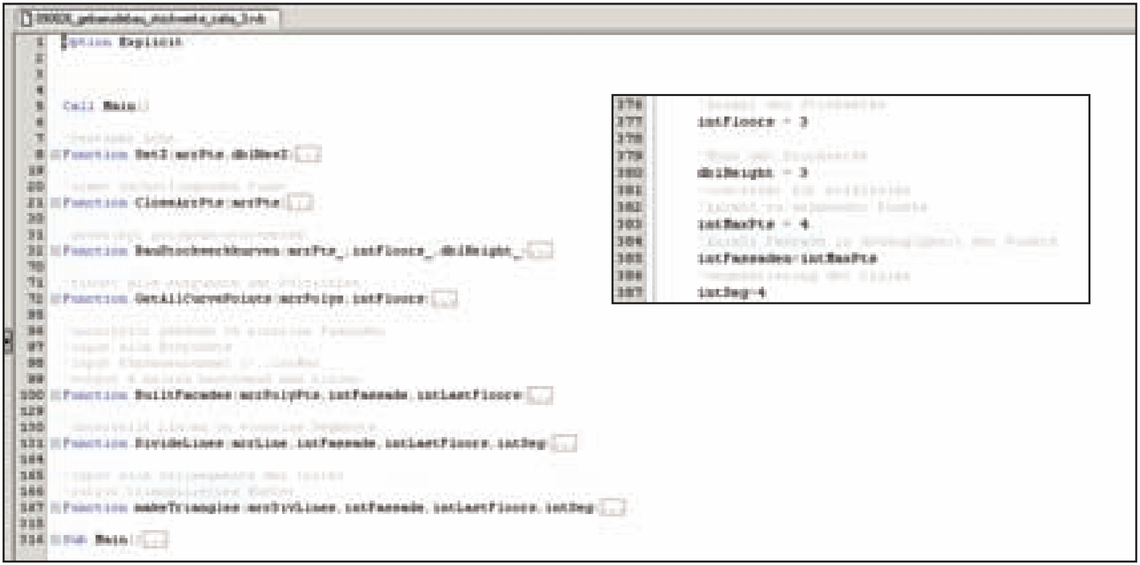


1 Episode of Rhinosscript

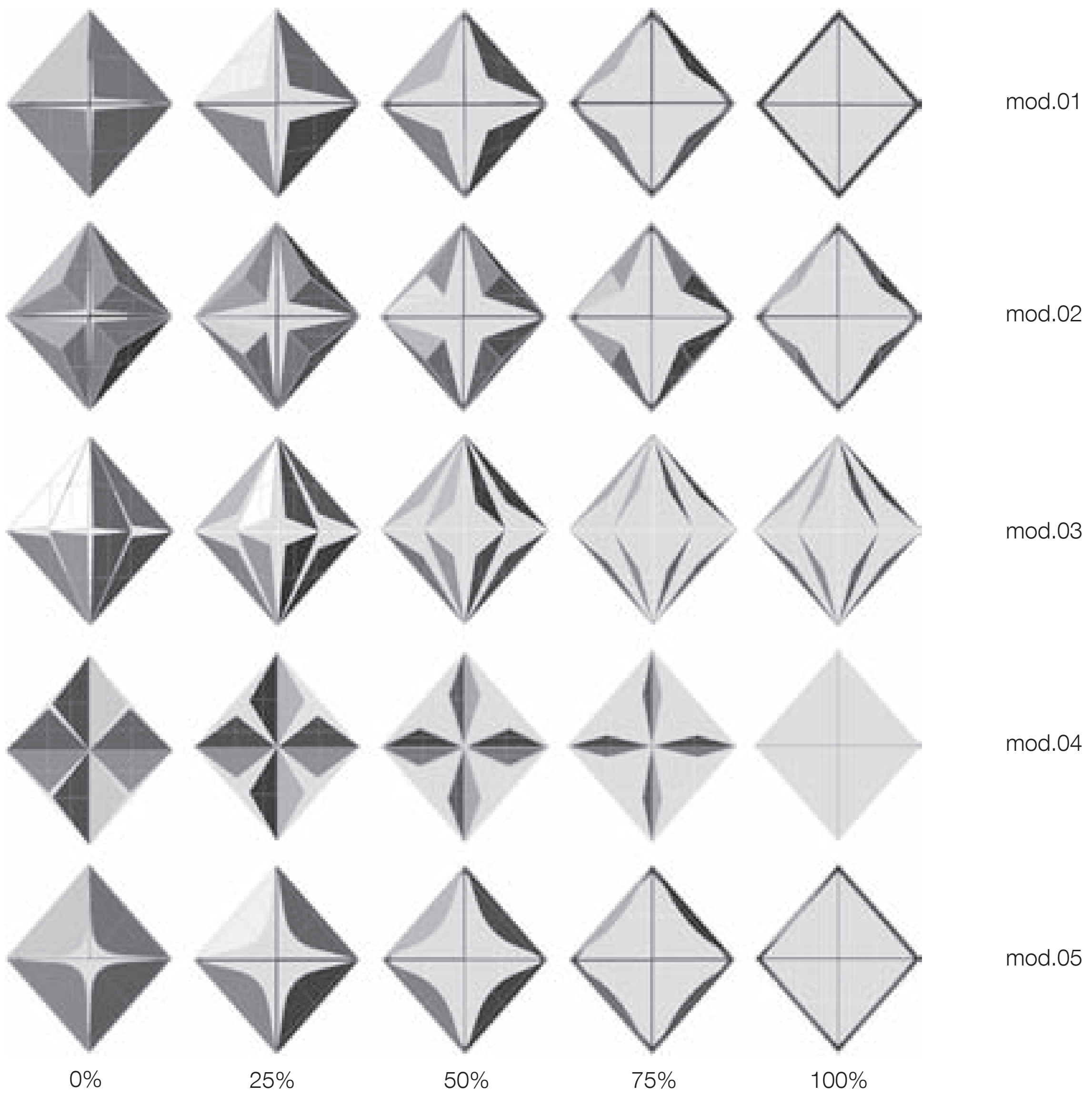


2 Episode of Rhinoscript: three pictures first row: output of Rhinoscript, next pictures: building after import from Digital Project with the façade elements mod.05, last picture: possible freeforms

Technique
The new script was completely made in Rhino, which simplifies the procedure of generating the façade.
The first function „SetZ“ defines the height of the floors. The second function „CloseArrPts„ defines the edge points and takes the closest point to draw the 2D-Form of the building. The next function generates the polygons for each floor according to the basic form of the buliding - also definition of the angle of rotation and the adjustments of the floors. The function „GetAllCurvePoints“ reads out the edgepoints of the polygons. The next one divides up the building in façade sides and draws lines for each floor. The function „DivideLines“ seperates the lines in segments. These segments and all lines of each façade are read in the final function „makeTriangles“ where the form will be triangulated *(1 Episode of Rhinoscript*).
The „Sub Main„ the main part of the script includes the defintion of the parameters. Other important parts are the user input for constructing the building and the process for saving the output as an CAT.part- file (the CATIA-format).This step makes the script self-sustaining.
The user defines his input for the building in the script (*user input*- floors, height, edgepoints, façade segments and the path to save the CAT.part-file). The first step after having started the script is to draw manually the edgepoints of the building. After some seconds the building is generated. For the final composition the CAT.part- file will be imported in Digital Project and afterwards saved as a stp-format for Rhino3D (*2 Episode of Rhinoscript)*.

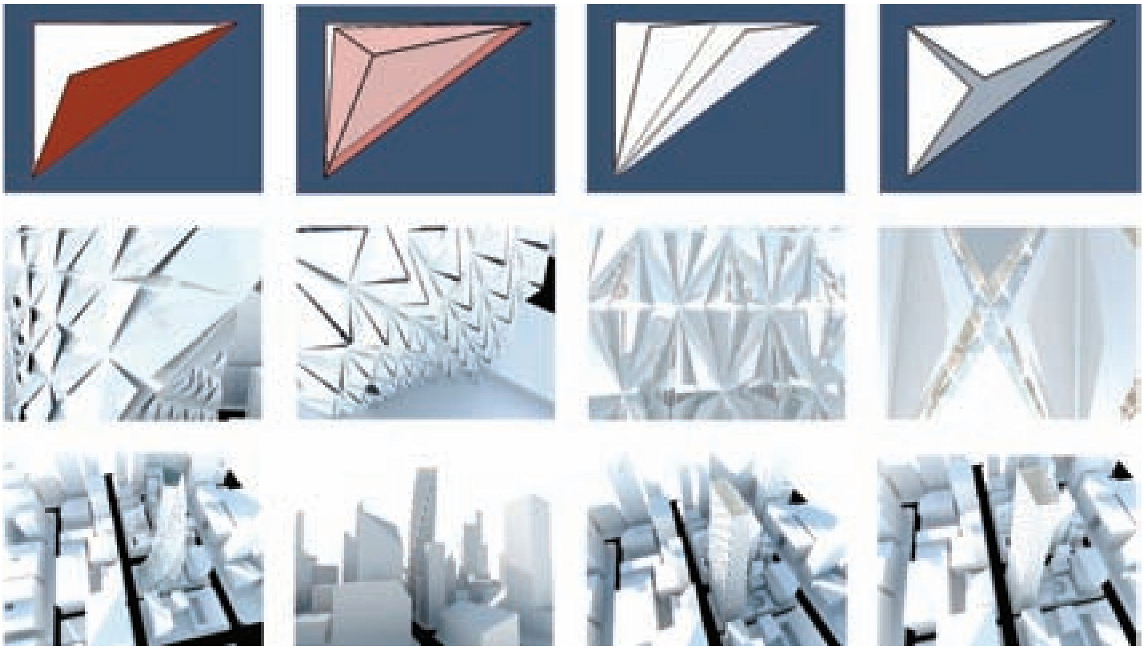


screenshot of Rhinoscript: functions and user input


mod.01
mod.02
mod.03
mod.04
mod.05
0%
25%
50%
75%
100%

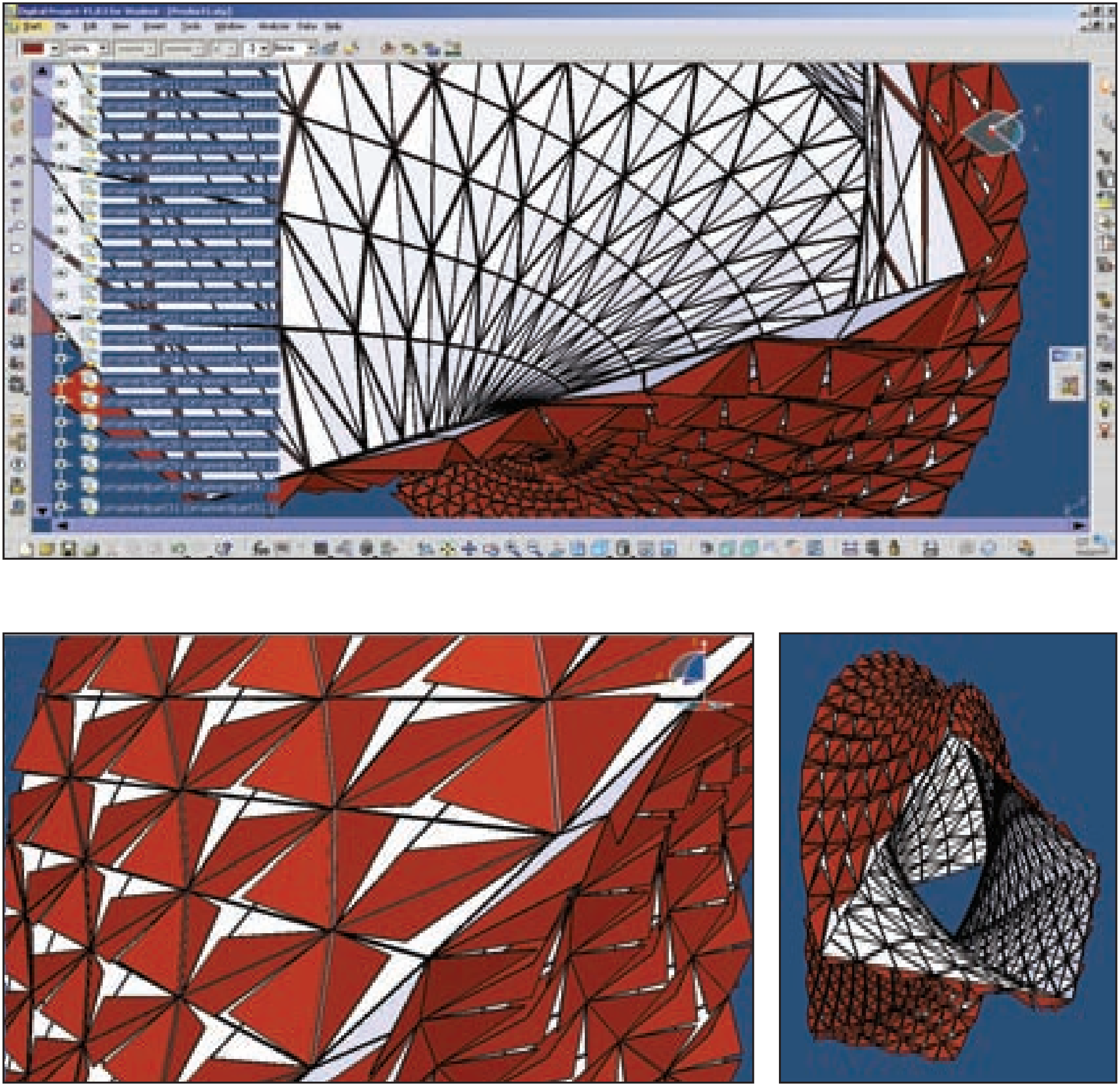
Technique
As mentioned before the elements for the façade in Digital Project are based on triangle-forms. One parameter controls the distance between the window and the sun panel. The second one steers the size of the opening (the first two parameters react according to the function and the surrounding area) and the last parameter steers the degree of the sunprotection (orientation of the building). The lower triangles react different like the upper ones. They create a visual breast in contrast to the triangles above.

Different elements had been testet to improve the technique to develope results faster and easier. The first four elements consist of more hard shaped forms and gave the tower a radical and aggressive appearance (*triangle façade elements*).
The final element (mod. 05) that I chose for the Warteck Tower in Basel is characterized by it`s chamfered curves and gives the tower a softer aesthetics in contrast to the crude environment.



mod 01 mod 02 mod 03 mod 04

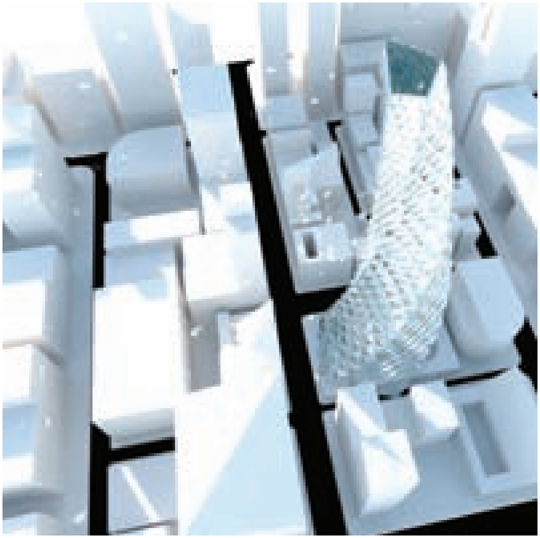
triangle façade elements: first row: Digital Project- second and third row: renderings MX Studio

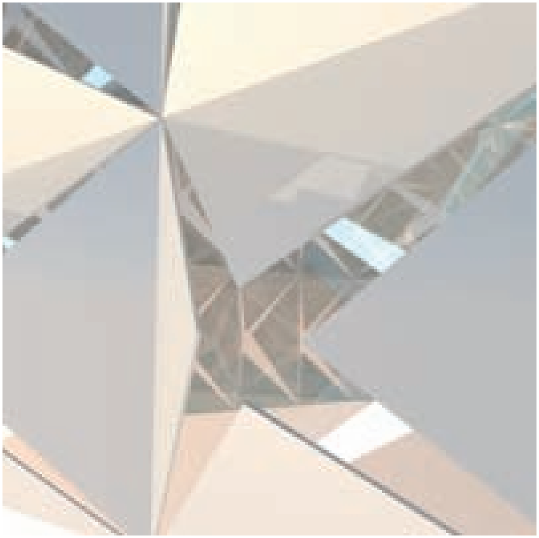


screenshots of Digital Project/CATIA: import of CAT.part-file: building with module 01

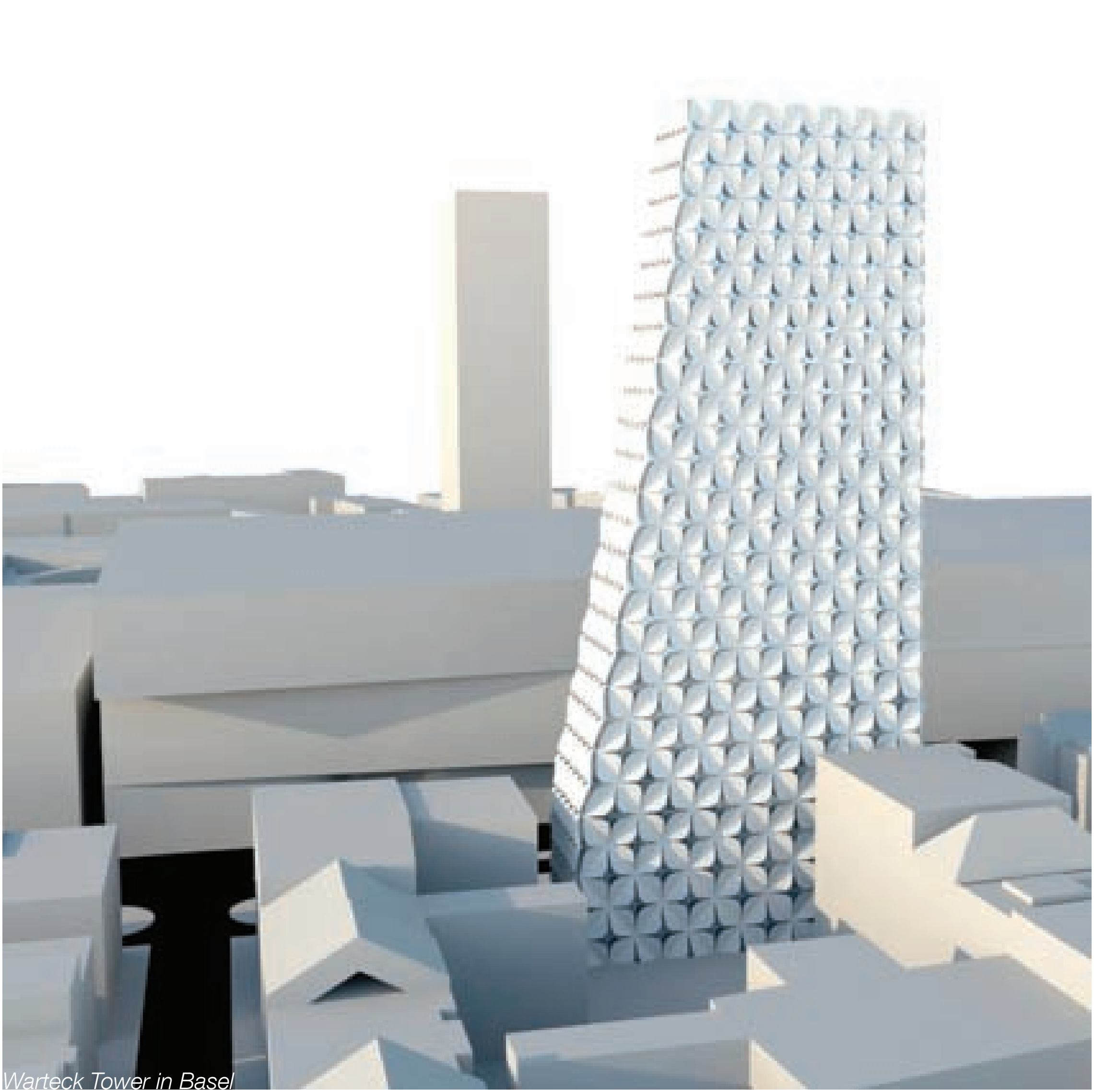
Result
The new script is quiet easy to use in terms of making simple parametric façades. The whole script is done in Rhino and the export from Rhino directly in Digital Project facilitates the process. Variations, changes, extensions can be done faster which is important especially for the design process. The work with two visual programs gives the user a better control.
As you can see on the following pages I experimented a lot with the new script according to variations and changes. It also offers the possibilities to create freeformed surfaces which extends the scope for development in the design process of the form. The user input helps to specify the building in an uncomplicated way.

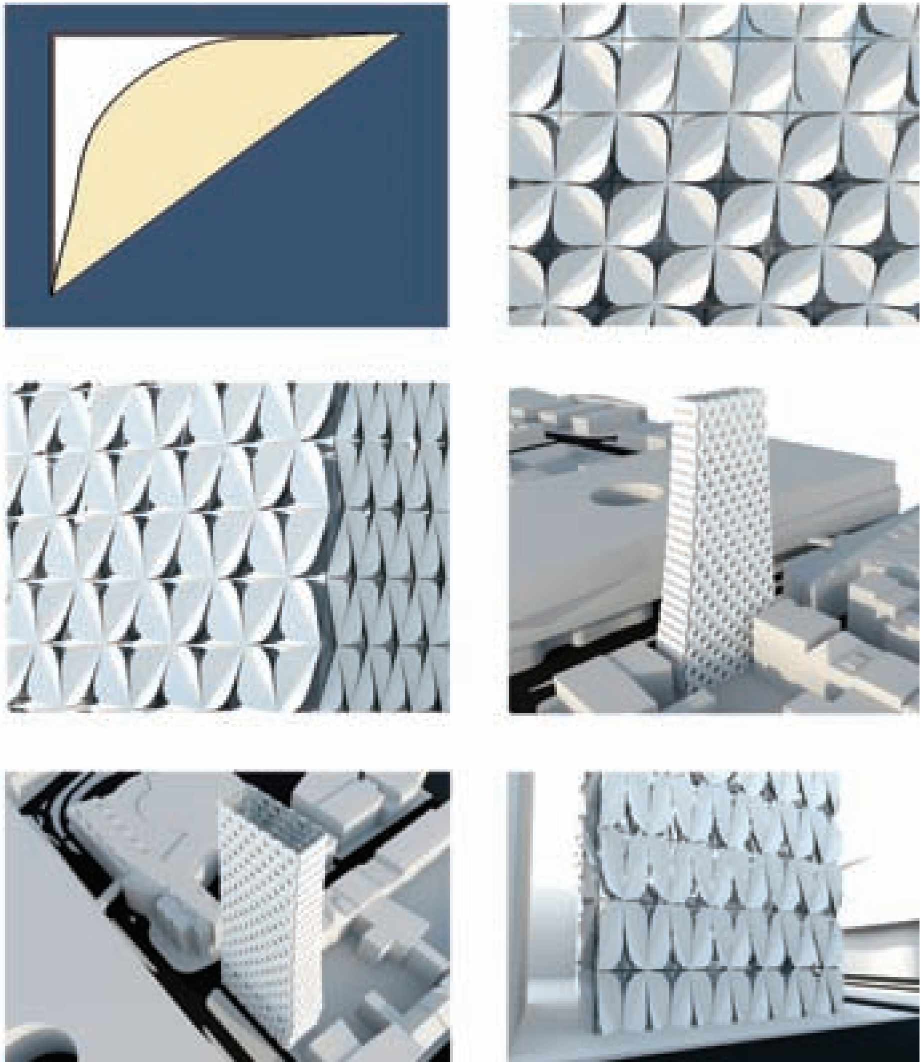
Of course there are other options to deal with freeformed surfaces. This method of triangulation provides a geometrical form for the façade element - it has to be a triangle in his basic shape. Consequently there are four pieces which have to be drawn in Digital Project because one element consists of this number. In contrast to the first case study the new script limits the variety of the openings - because the script generates a homogeneous „picture„ of a façade.
This script gives a good access to work with this practise in generating façades in an easier and quicker way for complex forms. And it also sensibilizes the users contact with digtital tools.





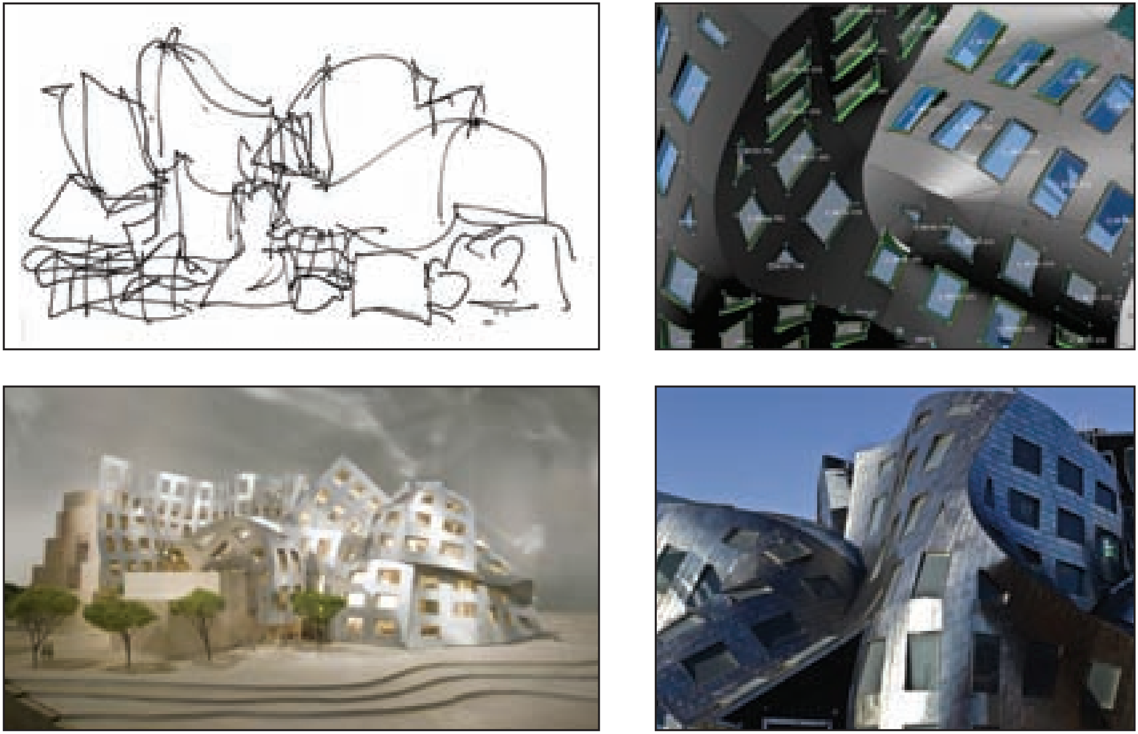
renderings of module 01: transfer Digital Project > Rhino3D > MX Studio


Warteck Tower in Basel



renderings of the final building (mod 05): transfer Digital Project > Rhino3D > MX Studio

„…it is becoming apparent that here as well, the generation of geometries using scripts should be replaced by their generation using parametric CAD software in the long term in order to be feasible by architects without specialized skills in computers.“



Gehry Partners LLP, Lou Ruvo Brin Institute- fluent practice: the process of a sketch > model > building.

FURTHER DEVELOPEMENTS | OUTLOOK

Digital Façade - Instructions for design parametric façades gives a good overview about the parametric procedures concerning an easy handling with the design of façades. My aim was to implement a new tool for architectural offices that don‘t have a special background with it. The scripts are already sufficient to use them in competitions where the level of detail is not that deep. The usage of them help to optimizes the design process a lot.
Every example has it‘s own way to create parametric façades.There‘s not only one solution how to generate them. This guide tries to analyze them and shows all the possibilties, advantages and disadvantages they‘ve got.

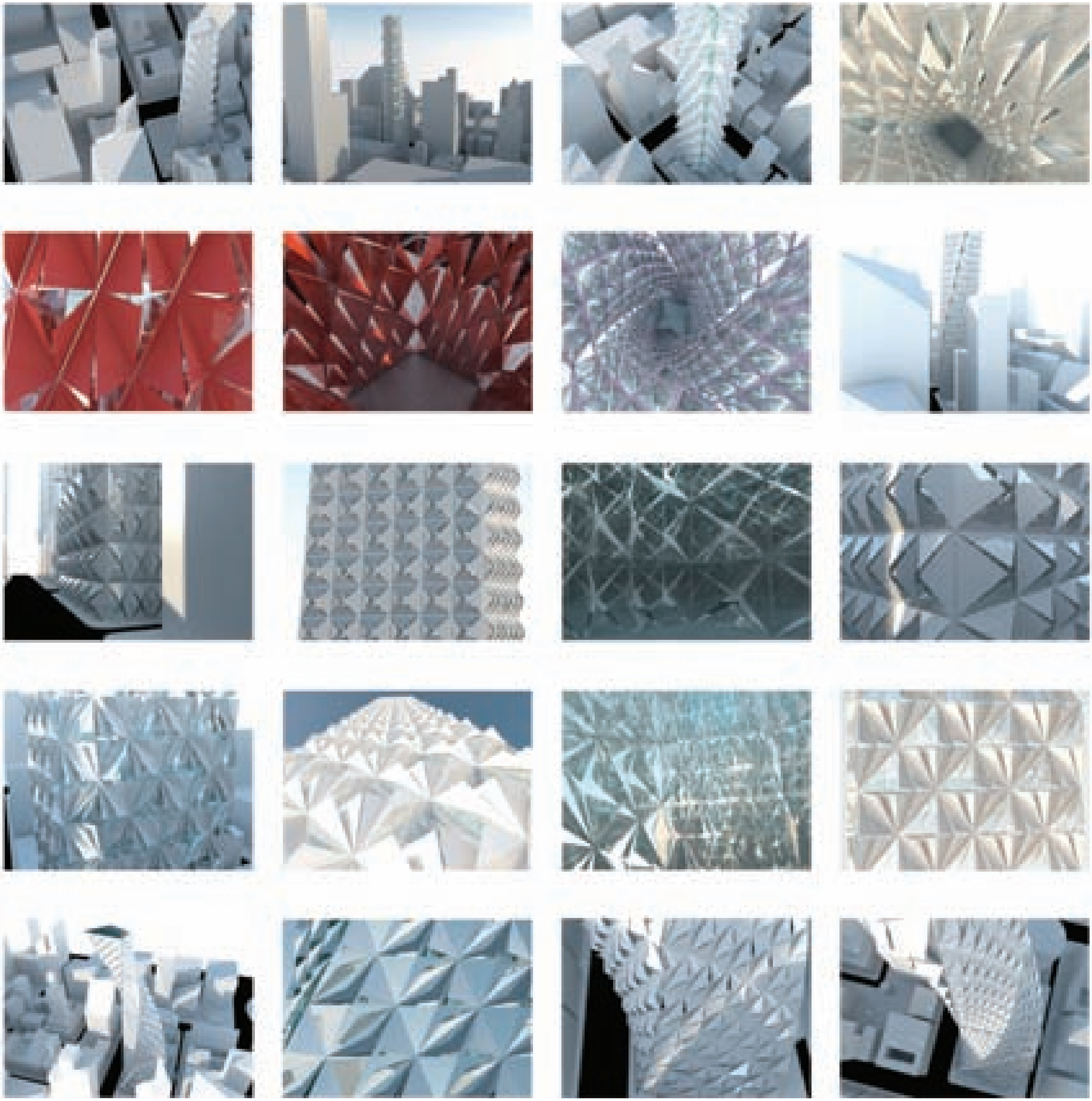
The results of the first two case studies leads to the third (case study 3) that extend the design possibilties and improves the process. I think this script has potential for further developements. In this context I collect some next steps which can be implemented. First of all the handling with free forms resp. with modern geometry can be described with other new methods. This would delete the constraints of triangulation forms (the façade elements). The second step would be to solve the problem with the sizes of the elements. In the first case study this problem is negotiated through drawing the façade grid manually - a time consuming procedure. The solution must be a combination of these two case studies. The third step could be to work with interfaces/applets where the user has a better overview of the program and his possibilities to deal with.

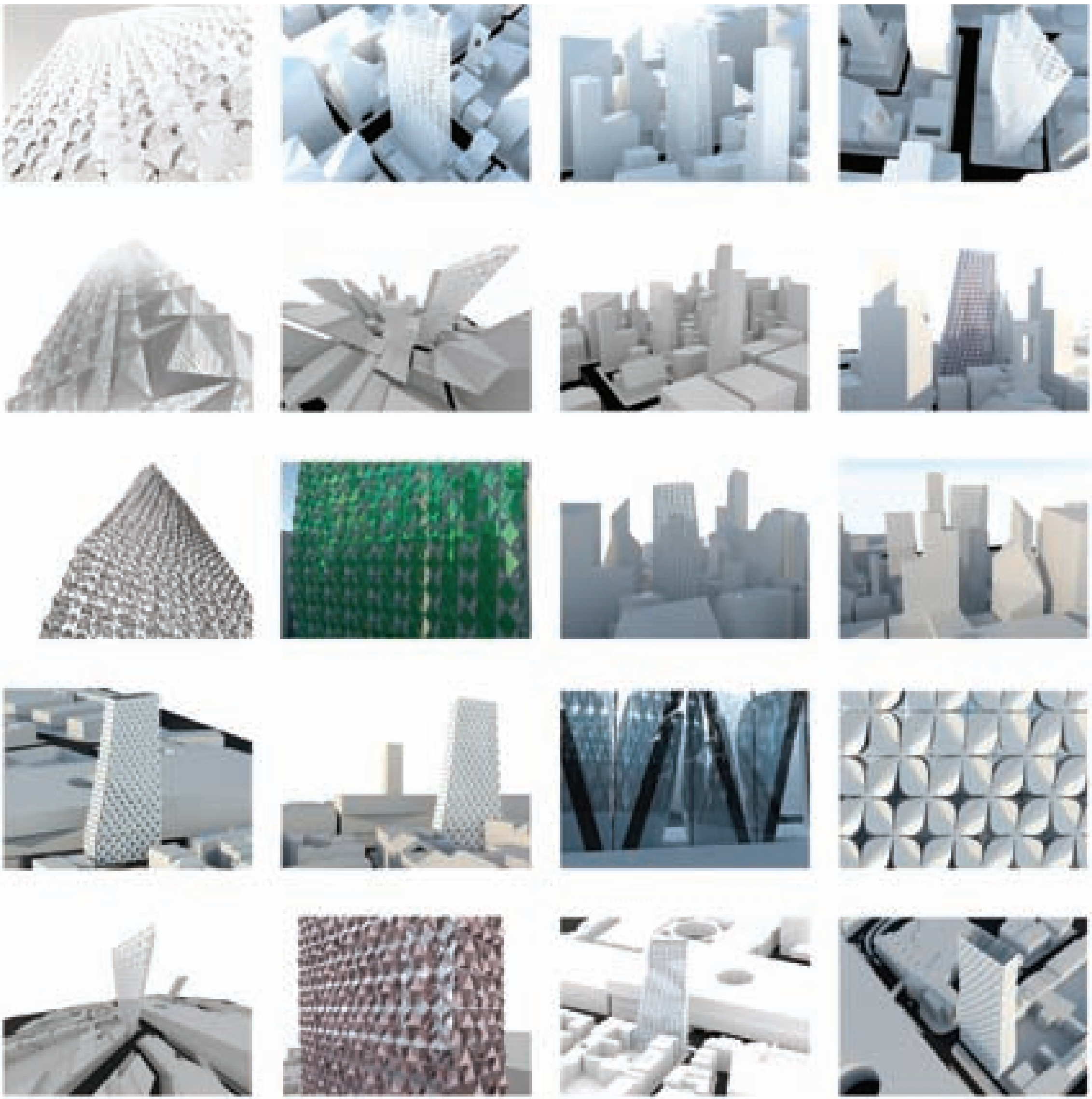
It would be interesting how to work with these case studies after the competitions. The office of Frank Gehry created in the early nineties buildings with the support of the program CATIA. Since that time they improve a lot that working process of the digital chain. They are already able to use CATIA in a detailling level that gives them the opportunity to create outputs that can be use for productions. This technique allows them a fluent practice: the process of a sketch to a model and to a building. And especially for Gehry himself offers this method a great freedom who knows that his sketches are „producable„.

But the disadvantage is that the process is still very difficult to handle without any experience in programming. It would be a challenge to develope methods that simplify the procedure for „laymen„-architects.

Including the further developements in programming parametric façades it will be necessary to work with algorithms. They help to optimize the façade in certain ways. For example the tower of the last case study consists of over thousand different elements. An algorithm could calculate each variance and make ten different groups of similar elements. Through this it is possible to have at least 10 different elements instead of thousand. This is an enormous advantage to reduce time and cost.

Digital Project already offers some algorithms for optimizations. It is up to the users and developers to test them and give that knowledge to architects. One day the handling with parametric façades and their algorithms will be naturally and eventually necessary.
This guide is one step to give an idea about the new comprehension and work with digital tools.





REFERENCES

Books
Lindsey, Bruce, *Digital Gehry, Material Resistance Digital Construction*, Birkhäuser, Turin 2001

Schumacher, Patrik, *Digital Hadid, Landscapes in Motion*, Birkhäuser, Turin 2004

Jormaka, Kari, Basics, *Methoden der Formfindung*, Birkhäuser, Berlin 2008

Kolarevic, Branko, *Architecture in the Digital Age: Design and Manufacturing,* Spon Press, 2003 New York

Magazines, Essays, Readers
Trentin, Luigi, Werk, Bauen+Wohnen: *Fassade als Darstellung*, *Die Architektur von Livio Vacchini*, 12|2005 Zürich

Various, Archithese:*CAAD*, 4|2006 Zürich

Various, AAG 2008: *Advances in architectural Geometry, First Symposium on Architectural geometry*, 2008 Vienna

Lemmerzahl, Steffen, *Nachdenken über parametrische Architektur*, ETH Zürich- CAAD Professur Ludger Hovestadt 2008

Internet Sides
http://www.gehrytechnologies.com/

http://www.cad.de/

http://www.3ds.com/ or *http://www.dsweb.com/*

http://www.arcspace.com/

http://www.gtwiki.org/

http://www.mas.caad.arch.ethz.ch/