„Recent theories of form in architecture have focused on computational methods of formal exploration and expression. From topological geometry and hypersurfaces to blobs and folds, there is a clear tendency to seek and explore formal properties as sources of ordering systems. For the last two decades, designers have been concerned with the use of computational mechanisms for the exploration of formal systems. These practices have attempted to readdress formal issues using new techniques and methods. Computational tools are central protagonists in this exploration.“

Kostas Terzidis

Algorithmic Architecture

Introduction to the MAS Colloquia 2006/07

- **Computer as algorithmic machine**
- **Development of digital architecture**

Computer as Algorithmic Machine

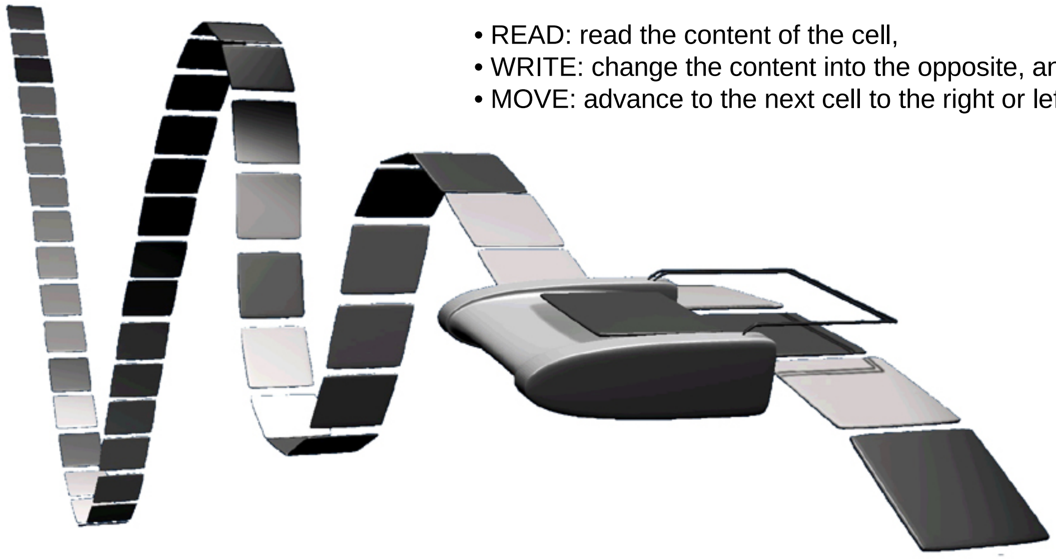
A computer is a machine that transforms input data into output data. Thereby data takes the form of a finite sequence of bits. Hence, data can be coded as a natural number and the transformation **f** can be viewed as partial function on the set of natural numbers $\mathbf{N}$ with output **out** ϵ $\mathbf{N}$ as result of a computation of the input **in** ϵ $\mathbf{N}$ that is **f**(**in**) = **out.**

$$\begin{tikzcd} \text{in} \arrow[r] & \boxed{f : \mathbf{N} \longrightarrow \mathbf{N}} \arrow[r] & \text{out} \end{tikzcd}$$

What kind of functions can be computed? In the 1930’s various mathematicians started to develop precise, independent definitions of what it means to be computable. It is a remarkable mathematical fact that all the different precise definitions of computability lead to the same class of functions. The most intuitive one was given by Alan Turing in 1936 and the underlying logic is closely connected to the later development of real computers.

A Turing machine consits of an infinite one-dimensional tape divided into cells, a movable read-write head with a specified starting position, and a table of transition rules that serves as the program for the machine. Each cell of the tape contains one symbol, either 0 or 1, and the head can move along the tape to scan one cell at a time and perform three different activities:

- READ: read the content of the cell,
- WRITE: change the content into the opposite, and
- MOVE: advance to the next cell to the right or left along the tape.



Conclusion: computable functions are precisely the algorithmic ones!

Example of a Turing-machine: Adding two numbers

input

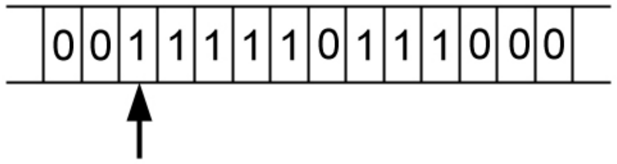


table of transition rules

$C_1,0,1,C_2$

$C_1,1,>,C_1$

$C_2,0,>,C_3$

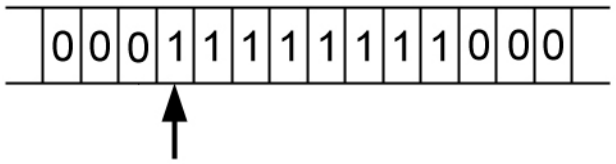
$C_2,1,<,C_2$

$C_3,0,>,C_4$

$C_3,1,0,C_3$

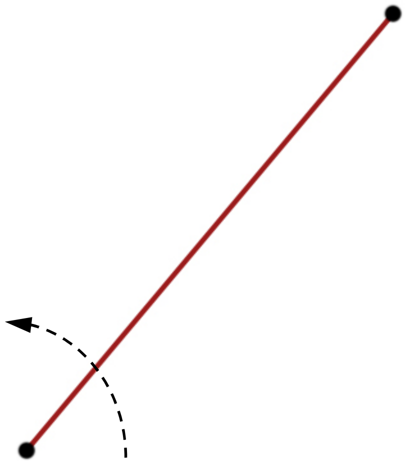
$C_4,0,>,C_4$

output



The pointer of this program runs through the left block of 1‘s from left to right till it reads a 0 the first time. Then it changes this 0 into 1 and moves back from right to left to the starting position. There the 1 gets overwritten by a 0 and the program stops.

Drawing a line in a CAD-system
=
Scripting a line in a CAD-system



select line-tool
draw line from start point to end point
MoveTo(. , .)
LineTo(. , .)

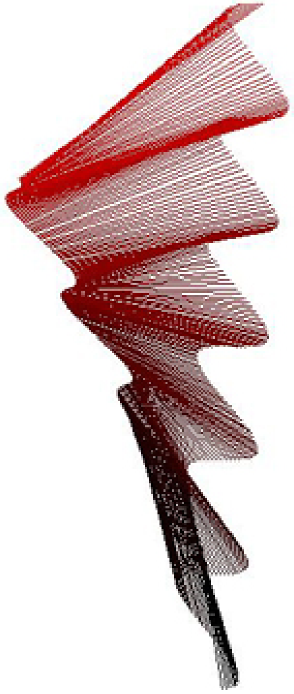
select a color
PenFore(. , . , . , .)

select tool to manipulate line
RotatePoint(. , .)

Algorithmic description:

```
PenFore( . , . , . , .);
MoveTo( . , . );
LineTo( . , . );
RotatePoint( . , . );
```

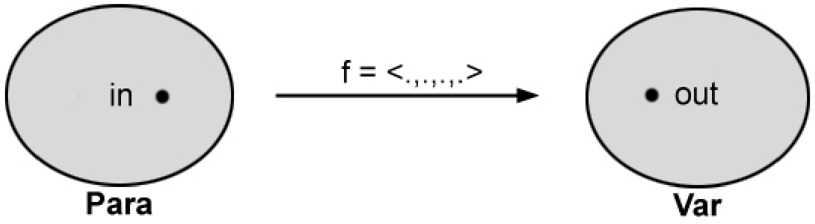
Example of scripted line-drawing
The aesthetics of complexity

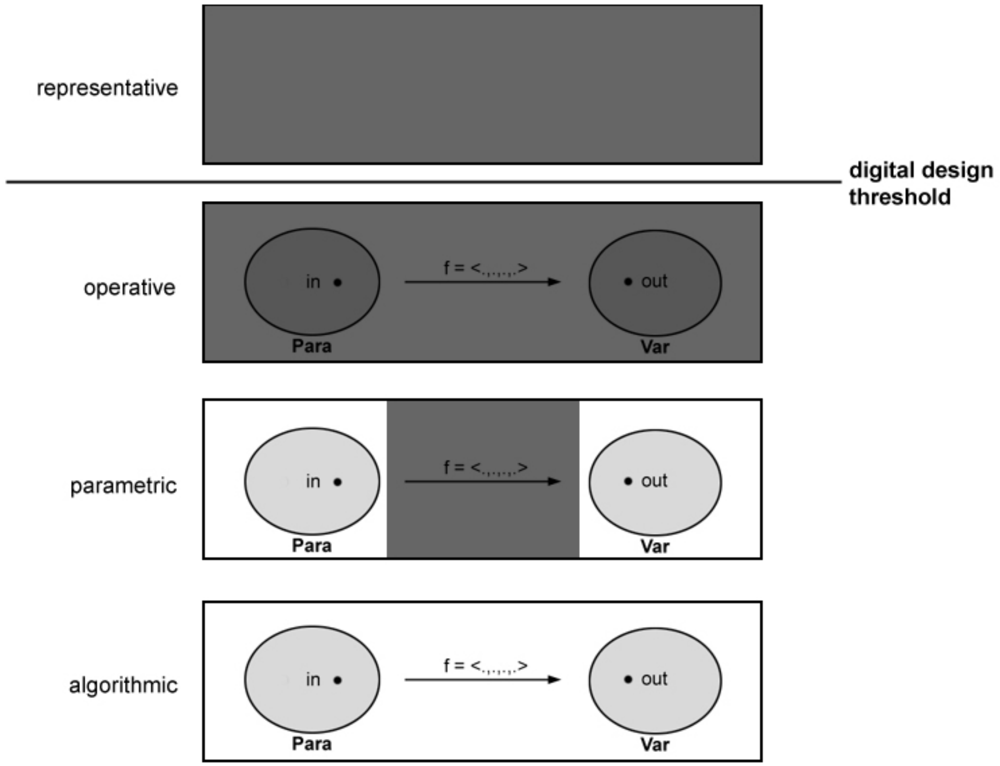


```
PROCEDURE Script1;

VAR
  i,j : INTEGER;      (* counters *)
  r,g,b : LONGINT;  (* variables for rgb-colors *)

BEGIN
  r:=0;g:=0;b:=0;   (* set color to black *)
  j:=12;                      (* parameter for manipulation *)
  FOR i:=5 TO 300 DO
    BEGIN
      PenFore(r+i*175,g,b);                              (* shift of color from black to red *)
      MoveTo(2*i+10*j,Cos(2*pi*i/100)*15+10*j); (* line from start point *)
      LineTo(i+20*j,Sin(4*pi*i/100)*25+18*j);        (* to end point *)
      RotatePoint(i+2*j,2*j+i,-10*j);                    (* rotate line *)
      DSelectAll;
    END;
END;
RUN(Script1);
```

The Algorithmic Perspective

The repeated use of the function **f** that describes the drawing of a line demonstrates an important point: Every function **f** has a set of valid input, the parameter space **Para** ϵ **N** and the image **f**(**Para**) of the parameter space is the set of possible variations **Var** e **N** in the output of the Turing machine. The potential of digital design lies in the activation of this space of variation!





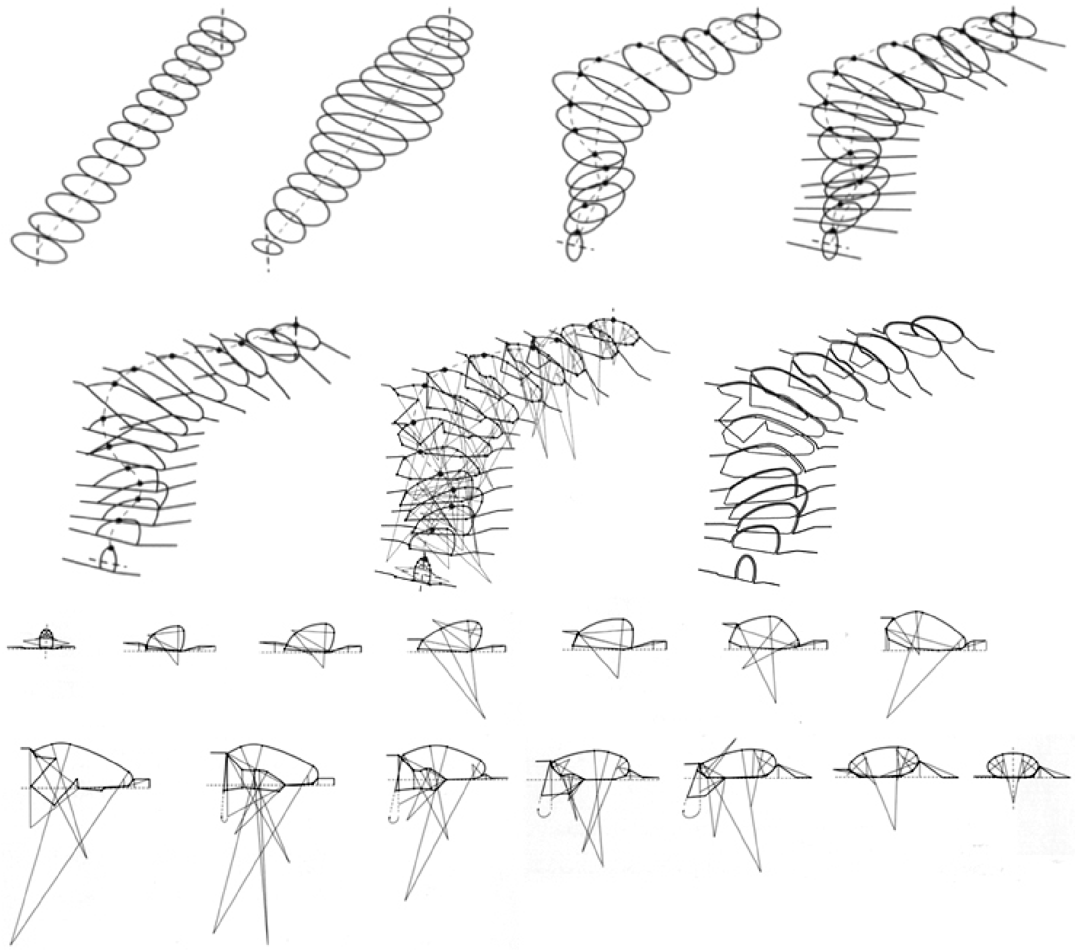
The threshold to digital design in architecture can be defined as the conscious overcoming of the traditional level of representation in the use of the computer as design tool.

Looking at the development in architecture since the 1990s one can distinguish three degrees of computational awareness in this process of acquisition of the computer into architectural design:

- the operative,
- the parametric, and
- the algorithmic.

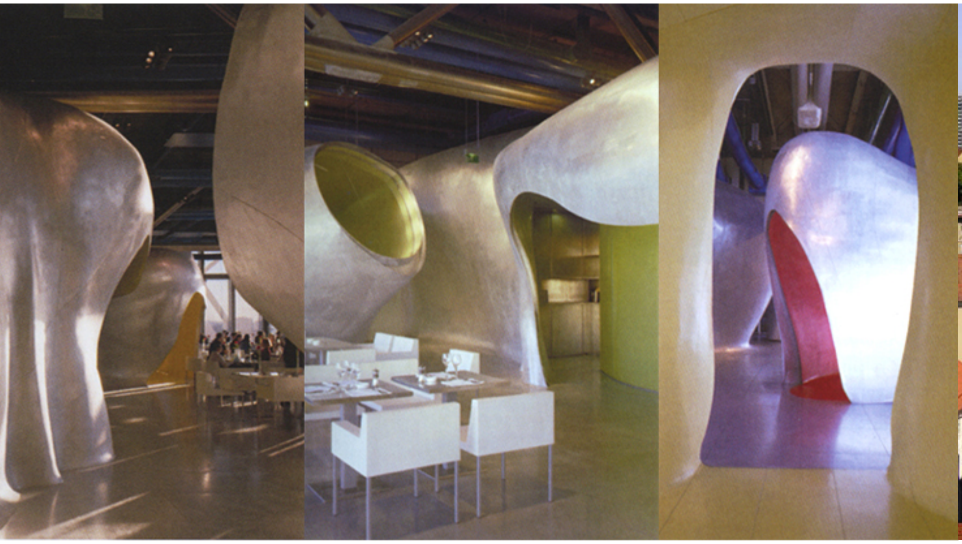
Operative Architecture

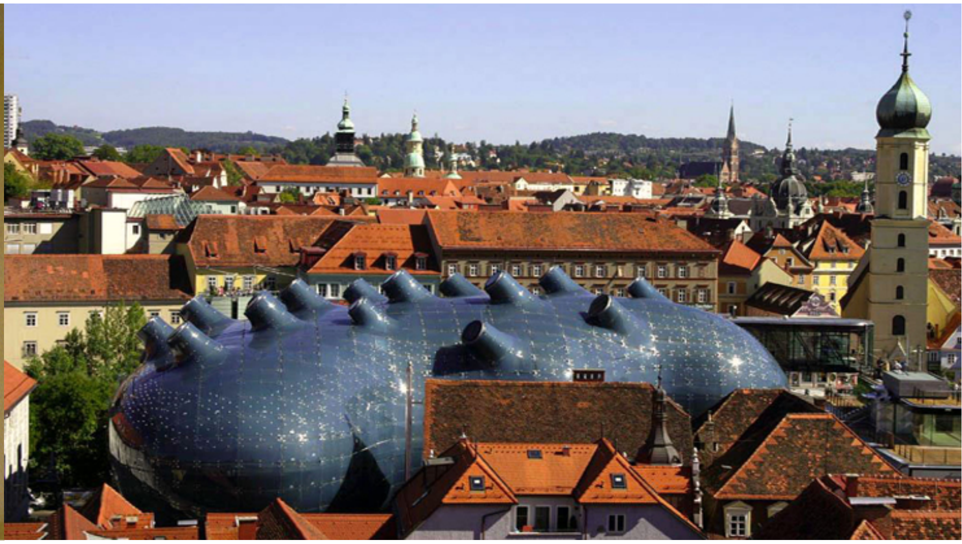
On the operative level, the computer gets used for modelling in a pre-defined geometric way. That is implemented geometric operations of the software in use are explored in an architectural context in order to deform the classical formal language of architecure by means of controlled transformations. What distinguishes the operative from the representative is the type of geometric operations used for modeling, for example the concatenation of circular segments in the process of meodeling the frame structure of the water pavilion by Lars Spuybroek.

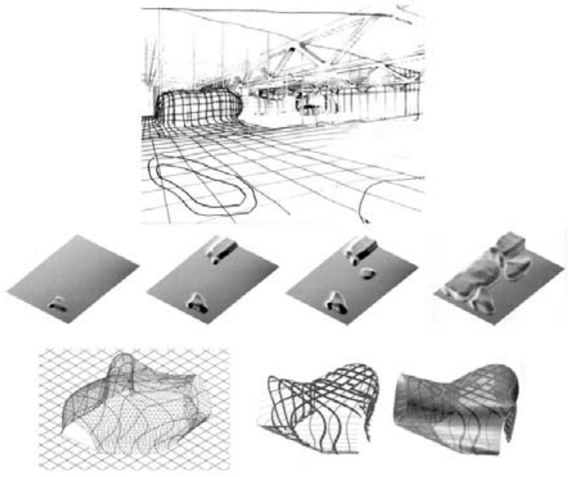


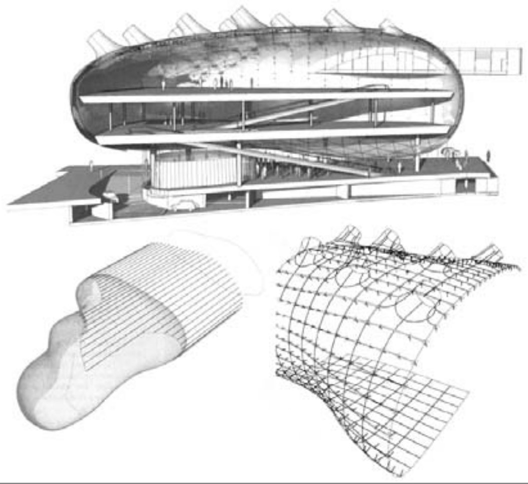
Lars Spuybroek: Water-experience pavilion, NeeltjeJans, Netherlands, 1993-97

Without a computer, such operations were not used widely in architecture because of the inherent increase of geometric complexity which limits the ability to handle them in an efficient way by means of drawing as well as imagination. In the case of the operative, it is the computational power of the computer that opens up a new field of geometric possibilities in modeling. This gets obvious in the architectural use of the curvilinear language of contemporary CAD-softwares, the NURBS-geometry, like in the Kunsthaus Graz by Peter Cook and Colin Fournier or the design of the restaurant in the Centre Pompidou by Jakob & MacFarlane.


Jakob & MacFarlain: Restaurant Le Georges, Centre Georges Pompidou, Paris, 1999

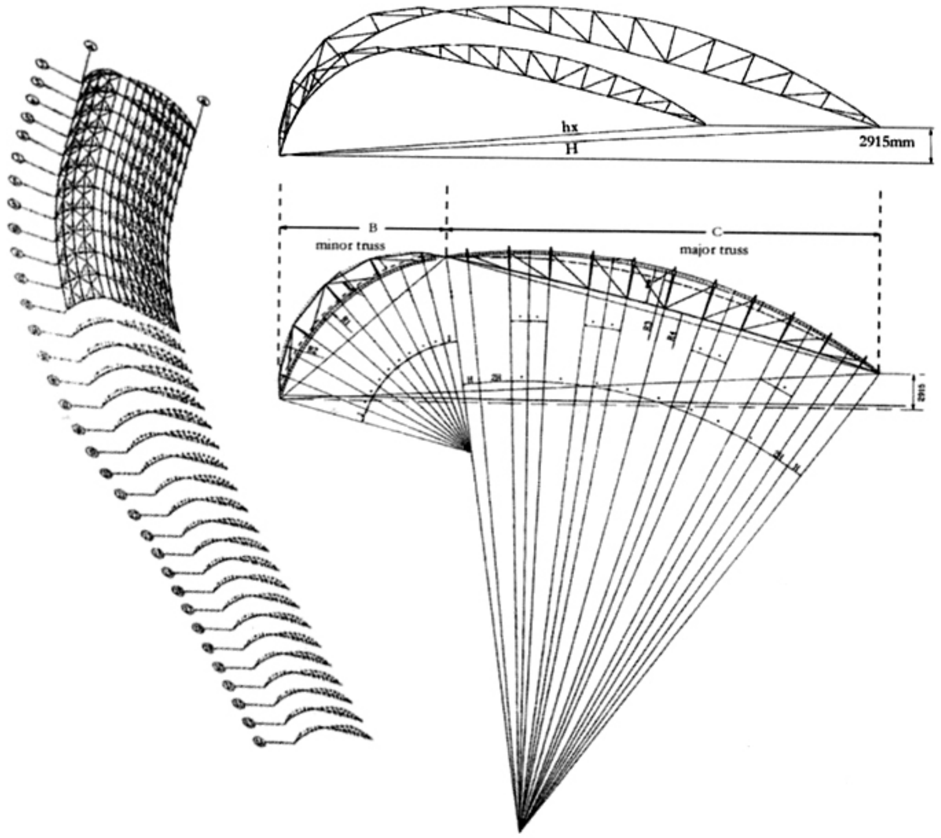

Peter Cook & Colin Fournier: Kunsthaus, Graz, 2002-03





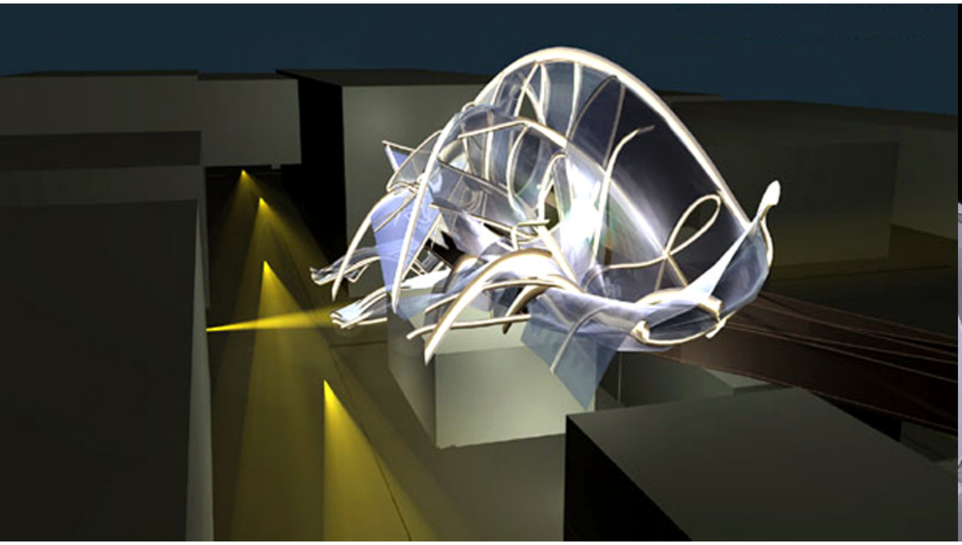
Parametric Architecture

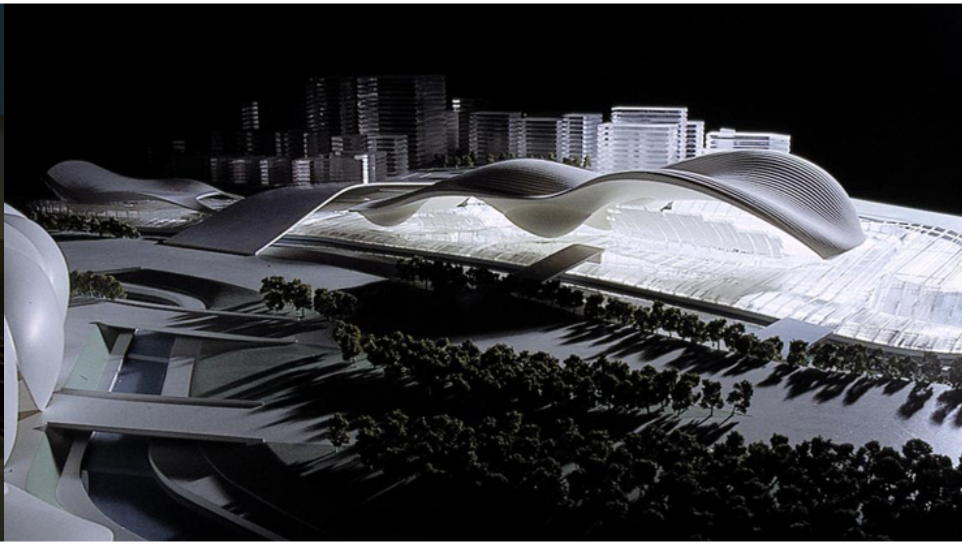
Especially the closer examination of the NURBS-geometry has fostered the parametric awareness and has helped to shift interest away from drafting and modeling towards a more mathematically based view on architectural design. This has to do with the fact that every NURBS object is defined within a local space of parameter given by control points and weights. These datas are not fixed but can be changed throughout the whole design process. That is why “parametrics can provide for a powerful conception of architectural form by describing a range of possibilities, replacing in the process stable with variable, singularity with multiplicity.” (Kolarevic) The design of thirty-six dimensionally different but identically configured three-pin bowstring arches for the International Terminal of Waterloo Station in London by Nicholas Grimshaw and Partners is an example for this parametrized variation.



Nicholas Grimshaw and Partners: International Terminal, Waterloo Station, London, 1993

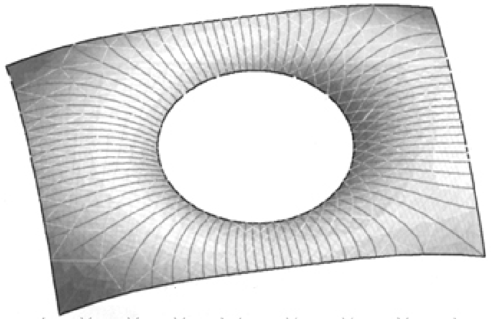
But by far the most popular way of using parameters in contemporary architecture is the utilization of time as primal parameter. Time-based techniques as morphing, keyframe animation, kinematics, force fields, or particle systems are widely used in the design process nowadays and are all based on the idea of gradually deforming a given NURBS-geometry by changing the parameters over time. Examples for this approach are Greg Lynn's project for the Port Authority Gateway or the Aquatic Center by Zaha Hadid.

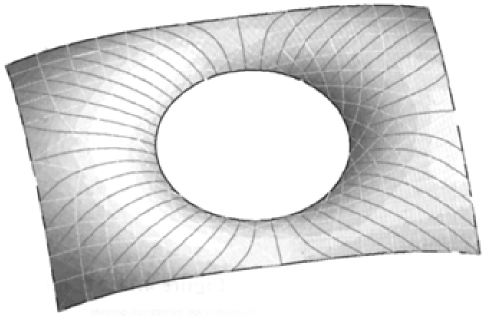

Greg Lynn: Port Authority Gateway, New York, 1995


Zaha Hadid: Aquatic Center, London, 1005-09

Algorithmic Architecture

As the Turing model shows, the strength of the computer as device is the flexible series of commands and logical procedures that can instantly transform it from one function to another. However, on the operative as well as one the parametric level, architects are forced to conduct the process of design using fixed Turing machines originally developed to solve the problems faced in different areas of use, for example in aircraft design or film-making.Therefore, over the last years many architects have turned to the inhouse creation of code appropriate to their specific needs. Only this step towards the algorithmic decription of the design made projects like the British Museum Great Court Roof by Norman Foster and Partners, the Serpentine Gallery Pavilion by Toyo Ito, the architecture for the Olympic Games in Beijing by PTW and Herzog & de Meuron, or Ocean North’s design for the Music and Art Center possible.

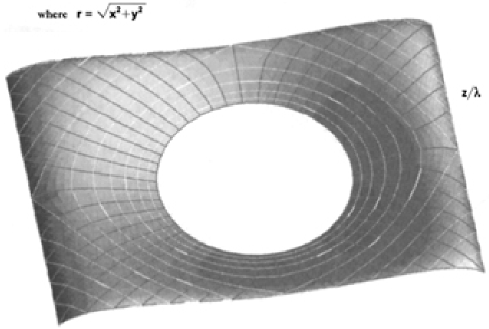




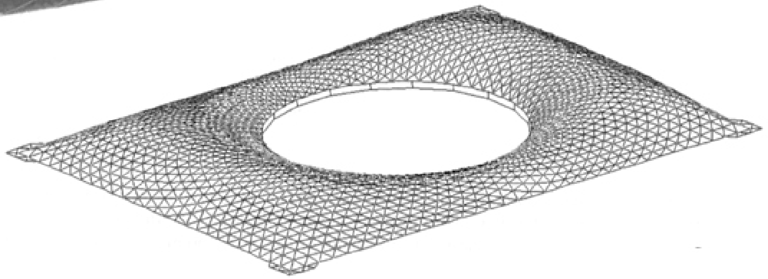
$$z/h=\left(1-x/b\right)\left(1+x/b\right)\left(1-y/c\right)\left(1+y/d\right)/\left(1-ax/rb\right)\left(1+ax/rb\right)\left(1-ay/rc\right)\left(1+ay/rd\right)$$

where $r=\sqrt{x^2+y^2}$

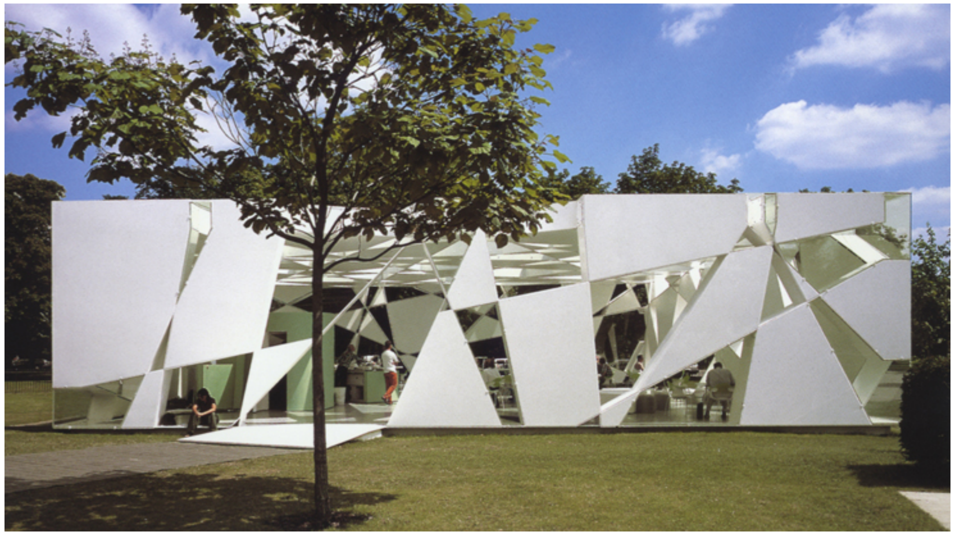
$$z/H=\left(1-x/b\right)\left(1+x/b\right)\left(1-y/c\right)\left(1+y/d\right)\left(\sqrt{x^2+y^2}/a-1\right)$$

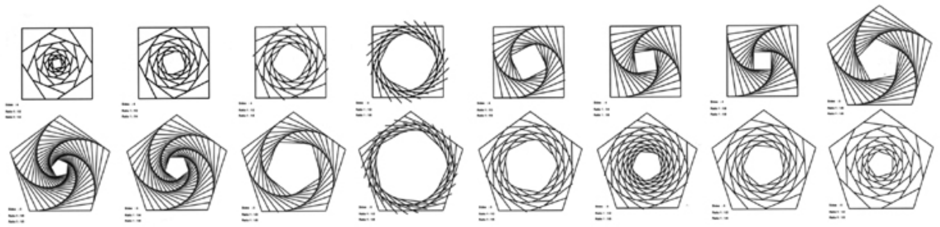


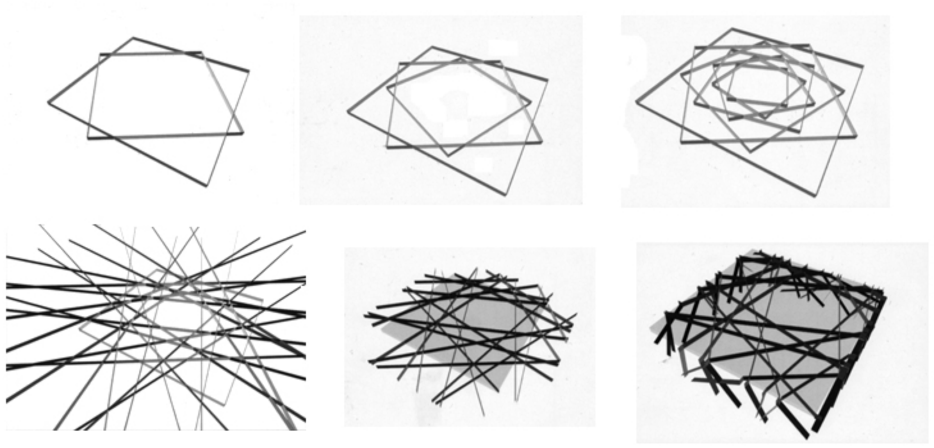
$$z/\lambda = \left(\sqrt{x^2+y^2}/a-1\right)\Bigg/\begin{bmatrix}\sqrt{\left(b-x\right)^2+\left(c-y\right)^2}/\left(b-x\right)\left(c-y\right) & + \\ \sqrt{\left(b+x\right)^2+\left(c-y\right)^2}/\left(b+x\right)\left(c-y\right) & + \\ \sqrt{\left(b-x\right)^2+\left(d+y\right)^2}/\left(b-x\right)\left(d+y\right) & + \\ \sqrt{\left(b+x\right)^2+\left(d+y\right)^2}/\left(b+x\right)\left(d+y\right) & \end{bmatrix}$$



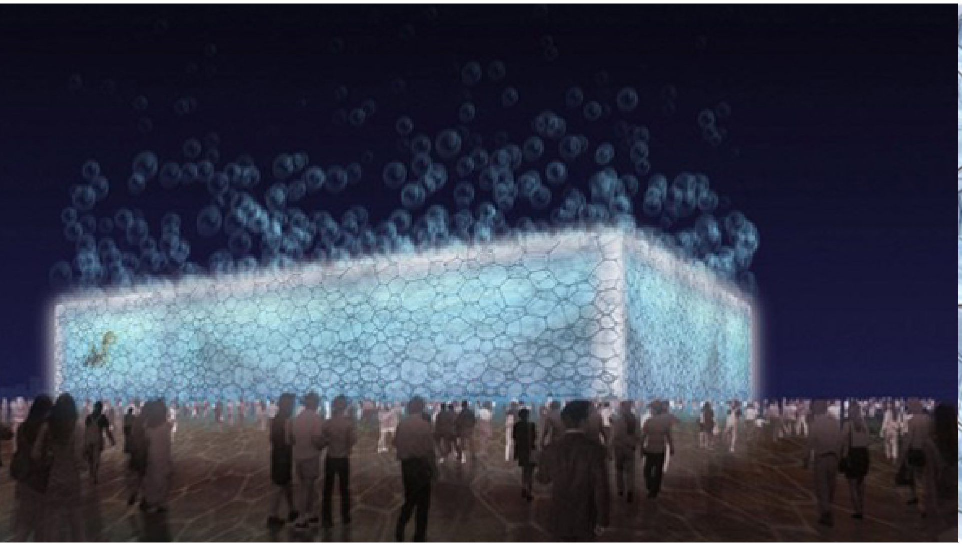
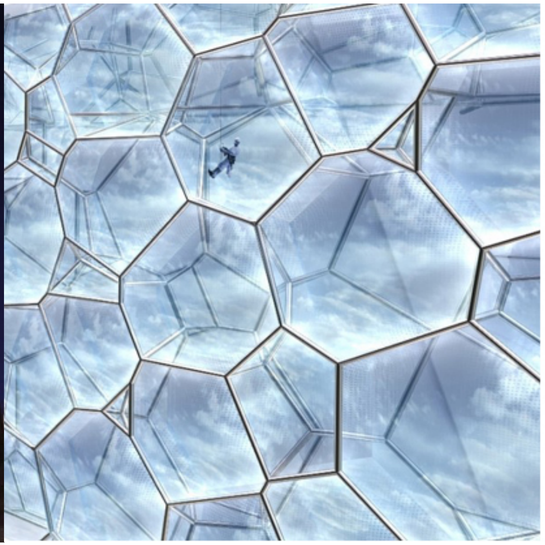
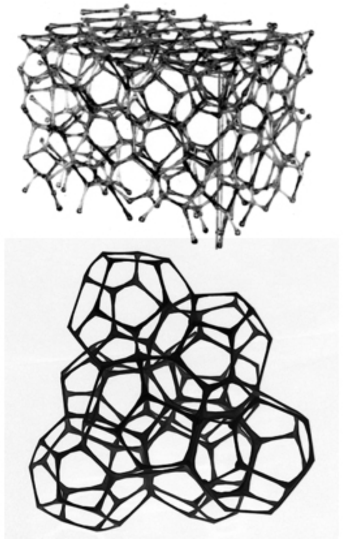
Norman Foster and Partner: Great Court Roof, British Museum, London, 1999-2000



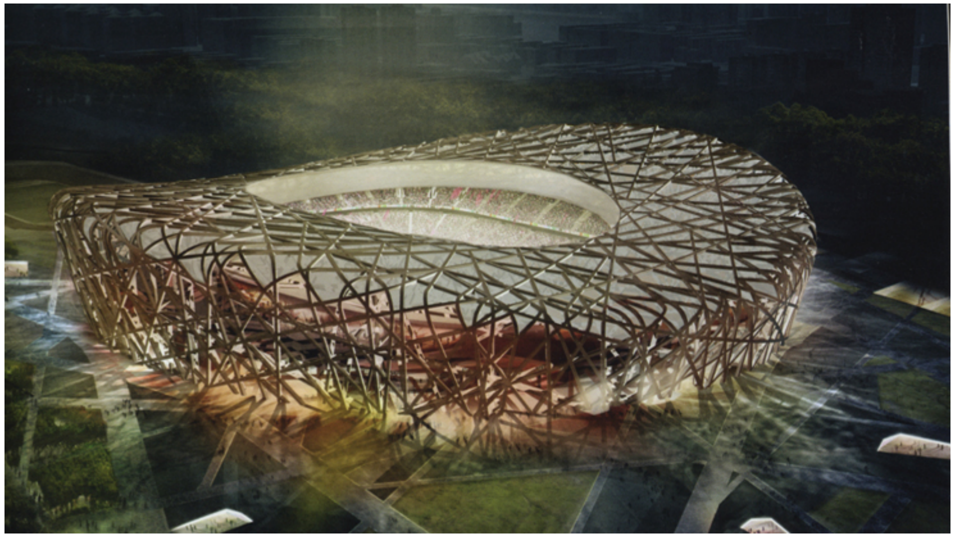


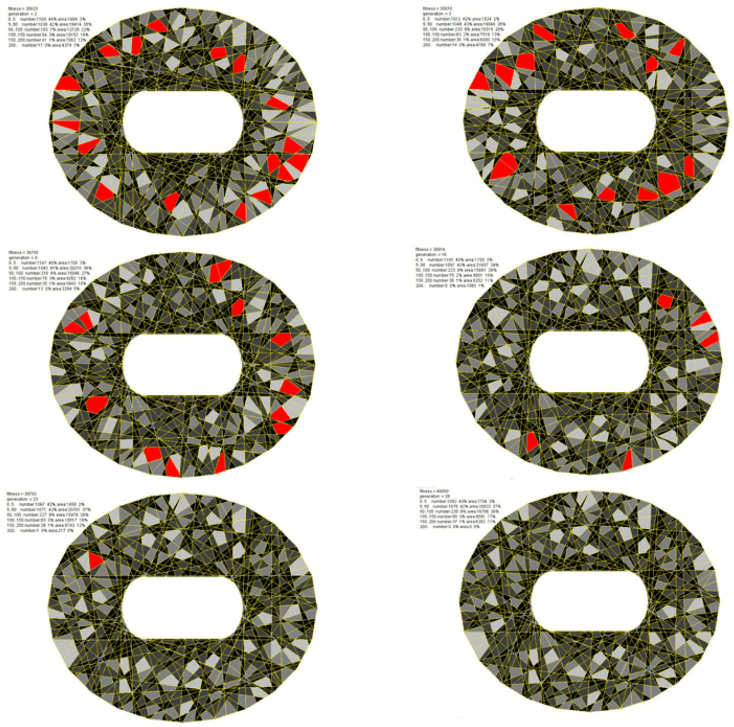


Toyo Ito: Serpentine Gallery Pavilion, London, 2002

PTW: National Swimming Center, Beijing, 2003-06





```
fitness = 39625
generation = 2
0..5    number:1100  44% area:1664  3%
5..50   number:1038  42% area:19414  35%
50..100 number:183  7% area:12729  22%
100..150 number:84  3% area:10152  18%
150..200 number:41  1% area:7062  12%
200..   number:17  0% area:4374  7%
```

```
fitness = 39810
generation = 3
0..5    number:1012  42% area:1524  2%
5..50   number:1048  43% area:19848  35%
50..100 number:233  9% area:16315  29%
100..150 number:63  2% area:7516  13%
150..200 number:36  1% area:6000  10%
200..   number:14  0% area:4189  7%
```

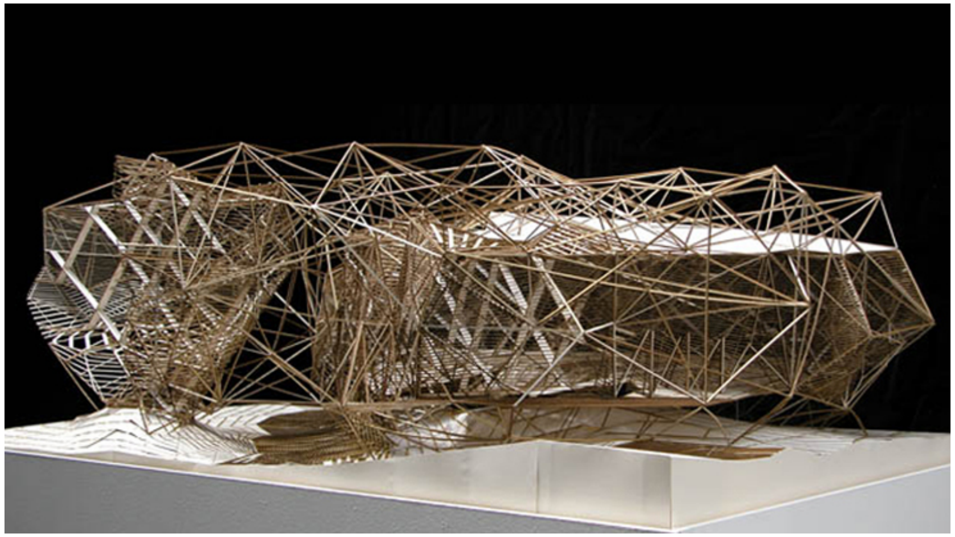
```
fitness = 39705
generation = 9
0..5    number:1147  45% area:1705  3%
5..50   number:1043  41% area:20215  36%
50..100 number:216  8% area:15046  27%
100..150 number:76  3% area:9292  16%
150..200 number:35  1% area:5843  10%
200..   number:13  0% area:3294  5%
```

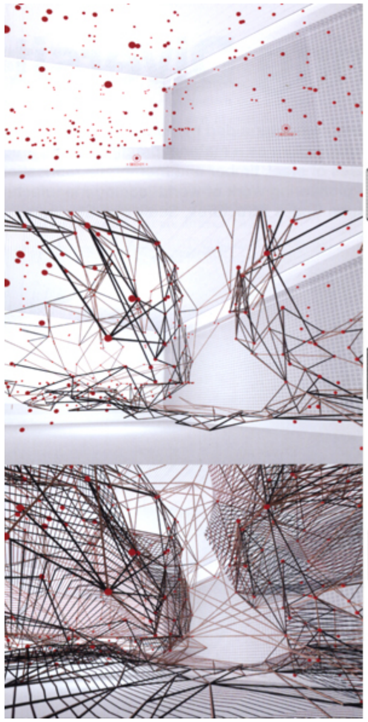
```
fitness = 39916
generation = 16
0..5    number:1101  43% area:1720  3%
5..50   number:1097  43% area:21657  39%
50..100 number:223  8% area:15881  28%
100..150 number:75  2% area:9001  16%
150..200 number:36  1% area:6252  11%
200..   number:5  0% area:1083  1%
```

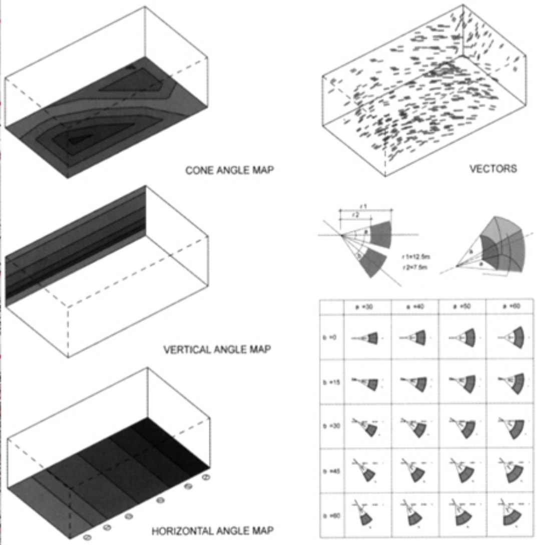
```
fitness = 39762
generation = 23
0..5    number:1067  42% area:1650  2%
5..50   number:1071  43% area:20781  37%
50..100 number:227  9% area:15870  28%
100..150 number:83  3% area:10017  18%
150..200 number:39  1% area:6743  12%
200..   number:1  0% area:217  0%
```

```
fitness = 40000
generation = 28
0..5    number:1093  43% area:1704  3%
5..50   number:1076  42% area:20933  37%
50..100 number:238  9% area:16786  30%
100..150 number:80  3% area:9591  17%
150..200 number:37  1% area:6382  11%
200..   number:0  0% area:0  0%
```

Herzog & de Meuron: National Stadium, Beijing, 2002-07







Ocean North: Music and Arts Center, Jyväskylä, 2004-05 (design phase)

The aesthetics of complexity!

Computational Geometry
• Voronoi diagram
• A* algorithm

Rule-based systems
• L-System
• Shape grammar

Self-organising systems
• Cellular automata
• Swarm system

Optimization
• Genetic algorithm

• the break down of the division between designing and making because of the developments in computer-aided design and manufacturing

• the phenomena of emergence in complex systems of organization and its use as morphogenetic design strategy; in general the question of science and architecture

Nov.	Dec.	Jan.	Feb.	March	April	May	June
Lecture 1	Lecture 2	Jan I	Feb I	March I	April I	Lecture 3	Lecture 4
		Jan II	Feb II	March II	April II		

http://wiki.arch.ethz.ch/asterix/bin/view/MAS0607/WebHome