CONTINUING
EDUCATION
CENTER

TU
VIENNA

MSc Program "Building Science & Technology"

Parametric Form Finding in Contemporary Architecture

A master's thesis submitted for the degree of
“Master of Science”

Supervisor: o.Univ.-Prof. Dipl.-Ing.Dr.phil. Georg Franck-Oberaspach
Univ.-Lect. Dipl.-Ing. Gabriel Wurzer

Vassileios Kourkoutas

Vienna, June 2007

0527086

Affidavit

I, **Vassileios Kourkoutas**, hereby declare

1. That I am the sole author of the present Master Thesis, "Parametric Form Finding in Contemporary Architecture ", 82 pages, bound, and that I have not used any source or tool other than those referenced or any other illicit aid or tool, and
2. That I have not prior to this date submitted this Master Thesis as an examination paper in any form in Austria or abroad.

Vassilis Kourkoutas

Digitally signed by Vassilis Kourkoutas
DN: cn=Vassilis Kourkoutas, o=Techniki 77, ou, email=clarkrulez@gmail.com, c=GR
Date: 2007.07.22 12:36:22 +03'00'

Vienna, July, 5th 2007

Date

Signature

ABSTRACT

The search of innovative geometry has been during the last years an interesting subject for Contemporary Architecture. As the 21st century brought a new era for architectural design, CAD programs have evolved together with the idea of Form Finding. The possibilities offered make the collaboration of the architect with the computer now possible in terms of searching the appropriate form for given cases. As the analysis of contemporary architectural pieces has indicated, the procedure of architectural design can be semi-automated. Parametric Form Finding, meaning the procedure followed offering solutions to spatial problems by using changeable variables, is transferring generative approaches into the architectural design workflow by introducing a set of rules to describe the constraints of the form.

Given this context, two methods have been realized, which are guided by the user by providing basic two dimensional shapes, restrictions and form characteristics. The approach is fitted in a plug-in for the modeling environment of Rhinoceros that generates three dimensional form based on the user’s input. The methods followed are being evaluated

Keywords: Parametric Modeling, Form Finding, Rhinoceros, plug-in

Table of Contents

List of Figures

1 INTRODUCTION

1.1 Motivation

In the search for appropriate form for the buildings, architects have always been trying to treat the appearance of their work as a personal signature (Burry, 2003). The technological evolution has offered innovative ways for treating the design process as the era of hand drawing has come to an end. Numerous three-dimensional modeling applications have replaced not only the process of designing the desired geometry but also the early design phases, the basic napkin sketches. This was a reasonable progress since human mind is capable of treating well two-dimensional issues, but when adding the third dimension, this inspiration is limited to basic primitives and formal architectural outlines. Contemporary architecture is in a continuous search of different forms as a result of the capabilities of the contemporary design process. Although the procedure does not differ from the previous state, as the same steps are being applied in order to reach the solution, the different possibilities of development start to grow as the 3d modeling packages being used these days widely by the architects are adding more features of treating the form. A new-found architectural language is being developed and is growing according to these tools. The architect is having difficulties to follow and is spending enormous amounts of time just to try different combinations of them.

The technological explosion of the 21st century, besides the new way of treating three-dimensional geometry, has to offer different possibilities as the collaboration of the architect with the computer in solving various problems. With the help of programming languages and the scripting interface that every contemporary modeling environment has to offer, automated procedures can be created in order help the user reach the solution faster and in a more efficient way. Although the computer is not able to understand aesthetics, it is capable of performing millions calculations per second and to visualize the results. By defining the design problem correctly, the program is able to handle the data and export only correct results, a procedure that would have been not only time-consuming but also hard to handle by the architect.

1.2 Claims

The following assertions are being analyzed and proved during the thesis. Every fact of the following is fundamentally important for the result and is supported by evidence in the corresponding paragraphs.

- Early architectural design can be semi-automated. (See paragraph 3.1.3)
- Work is based on rigorous analysis of existing contemporary architecture. (See paragraph 2.1 – 2.11)
- By using a set of two-dimensional shapes, complex three-dimensional form can be generated. (See paragraph 2.1 – 2.11)
- Transferring generative approaches such as L-Systems into the architectural design workflow by introducing a set of rules to describe the constraints of the form. (See paragraph 3.2 - 3.3)
- Every building can be described by shapes and their movement, rotation and scale. (See paragraph 2.1 – 2.11)
- Basic parameters such as constraints, preferences, etc can be imported to the application with the form of computer script. (See paragraph 3.2 - 3.3)
- The procedure of Form-Finding can boost up early architectural design – suggest also complex forms hard to imagine. (See paragraph 3.6)
- The design logic used in the algorithm can generate multiple equal solutions, so the user will be able to decide for the best one, as aesthetical rules cannot be programmed. (See paragraph 4.1)
- The algorithm can generate both organic and non-organic results – the final form is a function of the provided shapes. (See paragraph 3.2 - 3.3)
- The user is the one that evaluates the output, if he is not satisfied, the procedure calculates more results. (See paragraph 3.6)

1.3 Past research

Smith (Smith, 1984) presented a method to describe the structure of certain plants, originally developed by Lindenmayer (Lindenmayer, 1968). By using L-Grammars (graph-grammar language), called graftals, a new language was formed to describe various cases that have to do with graphical symbols. The language contains alphabet letters and brackets as well as a unique syntax to combine those. The language was inspired by the method of branching which is a natural procedure, found in flowers and trees. The grammar itself has no geometric content – using a grammar-based model requires both a grammar and a geometric interpretation of the language (Smith, 1984). Graftals were used to simulate the growth process, starting from tree evolution and proceeding to more advanced geometrical forms that follow similar systems.

1.3.1 Genr8

Genr8 is a plug-in for the modeling environment of Alias Maya developed by Martin Hemberg in collaboration with the Emergent Design Group at MIT in 2001. Genr8 is a Form Finding application that creates surfaces using Genetics. It is based on different concepts than traditional design logics as its core is strongly inspired from biology. It is using "Biomimetics", a method that imitates natural form, as its aesthetic and functional values are higher than the ordinary ones.

Genr8 is using local interactions to produce a global unique behavior – the algorithm is using the method of flocking and by applying certain rules (such as alignment, cohesion, separation, etc). The artificial life is based on emergent properties and the agents used basically in the algorithm follow the same direction. The evolutionary computation of form, based on randomized optimization algorithms, is inspired by natural evolution: Genetic Algorithms, Bit – Arrays, Genetic Programming, executable tree structures, Evolutionary Strategies and Floating Point Numbers (Martin Hemberg). The optimization methods used in the algorithms are trying to find the best possible solution from a wider field of solutions for a given problem. The global maximum is the best possible solution, which is being approached is a step – wise logic. The population of solutions for a specific problem adapts generation by generation. The Neo Darwinian Evolution Theory is supporting reproduction through the "survival of the fittest". Based on blind variation, each individual is being checked and ranked for its performance based on certain fitness rules. A numerical value is assigned to each member that represents its performance score among other individuals of the same population. The fitter individuals have a higher probability of selection and reproduction.

The algorithm results in surfaces that offer a decent solution to the problem given. The surface is growing within a given bounding volume following certain expansion rules.

Genr8 is a design tool that suggests multiple solutions to the problem described. As the factor “aesthetics” differs from user to user (STIN78), the program is using neural networks to learn from the user’s actions and have a clear image of his preferences for future reference.

1.3.2 Groboto

Groboto is a design tool created by Braid Art Labs and is used for 3D modeling and animation purposes. The program is used to create complex form based on repetition of basic forms. According to the manufacturers the concept of complexity is arising from simple operations repeatedly applied to simple forms. (Groboto). The idea of relativity, describing the influence of single objects on neighboring objects, is the basic idea behind the application. Every action on a single object is affecting the design of the final form in a smooth way – the user is able to use responsive tools to interact in real-time with the medium. Groboto is not parametric; the design process is not using the traditional Form – Finding methods based on complex algorithms. The form is being created based on the user's moves; certain principles are being applied to help the evolution from basic patterns. By repeating the same basic forms after transformations are being applied to them, complexity is being added to the designed forms. The workflow does not require special knowledge, as the basic terms that are used in the most common 3D modeling and animation packages are being used here.

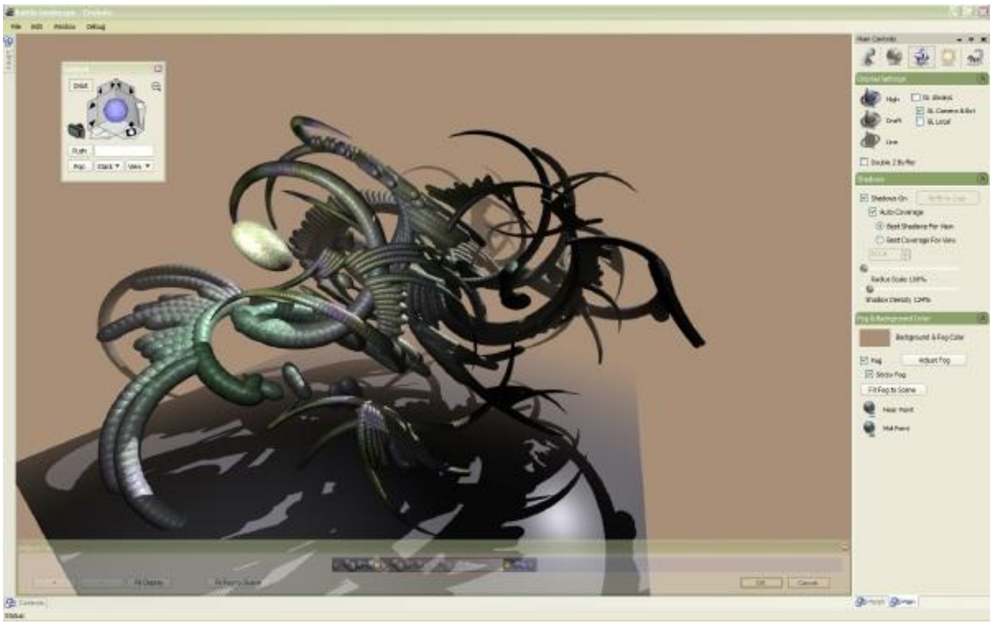


Figure 1: Main Interface of the application

Except of form creation, Groboto is also offering texture mapping, rendering and animation packages as well as an exporting tool to be able to collaborate with other modeling packages.

2 ANALYSIS

In order to detect the design methods followed by contemporary architects, an analysis of various pieces of modern architecture needed to be done. The buildings were not criticized for the aesthetical result – they were examined in a way that the abstract generic form of every building was analyzed to basic two-dimensional shapes and the transformations applied to these shapes as well as the connection between them was required. The basic transformation modifiers have been used in order to simplify the procedure.

This procedure is used widely as a design method, especially for architectural data, among others, e.g. Subdivision Modeling, which is used more for Computer Graphics, Character Modeling, etc. This method is based on repetition and the basic two-dimensional figure is copied every time before applying the transformations that are described next.

The Cartesian coordinate system is being used. The physical ground is being represented by the surface created by the X- and Y- axes, while the Z-axis is perpendicular to the surface generated by the two axes, looking up.

Basic transformation modifiers are being applied to the curves provided. These functions include movement, rotation and scaling.

The *MOVE* function is responsible for relocating elements from one fixed position in space to a new one.

The *ROTATE* function is responsible for turning elements around a rotation axis. The amounts of rotation as well as the rotation axis are being provided by the user.

The *SCALE* function is responsible for increasing or reducing the size of the elements based on scaling factors. Scaling can occur to all three axes directions or to a single direction. In these cases the element’s size is being changed unregularly, transforming it.

There are more advanced transformation modifiers, like *TWIST*, *TAPER*, etc, that are being analyzed to simpler functions until a combination of the basic transformation modifiers occurs.

2.1 Eric Owen Moss – 3505 Hayden

The 3505 Hayden project is an office space located in Culver City in California. The form of the building interacts with significant points of the environment. It consists of a three-storey building following a vertical extrusion, which is interrupted by a freeform tube that is running across the entire building. The tube is divided into three other segments, disconnected from each, one of which lands vertically on the one corner of the site (Eric Owen Moss Architects).

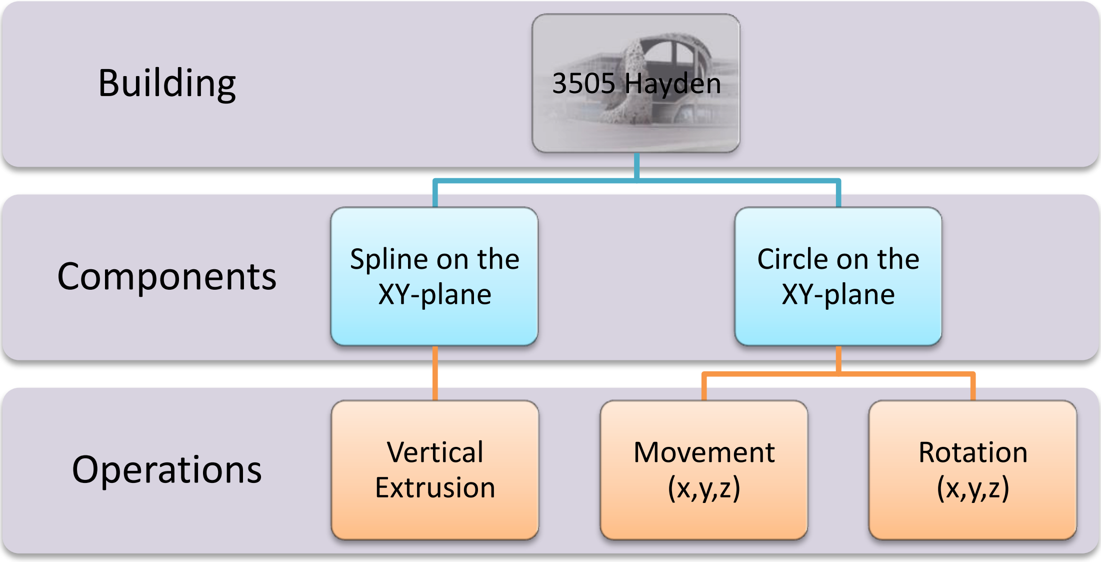


Figure 2: 3505 Hayden building analysis

The main component of the building is analyzed to a closed spline on the XY-plane, which is following the perimeter of the site. This rectangular shape is softened by introducing filleting operations to the main corner of the site (National Boulevard and Hayden Avenue), according to the entrance of the resulting building. A vertical extrusion is applied on the base shape of this component in order to transform it to a three-dimensional object. The extrusion operation can be analyzed more particularly into movement along the Z-axis of the basic shape after copying it.

The tube is analyzed into a circle on the XY-plane, which is a vertical section of the three dimensional object running across the building. Movement along the X-, Y- and Z- axes and rotation around the X-, Y- and Z- axes are applied on the starting shape. The path and the resulting end shape of the running tube are created according to the points of interest where this object should pass through (i.e. starting point, ending point).

2.2 Eric Owen Moss – Gateway Art Tower

The Gateway Art Tower is located at the prominent corner of Hayden Avenue and National Boulevard in Culver City, California. The structure is acting as an "Entrance Gateway" that welcomes the visitors for the nearby situated Conjunctive Points Theater Complex. The project is a lightweight tower of 22 m height, opposing to the massive two-storey masonry building that has replaced. The geometry of the tower consists of 5 conical segments of 3 – 3.5 m height that are held by repeated steel rings. The base of the Gateway Art Tower is excavated and is being used as an external amphitheater. (Eric Owen Moss Architects)

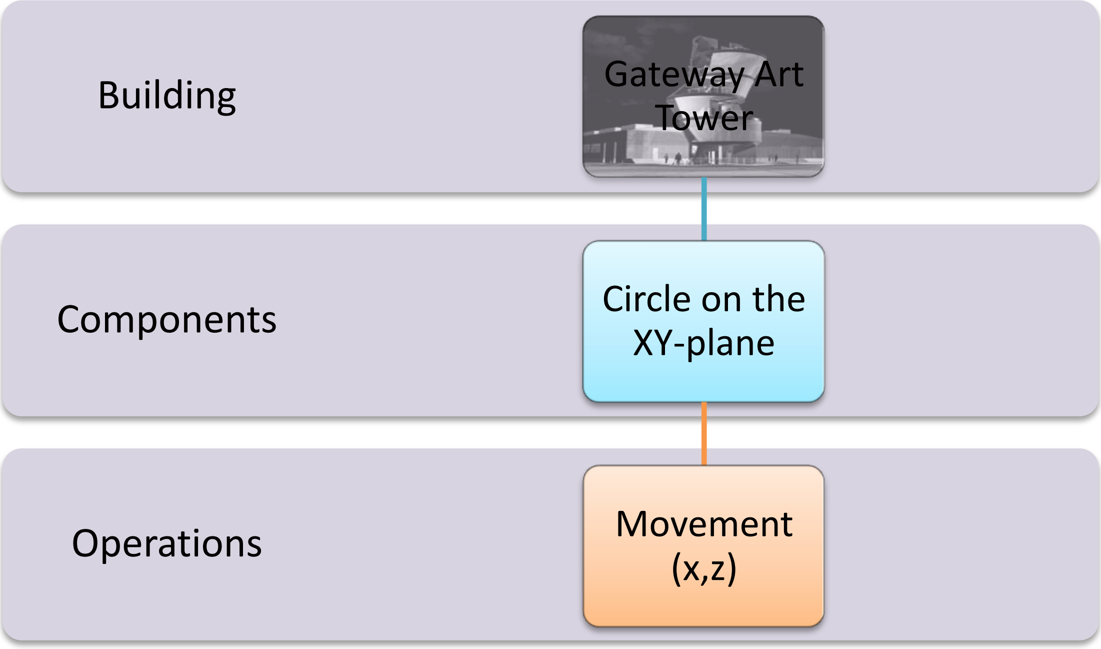


Figure 3 : Gateway Art Tower building analysis

The building is analyzed into one single component, a circle on the XY-plane, which is repeated several times along the Z-axis. The starting shape is being moved along the x axis in order to provide the slanted design of the building. This procedure takes place 5 times and every circle represents a storey of the tower. The shapes are being connected with straight lofting techniques.

As every iteration of the algorithm represents the perimeter of one storey, and the real building is a lightweight structure made out of glass that is being held by a steel structure, it could be said that for such cases every iteration of the design algorithm not only represents one storey, but also the structural form of this storey. The structural design logic is hidden behind the architectural design as every decision is supported by the structural judgment of the building.

2.3 Eric Owen Moss – Conjunctive Points Theater Complex

The Conjunctive Points Theater Complex is a multiplex that contains innovative theater rooms in addition to office, restaurant and retail space. The building maintains a constant cross section along its development, although it has been bent out of shape in a way to retain minimum footprint because of the limitations. Its main facade is facing south in order to maximize the southern exposure. The bending of the starting geometry was realized after taking into consideration the utilization of solar radiation and the inclination that a unique state-of-the-art theater should have. The structural system consists of orthogonal concrete frames along the main axis of the building and bent steel pipe frames that are used to hold them together.

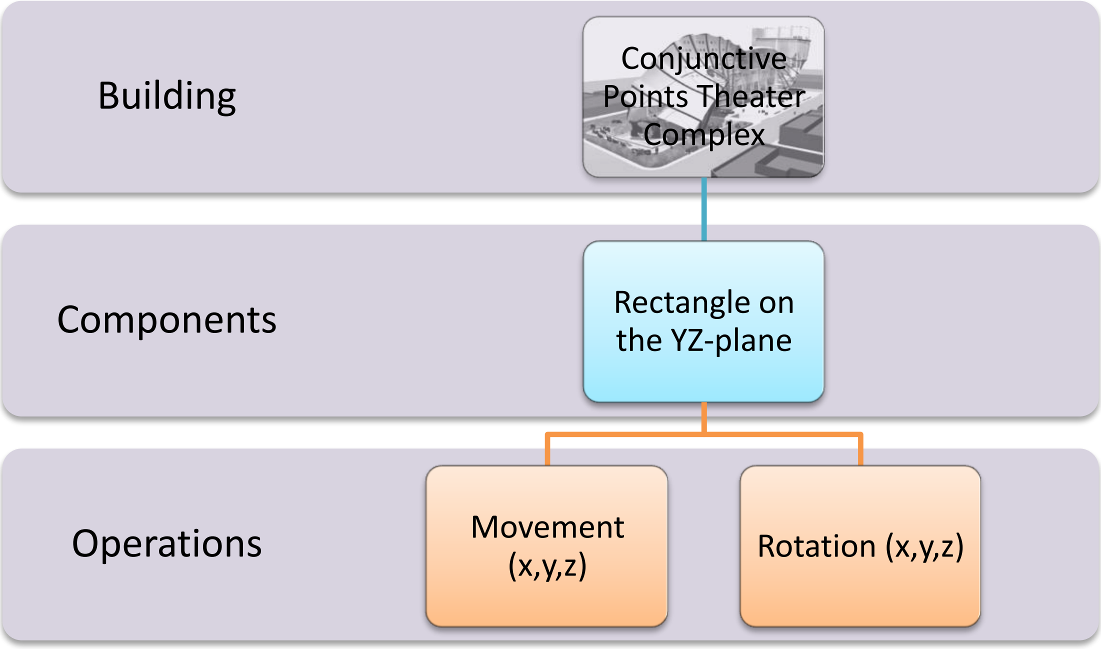


Figure 4 : Conjunctive Points Theater Complex building analysis

The Conjunctive Points Theater Complex building is analyzed into a 30x30m rectangle which is repeated and rotated along a main axis. This main axis is deformed resulting to a flow-along-curve technique. The operation can be more simplified by moving and rotating the starting rectangle on the three axes. By lofting the repeated shapes, the abstract form of the building is generated. In this example, the iterations behind the design idea represent also the structural system. Each rectangle, after the transformations applied to it, represents one of the sectional concrete frames.

2.4 Frank O Gehry – Nationale Nederlanden

The building is located in the historic center of Prague, Czech Republic and it was created by Frank O Gehry in cooperation with Studio Vlado Milunic in the years 1992 – 1996 and is the main office building of the insurance company Nationale Nederlanden. Although the regulations for the area are strict for new designs, Gehry created an object that is expressing fluidity and coheres to the environment (Guggenheim.ORG). Its role is also to connect the public square and the major bridge that constitute significant elements for the city. The two towers that compose the Nationale Nederlanden building have been likened to the dancers Fred Astaire and Ginger Rogers because of their movement.

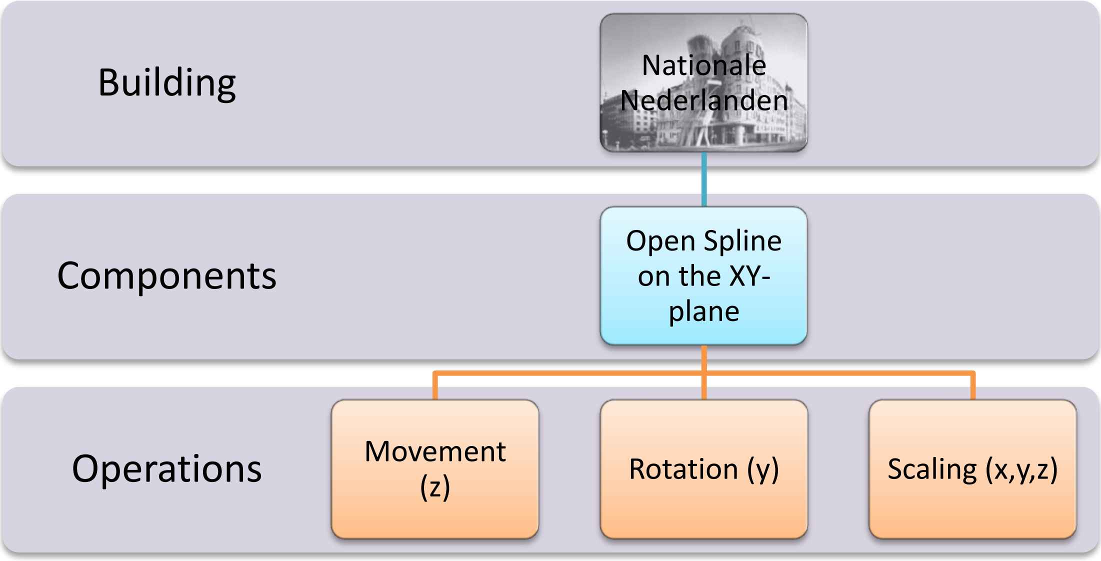


Figure 5 : Nationale Nederlanden building analysis

The design of the two towers is based on an open spline on the XY-plane which is repeated along the z axis. The transformations applied on the starting shape are movement along the Z- axis, rotation around the Y- axis and scaling along the X-, Y- and Z- axes. As the movement of the starting shape is responsible for the repetition line that is followed, the rotation around the Y-axis and the scaling pilots the "fluidity" of the design. The rotation leads to a Gotham-City-like unit, where inclined rounded walls offer the desired movement to the building. The final geometry is attained in three steps, where every iterations is responsible for the direction's changes of the exterior walls. Although every iteration represents also structural elements, further segmentation of the geometry could lead to a more detailed structural solution.

2.5 Gregg Lynn – H_2 House

The H_2 House is a multifunctional exhibition center that was designed for the Austrian Mineral Oil Processing company OMV. The building located in Schwechat, Vienna, Austria is pointing to the visitors of the facilities of the company as an educational resource. Innovative materials have been use in order to offer the possibility for the geometry to change. The aqueous aesthetic of the building is realized by introducing transparency to the exterior walls, that reveals the energy system and function of the building to the exterior. The design of the H_2 House is based on the environmental conditions and it is able to activate mechanism that will be responsible for maintaining comfort of the interior (Architekturzentrum Wien).

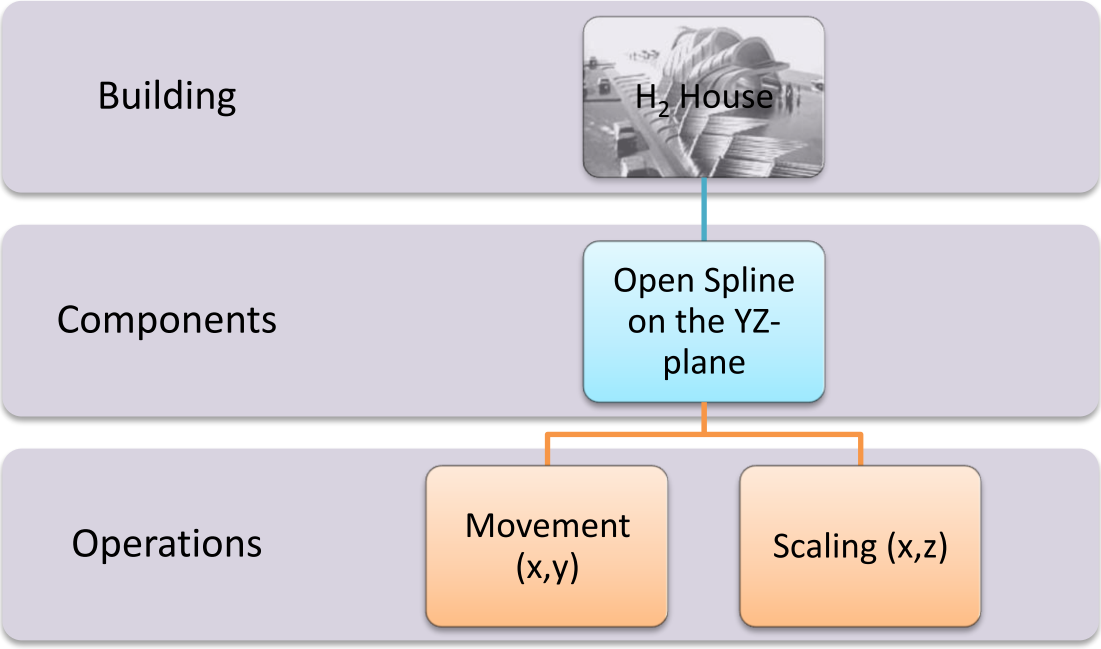


Figure 6 : H2 House building analysis

The H_2 House is deconstructed into an open spline on the YZ-plane which is repeated along the Y-axis. In order to add flexibility to the design, there should not be major changes along the development, the cross sectional should remain transformatted slightly. As the movement along the X- Y- axes results to the main design pattern, the separate scaling along the X- and Z- axes is responsible for the variations of the starting shape. In every two iterations of the design phase, lofting techniques are being used to connect the elements and to create a shelter-like design that borders the interior. The skin manipulation of the exterior walls and ceilings offer the desired flexibility that is attained through the special materials applied. Every iteration represents also a frame of the structural system of the building.

2.6 Norman Foster – London City Hall

The London City Hall, created from 1998 to 2003, is located in London, UK and is a space designed for the London's local government (The Great Buildings Collection).The building's geometry was a result of various factors, including environmental, by minimizing the total exposure of the façade to the solar radiation, and democratic in a metaphoric way, as the movement inside the building is realized through a spiral ramp, giving the opportunity to the citizens to walk "around their representatives" (Galinsky).The London City Hall is designed to use only one quarter of the typical annual energy being used by a typical air-conditioned office building. This is achieved by introducing the iconic-sphere like geometry that minimizes the perimeter of the building, which is responsible for heat losses and heat gains.

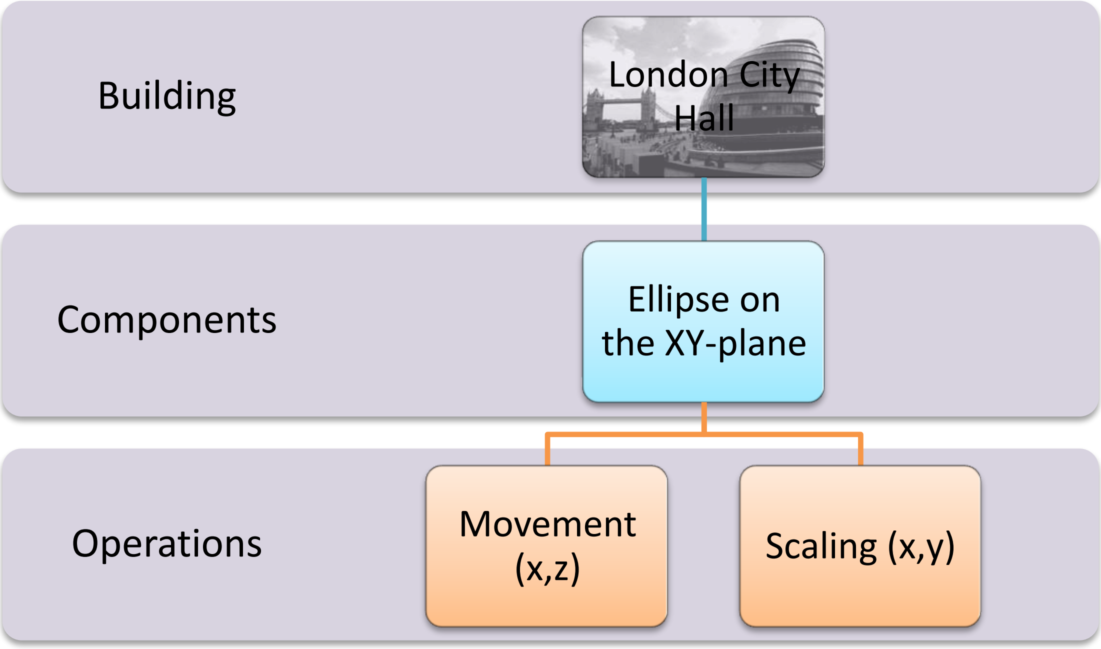


Figure 7 : London City Hall building analysis

The building is deconstructed into a single ellipse on the XY-plane, which is repeated along the Z-axis. The basic transformations applied to the starting two-dimensional shape are movement and scaling along the X- and Z- axes. Although the movement of the shape is producing the nine-storey building, the scaling by keeping a constant point on the ellipse creates the desired inclination that is responsible for the basic environmental issues. This design offers maximum performance, indicated by studies and other simulation methods. Every iteration of the design technique represents structural elements; in this case every ring represents a floor element.

2.7 Norman Foster – Bilbao Metro Entrance

The Bilbao Metro Entrance is a part of the Metro project in Bilbao designed by Norman Foster + Partners in the years 1988 – 1995. The task was conceived as a totality – architectural, engineering and construction skills were put together in order to create a unique space that mirrors the era of the 21st century. The dramatic curved design that is being followed inside and outside of the metro station offers various possibilities for future changes. As the metro entrance follows the same design pattern as the interior, it stands as a connection between Foster's world and the neutral, time-changing environment of the outside. The oppressive feeling of the subway has been substituted by the idea of space travelling and the entrance to the station references some futuristic design (Foster + Partners).

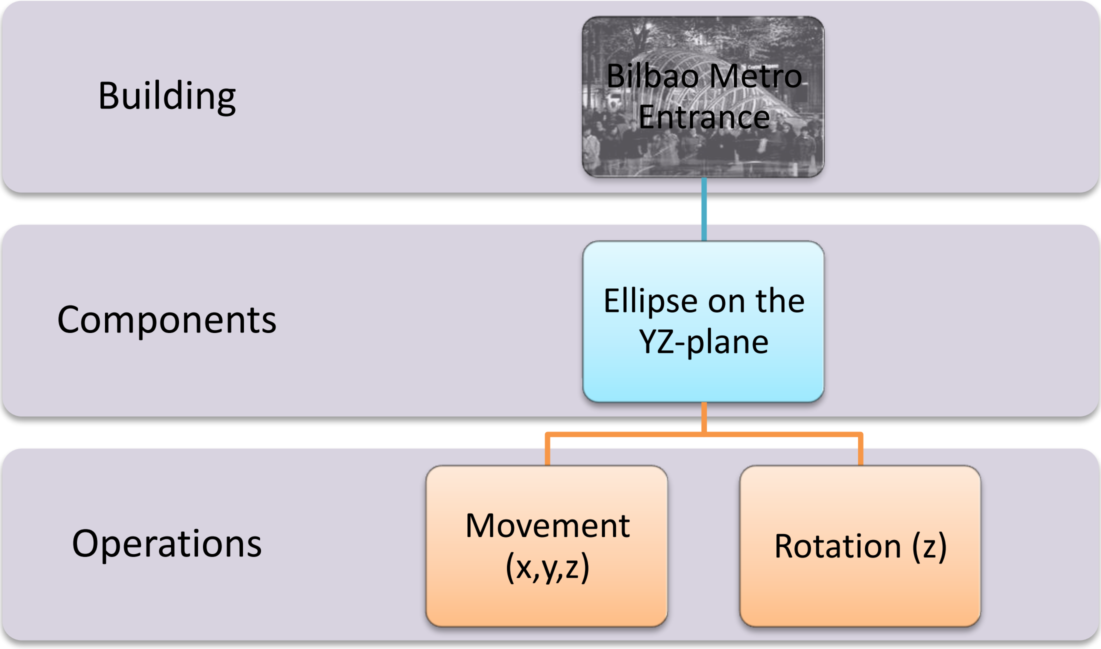


Figure 8 : Bilbao Metro Entrance analysis

The main design pattern of the Bilbao Metro Entrance is based on an open curve that is being copied, moved and rotated around the Z-Axis. Although the movement function forces the starting shape to move along constant paths, the rotation function forces the shape to rotate for different angles to achieve smooth transitions between the ground and the storey levels. The result of such a function is a wireframe that refers to the structural system of the construction. By lofting the resulting shapes using normal loft, the final geometry of the construction is given. Every edge of the designed geometry represents a piece of the steel structure that holds the construction and the surfaces created after lofting refer to the various glass parts.

2.8 Peter Eisenman – Max Reinhardt House

The Max Reinhardt House is a single-family house designed in 1992 by Peter Eisenman. The project is located in Berlin and has never been built in reality. The building is based on the Moebius curve as the architect tried to achieve continuity of its basic surfaces. Eisenman pioneered the Moebius Form by roughly translating it into the Max Reinhard House building (Eisenman, 1999). The building is being sliced at its section with the physical ground, thus failing to achieve visual continuity of the Moebius as a whole (Krawczyk). The main functions of the house were split to the two longer parts of the curve and the continuity offered within the house had a major influence to the placement of the rooms.

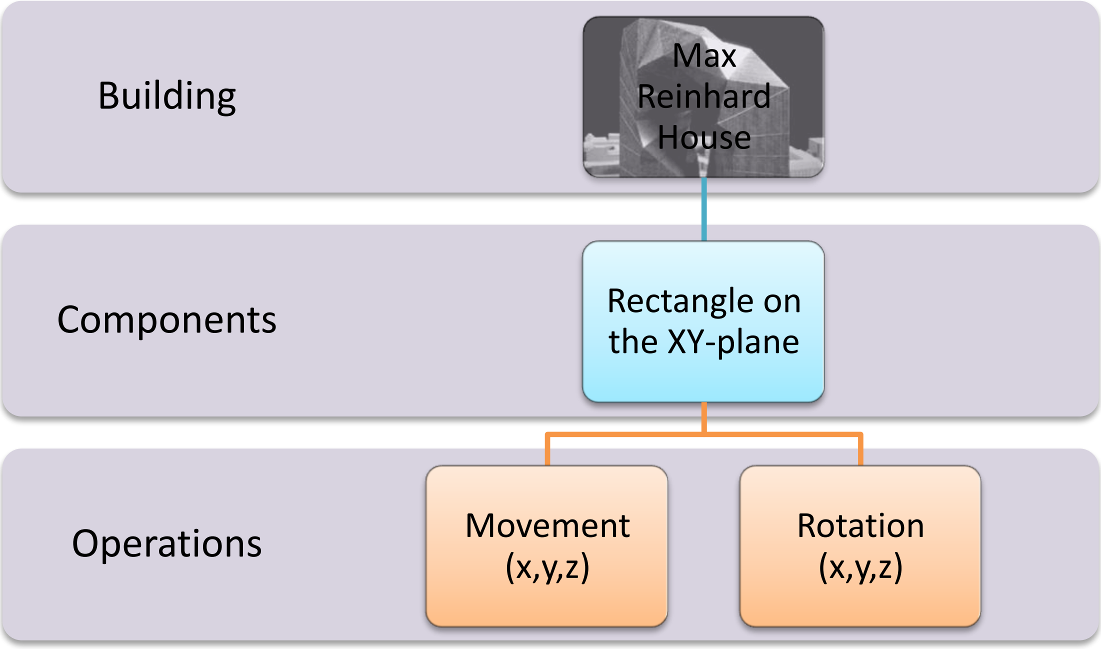


Figure 9 : Max Reinhard House building analysis

The building is analyzed into a basic rectangle that is being coped every time along the building axis, following movement and rotation as the Moebius Form dictates. The movement distances are constant – the rotation angle differs to create a stronger visual effect. The basic axis of the building is not the well known Cartesian axis, but an ellipse; the transformations are being executed along it.

The result is being lofted using straight section lofting. The basic rectangle that is being repeated along the building represents a frame that consists the basic wireframe of the house. The skinning is being realized with lightweight materials.

2.9 Rem Koolhaas – Seattle Public Library

The eight-storey Public Library was created by Rem Koolhaas and Office for Metropolitan Architecture (OMA) in 1999. The project located between the 4th and 5th street in Seattle was finished in 2004. The building consists of eight horizontal layers of different size that are connected to diverse functions of the library. A structural glass-and-steel skin smoothes the multi-faceted form. Although the exterior of the building seems more static, the interior space organization indicates continuous movement as the four-storey ramp allows people to browse through books and other elements in a continuous sequence. (arcspace.com)

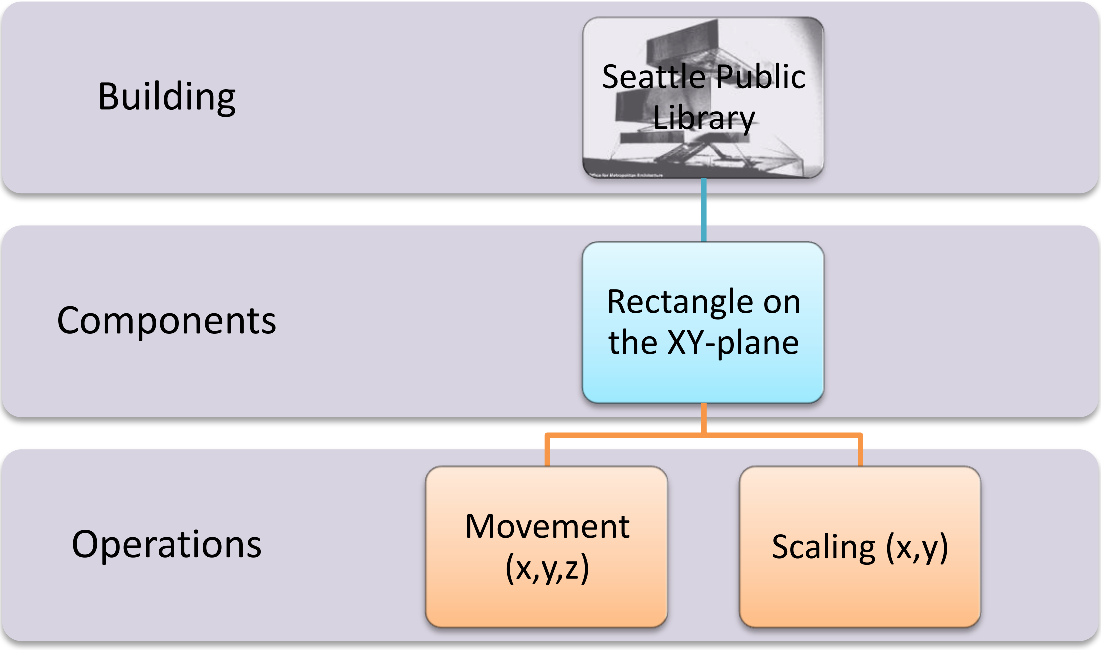


Figure 10 : Seattle Public Library building analysis

The main design pattern of the Seattle Public Library is a rectangle that is being copied, moved and scaled in order to fulfill the volume expectations. Although the vertical evolution along the Z-axis of the two-dimensional shapes determines the storey height according to the step followed, the Seattle Public Library uses different storey heights, which are multiples of a common number.

The resulting shapes are being lofted using straight loft. The horizontal edges of the resulting mesh indicate the top and bottom of every storey – the optimized structural system of the building is represented through the wireframe of the mesh. The surfaces created by lofting represent the glass-and-steel skin that covers the steel wireframe and comprises the basic ingredient for presenting the unique form.

2.10 Zaha Hadid – Contemporary Arts Center

The Contemporary Arts Center was designed in 1999 by Zaha Hadid Architects and lies in Rome, Italy. The project is characterized from a reaction with the physical ground; an approach of blurring the limit between real and artificial is visible. Its design makes it act as a 'second skin' for the area, as it emphasizes its limits at the same time. The Arts Center consists of spaces that flow freely between interior and exterior and different levels, as the ramp-like hallways where the exhibitions take lead the guest to different hotspots of the building without noticing.

The distinctive continuous walls of the project have several roles except of acting as a structural element or as a background for the exhibition paintings. The solidness of the walls is being interrupted by long glass surfaces (window to the city) or even big glass parts. The walls become floors and ceilings, they transform to sloped surfaces along the evolution of the linear spaces. (DesignBoom)

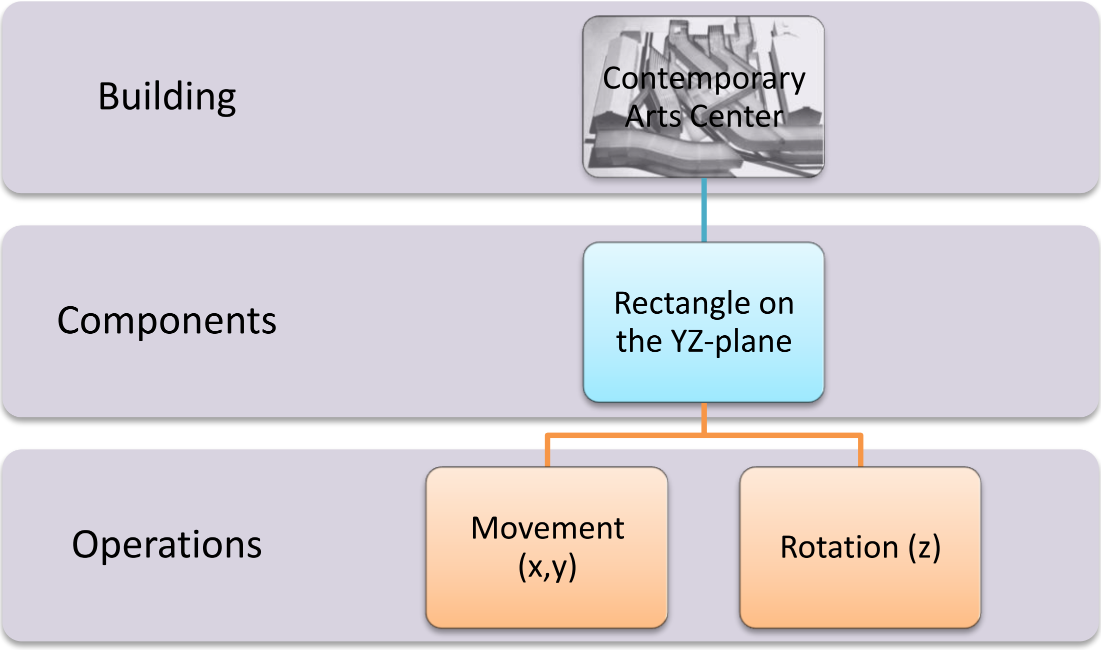


Figure 11 : Contemporary Arts Center building analysis

The Contemporary Arts Center is analyzed into a single rectangular shape XZ-plane that is being copied, moved and rotated. Every linear shape follows a different unique handling, but the design pattern remains the same. The resulting shapes are connected by using loose lofting techniques to maintain smooth transitions between the path lines. If more starting shapes are being used before the loft function, the selection of shapes determines the cross sectional changes. By shearing the starting rectangle along the linear evolution path, the walls get slanted. The different generated rectangles along the path represent steel frames that form the basic structural system.

2.11 Zaha Hadid – Twin Towers Redux

The Twin Towers Redux is a proposal by Zaha Hadid Architects, answering to the call for the replacement of the World Trade Center Towers after the 9/11. The new proposal consists of two pairs of two towers that are higher and more complex than the original towers of the previous state. The four towers have different uses, the thinner pair for residential use and the thicker one for offices. The typical form of a skyscraper is substituted by the new proposal; additional spaces are being created as the thought behind the main design idea is to connect people rather than putting them apart on different levels. The skyscrapers bend and twist in order to fit the program needs and as tribute both to the dead as to the World Trade Center, hollow tubes have been cut into the footprints of the towers.

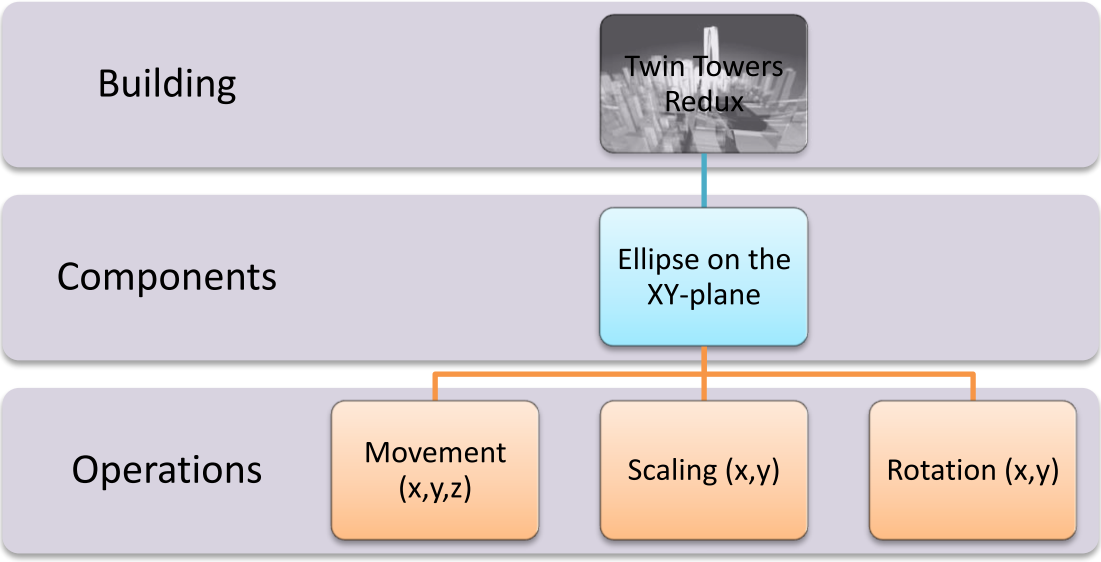


Figure 12Twin Towers Redux building analysis

The Twin Towers Redux is analyzed into four different parts. Every tower is represented with a different starting shape, an ellipse on the XY-plane. The ellipse is being copied, moved, scaled and rotated along the X- and Y- axes. The effects of bending and twisting occur by using rotation around the Z-axis for twisting and around the Y-axis for bending.

The structural system is difficult to be represented for this example because of the amorphous design and the special structural handling that such a building should have.

3 APPROACH

3.1 Basic Definition

3.1.1 Two Dimensional Shapes

1) Spline:

A smooth curve that runs through a series of given points. The term is often used to refer to any curve, because long before computers, a spline was a flat, pliable strip of wood or metal that was bent into a desired shape for drawing curves on paper. (Wolfram Mathworld)

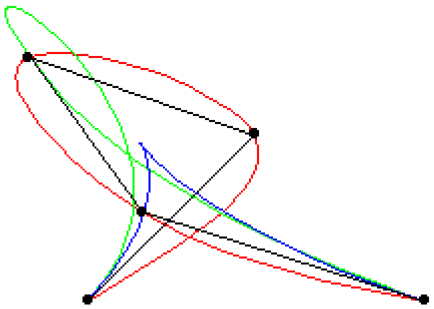


Figure 13: Typical examples of Splines

2) Bezier:

A curve that is generated using a mathematical formula that assures continuity with other Bezier curves. It is mathematically simpler, but more difficult to blend than a b-spline curve. Within CAD and drawing programs, Bezier curves are typically reshaped by moving the handles that appear off of the curve. (Wolfram Mathworld)

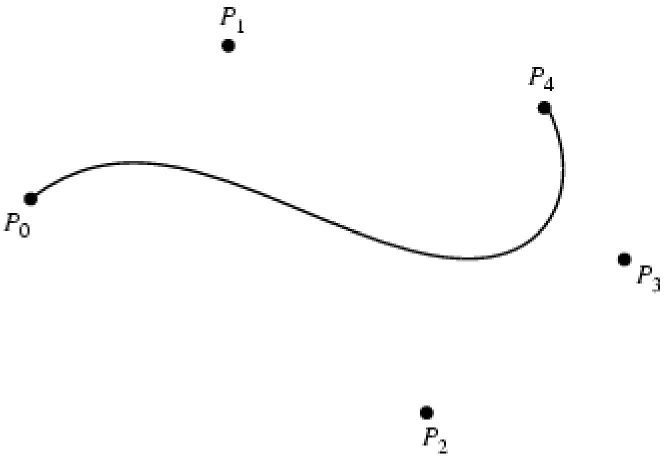


Figure 14: A typical Bezier Curve with the corresponding Control Points

3) *B-spline:*

A curve that is generated using a mathematical formula that assures continuity with other b-splines. (Wolfram Mathworld)

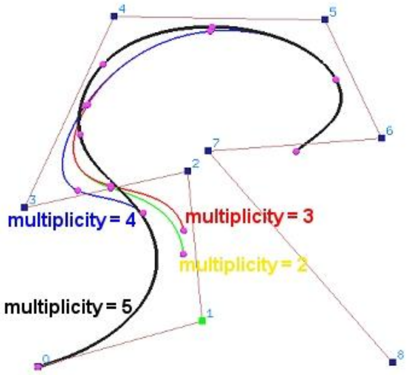


Figure 15: Typical examples of B-Splines

4) *Linear Spline:*

A B-Spline of degree 1, where degree is the degree of a polynomial (maximum of the degrees of all terms in the polynomial)

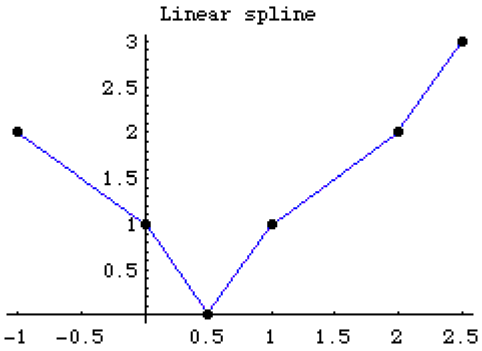


Figure 16: A typical Linear Spline

5) *Nurbs:*

(**N**on **U**niform **R**ational **B**-spline) A type of b-spline that is very flexible. NURB curves can represent any shape from a straight line to a circle or ellipse with very little data. They can also be used for guiding animation paths, for approximating data and for controlling the shapes of 3D surfaces. NURBS are known for their ability to control the smoothness of a curve. (Wolfram Mathworld)

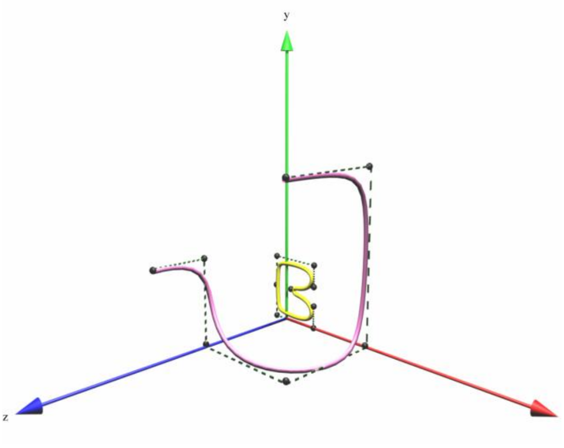


Figure 17: Typical examples of NURBS Curves

3.1.2 Skinning

Lofting is a skinning technique achieved by using NURBS surfaces to create a surface fit through selected profile curves that define the surface shape.

Other types of skinning are:

Sweep:

Creates a surface through profile curves that define the surface shape and one curve that defines a surface edge. (Foley, 1990), (Watt, 1993)

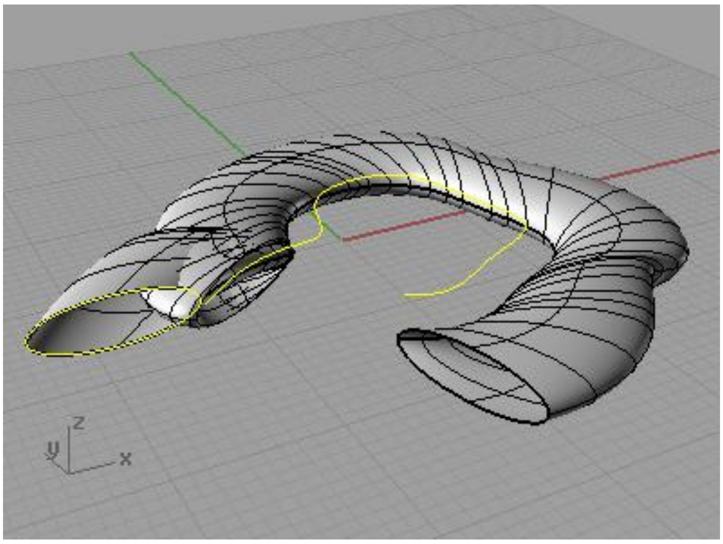


Figure 18: Example of the Sweep command

Extrude:

Creates a surface or solid by driving a curve in a straight line perpendicular to the construction plane.

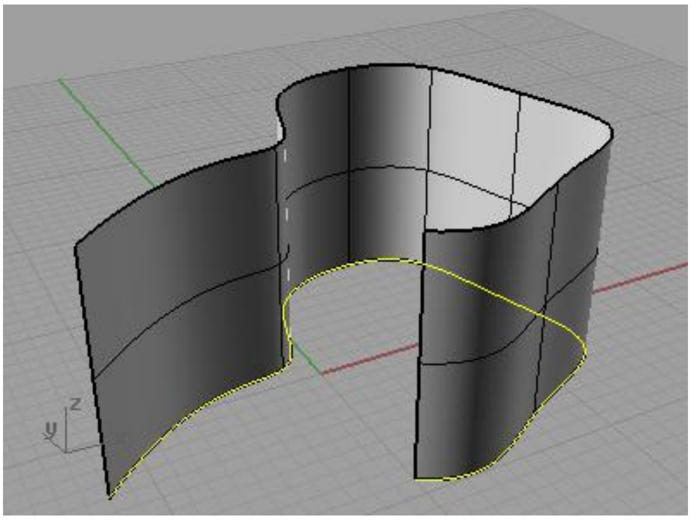


Figure 19: Example of the Extrude command

Revolve:

Creates a surface by revolving a profile curve that defines the surface shape around an axis.

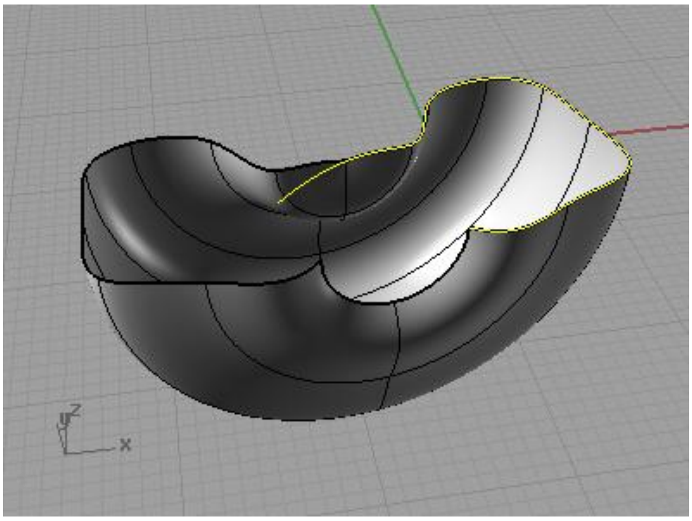


Figure 20: Example of the Revolve command

Drape:

Creates a surface through points defined at the intersection of objects and points projected toward the construction plane in the current viewport.

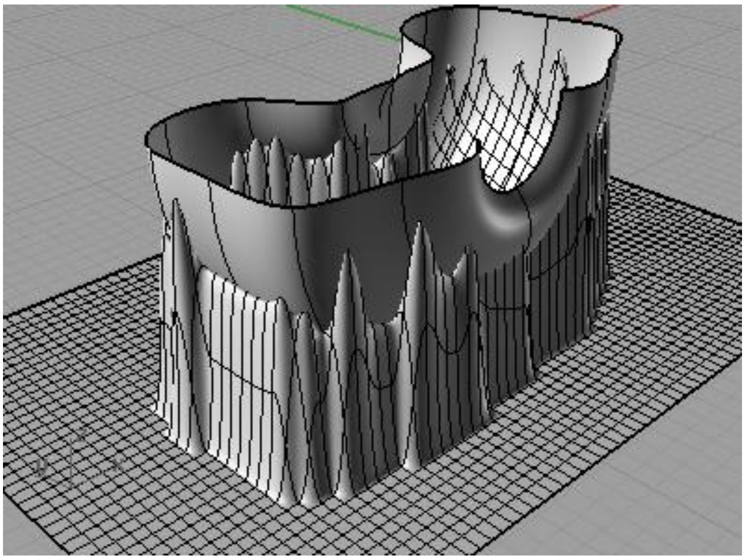


Figure 21: Example of the Drape command

3.1.3 Why use NURBS to represent 3 D geometry?

NURBS geometry has important qualities that make it a good choice for computer aided modeling.

- There are several industry standard ways to exchange NURBS geometry. Different modeling, rendering, animation, and engineering application can collaborate using the NURBS standard. Information can be stored that will be used after the completion of the project.
- NURBS can accurately represent both standard geometric objects and free-form geometry. (Watt, 1993)
- The amount of information required for a NURBS representation of a piece of geometry is much smaller than the amount of information required by polygon data. This gives an important head-on to the method that makes it ideal for applications such as architectural design.
- Remarkable standardization work for Architecture and Engineering has been done until today, in order to be able to exchange NURBS geometry. (IFC Standard)

The analysis of contemporary architectural pieces indicates that form can be described as a function of two-dimensional curves. By introducing variables that stand for design decisions and user's limitations, the procedure can be semi-automated.

In order to solve the non-deterministic problem of creating three-dimensional form that fits the design needs and follows the user's limitations based on two dimensional shapes, several techniques have been used during the thesis. All of these techniques are approaches whose result is always based on the restrictions provided by the user before the form creation and are using a step-wise logic based on repetition: the given shapes are being copied along with various operations being applied on them attempting to form building elements. By changing parameters, such as total height, number of storeys, maximum horizontal displacement, etc, generated form can be shifted to different architectural styles following particular design logics.

The following approaches were used in the thesis:

- Parametric Form Finding using Transformation Modifiers (see paragraph 3.2)
- Parametric Form Finding using Stochasing Choice (see paragraph 3.3)
- Architectural Constraints and Rules (see paragraph 3.6)

3.2 Parametric Form Finding using Transformation Modifiers - Overview

The technique of parametric Form-Finding is approaching the problem based on numerous operations applied to a two-dimensional curve. These operations are transformations whose influence can be limited based on the user's limitations and restrictions.

The analysis of contemporary architectural pieces indicates that building design can be based on design decisions and form can be analyzed to two-dimensional curves and operations connecting these curves. This connection procedure of as well as the algorithm generating these curves is being analyzed in this approach. The opposite procedure of the analytical form breaking down would be a good starting point.

The form-finding is divided into two basic problems: the choice of a set of succeeding shapes and the connecting of the shapes using a skinning operation such as lofting.

At the beginning of the algorithm the curve that defines the building plot boundaries is needed. The program recognizes this as the "area allowed to be used for the building", the so called "Site Curve". Subsequently, the program requires a new numeric value by the user, the total height allowed to be built. This value is usually given by the specifications of the building plot and depends on the law, on the area or the boundary conditions. This value is unique for every case and stands as a limit for the height of the resulting building. Every solution generated by the application will be limited to the height provided at this step. By combining the Site Curve with the Total Height allowed, the Bounding Volume is automatically created.

The program is using the concept of stores to build up form. After having created the bounding volume, it requires a numeric input from the user to define the storey height. By dividing the total height with the story height, the number of storeys results. Similar to the total bounding volume, smaller storey bounding boxes are being created that offer more flexibility to the process. They can be characterized as the "Expansion Volume" that every storey can take along the generative process.

After having set these conditions, the algorithm is ready to start creating three-dimensional form. The procedure is based on repetition. Common transformation modifiers such as *MOVE*, *ROTATE* and *SCALE* are being used after the initial curve object is being copied. The amounts of transformations applied are based on the randomizer of the internal engine of the CAD program, which generates random values within desired numeric fields. The user is able to limit these operations by defining the desired fields to lead the result to more definite design paths.

The algorithm is using step-wise methods to create the skeleton of the forthcoming building. It is using iterations to reach the total height allowed by starting from the ground level and using growing techniques similar to plant architecture. At every iteration the previous curve is being selected, and after being copied, transformations are being applied to it. The amounts of transformation (translation, rotation, scaling) are determined randomly, but within sensible limits. This preserves harmony and proportion for the result as it restricts the growth, so that dramatic changes in movement, orientation and size do not occur.

After the skeleton is being created, skinning techniques such as lofting are being used to connect the generated curves. Different types of lofting are available for the user to choose, although a random selection is also available.

The result, after being lofted, is being checked for its fitness according to the bounding volume created earlier. A check is done whether or not the volume of the generated story is within the bounding volume. If the result is not accepted, the procedure is started from the beginning by keeping the same data and preferences, until a valid solution is given.

Surface and volume characteristics can be calculated at this point, such as surface curvature and volume of the created storey or of the whole object. These values are being compared to the set preferences of the user, and act as further restrictions on the form generation algorithm. As a matter of fact, the design process can become stricter by limiting more the results.

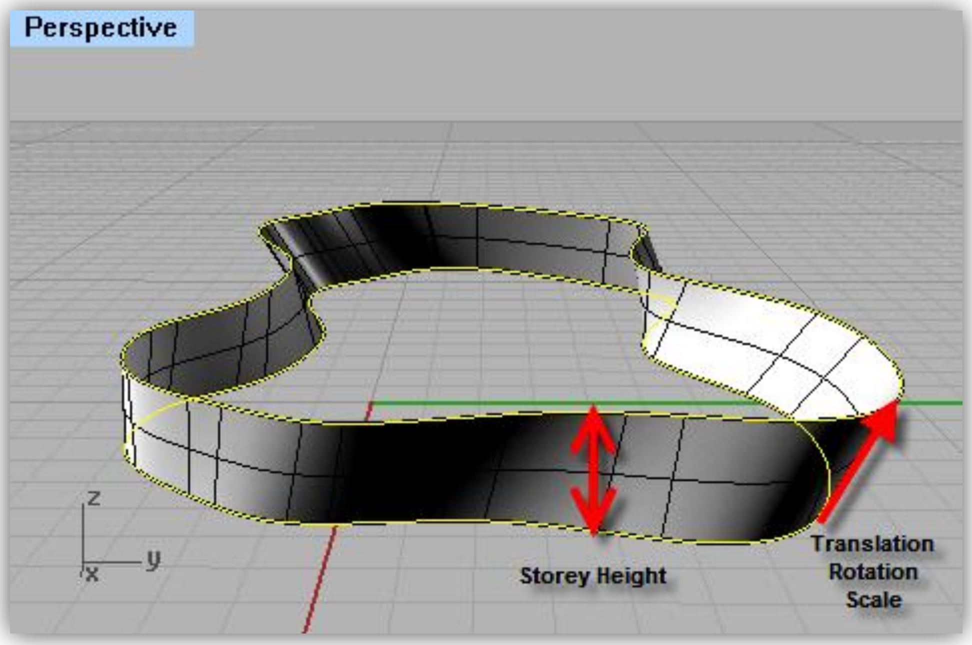
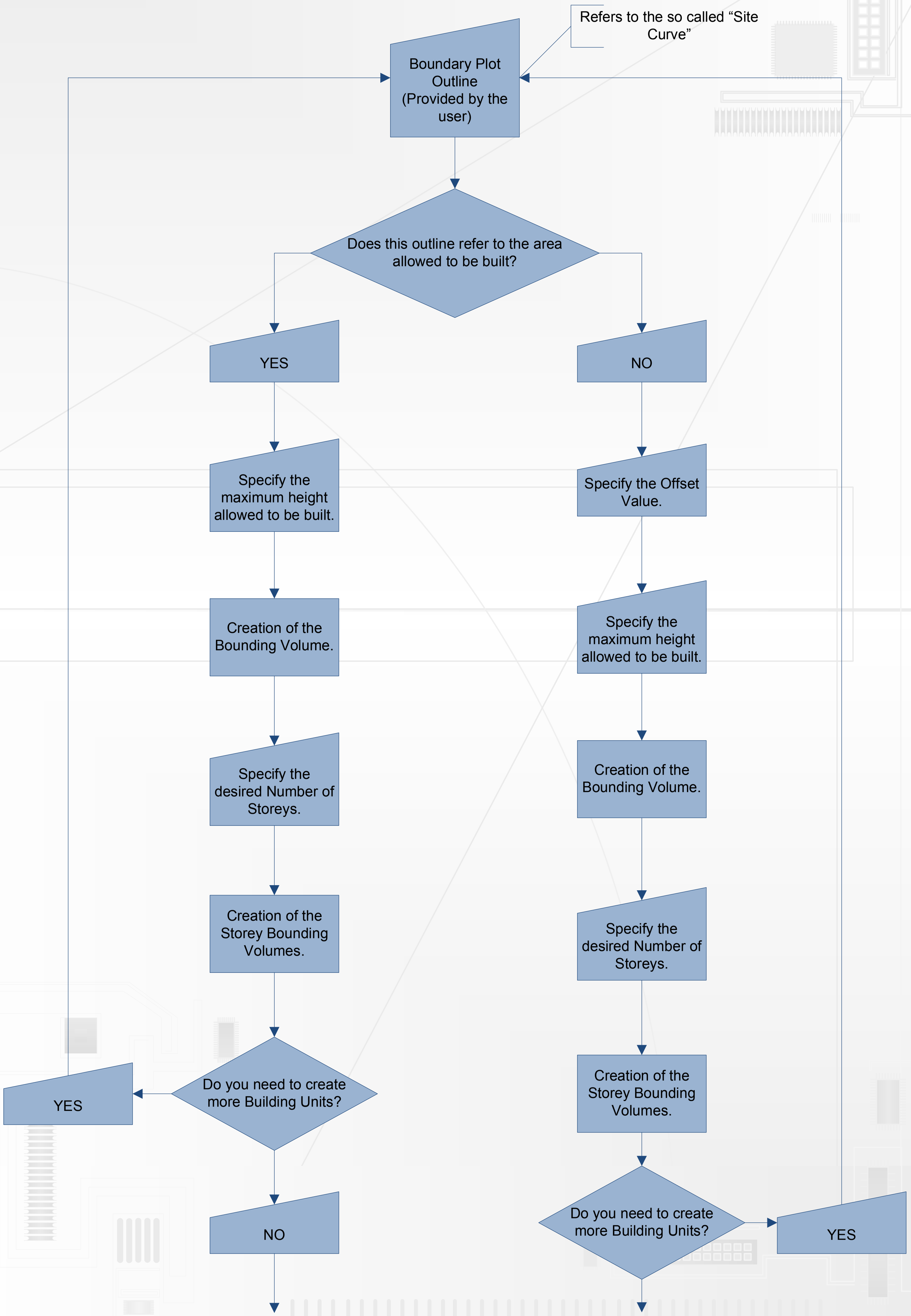
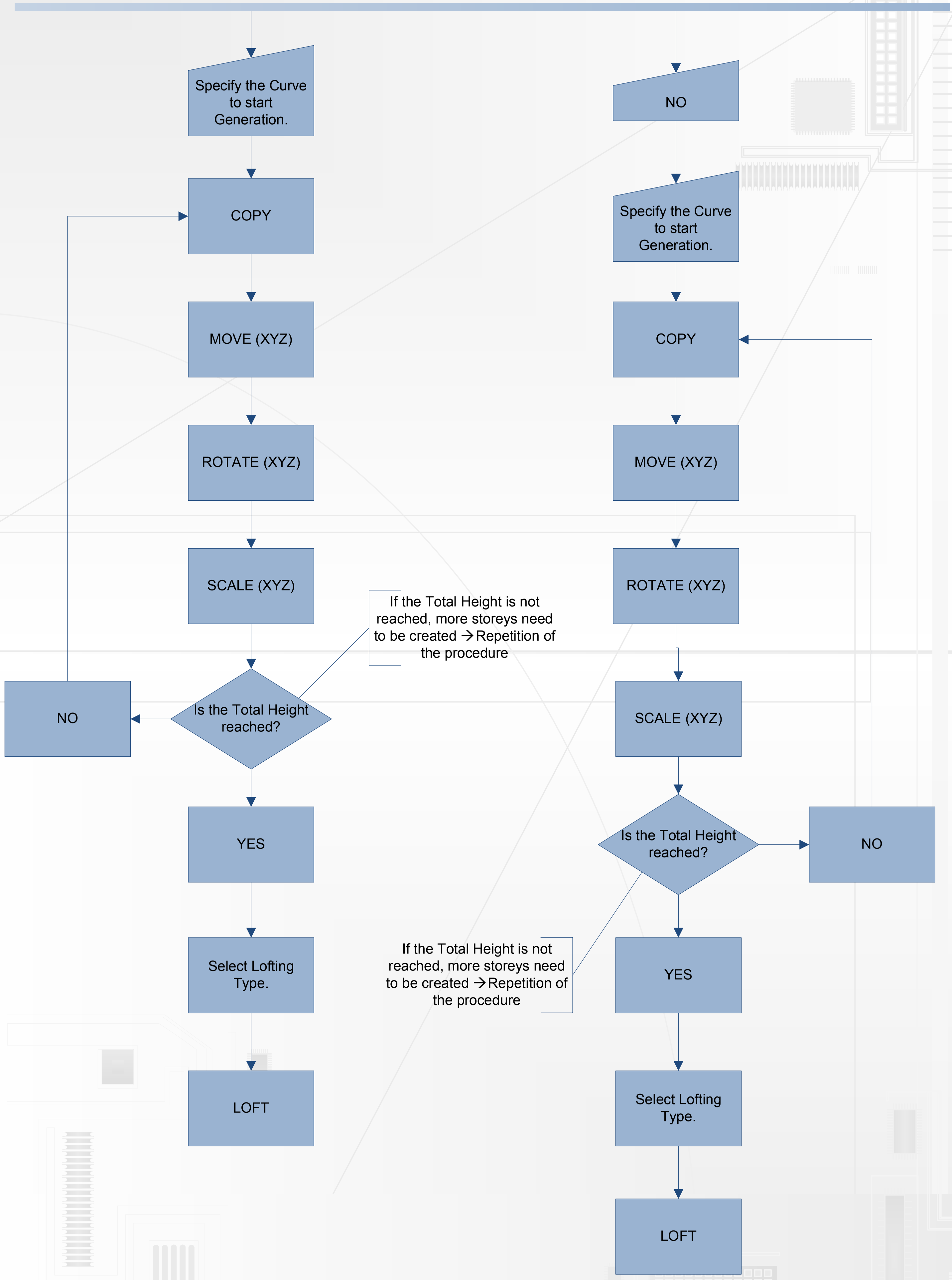


Figure 22: Screenshot showing two sequential curves and the lofting technique

Parametric Form Finding using Transformation Modifiers


Refers to the so called “Site Curve”
Boundary Plot Outline (Provided by the user)
Does this outline refer to the area allowed to be built?
YES
Specify the maximum height allowed to be built.
Creation of the Bounding Volume.
Specify the desired Number of Storeys.
Creation of the Storey Bounding Volumes.
Do you need to create more Building Units?
YES
NO
NO
Specify the Offset Value.
Specify the maximum height allowed to be built.
Creation of the Bounding Volume.
Specify the desired Number of Storeys.
Creation of the Storey Bounding Volumes.
Do you need to create more Building Units?
YES

Parametric Form Finding using Transformation Modifiers


Specify the Curve to start Generation.
COPY
MOVE (XYZ)
ROTATE (XYZ)
SCALE (XYZ)
Is the Total Height reached?
NO
YES
Select Lofting Type.
LOFT
If the Total Height is not reached, more storeys need to be created → Repetition of the procedure
NO
Specify the Curve to start Generation.
COPY
MOVE (XYZ)
ROTATE (XYZ)
SCALE (XYZ)
Is the Total Height reached?
NO
YES
If the Total Height is not reached, more storeys need to be created → Repetition of the procedure
Select Lofting Type.
LOFT

3.3 Parametric Form Finding using Stochastic Choice - Overview

The method of parametric form finding using stochastic choice is using a different approach to the problem of generating form starting from basic curve objects given by the user. The method is called parametric as the design decisions taken are based on parameters set by the user and the result generated depends a lot from these parameters.

As in the previous example, parametric form finding using transformation modifiers, the algorithm used requires at its beginning various input data, both numeric and object-based, from the user. This data frames the parameters of the design, which are in fact restrictions and requirements for the program to use along its continuation. The algorithm requires from the user to select the Site Curve. As in the previous example, the term Site Curve is used for the area allowed to be used for the building, not necessarily for the outline of the real building plot.

The next step is for the user to provide characteristics for the forthcoming building such as total height allowed and desired number of storeys. Both of these values are being put into variables as they will be needed later for the generative algorithm.

After the preferences have been set, the bounding volume is being created by combining the Site Curve and the total building height allowed, both provided by the user. The characteristics of the bounding volume are described explicitly in the previous section.

The program requires from the user to provide the starting curve that lies within the boundary limit. The curve objet has to be drawn from before or should be imported from a different CAD program. After the curve object is selected, the user has to clarify whether this will be the only curve used or he desires to select additional objects that will be used during the generation process. Although the algorithm is using the basic curve as a starting point for the growth, if the user decides to use more than one shape, they will be chosen randomly using the internal randomizer.

After all the curves that are going to be used in the process have been selected, the program tries to distribute them randomly inside the bounding volume. The random distribution of curve objects is based on repetition and is using the objects provided after copying, moving, rotating and scaling them in space. The user needs to set a numeric value which describes how many times a curve should be used inside the bounding volume.

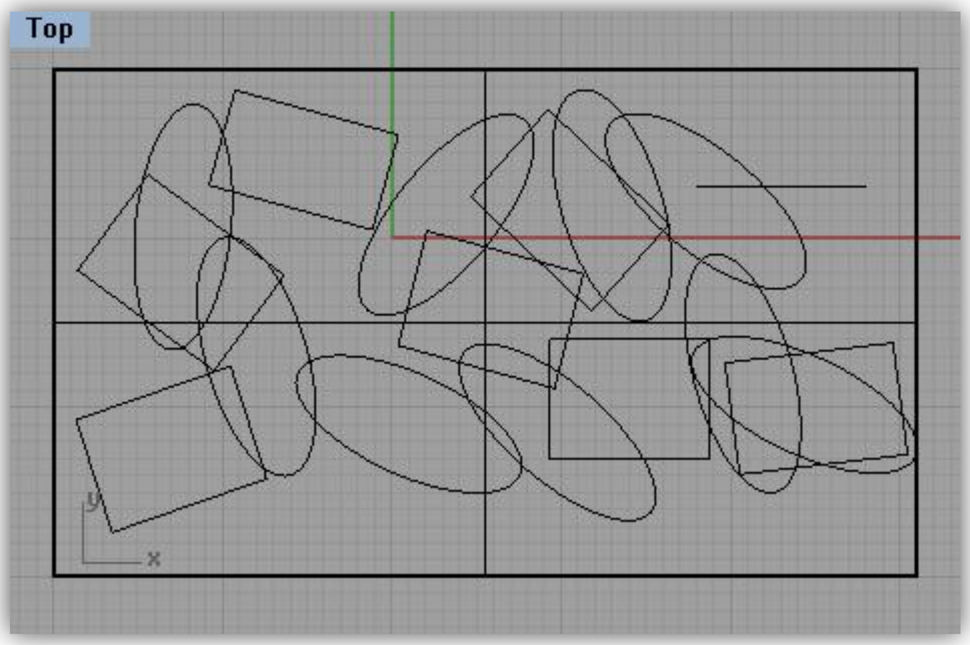


Figure 24: Random distribution of starting shapes (Top View)

Every curve object that is being distributed randomly in space needs to be checked for its position. The object should not overpass the strict outline of the bounding volume. This is checked using Boolean operations, as it was done in the first method (See Chapter 3.5.2). If a shape does not lie within the limit, it is deleted by the algorithm and a new one is being generated, until the desired number of shapes has been reached.

The generated shapes have a random position, orientation and size within the bounding box after they have been distributed in space according to the user preferences.

At this point, the user needs to define a numeric value that describes the maximum horizontal displacement allowed between two sequential curves. This is adding an architectural character to the whole procedure as it is filtering the results to surfaces that can stand as buildings.

The algorithm is starting from the basic curve and is searching within the area of the horizontal displacement allowed for generated objects. This is done by using the maximum allowed horizontal displacement value as radius for the theoretical cylinder with center the midpoint of the starting curve and height the total height allowed. Every generated shape that is lying within the boundaries of the cylinder is being selected.

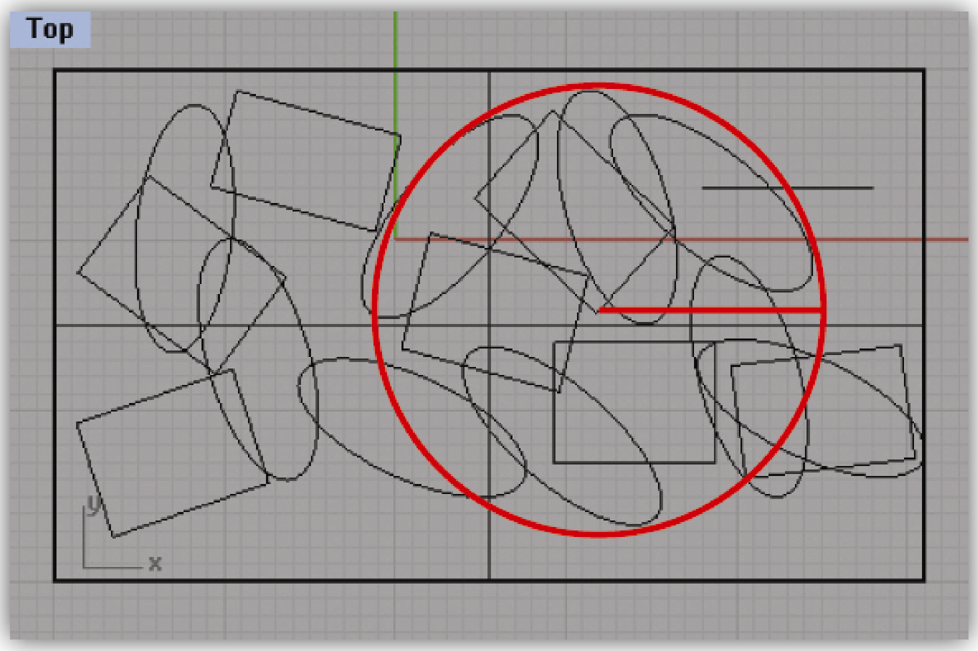


Figure 25: Random distribution and selection filter

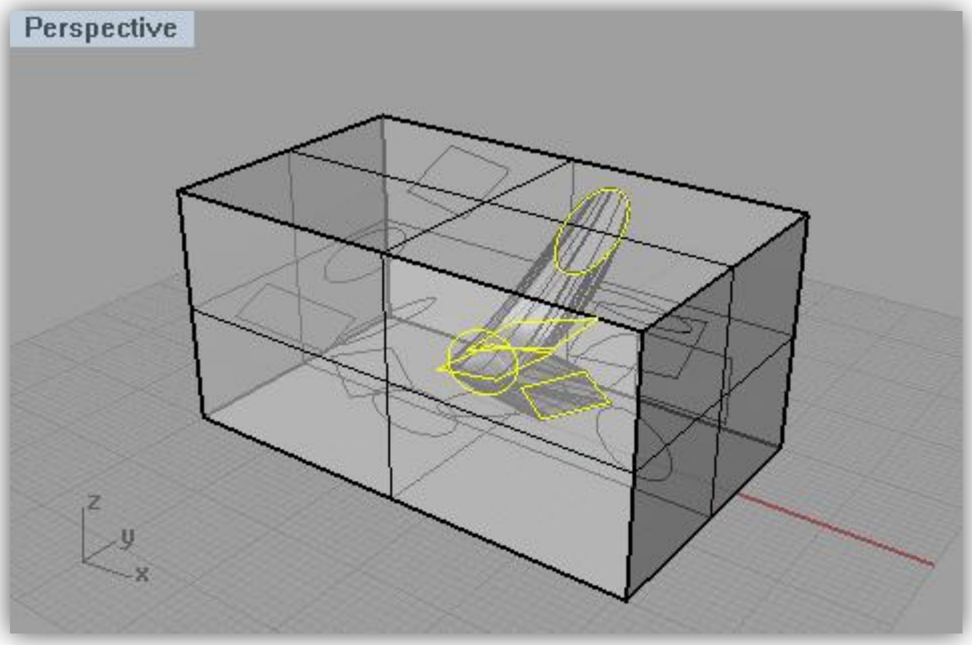
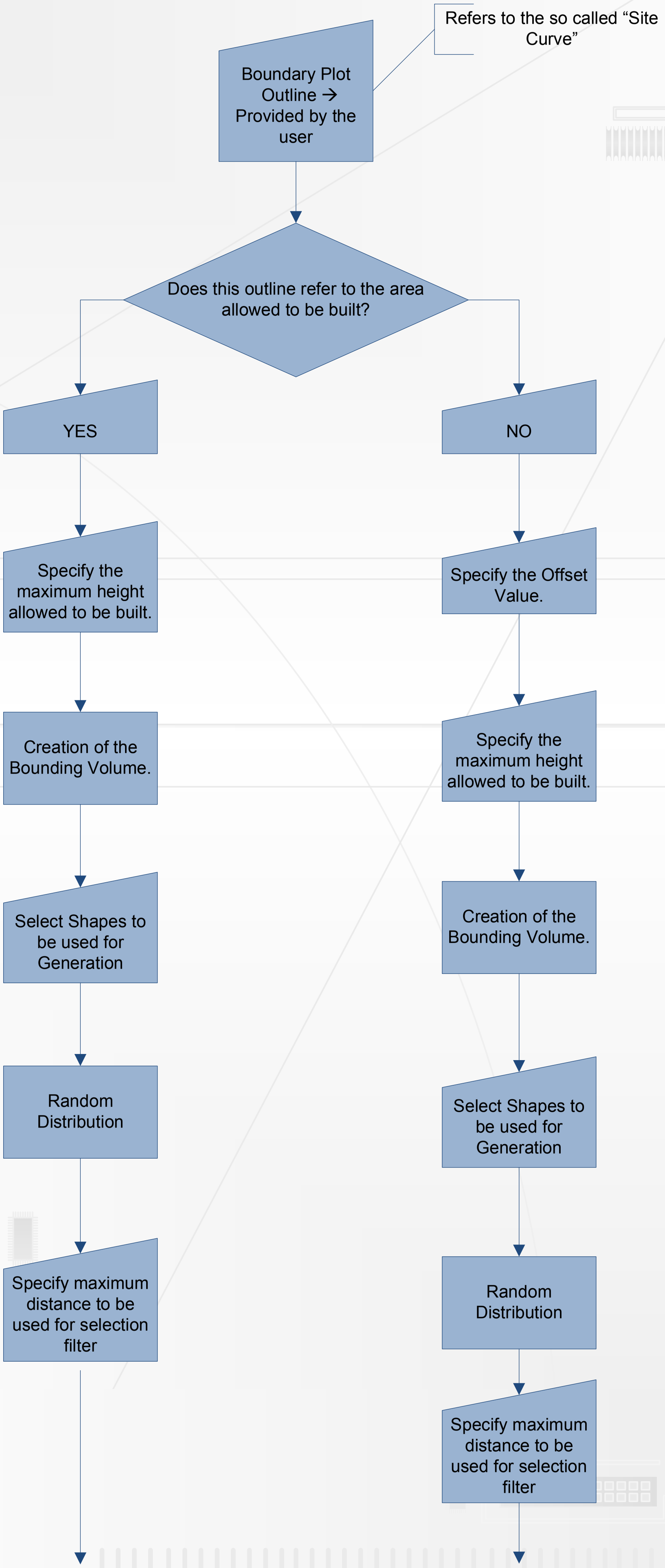


Figure 26: The selected and lofted shapes after the selection

After all the randomly distributed shapes have been tested and some of them have been selected, a wireframe of the resulting surface is being formed. By using skinning techniques such as lofting (see Chapter 3.5.4) the final surface is being created. The lofting type can be again chosen by the user or be chosen randomly by the algorithm if he decides to do so.

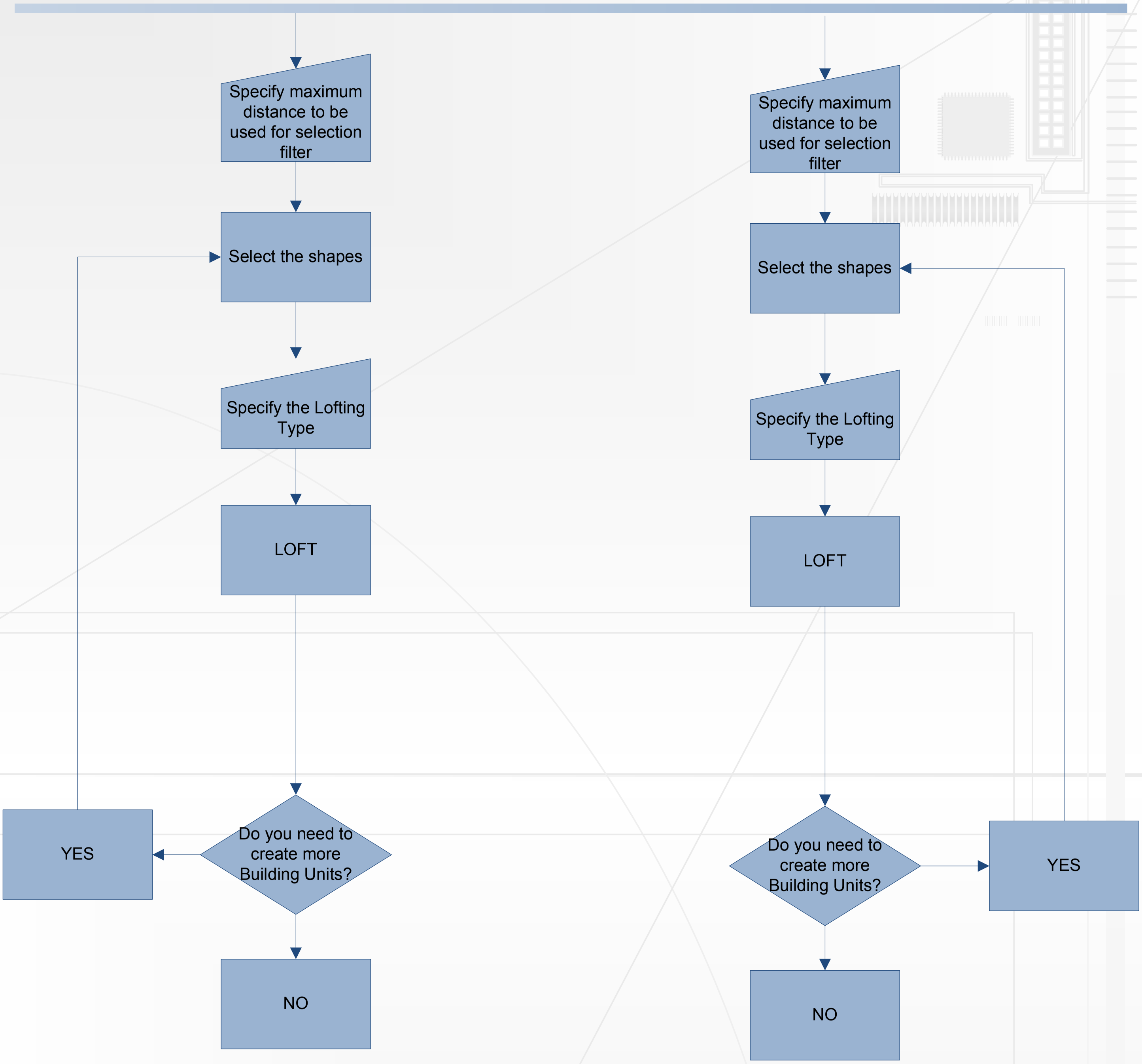
There are some cases where the user desires more units in the same building plot. In these cases more starting curves have to be selected. Every starting curve will form a cylinder around its midpoint. If two subsequent cylinders cut each other, the maximum horizontal displacement allowed will be limited automatically to the radius that the two cylinders osculate. Every shape will be treated as a separate unit obeying to the same rules and restrictions.

Parametric Form Finding using Transformation Modifiers



Refers to the so called “Site Curve”
Boundary Plot Outline → Provided by the user
Does this outline refer to the area allowed to be built?
YES
Specify the maximum height allowed to be built.
Creation of the Bounding Volume.
Select Shapes to be used for Generation
Random Distribution
Specify maximum distance to be used for selection filter
NO
Specify the Offset Value.
Specify the maximum height allowed to be built.
Creation of the Bounding Volume.
Select Shapes to be used for Generation
Random Distribution
Specify maximum distance to be used for selection filter

Parametric Form Finding using Transformation Modifiers


Specify maximum distance to be used for selection filter
Select the shapes
Specify the Lofting Type
LOFT
Do you need to create more Building Units?
YES
NO
Specify maximum distance to be used for selection filter
Select the shapes
Specify the Lofting Type
LOFT
Do you need to create more Building Units?
YES
NO

3.4 Comparison of the suggested methods

The two approaches described before are being used to create three-dimensional architectural form starting from two dimensional shapes, given by the user. Both methods indicate advantages and disadvantages – their value or disvalue lies on their usage. Although the two algorithms point to similar results, their structure has important differences, especially in the way that the two-dimensional curves are being treated.

At the beginning of both methods the same input data is required. This data is being used to provide information about the outline of the building plot and set the constraints. After having set all the necessary data, the two algorithms differentiate: the Form Finding method using Transformation Modifiers is requesting from the user to provide the desired shape to start the generation. At the same step, the Form Finding method using Stochastic Choice is requesting a set of curves that are going to be used by the algorithm.

The Transformation Modifiers' method is now starting to apply various transformations to the selected object (or objects) to form the first iteration. The procedure is being repeated according to the desired number of storeys. At the other side, the Stochastic Choice's method is analyzing the distributed curves one by one in order to select which one fits the needs to continue. This kind of searching stops as soon as the algorithm finds the first curve that fits the needs described – the algorithm is unable to distinguish between good solution, better solution and best possible solution. The first correct solution will be selected and the result will be based upon these selections.

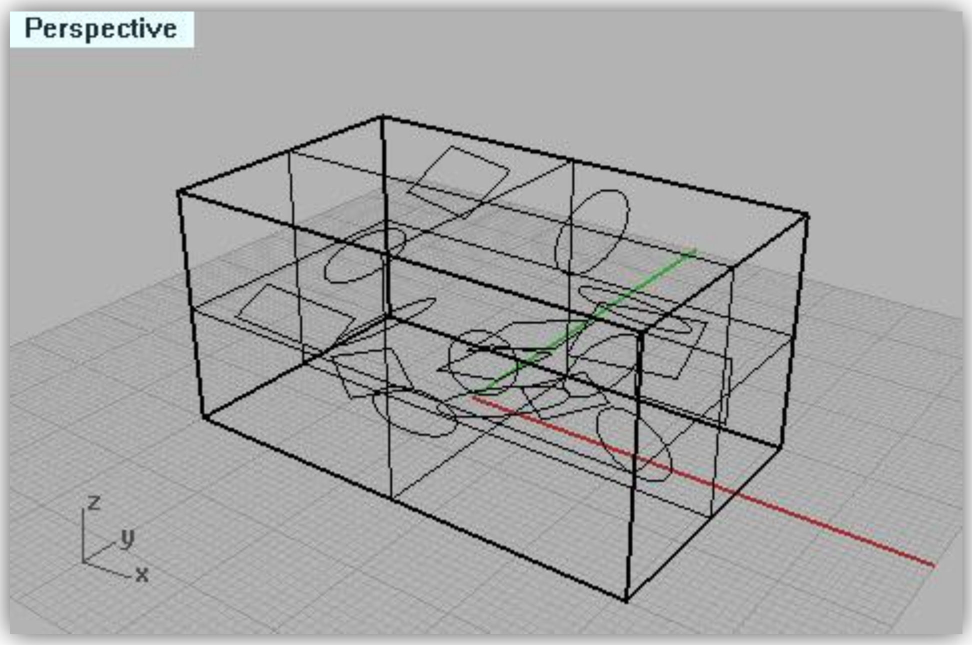


Figure 28: Screenshot of the plug-in showing the random distribution of the curves

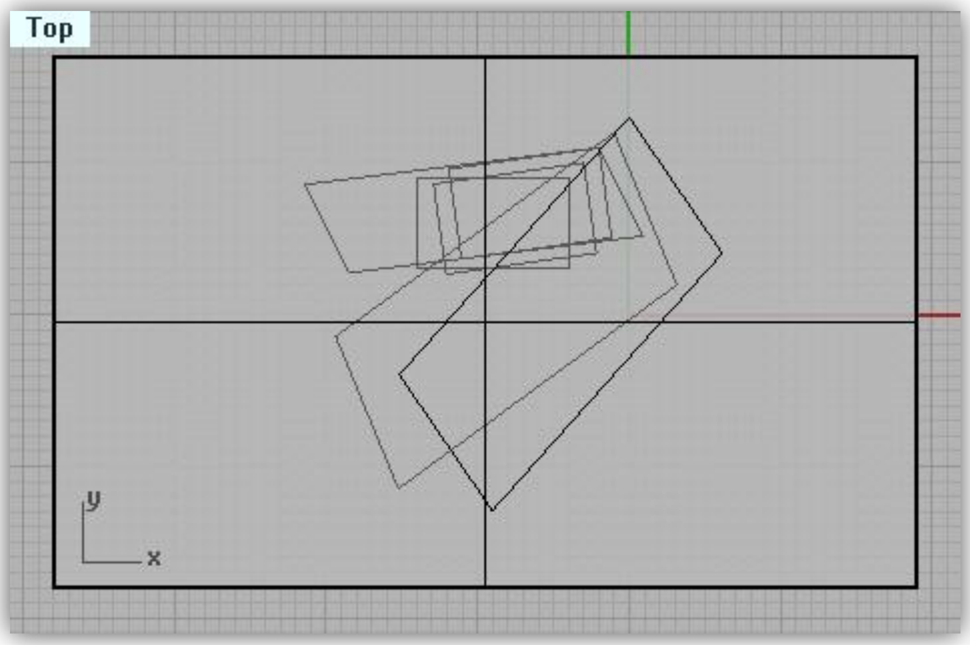


Figure 29: Screenshot showing generated curves within the bounding volume

In general terms, the Transformation Modifiers' method is first selecting and then creating, where the Stochastic Choice method is first creating and selecting afterwards. For this reason, the Form Finding method using Transformation Modifiers is performing faster. As this method is applying various transformations to a specific curve and is performing a similar procedure more than one time (according to the desired number of stores), in contrast to the Form Finding Algorithm using Stochastic Choice, that has to analyze all the neighboring shapes and decide which is the one that will be selected, it is obvious that the time needed for the first method to complete will be dramatically smaller than the time needed for the second one. As time is a very important factor in computer science, the first approach is preferred as this will dramatically decrease waiting times.

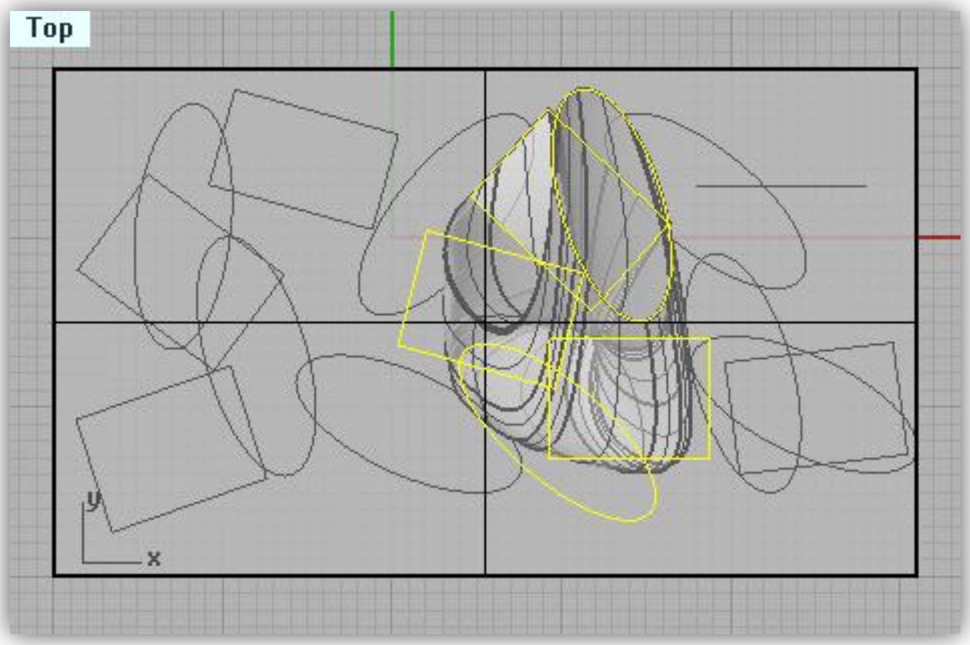


Figure 30: Screenshot showing the selection step and the lofting technique

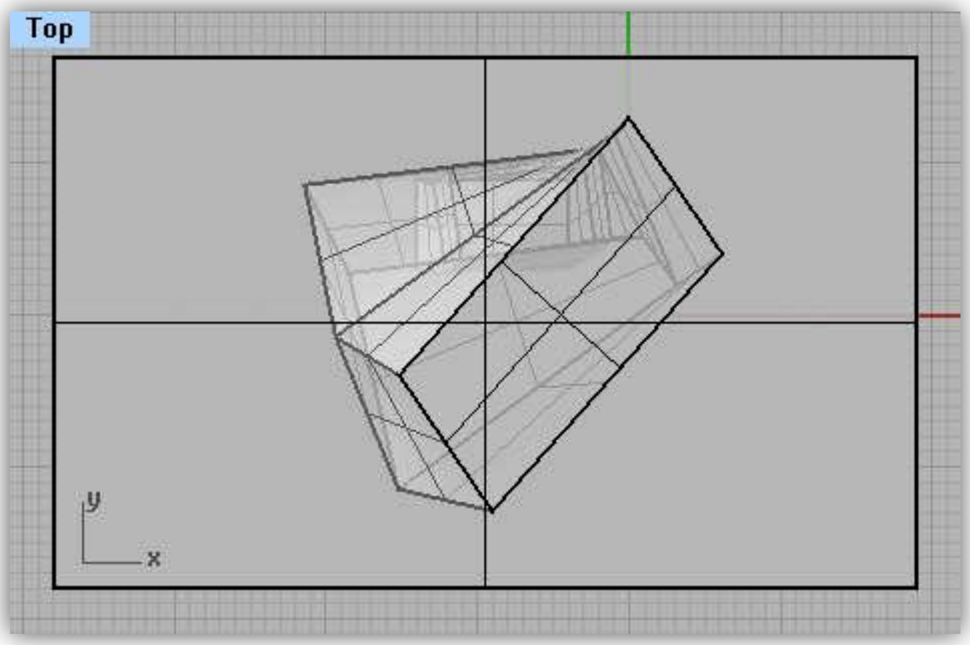


Figure 31: Screenshot showing the skinning technique within the bounding volume.

It is hard to distinguish which method is better, as both of them serve different requests. For more organic solutions, the approach using transformations modifiers seem to point out to more complete solutions than the approach using the stochastic choice. From the other side, the transformation modifiers' method is harder to add a chaotic note in terms of curve's position and rotation randomness to the resulting form.

Parametric Form Finding using Transformation Modifiers - Elaboration

The analysis of contemporary architectural pieces indicates that building design can be based on design decisions and form can be analyzed to two-dimensional curves and operations connecting these curves. This connection procedure of as well as the algorithm generating these curves is being analyzed in this approach. The opposite procedure of the analytical form breaking down (see section XXX) was used as starting point in the early stages of this thesis.

Our approaches deal with:

- a group of curves that are used as a basic element of the algorithm,
- variables that are used to provide the desired limitations of the three-dimensional form and
- Boolean values that are used as decisions for the program to follow specific approaches of architectural design.

3.4.1 Elaboration of the boundaries

The first step of the algorithm is a user-provided input, defining the basic curve that holds the building plot. This curve (the "Site Curve") should be drawn by the user or imported into the CAD program if drawn in another application. The Site Curve is drawn in the top window as this is the one that provides a floor plan-like view of the data

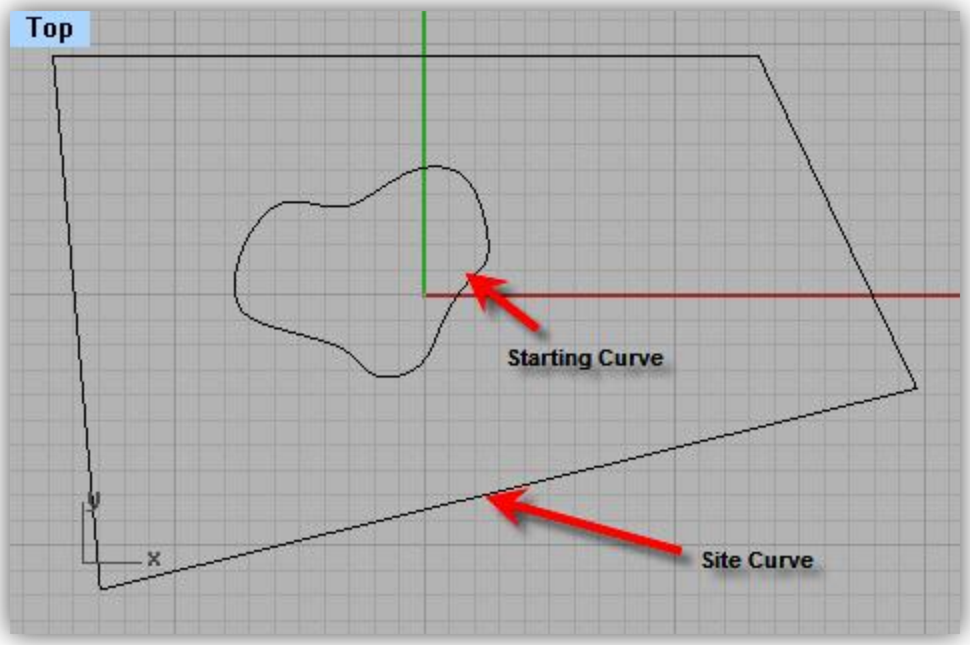


Figure 32: Screenshot of the plug-in showing the Site and Starting Curves

Occasionally, the boundary outline of a building plot is different from the outline that frames the building because of regulations that do not allow the exterior walls to touch the neighboring buildings. The program is taking into consideration this distinctiveness and in order to maintain its simplicity it is first asking for the original boundary outline of the building plot.

The next step is to ask the user whether the Site Curve refers to the area that is allowed to be built or there are some other regulations that need to be specified. This Boolean value represents a decision that needs to taken by the user for the program to be able to continue based on the answer. In case the user specifies that the actual Site Curve given can be taken as it is, the program continues without any further questions to the next step, presuming that the curve provided is the total area allowed to be used for placing the prospective building. In all other cases, the program requires numeric data about offsetting the given curve before selecting it. The offset function requires that the starting curve is exploded into linear pieces as the boundary displacement may not occur on every side of the provided curve. After exploding the given shape, the program asks the user to select the sides that need to be offset. After the linear elements are selected the program performs the offset function and joins the linear parts into a curve again.

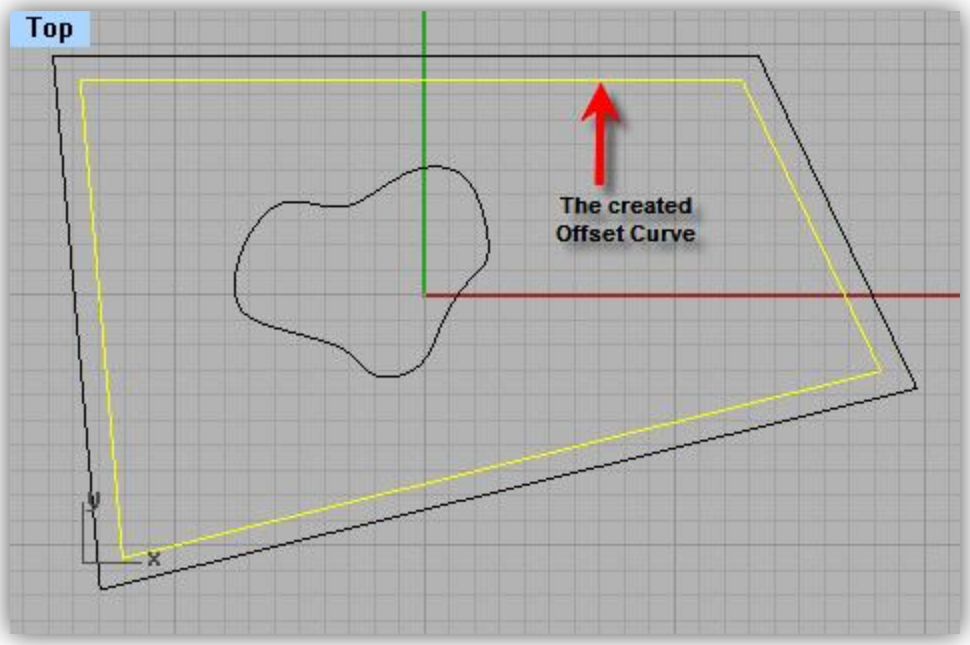


Figure 33: Screenshot of the plug-in showing the Offset Curve

The whole procedure performed to the Site Curve is mandatory in order to create a bounding volume that defines the boundaries of the resulting building. The bounding box will be able to check if the result fits within it so it will be allowed or in case it will not fit, it will be objected and the generating algorithm will search for another solution.

After having defined the Site Curve, a numeric value, that defines the maximum height that the building is allowed to reach, is requested. This value differs between cases, as it is being influenced by diverse factors such as the building law, the boundary conditions, the structural system, the design idea etc. The number provided by the user is the maximum value that the height of the result is acceptable to reach – every solution that overpasses this height must be rejected.

By combining these two elements, the Site Curve and the Total Height, a Bounding Volume is created that defines the boundaries of the possible results. It is hidden from the user as he doesn’t need to get into the procedure of comparing neither he needs to notice all the solutions that are generated and automatically rejected in case that they do not obey to the restrictions provided. The bounding volume covers the maximum volume that the resulting building will be taking if the Site Curve would be taken as starting curve and straight extrusion would be used. From a programmatic point of view, the bounding volume is a fitness function that checks whether a generated form fits inside the proposed bounding volume which represents the building limits.

3.4.2 Discussion of the stories

The model is using a storey-based logic that adds characteristics to the solution, such as harmony, proportion, freedom and flexibility. The step-wise logic is being limited within storeys and every storey is treated separately as a unit. The user inputs the desired number of storeys that the resulting form should have. The total height allowed is then being divided by the number of storeys to obtain the approximate storey height. This is then used to define a smaller storey bounding volume –s similar to the total bounding volume is acting: it defines the storey limits. The storey-building is done by adding horizontal cross-section of the building, but defining the storey limits instead of the building limits. Every cross-sectional surface represents a slab and the distance between two sequential surfaces is equal to the approximate storey height.

The idea of storey is adding a unique architectural character to the algorithm as this concept is widely used in architecture. Every building in fact consists of a fixed number of storeys. Even contemporary organic architecture using amorphous skin techniques includes the concept of story in a more abstract sense, meaning the height of spaces between floor and ceiling. Rem Koolhaas and the Office of Metropolitan

Architecture like to create voids within the amorphous building mass, which use double or triple storey height. As contemporary architecture negates the idea of a column-and-slab-based building where the importance of storey is clearly visible, the concept of the storey still remains as the vertical distance between floor slab and ceiling that is needed to maintain comfort within a fixed architectural space.

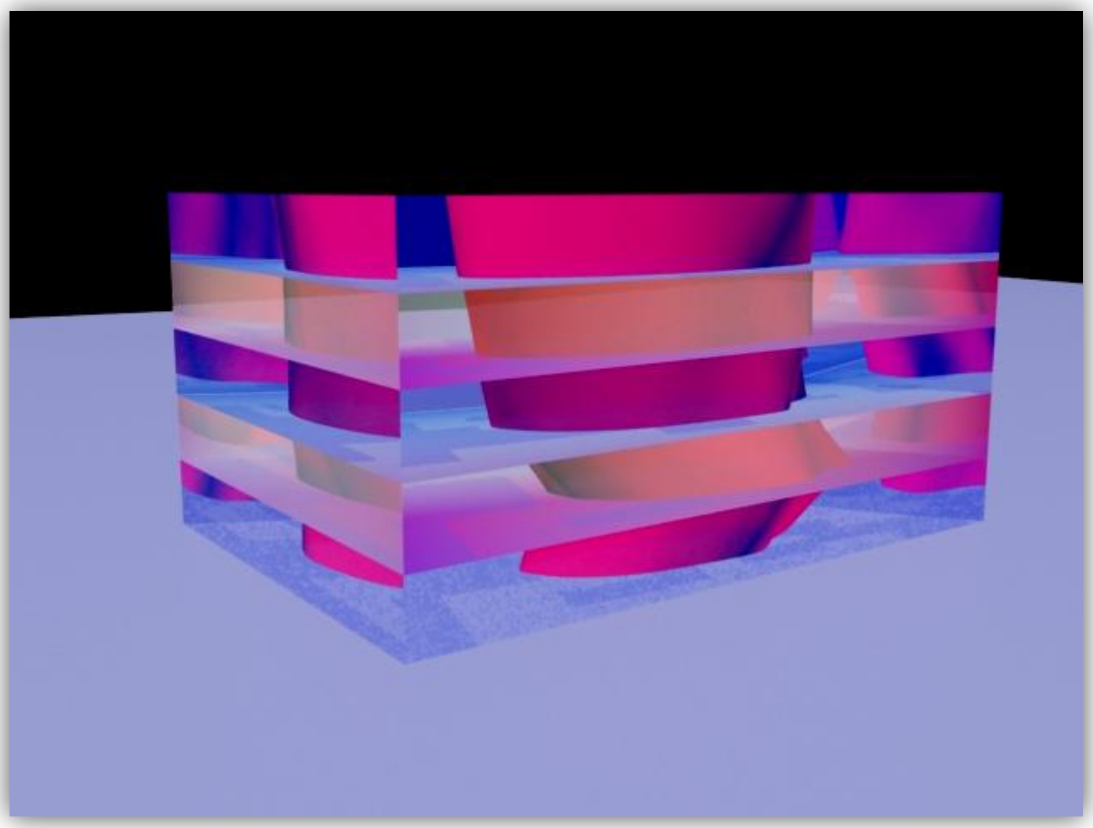


Figure 34: The bounding volume, storey height and generated form in a typical example

Le Corbusier based the architectural proportions onto the golden ratio in his Modulor, which was a system continuing the work of Vitruvius and Leonardo da Vinci concerning the proportions of human body aiming to improve the quality of architectural projects. He was based on human measurements, Fibonacci numbers and the double unit. Le Corbusier’s philosophy about the relation of the human body and the architectural spaces forced systems of harmony and proportion to be placed in the centre of design techniques and is followed widely by contemporary architects. The Modulor philosophy that stands responsible for the fitting and comfort of the human form in a fixed architectural space is strongly related to the concept of storey and storey height. (Padovan, 1999)

3.4.3 Elaboration of the Form Generation Step

After having defined the Site Curve, the total height allowed, the desired numbers of storeys, the total and sub-bounding volumes have been calculated, the algorithm is ready to start generating three-dimensional form. This is a procedure based on repetition as a curve is being copied every time and various transformation modifiers are being applied on it. The amounts of transformations applied as well as the iterations performed are being based on a randomizer.

The generation algorithm starts by requesting from the user to select the starting shape. This is always a two-dimensional curve drawn by the user and should be always on the ground level. After checking if the object selected is indeed a two-dimensional curve object, it is stored in a variable and various transformations are being applied on it.

The first step is to create a copy of the curve that will be accepting the random transformations. The curve stored in the variable is being replaced by the copied curve object in order to remain unaffected. The object is pasted in place – the two identical curves are on top of each other.

The next step is to use the *MOVE* function to randomly move the copied object along the three axes. Without having defined any minima or maxima of movement freedom, the randomizer is being asked to create three random positive numbers that will include the movement values along the three axes. The object is able to take different movement values along every axis as the three degrees of freedom that emit from the movement function are absolutely independent. By starting from the initial position where the starting curve was originally drawn by the user (not necessarily the point (0, 0, 0)), the object is being moved to a different point of the Cartesian System by a value created by the internal randomizer.

The Z-movement is determinative for the rich three-dimensional form creation. The more times an object is being copied and moved along the total height value, the more interesting the resulting form will be. The algorithm needs to ensure that every storey bounding volume will contain at least two curve objects. This is done by introducing a maximum value for the randomizer of the Z-movement. This maximum equals the approximate storey height.

The next step is to use the *ROTATE* function, which stands responsible for the orientation of the object. A similar procedure to the movement function is being followed: the rotation occurs around the three axes and each axis has a unique value of rotation generated by the randomizer. These values are generated numbers between 0 and 360. The rotation function is being applies three times, one time for each axis.

As the starting shape is being drawn in top view window, its normal vector is always corresponding to the Z-axis of the Cartesian System used. As the curve is being rotated along time, the local axes are being used instead of the Cartesian ones. The Z-axis always falls on the normal vector of the surface created by the corresponding curve and agrees to the normal vector's direction and orientation. The X- and Y- axes are always perpendicular to this vector.

The last step is to use the *SCALE* function which stands responsible for the size of the copied object. Scaling also uses three axes, imitating the procedure followed with the two previous functions, but this time the object is being deformed. When scaling has not the same value along the three axes, the object is pushed and pulled along the axis line and interesting effects can be generated with this procedure. As the starting shape does not remain, the *SCALE* function is the one that offers diversity to the generated results. It is using a scaling factor and as this factor is generated by the randomizer, each resulting shape is different than its previous one, but in a way that it could emerge from it.

In order to maintain harmony and proportion, at every iteration of the algorithm, the generated shape is being chosen as a starting shape for the next iteration. This ensures that every method followed will be taking into account the previous step and will be actually based on it. Especially methods as the *SCALE* function, that deform the given curve, need to be based every time on the previous step in order not to allow sudden changes in size.

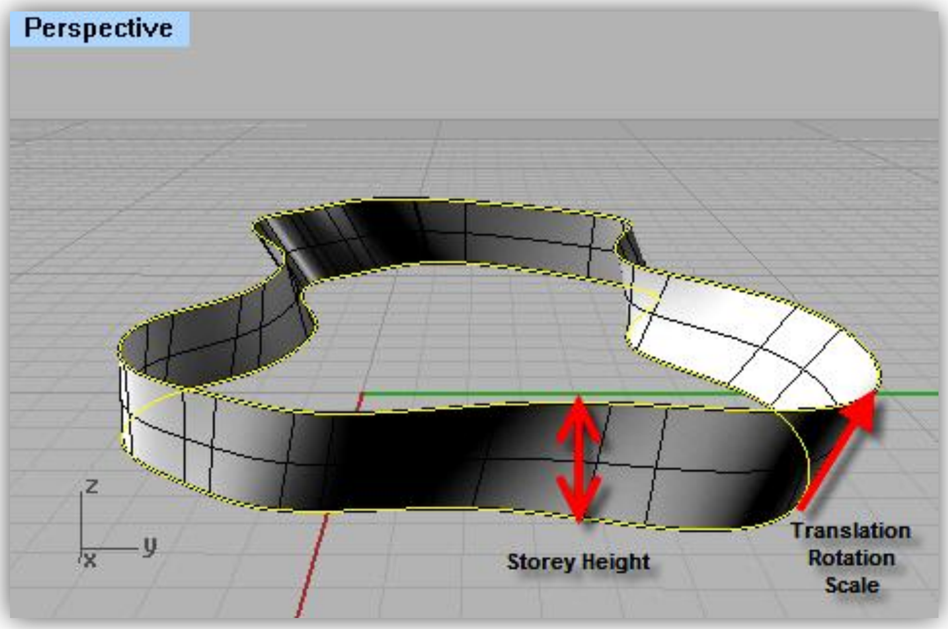


Figure 35: Screenshot showing two sequential curves and the lofting technique

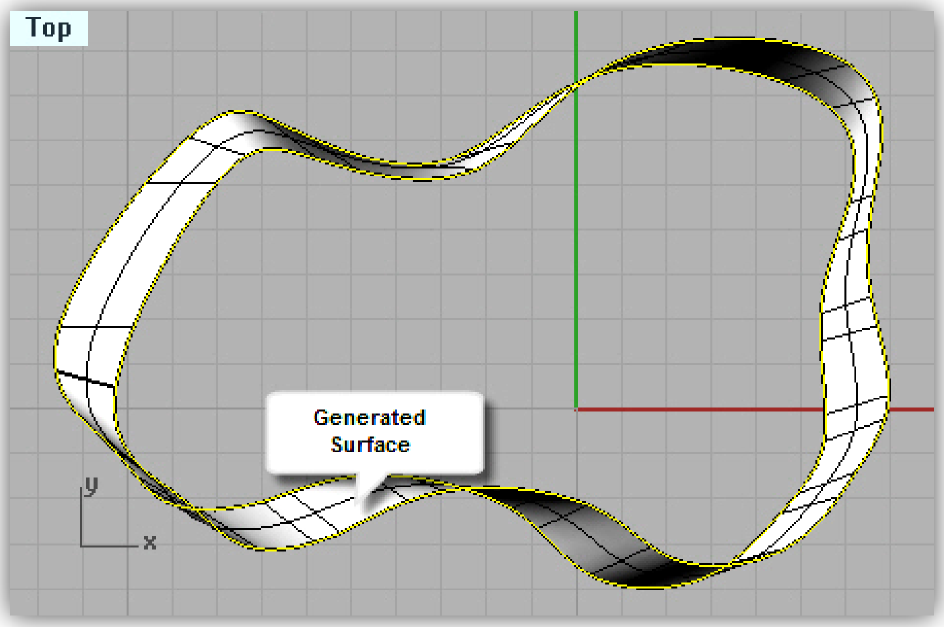


Figure 36: Screenshot of the plug-in showing the generated surface

At the end of every set of transformations the procedure is being repeated until the total height provided by the user is being reached. As it is seldom to reach exactly the value provided by the user as the total height allowed, a threshold of -/+2% is being used.

3.4.4 Types of Lofting

After being performed several times, the algorithm has created the wireframe of the resulting three-dimensional form. The curves created by the algorithm need to be created by using a skinning technique such as *LOFTING*. There are multiple types of Lofting, such as normal, loose, tight, straight sectional and developable:

1. *Normal*

The surface has a "normal" amount of stretching between the curves. This is a good choice when the curves are proceeding in a relatively straight path or there is a lot of space between the curves.

2. *Loose*

The surface control points are created at the same locations as the control points of the loft input curves. This is a good option if the control points will be edited later or when the curves are just a hint of the final shape.

3. *Tight*

The surface closely follows the original curves. This is a good choice when the input curves are going around a corner.

4. *Straight sections*

Creates a ruled surface: the sections between the curves are straight.

5. *Developable*

Creates a separate developable surface from each pair of curves.

The lofting type is left to the user to decide, although there is a random option for the program to decide on its own. After the generated curves have been lofted, the result is ready to be checked for its fitness.

3.4.5 A Note on Fitness Rules

There are two approaches to checking - one is to check after each iteration, the other is to check after the whole process. Since the buildup relies on randomness, and the checker throws away solutions it does not like - it is true that the user probably will have to wait for a long time when tall buildings are specified. However, we can constrain our randomness so that the algorithm will not produce so many superfluous storeys or unnecessary buildings, at the cost of producing less "random-looking" architecture. Furthermore, we may incorporate that the checker continues throwing away the "non-fit" storeys buildings. This would at least ensure correct termination of the algorithm.

The verifying process checks whether the created object lies within the user defined boundaries. This is done by using geometrical Boolean operations such as “difference” that cuts away the shared areas of selected surfaces with another set of surfaces. If the result of this Boolean method is zero, the bounding volume and the generated object do not cross, meaning that if the user created the initial curve inside the boundaries of the bounding box, the created object stays within the limit. If the result of the Boolean method differs from the value zero, this means that the two objects are crossing and the created object overpasses the limit of the bounding volume. In these cases the procedure has to start from scratch by using the same input data and preferences to create a different solution.

More surface and volume characteristics can be measured here to make the procedure stricter and to limit more the field of results. Characteristics such as volume of the generated form and surface curvature could be measured at this point or shown to the user as information (e.g.: "The total generated volume of the building is..."). If the user agrees with these characteristics the results are being kept – there is also the option to defined desired values for these fields and the algorithm will search for solutions that fulfill these expectations. If the desired volume and surface curvature are set from the beginning of the algorithm, according to some threshold used, the design methods will become more exact.

3.4.6 More units in the Building Site

In several cases more building units need to be created in the same building plot. The same procedure needs to be repeated more than once, according to the desired number of units. More “Site Curves” need to be defined and every time the algorithm will search for possible solutions that fit into the boundary selected. The solutions will not necessarily follow similar design techniques as they will be based on the patterns given by the user initially, although this would be a good extension for future work to come. After the first three dimensional form is created, the user is asked whether the generative process is finished or should be start again using the same settings but a different Site Curve.

3.5 Parametric Form Finding using Stochastic Choice - Elaboration

The method of Stochastic Choice is approaching the problem of architectural Form Finding from a different path: instead of creating at every iteration the next shape that will be used later for lofting, as it was done in the previous section, the curve objects are being selected from a wide field using criteria. The method of Stochastic Choice checks the already created object if it fits the given function consisting of rules and restrictions.

3.5.1 Elaboration of the Boundaries

At the beginning of the algorithm the Site Curve is needed. As it was done in the previous approach, the Site Curve (meaning the area of the building plot allowed to be used for the building), is needed to indicate the boundaries of the generation process. In collaboration with the total height allowed to be built, the bounding volume is being created, as it was done in the Parametric Form Finding using Transformation modifiers. The desired number of storeys is also requested.

The user is asked to select all the curves that are going to be used in the process. These objects are stored in an array for being used in the process. Instead of selecting every object individually, after copying them multiple times, the algorithm makes a random distribution of the curve objects within the bounding volume. As the parts are being moved and rotated randomly, they are being checked if they lie within the desired space. If not, they are being deleted by the algorithm and new copies are being created. The procedure continues until the set number of desired distributed shapes has been reached.

The curve objects generated with the random distribution in space are being now checked if they lie within the bounding volume. In some cases the surface of the sphere overpasses the outline of the bounding volume; there is a volume in-between where generated curves can be stored. The algorithm needs to check for the existence of these objects and delete them. This is performed using Boolean operations – every surface is being compared to the bounding volume. If it crosses its surface, the algorithm deletes it, in all other cases it moves to the next shape. As the shapes provided by the user are not constant, there is a chance that curve objects might not cross the boundary volume surface, but still lie in the volume between the sphere object and its surface. This is why an additional restriction needs to be introduced –

the radius of the created sphere object has to be checked. If the smallest length of the provided curves is smaller than the distance between the surface of the bounding volume and the surface of the sphere, then the radius of the sphere object needs to be modified until this condition is not true. If this condition is not met, there is no way the randomly distributed curve objects lie between the two surfaces, of the bounding volume and the sphere object, without crossing one of those surfaces.

3.5.2 Selection Step

Next, the program asks the user to define the maximum allowed horizontal displacement of two subsequent curves. The numeric value provided stands as an additional restriction for the next step, the selection of shapes from the random distribution.

For the selection of the curves, an action field needs to be defined. A curve object within the bounding volume is being selected randomly. According to the maximum allowed horizontal displacement given by the user earlier, a cylinder of radius the value of horizontal displacement and height the total height allowed is being created. This object represents the selection field for the second curve. If a second curve is lying within the cylinder volume, its horizontal distance from the first object is smaller than the maximum horizontal displacement allowed, which is also the radius of the cylinder object. The algorithm selects a second curve randomly and checks whether it lies within the selected field. If yes, the object is being selected and is moved to a new layer. For the next iteration, the selected object acts as a starting curve and a new one is being searched. If the algorithm does not indicate that the curve object analyzed lies within the desired field, the next object is being selected and the same procedure is being performed until this condition becomes true.

There are some cases where the specific random distribution of curve objects fails to give a curve that fits the needs for selection. In these cases, the generated objects are being deleted and distributed randomly for a second time within the bounding volume, as it was done in the beginning. This is done every time in case the algorithm does not find a curve that lies within the desired selection field until a second object is being selected.

3.5.3 Discussion on the storeys

The concept of “storey” is also introduced in this approach: after the cylinder that includes the selection field for the next object is being created and before the algorithm starts selecting any new shapes, the user is being asked whether he desires to set a specific number of storeys for the total height provided. A decision needs to be made which specifies in which way the algorithm will continue: if the user selects the first option, that he wants to specify a specific number of storeys, a numeric value is being required. This data is being stored within a variable for being used later. After the cylinder that represents the selection field is being created, its height is being divided to the desired number of storeys value. Smaller cylinders are being created which represent the selection field for every storey, similar to the total bounding volume and the storey bounding volumes from the previous procedure. Now, in order for the algorithm to continue, at least one curve object is needed within the smaller cylinder that represents the first storey – in case this is not found, the random distribution is performed again. As the desired shape is found, the algorithm checks if there are other shapes within the storey selection field. If there are, they are being selected and moved to a different layer, together with the ones selected by the program. In all other cases the algorithm moves to the next storey and performs the same procedure as before.

The procedure of storey setting requires more time, as the objects need to be distributed randomly more times. Although time is an important factor for the generating procedure, the idea of storey adds freedom and flexibility to the design process that balances the extended time factor.

3.5.4 Extended Lofting

In order for a surface to be created based on curve objects, at least two curves are need. As soon as at least two curves are being found that lie within the boundary, the algorithm does not need to distribute the objects again. After the second curve is being found and selected, the program is searches for more curves within the selection field. Every curve is being chosen and analyzed for its position; if it fits the needs of the selection function it is also chosen and moved to a different layer. This is done for all the curve objects in the scene, as long as at least two of them have been selected.

As soon as the algorithm checks all the objects in the scene, it selects then all the curve objects that have been moved to the new layer. These objects are being put to an array to be used for lofting. The contents of the layer form the skeleton of the proposed building – a skinning operation such as lofting is required to create a surface through these curves. The lofting operation, as it includes different types, it is left to the user to decide which one to use, as it was done in the previous approach.

There is also a random option, where the algorithm creates randomly the type of lofting, chosen from a fixed variety that the CAD application that hosts the algorithm has to offer.

3.5.5 More units in the same Site Curve

Occasionally, more building units need to be created in a given building plot. The approach of Parametric Form Finding using Stochastic Choice includes this option in its internal structure: as the curve objects are being distributed randomly within the bounding volume, the algorithm can choose any shape close to the ground level and make it a starting shape. In case the user declares that he wants to have more building units within the same plot, after the first unit is being created, the procedure starts from scratch using the next curve that lies close to the ground level as a starting curve. The algorithm distinguishes the curves that lie close to the ground level by their Z-value.

3.6 Analytical Approach of the Form-Finding Process in RhinoScript

3.6.1 Introduction to RhinoScript

RhinoScript is the scripting environment of Rhinoceros, developed by McNeel & Associates. Rhinoceros, aka Rhino, is a complete easy-to-use three dimensional modeling package that is widely used by architects, designers, etc. RhinoScript is based on the Visual Basic Programming Language.

The environment of Rhinoceros has been chosen to host the Form-Finding plug-in suggested in this thesis. The reasons are the following:

- Rhino is a well-known application and is being used by many designers.
- RhinoScript is based on Visual Basic which is a programming language based on the English language and fairly forgiving.
- Rhino is working with NURBS objects.
- Rhino supports more ways for creating surfaces than the other similar modeling packages.
- Rhino is a complete package, including modeling, rendering and animation interfaces.
- It has many exporters for all the famous formats of 3D modeling, something that widens the collaboration possibilities between different environments.

The method followed in this approach was the one using ‘Transformation Modifiers’. This approach showed better and faster results than the other method.

3.6.2 Analytical Workflow

3.6.2.1 Variables

The first step of the approach was to define different variables for all the non-constant fields of the program. In order to use loops, variables are necessary to store different values of the same field for every iteration. In this case fields such as the maximum distance between two sequential curves, the maximum storey height allowed, the Site Curve, the Lofting Type, etc, were put in variables, as they would change in every run of the program. Many of these variables are user-triggered, meaning that they request from the user to specify a specific curve or to provide a fixed number, distance or angle. Most of the variables that were being used in the final version of the program appeared as need during the coding procedure.

3.6.2.2 Creation of the Bounding Volume

After the variables have been defined, the user is asked to provide numeric values for the Maximum Distance between two sequential shapes, the Maximum Total Height for the building to reach and the desired number of storeys. This information is mandatory for every example as it includes the limitations for the resulting form. This data leads to specific design paths and methods that will be able to satisfy both the program and the user.

The next step is to ask the user to select the Site Curve. This curve represents the outline of the building plot, the area allowed to be built, which can either be imported from other applications or be drawn in the existing environment before the start of the program. By combining the Site Curve with the Maximum Height allowed (provided before), the program creates the Bounding Volume using Extrusion methods. The Bounding Volume acts as a limit for the form generation – every object that will slice the bounding surface will be denied and the procedure will start again.

3.6.2.3 Starting Object and additional Curves

After having defined the Bounding Volume, the user is asked to select the Starting Curve. This Curve is the object that the generation will start from. This object is the Base Object of the resulting form and the continuation of the algorithm depends strongly on it. The resulting form will follow its shape, by differentiating slightly from it. The algorithm stores the Starting Curve in a variable and checks if the object selected is a curve object indeed. As the algorithm interacts with the user, he is free to select any object, or press a button instead, or even make no selection. The checking is compulsory and is called Error Trapping. If the algorithm finds an error caused by the user in the selection field, the procedure is being ended and an error message appears, warning the user to start again and select a valid curve object. Error Trapping takes usually more space than the actual algorithm, as it is responsible for preventing all the possible mistakes in order for the algorithm to run smoothly.

After the Starting Curve has been selected and the program has checked that the object selected is a valid curve object, the algorithm requests from the user to specify whether he is going to use just the starting shape or more shapes for the form generation. At this point there are two cases and according to the user’s choice, the algorithm will follow two different paths. If the user selects to use more than one shape, the algorithm asks to select them. As the number of the shapes that are being selected is unknown, no fixed number of variables can be defined to store the additional curve objects. Instead, they are being stored in an array that has no upper limit. After the curves are being selected, the same procedure of Error Trapping checks whether the objects selected are curve objects indeed.

3.6.2.4 Form-Finding using the Starting Shape

In case the user chooses to continue the form generation using just the Starting Curve as a basic shape, the algorithm is ready to proceed to the next step. Initially, the user is being asked to specify numeric values for the minimum and maximum rotation angle as well as the minimum and maximum scaling factor. The maximum distance between two sequential curves has been provided earlier.

The Starting Object is being copied and is being moved using random values along the axes. The values generated do never overpass the limits of the Bounding Volume, neither the limits of the maximum distance allowed given earlier. After the object has been moved, it is being rotated using the randomizer again and taking into consideration the minimum and maximum rotation angles. Finally the object is being scaled randomly, using values between the minima and maxima provided by the user. This procedure makes possible for the algorithm to transform the starting shape by copying, moving, rotating and scaling it by maintaining within the limits. The randomizer makes rapid transformations and every time the generated curve is different than the previous ones.

The procedure is divided into specific steps that equal the desired number of storeys. At the beginning of the algorithm a counter variable counts the iterations for copying, moving, rotating and scaling – as soon as this number equals the desired number of storeys, the procedure has been completed.

3.6.2.5 Form-Finding using additional Shape

In case the user chooses to continue using additional shapes for the form generation, the same procedure is being followed as in the previous choice. The only different is that as the user has to draw the additional shapes, the algorithm is not sure for the position of these shapes. As the drawing space of Rhinoceros is indefinite, the shapes could be drawn in completely different places, something that makes impossible the connection between them for formatting a building.

In order to proceed, these shapes have to be moved close to the starting shape. This is performed by moving every additional shape from its gravity center to the gravity center of the Starting Curve. After all the selected curve objects are on top of each other, the same procedure is executed as in the previous path with the only difference that before copying the shape, the randomizer selects one of the shapes to continue.

3.6.2.6 Lofting and Tapping

After having created the basic wireframe based on the objects provided, the algorithm needs to loft the resulting curves by creating a surface that connects them. As Rhino offers different lofting possibilities, including Normal Lofting, Loose Lofting, Tight Lofting, Straight Section Lofting and Developable Lofting, the user is asked to select the lofting type that will be used to connect the generated curves. The numeric choice for the lofting type includes a random lofting type, which generates a random number that corresponds to a random lofting type of the available ones. This feature randomizes the whole procedure, from the curve transformation to the lofting type.

After the Lofting procedure, the algorithm performs tapping to the resulting surface in order to close the open top and bottom of the resulting surface by adding two more planar surfaces on the top and bottom curve. This is done by checking whether the top and bottom curves are planar indeed, if not the generated surface is left as it is.

If the surface has been closed, useful data is calculated such as resulting volume, surface area and surface curvature. These values are printed for the user and in case he is not satisfied with the results, the algorithm starts from the beginning using the same curves and restriction values to generate a newborn result.

4 Future Efforts and Discussion

The search for appropriate form has always been approached with innovative up-to-date solutions – especially in the 21st century, a technologically advanced and architectonically saturated era, contemporary methods are being called to offer solutions to the problem of Form-Finding. As the process of early architectural design can be semi-automated using algorithms (See paragraph 3.1.3), generative approaches, such as L-Systems, can be transferred into the design workflow (See paragraph 3.2 - 3.3) and basic restrictive parameters as well as preferences can be imported to a system that offers different solutions to the specific problem based on these parameters (See paragraph 3.2 - 3.3), a plug-in for the modeling environment of Rhinoceros has been implemented that is offering three-dimensional architectural Form-Finding based on two-dimensional user-provided curves. Although the plug-in is covering both organic and non-organic architectural cases, the lack of aesthetical consciousness leads many times to numerous unwanted results and the procedure takes more time to complete.

4.1 Aesthetical Consciousness

As the algorithm generates three dimensional form based on user’s restrictions and preferences, targeting architectural elements, aesthetics have a major impact whether the result will be kept by the user. Aesthetics is an important factor that adds value to the existing perfectness of form, function and ease of use. The aesthetics of form and the surface consciousness differ between different people and it is hard to teach a machine how to distinguish between beautiful and ugly. Although computers are able to understand functionality, ease of use and other form attributes –attributes that can be expressed through mathematical expressions, the lack of aesthetical consciousness can lead to non-desired results. The search for appropriate form can go through many different solutions of the algorithm, which is indeed faster and more efficient than the traditional methods of Form-Finding, yet time-consuming.

4.2 Neural Networks

The solution to the differentiation of the Aesthetical Consciousness can be given through Neural Networks. If the application was able to understand the user’s preferences concerning form aesthetics, the results provided would be filtered down to the ones that really fit the needs of the specific problem and agree with the user’s likes and dislikes. This can be achieved by introducing an interactive environment, where every user's action would be written down and classified within a database. As this database would grow, the program would become more definite about the actions to take and the result would come closer to the user expectations. The more the program would be used, the more filtering would take place for the resulting buildings and the more aesthetically correct the solution would be for the specific user.

Although Neural Networks would be an excellent solution to the problem of Form - Finding, its realization, although tested, is hard to subsist at this time. New approaches are being done to bring this field to everyday life, where user's preferences will be easier to access and be learned.

Bibliography

(n.d.). Retrieved from An introduction to L-Systems: http://www.mh-portfolio.com/lsystems.html

Architekturzentrum Wien. (n.d.). Retrieved January 15, 2007, from Architekturzentrum Wien: http://www.azw.at/otherprojects/

arcspace.com. (n.d.). Retrieved March 12, 2007, from http://www.arcspace.com

Burry, M. (2003, March). Between Surface and Substance. *Surface Consciousness* , pp. 8-19.

Cheng, C.-W. (n.d.). *Genetic Housing Studio*. Retrieved January 3, 2007, from http://www.arch.columbia.edu/Students/Fall2003/Cheng.Chih-Wei/

DesignBoom. (n.d.). Retrieved March 13, 2007, from DesignBoom: http://www.designboom.com

Eisenman, P. (1999). *Diagram Diaries.* Universe Publishing.

Eric Owen Moss Architects. (n.d.). Retrieved 10 10, 2006, from Eric Owen Moss Architects: http://www.ericowenmoss.com

Foley, J. D. (1990). In *Computer Graphics: principles and practice, Second Edition* (p. 540). Addisson-Wesley Publishing Company, Inc.

Foster + Partners. (n.d.). Retrieved March 12, 2007, from http://www.fosterandpartners.com

Galinsky. (n.d.). Retrieved January 11, 2007, from Galinsky: http://www.galinsky.com/

Groboto. (n.d.). Retrieved May 3, 2007, from What it's all about: http://www.groboto.com/About.html

Guggenheim.ORG. (n.d.). Retrieved January 24, 2007, from Guggenheim.ORG: http://www.guggenheim.org/

Hadid, Z. (1998.). The Complete Buildings and Projects. In Z. Hadid, *The Complete Buildings and Projects* (pp. 110-115). Thames and Hudson.

Hemberg, M. (n.d.). *Genr8*. Retrieved February 12, 2006, from A Surface Design Tool: http://projects.csail.mit.edu/emergentDesign/genr8/index.html

Jahn, G. V. (1997). The Exploration of Physical Space in Architecture: New Design Elements Derived from. *Proceedings of The Second International Conference: Mathematics and Design 98* (pp. 207-214). The University of Basque Country.

Krawczyk, J. T. *Möbius Concepts in Architecture.* Chicago, IL, 60616, USA: College of Architecture.

Lindenmayer, A. (1968). Mathematical Models for Cellular Interactions in Development, Parts I and II. *Journal of Theorethical Biology,18* , 280-315.

LyndyHOP. (n.d.). Retrieved March 24, 2007, from http://www.lab4web.com/chelmiger/lyndyhop/lh_start.html

Martin Hemberg, U.-M. O. (n.d.). Genr8: Architects' Experience with an Emergent.

Ochoa, G. (n.d.). *An Introduction to Lindenmayer Systems*. Retrieved March 13, 2007, from http://www.biologie.uni-hamburg.de/b-online/e28_3/lsys.html

Padovan, R. (1999). In R. Padovan, *Proportion: Science, Philosophy, Architecture* (pp. 320-326). Taylor & Francis.

PC Magazine. (n.d.). Retrieved May 1, 2007, from PCMAG.COM: http://www.pcmag.com/encyclopedia_term

Perry, C. (1993,). "In The Edge of Science: The Role of the Artist's Intution in Science" in The Visual Mind: Art and Mathematics. In C. Perry, *"In The Edge of Science: The Role of the Artist's Intution in Science" in The Visual Mind: Art and Mathematics* (pp. 59-61).

Prusinkiewicz. Development Models of Herbaceous Plants for Computer Imagery Purposes. *Proceedings of the ACM Siggraph 88*, (pp. 141-150).

Reffye. Plant Models Faithful to Botanic Structure and Development. *Proceedings of the ACM Siggraph 88*, (pp. 151-158).

Smith, A. R. (1984). Plants, Fractals and Formal Languages. *Proceedings of ACM Siggraph*, (pp. 1-10).

Stiny. (1978). *Algorithmic Aesthetics: Computer Models for Criticism and Design in the Arts.* Berkeley: University of California Press.

The Great Buildings Collection. (n.d.). Retrieved November 11, 2006, from The Great Buildings Collection: http://www.greatbuildings.com

Watt, A. (1993). *3D Computer Graphics, Second Edition.* Addisson-Wesley Publishing Company, Inc.

Wired New York. (n.d.). Retrieved February 13, 2007, from Wired New York: http://www.wirednewyork.com

Wolfram Mathworld. (n.d.). Retrieved June 6, 2007, from http://mathworld.wolfram.com

Zaha Hadid Architects. (n.d.). Retrieved March 15, 2007, from Zaha Hadid Architects: http://www.zaha-hadid.com/

APPENDIX

Examples of Parametric Form Finding

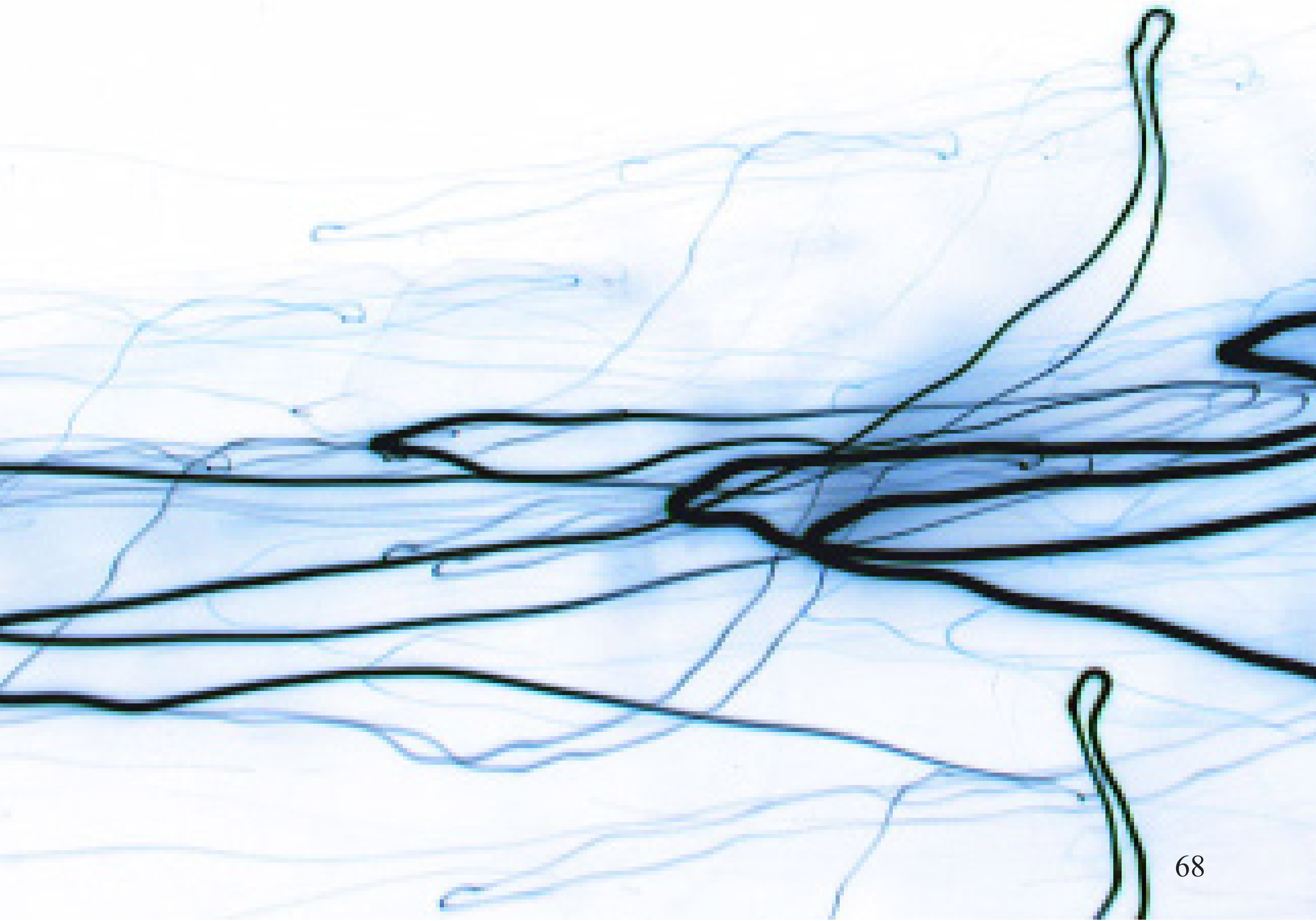
1st Example

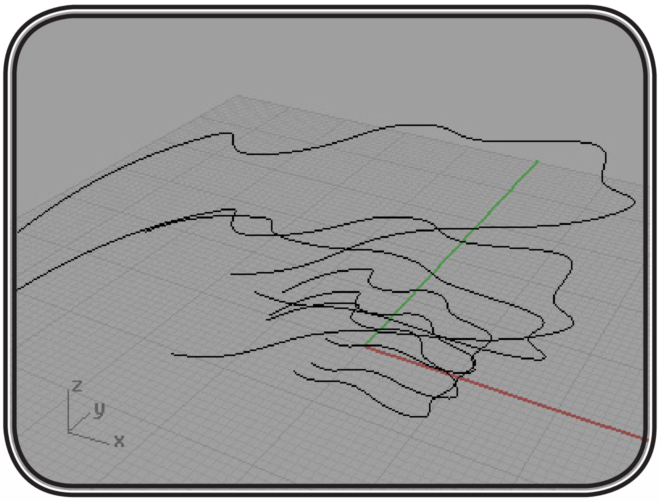
Criteria influencing the Form

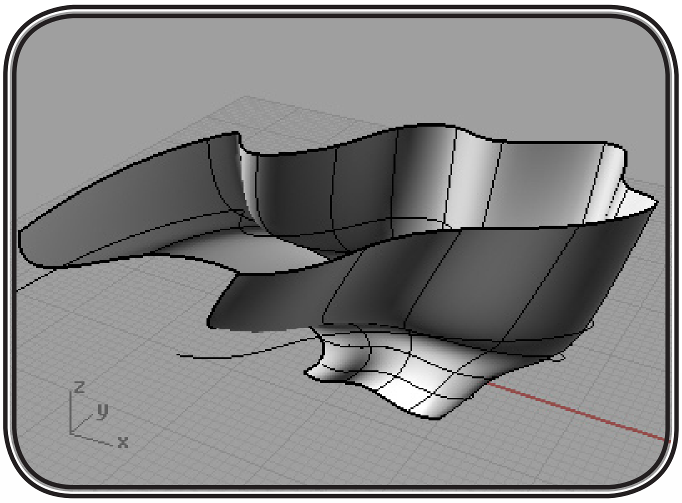
- 1 two-dimensional Basic Curve
- 15 m Total Height allowed
- 5 Storeys
- 3 m desired Storey Height
- Rectangular Site Curve 20 m x 15m

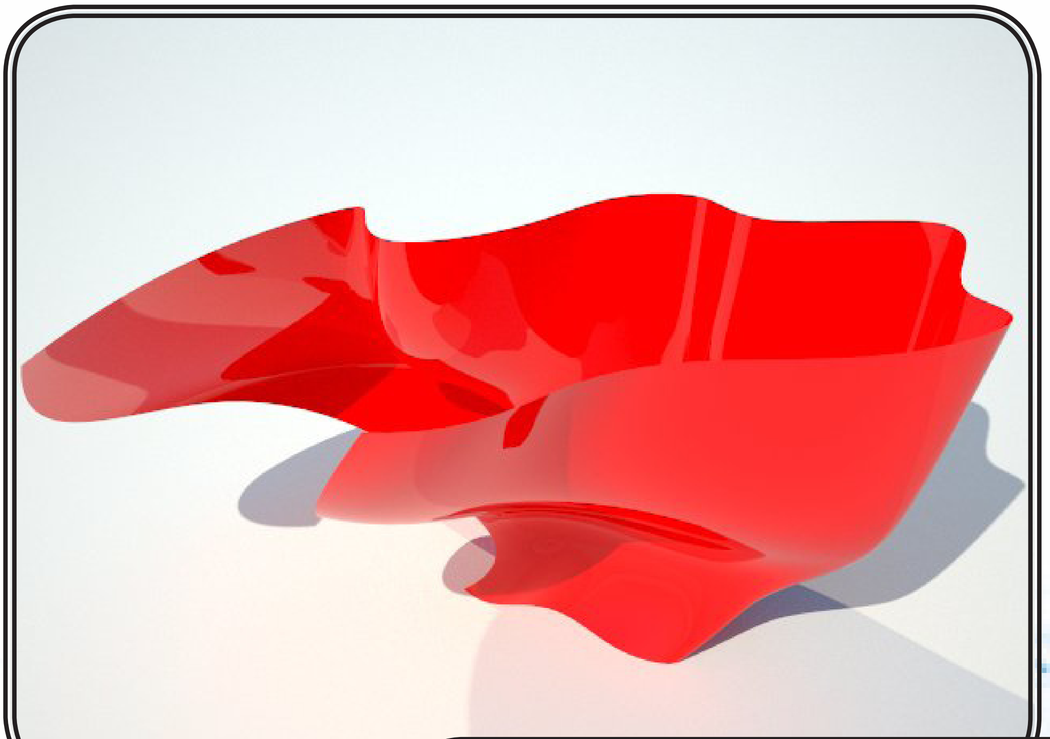
Architectural Data

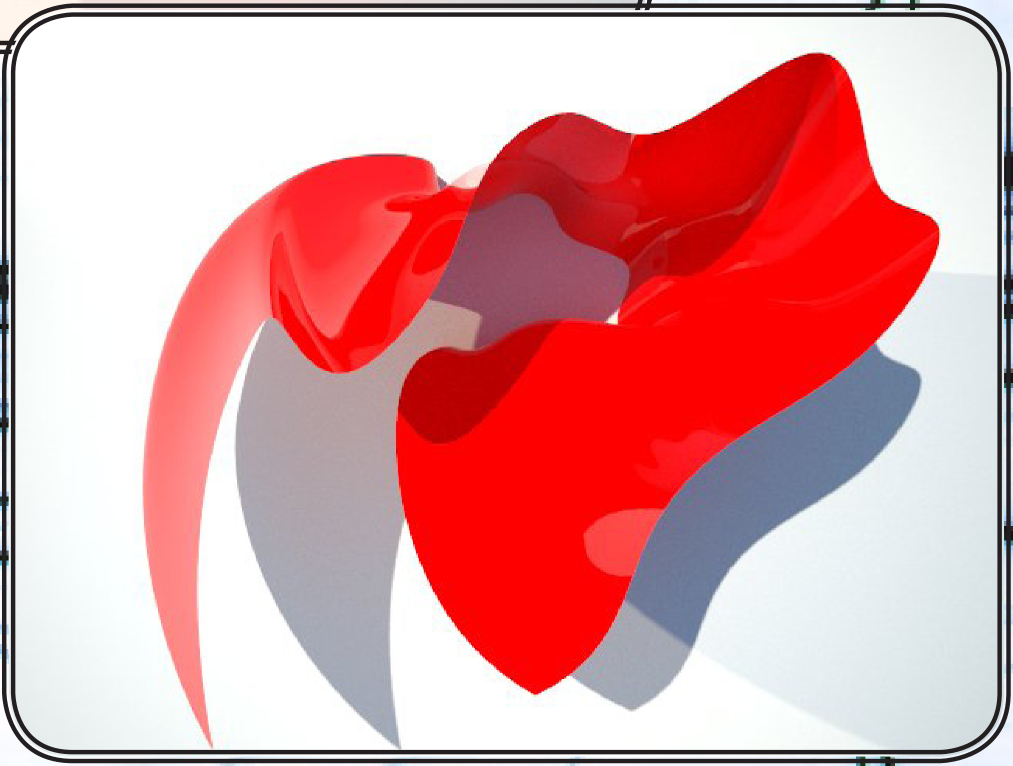
- Typical example of Free-Form Architecture
- Buildable Form by applying Frames to the resulted Wireframe
- Additional colums to hold the non-supported parts.




z
y
x


z
y
x





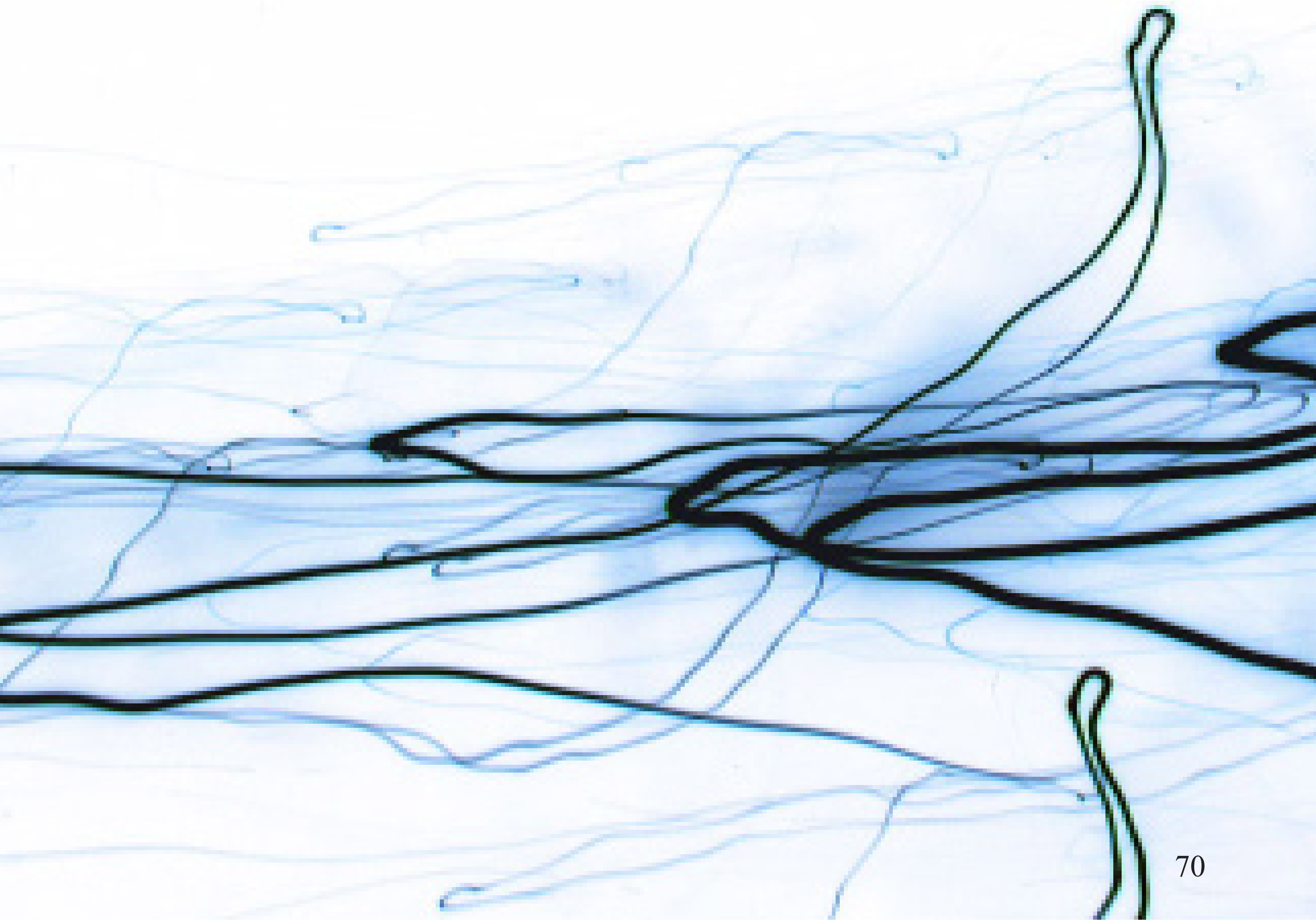
2nd Example

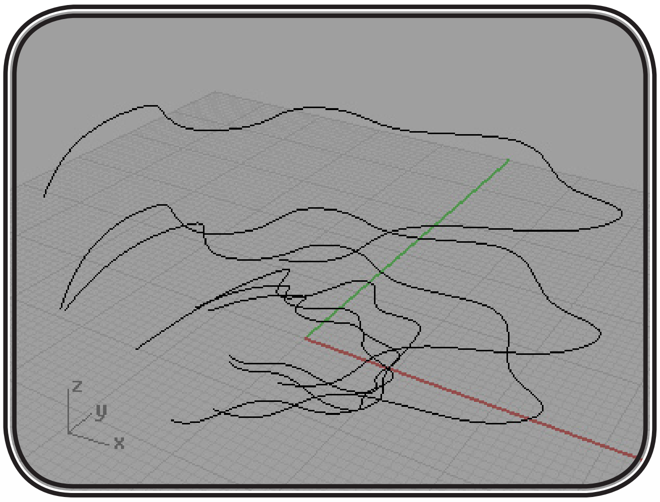
Criteria influencing the Form

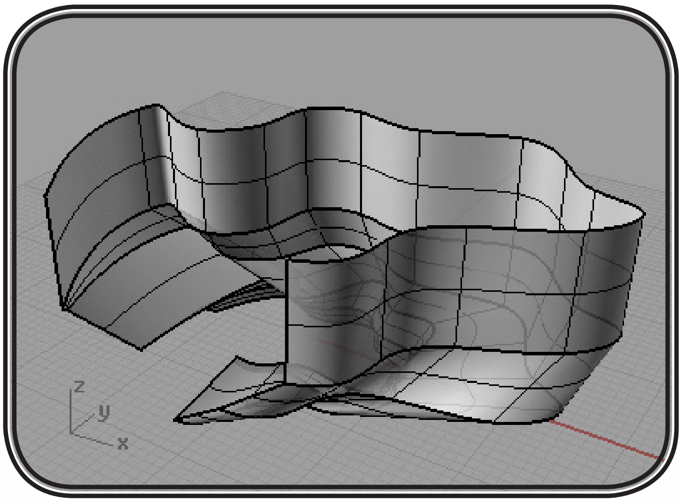
- 1 two-dimensional Basic Curve (same curve as in the 1st Example)
- 20 m Total Height allowed
- 5 Storeys
- 3 m desired Storey Height
- Rectangular Site Curve 20 m x 15m

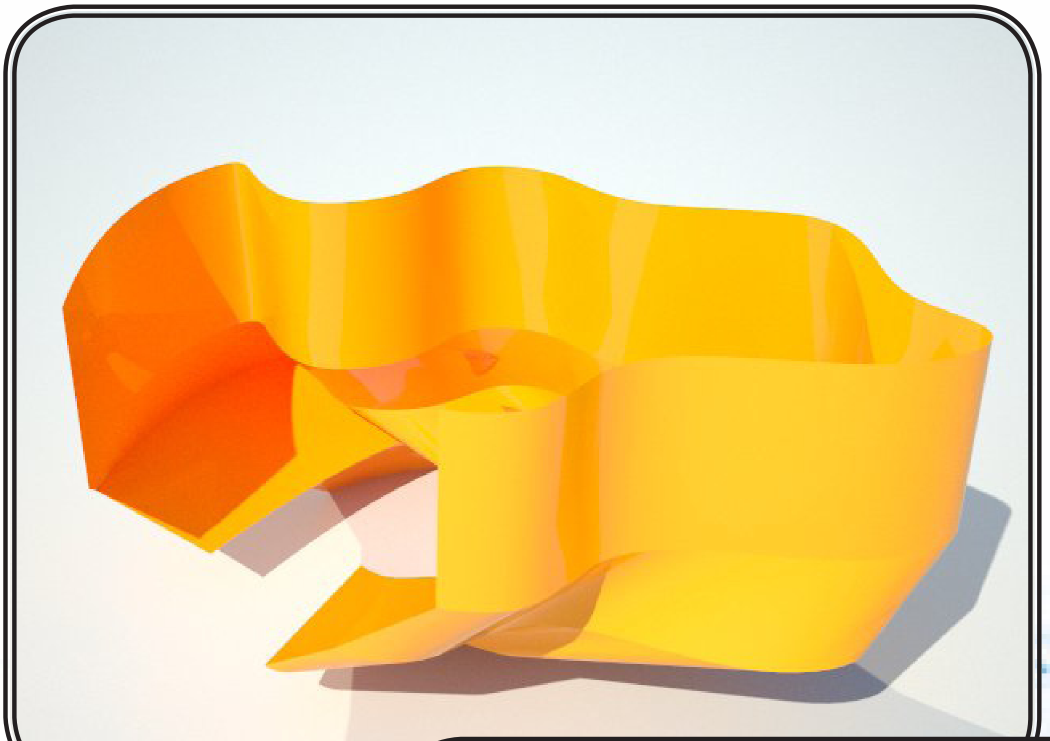
Architectural Data

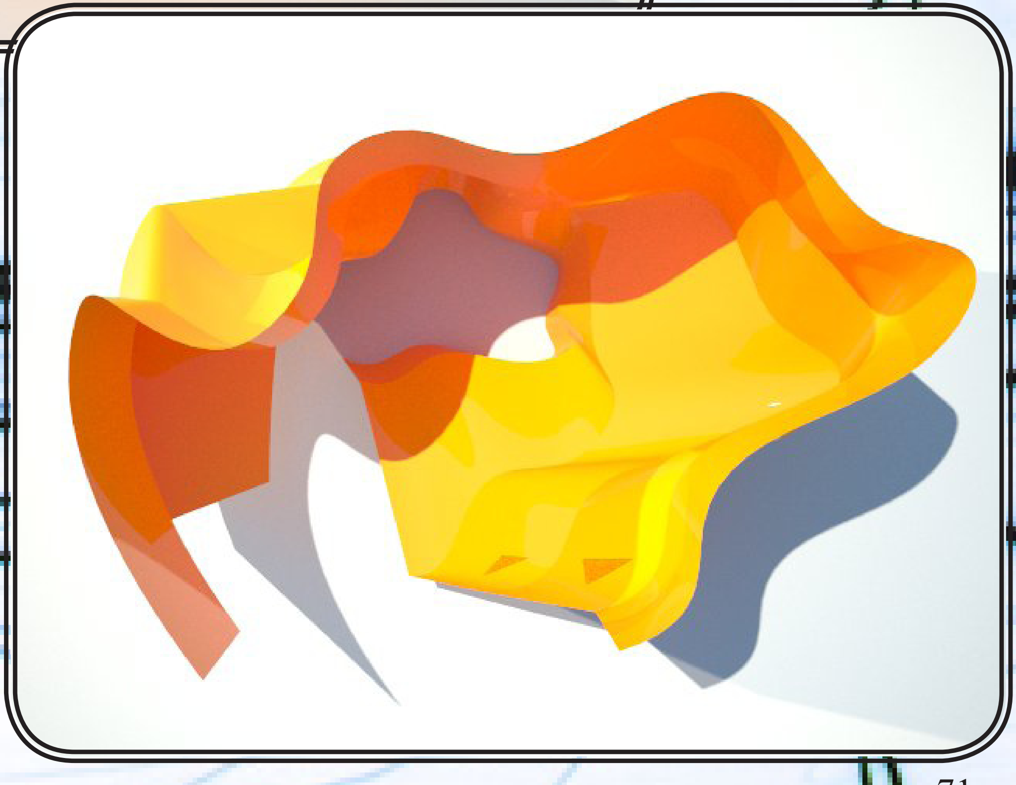
- Typical example of Free-Form Architecture
- Buildable Form by applying Frames to the resulted Wireframe
- Additional colums to hold the non-supported parts.




z
y
x


z
y
x





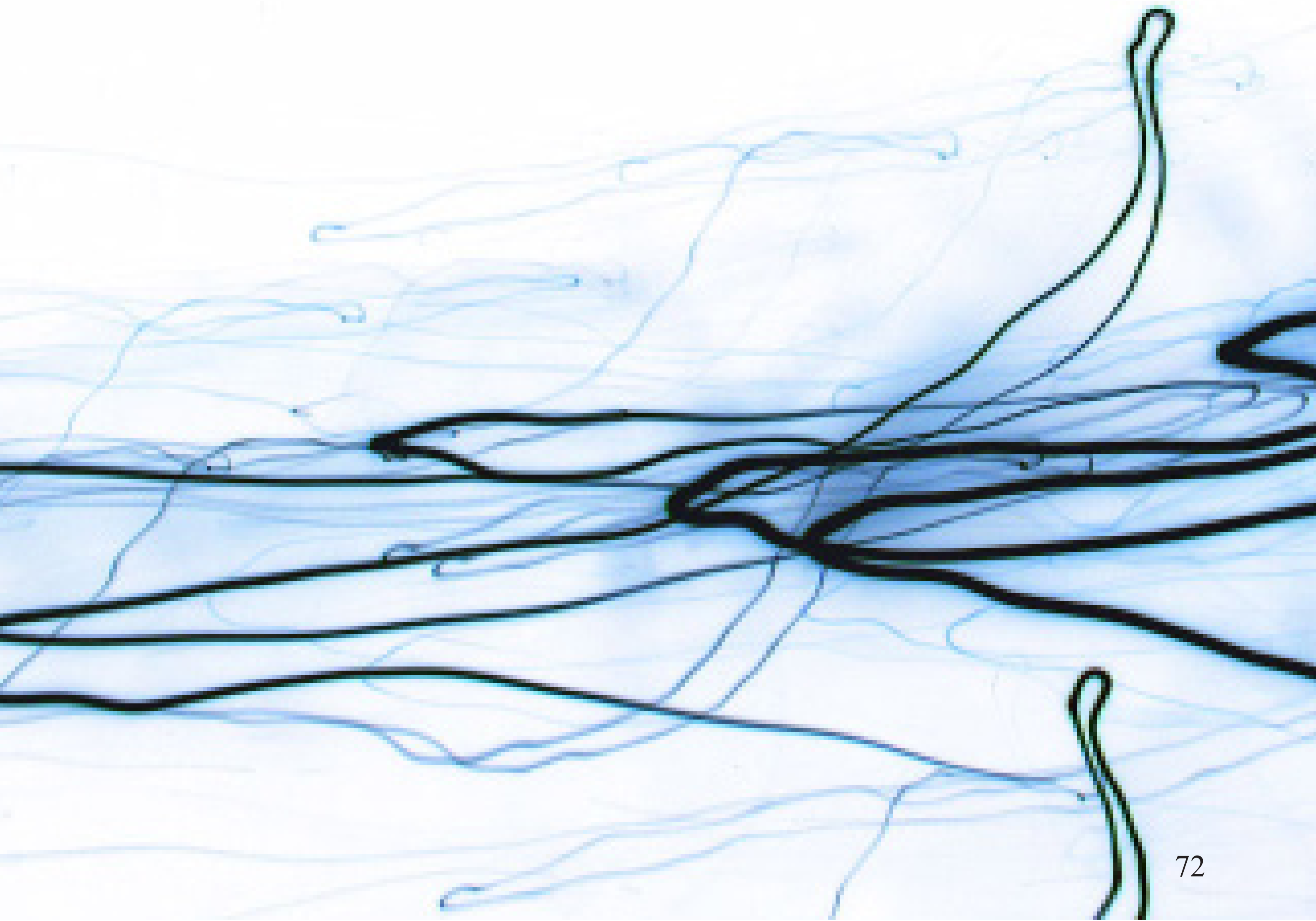
3rd Example

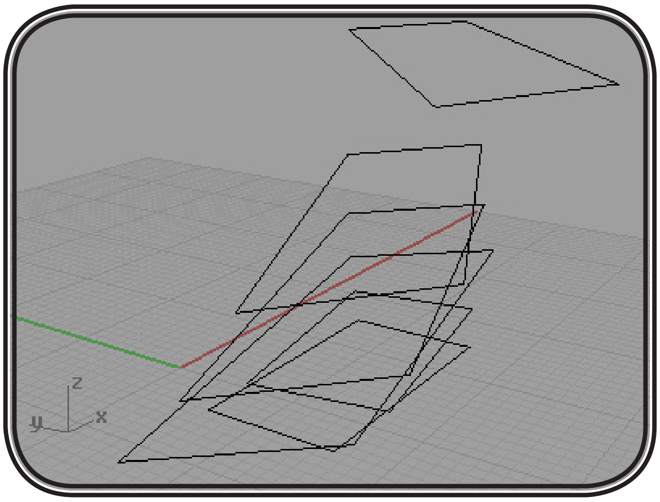
Criteria influencing the Form

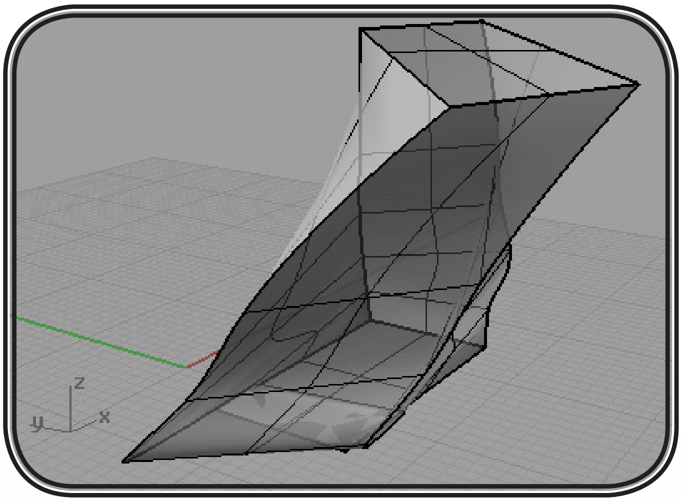
- 1 two-dimensional Basic Curve
- 30 m Total Height allowed
- 5 Storeys
- 3 m desired Storey Height
- Rectangular Site Curve 20 m x 10m

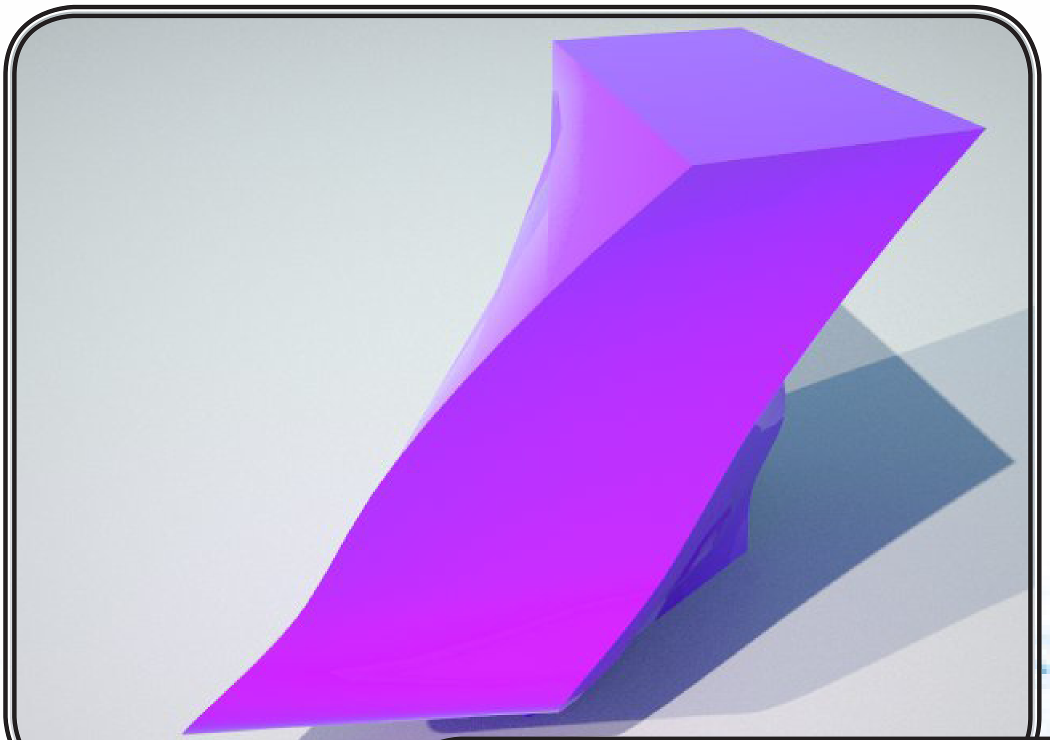
Architectural Data

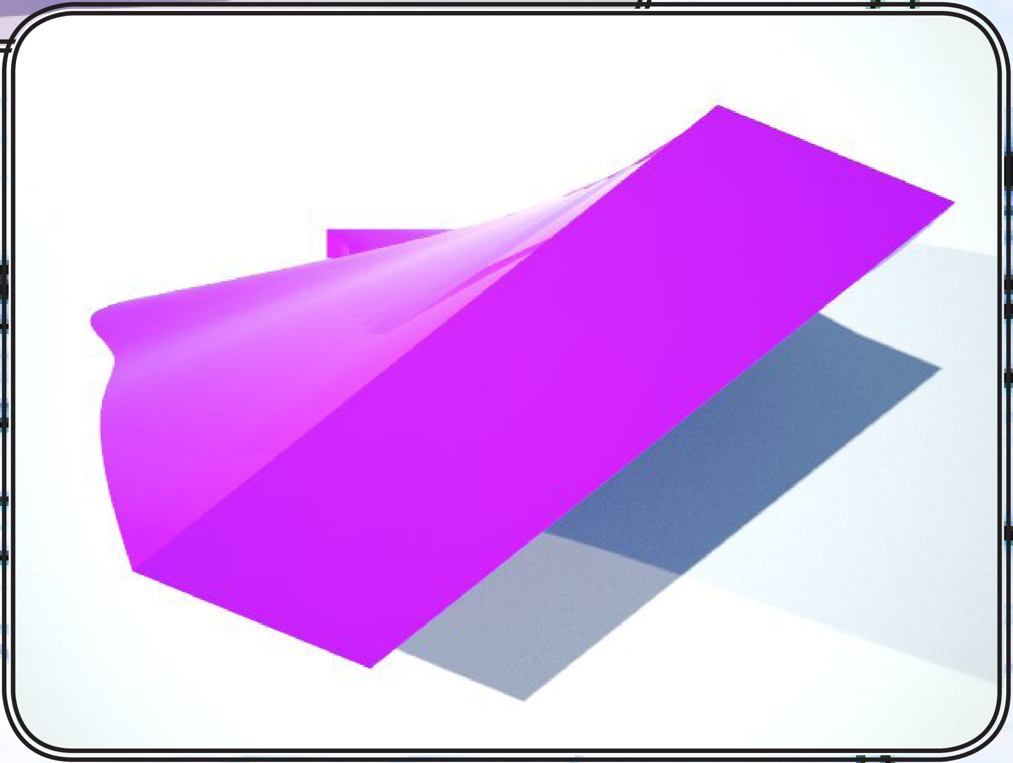
- Typical example of Free-Form Architecture (Twist Effect)
- Buildable Form by applying Frames to the resulted Wireframe
- Additional colums to hold the non-supported parts.




z
y
x


z
y
x





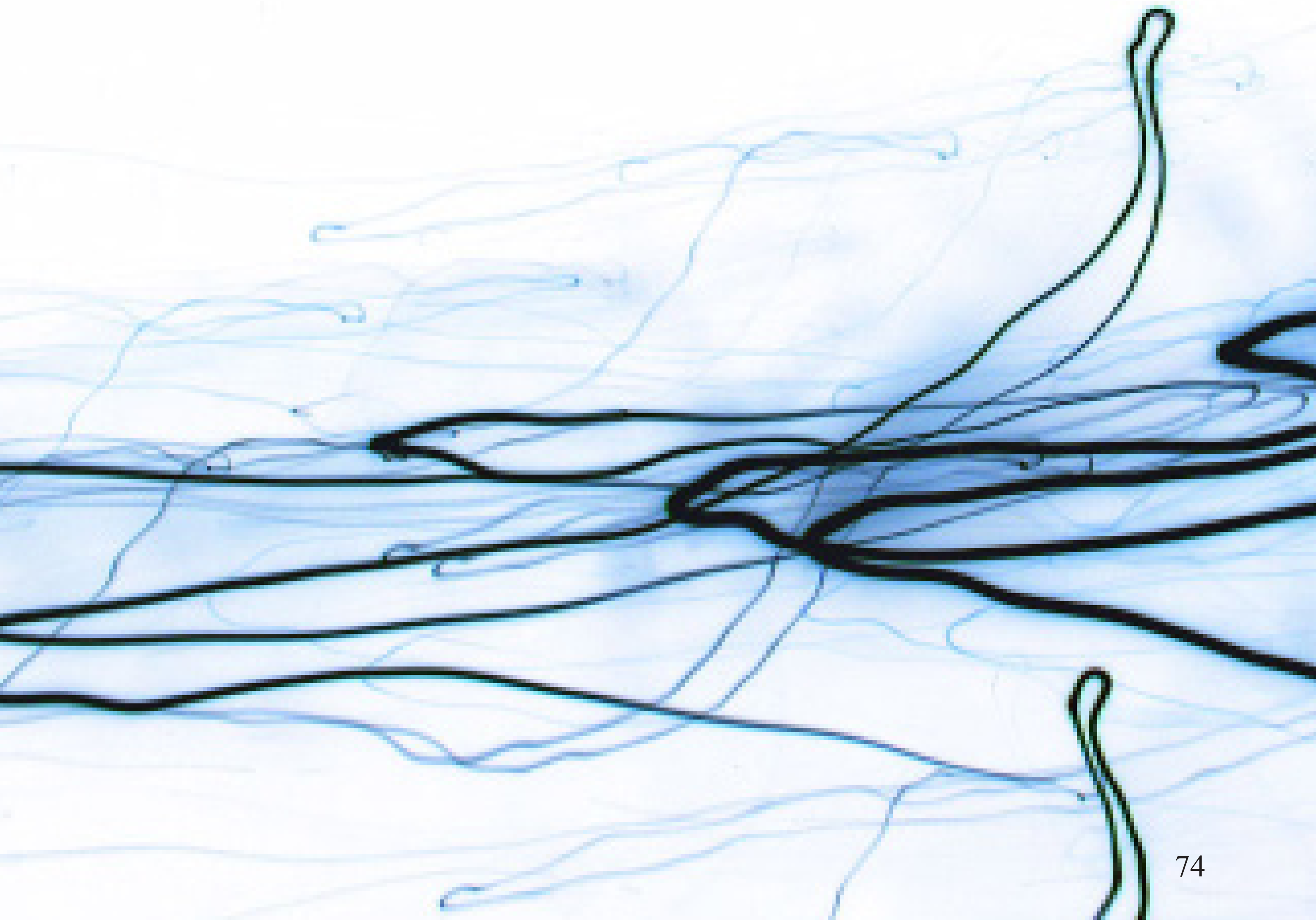
4th Example

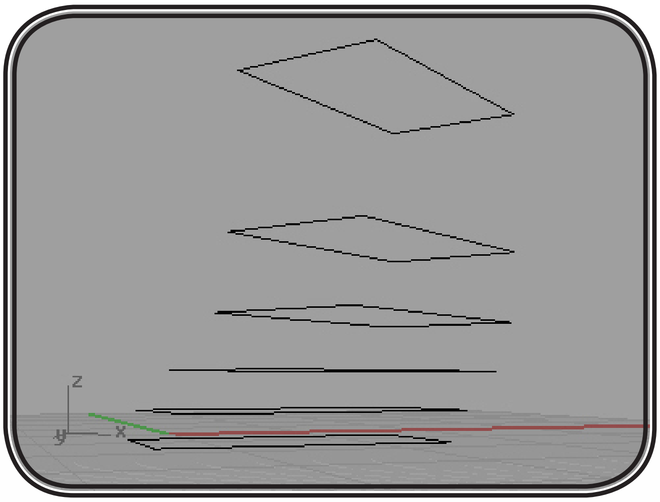
Criteria influencing the Form

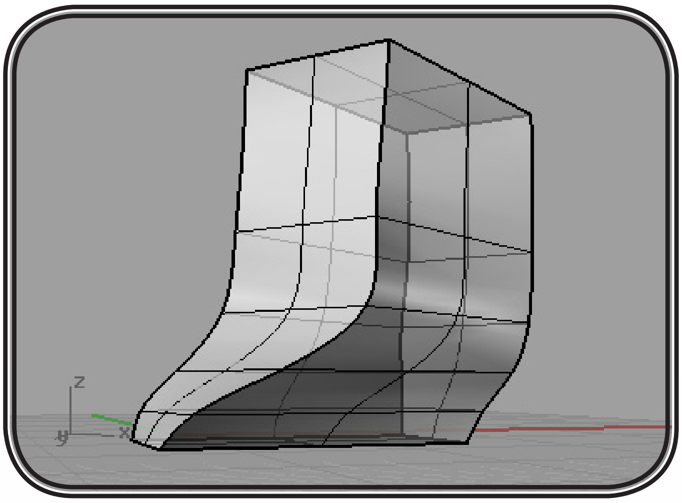
- 1 two-dimensional Basic Curve (same curve as in the 3rd Example)
- 45 m Total Height allowed
- 5 Storeys
- 6 m Maximum Storey Height
- Rectangular Site Curve 20 m x 10m

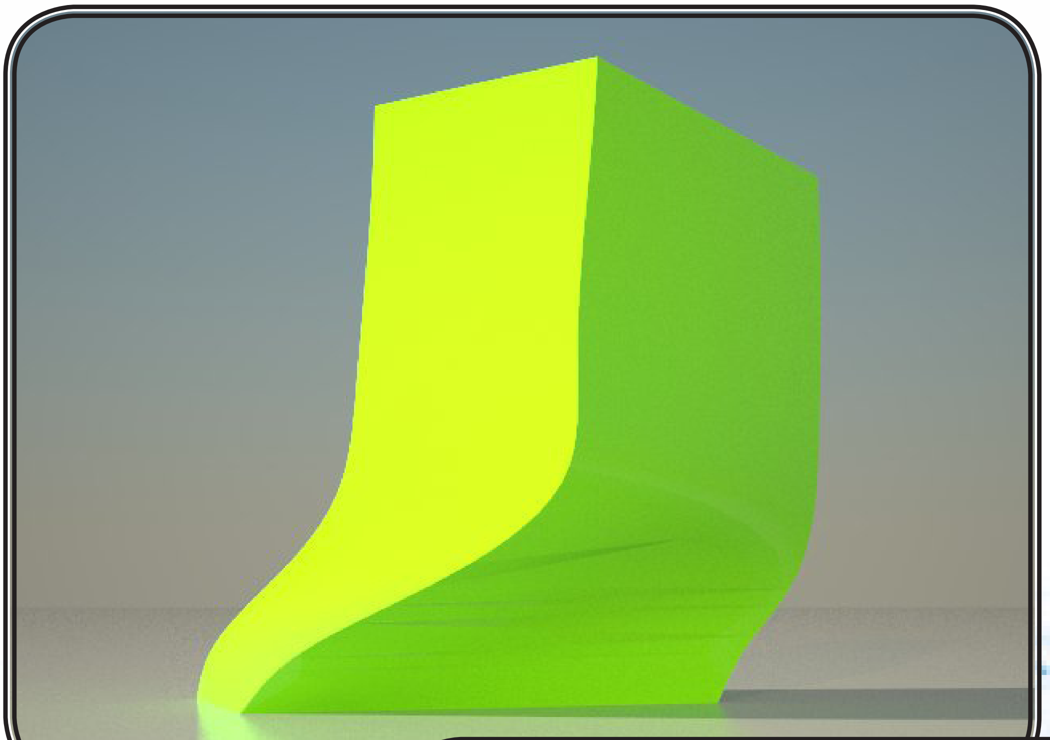
Architectural Data

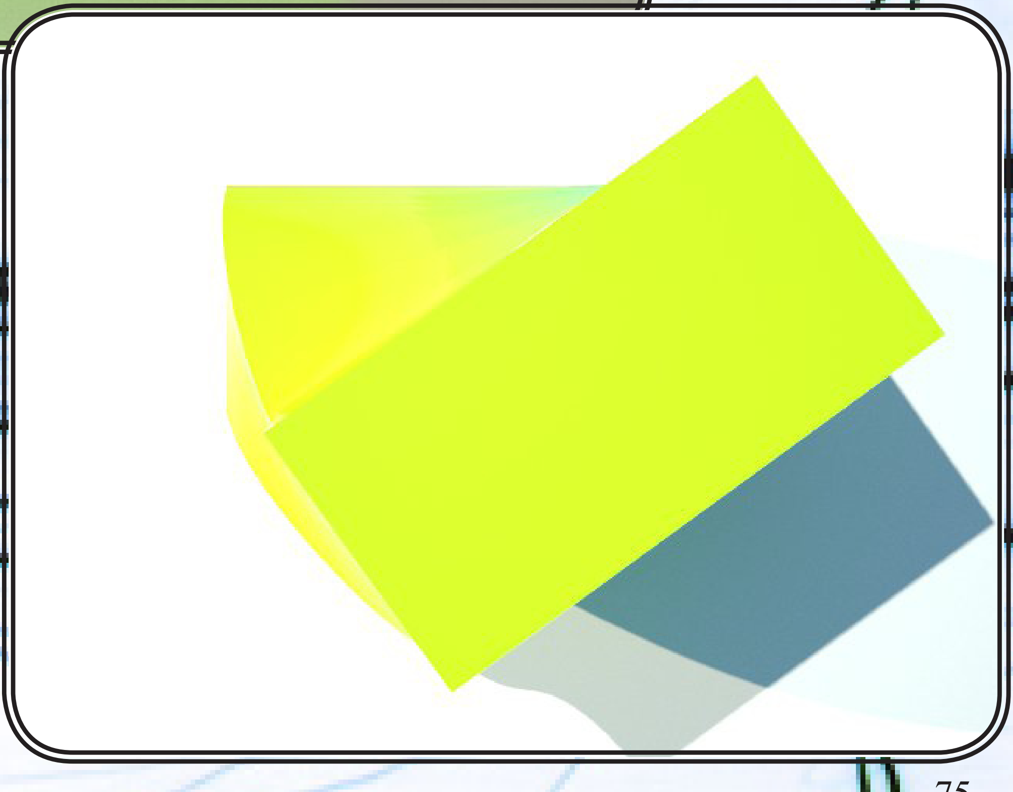
- Typical example of Free-Form Architecture (Twist Effect)
- Buildable Form by applying Frames to the resulted Wireframe
- Additional colums to hold the non-supported parts.




z
y
x


z
y
x





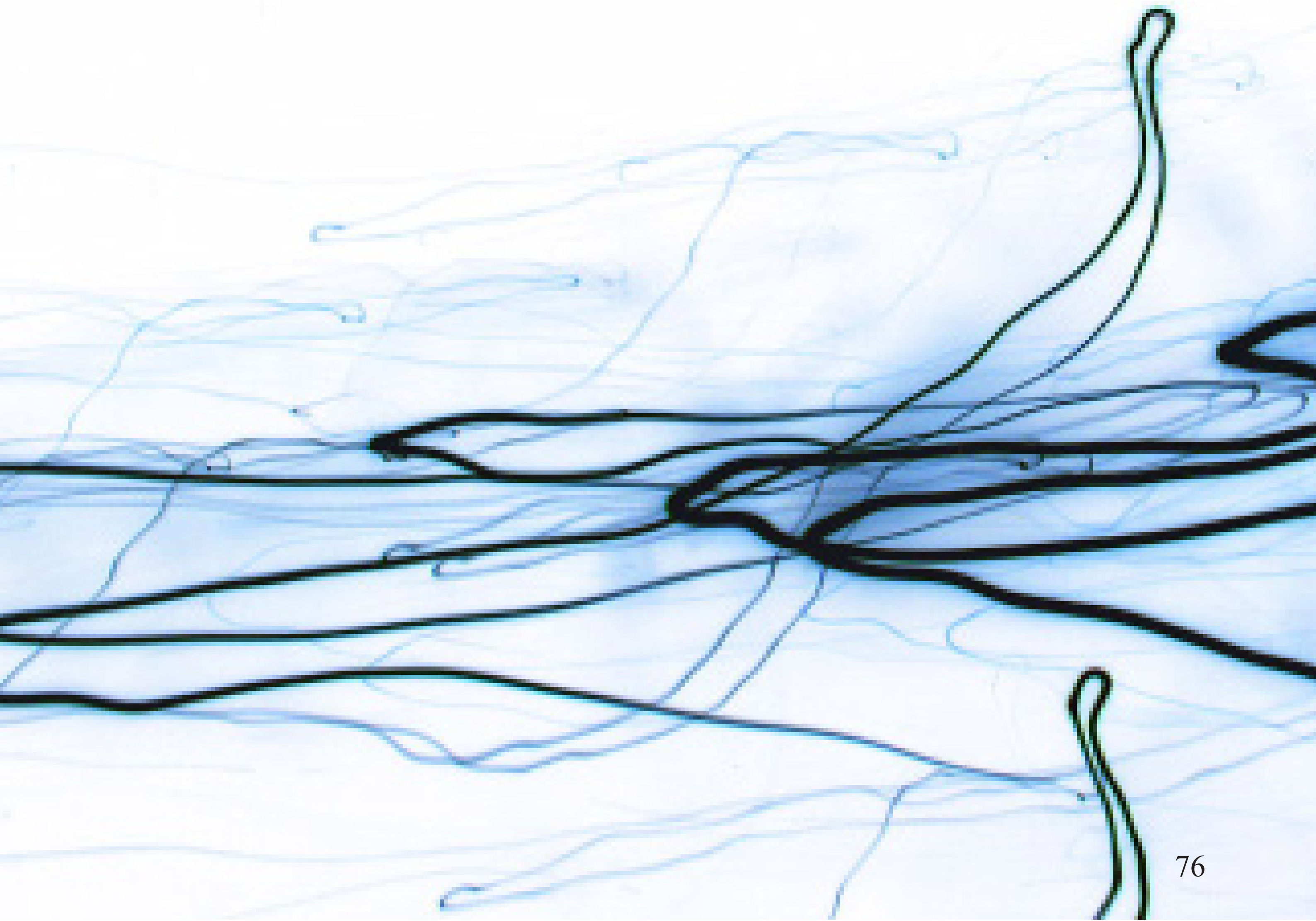
5th Example

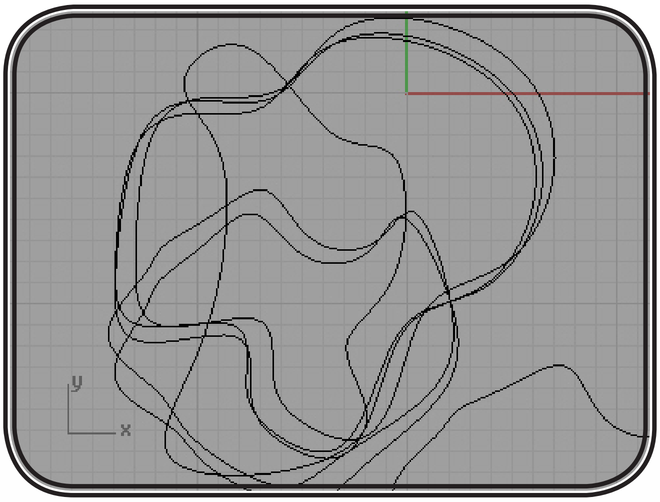
Criteria influencing the Form

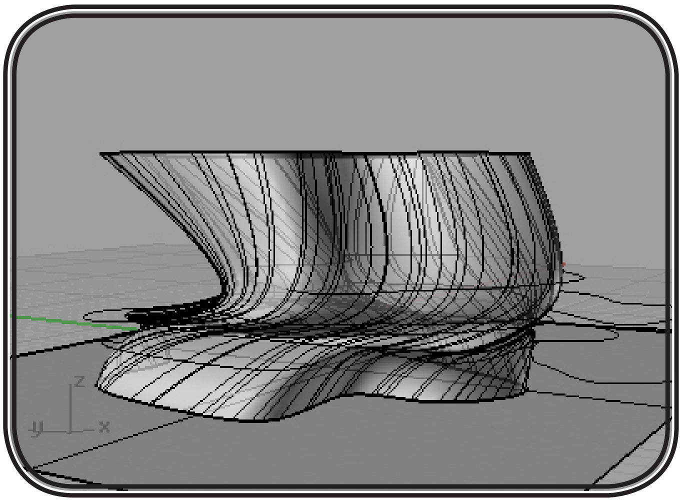
- 3 two-dimensional Basic Curves
- 30 m Total Height allowed
- 5 Storeys
- 4 m desired Storey Height
- Rectangular Site Curve 20 m x 10m

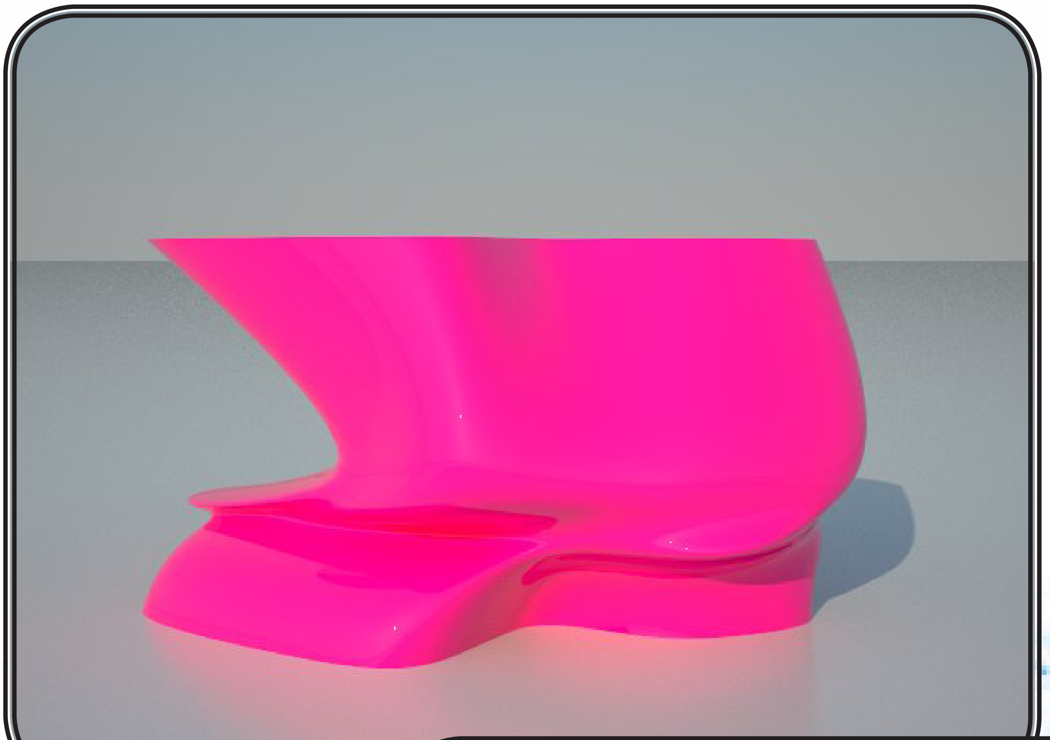
Architectural Data

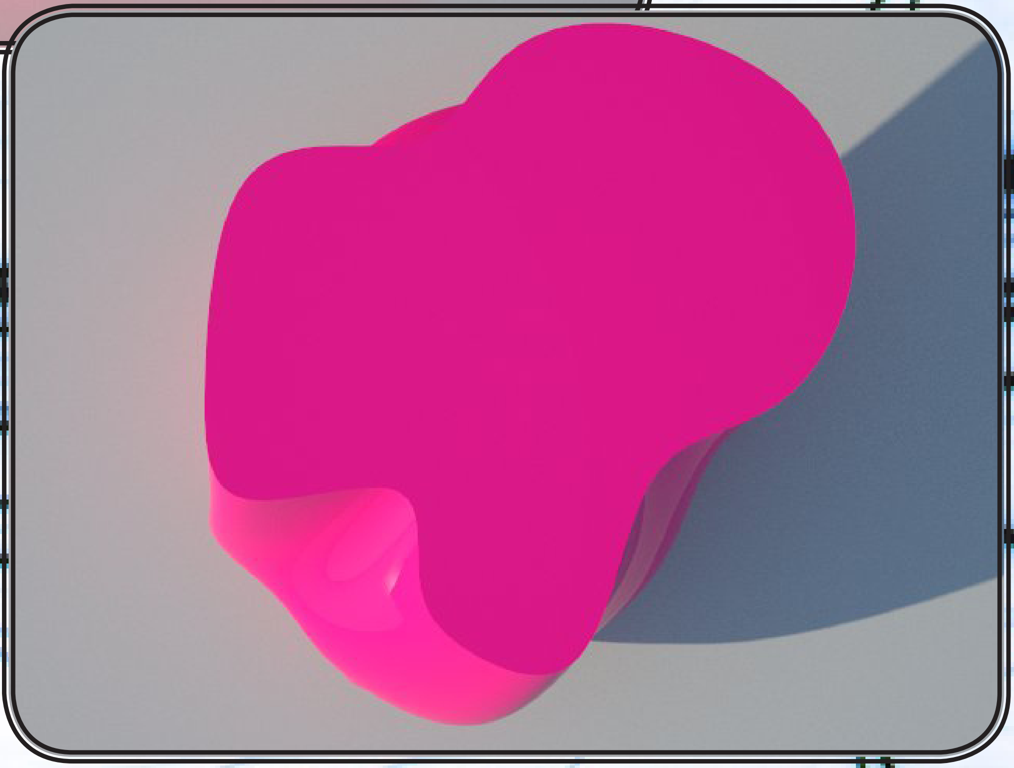
- Typical example of Free-Form Architecture
- Buildable Form by applying Frames to the resulted Wireframe
- Additional colums to hold the non-supported parts.




y
x


z
y
x





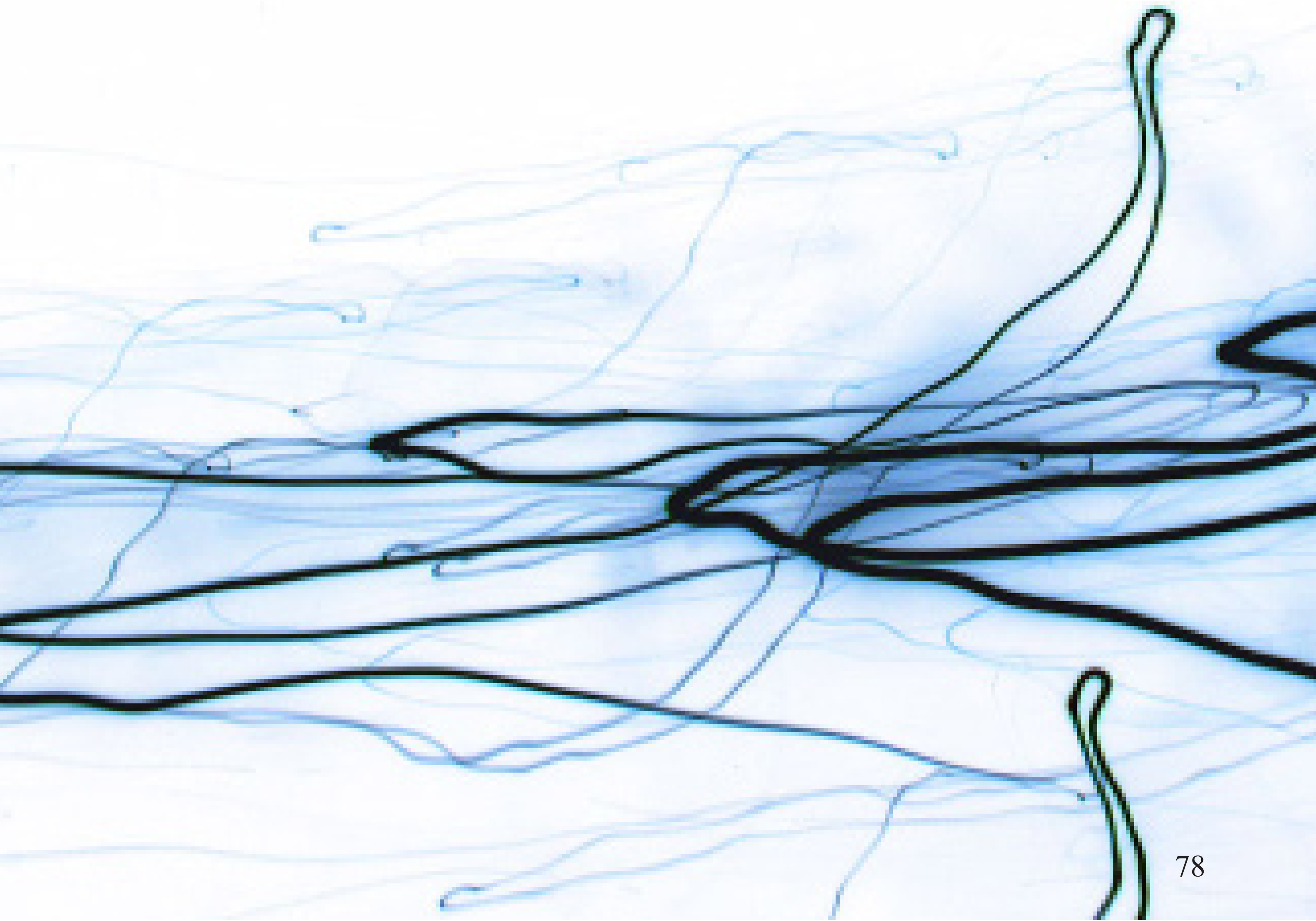
6th Example

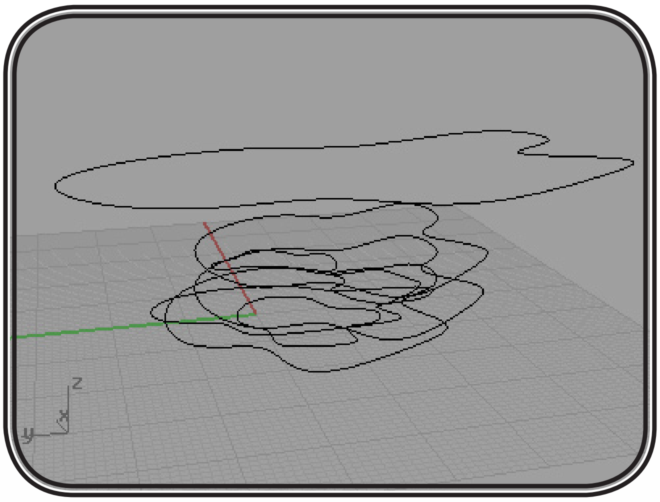
Criteria influencing the Form

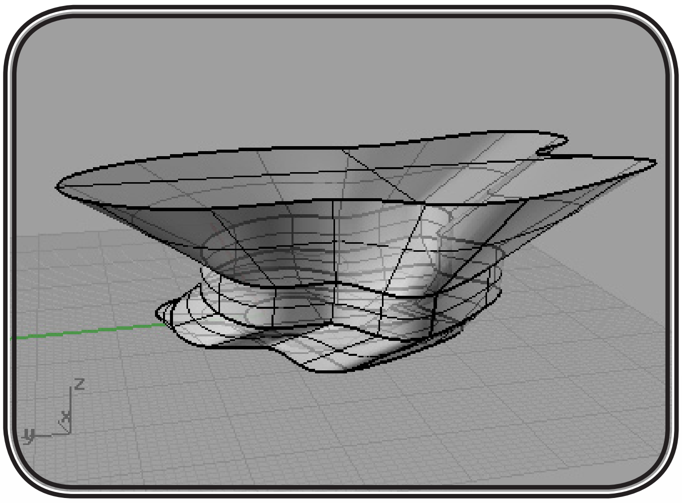
- 3 two-dimensional Basic Curves (same curves as in the 3rd Examples)
- 30 m Total Height allowed
- 5 Storeys
- 4 m desired Storey Height
- Rectangular Site Curve 20 m x 10m

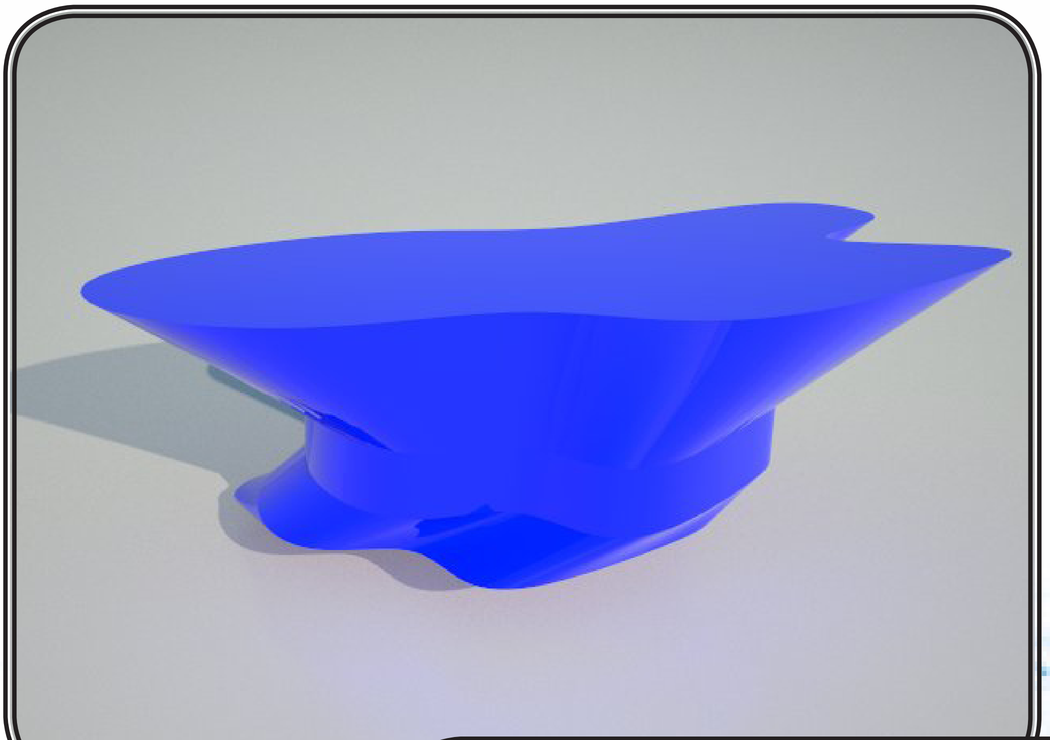
Architectural Data

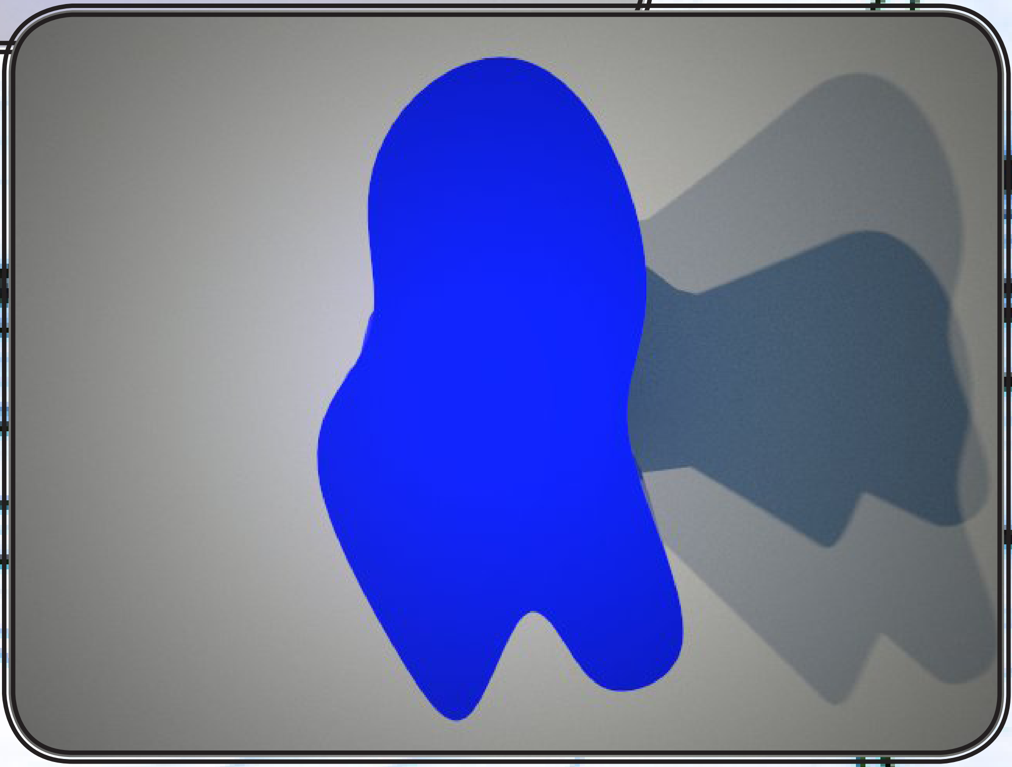
- Typical example of Free-Form Architecture
- Buildable Form by applying Frames to the resulted Wireframe
- Additional colums to hold the non-supported parts.




z
y
x


z
y
x





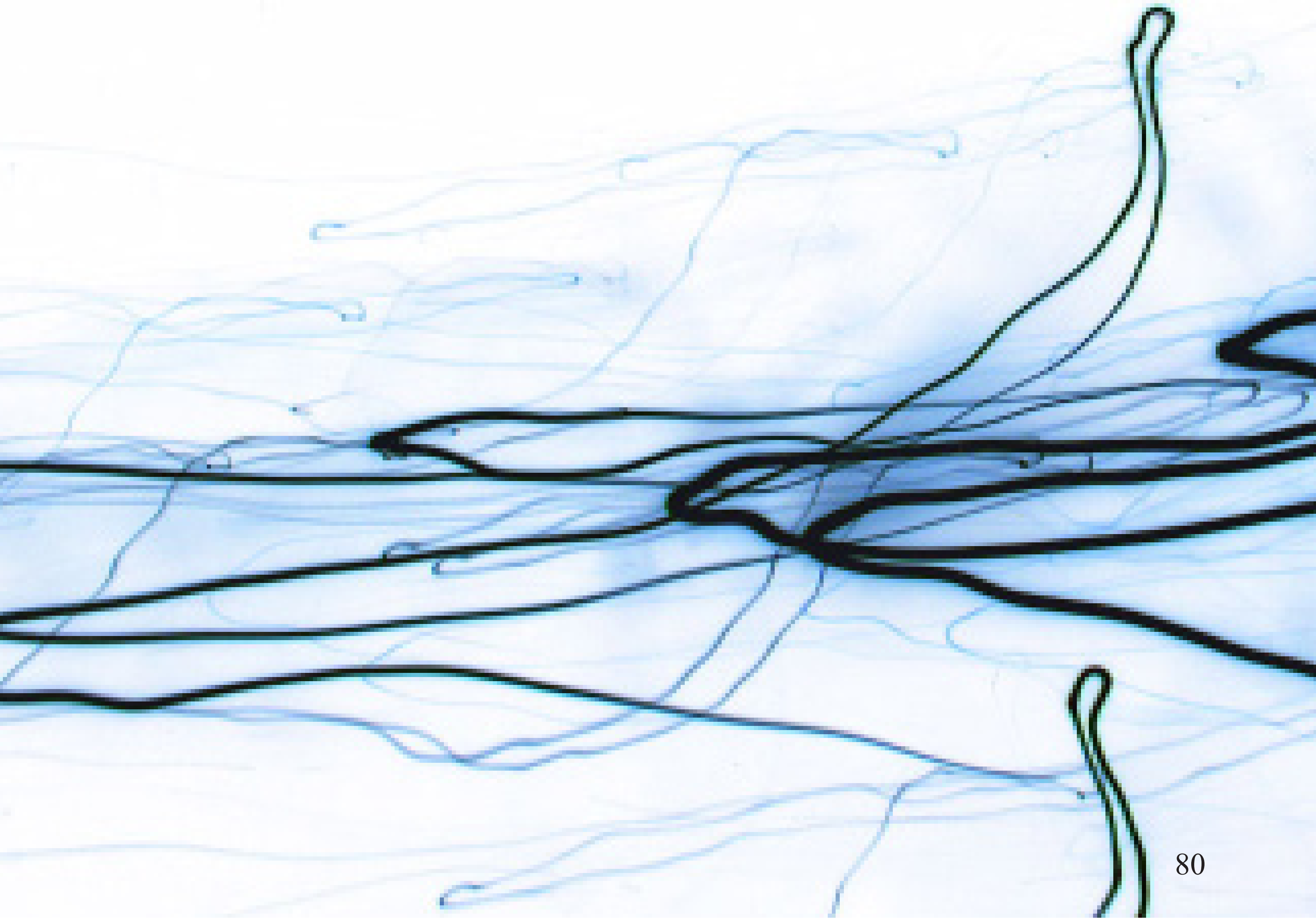
7th Example

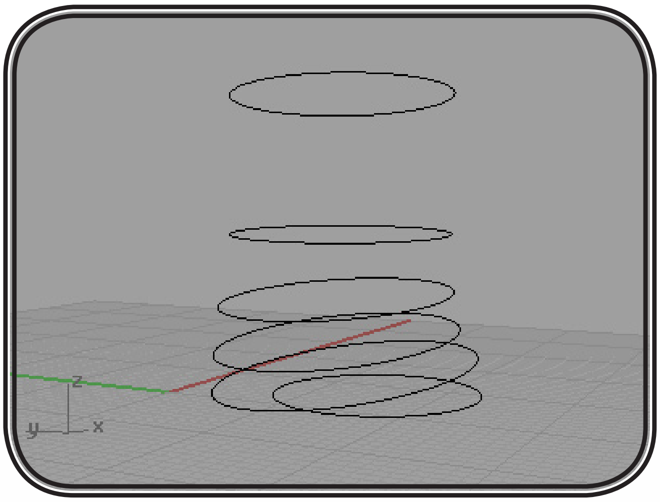
Criteria influencing the Form

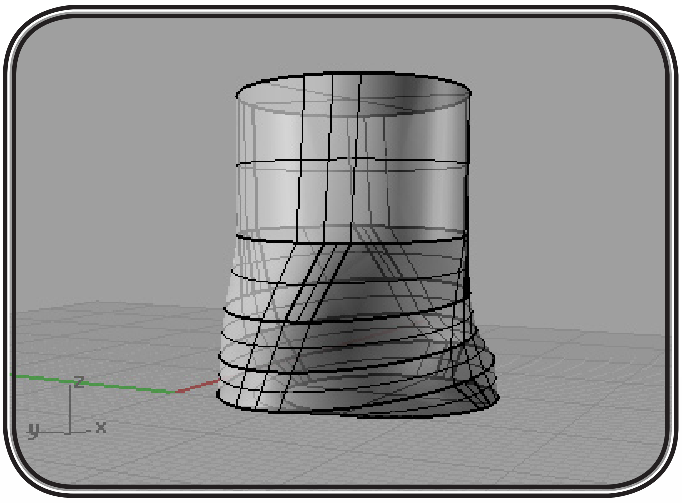
- 1 elliptical two-dimensional Basic Curve
- 45 m Total Height allowed
- 5 Storeys
- 7 m Maximum Storey Height
- Rectangular Site Curve 15 m x 10m

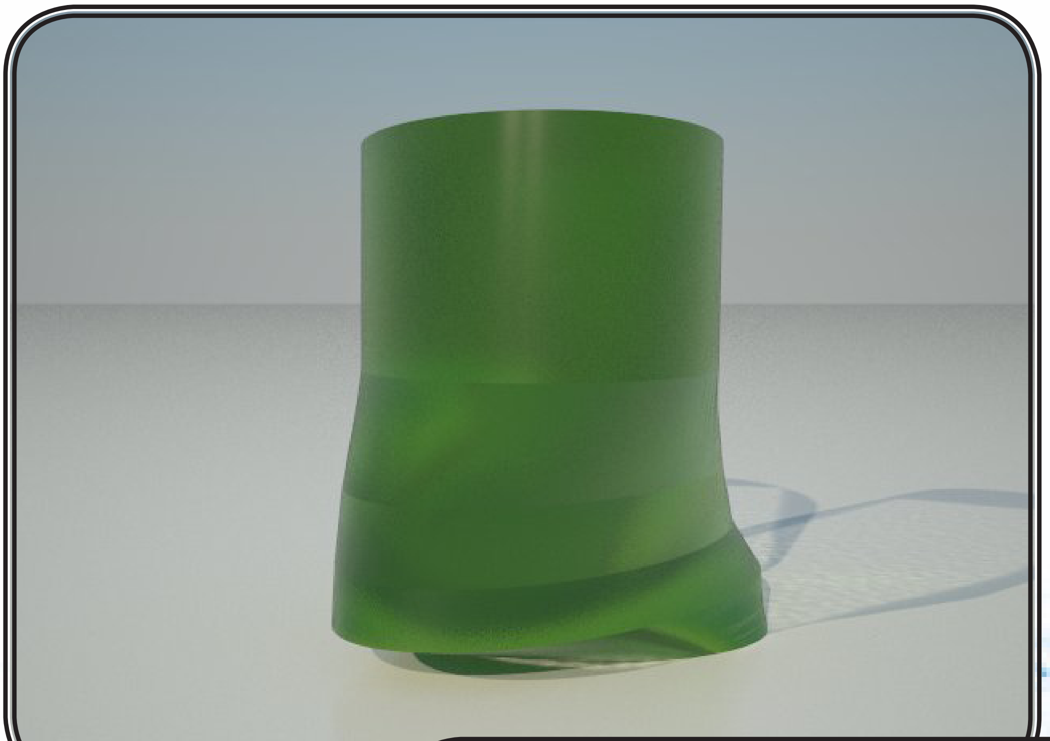
Architectural Data

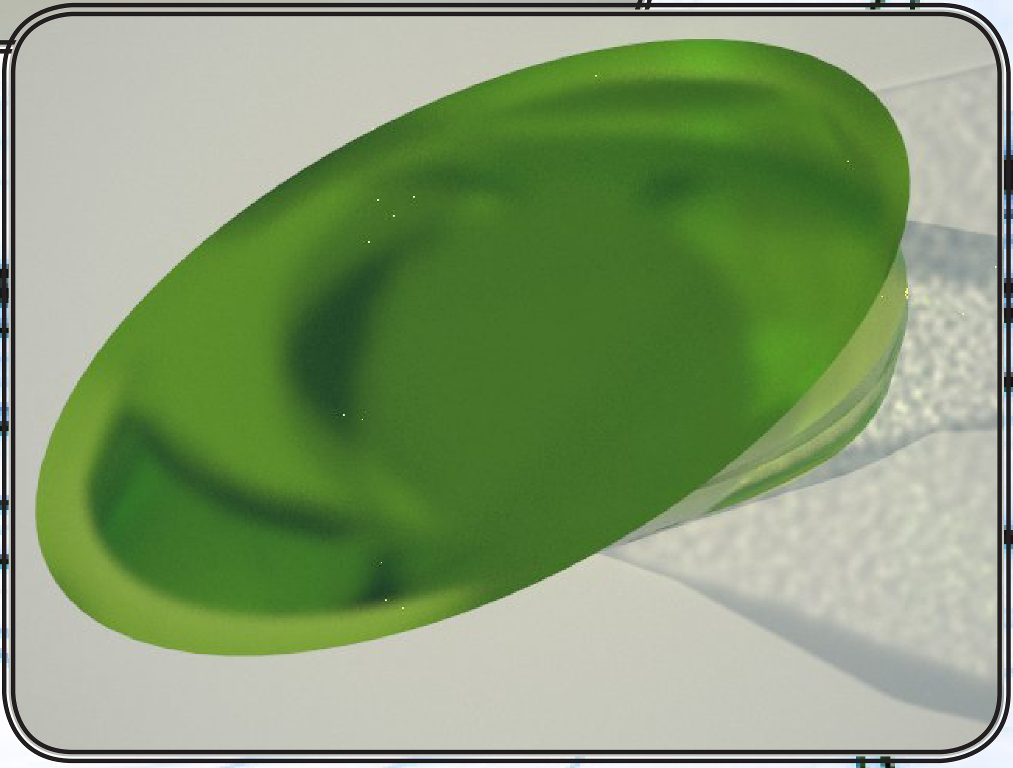
- Typical example of Free-Form Architecture
- Buildable Form by applying Frames to the resulted Wireframe
- Additional colums to hold the non-supported parts.




z
y
x


z
y
x





Short discussion of the thesis

The thesis focuses on a self-chosen topic that is truly interdisciplinary (Parametric Form Finding: architecture and computer graphics). The elaboration of the naive idea is solid, focusing on approaches that have been in existence since the early 1970ies and can be combined to produce architectural form: Given a number of basic splines to start from, the CAD package can now suggest building designs, taking into account previously entered constraint such as total area allowed to build and total allowed building volume. At all times, the approach leaves the aesthetics to the user (thereby avoiding the need to take up form aesthetics into the related research). The tool fits well into the architectural workflow, where design alternatives are subsequently elaborated and discarded. A shorter iteration time between prototypical forms can lead to faster and better work, especially when considering that architects need to participate in a lot of design contests before getting a reputation in the field. Also interesting to note is that the (artistic, non-scientific) design of a form can be reverse-engineered into a truly formal composition of basic shapes and transformations, as was shown for the buildings in the analysis section. This embeds the thesis into a wider body of works that try to extend the definition of what science is to the wider field of design.