Material Systems
- a design approach

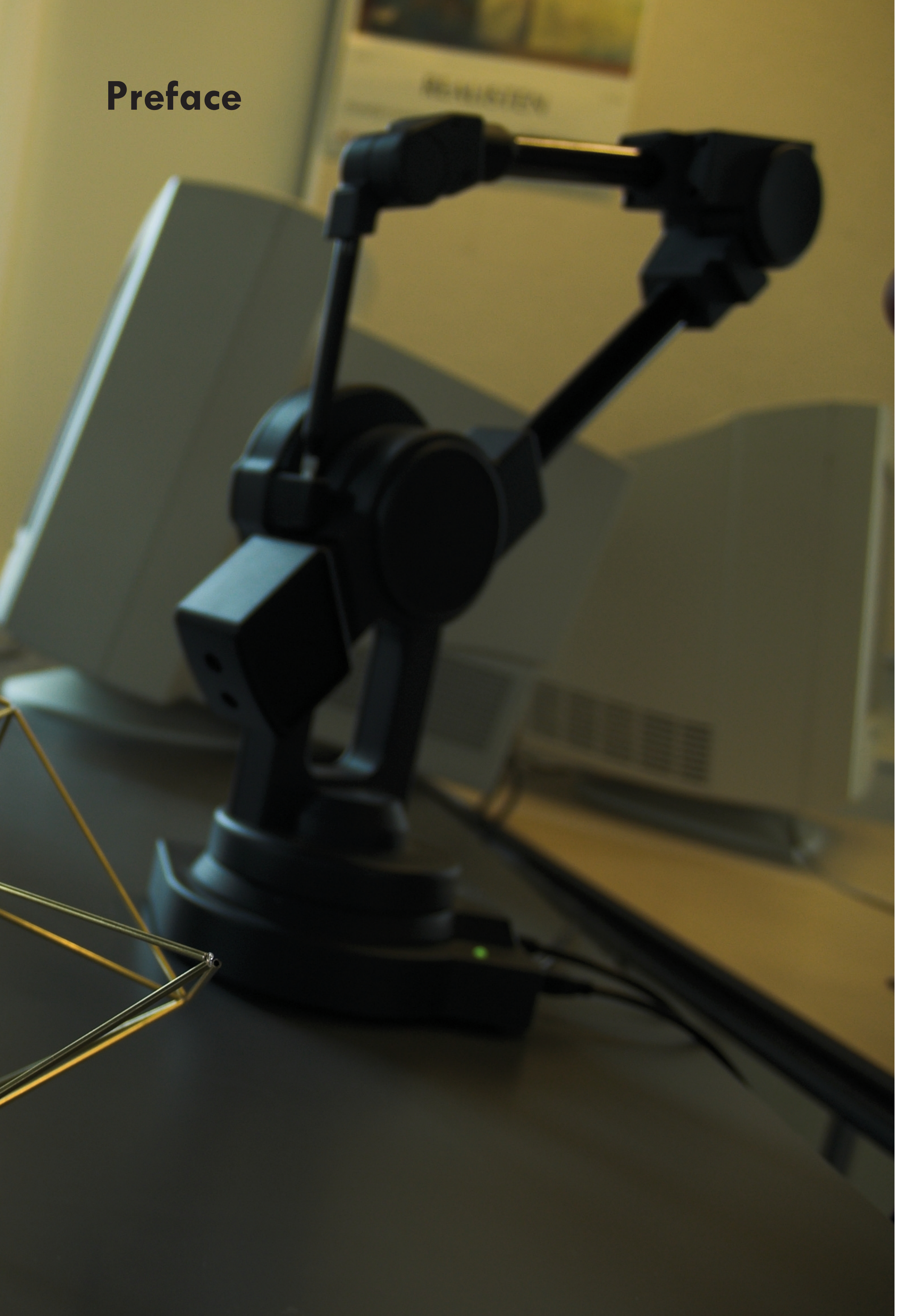
Master Thesis, 10th semester
Digital Design (M.Sc. Eng.)
Faculty of Architecture & Design, Aalborg University

Copies: 6
Pages: 105

[Supervisors:]
Poul Henning Kirkegaard, Assoc.Professor, Dep. of Civil Engineering
Michael Mullins, Architect MAA, Assoc.Professor, Ph.D.

Contents

Preface



In the end of the 20th century and the beginning of the 21st century, the applications of computers have in many aspects redefined architecture. From a manually driven tool-based design and practice profession, architecture has been transformed into a computer-driven form-based design practice. Many designers today use computers as an advanced tool for running programs that enable them to produce sophisticated forms and for graphically representing these design outcomes. This new type of architecture produced by the combination of new architectural efforts and digital technology is generally called digital architecture.

For some years blob architecture has been almost synonymous with digital architecture. By now almost a cliché in itself, this digital morphogenesis has been occupied with various processes of form generation. Utilizing current CAD-CAM technologies this approach has led to a new array of forms and surfaces not previously available and has occasionally lead to innovative structures and novel spatial qualities. It is important to recognize that although the use of CAD-CAM technologies is an inseparable part of this approach, it seems that more often than not, these, by now ubiquitous technologies, serve merely as a facilitative and affordable means to play around with free form architecture. The technology ends up being merely an extension of well-rehearsed and established design processes.[1]

Another thing that needs to be pointed out is that with this approach the notion of form-generation has been prioritized to such a degree that it has detached itself from material and construction logics. Being so fixated with the geometric outcome of the process the resulting digital shapes are often impossible or highly expensive to materialize, requiring top-down post-engineering, often at the expense of the desired expression of the formal outcome.

If the logic of computation was utilized in a way that integrated, both material characteristics, assembly logics, and manufacturing constraints, one could, without differentiating between the generation of form and the following materialization, explore morphological complexity and performative capacity. Utilizing this approach in the generation of digital architecture would open up for the potential of CAM technologies as they would turn into one of the defining factors of a design approach seeking the synthesis of form-generation and materialization processes.[2]

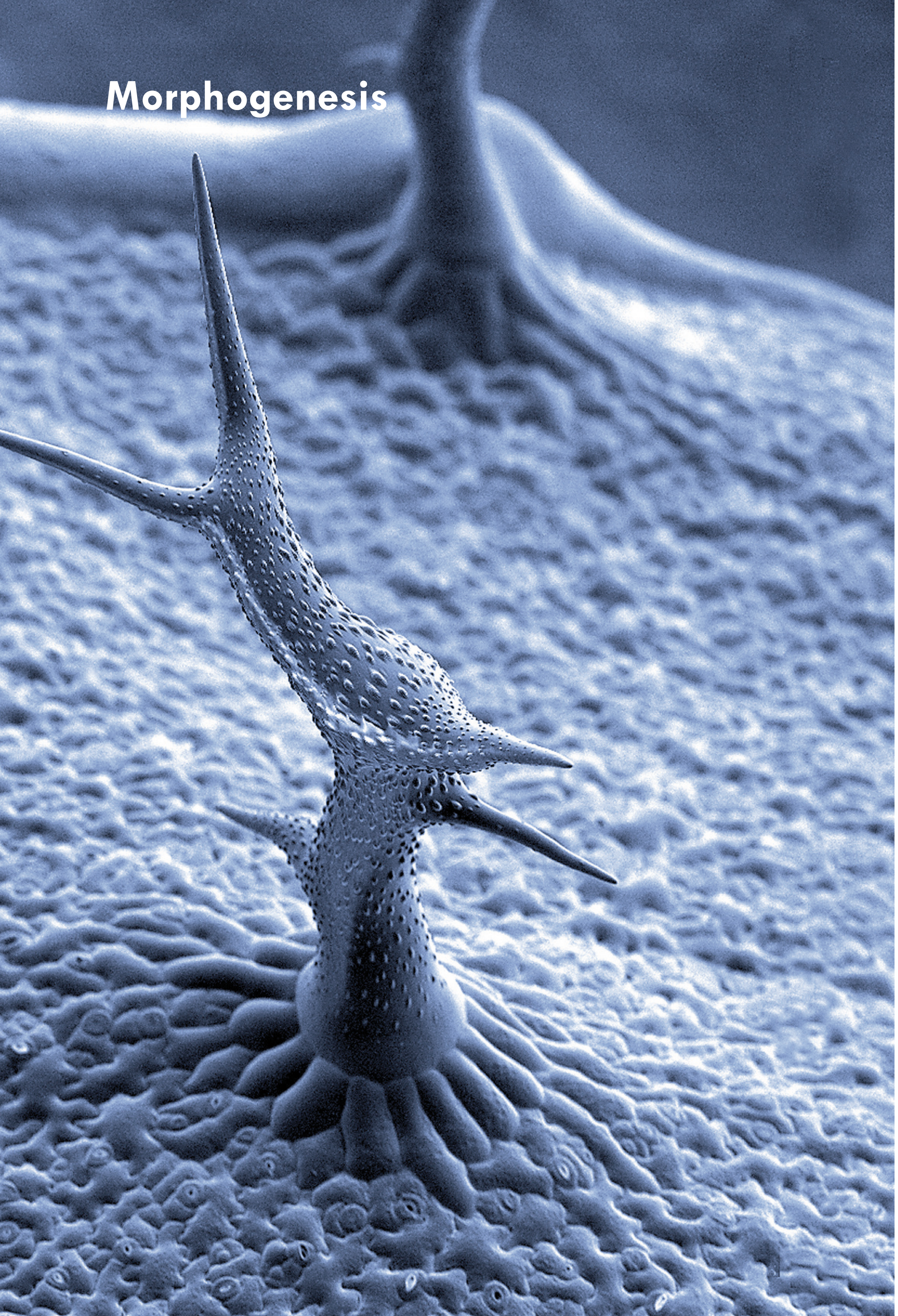
The core of such an approach is an understanding of material systems, in this case a material system that is extended to include its material characteristics, assembly logics, and manufacturing constraints. Adopting this material system approach will include a setup of a parametric computational model that can negotiate the different inputs and: *"…promote an understanding of form, material and structure not as separate elements, but as complex interrelations in polymorphic systems…"*[3]

Notes

[1] **Achim Menges, *Computational Morphogenesis*, ASCAAD Conference, Alexandria, Egypt, 2007**

[2] **Achim Menges, *Computational Morphogenesis*, ASCAAD Conference, Alexandria, Egypt, 2007**

[3] **Michael Hensel and Achim Menges (eds.), *Morpho-Ecologies*, Dexter Graphics, London, 2006, p. 21**



Morphogenesis

In this chapter we will look closer at the different areas of morphogenesis starting with natural morphogenesis, which can be observed in all plants and living beings. Subsequently the term digital morphogenesis will be examined so as to give a clear insight into why an approach based on this term is rejected in favor of an approach that taps into the logics of computational morphogenesis.

Natural morphogenesis

Within the field of biology morphogenesis, or the “creation of life”, involves an attempt to understand the processes that control the organized spatial distribu¬tion of cells and the creation of the characteristic forms of organs and overall body anatomy.

Dealing with the process of evolutionary development and growth, natural morphogenesis generates systems where processes of formation and materialization are always inherently and inseparably related. As these systems obtain their complex form and organization from the interaction between material capacities and external environmental influences and forces, natural morphogenesis is known for deriving polymorphic systems.

Looking at the world one can observe countless examples of how plants and living beings through natural morphogenesis have been “shaped” by the environment. Being successful in this field necessitates that morphological complexity and performative capacity is derived without differing between the generation of form and the processes of materialization.

When looking at the potentials derived from these processes it is striking that architecture as a material practice is still mainly based on an approach that favors the definition and generation of form over its subsequent materialization.[1]

Digital morphogenesis

During the last decade advancements in the computer industry has spawned a remarkable increase in computational power. As a result the use of computers has replaced former analog tasks within the field of architecture. Especially the introduction of CAD software has changed the way that design is created and represented.

Starting out in 1980’s with CAD programs capable of recording command-line sequences of its drawing operations, the developments of the principles behind the interface soon moved onto ‘direct manipulation’, a graphical user interface that made computing accessible to nonspecialists. This development continued and manipulating spline surfaces all day became a part of the architectural practice - the most favored genre of these improvised objects became known as ‘blobs’.[2]

Left: Trichomes, that are found on most cells, affects plant growth and development. (Photo courtesy of Dan Szymanski)

Peter Cook and Colin Fournier, Kunsthaus Graz, Graz, Austria, 2003 (www.kunsthausgraz.steiermark.at)

Frank Gehry, The new Guggenheim Museum, Bilbao, Spain, 1997 (http://www.wikipedia.org)

Frank Gehry, Experience Music Project, Seattle, USA, 2000 (http://www.empsfm.org/)

Top, Bottom, Left; Examples of blobarchitecture. The term "blob" comes from the technical description of a computer-formed shape - a "binary large object." It was coined by architect Greg Lynn in 1995 in his experiments in digital design with metaball graphical software

The term digital morphogenesis refers to various processes of form generation resulting in shapes that often remain elusive to material and construction logics. Being essentially about appearance and form this approach dismisses both the underlying principles of natural morphogenesis as well as the capacity of computational morphogenesis to encode logic, structure and behavior. Because of this the issues concerning materialization, production and construction needs to be subsequently pursued as top-down engineered material solutions. As a result of advances in digital fabrication, there is now far more incentive to work with design approaches that express their outcome in form of variables based on machining processes. Such approaches, dealing more with the logics behind the form than the form itself, are greatly benefitting from the power of computation.[3]

Computational morphogenesis

"No instrumental concept or logic of implementation since the invention of the wheel has fostered so much enthusiasm and promise as computation."

Karl Chu, *Metaphysics of Genetic Architecture and Computation*, 2006[4]

As the term computation is often confused with the term computerization it is necessary to fully understand the differences that separate these two terms, so as to get a clear insight into what lies behind the term computational morphogenesis. The majority of architects today are using computers for computerization; ideas or processes already conceptualized in the mind of the architect are entered into a computer system where they are stored and later manipulated. Computerization is about automation, mechanization, digitization and conversion. In contrast, computation is about exploring not yet defined concepts. It is about calculation, rationalization, logic, algorithm and exploration. Using the computational power of the computer designers can set up systems that allow for the exploration of complexities that extends the limits of human prediction, thereby embracing the possibility for exploring the unpredictable, implausible and unknown.[5]

Two reasons can should be addressed to explain the motives behind this favoring of computational morphogenesis over digital morphogenesis, explaining both why it is the chosen approach in this project but also why it seems to be gaining ground in present architecture.

First, and probably most important, is the advances in the field of digital fabrication. With the increasing availability of computer numerically controlled (CNC) machining much can be gained if design is expressed in terms of variables. Rapid prototyping and the fact that most practices are supplying a rapidly building world where a fast supply chain is a necessity has made the use of CNC machining an important factor in the competition.

The other reason is biology. With the present fashion in architecture being increasingly informed by biology, both when it comes to form and theory, terms like growth and emergence have become of great interest. Moving past mere mimicry of biologic idioms architects are now moving towards generative systems which, inspired by developmental biology, are being utilized in an attempt to uncover emergent properties and processes in complex systems. It is to be able to work with such morphogenetic systems, and their inherent processes, that architects have adapted a computational approach towards architecture.

"In its simplest form, computation is a system that processes information through a discrete sequence of steps by taking the results of its preceding stage and transforming it to the next stage in accordance with a recursive function."

Karl Chu, *Metaphysics of Genetic Architecture and Computation*, 2006[6]

Understanding the term computation one can state that, contrary to digital morphogenesis, computational morphogenesis utilizes the underlying logics of computation to construct models that describe behavior rather than shape. By deploying the geometric rigor and simulation capability of computational modeling, this approach seeks to integrate parameters, ex. in form of restrictions or constraints, so as to define material and construction systems. The power of computation, involving calculations, analysis, randomness, or recursion, can be seen as an "idea generator" which based on computational schemes have the ability of not only expanding the limits of human imagination but also of pointing out to new "thought" processes that may have otherwise never occurred to the human mind.[7]

Notes

[1] **Achim Menges, *Computational Morphogenesis*, ASCAAD Conference on Em'body'ing Virtual Architecture, Alexandria, Egypt, 2007**

[2] **Malcolm McCullough, *20 Years of Scripted Space*, in ADProgramming Cultures, London, Wiley Academy, vol.76, no.4, p. 12-15, 2006**

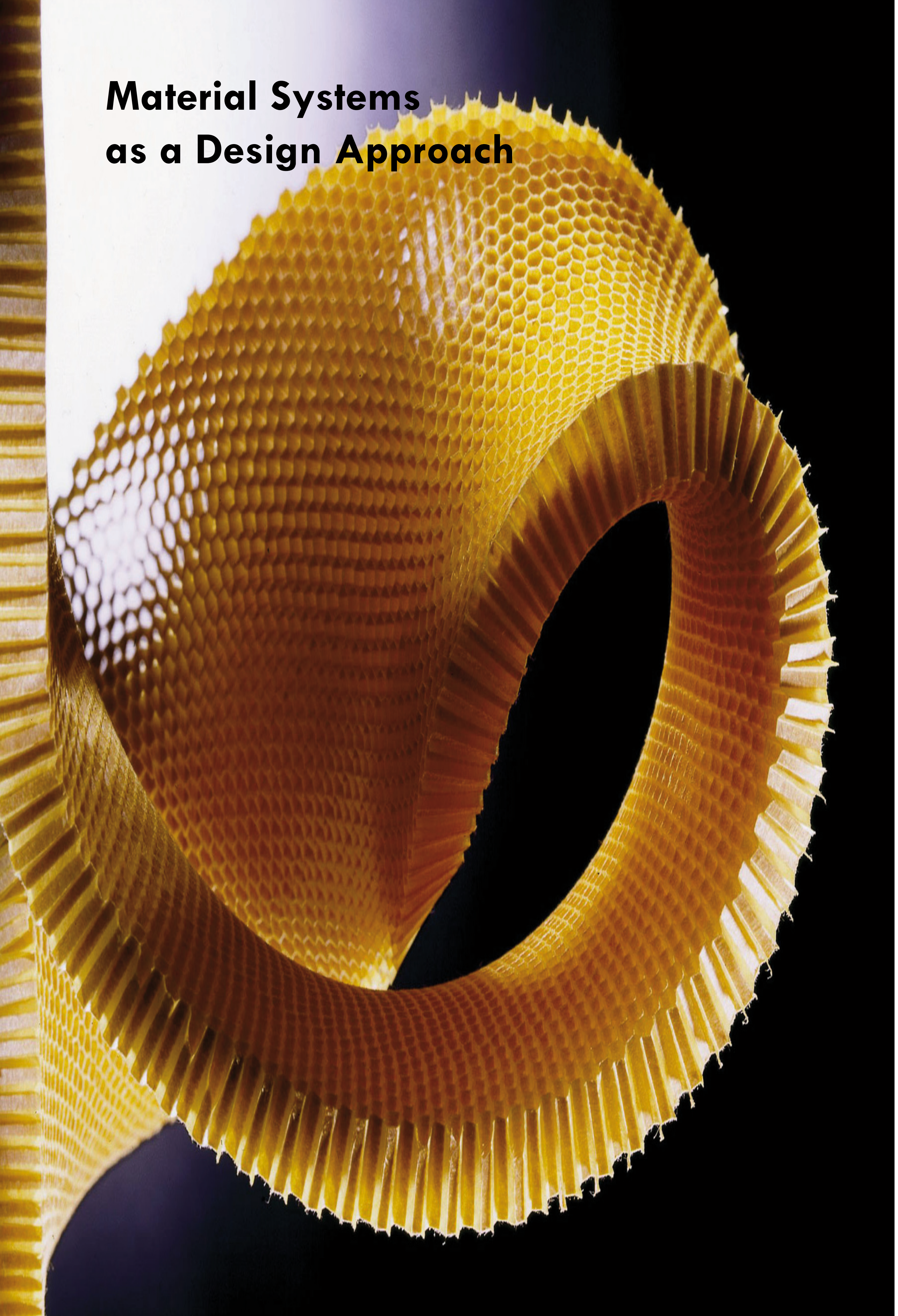
[3] **Malcolm McCullough, *20 Years of Scripted Space*, in ADProgramming Cultures, London, Wiley Academy, vol.76, no.4, p. 12-15, 2006**

[4] **Karl Chu, *Metaphysics of Genetic Architecture and Computation*, in Mike Silver (ed.), AD, Programming cultures, Vol 76, no.4, Wiley Academy, London, 2006, p. 39**

[5] **Kostas Terzidis, *Algorithmic Architecture*, Elsevier Ltd., Burlington, 2006**

[6] **Karl Chu, *Metaphysics of Genetic Architecture and Computation*, in Mike Silver (ed.), AD, Programming cultures, Vol 76, no.4, Wiley Academy, London, 2006, p. 40**

[7] **Kostas Terzidis, *Algorithmic Architecture*, Elsevier Ltd., Burlington, 2006, chap. 2**



Material systems are often understood as derivatives of standardized building systems and elements facilitating the construction of predetermined design schemes. Inspired by the morphogenetic approach presented in 'Morpho-Ecologies', a book by Michael Hensel and Achim Menges, this project will treat material systems as an approach based on the deliberate differentiation of material systems beyond the established catalogue of types. The project will seek to extend the notion of material systems to include its geometric behavior, manufacturing constraints, assembly logics and material characteristics, so as to pursue new designs through the system's intrinsic performative capacities. Taking such an approach to architectural design one can promote the understanding of form, material and structure as interrelated elements of a complex and parametric system capable of responding to environmental inputs.[1]

The core of this approach is an understanding of material systems as generative drivers in the design process and therefore it is important to perform investigations into inherent material properties as this will secure that further development of the performative system takes offset in the potentials of inherent material behavior and geometric capabilities.

The intention behind the creation of a material system is based on a desire to obtain a tool that through a series of operations and manipulations can spawn numerous outputs. These outputs, being the result of a complex system, will be negotiations between material properties, environmental inputs and performance criteria, thereby enabling the architect to explore previously hidden potentials within areas such as function, structure, aesthetics, social performance, spatial creation, etc.

Characteristics, logics, and constraints

As mentioned earlier this approach is based on an understanding of material characteristics. Therefore an important part of the process will be to examine the chosen material so as to be able to extract its inherent potential. One will have to ask oneself: what is this specific material good at? Can it withstand load-pressures? Can it bend? and if yes, how much can it bend before breaking, etc.? Inputs like this will inform the material system and ensure that it utilizes the inherent potentials of the chosen material.

All materials have different properties and geometric behavior. Exploring and investigating the material and its geometric properties will therefore guarantee that any possible formation within the material system is coherent with the inherent characteristics of the material and its geometric behaviors and constraints.

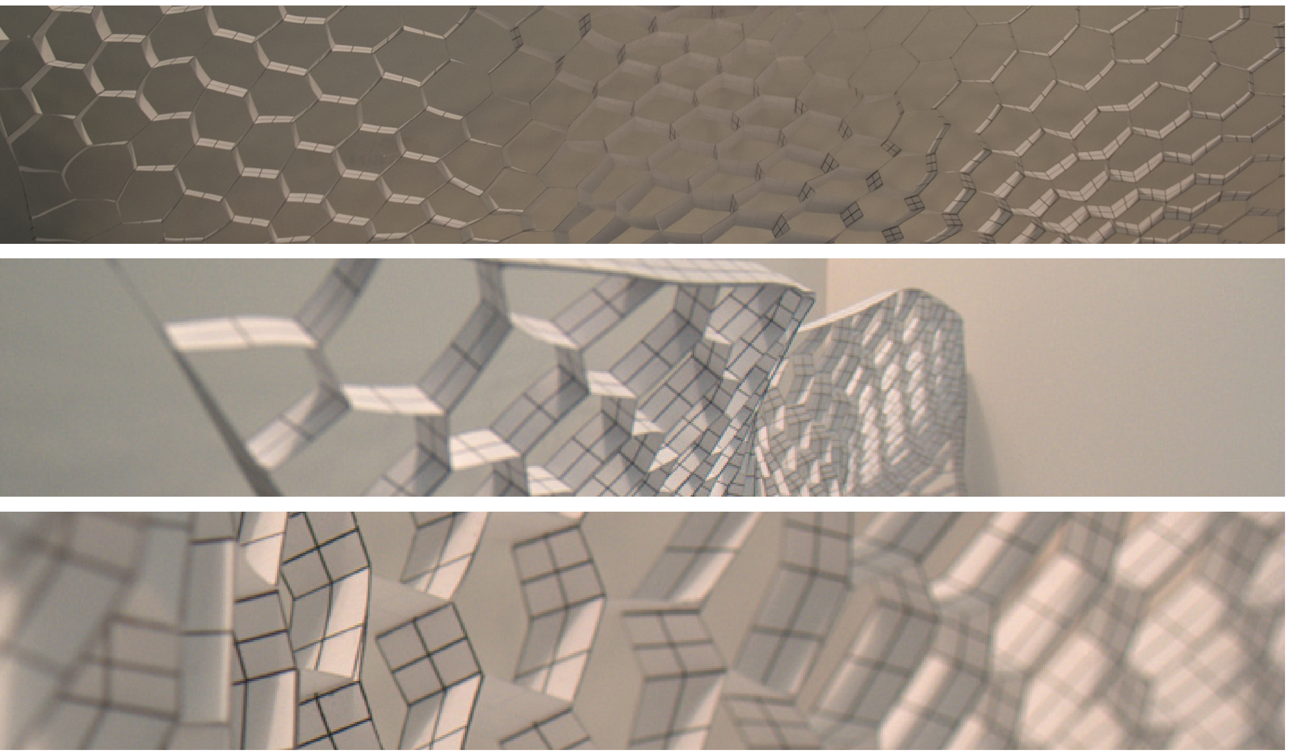
Besides the behaviors and constraints of the chosen material the approach also takes into consideration the restrictions and constraints of fabrication and the logics of assembly. As an example one could take specific restrictions of possible manufacturing processes and use these as generative drivers in the setup. Doing so ensures that any result emerging from the exploration of the material system will remain coherent with

Left: Honeycomb material used in Aircraft Infrastructure (http://www.prnewswire.com/mnr/dupont/27300) material, fabrication and assembly constraints. It is then obvious that one of the critical tasks for the architect will be to define the range in which these parameters can be operated.

Functional models

The material system approach relies deeply on both physical and digital form-finding but to be able to utilize the approach as an instrument in the design phase it is necessary for the physical models to shift away from being merely representational. Therefore this approach will work with physical models that have the ability of becoming (i) scaled functional models capable of being used for form-finding and analysis of performance capacities; (ii) scaled rapid prototype models for checking geometric and topological coherency of larger assemblies of elements.[2]

Using functional models throughout the project has a number of advantages. First of all it will be possible to reveal potential behavioral characteristics, the opportunity for load-testing and subsequent registration of deformations, and the models can be tested for their capacity to modulate micro-environments. Thereby functional models will become central to this performance-oriented design approach.



Above: Paperstrip model, exploring the manufactoring and assembly logics from the manifold project by Andrew Kudless (http://www.materialsystems.org)

Computational Framework

To be able to deal with all the different parameters, restrictions and characteristics found in both the explorations of the functional models and inferred from material, fabrication and assembly logics and constraints, a computational framework has to be set up. This framework has to be open and extendable and most importantly parametric. The reason for this is that it has to deal with the geometric relationships between all the parameters just mentioned.

This open and extendable geometric framework will thereby be based on the logics of the material system, a system that will integrate the possibilities and limitations of fabrication, and the self-forming behavior and constraints of a chosen material. Through evaluation of the material, fabrication and assembly logics of the functional models it will be possible to inform the framework step by step with a series of parameters, restrictions and characteristics. This will include the specific material and geometric behavior on the formative processes, the size and shape constraints of involved machinery, the procedural logistics of assembly and the sequences of construction. In other words this will ensure that any morphology generated within the computational framework can be materialized without contravening with any of the fabrication logics thereby avoiding the need for post-rationalization of the design.[3]

As mentioned earlier one of the reasons for creating functional models is for analysis purposes. Analysis plays a critical role during the entire morphogenetic process, not only in establishing and assessing fitness criteria related to structural and environmental capacity, but also in revealing the system's material and geometric behavioral tendenciesand possible emerging potentials. For example, if one expose multiple varying instances to a digitally simulated light flow, it would be possible to register how the system performs when being manipulated in different ways. In addition to this it could also be beneficial to explore the behavioral tendencies of the system when interacting with different external forces across various individual morphologies. The results of these tests could then be used to inform the computational framework making it capable of negotiating between multiple performance criteria. The computational framework will therefore be exposed to recurring evaluation cycles throughout the design process and be put through a series of manipulations, not to develop variations, but to trace the behavior of the system across a collection of varying instances.

Notes

[1] Michael Hensel and Achim Menges (eds.), *Morpho-Ecologies*, Dexter Graphics, London, 2006

[2] Michael Hensel and Achim Menges (eds.), *Morpho-Ecologies*, Dexter Graphics, London, 2006, p. 34

[3] Achim Menges, *Computational Morphogenesis*, ASCAAD Conference on Em'body'ing Virtual Architecture, Alexandria, Egypt, 2007

Problem statement

This project intends to explore the potentials of the setup of a material system - a parametric and generative assembly consisting of and taking into consideration material properties, manufacturing constraints and geometric behavior. Approaching the project through the construction of this kind of logic-driven system the aim will be to explore the possibilities of a material system that fulfills spatial, structural and performative requirements concurrently and how to negotiate with these in situations where they are conflicting.

The implication of such a material system approach is the need for a:

"*...shift away from programme as design-defining towards design as programme-evolving.*"

Michael Hensel, *Morpho-Ecologies*, 2006[1]

This shift will greatly affect the design process of this project. We will not be imposing a predefined form onto a programme but rather attempt to tease out or elicit the emergence of a changing form from a flow which has its own intrinsic behavior.

Notes

[1] **Michael Hensel and Achim Menges (eds.), *Morpho-Ecologies*, Dexter Graphics, London, 2006, p.58**

Objective

The central aim of the thesis project is the development of a material system containing a high degree of integration between its design and performance. A system capable of adapting to varied performance requirements through the modulation of the system's inherent geometric and material parameters while remaining within the limits of chosen production technologies. In this synergetic relationship, factors as: unpredictable, implausible, and unknown will not be seen as factors of fear but rather invi¬tations for exploration. When situated in an urban space the material system should be capable of responding to the immediate context and the dynamics of the urban life so as to be capable of adding new spatial configurations and environments. Therefore, although the material system, during its development phase, will be generated and motivated to grow and evolve in a contextless space, the aim is to implement specific site parameters and values into the computational framework.

Digital architecture is often characterized by prioritizing the generation of form over inherent material properties resulting in shapes created without any consideration for their inherent material properties. Instead, this project, through the use of computational techniques and digital construction technologies, seeks to explore and unfold the inherent capacities of materials. The project will also explore the potential of utilizing the power of computation - involving vast quantities of calculations, combinatorial analysis, randomness, or recursion, to name a few - as an 'idea-generator' that can point out new 'thought' processes that might never have occurred in the human mind, thereby expanding the limits of human imagination.

Since this project is of an experimental character documentation of the process is a very important factor. Recurring cycles of analysis and evaluation will ensure a continuous adding of new parameters and restrictions to the system and here rapid prototyping plays an important role enabling an interchange between digital test and analog material-dependent models. This ensures a coherency within the material system.

The criteria's for evaluating the installation is therefore: (i) Ability to adapt to site-specific parameters, (ii) the degree of integration between design and performance, and (iii) the degree of coherence between the different logics of the material system.

Limitations

Due to the limited timeframe of this thesis project some aspects has to be leaved out or only touch upon in a restricted manner. The following text will therefore seek to list the limitations of this project so as to enhance the clarity regarding this project's field of work.

First of all, the aspect concerning assembly logics will not be taken into consideration during the design phase. Although potential restrictions to the computational framework could be found by thoroughly exploring this part of the material system, this project will only take into account those general parameters uncovered through explorations of other interrelated parts of the system. So although the aspect won't be dealt with directly it will still be a part of the material system.

As the weight of this project is on the setting up and the implementation of a material system approach and not on the actual production of a functioning computational framework, the implementation of calculations for structural analysis will likewise be omitted. Yet, as in the case of the assembly logics, this aspect will still be discussed in parts of the project regarding the potential benefits of utilizing such methods within the computational framework.

As just mentioned in the objective, one of the aims of this project is to implement the system in an abstract urban environment, where the material system will be tested to explore how it adapts to specific site parameters and values. In this process the project will refrain from performing an in-depth analysis of a given site, but instead construct an abstract test environment containing all the contextual parameters needed for a successful implementation of the material system.

Theoretical background

Morphogenetic systems

Seeking to set up a material system, a parametric and generative computational framework, it is essential to build up some general knowledge concerning the theoretical aspects connected to this field of architecture. In a general manner the following writings will introduce the important theoretical terms and thoughts associated with morphogenetic systems, algorithms, and parametric architecture.

As mention earlier, contemporary architectural discourse is greatly inspired by another profession; namely biology, or more precise developmental biology, which is motivating architects to adopt the theoretical motivations behind the morphogenetic systems' approach. Morphogenetic systems, or genetic architecture, are a very fundamental approach to the design and construction of buildings as it deals directly with the construction of objects.[1]

Genetics is a term that can be traced back to William Bateson, a British geneticist who used the word to describe the study of inheritance and the science of variation. The meaning of the term 'genetics' is abstract and general enough to be used in architecture as it can be reduced to the idea of replication where a new cell is generated based on rules inherent in the genetic code of a previous cell.

To explain how these mechanisms of replication works we have to introduce another term called: recursion. Recursion is a generative function, or rule, that repeatedly calls itself by applying the same rule successively, thereby generating a self-referential series of transformations. It is this logic that bridges genetics and computation since this generative logic also lies at the base of computation.

It is important to note that the term genetic within genetic architecture isn't referring to a form of biomimesis, but to the concepts of recursion and self-replication. In fact the theoretical origins of genetic architecture can be traced back to John von Neumann, a Hungarian born American mathematician who developed what came to be known as the von Neumann architecture - a concept of a computer design model that used a stored memory program. This rather abstract proposal is based on two central elements: a Universal Computer and a Universal Constructor. The way it works is that the Universal Computer is embedded with a program that can direct the behavior of the Universal Constructor, which, in turn, can manufacture both a new Universal Computer and a new Universal Constructor. Once this was done, the Universal Computer would be programmed by copying the program of the original Universal Computer, and execution of the program would then begin. The von Neumann architecture is perhaps the ancestral and archetypical proposal for a self replicating system and as it addressed a notion that lies at the very heart of biology: the ability for a system to contain a complete description of itself and to use that to create new copies (the idea of a machine capable of self-replication), the von Neumann architecture can be seen as a precursor to the architecture of genetic systems.[2]

Left: The entire structure of the watercube is based on a unique lightweight-construction, developped by PTW with ARUP, and derived from the the structure of water in the state of aggregation of FOAM. Behind the totally randomized appearance hides a strict geometry as can be found in natural systems like crystals, cells and molecular structures (http://www.chrisbosse.de/watercube/)

Algorithms

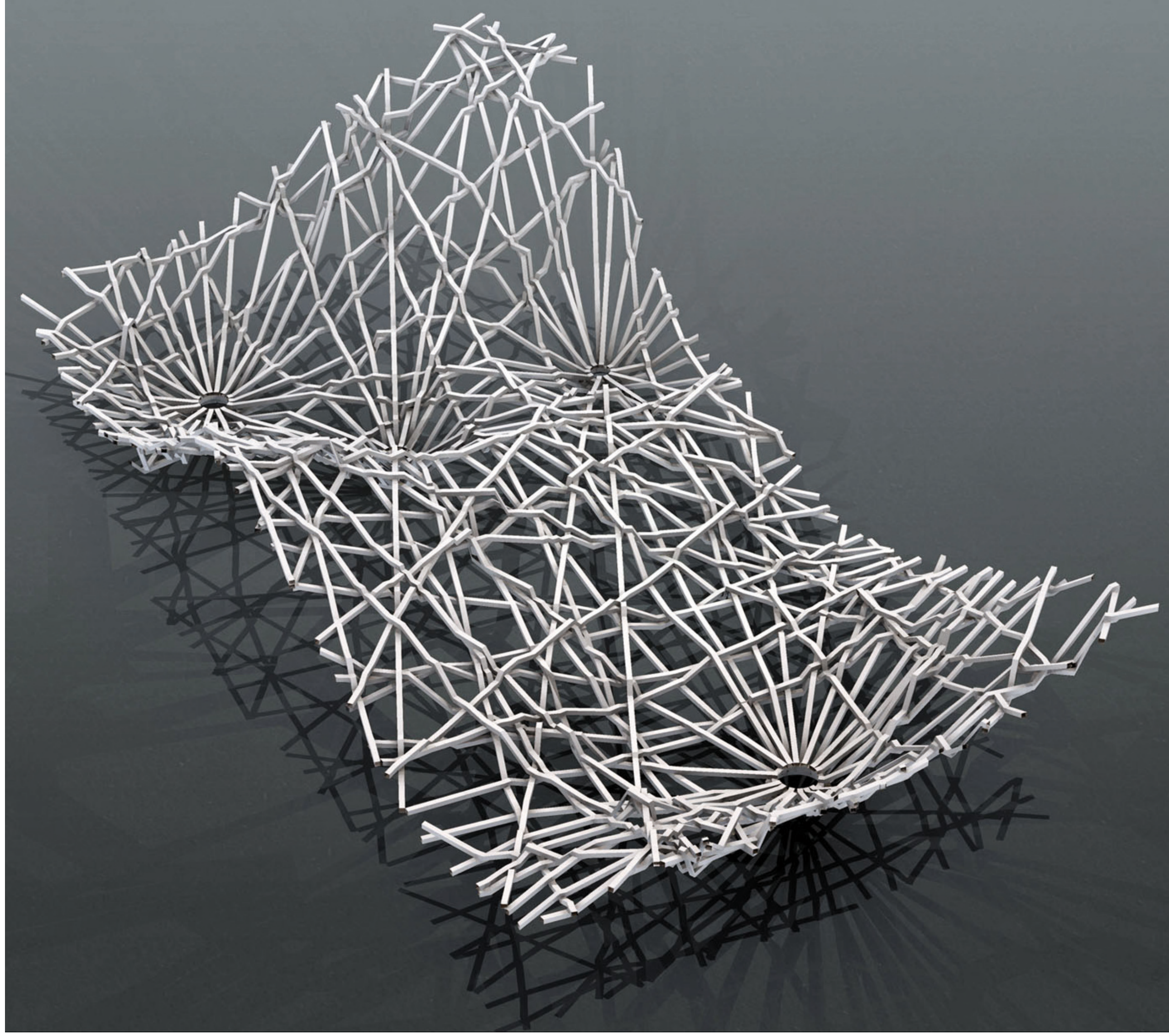
When talking about the term recursion as well as von Neumann's idea for a self-replicating system we're dealing with a line of thought that focuses on the application of rules or instructions, and on the idea of breaking up a process into a series of transformations. These aspects are identical to the ones dealt with when working with algorithms.

An algorithm can be described as the process of addressing a problem in a finite number of steps and can thereby serve as a method for codifying a problem through a series of consistent and rational steps. By the fact that algorithms can be seen as mediators between the processing power of the computer and the human mind makes them essential for this particular project as it namely seeks to utilize the power of computation.

As mentioned in the chapter on 'Computational Morphogenesis' this project tries to utilize the computer as more than merely a tool for productivity or presentation; thereby going against the common belief that the mental process of design is conceived and envisioned entirely in the human mind. Contrary to common beliefs algorithms can produce results for which there is no intention and which behavior cannot be predicted. An example that supports this fact can be found in a system called the Dada Engine. This engine is a system for the generation of text from a text file containing rules in the form of a grammar. Based on the principle of recursive transition networks, or recursive grammars, the system generates strings of text and by applying randomness in the arrangement of the text it can produce sentences that are unpredictable, but also accidentally meaningful, and unlike chaos, defined by the rules that govern the system.[3]

Design is often considered an act of conscious decision-making based on an intention derived from the human mind. An act based on the presence of a human mind. But if one were to accept that the decision was made by an algorithmic process instead, then the designer would no longer be confining the act of decision-making and results not initially intended might occur, results that the designer may assess as "successful" and adopt as one's own ideas, thereby assigning intention after the decision-making. Such a shift will advocate for an understanding of algorithms as being about exploration, codification, and extension of the human mind.[4]

When utilizing this explorative side of algorithms in the design process the keyword becomes synergy. The process will be based on the mutual contributions of both the human mind and the extendibility of the machine. In his book "*Animate Form*", Greg Lynn writes about this possible synergy: *"A machine, meanwhile, could procreate forms that respond to many hetero un-manageable dynamics. Such a colleague would not be an omen of professional retirement but rather a tickler of the architect's imagination, presenting alternatives of form possibly not visualized or not visualizable by the human designer."*[5]



For architects to be able to apply algorithms in their design process they would need to first resolve the problem of presenting their final product (likely a building) as the process that generated it, and next to break this process up into a series of well-defined steps or operations. This sequence, or actually the computer code that described it, will then become the "genetic code" of the product. Architects are already using CAD programs as an essential part of their work and as CAD models containing architectural structures are already described by a series of operations, this greatly simplifies the problems. A simple wall element for example, is produced by a series such as this: 1) Draw a rectangle defining the length and width of the wall; 2) extrude the rectangle to the desired height; 3) perform a few "Boolean subtractions" to carve out holes for windows and doors and other details. Some software packages allows the user to get a hold of the computer code corresponding to this sequence, so this code now becomes the "virtual DNA" of the wall element.[6]

Above: Fibrous self-interlocking derived through synthesised digital processes of component differentiation, mapped propagation and digitally simulation. Achim Menges (http://www.achimmenges.net)

"To be original and to be in control, the designer has to understand, if not originate, his own algorithm, and know how to 'drive it'."

Robert Aish, *Exploring the analogy that parametric design is a game*, 2006[7]

Parametric architecture

For many years, mechanical engineers and industrial designers have benefitted from parametric design software, enabling them to generate and study many alternative solutions of the same design. It is not until recent years that this way of working with computers has been made available for architects. Based on consistent relationships between objects, parametric design allows changes in a single element to affect another element or to lead to changes throughout an entire system. Robert Aish, former Director of Research at Bentley Systems and one of the leading forces behind the development of the parametric software Generative Components, has given the following description about the value of parametric/associative design:

"Design involves both exploration and the resolution of ambiguity. Therefore, it is not sufficient that computational design tools can model a static representation of a design. What is important is that the design tools are able to capture both the underlying design rules from which a range of potential solutions can be explored, and facilitate how this 'solution space' can be refined into a suitable candidate for construction…"

Michael Hensel, *Techniques and Technologies in Morphogenetic Design*, 2006[8]

In order to apply parametric modeling to ones design practice it is important to understand that any object or form can be specified in countless ways. Its geometric properties can be described graphically, as through drawings, where objects can be depicted in detail and accurately constructed or replicated. Alternatively, the behavior of objects can be described by means of their desired performance. However, it is also possible to explain objects properties through how they relate to other objects or entities, an object's mass for example can be found by multiplying its dimensions with its material density.

As the setting-up of such relationships in parametric design software enables the designer to formulate links between huge amounts of data from which an infinite number of geometric forms can be generated. Such a parametric system, or in this project computational framework, entails a procedural, algorithmic description of geometry. This entails the need for a fundamental understanding of geometric primitives: points, planes, line arc, curves, surfaces, and solids. An understanding of for instance what the 'order' of a curve means, how curves and surfaces are parameterized and how geometric operations work.[9]

Using a parametric approach in the design process not only makes it possible to manage and take into account a lot of different parameters simultaneously, but it also opens up for the possibility to explore many alterations of the same system, and to make minor change in the design, without it requiring major updates to the model, thus restricting the number of design alternatives due to time constraints.

Notes

1 Karl Chu, *Metaphysics of Genetic Architecture and Computation*, in Mike Silver (ed.), AD, Programming cultures, 76(4), Wiley Academy, London, 2006

2 Von Neumann architecture: *http://www.zyvex.com/nanotech/selfRepJBIS. html#vonNeumannArchitecture*

3 The Dada engine: *http://dev.null.org/dadaengine/manual-1.0/dada.html#SEC2*

4 Kostas Terzidis, *Algorithmic Architecture*, Elsevier Ltd., Burlington, 2006, p.25

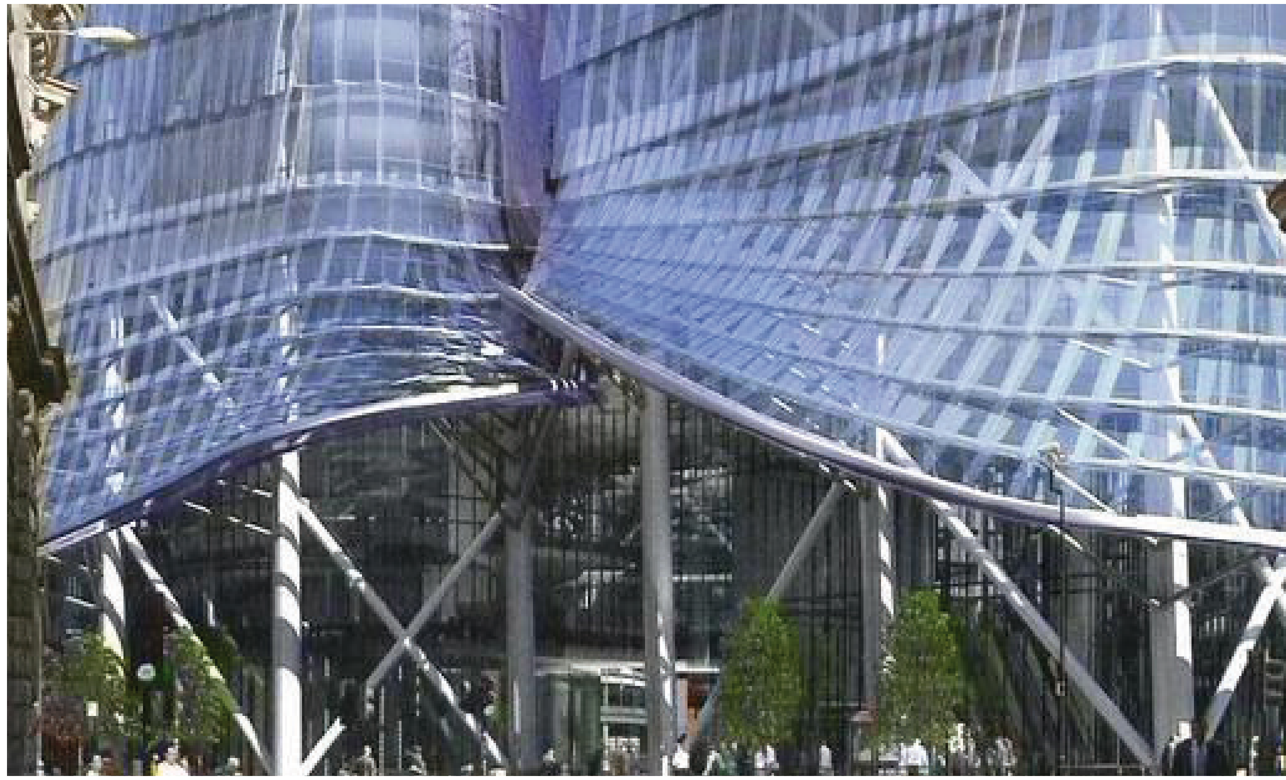
5 Greg Lynn, *Animate Form*, Princeton Architectural Press, 1998

6 Manuel DeLanda, *Deleuze and the Use of the Genetic Algorithm in Architecture*, http://www.cddc.vt.edu/host/delanda, 2001, p.1

7 Robert Aish, *"Exploring the analogy that parametric design is a game"*, Game Set and Match II, Episode publishers, Rotterdam, 2006, p.204

8 Michael Hensel, Achim Menges and Michael Weinstock (eds). *Techniques and Technologies in Morphogenetic Design*, AD, 76(2), London: Wiley Academy, 2006, p.47

9 Robert Aish, *"Exploring the analogy that parametric design is a game"*, Game Set and Match II, Episode publishers, Rotterdam, 2006



Top: BishopGate Tower, Kohn Pedersen Fox Associates, 2010 (http://www.skyscrapernews.org)

To sum up, this project seeks to construct and explore a material system, a logic-driven system that fulfills spatial, structural and performative requirements concurrently. Looking into the field of parametric architecture it has subsequently been shown that adopting a parametric approach not only makes it possible to manage and take into account a lot of different parameters simultaneously, but also opens up for the possibility to explore many alterations of the same system. Also evident is the fact that taking a parametric approach to setting up a material system entails the incorporation of a procedural and algorithmic description of geometry.

Having deduced that the aspects identical between genetic systems, more specifically von Neumann's idea for a self replicating system, and the use of algorithms, is the application of rules or instructions and the idea of breaking a process into a series of transformations, it is evident that incorporating these lines of thought will greatly assist in informing the material system. In the setting up of the material system an algorithmic approach will therefore be utilized so as to make sure that every limitation, restriction or behavioral tendency found, will be broken into a logic series of transformation before added to the computational framework. As stated in a quote by Robert Aish, it is: *"important to capture the underlying design rules from which a range of potential solutions can be explored"*[1]

With regards to the specific task of informing, or setting up, a material system it can also be concluded that it is important to shift towards using the design, or the behavior of the material system, as program-evolving. In other words it is important to repeatedly analyze and explore the emerging potentials of the material system and let these behavioral outputs guide the evolution of form. Not doing so will increase the probability of decisions being based on mere commonplace habits and established archetypes, or on individualized artistic intent based on self-willed expression.[2]

The following chapter, titled 'The Design Process', will describe the process of utilizing a material system approach. The described explorations will be based on a selected material and will be based on recurring analysis, and on an extensive use of both physical and digital models. An important factor throughout this design process was to elicit the inherent behavior of the selected material, as well as exploring and unfolding the emerging potentials. What are the features or qualities of this exact material system and how can it be utilized in an architectural sense?

Notes

[1] **Robert Aish, *"Exploring the analogy that parametric design is a game"*, Game Set and Match II, Episode publishers, Rotterdam, 2006**

[2] **Toyo Ito, *Diagram Architecture*, El Croquis 77, Madrid, 1996, p.19**

 Above: London City Hall, Norman Foster, 2002 (http://www.greatbuildings.com)

The Design Process

In chronological sequence, this chapter will present the project’s design process. As this project’s main focus is on the utilization of a design approach that seeks to explore the potentials of a material system, the design process involved a setting up and execution of a range of different physical and digital experiments. The chapter will therefore consist of recurring cycles of analysis and evaluation ensuring a continuous adding of new parameters to the system. Throughout the chapter the computational framework will gradually be informed by the uncovered potentials and at the end the system will be implemented on a site where it will adapt to site-specific parameters and values.

“Building with membranes is emerging from the shadow of the early pioneering achievements. Several decades of practical experience have led to a technology that is future oriented and that deserves to be more widely established.”

Klaus-Michael Koch, *Membrane Structures,* 2004[1]

As earlier stated the explorations of the material system will be based on a selected material. Having seen a great potential in the utilization of membrane structures, both with respect to its material performance and its self-organizational behavior, these were chosen as the underlying basis for an exploration into material systems, or more precisely into membrane systems.

Notes

[1] **Koch, Klaus-Michael (ed), *Membrane Structures*, PrestelVerlag, Munich, 2004, p.8**

Membranes

Membrane constructions are most frequently associated with tents - an architectural archetype that has survived through all the epochs of history, mainly serving as roofing for temporary purposes. Membrane has through history been

Through history, membrane constructions has evolved from simple transportable tent made of animal skins, to more elaborate constructions for dwelling serving to protect against wind and cold. In recent years membrane structures has moved past the genre of being mere tents for temporary use at festivities and circus events. Through the works of Frei Otto, a German architect and structural engineer, membrane systems evolved into large-scale, highly complex, cable-membrane constructions capable of covering huge areas. The works of Frei Otto also lead to the emergence of new production and material technologies, which concurrently with the developments in the areas of computer technology and programming, made it possible to develop and construct structures of with higher structural complexity.[1]

"A membrane is a thin, synthetic or natural, pliable material that constitutes the lightest material means for spatial organization and environmental modulation."

Michael Hensel, *Versatility and Vicissitude*, 2006[2]

Looking at the structural aspect of membranes an important aspect is that they only transmit tensile forces and therefore belong to the form-active tension systems. This fact entails that membranes, according to the applied forces, take shape into minimal surfaces. This is not to be understood as the construction of minimal surfaces, but rather that the shape of a membrane is found through its self-organizing behavior - as the state of equilibrium of internal and external forces.

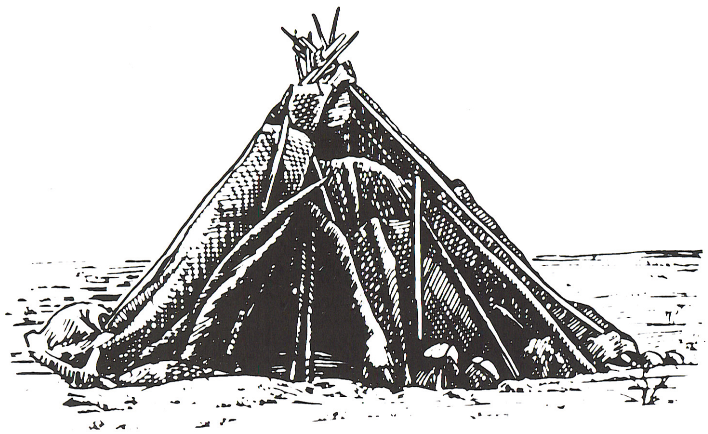
Transferring this knowledge to form-finding experiments, the shape of a membrane structure can be found by applying differentiated forces and then utilizing the self-organizational behavior of the membrane to derive the resulting shape. Such experiments not only depend on extrinsic forces but also the tension of the membrane itself; both having an influence on the final state of the relaxed membrane geometry.

Today such form-finding processes can be conducted both physically, as in the case of the earlier projects of Frei Otto, and digitally, by means of dynamic relaxation. Dynamic relaxation is a finite element method, which, based on the positioning of boundary control points and the specific elasticity of the membrane, settles a digital mesh into an equilibrium by performing iterative calculations.[3]

These aspects lead to the conclusion that any membrane structure consists of two principal components: the membrane itself, which may be a single field, or region, or multiple fields with a variety of sub-divisions, and the primary (or support) structure that equilibrates the tensile forces from the membrane and transmits them to the ground under both pre-stress and applied loading conditions.[4]

Sangesari tent, northan Iran, 1970 (Membrane Structures, Prestel Verlag, Munich, 2004)

Olympia roof in Munich, Frei Otto, 1972 (Membrane Structures, Prestel Verlag, Munich, 2004)



Transpotable tent of the Nganasan people, Siberia (Membrane Structures, Prestel Verlag, Munich, 2004)

Notes

1 Klaus-Michael Koch (ed), *Membrane Structures*, Prestel Verlag, Munich, 2004, p.18-35

2 Michael Hensel and Achim Menges (eds), *Versatility and Vicissitude*, 78(2), AD Wiley Academy, London, 2006, p.75

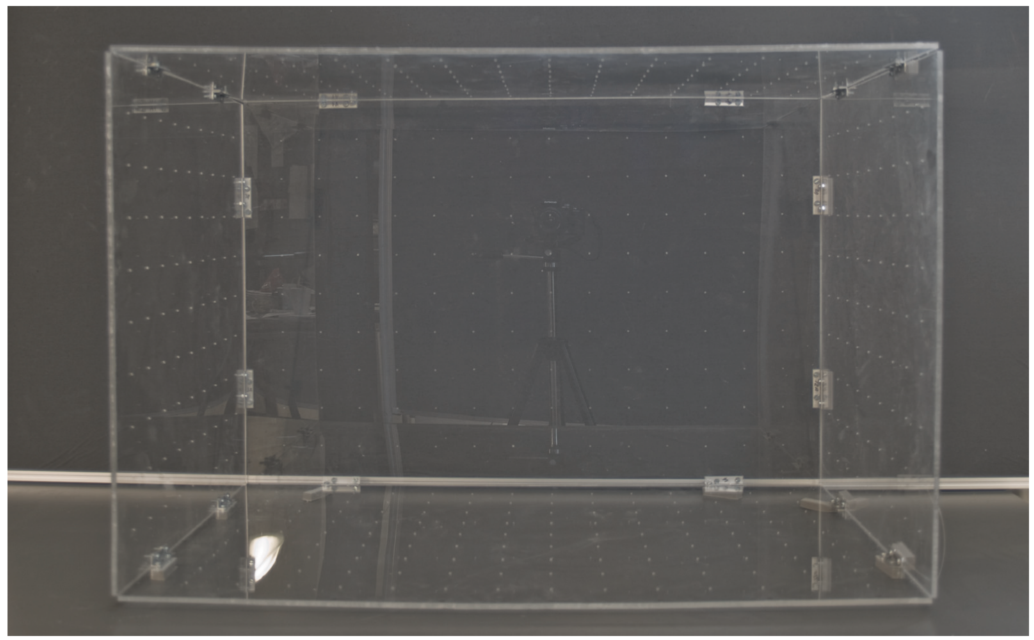
3 Michael Hensel and Achim Menges (eds), *Versatility and Vicissitude*, 78(2), AD Wiley Academy, London, 2006, p.75

4 Klaus-Michael Koch (ed), *Membrane Structures*, Prestel Verlag, Munich, 2004, p. 71

Experimental Setup

As mentioned, membrane systems are form-active tension systems, which mean that they transmit only tensile forces and take shape according to the applied forces. As result of this, membrane systems must be form-found, utilizing the self-organizational behavior of membranes under extrinsic influences. To be able to work with this kind of form-finding approach it was important to construct some sort of experimental setup that would allow for the introduction of tensile forces, or constraining control points, capable of collecting or transmitting the tensile forces of a membrane in tension. A solution was found in the setup of an acrylic box: a rectangular box consisting of five acrylic plates which all have a series of holes drilled so that they create a point grid with five centimeter spacing. Using a standard fishing line and regular elastic fabrics the membrane patches can be strung up inside the acrylic box and visually assessed from all angles. Assigning a coordinate system to the box enables an extraction of the various control point coordinates, making a smooth transition from the physical membrane experiments to any CAD software; thereby facilitating further explorations.

Skeptic minds could point out that experiments conducted within a spatially confined acrylic box, with a limited number of holes, and the fact that regular elastic fabrics are used as membranes, would limit the outcome. Therefore it should be emphasized that the experiments are conducted with the intention of registering and extracting general geometric behaviors and environmental performances, and to get an understanding of the different parameters present within a membrane system; parameters that can subsequently inform the computational framework. As a result of this it is not believed that the chosen setup is limiting the outcome.



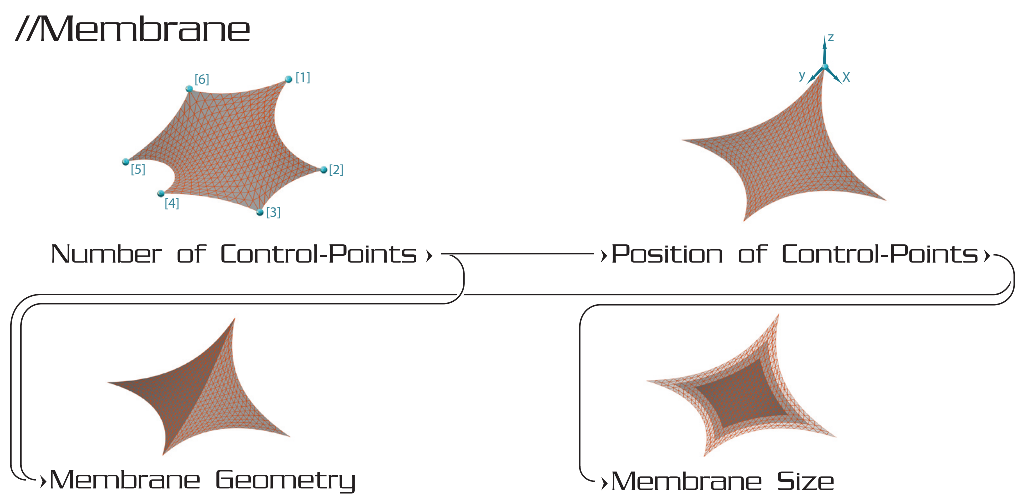
Left: Membrane test
Above: The perspex box used for membrane testing

Initial Experiments

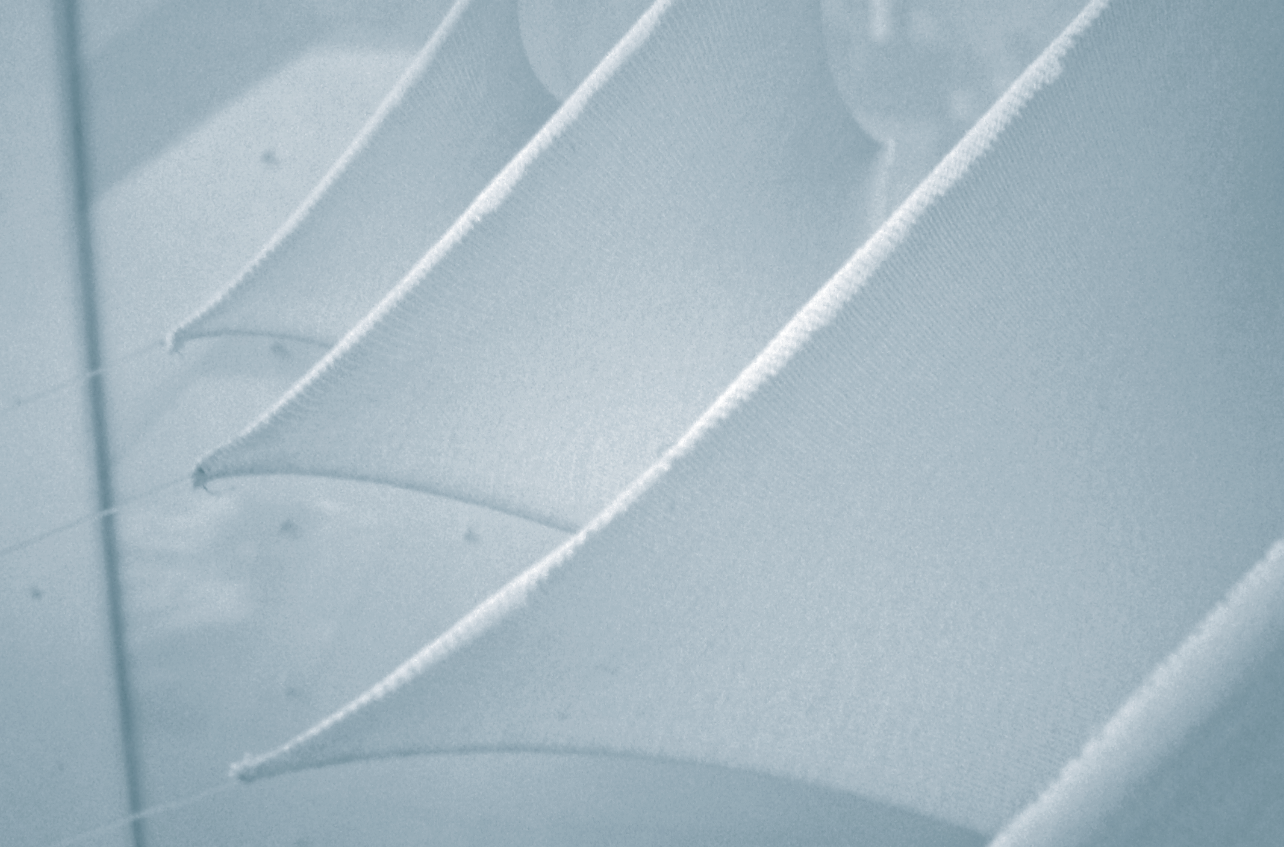
To get an insight into the technical aspects of cutting and mounting membrane patches as well as identifying the parameters present within the membrane system, our initial experiments commenced with the setting up of three basic membrane geometries: cone, hypar and barrel.

Besides giving a basic understanding of how membranes react when exposed to varying tension, our initial observations also lead to the conclusion that the membrane systems consisted of four basic parameters: the size of the membrane patch, the geometry of the patch, the number of control points connected to the patch, and the spatial placement of these control points.

With three different physical experiments set up and registered, the transition from physical models to digital three-dimensional models could also be tested. Taking advantage of an existing surface relaxation script[1] made for Rhino it was possible to recreate the physical experiments as well as making further experiments with regards to tensile forces. When setting up the surface relaxation process in Rhino one has to input the level of tensile forces for the membrane as this determines its elasticity.
Altering this parameter, while at the same time keeping the initial setup of the control points, changes the geometric expression of the membrane. Making these experiments with the physical models would have required a series of different fabrics with varying elasticity which in turn had to be mounted within the box; time consuming and demanding in physical tests yet a simple and fast manoeuvre within the digital software.



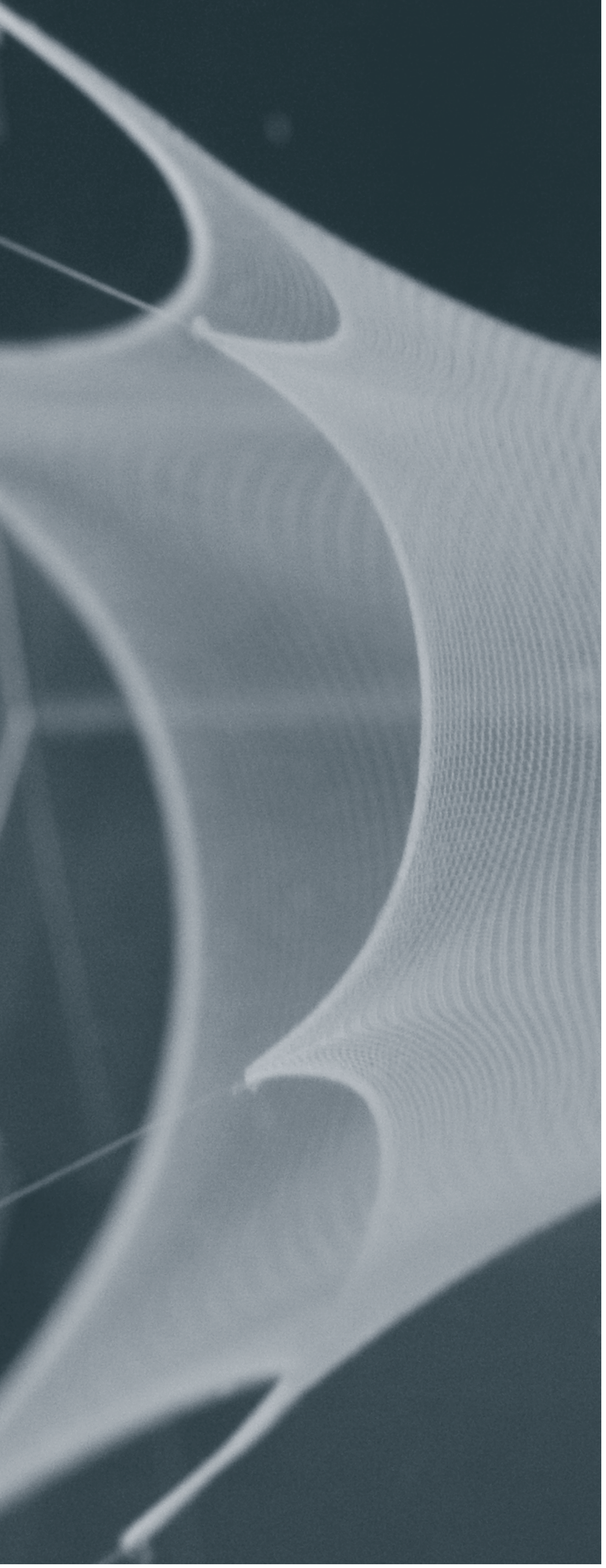
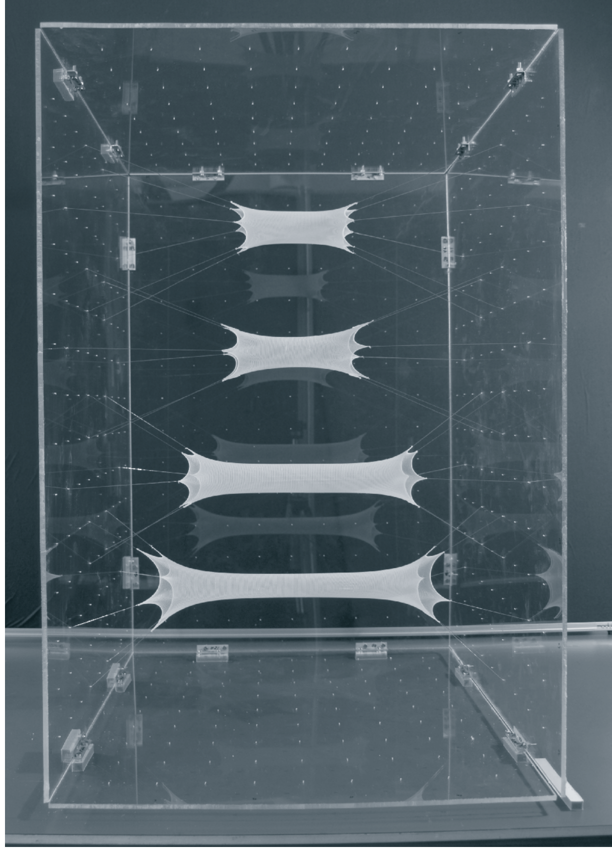
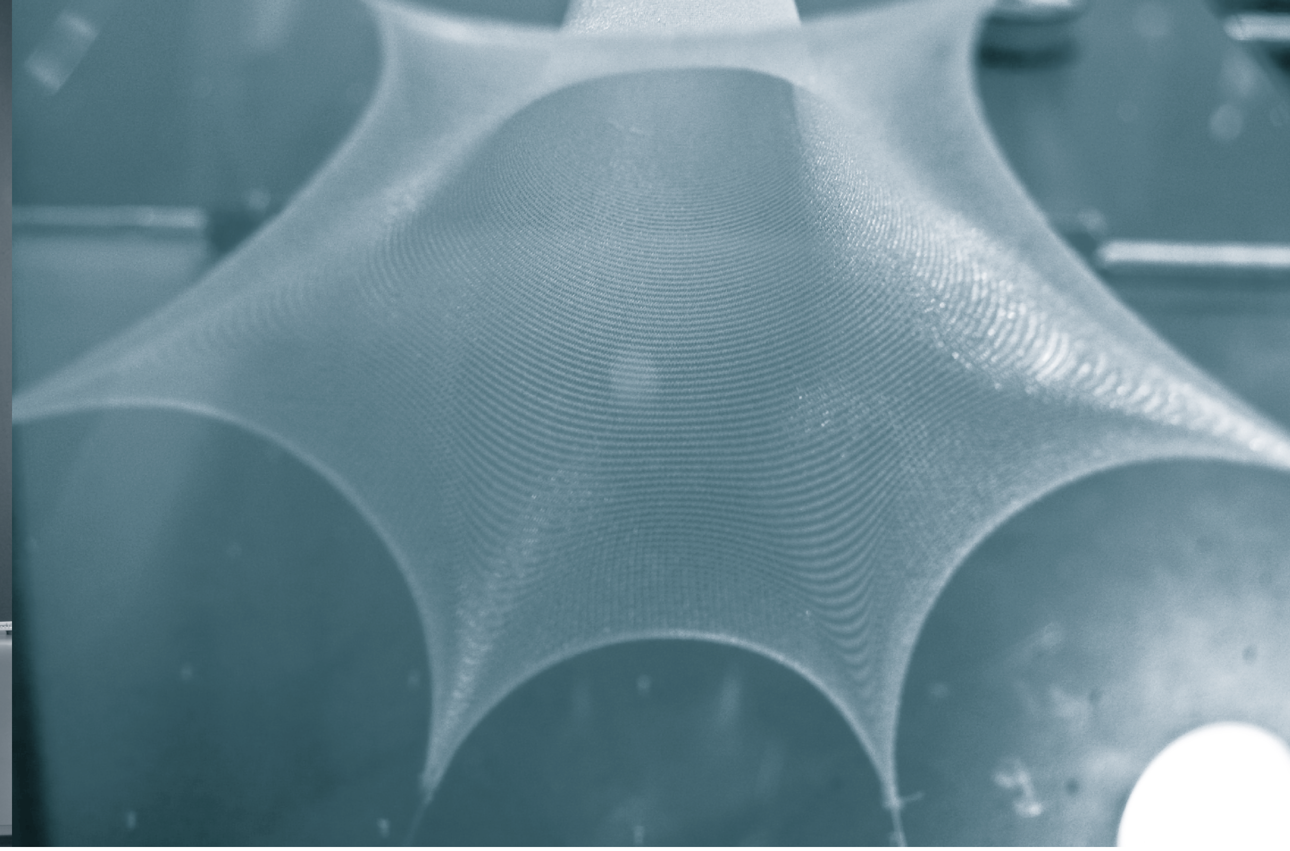
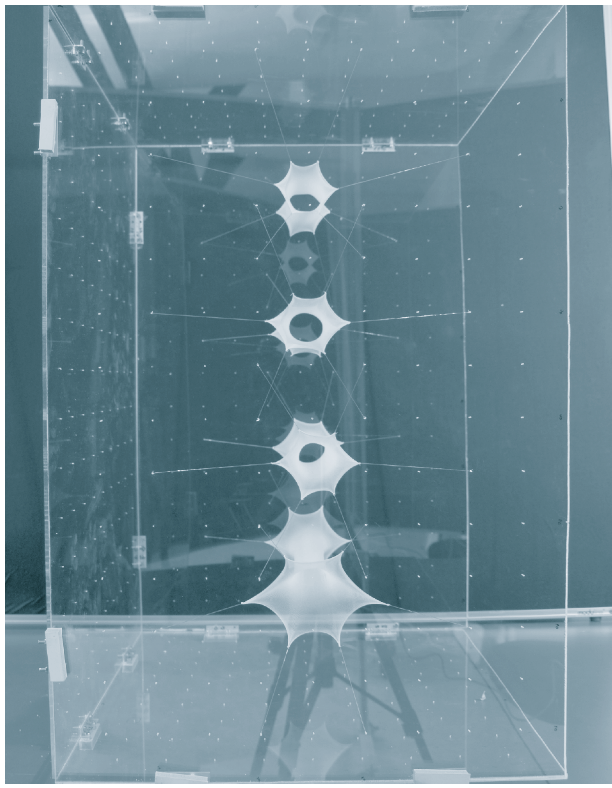
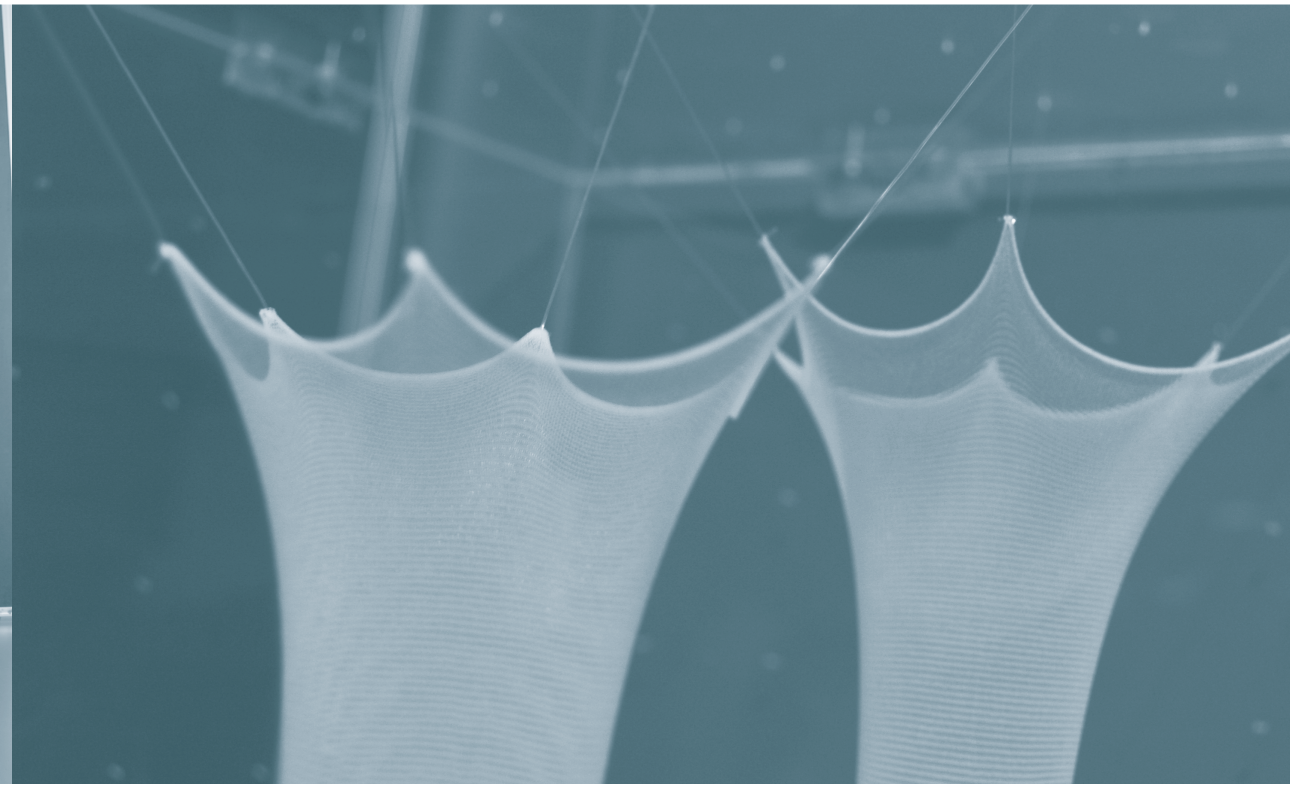
Left: Membrane setup in the perspex box
Above: Diagram of relationship between the membrane elements




 Both Pages: Saddleshape membrane




Both Pages: Coneshaped Membrane

Both Pages: Tube Membrane

On the basis of the initial experiments the manufacturing logics, one of the four essential factors of the material system approach, could also be tested and thereby serve to inform the setup of the material system. Deciding that the membrane system should be produced in the form of pre-tensioned membranes, our current membrane experiments could be assessed as to how they could be produced.

As the word indicates, pre-tensioned membranes are produced with non-elastic fabrics, so, in order to create curvature in two directions this production technique requires for a surface to be unfolded, cut out in patterns, and subsequently jointed together through sewing. The number of panels is depended on the degree of curvature in the desired membrane, the higher the degree, the more panels needed. The dimensions of the desired membrane also determines the number of panels as a 10 meter wide membrane for example, can't be constructed of only three panels if the maximum width of the fabric in use is only three meters. This restriction of course has an influence on the aesthetic expression of the resulting membrane, as more panels means more seams, resulting in more evident subdivision of the membrane. To translate, or unroll, the relaxed surface into two-dimensional panels, a software application from Meliar Design[2], called MPanel, was utilized throughout the project.

Notes
1 David Rutten, *http://www.reconstructivism.net*
2 Meliar Design, *http://www.meliar.com*

Above: Test of pretensioned membrane manufactoring

Contextual Dependency & Environmental Modulation

The initial experiments showed that the membrane system featured four basic parameters: the size of the membrane patch, the geometry of the patch, the number of control points connected to the patch, and the spatial placement of these control points. To ensure that coherency is present within the material system and to ensure that it utilizes the inherent potentials of the membrane, these parameters needed to be explored and investigated.

Exploring first the parameter: 'number of control points' the focus was on the behaviour of the membrane system when a membrane patch was added a varying number of control points. The conclusion that could be drawn from these experiments was that increasing this parameter resulted in additional geometrical definition and more curvature of the membrane patch, but of course at the cost of an additional number of connection points to the surrounding environment (acrylic box). From observing this experiment it became obvious that the need for physically connecting the membrane system to a potential context or site would result in a dependency to those specific surroundings. The context would in other words have to be incorporated into the membrane system and further exploration would depend on the availability of contextual connection points. This issue wasn't seen as a problem for the further exploration of the membrane system, but merely as a situation where one had to make a choice of which path to walk. Having stated earlier that the objective of this project was to generate and let evolve a material system in a contextless space, the choice of path was evident.

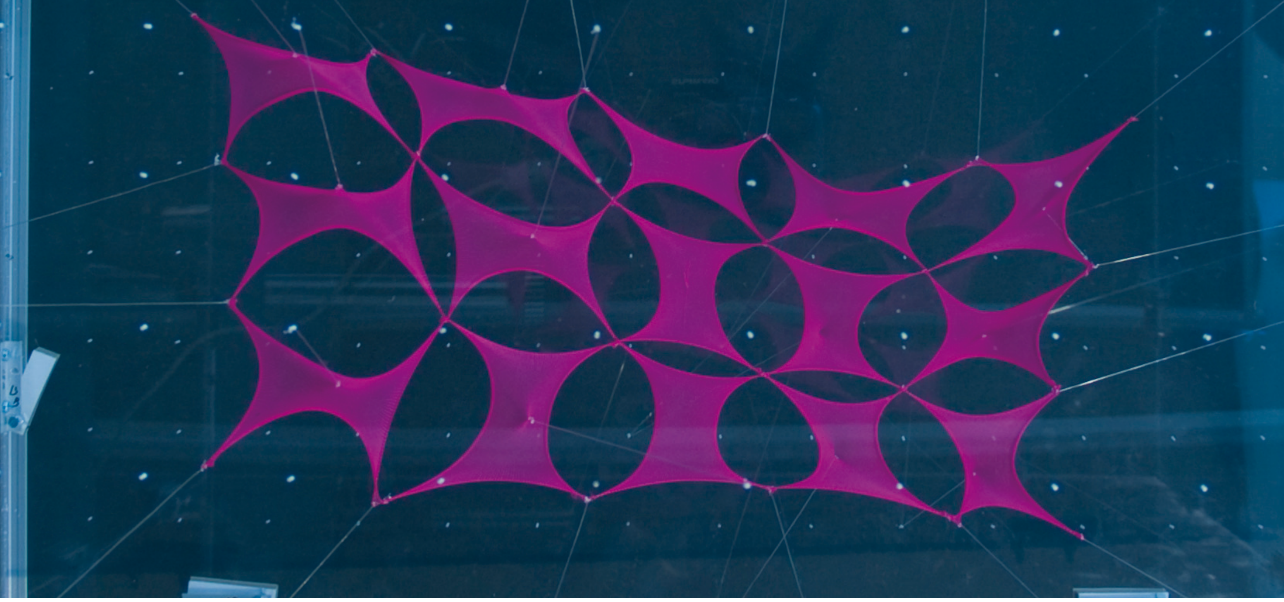
Taking a closer look at the present experiment and the current schematic setup of the membrane system it was realized that manipulating the ratio between the number of control point and the number of attachment point, a ratio that had until now been one to one, might hold some potential.

The physical setup of this experiment consisted of fifteen membrane patches arrayed in a three-by-five matrix. From the experiment it was evident that this kind of setup entails the introduction of an assembly. Looking at the diagrammatic setup of such a component-assembly configuration it is evident that the flow of information runs from the assembly and down to the individual component. This hierarchical relationship means that a manipulation of the control points belonging to the assembly affects the placement of the control points belonging to the component.

Since it is now possible to manipulate just the assembly and not all the components individually, we can talk about manipulating the membrane system globally. For example, changing the placement of the control points for the assembly will have a global effect as it changes the placement of all the control points belonging to the components as well.

To further test the potentials of such an assembly the parameters from the physical experiment was used to create a digital model enabling an examination of the effects derived by performing global manipulations on a membrane assembly. The examina-

Left: Twisted strip membrane in perspex box



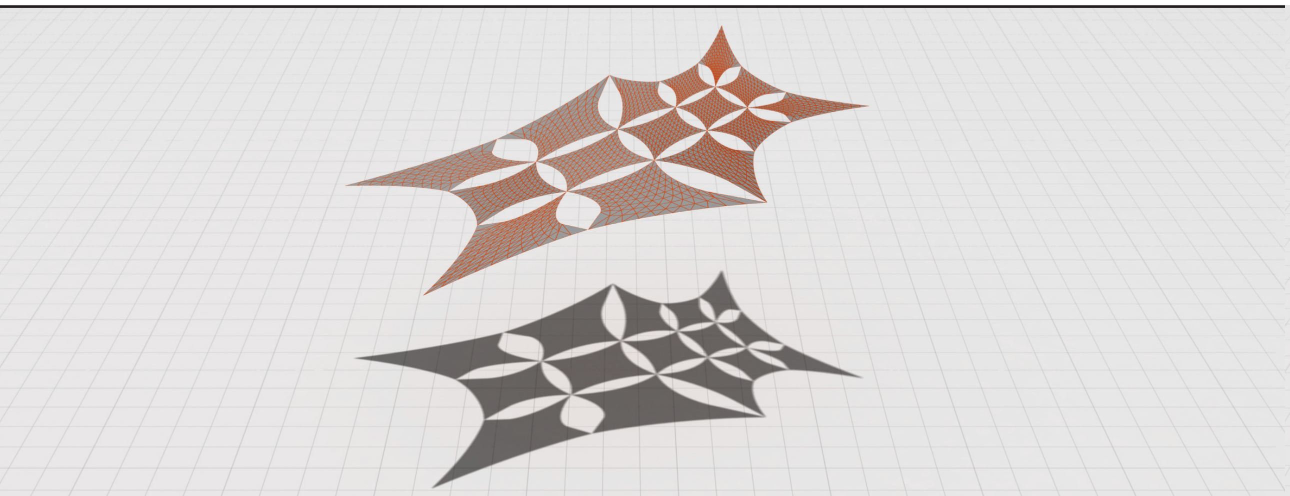
 Both Pages: Membrane Assembly in the perspex box tion was based on multiple instances of the same membrane system so as to be able to trace the systems behavioural tendencies.

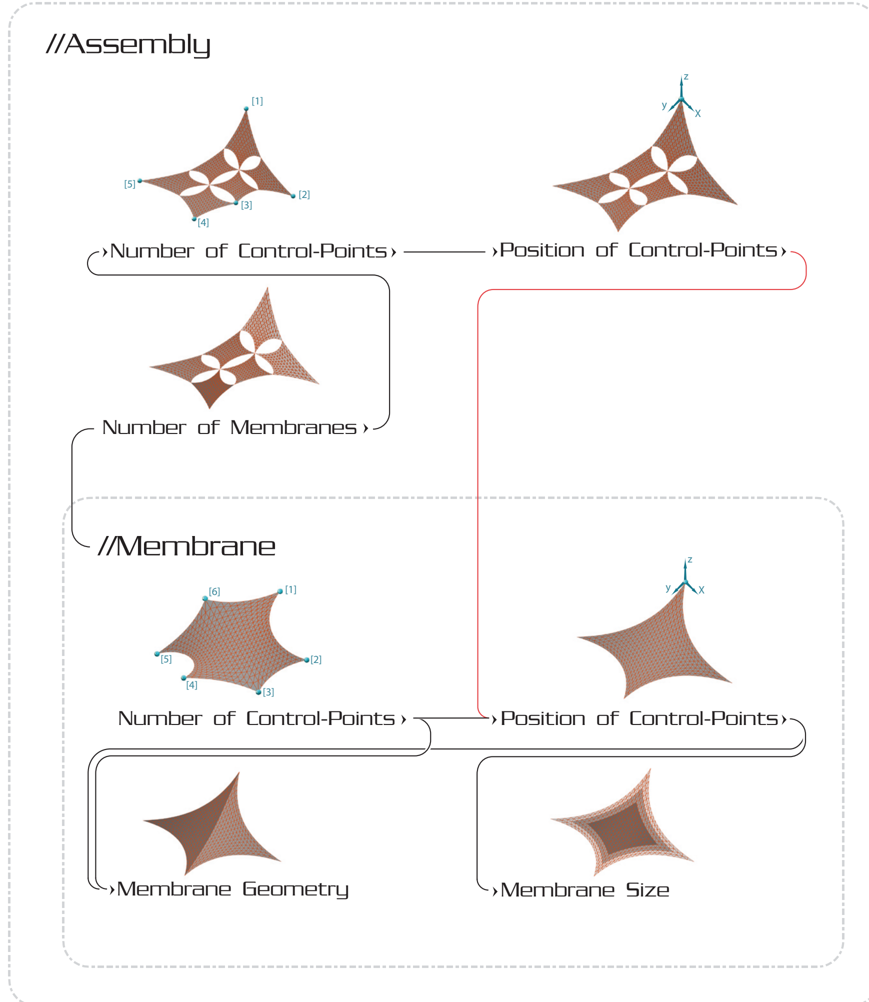
Searching to minimize the number of attachment points, the next experiment investigated the possibility of interconnectedness among the different patches, an approach that divided the control points into two groups: internal and external control points. In this way only the external control points were dependent on the presence of a context.

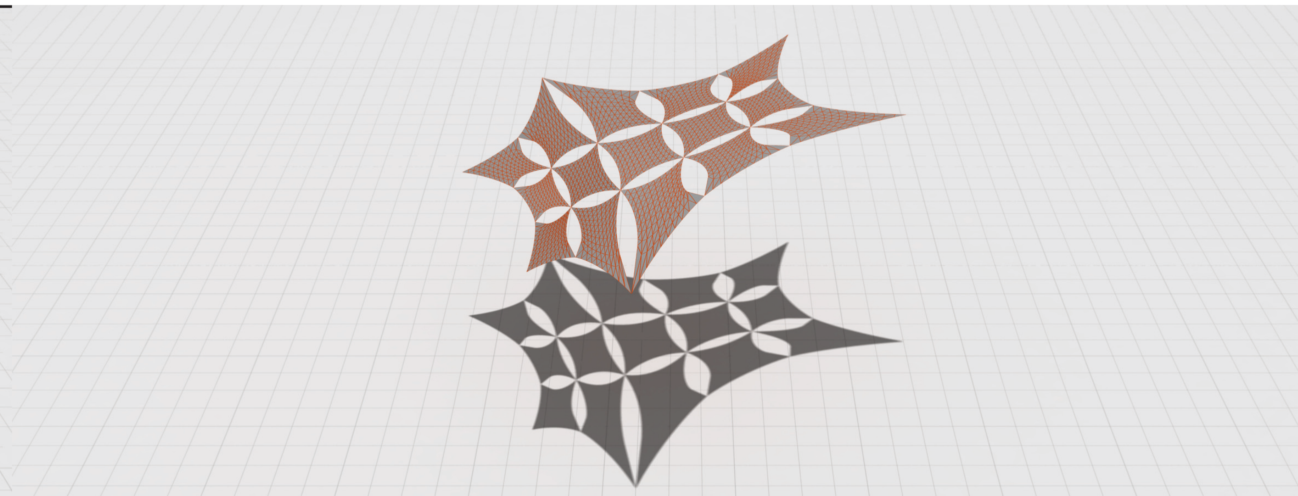
Although there is a huge potential in the construction of a system capable of generating differentiated sub-environments around its physical presence, this component-assembly system still seem to lack the potential of surprise. When taking a component and populating it along a 'host surface', which in this case is the surface spanned between the external control points, it is, depending on the number of components in the assembly, to some degree foreseeable how the result is going to be like. This is not to say that this approach cannot foster very articulated and complex structures, but more to point out that because such an approach is based on the use of a 'host surface' it is already from the beginning confined and kept 'under control' so to say.

First of all these digital experiments showed that an assembly, consisting of a fairly large amount of internal control points, could be constructed with very few connections to the environment thereby minimizing the contextual dependency. Achieving a minimal contextual dependency within the assembly would result in a more versatile material system – a context dependent system, but not a context specific system.

Interestingly, the digital explorations also unveiled another potential, namely, that when having an assembly consisting of many patches, it is possible to perform differentiated environmental modulations, enabling the creation of differentiated locally-defined environments. Manipulating the membrane assembly it's possible to change the levels of shading throughout the material system.







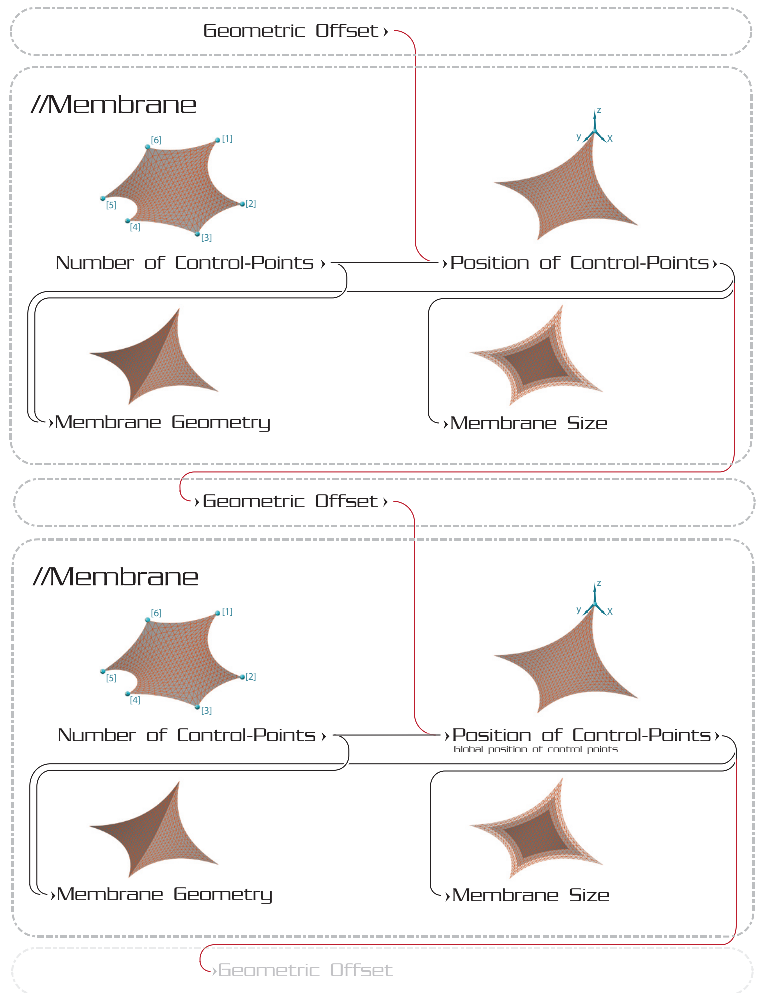
Above: Digital variation of membrane assembly. Altering the overall performance

Top: Diagram showig the hierarchy of the Membrane-Assembly system
Bottom: Digital variation of membrane assembly. Altering the overall performance

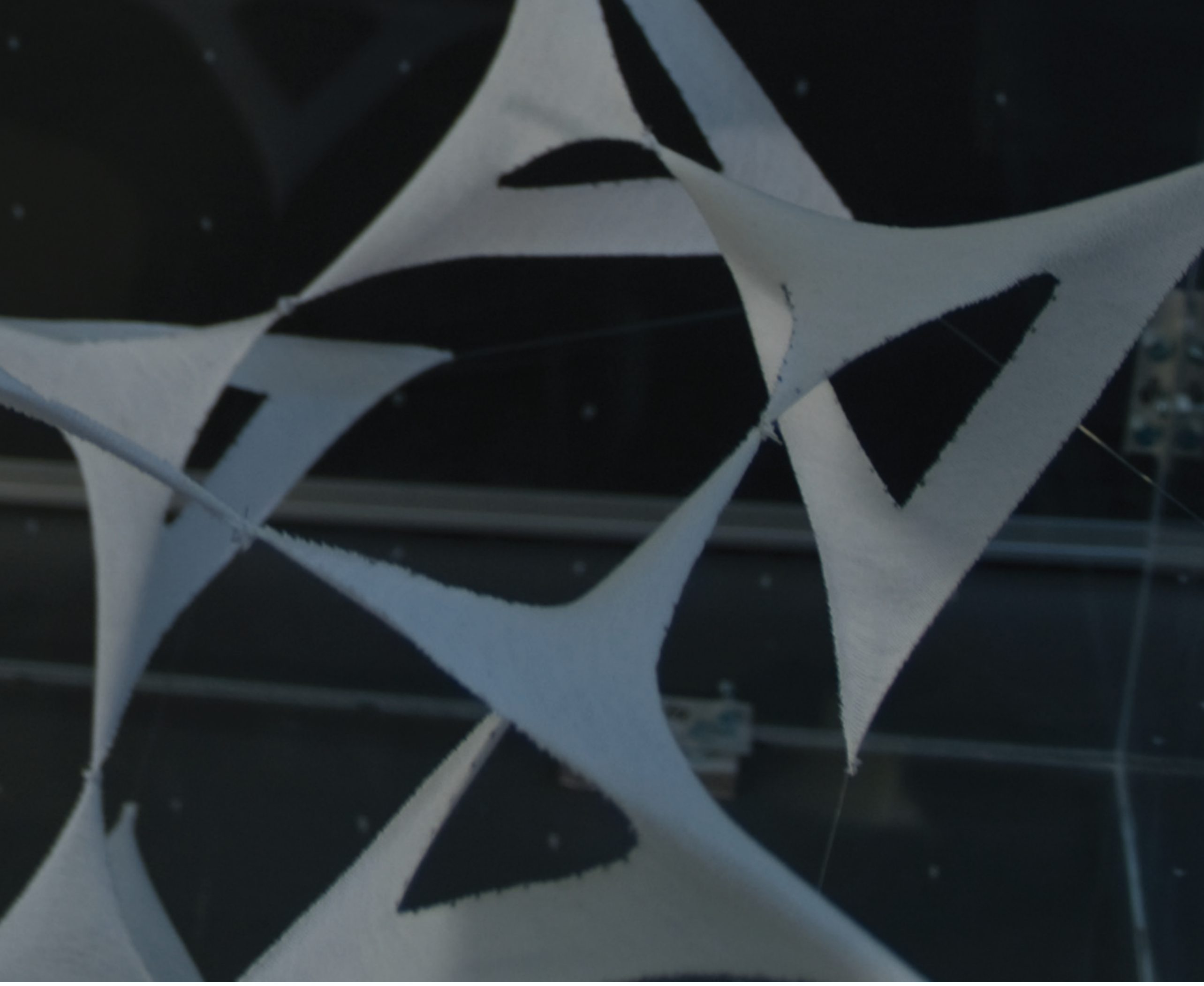
Algorithmic Growth Systems

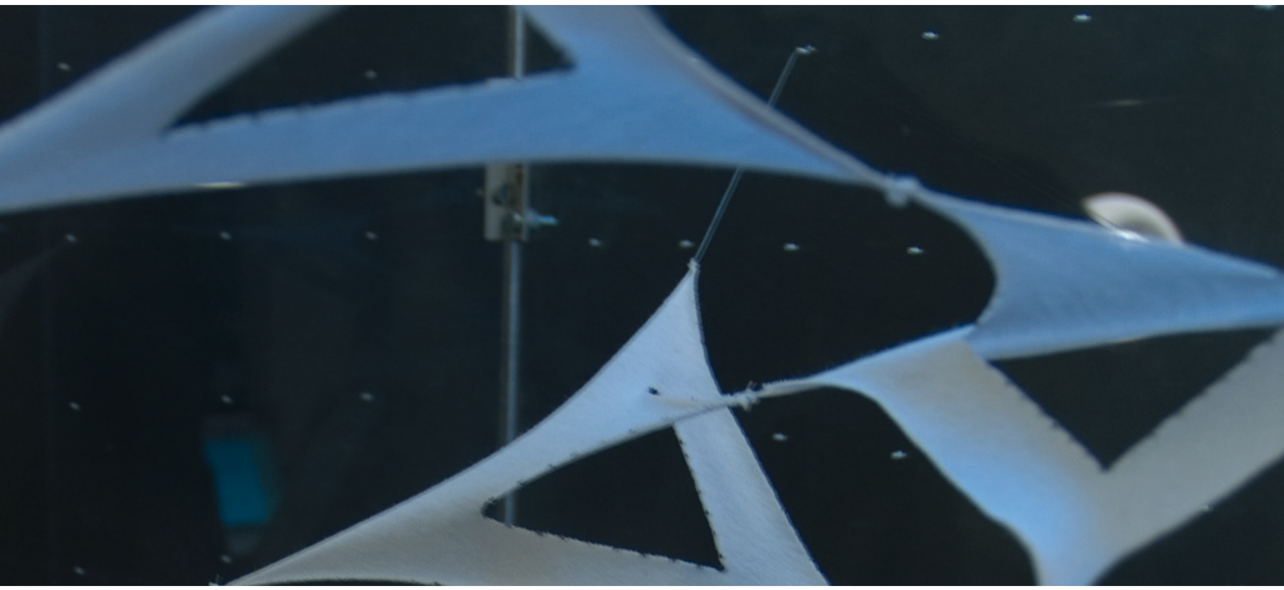
Having explored the potentials of the ‘component-host’ approach regarding its ability to populate a system it was apparent that too much control on such an approach would result in a lack of unforeseen behaviour. Seeking to design a system that contains the ability to perform differentiated environmental modulations, but did so without being linked to a ‘host surface’, the next experiment looked at a system where the population of components was controlled by an initial rule that was sequentially passed on to the next component.

This approach, inspired by the description of the Von Neumann Architecture, deals with a system where the first component holds the rule, or ‘genetic’ code, of how to manufacture a new component based on its own existence. Within such a system the designer will only control the construction of the initial rule-set, and not have any opportunity to directly manipulate the population on a global level. This is also evident when looking at the diagrammatic setup where it’s clear that no parameters are available for controlling the population on a global level.

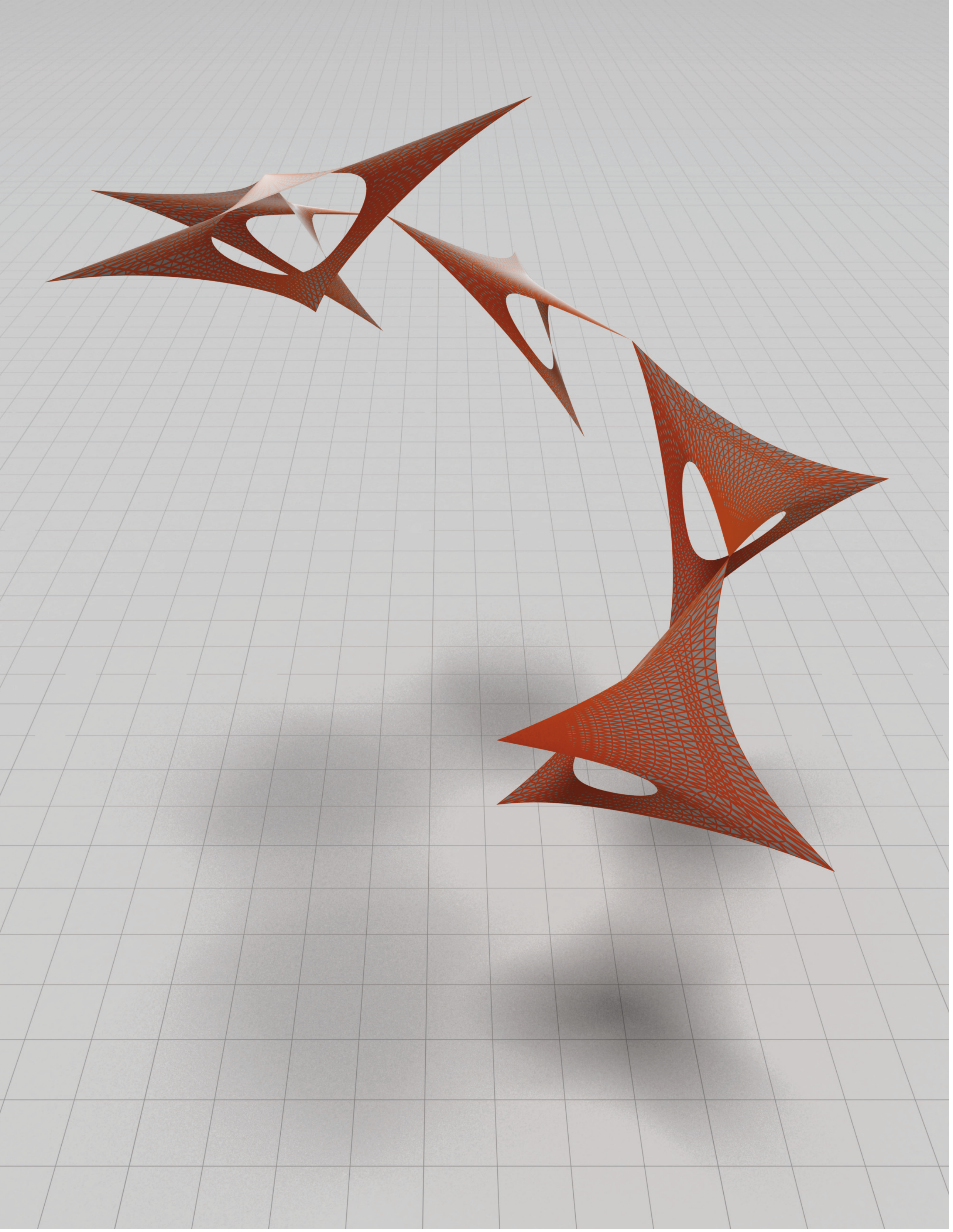


Left: Membrane test in the perspex box
Above: Diagram, showing that the relationship in the component-component system. The definition of each component lies within the predecessor for creation and the inherent desire for performance controls the geometric development.





Both pages: Membranes distributed with a growth rule. The distribution was a simple set of rules, transforming only the position of control points based on the previous component. Even though the rule was simple the outcome of the system was much more complex, and although the rule in this case was very simple it was evident that this approach had the potential of spawning unpredictable behaviour within the system.



 Above: Digital test showing an alternative growth rule

When adopting such an approach towards populating a system the focus shifts from dealing with an arrayed assembly to working with a growth system. The components geometric properties are no longer controlled by the parameters of the 'host surface', but based on an inherent rule of growth.

To investigate the act of setting up such a growth-rule, and how to incorporate it within a membrane system, a physical experiment featuring a population of identical membrane patches was constructed in the Perspex box. To be able to identify the impact, regarding the population, on the membrane system when applying such a rule, everything else than the parameter for positioning the control points were kept at a fixed value. Although the experiment focused on the implementation of a growth rule, the geometry of the membrane was also introduced to minimal holes, or 'cuts', to investigate how these could alter the behaviour of the membrane by achieving varying degrees of permeability.

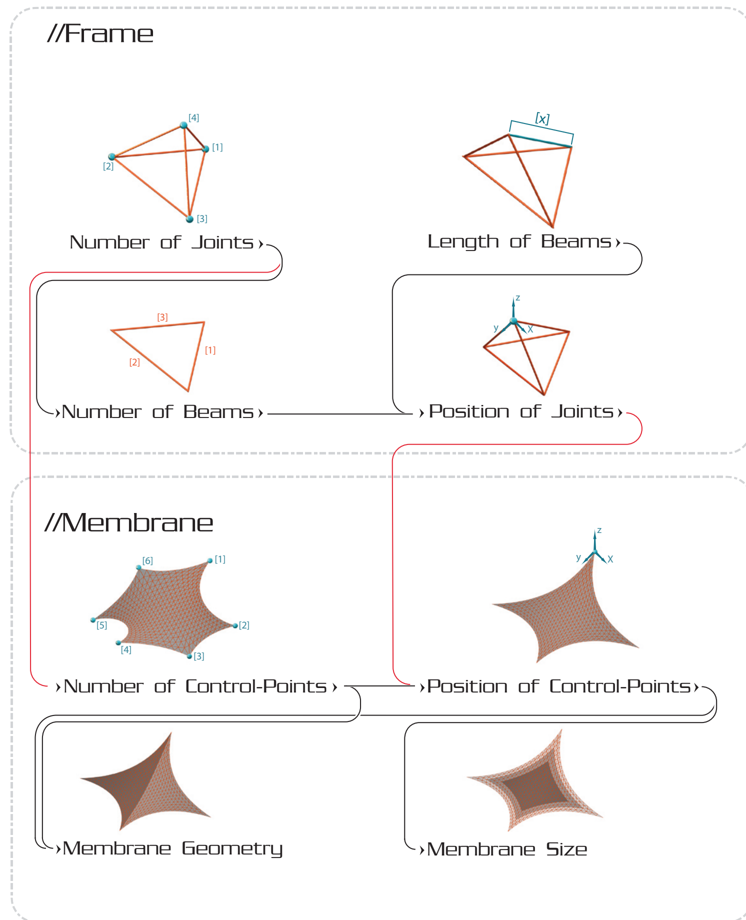
Reflecting on the outcome of this experiment it was evident that by utilizing an algorithmic description to guide the population of membranes it is possible to achieve rather complex results through simple rules. This is not to say that the geometric outcome of this particular experiment was highly complex. The experiment was guided by a simple and predictable algorithm that merely transformed the position of the coming component based on the position of the previous one, but if one were instead to utilize context-based values as inputs for all the parameters of the component, the guiding algorithm could end up being extremely complex, computing many different inputs, thereby making it rather difficult to predict the outcome. Expressing the membrane system through generative rules informed by contextual inputs, it would be possible to maintain the potential from the previous experiment; the potential of a system capable of generating differentiated local environments. When using this evolutionary approach as the generation process for architectural form, the central issue becomes the modelling of the inner logic rather than the external form.

By setting up and informing an algorithm, the computational framework, with context- and environmental based inputs, it is possible to construct a material system capable of taking into account several parameters simultaneously. Utilizing an algorithmic approach in the construction of the material system would thereby enable the generation of outputs that are based on and restricted by those logics and rules present within the computational framework.

Contextual Independent Structure

Although working with a membrane system containing a population of membrane patches enables one to modulate the environment in some degree, each membrane component is still dependent on the context. In a previous chapter it was concluded that a more versatile material system could be achieved if the contextual dependency within the assembly was minimized. This potential was further explored by the introduction of a frame system, a primary support structure, consisting of cylindrical beams.

Taking a starting point in the membrane system from the previous experiment, beams were placed between each of the control points of a single membrane patch, thereby obtaining the inherent forces of the membrane.
Implementing this kind of frame system within the material system involves a reconfiguration of the component and the hierarchies between the inherent parameters. From the construction and subsequent evaluation of a functional model of the membrane-frame component, it is obvious that the information flow within this runs from the frame system down to the membrane system. To reveal the possible effect or limitation that one parameter would have on another the new membrane-frame system needed to be exposed to a series of experiments.

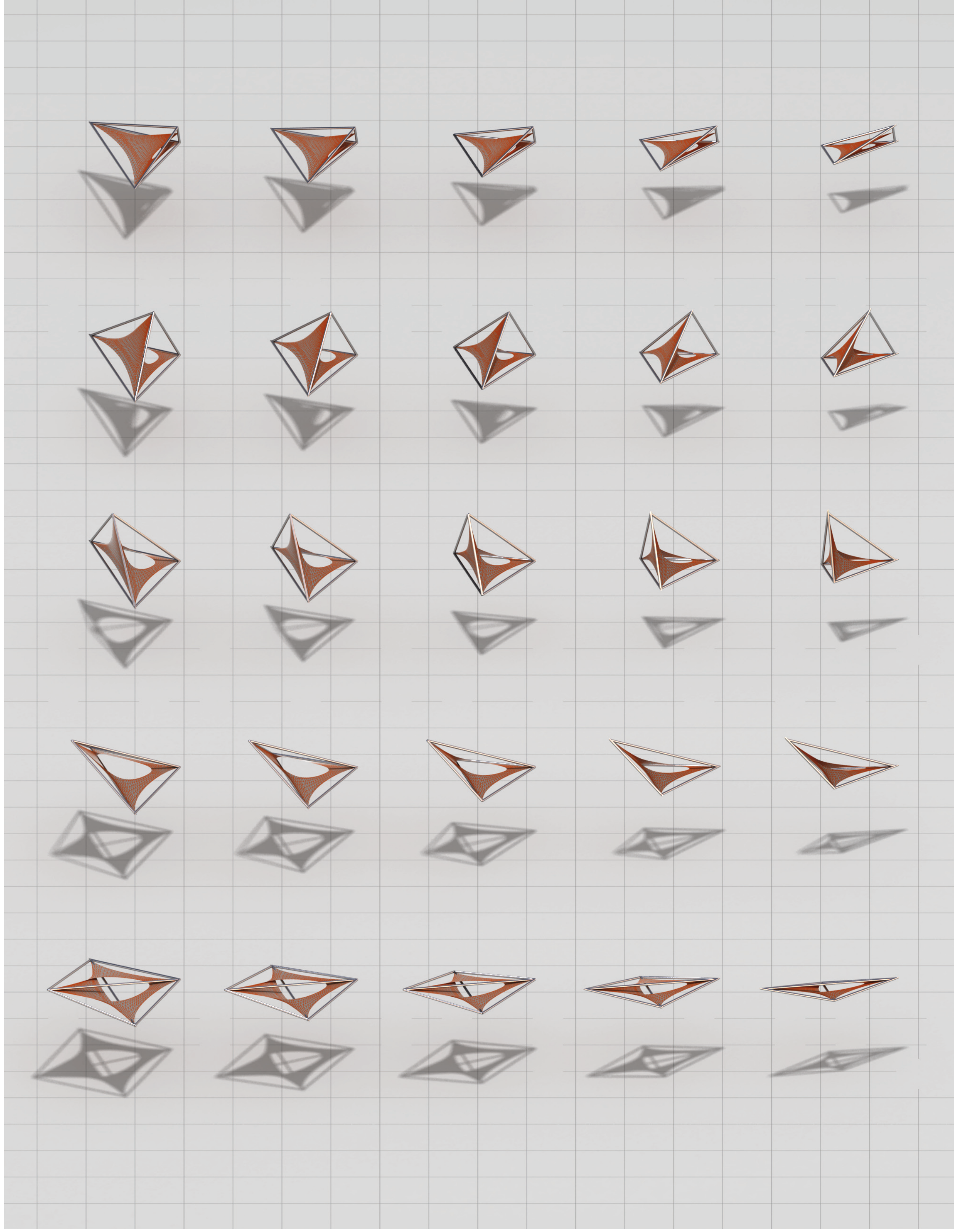


Above: Diagram of the Frame-Membrane system

As many parameters are affected by the length of the beam this parameter was tested first. Conducting the experiments in a contextless and abstract space, it is not the beam length in metric units that are manipulated, but instead the interrelated ratio between the length of the different beams. Changing the relationship between the lengths of the beams in the component, transformed the position of the membrane's control points, thereby altering the performance of the membrane itself.

On the basis of this experiment it could be concluded that the geometric behaviour of the membrane is the limiting factor when manipulating the ratio among the beams. When this parameter approximated its minimum and maximum ratio, resulting in an almost open or closed frame, the membrane became incapable of accommodating the geometric structure of the frame. The constraints, regarding the ratio between the beams, are therefore restricted by the material properties of the membrane. Applying this constraint to the membrane-frame system will ensure that, no matter the level of manipulation, the system will always retain the possibility of being manufactured without any need for post-rationalization.

Relying on the frame to act as a support structure the specific articulation of the membrane patches are now hierarchically related to the arrangement of the frame system, which entails a contextual independency of the material system. With the frame system being a rigid and self-supporting structure the material system takes on a more universal definition and enables a careful environmental modulation that surpasses the capacity of a typical membrane roof. Instead of being dependent on a given environment and its surrounding context the material system now has the potential of reacting to it – giving it a higher degree of contextual adaptability.



Above: Matrix of frame modulatin with membrane attached to show the differentiated performance

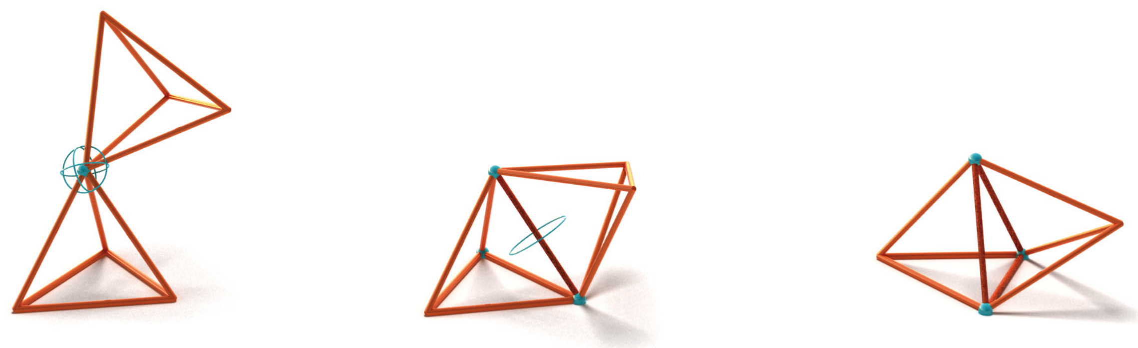
Proliferation

Still working towards the implementation of an algorithmic description that can guide the population within the material system, the central issue is, as mentioned earlier, the modelling of the inner logic rather than the external form. To be able to formulate a 'rule' as to how the membrane-frame system should proliferate, possible solutions needed to be examined.

As the frame contains structural rigidity it was natural to examine the possibility of simply connecting one frame to another. Three approaches were explored and involved using either; one, two, or three mutual joints in the interconnection of the frames. These methods resulted in membrane-frame systems that would contain three, one, and zero degrees of freedom, respectively.

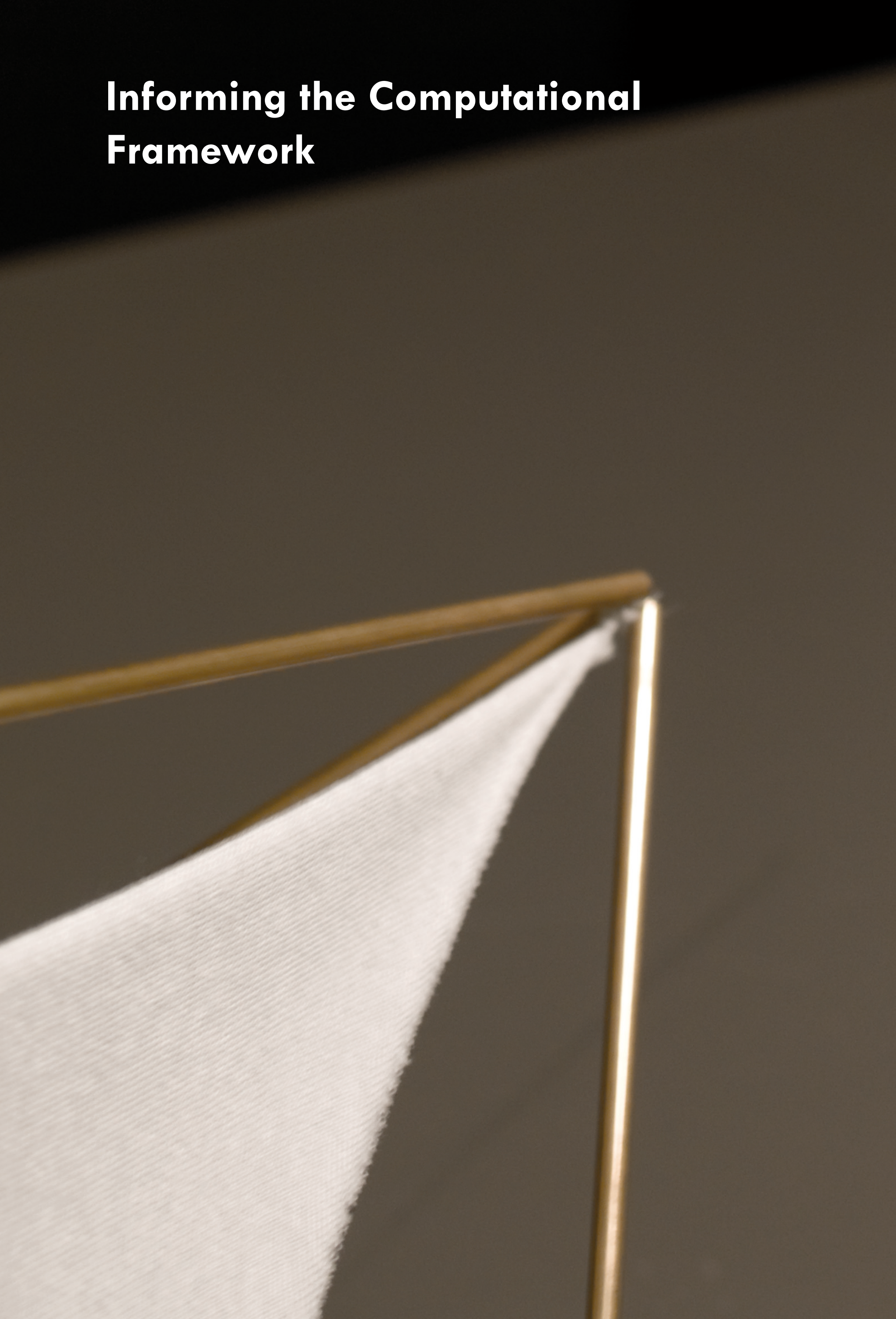
Looking from a structural point of view the method having zero degrees of freedom would create most rigidity within the system, whereas both the method of having one, and two, mutual joints would require great strength in the joints or additional elements to secure stability within the frame system.
Aiming to generate a rigid structure capable of constructing itself independent of any external factors, the approach using three mutual joints were seen as the most fitting.

To register the outcome when proliferating the membrane-frame component both a physical and a digital prototype was produced. Analysing and evaluating both prototypes revealed the effects and behaviors created from the proliferation of the specific component, thereby exposing the potential of this setup. It was found that the membrane-frame system was capable of 'growing' into a population consisting of several components, without any need for contextual requirements. From the physical prototype it could also be registered that the system derived a very rigid frame structure, and with the potential of letting the frame-membrane system be informed by environmental parameters, the frame-membrane system was set for further explorations.



Above: Frame connection examples

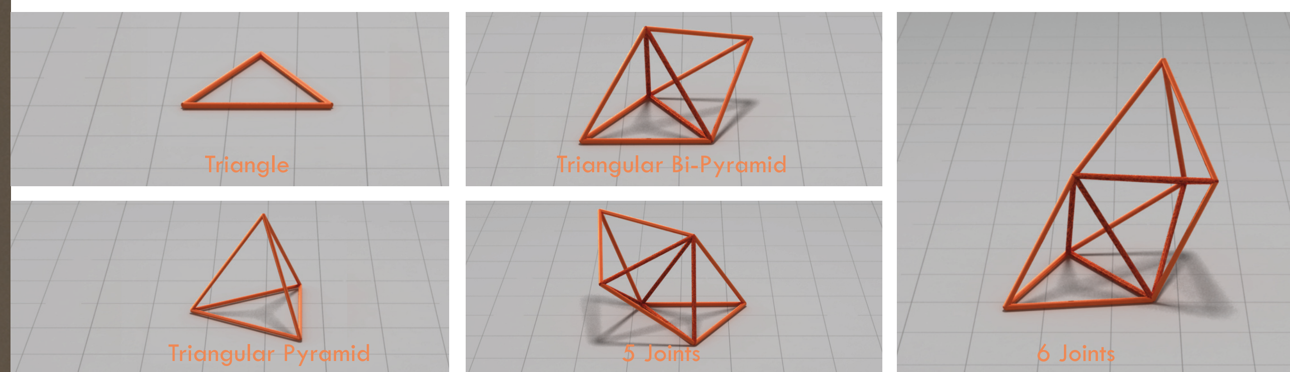
Informing the Computational Framework



As the material system now contains a frame system as well, it is necessary to explore what effects this has on the already examined parameters within the membrane system. As earlier experiments showed, the repositioning of control points directly changes the geometric behavior of the membrane patch. These control points are now directly controlled by, or associated with, the position of the frame joints. This means that in order to regulate the number or placement of control points one has to move up a level in the hierarchy and instead change the number and placement of the frame joints.

Conduction an experiment that investigates this relationship between the number of joints and the choice of membrane geometry will involve a few steps; (i) first a study of the influence that a number of joints will have on the geometric definition of the frame, (ii) then an examination of the geometric membrane shapes capable of accommodating a frame consisting of a given number of joints, (iii) and finally how the component performs when populated in a contextless space.

Starting by changing the parameter controlling the number of joints, it is evident that, if the desire is to obtain the most rigid structure with the least amount of beams, then for each additional joint three new beams has to be added the frame. With this simple experiment some of the generated frame geometries can quickly be excluded. Starting with the frame consisting of merely three joints, it is obvious that this triangular frame doesn't meet the earlier stated demands, which specified that three mutual joints were needed to generate a rigid structure. Furthermore it is also clear that this two dimensional frame will not be able to support any curvature within the membrane patch. The frames made up of four and five joints, respectively defined as a triangular pyramid and a triangular bi-pyramid, has a clear geometric definition and opposed to the previous frame they fulfill the demand regarding three mutual joints and are capable of supporting membrane curvature. Frames based on six joints or more can no longer be said to contain a clear geometric definition, and, even though they have the potential of supporting more articulated membrane geometries, they cross the threshold where a component turns into what could be considered an assembly in itself. Based on these discoveries it can be concluded that only a frame consisting of either four or five joints meets the demands of being able to perform as a geometrically simple defined frame with the ability to support membrane curvature.

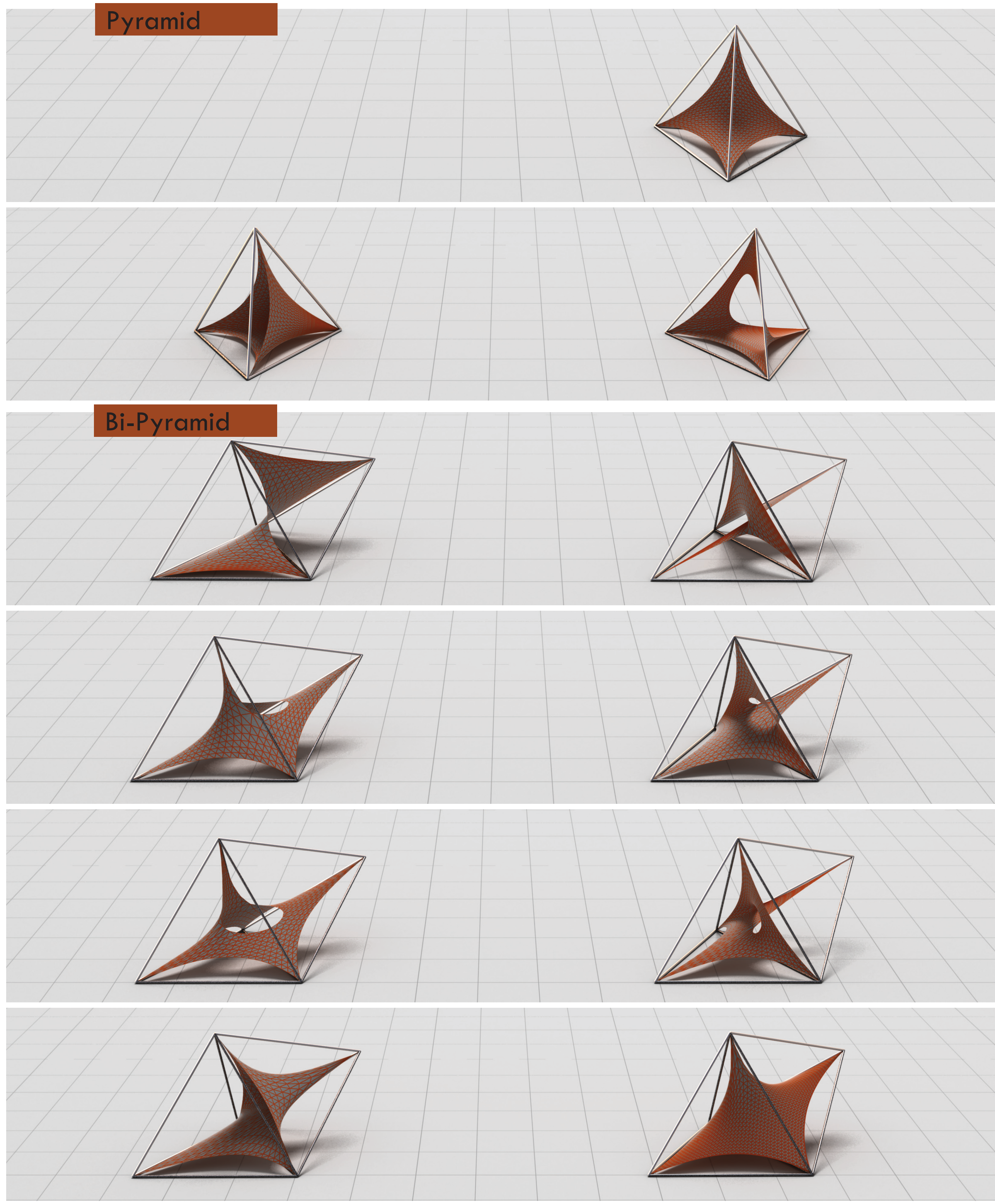


Above: Different component configurations according to number of joints and beams

Having identified the two possible frame configurations it is necessary to explore all possible membrane geometries capable of accommodating the two frames and simultaneously achieving curvature. A common rule throughout all the tests was that the relation between the number of joints and the number of membrane control points was kept at a 1:1 ratio. During the analysis of all the different membrane variants it was clear that every one of them in more or less degree possessed the ability to create shadow. Testing their flexibility and their capability to rearrange themselves within the setup of the frame revealed that some of the membrane geometries had bigger potential of adapting to local requirements. Although the aesthetic expression of the membranes is not directly used as a criterion for success it is nonetheless important that the materials employed stay true to, and express, their inherent structural, tactile and visual qualities. As one of the qualities of a membrane is its ability to visually express its inherent tensile forces, it is evident that the simpler the visual expression is the easier the forces can be 'read', resulting in a favoritism towards 'simple' membrane expressions.

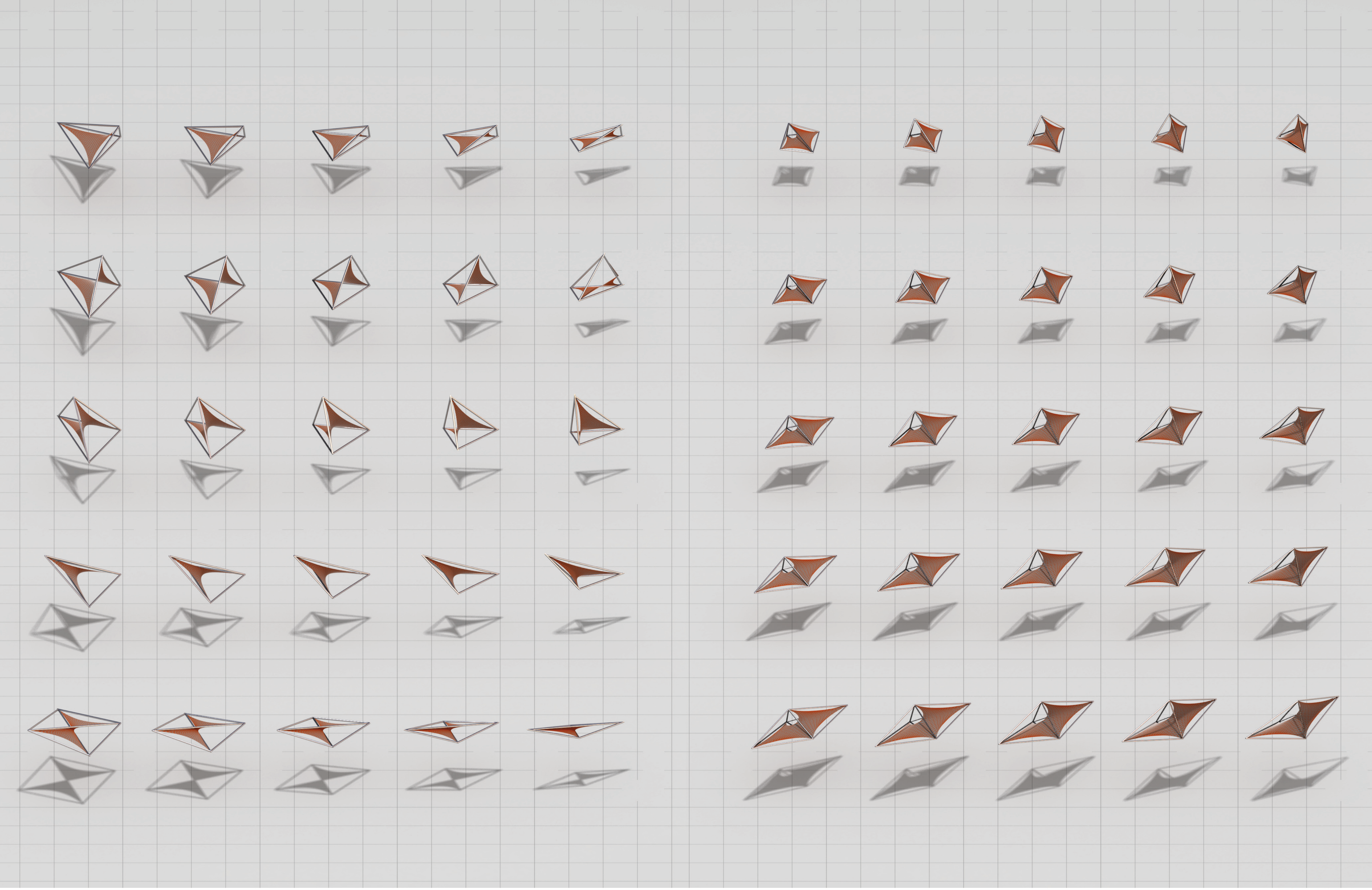
Performing these experiments on a frame with all beams being of equal length, partly ensured that the component would be able to add itself to every side on the previous frame, and the equal length of the beams also secured a homogeneous frame in which the membrane didn't rely on the orientation of the frame for it to perform differently. The equal distance between the joints of the frame had the effect the repositioning of the membranes control points didn't affect the behavior of the membrane but merely change the spatial orientation

A common feature in two of the membrane configurations is their ability to, in a simple and logical manner, express their material characteristics. Therefore these membranes, one for each frame, were selected for further exploration. Knowing that these membrane-frame components were to be proliferated into a population of varying configurations, it was important to examine their geometric behavior within their respective frame, so as to get an insight into their performative capacities. This was, as earlier, accomplished by setting up a matrix of varying frame configurations (achieved through an alteration the beam ratios), and subsequently analyzing the differentiated behaviors.

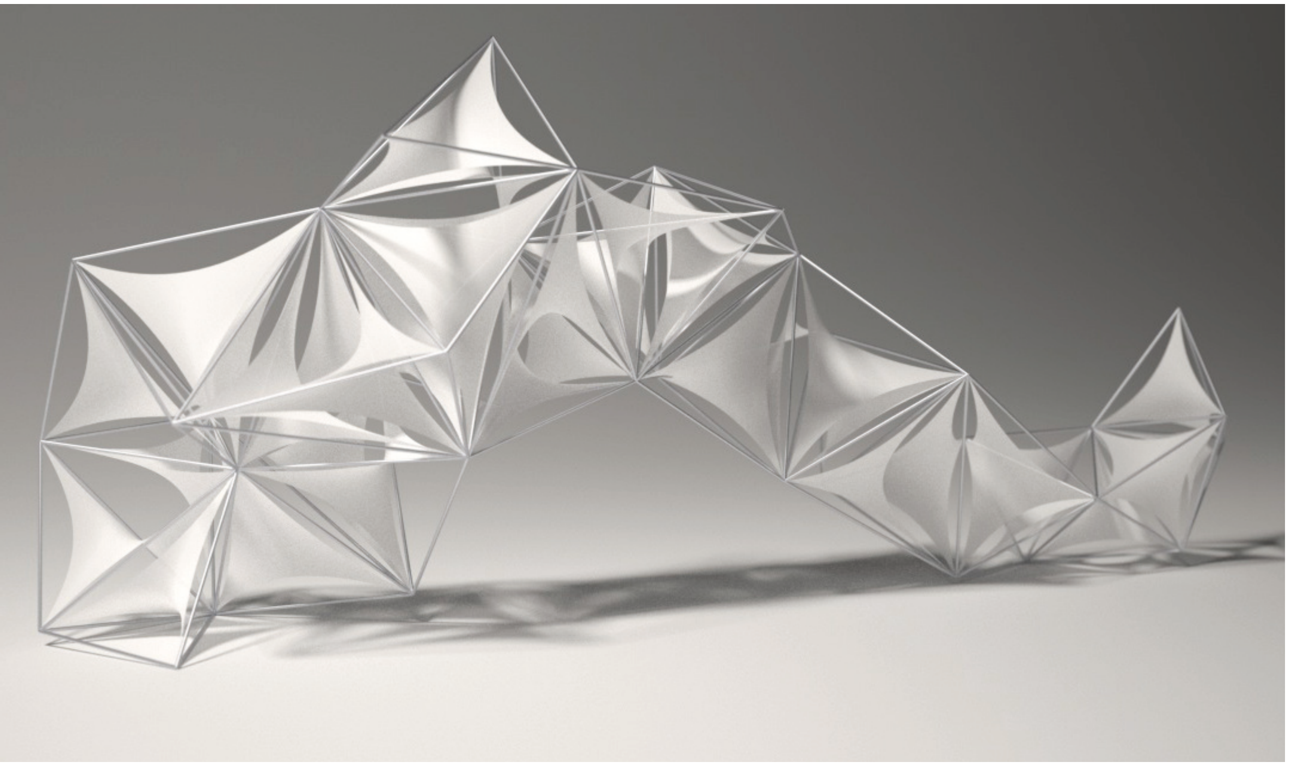


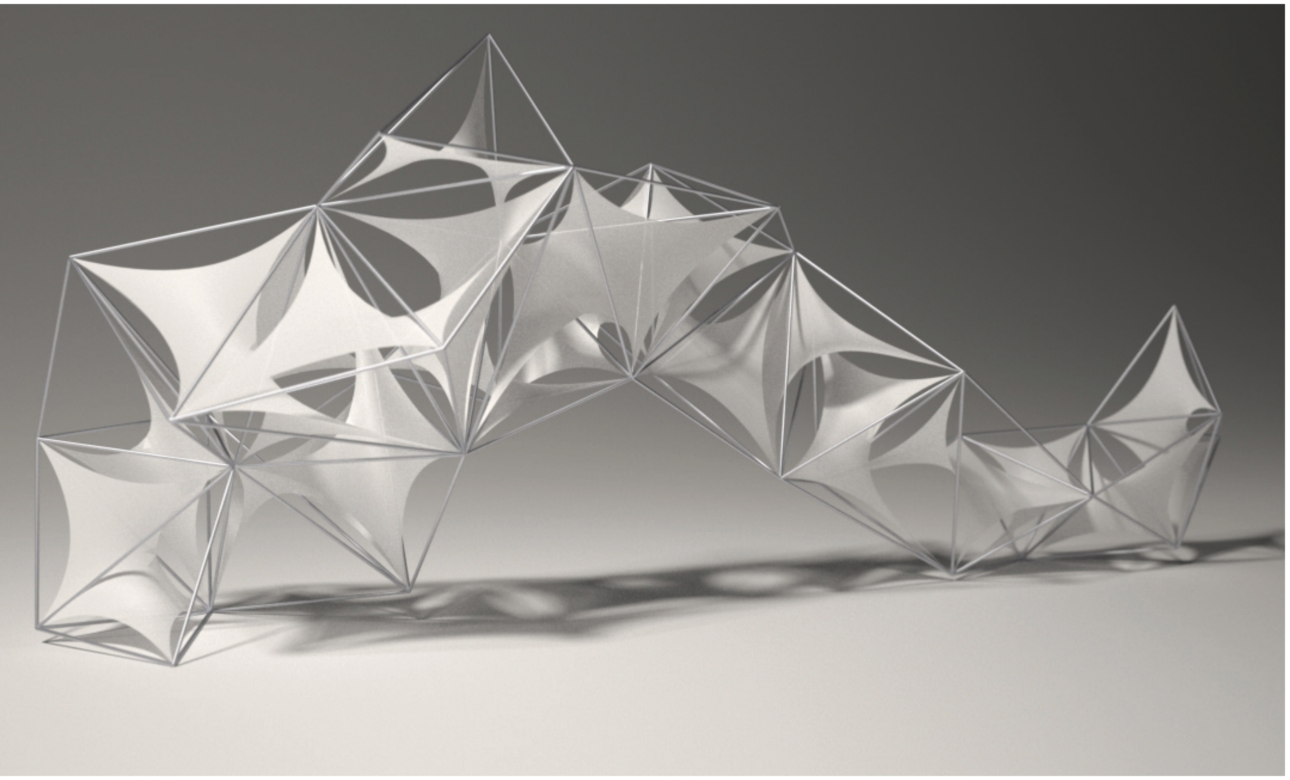
Above: Different types of membranes in the Triangular pyramid and the Bi-Triangular pyramid

Both pages: Three different membranes selected for physical testing.



Both Pages: Matrix of frame modulatin with membrane attached to show the differentiated performance for both the pyramid and the Bi-pyramid

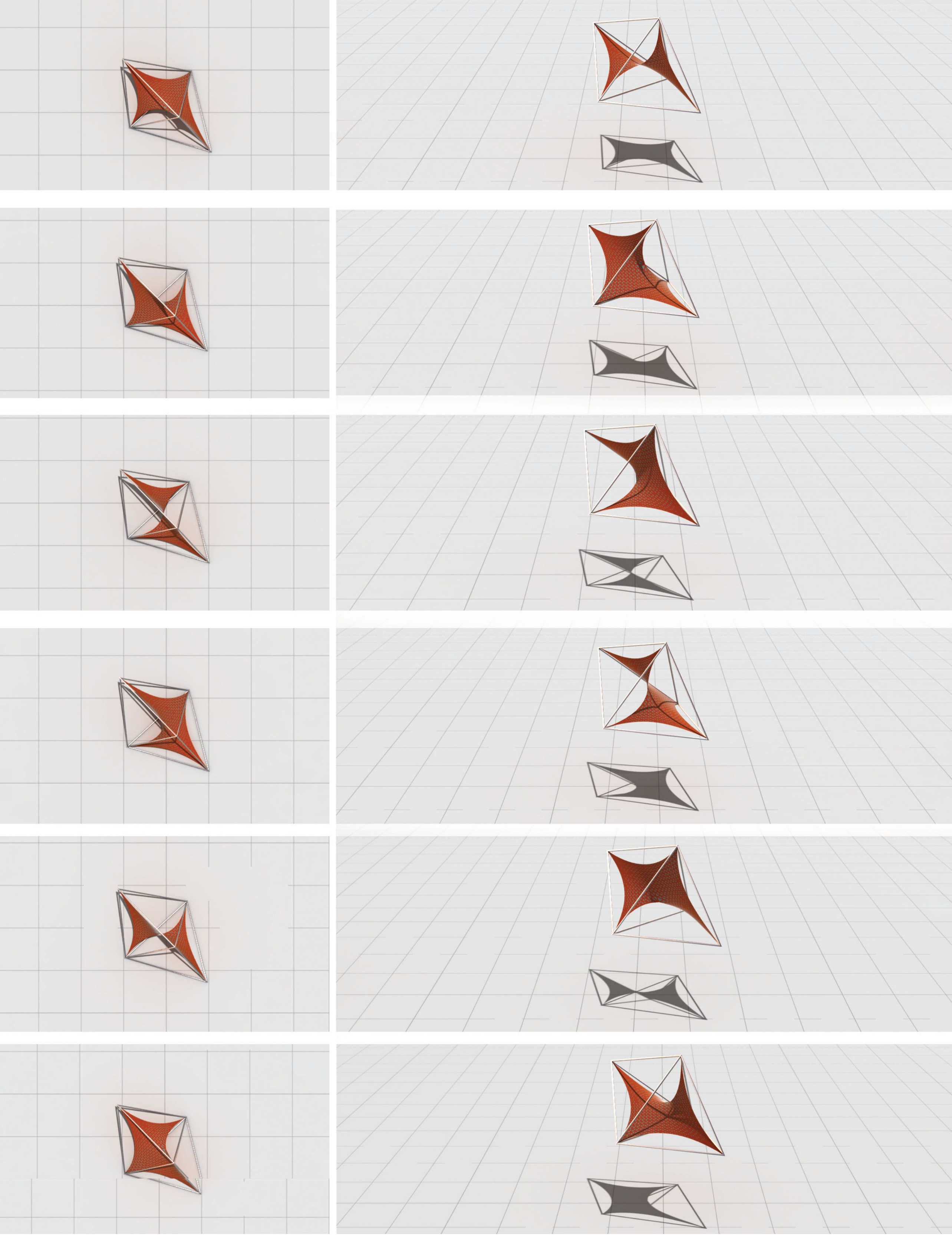




To trace what effect the behavior and the individual performance of a single membrane-frame component would have when sited amongst a larger number of varying instances, an assembly was constructed. This assembly didn't focus on how the frame system would grow but on how the two selected membranes would perform throughout the system. To examine the behavior and performance of these membranes when exposed to a population of varying frames, the frame geometry went from a number of equilateral frames, over a series of transformations, and ended up in configurations of total randomness.

Comparing the two assemblies shows that both components have the potential of creating differentiated behaviors, but a noticeable difference is that the membrane within the bi-pyramid has a bigger potential of adjusting to local contextual requirements. This effect is obtainable through a repositioning of the membrane and its orientation within the frame - altering for instance the level of shading. Evaluating the two selected components also reveals another argument for the selection of the bi-pyramid component. As also documented in one of the earlier experiments, the presence of an extra joint, and thereby an extra control point, gives the membrane within the bi-pyramid component the ability to obtain more definition and curvature, than the membrane in the pyramid frame. Based on these arguments, particularly the aspect concerning local adjustability, it was evident that the bi-pyramid was the preferred component and therefore selected for further exploration and development.

Top: Assembly with the pyramid membrane attached
Bottom: Assembly with the bi-pyramid membrane attached



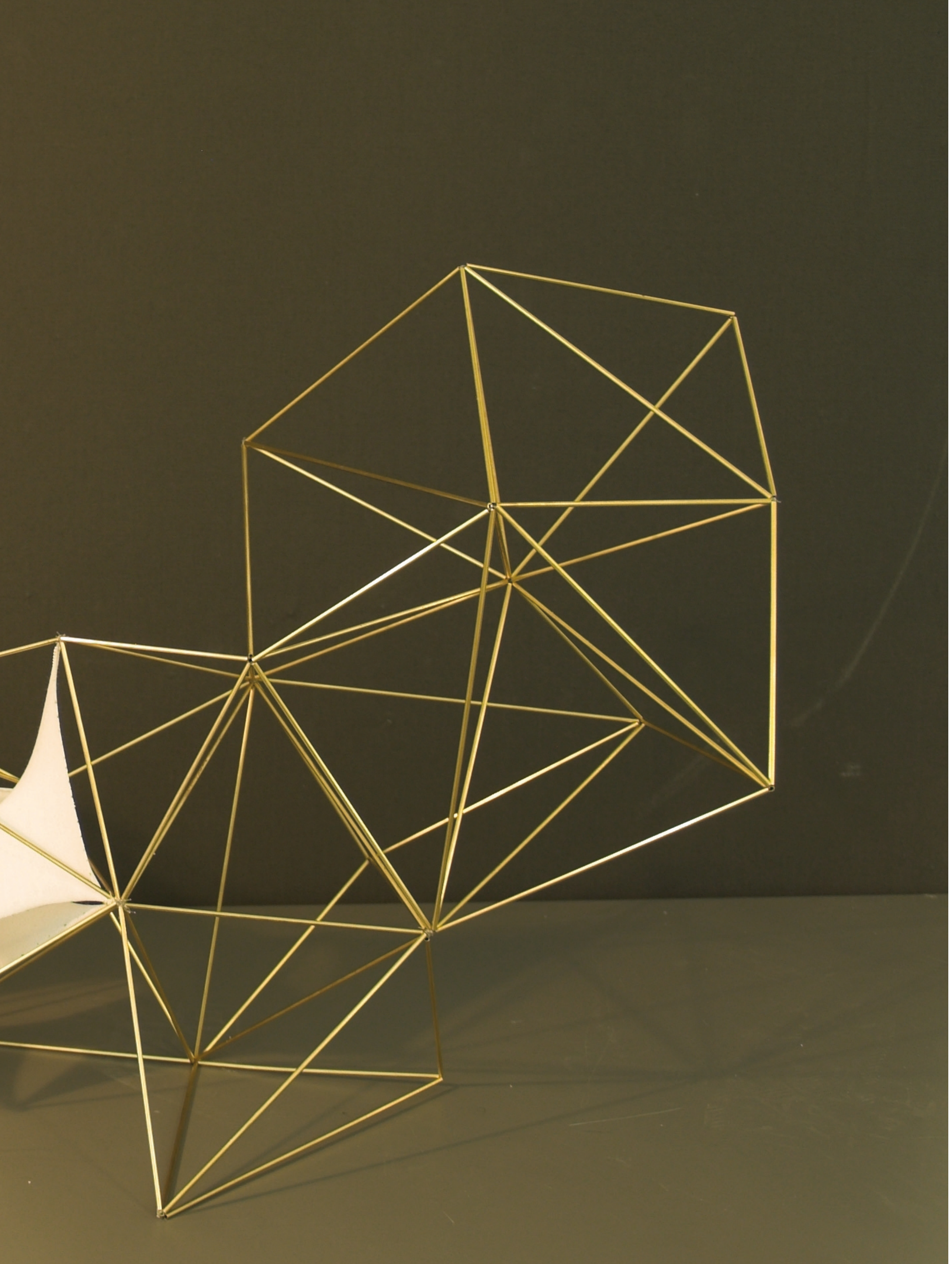
Above: Same frame configuration and orientation with the same geometrically defined membrane attached. Onle the membranes attachment order is changed

Selecting the bi-pyramid component entailed that certain parameters in the membrane-frame system needed to be locked at a certain value, so that the experimental-based findings would inform and constrain the system, ensuring that further development would take offset in these unveiled potentials. One of the parameters to be locked was the one controlling the number of joints, which, as a result of selecting the bi-pyramid, were set to a value of five. As all the parameters within the system are more or less interrelated, specifying this value also has an effect on the parameter regarding the number of beams, which is fixed at nine beams, and on the membrane geometry which is bound to the selected membrane geometry.

Continuing this process of informing the computational framework it is evident that the parameter dealt with in the earlier experiment concerning altering frame configurations, shouldn't be locked at a certain value, but rather restricted and only allowed to vary within a given minimum and maximum threshold. This restriction is derived by analyzing the range of altering ratios, where it is noticeable that certain settings result in almost two dimensional membrane surfaces that don't contain the desired curvature.

Informing the material system in a step by step manner it is ensured that the observed behaviors are extracted from the experiments and explorations of both the physical models and the digital tests. This allows for further modulation of the material system enabling additional differentiations that remains coherent with the already revealed behaviors and established restrictions.

Structural Considerations



Up until now the membrane-frame system has been develop in an abstract space without any sense of scale, so to shift towards a scale-defined environment the system has to be able to take into account the issue concerning dimensioning. The fact is that when scaling up a membrane-frame component the diameter of the beams should not increase with the same ratio as the length as this will result in a collapse of the beams due to the fact that they can no longer withstand their own weight. This problem could be solved by working out the proper ratio, so that the diameter of the beam would increase at a specified ratio higher than the length of the beam. Although this is a usable approach when dealing with the structural performance of a single component, it is not adequate when populating a number of components into an assembly. The components will in this case be structurally affected by the addition, and physical placement within the assembly, of each new component, thereby making the structural calculations far more complex and necessitate the introduction of a finite element method. Utilizing such a method within a material system would facilitate the possibility for adaptive growth, meaning that the system would be able to re-evaluate and re-configure its structural performance after the addition of each new component. Additionally containing constraints and logics regarding manufacturing, assembly, and material performance, it would be possible for the system to control or limit its own growth while keeping within the restrictions of the material system.

Although the incorporation of such a method would be feasible it is beyond the scope and time frame of this particular project. What is of interest here however is to acknowledge that scaling the component will change the interrelated ratios between some of the parameters of the membrane-frame system, and that this will affect the expression of the assembly. Having clarified the structural effects achieved through scaling the membrane-frame components, experiments can likewise be conducted to investigate the potential performance of the system related to shading effects.

Performance Explorations

```
Option Explicit
'Script written by <Rubaekbrath>
'Script version: 22.05.2008
Call Computatioal Framework
Sub Computatioal Framework
'-Parameters----------------------------------------'
Dim arrSunAngle, dblBeamLength, arrAttractor, strAttractor
Dim dblDistance, dblMaxHeight, dblAttractorForce
Dim strFlowCurve, dblFlowCurveForce, arrFlowCurvePoints
Dim dblSizeControllerArea, arrSizeController, strSizeController
dblDistance = 5.774 'Gives sides that are 10 units long'
arrSunAngle = Array(0.07,-0.569,0.819)
dblMaxHeight = 50
dblAttractorForce = 60
dblSizeControllerArea = 30
Rhino.Print("Pick Attractor: ")
arrAttractor = Rhino.GetPoint
strAttractor = Rhino.AddPoint(arrAttractor)
Rhino.Print("Pick SizeController: ")
arrSizeController = Rhino.GetPoint
strSizeController = Rhino.AddPoint(arrSizeController)
arrFlowCurvePoints = Rhino.GetPoints(,,"Select points for Flowcurve.")
strFlowCurve = Rhino.AddInterpCurve (arrFlowCurvePoints , 3)
dblFlowCurveForce = Rhino.GetInteger("Type in FlowCurve Force: ")
'-Calculate BeamLenght----------------------------------------------------------'
Dim dblNumberofDiffBeamSizes, dblNumberofBeams
Dim BeamSize, dblFactor
Dim arrSelectedCenterPoint
Dim dblBeamLenght, SelectedBeamSize()
dblNumberofDiffBeamSizes = Rhino.GetInteger("How many different beam sizes do you need?: ")
dblNumberofBeams = (dblNumberofDiffBeamSizes-1)
For i = 0 To dblNumberofBeams
BeamSize = Rhino.GetReal("Pick size for beam " & (i+1))
ReDim Preserve SelectedBeamSize(i)
SelectedBeamSize(i) = BeamSize
Next
dblFactor = 1/(Ubound(SelectedBeamSize)+1)
Rhino.Print "dblFactor: " & dblFactor
arrSelectedCenterPoint = Rhino.GetPoint("select CenterPoint:")
dblBeamLenght = CalcBeamLengthFirstPoint (dblSizeControllerArea, arrSelectedCenterPoint, dblNumberofBeams, SelectedBeamSize,
arrSizeController)
'-Get point 01, 02 and 03----------------------------------------------------'
Dim arrBasePoints, arrPoint01, arrPoint02, arrPoint03
arrBasePoints = BasePoints (arrSelectedCenterPoint, dblBeamLenght)
arrPoint01 = arrBasePoints(0)
arrPoint02 = arrBasePoints(1)
arrPoint03 = arrBasePoints(2)
'-Find Point04 (by using sphere-intersecions)-----------------------------------------------'
Dim arrIntersectionPoints, dblDistancePoint04a, dblDistancePoint04b, strPoint04, arrPoint04
arrIntersectionPoints = IntersectionPoints (arrBasePoints, dblBeamLenght)
dblDistancePoint04a = Rhino.Distance(arrIntersectionPoints(0,1), arrSunAngle)
'Rhino.Print "Distance Point04a: " & dblDistancePoint04a
dblDistancePoint04b = Rhino.Distance(arrIntersectionPoints(1,1), arrSunAngle)
'Rhino.Print "Distance Point04b: " & dblDistancePoint04b
If dblDistancePoint04a > dblDistancePoint04b Then
arrPoint04 = arrIntersectionPoints(1,1)
Else
arrPoint04 = arrIntersectionPoints(0,1)
End If
strPoint04 = Rhino.AddPoint(arrPoint04)
'Rhino.Print "Point04: " & Rhino.Pt2Str(arrPoint04)
'-START LOOP----'
Dim i
For i = 0 To 14
'-Find Point04 (by using sphere-intersections)------------------------------------------------------'
If i >= 1 Then
Dim dblDistancePoint04
dblBeamLenght = CalcBeamLenght(dblSizeControllerArea, arrBestSurfaceCentroid, dblNumberofBeams, SelectedBeamSize, arrSizeCon-
troller)
arrIntersectionPoints = IntersectionPoints (arrBasePoints, dblBeamLenght)
If IsArray (arrPoint04) Then
'Rhino.Print "arrBestSurfaceFreePoint is not an array!"
Else
'Rhino.Print "arrBestSurfaceFreePoint is an array!"
End If
dblDistancePoint04a = Rhino.Distance(arrIntersectionPoints(0,1), arrPoint04)
'Rhino.Print "Distance Point04a: " & dblDistancePoint04a
dblDistancePoint04b = Rhino.Distance(arrIntersectionPoints(1,1), arrPoint04)
'Rhino.Print "Distance Point04b: " & dblDistancePoint04b
If dblDistancePoint04a > dblDistancePoint04b Then
arrPoint04 = arrIntersectionPoints(0,1)
Else
arrPoint04 = arrIntersectionPoints(1,1)
End If
strPoint04 = Rhino.AddPoint(arrPoint04)
dblDistancePoint04 = Rhino.Distance(arrPoint04, arrPoint04)
'Rhino.Print "Distance Point04: " & dblDistancePoint04
End If
'-Draw sides 01-03----------------------------------------------------'
Dim arrPoints, arrPointsSides
Dim arrContainerSide01, arrContainerSide02, arrContainerSide03
arrPoints = Array(arrPoint01, arrPoint02, arrPoint03, arrPoint04)
arrPointsSides = DrawSidesFrom4Points(arrPoints)
arrContainerSide01 = arrPointsSides(0)
arrContainerSide03 = arrPointsSides(2)
'-Evaluate sides 01, 02 and 03------------------------------------------------'
Dim arrSurfaceNormals, dblSunSurface, arrSurfaceBase, arrSurfaceFreePoint
Dim dblSunSurfaceScore, arrCurvePoints, dblAttractorSurface, dblAttractorSurfaceScore
Dim ScoreSide01, ScoreSide02, ScoreSide03
Dim dblFlowCurveDegrees, dblFlowScore, dblSunSurfaceDegrees, arrFlowCurvePoint
arrSurfaceNormals = FindSurfaceNormals (arrContainerSide01, arrContainerSide02, arrContainerSide03)
ReDim Preserve    arrContainerSide01(5)
arrContainerSide01(5) = arrSurfaceNormals(0)
ReDim Preserve    arrContainerSide02(5)
arrContainerSide02(5) = arrSurfaceNormals(1)
ReDim Preserve    arrContainerSide03(5)
arrContainerSide03(5) = arrSurfaceNormals(2)
dblSunSurfaceDegrees = SunSurfaceDegrees (arrContainerSide01, arrContainerSide02, arrContainerSide03, arrSunAngle)
ReDim Preserve    arrContainerSide01(6)
arrContainerSide01(6) = dblSunSurfaceDegrees(0)
ReDim Preserve    arrContainerSide02(6)
arrContainerSide02(6) = dblSunSurfaceDegrees(1)
ReDim Preserve    arrContainerSide03(6)
arrContainerSide03(6) = dblSunSurfaceDegrees(2)
dblSunSurfaceScore = SunSurfaceScore (arrContainerSide01, arrContainerSide02, arrContainerSide03, dblMaxHeight)
ReDim Preserve    arrContainerSide01(7)
arrContainerSide01(7) = dblSunSurfaceScore(0)
ReDim Preserve    arrContainerSide02(7)
arrContainerSide02(7) = dblSunSurfaceScore(1)
ReDim Preserve    arrContainerSide03(7)
arrContainerSide03(7) = dblSunSurfaceScore(2)
dblAttractorSurface = AttractorSurfaceDegrees (arrContainerSide01, arrContainerSide02, arrContainerSide03, arrAttractor)
ReDim Preserve    arrContainerSide01(8)
arrContainerSide01(8) = dblAttractorSurface(0)
ReDim Preserve    arrContainerSide02(8)
arrContainerSide02(8) = dblAttractorSurface(1)
ReDim Preserve    arrContainerSide03(8)
arrContainerSide03(8) = dblAttractorSurface(2)
dblAttractorSurfaceScore = AttractorSurfaceScore (arrContainerSide01, arrContainerSide02, arrContainerSide03, arrAttractor, dblAttractorForce)
ReDim Preserve    arrContainerSide01(9)
arrContainerSide01(9) = dblAttractorSurfaceScore(0)
ReDim Preserve    arrContainerSide02(9)
arrContainerSide02(9) = dblAttractorSurfaceScore(1)
ReDim Preserve    arrContainerSide03(9)
arrContainerSide03(9) = dblAttractorSurfaceScore(2)
arrFlowCurvePoint = FlowCurvePoint (arrContainerSide01, arrContainerSide02, arrContainerSide03, strFlowCurve)
ReDim Preserve    arrContainerSide01(10)
arrContainerSide01(10) = arrFlowCurvePoint(0)
ReDim Preserve    arrContainerSide02(10)
arrContainerSide02(10) = arrFlowCurvePoint(1)
ReDim Preserve    arrContainerSide03(10)
arrContainerSide03(10) = arrFlowCurvePoint(2)
dblFlowCurveDegrees = FlowCurveDegrees (arrContainerSide01, arrContainerSide02, arrContainerSide03, strFlowCurve)
ReDim Preserve    arrContainerSide01(11)
arrContainerSide01(11) = dblFlowCurveDegrees(0)
ReDim Preserve    arrContainerSide02(11)
arrContainerSide02(11) = dblFlowCurveDegrees(1)
ReDim Preserve    arrContainerSide03(11)
arrContainerSide03(11) = dblFlowCurveDegrees(2)
dblFlowScore = FlowCurveScore (arrContainerSide01, arrContainerSide02, arrContainerSide03, dblFlowCurveForce)
ReDim Preserve    arrContainerSide01(12)
arrContainerSide01(12) = dblFlowScore(0)
ReDim Preserve    arrContainerSide02(12)
arrContainerSide02(12) = dblFlowScore(1)
ReDim Preserve    arrContainerSide03(12)
arrContainerSide03(12) = dblFlowScore(2)
ScoreSide01 = arrContainerSide01(7)+arrContainerSide01(9)+arrContainerSide01(12)
'Rhino.Print "ScoreSide01: " & Int(ScoreSide01)
ScoreSide02 = arrContainerSide02(7)+arrContainerSide02(9)+arrContainerSide01(12)
'Rhino.Print "ScoreSide02: " & Int(ScoreSide02)
ScoreSide03 = arrContainerSide03(7)+arrContainerSide03(9)+arrContainerSide01(12)
'Rhino.Print "ScoreSide03: " & Int(ScoreSide03)
'-Choose best side-----------------------------------------------------------'
Dim dblHighestScore, arrBestSurface
dblHighestScore = Rhino.Max (Array(ScoreSide01, ScoreSide02, ScoreSide03))
'Rhino.Print "HighestScore: " & Int(dblHighestScore)
If ScoreSide01 = dblHighestScore Then
arrBestSurface = arrContainerSide01
Else
If ScoreSide02 = dblHighestScore Then
arrBestSurface = arrContainerSide02
Else
arrBestSurface = arrContainerSide03
End If
End If
'-Find Centroid-------------------------------------------------'
Dim arrBestSurfaceCentroid
arrBestSurfaceCentroid = Rhino.CurveAreaCentroid(arrBestSurface(4))
If IsArray(arrBestSurfaceCentroid) Then
Rhino.AddPoint (arrBestSurfaceCentroid(0))
End If
'-Mirror point04 to create point05-----------------------------------------------'
Dim arrPoint05, strPoint05
arrPoint05 = FindPoint05 (arrBestSurface)
strPoint05 = Rhino.AddPoint(arrPoint05)
'-Draw sides 04-06--------------------------------------------------'
Dim arrContainerSide04, arrContainerSide05, arrContainerSide06
arrPoints = Array(arrBestSurface(0),arrBestSurface(1), arrBestSurface(2), arrPoint05)
```

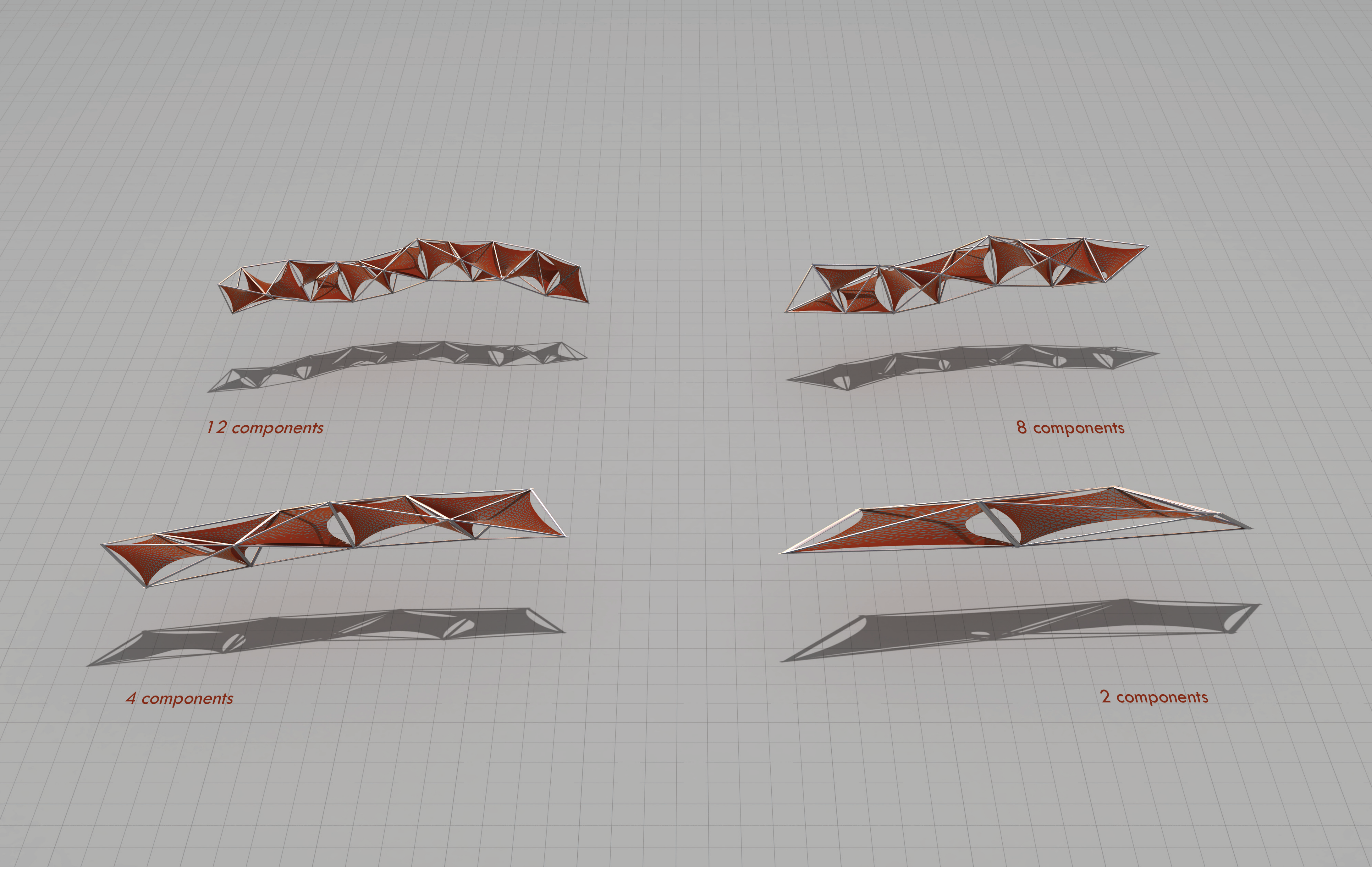
Size and performance

To explore the relationship between the size of the components and the effects regarding the generation of shadows, a number of different sized components were constructed in a digital environment. What is evident in the scene testing shadow effects is that the larger the components, the larger the membrane patches, and the less the opportunity for local variation. As the word 'local' is relative it will of course vary according to the site in question, but nonetheless it is a fact that a change in scale will affect the level of intimacy as well as the way the assembly is perceived.

Manipulating this parameter enables the generation of varying local environments, in areas with large components present the will be big fields covered with shadow or direct sunlight, while areas surrounded with small components will have a graduated pattern of shadow and direct sunlight. It will thereby be possible for the designer to define 'zones' or 'fields' wherein the membrane-frame system will be more or less likely to generate a certain sized component and in doing so foster differentiated local behaviours.

Dealing with the aspect of scale these experiments can inform the computational framework about the different performative effects that can be obtained by altering the parameter that controls the size of the frame. Based on the experiments it can also be derived that the allowed range of beam dimensions should inform the membrane-frame system of its limitations when it comes to their maximum and minimum size. These limitations could be introduced by two different inputs: the manufacturing constraints and the scale of the site. Depending on the available manufacturing facilities there could for instance be some constraints connected to the size of the beams. The size of the given site would also have to be taken into account. Using beams with a diameter up to as much as one meter would possibly be fitting for a system set up to cover part of a football field, but completely misplaced when covering a small area of a playground.

From these explorations into the aspects of size it can be concluded that the material system, through its inherent relationships, is capable of accommodating varying scales and therefore contains the potential of being utilized in sites of diverse extents. Additionally, the ability to vary in size also facilitates a generation of differentiated local behaviours, rendering it possible to generate various degrees of shadowing effects.



Both Pages: Modulation of the component sizes within the membrane showing the alteration in how the same area is covered according to how many membranes the assembly consist of.

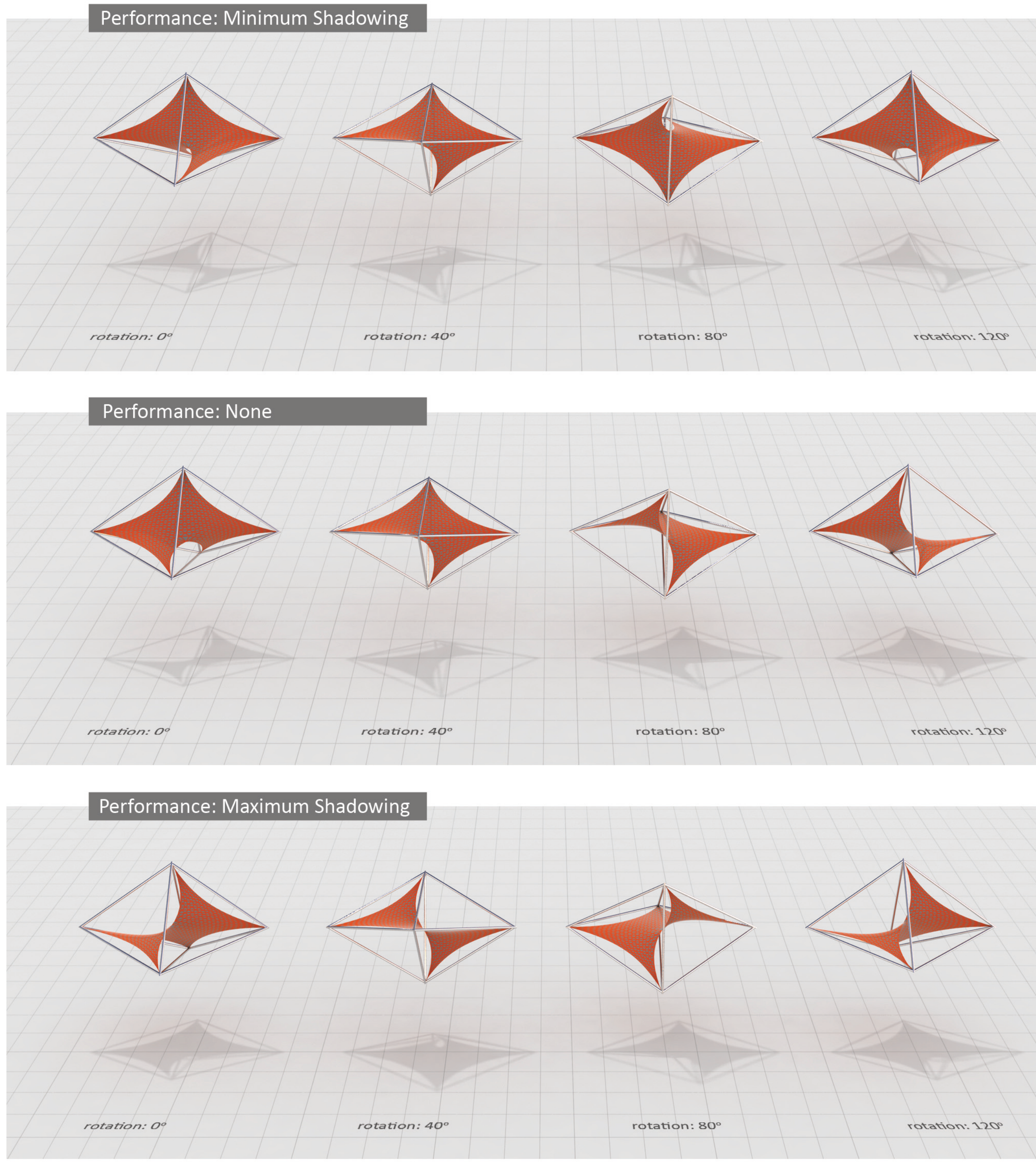
Earlier in the process the two components were compared to each other in order to select the one holding the biggest potential. At that time it was noticed that the membrane spanned within the bi-pyramid had a potential of adjusting to local requirements by repositioning itself within the frame. To be able to integrate this potential into the computational framework it is necessary to identify both the inputs that are to inform the actual placement of the membrane, and how these inputs should be translated to generate a desired performance.

Investigations into the possible effects of the repositioning of a membrane within a frame revealed that differentiated effects regarding shadowing were obtainable. Testing the performance of the membrane within a series of stepwise rotated frames of equal size it was possible to trace the degree of shading for each of the possible membrane configurations. As controlling the degree of shading for each component would enhance the performance of the membrane-frame system it was important to figure out how a shading system could be set up for the membrane.

To simplify the task of computing the degree of shading the membrane was considered as being defined by two of the six sides comprising the frame and that these sides were fully 'covered' by the membrane. From this abstraction, or simplification, the task would be to calculate the difference in degrees between the normal vector of each of the frame's sides and the angle of the rays emitted by the sun. The smaller the difference, the closer the frame side is at being perpendicular to the sun, and the larger the area will be for creating shade. Selecting the two sides with either the smallest or largest difference in degree will result in the component shading as much or as little as possible, respectively. The only rule that needs to be enforced in this decision-making process is that the two sides chosen can only share one joint, as this will ensure that the desired membrane geometry can be strung between the two sides.

These explorations of the membranes environmental performance reveals that, by utilizing the membranes possibility to reposition itself within the frame, enables the membrane system to perform environmental modulations with regards to differentiated shadow patterns.

As both these processes can be described in a number of logical steps they can both be converted to an algorithm and implemented within the computational framework. It is of course important to remember that these two processes, as well as the ones concerned with frame size and beam ratio, are in need of inputs to direct the value of their resulting output. In order to manipulate and guide the membrane-frame system a method needs to be created that has the ability to inform these evaluating processes. In other words, how would the material system behave when exposed to different contextual parameters and how would the designer/architect inform the system of their presence?



Above: The component differentiated abillity to create shadow according to the membranes orientation towards the sun.

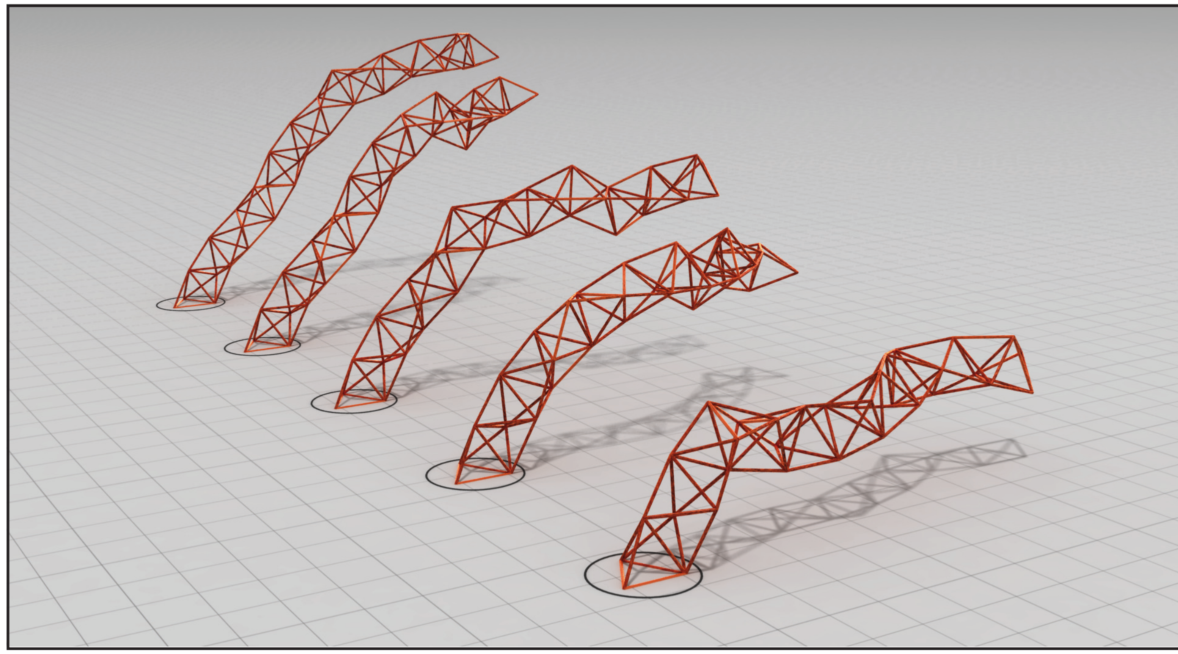
Guiding the Material System

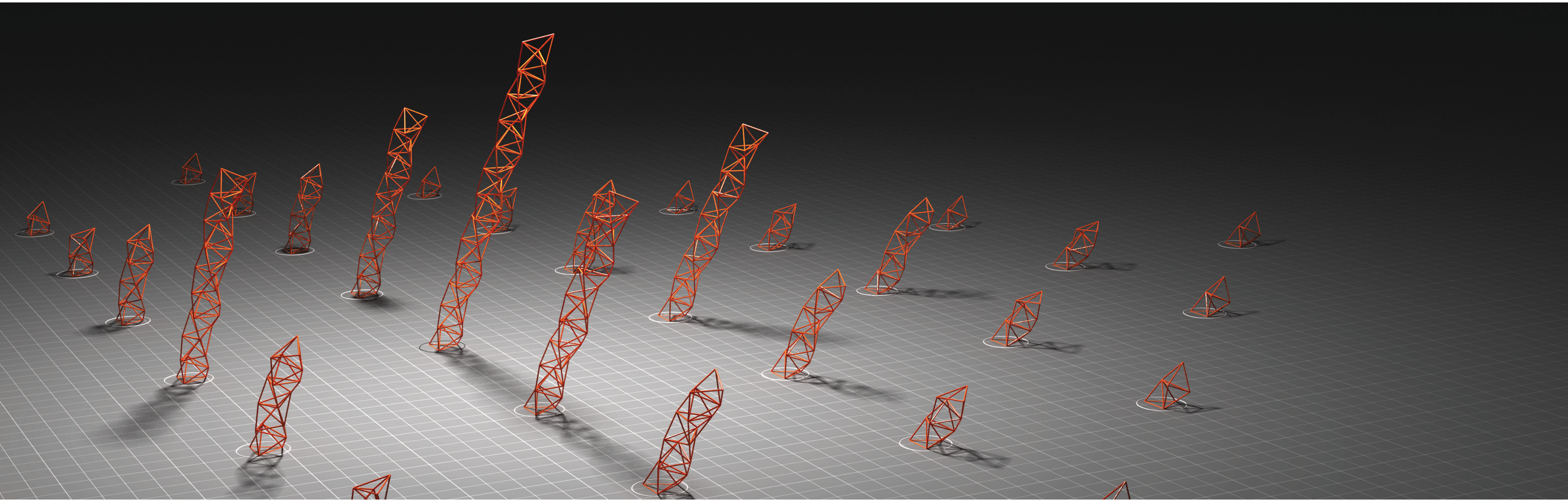
Having obtained a computational framework with restrictions and behaviors extracted from various experiments it is now necessary to define a method for guiding and informing the system. From those behaviors detected in earlier explorations it has been uncovered which parametric manipulations that yields certain desired effects. Recapitulating on the potentials discovered so far, there are a number of different parameters that are in need of an input to determine or guide their behavior.

First of all there is the aspect concerning 'growth direction'. As the material system has the potential of creating a population of components by attaching a new frame on the existing frame a recurring evaluation needs to be made so as to decide on which of the three possible sides the next frame should be attached. Secondly, by altering the parameter controlling the length of the beams the component also has a potential of varying in size, a variation that is restricted by a maximum and minimum dimension of desired or available beams. In addition, the minimum and maximum dimensions of the membrane will also restrict the size of the component as the frame cannot be larger than the membrane can span. And finally there's the orientation of the membrane. Being able to configure itself in six different ways within the frame the membrane has the potential of modulating the nearby environment through differentiated patterns of shading.

When dealing with the 'growth' aspect of the material system the initial parameter implemented into the computational framework was the direction of the sun. Informing the system with an average sun-angle it was possible to set up an algorithm that calculated the degree of deviation between the sides of the component and the sun angle. Choosing the side most perpendicular to the sun and subsequently applying the next component on this 'winning' side, it was possible for the system to grow in an almost straight spiraling line towards the sun. Looking at the behavioral effects generated by utilizing this recursive rule a new relation was introduced to the computational framework. Restricting the number of components in an assembly by the local level of sun light the system was indirectly informed that light is necessary for the process of growth. This restriction was of course implemented so as to promote the potential of the membrane generating differentiated shadow patterns.

To support the possibility of informing, or restricting, the material system about a desired, or required, height limit on a given site the computational framework was expanded with a new feature enabling an input of a maximum height. Instead of merely ending the growth-process when a certain height was reached the framework was informed in such a way that, when reaching its limit, it would change its selecting criteria so as to select instead the side most perpendicular towards the ground plane, thereby changing the direction of growth.

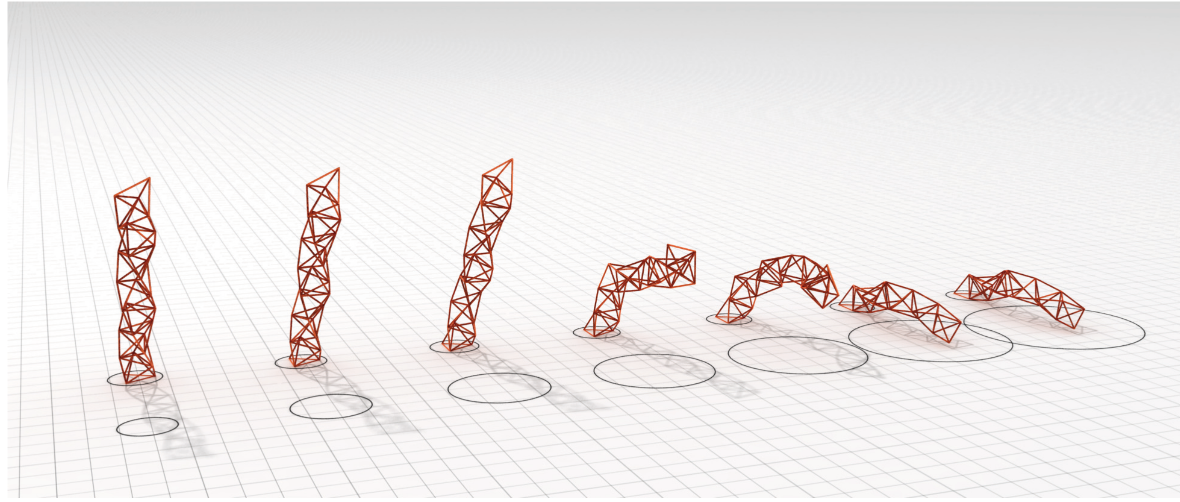


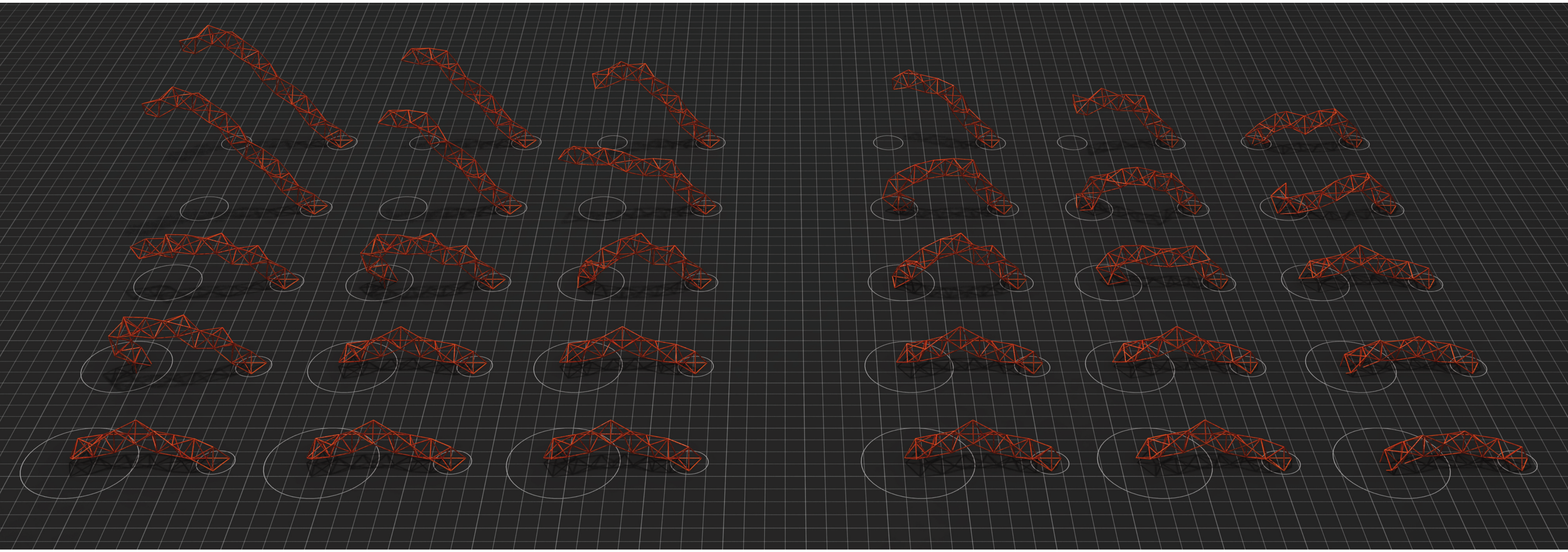


Above: Material system growing in the direction of a simulated sun. The number of components in each assembly is linked to the received amount of light.

Top: Digital test exploring the effect of varying height limits.

Dealing so far with a system 'attracted' by the presence of the sun an interesting new step is present in the introduction of an additional 'attractor'. A negotiation between more than one potential target necessitates a definition of the importance, or the force, of each constituent. In respect of this fact the extra attractor was defined by its position in space, thereby rendering it possible to incorporate the distance between this and the current position of the component as a 'distance to goal'-value, and its inherent attraction force. This expansion of the computational framework enables the designer to, based on a site analysis, introduce a range of different sized attractors and then subsequently employ the computational framework to calculate and hopefully output a series of 'successful' instances. In the experiment beneath such a population has been generated so as to explore the behavior of the frame structure when asked to take into account varying values of the parameters regarding both the direction of the sun, the height limit, and the force of the attractor.



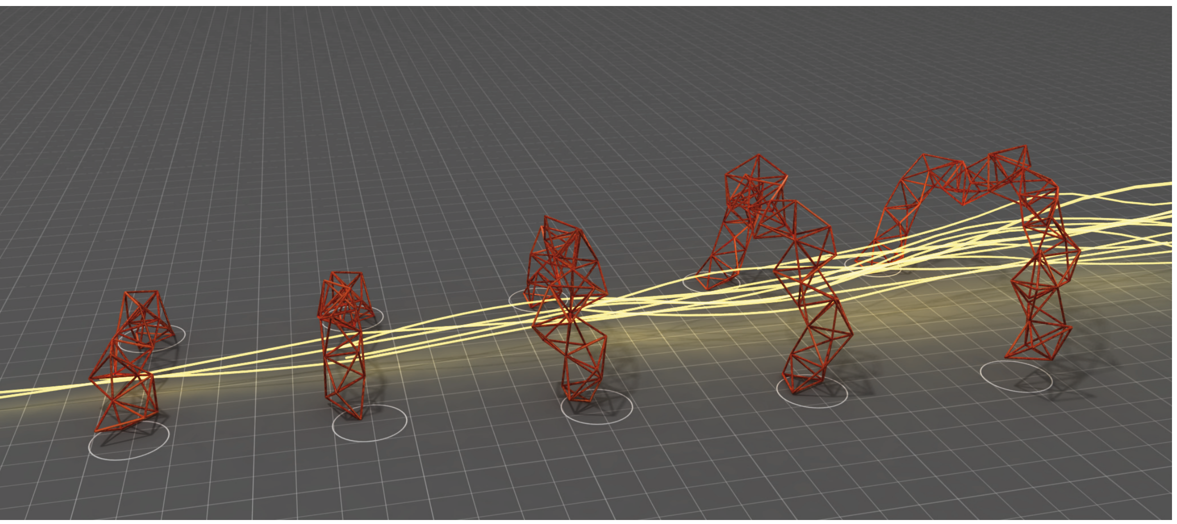


Above: An array of assemblies exposed to a variation in height limit (increasing from right to left), and varying attractor forces (increasing from top to bottom).

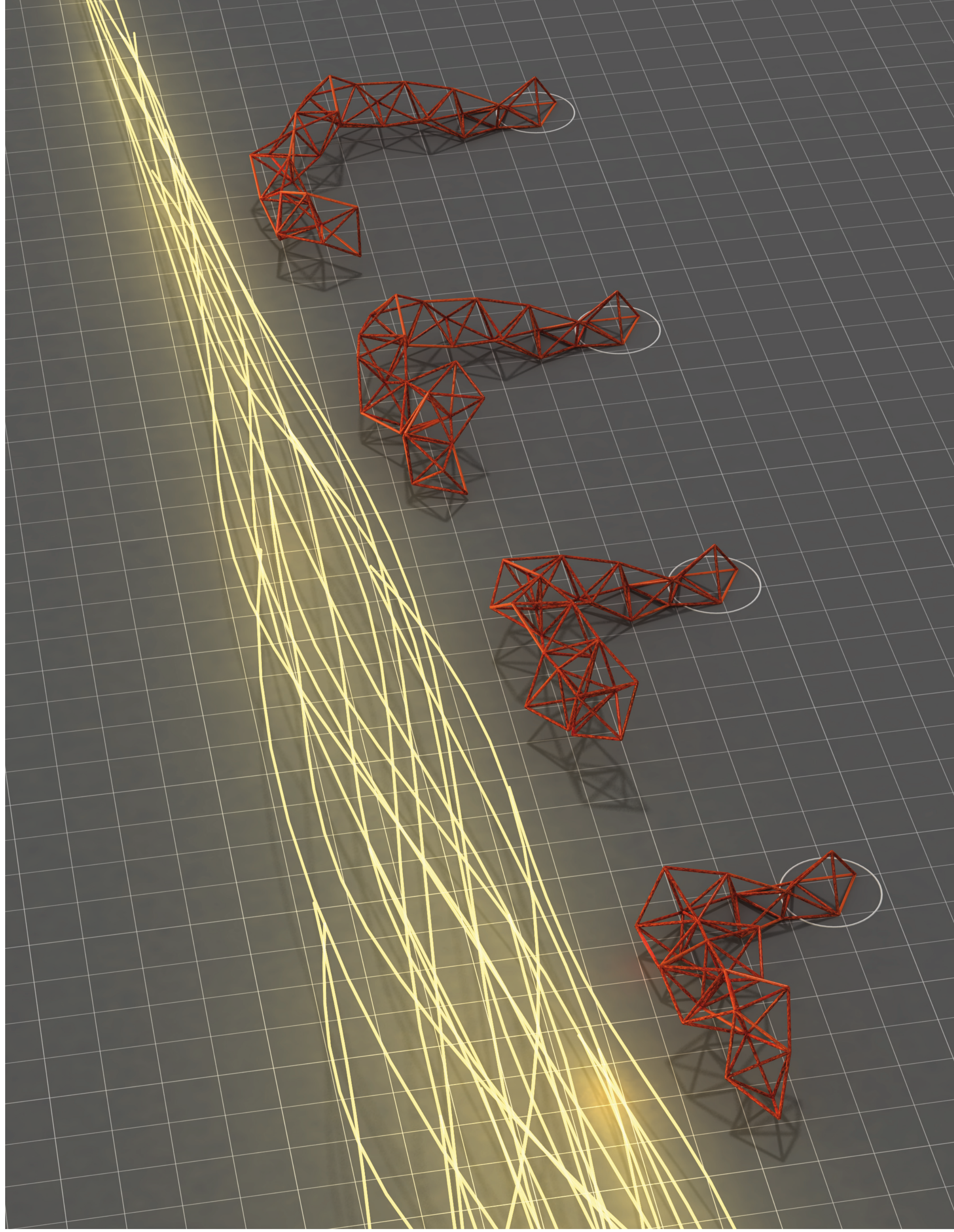
Top: Digital test exploring the effect of varying attractor forces.

Earlier in the design process the potential for implementing flow as a parameter was discussed. Unlike the attractor this parameter needs to be defined not by a point but rather with the use of a line so as to better represent its dynamic behavior. In technical terms the algorithm implementing this bases its calculations on the current position of the component and on that point on the line that yields the shortest possible distance to this. The way that the algorithm is set up it is utilizing the value derived from the flow as both an attracting and deflecting factor in the calculations. This means that when far away from the flow (large distance between flow-line and component) the component will be attracted. At a certain distance though the effect is going to shift towards a repelling kind and thereafter, if not affected by other parameters, the assembly will continue growing along the direction of the flow (a behavior visualized on the illustration at the left side).

Keeping the same flow-line while raising the height level and adding an attractor on the opposite side results in a changed behavior of the assembly. When generated with a restricted height limit the assembly grows towards and thereafter along the flow-line, but now it has the opportunity of growing up and over it, thereby reaching its new goal. With this ability it's possible for the designer to set up or inform the system about desired and undesired areas for growth.



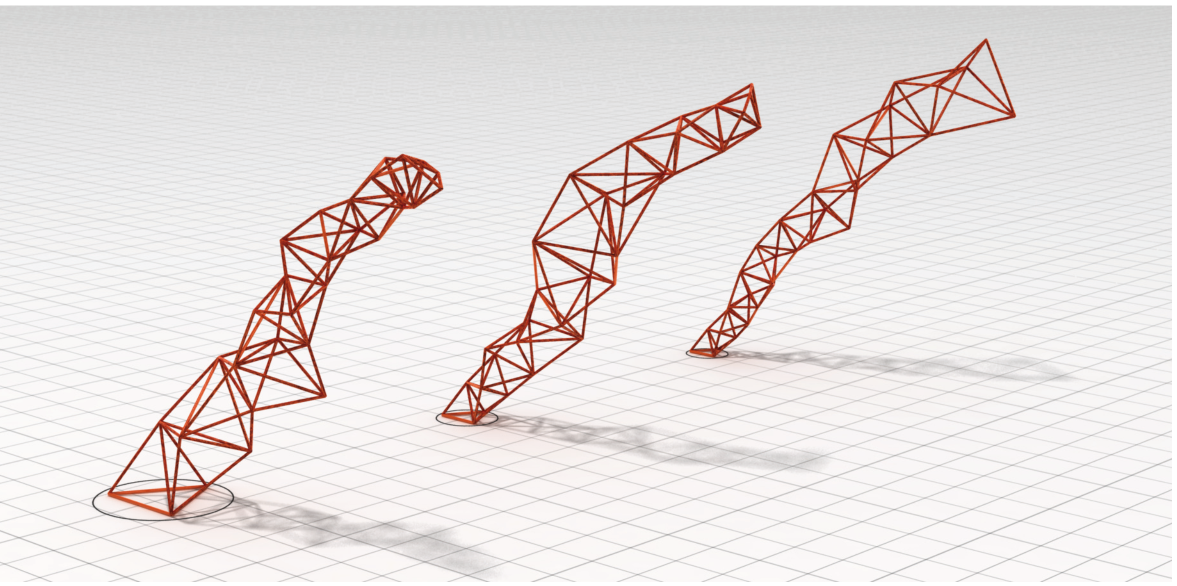
Above: When the material system isn't restricted by a height limit the assemblies will grow up and over the increasing flow of people.



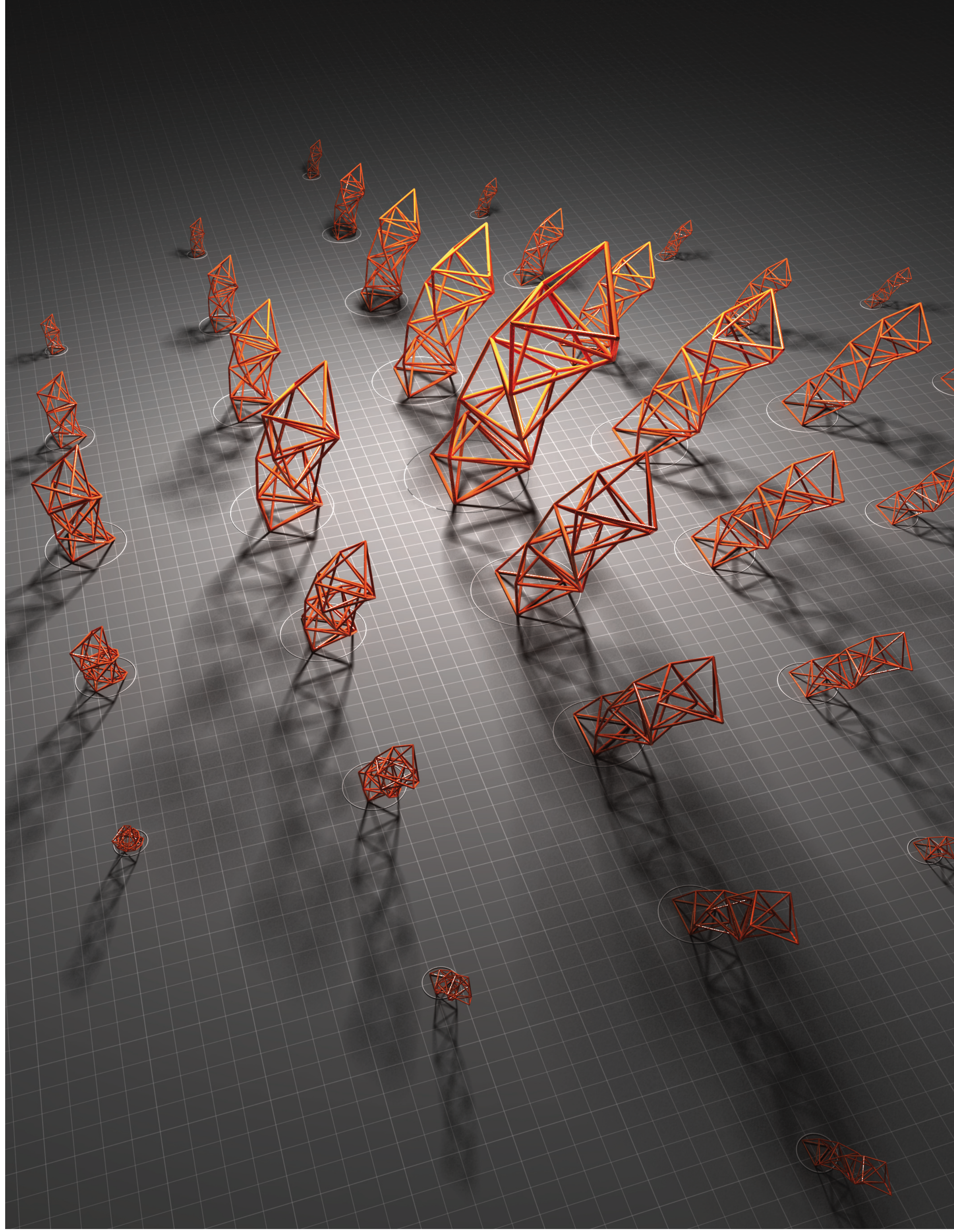
Above: However, when the material system is restricted by a height limit the assemblies will be unable of growing upwards and are therefore forced to grow alongside the flow of people.

Through a step by step implementation of extracted parameters and restrictions a computational framework has been obtained. Constructed in a parametric and hierarchical manner the framework is capable of withstanding manipulations of its inherent relationships, also when it regards a manipulation of its scale. This means that when changing the length of the beams their dimension changes as well.

To make it possible for the designer to guide the behaviour of the material system when dealing with scale an additional function was written that utilized the placement of a control point to determine the effect. When manipulating the scale of the material system two approaches can be put into use: controlling the size of the overall assembly, or the size of the individual component. As illustrated this derives to different effects. When applying the first method all beams within the assembly retains the same dimensions resulting in a homogeneous expression. Contrary, adopting the second procedure results in a heterogeneous assembly, but to not end up with each beam being unique, or to at least have the choice, this method additionally contained a feature allowing for an input of the desired number of beams and their respective dimensions.



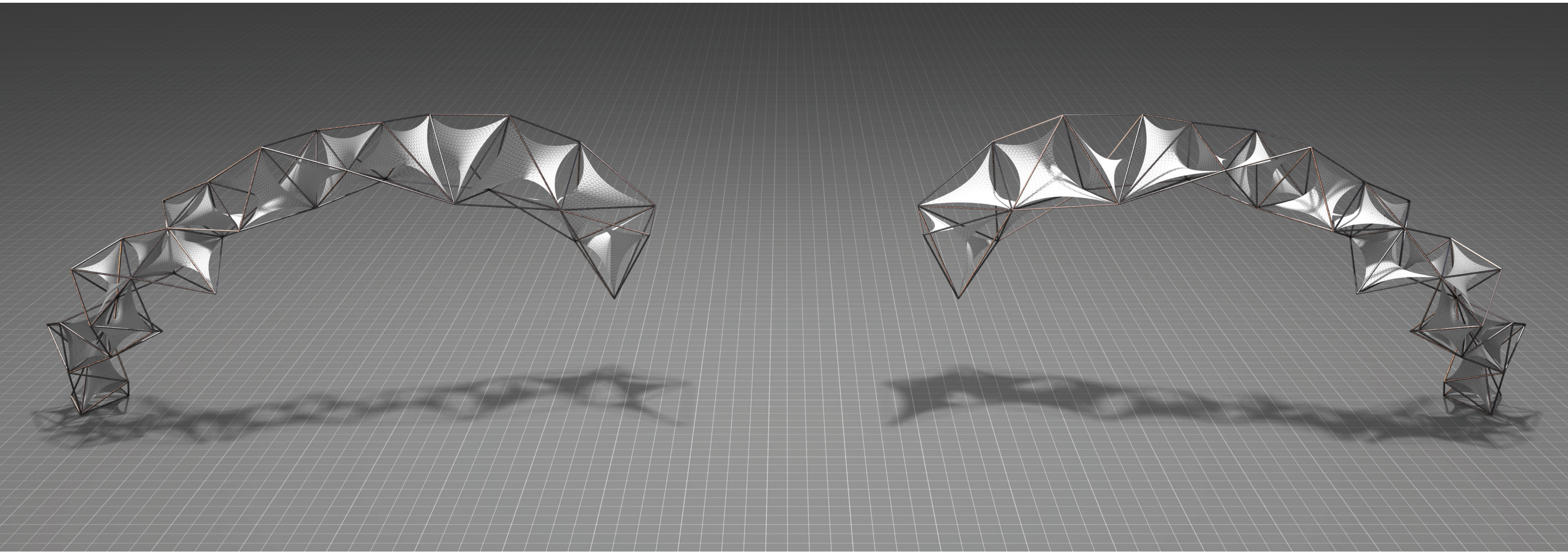
Above: Illustration showing three assemblies being exposed to varying component sizes.



Above: An array of assemblies showing the effect of applying varying scales to the same frame topology.

As showed an elaborated earlier in the design process the membrane patch contains the potential of repositioning itself within the frame, thereby facilitating differentiated environmental modulation. This means that no matter how the frames are positioned within the assembly the membrane will always be capable of adjusting for a minimum and a maximum creation of shadow. Contrary to large membrane roofs the material system is capable of generating several differentiated membrane patches within the same structure.

Through this showcasing of the material system's performative abilities it is obvious that further explorations based on the material system approach could yield numerous emerging morphologies with a high level of integration of form generation, materialization and construction. Due to the research-oriented approach of this project the material system presented remain in a proto-architectural state still awaiting a specific architectural implementation.



Above: Two identical frame constructions showing the maximum and minimum effect achieved by a reconfiguration of the membranes' placements within their respective frames.

Conclusion

The work showcased in this report demonstrates that utilizing the logic of computation in a way that integrates, both material characteristics, and manufacturing constraints, can enable an unfolding of morphological complexity and performative capacity, without differentiating between the generation of form and the following materialization. A material system approach that has the potential of promoting an understanding of form, material, and structure, as elements of a complex system capable of generating architectural constructs that modulates specified gradient conditions across a given space

Extracting those behaviours observed through experiments and explorations of physical and digital models it was possible to construct and inform the parametrically-based computational framework, thereby ensuring that later outputs were based coherent with the already revealed behaviors and established restrictions. From this it can be concluded that utilizing a computational framework renders it possible to construct a material system capable of taking into account several parameters simultaneously, thereby achieving a potential of dealing with complex scenarios where several different factors can be negotiated concurrently.

Although some aspects of the material system were left untouched, due to the limitations of the project, it was still possible to setup and explore the effect of utilizing a material approach throughout a design approach. Yet, it is evident that in order to setup a system capable of generating useable architectural constructions it is necessary to incorporate both assembly logics and structural analysis. Doing so will also enable the material system to generate outputs with an even higher complexity as it would then be negotiating between even more restrictions.

Having succeeded in the construction of a material system containing both a membrane system and a frame system it has been possible to obtain a setup that enables a structural independency towards a prospective context. The result of this system intrinsic feature is a more universally defined computational framework with a low contextual dependency and a high degree of contextual adaptability. Dealing with two hierarchically related systems also made it possible to generate outcomes capable of accommodating varying scales, with the consequential potential of being utilized in sites of diverse extents, and additionally facilitating the generation of differentiated local behaviours through various degrees of shadowing effects.

Reflection

Summing up on the design process, it is clear that the implementation of a material system approach has great impact on the way in which to go about a project. Simplified to some extent this approach is about selecting a material and in a Louis Khan manner asking it: “what are you good at?” and then subsequently guiding the generation of form on the basis of the potentials uncovered through recurring experimentation and evaluation cycles. As this approach shift towards design as program-defining it is well suited for research-minded projects, but would pose some problems if working at an architecture studio with a specific site and task at hand as it would be difficult not to let these factors affect the selection of parameters thereby coloring the final design.

Intention

Since the introduction of the very first CAD programs architects has been afraid, or at least worried, about losing control of their designs. Divided into two camps there are, to put it bluntly, the ones that completely rejects the computer as a design generator, and their counterparts that almost worship their use. As this project has adopted the computer as an important tool in the design process it is important to realize that: “…*design is strongly dependent on the tools utilized and, reversely, tools have a profound effect in design*.”[1] Contrary to traditional tools there is with the computer a possibility of having a tool that is not entirely under control and which therefore might spawn unexpected and surprising result. Embracing such unexpected behavior one might find that it could lead to results better than the ones intended. Adopting such an approach: “…*challenges one of design’s most existential qualities, that of intention. Is intention necessary in design? Is intention a human privilege only?*”[2]

Intention, a determination to act in a certain way, is often associated with having a deliberate objective. In his book ‘*Algorithmic Architecture*’ Kostas Terzidis explains that: “… *design is traditionally considered an act of conscious decision-making with an intention in mind*”[3] and immediately after states that the problem with this is that: “… *it assumes that behind every decision a conscious mind must be present*.”[4] Following this line of thought one then needs to separate the act of decision making from being only performed by a human mind to also include non-human decision-makers. Doing so would enable that decision may be made by an algorithm and if assessed to be ‘successful’ by the designer they can be adopted as one’s own idea.

Reflection on the design process it is evident that the utilization of non-human decision making has been incorporated in certain parts of this thesis project. By constructing a computational framework it has been possible to set up an algorithm that made decision regarding for instance the direction of growth. These decisions was subsequently assessed and based on the ‘successful’ results new restrictions were added to the framework. Based on such a decision-making process one can derive that the notion of intention is not associated with an outcome but rather with the process itself. This emphasis on the importance of the process instead of the goal can also be found in Bruce Mau’s ‘*An Incomplete Manifesto for Growth*’ where his third out of forty-three points states that:
“*When the outcome drives the process we will only ever go to where we’ve already been. If process drives outcome we may not know where we’re going, but we will know we want to be there.*”[5]

Genetic Algorithms

During the design process explorations into the material system disclosed a potential for the implementation of an inherent growth rule, or ‘genetic’ code, that could inform the component on how to manufacture a new component based on its own properties. Reflecting on this way of utilizing algorithms it can be stated that the process has lead to a use of a ‘genetic algorithm’.

Used within computer simulations of evolutionary processes this kind of search algorithm is a well established part of the computer programs used within the study of biological dynamics. In design the use of the genetic algorithm has the potential of allowing the designer to breed new forms rather than specifically designing them. Although the term breeding gives the impression that new forms can be derived through a simple routine there is more too in than just pressing a button on a keyboard.[6]

“*Only if virtual evolution can be used to explore a space rich enough so that all the possibilities cannot be considered in advance by the designer, only if what results shocks or at least surprises, can genetic algorithms be considered useful visualization tools.*”[7]

In his essay, ‘*Deleuze and the Use of the Genetic Algorithm in Architecture*’, Manual DeLanda argues that a productive use of genetic algorithms implies a deployment of two forms of philosophical thinking: populational, and intensive. Regarding the aspect of populational thinking DeLanda states that: “…*an entire population of such buildings needs to be unleashed within the computer, not just a couple of them. The architect must add to the CAD sequence of operations points at which spontaneous mutations may occur…*”[8]. As the material system in its last state had the potential and the necessary parametric setup to be ‘unleashed’ there’s a great potential for provoking mutations and breeding new polymorphic outputs. Doing so, would make it possible to extract new behaviors thereby furthering the complexity of the computational framework.

According to DeLanda the term ‘intensive’ refers to quantities like temperature, pressure and speed. In the same essay he discuss what intensive thinking means for the architect: “…*unless one brings into a CAD model the intensive elements of structural engineering, basically, distribution of stress, a virtual building will not evolve as a building.*” and he continues: “*The only way of making sure that structural elements do not lose their function, and hence the overall building does not lose viability as a stable structure, is to somehow represent the distribution of stresses…*”.[9] Linking back to the design process, more specifically the chapter 'Structural Considerations', it was stated that an implementation of a structural analysis method would make it possible for the material system to facilitate adaptive growth. This limitation of the material system confines it from making sure that structural elements, in this case the beams, do not lose their function. Without this ability it's not possible to ensure that the assembly does not lose viability as a stable structure. More technically, when proliferating the material system it has not been possible to simulate the distribution of stresses within the assembly thereby losing the possibility of ensuring that elements made for carrying loads aren't asked to transfer tension.

The components will in this case be structurally affected by the addition, and physical placement within the assembly, of each new component, thereby making the structural calculations far more complex and necessitate the introduction of a finite element method. Utilizing such a method within a material system would facilitate the possibility for adaptive growth, meaning that the system would be able to re-evaluate and re-configure its structural performance after the addition of each new component. Additionally containing constraints and logics regarding manufacturing, assembly, and material performance, it would be possible for the system to control or limit its own growth while keeping within the restrictions of the material system.

Although the implementation of a structural analysis has been excluded from the scope of this project it is evident that the step of bringing together a CAD package and a structural engineering package is holding great potential for further explorations of new morphologies. It will be possible to investigate the behavior of simulated evolution and trace emergent potentials that arise from the interaction between many components and parameters.

But what effect will such an approach have on the role of the architect? What will happen to that personal artistic style that almost all architects seeks to obtain? When using genetic algorithms or virtual evolution as a design tool DeLanda argues that: "*… the fact that the only role left for a human is to be the judge of aesthetic fitness in every generation (that is, to let die buildings that do not look esthetically promising and let mate those that do) may be disappointing*."[10] The artist role will then be to guide the evolution of these generated forms. Will the architect then (merely) play the role of aesthetic or functional judge? A race-horse breeder?

Notes

[1] **Kostas Terzidis, *Algorithmic Architecture*, p. 25**

[2] **Kostas Terzidis, *Algorithmic Architecture*, p. 25**

[3] **Kostas Terzidis, *Algorithmic Architecture*, p. 25**

[4] **Kostas Terzidis, *Algorithmic Architecture*, p. 25**

[5] ***http://www.brucemaudesign.com/manifesto.html***

[6] **Manuel DeLanda, *Deleuze and the Use of the Genetic Algorithm in Architecture*, presented at Between Bladerunner and Mickey Mouse: New Architecture in Los Angeles, Madrid, Spain, 2001**

[7] **Manuel DeLanda, *Deleuze and the Use of the Genetic Algorithm in Architecture*, presented at Between Bladerunner and Mickey Mouse: New Architecture in Los Angeles, Madrid, Spain, 2001, p.1**

[8] **Manuel DeLanda, *Deleuze and the Use of the Genetic Algorithm in Architecture*, presented at Between Bladerunner and Mickey Mouse: New Architecture in Los Angeles, Madrid, Spain, 2001, p.2**

[9] **Manuel DeLanda, *Deleuze and the Use of the Genetic Algorithm in Architecture*, presented at Between Bladerunner and Mickey Mouse: New Architecture in Los Angeles, Madrid, Spain, 2001, p.2**

[10] **Manuel DeLanda, *Deleuze and the Use of the Genetic Algorithm in Architecture*, presented at Between Bladerunner and Mickey Mouse: New Architecture in Los Angele, Madrid, Spain, 2001, p.2**

Litterature

Achim Menges, ***Computational Morphogenesis***, ASCAADConference on Em'body'ing Virtual Architecture, Alexandria, Egypt, 2007

Albert Ferre and more, ***Verb Conditioning***, Actar, Barcelona, 2005

Ali Rahim, and Hina Jamelle, (eds), ***Elegance***, 77(1), ADWiley Academy, London, 2007

Andres Ballentyne, ***Deleuze&Guattari for Architects***, Routledge, New York, 2007

Benjamin Aranda and Chris Lash, ***Tooling, Princeton Architectural Press***, New York, 2006

Greg Lynn, ***Animate Form***, Princeton Architectural Press, New York, 1999

Karl Chu, ***Metaphysics of Genetic Architecture and Computation***, in Mike Silver (ed.), AD, Programming cultures, Vol 76, no.4, Wiley Academy, London, 2006

Kas Oosterhuis, & Lukas Feireiss, (eds.), ***Game, Set and Match II***, Episode publishers, Rotterdam, 2006

Klaus-Michael Koch, (ed), ***Membrane Structures***, PrestelVerlag, Munich, 2004

Kostas Terzidis, ***Algorithmic Architecture***, Elsevier Ltd., Burlington, 2006

Kostas Terzidis, ***Expressive Form***, SponPress, London, 2003

Lucy Bullivant, ***4dsocial: Interactive Design Invironments***, 77(4), ADWileyAcademy, London, 2007

Manuel De Landa, Manuel, ***A Thousand Years of Nonlinear History***, Swerve Editions, New York, 2000

Manuel DeLanda, ***Deleuze and the Use of the Genetic Algorithm in Architecture***, presented at Between Bladerunner and Mickey Mouse: New Architecture in Los Angeles Exhibition, Madrid, Spain

Manuel Gausal and more (eds), ***Verb Connection***, Actar, Barcelona, 2004

Malcolm McCullough, ***20 Years of Scripted Space***, in ADProgramming Cultures, London, Wiley Academy, vol.76, no.4, p. 12-15, 2006

Mark Garcia (eds), ***Architextiles***, 76(6), ADWileyAcademy, London, 2006

Michael Hensel, Achim Menges and Michael Weinstock (eds), ***Emergence: Morphogenetic Design Strategies***, 74(3), ADWiley Academy, London, 2004

Michael Hensel and Achim Menges (eds.), ***Morpho-Ecologies***, Dexter Graphics, London, 2006

Michael Hensel, Achim Menges, and Michael Weinstock (eds), ***Techniques and Technologies in Morphogenetic Design***, 76(2), ADWiley Academy, London, 2006

Michael Hensel and Achim Menges (eds), ***Versatility and Vicissitude***, 78(2), AD Wiley Academy, London, 2008

Robert Aish, ***"Exploring the analogy that parametric design is a game",*** Game Set and Match II, Episode publishers, Rotterdam, 2006.

Yu-Tung Liu, (ed.), ***Demonstrating Digital Architecture***, Birkhäuser, Switzerland, 2005

Yu Tung Liu, (ed.), ***Diversifying Digital Architecture***, Birkhäuser, Switzerland, 2004