Generative Algorithmic Techniques for Architectural Design

Larsen, Niels Martin

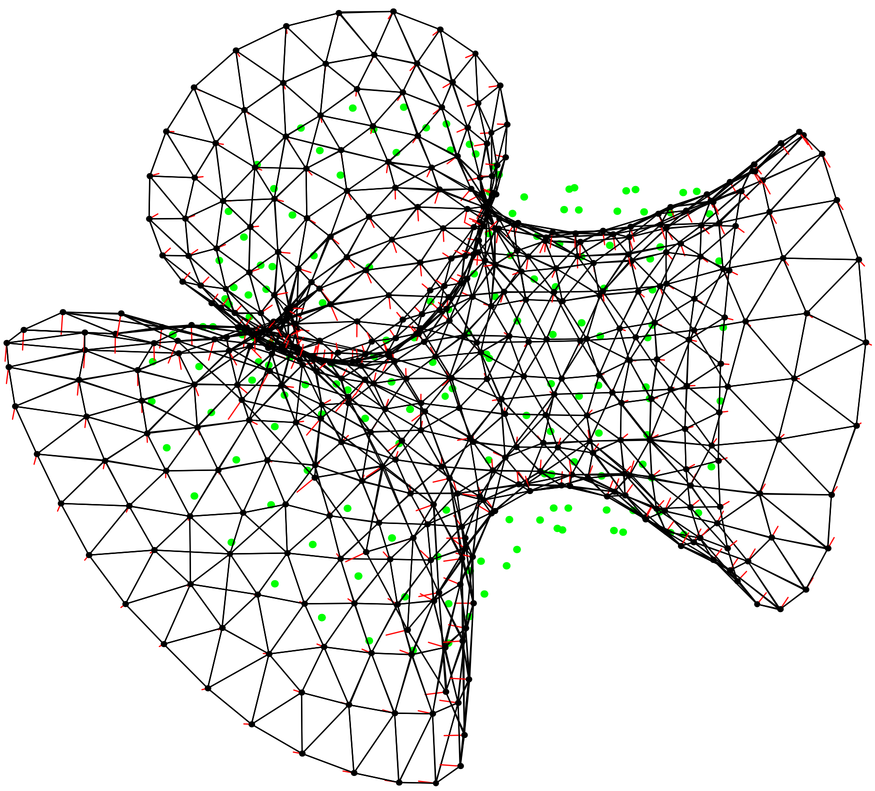
Publication date:
2012

Document Version:
Publisher's PDF, also known as Version of record

GENERATIVE ALGORITHMIC TECHNIQUES FOR ARCHITECTURAL DESIGN

Niels Martin Larsen

Thesis submitted for the degree of Doctor of Philosophy
Aarhus School of Architecture - 2012



Generative Algorithmic Techniques for Architectural Design

Thesis submitted for the degree of Doctor of Philosophy
by Niels Martin Larsen

Aarhus School of Architecture, July 2012

Principal supervisor: Claus Peder Pedersen
Secondary supervisors: Rob Stuart-Smith & Roland Snooks

Print: Sun Tryk, Aarhus.

Abstract

Dansk

Arkitektonisk designmetodologi udvides gennem muligheden for at fremstille særlige teknikker, baseret på computerprogrammering, som en integreret del af designprocessen. Den stigende udbredelse af digitale produktionsteknikker inden for byggeriet giver nye muligheder for at etablere smidige flow mellem digital formgenerering og realisering. En tendens i nyere praksis viser et øget fokus på at udvikle unikke tektoniske løsninger som en vigtig ingrediens i designløsningen. Disse tendenser konvergerer, og udgør den bredere kontekst for denne afhandling.

I arkitektonisk design er digitale værktøjer hovedsageligt blevet brugt til repræsentation og specifikationer, og generelt som en effektiv udskiftning af tilsvarende manuelle redskaber. Den seneste udvikling gør det muligt for arkitekter at integrere computerberegning i designprocessen, og derved generere geometri og information i forhold til realisering. I denne situation rettes arkitektens opmærksomhed mod de formgenererende regler, som så afspejler designhensigter, kontekstuelle parametre og produktionsforhold. En fordel ved denne fremgangsmåde er muligheden for at håndtere større grad af kompleksitet, både med hensyn til geometri og ydeevne. Det skyldes til dels, at den underliggende matematiske logik danner basis for at generere information på mange niveauer, afhængigt af projektets udviklingstrin og modtagerbehov.

Forskningprojektet retter sig imod behovet for at etablere en arkitektonisk ramme for at inddrage og diskutere generative teknikker. En række metoder, rettet mod bestemte typer af geometriske og arkitektoniske problemer er udviklet, og danner grundlag for at diskutere muligheder og konsekvenser relateret til hver problemtype. Metoderne er ofte inspireret af modeller udviklet inden for naturvidenskab, især biologi. Principperne er således videreudviklet med henblik på artikulation indenfor arkitektonisk design. Visse metoder er bidrag, som viser potentiale for fremtidig brug og udvikling. En metode er således rettet mod ”bottom-up” generering af overfladetopologi ved anvendelse af en agentbaseret logik. Der er udviklet en måde at integrere mønsterdannelse og kontekstuelle parametre i den formgenererende proces. I et tredje eksempel etableres et flow af information og materiale fra formgenerering til realisering ved hjælp af generative værktøjer og feedback loop i designprocessen. Der gennemgås yderligere tre metoder, og via specifikke referenceeksempler etableres en overordnet diskussion på tværs af de algoritmiske og arkitektoniske principper. Her diskuteres potentialer og implikationer for arkitektonisk formgivning. Det gælder både på det metodiske niveau, hvor forskellige tilgange sammenlignes og i et bredere perspektiv i forhold til arkitektonisk designmetodologi.

Abstract

English

Architectural design methodology is expanded through the ability to create bespoke computational methods as integrated parts of the design process. The rapid proliferation of digital production techniques within building industry provides new means for establishing seamless flows between digital form-generation and the realisation process. A tendency in recent practice shows an increased focus on developing unique tectonic solutions as a crucial ingredient in the design solution. These converging trajectories form the contextual basis for this thesis.

In architectural design, digital tools have predominantly been used for representation and specification, and as an efficient replacement of equivalent manual tools. Recent developments allow architects to integrate computation in the design process. Thus, computation is used for generating geometry and information concerning realisation. In this situation, the architect's attention is directed towards the form-generating rules, which then reflect design intents, contextual parameters and production constraints. An advantage of this approach is the capability of managing substantial complexity in terms of geometry and performance. This is partly due to the fact that underlying mathematical logic allows information to be generated on many levels with respect to development stages and the needs of the receiver.

A necessity for establishing an architectural framework for adopting and discussing generative techniques is addressed in this research. As such, a series of methods directed towards specific types of geometric and architectural problems is developed, and form the basis for discussing the potentials and implications related to each problem type. These methods are often inspired by models developed within natural sciences, biology in particular. The principles are further developed to form new modes of articulation in architectural design. Certain methods are contributions, which suggest a potential for future use and development. Thus, a method is directed towards bottom-up generation of surface topology through the use of an agent-based logic. Another method embeds a negotiation between pattern-formation and contextual parameters in the form-generating process. A third example demonstrates a flow of information and matter from form-generation to realisation through the use of generative tools and feedback loops in the design development. Three additional methods are described, and reference examples help to establish an overall discussion across the algorithmic and architectonic principles. Here, potentials and implications for architectural design are discussed. Both concerning a method oriented level, where different approaches are compared, and a general perspective with respect to architectural design methodology.

Acknowledgements

I would like to thank The Danish Council for Independent Research | Humanities for supporting the project financially. I would also like to thank all of the people who have supported me during this project. First of all, my principal supervisor, Claus Peder Pedersen, who has been deeply engaged in the project in many ways. His engagement in developing the initial application for gaining financial support was determining, and his research experience has been invaluable in terms of developing the theoretical discourse. Also, many thanks to my secondary supervisors, Rob Stuart-Smith and Roland Snooks, for introducing me to many of the principles that form the basis for this research. Their advice and support has been crucial for the advancement of the project.

I would like to thank Sebastian Gmelin for always taking time to discuss the research, and for bringing knowledge into our common teaching activities, which again have informed this research. Asbjørn Søndergaard for his support, and particularly for taking part in the first case study experiment. Ole Egholm Pedersen and Dave Pigram deserve many thanks for inviting me to collaborate in joint research, based on tools and techniques from their previous research. I would like to thank Morten Bülow for his collaboration, particularly with respect to some of the visualisations in the thesis, and Caroline De Dear for proofreading within a limited, yet changing, time frame. Karl Christiansen has been very supportive with respect to perspectives for future research and collaboration.

I Especially want to thank my wife, Birthe, for her incredible patience during this long journey. Without her amazing support, this project would never have been possible. Also, thanks to our wonderful daughters, Marie and Kamille, who had to bear with an absent father for long periods of time.

Contents

1 Introduction

1.1 Research topic and terminology

New ways of integrating computation in the design process suggest a possibility of rethinking the architectural design process. The possibility of playing out a whole range of iterative negotiations, in the process of establishing a tectonic system, allows a completely different hierarchy of design decisions to emerge. This effectively changes the meaning of authorship in relation to architectural design. These tools, referred to as *generative techniques* in this thesis, increase the field of complexity manageable in an architectural design process, and allow a notion of time, through iterative calculation, to become part of it. The inclusion of factors, normally considered in the phases close to realisation of the project, allow aspects concerning materials, manufacturing and tectonics to affect the form generating process. By equipping the architect with these tools for generating and controlling complex information on all levels concerning the actual realisation of the project, the architect is potentially brought closer to the centre of the project development. Furthermore, a range of scientific algorithms can be brought into the architectural realm of tectonic solutions, thereby expanding the architectural vocabulary. This research project investigates the potential of generative techniques in architectural design, and is connected to computational developments, architectural design and manufacturing. It studies how designers can begin to use computation as part of the design process for generating form and information, rather than mere representation. A renewed understanding of tectonics grounds digital computation and a history of extracting useful knowledge and techniques from science and architectural design are reinforced in the discussion. This research is induced by the increase in use of advanced digitally controlled production techniques in the industry, which allows information to flow seamless between systems controlled by the designer and systems controlled by the manufacturer. I am concerned with recent developments in architectural practice, where there is a tendency towards seeking characteristic tectonic solutions as a crucial part of the architectural design process. Originally, it was planned to develop a design vocabulary where the computational part was peripheral. During the process of formulating research questions, the potential of exploring new approaches to the design process emerged. Therefore, it was decided early on in the project that emphasis would be placed on the digital approach whilst maintaining a tectonic discussion in order to keep the research project within the realm of architecture.

The term *generative techniques* in relation to architecture is understood similarly to that of *generative art.* Philip Galanter has defined generative art as: 'any art practice where the artist uses a system, such as a set of natural language rules, a computer program, a machine, or other procedural invention, which is set into motion with some degree of autonomy contributing to or resulting in a completed work of art.' [1] However, the generative techniques referred to in this thesis are computational. Typically, a generative design process is initiated by defining a number of basic rules, a set of parameters, and an environment. The rules are played out in a sequence of operations. Often an interaction between units that represent the design components produces a series of results. One or more of these results can then directly become part of the design, or the starting point for a new process with a different set of rules and techniques. In both cases, the designer can change the rules, the parameters and the environment in order to regulate the design. On behalf of the designer, the focus shifts from the final design, towards the fundamental ingredients that affect the form generating process. As indicated, this approach to the design process could lead to a slightly different comprehension of the designer's authority, since part of the decision-making is moved away from the designer over to the generative system. This is true in the case where the system itself is not part of the design process. However, it is argued here that the design of the generative system, or at least a modification of it, should be based on the specific design intent and project context, and therefore also a part of the design process. Neil Leach expresses this understanding of the architect's role: 'the architect is recast as the controller of processes, who oversees the 'formation' of architecture.' [2]

This formation sometimes occurs through a process that can be described as *self-organisation*, and in some cases the system shows *emergence*. As marked by an issue of *Architectural Design* in 2004, the terms have gradually entered architectural discourse, inspired by science in the field of biology, physical chemistry and mathematics.[3] Within these different fields of science, the definitions of self-organisation and emergence vary. Therefore, it is shortly explained how the terms are used in this thesis. [4] *Self-organisation* is understood as a self-regulating mechanism, where individual parts of the system interact through negotiation, thereby displaying ordered behaviour. This corresponds with the notion suggested by Tom De Wolf and Tom Holvoet: 'Self-organisation is a dynamical and adaptive process where systems acquire and maintain structure themselves, without external control.' In biology, *self-organisation* and *emergence* are often interconnected, basically, referring to the same thing. For instance, Scott Camazine and others define *self-organisation* as 'a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of the system. Moreover, the rules specifying interactions among the system's components are executed using only local information, without reference to the global pattern.' [5] In this thesis, *self-organisation* is related to the ordering principles between the interacting elements of the system, and not necessarily an emerging pattern that displays order on a level that is different from the level of the interacting parts.

There seems to be larger consensus on the definition of *emergence*. Camazine and others refer to *emergence* as 'a process by which a system of interacting subunits acquires qualitatively new properties that cannot be understood as the simple addition of the simple addition of their individual contributions.' [6] This corresponds largely with the use of *emergence* in this thesis. Here, *emergence* is used to denote a situation where order on a global level is generated through local interactions between elements on a lower level. Level and order depend on the viewpoint and criteria defined by the observer. In short, this thesis articulates a distinction between *self-organisation* and *emergence*, where the former refers to an ordering principle between the elements, and the latter refers to appearance of order on a 'higher' level. A form of hierarchy is then related to the emergent property. These subjects are further discussed in relation to the developed methods, particularly with respect to agent-based systems.

One of the topics in the thesis is how generative techniques are related to *tectonics*. Within this thesis, *tectonics* refers to the way buildings are constructed, and the control of the underlying logic and the methods necessary for forming, organising and producing the architectural elements. An emphasis on generative techniques can be seen as a continuation of the theoretical discourse on tectonics, rather than being opposed to it. The term *digital tectonics*

1 Philip Galanter, 'What is generative art? Complexity theory as a context for art theory', *Proceedings of the 6th international conference: Generative Art 2003,* Milan, 2003

2 Neil Leach, David Turnbull & Chris Williams, *Digital Tectonics*, Wiley-Academy, West-Sussex, 2004

3 Michal Hensel, Achim Menges and Michael Weinstock, *Emergence: Morphogenetic Design Strategies*, Wiley-Academy, London, 2004, page 7.

4 Tom De Wolf and Tom Holvoet, 'Emergence versus self-organisation: Different concepts but promising when combined', *Engineering SelfOrganising Systems, Volume: 3464, Issue: 1675*, Springer, 2005, pages 1–15

5 Bonabeau, Camazine, Deneubourg, Franks, Sneyd, Theraulaz, *Self-Organisation in Biological Systems,* Princeton University Press, USA, 2001, page 8.

6 Bonabeau et. al., *op.cit.,* page 31.

has become part of the architectural vocabulary.[7] Neil Leach and others state in the introduction to the publication *Digital tectonics* that '… the old opposition between the digital and the tectonic has begun to collapse …' What is emphasised is the fact that digital tools have infiltrated every aspect of architectural production, and that they allow a new kind of structural integrity. This dissolution of the dichotomy between tectonics and the digital is part of the discussion in this thesis. Another term that appears frequently in this thesis is *pattern,* and often, *spatial pattern.* Here, the term *pattern* is not confined to the general use of the word as a repetitive decoration, but relates to its scientific significance. It refers to the existence of an underlying logic or system that is recognisable and therefore, in principle, possible to describe through a set of rules. This does not necessarily mean that we are able to define those rules, since they may be of immense complexity. The reason for using this reference is to engage with some of the recognisable patterns discovered in natural science, in order to investigate their potential in relation to architecture. This research project is focused towards the mechanisms that generate these patterns rather than the emerging patterns themselves. Hence, spatial patterns that can be generated through the use of known algorithms are seen as candidates for becoming part of the design process. This project seeks to extract computational methods, developed in science for the simulation of natural phenomena, and use these methods to generate patterns that can become useful in relation to architectural design through simplification, alteration, expansion and combination with other methods or external parameters. As a consequence, the methods are often deprived of their original function of correct simulation. Instead of reflecting patterns found in nature, the developed methods can then be used for generating new types of geometric organisation with tectonic potential. Furthermore, the altered methods establish new types of negotiations in the form generating process: a subject that appears frequently in this thesis.

1.2 Research question and methodology

What is the potential of using generative techniques in architectural design? Whilst this question can be seen as the overall focus of my research, it is necessary to specify the perspective from which the research experiments has been carried out. Digital tools have become intrinsic for developing and realising contemporary architecture. Digital tools provide means for visually presenting proposals and for managing information concerning manufacturing and construction.

7 I.K.Andersson & P.H.Kirkegaard, *A discussion of the term digital tectonics*, WIT Transactions on The Built Environment, Vol 90, 2006 WIT Press.

Different layers of information are increasingly being integrated through the use of building information tools, reflected in the related research.[8] Besides presentation and managing information, some architects seek the use of digital tools for generating new types of formal expression, based on the fact that Euclidian geometry seems to dissolve in the abstract world of advanced modelling software, such as those developed for production of animation movies. This last issue appears to be less simple than such in terms of realisation, and most examples of so called 'parametric designs' are to be considered as either explicitly formed with a top-down approach, or generated from linear methods of form-generating techniques. Other types of projects, typically not realised or even fully explained as architecture, succeed in demonstrating whole new ways of establishing material organisation through bottom-up approaches. The research described in this thesis connects some of these approaches. It departs from the formal approach towards the development of underlying organisational patterns. It breaks away from linear design methodology and arrives at a more bottom-up oriented method for architectural design. Whilst the initial emphasis has been on developing these new types of internal logic, a secondary goal has been to connect these methods with different kinds of tectonic systems (or at least geometric principles) and performance oriented simulation techniques. The intention has been to investigate how the intertwining of complex algorithmic principles and tectonic logic can lead to a more rational and 'intelligent' type of expressive architecture. An ambition is to establish a vocabulary of algorithmic techniques from the viewpoint of architectural design. Several recent publications communicate knowledge related to the field of computational design. In some cases they are oriented towards digital design in a broad perspective, such as Antoine Picon's *Digital Culture in Architecture*.[9] More often they are collections of writings and projects that forms a collage of relevant topics and examples. Here could be mentioned *Architecture in The Digital Age: Design and manufacturing*[10], edited by Branko Kolarevic and Kevin Klinger, which shows series of projects where digital tools have played various roles in terms of design development and realisation. Other publications are preoccupied with the generative logic. An example of this is *Programming Architecture*[11] by Paul Coates, which

8 Jason Underwood & Umit Isikdag (Editors). *Handbook of Research on Building Information Modeling and Construction Informatics: Concepts and Technologies*, IGI-Global, 2010

9 Antoine Picon, *Digital Culture in Architecture*, Birkhäuser, Basel, 2010

10 Branko Kolarevic, ed. Architecture in the Digital Age: Design and. Manufacturing. Spon Press, 2003

11 Paul Coates, Programming Architecture, Routledge, London, 2010

is structured from an algorithmic perspective and less occupied with the geometric and spatial implications. The ambition with this thesis is to describe an array of methods and algorithms that demonstrate specific approaches to different types of geometric problems from an architectural perspective. While, this forms a palette of methods, the goal is more to unfold potential and implications that emerges from thorough investigation. The techniques are based on known algorithmic principles. However, in some cases the techniques are further developed or implemented in new ways in order to establish the described methods. In this sense, this thesis contributes with approaches to generative techniques, both on a general and a deeper level.

The primary source of knowledge is experimental development of a series of algorithmic methods for architectural design. This approach was chosen for several reasons: it was crucial for the researcher to gain a deep understanding of the techniques involved in order to be able to identify their potential, and more importantly, their implications and limitations. Without testing out some of the techniques, a critical discussion related to the field would have to rely on statements, judgements and analysis of projects. As mentioned, this type of research can provide an understanding of tendencies within the field, but is less capable of providing specific details concerning the advantages and disadvantages of the exact methods used in the projects. As a result of these experiments and related theories, a series of additional questions have emerged:

1. In which ways do the tools allow complexity in architectural design?
2. How do the techniques offer ways of embedding new types of negotiations of the form generating process?
3. What are the possibilities of using generative tools to establish new kinds of feedback loops in the project development process?
4. How do the techniques question some of the profound hierarchies that exist within architectural methodologies?

The main issue here is that use of generative techniques challenge the normal design practice in ways, which exceed the challenges that computer aided design challenged the praxis when it was first introduced. As put by Koralevic: 'The digital generative processes are opening up new territories for conceptual, formal and tectonic exploration, articulating an architectural morphology focused on the emergent and adaptive properties of form.'[12] Interestingly, Koralevic emphasises form, whereas, topology is discussed as a means for maintaining consistency: '…the primacy over form of the structures of relations…' This matter is further pursued in Chapter 3.

12 Koralevic, *op. cit.*, page 13

This research has been carried out as a form of research by design. Without entering current discussions on terminologies concerning practice-based research, the approach focussed on new methods for use in architectural design through experimentation, rather than analysis of works within the field. The hypothesis was that this approach would reveal insights concerning intrinsic potentials and implications deeper than if projects developed by second parties were discussed. Presumably, the latter approach would have allowed a broader spectrum of methods and projects to be part of the discussion, and thereby provide a more substantial foundation for discussing general developments within the field. However, the more technical approach, as chosen by the researcher, led to an understanding of the field on many unexpected, deeper levels. The method development that forms the core of this research reflects a range of dimensions and scales related to architectural design. The goal was to develop methods for use in architectural design, based on a series of established algorithmic principles. Each method demonstrates the extent of which it is based on, and differs from, established methods. Generally, new mechanisms or combinations have been implemented for each method. Method development consists of three stages:

1. Form-generating method
2. Possible use and variations with respect to architectural design
3. Physical realisation

The first stage is the development of the algorithmic principle and the geometric form-generation that relates it to architectural design. The second stage is where the method is explored in terms of variations and possible outcomes, still in a virtual environment or in the form of abstract models. In the third stage, the method is tested out in 1:1 scale, and material and structural properties are embedded, as part of the functionality of the method. The third stage marks a large increase in complexity, due to the practicalities related to actual production and construction. In a single case, the method has been fully explored in the third stage, namely, the method *Complex Gridshell*. Logically, a fourth stage could be to use the method for the design of an architectural project, perhaps even with a separate fifth stage, representing the realisation process. The increase in detail that follows with a specific design solution would also demand further development of the generative tools. These stages concerning actual design were not part of the research.

Generally, the focus of this thesis is on the organisational principles and the generative principles, where actual realisation is discussed as a field of possibility and as an essential precondition for the relevance of the methods. When the realisation was not directly part of the experiment, one or more types of specific production would allow for the use of the method in realising a construction. As such, the focus has mainly been directed towards the first stage, concerning development of form-generating methods from algorithms. With a particular type of method (agent based systems), the development process is explained in depth, in order to unfold its complex functionality, whereas, the specific use with respect to architectural design remains on an abstract level.

These methods cover a range of architectural dimensions, spanning from surface morphology to three-dimensional aggregate growth. In this way, the methods address a spectrum of different problem types, or situations, related to architectural design on different levels. These levels are not necessarily related to scale, but rather to different types of form-generating principles. One end of this range are occupied with surface morphology, and in this sense, a two dimensional approach. The other end deals with aggregate growth from an inherent three-dimensional approach. In between these poles, other methods represent gradients of three-dimensionality. Six different methods have been developed:

1. Complex Gridshell
2. Self-organising Bezier Curves
3. Branching Topologies
4. Self-organising Surface / agent-based systems
5. SAGA
6. Solar DLA system

The first four methods deal with self-organisation in different ways, and the algorithmic principles in these methods can be compared to some extent. They all have a type of particle system, where a set of rules direct the local behaviour of the particles, or agents, which then self-organise to form distinct forms or patterns. In relation to Self-organising Surface, a series of procedures for generating geometry with agent-based systems were developed. The last two categories are types of aggregate growth. Here, predefined components are distributed from either a recursive or random logic, gradually filling up the design space.

1.3 Thesis contents and structure

The thesis consists essentially of two parts. The first part, Chapters 2 to 6, is concerned with a discussion of the developed methods, and how they relate to generative techniques and architecture on a general level. Two architectural examples are presented with some detail, in order to position the discussion of particular methods in an architectural frame. In the last part, the level of technical detail increases, since these chapters are concerned with the main body of research. In this part, the methods and experiments performed within the research project are described in detail. The two parts are inextricably linked, as the methods in the latter part serve to establish the discussion that is unfolded in the first part. The structure can be represented as such:

Chapter 1:	Introduction, research topic and methods
Chapters 2-6:	General discussion of developed methods and architectural examples
Chapter 7:	Conclusion
Chapter 8:	Detailed description of developed methods

The discussion in Chapters 2–6 is organised in order to reflect a gradual transition from two dimensions to three, corresponding with the character of the methods. The first four chapters, concerned with self-organisation, represent each a separate method, and the methods concerned with aggregate growth are discussed in the same chapter. The discussion relates both directly and indirectly to architectural practice, and some key topics are introduced by referring to specific architectural projects. This is to frame the subsequent detailed descriptions in an overall architectural context, which otherwise could become less apparent in the more abstract examples. Furthermore, the discussion of methods is assisted by theoretical references, which mainly address the underlying algorithmic principle. By placing the theoretical references directly in connection with the experiments, the necessity of external sources is reduced. A different approach would have been to gather all theoretical background as an introductory section, but this may have made the, afore mentioned, objectives difficult to achieve.

Chapter 8 contains the main body of the research, namely a detailed explanation of the functionality of the methods. This establishes a systematic way of describing, analysing and discussing the examples, thus developing an informational matrix. While the developed methods display a large degree of variety, the matrix helps to mark where there are similarities and differences. It raises and locates topics that can be discussed across different methods. The matrix consists horizontally of three categories: The first category, *intents and conditions,* contains a description of the idea behind the experiment, what the conditions have been for developing it, and what types of questions have been proposed. The second, *organisational logic*, is focused on the underlying logic and the geometrical principles that are part of the methods used.

The final category, *realisation*, discusses how the experiments have been realised as part of the research, and how they relate to the implementation in architecture. In short, the three parts describe the conditions, techniques and results of the experiment. By emphasizing the design intents and realisation aspects, the matrix supports an architectural approach, rather than focusing on the algorithmic categories, also reflected in the thesis structure. However, the method development has been focused on the organisational logic in most cases. An exception exists in Chapter 8.1, which describes a method that has been developed and tested through 1:1 case studies. Here, the realisation technique is closely linked with the generative logic. The matrix is horizontally divided into two parts: *properties* and *observations*. This division enables a clear distinction between the experiment and the discussion of it. The division between experiment and observation has much importance in the field of natural science, where it is necessary to produce ample proof for results. However, in the present research, the motivation is not to give the impression of the research as being objective in any sense, nor is it to give any type of proof for the results. Rather, the idea is to establish a key for gaining an understanding of the individual experiments together with a principle that can help to look across them. Within the method chapters, each vertical category is described in separate subchapters in terms of their properties. The observations of each category are described in a single subchapter, both for simplicity and to be able to discuss across the categories. It should be mentioned that the method described in Chapter 8.4, *agent based formations,* escapes the structure of the diagram, since the chapter is organised as a progressive development of the organisational principle rather than as a study of a single method. Beneath is a diagram of the matrix of categories:

	Intents and conditions	Organisational logic	Realisation
Properties	Keywords	Keywords	Keywords
Observations	Keywords	Keywords	Keywords

1.4 Larger thesis context

The background for this project is a convergence of trajectories. One trajectory is the development of the use of computers in the design process. Until recently, the use of computers in architectural design has been (almost entirely) for the purpose of representation and organisation of information related to the projects. Generally, computer-based tools have been used as a more effective way of completing work that had previously been carried out manually. Naturally, the possibility of using digital models has allowed designs that demand more complicated solutions to be realised. What remains the same, however, is the creation of designs through sketches and the creation of models. Simulation tools have become increasingly important, particularly for the analysis of technical performance on the engineering side of the project development. Simulation methods have been implemented in animation software developed for the film industry, in order to be able to generate realistic visualisations of dynamic systems, such as waves, smoke and explosions. This type of software was developed during the 1990s, and was adopted by several architects, such as Greg Lynn, who advocated engaging with 'topology, time and parameters' in the design process. This radical change demonstrated that computation could be used to generate information as part of the design process, not just as an informational representation. Lynn sees complex geometries, such as splines and NURBS surfaces, as capable of embedding notions of time in form, and demonstrates how animation tools provide an iterative form-generating process where negotiations of different parameters lead to the result. He states: 'Instead of a neutral abstract space for design, the context for design becomes an active abstract space that directs form within a current of forces that can be stored as information in the shape of the form.'[13] Lynn claims that due to the limitations of having to specify architectural construction through simple algebra, architecture is most often conceived as primarily dealing with gravitational forces, leading to an exaggerated emphasis on verticality. [14] He recognises that the complex iterative calculations necessary for controlling dynamic systems, are too complex for designers in general. This is where the animation software is considered a useful tool. In principle, Lynn's work reflects a turning point for architecture, mainly within academia, where dynamic systems and advanced algorithms have become part of the designer's vocabulary. This notion was particularly promoted by Columbia University, where Lynn taught during the 1990s.

13 Greg Lynn, *Animate Form*, Princeton Architectural Press, New York, 1999, page 11.

14 Lynn, *op. cit*. page 16.

Subsequently, this particular type of animation software has become widespread within architectural design. Important nuances exist, however, which differentiate these methods from the generative techniques dealt with in this thesis. Firstly, Lynn accepts the constraints that follow from using predefined software. This results in the difficulty of allowing different types of information to become part of the form generating process. Another aspect that dominates Lynn's approach is a resistance to engage with tectonics as part of the initial design process. The search for smoothness fits well with the abstract formations produced in the animation software, but also means that a traditional process of translating the formations into a construction must be applied in the case of realisation.

The present research seeks to demonstrate how other types of possibilities emerge from being able to tailor the generative system to the task, and also how geometric information that is useful in relation to realisation of the construction can be part of the initial process. In this sense, the research bridges the gap, as suggested by Lynn, between a tectonic approach and an approach engaging with complexity. Still, many of Lynn's analysis of the potential and implications of the tools are relevant to consider. For instance, he notices how testing out the character of the algorithms was crucial for gaining an intuitive understanding: 'In order to bring these technologies into a discipline that is defined as the site of translation from the virtual into the concrete, it is necessary that we first interrogate their abstract structure.[15] Lynn's methods have paved the way for a radically different way of using computational tools in the design process. However, where Lynn in his early work was confined to manipulating the way existing tools generated and exchanged information, many recent designers have been attracted to creating their own form-generating tools as part of the design process. Lynn also takes on this new challenge, and his view on the field of computational design has gradually changed. In 2006, he states in an interview that 'Architecture has a disciplinary history and responsibility to express parts-to-whole relationships and hierarchy [and] to ignore the history and richness of assembly is to miss the real impact of calculus.' In this sense, Lynn recognises the potential of addressing the form-generating mechanisms as such, rather than just accepting predefined algorithms, but still emphasises the importance of the designer's responsibility for establishing a form of unity within the design result.

A second trajectory is the proliferation of computer-aided manufacturing (CAM). The industries have during the last decades increasingly adopted computer-based production techniques, bringing the number of robots in industry to more than a million in

15 Lynn, *op.cit*. page 40

2010.[16] The development is generally driven by search for efficiency and optimisation, but what is so far not generally realised by architects, this situation allows entirely new forms of manufacturing processes to occur, linking the information generated as part of the design process directly to the technologies that are used in the industry to produce the actual building components as part of the realisation of the project. Furthermore, if this mode of exchanging information is brought to its full potential, architects will begin to combine manufacturing information with specialised tools that are part of the design process. On a political level, this may in some cases reposition the architect in the centre of the project development.[17] In regards to this research project, the concern is primarily towards the possibility of engaging with new types of complexity in the design process. The developments in manufacturing is a crucial part of what makes generative tools relevant for architects, as otherwise, the generative designs would in many cases end up as purely theoretical. An interesting issue is that the details of actual production have the potential to affect, and become part of, the design process. This is not dissimilar to the way architects of the past had knowledge about the crafts involved in construction.

A third trajectory is directly grounded in observations of developments in recent international architectural practice. This can be seen as a critical reaction to the cold, uniform appearance of late modernist architecture that continues to prevail throughout most of the world. One particular tendency is a renewed interest for developing and using characteristic tectonic solutions within architectural expression. In some cases, the main focus for developing the project has been towards a tectonic solution that can serve as both a technical solution for a specific challenge and at the same time as a pattern that covers the building, thereby providing it with a unique character. In some cases, the main goal is to find a tectonic solution with maximum performance. In other cases, the decorative result is the main driver for the development process. Often, these projects gain entirely new types of expression, which again helps to transcend the general tendency towards placelessness. This thesis does not provide an extensive analysis of this tendency. Rather, this concept is used as a theme that supplements the discussion about relating generative techniques with architectural practice. In this sense, the projects relate to Stan Allen's writing on *field conditions*. Allen states:

16 IFR Statistical Department, '2011 Executive Summary', *World Robotics - Industrial Robots 2011*, viewed 2 February 2012, <http://www.worldrobotics.org/uploads/media/2011_Executive_Summary.pdf>

17 Edwin Chan, Gehry Partners: 'It is absolutely possible that within a very short time … the technology could be developed in such a way that you just make the whole building directly from the database.' in *Digital Project, Frank Gehry's Vision,* video, Dansk Arkitektur Center, Copenhagen, 2007.

'Overall shape and extent are highly fluid and less important than the internal relationships of parts, which determine the behaviour of the field.'[18] This expresses that generative techniques can support this pattern-oriented approach to architectural design, since the tools exactly can help to generate and control complex patterns and large amounts of information, which is often necessary when realising complex tectonic solutions.

1.5 Digital tectonics

The following information connects the research described in this thesis with a broader historical discussion of the term *tectonics.* This is initially demonstrated through a brief interpretation of Gottfried Semper's writings on the subject, and notes on his interest in science. Then follows a reference to Kenneth Frampton's revitalisation of the discussion on tectonics, as a means to counteract the increasing tendency towards formalism. Finally, a new understanding of tectonics is delineated, including a discussion of computational tools and digital production techniques with reference to the writings of Neil Leach and others.

The term *tectonics* derives from Greek *tekton*, which means carpenter or builder. In ancient Greece it gained a broader meaning, referring to the act of making or production. The term later lead to the word *architekton*, which means 'master builder.' The term tectonic emerged in the 19th century in relation to a renewed interest in the ancient handicrafts and construction techniques. In Gottfried Semper's *Four Elements of Architecture,* tectonics specifies framework/roof as one of four basic elements, where the other three are: the earthwork/mound, the hearth and the enclosure/ lightweight membrane. These elements are related to different skills, where carpentry has to do with tectonics, ceramics and metal works are linked with the hearth, masonry is connected with the mound and weaving is related to the enclose, or lightweight membrane. An intimate relation between material and craft exists in the sense that technical skills evolve by the gradual manipulation of materials. Semper is preoccupied with material translations, which he denotes *stoffwechsel*.[19] Textiles, and how they are translated to wall fitters, are of particular interest. He constructs a theory concerning the cultural and historical evolution of architecture, and seeks to explain the change in expression, by relating it to varying conditions, needs and changes in materials and technical developments. His definition of basic architectural elements has proved a useful model in terms of understanding architectural works, which shows in the more recent writings on the tectonic. In this, the writings were part of a discourse in the 19th century concerning the role of architecture in relation to industrialisation. Semper criticises how former handicraft objects and architectural motifs were increasingly industrially produced without regards to how change in material and technique may affect the symbolic meaning.[20]

In 'Style in the Technical and Tectonic Arts' Semper expands his original theory, even adopting natural phenomena, due to historical scientific revelations at the time. During this period, geologists established that the age of the world was millions rather than thousands of years old. This discovery, among others, served to diminish the role of the biblical Creation in both artistic and scientific theories.[21] Semper's theories emphasised the role of techniques in architectural creation, where later critics such as Alois Riegl stated that style was entirely on behalf of the artist or architect's vision, or *Kunstwollen*. Semper advocates for seeing artistic creation as a process, or *the becoming of art*. His emphasis is: 'to explore the inherent order that becomes apparent in phenomena of art during the process of becoming and to deduce universal principles from what is found, the essentials of an empirical theory of art.' Again, he is focusing on the technical means as a factor that, together with other driving forces, is an important part of artistic form making, compared to a more abstract approach that only considers compositional aspects, such as proportion and symmetry.[22] However, due to Semper's interest in natural science, he does in fact analyse exactly these more abstract phenomena with respect to form making. In this sense, the theories become more formally based, compared to the ideas generally presented in his writings, which otherwise were rooted in historical and cultural analysis. As such, this more formal approach is perhaps not the strongest part of Semper's theories, and probably not the most well known. However, they suggest that architectural theory has a history for engaging with ideas and methods developed in science, which is also why they are worth mentioning here. Symmetry, proportionality, and direction are explained through references to natural phenomena and are

18 Stan Allen, 'From object to field', *Architectural Design Vol 67. May-June,* 1997

19 HF Mallgrave, 'Introduction', in HF Mallgrave & W Herrmann (eds, trans.), *The four elements of architecture and other writings,* Cambridge University Press, New York, 1989, page 36.

20 Kenneth Frampton, *Studies in Tectonic Culture*, The MIT Press, Cambridge, Massachusetts, 1995, page 87

21 Op.cit. Mallgrave, page 19.

22 Gottfried Semper, 'Style in the technical and tectonic arts or practical aesthetics', in HF Mallgrave & W Herrmann (eds, trans.), *The four elements of architecture and other writings,* Cambridge University Press, New York, 1989, page 183.

described as the primary conditions of formal beauty.[23] The primary examples are snowflakes and plant growth that show a number of variations of the three qualities, where the term *eurythmy* denotes a particularly subtle form of symmetry related to an experience of rhythm. Semper emphasises the role of *direction*, as reflecting the vital forces in nature. The forces are directed opposite the conditional forces, such as gravity. This again is explained further as resulting in different types of hierarchies, or *authorities,* visible in nature. Semper uses these references to link art and architecture with natural phenomena. Although it is questionable whether Semper succeeded in qualifying the relevance of his own theories in this way, it is interesting to notice how he seeks to bring these early theories of form-generation in nature into the field of artistic creation. Evolution theory was an unavoidable part of scientific discussion during Semper's historical context. Here, I will note that some of Semper's qualitative conditions for beauty, particularly the ones of direction and authority, relate to parameters playing roles in the experiments that also appear in this thesis.

The term 'tectonic' was re-introduced as part of the architectural discourse by Kenneth Frampton, when he published 'Studies in Tectonic Culture' in 1995. The book is primarily based on the analysis of works and writings of major architects in the 20th century. It was received as an important critique of the tendencies of the postmodern style towards abstract formalism, where fundamental aspects of architecture were neglected. The attention is drawn towards type, site and tectonic, which are elements that are able to 'counter the present tendency for architecture to derive its legitimacy from some other discourse'. Frampton problematises how architects import ideas from other fields, such as figurative art or philosophy, rather than basing the design on the fundamental elements that must inevitably be part of architecture. While type, site and tectonic are considered as equally important, the book focuses on the tectonic as a 'poetics of construction.' Directly in correspondence with Semper, Frampton raises the question of symbolic versus technical, or representational versus ontological. According to Semper's theory, particular architectural elements are related to symbolic values, namely the hearth and the infill wall, where Frampton points out that these principles must be rearticulated, depending on the particular conditions. By referring to Eduard Sekler, Frampton notes that a tectonic quality is achieved when structure and construction appear to be mutually interdependent.[24] Then, based on quotations of Vittorio Gregotti, Frampton states that 'The full tectonic potential of any building stems from its capacity to articulate both the poetic

23 Semper, *op.cit.*, page 198

24 Frampton, *op. cit.*, page 20

and the cognitive aspects of its substance.' He further states, 'thus the tectonic stands in opposition to the current tendency to deprecate detailing in favour of the overall image. As a value it finds itself in opposition to the gratuitously figurative…' These statements suggest an approach to architectural design where tectonic patterns are considered as primary architectural motifs.

By analysing a number of projects in detail and comparing them with related theories, Frampton provides a precise understanding of the role of the tectonic in architectural production. The necessity of the architect's knowledge of craft and production can be seen through the example of Mies Van Der Rohe, in terms of his chosen materials and the acknowledgement of their properties. For instance, he would visit brickyards and inspect the firing in order to control the dimensions and colours of the bricks that were often imported from Holland. Frampton unfolds a series of analyses concerning the tectonic aspects, and many others, of Mies' work. It is remarkable that during early projects, where he was able to manifest a new understanding of spatial continuity, Mies was very much aware of the tectonic implications. One of the conclusions from Frampton's series of thorough analyses points to: 'the crossroad at which the profession stands, for the fact that either architects will maintain their control over the metier of building design…or the profession as we know it will cease to exist.' [25] He suggests that the architect must learn to control the technological complexity of the project in order to 'reinstate their authority and to overcome, as it were, the redundancy of the somewhat circular working drawing shop drawing procedure as it presently exists.' By referring to Renzo Piano and the Centre Pompidou, computer technology is brought forward as a promising tool for managing the constraints and tolerances of the procedures involved in construction. This final remark is interesting in the sense that Frampton's book has been taken into account for a critical approach for the use of computers in architectural design.[26]

Bringing the discussion of the tectonics, and more precisely *digital tectonics* into the present context, it is relevant to refer to the book, edited by Neil Leach, David Turnbull and Chris Williams with exactly that title.[27] As they point out in the introduction to the anthology, 'a new tectonics of the digital – a digital tectonics – has begun to emerge.' They explain that computer technologies have become an integrated part of architectural production and that new understandings of material and structural properties have become

25 Frampton, *op. cit.*, page 386

26 Antoine Picon, 'Architecture and the Virtual: Towards a New Materiality,' *Praxis 6*, 114-121.

27 Neil Leach, David Turnbull & Chris Williams, *Digital Tectonics*, Wiley-Academy, West-Sussex, 2004

available through the use of computation. Digital tools currently exist as a bridging platform that enables architects and engineers to exchange ideas and information during the development of an architectural solution. These new types of collaboration between architects and engineers mark a paradigm shift. An example of this is reflected in this thesis in the method example in Chapter 8.1, where a relevant collaboration was established on a small scale. An important consequence of recognising the digital as intrinsic to a present understanding of tectonics is that *controlling* the geometry becomes a crucial aspect. An important question is then, at what point in the design process does the connection between the digital and the tectonic become established? Generative techniques allow this connection to be made early on in the project development, establishing intrinsic relations, both with respect to tectonic and contextual parameters. This aspect forms an important basis for the methods, described in this thesis.

Perhaps it is possible to detect some similarities and differences between the three different situations that the previous theorists respond to. Semper can be seen as reacting to the lack of consistency between the way architecture was articulated and the way it was constructed in his own time. In this sense he was a precursor for some of the radical developments in architectural expression that occurred in the early 20th century. The strongest of these developments are demonstrated in the works produced at the Bauhaus schools. Here, the goal was to seek new types of expression within all art forms, including industrial design and architecture, based on modern production techniques and experiments with fundamental physical phenomena, such as light and material properties. The strong link between architecture and other art forms is one of the aspects criticised by Kenneth Frampton, who sees this as a reason for typical architecture of the late modernist period losing its natural basis in structural and material properties. In this way, tectonics is seen as a way of informing architectural production with a more natural connection to its context, in the broadest sense. This approach aids in the reduction of the formalistic top-down approach, imposed through artistic image-driven design methodologies. In recent projects, based on digital tools, there is a lack of internal logic, let alone an interest in material or structural elements. These can be considered equally formalistic to the projects that Frampton is criticising. The approach, formulated by Neil Leach and others, suggests recognition of the possibilities of integrating digital tools with architectural design. Furthermore, there is a differentiation between methods where digital tools are used as merely for abstract modelling, and methods where integration of the digital is seen as an intrinsic part of the design development, which can ensure that both internal logic and tectonic aspects appear as a natural part of the design solution. In this sense, the dichotomy between analogue and digital does not apply to architectural production. Rather, a fundamental difference exists between design methodologies. One type of method is bottom-up oriented and seeks to address all aspects of the result, including tectonic aspects, early in the design process. Other types represent a more linear top-down process, where the overall form is defined first, and decisions concerning lower level relations are made closer to the time of realisation.

Generative techniques point toward a rethinking of tectonic hierarchies. In many of the examples in this thesis, a type of structural hierarchy, or a hierarchy between primary compositional volumes and secondary smaller building parts is not initially defined. Rather, a mechanism for negotiating certain relations is established. Through iterative negotiations, the form or articulation is gradually developed. The controlling mechanisms represent a hierarchy, the main difference being that it is not necessarily a formal hierarchy, but a hierarchy that balances different parameters, essential to the design solution. This leads to the proposal of a different view of tectonic hierarchies, compared to Semper's categories, and to the reinterpretation of the notion of tectonics. This notion includes the digital and acknowledges emergent hierarchies. The majority of methods described in this thesis do not display a tectonic outcome in the traditional sense, that is, in the tradition of Semper and Frampton. Rather, many of the methods relate to a discussion of digital tectonics. Thus, it can be recognised that the underlying logic of the generated geometry is essential to arrive at a consistent tectonic solution. The digital logic as a platform for exchange and generation of design information, which enables smoother collaboration between architects and engineers proves to be beneficial to structural problems. The same logic can be directed towards aesthetic or contextual parameters, thereby establishing simultaneous negotiations between parameters that are normally treated independently as part of the form-generating process. This occurs in most of the methods described in the thesis. In these cases, a type of digital logic exists, even if material or structural properties are not specifically defined. It can be discussed if the term tectonics, or more specifically *digital tectonics*, applies to this situation. However, in terms of computation, the mechanisms concerned with structural logic and the mechanisms concerned with other parameters can be entirely integrated.

1. 6 Tectonic Patterns

In certain recent architectural projects, there is a direct relation between part and whole, perhaps in a slightly different way from the type of coherence, Greg Lynn addresses in Chapter 1.4. In these

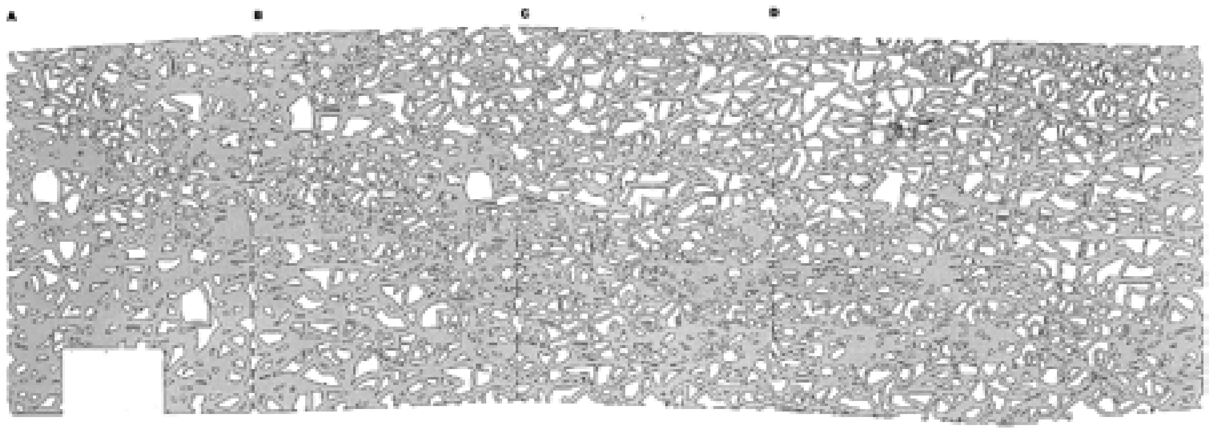
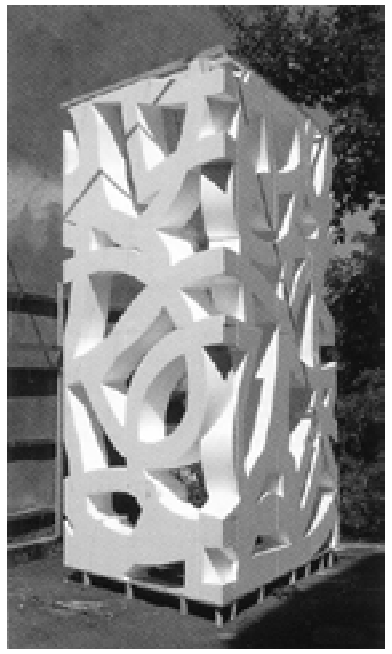


Figure 1: Herzog & De Meuron, Ciudad del Flamenco. Unfolded tower elevation.

works, major building parts are not assembled in a hierarchical composition. Alternatively, the buildings consist of small components and joints creating a façade pattern that is differentiated and functions as a netted mask that covers the building and creates both a homogenous and varied expression. The part is crucial to the whole as the buildings are shrouded in, and perhaps even defined by *tectonic patterns*. The main architectural identity of the building is created by a unique tectonic system, developed for the individual project. Where Greg Lynn is engaged in parts-to-whole relationships in terms of dynamically shaped forms, the attention here is much more directed towards pattern formation and tectonics. The tectonic pattern, covering the building volume, leads to the building volume being perceived as a coherent whole. This notion is consistent also in cases where the building does not constitute a regular shape. The tectonic pattern articulates the building, and is the prerequisite for experiencing it as a whole. The pattern can in some cases be considered as a type of material-based decoration. In other cases, the pattern becomes the façade structure itself, or becomes the primary supporting structure of the building. Here, a three dimensional pattern with structural properties emerges. It is impossible to experience the architectural whole without including the character of the individual parts that create the structure. Such an effect occurs in Herzog & De Meuron's project *Ciudad del Flamenco* in Jerez, Spain. In this case, a pattern of geometries derived from Arabic symbols is distributed over the whole façade, forming a concrete structure. Something similar is evident in a project by the same office *Campus Tree Village* for China, where patterns pervade the project on different levels of scale, from the bearing structure to the detailing of the façade. Both *Ciudad del Clamenco* and *Campus Tree Village* demonstrate tectonic patterns, and can be compared to the way Farshid Moussavi refers to the ornament.[28] In *The Function of Ornament* she states, 'ornaments are intrinsically tied to architectural

Figure 2: Herzog & De Meuron. Ciudad del Flamenco, Facade mockup.



28 Farshid Moussavi & Michael Kubo, *The Function of Ornament*, Harvard Graduate School of Design, Actar, New York, 1996

affects.' This supports the idea that the ornament is essential for the architectural identity, as suggested earlier. When specifying the nature of the ornament, Moussavi also says 'ornament can relate to depth in a number of ways. It can work with the entire form, with the load-bearing structure, or exploit the sectional depth of the cladding.' This approach considers ornamentation in a way that is similar to the discussion of tectonic patterns. As previously noted, patterns are not merely understood as decorative in relation to this research project. It is therefore less obvious to use the term *ornamentation.* It should be mentioned that Moussavi's claim is that ornamentation goes beyond the decorative. However, the emphasis here is primarily on the notion of pattern formation, and the goal for discussing tectonic patterns is to emphasise the potential for adopting methods for pattern generation into the field of architecture. Furthermore, this thesis hypothesises that generative techniques are capable of supporting a pattern-oriented approach to architectural design. Some architectural projects demonstrate how such methods can

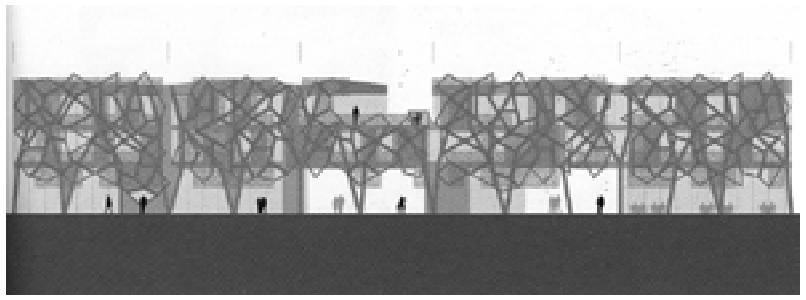


Figure 3: Herzog & De Meuron, Campus Tree Village. Unfolded courtyard elevation.

be implemented in architectural design. One of the few examples of realised architecture that is directly referred to as algorithmic is the *Serpentine Pavilion* by Cecil Balmond and Toyo Ito,[29] shown in Figure 13 in Chapter 3.3. The algorithmic principle within the structure's design is relatively simple, based on rotation and scaling of a square. Through an original tectonic solution, primarily based on steel flat, a unique tectonic pattern was created, essentially defining the architectural expression or identity. Another iconic project where generative techniques have played an important role is the *National Aquatics Centre in Beijing*, which is further described in Chapter 6.1. Here, a type of spatial pattern, inspired by the way soap bubbles self-organise to form complex structures, was used as the underlying organisational principle, effectively defining both the structure and the appearance of the building. In this case, the tectonic pattern is both understood as the spatial structure and the facade tectonics, which directly reflects the former. These types of architectural projects serve as support material for my investigation

29 Leach et. al., *op. cit.*, 129.

of generative techniques. My hypothesis is that through the use of generative techniques, it is possible to arrive at architectural solutions that have unique character through a methodology based on tectonic patterns. Algorithmic logic is not only beneficial in terms of generating advanced forms of expression and increasing performance, but also as an underlying structure that supports the realisation of the project, particularly when it is relevant to make use of digital production techniques.

2 Surface morphology

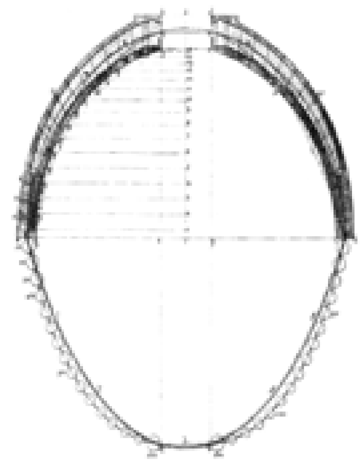
2.1 Structural self-organisation

This chapter addresses morphological self-organisation and the use of algorithmic tools for linking with digital production techniques. Additional topics are digital versus analogue self-organisation, aspects of materiality and feedback in the design process. This discussion is mainly focused around the *Complex Gridshell* method for designing and constructing a concrete gridshell of discrete components, explained in detail in Chapter 8.1. The *Mannheim Multihalle* serves as a key architectural reference. Additional references also aid the discussion. These examples are based on a surface topology. This is not an inherent property of the form-generation method, but rather a result of the design intent and general conditions of the projects. As such, these examples belong to one of two principally different approaches, which are both represented in this thesis. The contrasting approach, discussed in Chapter 6, is concerned with aggregation rather than morphology.

The most famous example of the use of analogue form finding methods is Antoni Gaudi's hanging chain models of the *Colonia Güell*. The principle was described first by Robert Hooke in 1671, while he was working with Christopher Wren on the project for *St. Pauls Cathedral*. [1] Hooke proposed that the inverted shape of a hanging chain represents the optimal form for an arch construction. While a hanging chain supports only tension, a masonry arch acts in compression. In 1748 Poleni used a chain with differentiated loads to analyse the stability of the dome of *St. Peters church* in Rome. Since the catenary could be inscribed inside the construction, Poleni concluded that the dome was safe. In the late 19th century Antoni Gaudi explored the use of catenary shapes much further. He realised that in the case of a three-dimensional vault structure, it was necessary to use three-dimensional models to generate the optimal shapes. By hanging small sandbags with string, he made large models for simulating an optimised compression structure. The sandbags were weighed so as to represent the loads on the structure. Gaudi experimented with different methods for transferring the shape of the complex models through drawings, photography and by covering them with cloth in order to extract their form.

Later in the 1950's, Frei Otto began to explore the field of analogue form finding methods, and his experiments included hanging

Figure 1: Poleni's drawing of the hanging chain principle from 1748.



1 P. Block, M. DeJong and J. A. Ochsendorf, 'As Hangs the Flexible Line: The Mechanics of Masonry Arches', *Nexus Network Journal on Architecture and Mathematics Vol. 8, No. 2*, Birkhäuser Basel,2006, pages. 9-18.

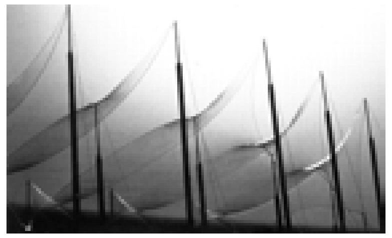


Figure 2: Top: Frei Otto. Soap film model of parallel wave tent. Above: Frei Otto and Peter Stromeyer's form-optimised tent construction for the Garden Exhibition in Hamburg, 1963. Photos: Institut für Leichte Flächen-tragwerke, University of Stuttgart.

chain models similar to Antoni Gaudi's. Frei Otto was occupied with the ability of basic materials to self-organise, creating an equilibrium state that reflects the minimum level of energy. As Manuel DeLanda states 'The processes that generate these geometrical forms may be viewed as searching a possibility space until they reach the singularity.' [2] In 1982 Frei Otto led a team that reconstructed Gaudi's model of the *Colonia Güell* Church in Barcelona (see Figure 3). Besides his experiments with funicular structures, Frei Otto explored a wide range of analogue form-generating principles, such as soap-film experiments for producing minimal surfaces.[3] These experiments formed the basis for large-scale projects with textile membranes as tectonic equivalents to the minimal surfaces, generated with soap film. As the example in Figure 2 shows, a series of optimised shapes for tent constructions were developed through the use of soap film models. As computer technology developed, Frei Otto's analogue experiments were gradually merged with digital simulation tools, particularly with respect to the realisation of actual constructions. The *Multihalle* in Mannheim is such an example that makes use of analogue models and is one of the very early large scale constructions that was realised through the use of computer-based generative systems.

2 Manuel DeLanda, 'Material Elegance', *Elegance*, Wiley, London, 2007, page 22.

3 Frei Otto, Bodo Rasch, *Finding Form: Towards an Architecture of the Minimal*, Edition Axel Menges, 1995, page 58.

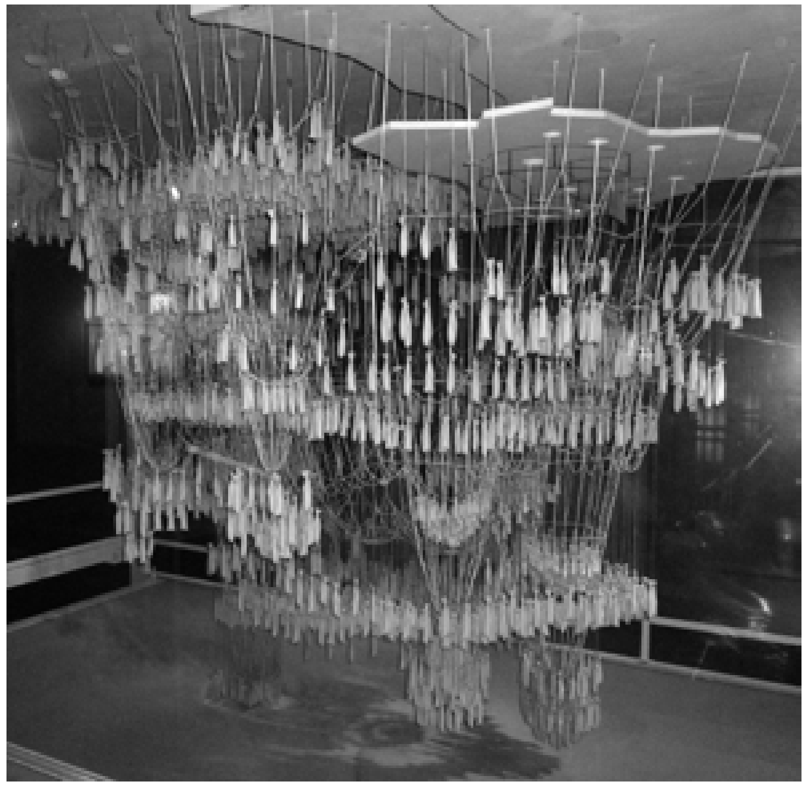


Figure 3: A team lead by Frei Otto reconstructed Antoni Gaudi's hanging chain model for the Colonia Güll church in 1982. The model is exhibited in Barcelona at Museum der Sagrada Familia.

Figure 4: Aireal photo of the Multihalle Mannheim.

2.2 Multihalle Mannheim: general background

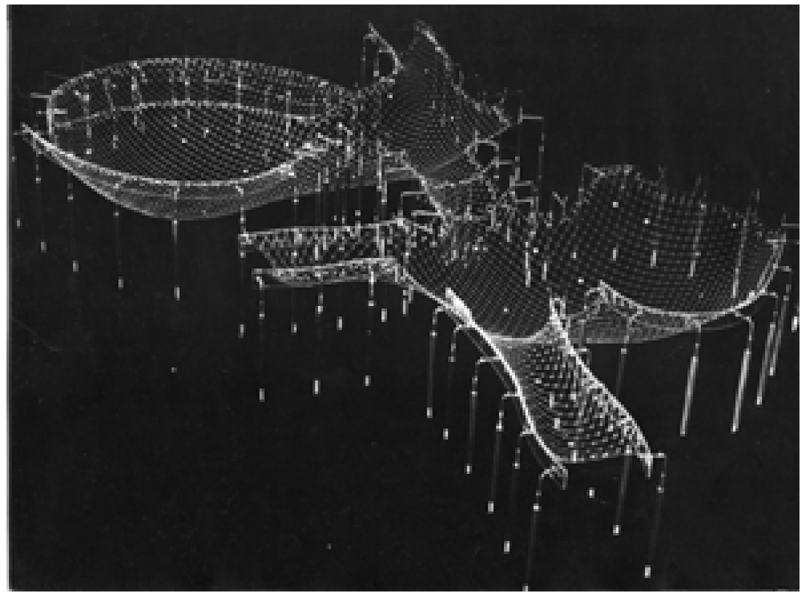
The *Multihalle* was a significant achievement in terms of developing and testing out new methods for realising constructions based on dynamic form finding techniques. The project established new grounds for form-generating techniques, the tectonic principle and the realisation of the construction. It was built in order to provide a covered space for cultural events. Originally, the primary occasion was the 1975 flower exhibition, Bundesgartenschau. Later on, other types of events such as concerts, political meetings and exhibitions took place. According to one of the architects involved, Carlfried Mutschler, the focus of the project was to create an unconventional atmosphere: a fully air-conditioned marketplace. Inspired by Frei Otto's previous work, the architects invited him to work on the project, and together they decided to generate the form as a grid shell structure based on the hanging chain principle. As Mutschler states: 'A spatially plastic total organism was then to be realized by means of the wooden grid shells spanning everything.'[4] Some of the criteria were that the hall should be inexpensive and translucent, which fitted the properties of the lightweight grid shell construction, suggested by Frei Otto.

2.3 Structure and realisation

A crucial part of developing the design was to establish precise models for simulating the double curved mesh structure, which would eventually define the shape of the realised construction. The wire model of the preliminary design consisted of a suspended net. Similarly to Gaudi's hanging chain models, it simulated the grid

4 Carlfried Mutschler in: Berthold Burkhart (ed.), *IL 13 Multihalle Mannheim*, Institut für leichte Flächentragwerke, Germany, page 28

Figure 5: The final analogue model of the structure with the suspended wire mesh.

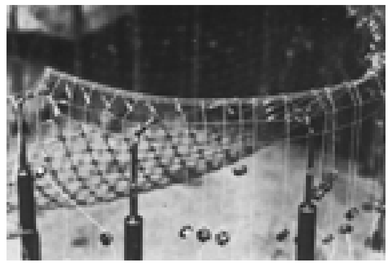


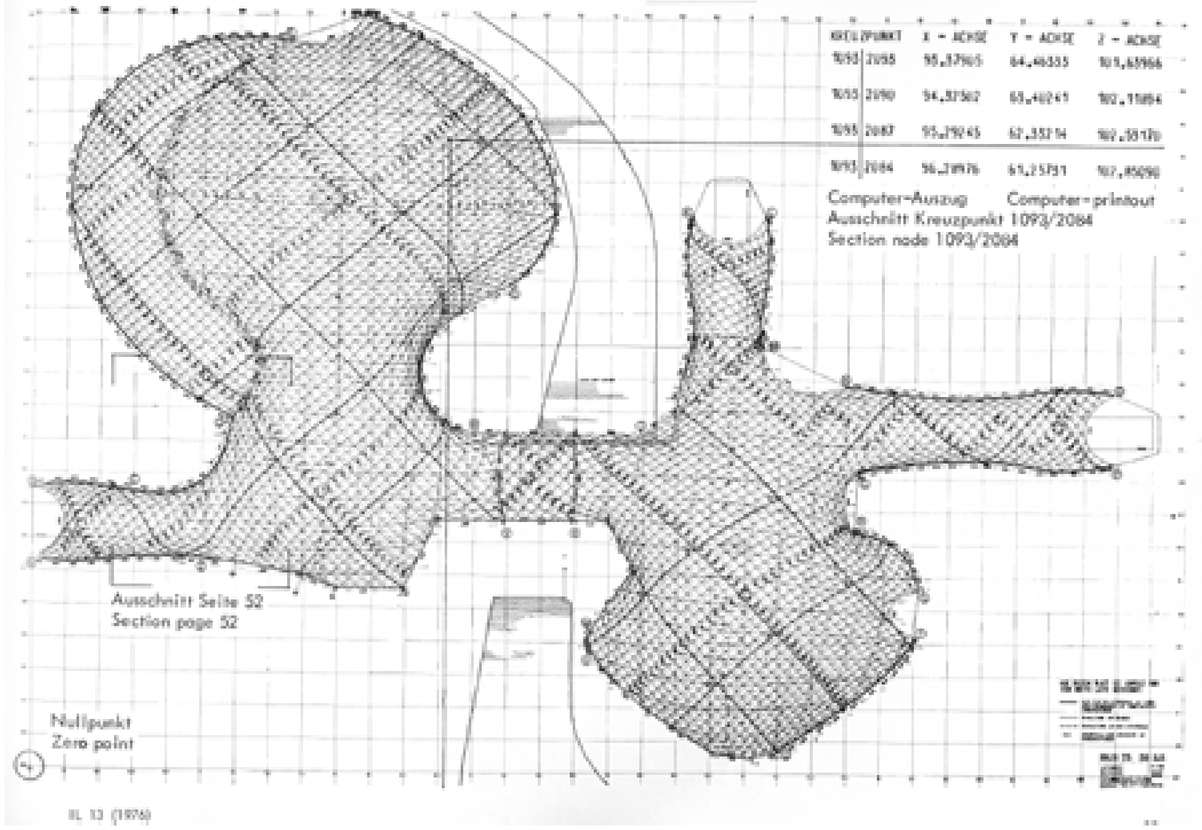
meshes of the standing compression stressed grid shell. With no outer loads (such as wind and snow), there would be no bending strains in the structure. The model had a coarser grid than the realised building, which meant that the missing mesh curves had to be interpolated from the ones represented in the model.

The model parts were produced at the Institute of Lightweight Structures at the University of Stuttgart. The system consisted of members and nodes, represented by wires that were bent into hooks in the ends and rings. This allowed the mesh to form freely according to adjustments made by weights at the edges. As the model was precisely measured and informed the actual project, it was necessary to determine the exact scale of the model components. The grid was oriented diagonally in relation to the plan in order to avoid imbalances between the structural layers, which would otherwise stem from structural lines of varying lengths. This problem was difficult to overcome, and in the end it was discovered that a slightly different orientation would have been optimal. Foreseeing the following developments in digital simulation tools, Ewald Bubner (who worked on the wire model) notes, 'As manual work is expensive in constructing models, the question has to be asked whether, instead of the model construction, other methods for form finding could not be found, i.e. by means of drafting or calculating.' He discusses the possibility of using computers in the calculation process, but finds it difficult to imagine that a process of iteratively calculating and drawing the states of the shell, as it reforms, would be possible within a reasonable amount of time.

Although the physical simulation model formed an important starting point for the calculations, the precise information concerning the positioning of nodes, lath lengths, forces and angles was actually generated through use of computers. Despite the high level of detail

Figure 6: It was possible to adjust the shape of the mesh indirectly by adjusting small weights hanging from the edges of the mesh.



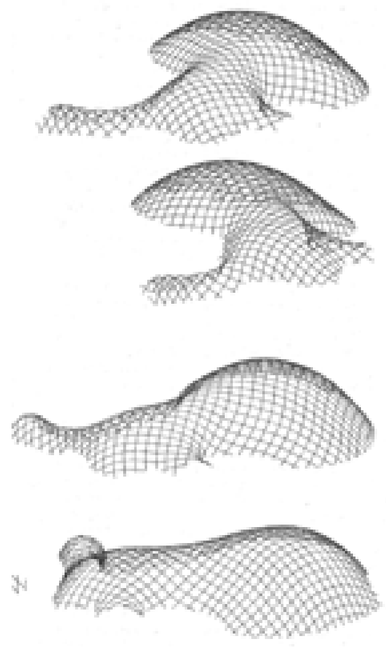


in the analogue model, it was necessary to calculate the data with technical precision, as it was impossible to realise the scale model with all details in the actual construction. The analogue simulation model was translated from photogrammetric procedures into numerical data, which were then fed into the calculations performed on computers.[5] Plan drawings and data necessary for the realisation process, referred to as 'pattern data', was then derived from the calculated structure. The plan drawing was of immense importance, since it specified the exact positions of all of the nodes in the system. As Joachim Langner mentions, 'the architects received a twelve-meter long, 5 column, closely printed computer print-out', referring to a table of node coordinates.[6] Fortunately, the available technology also helped to generate a much more useful plan drawing from the data.

The structure of the *Multihalle* is essentially a grid shell similar to the structures developed by Vladimir Shukhov around 1900. The *Multihalle* structure is realised as a double wooden grating. The structure has four layers of laths, creating a nearly rectangular mesh. The suspended net in the model wire model simulates the grid mesh of the standing compression grid of wood. The basic construction principle of the grid shell is that the mesh

Figure 7: Above is the generated plan drawing with examples of coordinate data at the top right.

Figure 8: Perspective views of the calculated mesh, still with a limited amount of mesh lines.



5 Frei Otto in: *IL 13, op. cit.*, page 11

6 Joachim Langner in: *IL 13, op. cit.*, page 51


Figure 9: The process of erecting the structure was a complex task of gradually raising the supporting scaffolds and moving the lifting points, depending on the behaviour of the mesh.

is assembled as a flat mesh, but with joints that allow the laths to change relative angles. The grid is then gradually raised from the ground, and because all of the edges become fixed, the double curved shell stays in place. Additionally, the joints must be tightened to eliminate the flexibility when in position. To provide the necessary stiffness of the *Multihalle*, a series of wires running diagonally across the grid are post-tensioned. The erection of the structure was achieved through the use of scaffolding towers, which could be elevated gradually depending on how the shaped formed. In order to achieve translucency and lightness, a membrane of Trevira fabric coated with PVC was used to cover the structure.

2.4 Multihalle Mannheim: conclusive remarks

It is remarkable, how Frei Otto managed to explore the full potential of the 3-dimensional catenary shapes through the use of the most advanced technologies of the time. By translating the suspended model into a complex post-tensioned structure consisting of lightweight components, it was possible to realise a structure of very large proportions. The project demonstrates a turn in time, where the use of computers became useful in the development of complex architectural projects. It is interesting to note how these

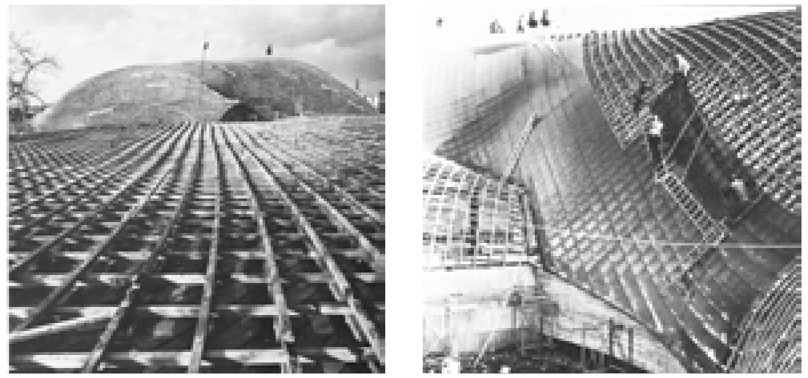


Figure 10: The final structure with four layers of laths, joined and fixed in position with a bolt in each crossing. Right: the mounting of the PVC-coated textile.

early uses, were mainly oriented towards calculating data and generating geometric information. In the following period, as the proliferation of computer technology continued, it became much more considered as an advanced drawing or modelling tool. Only in recent years has the idea of using the computer's fundamental capacity of calculating complex mathematical relations re-emerged as a useful tool in the design phase. Not only is this technology used for solving complex calculations of construction dimensions etc, but as a way of developing the architectural design. While the *Multihalle* demonstrates the great potential of using dynamic form generating methods in the architectural design process, it is also worth noting the amount of work involved in resolving the geometry, the tectonic system and erecting the construction. It is reasonable to state that a large volume and a unique and expressive construction has been realised with very limited material use, but also that an unusual


Figure 11: Multihalle mannheim. Flower exhibition Bundesgartenschau in 1975, Entrance area.

amount of time was spend during the phases of designing and calculating. As Frei Otto notes, 'The work on such buildings does not take place consecutively, much is done simultaneously. Innumerable negotiations take place.'[7] In this sense, the project is comparable to a research project, and many parties involved with its development considered it as such. This altered balance between developmental resources compared to construction resources reveals why this type of project is relatively rare. A very strong economic structure exists, directing these balances in the vast majority of architectural projects.

---

7 Frei Otto in: *IL 13, op. cit.*, page 13

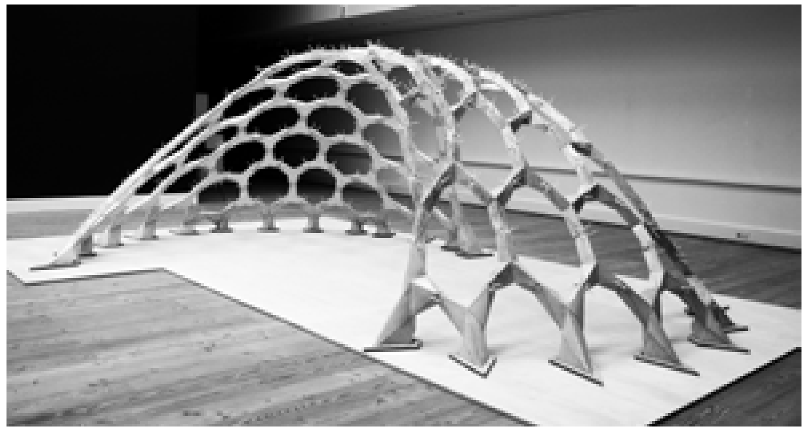


Figure 12: Concrete Gridshell Pavilion. Ole Egholm Pedersen, Dave Pigram and Niels Martin Larsen, Aarhus, 2010.

2.5 Method for constructing a concrete gridshell.

The use of three-dimensional hanging chain models, first used by Gaudi and later by Otto, is the basis for digital methods for simulating dynamic relaxation, as used in the method *Complex Gridshell*. In Figure 12 is shown the case study *Concrete Gridshell Pavilion*, which was constructed as part of the method development. This project is described in detail in Chapter 8.1 and is compared with the *Mannheim Multihalle* in the following section. *Concrete Gridshell Pavilion* exists on a different level, due to its character as a quickly developed temporary project of small scale. Still, it is relevant to consider the similarities and differences between the two approaches.

The concrete gridshell was produced in Aarhus, as a result of collaboration between the University of Technology in Sydney and the Aarhus School of Architecture in October 2011. The project was based on previous research in algorithms for dynamic relaxation and sustainable methods for concrete casting through the use of PETG sheet. Scripting tools and digital production techniques, in the form of laser cutting machines, were an essential part of both the development and realisation of the project. The form-generation process was controlled through the ReVault application, developed by Dave Pigram and Iain Maxwell. The application imports a planar mesh, consisting of connected lines, and through thousands of iterations, the optimised shape gradually reaches an equilibrium state where all members contain only tensile forces, as shown in Figure 13. The nodes where the structure is connected to the ground are defined as fixed points. When the structure is inverted, the forces are translated to pure compression, similarly to the physical suspension models used by Frei Otto. The algorithm simulates to some extent a physical spring system by gradually repositioning the nodes of the grid towards equilibrium states. That is, when the springs' force and gravitation is equally balanced. As the repositioning happens gradually, a negotiation between the nodes is established through the connecting springs. Thereby, the system slowly approaches the optimised shape. Parameters, such as member length and massing of nodes can be adjusted, allowing a wide spectrum of optimised solutions to be tested in a short time span. As with Frei Otto's physical models, the positioning of the anchor points as well as the structure of the initial mesh is of great importance. The latter is predominantly concerned with the subsequent generation of volumetric geometry and the production of components, since only certain types of components were defined within the generative system. Although the process is described in detail in Chapter 8.1, here is a short explanation of the process from form-generation to realisation. When the dynamic relaxation system arrives at an equilibrium state, the geometry of the mesh is exported from the ReVault application to a 3D modelling environment. Here, custom-made scripting tools are used to generate volumetric geometry and cutting patterns for laser cutting the PETG sheet. The latter is subsequently folded and assembled to create the templates for casting the concrete components. The scripting tools allow all the components to have individual geometry. The developed scripts are able to analyse the mesh structure, and detect the type of the components. The generic component is Y-shaped, and comprises of three individually folded arms. Special component types appear along the edges in the form of linear pieces with two arms, and at the anchor points, where larger components ensure a firm connection to the ground. Additional scripts help to generate drawings and measures for the scaffolding, which is made of laser cut cardboard. In the case study in Aarhus, recycled cardboard was used. For the assembly of the structure, the 3D model serves to direct the construction process, since all components are tagged and identifiable.

Besides the large scale and the substantial complexity in *Multihalle Mannheim*, it is worth noting some additional differences between the *Multihalle* and the *Concrete Gridshell Pavilion*. In regards to the *Multihalle*, the construction is lightweight and consists of layers of laths. The mesh geometry in the two examples is fundamentally different. Within the *Multihalle*, a somewhat quadratic grid is established, where the concrete gridshell is based on a hexagonal grid. This results in the forces being unable to travel in direct lines across the structure, thus producing bending forces

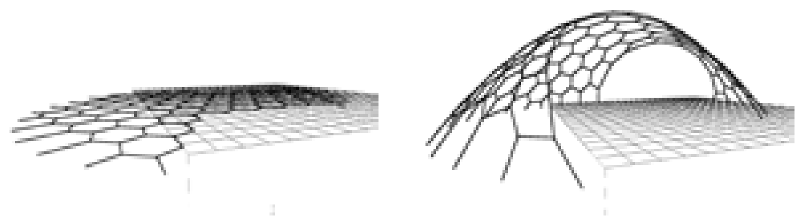


Figure 13: Concrete Gridshell Pavilion, dynamic relaxation simulation. Top: Initial state of the planar mesh. Bottom: Mesh in equilibrium state. Gravity is inverted.

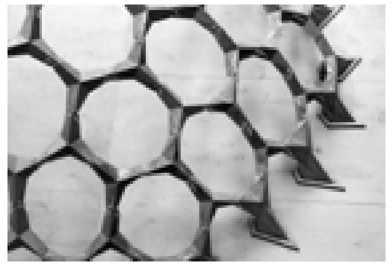


Figure 14: Concrete Gridshell Pavilion. Case study, Aarhus, October 2011. The components are differentiated both in proportion and type. The base components are a of a distinct type.

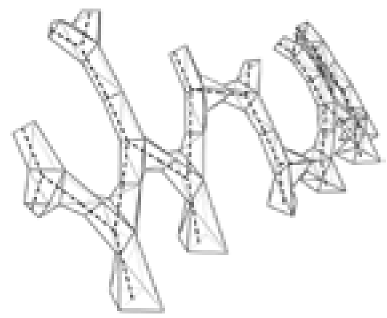


Figure 15: The method Complex Gridshell. Scripting tools serves to generate volumetric geometry and cutting patterns.

within the building components. Material-wise, the use of lightweight material was crucial for the realisation of the *Multihalle*, which was one of the largest compression-loaded structures at the time.[8] The choice of material for the concrete gridshell is to be considered as an a priori decision, rather than the most efficient for a particular purpose. The realised case study in Aarhus was not developed by a programme, but rather from loose design intents. These intents were to realise a structure that would suggest an architectural scale, and that the structure would serve as a landmark with respect to a temporary exhibition. The method development was directly based on experiments with concrete, and the main goal was to reveal the potential for using concrete in new ways. That is, sustainable methods for applying discrete components to architectural constructions. In this sense, the dynamic relaxation method achieves a minimal use of concrete compared to the size of the structure. The structure in Aarhus was not dimensioned in order to resist loads other than gravity. This was reflected in the joints, where simple cable ties were used to resist a limited amount of sheer forces, mainly from lack of precision and dynamic loads during construction and from people interacting with the structure. Besides the limited strength of the joints, it is reasonable to assume that the strength of the concrete material could be an advantage in terms of additional loads from cladding or people climbing the structure. The latter condition occurred in a third case study, briefly described in Chapter 8.1. In this sense, the use of concrete holds potential that is different from those obtained with lightweight materials, as used in the *Multihalle*.

2.6 Analogue and digital methods

These projects display interesting differences between analogue and digital form-generation. As stressed in the previous chapter, Frei Otto's team did in fact use computation to arrive at the final precisely positioned mesh structure. However, the overall form-generation was achieved with analogue computation. Contextually, it would not have been possible to generate the whole morphology from an initially planar grid, as it was done with the concrete grid shell, with computer technology in 1971. Analogue computation supersedes even modern computers in calculation time, if we consider a situation where the model is already constructed. In other words, if a suspended net (such as the ones Frei Otto's team used) is de-stabilised by a disturbed equilibrium state, it very quickly regains, or recalculates, the optimal shape. The main difficulty is to construct the model, often through carefully orchestrated steps. Due to the substantial amount of resources related to this, it is crucial that the designers have an extremely well developed understanding of the way that the model will behave once constructed. Even though the physical models by nature encourages an intuitive approach to the form-generation process, digital tools provide means for testing out a much larger variety of possible solutions. Simply by editing the two-dimensional drawing of the initial mesh, the whole structure can be entirely re-organised and re-shaped within a few minutes. This should be compared with the time it would take to rebuild the whole physical model with chains and weights etc. This rapid search through a field of possible solutions is reflected in Figure 16, which shows various configurations of *Complex Gridshell*.

8 Frei Otto, Bodo Rasch, 1995, *op. cit.*, page 140.

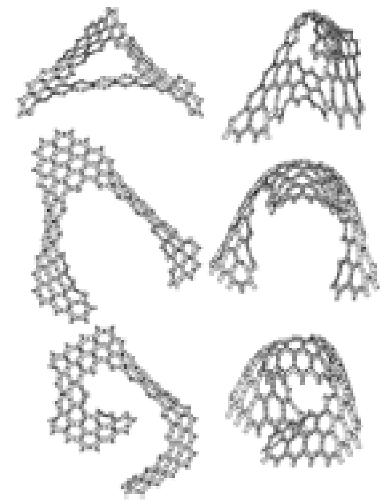


Figure 16: Complex Gridshell. Digitally generated variations of the structure.

Also, in terms of extracting the precise data from the model the digital version has huge advantages. The first advantage concerns precision. It is a complicated task to measure the three-dimensionally and exact position of every node in the model. Similarly to the Mannheim project, the physical model sometimes lacks the necessary resolution to adequately represent the realised structure. Therefore, additional positions must be extrapolated from the measurement data. It should be noted that realistically, a digital simulation for a structure as complex would probably be generated from a reduced mesh. Subsequently, a difference would be that the extrapolation could be carried out automatically from the generated result. The second advantage concerns the transmission of the generated data to further processing with respect to manufacturing and construction. With regards to the Mannheim project, computation played an important role in producing data sheets, necessary for the control of the structure's production and construction. A challenge was then to transform the measurements from the physical model to a digital format for further processing. This transformation occurred seamlessly within an entirely digital process, as with *Complex Gridshell*.

The physical and digital outcomes of the Form Finding methods are primarily controlled by the adjustment of anchor points, mesh geometry and member lengths. However, with digital form-generation it is possible to embed parameters and forces other than gravitation, in the logic of the algorithm. These additional parameters reflect programmatic issues, constraints from manufacturing or construction, design intents and additional performance criteria. This allows different parameters to negotiate within the form-generating process. In order to establish this type of method, it is necessary to be able to modify the internal logic of the form-generating software, or at least to have an application where the behaviour of the system can be modified. This was partly the case with the *Concrete Gridshell Pavilion*, as one of the responsible researchers was a co-author of the form-generating application. In this case, additional parameters were disregarded, as it was necessary to test a version where the

Figure 17: Concrete casting of distinct components with PETG templates.

structural capacity was prioritised. However, there was a degree of control of the outcome through the adjustment of member length, due to the limitation of component size.

2.7 Materiality and scale

With regards to the *Multihalle*, essential properties of the wood laths were taken into account with respect to the overall design of the structure. For instance, it was crucial for the laths to bend and contain tension forces, not only compression. Similarly to *Concrete Gridshell Pavilion*, material properties informed the design and realisation process in different ways. The PETG sheet used for the case study in Aarhus was of a very limited thickness, and therefore subject to deformations from the weight of the fluid concrete. This placed limitations on both the component extents and the thickness of the arms. It was realised that the templates could be reinforced, by adding a number of folds and ridges, as seen in Figure 17. These features were then implemented in the algorithms that were used to generate the cutting patterns for the templates. This helped to maintain precision of the components during production. The thickness of PETG sheet and the weight of concrete were balanced to minimise material use and avoid deformations. Due to these implications, it would be very difficult to perform a similar case study in a different scale without making substantial changes to the materials used and their dimensions. An optimisation of material use and production technique was present, besides the structural form-optimisation. This was at the expense of the possibility of direct scaling of the method. However, through changes in thickness of the PETG sheet for the templates, and the possible integration of other materials such as steel, the experiment is reproducible at a larger scale. This again affects the choice of production technique, which may have to shift from laser cutting to other production techniques such as water jet. Other issues, such as sustainability and methods for assembly, will also have to be reconsidered.

2.8 Feedback loops in the development process

During the development of the *Complex Gridshell* method, an integrated system with a series of feedback loops, ranging from initial form-generation to final construction, was established. In Chapter 8.1, Figure 2 reflects the flow of information and material. The diagram is vertically divided into two main parts, where the upper section represents the digital generation of information, and the lower section represent actual production and construction. As the arrows suggest, information from each part can feed back to the first part of the system. This denotes that the project on all levels (from definition of overall form, through component design, to construction method is accessed by the design team from the beginning of the process. Although this differs from most architectural projects, it is interesting to note that a similar diagram could be drawn for the *Mannheim Multihalle*. Here, the detailed solution was embedded in the logic of the system that generated the overall form from the beginning. Generally, architectural projects are resolved in a way, where the overall contextual relations and shape of the building is defined at an early state, together with main programmatic distribution of spaces. Often only vague ideas concerning materials and tectonics are present in the early phases. Phillip G. Bernstein articulates that the 'Construction insight is wholly missing from the conceptualization of the design, and design insight is applied only sparingly during its execution.'[9] In this sense, both the *Multihalle* in Mannheim and *Complex Gridshell* represent an alternative to prevalent architectural design methodologies. Although the *Concrete Gridshell Pavilion* in Aarhus was manageable due to its very limited scale, the *Multihalle* project demonstrates its great potential. Further examples suggest that this design methodology is beneficial, particularly when more experimental types of architectural projects are concerned. A more recent example of this approach is the *National Aquatics Centre in Beijing*, described in Chapter 6.2. A key factor of these projects is the fact that architects and engineers have collaborated in order to establish structural properties as an integrated part of the design process. In these projects, the structural, material and constructional aspects were not seen as secondary in relation to the formative. On the contrary, these factors supported the essential design intents. These projects did not compromise their essential parameters, but merged them through a process of renegotiation.

2.9 Conclusion

Complex Gridshell and the *Multihalle* in Mannheim serve as examples of self-organised surface morphology. Both methods illustrate the use of funicular models in analogue and digital forms. A surface-based approach may have explored a broader perspective without the funicular principles. On the other hand, some related methods have even more emphasis on the role of the surface. An example of this is the method for constructing masonry vaults,

9 Phillip G. Bernstein, 'Models for Practice: Past, Present, Future', *Building (in) the Future*, Princeton Architectural Press, 2010, page 194

as carried out by Philippe Block and colleagues.[10] However, the examples in this chapter help to raise relevant topics on different levels. It is illustrated how physical simulation can be an integrated part of the design process, and how generative techniques can help to establish feedback loops between construction aspects and design intentions. A new sustainable technique for concrete casting was implemented in this case study through the use of digital production. Subsequently, material properties have influenced the method development. Both examples demonstrate how an iterative collaboration can inform the design process, and improve the result. These methods are concerned with morphology based on a priori topology. Self-organised topology with respect to the overall form is discussed in Chapter 4. First, we take a look at self-organised pattern distribution.

10 L. Lachauer, m. Rippmann, P. Block, 'Form Finding to Fabrication: A digital design process for masonry vaults', *Proceedings of the International Association for Shell and Spatial Structures Symposium 2010*, Shanghai

3 Pattern distribution

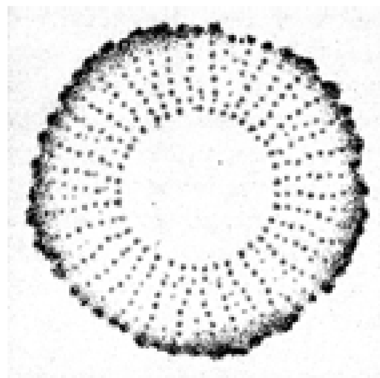
3.1 Surface based pattern distribution

The discussion in the previous chapter was primarily concerned with morphological challenges, or more specifically, how a surface structure can self-organise into an optimal shape in terms of energy level. In the following section, the focus is alternatively directed towards ways of modifying and articulating an a priori formed surface at a detailed level in the form of a surface pattern. The discussion is centred round the *Self-organising Bezier Curves* method, further described in Chapter 8.2. This method demonstrates principles for generating self-organisational patterns, and how these patterns can be linked with a form of architectural expression where both aesthetic design intents and performance criteria negotiate the pattern articulation. The method is not based on the structural or material-wise tectonic principle, but rather deals with geometric articulation. Chapter 8.2 outlines possible techniques for the method's realisation. This chapter focuses predominantly on pattern formation within architectural 'deep' surfaces. Here, the 'deepness' refers to the three-dimensionality of an architectural surface construction, or facade. Certain topics related to pattern formation are discussed, such as linearity versus nonlinearity, periodicity, self-organisation, adaptability and complexity versus consistency. Initially, the fundamental algorithmic principle of the method is explained and related to examples from science and algorithmic design research. Examples of the use of surface patterns in architectural design through architectural and student work performed at the Aarhus School of Architecture will follow. Finally, the method is related to additional developed methods that are relevant to this research.

3.2 Algorithmic logic

The self-organisational principle that forms the basis of the method *Self-organising Bezier Curves* corresponds with some of the investigations concerning self-organisation, described by Paul Coates.[1] In some of his examples, very basic agents, represented by a point in a two-dimensional space, self-organise to form geometric patterns of varying complexity. The basic behaviours that force the agents to shift position are repulsion and attraction. For instance, when repelled by a single point with a certain force, the agents move away. If the agents are simultaneously attracted to the same point,

Figure 1: Computer simulation of E. Coli bacteria. Eshel Ben-Jacob.



1 Paul Coates, *Programming Architecture,* Routledge, Oxon, 2010, page 13.

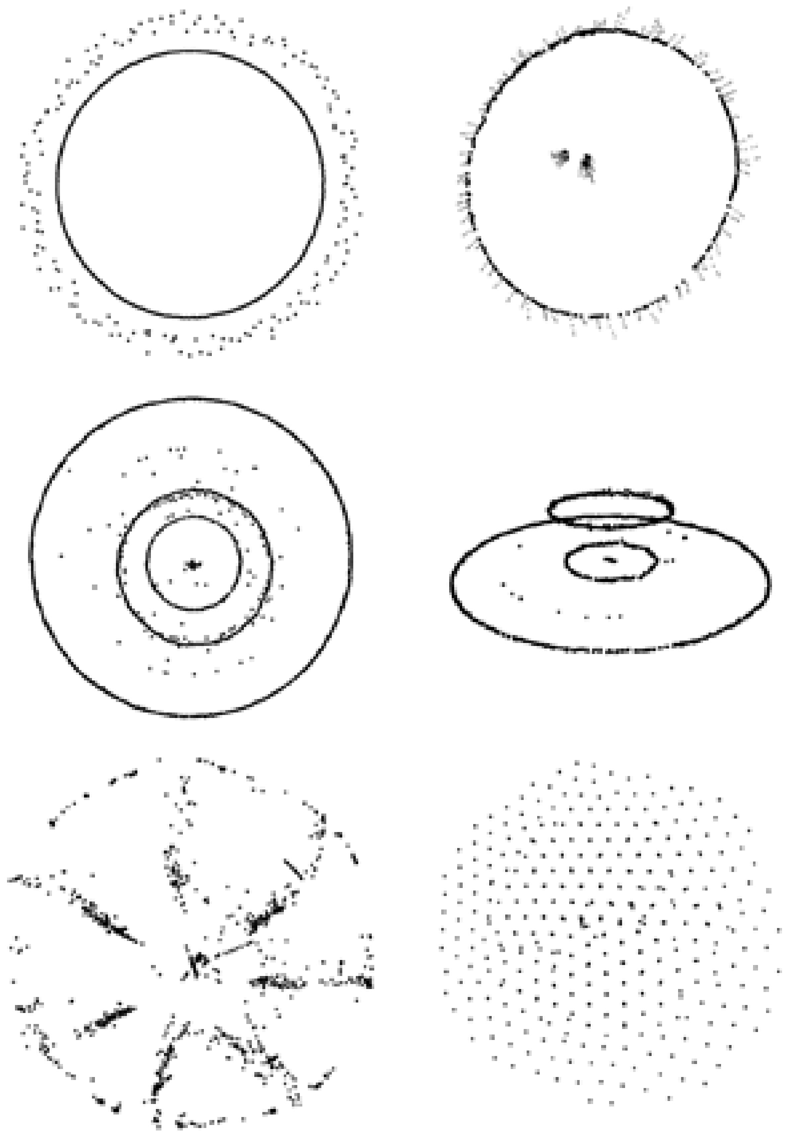


Figure 2: Patterns formed with self-organising agents.

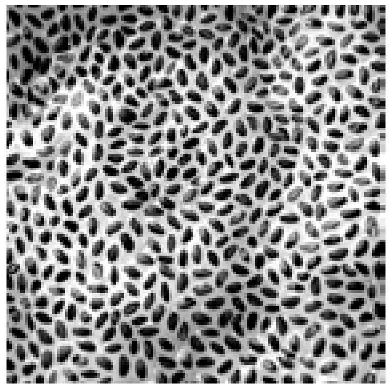


Figure 3: Cress seeds on wet cotton self-organise into a homogenous pattern.

they will begin to form a circle and remain at the perimeter where the attraction overrules the repulsion. This basic behaviour is also used in scientific algorithms that simulate bacterial behaviour. An example is illustrated in Figure 1, where the behaviour of E. Coli bacteria has been simulated by a team led by Eshel Ben-Jacob at Tel Aviv University.[2] The model simulates how complex radial patterns emerge from basic behavioural rules, reproducing patterns recognised in natural bacteria colonies. Figure 2 illustrates a series of experiments where agents self-organise and form complex patterns, as performed by the author. The agents are governed by rules of repulsion and attraction. All patterns were generated in a three-dimensional environment, influenced by gravity and a ground plane. The two results at the top of the chart are two-dimensional, whereas the others outline spatial formations. The simple mechanisms of repulsion and attraction are sufficient to generate order on a level that is different to the rule set that governs the form-generating process. In this sense, the pattern formation is an emergent property. Within a system of fixed agents, representing centre points in cells and a logic based only on repulsion and attraction, it is possible to make a population of agents that self-organise into a Voronoi pattern. A two-dimensional example by Paul Coates is shown in Figure 4, and as he documents in *Programming Architecture*, the code for generating a self-organised Voronoi pattern is far shorter and less complicated than a method constructed from computational geometry. [3]

As indicated, the method developed for constructing a surface pattern of self-organising Bezier curves is based on these basic behaviours. Figure 5 shows how the first step of the process manages the distribution of a field of centre points, only from a rule of repulsion. The image on the left shows the initial disorganised field of circles. The centre of the image demonstrates how after 25 generations the circles have moved away from their neighbours, but not far enough to create the specified distance between them. The

2 Phillip Ball, *Shapes*, Oxford University Press, Oxford, 2009, page 147.

3 Coates, *op. cit.*, page 15.

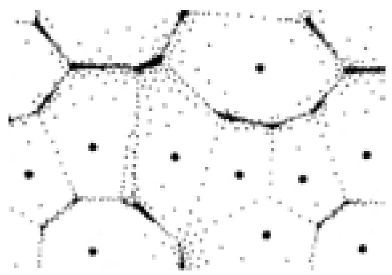
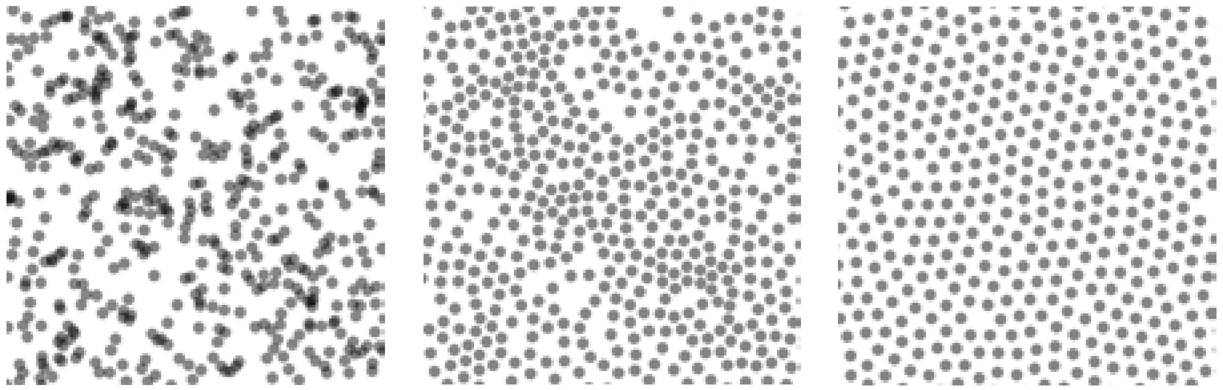


Figure 4: Self-organising agents forming a Voronoi pattern. Paul Coates.

Figure 5: Self-organising circles.First step of the method *Random Bezier Curves.*



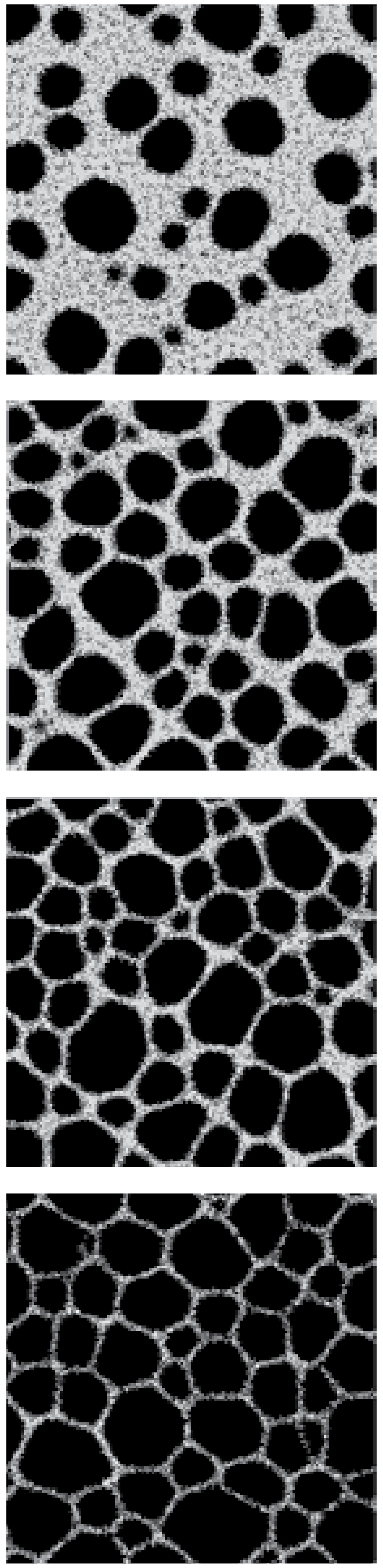


image on the right shows the pattern in an equilibrium state after 200 generations. The circles have stopped moving because they all have the specified minimum distance to their neighbours. The result is a homogenous pattern. The mechanism can be compared with a simple self-organising behaviour that occurs when cress seed are placed on wet cotton. The seeds form a gel, which forces them to rotate and distribute evenly, as shown in Figure 3. These patterns are aperiodic since there is no precise repetition. Furthermore, they display a nonlinear form of organisation because the position of each element is the result of continuous local interactions. Each element cannot be positioned independently from the adjacent parts, but only through iterative negotiations between the element and its immediate neighbours. When one element is moved, it affects the entire system. In a sense, the field can be compared with a *cellular automata*, where the state of each cell depends on the current state of its neighbours.[4] The difference here is that the state of the cell is represented through its positioning, and more importantly, the relations between cells are not a priori, but dynamic.

Figure 6 shows a more complex type of simulation directly related to biology. Here, the behaviour of 'true slime mould' Physarum polycephalum is simulated through the use of a dynamic agent based system. This type of slime mould is a single celled organism that can form complex structures in order to distribute nutrients and to find food sources. As Jeff Jones explains, 'the true slime mould Physarum polycephalum, exhibits a very wide repertoire of pattern formation behaviours used for growth, movement, food foraging, nutrient transport, hazard avoidance, and shape maintenance.'[5] The behaviour of slime mould, Physarum in particular, is a well-studied phenomenon, and has demonstrated how complex transportation systems can be simulated through experiments with slime mould. The agent-based simulations are driven by basic behavioural rules, and the pattern formations are generated spontaneously, that is, they emerge from the agent's interactions. The agents represent particles of the Physarum plasmodium gel-sol structure, and move forward or randomly change direction, depending on a 'chemo-attractant gradient'. Essentially, local interactions within the organism result in gradually changing formations that are optimised in terms of supporting the survival of the organism. In other words, this is an explicit example of self-organisation. As the figures of simulations demonstrate, behavioural rules are sufficient to generate a variety

Figure 6: Simulation of behaviour of 'true slime mold' Physarum. Different populations size. Illustration: Jeff Jones.

4 Wolfram, Stephen, *A New Kind of Science,* Wolfram Media Inc, USA, 2002, page 28..

5 Jeff Jones, 'Characteristics of Pattern Formation and Evolution in Approximations of Physarum Transport Networks', *Artificial Life, Spring 2010, Vol. 16, No. 2* , pages 127-153.

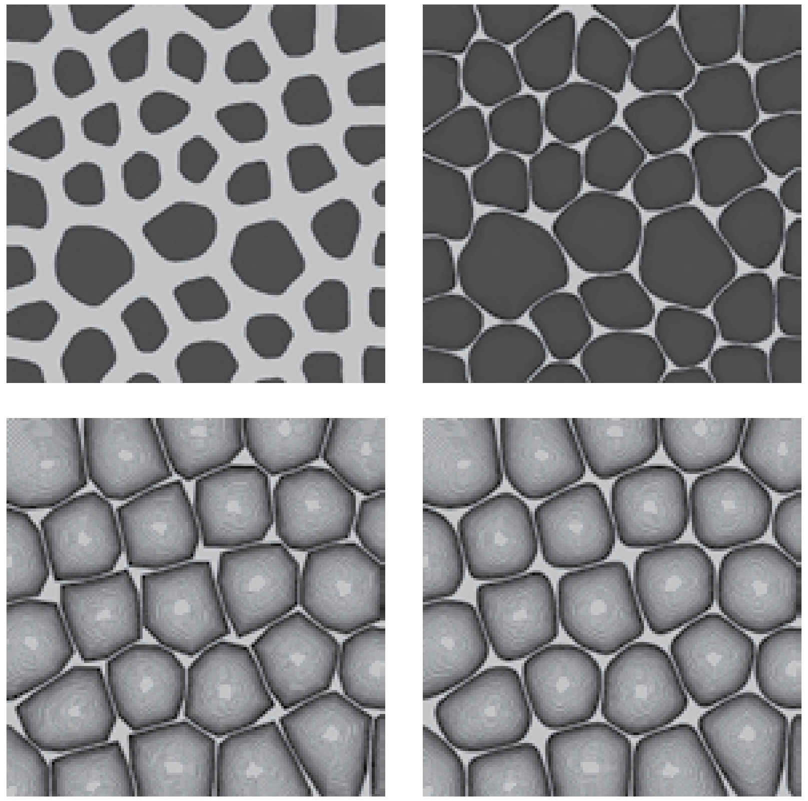


Figure 7: Self-organising Bezier curves. Top: Different adjustment of distance parameter. Below: shapes grow from centre points outwards. Left: underlying polygonal geometry. Right: Resultant Bezier curves.

of complex patterns.

The method *Self-organising Bezier Curves* was not developed for creating patterns that resemble slime mould patterns. Interestingly, the appearance of the patterns in Figure 6 and those generated with Bezier curves in Figure 7 share similarities. However, there is a fundamental difference in the way they are generated. The behavioural rules that direct the self-organising curves are unrelated to natural phenomena but based on geometric relations. With regards to the slime mould simulation, dark holes occur in passive areas, surrounded by the dynamic structure of the slime mould organism. With regards to the Bezier Curves, neither the interior of the curves or the area between them is 'active.' Only the borders that separate the two area types is generated and manipulated. As described in detail in Chapter 8.2, the curves are inscribed in polygons and

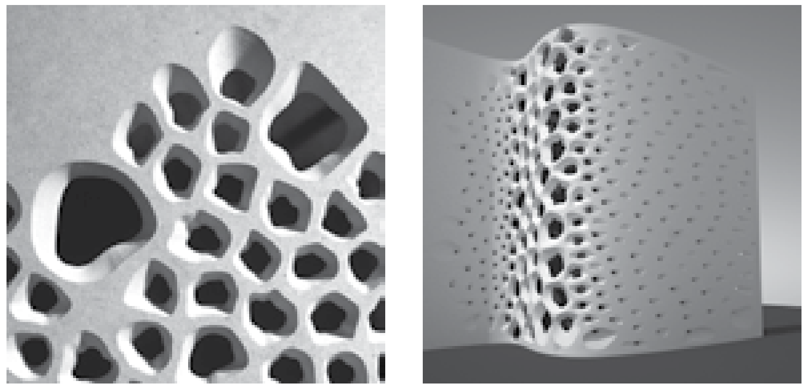


Figure 8: Self-organising Bezier Curves, volumetric geometry. Left: CNC milled sample of three-dimensional pattern. Right: Rendering of adaptable surface pattern.

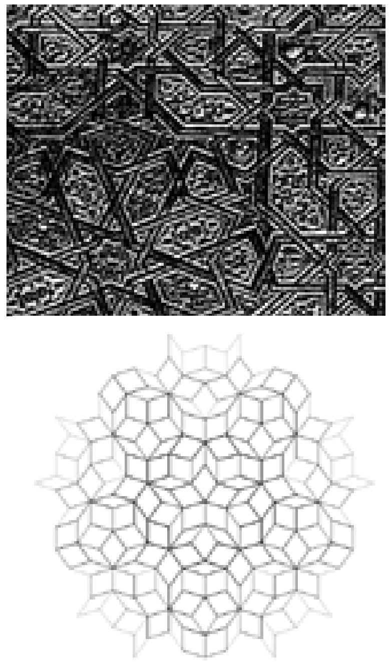


Figure 9: Top: Arabic patterns in the Alhambra Palace, Granada, Andalucia, Spain. Below: Aperiodic Penrose tiling

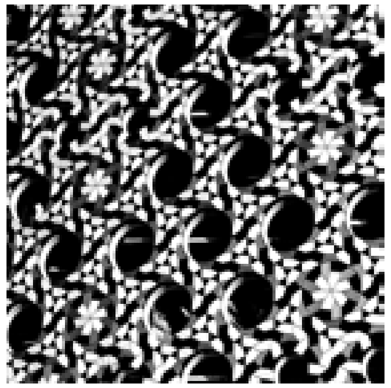


Figure 10: Forreign Office Architects. Ravensbourne College. London, 2010.

Figure 11: Forreign Office Architects. Spanish Pavillion. Expo 2005.

are constructed from straight lines. The pattern formation emerges through an interaction between the polygon vertices and the lines connecting them. Figure 7 shows the underlying polygon geometry and the visualised pattern of the Bezier curves. The illustrations also show how the shapes are initially formed near the centre of the polygon, and then grow outwards until a certain distance from their neighbouring shapes is reached. The self-organisation takes place in two steps. First, distribution of the circles, or centre points occurs. Then follows generation and expansion of the polygons and Bezier curves. In order to address the issue of surface depth, the different states of curve generation are related to different depths. The curves generated initially are placed on the interior of the surface thickness, and the proceeding curves are placed on the exterior. Subsequently, they are connected by new surfaces forming the inner sides of the openings. On the left-hand-side of Figure 8, a CNC milled sample of the 'deep' pattern is shown. The black background is seen through the penetrated surface. On the right-hand-side, an example of how the method was further developed towards a self-organising adaptive facade system is illustrated. In Figure 8, the orientation of the surface affects the distance between the curved penetrations, the initial size of the polygons, and how far they can expand. As such, the facade pattern is nonlinear, self-organising and adaptive.

3.3 Facade patterns

This section will focus on periodicity and linearity with respect to facade patterns. Some projects reflect a search for variation in the architectural expression, and one strategy is to use aperiodic patterns. In architectural history, Arabic patterns are well-known for displaying a large degree of complexity. An example of a facade relief from *Alhambra* in southern Spain is shown at the top of Figure 9. Generally, these patterns are periodic through translations or radial symmetry. Some Arabic patterns can be seen as an attempt to overcome obvious repetition. The ordering logic of these patterns is so complex that it almost disappears from the observer. Some patterns display symmetries similar to the type of aperiodic tilings, discovered by Roger Penrose in the 1970s.[6] An example of digitally generated Penrose tiling is seen in Figure 9. Despite the pattern being aperiodic, it is highly structured and consists of only two different tiles. Penrose tiling has been directly used in subsequent architectural projects. A recent example is the *Ravensbourne College* by FOA, illustrated in Figure 10. Here, facade penetration

6 Raymond Tennant, 'Medieval Islamic architecture, quasicrystals, and Penrose and girih tiles: Questions from the classroom', *Symmetry: Culture and Science, Volume 19, Numbers 2-3*, 2008, pages 113-125.

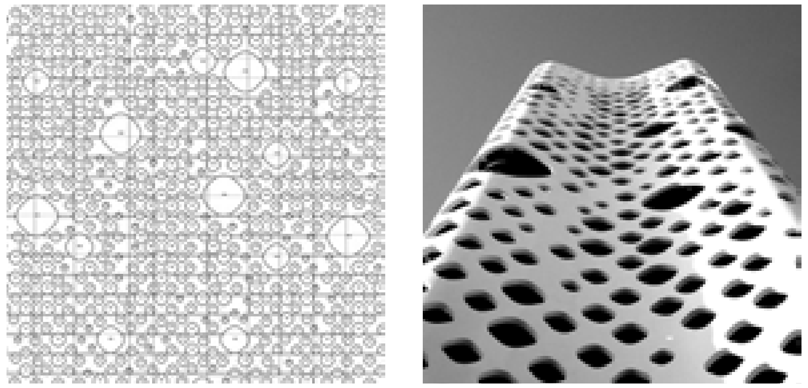
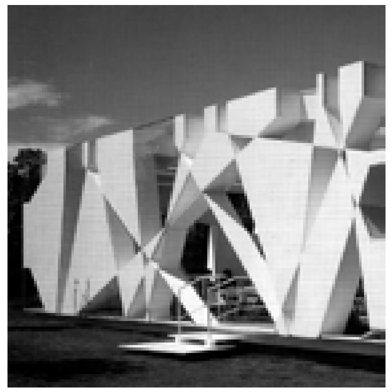


Figure 12: Reiser Umemoto. 0-14 tower, Dubai, 2007. Left: Part of facade drawing. Right: Realised facade. Illustrations: Reiser Umemoto.

in the form of approximate circular openings, retains the nature of the pattern so that the shape of the individual tiles is not obstructed. In another project by FOA, the *Spanish Pavilion for EXPO 2005*, a large degree of variation in the facade expression is achieved through the instigation of simple geometric manoeuvres (See Figure 11). Here, a cluster of six hexagons forms the repeated component. Some of the hexagon edges that reside inside the cluster are then adjusted, effectively skewing the hexagons. Because the external border of the cluster remains intact, repetition through simple translation is possible. In this method and through random colouring that transcends the cluster divisions, the result is a facade pattern that appears to be varied. Another example of an aperiodic pattern is the *0-14 Tower* in Dubai by Reiser Umemoto (Figure 12). Here, a pattern of circles of five different sizes is randomly distributed. Initially, an algorithm was used to generate the surface pattern, and its functionality reflected a variety of concerns such as structure, exposure, program, views, and luminosity.[7] These investigations helped to develop the project, whereas, the algorithmic approach was at some point replaced by three-dimensional modelling techniques in order to optimally balance various parameters. Still, the modelled geometry could be extracted through computation for further detailing in terms of realisation. A closer perspective reveals a regularity of distribution, as both horizontal and diagonal lines can be detected on the facade drawing. The facade is constructed from concrete, and the pattern ensures that the positioning of the holes corresponds with the structural demands concerning reinforcement.[8] The pattern appears to be random, which is amplified by the facade's curvature. This also increases the complexity of the geometry of the holes.

A more radical example of an aperiodic facade pattern is the

Figure 13: Toyo Ito and Cecil Balmond, Serpentine Pavilion, 2002.



7 Nicholas Solakian, 'Editor's Pick: Swiss Cheese-Laced Exoskeleton', *http://www.architizer.com/en_us/blog/dyn/15558/editors-pick-swiss-cheese-laced-exoskeleton/*

8 Thomas Lane, ''0-14 tower, Dubai: The hole story', *Building.co.uk, February, 2009*.

Serpentine Pavilion by Toyo Ito and Cecil Balmond, shown in Figure 13. The facade pattern, or tectonic pattern, is based on a geometric algorithm, developed for the project. The algorithm scales and rotates a square, and the crossing edges of the square define the pattern, which is eventually mapped onto the building's roof and facade. The tectonic solution is based on a structure of steel flat. The panels shift between glass and white aluminium, effectively generating an expression of a complex tectonic pattern.[9] The pattern is aperiodic, but also linear in the sense that there is a direct correlation between the initial state and the generated result. The positioning and form of individual parts does not, as a pattern, depend on other parts. Another case is an actual tectonic solution where the parts are interconnected. But in this case, this is a result of the pattern formation. The example demonstrates the application of a pattern that is generated separately from the volume it is covering. Naturally, the design of a pattern is directed towards its application. It is not based on, or initially constrained by, the surface geometry of volume. In this sense, it may be considered as a bottom-up approach, despite the fact that the pattern elements do not generate wholeness. Rather this approach is applied to the whole as a separate pattern, or image. The algorithmic pattern-oriented wholeness is largely achieved via the expressive tectonic solution. The solution can be compared with the *National Aquatic Centre in Beijing*, which is described in detail in Chapter 6. Here, the building also appears as a regular box, covered with a complex pattern. A difference exists, however, as the pattern reveals itself as a cut through a three-dimensional structure. Consequently, there is a stronger part-to-whole relationship.

3.4 Nonlinearity and adaptation

Certain types of computational tools allow parametric control of variation in the surface articulation. A typical example of this approach is shown in Figure 14. Because the pattern is controlled by an underlying regular grid structure, the pattern formation is linear. There is a direct link between underlying geometry, surface morphology, local surface articulation and building components. Also, the position and forming of each surface part can be generated directly from the surface geometry, independent of neighbouring parts. This is less obvious in the appearance of the building, since a smooth transition between the surface-parts has been ensured. In this sense, the surface articulation and the underlying geometry are

9 Neil Leach, David Turnbull & Chris Williams, *Digital Tectonics*, Wiley-Academy, West-Sussex, 2004, page 129.

directly correlated, as pointed out by Patrick Schumacher.[10] In terms of articulation, the pattern is aperiodic since all parts are different. However, the underlying structure can be seen as a distorted grid, which is entirely repetitive. It can be compared to the difference between a regular grid on a flat balloon, compared to a distorted grid that appears when the balloon is inflated. Another aspect is that calculations based on the local conditions of the surface with respect to orientation, lead to gradual change in the articulation and the accentuation of apertures, as Schumacher exemplifies. These variations then become a crucial element of the architectural expression, besides from providing an environmental response. Because the pattern formation is directly constructed from a subdivision of the underlying surface there exists a hierarchy where the overall shape of the building is prior to the pattern formation, that is, a top-down relationship.

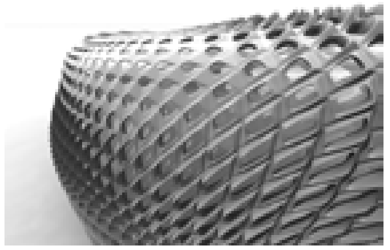
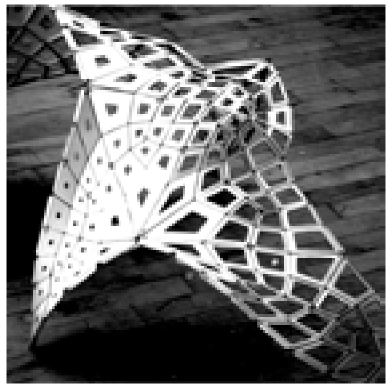


Figure 14: Zaha Hadid Architects, Civil Courts, Madrid, 2007

None of these architectural patterns could be denoted as self-organised or nonlinear, despite the fact that they display a large degree of variation within architectural expression. A hypothesis for the research in this thesis is that nonlinear properties are relevant to architectural design. An example of morphological self-organisation has previously been discussed in Chapter 2. With respect to self-organised pattern distribution, a series of tools has been developed. by Daniel Piker. In 2011 he lead a workshop at the Aarhus School of Architecture, where students tested some of these tools, which are accessible in the 3D modelling software Rhinoceros through Piker's Kangaroo plug-in. Figure 15 shows an example of a surface where an algorithmic tool has been used to generate a self-organised surface pattern, based on influences from predefined attractor points. The surface partition is achieved through random distribution of points on the surface. The points, or particles, distribute on the surface through use of repulsion and attraction, similarly to the distribution of circles shown in Figure 5. Eventually, the points define the subdivision of the surface into irregular homogenous mesh faces. The predefined attractor points are used to direct the articulation of individual components. In terms of nonlinearity, the main difference from the example in Figure 14 is that the subdivision of the surface is not a priori, but generated through an iterative negotiation process.

Figure 15: Digital Physical workshop at Aarhus School of Aarhus, cluster lead by Daniel Piker, 2011.



As part of teaching programme, led by the author and colleagues in 2010, a group of students at the Aarhus School of Architecture developed form-generating methods with inspiration from slime mould behaviour. The organisms' response to food resources in the environment was of particular interest. Parallel to the structure of slime as consisting of numerous nuclei, the group used a large number of components for a conclusive installation in

10 Patrick Schumacher, 'Parametric Patterns', *Architectural Design – Patterns of Architecture, Vol 79, No 6*, 2009

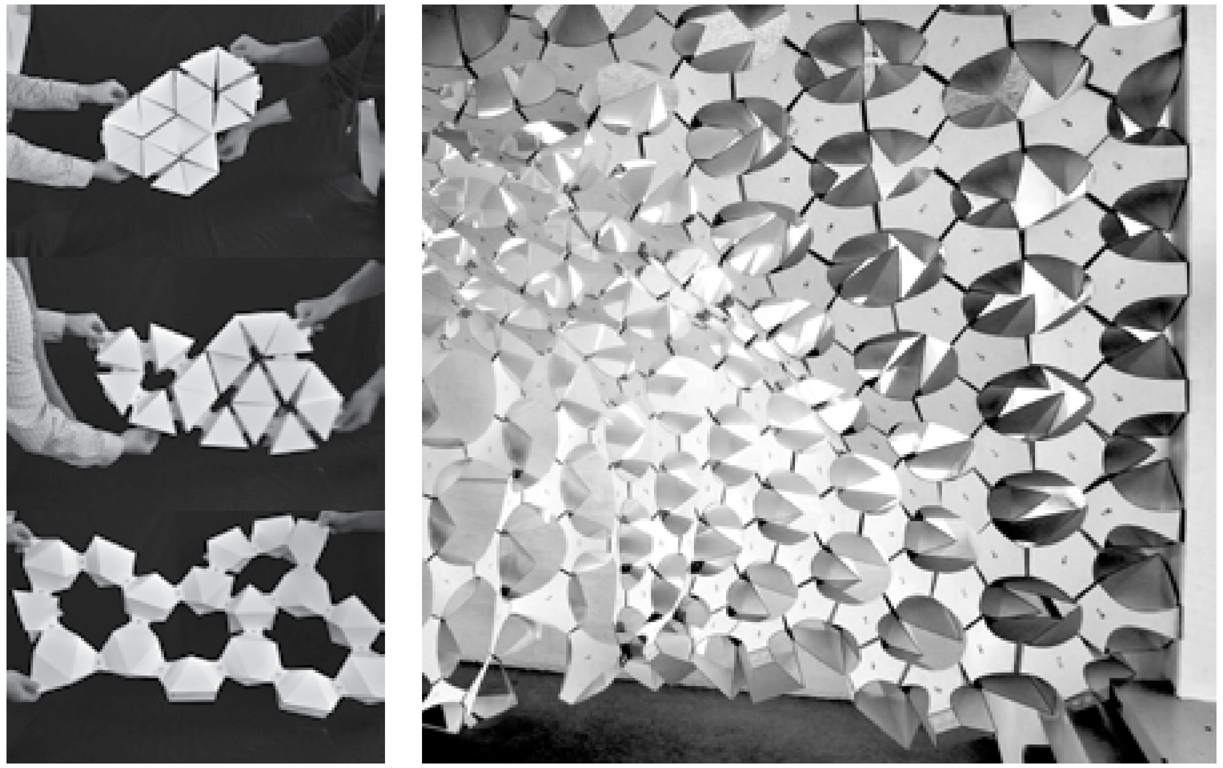
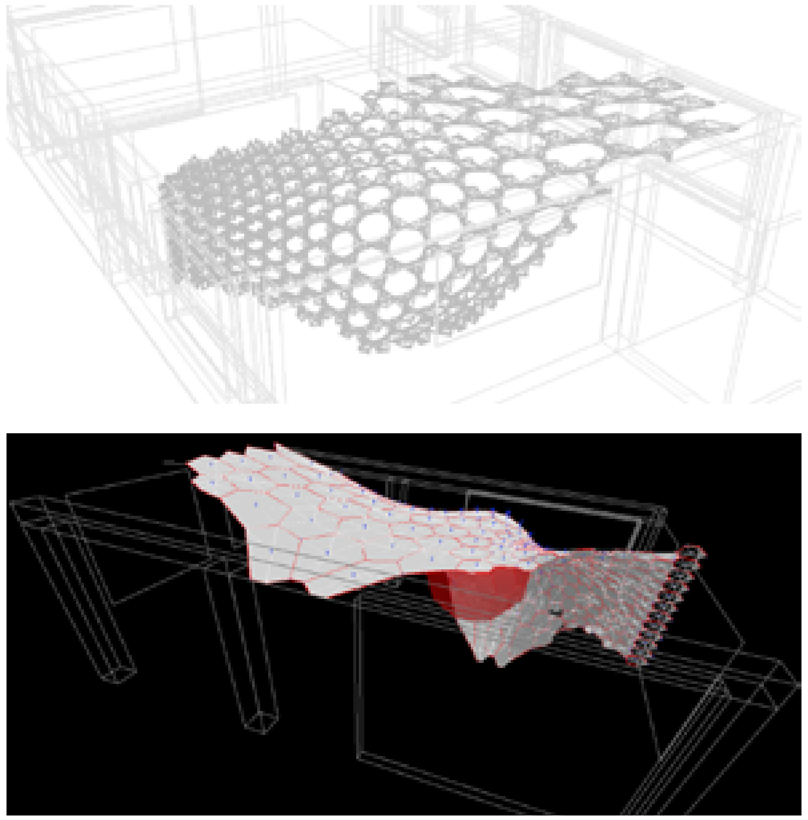


Figure 16: Student project inspired by slime mold behaviour. Above, left: stretchable components. Above, right: Installation in the studio space. Below: Spring based deformation. The red sphere represents an attractor point. Morphogenetic Studio, 2010. Aarhus School of Architecture. Anita Vårbro Berglund, Pelle Hviid, Mateusz Bartczak. Tutors: Sebastian Gmelin, Claus Peder Pedersen and the author.



the studio space. The top left corner of Figure 16 illustrates how a basic triangular component enabled the structure to expand and contract. The sides of the component that formed the connection points unfolded gradually in different angles, thereby responding to various forces affecting the structure. A stable structure was constructed as a simulation of a dynamic process, rather than a structure that was dynamic itself. This meant that initial ideas about forming the structure from a growth principle were discarded. Instead, the topology of the structure was predefined, but deformed and stretched over time through use of attractor points, as shown in the lower section of Figure 16. This form-generating method resulted in a structure that consists of discrete components. The rules embedded in the algorithm that control the deformation reflect the constraints necessary for producing the physical components and eventually assembling the structure. The components of the realised structure were individually laser cut and assembled, prior to constructing the large structure in the studio space. The tectonic principle allowed the structure to meet with planar surfaces, as shown in Figure 16, above. With respect to topology and pattern distribution, the realised structure was not self-organising. Rather, the morphology was self-organised and comparable to the dynamic relaxation examples observed in the previous chapter. In this case, a particle spring system was used to control the relations between the components and to arrive at an equilibrium state. A difference is, however, that gravity was not taken into account as part of the form-generating process. In this example, self-organisation is closely related to the mechanisms that form minimal surfaces, also mentioned in the previous chapter. The structure can be compared with a stretched elastic membrane. Because the relations between the elements that define the surface were predefined, the surface pattern did not exhibit self-organisation.

It has been noted that some pattern distribution methods are nonlinear, whereas, facade patterns are generally linear. This question of linearity is related to a degree of complexity versus necessity for consistency. Linearity is beneficial because there is direct control over the ordering of pattern elements. This helps to ensure that there is consistency between the pattern and overall geometry. Nonlinearity appears through iterative negotiations between the elements of the pattern, where the state of each part depends on the state of other elements and vice versa. This results in an indirect relation between controlling parameters and the resultant pattern. Consequently, this can present difficulties in terms of achieving coherence between the pattern and the overall geometry. However, as demonstrated by Daniel Piker and through *Self-organising Bezier Curves*, in cases where the morphology is complex, nonlinear methods can help to distribute a pattern evenly over the surface. More importantly, nonlinear methods allow complex negotiations between different parameters to affect the pattern formation. Through linear methods, the articulation of the pattern may depend on various parameters, as seen through the project for *Madrid Civil Courts (Figure 14)*. However, nonlinear methods allow the positioning and relations between pattern elements to be part of the negotiation. In other words, linearity and nonlinearity provide different degrees of adaptability. It should be said that with a nonlinear pattern it will often be necessary to refine the pattern, either through a high level of specification in the form-generating algorithm, subsequent translation to a more linear algorithm or even explicitly remodelling the pattern as seen through the *0-14 Tower* project. Through the subsequent proliferation of digital production techniques, it is becoming increasingly relevant to link the realisation with the form-generating process, as exemplified in the former example. In such context the need for 'manual' control is less evident. An example of this can be seen through the highly adaptable facade pattern, generated by the *Self-organising Bezier Curves* method, which is realised through the use of robotic laser cutting of foam templates for concrete casting. It is possible that the realisation of the *0-14 Tower* could have benefitted from a similar technique, even though these techniques have mainly been tested on smaller scales thus far.

3.5 Conclusive remarks

Although morphology plays an integral role in the examples illustrated in this chapter, the focus of this thesis is on strategies for distributing surface patterns as a precondition for tectonic solutions. In this sense, the topic is focused on the detail of the surface, rather than the morphology described in Chapter 2. As demonstrated with the last student project, these matters are intertwined. I have described some of the potential and the implications of self-organisation. At times topological constraints of the pattern enable self-organisation of the morphology. In other cases a self-organising pattern can be constrained to follow a surface, and eventually define the geometry of the surface in terms of realisation. I have examined various strategies for enabling adaptability in surface patterns, and have discussed various degrees of linearity in terms of controlling the geometry. Nonlinearity allows types of negotiations to be embedded in the articulation of the pattern, and can be a strategy for solving patterning of complex geometries. The discussion of self-organising patterns in this chapter addresses essentially topological organisation in a two-dimensional perspective. In the following chapter, the notion of topology and overall geometry is examined further.

4 Topological negotiations

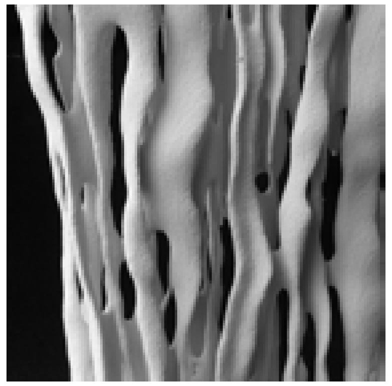


Figure 1: 3D printed plaster model of geometry generated with the method Branching Topologies.

4.1 Introduction

The previous chapters explored generative approaches to surface morphology and pattern distribution. These approaches can be said to have a two-dimensional character. The forming of a surface, as with the *Complex Gridshell* method, has of course a spatial dimension, but each node in the surface mesh is positioned in a two-dimensional topology. In the case of *Self-organising Bezier Curves*, the 'deep' patterns are directly related to a two-dimensional surface coordinate system, namely as UV-coordinates on NURBS surfaces. The discussion in this chapter considers ways of generating geometry with a complex topology in relation to the method *Branching Topologies*, which is described in detail in Chapter 8.3. The method was developed in order to create self-organised geometry with a topology that is not defined a priori. The functionality of the system is not completely three-dimensional, as the transformation from two to three dimensions happens in a linear way. This will be deliberated further throughout this chapter. This method is based on the previously discussed method *Self-organising Bezier Curves*. An isosurfacing tool was used for extracting volumetric geometry, and various studies and projects where isosurfacing has previously been used are discussed. Greg Lynn's writing on transections is incorporated with the discussion of *Branching Topologies*, since, in this method the geometry is established through combined transections. The experimentation of this method yielded a series of variations for possible outcomes, and the relation between variation and constraints is further discussed in the chapter. Moreover, the implications of using pre-defined algorithms as part of the form-generating process are discussed.

4.2 Bottom-up topology

The introduction established that the use of generative techniques leads to a shift in perspective with respect to the design process. The mathematics involved can be relatively simple, yet allows a large degree of complexity to emerge through numerous iterative calculations. Rather than explicitly forming the result, the attention is directed towards the rules that control the form-generating process. The two preceding chapters delineated how these techniques inform a large degree of geometrical complexity. Simultaneously, certain barriers for freely forming complex shapes are associated with the need to precisely define the underlying mathematics involved. As showed with *Complex Gridshell*, the morphology of a predefined surface can be manipulated by establishing negotiations between the control vertices. *Self-organising Bezier Curves* demonstrates how surface patterns can be distributed in a self-organised manner through similar local negotiations. In these cases, the geometrical complications are relatively simple. In terms of dynamic relaxation, it is mainly a question of implementing a mathematical formula for controlling the spring-based system of nodes and members. The mesh of nodes and members represents the generated surface shape. Additional geometrical challenges emerge when the mesh is translated into volumetric component geometry. However, the articulation of each detail of the geometry is not complicated in itself. With regards to the Bezier curves, the geometric control of pattern generation is based on a small number of trigonometric calculations, ensuring that control points and edge lines do not intersect.

When specifying geometry through use of advanced modelling software it is possible to 'manually' sculpt shapes that display a large degree of topological complexity. The modeller intuitively ensures that it remains consistent, i.e., the topology is not erroneous. When creating three-dimensional geometry with a generative system, it is necessary to specify the possible operations. If the geometry is constrained, for instance, to a three-dimensional lattice of equal cells, the amount of combinations is limited. Examples of this are discussed in Chapter 6, concerning *aggregate growth*. However, if the objective is to generate a non-Euclidean doubly curved and topologically complex geometry, as with the development of *Branching Topologies*, the geometrical complications are substantial. This can be seen through the problem of connecting volumes. If two existing volumes, or two different parts of a volume, are connected as part of the form-generating process, the procedure changes the topology. One approach would be to specify a procedure for the operation, another would be to analyse whether the conditions for the operation are valid in a geometric sense. Typically, the volumetric geometry is represented as meshes, which are defined as vertices, edges and faces. It is necessary to analyse whether all of the geometric entities are positioned in a way where the connecting procedure results in a consistent result. From the perspective of a designer who is not a mathematician, this large field of possible configurations is difficult to manage in terms of programming. Through direct 'manual' modelling of the geometry, the modeller ensures that the conditions for connecting the geometry are valid. Predefined algorithms assist in the process of performing the actual procedure. In essence, the issue of topological transformation entails a challenge for generative design.

4.3 Isosurfacing

One method for handling complex formations is to process the volumetric formation as a point cloud, and to evaluate the cloud in a single step. This means that the whole geometry is generated in a separate procedure, rather than being gradually built and transformed. A method for this approach is known as isosurfacing. *Branching Topologies* makes use of isosurfacing software, based on a *marching tetrahedrons algorithm*, as described by Graham Treece, who developed the specific tool with colleagues.[1] Similar methods for fluid dynamic simulations can be found in areas of science, engineering and animated films.[2] As this indicates, these calculations are sufficient to generate the volumetric geometry in real time, step-wise, as continuously updating frames. The simulation speed is naturally linked to the amount of data and hardware.

A weakness of this approach is that the form-generating process is only indirectly connected to the resultant geometry. The form-generating process directs how matter is distributed as volumes in space, but does not specify the precise points, edges and surfaces, because the secondary algorithm is confined to a certain logic. Furthermore, if the algorithm functions as a separate module or software, as seen through *Branching Topologies*, the designer cannot modify the way that the geometry is generated. Whether this is problematic or not depends on the actual design objective. For instance, if the generated result represents actual mass, as suggested with the column-wall examples in Chapter 8.3, it is perhaps sufficient that the geometry is constructed as homogenous mesh geometry. Here, the precise patterning is less crucial, as long as the form contains the correct degree of smoothness and precision. Another case is if the generated geometry is further processed to

1 G. M. Treece, R. W. Prager and A. H. Gee, 'Regularised marching tetrahedra: improved iso-surface extraction', Technical report CUED/F-INFENG/TR 333, Cambridge University Department of Engineering, September, 1998.

2 Mike Seymour, '*The Science of Fluid Sims*', http://www.fxguide.com/featured/the-science-of-fluid-sims, 2011

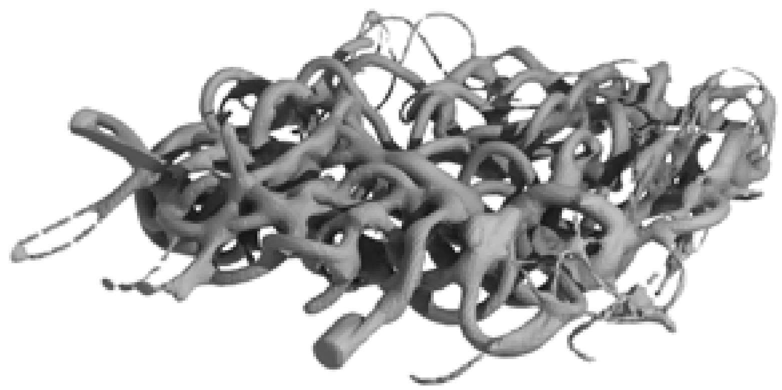


Figure 2: Student project, Morphogenetic Studio, Aarhus School of Architecture. Agent-based form-generation. Volumetric geometry through isosurfacing. Ragnar Zachariassen and Matheusz Bartczac. Tutor: Niels Martin Larsen.

form discrete building components, for instance. An example of this is the method *Complex Gridshell*, described in Chapter 8.1. Here, it is crucial to be able to directly address the mechanisms that define the mesh geometry. Otherwise, subsequent architectural detailing is confined by the geometry that the predefined module or software can produce. In reality, this would most often be unacceptable, and therefore lead to either the development of separate algorithms for extracting the necessary geometry, or to a form of 'manual' specification of the geometry. That is, the reconstruction of a three-dimensional model from the generated geometry. This dilemma is one of the incentives for developing the *Self-organising Surface* method, described in Chapter 8.4.9. The following section explores a range of experiments that utilise isosurfacing techniques.

In 2011, in relation to a programme led by the author, a group of students used an isosurfacing technique as part of the form-generating process. An agent-based algorithm aided the distribution of matter in the form of particles. Agent-based systems are further discussed in Chapter 5. The key algorithm was implemented in the programming language Processing, and isosurfacing was handled with the Toxiclibs module. Figure 2 displays an example of initial investigations where the movement of the agents is constrained to relate to a predefined grid. The effect is a form of weaving, but an irregular structure is generated because the constraint is loose. The thickness of the volumes varies between different agent types, which accounts for the thin strands. The agents' trails are saved as they gradually move through the field. At each time step or state, the entire volumetric geometry is re-generated with the isosurfacing procedure. Figure 3 displays a different study, where it is not the traces of the agents that are translated into volumes. Instead, the agents represent nodes in a structure. The agents are connected by lines of different thickness, which again behave like springs. These lines are then translated into volumetric geometry with the isosurfacing procedure. This principle was expanded through the

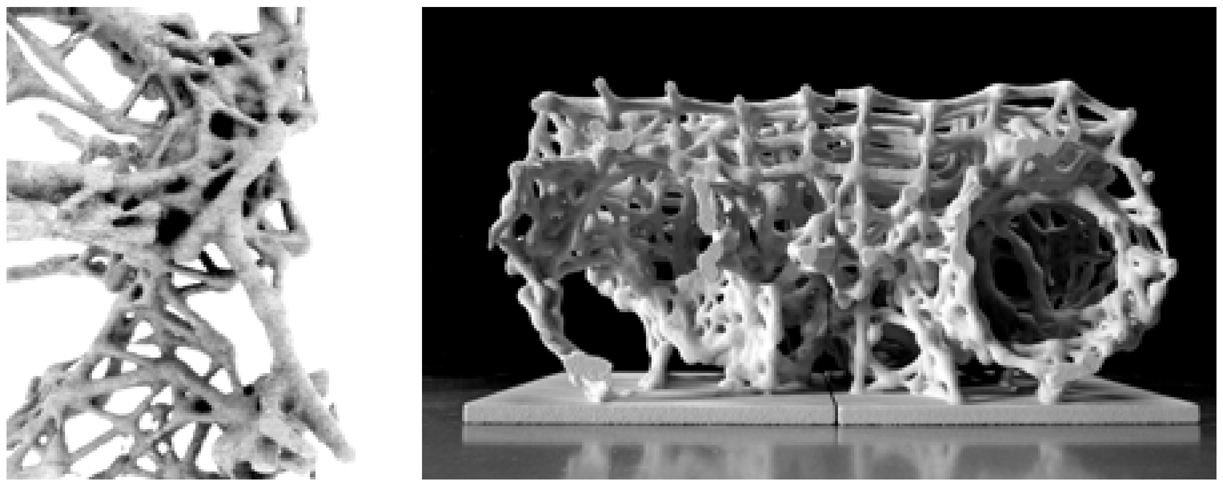


Figure 3: Student project, Morphogenetic Studio, Aarhus School of Architecture. Left: Experiment with spring systems and isosurfacing. Rendering of detail. Right: 3D printed plaster model of cross-section of structure generated with isosurfacing. Ragnar Zachariassen and Matheusz Bartczac. Tutor: Niels Martin Larsen.

project development, as the 3D printed section model in Figure 3 demonstrates. Here, the structure for a building was generated similarly to a spring system that is dynamically established during the form-generating process. The system adjusted according to attractors related to the building programme and the context. The geometry of the structure was again generated through the technique of isosurfacing. Both images in Figure 3 reveal the limitations of isosurfacing. The method is suitable for smooth curved volumes, as demonstrated with *Branching Topologies*, but less suitable for sharp edges and thin lines. This is because the geometry is created as an additional layer outside the geometry that directly represent the 'original' form-generating process.

The architectural office Kokkugia has developed methods for instantiating complex topology through the use of Isosurfacing as part of their research project *Swarm Matter*[3]. An example of this is shown in Figure 4. The research explores the possibilities for generating nonlinear hierarchies and emergent patterns. The underlying logic is an agent-based system, similar to those discussed in regards to additional Kokkugia projects in the following chapter. This example is relevant because of the use of isosurfacing techniques to generate volumetric geometry. The research focuses on organising matter with a bottom-up approach opposed to having an a priori topological hierarchy. As Roland Snooks states, 'The research project questions the contemporary understanding of component logic as elements which are subservient to a topological ordering device such as surface. Instead this exploration looks at the ability of the macro order to emerge from the interaction of components at a local level.'[4] The focus of this perspective is on mechanisms that allow the agents to self-organise and lead to temporarily emergent

3 Roland Snooks, 'Swarm Matter', 2009, viewed 12 April 2012,<*www.kokkugia.com*>

4 Snooks, *op.cit.*

Figure 4: Kokkugia. Swarm Matter. 2009. Roland Snooks and Pablo Kohan

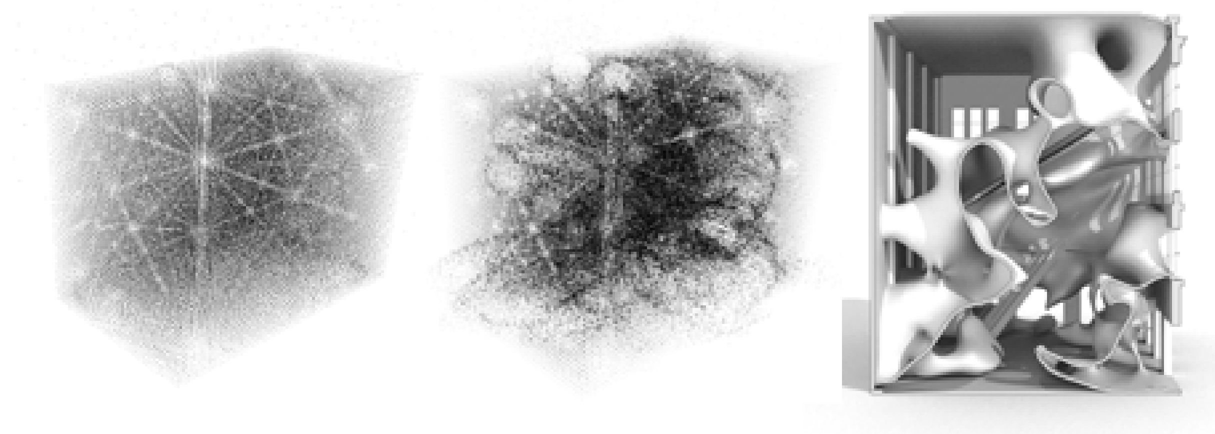
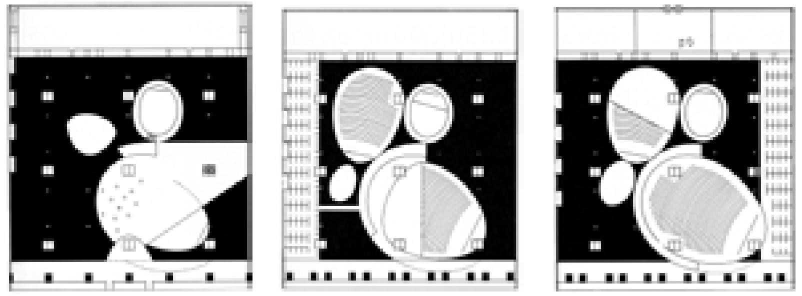


Figure 5: Kokkugia. Little Collins Baths, 2004. Rob Stuart Smith. Left: Initial particle field. Middle: Dispersed particles. Right: Cross-section through amorphous space division.

Figure 6: OMA, proposal for Bibliotheque Nationale de Paris, 1989. Floor plans of level - 4, -3 and -2.



hierarchies. As such, the goal is not to arrive at a solid topology in the same way as the method for *Branching Topologies* demonstrates. Rather, the emphasis is found in the complex formations that arise from the agent interactions. The isosurfacing technique remains crucial in terms of suggesting the existence of matter, rather than an entirely abstract geometry, such as a point cloud. With regards to *Swarm Matter*, there is a natural consistency between the logic of the system and the curved smoothness of the isosurface geometry. An architectural project by Kokkugia uses isosurfacing to divide an existing building into two intertwining but separated spatial zones. In the project for *Little Collins Baths* (see Figure 5), the topology is generated with a nonlinear dynamic system. A spring model of attractor points self-organise in the existing building volume from contextual and programmatic parameters. The movement of A particle cloud is simultaneously affected by the movement of the attractors, which leads to variations in particle density and the emergence of spatial formations. Figure 5 shows the existing building charged with a field of particles. The image in the centre shows the particle cloud after displacement through the nonlinear transformation process. Through the use of isosurfacing, it is possible to generate the surface that separates parts of the volumes where the particle density is high from the parts which are low. The geometry of the point cloud was extracted through a series of cross-sections in advance to the isosurfacing procedure.

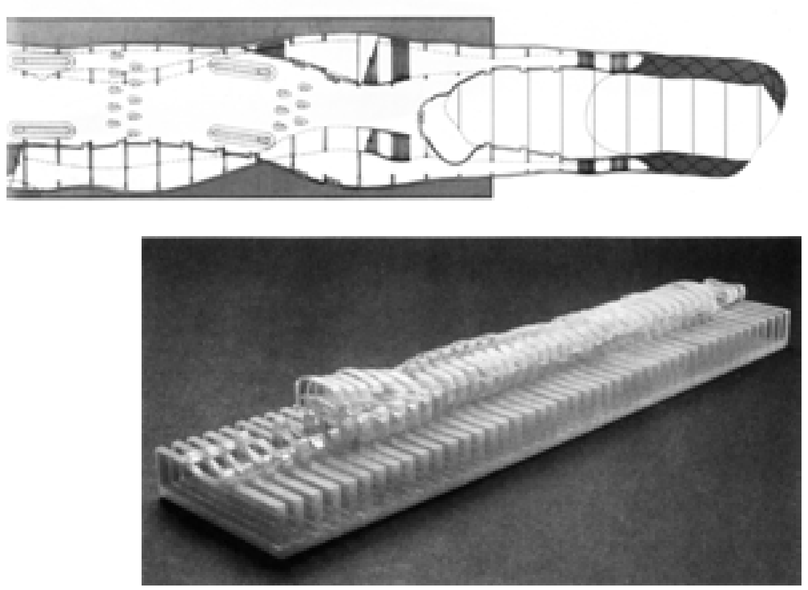


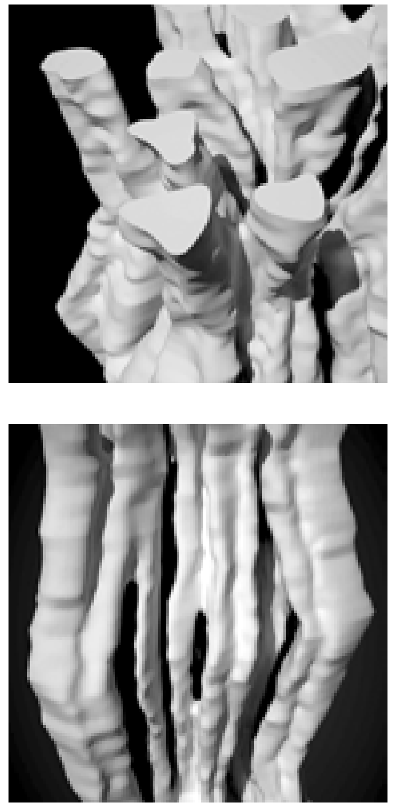
Figure 7: Greg Lynn, Yokohama Port Terminal, 1994. Uppermost: Laser-cut model of acrylic sheets. Lowest: Part of First-level plan.

4.4 Transections

The projects discussed in the final section of the previous chapter demonstrate formations that are three-dimensionally generated. *Branching Topologies* functions in a slightly different way. Rather than generating a point cloud in three-dimensional space, a series of cross-sections are generated. Figure 9 in Chapter 8.3 illustrates a dynamic self-organising process where shapes gradually transform, join and split. The formations take place in two-dimensional space, and only because the drawings are interpreted as a sequence of section drawings, a volumetric geometry is generated. The method is therefore linear in terms of spatial formation, even though the two-dimensional self-organisation is algorithmically nonlinear. Because of this constraint, the method can be said to be two-and-a-half-dimensional, compared with the previously mentioned projects where the form-generation was inherently three-dimensional. Although the spatial linearity of the method limits the functionality, these constraints can also be beneficial. First, I would like to expand upon using transections as an architectural approach. Greg Lynn stresses the topic in his article: *Probable Geometries: The Architecture of Writing in Bodies,* from 1993.[5] He discusses how transections are used in science to describe amorphous shapes. He states, 'In these stereometric examples, possible three-dimensional areas and shapes are projected from two-dimensional transections through a radical orthogonal technique that seems to be already natural to architecture.' Some of the architectural references are OMA's library projects from the period. In the proposal for the *Bibliothèque Nationale de Paris*, the floor plans appear as transections through amorphous shapes that contain spaces for individual purposes, similar to distinct organs in a human body (see Figure 6). In the 1990s, many of Greg Lynn's projects that were documented in his book *Animate Form*, the idea of using transections as a design approach is evident. An example of this is his proposal for Yokohama Port Terminal. An acrylic model of the project is displayed in Figure 7. Notably, the presentation model itself consists explicitly of transections. The organisation of the building structure also reflects this design logic. As seen in the plan drawing in Figure 7, stringent linear order in the structural lines that are placed as orthogonal section lines exists. As with OMA's

5 Greg Lynn, 'Probable geometries: The Architecture of writing in bodies', *Any 0*, 1993

Figure 8: Branching Topologies. Basic properties. Top: visible cross-section. Lowest: Branching topology showing that the 'branches' can merge.



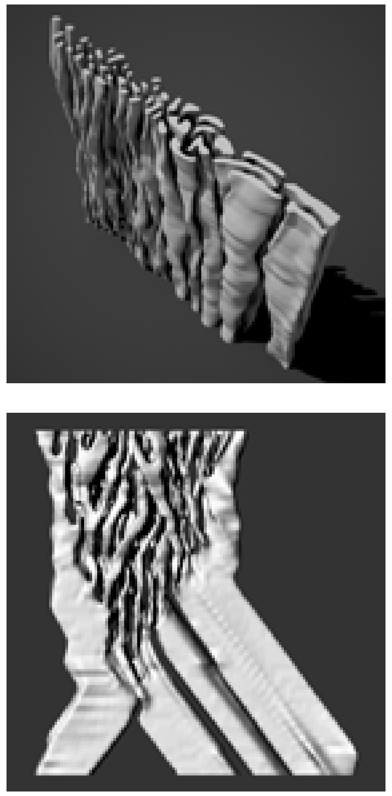


Figure 9: Branching Topologies. Graduation of the geometry in different ways. Top: The average cross-section area of each ‘trunk’ is initially defined. Above: The cross-sectional area decreases gradually as part of the process of generating the sections.

project, transections influence the project on numerous levels. They operate on a conceptual level, where the transections address the otherwise diffusely defined amorphous shapes. Another function is that the transections are directly part of the structure of the building. This again becomes part of the spatial experience by contrasting the amorphous shapes. As described in *Animate Form*, Lynn made use of animation tools as part of the design process. In this project, the primary role of transections is to describe, control and contrast the dynamic amorphous shapes. The method *Branching Topologies* can be seen as a continuation of this method for describing complex shapes through the use of transections. However, instead of ‘cutting through’ the shapes, the transections themselves create the amorphousness. The design intent determines whether transections may be utilised or not. For instance, Figure 10 illustrates a column wall where the transections have completely vanished, except perhaps from floor and ceiling. The high-rise building example, shown in Figure 10, illustrates transections that directly relate to floors in the structure. A relation between the architectural references and the aforementioned isosurfacing examples is evident. This is due to the fact that in all cases, the amorphous geometry is inscribed in a regular grid. Through the use of an isosurfacing technique, there must be a defined three-dimensional lattice that inscribes the generated data before the surfaces can be created. As such, the extent of the geometry is limited. This is an additional reason for developing alternative strategies for generating amorphous shapes, such as the agent-based systems discussed in Chapter 5.

4.5 Controllable behaviour

As previously discussed, the inherent logic of *Branching Topologies* poses a specific formal character on the outcome. However, within these limits it is possible to adjust and regulate the behaviour through subsequent adjustments. Examples of this are discussed in the following section. Two of the main properties of the method *Branching Topologies* are illustrated in Figure 8. The uppermost image indicates how the ‘trunks’ of the geometry are created from cross sections. The lowest image indicates that the volumes split and merge during the generation process. The cross-sections are ordered like layers in a vertical structure, where the first layer is positioned on the ground plane. This process is explained in further detail in Chapter 8.3. As the behaviour of the self-organising curves is only indirectly specified, effort has been made to investigate ways of controlling and varying the outcome of the form-generating process. Figure 9 shows two different ways of graduating the geometry. The uppermost image displays a situation where the area of the cross-section of each ‘trunk’ is gradually increased from one end of the

structure to the other. In this case, the graduation occurs in the first step where the curves are distributed, before generation of sections is initiated. The lower image of Figure 9 illustrates an example where the area of the cross-sections gradually decreases during the form-generating process. Initially, the area of the ‘trunk’ cross-sections is large at the bottom and gradually splits into many smaller ‘branches’. This way of varying the character of the geometry across the system is similar to the pattern variations with *Self-organising Bezier Curves*, described in Chapter 8.2. Variation occurs as a negotiation between the internal logic and various external parameters. This was demonstrated with the surface pattern where the orientation affected the character of the pattern. These negotiations allow the system to adapt to environmental conditions, thereby enhancing the functionality and simultaneously creating a more varied expression.

In continuation of the topic of variation, the issue of constraints arises once more. As previously stated, *Branching Topologies* is a constrained and linear method. The result has the potential to be a complex topology, but the generative process and the character of the result lies within a limited field. However, these limitations can also be seen as strengths. As explained in detail in Chapter 8.3, some of the constraints ensure, or at least support, that certain criteria are fulfilled. This is because the sequential logic of the form-generation ensures that each volumetric transection is directly continued from the previous transection. This is an advantage in terms of structural implications, since no cantilevered parts have to be considered. Another aspect of this method is that due to the fact that volumes are generated in layers, it is possible to control the cross-sectional area, and the volume size, during the form-generating process. If the geometry is representative of a building, it is relevant to consider a balancing of floor area and surface area. This is supported by the logic of the system. A series of examples concerning cross-sectional areas are shown in Figure 28 in Chapter 8.3.

As indicated previously, an important issue related to *Branching Topologies* is that there is a series of procedures connected to the form-generating process. Figure 3 in Chapter 8.3.1 illustrates this. The four final steps are connected to the use of isosurfacing in the project. A topic raised in this thesis is the possibility of embedding aspects concerning the final result directly in the form-generating process. When the geometry is translated during the process, it is impossible to address these aspects in detail from the beginning of the process. It is merely possible to further process the generated geometry, for instance, transform the typically triangular geometry into other formats, or generate the connection detail. A possibility would be to implement a bespoke version of isosurfacing. However, this would increase the complexity

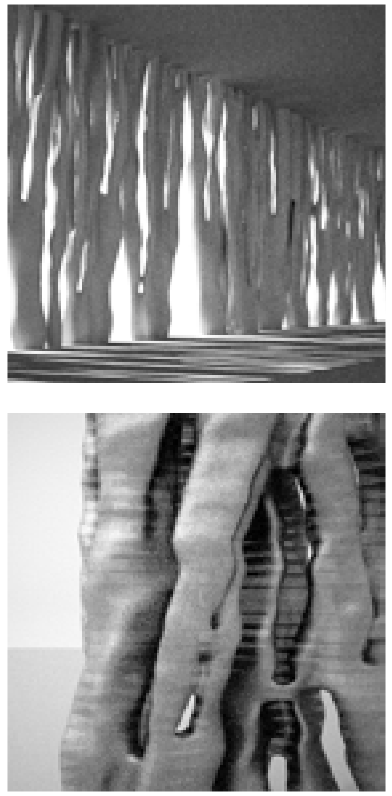


Figure 10: Branching Topologies. Top: Column wall. Above: High rise building experiment.

of the method and the fundamental logic of the algorithm would continue to pose constraints on the outcome. Another approach could be to encompass the isosurfacing algorithm. For instance, if the generated cross-sections are directly exported to a 3D modelling environment, and scripts are developed to interpret the curves into more detailed structures and components, a surface may be redundant. Situations where the geometries merge or split would appear as geometric challenges in this case.

4.6 Conclusion

Branching Topologies has been discussed as an example of a method that uses two-dimensional self-organisation to arrive at geometry without predefined topology. As such, this method and the related discussion connect a two-dimensional and three-dimensional approach. Although this method has a linear property, it is possible to generate highly complex geometry in terms of architectural design in general. It is relatively constrained, compared with the agent-based methods that are discussed in the following chapter. Simultaneously, these constraints ensure that the outcome has certain properties. This also relates to formal character, volumetric extents and topological consistency. It is controllable and allows a degree of variation. However, since the outcome is so specialised, it would not be relevant to many projects. As such, it may be considered as a demonstration of how the aforementioned properties can be achieved through the development of a generative technique. Due to its constraints, if the method is relevant to a specific task it would be much more successful than the more generic methods mentioned earlier in the chapter. The development of the method fulfilled the objective of transcending the need for a priori topology. The possibilities for controlling cross-sectional area, and indirectly surface area, are properties that are highly relevant in terms of use in actual projects. The method relates back to Greg Lynn's discussion of transections, and re-formulates its potential in generative design. Rather than playing the role of a skeleton that holds and frames amorphous shapes, the amorphous shapes are generated directly from the transections.

5 Emergent hierarchies

5.1 Introduction

Different approaches to generating architectural geometry have been discussed in the previous chapters. Most of these methods have a nonlinear character and are to some extent self-organising. In the *Complex Gridshell* method, the morphology is found through Form Finding. *Self-organising Bezier Curves* generates a self-organised pattern, and *Branching Topologies* enables creation of complex topologies through a nonlinear process, which is also constrained by a linear structuring of the geometry. The following section explores self-organisation further in terms of spatial complexity. This chapter will observe surface geometry or linear geometrical organisation methods that are fundamentally three-dimensional. Agent-based systems will be discussed in relation to experimental methods, described in detail in Chapter 8.4. In general, agent based systems do not adhere to spatial dimensions, and if they do, they are often addressing a two-dimensional space. However, the agent-based systems developed through this research are inherently three-dimensional. In this sense, these methods are less constrained than the methods mentioned previously. Although, some of the experiments and developed behaviours constrain the movements of the agents to two dimensions or follow a predefined surface. The difference here is that three-dimensionality is embedded in the logic, and allows the behaviour of the agents to respond to three-dimensional aspects, as well as generating three-dimensional results.

In this chapter agent-based systems will first be discussed in general. A series of examples form a basis for discussing the use of agent-based systems in architecture. The extent of which the methods are self-organising and display emergent properties, as well as the implications of using nonlinear systems for architecture are discussed. Certain emergent effects and behaviours are explored in order to unfold the general properties of agent-based systems. Finally, examples are used to cement the potential of agent-based systems in architectural design.

5.2 Agent based systems

Agent-based systems consist of numerous autonomous entities that interact through a basic set of rules on a local level, thereby generating global behaviours. Agent-based computer models are used for the simulation of a variety of systems, such as swarm behaviour within social insects, stock markets[1], human crowds[2] and artificial intelligence.[3] Digital computation allows models consisting of a large number of both agents and iterations, thereby enhancing the possible complexity of the system. A large quantity of agents and their interaction over a number of time frames is necessary for generating emergent effects, which is often a main property of the systems. Different biological systems have been simulated with agent-based models.[4] The method described in Chapter 8.4.2, was developed from an agent-based algorithm, originally designed for simulating flocking behaviour among birds. Other examples for its use are bacterial colonies[5], ant societies and fish schools. The philosophy behind agent-based systems is that complex behaviour can be generated by establishing multiple iterative interactions within a system, based on a set of simple rules. As such, agent based systems are concerned with the mechanisms for self-organisation, and what types of rule sets lead to specific behaviours. Two main characteristics that the systems have in common are that they consist of a large number of individual entities that act independently, and that they display a type of emergence. Emergence can be understood as the appearance of order on one level, generated from entities that self-organise on a lower level. This type of order is not the result of explicit top-down control, since the entities are guided by internal properties and interaction with each other and their surroundings. This is similar to the way cells in an organism behave and form a well-functioning living system like a plant or an animal, or how termite behaviour is directly influenced by pheromones. In nature, emergence also appears in cases that would normally not be said to be agent-based. Examples of patterns occurring from material properties are cloud formations, sand dunes, dendrites and soap bubbles. The difference is that the entities in these systems lack the capability of 'acting' and are subject to external forces. The entities in these systems are considered passive, since they lack

1 W. Brian Arthur, John H. Holland, Blake LeBaron, Richard Palmer, and Paul Tayler. '*Asset pricing under endogenous expectations in an artificial stock market', The economy as an evolving, complex system II*, Addison Wesley, Redwood City, 1997, pages 15-44.

2 M. Moussaïd, D. Helbing and G. Theraulaz, '*How simple rules determine pedestrian behaviour and crowd disasters*', Proceedings of the National Academy of Sciences, Volume 108, No. 17, 2011, pages 6884-8.

3 Gerhard Weiss, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 1999.

4 Bonabeau, Camazine, Deneubourg, Franks, Sneyd, Theraulaz. *Self-Organisation in Biological Systems*. Princeton University Press, USA, 2001.

5 Lees, Michael, Brian Logan, and John King, '*HLA Simulation of Agent-Based Bacterial Models', Proceedings of the 2007 European Simulation Interoperability Workshop*

autonomous aims such as survival, which inevitably control their behaviour. The entities in these systems are considered passive, since they lack autonomous aims, such as survival, to control their behaviour. Agent-based systems occasionally operate with such entities, and are subsequently addressed to as 'passive agents'. Typically, a system dealing only with passive agents is called a particle system. The dynamic relaxation algorithm, described in Chapter 8.1, demonstrates a type of particle/spring system that is also a self-organising system.

Agent-based systems provide a versatile basis for implementing a variety of self-organisation methods in architectural design. As previously stated, there are different methods for producing self-organisational patterns and formations. An example of this is the *cellular automaton*, first developed by John Von Neumann and Stanislaw Ulam[6]. Methods that use *cellular automata* display a range of different self-organised patterns, and the agent-based systems are to some extent developed from *cellular automata*. With respect to architectural design, a key difference between agent-based systems and *cellular automata* is concerned with the relation between the internal logic of the algorithm and the generated result. It should be mentioned that *cellular automata* by nature are very generic and can be adjusted to produce a wide scope of outcomes. However, the inherent logic of agent-based systems suggests an approach where the relations between the entities are more negotiable, and not necessarily a priori to the form generating process. The cells in a *cellular automata*, which can be compared to the agents in agent-based systems, are usually organised in a grid, rendering the organisational relations constant. The relations in agent-based systems can change continuously, as well as agents can be removed or added to the grid. In this way, agent-based systems suggest a much more situation-dependent relational configuration. This is particularly relevant when addressing complex topologies, which will be demonstrated in the examples of the following section. Another relevant aspect of agent-based systems, compared to systems dealing with passive entities, is the possibility of defining rule sets for the agents, that reflect design intents or properties that relate to materials, manufacturing or contextual influences. Because the agents dynamically negotiate their spatial positions and general states through interactions, local negotiations of a series of parameters can be established. As part of the logic of the system, their behaviours can be balanced or weighed in order to give each parameter the desired influence on the system.

In order to extend the discussion of agent-based systems

6 Wolfram, Stephen. A New Kind of Science. (USA: Wolfram Media Inc, 2002), page 28.

further, it is relevant to refer to John Holland's description of complex adaptive systems in his book, 'Hidden Order: How Adaptation Builds Complexity.'[7] Holland outlines topics of key interest with respect to agent-based systems in general. The basic principles described by Holland form an interesting starting point for understanding and describing agent-based systems. According to Holland, the agent's behaviour is determined by a collection of rules. This is typically a stimulus-response rule: 'IF stimulus s occurs THEN give response r.' The system's performance is then the result of a succession of stimulus-response events. Holland discusses some of the properties and mechanisms relevant to agent-based systems, such as aggregation [*emergence*],[8] *nonlinearity, flow, diversity* and *tagging*. *Emergence* refers to the ability of the agents in the system to form 'complex large-scale behaviours form the aggregate interactions of less complex agents.'[9] *Nonlinearity* addresses the fact that there is not a direct way of predicting the outcome of the system through an analysis of its logic and environment. Mathematically, the variables are multiplied and lead to complex results. *Flow* denotes that the behaviour of the system is created through a process where agents continuously exchange information or matter. *Diversity* concerns the existence of a variety of agents in a system, whereas, *tagging* is a mechanism that enables this differentiation within the system. Tagging allows the construction of a system of initially generic agents that are then tagged in order to play different roles in the systems behaviour.

In the following section, some of the basic properties of agent-based systems will be used as a starting point for discussing various research methods and references. The examples are influenced by parts of the research project as well as architectural projects, where agent-based systems have been used. The focus of the following section is directed towards the ways in which the entities of the system are constructed, how they operate, and how they are able to establish emergent effect, i.e., order on a higher level than the level they are operating at.

7 John Holland, Hidden order: how adaption builds complexity, Helix Books, New York, 1995.

8 In Hidden order, John Holland reflects on emergence through discussion of aggregation. The topic of emergence is broadened in this following book on the subject: see note 6.

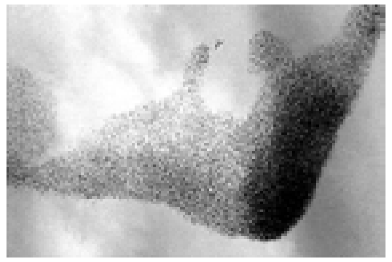
9 Op.cit. Holland, 1995.

5.3 Self-organisation and emergence

In this section, the topics of self-organisation and emergence are discussed in relation to agent-based systems. As the following examples reflects, emergence is intertwined with self-organisation. According to the definition provided in the introduction, it is possible to develop a self-organising system without emergent properties. At the same time, the possibility of generating emergent properties through the use of dynamic systems holds great potential for architectural design. Architects constantly negotiate problems that are not entirely different from generating emergence. Architecture is primarily concerned with generating a wholeness that represents order on a higher level than the materials and components it consists of. A difference exists, however, as this form of wholeness is not the result of a dynamic process. The individual parts do not negotiate in order to find their final state, as with self-organising systems. In fact, there are types of self-organisation on a material level, for instance with fluid concrete or pneumatic constructions.

Perhaps it could be said that self-organisation is the process of achieving emergence, thereby directly linking the terms. This statement particularly holds for dynamic systems, such as agent-based systems, even though, it is possible to discuss emergence in relation to static systems.[10] John Holland sees emergence as the appearance of a recognisable pattern, which also is recurring within the system.[11] The order is not random, but appears as a consequence of the internal logic of the system. An example that Holland puts forward is the maintenance of an ant-nest colony, where ants are seen as having 'emergent behaviour far beyond their individual capacities.' [12] Another well-studied emergent effect is bird flocks. A noted example is the flocks of Starlings, which form large murmurations in the vicinity of their roosts in Spring in southwest Denmark (see Figure 1). The birds may be considered as agents, guided by a simple set of rules. Without hierarchical control, they are able to form complex formations. The forming and behaviour of the flock is seen as an emergent property, as flocking assists the birds in different ways, such as to avoid predators and to forage. An agent-based model for simulating flocking behaviour was published by

Figure 1. Murmuration of starlings.
Photo: Ben Van Buren.



10 Tom De Wolf and Tom Holvoet, Emergence versus self-organisation: Different concepts but promising when combined, Engineering SelfOrganising Systems, Volume: 3464, Issue: 1675, Springer, 2005. Pages: 1-15

11 John Holland, Emergence: From Chaos to Order. Oxford University Press, Oxford, 1998, page 4.

12 Holland, *op. cit.*, page 5.

Figure 2. Mapping of flocking behaviour. Rendering: Morten Bülow

Craig Reynolds in 1987.[13] His method for implementing an agent-based system has been used as a generic model for agent-based systems, particularly the simulation of flocking behaviour and swarm formations. In 2009 alongside architect Morten Bülow, the author developed a method for generating geometries through the use of algorithms based on Craig Reynolds principles for flocking behaviour. Figure 2 illustrates a rendering produced with the method. The method development was initiated at the workshop: *Complex Formations*. The workshop was arranged by the research centre CITA at the Royal Academy of Fine Arts, Copenhagen, and led by architect Roland Snooks. The project was successful as a method for actually 'drawing' volumetric geometry in three-dimensional space was developed. The birds functioned as a drawing device, and the trajectories of their movements were the lines in space. A set of basic behavioural rules was expanded with rules that helped to guide the birds, or agents, in relation to their physical surrounding. For instance, they would avoid collision with walls, or search for specific locations in space. Furthermore, a secondary algorithm was developed for merging the trajectories into continuous structures. Two properties of the system are of significant importance. One was elegance, clearly reflecting bird movement that characterised the generated geometry. This was achieved by the vector-based calculation of the change in position from state to state. The other property was the large field of possible variations from a single start condition. By using a random generator, the formation could be formed differently in every simulation. The number, position and start direction of birds could also be varied. Even though the formations expressed complexity, it is questionable whether the system showed emergent behaviour. This due to the fact that the generated geometry displays a complexity that exists approximately at the same level as the exact movement of the birds. Both Craig Reynolds method and this method are described in greater detail in Chapter 8.4.2. Consequently, the agent-based system was developed into a system for generating surfaces with a completely bottom-up method, which is further described in Chapter 8.4.9. The objective here was to overcome one of the fundamental challenges within geometrical self-organisation in order to apply the method to architectural design, that is, to establish surface topology from a bottom-up approach. Within most methods, an a priori topology is defined and self-organisation mainly addresses the shaping of the surface. Other methods allow a particle cloud to self-organise into defined shapes, indicating a surface. A secondary algorithm is usually necessary to establish the actual surface. The self-organised geometry, typically a point cloud, only indirectly represents the surface. The final surface is accurately defined through the use of this secondary algorithm, which could potentially be an isosurfacing technique. An example of this is the previously discussed method Branching Topologies. Here, the self-organisational process does not produce a point cloud, but a series of sections, which are translated through a process of isosurfacing. The challenge concerning agent-based systems is reflected in a remark by Cecil Balmond, 'I would like to think that in time a swarm can create surface topology, but it can't until it can create membrane, because that is the source of all topology.'[14] Within the method *Self-organising Surface*, the agents are not mapped as seen in the previous example. Instead, they are equipped with behavioural rules that guide them towards forming a surface. They move from completely random positions, seek each other, and begin to negotiate factors such as distance between agents and their relative angles. These negotiations take place locally, though in some versions, the agents may individually affect global values that would then affect the whole system. Figure 3 demonstrates an example where the agents respond to a frame of attractor points that help to guide the formation of the surface. Without the guiding points, the agents will normally form an almost planar surface. The formation of a surface from agents representing points is an emergent effect, since the surface represents order on a global level compared to the original condition, which is completely disorganised. The example of a T-formed branching surface is the result of form-generation being strictly guided by 3 x 3 rings of precisely positioned attractor points. The surface demonstrates emergence, since the topological relations that define the surface and its shape, were initially undefined. The outcome is in this case predictable, since the process is so carefully controlled. John Holland stresses the issue of surprise in relation to emergence, and states, 'I do not look upon surprise as an essential element in staking out the territory.' [15] Hence, unpredictably is not a crucial parameter in terms of detecting whether a phenomenon is emergent or not. Another issue is whether the method is optimal for solving a particular problem if the outcome is predictable. However, this particular example was constructed in order to trial various possibilities for controlling surface formation. The main property that differentiates the method from similar methods is that the relations between the agents are unspecified in advance. Usually, when

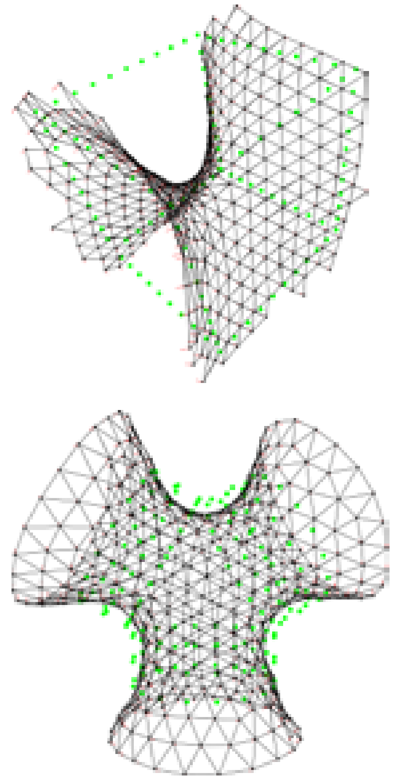


Figure 3. Method for generating a self-organised surface. Both the topology and the exact geometry is formed from initially randomly organised agents.

13 Craig W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model, Published in Computer Graphics, 21(4), July 1987, pp. 25-34. (ACM SIGGRAPH '87 Conference Proceedings, Anaheim, California, July 1987.)

14 Balmond, C, 'Informal Agency', in Leach, N, and R Snooks (eds), *Swarm intelligence: architectures of multi-agent systems*, Liaoning Science and Technology Publishing House, Liaoning, 2010, page 121

15 Holland, *op. cit.* Holland, 1998, page 5.

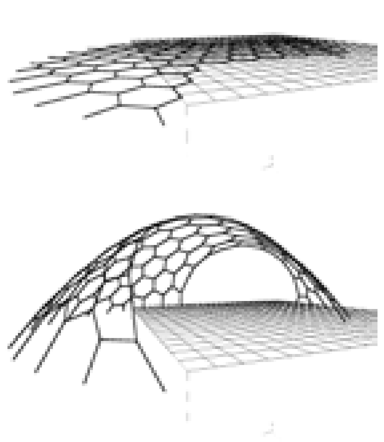


Figure 4. The structural property of the result of a dynamic relaxation can be considered to have emerged from the self-organising process.

working with geometries constructed from spring systems (or similar systems), the nodes and their connections are predefined in order to ensure geometric consistency. This does not mean that these systems cannot display emergent effects, since the precise form may display emergent properties that are not current within the initial conditions. An example of this is the method *Complex Gridshell* that has already been discussed, and will be described in greater detail in Chapter 8.1. Here, the self-organisation is linked to solving a very specific problem, namely the shaping of the shell form, as shown in Figure 4. Because the shape informs the curved grid with structural properties, the system can be seen as displaying emergent behaviour. In this case, the nodes can be compared with agents, even though they are hardly equipped with any sort of individuality or autonomy. This aspect has great meaning when different types of self-organisation in natural science are concerned. This difference exists mainly between physical and biological systems. Physical systems, such as the forming of sand dunes, waves, or clouds, can essentially be described through physical laws, affecting the particles. In biological systems, the complexity of the living components is much higher, as seen through ant colonies, neurons and bacteria. Here, the ability to develop new behaviours and refine the interaction between the agents through natural selection is a crucial aspect.[16] In terms of computer based simulation, when the aspect of evolution is not embedded in the system, for instance through the use of a genetic algorithm, the difference between the two types of systems is not as definitive. In terms of generative techniques for architectural design, it is important to clearly define systems that directly represent physical forces and those that do not. The latter category may comprise of pattern-generating rules constructed with respect to specific design intents. A property that is often mentioned in relation to the use of digital simulation of physical systems for architectural design is the possibility of merging physical rules with other types of behavioural rules. The system would then generate outcomes where both design intentions and physical realities could be constructed. Concerning the divergence between living and physical self-organisation, Rachel Armstrong has undertaken studies concerning the use of *proto-cells* with respect to the production of entirely new types of construction materials. Here, agent behaviour can be detected within cells that are considered to be in a pro-life condition.[17] Although the cells do not contain DNA, they are still capable of displaying certain types of behaviour, normally related to agency. Interaction through attraction and

16 Bonabeau et. al., *op. cit.*, 491.

17 Rachel Armstrong, 'Soil and Protoplasm', *Architectural Design vol. 81, no. 2*, 2011.

repulsion takes place, and as a result produces skin structures. Although these processes occur on a microscopic scale, they can be compared to the method for generating a self-organised surface. The research concerning using *proto-cells* for the production of new types of matter is predominantly concerned with chemical processes, and is thereby very different from the issue of controlling architectural geometry through the use of computation. However, it is interesting to note that some types of agency can be discussed in relation to physical systems, thereby challenging the sharp distinction between physical and biological systems with respect to self-organisation.

Another example of spatial self-organisation can be seen in a series of studies undertaken by a group of students in 2011 in the Morphogenetic Studio at the Aarhus School of Architecture. They developed a design strategy, partly based on a particle spring system, similar to the logic used for dynamic relaxation. The system was constructed in a way where the initial relations between particles, or agents, were undefined, which is similar to the method *Self-organising Surface*. The objective was not to arrive at a defined surface as with the previous examples, but rather to let the formation of self-organising agents create a complex spatial structure. The system is initialised with a set of nodes, where some serve as fixed anchor points. Similarly to the method of dynamic relaxation, the nodes are subjected to gravity and begin to move. An agent velocity exceeding a certain threshold activates a process of establishing connections, in the form of springs, to the neighbouring agents. Eventually, the system arrives at a stable state, similar to dynamic relaxation. Although the generated geometry does not represent a structure in pure tension, a level of structural coherence is achieved. Figure 5 shows two formations that have been generated with the method. The images do not represent different stages of the same process, but demonstrate the final equilibrium states achieved through two different sets of parameters for the spring system. This experiment can be seen as hybridised, as it consists of physical simulation and form generation, based on design intents. In regards to the development of a proposal for a building project situated in Barcelona, contextual parameters in the form of movement patterns and spatial organisation were embedded in the form generating process. An outcome of the generative process is shown in Figure 6. The system displayed self-organisational behaviour, due to the creation of complex structures through a simple grid of points. Still, it could be discussed whether the effect is emergent. The result was far from consistently organised, and therefore, it did not represent order on a different level to the starting point.

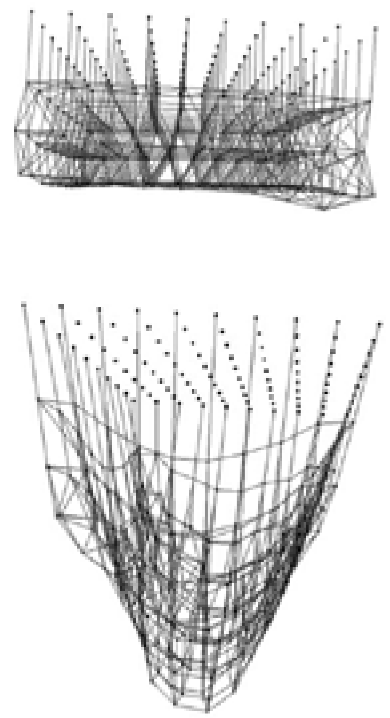
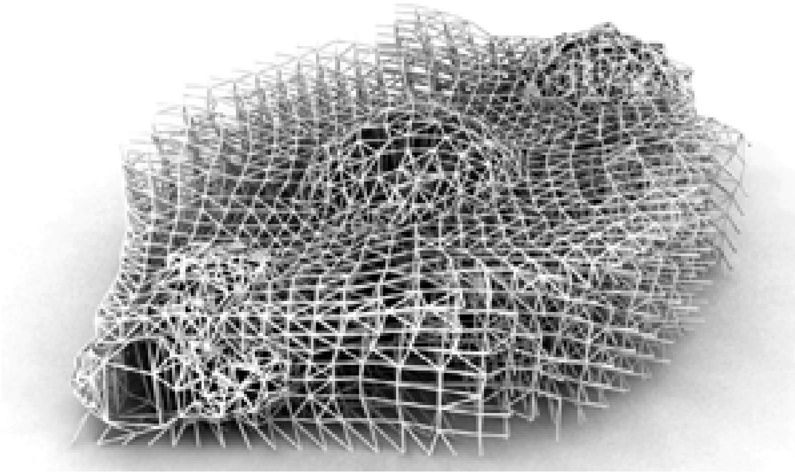


Figure 5. Hybrid method for use of a particle spring system. Two different spring settings. Mateusz Bartzak and Ragnar Zachariassen, Morphogenetic Studio, Aarhus, 2011.

An example of a method that is less constrained than the method for dynamic relaxation but more goal-directed than the previously mentioned method for generating a self-organised

Figure 6. Hybrid method for use of a particle spring system. An outcome of the generative process. Mateusz Bartzak and Ragnar Zachariassen, Morphogenetic Studio, Aarhus, 2011.



surface, is described in a paper by Vlad Tenu. This method is concerned with the generation of periodic minimal surfaces, and uses an agent-based approach to construct an initial region of the surface. This can then be reflected to establish the whole surface, as seen in Figure 7. The surface is subdivided through the use of Delaunay triangulation and a spring system is used to optimise the shape of the surface. One of the method's strengths is that it arrives at a completely resolved and optimised geometry, which is a major advantage with respect to realisation. In comparison to the method for generating a self-organised surface, Vlad Tenu's method is more constrained and less open-ended, since the overall organisation of the topology is predefined, despite the geometry being self-organised. It is directed towards periodic minimal surfaces, and confined to solving this type of problem.[18]

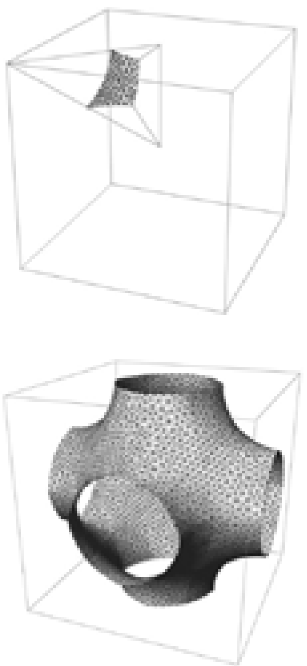


Figure 7. Method for constructing a periodic minimal surface. Illustration/ project: Vlad Tenu.

Another example of self-organised morphology is depicted in Figure 8, the 'Lamella Flock' project by Martin Tamke, Jacob Riiber and Hauke Jungjohann. It demonstrates an interesting agent-based method for generating a Zollinger structure. Rather than initiating the system from abstract points, the agents represent building components, consisting of four connected wood members. The members can adjust the length, relative angle and position while the whole component is able to scale. The emergent effect derives primarily from the fact that the shape of the structure is the result of complex negotiations between the individual parts, comparable to the logic of dynamic relaxation. The topology is not emergent as such, since the relations between the agents are predefined. In this sense, the method is self-organising to a smaller degree than *Self-organising Surface*, described earlier. However, the system can re-organise itself when disturbed, for instance, if the designer 'manually' re-positions individual components. Furthermore, the system is based on a specific structural principle, namely the Zollinger system, and has constraints regarding manufacturing and construction. As such, it shows a method for combining self-organisational properties with actual construction.

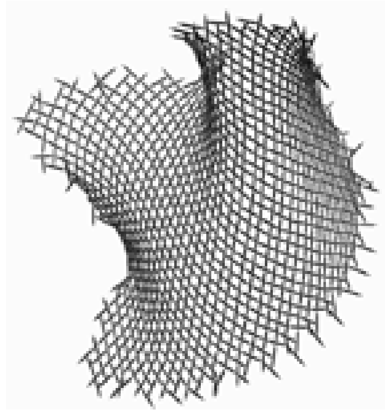


Figure 8. Lamella Flock. Martin Tamke, Jacob Riiber and Hauke Jungjohann. Illustration: Jacob Riiber.

Within some methods, the spring-controlled surfaces can self-organise and form new topologies. In these cases the generated geometry can possess a considerably higher level of complexity than the initial starting point. An advantage of this approach is that the transformed geometry is controlled during the self-organisation process. Because the initial geometry is consistent, and because the transformations are constructed to only result in consistent geometry, the final result is equally consistent despite an increase in complexity. Whether the formations correspond with the design intents and other parameters is of course a different matter. An example of this approach can be seen in the method *BodyToplogic*, developed by Kokkugia. Here, an a priori surface is defined as a swarm of autonomous agents. The agents re-position and interact through numerous vector-based negotiations, which then lead to an 'intensive topological formation through controllable and manageable self-organisation that is not bound to its starting topology or the quantity and configuration of constituent agents.'[19] As indicated earlier, a spring system is used to ensure that the topological relations remain intact during the transformation. Material and structural properties, as well as general design intents, can also be embedded in the code. This controls the agents' behaviour. *BodyTopologic* is demonstrates a versatile method for using an agent-based system to generate complex topologies as part of an architectural design process. An example of a project that makes use of this is the proposal for the *Busan Opera House* in South Korea. Figure 9 illustrates the initial spring controlled geometry. A number of agents have been tagged, or programmed, to perform certain interactions during the transformation process. On the right side of the diagram, the final formation of nodes and their connections define the surface of the building. In this way, a screenplay for the process was planned in advance in order to control the formation to some extent. The method allowed separate parts of the project to be post-processed without loosing the integration with the fixed parts. In this way, the project was developed partly as a bottom-up self-organised process, and partly as a more controlled design process, similar to architectural design methods in general. A more research oriented example that displays the potential of *BodyTopologic* more clearly, is the *BodySwarm* project, shown in Figure 10. Here, the

18 Vlad Tenu, 'Minimal Surfaces as Self-organizing Systems', *Proceedings ACADIA 2010*, New york

19 Robert Stuart-Smith, 'Formation and Polyvalence: The Self-Organisation of Architectural Matter', *Proceedings, Ambience 11*, Borås, Sweden, 2011

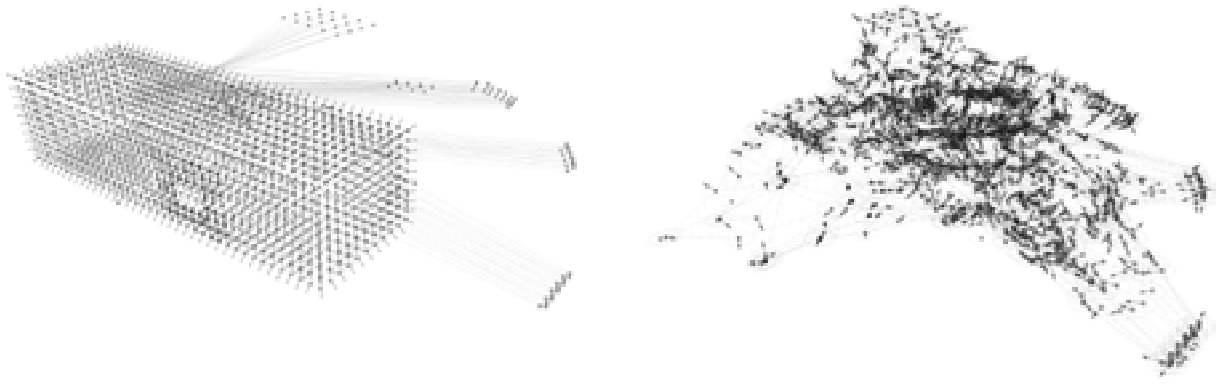
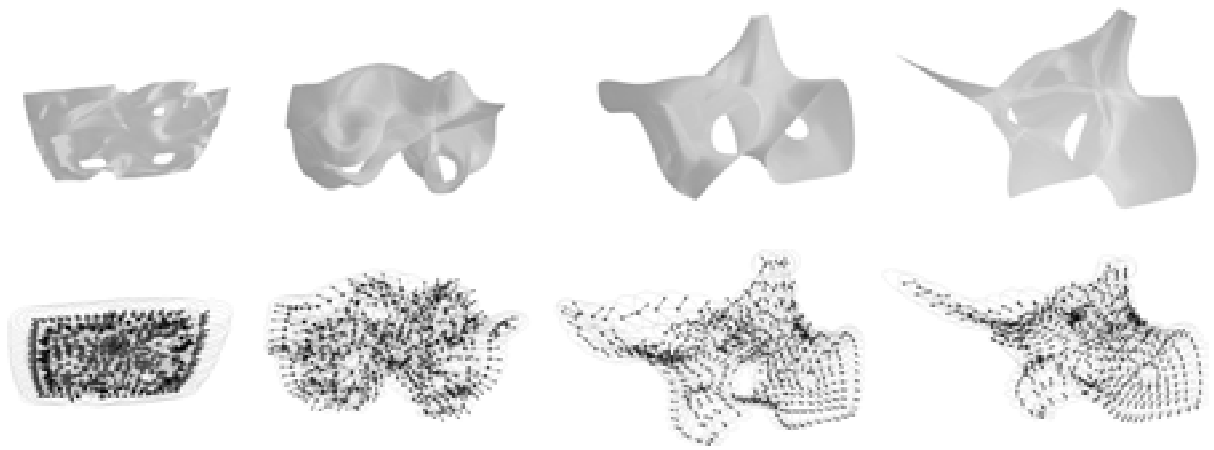


Figure 9. Busan Opera House, Kokkugia 2011. Left: Initial organisation of agent relations. Right: Final result.

Figure 10. BodySwarm research project, Kokkugia 2011. The topology self-organises through controlled interactions.

surface is self-organised to a larger degree. The interactions that re-configure the surface are not programmed in advance, but rather take place when certain spatial circumstances appear as part of the generative process. The decisions concerning the transformation of the topology happen locally, through the interaction between the agents that control the surface. In this example the result is completely emergent, both in terms of shape and topological relations. Kokkugia has developed a series of methods for the use of agent-based systems in architectural design. Another example is their proposal for the *AirBaltic Terminal* in Riga, which was developed with Buro Happold in 2010 (see Figure 11). Here, structural analysis software is integrated with an agent-based system. This tool is used for generating a complex fibrous roof structure. The form-generating process is guided by design intents and structural properties, both of which are embedded in the behavioural logic. In this way, the solution represents a negotiation between aesthetic and structural parameters. The parameters of the project are to be understood in a broader sense than numeric variables, since the logic of the system was an integrated part of the design process. With respect to the discussion on emergence, the agents that generate the formations works purely from local conditions and decisions. This results in a



complex unified design with global structural properties. Therefore, both structural and aesthetic properties are emergent effects of the process.

5.4 Nonlinearity

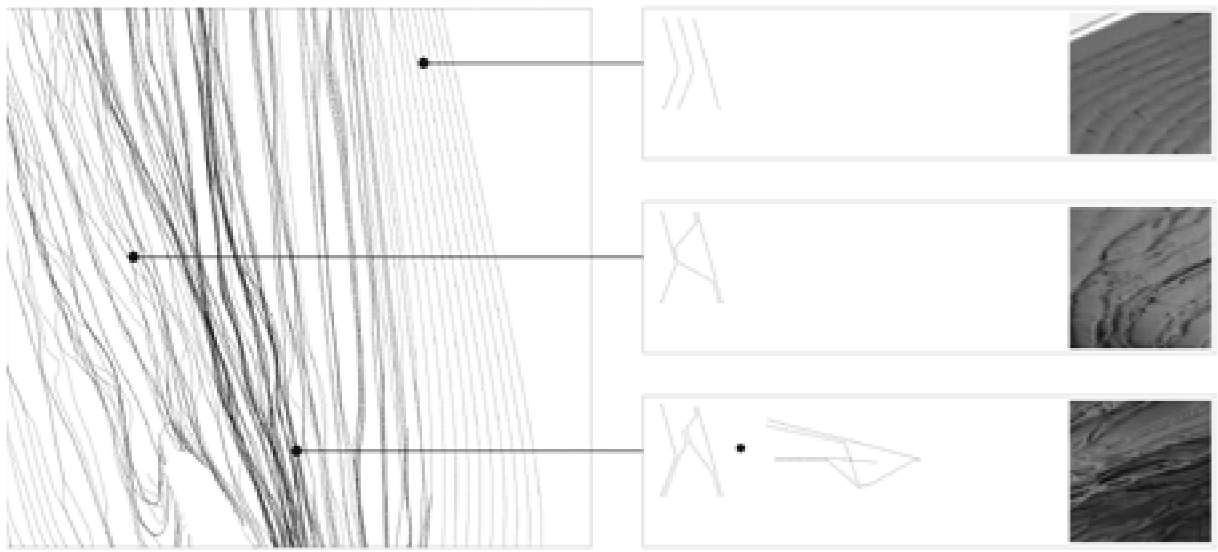
Agent-based self-organisation can be understood as nonlinear, and in this section the implications of nonlinearity in architectural design are discussed. Holland explains nonlinearity with a basic mathematical example where two variables in an equation are multiplied with two different addresses. There is not a direct way of predicting the outcome of the system from an analysis of its logic and environment. Mathematically, the variables are multiplied and lead to complex results. Holland uses the example of shifts in population sizes through predator-prey relations. Essentially, the factors influencing the shift in size also affect each other over time. When the predator population increases, the population size of prey corresponds by decreasing. Since the number of prey affects the number of encounters between predator and prey, which also affects the calculation of change of predator population size, the latter will at some point begin to decrease. The result is an oscillating change in size of both predator and prey populations. In agent-based systems, nonlinearity is influenced by the behaviour of each agent. This behaviour is the result of a series of multiplications that emerge from the current state of the agent population. Figure 2 in Chapter 8.4.2 illustrates how behavioural forces are first calculated and weighted individually, then added before affecting the state of the agent, which typically is its position and velocity in the kind of systems discussed here. If a system contains nonlinear behaviour, it is difficult to calculate or predict its outcome because the relation between input and output is indirect. It is interesting to consider the implications of this with respect to architectural design. Generally, designers have a need for having direct control over the outcome of the design process. When it comes to realisation of the project, it is crucial that the produced drawings and technical descriptions are consistent and precise in terms of directing the construction of the building. However, systems with nonlinear behaviour can also be relevant to architectural design. All of the previous examples stated in this chapter are nonlinear. The unpredictability is not a primary property, but rather a consequence of the complexity of agent-based systems. The systems are nonlinear because they enable negotiations between several parameters to be played out during the form-generation process. In terms of predator-prey relations, the global behaviour of the system is nonlinear yet predictable. Here, the property of emergence is perhaps what makes some systems more predictable and more useful for architectural design. For instance, the goal of dynamic relaxation is to arrive at a structurally optimised geometry rather than achieving a surprising form, even though the form is often expressive. In this sense, nonlinearity is perhaps an inherent property of self-organisational systems, rather than an advantage. Having said that, nonlinearity is a property that enables the development of systems to be an integrated part of the creative design process. Design intents are refined through feedback in the form of the behaviour of the nonlinear system, which changes through the development process. As Roland Snooks states, 'I see agent-based algorithms - or perhaps any generative algorithm – as operating in two main modes, although not mutually exclusively. One is self- organization to solve a complex problem. For example, this might be self- organization of program or structure. The second is the generation of emergent patterns or forms or affects, which is an attempt to capture non-linear behaviour.'[20]

5.5 Flow, diversity and tagging

When Holland addresses flow as a property, he is primarily referring to the fact that complex adaptive systems, such as urban societies or biological systems, can be seen as a continuous flow of matter and information. Through the use of digitally simulated systems that our interest is directed towards here, the meaning of flow becomes abstract in character. Flow has various meanings on different levels. The constant negotiation between the agents is essential to the functionality of the system, enabling the dynamic process of self-organisation to take place. With respect to geometric self-organisation, there must be forces between the agents that help

20 Balmond, C, *op.cit.*, 2010.

Figure 11. Studies for the propsal for AirBaltic Terminal, Riga by Kokkugia and Buro Happold, 2010. The agents generate strands, which again appears in the form of different structural patterns.



them to change position, effectively approaching an equilibrium state if possible. As this is a constantly changing system, it can be considered fluent. In many systems, individual agents, or nodes, can fluently be added or removed from the system without interfering with the general behaviour. In this way, the system functions as a living organism, where the system self-organises and reconstructs eventual gaps in case individual cells are destroyed. This approach is implemented in *Self-organising Surface*. Here, agents are constantly added to, and removed from the scene, in order to establish the surface. If an agent lacks a neighbouring agent, in terms of constructing a consistent mesh structure, it can activate the construction of a new agent, placed in a random position close to its own position. Often, the new agent will automatically search for a vacant position in the mesh. Alternatively, if an agent experiences the local environment as too crowded, that is, if too many agents are within the distance limit to which the agent makes connections, it can activate the destruction of one of these agents, thereby decreasing the density in the area. In this way, the population is in constant flux, or *flow* to use Holland's term. In the previous example, agents locally negotiate the separation distance between them. These values 'travel' across the structure, which is comparable to the way impulses would travel through a nervous system. This type of flow directly relates to the primary type of flow that Holland is referring to. It can be said that many agent-based systems are preoccupied with dynamic mechanisms that occur through the negotiations established between the agents. A huge potential for combining these mechanisms through the integration of flows of information from external systems exists. An example of this is the linking of simulation of contextual parameters, which is demonstrated in the *DLA Solar System* described in detail in Chapter 8.6. A combination of versatile agent-based systems and this kind of interaction between systems has the potential to reveal new forms of self-organisation relevant to architecture.

When Holland addresses *diversity* as a property of complex adaptive systems, he is mainly occupied with the biological capability of maintaining complex ecologies. Through evolution, if a potential niche in an ecosystem is vacated, the niche is eventually occupied by new species that may exploit it for food supplies. However, the term diversity leads towards other aspects, which relate to the use of agent-based systems for architectural design. We have already encountered diversity in some examples. For instance, in Kokkugia's proposal for the *Busan Opera House*, certain agents were tagged (a term that will soon be discussed in more detail), in order to perform specific tasks during the form generating process. This type of diversity assists in establishing advanced methods for controlling the generative process, as the project demonstrates.

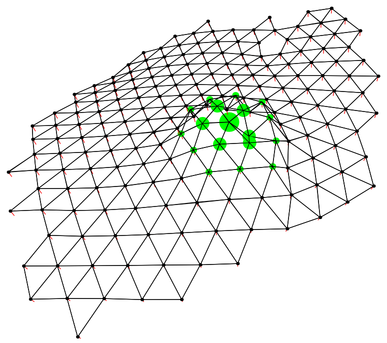


Figure 12. Self-organising surface. A single agent was coded to have an angle of 100 degrees between the normal and the surface plane, which suggests the forming of a hill top. The behaviour was transferred to the nearest neighbour agents.

Furthermore, diversity among the agents can reveal new levels of complexity in the generated formations. Some types of diversity were tested during the development of the method *Self-organising Surface*. An example of this is depicted in Figure 12. Here, particular agents were programmed to form a local hill in the otherwise entirely planar surface structure. As described in further detail in Chapter 8.4, these diversities can also lead to disturbances and unexpected behaviour within the system. Diversity on a different level, namely in the resultant formation, is perhaps one of the most essential properties of the agent-based system. For instance, in the project for the *AirBaltic Terminal*, a series of differentiated structural typologies emerged through the generative process, as shown in Figure 11. When tectonic logic is embedded in the algorithmic logic, it is also possible to incorporate tectonic diversity in the generative process. For instance, a diversity of agents representing joints, members and cladding, may be implemented in the generative system. This research project has not taken agent-based systems as far as to test this approach. However, an indication of the potential can be seen through the diverse components in the *Concrete Gridshell Pavilion*, described in Chapter 8.1.

Basically, *tagging* enables diversity to occur. When discussing agent-based systems for design purposes, it is relevant to take a closer look at the meaning of tagging. As previously mentioned, it makes sense to tag individual agents to take on certain roles in the form generating process. A group of students at the Morphogenetic Studio in Aarhus established a system, which consisted of three different types of agents. Two of the groups were essentially identical, but the division of groups allowed them to interact in a different way than if they were just one group. The third type was fundamentally different, since the agents were connected in a mesh structure, representing a surface. This project is depicted in Figure 13. The 'free' agents are initially placed above or below the surface depending on which group they belong to. The agents then start to push and pull the surface, depending on their tagging. This effectively makes cuts in the surface. As the surface is deformed and torn apart, the agents can move more freely and begin to interact directly with each other. The system was used to establish a negotiation between parameters deriving from a specific context in Barcelona and embedded design intents. Another type of tagging is crucial for establishing emergent effects, since emergence is enabled through changing the state of individual agents as part of the form-generating process. A simple example of this type of tagging is demonstrated in an experiment with the method *Self-organising Surface*, which is further described in Chapter 8.6.9. Figure 12 shows a situation, where the majority of the agents were set to form a relatively planar surface, but a single agent was coded to form a hilltop. When this agent established connections, its immediate neighbours were tagged to adopt the hill-forming behaviour, thereby forming a larger hill on the surface. This method makes use of tagging as the agents tag their neighbours in order to define the relations that eventually come to define the surface. In fact, it is the 'active' agent that is tagged, since it has a continuously updated list of its neighbouring agents, depending on the distance between the agents. A similar mechanism is seen with the *BodySwarm* project. When an agent detects certain geometric conditions it can activate state changes for itself and a group of agents, thereby tagging them with new relational states.

5.6 Conclusion

This chapter has discussed the implications of a using three-dimensional self-organisation in architectural design. The possibility for using agent-based systems to handle complexity in the form generation process has been demonstrated. Many of the examples reflect the challenge of exploiting the dynamic character of the agent-based system, whilst simultaneously extracting consistent geometry from the system. Basically, two different strategies can be detected. One is focussed on an emergent effect, which helps to order an otherwise disordered field of agents and their relations (as demonstrated with *Self-organising Surface*). The other approach is to ensure a priori topological relations, and then use the agent-based systems to gradually transform the geometry, whilst maintaining consistency throughout the process. This transformation is then often directed towards morphology, but can also include topological relations, as seen through Kokkugia's *BodyTopologic* method. One of the main advantages of agent-based systems is their capacity for establishing complex negotiations at a detailed level in the formation process. They are versatile in the sense that a large range of behaviours and parameters, concerned with design intents, tectonics or context, can be embedded in the algorithmic logic. The examples used in this thesis are vector based. There is direct potential for using this form of three-dimensional mathematics to represent three-dimensional physical space. This approach enables a dynamic, non-hierarchical and grid-less method for generating self-organised geometry, compared to other methods, such as typical examples of using *cellular automata*. What has not been stressed within the chapter is the fact that many of the methods are demanding of computational resources. An increase in the number of agents often leads to an exponential increase in calculation time. Therefore, natural limits for what type of solutions can be solved through the use of agent-based systems must be considered. This

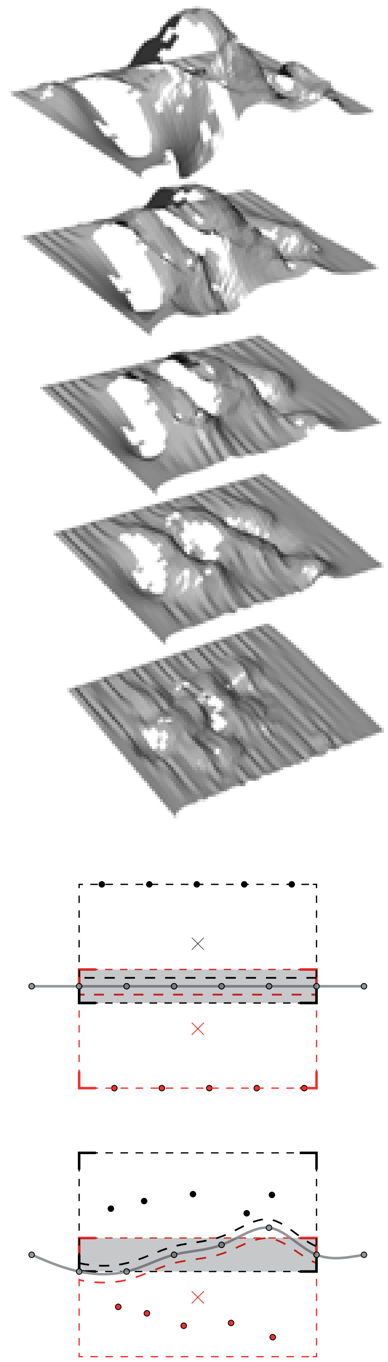


Figure 13. Differently tagged agents negotiate deformation of a surface. R. Marozaite, T. Rasmussen, S. Olson and D. Martinaitis, Morphogenetic Studio, Aarhus, 2010.

issue should be related to the general optimization of computer technology, which allows computation of greater amounts of data. Also, there are often possibilities for the optimization of the developed codes that are used. When designers without particular experience in computation are implementing the logic, there is potential for achieving a greater performance through optimization of the code. Whether it is necessary depends on the complexity of the system and the design task. Usually, the running time is related to a balance between the number of agents, behavioural and geometric complexity, and the number of iterations. In most cases more iteration is needed in order to establish emergence, compared to methods concerned with mapping the agents. However, with mapping, it is necessary to construct geometry from many system states, which is time consuming, whereas, only the final state of emergent formations defines the geometrical outcome. With these observations, we will leave the subject of self-organisation for a while, but retain a three-dimensional approach to generative design. Namely, when aggregate growth is discussed in the next chapter.

6 Aggregate growth

6.1 Spatial patterns from cells and components

The methods discussed in the previous chapters reflect different aspects of generating architectural geometry. Surface topology, pattern distribution, complex topology and agent-based systems have been explored. These methods cover a spectrum ranging from two-dimensional to three-dimensional form-generation, and their organisational logic is related to self-organisation. This chapter describes a different approach. The structures discussed in this section are not self-organised, since the components that they consist of do not iteratively negotiate their state. Some forms of local negotiations can still affect the form-generation. But these are rooted in contextual conditions or the state of the structure at the time of articulation and positioning of the component. Once the component is placed, it does not change its state. As such, there is a large degree of linearity within form-generating processes. However, they are not all completely deterministic. A degree of randomness is allowed to affect the outcome in two methods developed as part of this research. The two methods, *SAGA* and *Solar DLA System*, are described in detail in Chapters 8.5 and 8.6, and form the basis for discussing aggregate growth. This chapter discusses periodic cellular organisation, random versus deterministic behaviour and adaptability with respect to contextual parameters and realisation conditions. The *National Aquatics Centre in Beijing* demonstrates how a complex cellular pattern can be implemented as part of an architectural project. The cell lattice shares the property of being periodic like the *Solar DLA System*. The cell geometry of the two examples is comparable, even though they differ in geometric complexity. The original concept for the *Watercube* being based on self-organising soap bubble formations, has some relation to Frei Otto's experiments with soap film, mentioned in Chapter 2. The realised building structure is based on a specific principle for cell packing rather than nonlinear mechanisms as found in physical experiments with soap bubbles.[1] In terms of generative techniques, algorithmic tools played an essential role in the realisation of the building. I will initially examine the organisational logic of this method.

1 D. Weaire, S. Hutzler. *Nonlinear phenomena in soap froth*, Physica A 257, 1998, pages 264–274

Figure 1: PTW Architects and Arup. Watercube, 2008, facade. Image: PTW.

6.2 Watercube: cellular space filling

By observing a random bubble pattern, the design team decided to use a repetitive system of polyhedra in order to optimise the realisation of the building. Theories of the rules for self-organising soap bubbles generally originate from Joseph Antoine Ferdinand Plateau. His work is summarised in his book, 'Statique Expérimentale et Théorique Des Liquides Soumis Aux Seules Forces Moléculaires', published in 1873. Based on Plateau's work, Lord Kelvin developed a geometrically optimal solution for division of space with equal cells in 1887.[2] Each cell is a 14-sided polyhedron, or more precisely, a tetrakaidecahedron. As recent as 1994, Denis Weaire and student Robert Phelan developed an improved version of a space filling cell system. The new version consists of two different types of polyhedrons: a tetrakaidecahedron and a dodecahedron, which has 12 sides. These two types have equal volumes.

In the search for a rational, yet bubble-like structure, Tristram Carfrae of Arup first observed Kelvin's tetrakaidecahedron. This directed him to Weaire-Phelan's solution, which eventually was chosen as the organisational logic for the space frame structure in the *Watercube* project. Weaire-Phelan foam is less than one percent more efficient in terms of material efficiency, but since two different cell types are used, the expression coincides more effectively with the original idea. Additionally, the spatial organisation is rotated, which makes it difficult to perceive the repetition of the pattern. This is highly visible on the façades, since they are the result of sections though the pattern, and end up with great geometrical variety in the panels. Still, the fact that the system consists of a periodic spatial

2 Lord Kelvin (Sir William Thomson), *Philosophical Magazine, Vol. 24, No. 151,* 1887, page 503

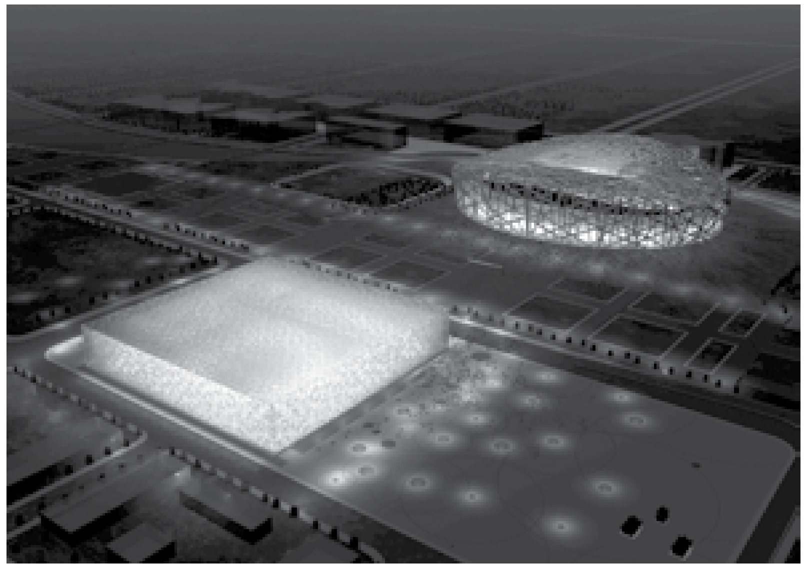


Figure 2: Competition design of Watercube next to the Olympic Stadium, Birds Nest. Image: PTW.

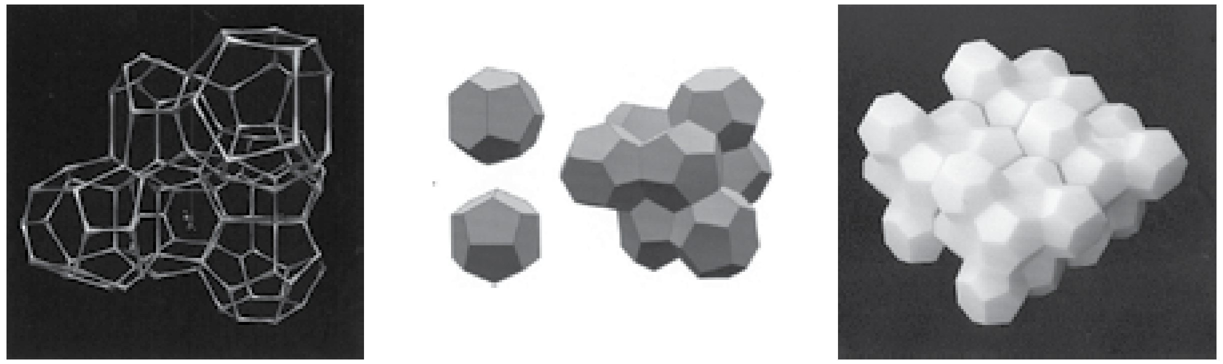


Figure 3: The Weaire-Phelan space division principle. The system is 0.3 percent more efficient than Lord Kelvins and consists of 2 polyhedrons types of polyhedrons.

pattern makes it substantially more rational to produce, particularly when addressing conventional industries.[3]

6.3 Design concept and realisation

The National Aquatics Centre, also known as the *Watercube*, was designed for the Olympic Games in Beijing, 2008. The project was chosen from 10 proposals in an international competition in 2003 and was designed by two Australian companies, PTW Architects and the Arup Australasia engineering group, together with the China State Construction Engineering Corporation (CSCEC) and the CSCEC Shenzhen Design Institute. The initial idea was to use a direct reference to water as a tectonic motive. A general interest in self-organisational form-finding and soap bubble patterns was an important driver in the design process. As Kurt Wagner from PTW states, 'For this project we were researching the meaning and relevance of water, and we were intrigued by images of foam, soap bubbles, molecules and corals, and the organic structures behind them.'[4] Simultaneously, local typologies affected the design and influenced the rectangular form of the building.

An underlying mathematical principle made it possible to use algorithms to generate a variety of digital models. The structure consists of 22 000 steel members. In order to reduce the amount of steel, a structural optimisation technique was carried out. Subsequently, a rationalisation algorithm was used to express the optimised structure as consisting of three different member types. Furthermore, it was possible to automate much of the production of drawings and information necessary for the realisation of the project.[5] Concerning the numerous joints, the solutions were

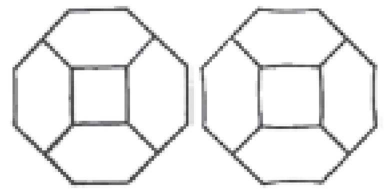


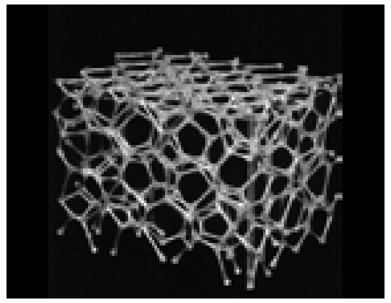
Figure 4: Projection drawings of the minimal tetrakaidecahedron, developed by Lord Kelvin in 1887. The edges are slightly curved.

Figure 5: Natural soap film bubbles. Usually bubbles vary substantially in size. However, it is possible to achieve naturally formed Weaire-Phelan bubbles, which could suggest that the principle reflects an optimal solution under certain conditions. Photo: Ruud Kempers

3 Henry Fountain, 'A Problem of Bubbles Frames an Olympic Design', *http://www.nytimes.com/2008/08/05/sports/olympics/05swim.html*, 2008

4 Peter Rogers, 'Welcome to WaterCube, the experiment that thinks it's a swimming pool', 2004, viewed February 4 2012, <*www.guardian.co.uk/science/2004/may/06/research.science1*>

5 Ethel Baraona Pohl, *WaterCube: The Book*, dpr editorial, Barcelona, 2008, page 190



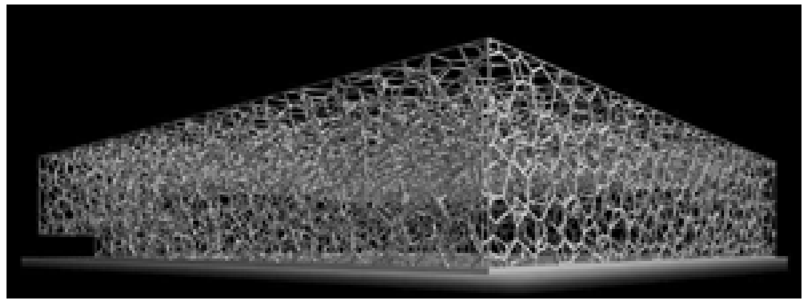


Figure 6: Left: Resin model of the periodic Weaire-Phelan structure. Right: Model of the space frame structure. The facade appears as entirely random, due to the transverse section through the spatial pattern. Photos: PTW

relatively manageable, due to the fact that the foam cells always meet at certain angles, particularly in a regular repetitive pattern. The façade is constructed from pneumatic cushions, restrained in aluminium extrusions. The cushions are made of layers of ETFE plastic, and are inflated with low-pressure air to provide insulation and resistance to wind loads. The material coincides with the structural requirement for great spans of the façade and roof panels, which would be impossible to construct out of glass without internal framing. The cushions cover the whole building with both an external and an internal layer. They vary in terms of heat reflection, where they are positioned, the season of the year and the current use of the building. Shading foils in the cushions between the two ETFE layers in the internal layer can be controlled to gradients of

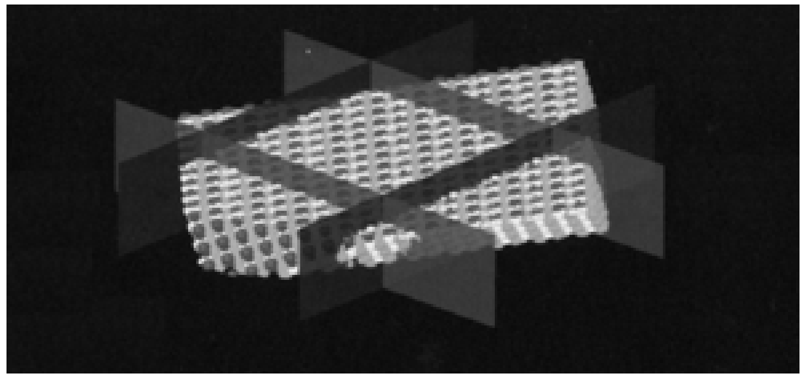


Figure 7: The periodic structure is rotated before cutting the sections that define the surfaces of the building. Illustration: PTW

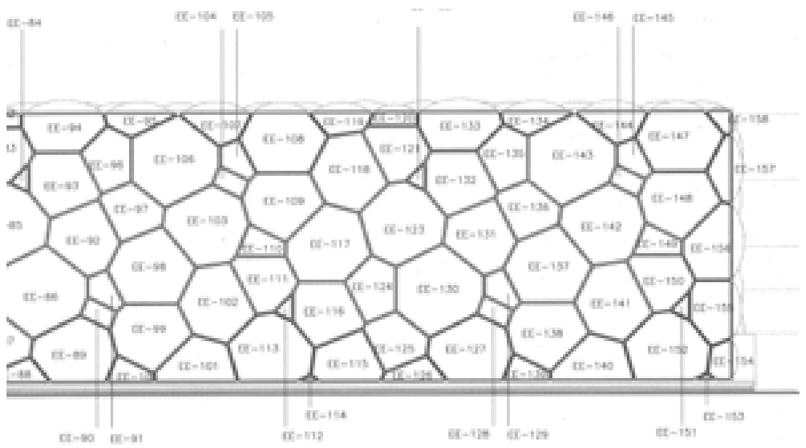


Figure 8: A numbering system helped to control the production of 3000 differentiated ETFE cushions. Illustration: PTW

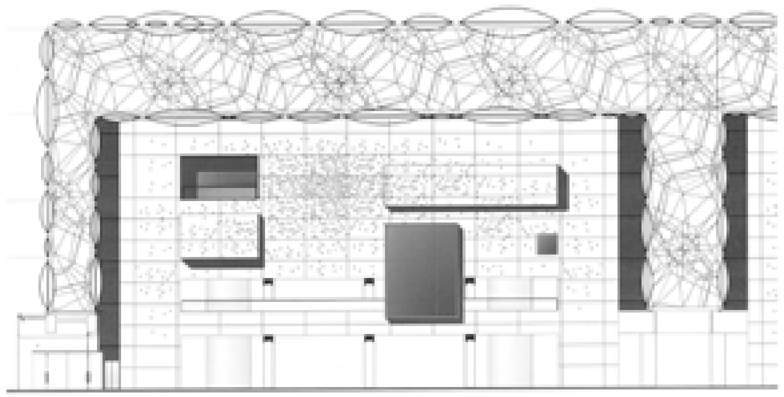


Figure 9: Section drawing showing the two layers of ETFE cushions. The air space between the layers helps to regulate the internal climate. The cushions in the inner layer are provided with controllable layers of internal foil in order to filter the light that enters the building. Illustration: PTW Architects, Arup, CSCEC.

openness, depending on the situation, providing the building with an advanced system for controlling illumination and the demanding climate of the swimming stadium.

6.4 Watercube: conclusive remarks

On an immediate level, the building fulfils the idea of achieving an image of water as a material, related to the functionality. Despite the fact, that the motive is somewhat representational, the emphasis on the tectonic pattern provides the building with a character of its own, beyond the pure image of bubbles. Additionally, the idea of soap bubbles is translated into a useful spatial concept based on mathematics, and becomes a structural and organisational principle.

The embedded mathematical logic enables analysis, optimisation and generation of information for the realisation process. The random expression derives mainly from the complex packing of the cells and the rotation of the spatial system. This again results in a large number of individually different components, and the variation occurs primarily in the lengths of members and façade component edges. These differentiated members affect the shape of the ETFE cushions, which then obtain highly differentiated proportions. The project demonstrates how mathematical principles and computation can be important drivers, both in design and production processes, and how general design thinking is essential, as with projects that are less computer based. I. K. Andersson and P. H. Kirkegaard state, 'The results from technical analyses are constructively and artistically worked into the design, and they are therefore an important design parameter, rather than an appliqué to a form. The process is hereby a hybrid process, with architects and engineers working closely together in a digital continuum.'[6] As such, the project demonstrates how the use of generative techniques

6 Ida Kristina Andersson & Poul Henning Kirkegaard, 'A discussion of the term digital tectonics', *Digital Architecture & Construction 2006, pages 29-39.*

Figure 10: Right: The realised ETFE facade. Photo: Chris Bosse.

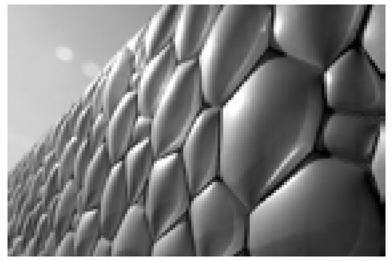


Figure 11: The original DLA model. Random aggregation of 3600 particles on a square lattice. Illustration by Witten and Sander.

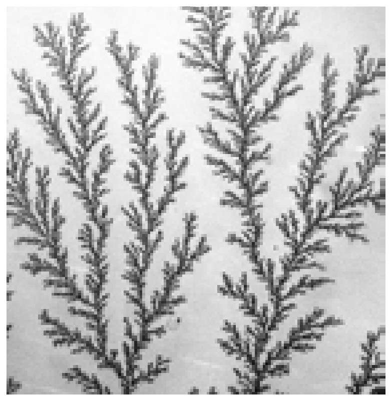
is able to encourage architects, engineers and manufacturers to exchange knowledge and participate during the development of the project, rather than solving separate problems in a linear process.

The surface character achieved through the use of ETFE plastic cushions corresponds with the project theme of bubbles and is a strong part of the building's tectonic appearance. Besides the soft convexities, the material has a semi-transparency, which perhaps creates an underwater atmosphere, corresponding with the general expression. Some of the initial sketches for the project displayed greater variation in bubble size, and if the developers were to rely on a completely digitised production process (rather than a conventional and labour intensive industry), it would have been possible to realise a building with even greater variety through a consistent use of generative techniques. However, the final result effectively corresponds with the initial design intents.

6.5 The Diffusion-Limited Aggregation model

Principles for cell packing used in the *Solar DLA System* method can be compared with those that formed the basis for the *Watercube*. Before this topic is approached, two general subjects that form a background for the *Solar DLA System* must be identified. The first is the organisational principle, which is an algorithm for simulating *diffusion-limited aggregation* (DLA). The second is the phenomena of *stigmergy*, a topic rooted in biology. Algorithms for generating complex patterns are usually developed in order to describe or simulate natural phenomena. This is also the case with the DLA model, as the reason for using it in relation to self-organising tectonics is the properties of the model, rather than its origin in natural phenomena. The DLA is essentially a principle for random growth. An important property of the model is that it results in characteristic branching and open structures, avoiding a compact packing of cells. Furthermore, the principle is defined by few elementary rules, making the model versatile and open to adjustments in relation to use in different situations.

Figure 12: Dendrites of copper crystals. Photo: Pauls Lab



T. A. Witten and L. M. Sander published a model for diffusion-limited aggregation in 1981.[7] The model was developed in order to describe the mechanisms that make dust particles aggregate in air. Subsequently the model has been used to describe how some types of crystals aggregate under certain conditions, and form dendrites. This term is derived from the Greek word for tree: *dendron*, and refers to the characteristic plant-like shape of the crystal formations.

7 T. A. Witten, Jr. and L. M. Sander, 'Diffusion-Limited Aggregation, a Kinetic Critical Phenomenon', *Phys. Rev. Lett. 47*, The American Physical Society, 1981, pages 1400-3.

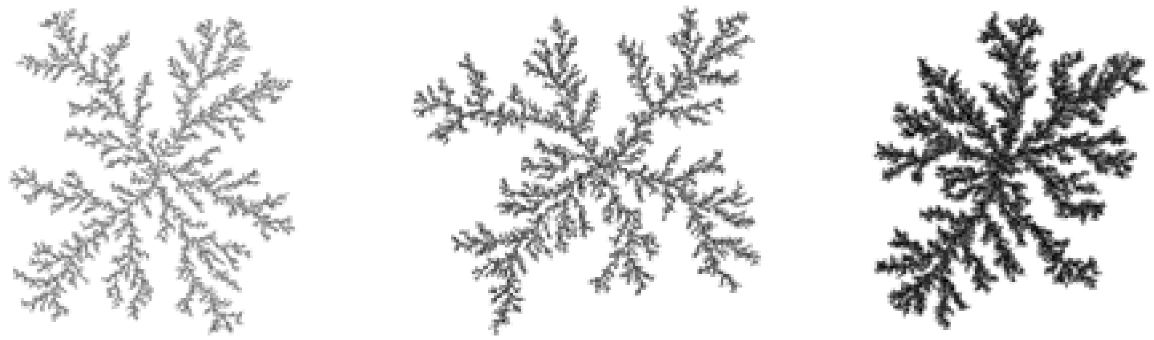


Figure 13: The density of the DLA pattern can be controlled by adjusting the 'stickiness'. The right image shows lowest tendency of sticking. Illustrations by Paul Bourke

When crystals form gradually, they tend to develop the well-known compact faceted shapes. If the crystals are formed under highly unstable conditions, as in an electrically charged solution of metal and salt, the crystals grow faster, becomes irregular, and branches grow out. These types of crystal growth form dendrite patterns. The DLA simulation model presented by Witten and Sanders is essentially based on a set of simple rules. First, a single seed particle is placed in an empty environment, defined as a lattice. Then a 'walking particle' is placed in a random position away from the seed. The walking particle moves from one empty position to the next, until it reaches either the edge of the environment or the seed particle. In the first case, the particle is cancelled. In the latter, the new particle is in a fixed position next to the seed particle, altering the structure that now consists of two particles. A third 'walking particle' is now placed, and the process is iterated. The result is a random and distinctively open structure. The openness is derived from the fact that every particle added to the structure arrives from a point away from the structure, and attaches to the first part of the structure it touches. With respect to dendrites, the pattern is recognised both in two and three dimensions. Similarly, the algorithmic principle can function both in two and three dimensions. As demonstrated by Paul

Figure 14: Ice flowers is a form of dendrites. Often the dendrites are layered.

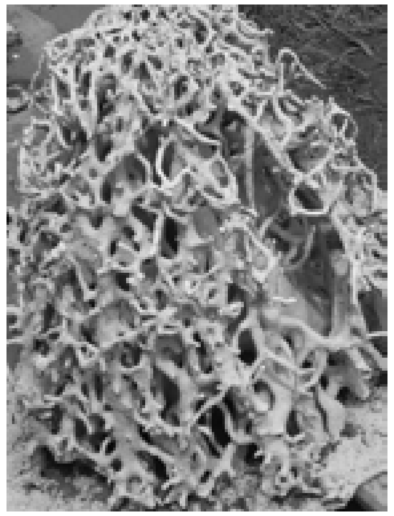


Figure 15: As part of research conducted by Rupert Soar has been made casts of the interior of termite mounds. The casts indicate the complexity of the transportation and climatic systems in the termite mounds. Photo by Rupert Soar.

Bourke[8], the density of the DLA pattern is adjustable. Normally, when the wandering particle reaches the existing structure, it will stick. By introducing a probability of sticking, the density is increased. Generally, with a lower probability of sticking, the result becomes more hairy and solid.

6.6 Stigmergy

The French zoologist Pierre-Paul Grassé introduced the term *stigmergy* in relation to his research in social insects in 1959. It is derived from the Greek words *stigma* and *ergon*. *Stigma* means mark or sign, and *ergon* means work or action. The term *stigmergy* refers to the phenomena where the work of agents is stimulated by the work that is previously performed by other agents.[9] Stigmergy occurs for instance when termites deposit or remove matter during the construction of a termite mound. When a termite adds a soil pellet to the structure, it also leaves a pheromone trail that encourages other termites to place pellets in the same location. The soil is often collected from the immediate surroundings, which results in both a densification of the structure and simultaneous excavation of surrounding spaces. The behaviour of termites is qualitatively different, depending of local state of the mound, and does not rely on communication between the builders, but rather the sensing of different pheromones left by the other termites when depositing soil. When termites are stimulated to repeat actions performed by other termites, such as depositing matter, a positive feedback loop occurs, and the accumulation of the pheromone can have a quantitative effect on the behaviour of the termites. Typically, when the magnitude of pheromone is increased, due to increased activity, a larger number of termites are encouraged to deposit soil in the area. These stigmergic reactions have been studied through experiments where termites are placed in an artificial environment with evenly distributed soil. Initially the termites randomly deposit soil pellets without any significant formation occurring. Due to random fluctuations, densification in some parts of the environment will occur, and from this point the behaviour of the termites shifts and becomes more consistent. This is due to the accumulation of pheromone in the dense areas, compelling the termites to concentrate their activities to these areas.

8 Paul Bourke, 'DLA – Diffusiont Limited Aggregation', 1991, updated 2004, viewed 7 January 2012, <*http://paulbourke.net/fractals/dla/*>

9 S. Camazine, J. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, E. Bonabeau, *Self-Organization in Biological Systems,* Princeton University Press, New Jersey, 2001

Although many levels of complexity arise from the study of stigmergy in nature, it does not completely explain the building processes carried out by social insects. When introducing the term in relation to an architectural building process, the idea is not so much to emphasize the obvious similarity between building processes in nature and human building processes. Rather, a goal is to extract some knowledge from natural systems in order to be able to implement a higher degree of complexity in the building process. Furthermore, an outcome may be to begin implementing local negotiations between various parameters during the form generation process, similar to the qualitative and quantitative differentiation recognised in relation to stigmergy.

Figure 16: Rendering of 14-sided polyhedrons.

6.7 Aggregate construction

The method *Solar DLA System* was initiated from the workshop cluster *Agent Construction* at the Smart Geometry conference in 2011. A large model was constructed at the workshop, and a detailed explanation of the model and *Solar DLA System* is detailed in Chapter 8.6. The agent-based approach to constructing the physical model was inspired by stigmergy as it appears in the form of termite mounds. Before a discussion of the presence of stigmergy in these projects can take place, the geometric principle that was used in *Agent Construction* and *Solar DLA System* must be explored. The principle is a three-dimensional lattice consisting of 14-sided polyhedrons, as shown in Figure 16. The polyhedrons fill the space completely, which is similar to the behaviour of Weaire-Phelan foam described in relation to the *Watercube*. The shape is also referred to as a truncated octahedron, and the symmetrical shape allows growth in 14 different directions, consisting of 6 orthogonal and the 8 diagonal directions. The geometry of this material is more advanced than a cubic lattice, but simpler than the Weaire-Phelan foam. This means that as a space-filling pattern it appears as completely periodic, unlike the Weaire-Phelan foam. However, in terms of the *Solar DLA System*, the goal is not to fill the lattice, but to provide a design space for growth logic. Despite the regular lattice, the results gain an organic appearance, as shown in Figure 19. Implementation of Weaire-Phelan foam would have made the realisation of a physical model and the development of generative logic more complicated.

The intention of *Agent Construction* was to let human agents interact by building a model during the workshop. Each participant behaved according to various elements in the environment, such as response to light conditions and structural consistency. The experiment was successful in the sense that a form of order in the pattern formation emerged during the building process. Still, it is questionable whether these tendencies can be described as

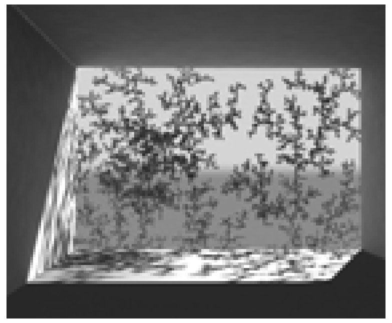


Figure 17: Solar DLA System. Example of interior view through solar protection structure.

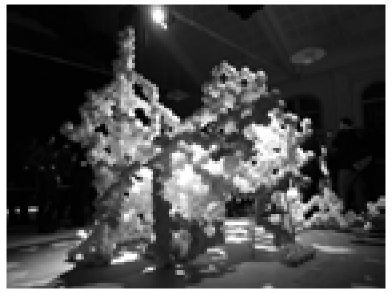


Figure 18: Agent Construction, Smart Geometry 2011. Completed structure.

emergent, as the ordering was a result of local top-down control over the process, even if the constructors did not directly communicate their individual actions. When the ‘agents’ discovered a potential for establishing an ordered formation, such as a wall topology, they would silently collaborate to complete the formation. This is fundamentally different to the entirely local instinct-controlled behaviour of termites. The rules that directed the human agents were formulated as goals or themes, and were therefore too vague to absolutely control the local behaviour. Another aspect of the experiment is that the human agents did not use a form of communication that can be compared to pheromones among termites. Only the shape of the structure at any current state encouraged the subsequent actions of adding or removing components. The builders were encouraged to limit direct communication, since, the goal for the experiment was to investigate a bottom-up approach. However, they were able to discuss strategies. From this perspective, it is questionable whether the growth process can be described as stigmergic. The use of the word ‘agent’, as suggested in the workshop title, is thus misleading. Studies in pattern formation in biology have used agent-based modelling. For instance, Eric Bonabeau has studied agent-based mechanisms for generating patterns that resemble those found among termites.[10] Although the participants in *Agent Construction* could perhaps be referred to as ‘agents,’ the building process itself was not self-organised like the agent-based systems described in Chapter 5. Because the rules were indefinite, the agent behaviour was effectively random and, or locally top-down controlled. Here, it is necessary to distinguish nonlinear self-organisation from random behaviour. This should not problematise the experiment, but distinguish that the way agents appear in this case is different from the previously discussed methods. Neither is it problematic as such that the *Agent Construction* experiment did not precisely demonstrate stigmergy. However, it is possible that a notion of stigmergy could have informed the realisation process with stronger self-organisational properties, which again could have resulted in bottom-up pattern formation.

6.8 Cellular solar protection

Some of these issues can also be raised with respect to the *Solar DLA System* method. The system does not make use of self-organisation understood as the nonlinear behaviour from interaction between entities. The components do not communicate directly or

10 Eric Bonabeau, ‘From classical models of morphogenesis to agent-based models of pattern formation’, *Artificial Life archive Volume 3 Issue 3*, MIT Press Cambridge, MA 1997, pages 191-211

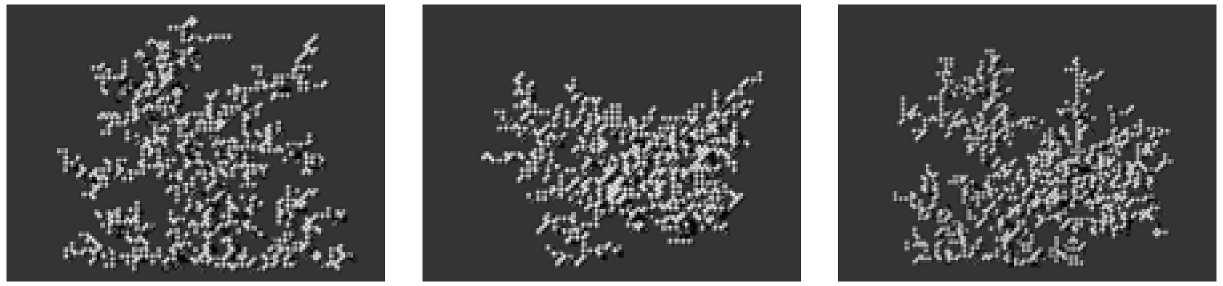


Figure 19: Rendering of the DLA solar protection pattern. Various degrees of protection tendency.

indirectly, as with stigmergy among social insects. The behaviour of the system emerges from rules that direct the local behaviour, which reflects contextual conditions. One condition is the state of the structure, as with *Agent Construction*. New cells can only be added to spaces that connect to the existing structure. Another condition is the level of solar radiation, since the decision to place a cell in a specific position depends on the effect in terms of protecting predefined areas from direct sunlight. The behaviour is stochastic because the decision of placing a cell relies on calculating a weighted probability. The degree of solar protection is controlled by adjustable parameters, which affect probability. The growth process is guided by local negotiations between the internal logic of the DLA pattern and solar protection probability. The *Solar DLA System* functions as a completely bottom-up controlled process. Only contextual conditions, inclusive of the framing of the design space, affect the form-generating process from an external perspective. As such, the system differentiates from the approach of *Agent Construction*. Figure 19 displays different degrees of solar protection. Towards the left of the figure, the DLA growth is 100% random without solar protection behaviour. Subsequently, only the internal logic of the DLA algorithm is expressed in the structure. Randomness and solar protection do not exclude each other. Rather, both behaviours can trigger the placement of a cell, with individual probabilities. Despite its two-dimensional character, there is a depth of six layers present in the structure. In the centre image, the system is regulated towards 100% solar protection and 1% random growth. The growth is concentrated in a specific area, because the attractor points that are to be protected are positioned there. The image on the right shows 20% random growth and 100% tendency for solar protection. In this sense, the pattern represents a negotiation between, or balancing of, the behaviour of solar protection and the internal logic of the DLA algorithm.

To return to the question of geometry and organisational logic, it is possible to imagine the *Solar DLA System* further developed with an entirely different approach to geometry. The original DLA method works from a lattice structure, but this is not an inherent property of the algorithm. It is possible to implement the

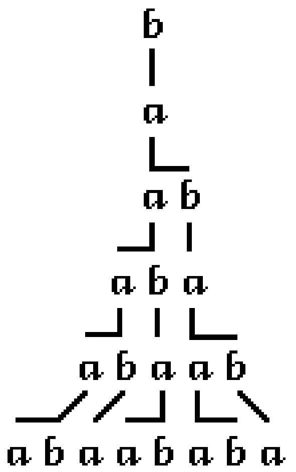


Figure 20: Example of derivation in a DOL-system. The letters are replaced for each generation. Rules: b -> a, a -> ab

DLA pattern though a vector-based approach. Leaving the three-dimensional grid would entail development of a different component solution. Taking the solar protection functionality a step further could be to implement this part of the logic with a completely different organisational logic. This could for instance be an agent-based system similar to those discussed in Chapter 5, allowing for other types of spatial formation to occur. This approach would then entail the development of different component geometry.

6.9 L-systems

Another example of aggregate growth is demonstrated in the method *SAGA*, described in greater detail in Chapter 8.6. Here, the components are not space filling cells, but planar squares. The recursive logic of the system is comparable to Aristid Lindenmayer's L-systems, which are briefly described in the following section. Lindenmayer developed algorithmic principles for simulating plant growth in 1968 which are unfolded in *The Algorithmic Beauty of Plants*.[11] One of the fundamental examples described in the book is the context-independent and deterministic DOL-system. It is a parallel rewriting system, where each letter in a string represents a rewriting rule. The system is iteratively updated, and for each generation each letter is replaced with alternative letters or letter sequences, as shown in Figure 20. Each state of the replacement sequence is represented as a string, where each character in the string corresponds with a future action, or production. The action may imply the drawing of a line segment or moving the drawing position. All of the actions in a line are processed in parallel, meaning that the whole system is updated in one step. Each letter, representing an action, is replaced by new letters, which represent new actions. The number of actions is limited, and the complexity of the result is dependent on the recursive subdivision and scaling of the components. Figure 21 shows the basic rewriting principle in relation to simulation of plant structures. Figure 23 illustrates how the logic may be used to direct other forms of drawing actions through *turtle interpretation*. The basic operation of the turtle resembles the logic used in the *SAGA* project, which will be discussed further in the following section.

11 P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*, Springer-Verlag, New York, 1990

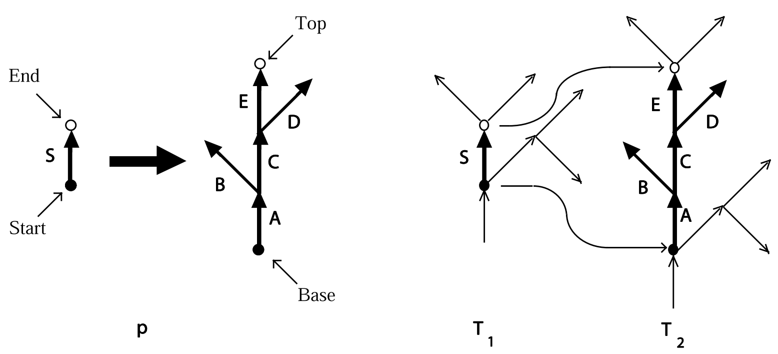


Figure 21: The drawing shows how the line S is subdivided. Illustrations by Prusinkiewicz/Lindenmayer

Generations: 5
Rotation angle: 25.7º
Initial set : F
Rule: F -> F[+F]F[-F]F

Generations: 5
Rotation angle: 20º
Initial set : F
Rule: F -> F[+F]F[-F]F

Generations: 5
Rotation angle: 22.5º
Initial set : F
Rule: F -> FF-[-F+F+F]+[+F-F-F]

The initial line is replaced with 5 shorter line segments of which two are rotated. The square brackets indicates that the drawing position is the same for the next action outside the brackets (branching).

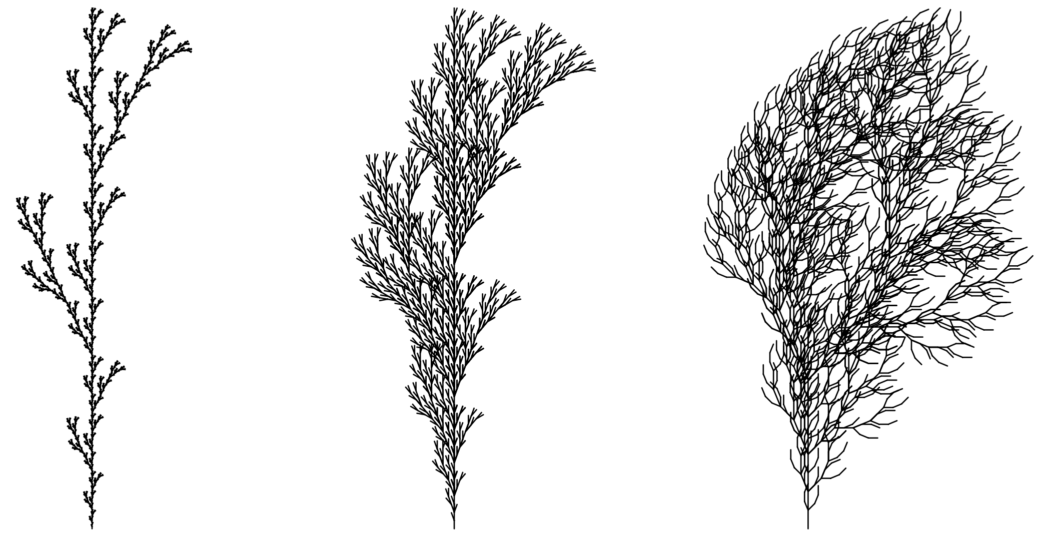


Figure 22: Examples of plant-like branching structures generated by DOL-systems. Illustrations by Prusinkiewicz/Lindenmayer

6.10 Recursive growth logic

Similar to the way *Agent Construction* and *Solar DLA System* are related, *SAGA* consists of an algorithmic method and a physical experiment. Of the methods described in this thesis, *SAGA* was the

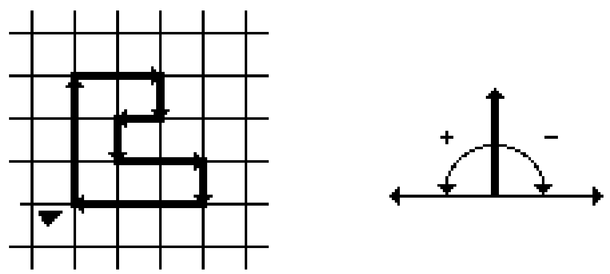


Figure 23: With so-called turtle interpretation, the letters are linked with drawing actions. F means move forward and +/- determines which side to turn to. Illustrations by Prusinkiewicz/Lindenmayer

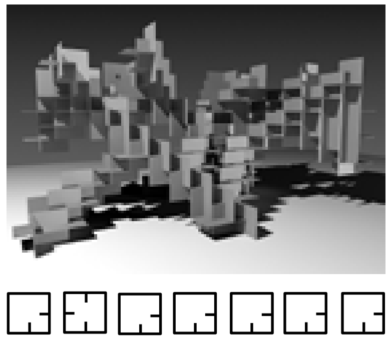


Figure 24: SAGA, algorithmic simulation. Growth sequence 1511111, 30 generations, 272 objects.

earliest developed. The generative principle is a recursive algorithm. This is linked directly with a geometrically defined component, that is, a planar square. In an abstract sense, the system treats one or more initial components as input for the recursive process. The component then 'copies' itself to an adjacent position according to a growth sequence. The sequence defines the new component positions and orients itself. Constraining these operations leads to seven possible operations. An example of this is shown in Figure 24. The first 'copy' positions itself in accordance with the 1st position in the growth sequence. When this new component 'copies' itself in the second generation of the recursive process, this operation is defined by the second step in the growth sequence. Figure 24 demonstrates this step, where branches grow in two directions: forward and left turn. These new 'copies' then continue from step three in the growth sequence and so forth. The growth sequence is comparable to the strings in DOL-systems, but the operations are not replaced in the same way as with the rewriting systems. The recursive approach here is extensive and does not address depth in scale as with L-systems, which can be used to generate fractal patterns. The *SAGA* method was not developed to a point were architectural potential was directly suggested. Still, the project helped to illustrate some aspects. At the algorithmic level it was demonstrated that a complex spatial structure could be generated from a very basic rule set. It was also shown that a large degree of variation in the pattern could be achieved through changes in the growth sequence. Realisation of the physical model helped to point out the necessity of feedback between different levels of the realisation process, which was not pursued in the experiment, but in the development of the *Concrete Gridshell Pavilion* in 2011. Before the physical *SAGA* model is discussed, a few topics related to the algorithmic principle will be explored.

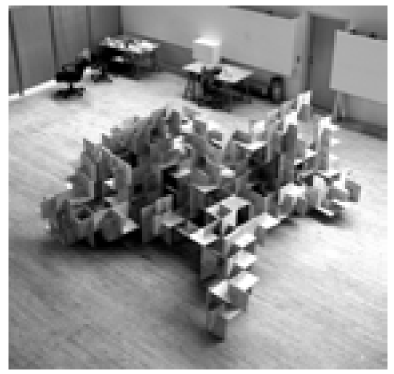
There are both similarities and differences between the recursive logic and the DLA-algorithm previously mentioned. Both types deal with a form of extensive growth. In the discussed examples, the design space is defined as a three-dimensional lattice. It is possible to implement versions without this constraint, but the lattice improves performance and ensures geometric and topological consistency, since only specific relations between components are possible. A notable difference between the two types is that the growth from recursion can be understood as internal in the sense that new components are positioned directly adjacent to the component in the previous generation. DLA growth is essentially generated externally. As explained previously, a *random particle* searches through the design space until it 'hits' the existing structure, and places a new component. This has two consequences. One is that the new components are randomly positioned and are unrelated to growth generations. Another consequence is that the result is a characteristically organic and open pattern, due to the fact that it is more difficult for the *random particle* to reach the inner parts of the structure without hitting the outer parts first. While the DLA is stochastic, the recursive algorithm used in the *SAGA* simulation is deterministic. Here, it is relevant to discuss the matter of randomness with respect to generative design.

6.11 Randomness and realisation

Randomness is not necessarily equivalent to unpredictability. Within the *Solar DLA System*, the random function is a means for using probability as part of the logic. Other calculation methods may create similar results, for instance by balancing local decision through counting. However, this would result in less versatile and more complicated logic. The random function makes it possible to adjust the probability, and thereby the pattern formation, seamlessly. Here, it is also worth noting that the randomness does not lead to an unpredictable pattern. The exact geometry of the pattern is unpredictable, but because the randomness is directed towards tendencies, the overall result is relatively unchanged between test runs if the parameters are unchanged. This could be compared to *SAGA*, where randomness is not part of the system. Needless to say, the geometry of the generated pattern is completely identical between test runs. However, if a single step in the growth sequence is changed, a completely different result may occur, as documented in Figures 1–7 in Chapter 8.5. While, the algorithm in the DLA example is hardly linear, when the parameters are gradually changed the pattern formation changes accordingly. The unpredictability lies at the level of detail, while the overall character of the pattern is more or less predictable.

If we take a look at the realised physical *SAGA* model, shown in Figure 25, a large amount of randomness was accepted. This approach is similar to the *Agent Construction* model, as the builders had control over the structure during its construction. In the *SAGA* project, however, the process was more constrained by the component logic. Only a few connection possibilities existed, and only one type of component was able to create branches. The idea was that the constraints would lead to 'self-organised' pattern formation, similar to the algorithmically generated examples. Despite these constraints, the realised model had a character that was more random than algorithmic. The building process demonstrated how actual building entails an exponential increase in complexity, even with the simplest tectonic principle. Some connections were only possible if part of the existing structure were detached. Tolerances, at times below 0.5 mm, played a major role as tight fits assisted

Figure 25: SAGA, realised model.



in stabilising the structure. This was problematic at times in terms of assembly. Concerning stability, it was necessary to ensure that all parts of the structure were supported during construction. Some 'bridges' in the structure were only possible because the builders intermediately held part of the structure while building. These issues could have been improved through iterative development of tectonics and algorithmic logic, i.e., the establishment of feedback loops between realisation and generative logic. But this approach was not explored in the early experiments. Contrastingly, it is worth pointing to the *Watercube*, which was developed on different levels simultaneously. Generative techniques enabled the handling of numerous discrete components necessary for realisation. The development process was influenced by feedback and parallel adjustments concerning detailing, performance and aesthetics. In this sense, the *Watercube* relates more to the *Concrete Gridshell Pavilion*, which also demonstrates a form of cyclic development process.

6.12 Aggregate growth: conclusion

If we compare the examples in the chapter with those previously discussed, the emphasis here is on the properties of the cells or the components, and less on topology and morphology. For instance, the surface of the *Watercube* appears through a secondary step. It emerges from cuts through the predefined cell pattern, and is subsequently refined through the development of a tectonic solution. The bottom-up logic of the *Solar DLA System* does not generate a complete surface, even though the result can be a protecting layer. The *Watercube* serves as an example of a realised building where a tectonic pattern forms the basis for a unique architectural expression and a rational construction. Although the project does not share the concept of growth with the research examples, there are similarities in the use of three-dimensional cellular organisation. The project demonstrates how generative techniques assist in establishing feedback loops that inform the realisation process. This type of optimization of the result is possible because a large part of the process is taken into account at an early stage. The generative tools make it possible to explore a range of solutions concerning manufacturing and construction. This project demonstrates how the linear hierarchy that usually dominates the project development can be transformed into a more holistic approach. Some of the scientific topics discussed are not directly implemented in the discussed methods. The topics occur as inspirational examples rather than scientific explanations that enlighten the functionality of the methods. In this sense, they facilitate how science can inform design thinking in an indirect way. Similarly, the design of the *Watercube* was originally inspired from studies of soap bubbles, and subsequently realised through the use of a regular geometrical principle.

SAGA demonstrates the potential of algorithmic logic on a basic level, and the physical realisation underlines the necessity of addressing tectonics as part of the development. This issue was not thoroughly pursued until the later development of *Complex Gridshell*. *Solar DLA System* shows how growth logic can be used to generate a pattern the represents a negotiation, or balancing, between internal logic and external parameters, that is, protecting areas against solar radiation. Both projects raise the question of control and predictability with respect to generative design. Particularly in terms of realisation it is necessary for the designer to be able to control the geometry completely. However, during the development process the question of control is more nuanced. Here, the question is perhaps oriented towards predictability. The examples that have been explored demonstrate how an essentially random system, the *Solar DLA System*, is more controllable in terms of adjusting the outcome than the deterministic system, *SAGA*. The design process can be stimulated through experimentation with unpredictable algorithms. However, at some point, the designer must be able to adjust the behaviour of the system towards a specific outcome. Here, it is often more important to be able to control the overall character of the pattern and the tectonic principle, rather than the exact positioning of individual elements. Therefore, a nonlinear adjustable pattern can be more useful than a deterministic pattern where changes have a strong impact on the outcome.

A generative process can be controlled on three different levels. The first level is the change of algorithmic logic. The second is the change of parameters that affect the form-generation. And the third is direct 'manual' intervention. The process of constructing the physical model of *SAGA* demonstrates a system with a high degree of local intervention. The builders were able to adjust the growth process to achieve certain properties. The system allows intentions related to local conditions in the structure to affect the pattern formation. This was also the case in the *Agent Construction* model, since the construction process was based entirely on local conditions and the builders' intentions. In an architectural project, this type of intervention is often necessary because exceptions to the rules occur as part of the design development. When an algorithmic logic drives the tectonic solution, it is an advantage if the designer can step directly in and adjust specific parts of the design without having to develop exceptions in the generative code. The *Watercube* is based on rigid periodic geometry, which means that local adjustments would have demanded the pattern to change at large, or that exceptions would have to be embedded in the generative methods. Within *Solar DLA System* a certain degree of local adjustment is possible through the adjustment of attractors, which could be linked with certain behaviours. However, it is impossible to directly specify the shaping of the pattern on a local level. A similar approach was discussed in the previous chapter in relation to Kokkugia's *Busan Opera* project. Here, the large part of the geometry could be 'frozen' while a small part is restructured through a local generative process. Generally, it is possible to 'manually' change the geometry of a project at a certain stage of the development process, at least after the final steps in the generative process. However, this situation also marks a break in the flow of the system. New versions of the resultant geometry must relate to the 'manual' intervention, or the process must be repeated.

7 Conclusion

7.1 Introduction to the conclusion

The conclusion discusses the outcome and the scope of this research, both in terms of research methodology and more detailed aspects concerning the developed methods. Essential questions raised in the introduction will be revisited. The process of developing and trialling methods based on generative techniques has revealed a series of potentials and implications, thus aiding the discussion of the topics. The potential for using generative techniques in architectural design remained as an anchor point for the research. As, there is not a definitive response to the research topic, this thesis provides alternative methodological perspectives. The conclusion addresses first contributions, scope and possible future developments directly related to this research. The methodology is observed in terms of approaching the field of generative techniques in architecture. Chapters 7.4–7.7 contain a broader discussion based on the research outcome, a form of outlook. The topics complexity and negotiations in the form-generating process are discussed, as well as the matter of controlling the geometry. It is discussed how new types of hierarchies emerge as a consequence of the methods. Then follows a discussion of the relation between design and science and considerations regarding the role of the architect.

7.2 Contributions

This research makes contributions on different levels. On a general level, this thesis discusses generative strategies from the perspective of architectural design. Therefore, it is structured from a geometrical and architectural viewpoint, discussing a range of topics based in architectural problems. On a specific level, methods directed towards different problem types have been developed. Three methods are mentioned in particular. *Complex Gridshell* demonstrates a holistic design process, based on generative techniques, *Self-organising Surface* addresses the problem of directly controlling complex geometry, and *Solar DLA System* shows an example of integration of contextual parameters. These methods relate to various aspects of algorithmic design, which will be discussed in the following section. Initially, the contribution of this thesis from an overall perspective is discussed.

This thesis seeks to unfold generative potential from the viewpoint of architecture rather than the viewpoint of computer-science. Although the main emphasis is on organisational logic, methods are discussed and explained from the perspective of architectural design. These methods are arranged in the thesis to represent a different fields, ranging from two-dimensional geometries to procedures that are inherently three-dimensional. Even though the method development was not initially focused on covering a spectrum of geometric problems, this approach to scrutinising and discussing the methods proved beneficial in terms of maintaining an architecturally oriented discourse. The ordering principle that places the method in a spectrum ranging from two-dimensional surface morphology to three-dimensional spatial aggregation emphasises the methods' potential within the realm of architectural design. Another approach may have been to prioritise the underlying logic of the system techniques and use it as a structuring method for the thesis and discussion. However, this approach would have privileged the overview in terms of algorithmic logic over geometrical and architectural topics. The following discussion draws attention to specific methods that demonstrate various potentials of generative techniques.

Complex Gridshell serves as an example of a method developed through cross-disciplinary collaboration and establishment of feedback loops. The method is highly dependent on custom-made generative tools that were configured as part of the development process. Constraints regarding production were taken into account in the form-generating process, and all parts of the system were developed simultaneously. Digital information was exchanged between generative tools and structural analysis software. As a whole, the system was conceived with a holistic approach.

Self-organising Surface addresses the problem of controlling the geometry without predefined topological relations. Generation of complex surface shapes generally demands either a priori topology or use of a secondary algorithm, such as marching cubes, to extract consistent surface geometry. Algorithms for extracting geometry pose constraints on the outcome, thereby limiting direct control of the geometry from within the form-generating logic. The technical issues are explained in Chapter 8.4, but here it can be said that *Self-organising Surface* allows complete bottom-up generation of surface topology through agent-based logic.

Solar DLA System is an example of a method where iterative negotiations between internal logic and external parameters are embedded within the generative logic. An algorithm that generates a distinct spatial pattern is linked with sun path simulation. Through specification of areas in the design space that are to be protected and a degree of protection, the growth process generates a solar protection wall with differentiated density. Some of the topics related to these methods are further discussed in a general perspective in Chapters 7.4–7.7.

7.3 Methodology, scope and future research

As emphasised in the introduction, the research approach was experimental, rather than analytical. This experimental approach has revealed a variety of insights into the potentials of and implications concerning generative techniques. It can be difficult to detect the strength of a method from a theoretical perspective. For each method there are often limitations and problems related to geometric consistency, which are seldom entirely explained in project presentation. These aspects are influenced by the fact that the techniques involved are relatively untested in actual building projects. It has been discussed for each method in the thesis to what extent it is based on well-known principles. Generally, new mechanisms or combinations have been implemented for each method. Another insight, gained from expanding the development of original methods, is related to the integration of method development and design development. Without directly engaging in the method development it would have been difficult to understand the connection between design and system logic that is crucial to generative design. The amounts of technical challenges and use of time resources superseded the estimations. These aspects should also be taken into account when discussing the potential of the methods investigated. This is often inadequately communicated in published projects. The goal of the project was not to arrive at a new theory concerning the field of digital tectonics or generative techniques. However, by gaining insight in the underlying logic of the techniques involved, the researcher established grounds for discussing the topics in general. Considering the experimental approach, the research covers a relatively large range of methods, whereas, it could be argued that each of the methods has not been substantially developed. It can be questioned whether the broadness of the field of methods compared to scope of development has been optimally balanced. Since some methods, such as the agent-based systems, have proved more versatile than others, the researcher could have potentially prioritised these elements differently in hindsight. However, the investigation of a range of methods provided means for comparison and discussion across the methods with respect to their relevance and limitations in relation to geometry and architecture. This underpinned the discussion of a design methodology, based on generative techniques.

Another issue of concern is the limited extent of experiments in physical models. As the development of the *Complex Gridshell* method shows, much knowledge is gained through these kinds of experiments. However, this thesis was mainly focused on developing the internal logic of the methods, particularly those concerning self-organisation and principles for implementing negotiations as part of the form-generation process. Arriving at a consistent geometry proved to be challenging in many cases. These experiments would have benefitted from further exploration through physical experimentation in order to gain more knowledge on the integration of digital production techniques. This approach contains much potential for future research.

This thesis raises some questions concerning the future potential of generative techniques. Future research may provide answers to these questions. The use of tectonic patterns as a main design driver played an important role in initiating this research project. Generative and digital production techniques may help to support this procedure and promote a bottom-up oriented approach to architectural design. Although the internal logic of algorithmic principles and the establishment of digital negotiations have previously been explored, how these methods help to create actual tectonic solutions is currently a focus for research within the field. Future research could help to unfold this potential, and a variety of approaches should be considered. One approach would be to generate tectonic systems based on algorithmic methods, and present them as examples of architectural design in the virtual realm. This would provide the basis for a discussion of the relevance of the methods. However, it would be more rewarding to approach the development through a material based investigation. As demonstrated in this thesis, physical experimentation reveals a variety of implications that guide the development of the methods even if the materials used differ from the material that would be used in the actual project. This is due to the increase in complexity involved in manufacturing and construction processes, compared to visualising the result. Recent publications show that these types of experiments are being carried out in a number of institutions. An example is the *ICD/ITKE* research pavilions constructed at Stuttgart University.[1] Another example is *Fab Lab House* produced at the Institute for Advanced Architecture of Catalonia as an entry into the 2010 Solar Decathlon Europe competition.[2] It is interesting to note that because this approach is new to architectural practice, these methods are being developed in the academic realm. Only a small number of examples exist where the digital production techniques are fully exploited in realised architectural projects. It seems only natural for these tools to be developed, investigated and tested on a smaller scale before they are applied to 'real' architecture, where the economic and practical implications must be fully controlled.

1 'ICD/ITKE Research Pavilion 2011', viewed 11 July 2012, <*http://icd.uni-stuttgart.de/?p=6553*>

2 'Solar Energy Design The Fab Lab House By IAAC', viewed 11 July 2012, <*http://www.kubodo.com/2011/03/06/solar-energy-design-the-fab-lab-house-by-iaac/*>

Therefore, these developments are a natural subject for future research.

Another subject that holds potential for future research is the integration of negotiations between different contextual parameters and design intents embedded within generative logics in the form-generation process. This research project only briefly acknowledges the subject, which confirms the potential for further exploration. This relates to matters of optimisation and sustainability. The researcher suggests the possibility of enabling complex negotiations where issues concerning material use and performance play a role with overall design intent that can reflect programmatic or aesthetic parameters. These investigations can be developed to a certain extent within a virtual environment. However, the actual materials and physical contexts of a model scale provide a more optimal basis for evaluating the outcome of the methods before realisation. Ultimately, issues concerning contextual parameters and the focus on tectonic patterns would be best investigated within an architectural project.[3] This kind of project often involves both practitioners and researchers, similarly to the architectural examples explored in the thesis.

7.4 Complexity and iterative negotiations

In the introduction it is suggested that generative techniques provide means for addressing complexity within architectural design. One type of complexity is concerned with geometry. The method *Complex Gridshell* (described in Chapter 8.1) demonstrates how a complex structure can be realised through the use of digital form-finding and scripting tools in 3D modelling software. The method *Branching Topologies* enables the generation of complex shapes as well as self-organised topologies. The algorithmic control of the geometry allows both complex shapes and differentiated tectonic articulation. This type of control is strongly expressed where digital production techniques can be used for the realisation of the project, as with the production of discrete components in the case study *Concrete Gridshell Pavilion*.

Another form of complexity is the possibility of establishing continuous negotiations between design parameters within the form-generating process. Parameters, which are normally controlled through trial and error experimentation, can be embedded in the form-generating process. Thus, a continuous flow of information from design intent to realisation emerges. Designers generally have to allocate a few components as key factors in the design process,

3 It is planned to further develop the Solar DLA System in relation to research in daylight undertaken by Carlo Volf.

leaving other parameters to become secondary. This usually occurs in an interchanging process where the hierarchies have the potential to change. The methods described in this thesis suggest other ways of balancing these influences. Rather than choosing primary parameters, a field of negotiations is established and utilised in the form-generating process. Through the establishment of the system, the designer becomes aware of priorities and indirectly guides the system towards a useful solution. Simulations and data regarding the project's context, internal algorithmic logic and constraints from manufacturing and construction can be embedded in iterative complex negotiations. This perspective is demonstrated in *Solar DLA System*, described in Chapter 8.6. In this example, the 'growing' structure responds to sun radiation amount and degree of protection of predefined areas.

Even if the result is not necessarily complex, it may represent the outcome of numerous complex negotiations between essential parameters, such as performance, realisation aspects and design intents. The method *Complex Gridshell* was guided by performance criteria through dynamic relaxation. The realisation aspects were crucial parameters in terms of setting the limits for component size and complexity. Negotiations rooted in design intent are present in more of the examples, produced with the methods. For instance, with the *Branching Topologies*, the research demonstrates how the regulation of parameters and re-configuration of logic can guide the system towards varying outcomes. More importantly, the design approach within this research demonstrates how design intents and the development of algorithmic logic can be considered equal aspects of the design process. Although very few examples are directed towards actual design solutions, the methods are generally demonstrated at a level where utilisation within architectural practice is suggested. For instance the methods *Self-organising Bezier Curves* and *Branching Topologies* demonstrate that the design solution and underlying algorithmic logic are fundamentally connected. During the development of both methods, the character of the outcome and the logic of the system emerged simultaneously.

The examples in this thesis demonstrate negotiation of geometrical and contextual parameters within the form-generating process. In some cases, such as the agent-based flocking algorithm in Chapter 8.4.2, a mapping of the produced states eventually forms the geometry. Similarly to the aggregate systems described in Chapters 8.5 and 8.6, every state adds new geometry to the field. In other cases, the system is constructed to arrive at a stable state, which then forms the result. This is the case with both *Complex Gridshell* and *Self-organising Bezier Curves*. Even the agent-based method *Self-organising Surface* in Chapter 8.4.9 moves towards an equilibrium state, though highly dependent on specific conditions.

7.5 Addressing the rules

Standard algorithms for extracting states of geometry changing from one form to another, like Voronoi or isosurfacing, can be compared with tools in animation software, such as the visualisation of fluid dynamics. This research does not stand in opposition to these methods, which is indicated by the use of standardised isosurfacing software for *Branching Topologies* in Chapter 8.3, but demonstrates the advantages of being able to access the form-generating rules. First of all, the methods developed in the research are generally not based on predefined tools, but rather conceived as original, in the same way as a particular design solution is original. With predefined tools, it is impossible to fundamentally change the tool in order to arrive at a different result. This is also a reason for many digital designs having similar appearance. The underlying algorithmic principles may result in recognisable characteristic types of formations and geometries. An appropriate response to this dilemma would be to consider a specific design problem and deliberate whether a type of algorithmic method is appropriate. The algorithmic tool would then be rebuilt in order to reflect the design intent and relevant contextual parameters. Because the internal logic of the system is addressed during its development and adjustment stages, the logic can be extracted and utilised for the realisation of the project. Animation software is generally structured towards the creation of a trustworthy visual representation and not towards the definition of exact geometry with respect to physical realisation. As mentioned in Chapter 4.2, not all examples in this research coincide with this idea of consistency. These cases can be said to lack complete control of the internal logic. However, it is possible to establish connections if such direct links should turn out to be crucial to specific projects. Concerning the isosurfacing tool, it is possible to either implement a custom written isosurfacing algorithm, thereby directly controlling the outcome, or simply develop a different way of generating the geometry. In some cases it is preferable for the generative process to be divided into separate stages. A set of conclusive information is collected at the end of each stage and transferred to the beginning of the next. This ensures that if a part of the system needs alteration, the whole process does not necessarily have to be restarted. Unfortunately there is a danger of loosing the potential for making changes to the preliminary form-generation if the initial conditions for the system are lost. This became clear with *Concrete Gridshell Pavilion*, since the initial settings were lost. It was impossible to reconstruct the exact form-generating process. Fortunately the generated geometry was saved, and since this was used as input for subsequent processes, the loss was less dramatic. Thus, the example also demonstrates the advantage of stages in the generative process.

7.6 Feedback loops and hierarchies

One of the issues that is addressed in this thesis is the possibility of establishing new kinds of feedback loops in the project development process through the use of generative tools. The flow of information is simultaneously a flow of materials in cases where actual production is embedded within the design process. *Concrete Gridshell Pavilion* showed how a system of this nature can be established on a small scale. Feedback loops enabled a cyclic development process. Each part of the system, from the initial conditions to the realisation, is accessible to the architect. Information, which is linked with later stages of the realisation process, can feed back to earlier stages. This information is then embedded directly in the generative process. Although this occurs within typical design processes, knowledge is imparted to subsequent projects, rather than affecting the current project. Specific information concerning realisation is taken into account but seldom leads to a reconfiguration of the initial design. With concern to generative techniques, feedback loops in the design process exist at several levels. As demonstrated with agent-based systems in Chapter 8.4, the development of the system is based on a feedback loop, cycling between re-configuring the internal logic, adjusting parameters and analysing the output from the system. This is not a purely technical or performance oriented process. Aesthetics and programmatic issues are equally important, particularly when addressing a design solution and not merely basic functionality. Furthermore, the potential of the method is gradually revealed during the system development, which supports an approach where system development is an integrated part of the actual design process.

The approach to the design process that is reflected in this thesis, questions the prevailing hierarchies within architectural methodologies. The linear sequence that characterises most architectural processes also poses a hierarchy that exists on the level of project development. Large-scale design is generally fixed from the early stages of the design process. Subsequently, major elements are defined, and precise decisions concerning materials and details are made. In many cases the process is divided into separate stages, enabling information concerning the more detailed realisation impact the initial design. As previously mentioned, the use of generative techniques suggests a less hierarchical process, which allows different scales of the project to play a role from the beginning. *Complex Gridshell* demonstrates this most coherently as all of the steps in the realisation process are embedded in the design process. The approach is present in all of the methods, as the internal logic would have to be adapted for particular conditions and realisation techniques. Again, this adaptation should be considered as an integrated part of the design development. The question of hierarchy also refers to negotiation in the form-generating process. Architecture has traditionally reflected hierarchical relations in many ways. This can be seen as hierarchy between structure and cladding, envelope and programme, material and form and many others. The use of generative techniques suggests a different understanding of 'hierarchy.' Rather than being imposed, hierarchies emerge within the form-generating process as the result of numerous iterative negotiations, as discussed in Chapter 7.4.

This research project demonstrates the possibility of incorporating factors in the initial stages of development that are normally only considered in the phases close to the realisation of the project. With *Complex Gridshell*, these factors are an integrated part of the form-generation process. A tectonic principle is developed and informs the configuration of the form-generating system. The designer addresses both the design and the realisation processes, dealing simultaneously with all scales and levels of information. As such, the architect is very close to the centre of the project development, as suggested in the introduction. Naturally, the case studies do not reflect the realisation of a building project, as only a basic structures in a small scale was produced. Architects are generally unable to adopt experiments with actual production, let alone construction. However, the cyclic project development has potential to be transferred to projects of larger complexity, involving many factors, without loosing the benefits of a holistic approach. This again would raise important questions concerning economical structures and priorities set by clients and other influential parties. As stated by Phillip G Bernstein: 'The motivations are apparent and the tools are evolving rapidly, but the underlying business models needs to continue to change to make further process improvements possible.'[4] Architects often deal with large scale and programmatic issues initially. Appearance, realisation time and costs also play crucial roles, and issues concerning sustainability and performance are increasingly becoming important parameters within architectural design. Many of these issues must be dealt with through a bottom-up oriented approach, allowing all aspects to affect the overall design at an early stage. These viewpoints relate to this research, although it should be mentioned that further implementation in architectural practice has not been studied.

---

4 Phillip G. Bernstein, 'Models for practice: past, present, future', in P. Deamer & P. G. Bernstein (eds), *Building (in) the future*, Princeton Architectural Press, New York, 2010, page 196.

7.7 Methods and science

Architects make use of algorithmic principles with aims that differs from scientists and computer programmers. Science is mainly occupied with studying natural phenomena, and programmers are concerned with implementation strategies. Architects' use of generative techniques is fundamentally different. Here, the goal is not to reflect an existing reality, but to generate a new reality, and the character of the outcome emerges through the development of the system. Geometrical properties of the outcome are crucial, since these define how the algorithmic principles are transferred to architecture. While, architects can seek complex form-generation, it is essential that the geometric outcome is controllable. Some of the developed methods address this problem. *Complex Gridshell* demonstrates a process where the code that defines the geometry can be accessed in all parts of generative process. That is, the mesh geometry, detailing of the components and the template production. Another type of geometrical control is addressed in *Self-organising Surface*, where the goal is self-organised topology.

This thesis articulates various methods that are developed with a basis in science. As suggested in the introduction, the methods were not used as simulations of natural phenomena, but as tools for generating complex self-organising patterns and spatial structures. During the development of these experiments, the possibility of using generative logic within these systems (even without gaining deep understanding of their relations with natural phenomena) was realised. This underpins the relevance for architects to engage with scientific methods during the creative process. It is crucial, however, to maintain a clear distinction between different types of use. In some examples, actual simulations of natural phenomena were embedded. These examples illustrate how physical phenomena can be interlinked with the internal logic of a generative system. In recent years, digital simulation tools used for the analysis of performance and negotiating the effects of specific solutions have become more common in architectural practice. The tools are normally used to evaluate a well-specified design, but the generative approach suggests ways of embedding these analytic tools in the form-generating process. With regard to the 'home-made' simulations that are reflected in this project, a reasonable approach would be to combine form-generation with a subsequent analysis of well-tested digital, physical, or environmental factors, in order to substantiate the performance of the solution. Optimisation was not a primary focus of this project. Within the realm of architecture, there usually is not a single primary parameter, but rather a series of parameters with varying amounts of importance that must be balanced against each other. Therefore, the research was oriented towards showing negotiations between different external parameters and different internal rules as part of the form-generating process. For instance, sun path simulation was used as a generative parameter to construct a sun protection structure, and the simulation of spring properties with respect to structural stability was used in the examples of dynamic relaxation. Furthermore, this research suggests a holistic understanding of optimisation, which acknowledges that generative techniques entail ways of reducing material use and production time.

7.8 Generative techniques and the role of the architect

As suggested in the previous section, some of the initial viewpoints have been consolidated through the demonstration of methods. Subsequently, a general discussion of the role of the architect and the practical uses of the methods is formed. Use of generative techniques can affect the architect's position in the process from design to realisation in different ways. As stressed in the thesis, the design process is significantly changed through use of generative techniques, mainly due to the focus on the generative mechanisms. Essentially, this is an internal shift of design methodology in terms of discussing the profession. A more interesting perspective emerges when looking at generative tools as a means for establishing cyclic development where realisation constraints affect the form-generation. Here, the architect's position can become central in the project development to larger degree than in typical processes. This is where the architect controls the information flow directly to building component manufacturers and contractors. Another perspective on the same situation is to consider the architect as part of a larger collaboration with common goals. A team of different professionals participate in the project development in all phases. In this view, the architect renounces control over the initial design in order to achieve higher quality in terms of both performance and aesthetic properties.

It is stated in the introduction that when using generative techniques, the designer's attention is less directed towards the final result than towards the system. This is certainly true for the sake of the researcher who, for many of the presented examples, spent weeks on developing the internal logic of the algorithm. Besides time, resources and interest, knowledge of the fields of programming and mathematics is necessary. In terms of practicing the approach, these resources must be taken into account. In case an already developed method can be used for a specific solution, the development time is remarkably shorter. However, it is also stated that the system itself, or at least a modification of it, should be a part of the design process. Through the development of various processes and subsequent experiments, the researcher concludes that this statement is verified. Some of the methods are relatively generic, however, and have the potential to serve as starting points for bespoke algorithmic solutions. This intermediate approach (where the basic behaviour of the system is well known, but the specific articulation is part of the design process) is probably most relevant to architectural design. An example of this situation can be seen through the link between *Self-organising Bezier Curves* and *Branching Topologies*, where one method forms the basis for another. A question is, how far it is relevant for architects to take on the role as computational expert, or just to learn about the techniques? As Rob Stuart Smith expresses, 'When generative techniques are used as an integrated part of a design solution, it is a task for the designer to develop the logic of the system, as part of the design process.'[5] The designer must have a basic knowledge of the structure and functionality of the code, even if a different person is responsible for its implementation. In reality, when dealing with generative systems, the system forms gradually during the design process, in parallel with the development of the actual design proposal. Therefore, it is crucial that the designer is able to control both the outcome and the logic behind the system. Another aspect of this issue is that a large number of architectural design methods are not originally developed for a design purpose. Rather, they come from science as a simulation of natural phenomena. To modify or reconstruct these methods for design purposes, designers must engage with the process, in order to arrive at a useful result. Again, this also suggests the relevance of designers engaging in computer programming.

Digital processes have already been proliferated within industries and are likely to become dominant in the same way that computer technology has become the central tool for architectural design. It must be considered, however, whether or not the control of digital production techniques should be a challenge for the architect. In principle, it may remain as a subject for the manufacturer to choose the most suitable production technique, in order to arrive at a result. However, the researcher is convinced that these new production techniques have the potential to propose new types of tectonic principles, and that these types are best discovered through experimenting and engaging with techniques. This approach is not dissimilar to the way experiments with new materials and production techniques were performed at the Bauhaus school in the 1920s in the search for new methodologies. An example of handling the process of architectural realisation in its entirety (from design to fabrication and construction) is Gramazio & Kohler at the ETH in Zürich. They have performed a series of experiments with robotic fabrication, from designing the algorithmic logic to actual construction. One of their primary concerns is that these techniques are already becoming part of the building industry, so it is important for designers to develop tectonic strategies that make use of the technological potential.

5 Robert Stuart-Smith, *personal communication*, 9 May 2012

In relation to more complex projects, it is evident that the architects must be able to use scripting tools and parametric models of the project, both during the development process, and for generating the necessary information for production and construction. This approach is predominantly concerned with managing representational information and not with bottom-up ways of arriving at tectonic solutions or architectural expressions. Here, the researcher proposes that the generative techniques provide a means for dealing with higher levels of complexity in the design process, and at the same time, gaining access to a larger vocabulary of architectural expression. Furthermore, a series of possibilities for optimising solutions can be embedded. To return to the role of the architect, it seems relevant that architects to some extent engage with both generative techniques and digital production. How deep this engagement must be in order to be productive is perhaps still a notable question. A realistic model could be to instigate a cross-disciplinary collaboration between architects, engineers, programmers, mathematicians and other experts. However, it can be assumed that the degree of technical engagement is somewhat reliant on the degree of influence, on behalf of the architect. As previously stated, these new tools support a redefinition of the architect as the project's central facilitator, but only to the extent that the architect is able to control the system and its functionality. Less ideologically, in order to be able to realise certain complex solutions, the architect must be able to specify how the manufacturing and construction is possible.

This research seeks to embrace digital technologies with an approach that can reinstate craft-based knowledge as essential in architectural design. The primary objective of this thesis is to support a bottom-up approach for arriving at new form generating techniques, not as a form of imagery but as an integration of material properties, performance and design intents. As a result, contextual, programmatic and aesthetic parameters are equally balanced. This research seeks to 'renounce the split between art and science', as Kenneth Frampton puts it.[6] He is referring to integration of craft, research and design, not to digital tools, which he is sceptical about in terms of design methodology. However, this research suggests that generative techniques can underpin a holistic approach to architectural design. Algorithmic principles for organising matter lead towards a new understanding of tectonic hierarchies, thereby demonstrating another type of exchange of knowledge between science and art.

6 Kenneth Frampton, 'Intention, Craft and Rationality' in Peggy Deamer and Phillip G. Bernstein, op. cit., page 35

8 Method descriptions and experiments

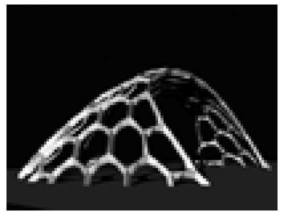
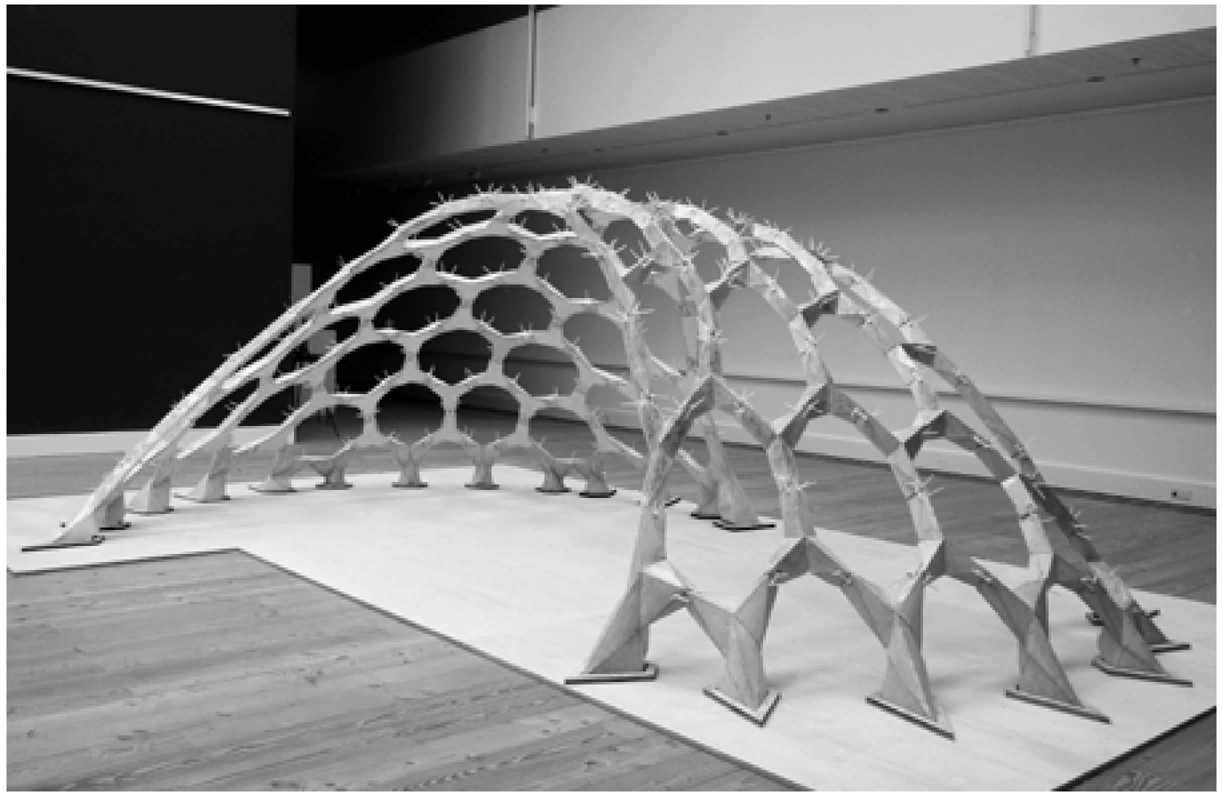
	**Intent and conditions**	**Organisational logic**	**Realisation**
Properties	Self-organisation through dynamic relaxation. Mass-customisation Discrete components Cyclic development Structural analysis	Digital/physical integration Feedback loops Dynamic relaxation Bespoke digital procedures Information flows in relation to production and assembly.	Sustainable concrete casting Component-based structure Laser cutting Folded PETG Templates Precision and tagging 1:1 scale case studies Temporary construction.
Observations	Flexibility Variation Complexity Flexibility Optimisation Sustainability	Digital flow Translation Structural analysis Automation	Material properties Tolerances Scalability Components Human labour

Figure 1: The Concrete Gridshell Pavilion at Aarhus School of Architecture, October 2011.



8.1 Complex Gridshell [1]

8.1.1 Intents and conditions

This method was developed by Ole Egholm Pedersen, Dave Pigram and the author, Niels Martin Larsen, as part of collaboration between Aarhus School of Architecture and the University of Technology in Sydney. One of the goals was to investigate how digital production technology and material properties can be interlinked through feedback loops in the architectural design process. Another goal was to investigate the possibility of realising a structure of advanced geometry through component-based construction. Furthermore, the method was expected to demonstrate new sustainable techniques for casting distinct concrete elements. The prerequisites for the project consists mainly of two parts: At the level of production technique, a number of experiments concerning concrete casting was carried out by Ole E. Pedersen in the beginning of 2011 at University of Technology in Sydney. A viable method based on concrete casting in PTEG sheet was developed, and forms the basis for the casting techniques that are used in the method described here. The second part is an application for simulating dynamic relaxation, previously developed by Dave Pigram and Iain Maxwell. The challenge was to develop a method that combines both techniques, and to test the method through production of a full-scale concrete structure. The idea was to integrate production and method development process, so aspects concerning realisation could be embedded in the form-generation method. In particular with respect to the digital tools that are necessary for generating drawings and information for the digital production. Besides the researchers and consultants, a group consisting of students from both Sydney and Aarhus participated in the development and realization of the construction.

8.1.2 Organisational logic

The method consists essentially of two parts. The first regards form-generation and generation of information for manufacture and assembly. The second concerns production and construction. Basically, there is a virtual and a physical part. In reality, the virtual and the physical parts are interlinked, which is a crucial aspect of configuring the method for a specific task. Figure 2 shows how

1 The chapter is based on a conference paper: Niels Martin Larsen, Ole Egholm Pedersen, Dave Pigram, 'A method for the Realisation of Complex Concrete Gridshell Stuctures in Pre-cast Concrete', *manuscript submitted for publication, ACADIA, 2012*.

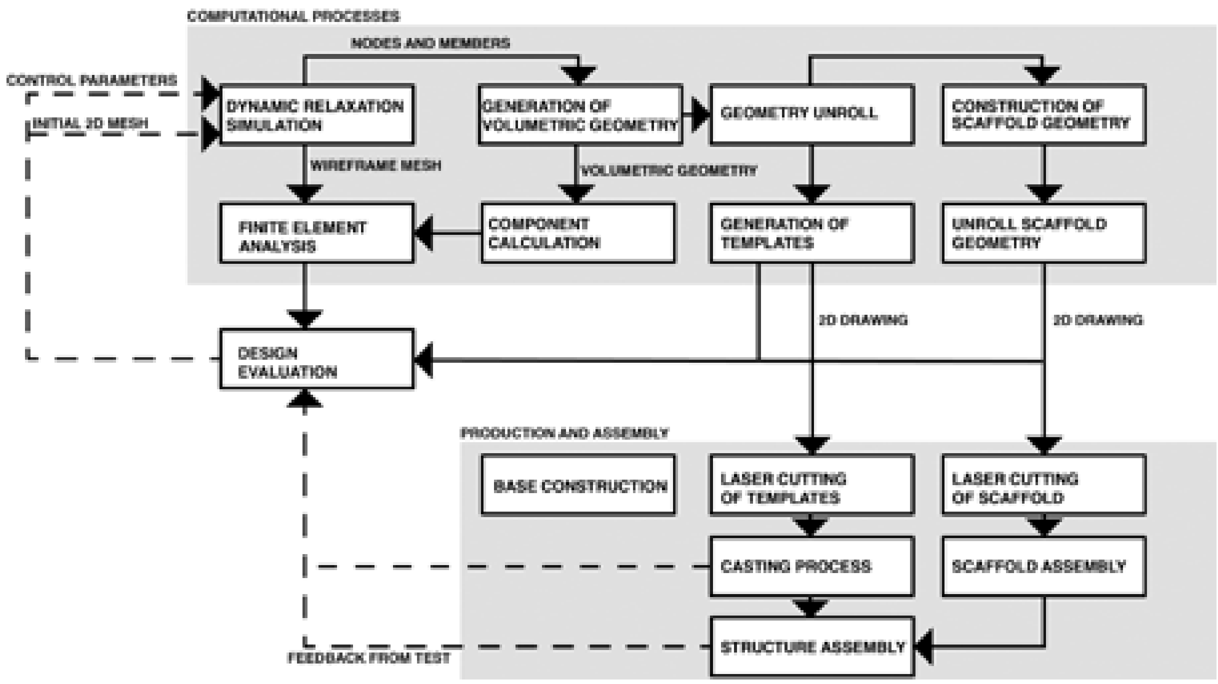
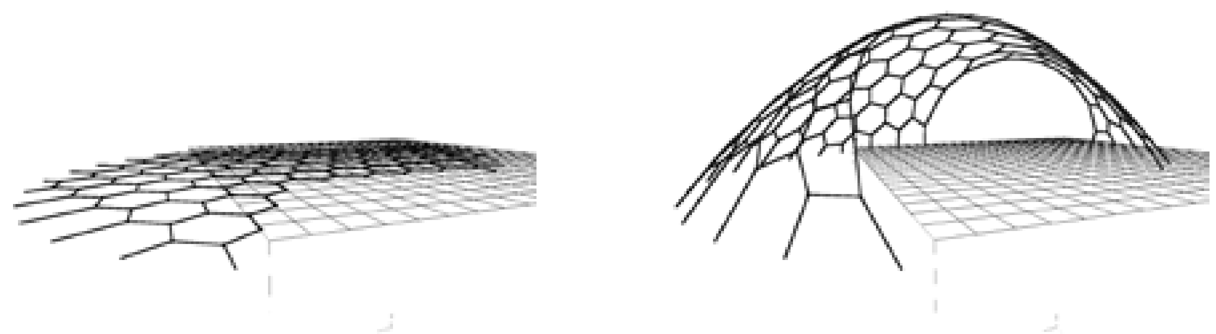


Figure 2: Method for constructing a concrete gridshell. Diagram of the flow of information and materials. The dashed lines indicate feedback loops during development.

information and materials flow through the system. The dashed lines indicate feedback loops during the development process. The cyclic design procedure includes all aspects of the realisation process, through form-generation, production and construction. It breaks from a linear design process, where information concerning production and construction is confined to the later stages of the development.

The form finding method is based on the principles for generating optimised vault structures, demonstrated in Antoni Gaudi's hanging chain models for the *Sagrada Familia* cathedral in Barcelona. Through physical self-organisation, the chains take forms that contain tensile forces only (catenaries). When the hanging form is inverted, the forces are translated into pure compression, resulting in funicular forms optimised for construction in materials such as stone. The self-organising process can be described by using Hooke's law, which states that for elastic deformations of an object, the magnitude of its deformation (extension or compression)

Figure 3: Dynamic relaxation algorithm. Left: Input mesh imported as a 2D drawing. Right: 3D mesh in equilibrium state.

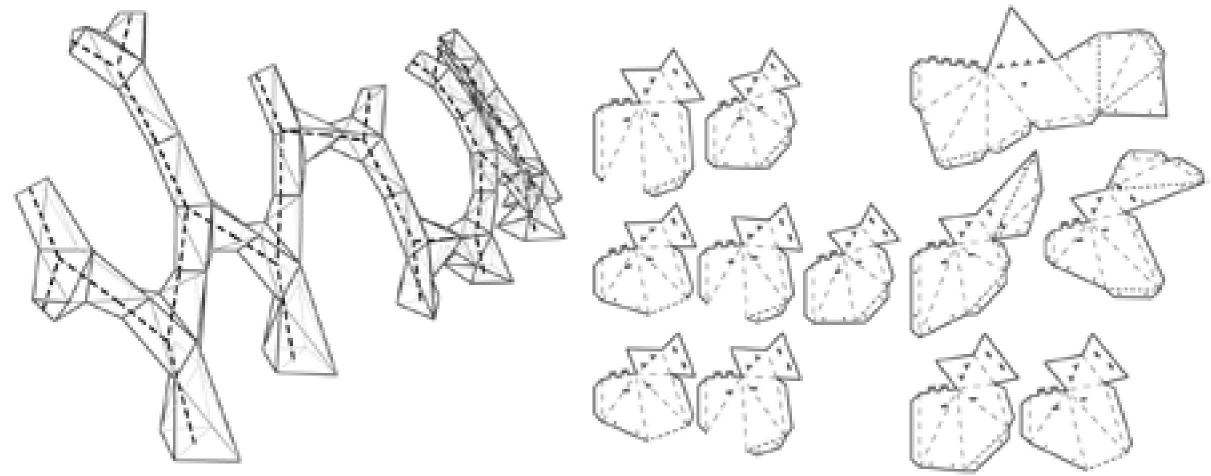


is directly proportional to the deforming force or load. Hooke's law[2] states that the applied force F equals a constant k multiplied by the displacement (change in length) x, thus: $F = kx$. The formula is implemented in the computer application ReVault, previously developed by Iain Maxwell and Dave Pigram. The application can be used to simulate the self-organisational behaviour seen in the hanging chain models by Antoni Gaudi. The simulation tool enables testing of a number of possible solutions within a limited time span, compared with testing through physical models. The time for establishing the geometry of the structural mesh represented a total of three working days for one person.

The application takes a two dimensional drawing of a mesh as input, and simulates a dynamic relaxation process, controlled by setting a list of essential variables, such as the relative rest-length of the members and damping of the system. Through iteration, the system arrives at an equilibrium state, meaning that all the forces in the system are balanced, and the velocity of the nodes is zero. In this state, the structure is in pure tension, or in pure compression, depending on the setting of the gravitational force. The generated three-dimensional mesh is then exported to a 3D modelling software. Forms generated through the dynamic relaxation form-finding processes are optimised in terms of compression-only load distribution from the structure's own weight. Live loads, such as wind load or point loads from people climbing the structure are not computed in the initial form generation. In order to both verify the output of the form-finding software, and to calculate the structure's performance, the generated wireframe geometry is transferred to Autodesk Robot software in order to perform *Finite Element* analysis. The wireframe geometry, generated through the ReVault simulation, is imported into the 3D modelling software Rhinoceros, and further processed through use of the IronPython scripting module. The

2 'Hooke's law.' *Encyclopædia Britannica Online*, 2011, viewed 14 Dec. 2011.

Figure 4: Left: Volumetric geometry, generated from mesh lines. Right: Patterns for laser cutting of templates. Illustration: Dave Pigram.



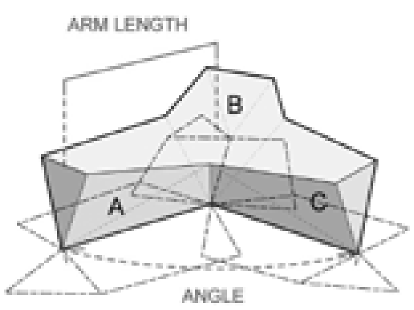


Figure 5: Basic diagram of folded template. Illustration: Ole .E. Pedersen.

geometry is developed into unique volumetric components via custom written algorithms (Figure 4 left). The code is configured so it reflects the design intents of the component geometry and the requirements of the production process. For instance, it is able to distinguish between regular and base components, generating a flat base for the latter. In the same script, input for the manufacturing process is generated. This includes scoring lines for folding, rivet holes, flaps for stability, holes for tube inserts to run the tension cables through, and the engraving of a unique number. (Figure 4 right) The three-dimensional component model is used for extracting the geometry of the scaffolding and for positioning the individual components during the assembly.

8.1.3 Realisation

The following describes the general realisation method and subsequently the details concerning realisation of three case study experiments. For production of the templates for concrete casting, laser cutting is used in order to produce unique components at relatively high speed, compared with other techniques for mass-customisation. Here, three-dimensional form is generated from flat sheets by means of folding, following scored lines. This means that pre-cast components can be designed with a number of parametric variables, which can cause, and be influenced by, differentiations in the component design. In the case studies, Y-shaped components were practical, but the overall method in general and the casting method in particular can accommodate other component geometries and structural forms. The mould material utilised by this method is PETG plastic. It is easily recycled, by melting, at 260 ºC, evaporating only CO_2 and water. It is a technical nutrient and should remain in a closed recycling process with no degrading.[3] This suggests that the plastic sheets used for moulds can be melted and reused. However, this aspect was not pursued in the case studies, due to the limited material amount. The PETG sheet, typically 1 mm thick, has a high degree of deformation, and must be reinforced by folds by triangulating large areas and limiting the area of planar surfaces. These factors contribute to the aesthetic characteristics of the method. Given the relative thinness of the elements, it is extremely important that the constructed structure matches the computationally found form so that all load paths remain within the sectional profile. Additionally, as is typical in non-Catalan vaults, the structure is not stable in an incomplete form. Therefore, it is necessary to use formwork to ensure the exact positioning of every component and to

3 William McDonough, Michael Baumgart, *Cradle to Cradle: Remaking the Way We Make Things,* New York, New York: North Point Press. 2002

Figure 6: Production of concrete components. Uppermost: Template assembled from laser cut and folded PETG plastic with rebar installed. Bottom: Template with concrete.

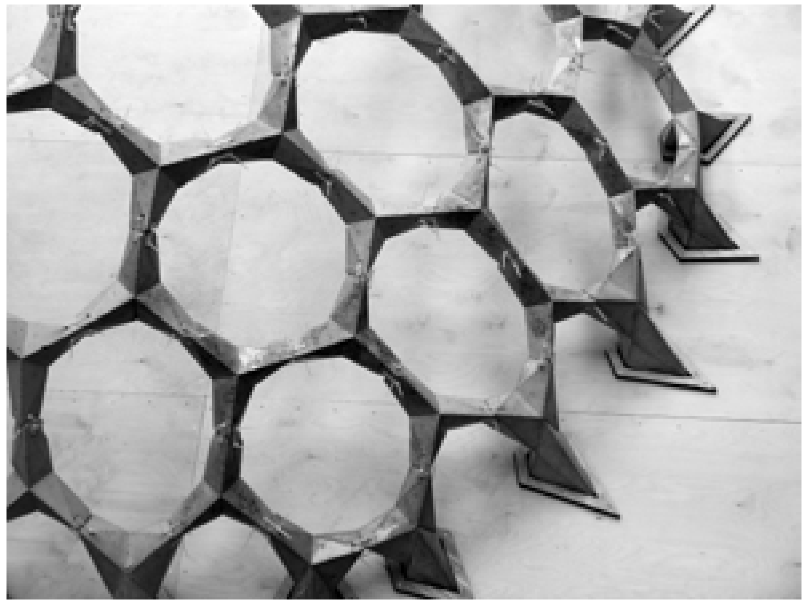


Figure 7: Concrete Gridshell Pavilion. The components form a hexagonal pattern. The base components are articulated in order to meet the ground.

support them during assembly. Like the precast components, each scaffolding element is unique.

Lateral forces that arrive at the springing point of a gridshell need to be resisted. This is most often achieved through in-ground footings, but where such footings are impossible, establishing connections across the structure at the ground level can also resist lateral forces. In Case Study 2, the *Concrete Gridshell Pavilion*, the forces were transferred through a plywood floor plate that engaged the bottom row of components. Component placement is directed by applying unique identifiers, referencing a component's arm to the corresponding arm on the neighbouring component.

The scaffolding can be produced from cardboard, laser cut and assembled, first into triangular tubes, then into larger clusters forming hexagonal geometries, reflecting the plan of the concrete structure (Figure 12). A plan drawing in full scale, mounted on the floor plates, can be used to position the scaffold. The function of the scaffold is both to support the structure during construction, and to ensure the exact positioning of the components. The latter aspect being most important, since very small deviations from the spatial geometry made assembly impossible. The cardboard can be recycled after use.

Three constructed case studies contributed to the development and testing of this method. The first case study was a deliberate test of a design that had known insufficiencies and thus represents something of a worst-case scenario. The second case study incorporated all of the lessons learned from the 'worst-case' prototype, and the third was an attempt to integrate some degree of usability with respect to construction at a Kindergarten playground. The first case study was produced in order to practically discover

Figure 8: Concrete Gridshell Pavilion. During construction, the structure was supported by a cardboard scaffold.

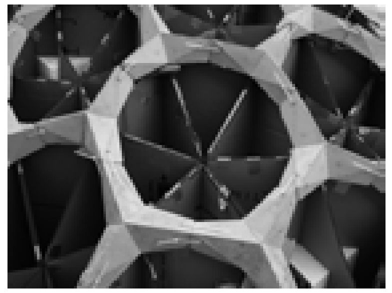


Figure 9: Case study 1. The structure could not be stabilised without connections.

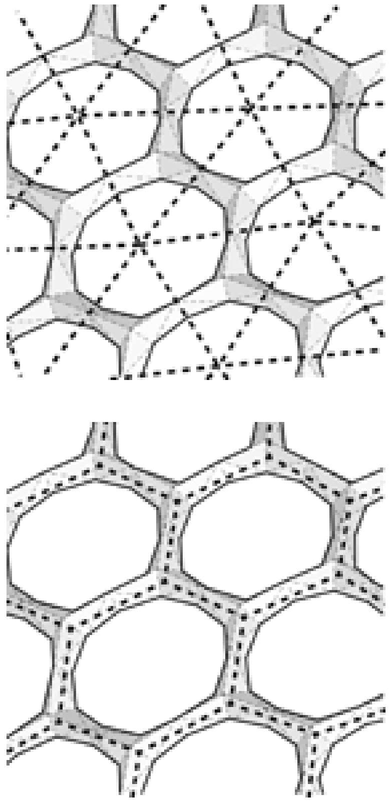


Figure 10: Translation of mesh to component geometry. Uppermost: Hexagonal geometry from triangular grid. Bottom: Hexagonal grid directly translated.

many of the constraints that eventually would guide the method. The prototype was based on a triangular mesh, which was transformed with the dynamic relaxation procedure. The structure was symmetric unlike the complete structure in the following case study. Through generation of the geometry, the mesh was translated into the hexagonal pattern, shown in the second case study, the *Concrete Gridshell Pavilion*. Finite Element Analysis showed key problems in the first case study, and provided essential information for realisation of the *Concrete Gridshell Pavilion*. The first analysis showed that a method, where the hexagonal geometry was generated on basis of a triangular mesh, resulted in deviations from the optimised shape that were bigger than the component joints would be able to obtain. Therefore, the principle was changed so the base geometry was defined as hexagons, and represented the geometry more directly in *the Concrete Gridshell Pavilion* (Figure 10). The prototype test illustrated the importance of connections. Correct positioning of each element is next to impossible without dealing with shear forces between the elements. This means that the assembled form will not match the computed load paths, the construction will not be compression-only (even only under self-weight), and will collapse due to the lack of tensile resistance at the joints (Figure 9). This translation resulted in a noticeable deviation from the funicular shape. Coupled with the fact that the joints in the prototype could not absorb and kind of shear or tension forces, this meant that it was practically impossible to assemble the prototype. Another finding was the importance of correspondence between initial and final mesh geometry. The test also showed that the use of tension cables to counteract the horizontal forces proved difficult to control. In the final structure, the components were connected with structurally rated cable ties (Figure 11). Each cable tie can withstand approximately

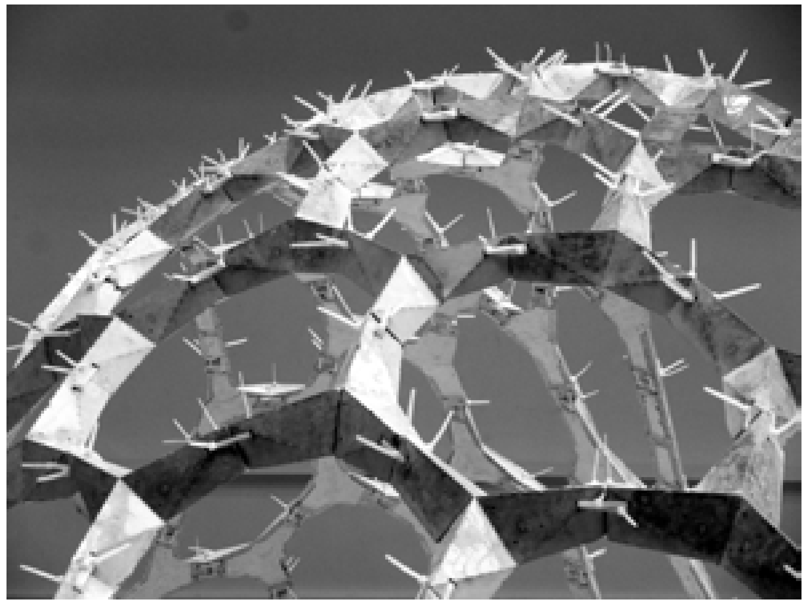


Figure 11: Concrete Gridshell Pavilion. Structural pattern after assembly. The components are connected with cable ties. Since the components are planar, the curvature is obtained in the joints where the components meet in an angle.

0.9kN, providing a safeguard against unexpected forces, e.g. from human intervention.

The Concrete Gridshell Pavilion was used as part of the cultural event 'Kulturnat Aarhus'. Given the experimental nature of the construction, it was necessary to perform an initial test assembly to verify its structural integrity. Hence the structure was assembled and disassembled twice: once at Aarhus School of Architecture and once opposite Aros, Aarhus Kunstmuseum. Anticipating this, the structure was designed with the capability of being disassembled. It was this requirement that made the use of many, smaller components the most suitable solution and which led to the use of mechanical (as opposed to cast) joints between them. It follows that the detailed design of a parametrically variable concrete component was a primary issue in the development of this method. The ultimate

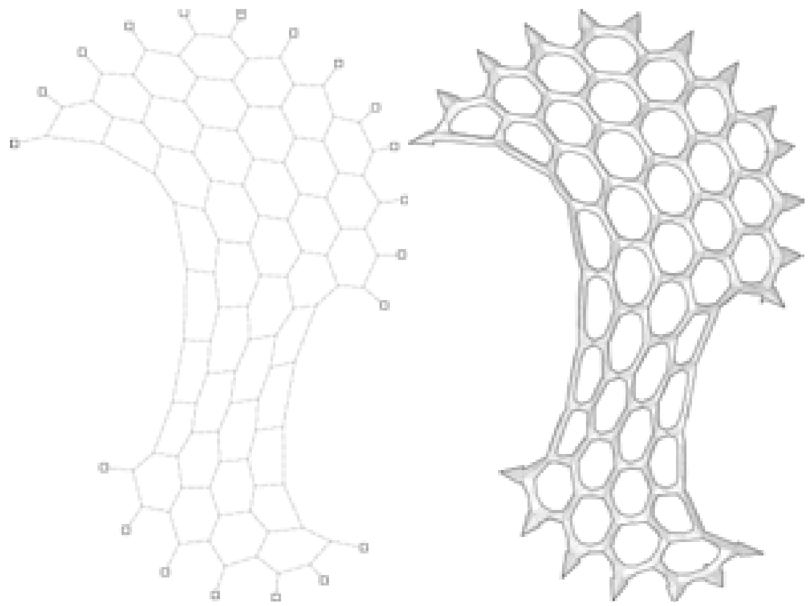


Figure 12: Plan drawings of the structure in equilibrium state after running the dynamic relaxation simulation. Left: Wireframe mesh with anchor points, showed as small boxes. Right: The volumetric geometry generated directly from the wireframe mesh. Illustrations: Dave Pigram.

Figure 13: Plan diagram showing the bending forces in the structure. Left: Preliminary design where the hexagons are generated from a triangular mesh and the edges defining the two big openings are rough with standard Y-shaped components. Right: Optimised final structure where the edges are solved with use of straight components and the grid is originally defined as hexagonal.

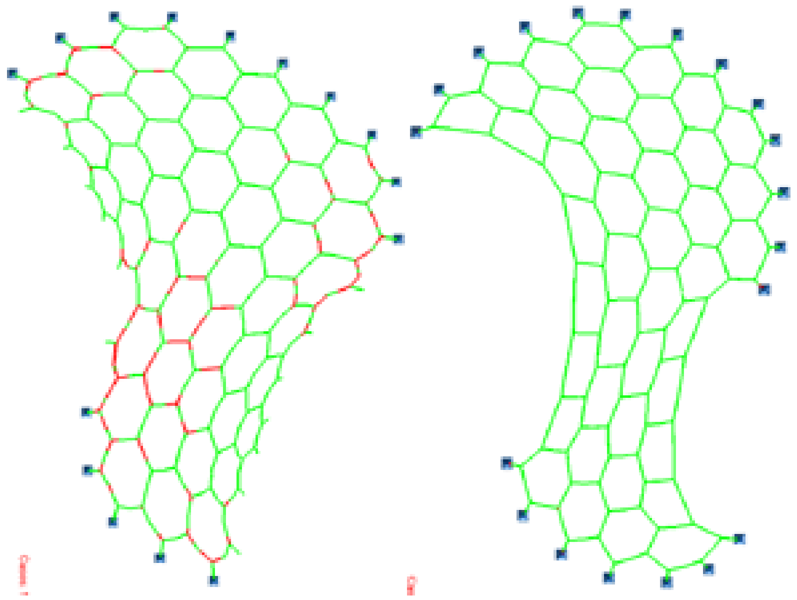


Figure 14: Figure 9: Case study two. Second assembly. The 3D model was used for referencing the components and the scaffold throughout the assembly process.

component design resulted from the negotiated input of a large number of constraints such as structural strength, reduction of weight, sufficient volume for reinforcement, fabrication tolerances and assembly time. Of similar significance and intimately related to the component design, was the challenge to find a suitable joint solution manageable in terms of both production and economy. A production workflow, where the geometry was transformed between two and three dimensions according to the different parts of the process, was based on digital form generation and digital production logics. Algorithmic work flows allowed for the components to obtain their necessarily unique form and dimensions without major increase of the production time. In earlier testing of the Aarhus Pavilion's form Finite Element Analysis had revealed problems at the borders of the passage openings in the structure, which was expected. By defining the edges as approximately continuous members, it was possible to reduce the tension forces, which again was confirmed by use of the FE analysis. Furthermore, the analysis was used to calculate the shear forces in the joints, which again was used for making decisions regarding the joint design and materials. Principally there would be no shear forces in the joints, the structure being in pure compression. In reality, shear forces did occur, due to the large passage openings in the mesh, and due to lack of precision, both in the production of the components, and in the process of assembling the structure.

Case Study 3 was carried out as a workshop, led by Ole E. Pedersen, with 40 students over a period of two weeks at Royal Academy in Copenhagen and had two purposes: to test the method in an industrial production outside the laboratory and to explore the potentials for the method to deal with a more complex overall form. A revised version of the dynamic relaxation algorithm that was

Figure 15: The Concrete Gridshell Pavilion served as landmark both for the event Culture Night and the exhibition Aarhus Urban Lab in Ridehuset, Aarhus, October 14 2011.

used in the ReVault software was developed by the author. It was implemented as a Grasshopper/Python component in Rhinoceros, in order to keep the digital development of geometry within a single piece of software. This allowed first year students to quickly learn the workflow of drawing a mesh, performing dynamic relaxations, and component generation. This enabled many designs to be quickly proposed and evaluated by the participating engineers. The original algorithm is a simplified version of a spring system, which does not include velocity as part of the calculation. The Grasshopper implemented version uses a Velocity Verlet calculation method, thereby simulating a physical spring system. In Case Study 3 any noticeable improvement from using the latter calculation method was not identified in terms of the form-optimisation. However, the enhancement enables larger degree of regulation with respect to future experiments.

8.1.4 Observations

The case-study pavilions demonstrated that the forces simulated in the form-finding software, corresponded to the forces acting on the physical structure. It also demonstrated that the method

Figure 16: Case study three. Dynamic relaxation procedure and generation of volumetric geometry. Drawings: Ole E. Pedersen.

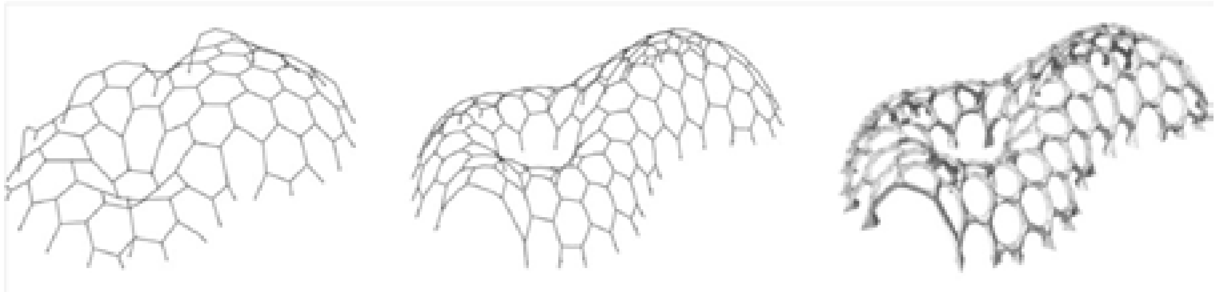


Figure 17: Case study three. Realised structure, Copenhagen. Photo: Ole E. Pedersen.

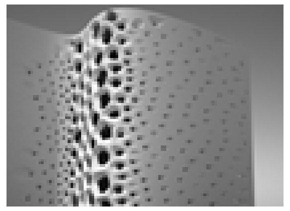
of translating an abstract wireframe mesh into physical concrete components worked and yielded a precise enough outcome to make for a viable construction. By designing a shell as a grid as opposed to a solid surface, overall weight is reduced considerably. If a 30mm thick, solid surface shell has a total weight factor of one, the shell of the *Concrete Gridshell Pavilion*, with its y-shaped components, had a total weight factor of 0.65.

Calculations from the engineers showed that a very high level of precision is needed for such a thin structure to rest in a funicular gridshell. The direct translation of computer generated form into the laser cutter meant that this precision was maintained, and concrete elements could be cast with a tolerance less that a millimetre. In practice the tolerance was a little higher, because some moulds skewed or twisted when the heavy concrete was poured. Further development would include upgrading the script to generate supports preventing such deformations.

The *Concrete Gridshell Pavilion*, constructed in a very short time, for low cost and with relatively unskilled labor demonstrates that the integration of algorithmic form-finding techniques, CNC fabrication workflows and the use of innovative PETG folded mould techniques enables the practical realisation of freeform funicular structures in pre-cast concrete. While the pavilion was a success in terms of precision and structural performance, Case Study 3 demonstrated that the proposed method can be utilized to generate more complex forms. This case study also showed that while it is a flexible method, it is also one sensitive to imprecision and to scale through the accumulation of dimensional variances. It also requires the use of digital tools for both form generation and evaluation.

To conclude, the project demonstrates how digital technologies can be used to gain larger variation and complexity in the design and realisation of an architectural construction. The self-organisational method for generating was successful in terms of arriving at an optimised structural shape within a limited time span. The method of including structural analysis proved to be beneficial, especially because the construction was assembled in a public space. On different levels it was possible to establish feedback loops on different levels. By embedding material properties in the form generating process, and more general in the way the production conditions became guidelines for design of the components. Despite the success in terms of implementing the digital tools, it should be pointed out, that human labour played a large role, mainly in the production phase. Improvement of the described method could imply optimisation and automation of the processes of production and assembly, particularly in case of implementation in a larger scale. A further development of the system could be to incorporate a genetic algorithm in the process of finding the best version of the possible formations. This would imply a direct connection from the structural analysis to the setting of variables and perhaps drawing of the mesh, which takes place in the beginning of the form generating process. Realistically, the genetic algorithm would work most efficiently in the process of fine-tuning the settings.

Acknowledgements. Students taking part in the method development and realisation of case study two: Jon Krähling Andersen, Anastasia Borak, Bing-Nian lan Choo, Lauren Foley, Kara Gurney, Alexandra Wright, Tara Fitzgerald Kennedy, Aleksander Czeslaw Tokarz, Jacob Lohse Ellung Christensen, Yi Lin, Andrew Stephen Fong. Many thanks to the enthusiastic engineers participating in the project: Ronni Lundoff Madsen and Jacob Christensen. Andreas Bak helped with the Verlet Integration algorithm. Also thanks to architect and tutor Stefan Rask Nors for assistance and participation.

	**Intent and conditions**	**Organisational logic**	**Realisation**
Properties	Self-organisational adaptive surface pattern Local variation Complex negotiations	Geometrical rules Dynamic grading Conditions and response Environmental parameters	CNC milling processes Generative techniques for realisation information Subtractive methods
Observations	Rule specification. Digital environment Relative optimisation.	Adaptive logic Differentiated articulation Self-organisation	Digital production tech-niques NURBS constraint

8.2 Self-organising Bezier curves

8.2.1 Intents and conditions

This method was developed as an investigation of a surface based self-organisation. The main property of the method is that the pattern is constrained to the surface on which it is drawn upon. In this sense, the method is two-dimensional. This constraint suggests a connection with more familiar architectural design methods. A sense of three-dimensionality is also established with the method. The pattern grows on a NURBS surface, therefore transcending the confines of a specific plane. By connecting patterns on different surfaces, the result of the procedure can be described as a ‘deep’ pattern. The pattern developed as a simple form of self-organisation. The underlying idea of the procedure is that the pattern is not constructed from a rigid grid or geometric system. Normally, when constructing a pattern, a dividing principle is defined and laid out in order to specify the size and position of each pattern element. One aspect of using a self-organised pattern is to transcend the constraints of repetition, but more interesting is that it opens a field, where a range of parameters can participate in negotiating the exact expression of the pattern. Perhaps this type of self-organisation can be seen as a steppingstone towards implementing Stan Allen’s ‘field conditions’ in terms of tectonics.[1] The most interesting aspect of this method is not the actual appearance of the pattern, but the generative mechanisms and how they can be implemented in different ways.

8.2.2 Organisational logic

The pattern is basically constructed in two separate steps. First, a number of circles are distributed at the surface. They are placed randomly within the surface field and are equipped with very little individual data, such as their position, their radius and their preferred distance to other circles. The distribution process can be described as such: Consider each circle as a type of individual that is able to move in small steps through a number of iterations. In each iteration the circles are individually updated. Each circle ‘detects’ if any circles are within its private zone. If this is the case, it moves in the opposite direction of the closest circle in a very small step, otherwise its position stays unchanged. This simple rule (shown in Figure 1) is sufficient to make sure that the circles distribute evenly, generating a homogeneous pattern. As indicated, the pattern is not completely

1 Stan Allen, ’From Object to Field’, *Architectural Design vol. 67 no. 5/6*, 1997, p.ages 24–31.

Figure 1: The basic principle of the circles moving gradually away from the nearest circle within the 'private' zone.

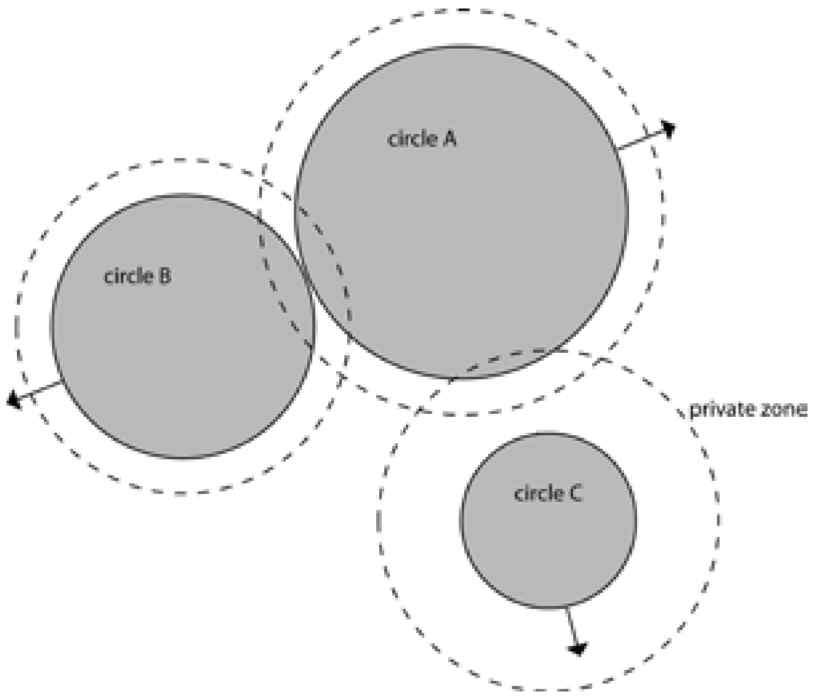
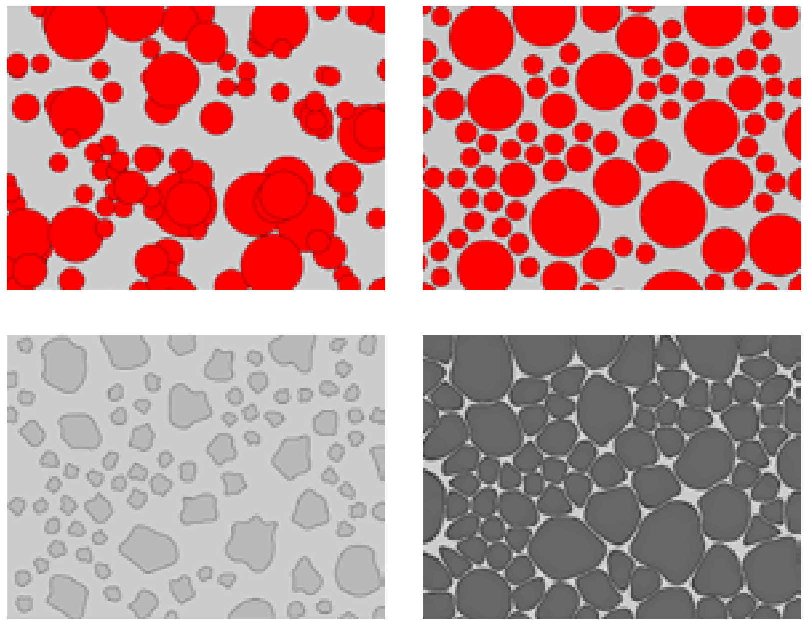


Figure 2: The pattern generation steps. Top, left: Randomly distributed circles. Top, right: Circles after initial self-organisation process. Bottom, left: Bezier curves randomly constructed from circle centres. Bottom, right: Bezier curves have reached an equilibrium state after iterative expansion process.



optimised. If this was the case, a hexagonal pattern of equally sized circles would emerge. In fact, this would be the result if the methods were supplement with a cohesion-functionality. But the goal is in this case not to arrive at an optimised packing of the circles. The second part of the system is the drawing of the Bezier curves. Bezier curves are basic mathematical functions for defining curves from control points. One closed Bezier curve is constructed for each circle on the surface. The construction is carried out in the following way: A vector with a random length is defined so it is aligned with the x-axis. The vector is randomly rotated around the centre, and marks the position for a control point. The vector length is altered, and the vector is rotated again in order to place the next control point. This is continued until all control points are set. Then, a closed Bezier curve is constructed from the control points. After this has been carried out for all the circles, an expansion process is initiated. Through a series of steps, all of the control points are gradually moved away from the centre of the polygon. The movement stops when the point gets within a predefined minimal distance from another polygon. Effectively the system arrives at an equilibrium state where all the points stop moving. Compared to the initial chaotic state where the circles are randomly placed, the new pattern can be said to have emerged.

The ways of controlling the pattern are relatively limited, at least at this abstract level. Still, circle radius and distance can be adjusted globally or varied locally. When the parameters are linked to local conditions, such as the position on the surface, these variations begins to demonstrate adaptability of the system. The images in Figure 6 show a range of possible adjustments within the system. The difference between the top row and the second row is simple change of radius. In the abstract diagram it is simply a matter of scaling, since the relations are unchanged. The third row demonstrates how the radius can change according to the position of the circle. If the image is seen as a coordinate system, the radius is proportional to the x-coordinate. To be precise, the radius is calculated as an exponential function in order to have 'more smaller circles.' In the fourth row, the distance between the circles is also varied. Again, if the vertical direction is seen as a y-axis, the distance is reduced in proportion to the y-coordinate. This diagrammatic variation reflects how the pattern could respond to both 'local' and 'global' conditions with respect to specific use. As will be explained in the following, certain input parameters could for instance control the degree of openness.

In order to create a volumetric pattern, different Bezier curves are extracted. In the example, the differently sized Bezier curves are generated on one single NURBS surface. The first Bezier curve is saved and placed on another NURBS surface when the

Figure 3: Principle diagram. The Bezier curves are constructed from control points defining a polygon shape. The points are positioned via a vector that is rotated around the centre of the original circle.

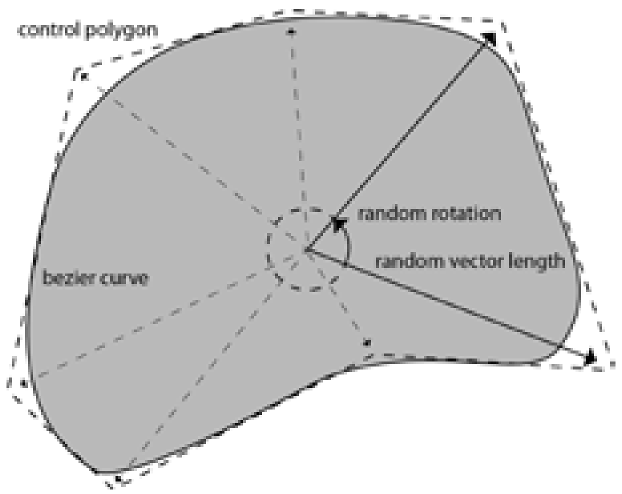


Figure 4: Each Bezier curve control point moves gradually away from the centre.

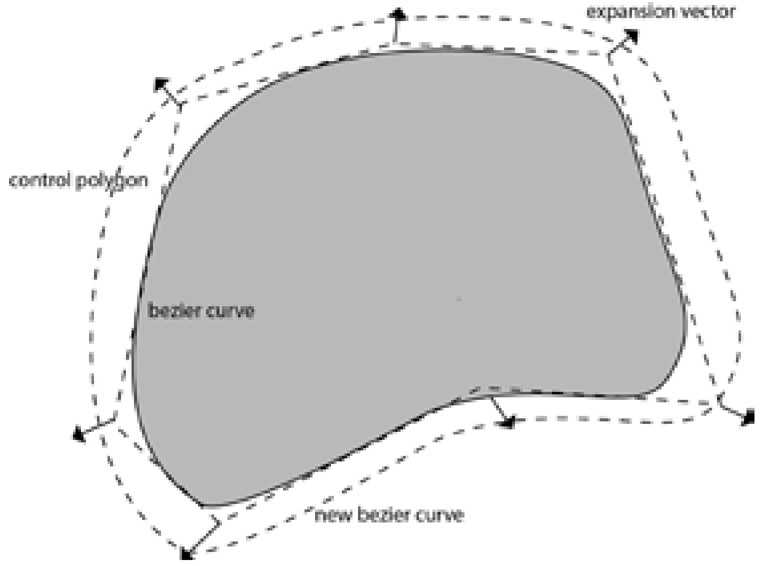
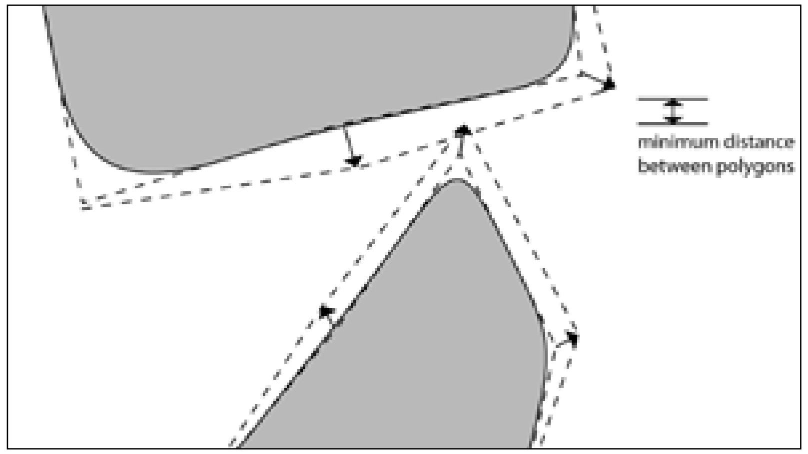


Figure 5: When a control point approaches the private zone of another polygon, the movement settles.



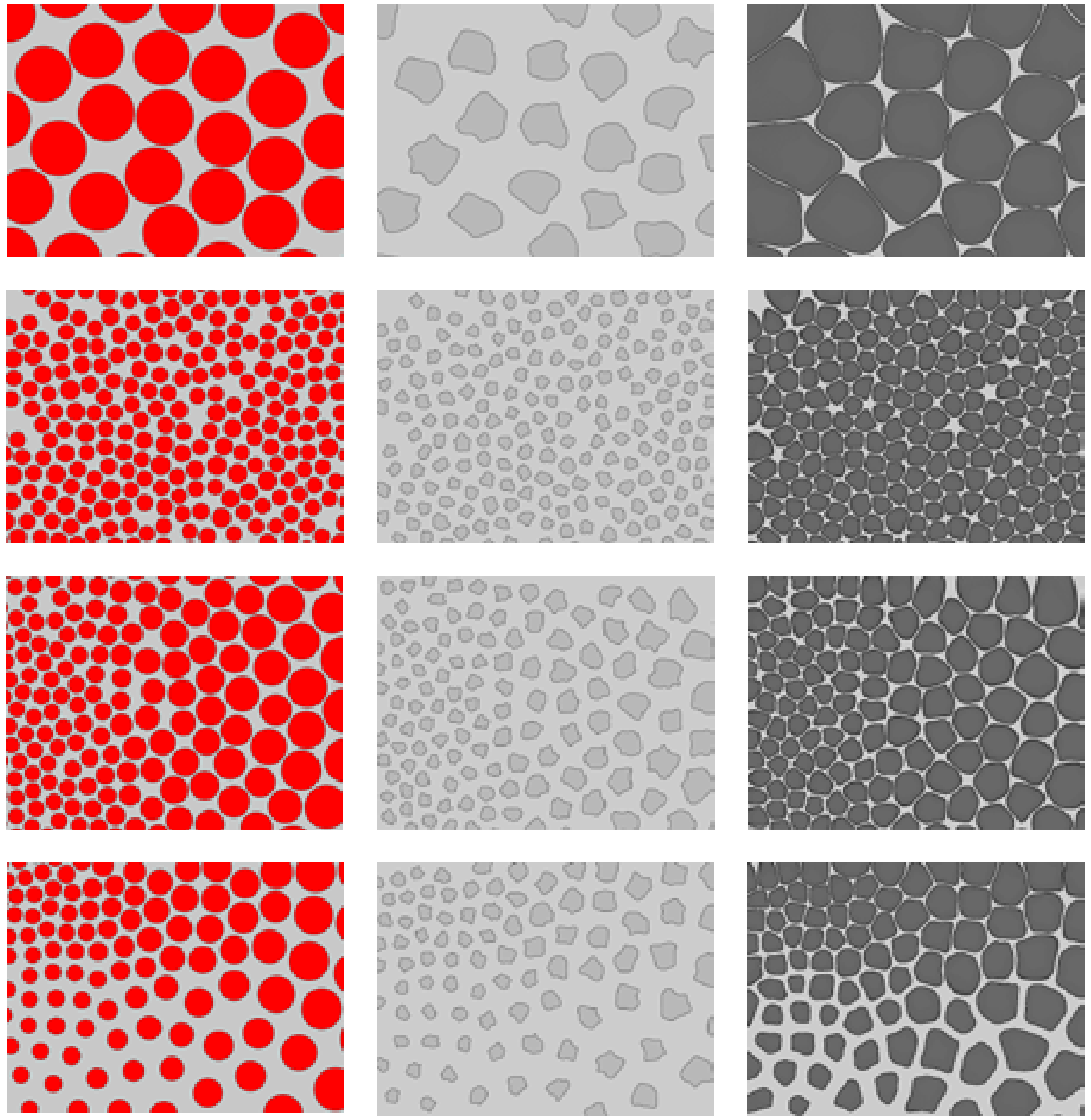
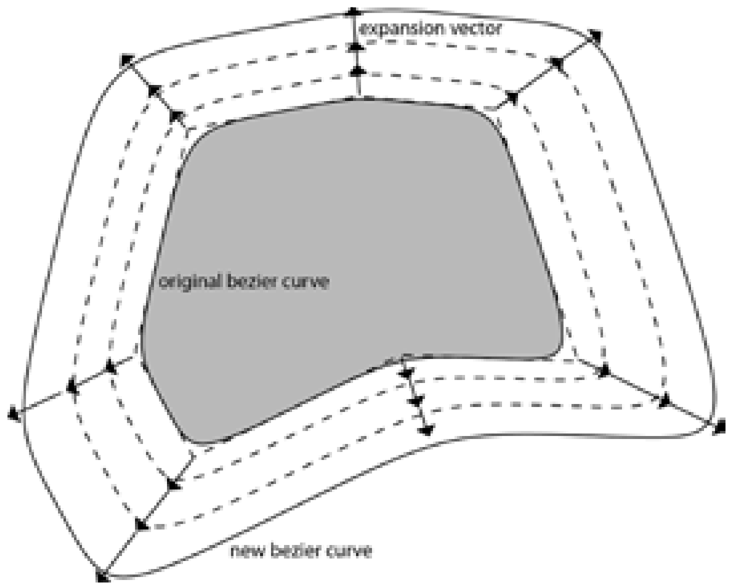


Figure 6: Global and local adjustment of control parameters. The two top rows shows the effect of globally changing the radius for the circles. Third row shows local adjustment of radius, and fourth row also have local adjustment of distance between the shapes.

Figure 7: Diagram of an expanding Bezier curve. The algorithm runs much more iterations than the 4 curves, shown in the diagram. For the volumetric pattern, the first and last curve are used.



final large Bezier curve is generated. This surface represents the exterior of the 'wall,' where the first represents the interior. The two surfaces have to be created simultaneously, or in a way that ensures that their surfaces parameters correspond. The interior and exterior curves are connected with so called 'loft' procedures, thereby creating the curved surfaces that define the sides of the opening. The corresponding holes are cut out of the original surface plane. This final procedure of cutting is partly a manual process, since the trimming of NURBS surfaces has to be completed in steps when complicated. By eliminating the intermediate Bezier curves and confining the three-dimensionality to connecting only two curves, the result is a ruled surface. The straightening of the interior of the holes is primarily for aesthetic reasons, but also to make an eventual construction much more rational, which will be explained more in the next section. Figures 8-11 show a variety of adjustments possible with the volumetric pattern. The difference from 8 and 9 is simply the radii of the initial circles that have been changed as a global value. This allows a scaling of the pattern. Figure 10 shows how this change can happen locally, leading to a gradual change in scale. Figure 11 also shows a gradual locally defined change in parameter, but here it is the minimal distance between the shapes that is changing. As indicated, these small regulations of the system have great impact on the character of the pattern, and it is crucial for the designer to have this kind of control in order to make use of it for a particular solution. Moreover, the controls come into play when the system begins to relate to environmental parameters, as explained in the following, where the pattern's adaptability is related to its use in architecture.

An example of a facade screen with orientation dependent openings is seen in Figure 12. The sketch model was developed for

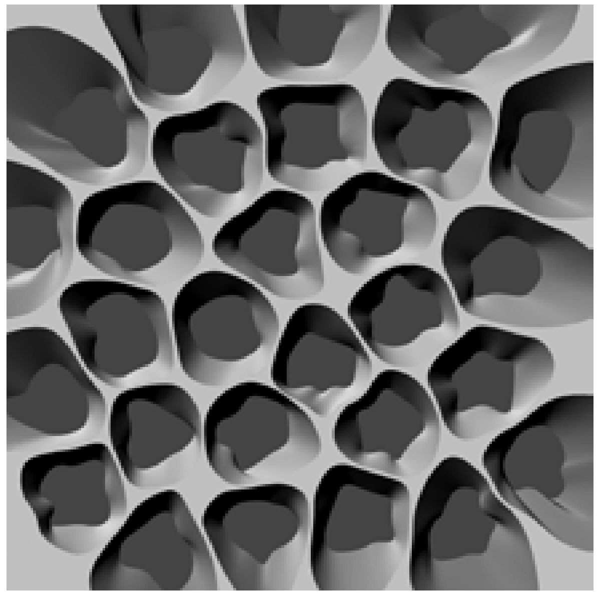


Figure 8: Volumetric surface. Relatively large radius, or distance between shape centres. Even minimum distance.

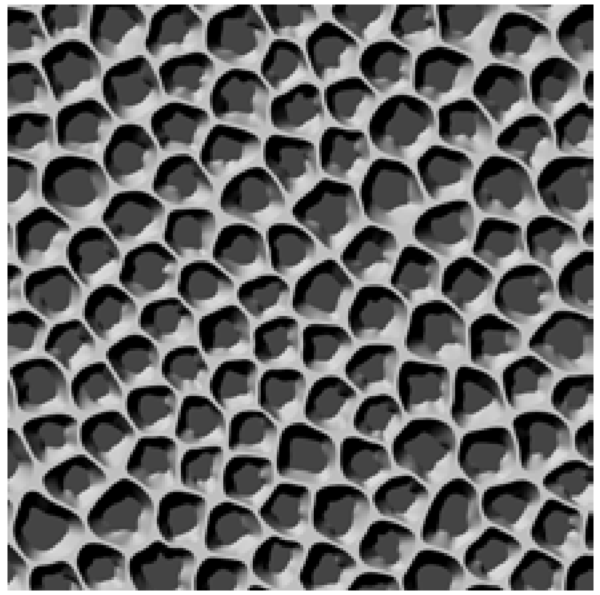


Figure 9: Compared to figure 9, the radius has been significantly reduced. Even minimum distance.

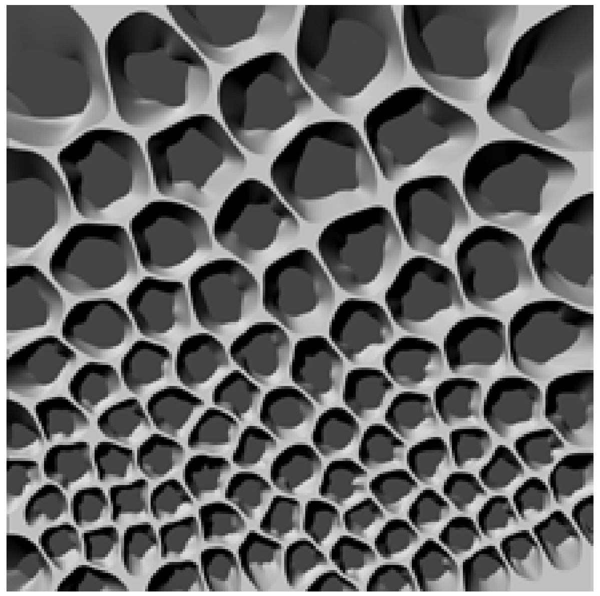


Figure 10: Gradually increasing radius for shapes closer to the upper part of the volumetric surface.

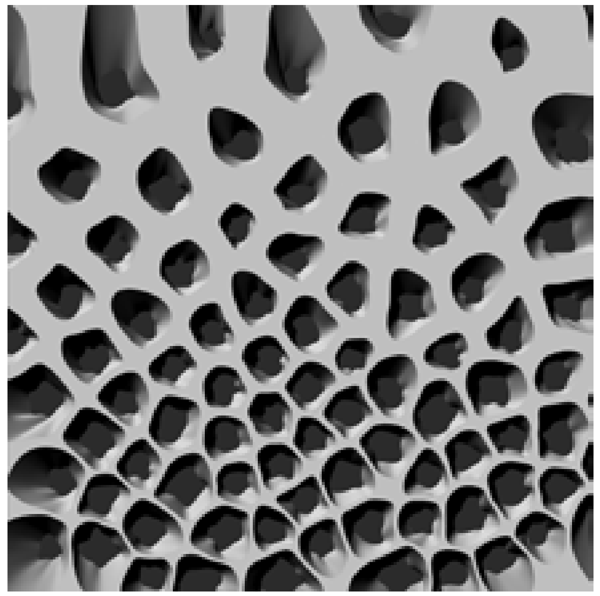


Figure 11: In this example, both the radius, or distance between centres, and the distance separating the shapes, are increased when the shapes approaches the top of the surface

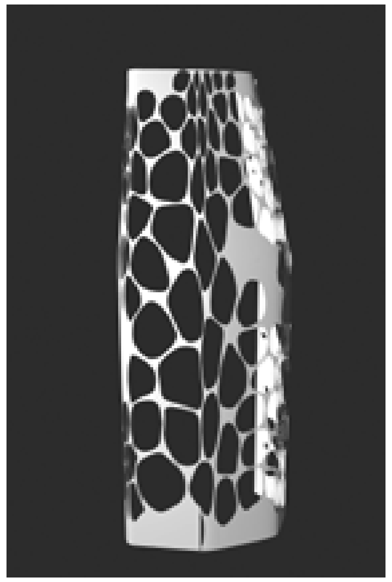


Figure 12: Sketch model for a self-organised facade pattern in relation to the Miami Civic Centre competition.

the *Miami Civic Center* competition in 2010. It is worth noting that more subtle regulations could also be implemented, for instance, the exact shaping of the Bezier curves. Another regulation could be size of interior polygons versus size of exterior polygons. The external parameters that affect pattern generation can be of different type. Clearly, if the pattern were used as a facade, the use of the building and the character of the interior spaces would need to play a role. Defining zones with certain parameters could control this. A more elegant solution would be to introduce attractors where, for instance, distances to predefined points are calculated and affect the generating process. Naturally, a response to the environmental conditions would be relevant. Depending on the altitude of the site, it may be desirable to have a more closed facade where the solar radiation is high. A similar example using DLA-growth also appears in this thesis. The example shown in Figure 13 clearly demonstrates how the system responds to orientation. The system was configured to gradually increase the size of the openings on the surface where they are oriented in a certain direction. Furthermore, the distance between the shapes is increased when the openings are pointing away from the view direction, thereby enlarging the difference. Technically, the surface normals are evaluated as the elements distribute on the surface.

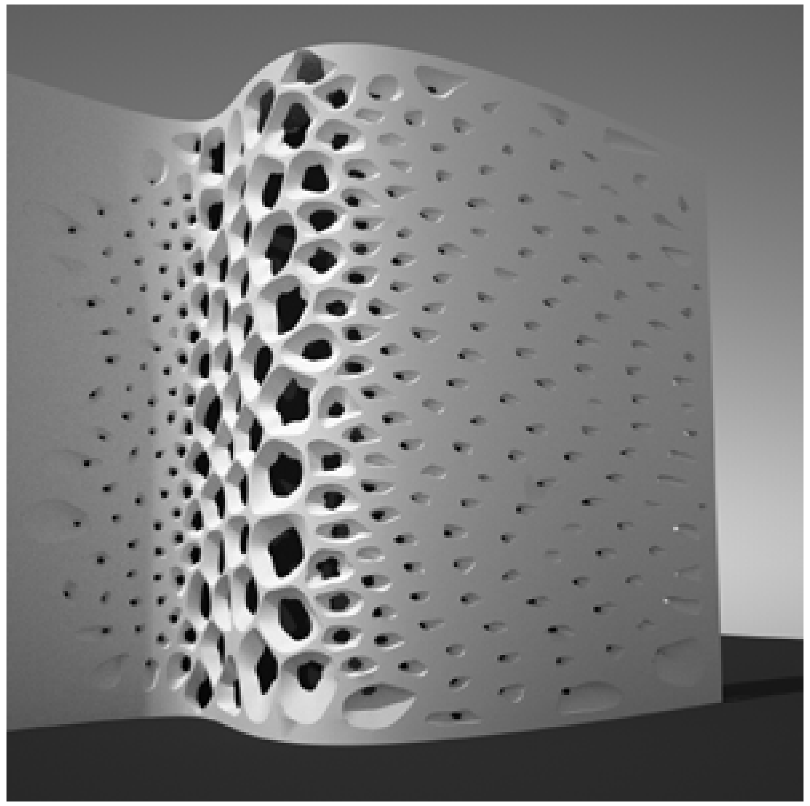


Figure 13: An example of the Bezier pattern implemented with a response to orientation.

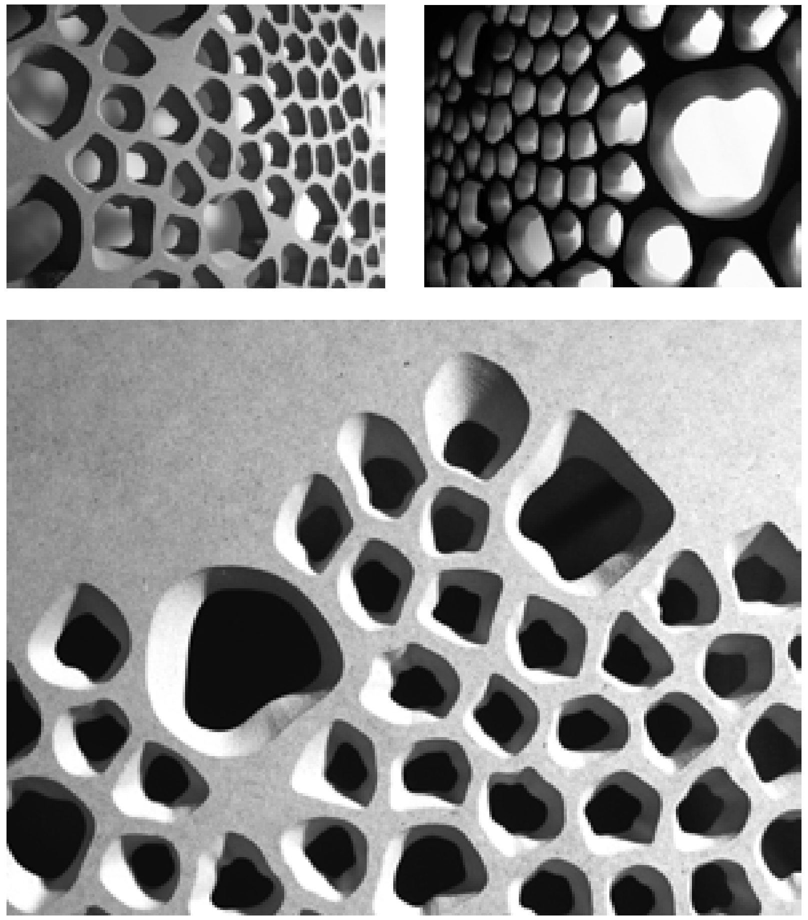


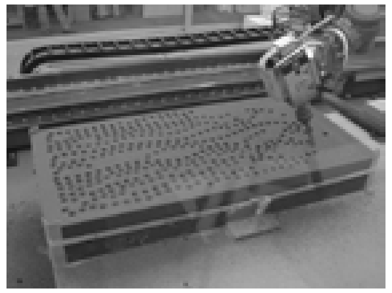
Figure 14: Pattern generated with the method Self-organising Bezier Curves realised in MDF though CNC milling.

8.2.3 Realisation

The pattern demonstrated clearly expresses a method of subtraction in terms of realisation. The immediate expression demonstrates a fully closed continuous surface that subsequently is perforated and hollowed out in a specific way. This approach has been tested out on a CNC milling machine, similar to the one shown in Figure 15. The machine has three axes, allowing it to move the drill vertically, but not to change the angle of the drill. The size of the drill is automatically adjusted according to the task, and the small sample displayed in Figure 14 shows the resulting smooth finish. This sample consists of a 1 cm thick MDF board of dimensions 40x40 cm. It was milled in approximately 10 hours. Consequently, this technique has certain limitations. First, the produced example is of the maximum size that the machine can handle. The use of larger machinery would rectify this issue. Secondly, and perhaps most importantly, the milling process, which removes all unwanted material, is very time consuming. Finally, the machine can only mill with a vertically oriented drill, essentially confining its use to flat panels.

A more viable production method would exercise the use of robots, enabling milling with more axes. An important step would be to ground the milling process on the fact that the shapes of the

Figure 15: The milling machine used for production of the example in figure 14.



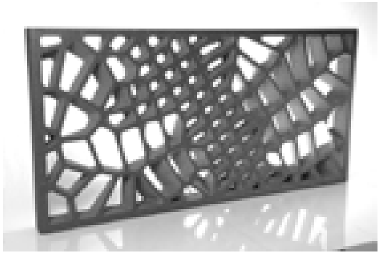


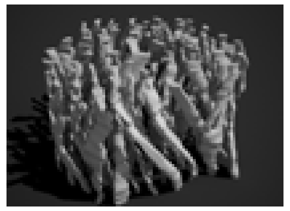
Figure 16: Top: Milling robot at the ETH, Zürich. The picture is from a workshop directed by Gramazio and Kohler. Middle: Example from the workshop. Lowest: Robot controlled hot wire cutting. Image from production of Matter Design's Foam Tower.

holes are ruled surfaces, which implies that they can be cut out from straight lines. This would result in the robot only having to follow the exact curve of the holes' edges. If this was the case, the drill would cut through the material by following the tracks of the inner and outer edge curves simultaneously. The advantage would be a substantial optimisation of the production time. This process has been tested in Zürich at the ETH, where different walls were cut out of foam. An example of this is shown in Figure 16. Without losing site of the method's realisation, it could be mentioned that the method for generating patterns at the ETH was based on randomly distributed points, not dissimilar to the initial procedure described in this chapter. The forming of the surface pattern was then generated with a Voronoi algorithm. It would be possible to use the same technique with a more solid material, such as lightweight concrete. This would again result in longer production time. Material waste would also become an issue due to the subtractive principle. In this sense, foam would have some advantages as a material as its waste is recyclable. Now, foam is often used as an intermediate material for further production. An example of this would be its use as templates for casting. In this case it would be possible to use an even more efficient method for cutting out the 'holes,' which would then shape 'negative' volumes, later becoming openings in the wall. These 'holes' could then be cut out by using a robot-controlled hotwire. Figure 16 lowest, demonstrates how this technique has been tested. It would be possible to conceive a production method where a concrete wall is cast with traditional mould plates, and curved blocks of foam, defining the openings, are mounted before the concrete is poured. Another possibility would be to use laser cutting or a water jet to cut out the unrolled surfaces that define the holes. These surfaces would be joined to form a continuous shape, defining the holes in the volume. This approach would be particularly relevant in situations where the tectonic principle is additive and based on assembly from sheet material, rather than from subtraction and casting.

8.2.4 Observations

The simplicity of the pattern provided insight for some of the fundamental implications of working with generative techniques in a surface based manner. The use of NURBS surfaces proved useful in terms of linking a two-dimensional method with a spatial realisation. However, there were implications from using the NURBS based methods, which arose due to the algorithms being implemented within a 3D modelling environment (in this case Python programming with the Maya program). The diagrammatic principle described earlier in the chapter was implemented in the Processing environment relatively easily and had an acceptable performance.

Working with the 3D environment in relation to the volumetric examples was time consuming, both in terms of establishing the system, and in terms of producing specific results. However, once the system was established, it was less difficult to adjust it to a particular situation. Importantly, the system proved to be successful in terms of responding to external parameters, such as orientation. This capability could have been explored further as part of the research, but hopefully future projects will continue this investigation. Besides providing functionality as a surface pattern, the method also served as a predecessor for the more complex spatial method, described in the following chapter. As such, it is a suitable starting point for explaining the basic logic used in these methods.

	Intents and conditions	Organisational logic	Realisation
Properties	Self-organising topology From 2D to 3D Consistent geometry Controllable behaviour	Isosurfacing Sequence Graduation Rule-based control Constraints	Contour crafting Robot milling High rise building Tectonic system
Observations	Formal character Contextual parameters Method specificity No a priori topology Constraints and properties	Control versus randomness Code configurations Topological limitations Extracting geometry Linearity	Technological development Scale Continuous formations Indirect geometry

8.3 Branching topologies

8.3.1 Intents and conditions

The method was developed directly from the surface pattern method, described in the previous chapter. The goal was to challenge the topological constraints that often come with generative techniques. Often it is necessary to establish a predefined system of relations, where the nodes in the system are connected in advance. For instance, if the goal is to generate a consistent mesh structure that describes a surface, the safest way of ensuring that the components are correctly connected in the final result, is to define these connections in advance. This means that only the shape of the mesh is establish through the generative process, and not the topology as such. The idea behind the described method was to overcome this limitation and develop a distinctive generative method that would be relevant for an architectural design solution. In order to achieve a three-dimensional free-formed topology, the surface pattern method was further developed and an isosurfacing technique was applied. As explained in the following section, the chosen isosurfacing technique has certain spatial limitations, which gives rigidity to the method. Again, these constraints also provide the system with a level of consistency and orientation that can potentially have great relevance in terms of realisation. Early examples generated with the *Branching Topologies* method is shown in Figure 2.

Isosurfacing is a method used for visualising data sets in higher dimensions. This method is often used for transforming a point cloud into enclosed volumes in order to give a better understanding of formations. Isosurfacing techniques are used in many different fields such as computational fluid dynamics, pharmacology, chemistry, geophysics, meteorology and medicine. The isosurfacing algorithm described in the following section, was developed by Graham Treece at Cambridge Engineering Department in 2005, for medical use. An example of the potential use of Isosurfacing is that it provides a way of visualising internal organs and skeletons from data gained through CT scanning. The computer program can be used to extract triangulated isosurfaces from a series of two-dimensional cross-sections. Figure 1 shows a model of the skin and skeleton from the female cadaver of the *Visible Human Project* data set, generated with the program. The underlying algorithm is based on a method called *regularised marching tetrahedra*. The methods *marching tetrahedra* and *marching cubes* are the most common ways of implementing isosurfacing.[1] The overall principle

Figure 1: Model of female skeleton and skin produced through isosurfacing of data from the Visible Human Project. Image: Graham Treece

1 Paul Bourke, 'Polygonising a scalar field', 1994, viewed 15 April 2012, <*http://paulbourke.net/geometry/polygonise/*>

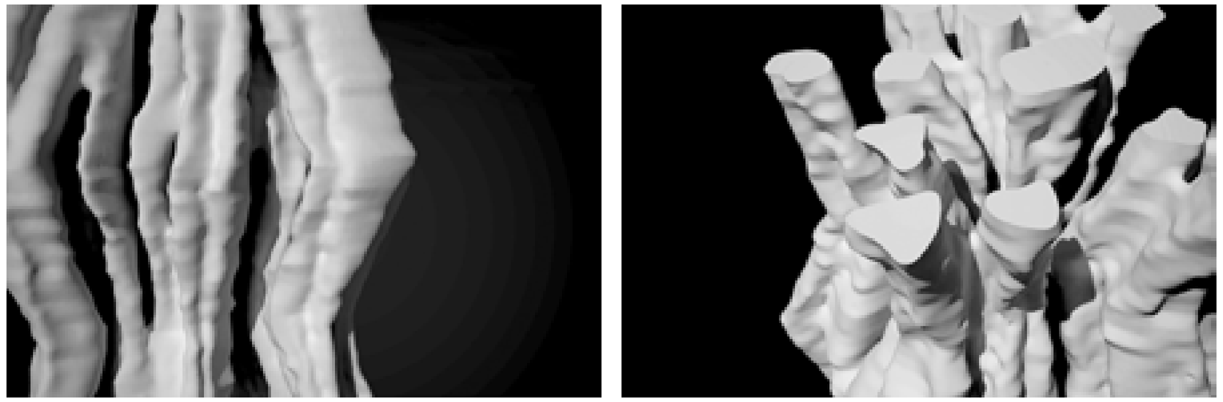


Figure 2: Early example of geometries generated with the Branching Topologies method.

is that a limited three-dimensional space is divided into cells. Then each corner point in each cell is analysed in terms of relation to a data set. This can be a specified distance to a point in a point cloud, or a colour gradient in a scanned field. If not all points are either below or above the so-called isovalue, the cell is divided with planes. When all cells have been analysed and defined, the result is a field of enclosed volumes. This method uses the principle of colour gradients. Isosurfacing is often used for visualising either collected data or data sets generated though simulation. The latter type comes closest to the way isosurfacing is used in this experiment. However, the 'simulation' does not reflect realistic phenomena, but rather generates something that is not pre-existing, based on algorithmic internal logic and design intent.

8.3.2 Organisational logic

In short, the surface pattern algorithm was further developed into a system that allows two polygons to join a larger polygon and vice versa. By carefully adjusting the process, it is possible to let the system gradually re-configure itself through hundreds of generations, allowing complex formations to occur. When the sections are translated via surfacing, a volumetric geometry emerges from the series of cross-sections. The diagram in Figure 3 demonstrates the basic process of generating the geometry. There are a number of steps in the process, mainly because the Isosurface program only accepts a certain format of data. After the cross-sections have been generated, they undergo conversion from TIFF format to RAW format. Blurring the edges allows the Isosurface program to better establish smooth transitions. The files are then merged into a binary file before they are used as input for the isosurfacing procedure. Finally, the output is of a VRML format, which can be read like a 3D modelling software, such as Rhinoceros. The following section will describe each step in detail.

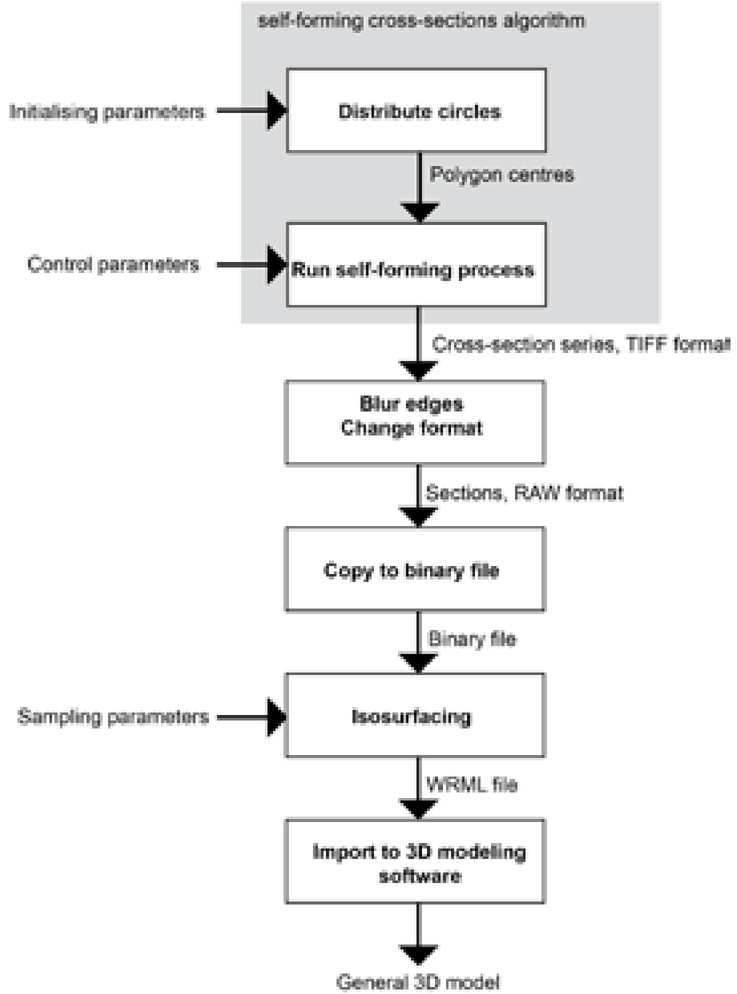


Figure 3: Basic diagram of the computational form-generating process.

The distribution of circles functions in the same way as shown in Figure 1 in Chapter 8.2, concerning *Self-organising Bezier curves*. The procedure serves to distribute the pattern evenly in the specified field. The algorithm does not automatically adjust directly to a field size. Rather the amount of and distance between centres is specified depending on the intent. The centre points are distributed randomly, and after balancing out the distances in order to create an ordered field, the centre points positioned outside the field are culled. As with *Self-organising Bezier Curves*, this process is performed as a type of initiation of the following form-generating process. Both the distribution of centre positions and the generation of polygons and Bezier curves are random. However, the system can be initialised with a random seed parameter, which allows an exact repetition of the process if the parameters stay unchanged.

The self-forming process begins with Bezier curves inscribed in polygons as shown in Figure 2 in the previous chapter. The main difference is that two polygons can join into one and vice versa. These procedures can become activated when a vertex of one polygon approaches an edge. There is still a tendency for the vertices to slow

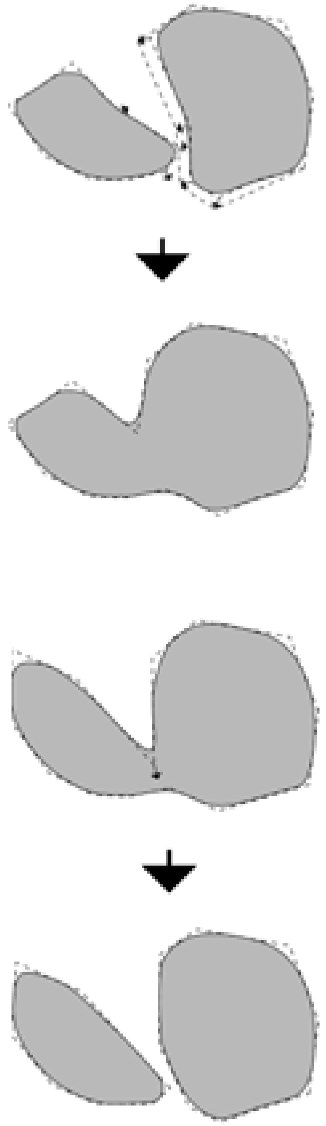


Figure 4: Topological transformations. Top: Two joining polygons. Above: Two splitting polygons.

down when they approach. But if the distance between a vertex and an edge exceeds a threshold, and the geometric conditions allow it, a joining procedure can take place (see Figure 4) Conversely, it is possible for a polygon to split into two polygons. The join procedure implies the loss of a vertex, where the splitting adds a new vertex to the geometry. Geometry balanced between joining and splitting will keep a constant number of vertices. If the polygons are generated from few points, as seen in the illustration, the operations change the topology in a relatively abrupt way. Because of the versatility of the isosurfacing algorithm,(explained in the next section), the spatial mesh geometries display smooth transitions. An important feature, necessary for stabilising the system, is a delay on the joining and splitting operations. The feature ensures that a certain amount of steps is processed between each re-configuration. This eliminates the constant flickering between joining and splitting, which would otherwise corrupt the system. The mechanism works by resetting a polygon's 'age' after a re-configuration, and checking whether the involved polygons are 'old enough' before the operation is carried out.

The vertices do not automatically grow away from the polygons' original centre (as with *Self-organising Bezier Curves*), since the idea of a centre dissolves as the polygons re-configure and change position. Instead, other types of movement forces are added in order to establish a dynamic system. Each vertex is gradually moved in a direction, thus reflecting its velocity. The velocity is a vector describing the distance and direction that the vertex is moved in each step, or frame. The velocity is gradually built up and modified by adding an acceleration vector, calculated in each step. The acceleration vector is calculated by adding all forces affecting the vertex in its current position. In this sense, the system is actually an agent-based system, similar to those described in the following chapter. The forces acting upon the vertices are of a different character. Some relate to neighbour vertices, both in the same polygon and in other polygons. Others relate to the polygon level or to the global level.

One of the forces acting upon the vertices is attraction to another polygon, as seen in Figure 6. Each polygon is 'trying to' approach another polygon, which again is trying to approach a third polygon, and so on. The attraction is distributed on a vertex level, so each vertex from one polygon is not attracted to the same vertex in another polygon. Although this logic does not make sense in relation to the simulation of physical phenomena, it serves to activate movement in the system. It allows the polygons to stay in relation to the group of polygons, similar to the coherence function described in relation to agent based systems. Increasing the attraction force accelerates topological re-configurations, that is: join and split

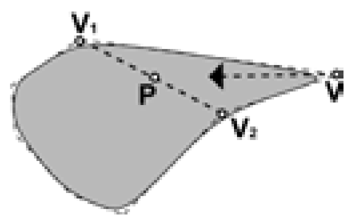


Figure 5: The averaging force function seeks to move the vertex V towards the midpoint P between neighbour vertices V_1 and V_2.

operations. Another force at work is the alignment function. It is based on a simple principle where each vertex is adjusted in order to balance the distance to its neighbours. Each vertex is repositioned in order to get 'in line', as shown in Figure 5. The purpose of the function is to avoid acute angles and large differences in edge length. If the function is not counter-acted by other forces it will cause the polygons to shrink. Although seemingly disadvantageous, in combination with other forces, the problem is less and can be seen as an improvement.

In terms of external control of the system, the main variable is area calculation. The area of each polygon is calculated in each step and summed up to give a total cross-sectional area of the system at the current state. The area of the cross-section indirectly represents the volume size of the corresponding 'slice' of the geometry when the isosurfacing procedure has been performed. Therefore, the area calculation can be used to regulate the volumetric size of the final structure. The area calculation is analysed for each polygon on two different levels. One force reflects the local situation on a polygon scale. This responds to the polygon size compared to a global setting of minimal polygon size and maximum polygon size. If the size goes beyond the limits, the result is a force directed orthogonally outwards or inwards on the polygon, according to the detected problem, as seen in Figure 7. If the other forces affecting the movement of the vertex do not supersede the response to area, the polygon will grow or shrink and approach the acceptable area. If a polygon is decreased in size, below a critical limit, (normally half the accepted minimum size), the algorithm removes the polygon. The state of the global system is analysed in a similar way. If the maximum or minimum values for total area of a cross section are exceeded, a responding force is added in order to stabilise the system, as shown in Figure 8. In some conditions, because of the joining and splitting operations, a polygon appears inside another. In this case the smaller polygon will be interpreted as a 'hole', or a hollow part of the structure. The area of the hollow part is then

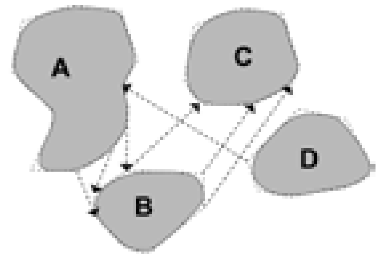


Figure 6: Basic diagram of the computational form-generating process.

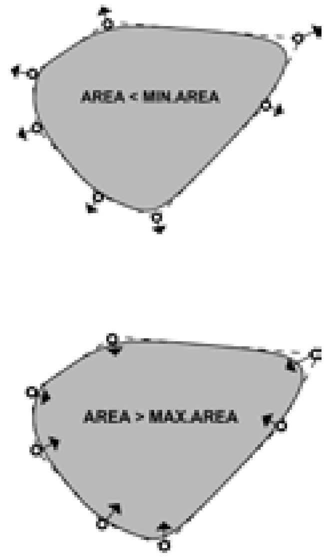


Figure 7: Above: Each polygon is evaluated in terms og local area size and the forces are applied to the velocity of the vertices if the values are exceeded.

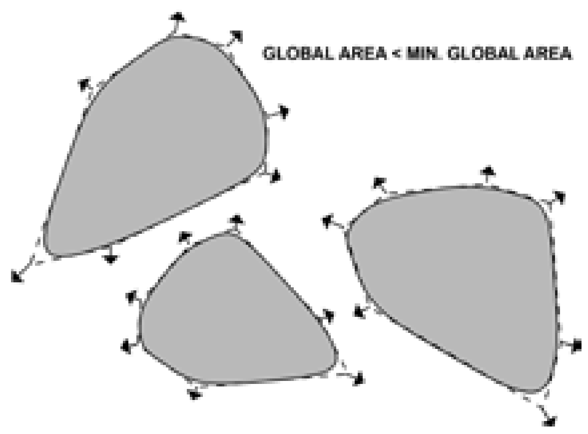


Figure 8: Each vertex is affected by forces responding to the size of the global cross-sectional area.

Figure 9: A series of generated cross-sections.

extracted from the global area of the cross-section, but not from the calculation of the individual polygons. In a situation where the exact local area is crucial and hollow parts appear, the calculation must undergo development. These parameters can be regulated in certain stages in the form-generating process. This occurs through changes in the code, but a more interactive control could be added if needed. Figure 10 demonstrates an example of a generated cross-section, where the area sizes are shown during the form-generation process.

It is possible to interactively add, move and remove circle shaped attractors, while the system is running. This advantage concerns the development of a setting. When the positioning and size of the attractors have been decided, it is much more useful to predefine the setting, which is also possible by adjusting the code as with the other parameters. An attractor function adds a force to the vertex acceleration, making the vertex move away from the centre of the attractor. This allows the generated structure to form a type of void or enclosure.

Besides the acceleration controls for the vertices, some rigid controls are embedded in order to adjust the re-configuration of the polygons. In some cases, the edge length becomes relatively short, despite the averaging function. In this case, the algorithm removes a vertex and connects its neighbours instead. Also, if an edge line exceeds a certain length, it is divided into two edges. In other words, a vertex is added at the mid point of the edge. More general forces are applied in order to adjust the positioning of the polygons. Without these functions, the system tends to move outside the displayed field, thereby escaping the following isosurfacing procedure. One function adds a force, pointing towards the centre of the field. Another function reduces the vertical field size in order to make the vertices stay in a certain horizontal zone. This function is particularly valuable when generating a wall-like structure. As with agent-based systems, a range of vision for the vertices was implemented. This means that other vertices outside a certain radius are not taken into account in terms of calculating the acceleration vector for the vertex. In this system, the movements are not a result of local collective behaviour as with bird simulation, which is explained in relation to agent-based systems in Chapter 8.4. Here, the range of vision is used

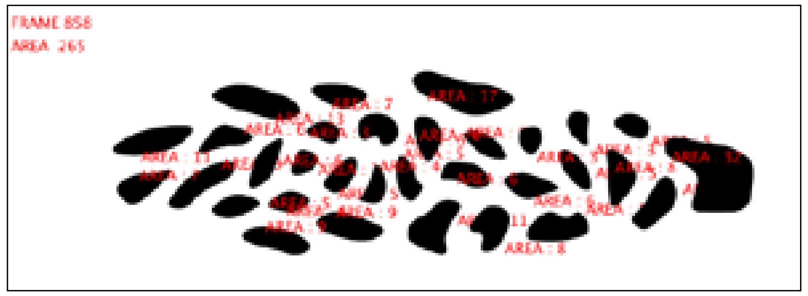


Figure 10: The algorithm keeps track of local and global area of the polygons.

RANDOM GENERATOR	
Random seed:	Index for initialization of the random generator
RECORDING SETTINGS	
Number of frames, first, last frame and step interval	Usually the system runs for 300 generations or more before the 'recording' is initiated
	There are typically 30-50 states between each saved cross-section
WINDOW SETTINGS	
Width and height of registered field	Reflects the number of pixel in x and y direction
Height of horisontal zone	Effectively the depth of a 'wall' structure
FORCE WEIGHTS, VERTEX ACCELERATION	
Seek polygon	Weight for approaching another polygon vertex
Average position	Weight for smoothing the geometry with shrinkage as sideeffect
Keep area	Weight for adjusting polygon area, locally and globally
Stay in horisontal zone	Weight for staying within 'wall' boundary
Internal separation	Weight for keeping distance to vertices in same polygon
External separation	Weight for keeping distance to vertices in other polygons
Attractor response	Weight for response to attractors
Centre force	Weight for moving towards field centre (simple coherence)
Damping	Small reduction of velocity
Maximum velocity	Maximum distance for movement of vertex in each generation
Maximum force	Maximum vector length for changing velocity in each generation
Range of vision	Maximum distance to other polygons for collision test
GENERAL CONTROLS	
Global maximum area	If total area is above, all vertices move 'inwards' in their polygons
Global minimum area	If total area is below, all vertices move 'outwards' from their polygons
DISTRIBUTION CIRCLES SETTINGS	
Number of circles	Amount of centre points before distribution
Minimum diameter and diameter range	Range of distances between circles (centre points)
POLYGON SETTINGS	
Maximum number of polygons/Bezier curves	If number of polygons reaches this value, splitting is not performed
Minimum number of polygons	If number of polygons reaches this value, joining is not performed
Generations before split or join operation	Number of generations before the polygons can split or join
Local minimum and maximum area	Control of area size
Minimum and maximum edge length	Control of edge length
Minimum internal vertex distance	Limit for distance between vertices in the same polygon
Minimum external vertex distance	Limit for distance between polygons
Approximate initial number of polygon points	Number of vertices in each polygon

Figure 11: Table of parameters that affects the form-generation.

for optimization of the calculation time of the system. Because the movements are small, the polygons only have to be evaluated with respect to their nearest neighbours, regarding collision detection. The list of nearby neighbours is updated with a fixed interval, such as 50 or 100 generations, allowing a noticeable improvement in running time of the system. In Figure 11 is a list of parameters that are directly adjustable within the system. The parameters are set prior to the generative process. However, as with the area sizes, a simple interface could have allowed interactive adjustment of the parameters if needed.

8.3.3 Isosurfacing procedure

As shown in Figure 3, the process of creating a 3D model with a consistent mesh has a few steps to it. The actual form-generation happens in the process, described above, where the procedure of generating a volumetric geometry will be described in the following. While the procedure cannot change the geometry as such, it is important to notice the constraints that automatically follow from using the method. As it will show, the procedure is not entirely fixed, since certain parameters can be adjusted. As explained, the outcome of the form-generating process is a series of 2D sectional drawings. A few drawings from a series is shown in Figure 9. Since the algorithm draws defined geometry, only black and white colours appear, describing the interior and exterior spaces in the field. The isosurfacing algorithm reads the drawing as a graduated field of gray tones, where values below a certain limit are read as being

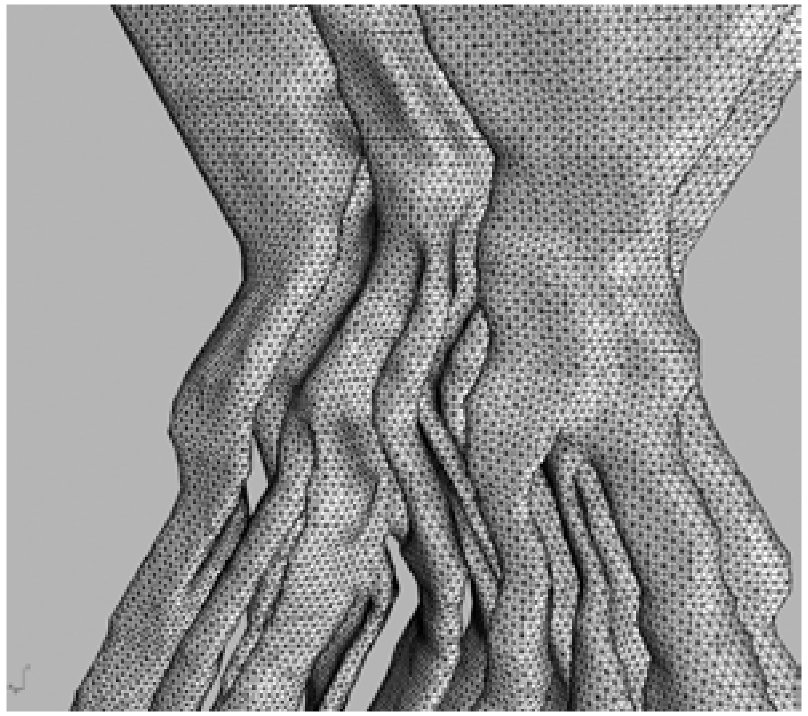


Figure 12: An example of a mesh, generated with the isosurface program.

part of the massing. The algorithm uses the graduation to establish smooth transitions between the cross sections. Since the output from the previous process does not generate these gradients, the images are imported to a photo-editing program where the contours are blurred before the isosurfacing procedure is carried out. The images are transformed from TIFF format to RAW format. The Isosurface program, written by Graham Treece, runs in the DOS operating system, and all the sections must be packed into a single binary file before they can be used as input. The program accepts different parameters, controlling how the meshes are generated. The primary adjustments are the resolution of the mesh and the relative proportions of the mesh. The program can generate meshes of high resolution, and takes a high number of sections as input. The question of resolution is therefore mostly related to difficulties in handling large data amounts in the following processes of handling the general design and realisation. The program produces a consistent mesh from the data, and saves the geometry as a WRML file, which then can be imported into a 3D modeling software. Figure 12 shows an example of a mesh, generated with the isosurface program. The resolution of the mesh is visible as a surface grid.

8.3.4 Branching topology experiments

The following describes a series of experiments with the described method. The focus has been to explore both the inherent properties of the system, but also to investigate to what extent the form-generating process can be controlled by external parameters, and thereby identify potentials and issues concerning use of the method for architectural design. The system has been studied in different scales, and a variety of behaviours have been developed. Some of the changes was achieved primarily through adjustment of parameters, where in most cases, these changes also lead to a change or improvement of the code for the program that generates the geometry. In all the experiments, the generated drawings were interpreted as horizontal sections, beginning with the bottom section first, then reading each drawing as sections step wise upwards throughout the geometry. While, the isosurfacing program very well handles geometry that ‘appears’ and ‘disappears’ during the sequence of sections, this ability has not been used in most cases. Rather, the column-like property of the first studies, where the structure has a ‘cut off’ bottom and top, has been kept as a general property.

The first series of experiments concerned investigation of using the method for generating a type of wall structure. The dynamic branching character, inherent in the algorithmic method, suggests a type of structure that is neither a solid wall, nor a row of columns, but rather something in between those types. And as demonstrated, it was possible to generate structures that reflect a kind of transition from wall type to column type. Figure 13 shows how a facade with such type of column-wall could appear, and below, in Figure 14, the interior space is rendered. As the algorithm essentially functions randomly, it was necessary to find methods of controlling the behaviour. The first step was to redefine the 'design space'. In other words, because the goal was that it should be possible to inscribe the final mesh structure in a long shallow box, the projected cross sections was a long shallow rectangle. A simple method would be to stop the motion of the vertices as soon as they move outside this limit, but this would result in entirely flat parts of the formations in these areas. Since a more fluid character was sought for, a different method was used. Instead, as described earlier, a force was added that gradually pushes the vertices back into the field, when they move outside it. This results in a more smooth regulation of the field, without abrupt change in character in the border zones. Figure 15 shows how the geometry is kept inside a narrow space and creates a type of wall formation. The rightmost wall end is straight because the formations in this case moved outside the image frame, causing the isosurfacing program to abruptly flatten out this part of the geometry, reflecting the problem mentioned above. The form-generating process calculated 300 states before recording the first frame. As with all the following examples, this was done because often the initial geometry is randomly positioned, and during the first generations it is adjusted to meet the specified parameters, particularly in terms of field size. The wall formation experiments are all formed from 50 generated cross sections. Typically there are approximately 1000 states in the form-generating process, which means that there is about 20 states between each recorded frame, or cross section.

One of the wall experiments was about enabling some type of gradual scaling of the structure. The goal was to establish a way of regulating the thickness of the 'trunks' in different parts of the wall formation. That is, in a zone where the thicknesses of the branches were reduced, the number of branches was correspondingly increased. Still, the total volume in the thinned areas would be much less than in the more massive areas. As figure 16 shows, this graduation was carried out in the beginning of the process. In fact, the difference in scale is settled in the initial distribution of the polygon centres, where also the radius of the circles and the distance between them is defined. A more general implementation would base the graduation on some type of attractors, which also will be discussed later in the chapter. In this case, just to demonstrate the principle, a simple function based on a circle's position gave a size which again was transferred to the initial polygon. Also, the maximal

Figure 13: Visualisation of an abstract facade where a type of column wall has been generated.

cross-sectional area was evaluated throughout the form-generation, ensuring that the polygons would adjust in size, according to their position in the field. Probably due to secondary parameters, such as number of generations between re-configuration of a polygon, there seem to be limited amount of splitting and joining of branches. However, this was not a focus for the experiment, and has not been analysed further.

As a continuation of the previous experiment, a method enabling a different type of graduation was developed. In Figure 17, left, the graduation was not resolved initially, but rather as a type of transformation process from one volumetric scale to another. To

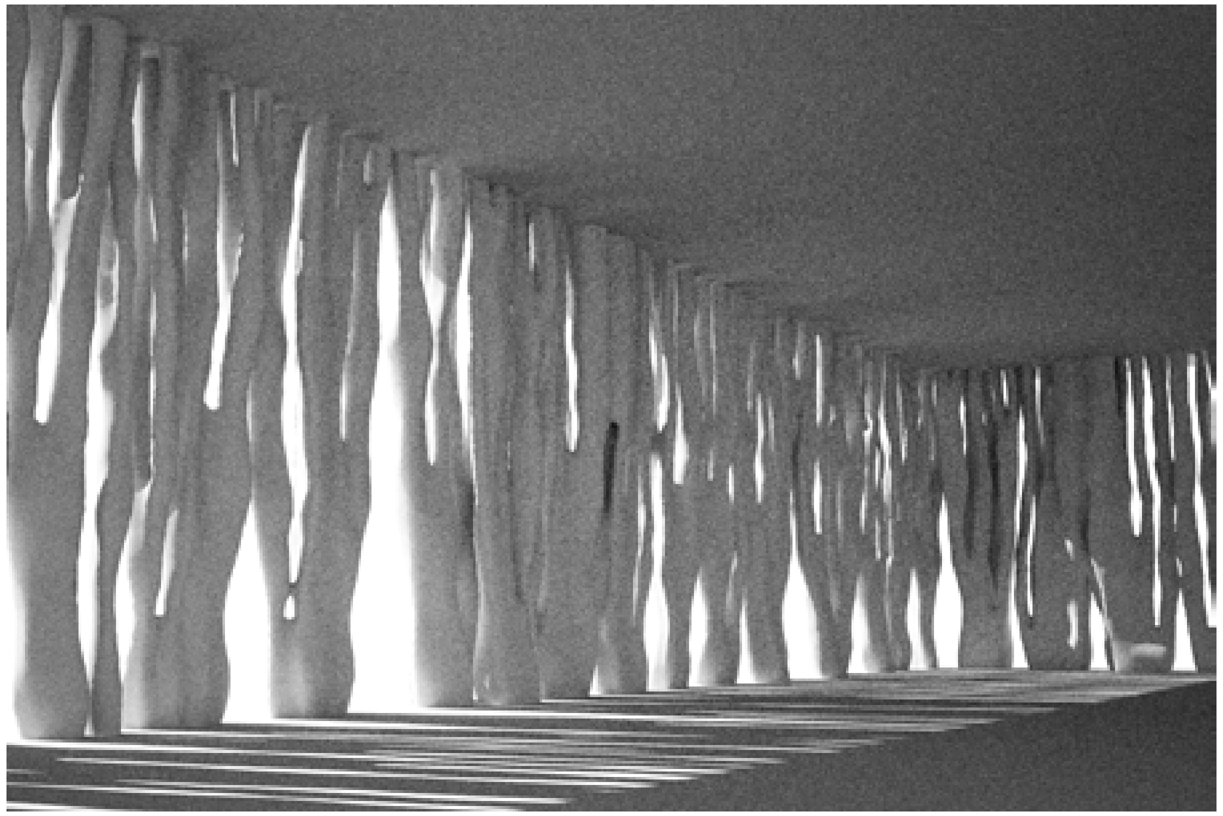


Figure 14: Visualisation of the interior of a column-wall enclosed space.

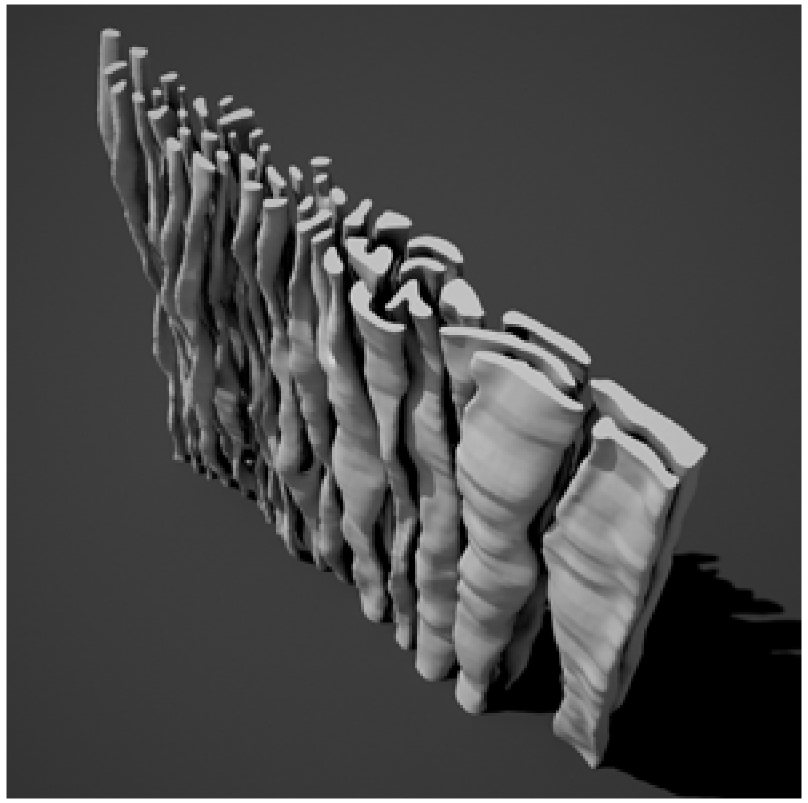


Figure 15: When the geometry is constrained to a narrow space, a type of wall formation is generated.

establish shifts in the system behaviour, a difference between the parameters controlling the initiating of the system and parameters controlling the form-generating process was established. The polygons were initiated with large areas, whereas, the minimum area size was set to a low value. Also, the possibility of joining polygons was cancelled. In fact, the behaviour that had the greatest impact, in terms of making the polygons split into smaller polygons, was increase in weight of the parameter controlling the attraction between the polygons. This can be explained from the fact that the polygon attraction function is the main dynamic force in the system, and the more the vertices move around and approach polygon edges, the more likely they are to cause splitting operations. Conversely, the result shown in Figure 17, right, was generated from a version, where the initial state consisted of many small polygons, the splitting operation was cancelled, and the weight of

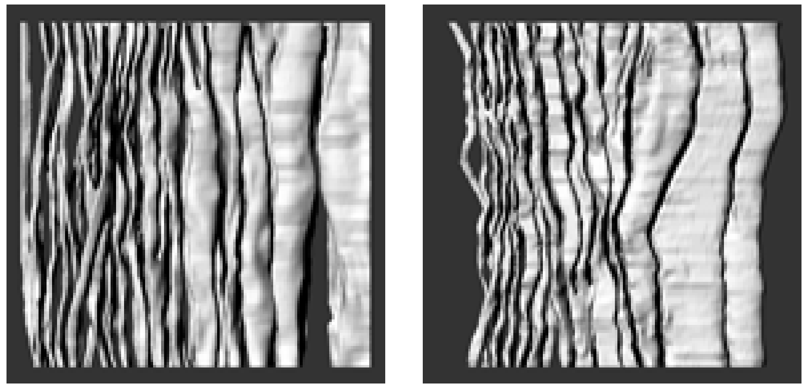


Figure 16: Gradual scaling of the structure in the initial distribution of polygon centres.

attraction force was high enough to ensure re-configuration of the polygons. The strong attraction also has a side effect in that it results the polygons creating a funnel shape. The reason why the system manages to spread in the top, is partly the area function, seeking to increase the area. More important, the polygons have merged into only on continuous geometry, meaning that the attraction function is cancelled out. Further development and adjustment of the system would most likely enable a pattern where the polygons can develop from small to bigger cross-sectional areas without forming a funnel shaped structure.

One of the aspects investigated in relation to the wall type experiments was the stability, or smoothness, of the generated geometry. In the early examples, such as those in Figure 2, the shape of the volumes tended to reflect a sort of oscillation, which later analysis revealed to stem from intense regulation of the cross-sectional area of the branches. Figure 18 shows how negotiations on different scales can affect the outcome. The example on the left shows a system where the global area of the cross-section is constantly either growing or shrinking. Only when the area reaches one of the limits, the behaviour is reversed, but never stabilises. This gives an undulating effect, which was recurring in most of the early experiments. Figure 18, middle shows a setting, where the effect mentioned before has been reduced, by eliminating the forces that regulate the area size when the total area is within the limits. Still, an undulating effect occurs due to a continuous negotiation of total cross-sectional area. This happens because some forces, such as the force that averages out the polygon angles, has a shrinking effect on the movement of the vertices. When the area size moves below, the area regulating function returns a force directed to increase the polygon size, counteracting the former. This behaviour is then constantly turned on and off, depending on the size of the area of the cross-section. Both effects were avoided in the example on the right by reducing the weight of the area-controlling forces. Then, the minimum area was set to a percentage higher than the ideal size, so the system is actually constantly below the limit, 'trying'

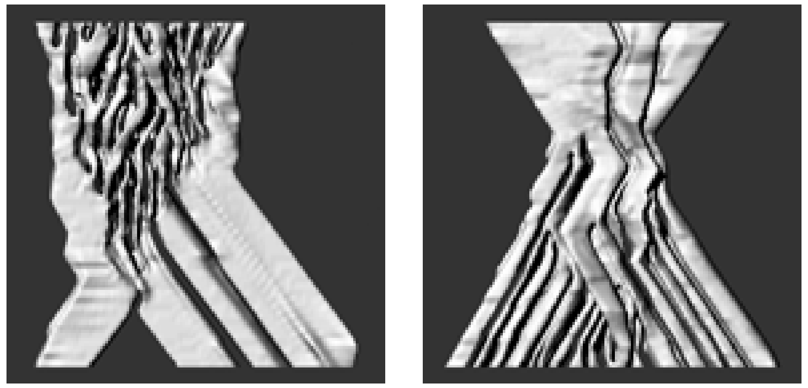


Figure 17: Gradual scaling of cross-sectional area of individual branches from one volumetric scale to another. Left: Decreasing cross-sections. Right: Increasing cross-sections.

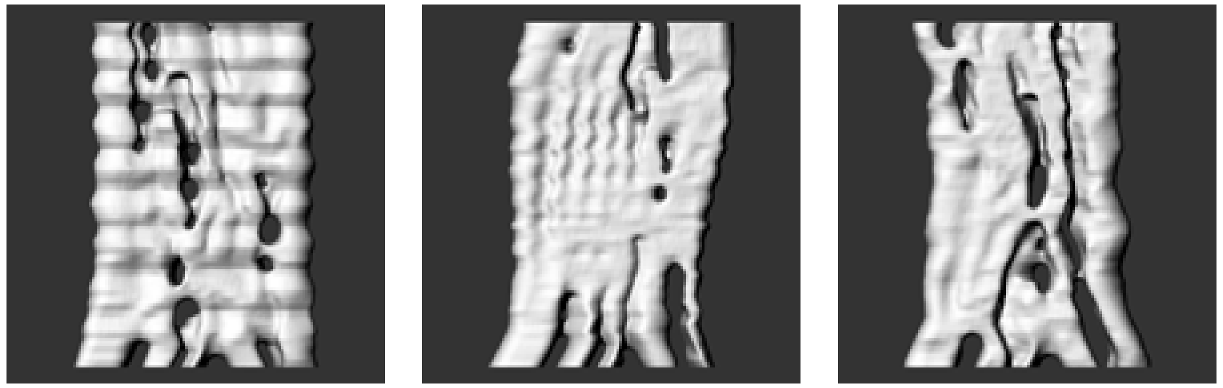
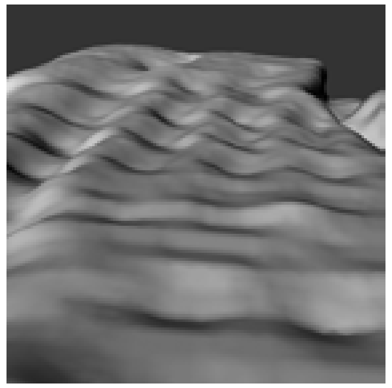


Figure 18: Negotiating the cross-sectional area at different levels.

to increase in size. Then, the abrupt shifts in behaviour that causes undulating effect are avoided. It is shown here, because it is possible to imagine situation where these effect are actually desirable. As indicated in Figure 19, it would be possible to enable the effects to occur on a local scale by adjusting the system. Figure 20 shows a development sequence, where a different type of solution has been tested regarding the same problem. On the left is the initial formation, almost a massive wall, clearly showing undulating effects in the mesh geometry. In the middle, the velocity of the vertex movement has simple been reduced, removing the undulation, but as it shows, also results in a radically different type of expression. On the right, the velocity has been reduced even more, leading to almost completely smooth geometry, but also with very limited topological and geometric variation.

Another experiment sought to investigate the systems capability of accommodating a basic type of spatial design intent, understood as a capability to relate to predefined voids. The voids could be seen as representing larger spaces in a building structure. In this experiment, the attention was directed only towards the behaviour of the system. The main interest was whether the system was capable of responding to existence of one or more attractor points, defining zones in the field that should be kept free of structure. In Figure 22, the attractors are shown with a green colour. The attractors repel the vertices, thereby establishing voids in the otherwise relatively occupied field. One of the main questions that investigated was the systems capability of adjusting to appearance and disappearance of attractors during the process. The image sequence shows how it takes more than 200 generations for the structure to clear the space for the attractor already present from the first frame. In the last part of the sequence, the small attractor disappears in fram 596, and the void is not dissolved until frame 827, more than 200 generations later. Figure 21 shows the volumetric outcome of the process. It was possible to generate a formation that appeared relatively dense from the outside and contained a type of internal space, formed from the two attractors. Depending

Figure 19: Undulating effect at a local scale.



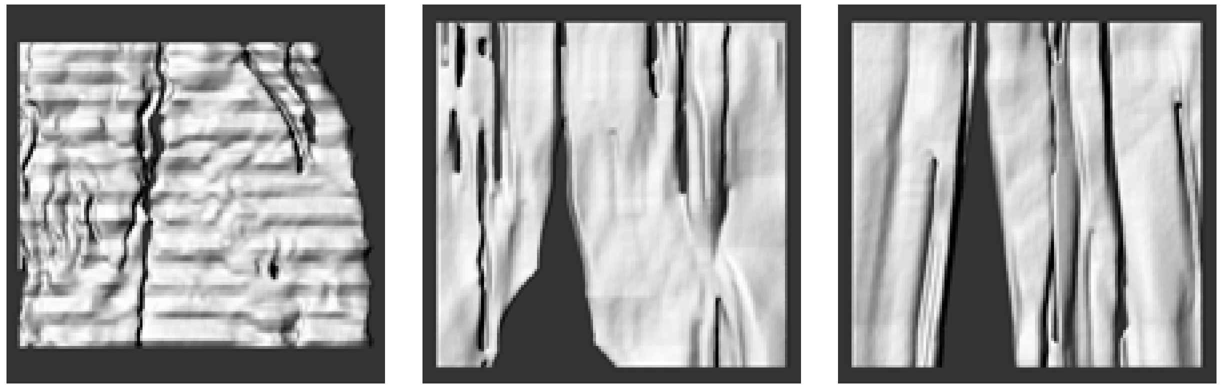


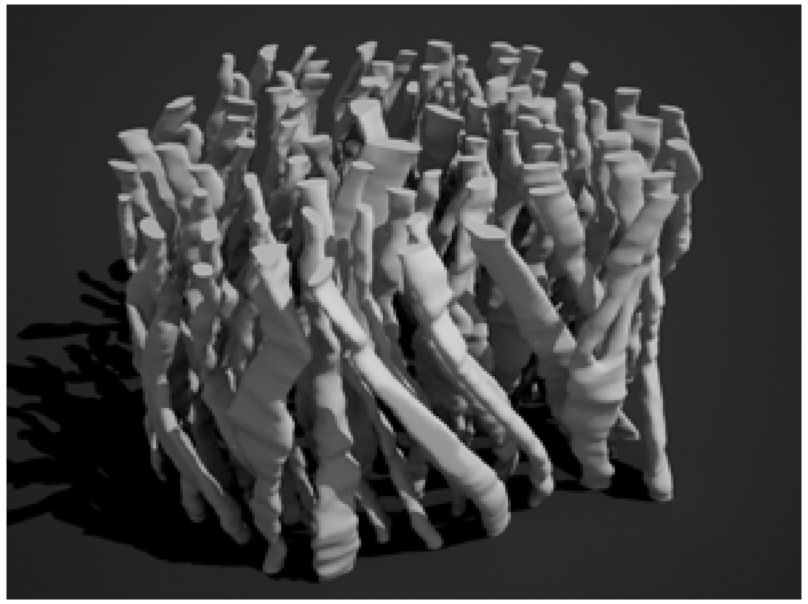
Figure 20: Smoothing the geometry through reducing the force vectors also affects the morphology.

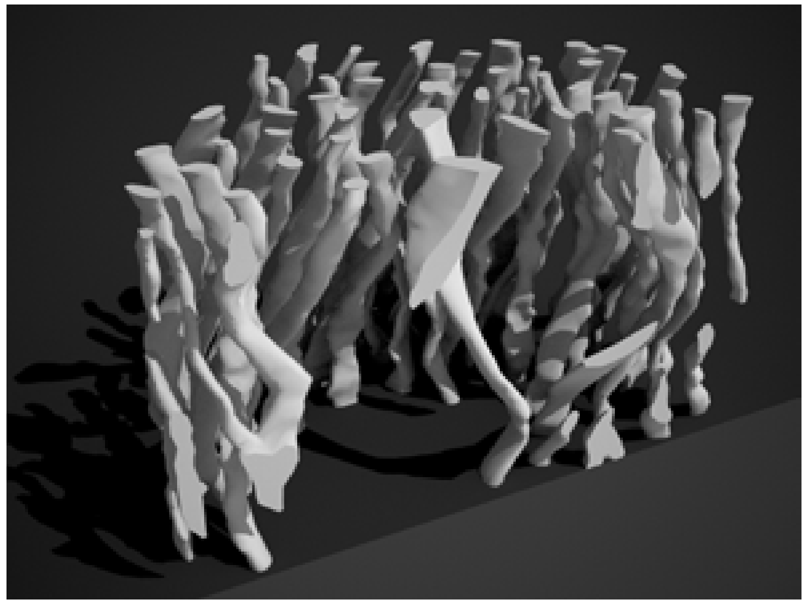
on the further articulation of the geometry, an architectural project would probably benefit from much lesser density in the formation. However, in order to demonstrate the mechanism, it was necessary to keep a certain density. Otherwise the difference between void and dense structure would be difficult to detect. In this sense, the demonstration showed that the system to some degree is capable of adjusting to an architectural program, but also that implementation of this behaviour in a specific project most likely would depend on further development and re-configuration of the system, to meet the requirements of the task.

A different type of experiment, suggesting use of the system on a high-rise building scale was carried out. The intent was to investigate if the system could be used as a tool for shaping of the overall form of such a project. The goal was not to arrive at an optimised method for modelling the project. On the one hand, if the shape was preconceived, and the problem was to specify the geometry in three dimensions, it would be relatively simple to use a normal explicit modelling technique. Furthermore, the system is difficult to control, so the efforts saved by automating the modelling would easily be spend on adjusting the system to generate the desired model. Rather, the question was whether a process of adjusting the system could be an integrated part of a design development in a way, where unexpected formations would appear and suggest topological and formal outcomes that perhaps would not occur with a more direct design approach. Simultaneously, because of the inherent logic of the system, a large variety of solutions could be tested, all with the same embedded logic, ensuring that certain structural properties are present. Furthermore, a hypothesis was that because of the systems was based on specific mathematics, the generated solutions would have a direct link to a possible realisation through translation of this logic into manufacturing data. This claim is perhaps not strongly underpinned because of the translation through isosurfacing, which will be discussed later in the chapter. It was decided to use the systems inherent capabilities of negotiation topology, to develop a high-rise building that differs from

Figure 21: Example where a spatial zone defined by attractor points is avoided. Top: The generated structure. Bottom: Section through structure, revealing the void.

Figure 22: Opposite page. Image sequence with attractor zones shown as green circles.

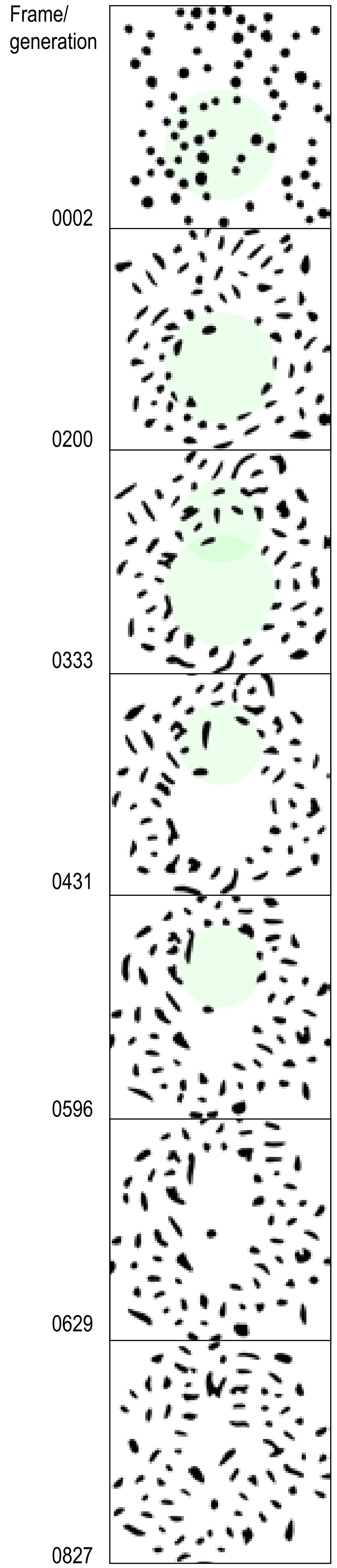




general typological approaches. The main inherent properties that were expected to inform the project were mainly four kinds:

1. The structural logic. All parts of the structure are supported by parts below. The whole structure is connected which allows ‘branches’ to support each other.
2. Infrastructure. The connectivity of branches melting together also provides a possibility of implementing a complex infrastructure, allowing the program of the building to adjust fluently, and the users to move across the branches.
3. Control of area. As mentioned earlier, the systems behaviour is to large extent based on control of the cross-sectional area of both the individual arms, and the area of the whole cross-section. This is useful in terms of ensuring both that each part of the building meets some standards for depth. Also, a desired total area can be specified and approximately fulfilled in the generated models.
4. Spatial configuration. Because of the system’s behaviour it ensures that in most part of the system there are certain distances between surfaces that are maintained, both internally and externally. This means that despite the seemingly randomness of the geometry, it is possible to make sure that the building shape functions well in terms of airflow, natural light and views.

The design process was initiated by adjusting the parameters to meet a ‘realistic’ setting based on the above mentioned properties. Without specifying a program for the building, it was thought of as a commercial building complex. A certain dynamic, which would encourage re-configuration of the topology throughout the generative process, was ensured, by giving the attraction between polygons sufficient weight. The area limits were defined as maximum 2600 and minimum 2400 m^2 per floor. A polygon shape, which would represent a separate floor plan should have more than 100 m^2. There were 38 floors, the floor height was set to 4 m, and the total building height was approximately 150 m. The total floor area of the building complex would then amount to approximately 90.000 m^2. 80 frames were recorded from the form-generation process, roughly representing a distance of 2 m between each cross-section. A range of 3-5000 generation of geometric states were run through for each version, depending on the behaviour of the system. The minimum distance between vertices and edges, reflecting the distance between the building facades, was set to 7 m, which also



was the minimum building depth. In reality, the units could not be given directly to the system, because all the parameters affecting the behaviour was originally defined as relative, without basis in actual units. Therefore the settings were adjusted by analysing examples of generated cross-sections, based on the input parameters. It would though be possible to fine tune the system to make it reflect precise units. Figures 24 to 28 shows a development process, reflecting the search for a possible answer to the proposed problem.

The first test, shown in Figure 24, left, was clearly affected by the undulating effect, previously mentioned. Simply be reducing the maximum acceleration of the vertex movement, the geometry became noticeably smoother, as shown in Figure 24, middle. Despite the reduction in acceleration there is still movement enough to make topological re-configurations. A number of parameters was changed to arrive at the next example on the right, but the main difference is that the random generator was initiated with a different number, which means that the random positioning of the initial polygons is changed. The next example in the development is shown in Figure 25, left. Here, the random generator was adjusted back to same setting as the first two examples. A range of parameters had been adjusted, and this resulted in the rough undulating geometry, even more 'out of control' than in the first example. The difference to the next example, figure 25 right, was basically a tightening of the design space. The field was decreased a in size and the minimum area for individual polygons was increased to 400 m^2. These more external adjustments seemed to indirectly reduce the roughness of the geometry. In Figure 26 the smoothness was increased by directly reducing the movement of the vertices. The velocity was set to 25 % of the setting in the previous experiment. The random setting was changed again, which again radically re-configured the geometry. While, the surface became smoother, also the topological re-configurations almost stopped appearing. Therefore, the polygon attraction was increased 1/3 and the maximum velocity of the vertex movement was doubled. The result, shown at the right in Figure 26, was accepted as a balanced response to the design intentions. Subsequently, a series of experiments was performed in order to investigate the potentials of changing basic parameters while keeping the achieved balancing of the system. Figure 27 shows a series of variations where only the setting of the random generator was changed. The behaviour is similar in the examples, but the outcome was topologically different. The version on the right was chosen for further detailing. A visualisation that vaguely suggests a building scale is shown in Figure 23.

Another series, based on the same setting as the final version of the tower, was about investigating ways of distributing the floor area. See figure 28. The example on the left was programmed to shift the total area of the cross-section from 30% above the average in the first 800 generations to 40% below the average in the last 1900 generations. The thinning out of the structure is clearly visible in the outcome. The middle image shows the opposite

Figure 23: Visualisation of high rise building experiment. A scale is vaguely suggested.

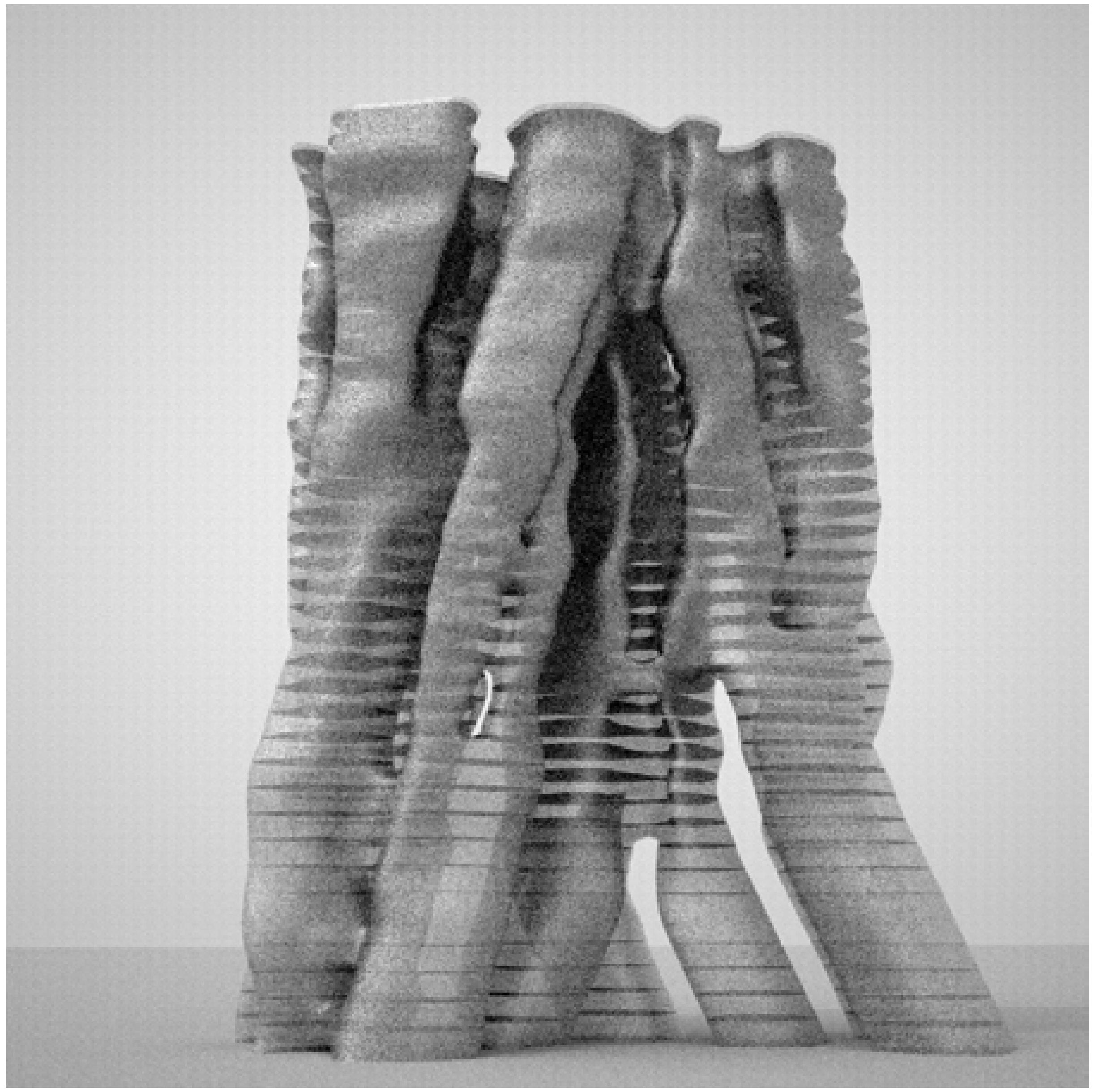
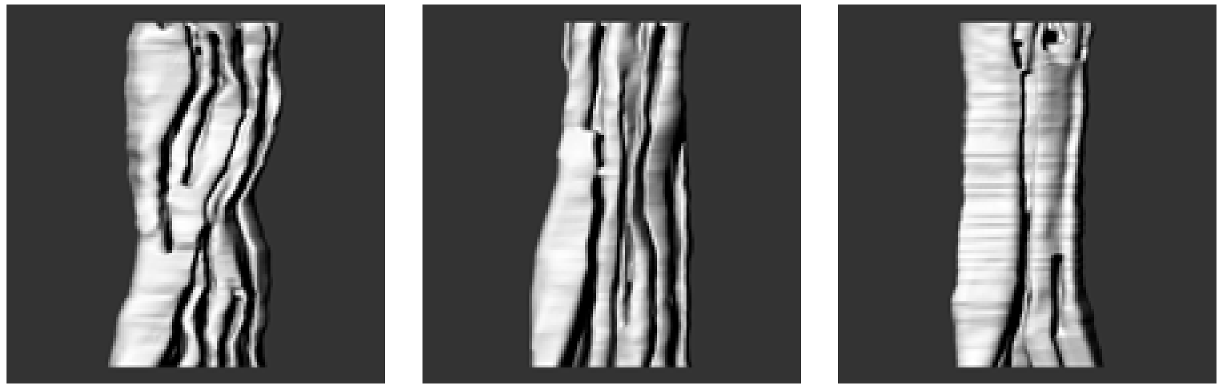


Figure 24: First examples of volumes for a high rise building



situation. Here, the area is reduced in the first generations and increased in the last generations. The structural consequences of this organisation would obviously be noticeable, but was not further investigated. The example on the right shows a more crude way of regulating the area. Here, a polygon was simply removed for every 350 generations, and the maximum total area for the cross-section was reduced correspondingly. Because the system enables the polygons to constantly divide and form new polygons, it took several attempts before an example that stopped its growth, before reaching the final frame, was produced.

8.3.5 Realisation

The question of realisation has only peripherally been part of the basis for this series of experimental development. However, there are developments in building manufacturing that suggest a future potential in combining the displayed method with specific realisation methods. In cases where the geometries are considered as massive supporting building parts, similar to columns, it would be logic to consider concrete, and particularly fibre-reinforced concrete, as building material. The fibre-reinforcement would support appearance of fine details in the structure. During the recent years, a variety of methods for casting advanced geometries in concrete, have been developed. Some are still under development, and only on the way to implementation in architectural projects. A couple of realised

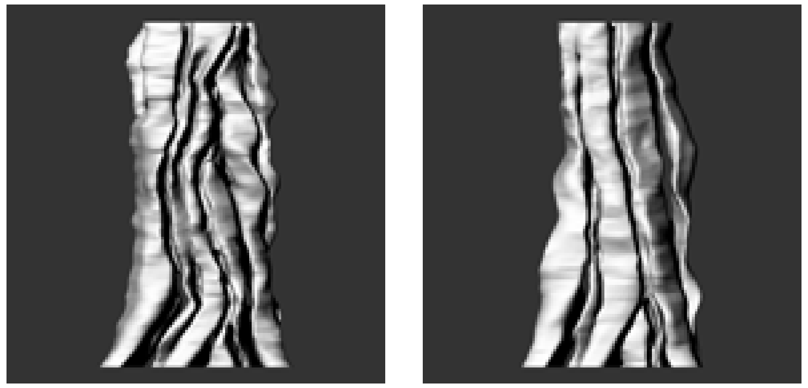


Figure 25: High rise examples. Left: undulating effect. Right: Tightening of the design space.

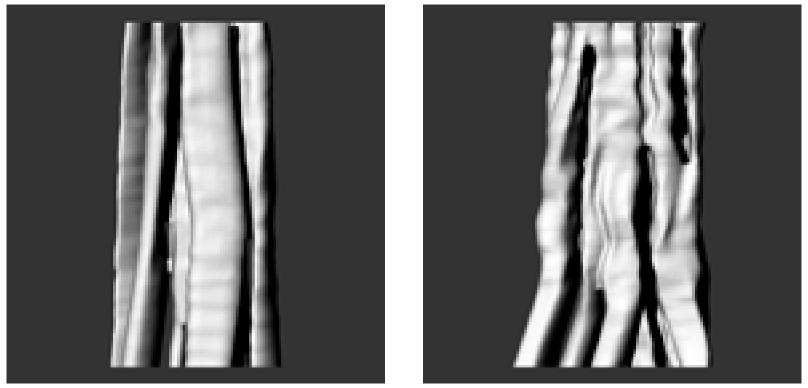


Figure 26: Left: Increased smoothness by reducing movement forces. Right: Increased dynamic through settings.

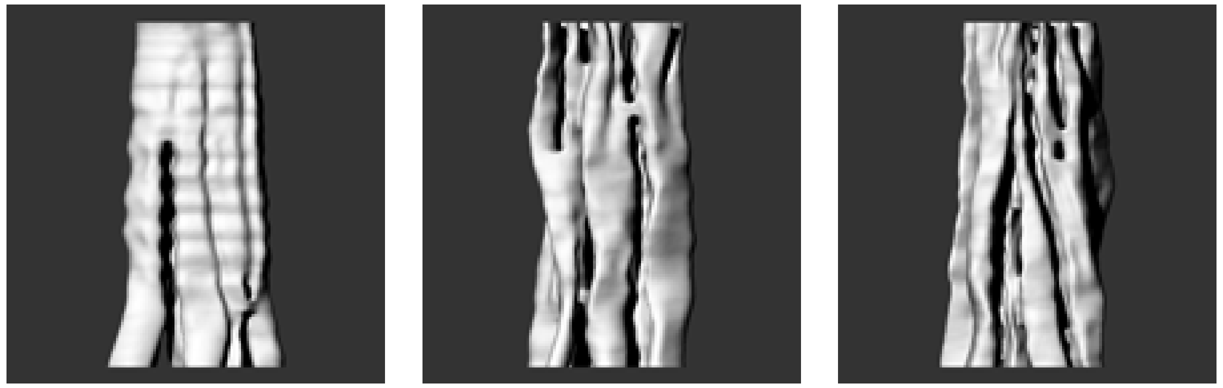


Figure 27: High rise examples. Variations where only the setting of the random generator was changed.

architectural projects does in a way reflect 'state of the art' in terms of what degree of complexity is possible within a relatively standard construction process. Baumschlager and Eberle's *Nordwesthaus* has a structure of organically shaped concrete walls. There are similarities between the project and the wall examples shown earlier, where the structure can be characterised as both columns and walls. There is a certain degree of complexity to the geometry, suggesting that a realisation of the system proposed here also would be possible. An important difference is though that in the Nordwesthaus the structure is two-dimensional, in the sense that a flat pattern has been extruded, which limits the complexity with respect to construction.

A method, likely to become integrated part of building industry in the near future, is robot controlled CNC milling in polystyrene for concrete casting. Technological Institute in Denmark has carried out a number of experimental projects with this technique. In 2010, the author participated in a workshop arranged within the professional network Digital Crafting. Here, experiments with robot milling were carried out, as shown in Figure 30. The robot allows milling of complex shapes in polystyrene blocks, which then can be used for concrete casting, both as components and as in situ casting. The *Unikabeton* project, shown in Figure 31, was a demonstration of the technique. On the left is an assembly of CNC-milled polystyrene blocks ready for casting. The mould is covered with a layer of epoxy in order to avoid cracks in the casting. It is possible to imagine a use

Figure 28: High rise examples. Different versions of floor area control. Left: Reduced area after 800 generations. Middle: Increased area after 800 generations. Right: A 'trunk' growth is stopped at every 200 generations.

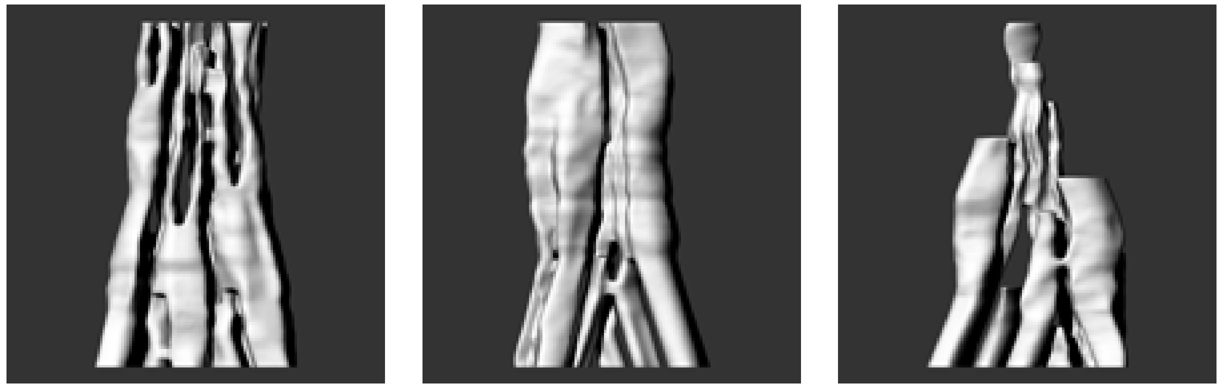


Figure 29: Baumschlager and Eberle, Nordwesthaus, 2008. Photo: Architects.

of the technique for casting the complex shapes occurring in the experiments described earlier. Probably, each layer of the geometry would be individually milled out, more or less corresponding to the generated cross-sections. The layers would be stacked and fixed before the concrete would be poured into the finished mould construction. The properties of the system would, to large degree, ensure that the fluid concrete could reach all parts of the mould. The technique is relatively sustainable, since the polystyrene is recyclable.

Another relevant technique, recently developed for building construction by Behrokh Khoshnevis, is Contour Crafting. The principle is to use an automatic system to gradually add layers of a quick-setting concrete-like material. The technique is still in an experimental state, but there occurs to be inherent constraints that correspond well with the constraints inherent in the examples shown earlier. The fact that the construction is very directional, that is, from the ground and upwards, is completely similar to the way, the cross-sections generate the geometry in the earlier mentioned experiments. Therefore, the form-generating system would ensure, that all parts of the structure would stay within the constraints of the contour crafting technique. A close analysis of the implications of using Contour Crafting has not been performed as part of the research, but it could become relevant in case of a future realisation of the developed form-generating system.

As a first step, in terms of approaching a realisation process, an example of the geometry has been produced through use of 3D printing. The sample is shown in Figure 33. Besides the quality of experiencing the shape in physical form, there are a couple of incentives for producing a physical model. In some way, the printing procedure is a qualitative test of the produced geometry. The meshes that define the shapes have to be geometrically consistent for the model to be correctly printed. In other words, errors in the form-generating process, resulting in corrupted geometry, would be discovered during the preparations for 3D printing. The procedures proved to produce consistent geometry, where in fact, some problems did occur in situations where the volumes had voids inside in the bottom or top. Problems that were solvable. Another issue that came about was how well the geometry was connected. Parts that were only loosely connected would fall off the printed model. Also the question of dimensions became present, since too small dimensions of the branches would cause them to break. These are aspects that would also become present in case of realisation on a building scale. Another interesting issue that began to appear from producing the 3D print was the question of detailed traces of the production technique. Even in the minimal size of the 3D print, less than 20 cm high, it was clearly visible how the model had been oriented inside the printing chamber. In case of realisation, the aspect, of balancing smoothness against decorative traces from production technique, would become an important part of the design solution. Particularly, because there often is a connection between the roughness of the material treatment and the speed of production, which was also is demonstrated in the *Unikabeton* project.

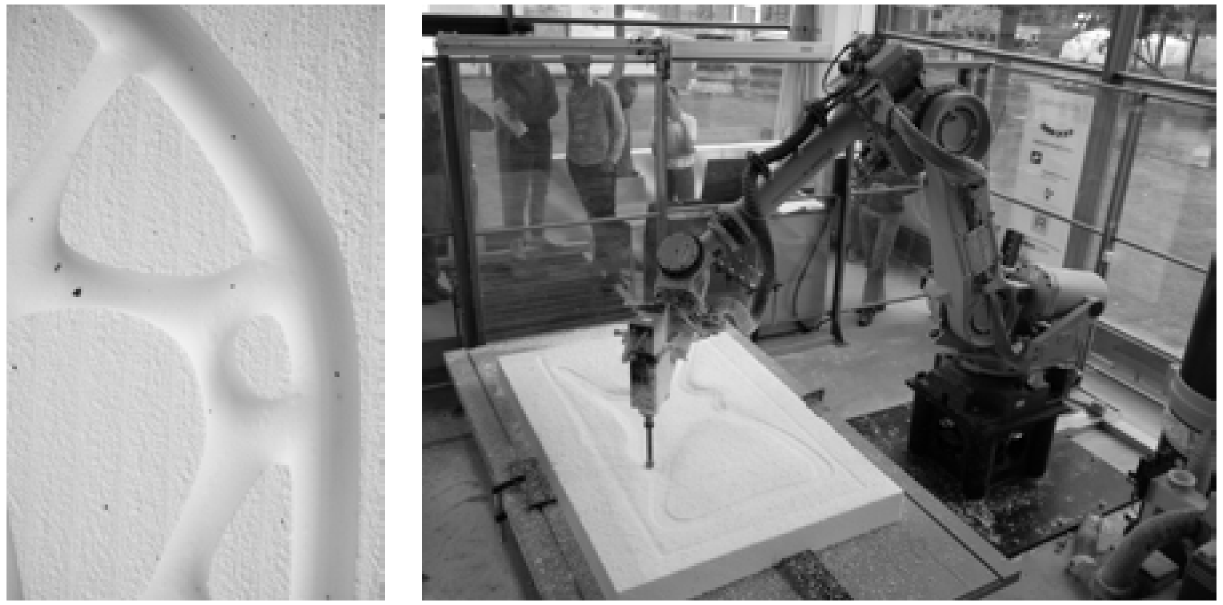


Figure 30: Technological Institute, Copenhagen, Denmark. Images from Digital Crafting workshop, 2010. Left: Polystyrene mould for concrete casting. Right: Robot milling of polystyrene.

Figure 31: Unikabeton, full scale experiment. Photos: Asbjørn Søndergaard.

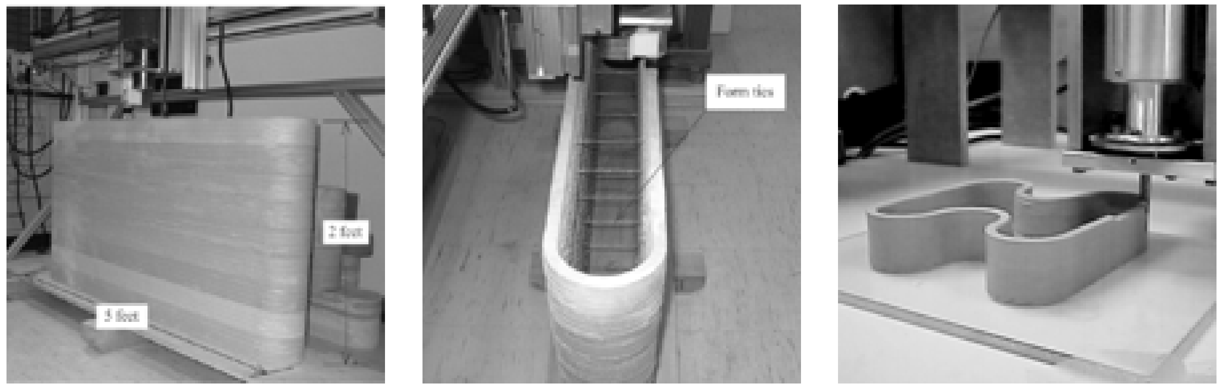


Figure 32: Behrokh Khoshnevis, experiments with Contour Crafting.

8.3.6 Observations

The method is successful in the sense that it provides means for generating geometry with no a priori topological relations, as was the primary goal for the development. The sequential procedure of generating the form entails remarkable constraints, since the result has a certain linearity, or direction. In principle, the method can be configured to generate more freely oriented shapes, but the logic of the system encourages a direction in the generation of the result. The linearity is one of the properties that also define the character of the outcome, thereby establishing a particular formal identity. The character of merging and branching trunks appear from this linear logic, and the appearance would be more generic with a method without orientation. This property also limits the range of possible uses, since only projects where this particular type of formal character is useful can benefit from the method. However, as described in the previous, the constraints that limit the use of the method can possibly be useful in terms of controlling the outcome and realisation of the project.

In terms of relevance for use in an architectural project, the precondition is then that the formal character of the design problem fits with the character that is producible with the method. If that is the case, then the method holds particular potential in terms of securing a variety of scale dependent aspects. In smaller scale, the method can help to control minimal structural capacity and limit material use through adjusting the area of the cross-sections of the structure. The

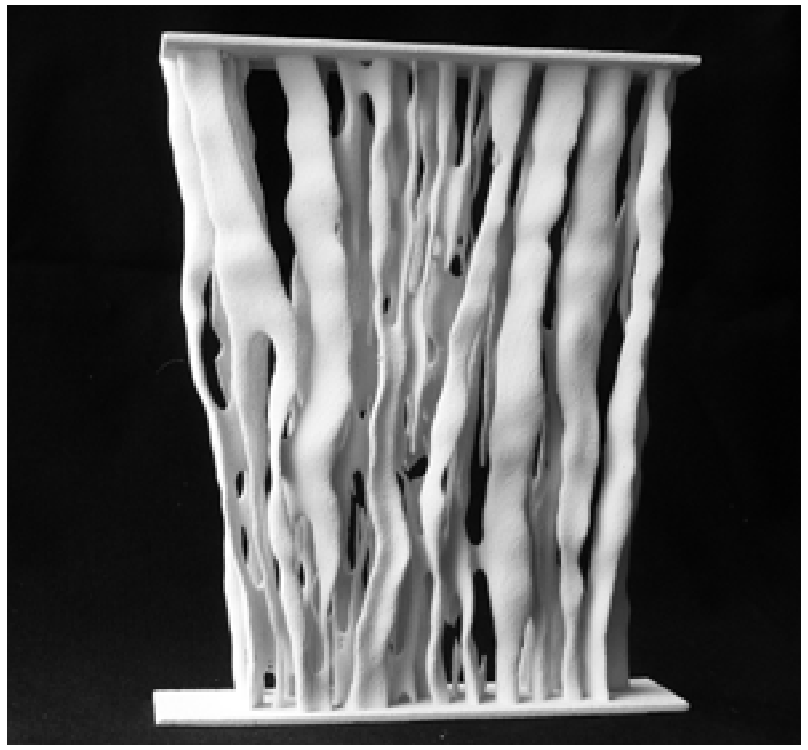


Figure 33: Sample produced with 3D printing technique.

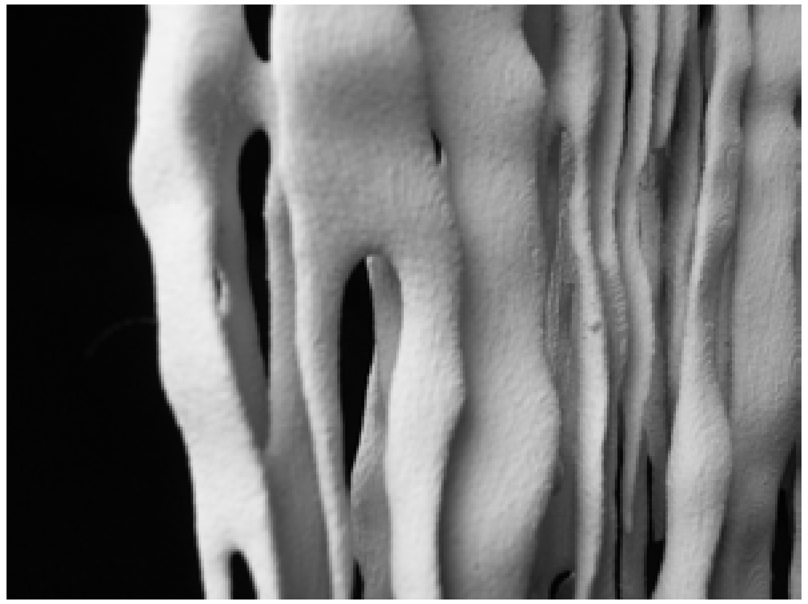
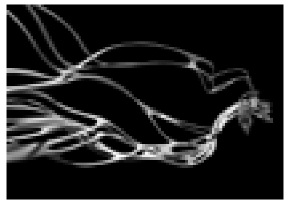


Figure 34: Sample produced with 3D printing technique. Close-up.

same mechanism can help to control floor area and surface area when used in a large building scale. In all cases, the method can secure that the formations are continuous and rooted in the base, as demonstrated in the chapter. It is possible to produce less consistent and more complex structures with the method, but this was not pursued in the development. Within the scope of the method's functionality, a range of parameters can affect the outcome. On the one hand, a series of solutions that have similar properties can be generated. On the other, the detailed behaviour of the algorithm can be adjusted, once, a particular initial setup have been chosen. However, these adjustments sometimes lead to major changes in the outcome, which again means that the time consumption increases in accordance with the specificity of the solution.

The many separate steps in the form-generating process, inclusive isosurfacing, limits the possibility of linking more specific information, for instance with respect to manufacturing of building components. Because the geometry is 'automatically' produced from a predefined algorithm, the exact positioning of, and relations between, the individual nodes in the mesh structure cannot be negotiated directly. If the method should be used for a large building project, it could be considered to link the algorithm that generates the sections with a component-oriented script instead of using the isosurfacing algorithm. Thereby, establishing an information flow where all parts can be addressed as part of the design process.

	Intent and conditions	Organisational logic	Realisation
Properties	Geometry from complex formations Controlling the behaviour Constraints Self-organisation Surface self-generation	Agent based logic Behavioural rules for flocking algorithms Internal negotiations Constraining logic Spatial self-organisation Surface generating rules	Attempts to implement in architectural design Generic logic with potential for physical realisation.
Observations	Dynamic appearance Strong flocking character Self-organisational potential Specific design problem	Self-organised topology Advanced rules Complex formations Three-dimensionality Vector-based calculations Fine tuning of parameters	Complexity and consistency Potential of integrated digital flow.

8.4 Agent-based formations

8.4.1 Experiments with agent-based systems

This chapter is structured differently from the other chapters, concerning method descriptions. With respect to agent-based systems, procedures have not been investigated through a single enclosed experiment, but rather through a series of developments that revolve around the same core principle. Therefore, the chapter reflects the process of developing different methods, and the implications that emerge. Another difference is that the topic of realisation was generally removed from the development of the methods. An exception to this is an attempt to incorporate some methods in an actual project, which is discussed later in the paper. Manufacturing techniques are similarly omitted. The emphasis is on the internal logic of the system, the possibility of using constraints, its relation to architecture and its potential as a generic tool for working with self-organisation in architectural design. As a consequence of the process-oriented structure of the chapter, most of the observations and discussions are embedded in the explanation of the methods.

8.4.2 Agent-based flocking algorithm

The following methods and experiments are based on algorithmic principles for simulating swarm behaviour as it appears in flocks of birds. In 1987, Craig Reynolds described a method for simulating bird flocks in the paper, 'Flocks, Herds, and Schools: A Distributed Behavioral Model.'[1] The method was based on processes for simulating particle systems, which were well known at the time. Reynold's strategy was to define each particle as representing a bird, and then equipping each bird with a set of basic behavioural rules, mainly for guiding the bird's movement in relation to other birds in the flock. Reynolds demonstrated how the unfolding of these local negotiations between the virtual birds, or 'boids' as he named them, was sufficient to generate a flocking-like behaviour. There was no need for explicit definition of each bird's movement path, or for an overall steering of the flock. Self-organisation as a guiding principle in swarms is recognised within science and occurs in many different forms and species. A number of simulation algorithms have been developed in order to describe these natural phenomena. However, because it was developed for a practical goal rather than scientific

1 Craig W. Reynolds. 'Flocks, Herds, and Schools: A Distributed Behavioral Model', *Computer Graphics, vol. 21, no.4,* July 1987, pages 25-34.

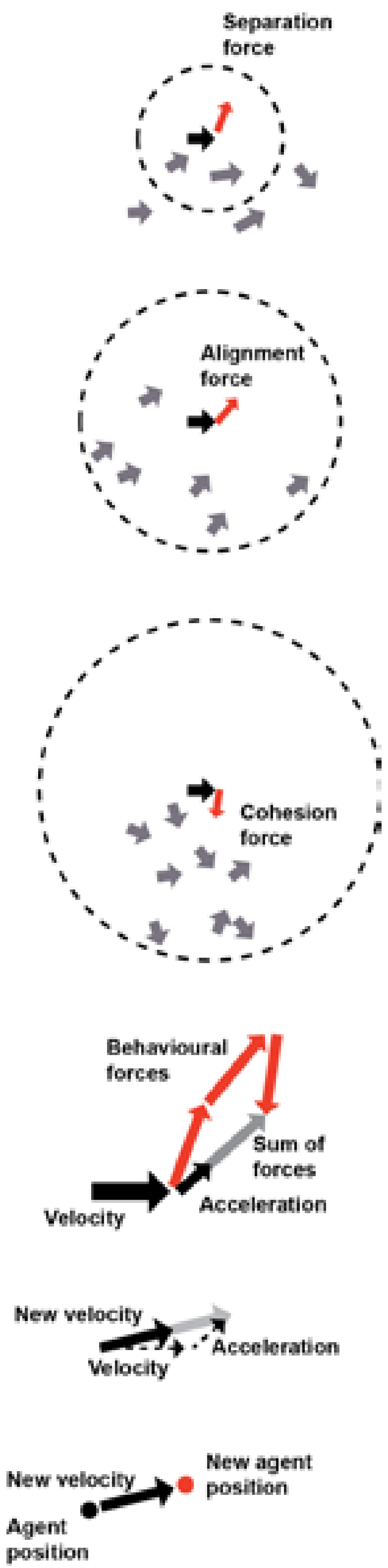


Figure 1. The behaviours are related to a differentiated range of vision. Each behaviour results in a force, affecting the movement of the agent.

proof, Reynolds' method serves well as a generic system for manipulating and developing an agent-based system for purposes other than the simulation of natural phenomena. The manoeuvres of Reynold's boids are primarily based on three essential behavioural rules: *separation*, *coherence* and *alignment*. Much other behaviour such as avoiding predators and collisions with the environment can be encoded, depending on the animation. The three basic rules are shown in Figure 1. All three behaviours are coupled with a field of vision that refers to the distance to other boids. Only boids within *range of vision* can affect the boids' movement, and the distance if differentiated, depending on the rule type. For instance, it is crucial that the range of vision for coherence is larger than for separation, since otherwise, the boids will not stay together. The system can be described as a state machine, similar to the principles developed by John Holland.[2] The movements are described as vectors, as are the changes in speed and direction. The latter is calculated for each boid in each state. In other words, for every state or generation, the position, speed and direction is known for all individual boids in the system, typically representing a flock. Each boid is analysed in terms of relation to neighbours according to the behavioural rules that are most often identical for all boids. It should be mentioned that it is possible to define different 'species' with different rules, and even vary the rule-set between the individual boids. Generally, if the goal is to achieve a flocking behaviour, this type of variation is not implemented, except from definition of special roles, such as predators attacking the flock. Figure 2 illustrates the sequence of updating a single agent, or boid. Each behavioural rule results in a calculated force, affecting the acceleration of the boids' movement. The forces, represented by vectors, are added and the resultant vector is added to the current velocity vector. In other words, the speed and direction of the boid is adjusted according to the influences of its neighbouring boids. In each generation, or time step, all of the boids are updated, gradually harmonising the relations between them. The rule of *separation* defines a force directed away from neighbour boids within the range of vision. The rule of *coherence* defines a force leading towards the centre of the flock. Still, it is only boids within the range of vision that are taken into account. The rule of *alignment* defines a force representing the average speed and direction of the neighbour boids. The three rules ensure that the boids stay together, do not collide, and move in more or less the same direction. Because the forces are indirectly conflicting, the dynamic movement of the flock is generated. The rules can be said to negotiate and gradually shift their influence on

2 John Holland, Emergence: From Chaos to Order. Oxford University Press, Oxford, 1998, page 33.

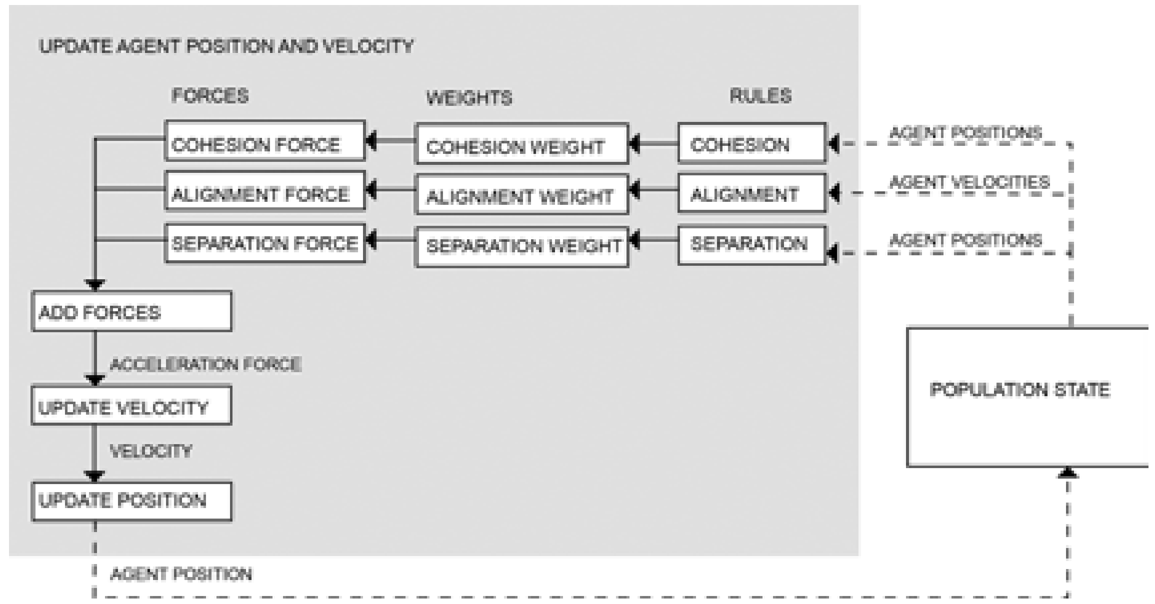


Figure 2. Diagram of the basic logic of the iterative updating of each agent from one state to another.

the individual boid, as the flock is changing shape. The main focus of the computation task is calculating the boids' acceleration as a vector that describes the change in direction. The acceleration is gained from summarising the behavioural forces that can affect the movement. Typically, calculation of a behavioural force entails some degree of analysis of the whole population of agents. The initial analysis of a population member, that is, another boid, is dependent on whether the boid is within the range of vision. If the boid is too far away, the boid's data are not included in calculating the behavioural force. Hildenbrandta, Carereb and Hemelrijka have discovered that in nature, it seems to be a limited number of close-by individuals that affect behaviour, rather than the exact distance[3]. In other words, it is a topological principle that guides the birds, compared to a metric principle. Since the goal here was not to construct scientifically correct simulations of a natural phenomenon, but rather to establish new form generating techniques, this detail has not been further investigated. However, the problem of topological versus metric relations is part of the method *Self-organising Surface*, described in section 8.4.9.

The next step is to extract the data, relevant to the particular behaviour. For *alignment*, the velocities of nearby boids are averaged, so the current boid can synchronise its own speed and direction with the neighbours, as shown in Figure 1. In the case of *cohesion*, the position of boids within the range of vision are averaged in order to find a centre point for the flocking, and construct a force vector pointing in that direction. The *separation* force is calculated from an

3 H. Hildenbrandta, C. Carereb,c and C.K. Hemelrijka, *Behavioral Ecology, vol. 21, no.6, 2010*: 1349-1359.

average of nearby positions, similar to the cohesion force. Here, the force is directed away from the neighbours. Importantly, the range of vision is different between cohesion and separation, which prevents them from being cancelled out. Each boids' behaviour is weighted in order to balance the system behaviour. Weighting the behaviours is a crucial part of developing and adjusting the system. In the case of flocking simulation, separation is normally weighted much higher than coherence, because the range of vision for separation is smaller. This means that the boids have to change direction within a smaller amount of states. To continue with the diagram in Figure 2, the behavioural forces are summed up and limited to a maximum acceleration magnitude. The boids already have a velocity vector, representing speed and direction. The change in velocity is calculated by adding the acceleration to the velocity. This leads to a gradual change of direction. Finally, the velocity vector is added to the boid's position, thereby propelling the boid forward. The system develops iteratively, and for each state, all boids are updated. In this way, the system manages to replicate a system of agents that gradually changes position in a way that is surprisingly similar to a flock of birds in nature. Therefore, the algorithm is very useful for movie animations, because the modellers avoid having to script the trajectory for every bird in the scene and bird-like behaviour is automatically ensured. One of the first films to make use of the method was Tim Burton's Batman Returns (1992), where bat swarms were simulated.

By starting with the algorithm for simulating bird flocks, a sequence of experiments have been performed in order to investigate the potential of using agent-based systems for architectural purposes. Mainly, the experiments concern new types of spatial organisation, and begin to suggest ways of establishing negotiations between both internal and external parameters during the generative process. Bird simulation was chosen because it is a relatively simple example. However, as the examples demonstrate, it takes some effort to encompass the strong expression of the bird flock. The methods described in the following section were developed in order to discover usages of agent-based systems for generating spatial formations and geometry. Subsequently, new ways of generating architectural structures in the broadest sense have emerged.

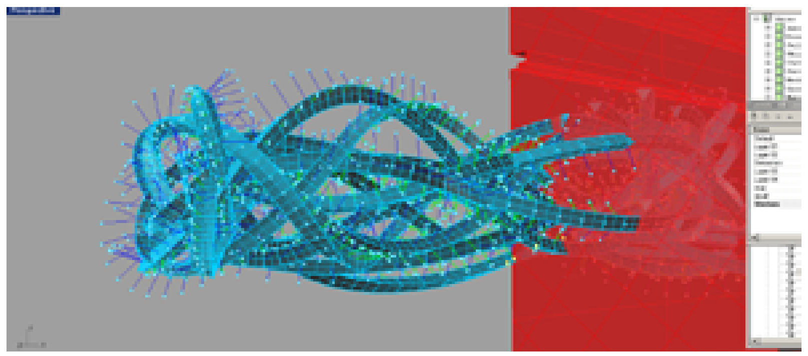


Figure 3. An example of generation of volumetric geometry from mapping of boid movements. VB script in Rhinocerous 3D modeling environment. The thin blue lines represent the rotational orientation of the boid.

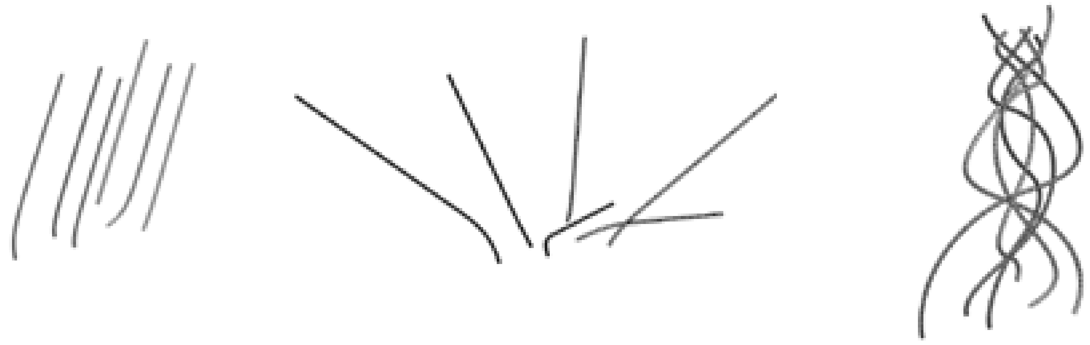
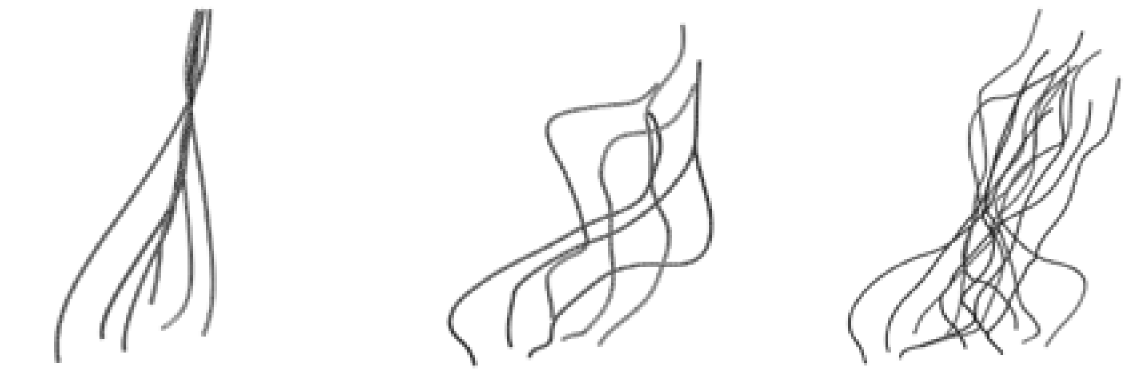


Figure 4. Mapping of distinct agent behaviours. The movements were initiated at the bottom of the field. Left: Alignment behaviour. Middle: Separation. Right: Cohesion.

8.4.3 Geometry and basic flocking behaviour

The first step was to find a way to capture the geometry of the agents' movement. One way to do this was to map the trajectory of each agent and construct a volumetric geometry from that data. In many of the examples shown, box geometry has been added to each trajectory in each state of the system. From the velocity vector and the rotational orientation, a seamless, continuous, and facetted mesh could be constructed. The screen capture shown in Figure 3, demonstrates the mechanism of mapping the agent's movement and constructing volumetric geometry from it. The blue lines pointing out from the 'tubes' represent the rotational orientation of the boids. Figure 4 shows a sequence of images demonstrating the three basic behaviours in different combinations. The system is portrayed in two dimensions in order to demonstrate the behaviours more clearly. The movements were initiated at the bottom of the field. The left image shows a combination of *cohesion* and *alignment*. These behaviours supplement each other without conflicting, and lead to the boids forming a tight cluster, moving with entirely synchronised speed and direction. The middle image shows the resulting behaviour when *cohesion* and *separation* forces are activated. The boids attempt to move to the centre of the flock, but are repelled when they get

Figure 5. Combination of behaviours. Left: Cohesion and alignment. Middle: Cohesion and separation. Right: Cohesion, alignment and separation.



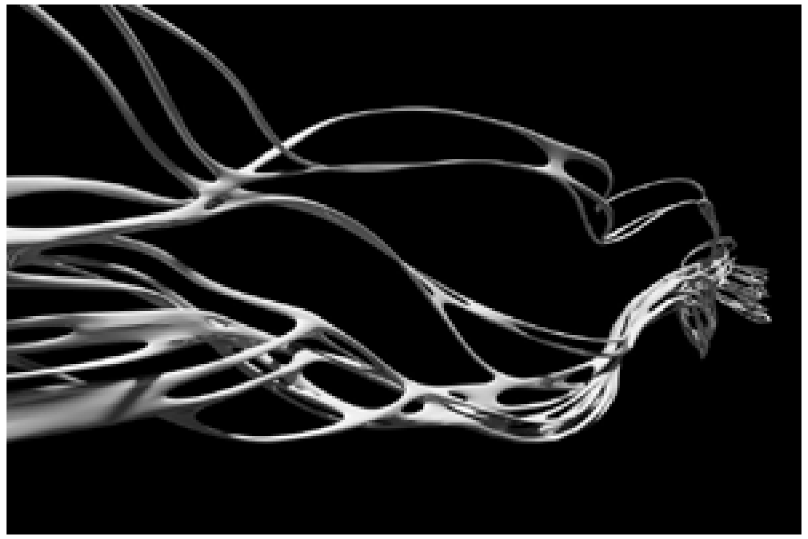


Figure 6. Mapping of flocking behaviour. Renderings, Figure 6-8: Morten Bülow

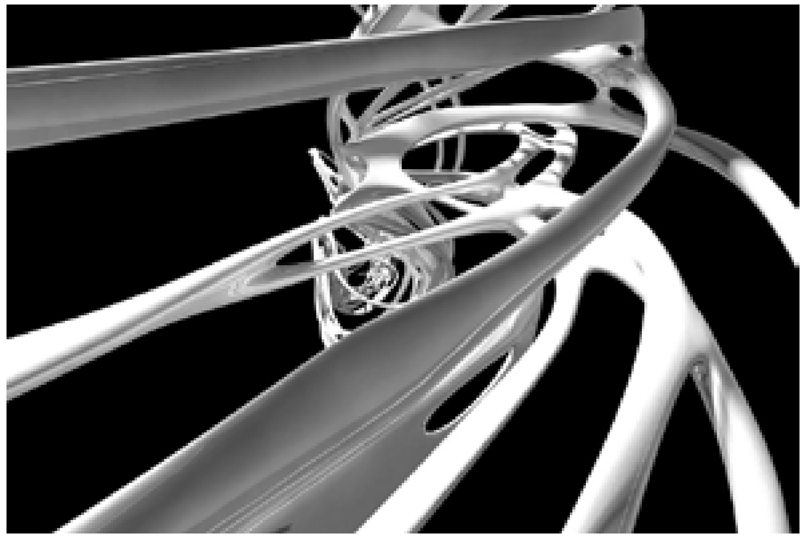


Figure 7. The geometry was subjected to a smoothing algorithm for emphasising the character of a seamless flow.

Figure 8. An experiment was produced for a specific spatial context with respect to development of contextual behaviours. Tectonic or programmatic aspects were not taken into account.

too close to their neighbours. Because the two behaviours are conflicting, the result is an oscillating effect, which often occurs in nonlinear systems[4] and also characterises natural flocks. The right image shows all three behaviours as active. As seen in the example, the flock begins to form a regular and dynamically formed group at the top of the field. The flock will continue, more or less, in the same direction.

8.4.4 Complex formations

To develop an understanding of the potential of the basic algorithm, a series of experiments were produced. The first examples, shown in Figures 6–8, were initiated at a workshop with Roland Snooks at the research centre, CITA, of The Royal Danish Academy of Fine Arts in 2009. The experiments were developed in collaboration with Morten Bülow. Behaviours, complementary to the three basic rules, were implemented in these early experiments. The focus here was mainly on the response to a representation of a physical environment. The enhanced boid could avoid collision with surfaces and move towards predefined attractors in the form of points, lines or surfaces. An algorithm that could merge the 'tubes' of the trajectories at places where they are close to each other, was implemented. The procedure turned out to be nontrivial. In order to arrive at the smooth finish, irregularities were manually carried out in the 3D modelling environment. This was not considered to be a methodological problem, but rather a practical problem, since the process turned out to be time consuming. In Figure 8, where the movements are mapped in a model of The National Gallery of Denmark, the boids were 'released' in one end of a long space, and aimed at a wall in the opposite end. The trajectories were not straight due to obstacles and because of the interactions between the boids. Because of the nonlinearity of the system, it was immensely difficult to balance the behaviours in order to get a useful result. Most of the time, one or more of the behaviours were weighted too high, resulting in some of the boids behaving irrationally. As Roland Snooks remarked during the workshop, 'With agent-based systems, you can guide the behaviour of the agents, but you cannot control them completely'. Still, the characteristic expression of flying birds was largely maintained throughout these adjustments. The resulting geometries were subsequently smoothed through the use of a predefined tool in Maya software, in order to express the idea of a seamless flow.

4 John Holland, *Hidden order: how adaption builds complexity*, Helix Books, New York, 1995, page 18..

8.4.5 Constraining the agents

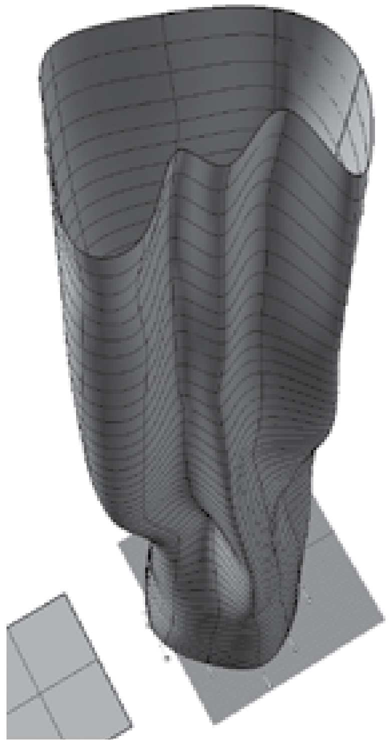
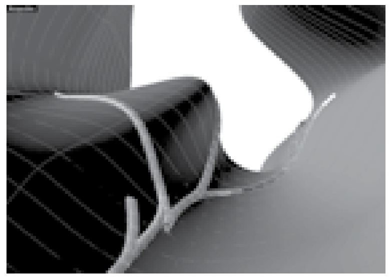


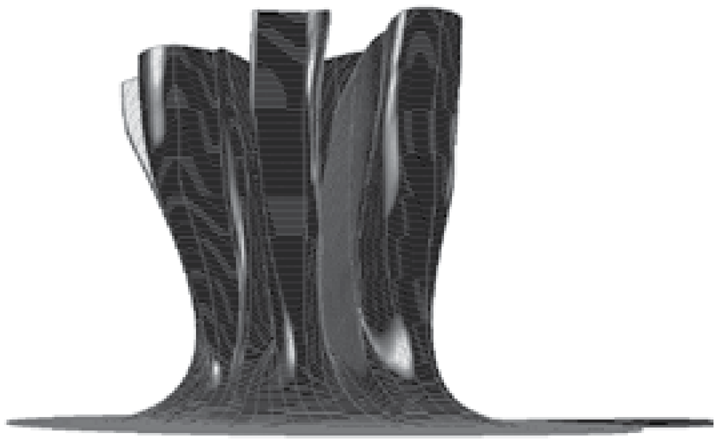
Figure 9. Experiment with a flocking algorithm. The movement of the agents was mapped as control points for constructing a surface.

The next series of experiments was focused on transforming the agent-based system from almost simulating birds to become more versatile with respect to an architectural design process. A method, were the agents create control points for generation of a surface, was developed. The main principle of mapping the agent's trajectories was maintained, but instead of generating tubes around the trajectories, each position of the agent, throughout the movement, was saved. During the generative process, a facetted mesh was constructed from the control points. Because the faces were instantaneously generated, as the agents were moving, it was possible to encode a behaviour that repelled the agents from the surface, while it was being constructed. This ensured construction of a surface that, generally, did not collide with itself. Still, as in line with the previous statement about complete control, the surface consistency depended on balancing all the behavioural parameters. After generating the facetted surface, a new curved surface was generated by drawing NURBS curves from the control points, and connecting them through use of the loft function in Rhinoceros. This is a predefined algorithm that enables construction of a curved surface from a series of curves. The topology of the surface was not further challenged at this point, so the practical use of the method is limited to relatively simple volumes.

The next series of experiments focused precisely on the question of gaining more control over the agents, with respect to arriving at methods more directly useful in architecture. This meant introducing more constraints in the generative process. One useful control mechanism was to constrain the movement of the agents to a surface. Figure 10 shows how the surface, generated in Figure 9 has become the state space for the agent's movements. The surface constraint could be achieved in different ways. One way could be to define the agent's movements as two-dimensional, and

Figure 10. Below: Example where the agents a constrained to follow a surface. A nearest point function is used, rather than UV-coordinates. Right: Example of a surface constructed with a flocking algorithm. Each constructed layer is visible.





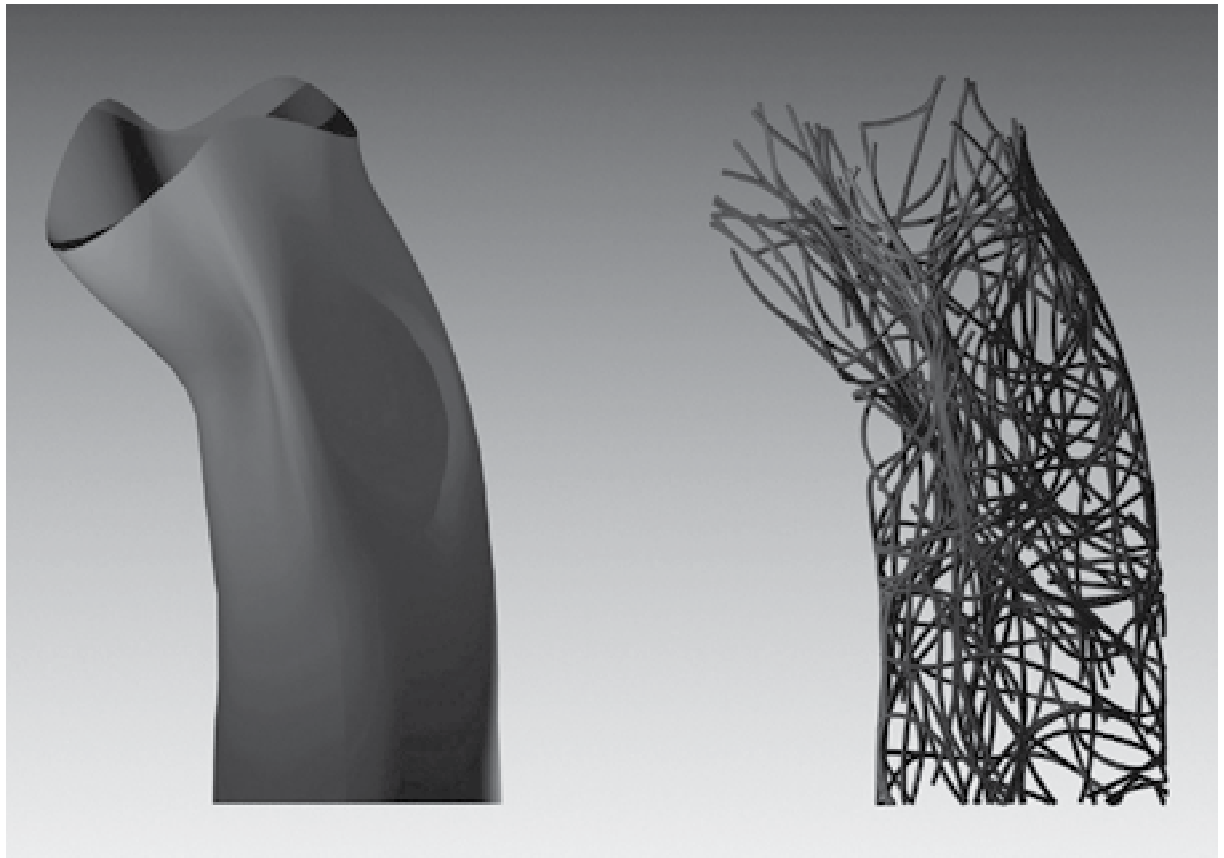
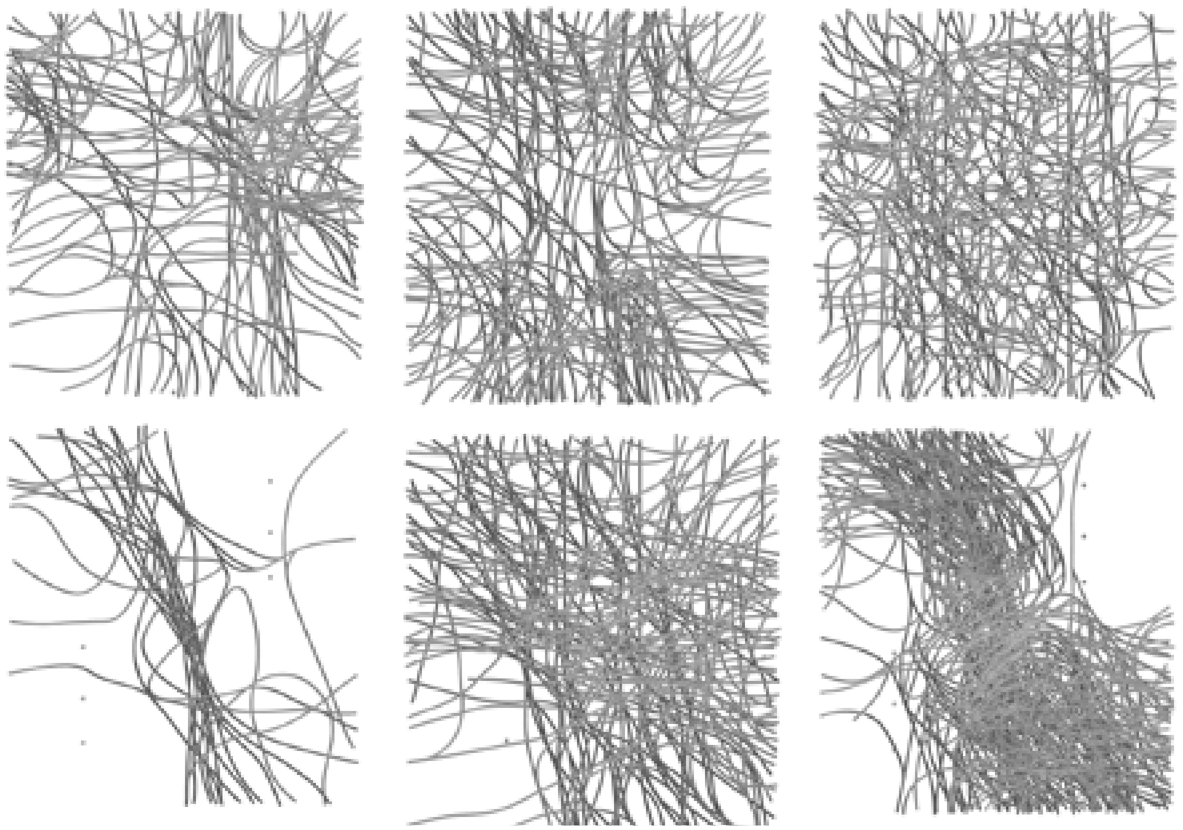


Figure 11. Left: Surface generated with flocking algorithm. Right: The same surface is used for constraining another flocking algorithm, where the agents are mapped as a form of tubes, creating a type of felted structure.

let the vectors correspond to the UV coordinates of the surface. This would be equivalent to the surface method that is described in Chapter 11.3. Here, an approach was chosen, where the agents move in 3D space. For every movement, the nearest point on the surface is detected, and the position is corrected, so it sits on the surface. An advantage of this is that the scale of the movements is homogenous over the whole surface, independent on irregularities in the distance between UV coordinates, which often is the case with curved NURBS surfaces. Also identification of the edges, and continuing across them can be a challenge. On the other hand, since the agent does not automatically move on the surface, but rather adjusts to it, some irregularities from the shortening of the velocity vectors can occur. With the shown example, where the surface describes a smooth curvature, this did not appear to be problematic. As figure 11 shows, it was possible to construct a type of weaved or felted pattern, reflecting the shape of the original surface. Merging of the generated meshes into larger meshes was problematic, due to limited performance of the 3D modelling software. This was mainly a representational problem, but could also be a barrier for using the method in an architectural design process, unless the 'tubes' are conceived as individual parts of the construction. It is possible to imagine that a structural capacity could emerge from the system because of its redundancy, similar to the stability of a basket. This was not further investigated as part of the research.

The observant reader has perhaps noticed in Figure 10 another functionality that was developed as part of the method for mapping the agents. This is a method for letting the agents 'branch', in reality a mechanism for generating new agents from the position of existing agents during the generative process. As hinted, the effect was a type of branching pattern, which again resulted in a more complex and interweaved structure. Yet, this new capability also had to be carefully balanced, since, the sequence of branching often increased exponentially, leading to both extremely high density, and slowing down the performance dramatically. A noticeable side effect from the branching was that the system became less determined by the positioning of the start positions and velocities of the agents. Only a few agents had to be placed, because after a few generations, or states, the agents would 'divide' and add new agents to the scene. For studying the possibilities of directing this type of pattern more precisely, the examples in Figure 12 were produced. The upper row shows different results from running the system without further control. The main difference is the number of generations calculated before stopping the process. The system runs randomly, but it is possible to initiate the random generator in a way where a result can be completely reproduced. This is an important aspect with respect to use of the method for a tectonic solution. For the experiments shown in the lower row of Figure 12, a behaviour that seeks to avoid

Figure 12. Top: Random felted structures of agents describing two-dimensional movements. Below: By adding attractor points, the density of the pattern was varied.



attractors in the form of points, placed in the field, was implemented. The pattern clearly shows the areas as almost empty where the attractors are placed. In these examples, the attractor-dominated areas are not completely avoided, due the issue of control, already mentioned. It would be possible to constrain the agents to completely avoid the attractors, either by simply stopping the trajectory if an agent gets within a certain distance, or by increasing the weight of the attractor-related behaviour.

To begin to approach a functionality that points more directly towards use for architectural design, some experiments with large degree of constraint were carried out. Figure 13 shows examples of two types of control. In both cases, the movement was constrained to two dimensions. Figure 13, left shows agents equally distributed in the top, moving downwards. Their weight for separation behaviour was sufficient to ensure that they did not touch or cross over each other's trajectories. The agent's movement behaviour was still based on the flocking rules, and resulted in a somewhat varied pattern. Figure 13, right shows an even more constrained example. The movement is still vertical, from top to bottom, but in this case, the movement is completely linear. The variation was achieved from a mechanism implemented in the function that generated the volumetric geometry. Each agent is constructing a profile as it moves forward, and this profile becomes larger in one dimension, while smaller in the orthogonal dimension, maintaining a constant cross-sectional area. The thickness is directed by the agent's closeness to a specified attractor point. Both these exercises served to illustrate how the method can be used for both complex formations and more controlled geometries.

8.4.6 Competition for Miami Civic Center

In relation to a competition concerning Miami Civic Center, some attempts to implement the principles in an actual design solution were made. Except generation of a guiding pattern, based on the algorithm for *Self-organising Bezier Curves*, also described in the

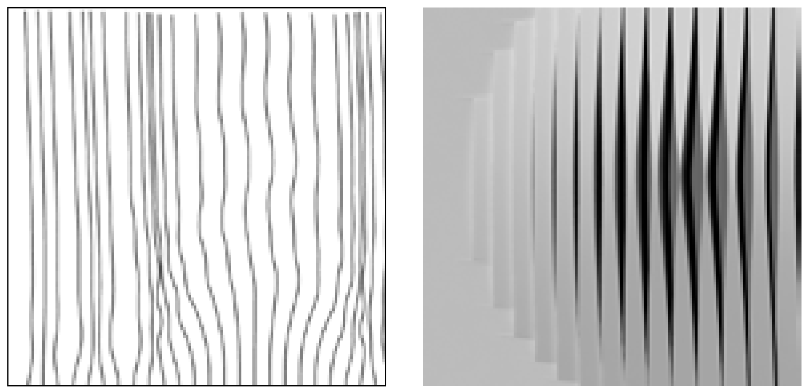


Figure 13. Left: The agents are constrained to avoid crossing trajectories. Some of the random dynamic is still visible in the pattern. Right: The agents can be completely controlled, and used for creating minimal variations. Here, the profile of the vertical elements change gradually from large width to large depth, depending on closeness to an attractor point.

thesis, none of the agent-based methods became part of the final proposal. The images shown here are by-products from the design process, and therefore only very roughly presented. Initially, the idea was to construct an agent-based system, capable of merging layers of different types of potential movement on the site into type of map, which again could be used for making decisions on how the area should be organised, see Figure 14. The plan was to construct a system of open-ended movements of agents based on generic flocking behaviour. Places of certain interest were defined as attractors that affected the agent's movements, leading to the agents constructing a map of traces that to large degree represented an image of flow between the main areas. The system was difficult to control, and was not developed far enough to reveal any surprising patterns or organisational solutions. Still, it did serve as a means for discussing priorities concerning distribution of the program on the site. Based directly on the agent mapping algorithm, explained earlier, a tool was developed for generating light solar protection 'umbrellas', as shown in Figure 15. The goal was not to arrive at perfectly ordered geometry, but rather to let the dynamic expression of the flocking algorithm come through and become part of the design solution. Explained shortly, the agents grow from the ground, upwards. When they reach a certain height, the separation force is increased, and the start to connect to each other in the form of a lath-like structure. The tectonics was not further developed, but the initial idea was a type of steel construction with a covering of wood. Another implementation was a method for generating graduated facades. While, the tectonics also here a unspecified, the goal was to use the constrained agent-based system to help

Figure 14. Miami Civic Centre competition. A map of dynamic relations between major attractors on the site was constructed from a flocking algorithm.

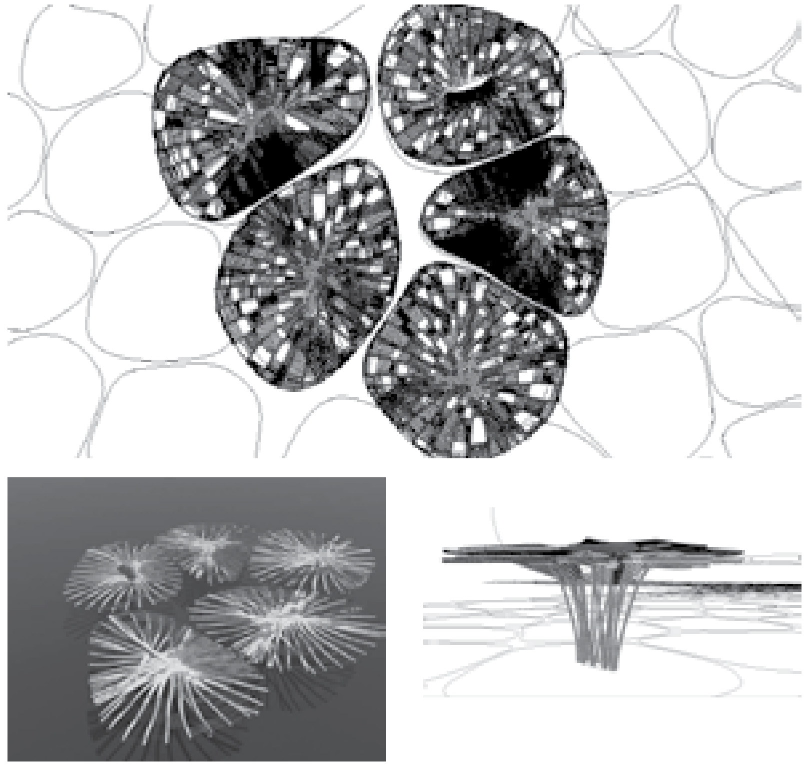


Figure 15. Miami Civic Centre competition. A flocking algorithm was configured to generate a special type of solar protection 'mushrooms' during development of proposal.

to create variation in the architectural environment, both externally and internally. As mentioned, most of the algorithmic material was left during the development process. One of the reasons is definitely that it was impossible to adjust the systems fast enough to fulfil the necessity to test out a range of solutions in a relatively short time in this type of projects. A lesson could be that the methods are best exploited when they can be linked directly with a specific design problem, rather than for displaying a range of possibilities.

8.4.7 Emergent organisation: molecular field

These exercises were done in order to investigate how different types of emergence could be established through use of agent-based systems. The previous examples were mostly focused on mapping the movement of the agents, mainly flocking algorithms. The next examples have a more generic and basic character, because they were developed in order to create systems that display different

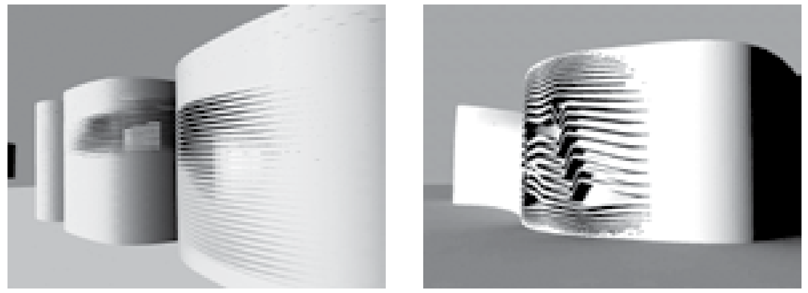


Figure 16. Miami Civic Centre competition. Some tests for implementing constrained agent based facade patterns were made.

kinds of emergent order. Generally, in the examples, the agents interact though thousands of negotiations and arrive at some type of equilibrium state. The geometric outcome is relatively simple, and in terms of tectonic solutions, the methods are unresolved. However, because the are truly self-organising, they display much more potential for future development, compared to the mapping methods, described above. The basic method is essentially a simple version of the flocking algorithm, since only behaviours for *separation* and *cohesion* is part of the fundamental logic. In this sense, it reminds perhaps of a system for simulating molecules, more than a system for showing agent behaviour, since generally, agents are understood as being able to 'act individually'. In terms of computational implementation, the border between 'free will' and being subject to forces, however, is not so precise. Also, more complex behaviours were added, as explained in the following.

First it was discovered that the mechanisms of *separation* and *coherence* was sufficient to make the agents self-organise into a completely regular grid, similar to stacking of invisible spheres. This is probably not surprising for a physicist, used to study a world of forces, but architects generally have a completely different way of creating order, namely, through explicit definition of geometry. Figure 17 shows a chaotic field of agents, and Figure 18 shows the ordered field of self-organised agents. The tests were performed in a way where a ground plane and a gravitational force were defined in order to stabilise the system. When the agents were 'floating around' they did not arrive at an equilibrium state, or rather, the equilibrium was in constant movement. Also, the showed system was not entirely stable, it continued to 'vibrate', also after self-organising, but the movements could be stopped by reducing the interacting forces. The regularity of the system was analysed by use of a Voronoi algorithm. The result is shown in Figure 19. Strongest regularity was seen in the middle of the field. The Voronoi algorithm created a cell-packing, consisting of two types of equally sized dodecahedrons, polyhedrons with 12 sides, which can be seen in Figure 20. One polyhedron consists of 12 equally formed rhombi, a so-called rhombic dodecahedron. The other type of dodecahedron consists of 6 rhombi and 6 equally sized trapezoids. Both types can be used for complete space filling, but where the first type reflects cubic sphere packing, the latter reflects hexagonal cell packing. [5] The field of agents seems to be organised in a hexagonal pattern, and the two types of polyhedra were organised in layers, where one of the layers closest to the 'ground' consisted of rhombic dodecahedrons, and two layers above was of the other irregular type of dodecahedron. The geometry was not further analysed, because the Voronoi analysis mainly served to document the regularity of the self-organisational system. An additional layer of complexity was added to the agent-based system to test if it was possible to expand the self-organisational behaviour to groups of agents. This proofed to be relatively simple, just be

Figure 17. Particle field. The particles are randomly distributed in a three-dimensional field.

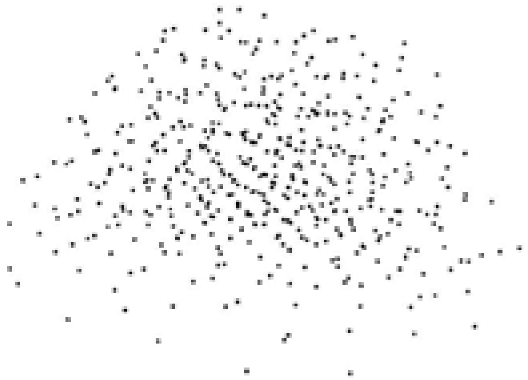
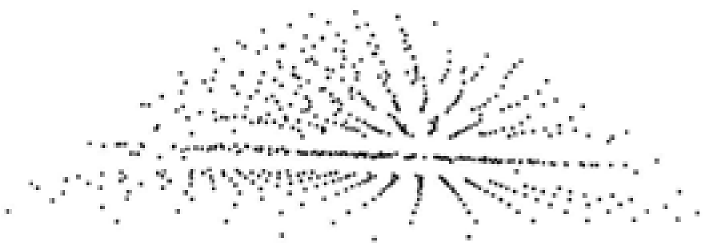


Figure 18. Particle field. With separation and cohesion rules, the particle can self-organise into a completely structured field.



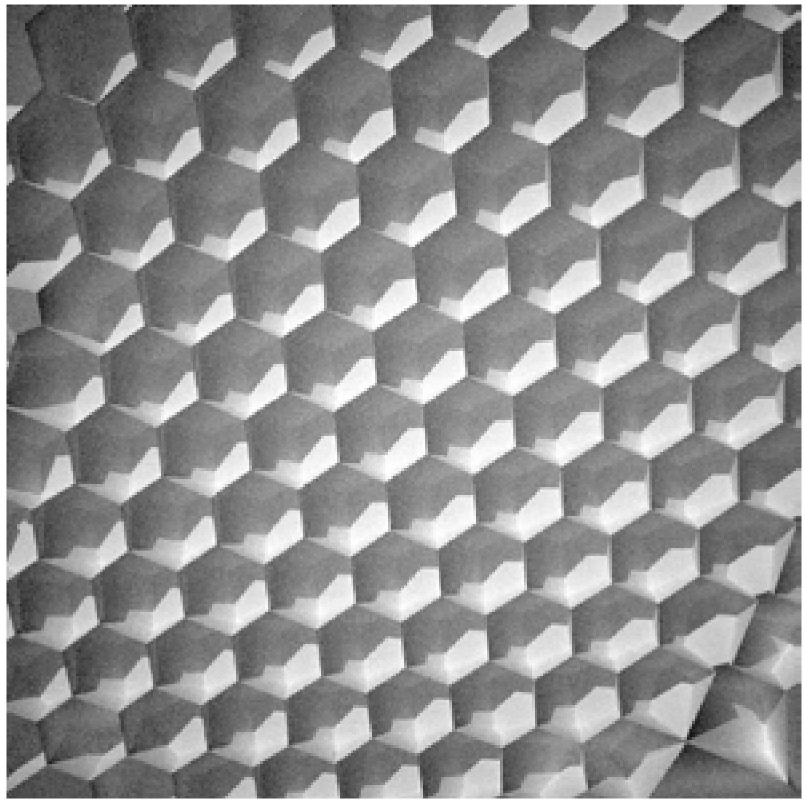


Figure 19. Particle field. When a Voronoi algorithm was applied to the particle field in Figure 19, the hexagonal organisation became clearly visible.

5 Weisstein, EW, 'Hexagonal Close Packing', *MathWorld--A Wolfram Web Resource*, viewed 1 May 2012, <*http://mathworld.wolfram.com/HexagonalClosePacking.html*>

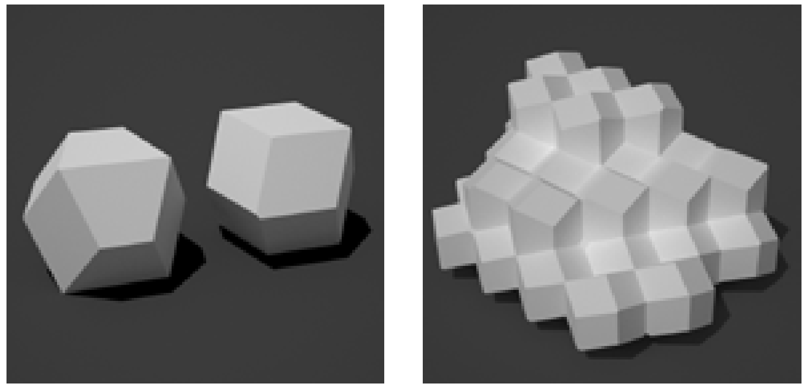


Figure 20. Left: Analysis showed that the Voronoi algorithm constructed a geometry, consisting of two basic types of dodecahedra. Right: The types were ordered in layers.

adding different 'ranges of vision' concerning group or individual behaviour, as shown in Figure 21.

The intention with the next method was to find ways of using the method from the previous example to create a system where the agents could self-organise into defining a surface. This was an attempt to address one of the fundamental challenges with respect to use of self-organisational systems for generating geometry. As Roland Snooks has formulated it: 'how does a bottom up design strategy comprehend a global condition such as topology?' [6]. It is understood that this is in case, the topological relations are not predefined. Predefined relations are, generally, the way that geometrical hierarchies, such as surface relations, are controlled in most digital generative systems. The experiences from the simple spatial self-organisation led to the belief that similar ways of making agents self-organise into defining a surface would be possible.

---

6 Cecil Balmond, 'Informal Agency' in Leach, N, and R Snooks (eds), *Swarm intelligence: architectures of multi-agent systems*, LSTPH, Liaoning, 2010, page 121.

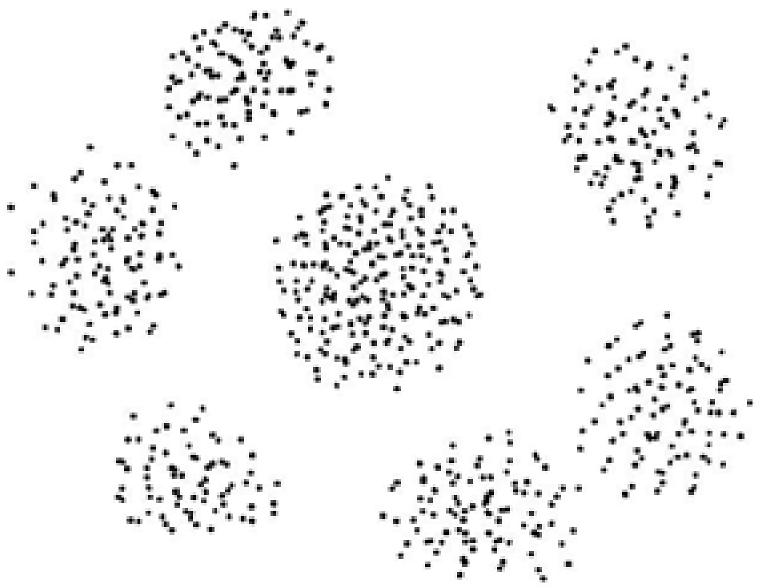


Figure 21. It was possible to distribute the particle field to smaller clusters, or molecules, equally separated with a certain distance.

8.4.8 In-direct surface generation: attractor field

The first step was to introduce attractors to guide the positioning of the agents. The agents were forced to position themselves within a certain distance from the nearest attractor point. The most important rule was the separation from the neighbouring agents. This leads to hexagonal patterning of the surface. If a regular distribution of agents, describing a virtual surface with a hexagonal grid, was established, grid lines could then be constructed between each agent and its closest neighbour. If the mesh was consistent, it could then be displayed as a closed surface, and exported to different software, such as 3D modelling environments. A similar logic has been demonstrated by Daniel Piker and his Kangaroo plug-in for Grasshopper, which again is a module for the Rhinoceros software. Piker's goal was to use a new approach to a so-called Metaballs logic[7]. Also in the experiment described here, the initial experiments were directed towards covering one or more attractors. A difference is that with Metaballs, a gradual transition from one 'sphere' to the next is part of the logic. Here, the experiment headed towards positioning the agents directly in relation to the closest attractor. Some new behavioural rules were added to the basic logic of the regular field algorithm, described before. The agents were coded to be attracted to nearest attractor point, but also to be repelled from it when in a certain distance. This resulted in all the agents positioned correctly in terms of distance from the attractor, and because of the separation behaviour that forces the agents to keep a certain distance from each other, they began to form an evenly distributed hexagonal grid. The agents could then form connections to their nearest neighbours as shown in Figure 22. A challenge was to get the agents to balance the distance between them, because a constant separation distance would normally not result in an enclosed volume. This was mainly achieved through the agents negotiating whether the separation distance should be increased or decreased. In the first examples there was a global value that held for all the agents, which meant that all the agent could affect, and were affected by, this value. An agent would seek to increase the value if it had less than 6 neighbour agents, and seek to decrease it if it had more than 6, since 6 neighbours was the ideal situation for creating the surface structure. Also, because the surface was expected to curve, the agents tried to decrease or increase the separation distance, if the curvature seemed to become too facetted, or too flat. The latter could indicate that the surface structure could be achieved with lesser control points. When

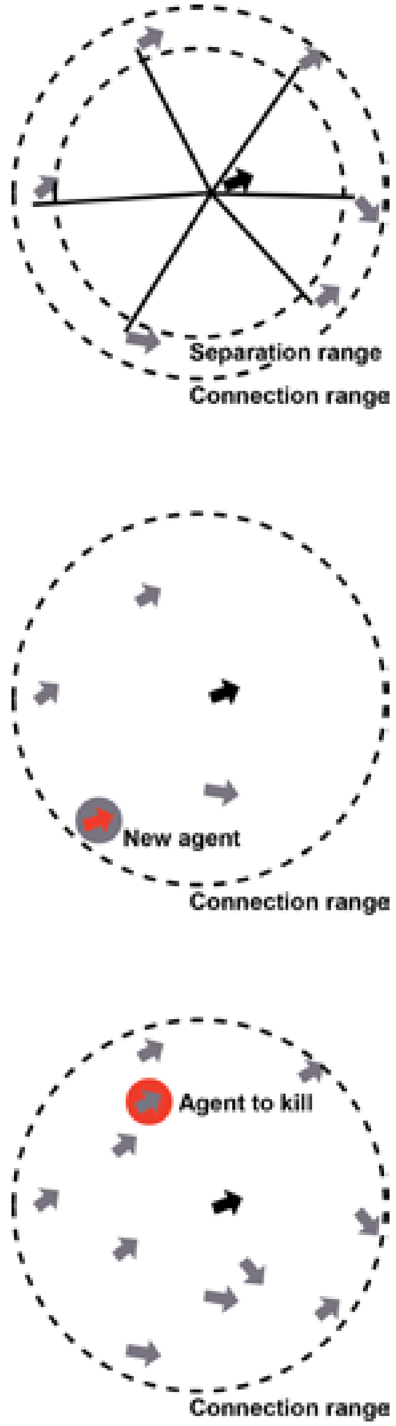


Figure 22. Relational behaviours. Top: An agent connects to its nearest neighbours. The separation range is negotiated between the agents. Below: Agents could for every 100 states destroy or add other agents, depending on the number of neighbours, ideally there would be 6.

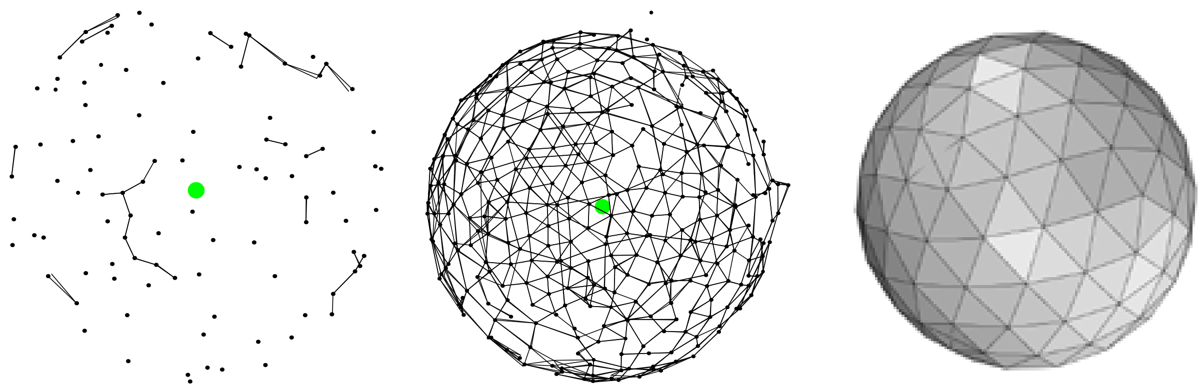
---

7 Piker, D, 'Metaballs in Kangaroo', 2011, viewed 2 May 2012, <*http://www.grasshopper3d.com/profiles/blogs/metaballs-in-kangaroo*>

the factors for these negotiations were adjusted, the system proofed to be able to find suitable values for being able to cover a number of attractor points. Figure 23 shows how a single attractor is gradually covered. First, the agents are randomly spread. Because there are only few connections, it can be concluded that the separation value is too low. Through negotiation the agents distribute evenly, and begin to cover the surface with connection lines. Finally a facetted and fully covered sphere is constructed. As it is shown, small errors occurred in the system at the stage to which it was developed. Figure 24 shows how the geometry can be further developed into a volumetric geometry with methods similar to the ones used in the concrete grid shell project described in the thesis. Here, the grid members have a rectangular cross section, rather than a triangular. Also, the members have been graduated in terms of thickness. The lower parts are thinner than the upper in the example. Note that the triangular grid has been exported as a hexagonal grid, in order to generate the Radiolaria-like appearance. An error, unless manually corrected, will also appear in the generated result. Here, the script that generates the geometry does actually recognise the situation and generates a relatively well-defined solution. The example is shown to demonstrate both the imperfection of the system, and indicate that it can be dealt with in different ways.

The next step was to try to cover a whole field of attractors. An example was established, where a field of evenly distributed attractor points was constructed from a randomly drawn surface geometry. In this case it soon proofed insufficient to just let the agents negotiate the separation distance, because the complexity of the geometry did not allow a reasonable solution with a limited amount of agents. The agents behaviour was expanded with the capability of adding and removing other agents. If an agent had too few neighbours, it could construct new agents, positioned close by. Alternatively, if an agent had too many close by neighbours, it could remove one of them. These mechanisms turned out to be efficient in growing or reducing the population of agents to a size that fitted the current task of covering a field of attractors. Also, the system

Figure 23. Generation of a sphere as a process of covering an attractor point.



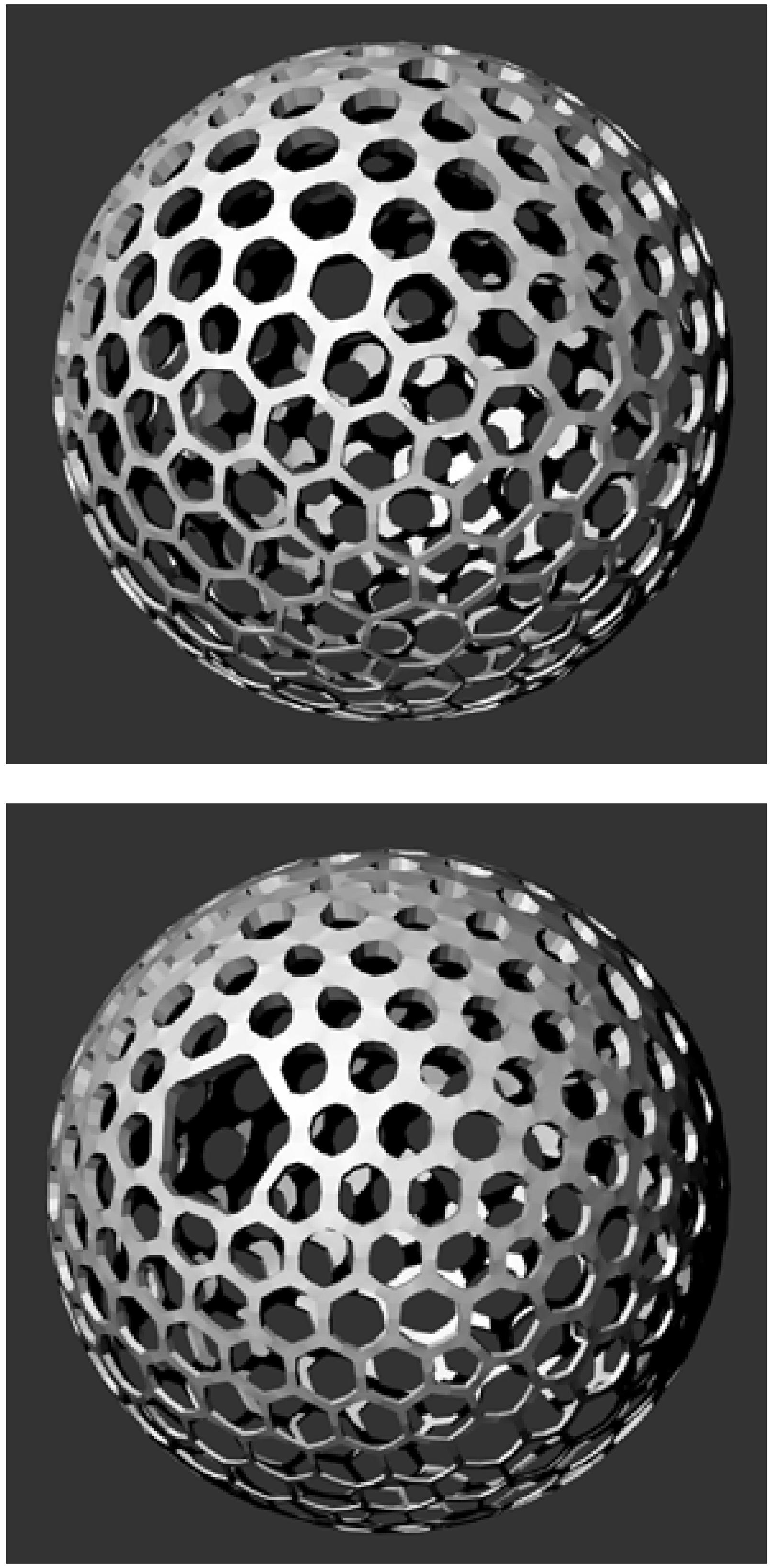
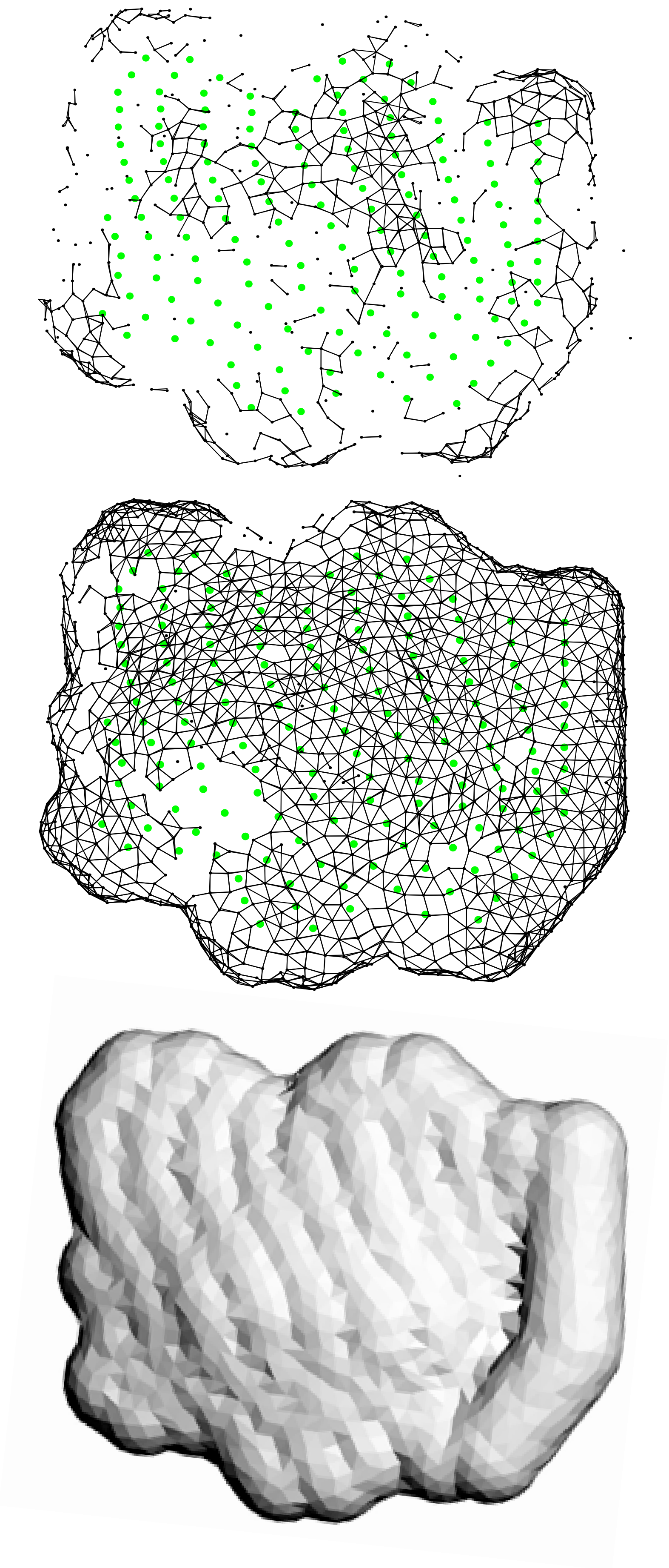


Figure 24. Top The structure constructed in Figure 25 was exported to a 3D modelling environment where a script was used for applying graduated volumetric geometry to the structure. Below: The mesh was not entirely consistent, which would show in the volumetric geometry.

Figure 25. Process of covering a field of attractors with use of the algorithm for generating a self-organised surface. The green points represent the attractor points.



was capable of handling an attractor field, where the attractors have individual field radii, so it was possible to construct surfaces with differentiated curvatures. The example in Figure 25 shows an example of how the process could form. The field of attractor points is shown as green dots. In the beginning, the agents are scattered with only few connections. Through a process of adding new agents, but also deleting agents in areas with high density, the field of attractors is gradually covered with a regular hexagonal grid. Finally, the field is fully covered with an enclosed volume.

While, the algorithm was not able to generate entirely resolved geometry, the pattern was resolved to a point where it seemed realistic that the system could be fine tuned in order to generate consistent meshes. The inconsistencies were relatively small, and could be repaired with tools that are found in 3D modelling software. Of course, if the goal is to establish an interlinked generative flow, the optimal would be to develop the algorithmic logic a step further. Probably, an approach even closer to the Metaballs, where the transitions between attractors are gradual, would help to avoid some of the problems. However, the method was mainly seen as a way of approaching some topological challenges, described in the following.

8.4.9 Self-organising surface

The next series of experiments were directed towards finding new ways of generating geometry from a self-organising system. Two typical approaches towards can be mentioned. One is to have topological relations predefined, and then let the agents change position through the form-generating process. Usually, the relations stay intact throughout the process, but in some cases, the relations can be altered within the process. The essence is that an ordered relational system exists from the beginning, and the order is maintained throughout the process. Another approach is to start without any predefined relations, a disordered field, so to say. The agents can self-organise into complex structures, but then the challenge is how to extract geometry from them, particularly a surface geometry, since the topological relations are undefined. This can be done with a secondary algorithm, such as isosurfacing. Generally, this procedure exists as a separate part of the generative process, not directly related to the movements of the agents. The positioning of the agents is not directly corresponding to vertices on the resultant geometry. This implies that these vertices cannot be directly negotiated within the process, which again can be problematic with respect to architectural realisation in the form of a specific tectonic system. It would be possible to control the isosurfacing algorithm by customising it. Anyway, the method has certain limitations, due to

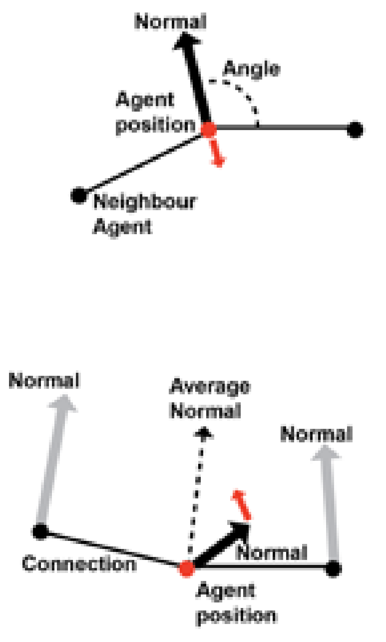


Figure 26. Behaviours related to surface normals. The current agent's position is the red dot, and the neighbours are shown with black dots. Top: The agent moves in the direction of the surface normal if the angle is bigger than 99 degrees, and vice versa if it is smaller than 81 degrees. This helps to flatten the surface. Below: The surface normals are gradually aligned.

the fact that it is based on a three-dimensional grid of cells.

Here, a different approach was tested, an approach where a cloud of agents without predefined relations self-organise to form a surface. It is an emergent system, since the formed surface represents a different and higher degree of order than exists in the initial system. Also, the rules that guide the agents do not define their roles in relation to a larger whole. Only local negotiations drive the agents towards generating a surface. In the previous example, the attractor points served to direct the agents, whereas, no guiding geometry 'helps' the agents in the method described here. Before, the virtual surface normals were derived from the agent's relation to the closest attractor point. Figure 26 shows how the surface normals are adjusted, so agents close to each other gets similar surface normals. Initially, the surface normals points in random directions, but gradually, as the system evolves, the normals self-regulate. The resultant surface is randomly oriented, since there is no directional force or defined base plane of any kind.

An additional behavioural rule was that an agent should try to move towards a position where the angle between the surface normal and the lines, connecting it with its neighbours, was a right angle. These two new behaviours were basically sufficient to let the system self-organise into a surface structure as shown in Figure 27. First, the chaotic field of agents is randomly connected. As with the earlier system, the ideal separation distance between the agents is negotiated locally between the agents. In the second image, the agents begin to form planes with different orientations. Then, a process of negotiation takes place. At some point, a primary surface has formed, as shown in Figure 27c. From Figure 27e onwards, the system seeks to resolve the imbalance from the secondary surface pointing out from the primary surface. The agent nodes in the joint between the two surfaces cannot find equilibrium, but continues to shift position, generating a dynamic movement. It can be seen in Figures 27e to 27k how the secondary surface, gradually, is pushed towards the edges, where it finally collapses and merges with the large surface. In the final images, the structure is in an equilibrium state and has stopped moving.

It was interesting to discover how different types of hierarchical negotiations appeared as part of the form-generating process. Also other types of emergent effects were detected. In Figure 28, some of the agents were equipped with a fixed separation ratio, so they could only connect with neighbours sitting in a specific distance from them. The hypothesis was that the remaining agents would adjust to the fixed ones, which then would lead the whole system to obtain the same separation distance. Since, most of the agents initially was equipped with a larger value for separation distance, and quickly formed stable relations without interacting

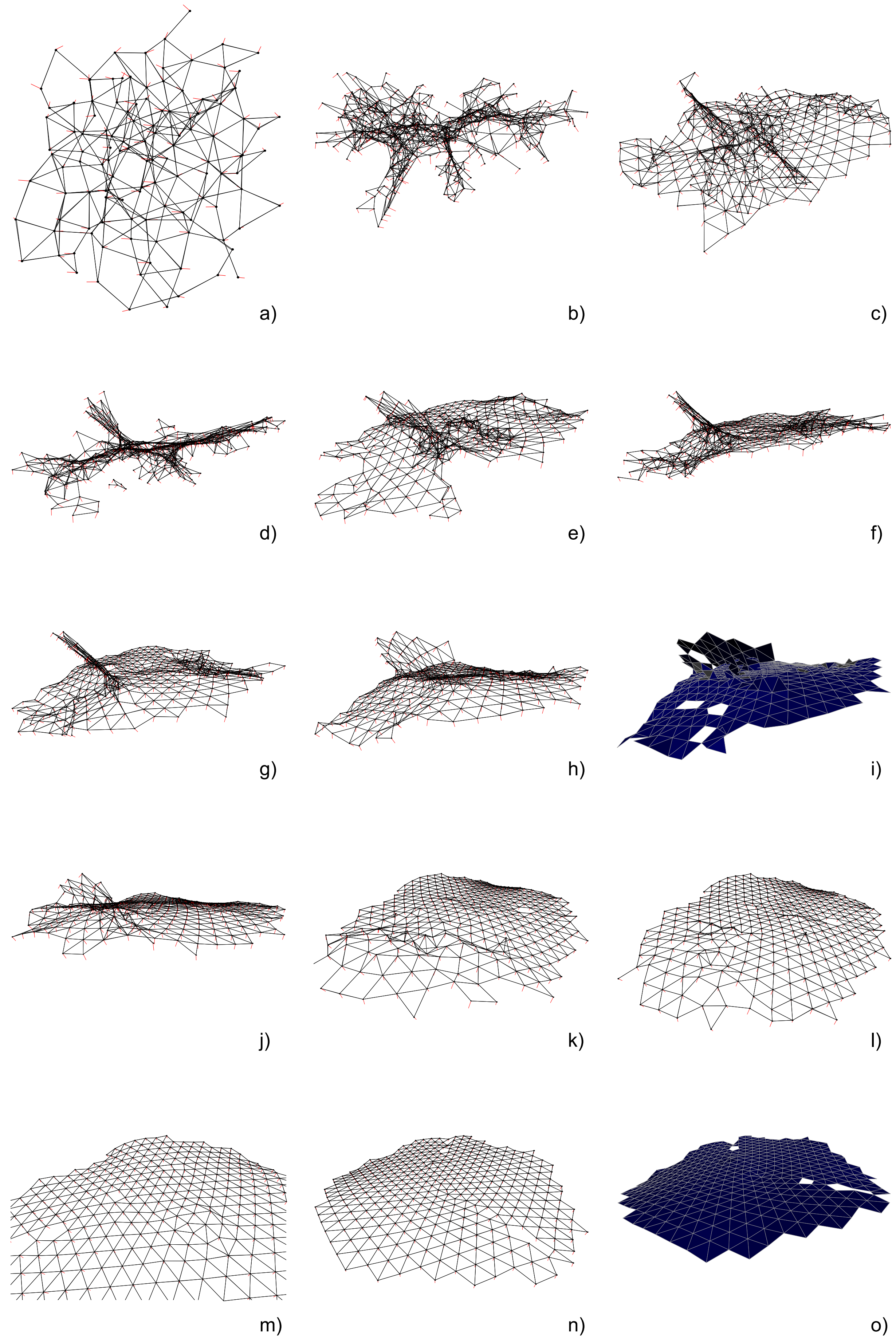


Figure 27. Opposite page: Sequence of self-organising surface. The agents negotiate the orientation of the surface normal, and eventually arrives at an equlibirum state.

with the ‘special’ agents, they were reluctant to reduce their values. Interestingly, the special agents would form a small cluster, almost excluding the normal agents. This happened despite that no type of attraction was implemented as part of the agent’s behaviour. Only general cohesion, ensuring that the agents stay together as a whole, was implemented.

Figure 29 shows an even more radical isolation of agents. Here, some agents were defined with fixed surface normals. In other words, the normals would no adjust to the neighbouring agents. Since the direction of the normal was random, it was impossible to construct a relatively flat surface, embedding all the agents, if the normals were to be oriented correctly. Figures 29a-d show how the system, as usually, arrived at a stage where a primary surface was formed. As in the previous example, the ‘difficult’ agents were transported to the edge, but were not embedded in the surface, since the normals could not be negotiated. Instead of staying unresolved, the system ‘decides’ to shoot off the unresolved agents, which then flies away, and the remaining surface settles in equilibrium.

Another example of exclusion is seen in Figure 30. Here, an attempt to introduce a shaping of the surface was performed. A single agent was programmed to try to establish an angle larger than 90 degrees to its neighbours. The rule also ‘infected’ the nearest neighbours in two rows away from the central agent. In this attempt, apparently, the angle was too steep for the system to accommodate, so after a number to states of negotiations, also this small enclave

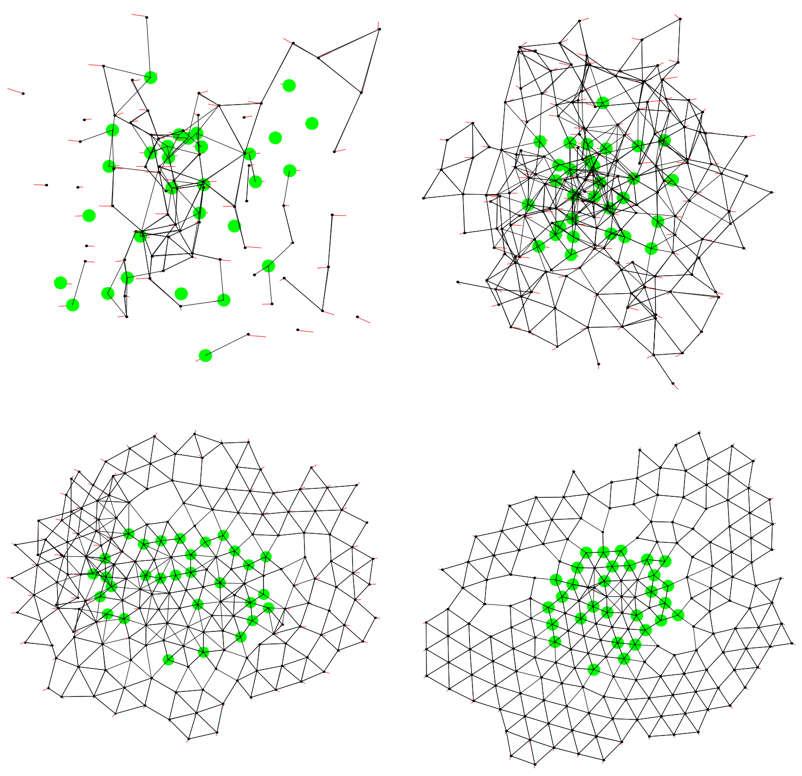
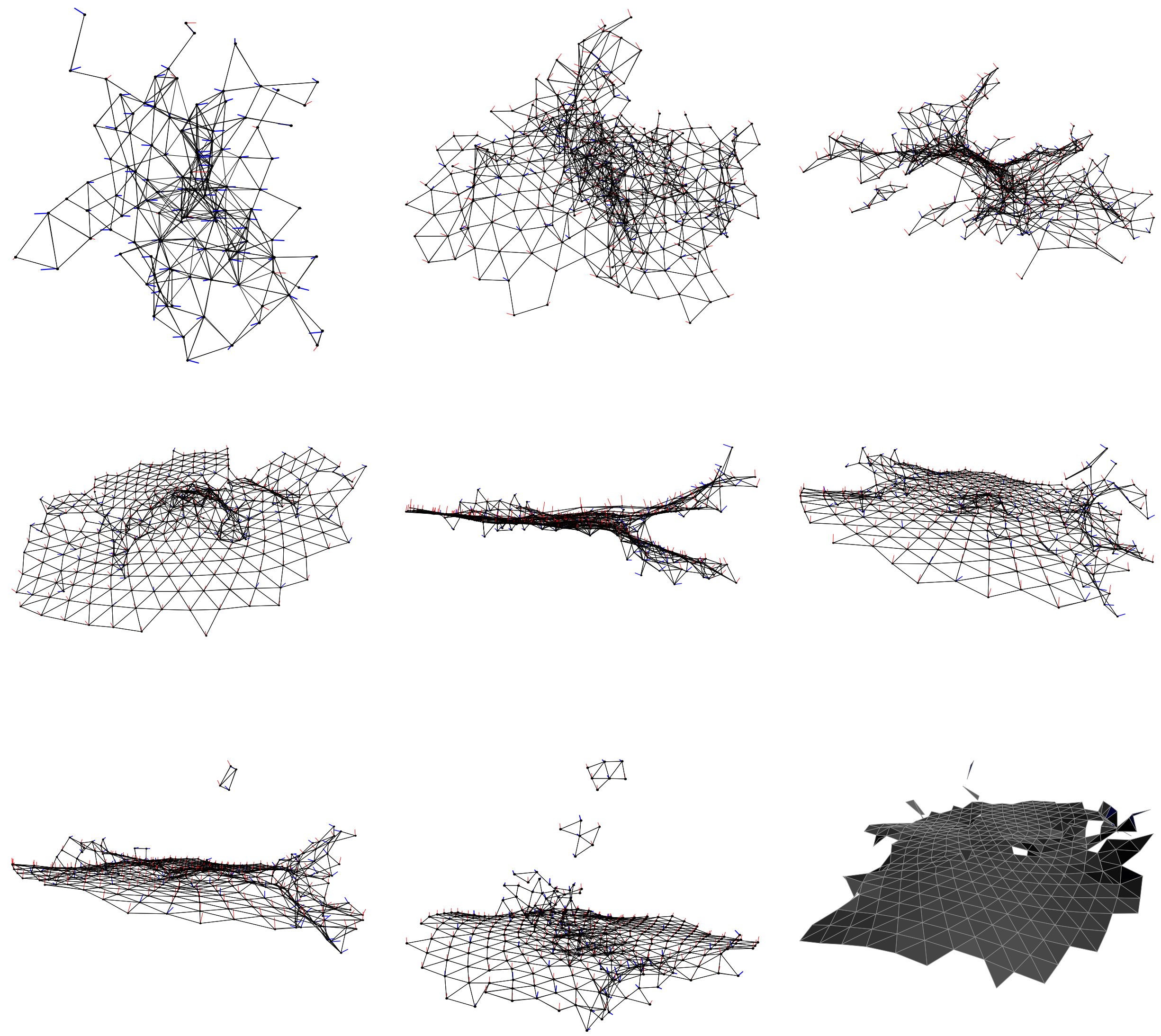


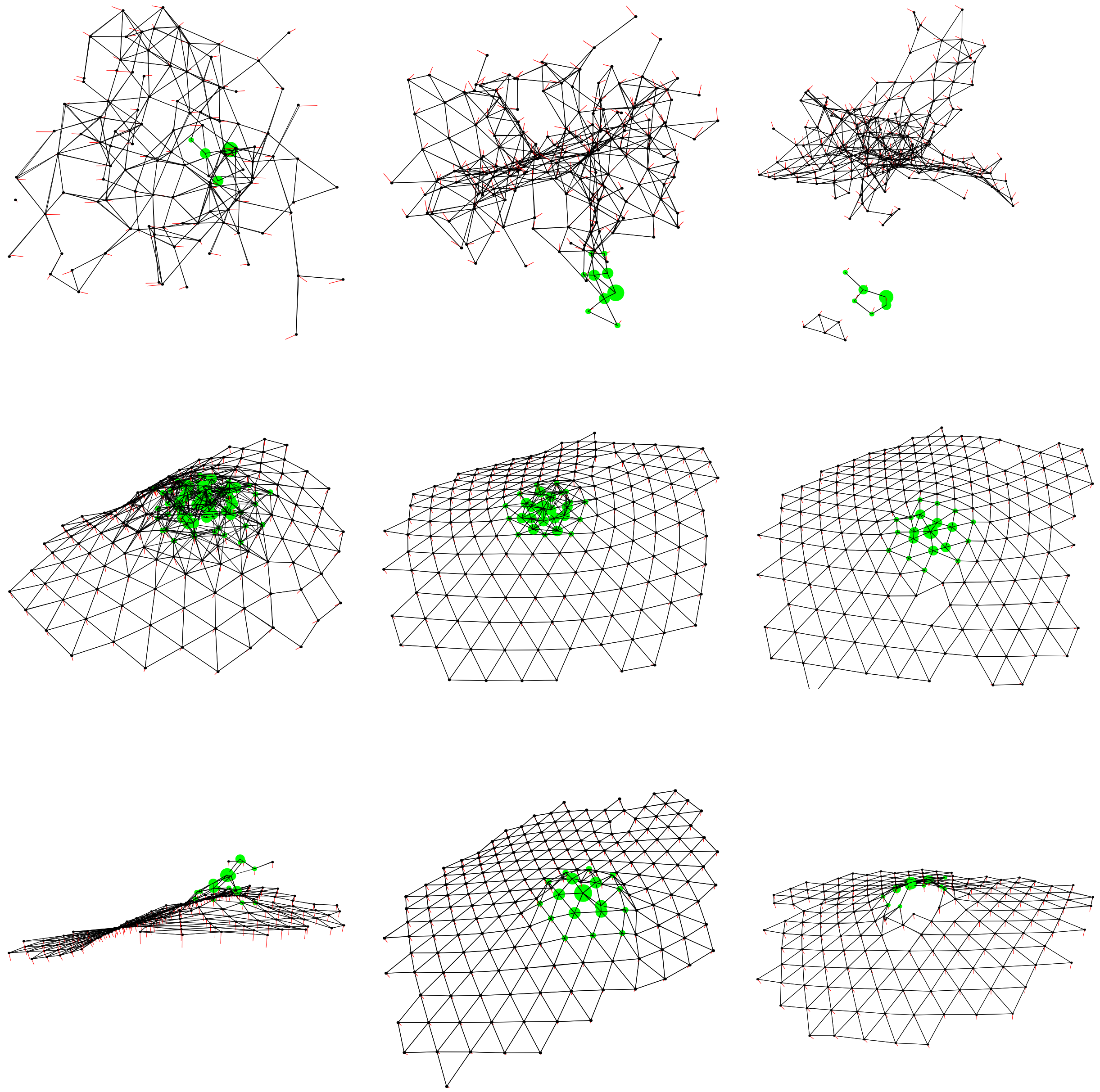
Figure 28. The green agents were set to a constant separation distance, lower than the global initial value. The rest of the system found a different average, and the green agents seemed to form a partly isolated cluster.



of ‘strange’ agents were shot off in a vertical direction from the surface, where after the remaining surface quickly healed. Later, the behaviour was explained from the fact that the change of angle has a side effect, since it results in a force, perpendicular to the surface. Figure 31 shows a different version, where the angle changes gradually towards the ‘peak’ agent’s position. This allows the surface to adapt to the curvature without breaking. Figure 32 shows a series where the behaviour of all the agents is changed in order to establish a doubly curved surface. The first images illustrates how certain parts were detached in the initial states, but reunites later again because of the coherence force. The system was gradually stabilised, forming the expected doubly curved surface. However, it was revealed that stronger curvature was difficult to achieve, partly because of the perpendicular force, mentioned before.

Finally, ways of directing the generated surface was studied. The goal was not to return to a method for *covering* a field of attractors, as with the previous method. Rather, the idea was to

Figure 29. A limited number of agents were set to have random, but fixed, orientation of surface normal. When an average orientation began to settle, the special agents were gradually transported to the edge. Finally they were shot off, and the remaining surface stabilised in an equilibrium state.



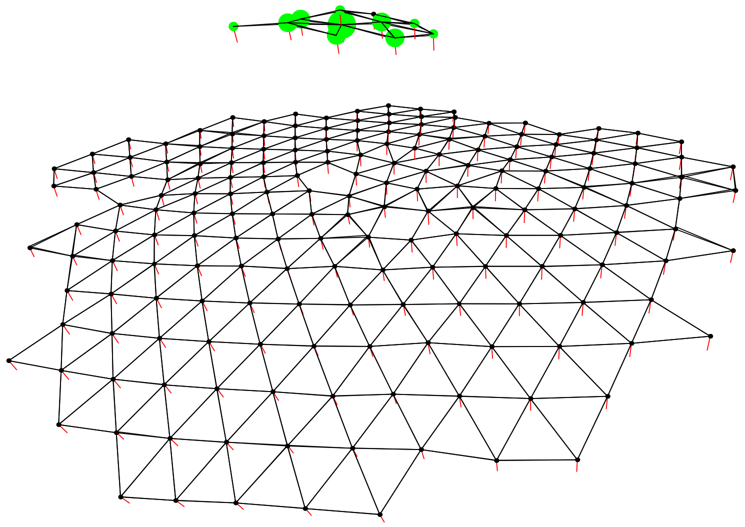


Figure 30. A single agent was coded to have an angle of 100 degrees between the normal and the surface plane, which suggests the forming of a hill top. The behaviour was transferred to the nearest neighbours. The relation could not be solved, and the infected agents were automatically excluded by the system.

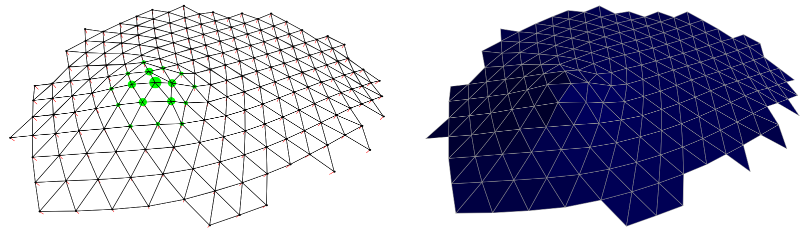
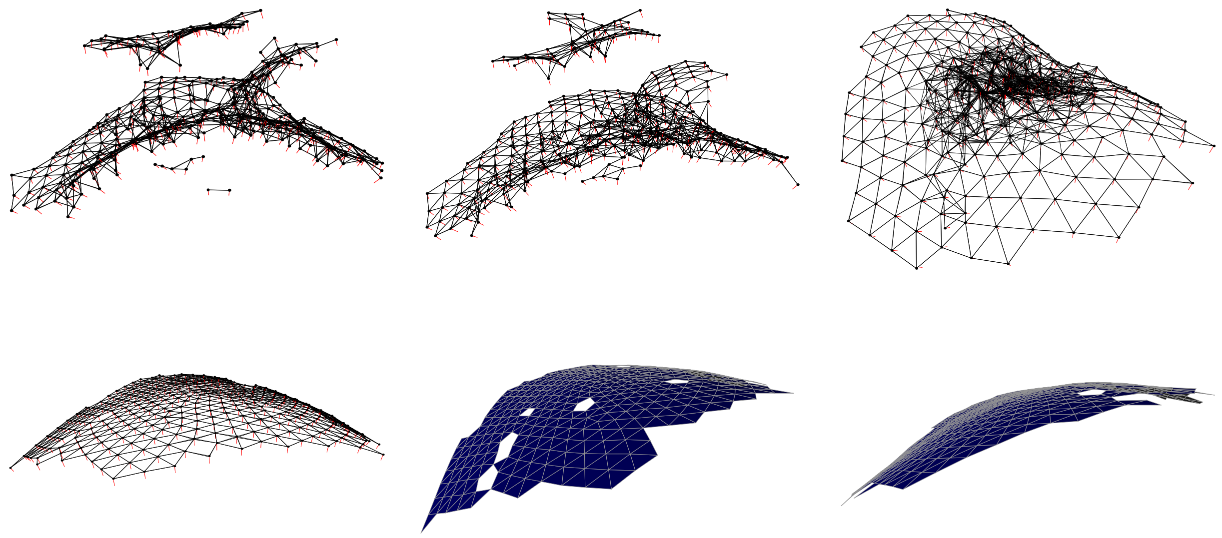


Figure 31. Same experiment as Fig. 32, but with an angle of 95 degrees. The more gradual transition allowed the forming of a hill top on the surface.

establish a negotiation between the self-organising surface and a regular geometry, represented as attractor points. Early attempts were made, where, instead of attractor points, some of the regular agents were defined as *fixed*. This meant that they would stay in the initial position. Because of the forces in the system, the other agents would form a surface where the agents to some extent were embedded, but the internal logic meant that the fixed agents often formed sharp irregularities within the surface, or even failed to become part of it. Therefore, a different approach with use of attractor points was implemented. The functionality is similar to the previous method, but without the agents being repelled by the attractor within a certain distance. This means that the agents seek towards the attractors, but because of the influence from the surface-generating forces, they do not directly touch them. Except, when the attractor points are all on the same plane. In order to improve the systems ability to adjust to curvatures, an additional function regarding the surface normals was implemented. The new behavioural rule is shown in Figure 33. Now, the agents adjust their normals to better reflect the surface they are part of, rather than just copying the normals of their neighbours. The previous simplicity guides the system towards a flat surface, and no other functionality affects the normals to point in other directions. In the altered system, a type of negotiation between *flatness* and *adaption* is established with regards to direction of the surface normals. In short, the system can better adapt to curvatures.

Figure 32. Continued experiments with curved surface. Here, all the agents seek a curvature, and after dramatic negotiations, a doubly curved surface is formed.



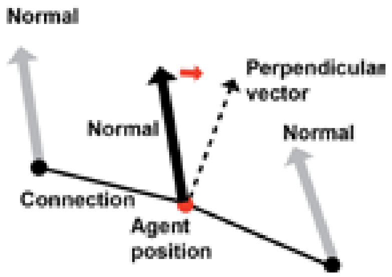


Figure 33. The agent's surface normal is adjusted towards a direction that is an average perpendicular angle to the connection lines.

In Figure 34 it is shown how the system behaved when exposed to a right-angled frame, similar to well-known examples with minimal surfaces, formed with soap film. As the figure shows, it was possible to achieve a similar surface with the self-organising agents. Besides the fact that the surface does not actually connect to the frame geometry, it is also not a minimal surface, because most of the forces that affect the system are directed towards establishing a surface structure, rather than optimising it. However, it would be possible to extract the surface geometry into a proper spring system and perform an actual dynamic relaxation process. It was very difficult to arrive at the shape by just letting the system run without any parameter adjustments, but a method was developed, where the system was initiated with only 250 agents. The first agents formed the first fragments of the surface, and their limited amount allowed them to synchronise their normals so all conflicts were resolved before more agents are added. The acceleration force was set to a high level to provide the necessary dynamic for the negotiations. Gradullay, more agents were then added, while maintaining the initial consistency, until the surface was fully established with approximately 650 agents. At this stage, the surface was in equilibrium, and the system was damped, in order to fine-tune the shape. Previous approaches, where separate surface parts were developed in the early phases, and then gradually joined, proofed to be difficult to control. The problem was that often, the surface parts would diverge in orientation, and when they began to merge an infinitely long process of negotiating the orientation would take place. The system would sit in a sort of deadlock, and only dramatic change in parameters could reboot the process. Sometimes, the system would accidently form harmonically, but the approach described above was more likely to give a successful result. However, after fine-tuning the system, it appeared to generate a smooth saddle shape in a majority of the times, also when it was initiated with 650 agents from the beginning.

Since the system seemed to have a behaviour reminiscent of soap film, a more complex setup was tested, namely, a full cubic frame. This is a classic example of soap film's self-organising properties, as shown in Figure 35. Figure 37 shows that it was possible to arrive at a result, almost representing a minimal surface, similar to experiments with soap film, as shown in Figure 36. However, the saddle shape was only generated in a small part of the tests. Rather, the system would arrive at equilibrium in a more unresolved state, as shown in Figure 37. Often surfaces would meet in a type of T-formed branching. This is also typical for experiments with soap film, even though, the generated shapes did not exactly represent resolved minimal surfaces. It should be pointed out that generation of minimal surfaces was not originally the goal for the experiments. Rather, the idea was to demonstrate and study the

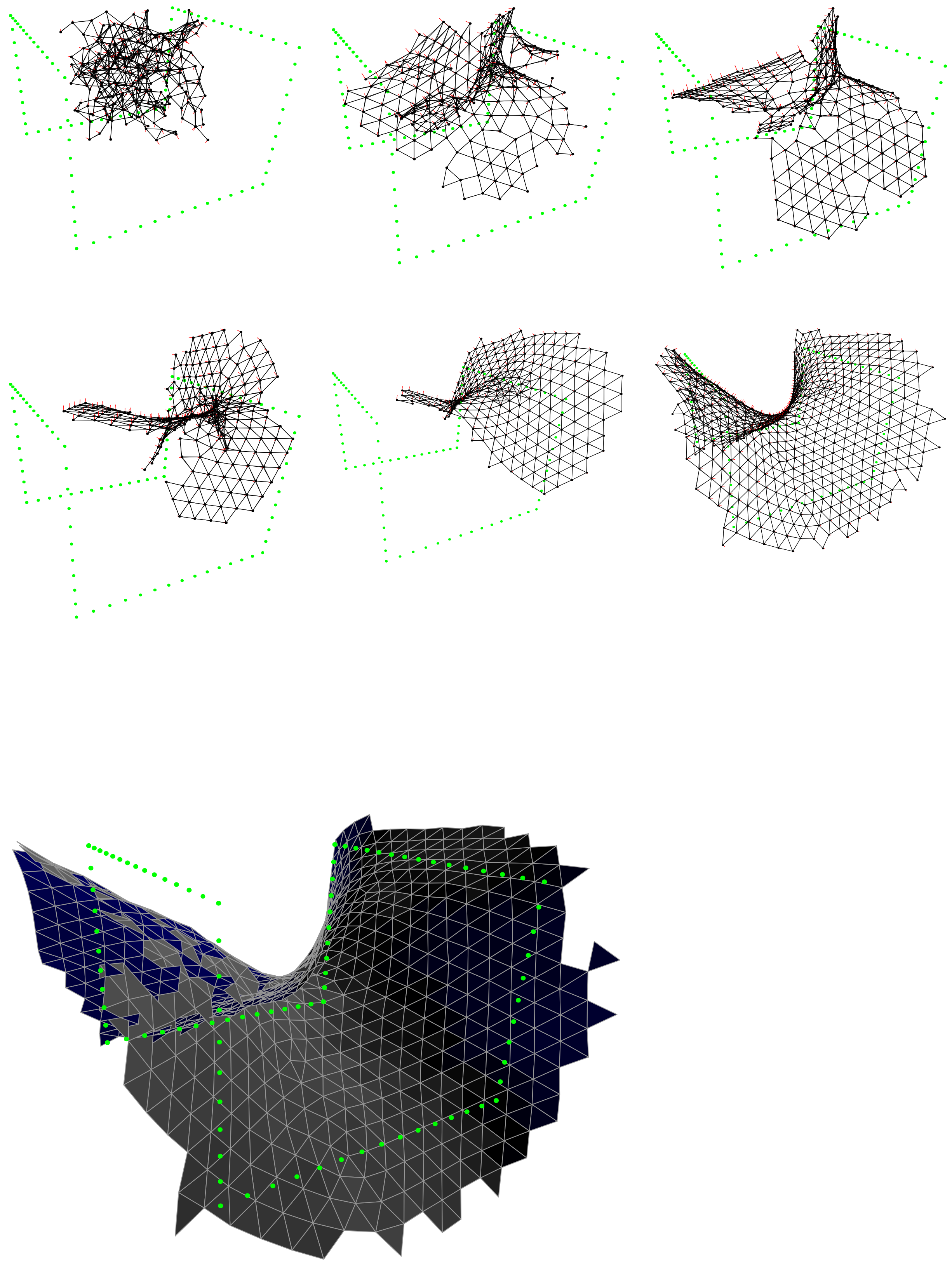


Figure 34. Forming of a saddle shape through use of guiding attractor points.

self-organisational potential of the system. The examples showed that the agents were capable of self-organising into a resolved surface structure, and that it was possible to guide the formation to a certain degree through use of attractor points. Some more attempts to guide the self-organising process was performed. Figure 38 shows an experiment with a spiralling shape, inspired by similar experiments with soap film. Because the logic of the system to does simulate soap film molecules, and because the generated surface is not constrained in extension by the attractor points, it only to some extent mimics a minimal surface. Rather than defining a physical frame for the process, the attractor points help to guide the formation of the surface. In the case with the spiral, the surface glides along the line of attractors, and the centre line allows the surface to break and define an edge close to the centre. The figure on the right shows how, often, the surface would have conflicts from different orientation of the surface normals. This resulted in the surface not arriving at an equilibrium state. The surface could appear stabile, but the agents near the problematic area would continuously shift position, trying to resolve the problem. Another test, shown in Figure 39, showed that it was relatively manageable to make the surface form into tubes. Attempts to make the system self-organise more complex shapes were also performed. The second image shows an experiment where the surface was guided by three times three circles of attractor points. The tight guidance was necessary to make the system generate a solution, where three tubes would meet in a T-joint. Interestingly, the edges were clearly defined at this stage, effectively stopping the growth. More precise, the system stopped adding new agents to the scene, despite that the number was below the defined limit.

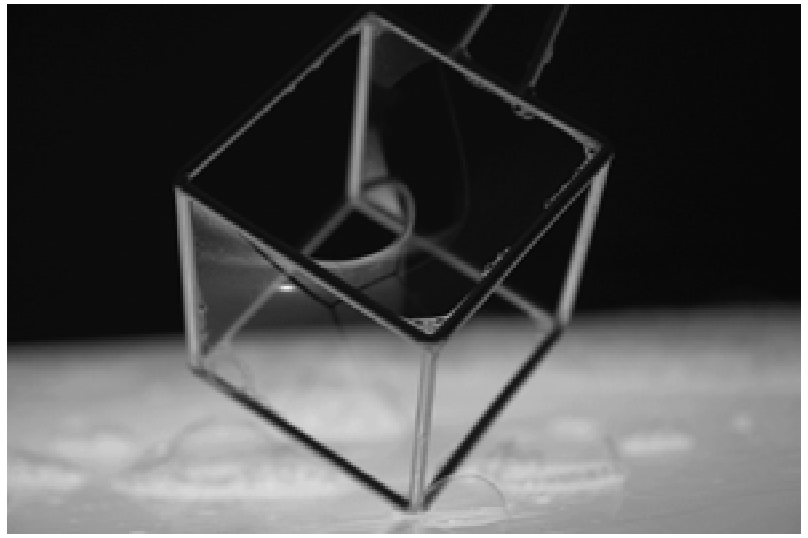


Figure 35. Minimal surface of soap film, generated in a cubic frame. Photo: Soapbubble.dk

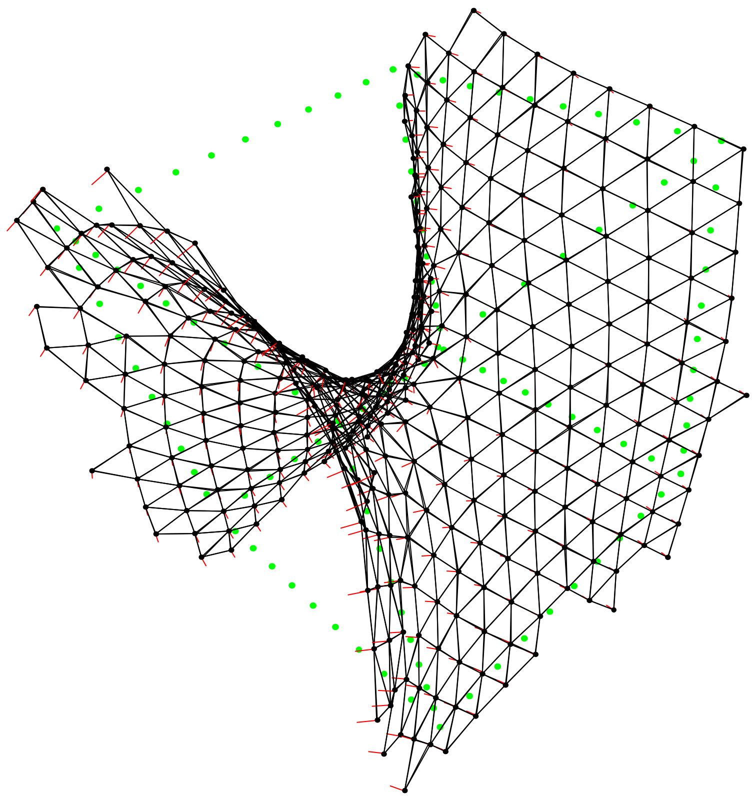


Figure 36. A shape, reminscent of a soap film surface, could occur when using a cubic frame of attractor points.

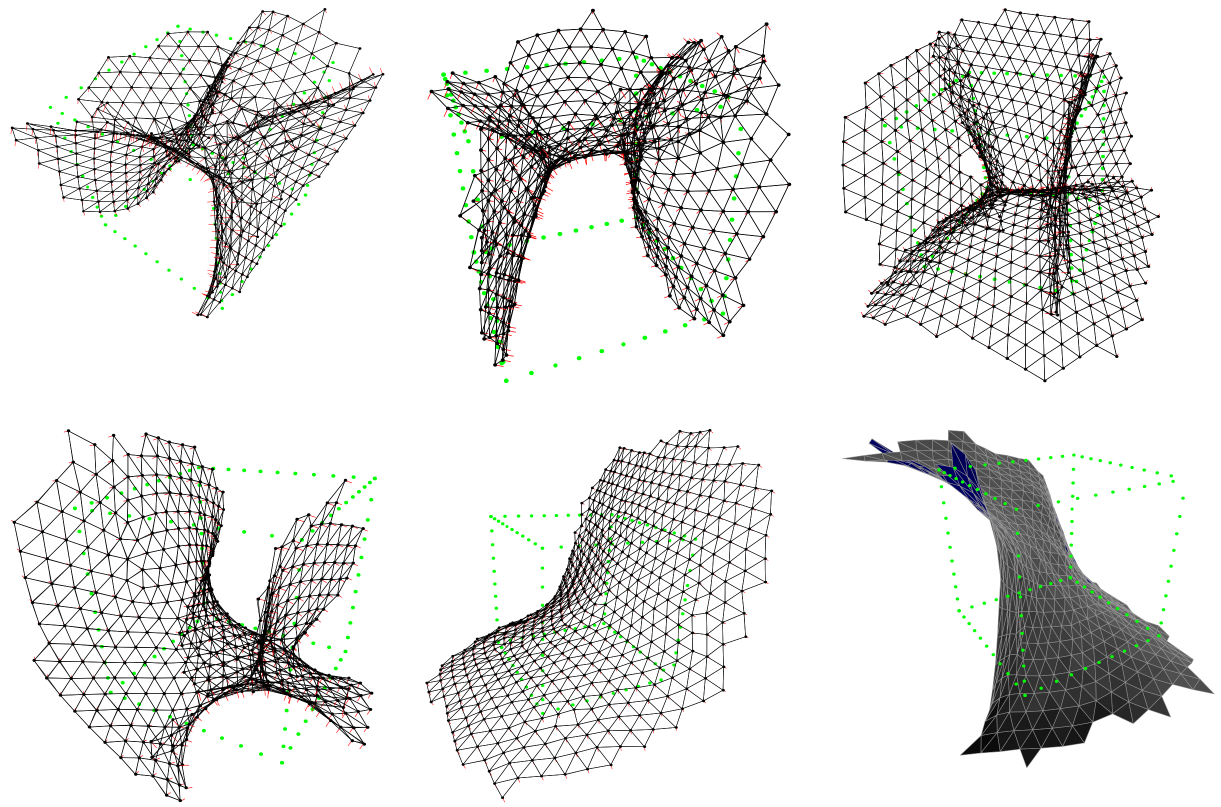


Figure 37. Generally, with the cubic frame, unresolved and complex shapes would appear. Where more surfaces were joined, the system would continue to be unstable despite that the majority of the agents were in an equilibrium state.

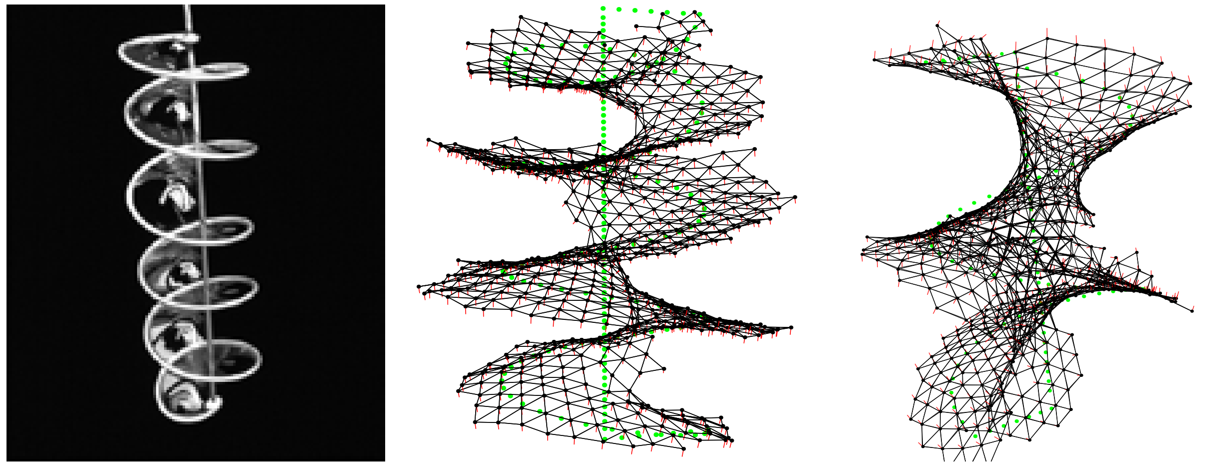


Figure 38. Spiralling minimal surface generated with soap film. Photo: Exploratorium. Middle: A similar shape is achieved with an experiment where the attractor points represent the base geometry of the model. Right: Often, the result would not be a continuous surface.

8.4.10 Observations

As mentioned earlier, most observations are included in the explanation of the methods. Here follows a few concluding remarks. Agent-based systems proofed to be well suited for developing self-organising systems. Because the agent logic could be linked with three-dimensional orientation, it was relatively simple to transfer the logic to spatial geometry. In this sense, the flocking algorithm served well as a starting point for investigating agent-based systems. Furthermore, the vector based calculations pointed towards use of non-Cartesian geometry. Particularly when compared with other methods for working with self-organisation, which often are based on a rigid grid. The flocking algorithm is inherently spatially complex, and the challenge was to learn to constrain it, rather than to expand the complexity of the geometric outcome. The strong dynamic of the system, deriving from the simulation of birds, appeared strongly in the initial experiments, but the methods for constraining the agents showed that the method can provide means for generating different types of varied surface patterns with potential for architectural design. While, these methods could be useful, it was difficult to maintain the original self-organisational potential, and control them in order to solve a specific design problem, at the same time. The method for generating self-organised spatial structures helped to bring the potential of self-organisation back into the picture, without returning to the image of the bird flock. Most promising for future use, with respect to self-organisational systems, was the method for a self-organising surface. The main achievement with this method was that it demonstrated how a disordered field can self-organise itself into a formation that shows topological order that was undefined in advance. The method is fundamentally different from other some of the other methods, described in the thesis. Isosurfacing is a procedure that generates geometry after the self-organisation has taken place. It can be done simultaneously, but it is technically difficult to integrate the self-organising logic with it. More directly related is perhaps the methods for dynamic relaxation. With these tools, the surface is predefined, or at least the nodes and the relations between them. The nodes can very directly be compared with the agents described here. The difference is exactly that the agents in the example were initiated with undefined relations and established these relations through the generative process. Importantly, the relations represented a self-organised topology. The crucial gain from this other approach is that it becomes possible to establish negotiations between a series of parameters directly in relation to the topology while it is being constructed. Shortly, it can be said that with isosurfacing the surface is constructed subsequently to the topological negotiations, and with dynamic relaxation the topological relations are predefined. The method of producing an entirely self-organised surface was, as explained, not fully developed. Future developments could seek to explore how the method could adopt ways of responding to external parameters such as structural properties, light conditions etc. These influences could perhaps also help to provide the system with a defined framework, which again could help to constrain and fine-tune the functionality.

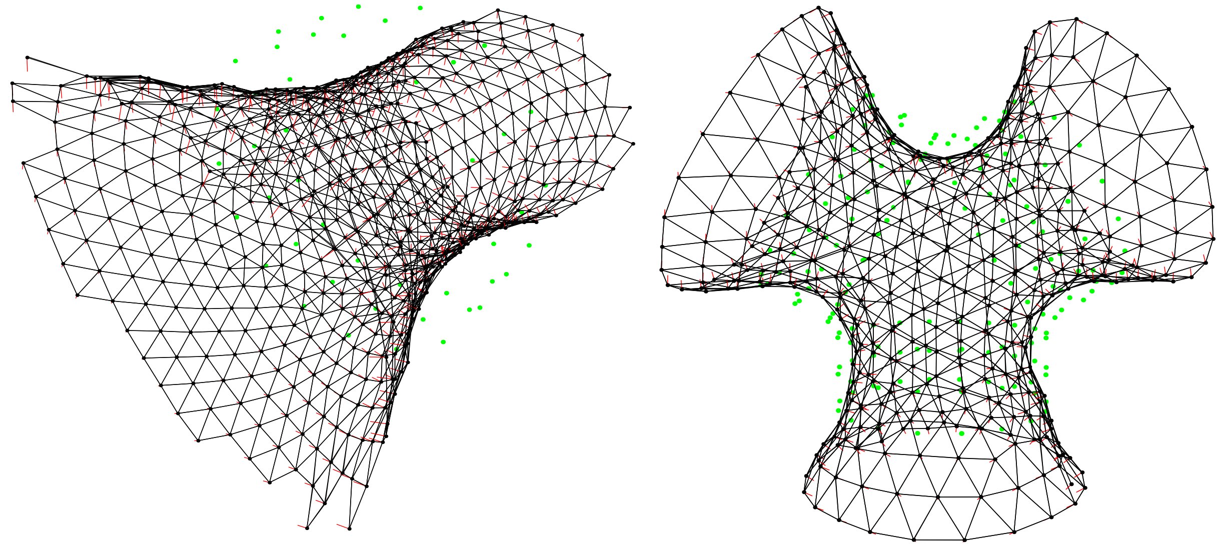
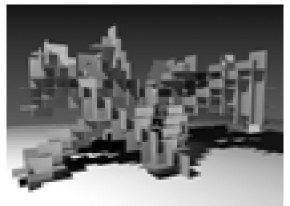


Figure 39. Left: A tubular shape could be formed with two rings of attractor points, placed within a certain range of distance from each other. Rigth: A branching surface could be generated via intensive guidance with 9 rings of attractor points.

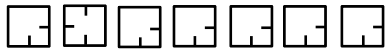
	**Intent and conditions**	**Organisational logic**	**Realisation**
Properties	Space filling growth User participation in build-ing process Computation / realisation Rule based differentiation	Recursive growth Deterministic principle Self-reproduction Randomness in the form generating process	Structural challenge. Use of specific technologies. Scale and realisation. Temporary construction.
Observations	Complexity from realisation Algorithmic variation Character and control. External goals and intents	Limiting the design space Variation through rules Structural/practical factors External parameters	Implications from realisation User influence increases randomness Simple joint Low tolerances

8.5 SAGA

8.5.2 Intent and conditions

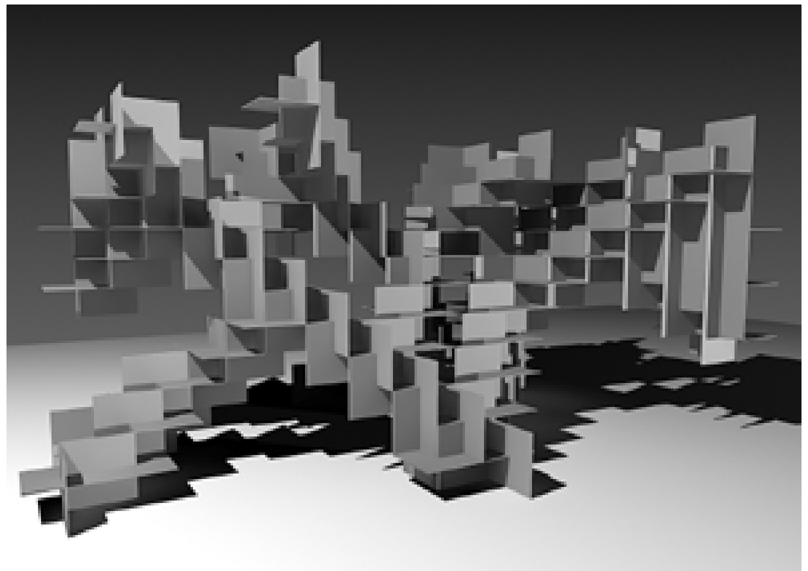
The aim for the project was to use an algorithmic method to create differentiated spatial structures from a simple component. The project was basically carried out in two steps. First part was the development of the tools needed for exploring the growth substitution principle. The second step was focused on production of and assembly of the physical model. The experimentation was based on a growth substitution algorithm with a seven steps growth sequence. It was decided to seek for a principle that would allow the growing structure to connect to itself, and this was solved in a straightforward way, by inscribing the growth in a virtual three-dimensional grid. This again led to use of a planar square building component. Part of the experiment was to investigate to what extent it was possible to change the aesthetic character of the structure, only by changing the sequence of operations. This was mainly explored in the virtual models, where the deterministic systems are unfolded in a precise manner. It was decided to allow an amount of randomness in the physical realization of the structure, giving a reduced emphasis on the algorithmic principle. The idea was to make people engage in the assembly of the structure, giving them a role as agents making decisions affecting the growth and the final outcome of the process. These conditions were defined as a framework for the experimentation carried out, and the project was developed in order to look at certain issues:

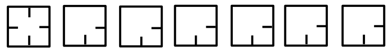
- To what extent would the minimal possibilities of changing the growth sequence provide a variation in the character of the structure?
- What implications occur when translating the abstract growth of the geometry to physical model?
- How would the randomness allowed in the physical model affect the character?
- Would the physical model achieve characteristic patterns besides the tectonic principle, when only partially controlled?



Growth sequence 1511111
Generations: 30
Objects: 272

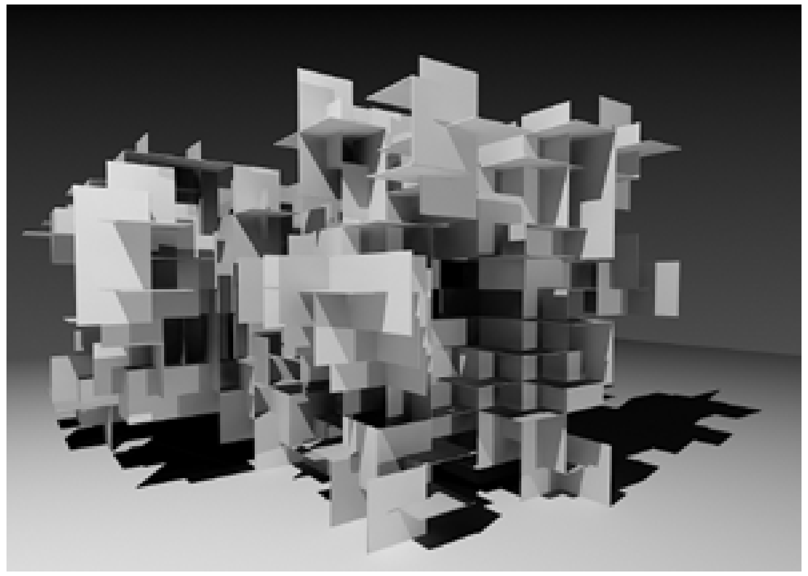
Figure 1: The 90º rotation, denoted by the digit 1, affects the growth in a way that gives the structure a kind of organic expression. The digit 5 denotes that a two way branching can take place, and since only one of the steps is a branching procedure, the structure maintain an amount of openness.

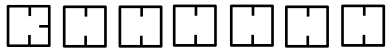




Growth sequence 4111111
Generations: 30
Objects: 1031

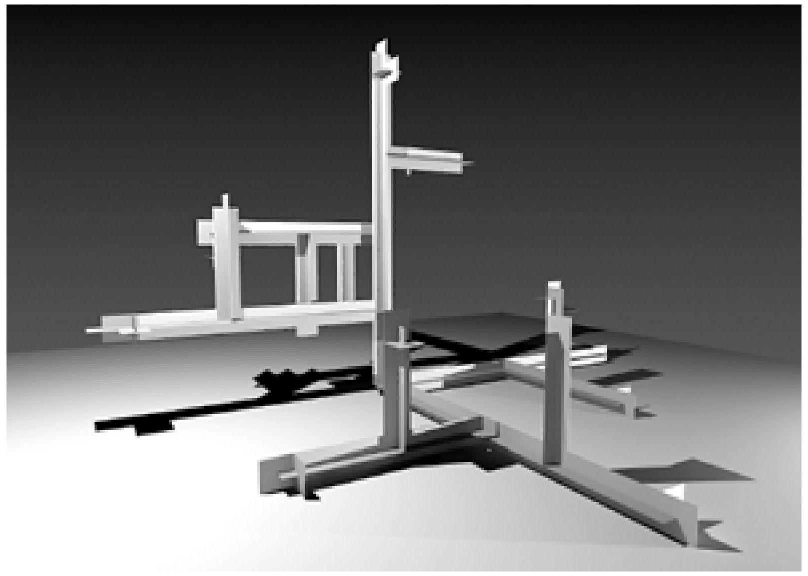
Figure 2: The three way branching, indicated by the digit 4 in the beginning of the growth sequence leads to high density in the structure, compared with the previous example.





Growth sequence 3000000
Generations: 30
Objects: 264

Figure 3: The 0 digit indicates a straight forward growth, resulting in a structure that resembles building structures with columns and beams. Also, the extent of the structure is maximised by use of the grow forward step.



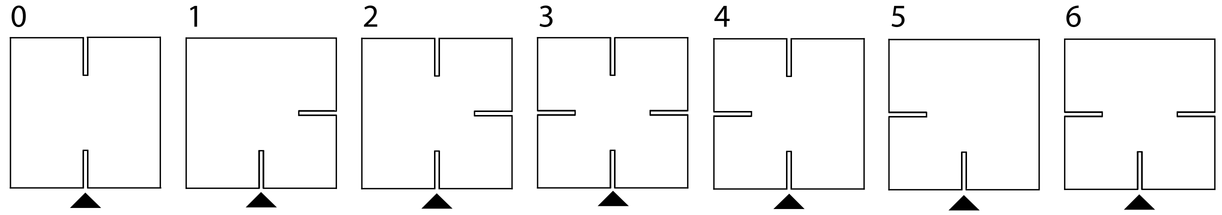
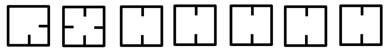


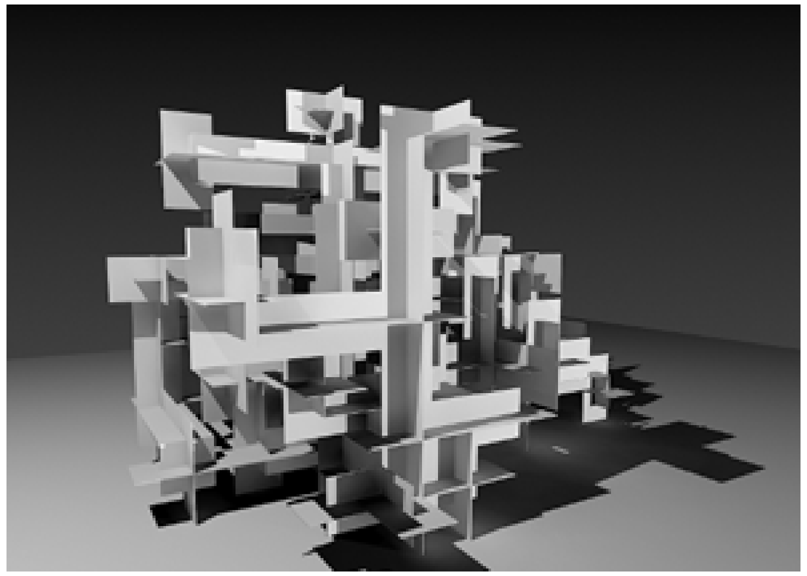
Figure 4: The 7 component types. The triangles show the point where the new component connects to the existing structure. The empty slits are the potential growth directions. In the virtual model, all possible growth directions are effectuated, whereas in the physical model the builders decide whether an empty slit was used or not.

8.5.3 Organisational logic

The SAGA project is based on a recursive growth algorithm. The method is comparable to a DOL-system. However, it is not exactly a substitution process. Rather that subdividing an existing geometry, the system could be seen as a process of recursive reproduction of a basic component. As part of the reproduction, a rule-set embedded in each new component is updated, as a step in a determined growth sequence. The operations can be explained as follows: An original object is placed in a certain position, in this case a square plate. It is possible to place more start positions but most of the examples have only one. The original object is copied to a position that have a certain relation to itself. In this case it is moved 1/2 square length forward and rotated 90 degrees. This operation follows a sequence telling whether the rotation should be to one side or the other, whether the new object should be rotated 90 degrees in relation to the original object (make a turn), whether a branching should occur and in what directions this branching should be. Since only 90 degree operations are allowed there are only 7 different operations. In this case we used a sequence with 7 steps. If a component for example is on step 2 it means that it should branch in 3 directions in the examples shown on the right. The 3 new copies will then generate copies that only continue forward and do not branch. The plate components have different possibilities of branching. Since one of the sides is the starting connection the component can have up to 3 new connections in 90, 180 and 270 degrees angles. The algorithm controls the sequence of connections and the direction of the added component.

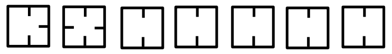
Some advantages derive from using a simple geometry, only using 90 degrees operations and a square component. The main achievement is the possibility of making simple joints between the different generations of objects. This means that if a branch is generated in the direction of the existing structure it is able to connect to it. The computational system works consistently to some extent. It simulates a building process in the sense, that if the growth sequence dictates positioning of a component below ground level





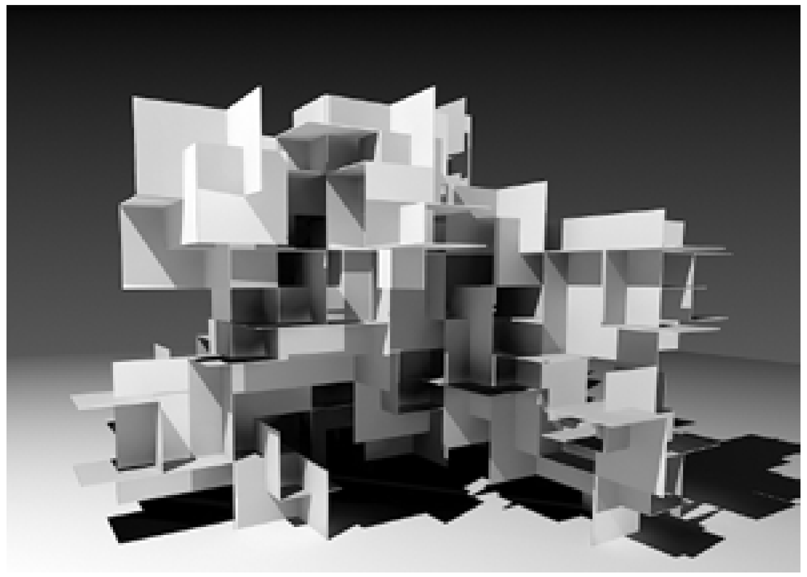
Growth sequence 1400000
Generations: 30
Objects: 824

Figure 5: By adjusting the forward sequence with a single 90º rotation, the character is change into a hybrid of an organic structure and an orthogonal skeleton.



Growth sequence 3400000
Generations: 30
Objects: 4018

Figure 6: The digit 3 is a way branching in two directions, where 4 indicates a three way branching. Generally, adding a branching step in the growth sequence increases number of components and the density of the structure exponentially.



Growth sequence 3511111
Generations: 30
Objects: 471

Figure 7: 3 and 5 both indicates branching, but since the sequence makes the structure rotate and connect to itself, the growth is limited compared to the example above.

or where another building component already is placed, the branch stops growing. However, it does not take gravitation into account, so very long branches can be generated. Also the practicalities of physically joining the components are not implemented. The script for generation the growth structure is developed in Rhino-script. The input for the script is the geometry defining the initial square component object, points defining the positioning of new copies, origin positions for the growth, the sequence of operations, defined as 7 figures and the number of generations. As a consequence of the growth principle, the extent of the structure is somewhat equivalent to the number of generations, where the density largely is defined by the growth sequence.

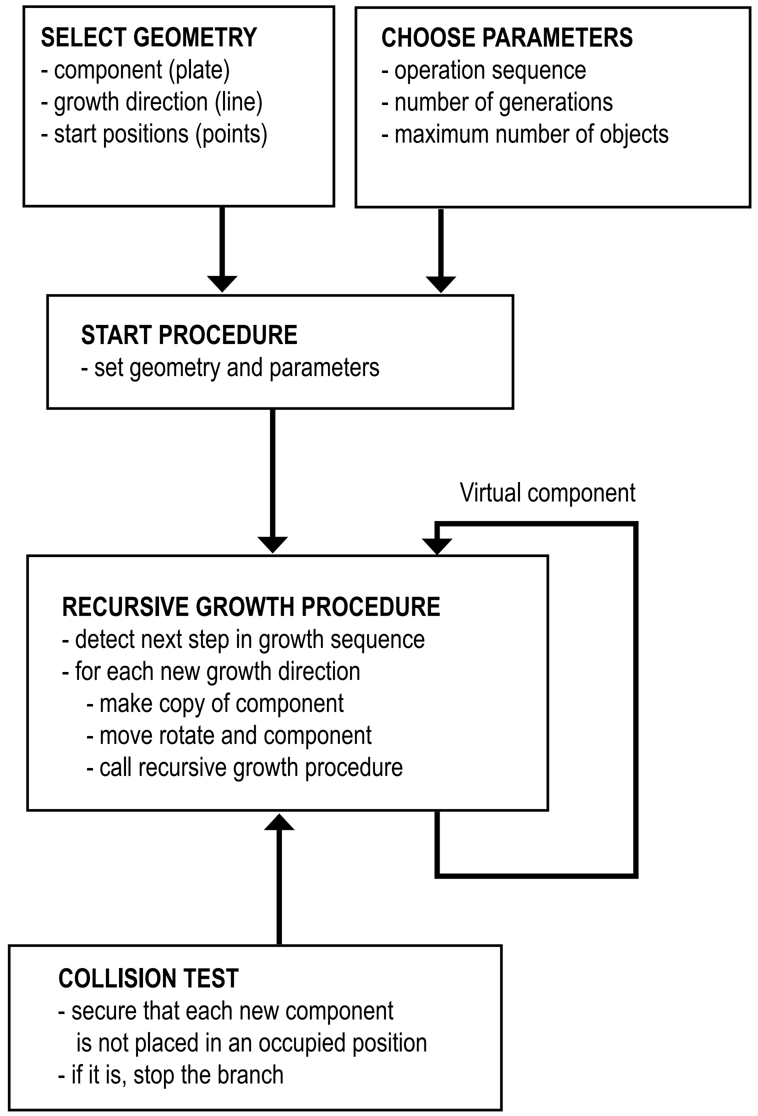


Figure 8: Diagram of the computational form generating process.

8.5.4 Realisation

One of the ideas behind the project was to build a structure based on algorithmic principles, simultaneously encouraging people to take part of the building process, and thereby affect the formation of the structure. The hope was, that it would be possible to guide the growth of the construction through constraints embedded in the building components. The idea was that a predefined sequence of component types, the physical practical conditions, and the engagement of the builders would encourage some type of order to appear, not too dissimilar to the character of the virtual model. Therefore it was decided to make components with a limited range of operations. The sequence of components consisted of rotation components type 1 and three-way branching components type 4 in a relation, where there were 4 rotation components for every branching component. In reality the type 1 component was also a type 5 component, leaving the direction of the rotation up to the builder. 128 components could branch in all 3 directions and 512 components only had 2 slits in a 90 degrees angle, thereby determining the positioning of the following component. The components were sorted into a sequence in order to distribute the different components evenly. A simple joint with interconnected slits in the component was developed in order to avoid any additional nails, screws or fittings. This meant on the other hand that the production of the components and especially the cutting of the slits had very low tolerances in order to prevent instability from loose joints. Despite the limited variety of operations that could influence the growth, the builders had substantial influence on the growth of the structure. Particularly two types of choices were made during construction:

1. The builders could decide where the growth should happen, since they were not restricted to update the whole structure in parallel, as it is the case in L-systems, and as it also happens in the computational model.
2. The design of the components did not imply in which direction the component should be rotated, where the virtual model followed a specified rotation sequence.

These factors meant that the realised model was fundamentally different from the virtual model. The growth process was stochastic, where the computational model was completely deterministic. This meant for instance that the extents of the realised structure was not directly linked to number of generations, or growth steps. In fact, the growth of the physical model did not happen in generations, but rather in steps for each added component.

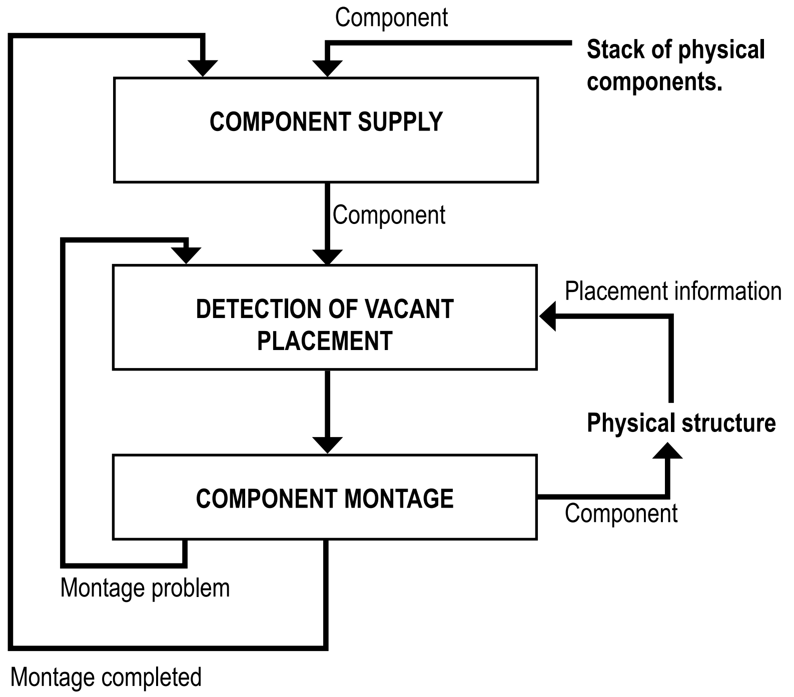


Figure 10: Diagram of the growth process in relation to the physical structure.

8.5.5 Observations

During the computational experimentation it was possible to generate an unexpected variety in the generated structures. It showed quickly, that the amount of branching in the growth sequence was determining for the character of the structural growth. The density was exponentially reduced when the amount of branching was increased. Another essential factor was the number of turning steps. It appeared that sequences with more turning steps achieved a sort of organic and complex character, despite the rectangular geometry. The sequences with mainly step forward actions achieved a kind of column-and-beam character resembling building components. In

Figure 9: Physical structure after completed growth process

terms of usability in relation to architecture, the method would need implementation of other types of controls. At least three trajectories could be relevant for further development of the virtual system:

1. Definition of a limited design space, constraining the growth. By setting up a predefined frame for the growth, it would be possible to generate structures with relevant architectural properties, such as light filtering or space division.
2. Implementation of structural and practical factors. If the virtual model should make any sense as a simulation of a structure for actual realization, it would be necessary to implement at least two factors. First, the structure should take into account the weight of the components and the strength of the joints. Then, it would need to have a way of analysing practical problems in terms of assembly. The latter would probably be most demanding, considering the difficulties appearing during assembly of the physical structure.
3. Influence from external parameters. Perhaps the most intriguing potential of further development would be to implement external parameters affecting the growth process, such as light sources and predefined attractors in the virtual environment. This would enable architectural goals to become part of the form generating process, perhaps leading to formation of walls and openings as a direct reflection of specific parameters. Also, in this case it could make sense to allow a certain amount of randomness in the growth, replacing the strict rules with tendency choices.

Figure 11: States of the building process. The builders could decide which growth possibilities were effectuated.

As expected, the realised structure became random, compared with the virtual model, since the builders could decide which branches should be developed. However, it is not quite as simple to make connections between the new branches and the existing structure in the real world as in the computer model. A branch would become unstable when reaching a length of 4 or 5 objects if they are not connected to the main structure in more the one place. This was an encouragement for the builders to develop the structure in order to make it more stable and at the same time giving it a certain character. In some cases it was even necessary to disassemble parts of the structure in order to be able to add new connecting parts. The goal was not to arrive at a final stage that would meet any specific requirements. Rather the structure would grow until people stopped or ran out of components. The realisation of the physical structure

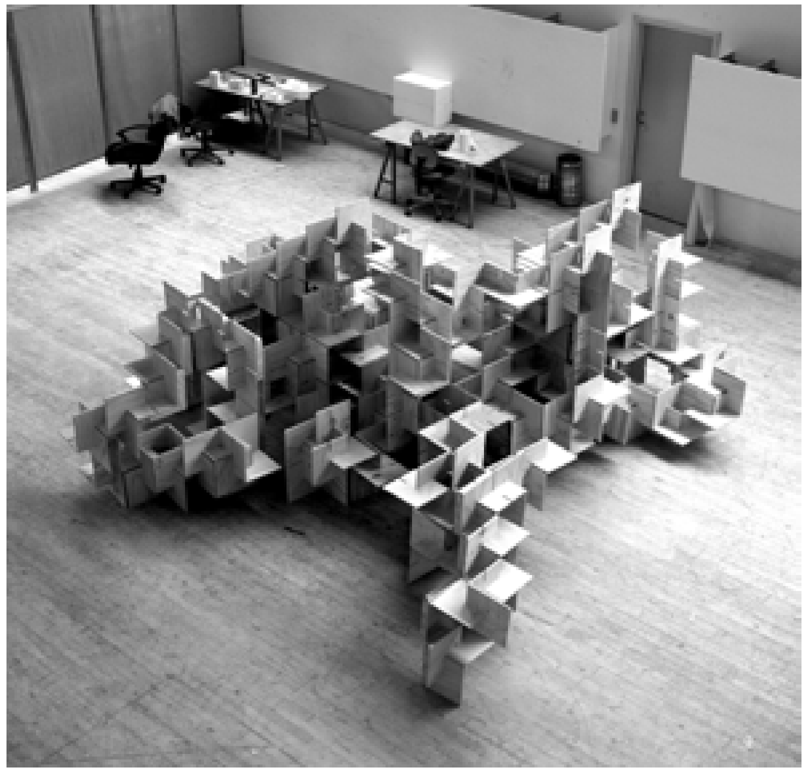


Figure 12: The realised SAGA structure. The structure gained a random expression. However, the rotational component type affected the tectonic expression substantially. The organic character, discovered via simulation, is also recognisable in the realised structure.

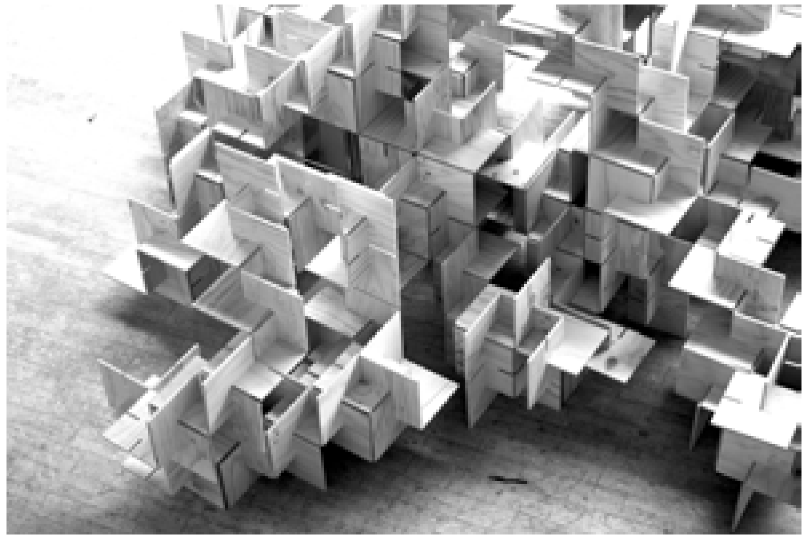
revealed a number of important issues, which were not taken into account in the computational model. Despite the simplicity of the building system it turned out to be relatively demanding to assemble the structure. Still, it only took approximately two hours to assemble 650 components with a group of 10 people. The difficulties derived partly from too tight fitting of the slits, instability of the structure and practical problems when adding connecting pieces. The realised structure has a certain tactile character, deriving from the material quality of the cross veneer material and the tectonics of the building system. The result is a kind of organic, porous but sharp-edged cloud. As implied, it has a very random character, and the constraints did not have any major impact on the result. A future development could be to constrain the growth to a larger extent in ways not dissimilar from the 3 issues mentioned above in relation to the computational model. In short, these issues were pointed out:

1. The structural properties of the component design were not optimal. Simply, the positioning of only one connection point in the centre of the plate made the structure relatively instable.
2. The simplicity of the joint resulted in extremely low tolerances for the production and difficulties in the assembly. Development of the concept would have to consider a more versatile joint solution. An additional joint piece could be relevant.

Figure 13: In some situations, the algorithmic logic occurred in the random growth, possibly as a result of conscious decisions by the builders.

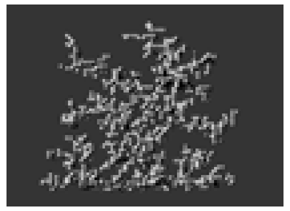
Figure 14: The orthogonal logic of the tectonic system lead to a certain order in the general expression of the structure, and in parallel allowed a large degree of spatial complexity to emerge.

Figure 15: When seeing the structure from certain viewpoints, the organic character of the tectonic system was apparent.



3. The realised structure appears random to a large extent. A future development could overcome this tendency in different ways. The building system could be more constrained, not allowing the components to rotate randomly, and a sequential numbering system could be added. Perhaps more interesting would be to add external goals for the growth process, such as interaction with the physical context or establishment of spatial enclosure.

To summarise the 4 questions posed in the beginning of the chapter, it could be said that the concept proved to be able to demonstrate great variety within the very limited scope of rule differentiation. The difference between virtual and physical model proved to be surprisingly large, especially when taking the simplicity of the building system into account. The physical model appeared as substantially random, differentiating it from the virtual experiments. Most surprising was the increase in complexity when transposing the method to the physical realisation. The realisation raised important issues of which some could be implemented in the virtual model through further development, but which probably would not have become evident without the realisation process.

	Intent and conditions	Organisational logic	Realisation
Properties	Generative growth system Negotiation in the form-generation process Computation versus realisation Degrees of optimisation	Diffusion-limited aggregation Conditional growth Iterative negotiations Sun path simulation	Agent interaction Feedback processes Self-organisation Stigmergy
Observations	Specified rules. Complexity controlled in the digital environment. Optimisation	Adaptable DLA as viable pattern Organic character Additional parameters	Local top-down approach Rigidity of rules for growth Tectonic solution undefined. Structural parameter

8.6 Solar DLA system

8.6.3 Solar DLA System: intent and conditions

The Solar DLA System is an experiment rooted in a workshop cluster at the Smart Geometry conference in 2011[1]. During the workshop a physical model was realised, referred to as the physical model in the following. During the workshop, different form generation methods were developed from a generic DLA algorithm. Subsequently one of the methods have been further refined and developed, namely the Solar DLA System. The system is a generative method that combines the DLA model with sun radiation simulation. Both the virtual and the physical models are based on a cell component shaped as a 14-sided polyhedron called a truncated octahedron. The components can be connected to each other in all 14 directions. The cell-packing principle implies that the components are positioned in an underlying spatial grid structure, which again means that the components always can attach to each other in a precise way, when parts of the structure connects to other parts. In principle, the system is as rigid as a square grid, but the diagonal growth directions adds a larger geometric variety to the character of the formations.

The aim for the initiating workshop cluster was to investigate how a complex structure can be generated through agent interaction. The phenomena of stigmergy and feedback processes were studied with inspiration from research in termite mound construction. The goal was to gain knowledge about how these processes could become useful in relation to development of self-organisational tectonic principles in architecture. The investigation was carried out partly through development of digital generative systems and partly through humans acting as agents on a physical structure being realised throughout the workshop. As David Andreen writes in the workshop introduction: ‘Several challenges must be overcome to be able to implement these strategies in architecture: extending our ability to sense and respond to fluctuating processes, building with much greater complexity, continuously altering structures in response to changing conditions, and developing software which can act, respond and negotiate in the same way organic agents do’. The growth processes involved interaction between both the agents and between the structure and the agents, understood as the agents responding to, and changing, the formation of the structure during the growth. Also, the similarities and differences between a purely digitally generated system and a physically grown system were studied.

Figure 1: The 14-sided polyhedron was the base component in the experiment. The illustration indicates how identical polyhedrons can be added in 14 different directions.

1. Rupert Soar, David Andreen and Petra Jennings were responsible for preparing and organising the Agent Construction workshop cluster.

Figure 2: Spatial DLA controlled growth. Top: The structure obtains an open branching structure from the inner logic of the DLA.

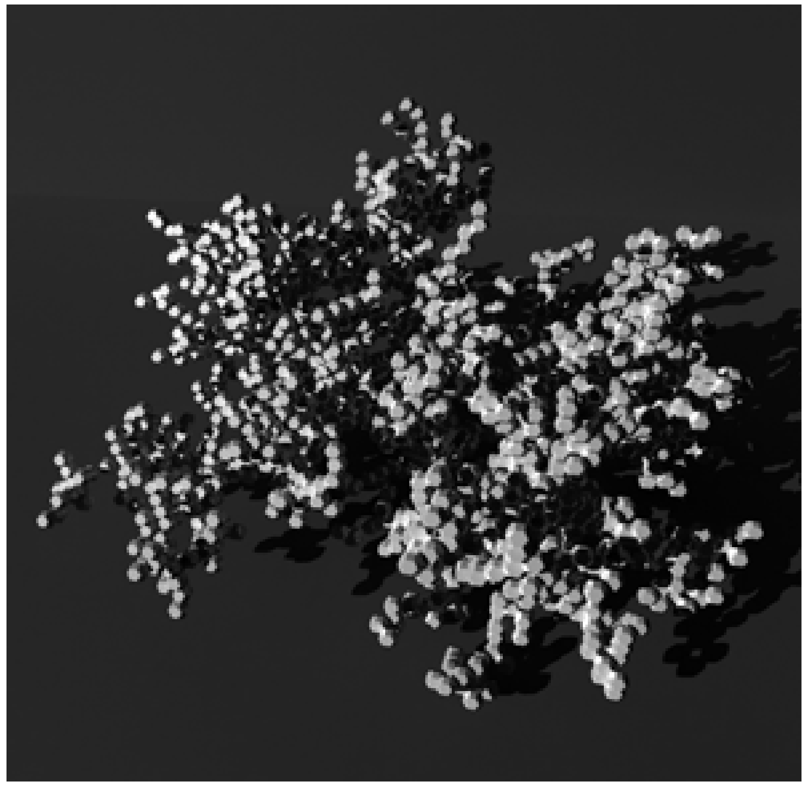
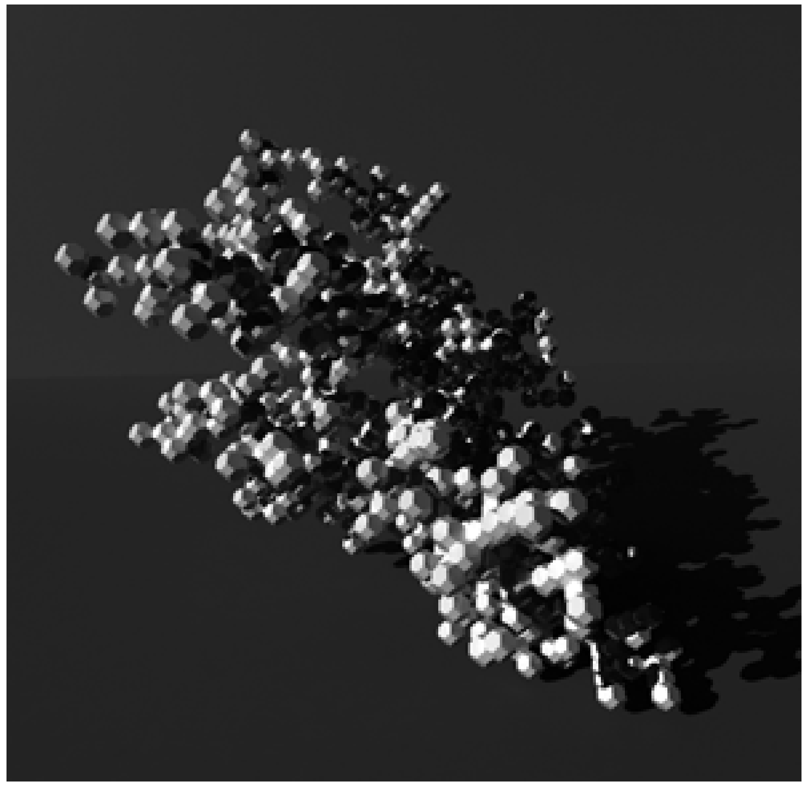


Figure 3: The growth is constrained to only add components when they can shield a defined point against sun radiation. Since the sun does not exceed a certain angle, the growth stops when it reaches this invisible plane.



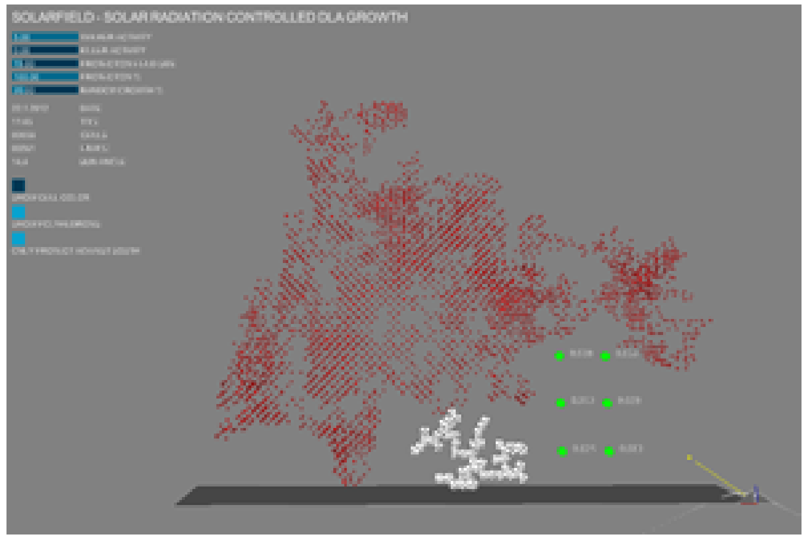
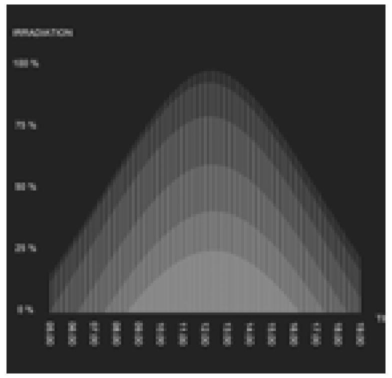


Figure 4: DLA generated growth. The read area describes the path of the random particle in search for a position for the next cell. When it touches the existing structure, the cell can be attached.

8.6.4 Organisational logic, digital growth

As part of the initiating workshop, a series of different algorithmic methods were developed and discussed. The thesis outlines only one method, which has also subsequently been further developed. This method is based on a generic DLA algorithm, implemented by David Andreen for the workshop. The algorithm is based on the DLA model, developed by Witten and Sander. By using a basic DLA model, the idea was to gain the basic open, branching character known from DLA, and to simultaneously be able to incorporate additional specific growth-controlling parameters. Having the scope of the experiment in consideration, it was decided to work with only one additional parameter, namely sun radiation. By linking the form generation with an environmental parameter, the method could be extended with a type of contextual relation, or even a form of optimisation. Moreover, it would be possible to establish some form of negotiation between the inner logic of the algorithm and the optimisation parameters. The virtual environment for the DLA growth was defined as a three-dimensional lattice. This made it possible to record the position of the cells both as a position vector and as an index in the three-dimensional array of points in the grid. The *random particle* that finds the random growth directions by randomly searching through the grid does not need to analyse the position of all cells in the existing structure, but only needs to test if the current *random particle* position and its immediate neighbours are occupied by a cell.

The solar radiation simulation was set up in a way, where the challenge is to protect a number of precisely identified points, later referred to as attractor points, from sun radiation. However, the goal was not to provide a solution for a specific problem, but rather to investigate possible implications of embedding a simulation



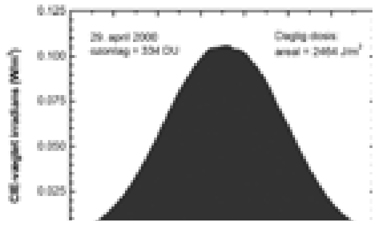


Figure 5: The intensity of the irradiation was in the simulation directly linked to the angle of the sun. Top: Relative irradiation levels , used in the simulation. Note that the diagram shows the months from January to June. Above: Irradiation diagram for Copenhagen in April.

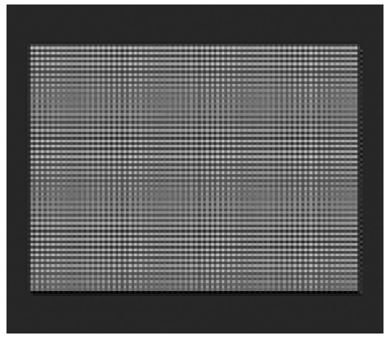


Figure 6: The lattice underlying the DLA growth. The spatial lattice was a grid with a depth of 6 layers of cells. The image shows a wall with 100% density.

method in the form generating process. Eventually it is possible that the system could be used for generating sun protection as part of a façade system or perhaps for covering an outside space. In this case it would typically be relevant to specify a whole array of points that needs protection. A property of the system would then be its capability of adapt to a large variety of both local conditions in the structure and differentiation due to orientation. A couple of illustrations indicate the use in architecture. The calculation of sun angles is based on a Java library implemented by Klaus Brunner[2], and uses latitude, longitude, date and time as input parameters. The irradiance value is then calculated directly from the sun angle, meaning that the higher the inclination of the sun is, the higher irradiation value. Despite the simplicity of the irradiance calculation, it seems sufficient for the method. An example of a precise calculation for Copenhagen in April is shown, and when compared to the diagram used in the simulation, there is a similarity in the relative increase in irradiance during the day. The irradiance values in the simulation were not quantified. In the cases where the simulation was used for generating a sun protection wall, the sun angle in plan also has importance. In the experiment, the sun protection wall was facing south. This means that in positions, where the sun was not precisely positioned in south, the radiation hitting the façade was reduced, and therefore the orientation was part of the calculation, similar to the sun inclination.

The cell growth was controlled by adjustment of some essential variables. Mainly it was important to be able to control the balance between random growth and solar protection. When the solar protection setting was set to zero, the growth was fully controlled by the DLA logic. When the solar protection setting was increased, the growth was intensified in areas where the protective values for the cells were calculated as high. If the solar protection setting was set to 100%, and random growth was set to 0, it meant that only cells with a protective value above zero could be added to the structure. The question was whether it was possible to generate a compromise between inner logic and solar protection, and it seemed that a setting where solar protection was set to 20% resulted in a formation still appearing as a DLA structure, but also with increased solar protection properties.

The growth process can be described as follows: The 'random particle' performs its search through the lattice until it 'bumps into' the existing cells. The chance of a new cell being added at the position is a stochastic decision, where the essential

---

2. The Java library used for calculating the sun angle is developed by Klaus A. Brunner, and based on the paper: Blanco-Muriel et al., 'Computing the Solar Vector', *Solar Energy Vol 70 No 5,* pages 431-441.

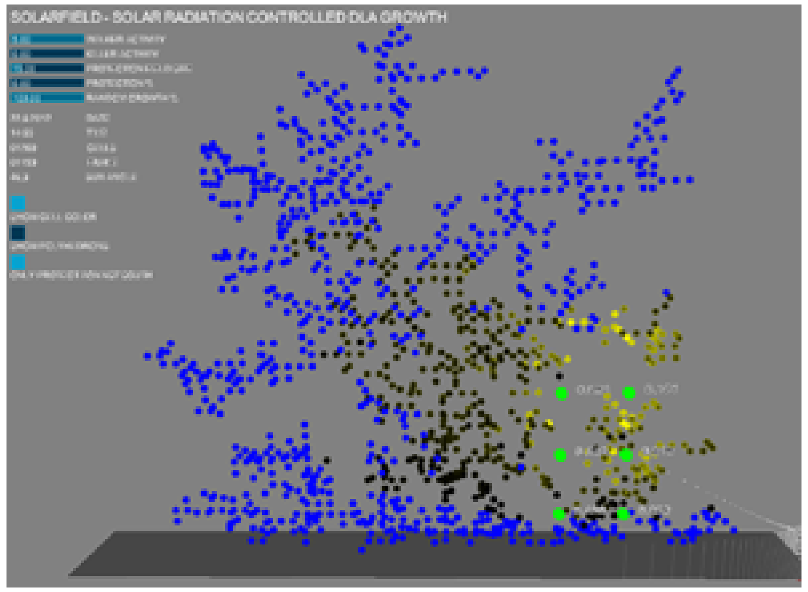


Figure 7: Generation of solar protection wall. The solar protection parameter is set to 0, which results in a pattern, completely reflecting the DLA growth logic. Still, the attractor points are protected from solar radiation to some extent. The protection percentages range from 16 to 94.

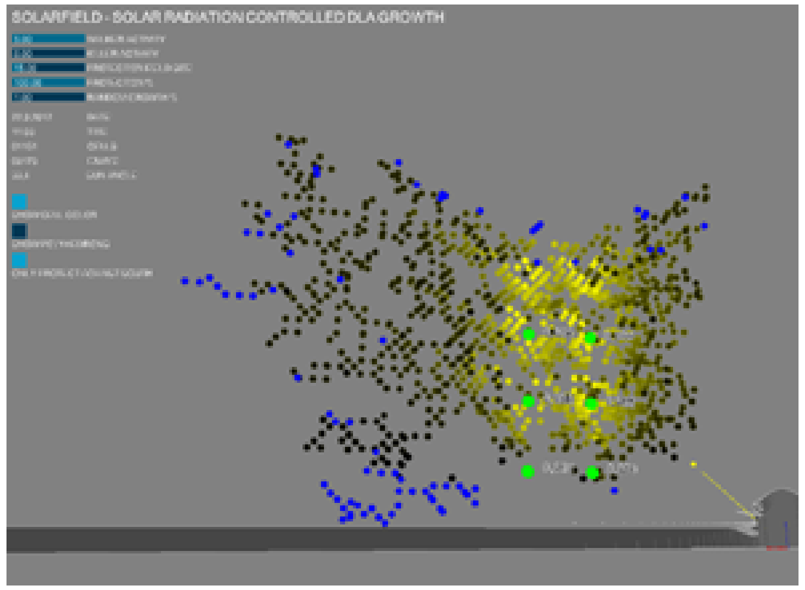


Figure 8: Generation of solar protection wall. The solar protection parameter is set to 100%, and random growth is set to 1%. This results in a pattern entirely formed to protect the attractor points. DLA is still the driving principle, but the character of the pattern is blurred. The protection ranges from 80% to 97%.

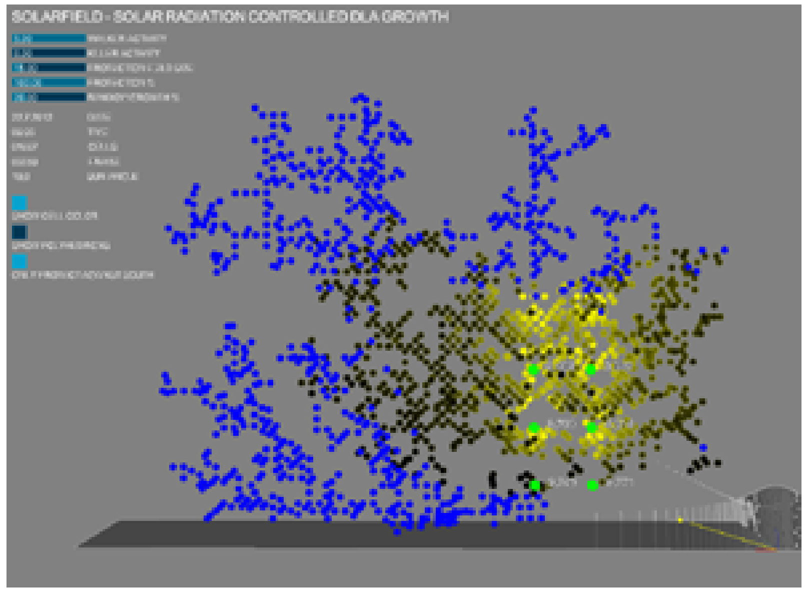


Figure 9: Generation of solar protection wall. The solar protection parameter is set to 100%, and random growth is set to 20%. The pattern still appears as a characteristic DLA pattern. The attractor points are protected from solar radiation with values ranging from 71% to 94%.

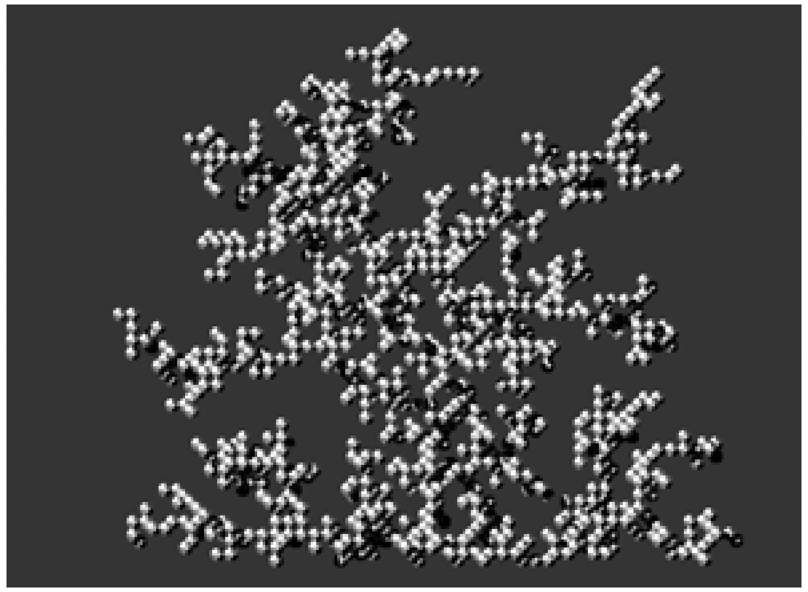


Figure 10: Rendering of the DLA solar protection pattern generated from 100% random growth.

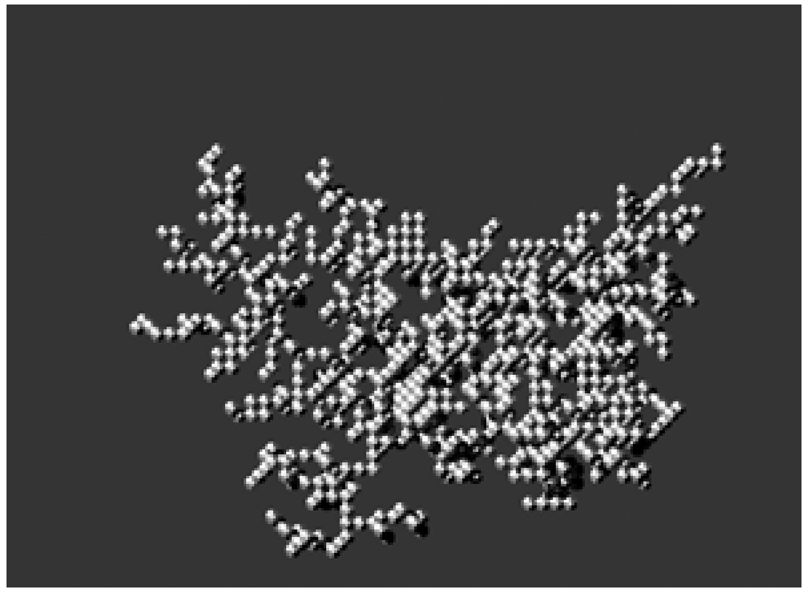


Figure 11: Rendering of the DLA solar protection pattern generated from 1% random growth and 100% tendency for solar protection.

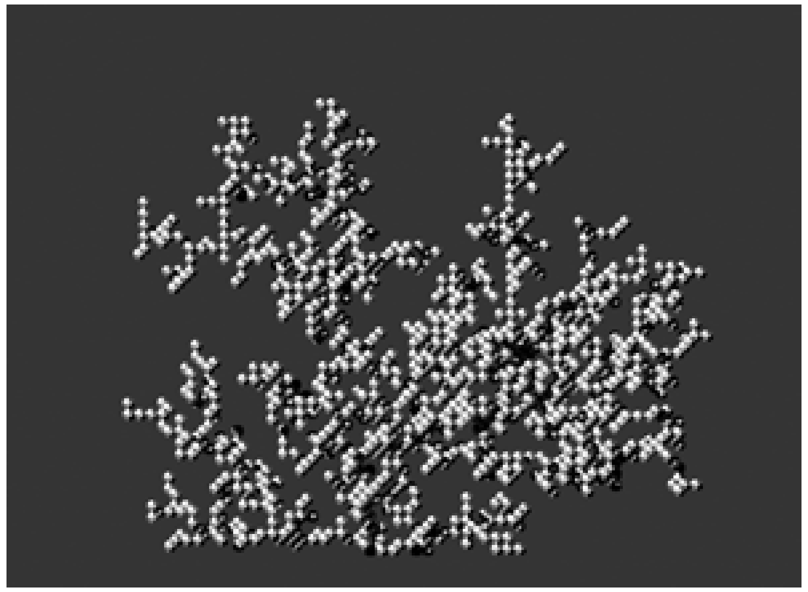


Figure 12: Rendering of the DLA solar protection pattern generated from 20% random growth and 100% tendency for solar protection.

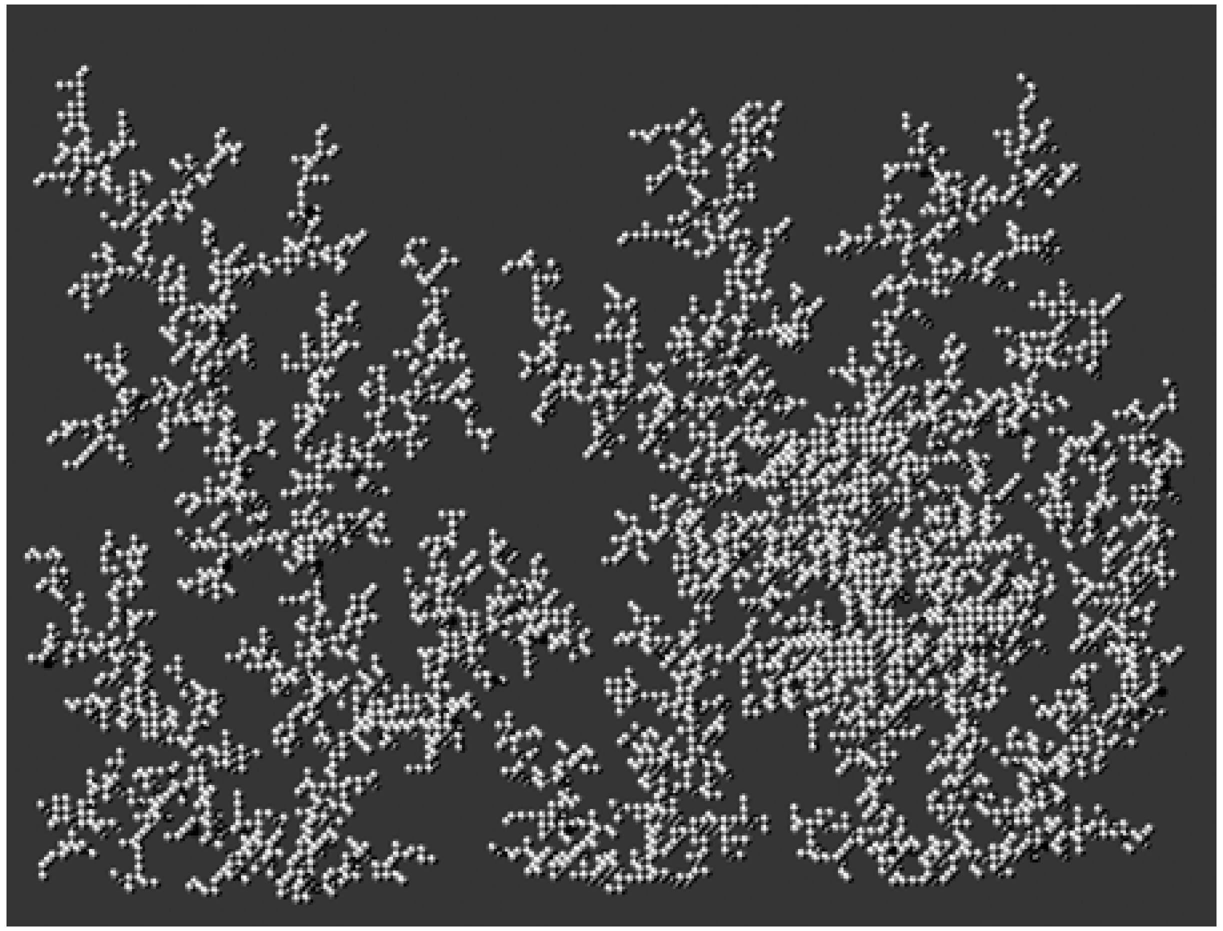
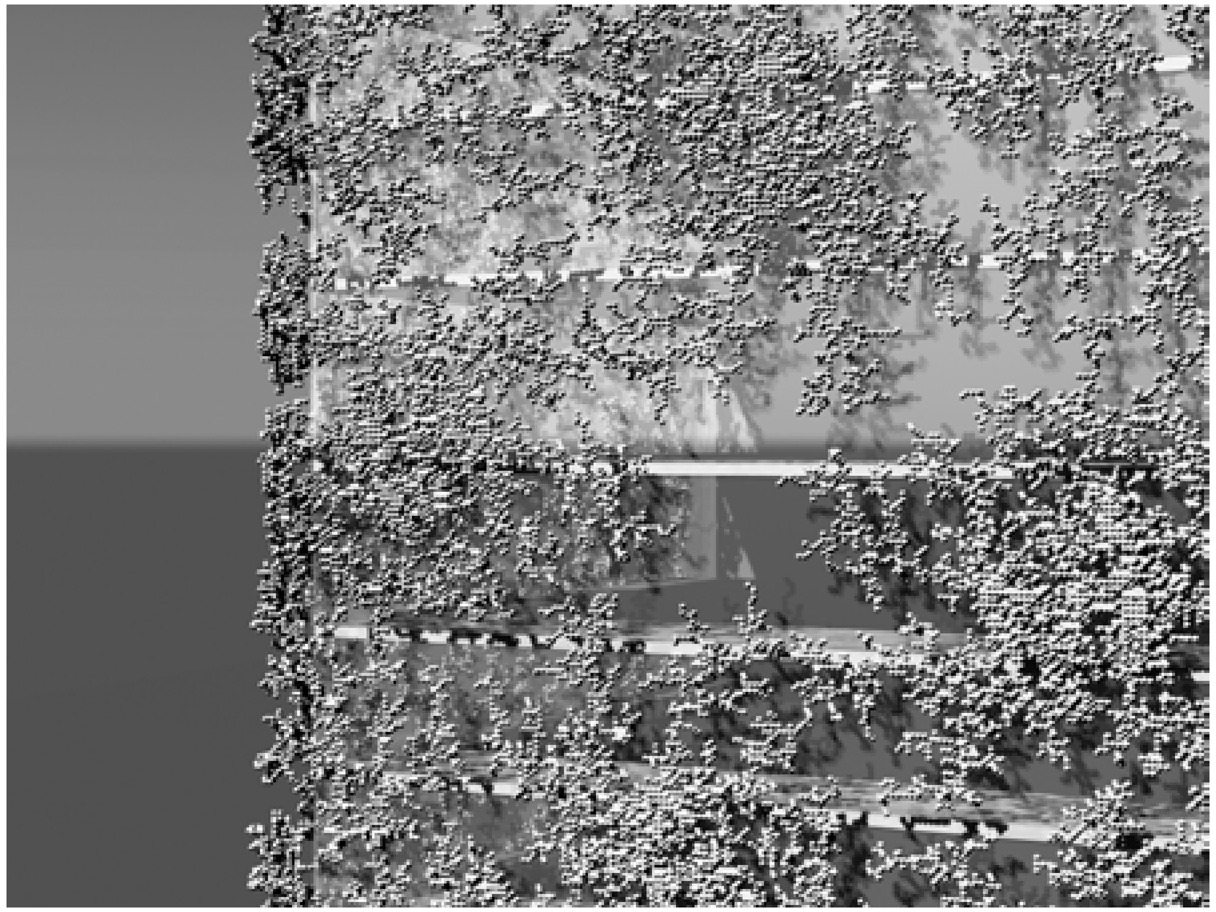


Figure 13: Rendering of the DLA solar protection pattern generated from 20% random growth and 100% tendency for solar protection.

parameters define the probabilities. It can be compared to the adjustment of stickiness in the generic DLA algorithm, but in this case the probabilities depend on the protective values of the cells. If the cell is added, the protective value of the cell is calculated. A simulation is run, where a number of sun positions with defined intervals are tested throughout a period. In this case, the simulation covers a year, where one day in each month is calculated with hours from 5.00 to 19.00, divided into intervals of 5 minutes. The days that are not included are in some cases readable in the growth, but this is countered in adjustments in the settings. Each cell has a protection value, reflecting the accumulated irradiance values, that the cell can prevent from hitting the attractor points during the simulation period, in this case a year. If the cell is placed in a position where it can block sunrays hitting one of the attractor points with high irradiance value at some time during the year, the cell gets a high protection value, and is showed with a bright yellow colour. If it does not block any rays it gets a protection value equal to zero, and is shown with a blue colour. If the cell only blocks sunrays with low irradiation value, the protection value is low, and the cell is shown with a graded dark yellow.

In order to address an architectural context more directly it was decided to produce examples, where the solar protection



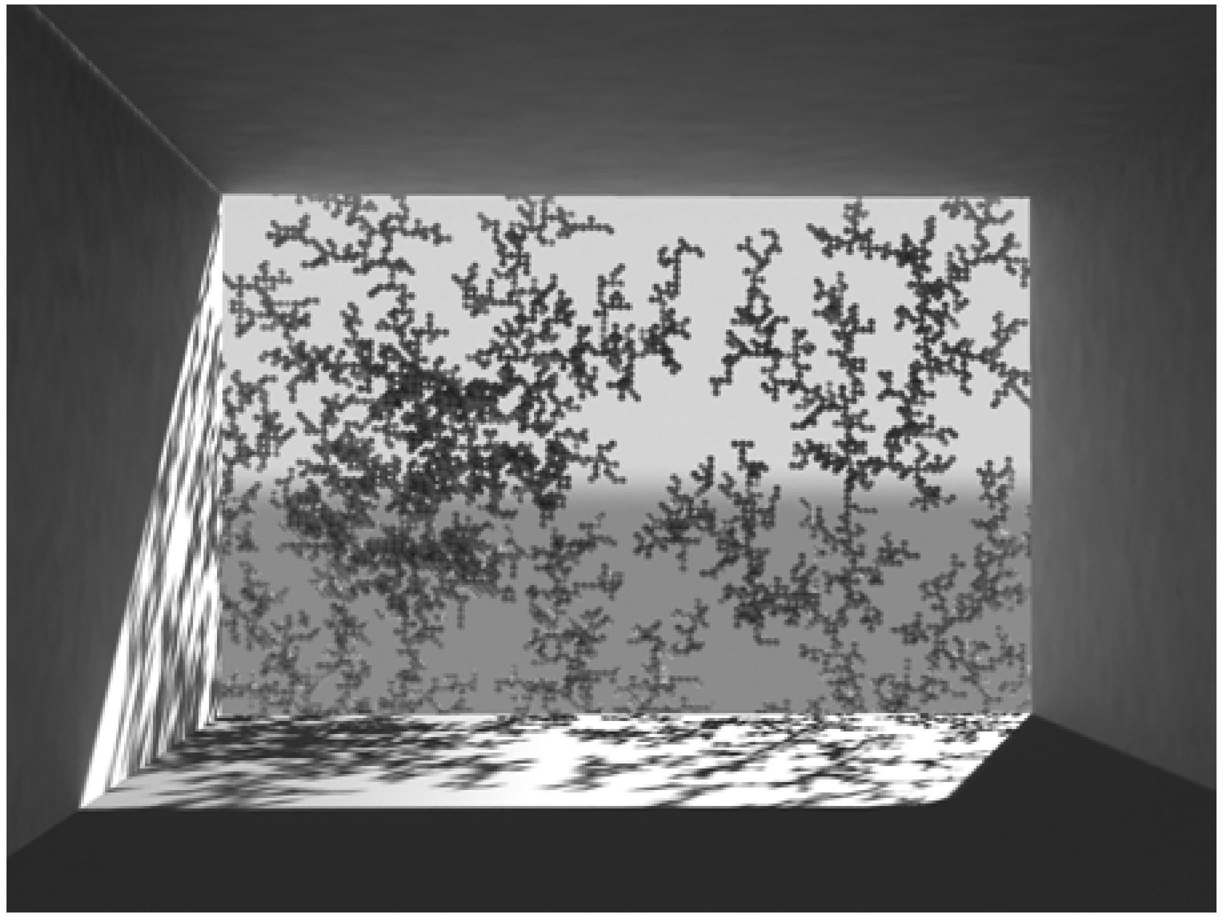


Figure 14: Agent Construction. Physical structure after completed growth process

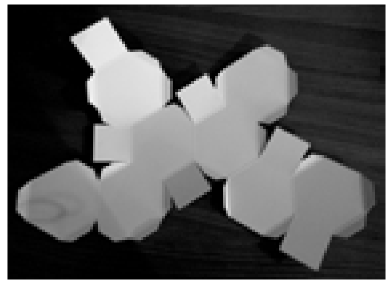
Figure 15: Opposite page. Top: Rendering of facade with sun protection structure, generated with Solar DLA System. Bottom: Rendering of an interior space.

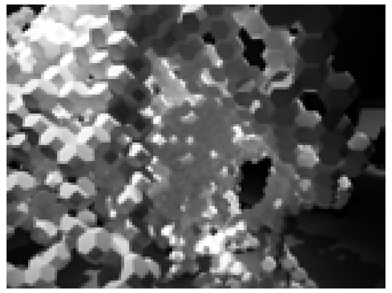
structure is organised as a wall. As described in the general explanation of the DLA model, the system can work, both in two and three dimensions, so the example could have been developed as a two-dimensional pattern. However, when developing the sun-protection wall method, the idea was to establish a formation of three-dimensional appearance and to use the spatial depth as part of the protection strategy.

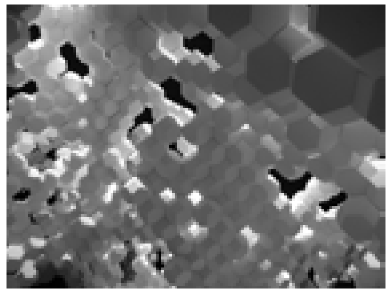
8.6.5 Realisation, the physical model

The physical system consisted of 5000 components assembled by approximately 15 persons with differentiated purposes in terms of aims for the structure. The components, formed as polyhedrons, were made from folded and glued cardboard templates. The material was mainly chosen for economic and practical reasons, but it enabled both addition and subtraction of components during construction, since the components were quickly connected with glue or disconnected by pulling them apart. The participants, or agents, affected the building process by adding or subtracting components, or cells, during the growth process. Each person was instructed to aim for a specific type of performance goal, such as structural strength, division of space, airflow or light conditions. The positions of the cells were digitally measured during the construction process, in order to be able to use digital simulation tools to analyse the performance of the structure. Part of the idea was to use the analysis as feedback affecting the building process. The agents were allowed to obtain additional goals during the process, depending on the development

Figure 16: Agent Construction: Unfolded polyhedron. Lowest: Builders discussing during construction.







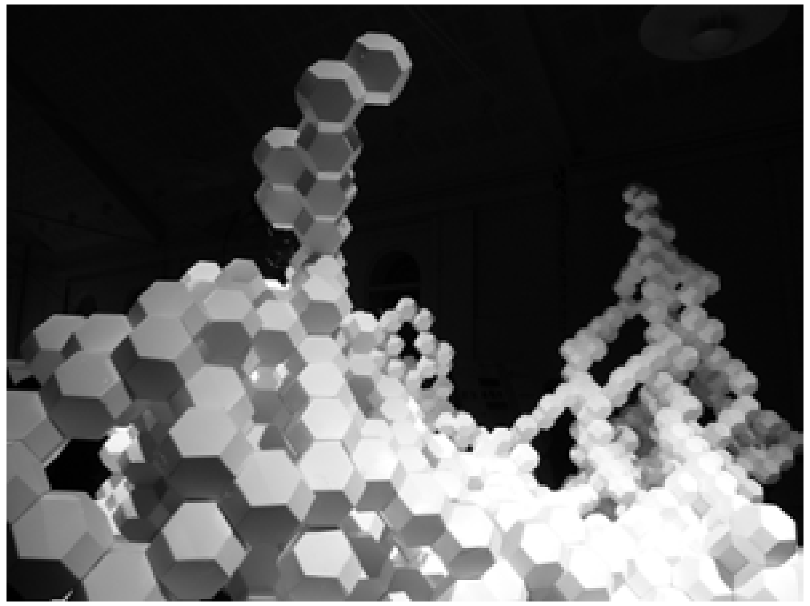


Figure 17: Agent Construction model. Surface formations and growth patterns were formed during the construction process.

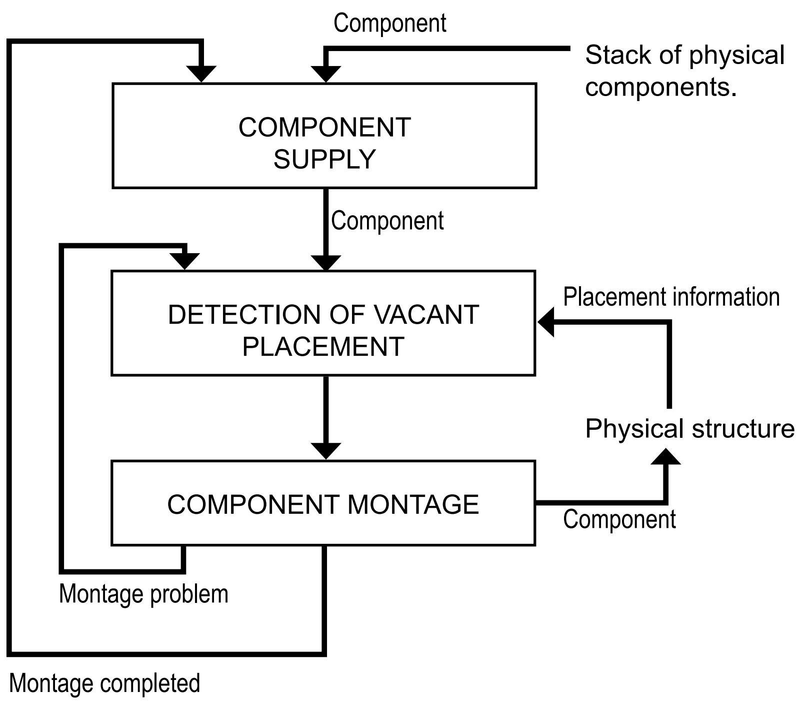


Figure 18: Agent Construction. States of the building process. The builders could decide which growth possibilities were effectuated.

of the structure. This could imply agents detecting a common goal, besides the predefined, from potentials occurring during the building process. For instance, as the structure emerged, it quickly became clear, that some agents worked for particular formations to happen, such as bridging to different parts of the structure, or making the structure reach out for a particular spot in the surroundings.

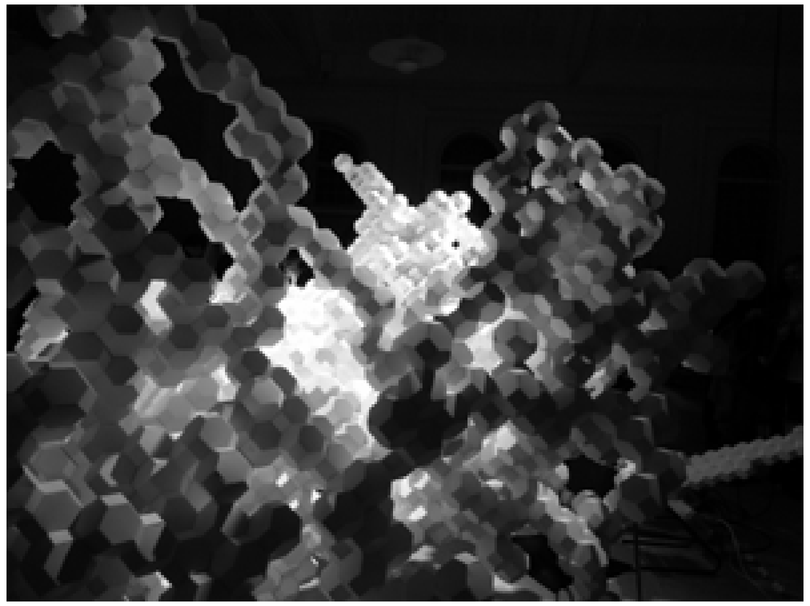
8.6.6 Observation, comparison

In terms of studying the phenomenon of *stigmergy*, the physical provided useful experience. Most input for the agents during construction was from the formation of the structure itself. It was primarily information embedded in the structure that guided the growth. However, the physical structure grew only partly based on local rules. This was due to the fact that the agents all the time during the building process were studying and discussing the development of the structure, and could take on individual goals. These additional goals were generally on a global level, reversing to a top-down approach. This could be to finish a type of bridge part or to establish a fully enclosing surface. Since these goals in some cases were reminiscent of patterns recognised in the termite mounds, it became difficult to separate the top-down guided growth parts from the more bottom-up controlled parts. So, in terms of analysing the result of rule-based growth the physical model did not reveal any clear conclusions. Furthermore, the conditions set for the agent behaviour were not strong enough to guide the actions. The most successful behavioural goal was the structural aspect, which was inevitable, even though the components were light. The spatial division easily became a top-down goal, where the light conditions in the environment were undefined and since the construction was placed in an interior space, the wind conditions were a difficult parameter.

Testing the potential of the principle in a new physical context, would benefit from a more rigid definition of the rules for growth and clearer measures for positive and negative consequences of placing each component, seen in the light of the separate goals, that the agents are predestined to follow. Also, it would help the experiment to define specific goals for the construction as a whole, even though these goals should be hidden from the constructing agents. This would establish a situation more similar to, for instance, a termite mound, where the overall goal is the survival of the colony, but the individual termites only work from a simple set of behavioural rules, not 'knowing' anything about the general goal.

Having pointed out the lack of bottom-up approach in the physical realisation process, it could be said that the digital method, described earlier, only deals with a very limited set of goals.

Figure 19: Agent Construction model. Patterns emerged during construction.



Compared with a termite mound, the experiment is immensely less complex. Still, it displays some essential potential in using these types of self-organisation in architecture. The process of negotiation between inner algorithmic logic and performance criteria is one of the more fundamental qualities provided by generative techniques. Some of the limitations in the example are listed in the following points:

1. The tectonic simplicity. In terms of tectonic principle, the component could be described as either extremely abstract or as very simple. Also, the underlying lattice, which ensures that all components can join, results in a rigid system. The system could perhaps embed differentiation in terms of local scaling, but to gain the full potential of the generative system, a more versatile geometric principle could be developed. Perhaps a combination, where the lattice is understood as a freeform surface, where the growth takes place, could be possible.
2. Sun radiation as single performance parameter. A new generation of the tool would start to adopt other types of parameters, such as structure or spatial experience. An obvious next step would be to set up zones where the façade system strives towards openness, for instance if the inhabitants need view to the outside.
3. Limited implementation of feedback systems. In the current tool, only the implemented sun position calculation affects the generative process. It would be a logic step to link the process with other types of analytic tools, such as climatic analysis.

However, the experiment did document some important potential in use of generative techniques. The DLA principle appeared to be a useful starting point for certain types of self-organising tectonics. Its basic simplicity allowed great variation in the growth process, due to additional parameters. It was possible establish a negotiation between the embedded organisational logic and the external conditions, in this case the sun radiation. This allowed simultaneous appearance of the algorithmic pattern and performance improvement.

References

Allen, S, 'From object to field', *Architectural Design, vol. 67, no. 5/6, 1997, pages 24–31.*

Andersson, IK & PH Kirkegaard, 'A discussion of the term digital tectonics', *WIT Transactions on The Built Environment, Vol 90, 2006*, *pages 29-39.*

Armstrong, R, 'Soil and protoplasm', *Architectural Design vol. 81, issue 2,* Wiley, London. 2011.

Arthur, WB, JH Holland, B LeBaron, R Palmer, and P Tayler. 'Asset pricing under endogenous expectations in an artificial stock market', *The economy as an evolving, complex system II*, Addison Wesley, Redwood City, 1997, pages 15-44.

Ball, P, *Shapes*, Oxford University Press, Oxford, 2009, page 147.

Balmond, C, 'Geometry, algorithm, pattern: The Serpentine Pavilion 2002 - Toyo Ito and Arup' in N Leach, D Turnbull & C Williams, *Digital Tectonics*, Wiley-Academy, West-Sussex, 2004, page 129.

Balmond, C, 'Informal Agency' in Leach, N, and R Snooks (eds), *Swarm intelligence: architectures of multi-agent systems*, Liaoning Science and Technology Publishing House, Liaoning, 2010, 121, 125.

Baraona Pohl, EB, *WaterCube: The Book*, dpr editorial, Barcelona, 2008, page 190

Bernstein, PG, 'Models for Practice: Past, Present, Future', *Building (in) the Future*, Princeton Architectural Press, 2010, page 194

Blanco-Muriel et al., 'Computing the Solar Vector', *Solar Energy Vol 70 No 5,* pages 431-441

Block, P, M DeJong and JA Ochsendorf, 'As Hangs the Flexible Line: The Mechanics of Masonry Arches', *Nexus Network Journal on Architecture and Mathematics Vol. 8, No. 2*, 2006, pages 9-18.

Bonabeau, E, 'From classical models of morphogenesis to agent-based models of pattern formation', *Artificial Life archive Volume 3 Issue 3*, MIT Press Cambridge, MA 1997, pages 191-211

Bonabeau, E, S Camazine, J Deneubourg, NR Franks, J Sneyd & G Theraulaz, *Self-Organisation in Biological Systems*, Princeton University Press, USA, 2001, page 8, 31, 56, 491.

Bourke, P, 'DLA – diffusiont limited aggregation', 1991, updated 2004, viewed 7 January 2012, <*http://paulbourke.net/fractals/dla/*>

Bourke, P, 'Polygonising a scalar field', 1994, viewed 15 April 2012, <*http://paulbourke.net/geometry/polygonise/*>

Burkhart (ed.), B, *IL 13 Multihalle Mannheim*, Institut für leichte Flächentragwerke, Germany

Chan, E, statement in *Digital Project, Frank Gehry's Vision*, video, Dansk Arkitektur Center, Copenhagen, 2007.

Coates, P, *Programming Architecture,* Routledge, Oxon, 2010, page 13.

DeLanda, M, 'Material Elegance' in A Rahim H Jamelle, *Architectural Design, vol. 77, no. 1, 2007*, page 22.

Farshid, M & M Kubo, *The function of ornament*, Harvard Graduate School of Design, Actar, New York, 1996

Fountain, H, 'A Problem of Bubbles Frames an Olympic Design', 2008, viewed 4 February 2012, <*http://www.nytimes.com/2008/08/05/sports/olympics/05swim.html*>

Frampton, K, *Studies in tectonic culture*, The MIT Press, Cambridge, Massachusetts, 1995, pages 20, 87.

Frampton, K, 'Intention, Craft and Rationality' in Peggy Deamer and Phillip G. Bernstein (eds), *Building (in) the Future*, Princeton Architectural Press, New york, 2010, page 33, 35.

Galanter, P, 'What is generative art? Complexity theory as a context for art theory', *Proceedings of the 6th international conference: Generative Art 2003,* Milan, 2003

Hensel, M, A Menges & M Weinstock, *Emergence: Morphogenetic Design Strategies*. Wiley-Academy, London, 2004, page 7.

Hildenbrandta, H, C Carereb and CK Hemelrijka, *Behavioral Ecology vol. 21, no. 6, 2010*, pages 1349–1359.

Holland, J, *Hidden order: how adaption builds complexity*, Helix Books, New York, 1995, pages 10–37, 11.

Holland, J, *Emergence: from chaos to order*, Oxford University Press, Oxford, 1998, pages 4, 5.

'Hooke's law', Encyclopædia Britannica Online, Encyclopædia Britannica Inc., viewed 14 Dec. 2011.

IFR Statistical Department, '2011 Executive Summary', *World Robotics - Industrial Robots 2011*, viewed 2 February 2012, <http://www.worldrobotics.org/uploads/media/2011_Executive_Summary.pdf>

Janmohamed, H, Digital Project - Frank Gehry's Vision', video, exhibition at Dansk Arkitektur Center, 2007, retrieved 3 March 2012, <http://www.arcspace.com/architects/gehry/dp/dp.html>

Jones, J, 'Characteristics of Pattern Formation and Evolution in Approximations of Physarum Transport Networks', *Artificial Life, Spring 2010, Vol. 16, No. 2* , pages 127-153.

Kelvin, Lord (W Thomson), *Philosophical Magazine, Vol. 24, No. 151,* 1887, page 503

Kolarevic, B (ed.), *Architecture in the Digital Age: Design and. Manufacturing*, Spon Press, 2003, page 13

Lachauer, L, M Rippmann, P Block, 'Form Finding to Fabrication: A digital design process for masonry vaults', *Proceedings of the International Association for Shell and Spatial Structures Symposium 2010*, Shanghai

Lane, T, '0-14 tower, Dubai: The hole story', *Building.co.uk,* viewec 3 April 2012, *February, 2009*.

Langner, J, in B Burkhart (ed.), IL 13, op. cit., page 51

Larsen, NM, OE Pedersen & D Pigram, 'A method for the Realisation of Complex Concrete Gridshell Stuctures in Pre-cast Concrete', manuscript submitted for publication, ACADIA, 2012.

Leach, N, D Turnbull & C Williams (eds), *Digital Tectonics*. Wiley-Academy, West-Sussex, 2004

Lees, M, B Logan and J King, HLA Simulation of Agent-Based Bacterial Models, *Proceedings of the 2007 European Simulation Interoperability Workshop*

Lynn, G, *Animate Form*, Princeton Architectural Press, New York, 1999, page 11, 16, 40.

Lynn, G, 'Probable geometries: The architecture of writing in bodies', *Any 0*, 1993

Mallgrave, HF, 'Introduction' in HF Mallgrave & W Herrmann (eds, trans.), *The four elements of architecture and other writings,* Cambridge University Press, New York, 1989, pages 19, 36.

McDonough, W, Baumgart, M, *Cradle to Cradle: Remaking the Way We Make Things*, North Point Press, New York, 2002

Moussaïd, M, D Helbing and G Theraulaz, 'How simple rules determine pedestrian behaviour and crowd disasters', *Proceedings of the National Academy of Sciences, Volume 108, No. 17, 2011*, pages 6884-8.

Mutschler, C, in B Burkhart (ed.), IL 13, op. cit., page 28

Otto, F, in B Burkhart (ed.), IL 13, op. cit., pages 11,13

Otto, F, B Rasch, *Finding Form: Towards an Architecture of the Minimal*, Edition Axel Menges, 1995, page 58, 140.

Picon, A, *Digital Culture in Architecture*, Birkhäuser, Basel, 2010

Picon, A, 'Architecture and the virtual: towards a new materiality', *Praxis 6*, pages 114–121.

Piker, D, 'Metaballs in Kangaroo', 2011, viewed 2 May 2012, *<http://www.grasshopper3d.com/profiles/blogs/metaballs-in-kangaroo>*

Prusinkiewicz,P & A Lindenmayer, *The Algorithmic Beauty of Plants*, Springer-Verlag, New York, 1990

Reynolds, CW, 'Flocks, herds, and schools: a distributed behavioral model', *Computer Graphics, 21(4), July 1987*, pages 25-34.

Rogers, P, 'Welcome to WaterCube, the experiment that thinks it's a swimming pool', 2004, viewed February 4 2012, *<www.guardian.co.uk/science/2004/may/06/research.science1>*

Schumacher, P, 'Parametric Patterns', *Architectural Design – Patterns of Architecture, Vol 79, No 6*, 2009

Semper, G, 'Style in the technical and tectonic arts or practical aesthetics', in HF Mallgrave & W Herrmann (eds, trans.), *The four elements of architecture and other writings,* Cambridge University Press, New York, 1989, pages 183, 198.

Seymour, M, '*The Science of Fluid Sims'*, 2011, viewed 3 June 2012, *<http://www.fxguide.com/featured/the-science-of-fluid-sims>*

Snooks, R, '*Swarm matter',* viewed 12 April 2012, <www.kokkugia.com, 2009>

Solakian, N, 'Editor's Pick: Swiss cheese-laced exoskeleton', viewed 15 May 2012, *<http://www.architizer.com/en_us/blog/dyn/15558/editors-pick-swiss-cheese-laced-exoskeleton/>*

Stuart-Smith, R, 'Formation and Polyvalence: The Self-Organisation of Architectural Matter', *Proceedings, Ambience 11*, Borås, Sweden, 2011

Tennant, R, 'Medieval Islamic architecture, quasicrystals, and Penrose and girih tiles: Questions from the classroom', *Symmetry: culture and science, vol. 19, no. 2/3*, 2008, pages 113-125.

Tenu, V, 'Minimal Surfaces as Self-organizing Systems', *Proceedings, ACADIA*, New york, 2010

Treece, GM, RW Prager & AH Gee, 'Regularised marching tetrahedra: improved iso-surface extraction', *Technical report CUED/F-INFENG/TR 333*, Cambridge University Department of Engineering, 1998.

Underwood, J & U Isikdag (eds), *Handbook of Research on Building Information Modeling and Construction Informatics: Concepts and Technologies*, IGI-Global, 2010

Weaire, D & S Hutzler, ‘Nonlinear phenomena in soap froth’, *Physica A, no. 257, 1998*, pages 264–274

Weiss, G, *Multiagent systems: a modern approach to distributed artificial intelligence*, MIT Press, 1999.

Weisstein, EW, ‘Hexagonal Close Packing’, *MathWorld--A Wolfram Web Resource*, viewed 1 May 2012, <*http://mathworld.wolfram.com/HexagonalClosePacking.html*>

Witten Jr., TA and LM Sander, ‘Diffusion-Limited Aggregation, a Kinetic Critical Phenomenon’, *Phys. Rev. Lett. 47*, The American Physical Society, 1981, pages 1400-3.

Wolf, TD & T Holvoet, ‘Emergence versus self-organisation: Different concepts but promising when combined’, *Engineering selforganising systems, vol. 3464, issue 1675*, 2005, pages 1–15.

Wolfram, Stephen, *A New Kind of Science,* Wolfram Media Inc, USA, 2002, page 28.

Illustrations

The author have made every effort to obtain proper credit information and permission to reproduce images that are not the work of the author. Copyright holders are welcome to contact the author with respect to correction.

1. Introduction

Figure 4: Herzog & De Meuron, ‘Ciudad del Flamenco, Unfolded Tower elevation’, *El Croquis 129/130*, 2004

Figure 5: Herzog & De Meuron, ‘Ciudad del Flamenco’, *Facade mockup*. *El Croquis 129/130*, 2004

Figure 6: Herzog & De Meuron, Campus Tree Village. ‘Unfolded Courtyard elevation’. *El Croquis 129/130*, 2004

2. Surface morphology

Figure 1: Poleni, Hanging chain principle, 1748. *Nexus Network Journal – vol. 8, no. 2, 2006*

Figures 2, 4–11: Images courtesy of Institut für Leichte Flächentragwerke, University of Stuttgart.

Figure 2: F Otto, soap film model, Frei Otto and Peter Stromeyer, the Garden Exhibition in Hamburg, 1963

Figure 3: NM Larsen,: Model for the Colonia Güll church, Museum der Sagrada Familia, 2011.

Figure 4: Multihalle Mannheim. *IL 13, 1978*

Figure 5: Multihalle Mannheim, suspended wire mesh, *IL 13,* 1978.

Figure 6: Multihalle Mannheim, *IL 13*, Suspended wire mesh, *IL 13,* 1978.

Figure 7: Multihalle Mannheim, plan drawing, *IL 13,* 1978.

Figure 8: Multihalle Mannheim, calculated mesh, *IL 13,* 1978.

Figure 9: Multihalle Mannheim, Scaffolds, IL 13, 1978.

Figure 10: Multihalle Mannheim, *IL 13,* 1978

Figure 11: Multihalle Mannheim, Flower exhibition Bundesgartenschau in 1975, *IL 13,* 1978, Entrance area.

Figure 12: NM Larsen: Concrete Gridshell Pavilion, 2010.

Figure 13: NM Larsen: Concrete Gridshell Pavilion. 2010.

Figure 14: NM Larsen: Concrete Gridshell Pavilion, 2010.

Figure 15: OE Pedersen: Complex Gridshell, 2010.

Figure 16: JLE Christensen: Complex Gridshell, 2010.

Figure 17: NM Larsen & OE Pedersen: Complex Gridshell 2010.

3. Pattern distribution

Figure 1: Eshel Ben-Jacob: Computer simulation of E. Coli bacteria.

Figure 2: NM Larsen: Self-organising agents, 2012.

Figure 3: NM Larsen: Cress seeds, 2008.

Figure 4: P Coates: Voronoi pattern. *Programming Architecture*, 2010

Figure 5: NM Larsen: Self-organising circles.

Figure 6: J Jones: Simulation of Physarum.

Figure 7: NM Larsen: Self-organising Bezier curves.

Figure 8: NM Larsen: Self-organising Bezier Curves.

Figure 9: NM Larsen: Alhambra Palace, 2007. Aperiodic Penrose tiling. 2011.

Figure 10: FOA: Ravensbourne College, London, 2010.

Figure 11: FOA: Spanish Pavillion. Expo 2005.

Figure 12: R Umemoto: 0-14 tower, Dubai, 2007.

Figure 13: C Balmond, Serpentine Pavilion, 2002.

Figure 14: P Schumacher: Zaha Hadid Architects, Civil Courts, Madrid, 2007

Figure 15: A Søndergaard: Digital Physical workshop, Aarhus 2011.

Figure 16: M Bartczak: Morphogenetic Studio, 2010

4. Topological negotiations

Figure 1: NM Larsen: 3D printed plaster model, 2010.

Figure 2: M Bartczac: Morphogenetic Studio, 2011.

Figure 3: M Bartczac: Morphogenetic Studio, 2011.

Figure 4: R Snooks: Swarm Matter. 2009.

Figure 5: R Stuart-Smith: Little Collins Baths, 2004.

Figure 6: OMA: Bibliotheque Nationale de Paris, 1989.

Figure 7: G Lynn, Yokohama Port Terminal, 1994. Uppermost: Laser-cut model of acrylic sheets. Lowest: Part of First-level plan.

Figure 8: NM Larsen, Branching Topologies, 2010.

Figure 9: NM Larsen, Branching Topologies, 2012.

Figure 10: NM Larsen, Branching Topologies, 2012.

5. Emergent hierarchies

Figure 1: BV Buren: Murmuration of starlings.

Figure 2: M Bülow: Complex Formations, 2009.

Figure 3: NM Larsen: Self-organising Surface. 2012.

Figure 4: NM Larsen: Complex Gridshell, 2011.

Figure 5: M Bartzak: Morphogenetic Studio, Aarhus, 2011.

Figure 6: M Bartzak: Morphogenetic Studio, Aarhus, 2011.

Figure 7: V Tenu: Periodic minimal surface, 2010

Figure 8: J Riiber: Lamella Flock. 2010.

Figure 9: R Snooks: Busan Opera House, 2011

Figure 10: R Snooks: BodySwarm research project, 2011.

Figure 11: R Snooks: AirBaltic Terminal, 2011.

Figure 12: NM Larsen: Self-organising Surface, 2012.

Figure 13: D. Martinaitis: Morphogenetic Studio, Aarhus, 2010.

6. Aggregate growth

Figure 1: PTW Architects: *Watercube, The Book, 2008*

Figure 2: PTW Architects: Competition design, 2003, *Watercube, The Book, 2008*

Figure 3: PTW Architects: The Weaire-Phelan space division principle, *Watercube, The Book, 2008*

Figure 4: L Kelvin: Minimal tetrakaidecahedron, 1887.

Figure 5: R Kempers: Natural soap film bubbles.

Figure 6: PTW Architects: Resin model. Watercube model, *Watercube, The Book, 2008*

Figure 7: PTW Architects: Diagram, *Watercube, The Book, 2008*

Figure 8: PTW Architects: Facade drawing, *Watercube, The Book, 2008*

Figure 9: PTW Architects: Section drawing. *Watercube, The Book, 2008*

Figure 10: C Bosse: Watercube, ETFE facade.

Figure 11: Witten & Sander: DLA model.

Figure 12: Pauls Lab: Dendrites of copper crystals.

Figure 13: P Bourke: DLA patterns

Figure 14: NM Larsen: Ice flowers, 2008.

Figure 15: R Soar: Cast of the interior of termite mound.

Figure 16: NM Larsen: Rendering of 14-sided polyhedrons.

Figure 17: NM Larsen: Solar DLA System. 2012.

Figure 18: NM Larsen: Agent Construction, Smart Geometry 2011.

Figure 19: NM Larsen: Solar DLA System, 2012.

Figure 20: Prusinkiewicz/Lindenmayer: DOL-system, *The Algorithmic Beauty of Plants*, 1990.

Figure 21: Prusinkiewicz/Lindenmayer: DOL-system, *The Algorithmic Beauty of Plants*, 1990.

Figure 22: Prusinkiewicz/Lindenmayer: DOL-systems, *The Algorithmic Beauty of Plants*, 1990.

Figure 23: Prusinkiewicz/Lindenmayer: Turtle interpretation, *The Algorithmic Beauty of Plants*, 1990.

Figure 24: NM Larsen, SAGA, 2009.

Figure 25: NM Larsen, SAGA, realised model, 2009.

8.1 Complex Gridshell

Figure 1: NM Larsen: Concrete Gridshell Pavilion, 2011.

Figure 2: NM Larsen: Complex Gridshell, diagram, 2011.

Figure 3: NM Larsen: Concrete Gridshell Pavilion, 2011.

Figure 4: Left: OE Pedersen, Complex Gridshell. Right: D Pigram: Patterns for laser cutting. 2011.

Figure 5: OE Pedersen: Complex Gridshell, diagram.

Figure 6: NM Larsen: Complex Gridshell, templates. 2011.

Figure 7: NM Larsen: Concrete Gridshell Pavilion, 2011.

Figure 8: NM Larsen: Concrete Gridshell Pavilion, 2011.

Figure 9: OE Pedersen: Case Study 1, 2011.

Figure 10: NM Larsen: Concrete Gridshell Pavilion, diagram, 2012.

Figure 11: NM Larsen: Concrete Gridshell Pavilion, 2011.

Figure 12: D Pigram: Concrete Gridshell Pavilion, plan drawings, 2012.

Figure 13: J Christensen: Concrete Gridshell Pavilion, structural analysis, 2011.

Figure 14: NM Larsen: Concrete Gridshell Pavilion, Montage, 2011.

Figure 15: NM Larsen: Concrete Gridshell Pavilion, exhibition, 2011.

Figure 16: OE Pedersen: Case Study 3, Copenhagen, 2012.

Figure 17: OE Pedersen: Case Study 3, Copenhagen, 2012.

8.2 Self-organising Bezier curves

Fig. 1-11: NM Larsen, Self-organising Bezier Curves, 2012.

Figure 12: NM Larsen, Miami Civic Center, 2010.

Fig. 13,14: NM Larsen, Self-organising Bezier Curves, 2012.

Figure 15: Roland: MDX-540

Figure 16: Top, middle: F Gramazio, robot workshop.

Figure 16: Lowest: Matter Design's Foam Tower.

8.3 Branching topologies

Figure 1: G Treece: Isosurfacing, Visible Human Project.

Fig. 2–28: NM Larsen, Branching Topologies, 2012.

Figure 29: Baumshlager and Eberle: Nordwesthaus, 2008.

Figure 30: A Søndergaard: Polystyrene, Robot, 2010.

Figure 31: A Søndergaard: Unikabeton, 2010.

Figure 32: B Khoshnevis, Contour Crafting.

Fig. 33–34: NM Larsen, 3D plaster print, 2010.

8.4 Agent-based formations

Figure 1: NM Larsen: Agent-based systems, behaviours, 2009.

Figure 2: NM Larsen: Agent-based systems, diagram, 2009.

Figure 3: NM Larsen: Agent-based systems, screen grab, 2009.

Figure 4: NM Larsen: Agent-based systems, behaviours, 2009.

Figure 5: NM Larsen: Agent-based systems, behaviours, 2009.

Figure 6: M Bülow: Complex Formations, Copenhagen, 2009.

Figure 7: M Bülow: Complex Formations, Copenhagen, 2009.

Figure 8: M Bülow: Complex Formations, Copenhagen, 2009.

Figure 9: NM Larsen: Agent-based systems, 2009.

Figure 10: NM Larsen: Agent-based systems, 2009.

Figure 11: NM Larsen: Agent-based systems, 2009.

Figure 12: NM Larsen: Agent-based systems, weaving, 2012.

Figure 13: NM Larsen: Agent-based systems, constraints, 2012.

Figure 14: NM Larsen: Miami Civic Center, preliminary investigations, 2010.

Figure 15: NM Larsen: Miami Civic Center, preliminary investigations, 2010.

Figure 16: NM Larsen: Miami Civic Center, preliminary investigations, 2010.

Figure 17: NM Larsen: Particle field, 2012.

Figure 18: NM Larsen: Particle field, 2012.

Figure 19: NM Larsen: Particle field, Voronoi geometry, 2012.

Figure 20: NM Larsen: Particle field, polyhedrons, 2012.

Figure 21: NM Larsen: Particle field, clusters, 2012.

Figure 22: NM Larsen: Attractor field, behaviours, 2012.

Figure 23: NM Larsen: Attractor field.

Figure 24: NM Larsen: Attractor field, volumetric geometry.

Figure 25: NM Larsen: Attractor field.

Figure 26: NM Larsen: Self-organising Surface, 2012.

Fig. 27–34: NM Larsen: Self-organising Surface, experiments.

Figure 35: Soapbubble.dk: Minimal surface of soap film. 2012.

Fig. 36,37: NM Larsen: Self-organising Surface, experiments.

Figure 38: Exploratorium: Spiralling minimal surface.

Fig. 38,39: NM Larsen: Self-organising Surface, experiments.

8.5 SAGA

Fig. 1–15: NM Larsen: SAGA, renderings, diagrams, photos.

8.6 Solar DLA system

Fig. 1–13: NM Larsen: Solar DLA System, renderings, diagrams, photos.

Figure 14: NM Larsen: Agent Construction, Smart Geometry, Copenhagen, 2011.

Figure 15: NM Larsen: Solar DLA System, renderings.

Figure 16: NM Larsen: Agent Construction, Smart Geometry, Copenhagen, 2011.

Figure 17: NM Larsen: Agent Construction, Smart Geometry, Copenhagen, 2011.

Figure 18: NM Larsen: Agent Construction, diagram, 2012.

Figure 19: NM Larsen: Agent Construction, Smart Geometry, Copenhagen, 2011.