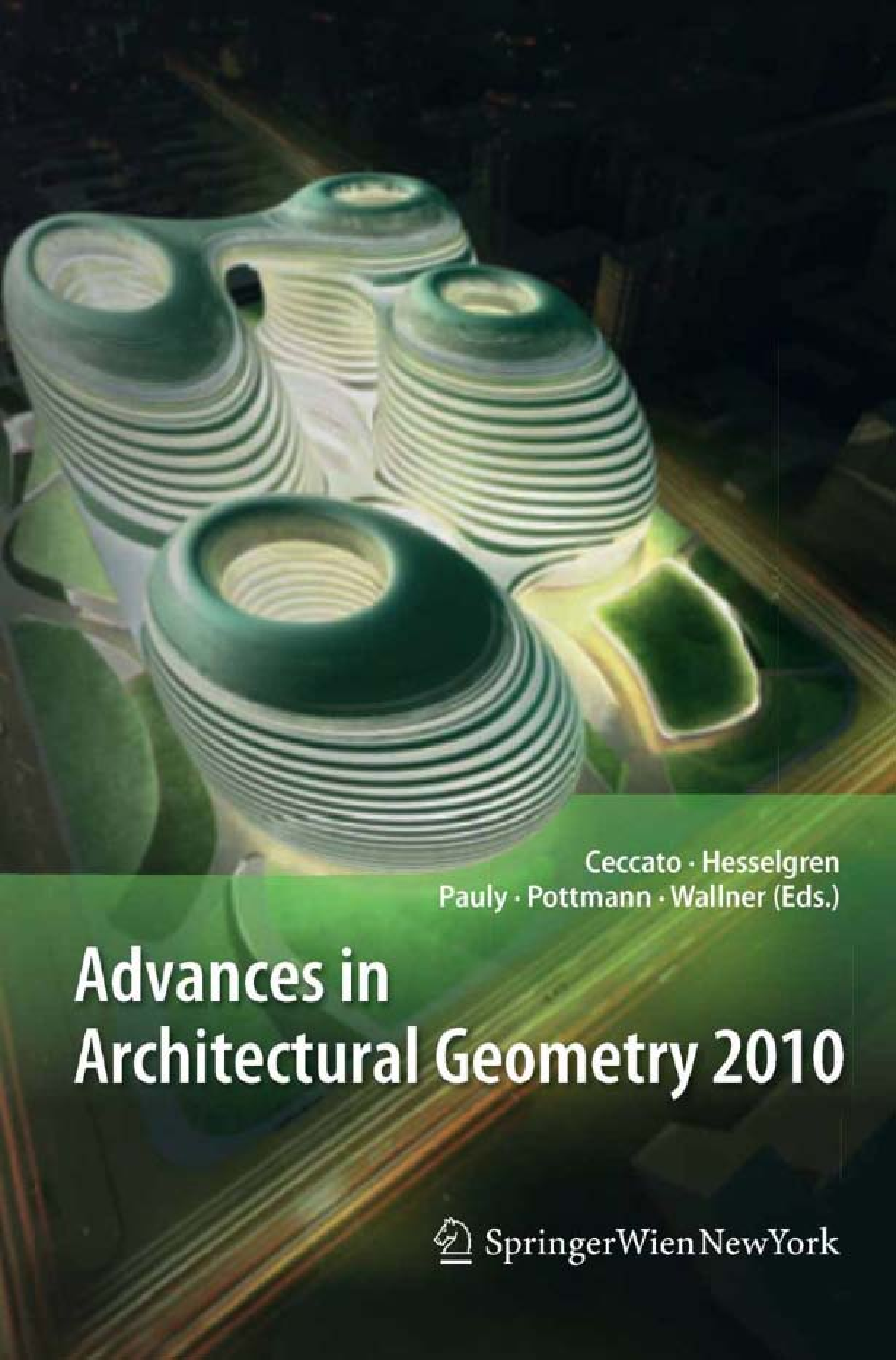


Ceccato · Hesselgren
Pauly · Pottmann · Wallner (Eds.)

# Advances in Architectural Geometry 2010

SpringerWienNewYork

SpringerWienNewYork

Cristiano Ceccato · Lars Hesselgren · Mark Pauly
Helmut Pottmann · Johannes Wallner
*Editors*

# Advances in Architectural Geometry 2010

SpringerWienNewYork

Editors
**Cristiano Ceccato** · Zaha Hadid Architects London, UK
**Lars Hesselgren** · PLP Architecture London, UK
**Mark Pauly** · EPFL Lausanne, CH
**Helmut Pottmann** · KAUST and TU Vienna, AUT
**Johannes Wallner** · TU Graz, AUT

Printed in Austria
SpringerWienNewYork is a part of Springer Science+Business Media
springer.at

Cover photo: Soho Galaxy project, Zaha Hadid Architects
Cover Design: WMXDesign GmbH, Heidelberg, Germany
Printing and binding: Holzhausen Druck GmbH, Vienna, Austria

Printed on acid-free and chlorine-free bleached paper

SPIN: 80013827
Library of Congress Control Number: 2010934109

With numerous colored figures

ISBN  978-3-7091-0308-1 SpringerWienNewYork

# Foreword

Geometry lies at the core of the architectural design process. It is omnipresent, from the initial form-finding to the final construction. Modern geometric computing provides a variety of tools for the efficient design, analysis, and manufacturing of complex shapes. On the one hand this opens up new horizons for architecture. On the other hand, the architectural context also poses new problems to geometry. Around these problems the research area of architectural geometry is emerging, situated at the border of applied geometry and architecture.

The conference on *Advances in Architectural Geometry* which takes place from September 18 to September 21, 2010 in Vienna brings together researchers from the fields of architecture and geometry to discuss recent advances in research and practice, and to identify and address the most challenging problems. We aim at connecting researchers from architectural and engineering practices, academia, and industry.

This book reflects the substantial progress in the field since the last event two years ago. For AAG 2010 we received 59 high-quality submissions out of which 15 have been selected for this volume based on the reviews of an international program committee. We would like to express our sincere gratitude to all reviewers!

The interdisciplinary nature of architectural geometry is reflected in the diversity of backgrounds of the contributing authors. Renowned architects, engineers, mathematicians, and computer scientists present novel research ideas and cutting-edge solutions at the interface of geometry processing and architectural design.

The papers by Bollinger et al., and Koren provide fascinating insights into the essential role of geometry in the design and realization of contemporary architectural projects. Cutler et al. leverage computer vision techniques to automatically computed consistent digital 3D models from rough architectural sketches. Tamke et al. demonstrate how methods of self-organization in a dynamic process model enable interactive design of interlinked freeform wood structures. Novel computational methods for rationalization of freeform surfaces with planar, straight, or curved panels are discussed in the papers of Schiftner et al., Wallner et al., and Eigensatz et al., These contributions demonstrate the power of global optimization algorithms to handle the complexities of the freeform paneling problem. New means of form finding are explored in the works of Ahlquist and Menges, illustrating the effectiveness of computer simulation and dynamic systems. The geometry of bonds are investigated by Brocato and Mondardini for the design of stone domes and by Bärtschi et al. for generation of complex brick walls assembled by an industrial robot. Other computational design methods include the use of flat-foldable freeform structures by Tachi, the integration of acoustics simulation into the architectural design process by Berkeron-Mirsky et al., and the interactive generation of parametric truss geometry by Lachauer and Kotnik. Finally, Bentscheff and Gengnagel investigate the integration of digital design tools into the academic discourse and teaching.

It remains to say thanks to our sponsors and to the European Community's 7th Framework Programme under grant agreement 230520 (*ARC*), which makes it possible to financially support young participants of the conference.

*Cristiano Ceccato, Lars Hesselgren, Mark Pauly* (scientific co-chairs)
*Alexander Schiftner, Helmut Pottmann, Johannes* Wallner (organizers)

# Contents

1. The Master-Builder-Geometer
   *Cristiano Ceccato* ........ 9
2. Interpreting Physical Sketches as Architectural Models
   *Barbara Cutler, Joshua Nasman* ........ 15
3. Lamella Flock
   *Martin Tamke, Jacob Rüber, Hauke Jungjohann et al.* ........ 37
4. Case Studies in Cost-Optimized Paneling of Architectural Freeform Surfaces
   *Michael Eigensatz, Mario Deuss, Alexander Schiftner et al.* ........ 49
5. Tiling Freeform Shapes With Straight Panels: Algorithmic Methods
   *Johannes Wallner, Alexander Schiftner, Martin Kilian et al.* ........ 73
6. Freeform Rigid-Foldable Structure Using Bidirectionally Flat-Foldable Planar Quadrilateral Mesh
   *Tomohiro Tachi* ........ 87
7. The Sphere Project – Negotiate Geometrical Representations From Design to Production
   *Klaus Bollinger, Manfred Grohmann, Oliver Tessmann* ........ 103
8. Towards Teaching Generative Design in Architecture
   *Ilija Bentscheff, Christoph Gengnagel* ........ 113
9. Architectural Acoustics for Practioners
   *William Bergeron-Mirsky, Jason Lim, John Gulliford et al.* ........ 129
10. Wiggled Brick Bond
    *Ralph Bärtschi, Michael Knauss, Tobias Bonwetsch et al.* ........ 137
11. Geometric Methods and Computational Mechanics for the design of Stone Domes Based on Abeille's Bond
    *Maurizio Brocato, Lucia Mondardini* ........ 149
12. Louvre Abu Dhabi 1/33 – Fabrication of a Large-Scale Physical Light-Test Model
    *Benjamin S. Koren* ........ 163
13. Ortho-Pictures: 3D Objects From Independent 2D Data Sets
    *Gershon Elber* ........ 175
14. Geometry of Structural Form
    *Lorenz Lachauer, Toni Kotnik* ........ 193
15. Realizing Formal and Funcional Complexity for Structurally Dynamic Systems in Rapid Computational Means
    *Sean Ahlquist, Achim Menges* ........ 205
16. Statics-Sensitive Layout of Planar Quadrilateral Meshes
    *Alexander Schiftner, Jonathan Balzer* ........ 221

*List of Contributors* ........ 237

# The Master-Builder-Geometer

**Cristiano Ceccato**

Zaha Hadid Architects

*Let no man destitute of geometry enter my doors* — Plato

*Practice without Theory is blind; Theory without Practice is sterile* — Confucius

The Advances in Architectural Geometry conference consciously brings together two Greek terms, ‘architect’ (ἀρχιτέκτων) and ‘geometry’ (γεωμετρία) and Plato’s famous quote, which was said to hang above his door, sets the theme of the conference in terms of intellectual rigour as Confucius’ does in terms of practical aspiration; each of these two concepts is fundamental to architecture and its practice. The term ‘advanced’ is included as a catalyst for re-examining an ancient relationship that is presently finding new interest in contemporary architectural discourse through the advent of ubiquitous computation, and a current fascination with natural, scientific and mathematical concepts and their potential for expression through built form.

Architecture and geometry have been interwoven for thousands of years. Throughout this time, the architect and the geometer have been one and the same person. The term ‘architect’ means *master-builder*, while ‘geometer’ can be translated as *Earth-measurer*. This *master-builder-geometer* had a comprehensive knowledge of his trade, from contemporary design rules of mathematics and geometry, to a profound understanding of its expression through material and structure, and of course considerable skill in financing, managing and constructing a complex structure, of which we find the highest expression in the work of the most masterful of these architect-geometers in their respective eras. As we consider what ‘architecture’ and ‘geometry’ imply for us today, it is worth remembering what their significance has been in the past.

The earliest scientists were physicians, astronomers and mathematicians. Plato, Euclid, Pythagoras, Archimedes, and on through the likes of Piero della Francesca, Leonardo da Vinci and Descartes provide a continuous and unbroken line of geometric knowledge that has been used by architects and builders throughout history. The acquisition of this knowledge, the achievement of *mastery*, was a necessary requisite for architects as unlike today, knowledge was held in the minds of a few, and ‘consultants’, as one might think of them today, did not exist. In other words, the integration of multiple domains of knowledge into a unified set of skills and mental abilities were fundamental to the successful *master-builder*, such that geometry and architecture were inextricably fused into a single body of knowledge.

A distant point of departure could be Imhotep, the architect and engineer who designed and built the Step Pyramid of Djoser at Saqqara in Egypt (2468 BC), in fact predating Greek thought on geometry. Such was the level of accomplishment of Imhotep that he was deified and revered in antiquity as one of the great thinkers of his time. Though layered in its geometry, the Step Pyramid proved to be the first viable pyramid design in Ancient Egypt and formed the basis, through multiple refinements, of the great pyramids in Egypt, including those at Gaza. Indeed, the pyramid builders of Ancient Egypt ultimately achieved a level of execution on par with their vision of the buildings: the fact that the Great Pyramid of Khufu at Gaza built by Hemon is less than 0.001% out of horizontal level and that the gap between most stones is less than the width of a human hair is a remarkable achievement of executed geometry even by today's exacting standards.

Just as the pyramid-builders had understood the material geometry of stone, the formidable Gothic cathedral-builders of medieval times further pushed the boundaries of the same material by expressing the geometry of structural forces — in their most sublimated form — through an intimate knowledge of stone masonry, construction and ultimately the ability to coordinate and manage a multi-man-year, multi-trade construction project at all levels, from aesthetics, to construction, to politics and finance. In other words, geometry formed one of the core components of the early *universal man*, the master-builder, who later found his maximum expression in the Renaissance.

Vitruvius and his *Ten Books of Architecture* laid the foundations not only for architecture in the Roman Empire, but for Classical Architecture through the ages. Viturivus' books are the first to bring together abstract geometry (Plato) and then-contemporary construction techniques for a wide range of building and engineering requirements. The rules laid down in his work influenced all subsequent architects of the past two-thousand years, from Michelangelo to Leonardo da Vinci and Leon Battista Alberti, who issued his own interpretation of the *Ten Books*, just as Palladio codified his own rules of architectural geometry in his *Four Books of Architecture*, as a means for others to be able to achieve the same geometric proportions in their work.

On a human scale, Alberti formally explored the subject of perspective in *De Pictura*, as did the painter and mathematician Piero della Francesca. While it is not a constructive geometry in the sense of formal proportions or structural relationships, perspective can nonetheless be considered a fundamental geometric device that allowed architects to move beyond stereometric projections to a visual description that is more akin to the human perception of space and form — the constituents of architecture. The Vitruvian Man, described amongst others by Leonardo da Vinci, furthers an anthropocentric understanding of geometry that was later echoed by Le Corbusier in the *Modulor*.

Invariably, the legal ramifications of the modern construction industry and the universal desire for excellence and accountability have fragmented the *master-builder* into the multiplicity of professions that regularly come together to design, develop and execute buildings. The major disruption here is not only the break-down of the *design mastery* into the modern professions of Architect, Structural Engineer, Mechanical Engineer and so on, but their formal and legal separation from the *construction mastery* that is embodied by General Contractors, Fa cade and Steel Contractors, and — most importantly — the fabricating subcontractors who actually embody the knowledge required to translate an abstract design into built form.

While architects, construction engineers and contractors must necessarily work together, collaborations between architects and other parties on geometric solutions for building designs have been traditionally infrequent, but in many cases highly successful and innovative. To name but a few, the synergetic designs of Le Corbusier and composer Iannis Xenakis (La Tourette et al.), Bruce Graham and Fazlur Khan (John Hancock and Sears towers in Chicago) and, more recently, Herzog & de Meuron and Ai Weiwei (Beijing Olympic Stadium) or Rem Koolhaas and Cecil Balmond (CCTV Beijing) have been ground-breaking in their time and have emphasised how the cross-pollination of ideas from different domains can often result in a revolutionary, rather than evolutionary, rethinking of architecture and trigger the often necessary technological process innovations necessary to achieve a new vision.

The AAG 2010 Conference comes therefore at a critical time for the specific subjects of *architecture* and *geometry*, where contemporary architectural theory and discourse is once again focused on the ancient relationship between the two, which has been vigorously rekindled by the myriad possibilities now granted to any designer through the universal availability of inexpensive yet powerful computational design tools. The availability of such geometric authoring tools however does not absolve the architect from his responsibility over the constructability of the geometry he is producing. By responsibility we are not talking about contractual liability here, though that invariably plays a downstream role; rather the issue of maintaining intellectual rigour and technical mastery of the work being produced.

There are different conceptual frameworks for the exploration of architectural geometry through computation; including Pre-Rationalised and Post-Rationalised geometries which are often seen to stand at odds with each other. The first implies a pre-conceived notion of fixed or predetermined form, while the latter suggests the search for an acceptable physical embodiment of an uncontrolled, free-form shape. Both are misconceptions of what are, in both cases, much richer architectural geometric strategies, each with its own advantages and disadvantages, as well as distinctly different formal outcome and creative potentials.

Pre-Rationalisation is also referred to as designing with “First Principles”, using a set of given geometric rules and methods to produce a solution that is constructible from the outset. These rules may incorporate parameters of manufacturability, material performance, acceptable functional minima and maxima, and, often, an understood range of cost implications. These First Principles, or rules and parameters, are not necessarily immediately understood, but rather may emerge from a conventional trial-and-error design process through which a particular architectural aspiration may be pursued. Early families of rules and parameters may in turn inform the architectural concept itself, and so on. It is the architect's responsibility, in this case, to ensure that the methods and rules that are available adequately implement the solution to a particular architectural challenge — the formal aesthetics often emerge from the application of the rules and principles themselves.

Such Pre-Rationalised design strategies have been exemplified by the broad range of works by firms such as SOM, Foster and Partners, KPF and, more recently, Wilkinson-Eyre. In contrast, Frank O. Gehry's work appears to stand in contrast to this method, whereby a Post-Rationalised approach seeks to provide a solution of constructability to a formal design initially developed more for its artistic expression than a pre-set solution of buildable rules. Form-finding in this case is understood differently: the process is driven by aesthetic concerns, and a shape may be produced by a variety of means, but with little or no preconception as to its method of construction or geometric language. In Gehry's work, a minimum of rationale is sometimes applied through the use of physical models, whereby the curving and folding of sheets of card mimics the material behaviour of sheet metal quite faithfully, the latter being a favourite material choice of that firm. Implicitly, the mathematical concept of the Developable Surface ($C_{gauss} = 0$) is embedded into the geometric materiality of the paper model and scales reasonably well to the full-scale architectural cladding artefact. In this case, the computer serves as a means to capture the initial gestures and refine the underlying geometry — in Gehry's words, to "catch the curve, not create it".

It is precisely this "creation of the curve" — i.e., using the computer directly as a creative design tool — that contemporary architectural discourse is so preoccupied with and, in terms of technical solutions and scientific method, that this conference attempts to address.

Architecture firms are becoming increasingly interested in complex geometry as they see it as a means of achieving form that can meet far more demanding aesthetic and performative criteria than was previously possible. In an age of world-wide competition, global ecological awareness and increasingly complex financial and programmatic conditions, architecture must be able produce solutions that transcend the boundaries set by modernism and traditional means of industrial production. To achieve this, architects are increasingly exploring the boundaries of geometric form expression through the use of advanced digital design tools. They are developing in-house 'specialist' teams of geometrically-savvy, computationally-literate architects, such as Foster and Partners' Specialist Modeling Group or the "CODE" unit at Zaha Hadid Architects. Parametric design tools and end-user programming ("scripting") allow different aspects of the Pre-Rationalised and Post-Rationalised approaches to be combined to rapidly explore multiple related design iterations across a solution field of desirable shapes, while maintaining a common codified geometric logic throughout the generation of families of possible designs. Downstream, further performative criteria can be applied to optimise a particular design approach and achieve the most productive iteration based on prescribed needs.

Many of the tools in questions have been 'imported' into the architectural profession from other fields, such as Maya from the cinematic arts, Rhinoceros from industrial design, and CATIA from aerospace engineering. Other software packages still, such as Revit, have been produced specifically for architecture, but as with all of the above tools and others still, are being continuously questioned, expanded, modified and repurposed by the end-user and tailored in their function to specific design purposes. It is now easy for *anyone* to produce curvy, blobby or irregularly repeating or branching shapes in comparatively little time. The ease of controlling such form-generation tools is deceptive, in that it does not manifest the rationale of constructability, technical challenges, material behaviour or issues of cost in this process. It also equally challenging to understand what is really happening within the computation tool; successful software attempts to simplify operations and geometric transformations for its users, often with well-mean obscurantism that defeats the need for geometric clarity and understanding. The temptation for the designer, then, to abscond the 'sorting out' of his or her geometry to others is, at this point, great.

As the interest in complex form grows, and with it the need to resolve difficult problems of geometry, new advisory firms such as ARUP's Advanced Geometry Unit, or the Viennese team Evolute, provide geometric and mathematical consultancy services that under the right circumstances of collaboration can lead to spectacular results. Academic research has an equally significant impact on the advancement of the field, and informs the necessary dialogue between academics (especially computer scientists and mathematicians) and paracticioners. Clearly, for the architect to fully appreciate and make the most informed contribution to such collaboration the basics must be present — one needs to but remember Plato's quote, which now takes on a very literal significance.

The availability, and indeed the necessity, of formal geometric education for aspiring young scholars of architecture are therefore paramount. This is achieved in two ways: through calculus and geometric algebra; and the study of formal descriptive geometry, over a number of years, without any computational aid. Regrettably, this is nowadays more the exception then the rule in that this level of knowledge must be persistently acquired during high-school; at university, it must be there to be readily applied, or it is often too late. Like a pianist in his early years, practicing and perfecting the Études by Chopin, the young architect must master the fundamental basics of geometry *formally* before he can hope to understand, and therefore control, what is happening in advanced form *computationally*. This is as important if the architect wants to accomplish his own resolution of design geometry in-house, or wishes to collaborate with other geometric experts on more challenging problems, and yet have the intellectual foundations to be able to participate fully in a collaborative discourse.

It is equally important for collaborating geometers and mathematicians to understand both the aesthetic preoccupations and professional obligations of the architect; in particular, that geometry in architecture is not abstract, but that it is very real and tangible, and closely tied to issues of material performance, fabrication and assembly methods and — most often — imperfect tolerances that drag an undesirable "fuzziness" into what is mathematically known to be a pure and perfect domain.

Nor is geometry free, but that it comes at often significant cost to the ultimate arbiter of any project — the Client — and that geometrically elegant or optimised solutions often need to be discarded, however refined, due to their lack of financial or even legal viability.

If we think of Confucius' quote, then it is this rich interplay between theory and practice that will ultimately yield, either through the personification of the *master-builder-geometer*, or in its distributed collaborative embodiment in teams, an architecture that finds its true expression through the constructability of applied geometry, and yet implements the performative, aesthetic, financial and political ambitions that make our world so rich and complex in the 21st century.

# Interpreting Physical Sketches as Architectural Models

**Barbara Cutler**

Rensselaer Polytechnic Institute

**Joshua Nasman**

Rensselaer Polytechnic Institute

**Abstract.** *We present an algorithm for the automatic interpretation of a rough architectural sketch as a consistent 3D digital model. We compare our results to the designer's intended geometry. We further validate the algorithm by studying the variations in possible interpretations made by other humans for a set of relatively ambiguous sketches. In our system, the user sketches an architectural design by arranging small-scale physical wall modules and simple markers for windows on a table. These color-coded elements are captured by a camera mounted above the scene and recognized using computer vision techniques. The architectural design is automatically inferred from this rough physical sketch transforming it into a consistent and manifold 3D triangle mesh representation. The resulting digital model is amenable to numerous building simulations including lighting, acoustics, heating/cooling, and structural analysis.*

## 1 Introduction

Sketching, drawing, and diagramming are fundamental components in architectural design. The evolution, communication, and documentation of a design are performed through various styles of visualization. These broad categories of visual communication in architectural design use a variety of artistic techniques; however, these representations are usually stylized and employ common conventions.

One important category of architectural illustration is the *figure-ground diagram*. This visual representation uses two contrasting colors, positive and negative, to partition space into two sets by filling in large regions of the diagram with solid color. In architecture, a figure-ground diagram is most often drawn in plan (from above) to convey either the rough overall massing shape of the building or the public freespace; for example, a public plaza surrounded by private buildings. Often architects will execute diagrams in both forms when considering different aspects of the same project. Another important class of diagrams used in the early stages of architectural design, *circulation diagrams*, visualizes how people will use the space and highlights the common movement paths within the proposed design. By analyzing and anticipating common paths, the relative placement of spaces within a design and the relationship to the existing site can be optimized to minimize path lengths or to add interest or drama by highlighting views and enhancing the experience of circulating within the design.

Figure-ground and circulation diagrams are used primarily in the early stages of design when the spaces and relationships are still evolving. In contrast, technical architectural CAD drawing, used in the later stages of design and in construction documentation, is highly precise and detail-oriented and more strictly follows diagrammatic conventions. Few architects would claim that traditional CAD modeling tools and detailed technical drawings are essential to the early, creative stages of architectural design.

In addition to pen-and-paper sketches, small-scale physical 3D models (often built from scrap cardboard) are fundamental tools for architectural design. These study models can be essential for understanding complex spatial relationships, documenting the evolution of a design, and communicating the concept to the client. Even with the wholehearted adoption of computer technology for drafting and 3D visualization, the physical study model has not been abandoned as a tool for architectural design. In fact, rapid prototyping technology has increased the expectations for physical prototypes of complex designs.

### 1.1 Tangible User Interface for Architectural Design

The architectural modeling system at the center of our project uses a *tangible user interface*, which involves manipulation of physical props for interaction with computation (e.g., [Ben-Joseph et al. 2001]) rather than the typical mouse/keyboard/monitor interaction between human and computer. Well-designed tangible interfaces are attractive because they are inherently simple, natural, and intuitive. Furthermore, these interfaces generally support collaborative work environments.

In our system, shown in Figure 1, one or more users gather around a table and construct a small-scale (1:12, 1” = 1’) sketch of an architectural design using simple foam board flat and curved walls in three different heights (5”, 8”, and 10”). Special markers slip over the top edges of the walls to indicate windows, and the overall orientation of the architectural design on the site is specified with a “north arrow” token. This design environment is simple to operate and requires essentially no instruction to use. The only restriction on the designs is that wall elements must be upright, resting on small “feet”, so that each wall surface is perpendicular to the table surface. A new design can be quickly constructed in under a minute by selecting and arranging wall and window elements from a modest collection of parts on a neighboring table. Similarly, the design can be edited in seconds by adjusting any of the physical pieces. Image capture and processing of the detected geometry is completed in a couple of seconds. The system supports viewing and editing by multiple users who are gathered around the table. The interactive modeling environment encourages creativity and collaboration.

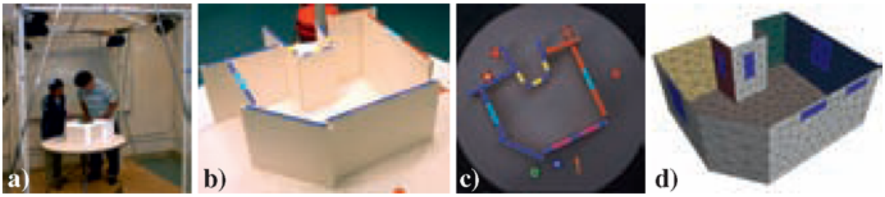


Figure 1: In our physical architectural design environment a) users gather around a table and construct b) a small-scale mockup of a design from a collection of wall elements and marker tokens. A camera above the table c) captures the layout of elements on the table. Our sketch interpretation algorithm processes these elements to construct a consistent and watertight triangular mesh of the implied architectural design (ceiling removed for visualization).

Within this design environment the designer cannot create a highly detailed model, but instead is compelled to focus on more abstract concepts appropriate for the early stages of design, including orientation of the building on the site and spatial relationships between the primary zones of the design, and in making key decisions about the structure, lighting, and acoustics. Importantly, computational simulations to analyze the performance of the structure, lighting, or acoustics of the space (which are currently underutilized during schematic design) require a consistent and watertight, yet simple, 3D representation of the design for efficient and accurate analysis. Thus, we believe our algorithms to produce such a model from these sketches will be an invaluable tool in the early stages of architectural design.

### 1.2 Contributions

In this paper we present the following contributions:

- An algorithm and implementation for the automated interpretation of physical sketches as consistent architectural designs which can be exported as either standard architectural floor plans or watertight 3D triangular meshes amenable for simulation.
- The collection of several hundred physical architectural sketches using our design environment. Each design is annotated by the original designer to indicate the intended interpretation of the design.
- Re-interpretation of these designs by other humans to provide a measure of the ambiguity present in these sketches. The set of human interpretations is compared to the algorithm's output for validation.

### 1.3 Overview

We summarize several important areas of related work in Section 2. In Section 3 we present our sketch interpretation algorithm, which was developed with extensive user testing and feedback from both architects and non-architects. In Section 4 we present the results of two formal user studies that we conducted to gather a large set of example designs and validate our sketch interpretation approach.

## 2 Related Work

Our project draws from research in a wide variety of areas including: sketching interfaces, sketch recognition, human perception, and computational models of gestalt and saliency. In the sections below we provide a brief overview of prior art in these fields and existing software for architectural design.

### 2.1 Modeling Interfaces for Architectural Design

Many existing computer software packages tackle the challenge of constructing 3D architectural geometry. Initially these packages focused almost exclusively on creating models with high geometric precision, requiring a significant investment of time and only limited support for editing a completed model. Unfortunately, because focusing on precision too early in the design process can stifle creativity [Lawson 2002], these tools are generally used only in the later stages of design.

The requirements for 3D models in architecture vary tremendously depending on the intended use of the model. Photorealistic renderings require high polygon count geometry and accurate normals and materials. In contrast, many simulations (e.g., acoustic, passive ventilation, heating/cooling, structural analysis) require a consistent and watertight geometric model for accurate calculations and often perform most efficiently with a simplified model constructed especially for that analysis [Autodesk 2000-2008].

More recently, software tools have become more amenable to the fast-paced, quick sketching of the early stages of design [Google 2010], including explorations of pen-based user interfaces for 3D modeling [Zeleznik et al. 1996; Igarashi et al. 1999; Lipson et al. 2002]. A limited construction interface (axis-aligned elements on a floor plan grid) can help ensure the construction of architectural environments that are appropriate for use in virtual reality walkthroughs [Mackie et al. 2004]. New drawing user interfaces and systems have been demonstrated that allow architects to leverage their pen and paper skills when interacting with the new media interface of the computer. Using projective geometry, freehand architectural sketches can be re-projected or warped to simulate novel viewpoints and an immersive experience [Tolba et al. 2001]. The Mental Canvas system allows designers to arrange 2D sketches on arbitrary planes in 3D, constructing an effective representation of complex architectural spaces [Dorsey et al. 2007].

### 2.2 Sketch Recognition

Sketch recognition systems are typically custom-developed for each application (e.g., recognizing hand drawn UML diagrams [Lank et al. 2000]) and leverage domain-specific knowledge to improve accuracy. General-purpose toolkits and languages for sketch recognition of diagram components can ease the development of these programs [Hammond and Davis 2007].

Well-designed sketching user interfaces minimize the number of times the user is prompted for additional information or design annotation, which interrupts workflow and concentration. Strategies for continuous and incremental recognition of drawings as they develop have been demonstrated [Alvarado and Davis 2001]. It is important to maintain an estimate of the confidence in the intermediate interpretations, which improves the accuracy of the final interpretation. An ongoing challenge in sketch recognition is the grouping of multiple individual strokes that form a single logical unit. The drawing sequence for individual strokes and other meta-data available from tablet displays [Wacom 2010] can be used as evidence when tackling this problem [Gross 1994]. Another challenge in sketch recognition comes from touch-up or continuation strokes the user might make to complete a drawing [Paulson 2010]. These strokes are disconnected from or overlap other strokes but are intended to be recognized as a single object. Spatially close strokes (often defined as the distance between two endpoints being less than 10% of their average length) can be merged; however this greedy approach, while fast enough for interactive sketching, is not optimal.

### 2.3 Sketch Recognition for Architectural Drawings

The precision, consistency, and standardization of architectural CAD drawings make this domain amenable to automatic processing algorithms, and an impressive collection of automated sketch recognition programs have been developed for architectural drawings [Ah-soon and Tombre 1997; Ah-Soon and Tombre 2001; Kulikov 2004; Lu et al. 2005]. The motivation behind many of these tools is the digitization and automated reconstruction of 3D building geometry from older construction documents. These methods have also been demonstrated for hand-drawn architectural designs that closely follow the diagrammatic conventions of CAD drawings [Aoki et al. 1996; Llados et al. 1997]. For example, the VR Sketchpad project automatically interprets a 2D floorplan sketch including furniture layout to create a 3D VRML model that can be used for walkthroughs [luen Do 2001]. Freehand sketching can be used to interact with digital models [Gross and luen Do 2000] and preliminary work was done to classify interior and exterior walls in quick floorplan sketches [Ramagupta and Hammond 2007].

Koutamanis and Mitossi describe three levels of automated recognition of architectural floor plans: recognition of geometric elements, recognition of building elements, and recognition of spatial articulation [Koutamanis and Mitossi 1993]. They argue that the third category is the most advanced: identifying solid mass versus space within the design. Our aim in this work is to specifically address this challenge for freeform architectural sketches using a tangible interface.

### 2.4 Gestalt Theory and Sketch Interpretation

Gestalt psychology, the laws of perceptual organization, and Pragnanz [Koffka 1935; Kanizsa 1979] describe how humans perceive and interpret incomplete diagrams or other modes of partial stimuli. The fundamental phenomena of closure, proximity, symmetry, and continuity can be explained by low level human vision processing. Gestalt theory describes why our interpretation of an incomplete or ambiguous diagram tends toward simpler forms, avoiding complexity. The rich vocabulary of pen and paper sketching in architectural design draws on the gestalt principles of collinearity, parallelism, continuation, and completion [Koutamanis 1999]. Our algorithm for automated interpretation of architectural sketches follows and implements these principles. Our user studies on human interpretation of architectural sketches provides validation to our proposed use of these concepts in recognition and analysis of architectural designs.

Research in computer vision has developed algorithms for image processing using computational models of gestalt. Attributes which define the form, i.e., thickness of a line, convexity, and parallelism, can be referred to as *gestalts* [Desolneux et al. 2002]. Computational gestalt research focuses on determining thresholds that indicate a pattern is significant. In other words, this work involves detecting and studying patterns and analyzing how likely those patterns would be to occur in the image randomly [Desolneux et al. 2002].

Similarly, saliency is a measure of how much an area stands out in comparison to the areas around it. A *saliency map* [Koch and Ullman 1985] combines different stimuli (movement, color, etc.) and the relative conspicuity to quantify changes in these characteristics. Itti, Koch and Niebur designed a computational visual attention model based on the primate visual system to estimate the saliency map and identify which areas of the scene are most likely to contain useful information and should be analyzed in more detail [Itti et al. 1998].

## 3 Sketch Interpretation Algorithm

In our physical sketching environment, the user selects from the provided collection of physical props and quickly assembles the chosen pieces on the table. Thus, the arrangement of elements truly forms a rough, approximate sketch. The selected wall pieces are likely too long or too short, yielding overlaps or gaps with adjacent walls. Similarly, the limited palette of curved components may not have the desired curvature and thus tangential connections are imprecise. Furthermore, the approximate assembly manner means that components intended to be parallel or meet at crisp 90° angles will include some unavoidable imprecision. The challenge is to sift through the available information to deduce and construct the clean and complete design as it was conceived in the architect's mind.

In the following sections we present the key steps of our sketch interpretation algorithm: image pre-processing, detecting parallel, perpendicular, and collinear elements, linking elements into chains, constructing an arrangement of polygonal cells, estimating spatial enclosure, assigning interior/exterior zones, managing details, and post-processing the floorplan geometry to make a 3D model.

### 3.1 Image Pre-Processing

A camera mounted above the table captures the details of the physical sketch. A controlled lighting environment and carefully chosen color-coded labels on the physical

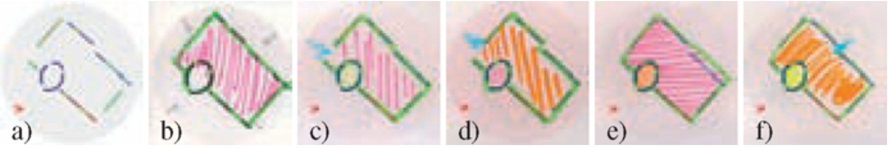


Figure 2: Intended collinearity can be ambiguous: a) detected primitives, b) annotation by the original designer, and c-f) annotation by other users.

elements allow this geometry to be robustly detected using standard image processing techniques, which are described in detail in our earlier publications [Sheng et al. 2009b; Sheng et al. 2009a].

The input to our sketch interpretation algorithm is the 2D projection of each detected wall module onto the table surface, labeled with the element height. The flat walls are rectangles and include any associated windows as inscribed rectangles. The circular arc curved walls are specified by a center, the inner and outer radii, and the start and end angles of the arc. An example of this input is shown in Figure 2a.

### 3.2 Intended Parallel, Perpendicular, and Collinear Elements

For practical construction and space efficiency reasons, most real-world architecture involves parallel walls and sharp 90° corners. Even within high-profile showpiece architecture, spaces are typically arranged into regular patterns and placed on a grid, sometimes with secondary grids that are offset and/or rotated. Many (but not all) designs created by participants using our tangible sketching interface follow these conventions and we can automatically detect the implied grid(s).

First, we cluster the flat walls into groups that are nearly parallel and *snap* all walls in each group to their weighted average direction. Similarly, we compare the wall groups to each other and those that are approximately perpendicular are snapped to be orthogonal. We found that an angle tolerance of 5° was an appropriate threshold across all of our collected design data. This tolerance was effective at cleaning up placement imprecision inherent in the physical sketching environment, yet was not large enough to introduce artifacts in the more freeform designs that eschewed parallelism and right angles.

In addition to the angle tolerance, it is necessary to identify flat walls that are approximately collinear. However, selecting a global setting for this offset tolerance distance is somewhat more difficult. We also saw variable tolerances for collinearity in human perception when different users were asked to interpret the same design (Figure 2). Some users perceived a slight line break as an intended straight line, while others interpreted the break as an architectural feature and possibly an entrance. We found that using a 1" offset distance tolerance in our physical sketching environment (equal to 1' in full-scale) was a good compromise for our automatic collinearity detection and adjustment, but the user may need control of this threshold for some designs.

We have not yet finished implementation of similar clustering and adjustment methods for circular arcs. The implementation is straightforward and will require determination of appropriate tolerances and smoothing procedures for several cases: multiple arcs placed to sketch a circle or ellipse (e.g., Figure 2), two arcs that form an inflection point, and adjustment of the tangent and/or curvature when a circular arc leads into a flat, straight wall.

### 3.3 Linking Walls to Form Continuous Chains

Following the gestalt principle of continuity we not only snap nearly collinear elements to a common line, but we also form explicit connections between elements with similar tangents (straight to straight, straight to curved, or curved to curved). Two or more elements can be connected into a *chain* that is traced through the working plane, separating space into two regions. Our angular and offset distance tolerances for establishing a connection between two elements is less strict than the parallel and collinear thresholds described in the previous section, allowing the establishment of longer freeform chains. Examples of these chains are shown in Figure 3c.

Our algorithm for establishing the chains is as follows. For each endpoint of each wall element we search all other walls for the best matching connection tangent. If a pairwise connection is mutual (element A selects element B as the best match for tangent direction and offset distance and element B also selects element A as the best match), then the connection is established. If no connection is made for a wall endpoint, then that end of the wall chain is simply extended to infinity following the tangent of the wall at that endpoint. When a wall chain connects curved arcs, the chain may form a U turn (Figure 3 second row), a closed loop (Figure 3 third row), or other interesting shapes.

Defined spaces in architecture can be created by real and/or implied boundaries. Each wall chain divides the working plane into two spaces, one on either “side” of the chain. The set of all wall chains in a model will divide space into many subspaces that can be labeled by their sidedness, which is visualized in Figure 3d. If a wall chain loops around and crosses itself (Figure 3 fourth row, blue and yellow chains), the loop portion of the chain is disconnected to define a new chain to allow the unique labeling of all subspaces. Note that if two wall chains cross two or more times (which is possible if one or both chains are non-linear), the resulting subspace organization may have two disconnected spaces that have the same sidedness (Figure 3 fifth row, blue and red cells). We perform a connected component analysis to identify this situation and define separate subspaces.

### 3.4 Arrangement of Cells and Enclosure

The wall chains described in the previous section are used to cut the working plane into a watertight planar convex polygonal mesh or arrangement of cells, which is represented using a half-edge data structure. Each wall chain explicitly represents the wall thickness, thus working plane polygons are constructed for both open space (interiors and exteriors) as well as the are comprising the construction thickness of real and implied walls.

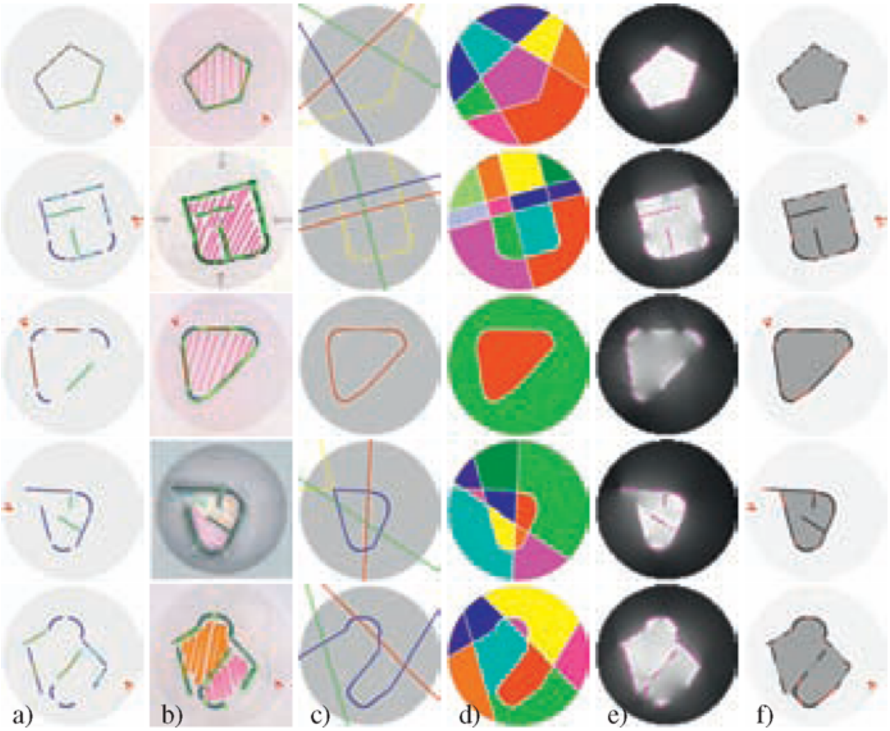


Figure 3: Dividing space into cells by extending tangents and connecting elements: a) detected primitives, b) annotation by original designer, c) wall chains, d) zones defined by the wall chains, e) enclosure, and f) our automatic interpretation.

For each cell in the arrangement, we calculate the *enclosure*, the probability that the cell is part of the building interior. We define the enclosure at a point as the lack of visibility to areas outside of the design from that point. We estimate this value by tracing rays from the point to infinity in a dense sampling of directions and record the fraction of all rays that intersect a wall of the design. We visualize enclosure over a dense point grid in Figure 3d. Points with high enclosure (nearly all rays intersect a wall in the design) are drawn light grey or white, and points with low enclosure are dark grey. We define the average enclosure of the cell in the arrangement by averaging the enclosure at many points within the cell. Similarly, we can compute the average and standard deviation of the point-based enclosure values for a subspace.

For relatively simple designs, a carefully-selected global threshold placed on the point/cell/subspace enclosure value can correctly classify each subspace as interior or exterior. However, as the gaps between the walls increase or decrease the threshold value must be adjusted accordingly (compare the first and third rows of Figure 3d). Furthermore, if the model contains concavities in the outer wall, nearby

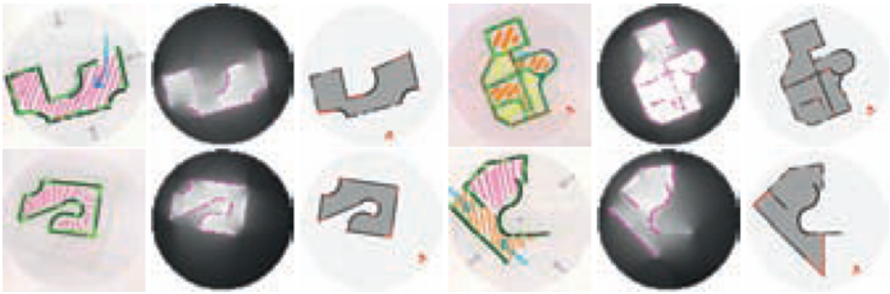


Figure 4: Designs with non convex boundaries may not be accurately extracted with a simple threshold on the average point or cell enclosure. By minimizing the lengths of unused wall and extra *inferred wall* necessary to enclose the interior zones we correctly interpret these complex designs.

areas may be incorrectly marked as interior spaces. Beyond the challenge of selecting a threshold, for more complex models a simplistic setting of a global threshold will not successfully separate the building interior from exterior (Figure 4).

One cause for incorrect interior/exterior division is high variance in enclosure within a single subspace. For example, consider the final row of Figure 3. The sharp bends of the blue wall chain loop through the design and interact with the other two wall chains to produce one large subspace, colored cyan and green. The enclosure values within this subspace have high variance, indicating that it should not be treated as a single space when identifying interior space. For each subspace with high enclosure variance, we solve a Minimum Cut graph problem to determine an optimal segmentation of this subspace. We build a dual graph from the polygonal cells in this subspace. Each polygonal cell in the arrangement is a node in the graph. If two cells share an edge in the arrangement, we create an edge between the corresponding nodes in the graph. The weight of the edge is defined to be the length of the shared edge in the arrangement. The source is defined to be the node whose corresponding cell has the highest enclosure, and the sink is likewise defined to be the cell with the lowest enclosure value. Using a basic textbook implementation of the Maximum Flow/Minimum Cut algorithm, we find a minimum length cut which divides the zone into two subspaces that will have lower variance and produce a more satisfactory interior/exterior segmentation of the design.

### 3.5 Assigning Interior and Exterior Zones

After the wall chains have divided the working plane into a set of zones, we need to label each zone as either an interior or exterior space. The average enclosure value for the zone can be used to make an initial determination, but that strategy frequently yields incorrect assignments for complex designs with non-convex boundaries (e.g., Figure 5). Furthermore, our method for constructing wall chains that extend each element to infinity will yield a division of space into zones far from the element's

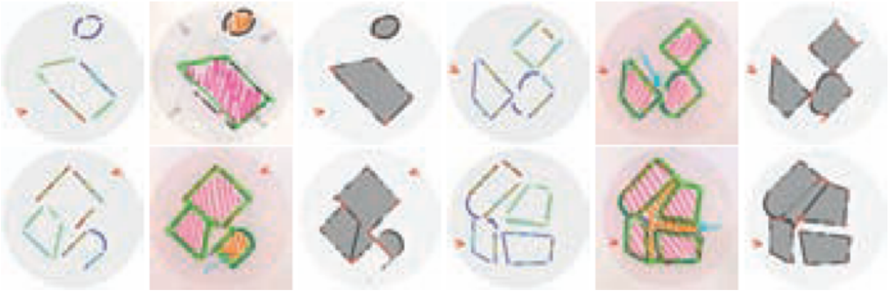


Figure 5: Challenging examples of designs with multiple rooms. The examples in the bottom row are somewhat ambiguous and have multiple reasonable interpretations for the passageways between rooms.

actual position, which may not be intended. When two neighboring zones are assigned opposite labels, the wall or wall chain between the zone is thus interpreted as an implied wall or boundary. If the original wall element is short and/or a significant distance from the implied boundary, the resulting automatically-constructed floor plan may be non intuitive and not match a human interpretation. Therefore, we must be careful to use an extended wall chain as evidence for a boundary only in close proximity to the original wall. Every hypothesized or inferred wall separating interior and exterior space should be checked against the evidence.

In solving this problem, we follow the Gestalt principles of closure and simplicity. We search for a closed form that is simple, uses most or all of the detected wall elements, and requires little length of additional inferred exterior wall to fill gaps between the original wall primitives of the sketch. We solve this optimization problem in a brute force manner by considering as interior space all subsets of the zones with enclosure values above a reasonable threshold and select the zone assignment that minimizes the sum of all unused walls and all inferred walls. An unused wall is defined as a detected element that has exterior zone on both sides. Note that wall elements may be partially used and the unused wall penalty is accordingly prorated. Similarly, an inferred wall is a portion of wall chain that is not represented in the sketch by a physical wall, but has exterior zone on one side and interior zone on the other. In the floor plan results diagrams used throughout this paper, real walls are drawn in black, interior zones are drawn in medium grey, exterior zones are white and inferred walls are drawn in dark red for visualization purposes.

Some of the collected designs contain an interior space that might have been conceived as an open courtyard rather than a room with no exterior walls (Figure 6). This distinction can be important for architectural simulations such as daylighting design or passive ventilation — whether the finished design includes a roof over this fully interior room will impact the simulated performance results. To resolve this ambiguity, we are considering user interface options for allowing the designer to indicate which of these two perfectly reasonable scenarios he/she intended. Importantly, we want maintain our minimalist interface and follow the most appropriate default interpretation.

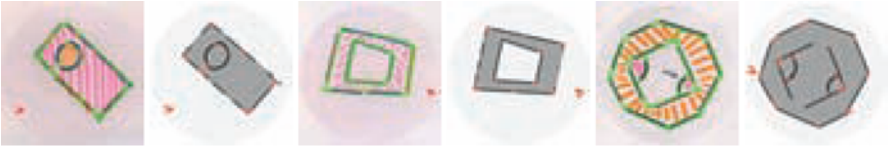


Figure 6: If the design contains a interior room with no exterior walls, this space may be intended as a courtyard space, uncovered by a roof. The system will require extra information from the designer if the default interpretation does not match his/her intentions.

### 3.6 Detecting Inferred Interior Walls and Trimming Unused Walls

Once the core interior/exterior spaces of the design have been determined and the true exterior and inferred exterior walls have been identified, the system performs several small cleanup steps to improve the quality of the final model and ensure that the extracted 3D geometry is manifold. When two wall chains that each define a portion of the boundary between interior and exterior cross, the quadrilateral zone defined by the thickness of each chain at the crossing should also be labeled as a wall to create proper interior and exterior corners.

Some designs incorporate interior partition walls that are meant to separate the interior into smaller rooms. Due to the physical sketching environment, generally these walls are a bit short and leave small gaps where they should meet other interior or exterior walls. However, not all gaps should be closed. Standard doorways in architecture are generally a minimum of 2’6” wide, so if the gap is less than 2” wide, we assume this was intended to be a solid wall and we close the gap by labeling the corresponding wall chain polygons as inferred walls. We found that this tolerance provided a reasonable match to the designer’s annotation of his/her intentions for the full range of example designs. Similarly, when the physical wall modules are slightly too long and protrude from the middle of an inferred boundary or corner, we should clip back this geometry to create a more polished design. We propose a tolerance of 1” (corresponding to 1’ in full scale) for this trimming. We note that it is important to not completely remove from the design all walls that are “unused” (not positioned on the exterior/interior boundary of the model or used as interior partitions). In many cases these extra exterior walls serve specific architectural functions including privacy screens, shading, acoustic dampening, and wind control.

Finally, we propose an initial strategy for labeling the primary entrance to a design, and augmenting the floorplan and 3D model with this information. Note that not all designs have a clear entrance, but many of our user study participants left specific openings within the outer boundary of their shape. Simply detecting the longest section of inferred exterior boundary (and greater than a minimum tolerance of approximately 4”) as the primary entrance will correctly label this feature in most of the designs. One notable example that breaks this rule is shown in the bottom row of Figure 7. Annotations made by other user study participants match the intention of the original designer: the obvious entrance is through an elaborate portico, rather than through one of the large gaping holes in the “back” of the de-

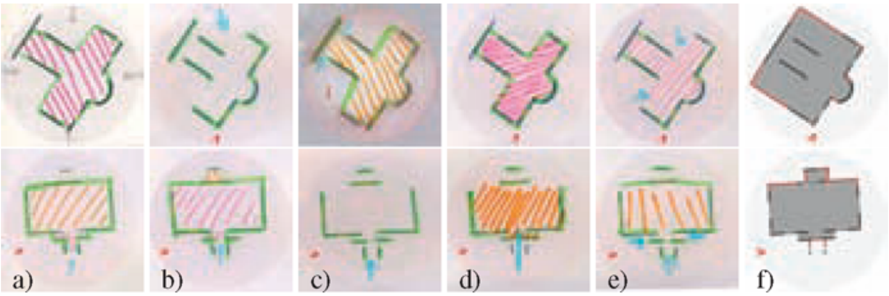


Figure 7: Domain-specific knowledge may be necessary to correctly interpret sketches that hint at common architectural forms, such as the cruciform used in church floor plans (top row) or to recognize an entrance portico (bottom row). Despite the potential for ambiguity, b-e) most users' interpretations matched a) the original designer's intention. Our automatic sketch interpretation results are shown in f).

sign. This sketch interpretation task will benefit from domain-specific knowledge of common architectural forms.

### 3.7 Post-Processing: Constructing a Watertight Triangle Mesh

At the end of our sketch interpretation algorithm, we have a precise, manifold, polygonal representation of the working plane. Each polygonal cell has well defined neighboring cells (no 'T' junctions) and has been assigned one of several labels: *solid wall*, *window within a wall*, *interior space*, or *exterior space*. This representation can be extruded and exported as a consistent mesh following the necessary CAD conventions for architectural rendering or performance simulation software. For example, we can construct a watertight mesh appropriate for a radiosity simulation of interior illumination as follows. Each interior cell is exported as two polygons, one floor plane polygon with normal pointing 'up' and one ceiling polygon with normal pointing 'down'. For every edge between an interior cell and a wall or exterior cell, we create a wall polygon stretching from the floor to the ceiling with normal pointing toward the interior cell. When an interior cell touches a window cell, the exported wall polygon is split vertically and assigned different materials as appropriate.

## 4 Validation of Physical Sketch Interpretation Environment

To validate our algorithm for sketch interpretation, we ran two user studies, one focusing on our physical sketching environment and the second on sketch interpretation. We recruited participants with a variety of backgrounds including architecture, visual arts, and computer science.

### 4.1 Design Collection Study

The purpose of the first user study was to sample the range of architectural designs that could be constructed in our physical sketching environment and to evaluate the potential for its use in the early stages of architectural design. The design task was open-ended and after a brief overview of the tangible interface, participants were instructed to use the wall and window primitives to create between 10 and 20 different designs.

After each participant completed the design stage, we prepared a single-page annotation form for each of his/her designs. The form contained two parts: *designer annotation* and *evaluation of automated sketch interpretation algorithm*. The form was folded in half so that only the annotation section was visible and the participant was instructed to first complete the annotation for all designs before unfolding the paper to see the output from our automated sketch interpretation algorithm. Thus, the participants were not influenced by the output of our program in either the design or annotation stages.

The annotation portion of the form presented two large images: the overhead photograph of the physical sketching environment (for reference) and a 2D rendering of the detected wall geometry (to be used for annotation). The participant was instructed to use a green highlighter to draw the complete intended wall geometry on the detected geometry rendering. The pink, orange, and yellow highlighters were used to shade interior spaces. Optionally, he/she could use a blue arrow to indicate an entrance or to sketch the circulation within the design. As guidance, users were provided with three example designs annotated in this manner.

The evaluation portion of the paper contained our automatically generated floor plan of the design. The users were asked to evaluate the quality of the automatic interpretation of each design, whether it matched the design intention, was an acceptable alternate interpretation, or was incorrect. Additionally, we encouraged them to mark or comment on which parts of the design were most challenging for the automated system to interpret. After completing the evaluation of all designs, the users filled out a short post-study questionnaire.

Each participant used our physical sketching environment for approximately 20 minutes and created 3-26 designs. Some users created just a few highly detailed designs, while others created many rough sketches or a series of variations that evolved from a base sketch. In total we collected 329 designs from 30 participants in the first user study. Fifteen of those participants (responsible for 154 of the designs) were architecture students, most with at least three years of formal architectural education and professional experience through internships. Of the other participants, eight were visual arts students (83 designs) and the remaining seven (92 designs) had no formal training in architecture or art. A broad selection of these designs are presented in Figure 9.

### 4.2 Re-Interpretation Study to Quantify Design Ambiguities

For the second study, we wanted to understand any discrepancies between our algorithm's interpretation and the original designer's intentions. We wanted to investi-

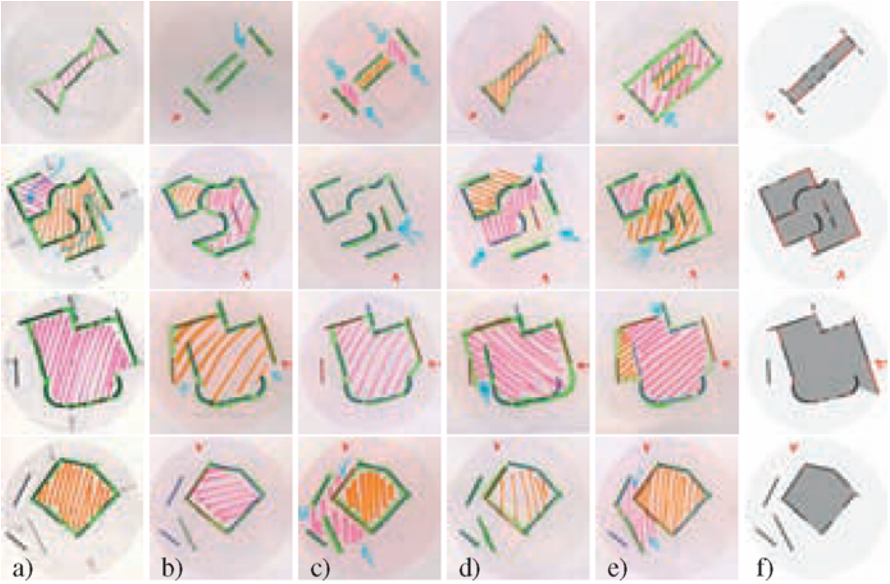


Figure 8: Some ambiguous designs: a) the original designer's annotation, b-e) annotation by other users, f) our automatic sketch interpretation results.

gate whether the differences were due to a flawed automatic interpretation strategy or the result of ambiguous physical sketches (Figure 8). In order to quantify the ambiguity of a particular design, in our second user study we asked the participants to annotate a selection of interesting sketches made by other users. All participants for the second study first performed the tasks of the first study (if they were not already subjects in that study) and thus all were familiar with the sketching environment and annotation instructions.

All designs from the first study that our initial sketch interpretation program struggled with were selected (omitting near duplicates), as well as other designs that we thought were ambiguous, complex, or interesting. We also included a number of simpler designs, which had a single reasonable interpretation, as controls. In total, 114 of the 329 total designs from the first study were selected. 60 of these designs were created by architecture students, 28 were made by visual arts students, and 26 were from students with no formal training in architecture or visual arts.

Each participant for the second study was presented with annotation paperwork for a randomly ordered, randomly selected subset of these designs. The annotation form consisted of three parts: *annotation*, *comparison to the original designer's intention*, and *evaluation of automatic interpretation*. The forms were folded to conceal the second and third parts.

As in the first study, the participants were asked to mark their interpretation of each sketched design and shade the interior spaces. After approximately 20 minutes each participant was asked to proceed to the second part of the study and any designs he/she had not yet annotated were collected.

Next, the participant was asked to unfold the paper (keeping the third part concealed) and compare his/her interpretation of each design to the original designer's intention, marking whether the interpretations matched or how the designer's physical sketch was ambiguous or unclear. Finally, after all comparisons were made, the participant unfolded the third and final part of the form and evaluated our computer algorithm for automated sketch interpretation.

We collected a total of 346 new annotations from 15 participants (124 annotations were made by architecture students, 82 by visual arts students, and 140 by other students). Each of the 114 designs received between 3-6 new annotations.

### 4.3 Validation Results: Subjective Feedback

The users' questionnaires revealed how excited they were about our physical sketching environment. A visual arts student said the system was "Very intuitive, very clear. Felt like playing with blocks as a kid, but each block had a meaning. Seeing each design in 3-D helped spike the creativity." Another visual arts student commented: "I was really impressed with the software–it did a great job mapping what I wanted." A second year architecture student said of the system "I was very surprised by the accuracy of the program for the most part. Despite some errors, the interior and exterior implied spaces were read pretty well." Other users were surprised at particular failings for rules we had not yet implemented. "The program filled in a void that was meant to be exterior, especially since I had windows on the exterior parts of these walls to make that distinction."

We used direct feedback from architecture students in pilot studies as well as general observations about the designs they created to improve our sketching environment and automatic interpretation algorithm. These improvements include: addition of curved walls and column primitives, control over window placement, detection of disjoint spaces and interior courtyards, and handling designs with large gaps in the exterior wall (typically, an implied entrance). We are motivated to continue this avenue of work and tackle the challenges of detection and labeling of the different subspaces within the interior, predicting interesting circulation patterns, etc.

The results of our second study can best by summed up by one student. Interpretation was "Often challenging. Many designs were unclear, difficult to interpret. Others were extremely clear and easy." Some users were surprised by the variety and complexity of designs possible in the system. A visual arts student said "I was surprised at the designs that the other users came up with – they seemed very complex in some cases – and the computer did a good job of interpreting them."

We found that for many of the designs that our algorithm struggled to interpret, other humans also found the design to be ambiguous. However, there were several notable examples where all humans interpreted the design quite similarly to the original designer, despite a lack of hard evidence for that shape within the sketch. Figure 7 presents a few of these examples, where the humans are quite consistent in their interpretation of the design. We believe this may be due to domain-specific

| | correct | | mostly correct | | incorrect | | total |
|---|---|---|---|---|---|---|---|
| clear | 155 | 78% | 17 | 9% | 26 | 13% | 198 |
| ambiguous | 74 | 56% | 35 | 27% | 22 | 17% | 131 |
| total | 229 | 70% | 52 | 15% | 48 | 15% | 329 |

Table 1: Statistics about the ambiguity of the physical design sketches and the quality of the interpretation results from our automated sketch recognition algorithm.

knowledge of architectural forms that have not yet been explicitly encoded in our sketch interpretation algorithm. One architecture student noted for the example shown in the top row of Figure 7: "Humans recognize this because it is a basic cruciform shape, but without more information, it may be difficult for an algorithm to determine this."

### 4.4 Quantitative Sketch Interpretation Results

After all of the designs and annotation figures had been collected, each physical sketch was categorized as "clear, straightforward, unique interpretation" or "ambiguous, multiple interpretations possible". This was determined by criteria such as if there were openings that may or may not be doors and whether it was clear which spaces were interior versus exterior. Secondly, each automated interior/exterior space partitioning by our algorithm was graded on the following scale: "correct", "mostly correct", or "incorrect". In order for a design to be marked correct, all interior spaces had to match and all walls that were part of the design had to match exactly with the designer's intention. An interpretation was judged to be mostly correct if at least 90% of the walls matched and if each interior space was mostly bounded by real walls. The results are shown in Table 1. In total, our algorithm found a correct interpretation for 70% of all designs and correct or mostly correct interpretation for 85% of all designs. Of the designs that were judged to have single clear interpretation, we made the correct interpretation for 78% of the designs. In contrast, for the ambiguous designs we found a correct interpretation (closely matching the original designer's intention) for 56% of the designs. Many of the errors in interpretation made by the system are minor robustness issues and we are confident that with additional development efforts the accuracy of the system will improve.

We analyzed the annotations to determine if there was any correlation between the architectural or visual arts training (or lack thereof) of the original designer or secondary annotators. We did not find a correlation between the background of the participant and their ability to correctly infer the original designer's intention.

## 5 Conclusion and Future Work

We have presented an algorithm for automatically interpreting approximate physical sketches of architectural designs, preparing detailed floor plans of these designs with explicitly represented interior and exterior space, and converting these floor plans into watertight 3D meshes that are appropriate for simulations of building performance. We presented a validation of the effectiveness of the physical sketching environment for modeling and of our algorithm for automatically and correctly interpreting these designs. Response from both architecture and non architecture students about the system has been positive and encouraging.

Our current interpretation algorithm is quite successful at interpreting complex designs and produces reasonable results even for rather ambiguous sketches. We will continue to improve the algorithm, revising the rules for linking walls and defining separate interior spaces. We would like to incorporate domain-specific knowledge of common forms in architectural design and leverage symmetry within the sketch. Prior work in computer graphics demonstrates how approximate symmetries within a model allow decomposition, identification of correspondences, compression, warping to make the mesh more symmetric, and hole filling of missing data missing data [Golovinskiy et al. 2007; Pauly et al. 2008]. Furthermore, we plan to explore the automatic recognition of circulation paths within a design and generate appropriate roof overhangs and sloped roof shapes for the detected geometry.

We believe the core of the interpretation algorithm described in this paper can be extended to other forms of architectural sketching. For example, a direct digital equivalent of the physical environment with drag & drop, translation, and rotation of components would be straightforward. The system could also be adapted to a tablet display environment using existing sketch recognition technology for parsing straight and curved pen strokes.

## References

AH-SOON, C., AND TOMBRE, K. 1997. Variations on the analysis of architectural drawings. In *In Proceedings of Fourth International Conference on Document Analysis and Recognition*, 347–351.

AH-SOON, C., AND TOMBRE, K. 2001. Architectural symbol recognition using a network of constraints. *Pattern Recognition Letters 22*, 2 (February), 231–248.

ALVARADO, C., AND DAVIS, R. 2001. Resolving ambiguities to create a natural computer-based sketching environment. In *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann Publishers, 1365–1371.

AOKI, Y., SHIO, A., ARAI, H., AND ODAKA, K. 1996. A prototype system for interpreting hand-sketched floor plans. *Pattern Recognition, International Conference on 3*, 747.

AUTODESK, 2000-2008. Ecotect Analysis. `http://www.autodesk.com/ecotect-analysis`.

BEN-JOSEPH, E., ISHII, H., UNDERKOFFLER, J., PIPER, B., AND YEUNG, L. 2001. Urban simulation and the luminous planning table: Bridging the gap be-

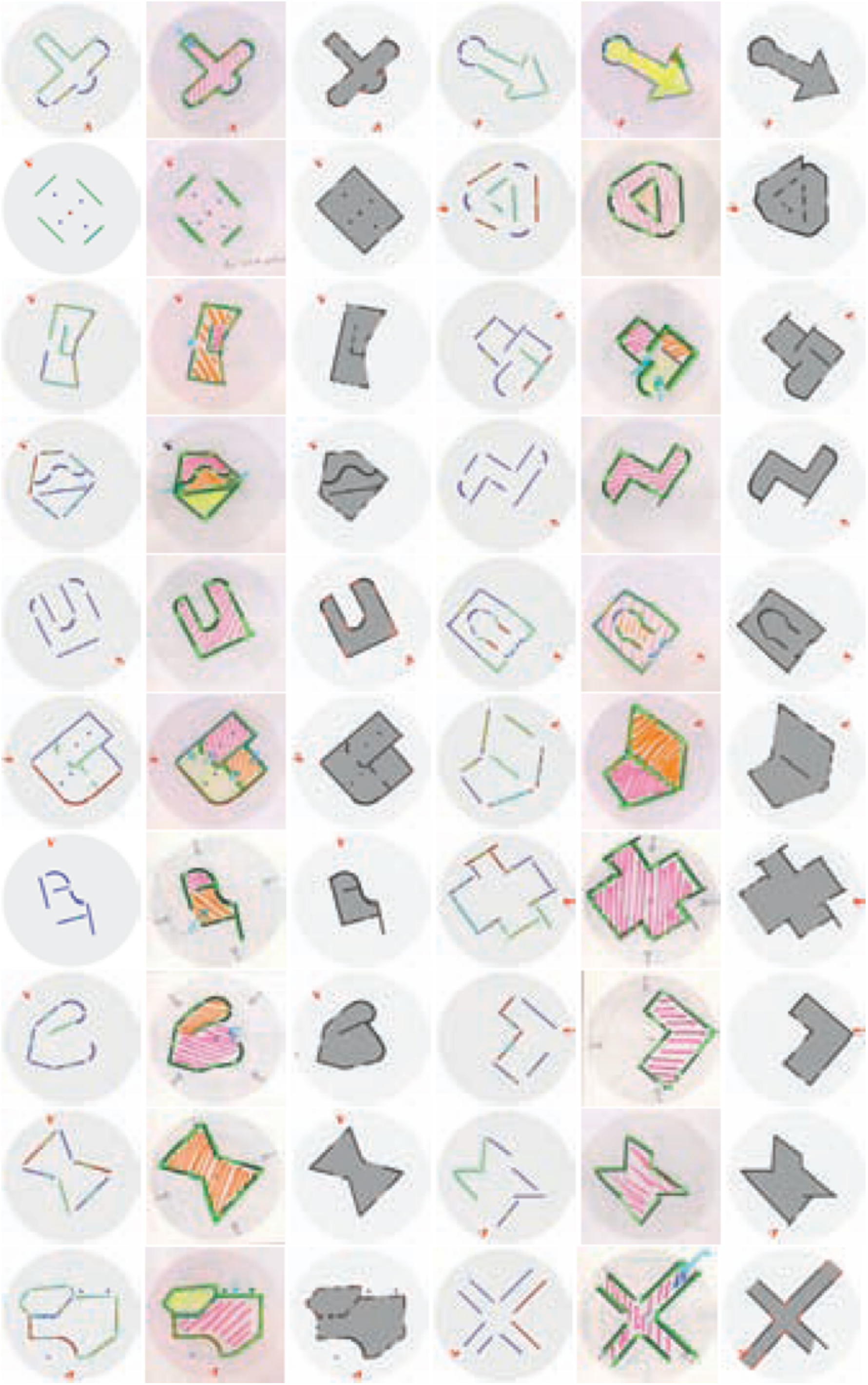


Figure 9: A sampling of the variety of collected physical sketch geometry: the original designer’s annotation, and our automated interpretation of the interior/exterior space.

tween the digital and the tangible. *Journal of Planning Education and Research*, 195–202.

DESOLNEUX, A., MOISAN, L., AND MICHEL MOREL, J. 2002. Computational gestalts and perception thresholds. *Journal of Physiology - Paris 97*, 2003.

DORSEY, J., XU, S., SMEDRESMAN, G., RUSHMEIER, H., AND MCMILLAN, L. 2007. The mental canvas: A tool for conceptual architectural design and analysis. In *15th Pacific Conference on Computer Graphics and Applications*, 201–210.

GOLOVINSKIY, A., PODOLAK, J., AND FUNKHOUSER, T. 2007. Symmetry-aware mesh processing. *Princeton University TR-782-07* (Apr.).

GOOGLE, 2010. SketchUp: 3D modeling software. `http://www.sketchup.com`.

GROSS, M. D., AND LUEN DO, E. Y. 2000. Drawing on the back of an envelope: a framework for interacting with application programs by freehand drawing. *Computers & Graphics 24*, 835–849.

GROSS, M. D. 1994. Recognizing and interpreting diagrams in design. In *AVI '94: Proceedings of the workshop on Advanced visual interfaces*, ACM, New York, NY, USA, 88–94.

HAMMOND, T., AND DAVIS, R. 2007. Ladder, a sketching language for user interface developers. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*, ACM, New York, NY, USA, 35.

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: A sketching interface for 3D freeform design. In *Proceedings of SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, 409–416.

ITTI, L., KOCH, C., AND NIEBUR, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 11 (Nov), 1254–1259.

KANIZSA, G. 1979. *Organization in Vision: Essays on Gestalt Perception*. Praeger.

KOCH, C., AND ULLMAN, S. 1985. Shifts in selective visual attention: Towards the underlying neural circuitry. *Human neurobiology 4*, 4, 219–227.

KOFFKA, K. 1935. *Principles of Gestalt Psychology*.

KOUTAMANIS, A., AND MITOSSI, V. 1993. Computer vision in architectural design. *Design Studies 14*, 1, 40 – 57.

KOUTAMANIS, A. 1999. A framework for architectural sketch recognition. In *4th International Design Thinking Research Symposium on Design Representation*.

KULIKOV, V. 2004. *Building Model Generation Project: Generating a Model of the MIT Campus Terrain*. Master's thesis, Massachusetts Institute of Technology.

LANK, E., THORLEY, J. S., AND CHEN, S. J.-S. 2000. An interactive system for recognizing hand drawn UML diagrams. In *CASCON '00: Proceedings of the 2000 Conference of the Centre for Advanced Studies on Collaborative Research*, IBM Press, 7.

LAWSON, B. 2002. CAD and creativity: Does the computer really help? *Leonardo 35*, 327–331.

LIPSON, H., , LIPSON, H., AND SHPITALNI, M. 2002. Correlation-based reconstruction of a 3D object from a single freehand sketch. In *AAAI Spring Symposium on Sketch Understanding*, 99–104.

LLADOS, J., LOPEZ-KRAHE, J., AND MARTI, E. 1997. A system to understand hand-drawn floor plans using subgraph isomorphism and Hough transform. *Machine Vision and Applications 10* (August), 150–158.

LU, T., TAI, C.-L., SU, F., AND CAI, S. 2005. A new recognition model for electronic architectural drawings. *Computer-Aided Design 37*, 10, 1053 – 1069.

LUEN DO, E. Y. 2001. VR Sketchpad, create instant 3D worlds by sketching on a transparent window. In *Proceedings of CAAD Futures 2001 (Eindhoven*, Kluwer Academic Publishers, 161–172.

MACKIE, C., COWDEN, J., BOWMAN, D., AND THABET, W. 2004. Desktop and immersive tools for residential home design. In *Conference on Construction Applications of Virtual Reality*.

PAULSON, B. 2010. *Rethinking Pen Input Interaction: Enabling Freehand Sketching Through Improved Primitive Recognition*. PhD thesis, Texas A&M University.

PAULY, M., MITRA, N. J., WALLNER, J., POTTMANN, H., AND GUIBAS, L. 2008. Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics 27*, 3, #43, 1–11.

RAMAGUPTA, A., AND HAMMOND, T., 2007. Archiassist: A sketch recognition system for floor plans, March. Poster Presentation at Texas A&M Industrial Affiliates Program.

SHENG, Y., YAPO, T. C., YOUNG, C., AND CUTLER, B. 2009. A spatially augmented reality sketching interface for architectural daylighting design. *IEEE Transactions on Visualization and Computer Graphics*, 20 (November).

SHENG, Y., YAPO, T. C., YOUNG, C., AND CUTLER, B. 2009. Virtual heliodon: Spatially augmented reality for architectural daylighting design. In *Proceedings of IEEE Virtual Reality 2009*.

TOLBA, O., DORSEY, J., AND MCMILLAN, L. 2001. A projective drawing system. In *ACM Symposium on Interactive 3D Graphics*, 25–34.

WACOM, 2010. Cintiq: Interactive Pen Display. `http://www.wacom.com/cintiq/`.

ZELEZNIK, R. C., HERNDON, K. P., AND HUGHES, J. F. 1996. Sketch: An interface for sketching 3D scenes. In *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, 163–170.

# Lamella Flock

**Martin Tamke**

CITA, Centre for Information Technology and Architecture,
Royal Danish Academy of Fine Arts, School of Architecture

**Jacob Riiber**

CITA, Centre for Information Technology and Architecture
Royal Danish Academy of Fine Arts, School of Architecture

**Hauke Jungjohann**

Knippers Helbig Advanced Engineering

**Mette Ramsgard Thomsen**

CITA, Centre for Information Technology and Architecture
Royal Danish Academy of Fine Arts, School of Architecture

**Abstract.** *The research project Lamella Flock questions how tectonic systems are usually formed and proposes self-organization as a means for future design. Lamella Flock investigates the possibility of designing as well as physically producing free-form interlinked structures based on multiple and circular dependencies.*

*The research takes its point of departure in the intersection between traditional wood craft, computer science, and a digital non-standardized production. Through computation and methods of self-organization the project investigates the structural abilities of the wooden Zollinger system; a structural lamella system distributed as a woven pattern of interconnected beams. By introducing an understanding of these beam elements as autonomous entities with sensory-motor behaviour the geometrically rigid Zollinger system is transformed into structures describing free-form surfaces.*

*By implementing dynamic processes, Finite Element Calculation, material and production constraints, and real-time interactive modelling in a hybrid environment Lamella Flock explores how to design and build with such a system. Hereby the agent system negotiates between design intent, tectonic needs, and production while creating a direct link between the speculative and its materialization.*

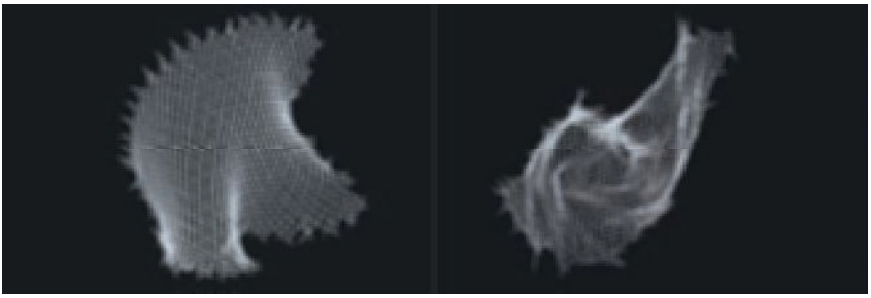


Figure 1. Renderings of two generated models with diverse characteristic

.

## 1 Non-Linear Processes in Architectural Design

During the last 20 years a new design practice has emerged in which architects become the developer of bespoke design environments that allow dynamic interfacing between design intention and contextual information [Kolarevic 2005, Shwitter 2005, Burry 2005]. This design practice has allowed for projects of high degrees of complexity that directly engage contexts such as day light [Whitehead 2005], spatial envelope [Goulthorpe 2008] or structure [Linsey 2001].

The organization of the related information is the domain of the building discipline. While the sources, nature and importance of these parameters are highly diverse the predominant strategy organizes them in carefully weighted linear flows. Parametric systems help to organize this flow of information, wherein one level of instructions is based on the previous.

This hierarchical approach allows for order and overview, yet has problems accounting for the complexity of design solutions that arise when multiple and highly interrelated parameters are incorporated.

We are surrounded by the success of this top-down strategy, as well as its failures. The downfalls have led to a wide interest in alternative design strategies such as bottom-up and performance based design methods [Kolarevic 2005]. Utilizing generative interactive procedures these methods relate to a notion of form-finding with direct interaction and feedback to the designer as in the practice of Gaudi, Frei Otto or Isler. Where these systems optimize mainly towards single goals (e.g. withstand gravity), computation allows optimization towards multiple goals of diverse nature. In this research we introduce the concept of an aware design model and ask how design can take place in environments that are characterized by multiple and circular dependencies governed by bottom-up principles?

## 2 Zollinger – A Lamella Wood System

The Zollinger construction is a type of lamella roof construction [Allen 1999] that was invented in the 1920s in order to create wide spanning constructions out of short pieces of timber (Figure 2 and 3).

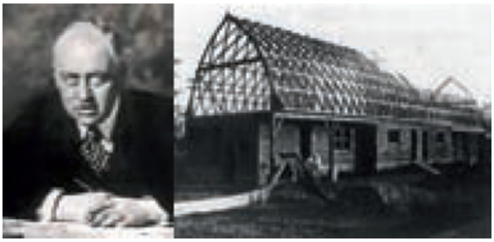


Figure 2 and 3. F. Zollinger and an original Zollinger Roof system in Merseburg, Germany.

The lamella's structural principle consists of a crisscrossing pattern of parallel arches of relatively short members. These are hinged together to form an interlocking network in a diamond pattern. The ingenuity resides within two constituents: Firstly, the efficient joint system that minimizes the amount of shared meeting points allowing for simple assembly, secondly, the structural strength given by the interwoven beams (Figure 4).

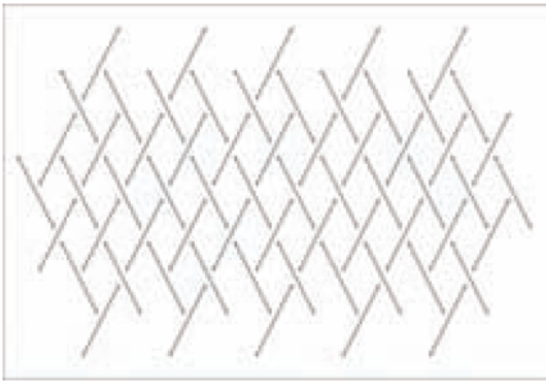


Figure 4. Principal pattern of the Zollinger lamella structure.

Where similar systems, such as reciprocal frame systems [Popovich 2008], usually form barrel or dome shapes, work from the AA [Hensel and Menges 2007], Shigeru Ban [Tristan, Self, and Bosia 2007] and Oliver Baverel [Popovich 2008] demonstrates the principal ability of the systems to form different shapes using the flex of the material, tolerances in the joint geometries and changes in beam orientation. In this bottom-up approach each element is threaded individually as it acts autonomously in a larger formation

Our own investigations revealed that freeform structures can be manually crafted from straight bamboo sticks by exploitation of tolerances in the joint. Yet this method relies purely on skill in crafting and negotiation with the physical model. The translation of this craft-based process into an architectural planning practice was a main concern of the investigation. This required the development of a parametric system that would allow control, anticipation and fabrication of geometry in relevant scale and tolerance. How could the non-linear relationships of the system be modelled within a computer?

## 3 Free Form Wood Structures and Previous Experience

Previous research on mass customized parametric wood constructions [Tamke, Thomsen, and Riiber 2008] indicated that digital production can provide the sought after flexible, effective fabrication of easily assembled wood beams. This approach is based on the conjunction of computation, digital fabrication, and traditional craft techniques. Herein modern CNC wood joinery machinery allows the cutting of monolithic joints in high speed and variable geometry (Figure 5). These joints allow for fast assembly as they incorporate self registering geometrical properties such as contemporary industrial snap fit joints [Schindler 2009]. The improved understanding of forces within massive wood, in its monolithic joints as well as in its assembly as structural systems through Finite Element systems [Holzner 1999] allows for new applications of traditional wood crafts. The combination of computational capabilities with digital fabrication allows therefore the introduction of craft related knowledge into contemporary practice that was previously bound to the skill and knowledge of the executing craftsperson.

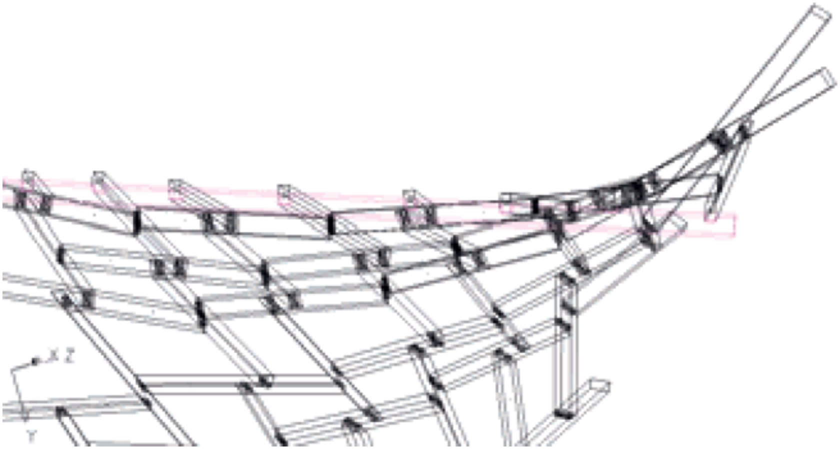


Figure 5. Wireframe view of the customized monolithic Tenon joints used in the construction.

## 4 Investigating Freeform Lamella Systems

In the initial stages of the research the distribution and computing of elements where investigated looking for the most suitable method of controlling the system and its non-linear relationships.

The lamella structure was at first distributed on pre-modelled surfaces. This presented two problems: When following a free-form surface all beam endpoints should be on the surface. Since all endpoints also connect to the midpoints of other beams this criterion cannot be met. Secondly this top-down approach lacked the possibility of exploring the performance of the structural principle. How would the rigidity of the reciprocal relationship between beams affect the scope of shapes possible?

The conclusion to use bottom-up approaches instead, gave at first problems in controlling the system. The elements were here structured through a rule based linear distribution where element were sequentially inserted. Due to the fact that in a networked lamella system one element is affecting all neighbours the linear distribution led to extreme and unpredictable conditions. This impeded design control but did result in compelling morphologies (Figure 6).

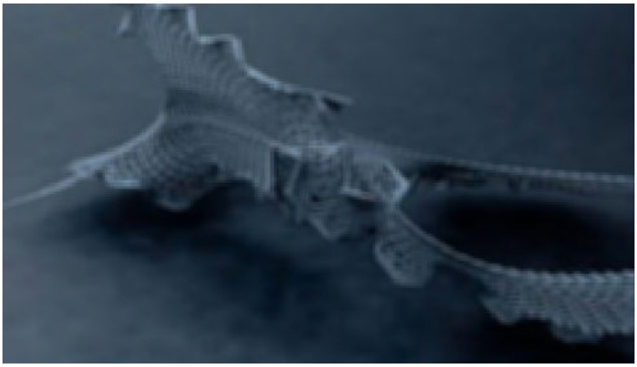


Figure 6. Output of a rule based distribution of lamella elements

The problems within the mentioned experiments made it possible to state the requirements of our lamella system: A bottom-up process with the ability of dynamic non-linear interaction where different design possibilities could be explored. We introduced an understanding of the structure as a self-organizing system of entities possessing a simple set of behavioural properties and relations to each other.

## 5 An Outline of Self-Organization

Theories of self-organization where originally developed in the context of physics and chemistry to describe the emergence of macroscopic patterns out of processes and interactions defined at a microscopic scale. Later it was found that these ideas could be extended to the simulation of social insects to show that complex behaviour may emerge from interactions among individuals that exhibit simple behaviour. Here social insect colonies where viewed as decentralized problem-solving systems, comprised of many relatively simple interacting entities [Bonabeau, Dorigo, and Theraulaz 1999].

This relies on the idea that a group of agents may be able to perform tasks without explicit representations of neither environment nor other agents, and where planning may be replaced by reactivity [Coates and Carranza 2000]. By re-contextualizing these abilities into numerous fields of knowledge powerful tools for developing dynamic and intelligent systems emerged.

The advantages of using self-organization to solve problems reside in a flexibility to function in changing environments and an ability to function even though some entities may fail to perform. The disadvantages can be located in the bottom-up approach to programming such systems. Here the paths to problem solving can never be predefined but are always emergent and result from interactions among entities themselves, as well as between entities and their environment. Therefore, using self-organization to solve a problem requires precise knowledge of both the individual behaviour of agents and what interactions are needed to produce a desired global effect [Bonabeau, Dorigo and Theraulaz 1999].

## 6 The generated lamella system, structure and behaviour

Our computer program is based on the interaction of four line segments coming together in a spiralling motion. In this way each entity exhibits within itself the non-linear relationship that also defines the global structure aimed at.

To initialize the program consists in determining the amount of entities, their sizes, and a preliminary distribution of these as a diagonal grid in space. The grid can either be coherent or fragmented depending on the desired modelling process. Positioning entities in space are in both cases done by either defining a distribution of point coordinates or loading a previously saved model into the system. This last feature allowed us to create models that could be evaluated through other tools and changed accordingly.

While running, the system is controlled through four behavioural algorithms that accumulate vector information (Figure 8). A method inspired by the division into goal types found in the simulation of flocks, herds and schools [Flake 1998]. Each algorithm produces directions and velocities that interact to produce the overall movement and transformation of an entity:

1. **Movement towards neighbours:** When not representing a corner or an edge each entity has four neighbours. By measuring the distance and direction from endpoints of line segments to a neighbour connection point, vectors are calculated. These vectors are added and weighted to calculate a mean vector by which all points in an entity are moved.
2. **Orienting towards neighbours:** By altering the configuration of angles

between segments each element tries to orient its segments towards their neighbours. A segment is in this way sought to be aligned with the trajectory towards its destination.

3. **Stretching towards neighbours:** Through the above orientation a segment will, within a certain tolerance, be able to stretch to connect to a neighbour. This is allowed when the orientation is correctly aligned and if it is happening within a predefined size limitation of a segment.
4. **Scale entity:** Each entity has the ability to scale up and down while keeping its proportions. This allows for a global push/pull effect within the lamella network.

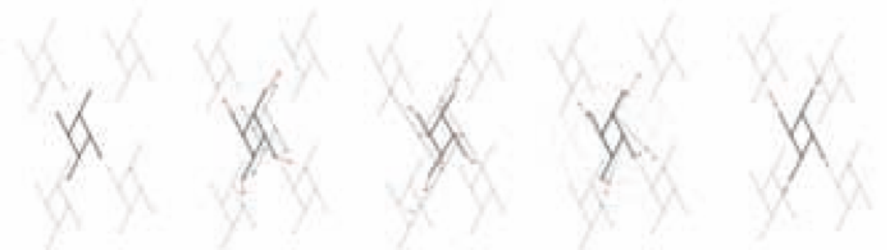


Figure 8. Initial state and the 4 main behavioural principles of Lamella Flock

Additionally production related constraints were introduced into the program. At the scale of the individual beam elements this means that the computation restricts the beam sizes and intersection angles to the specifications allowed by the machinery used for production. Also, the program is informed by the fact that two beams cannot share the same meeting point on a third beam, i.e. even though all non-edge beams will be connected to two other beams midway along their length, drilling for joints needs separate space for both connections. The program deals with this by slightly offsetting the shared meeting points of every element away from each other.

The generative design process was in this way informed by its implementation and realization in 1:1.

The global behaviour occurring from these functions produces a network of entities that attempts to obtain the shape of a surface. The global configuration is continuously and non-lineally renegotiated until a stable result is achieved.

## 7 A hybrid system

Experience [Tamke and Ramsgard 2009] has shown that in the context of architectural design a combination of generative and interactive modelling is practical. We introduced the possibility of manually manipulating entities while the system is running. This results in a tool where changes in the configuration of a surface can be made by altering local conditions, while self-organization deals with the global consequences of these actions (Figure 9).

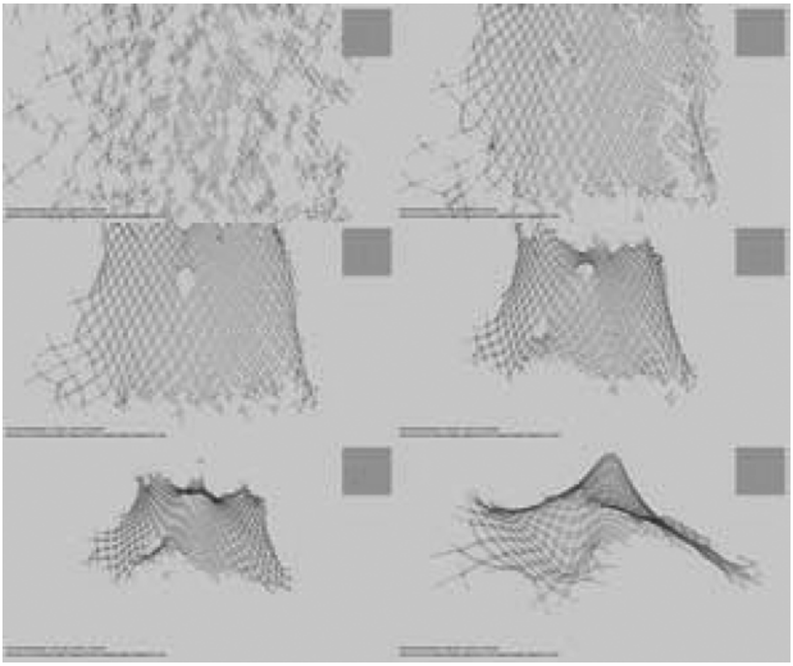


Figure 9. Formation and interaction of a lamella structure within the Processing interface

Actions include the ability to move an entity in any direction or change its scale, as well as fixate it in a given position. This last feature forces the surrounding network to adapt to the new conditions. Colour coding of elements and a navigational diagram helps to maintain an overview of these manipulations. Precision and localization of the design model where given through a millimetre based unit space and the ability to link in 3D models of the site (Figure 10).

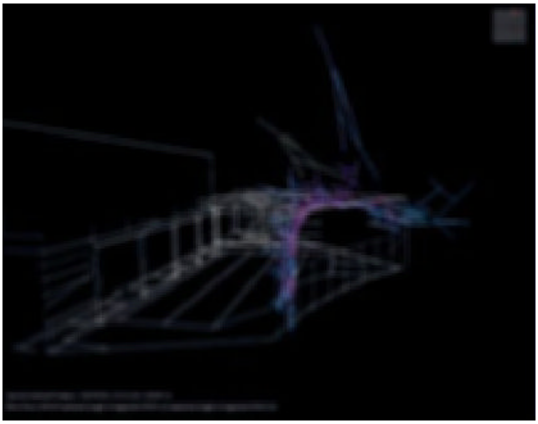


Figure 10. Processing interface with model of the ROM exhibition site

## 8 Implementation

The interface allows the model to interact dynamically with and inside an environment given by site, program, production and material. Changes to the environment through manipulation are instantly answered by the model through shape change. These transformations appear to the designer as a result of an internal reflection rather than direct answer. In this way designing starts by learning about the distinct character of the model and its behaviour.

The model exchanges through customized information transfer with different specialized tools: for structural FE-Analysis with Sofistik or for the generation of production data to Generative Components. The output can be adjusted to different model scales ranging from design speculation to 1:1 realization through machine code for Hundegger wood joinery machines (Figure 11, 12). Intense communication and testing through prototypes were crucial to determine the adequate types and dimensions of joints, fasteners, bearing and bracing for fabrication and assembly strategies.

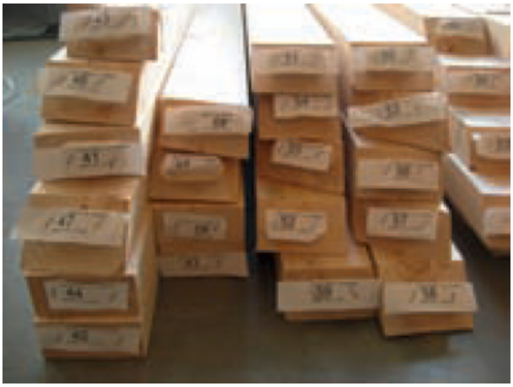


Figure 11. Labeled Non-standard wood beams ready for assembly

Feedback was integrated into the model which was becoming noticeably aware of its placement in the building process – its environment. The incoming information was handled in a pragmatic way where new insights were either encoded as internal conditions in the generative code or the visual interface was used for constraining the SO system.

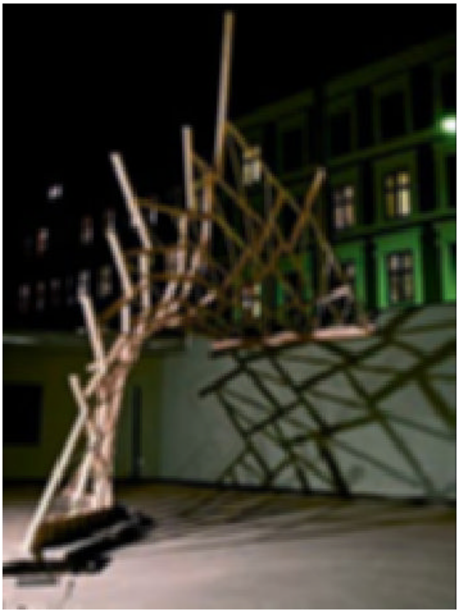


Figure 12. 1:1 Demonstrator at the ROM gallery Oslo / Norway

The intense preparation allowed us to exploit the capacity of digital fabrication and self registering joinery, demonstrated by only 3.2 hours of cutting time and two days of overall assembly of a structure consisting of 80 individualized beam elements.

## 9 Conclusion

The hybridization of generative processes and interactive modelling proposes a solution for integrating self-organization within architectural design and shows that non-linear systems can be used as a design tool. Here the different modelling methods are not mutually exclusive and work in parallel rather than in succession with individual strengths and weaknesses. Where programming is able to structure processes and relations that otherwise are beyond human capabilities specific design intentions are hard to test or change. Here manual interfacing rather than programming opens a space for design speculation where various constraints can be applied in an easy to handle fashion. Further research into the construction of customized user interfaces for hybrid dynamic-interactive processes might prove valuable for opening new territories for architectural design

The project shows that self-organization is capable of negotiating in an early architectural design context, characterized by interrelated requirements. It allows implementing global design intent as well as information regarding production, detail and material. The advantages of this are apparent in the speed and accuracy by which structures could be realized in 1:1 (Figure13). The open nature of the approach allows it to include an extended set of information, creating extended awareness of e.g. surroundings, gravity and tectonic stress.

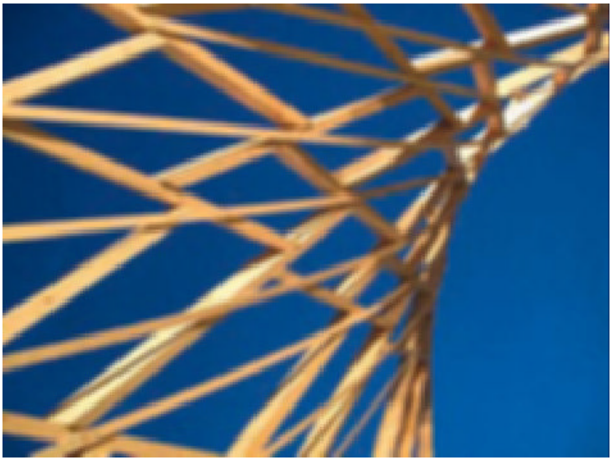


Figure 13. Detail of lamella demonstrator  in 1:1

## Acknowledgements

The Lamella system was only possible through support by HSB Systems, Hundegger GmbH, Trebyggeriet.no, Knippers and Helbig Advanced Engineering and Prof. Christoph Gengnagel/ TU-Berlin.

## References

ALLEN, J. S. 1999. *A Short History of 'Lamella' Roof Construction*. Transactions of the Newcomen Society, Vol 71, No 1.

BONABEAU, E., DORIGO, M., AND THERAULAZ, G. 1999. *Swarm Intelligence, From Natural to Artificial Systems*, New York: Oxford University Press.

BURRY, M. 2005, Between intuition and Process: Parametric Design and Rapid Prototyping. In *Architecture in the Digital Age: Design and Manufacturing*, ed. Kolarevic, B., Washington DC, Taylor & Francis.

CARRANZA, P.M., AND COATES, P. 2000, *Swarm modelling, The use of Swarm Intelligence to generate architectural form*, Proceedings of the International Conference on Generative Art

FLAKE, G. W. 1998. *The Computational Beauty of Nature: Computer Explorations of fractals, chaos, complex systems, and adaption.* Massachusetts: MIT Press.

GOULTHORPE, M. 2008, The Possibility of (an) Architecture. In *deCOI Collected Essays*, Routledge.

HENSEL, M., MENGES, A., 2007. *Morph-Ecologies: Towards Heterogeneous Space In Architecture Design*. Kfar Sava: AA Publications.

HOLZNER, H. 1999, *Entwicklung eines Nachweisverfahrens zur Bemessung von speziellen* (maschinell gefertigten) Zapfenverbindungen, Diplomthesis at Institut für Tragwerksbau - Fachgebiet Holzbau, TU Munich.

KOLAREVIC, B. K. 2005, Towards the performative in architecture. In *Performative Architecture*, ed. Kolarevic, B., Malkawi, A. Routledge Chapman & Hall, 205-211.

LINDSEY, B. 2001, *Digital Gehry* , Princeton, NJ: Princeton Architectural Press

POPOVICH, O. 2008. *Reciprocal Frame Architecture*. London: Architectural Press.

SCHINDLER, C. 2009. *Genagelt und geschraubt*.  ARCH+ 193, ARCH+ Verlag, Aachen. , p. 35.

SCHWITTER, C. 2005, Engineering Complexity: Performance-based Design in use. In *Performative Architecture – Beyond Instrumentality*, ed. Kolarevic, B., Malkawi, A., M., Spon Press.

TAMKE M. AND THOMSEN M. R. 2009, *Implementing digital crafting: developing: it's a SMALL world*. Berlin. Proceedings of the conference Design Modelling Symposium, Berlin, pp. 321-329

TAMKE, M., THOMSEN, M., AND RIIBER, J. 2008. *Complex Geometries in Wood*, Proceedings of the international conference Advances in Architectural Geometry, Technical University, Vienna.

TRISTAN, S., Martin, S., & Daniel, B. (2007). *Woven Surface and Form. Architextiles* , Architectural Design, Chichester: Academy Press. 82 – 89.

WHITEHEAD, H. 2005, Laws of Form in Architecture in the Digital Age. In *Design and Manufacturing*, ed. Kolarevic, B. Washington DC: Taylor & Francis. p. 123.

# Case Studies in Cost-Optimized Paneling of Architectural Freeform Surfaces

**Michael Eigensatz**

ETH Zurich, EPFL, and Evolute

**Mario Deuss**

EPFL

**Alexander Schiftner**

Evolute and TU Wien

**Martin Kilian**

Evolute and TU Wien

**Niloy J. Mitra**

KAUST and IIT Delhi

**Helmut Pottmann**

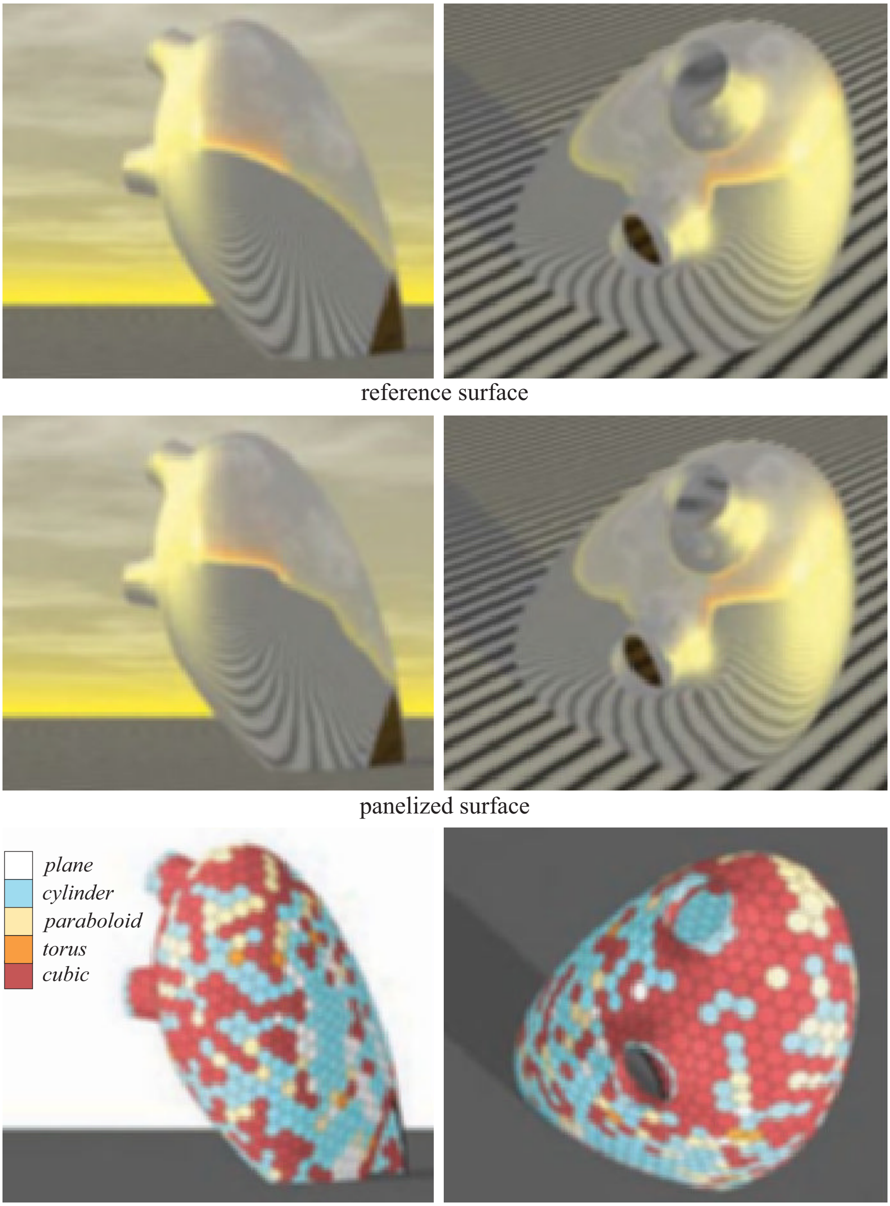
KAUST and TU Wien

**Mark Pauly**

EPFL

**Abstract.**

*Paneling an architectural freeform surface refers to an approximation of the design surface by a set of panels that can be manufactured using a selected technology at a reasonable cost, while respecting the design intent and achieving the desired aesthetic quality of panel layout and surface smoothness. Eigensatz and co-workers [Eigensatz et al. 2010] have recently introduced a computational solution to the paneling problem that allows handling large-scale freeform surfaces involving complex arrangements of thousands of panels. We extend this paneling algorithm to facilitate effective design exploration, in particular for local control of tolerance margins and the handling of sharp crease lines. We focus on the practical aspects relevant for the realization of large-scale freeform designs and evaluate the performance of the paneling algorithm with a number of case studies.*



reference surface

panelized surface

mold types

**Figure 1:** *Given a reference surface (top row), the* paneling algorithm *produces a rationalization of the the input. The paneling solution (middle row) employs a small set of molds that can be reused for cost-effective panel production (bottom row), while preserving surface smoothness and respecting the original design intent. The shown metal paneling solution is* 40% *cheaper than the production alternative of using custom molds for each individual panel. Figure 11 presents a variety of solutions that achieve cost savings of up to* 60%. *Figure 4 lists the metal cost ratios used.*

# 1 Introduction

Freeform shapes play an increasingly important role in contemporary architecture. Recent technological advances enable the large-scale production of single- and double-curved panels that allow panelizations of architectural freeform surfaces with superior inter-panel continuity compared to planar panels. However, the fabrication of curved panels incurs a higher cost depending on the complexity of the panel shapes, as well as on the employed material and panel manufacturing process (see Table 1). This gives rise to the so-called *paneling* task: The approximation of a design surface by a set of panels that can be manufactured using a selected technology at a reasonable cost, while respecting the design intent and achieving the desired aesthetic quality of panel layout and surface smoothness. The paneling task is a key component of the *rationalization* process for architectural freeform designs.

The challenge in paneling architectural freeform surfaces lies in the complex interplay of different objectives related to geometric, aesthetic, and fabrication constraints that need to be considered simultaneously. In this paper we discuss the paneling solution recently introduced in [Eigensatz et al. 2010], henceforth referred to as the *paneling algorithm*, and focus on the practical aspects relevant for the realization of large-scale freeform designs. We enhance the algorithm to handle spatially adaptive quality thresholds and propose an extension that allows incorporating sharp feature lines. With these new functionalities, the algorithm offers improved control for the architect to adapt the paneling according to the design specifications. We present three case studies to evaluate the performance of the paneling algorithm and provide insights into how the different parameter tradeoffs affect the quality of the results.

The rest of the paper is organized as follows: After discussing related work in the area of surface rationalization, we first classify different available panel types and fabrication processes (Table 1). We then formalize the paneling problem as stated in [Eigensatz et al. 2010] and review the main algorithmic contributions of their paneling solution. Section 4 presents our extensions to the existing formulation that allow processing freeform surfaces with sharp feature curves and enable local control of the paneling quality. In Section 5, we present three case studies to evaluate the performance of the algorithm, before concluding with a discussion of future research directions to address current limitations in Section 6.

### Related Work

A forward approach to surface rationalization is to use parametric design. An example for this was proposed by Glymph and coworkers [Glymph et al. 2002], where certain classes of surfaces are rationalized using planar quadrilateral panels. Parametric design is also available in many standard CAD tools nowadays. Such an approach introduces a logic into a geometric model by means of a generative sequence and relations between geometric objects. This logic helps in enabling simultaneous

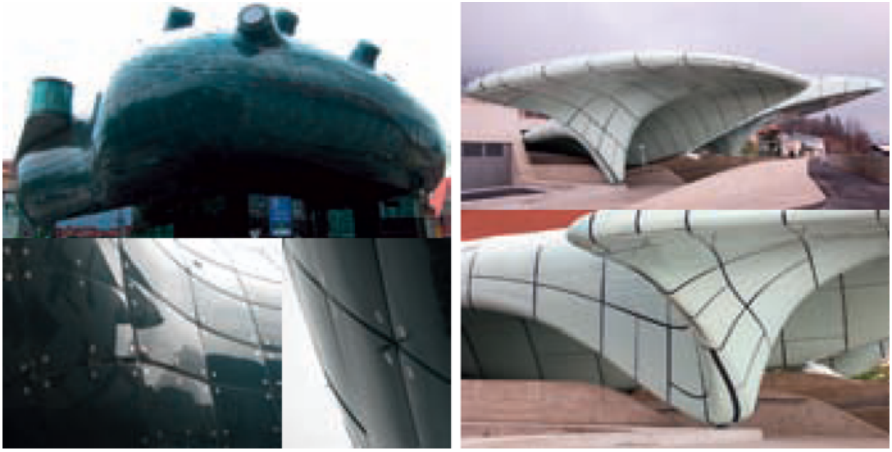


**Figure 2:** *Projects involving double-curved panels where a separate mold has been built for each panel. These examples illustrate the importance of the curve network and the existing difficulties in producing architectural freeform structures. (Left: Peter Cook and Colin Fournier, Kunsthaus, Graz. Right: Zaha Hadid Architects, Hungerburgbahn, Innsbruck.) Figure taken from [Eigensatz et al. 2010].*

control of the surface shape and the paneling layout. The simple causal chains inherent to parametric modeling, however, are insufficient for the rationalization of complex freeform geometries.

Other early contributions to the field of freeform architecture come from research at Gehry Technologies (see, e.g., [Shelden 2002]). These are mostly dedicated to developable or nearly developable surfaces, as a result of the specific design process that is based on digital reconstruction of models made from material that assumes (nearly) developable shapes. This approach is well suited for panels made of materials like sheet metal that may be deformed to developable or nearly developable shapes at reasonable cost. The approach is not sufficient, however, for panels made of materials like glass, for which the production processes limit shapes achievable at reasonable cost to very restricted classes of developable surfaces (see Table 1).

Most previous work on the paneling problem deals with planar panels. For various reasons, planar quadrilateral (quad) panels are preferred over triangular panels. Based on the theory of discrete differential geometry (see also [Bobenko and Suris 2008]), Pottmann and colleagues propose algorithms for covering general freeform surfaces with planar quad panels with new ways of supporting beam layout and for the related computation of multi-layer structures [Liu et al. 2006; Pottmann et al. 2007]. More recently, this approach was extended to the covering of freeform surfaces by single-curved panels arranged along surface strips [Pottmann et al. 2008b]. Figure 3 shows an example freeform surface rationalized using planar quads and developable strips, respectively. Additional results in this direction, e.g., hexago-

| **surface types** | **manufacturing possibilities** | | |
|---|---|---|---|
| | *glass* | *metal* | *fibre reinforced concrete/plastic* |
| **single curved**<br>isometric to the plane, no or little plastic deformation of material | | | |
| ***cylindrical***<br>parts of right circular cylinders | machine for bending and thermal tempering | roll bending machine | configurable mold or custom hot-wire cut foam mold |
| ***conical***<br>parts of right circular cones | configurable or custom mold, no thermal tempering | machine or reconfigurable mold | configurable mold or custom hot-wire cut foam mold |
| ***general single curved***<br>developable surfaces | custom mold, no thermal tempering | | custom hot-wire cut foam mold |
| **double curved**<br>usually plastic deformation of material is involved | | | |
| ***general double curved*** | custom molds, no thermal tempering of glass | machine or reconfigurable mold | custom molds commonly made of EPS foam |
| ***general ruled***<br>generated by a moving straight line | straight lines can be exploited | see above | foam molds can be hot-wire cut |
| ***translational***<br>carries two families of congruent profile | congruent profiles can be exploited | | congruent profiles can be exploited |
| ***rotational***, cf. Figure 6<br>carries one family of congr. profiles | | | |

**Table 1:** *Classification of panel types and state-of-the-art production processes for common materials in architecture. Although we do not cover all the relevant production processes, this table is for a rough guideline. Planar panels have been left out.*

nal meshes with planar faces, have been presented at “Advances in Architectural Geometry” [Pottmann et al. 2008a].

These approaches, however, focus on one specific type of panels (planar or developable) for rationalizing a given freeform surface, and do not explicitly consider the aesthetic quality of panel layout or surface smoothness. With these rationalization approaches it is difficult to freely choose the paneling seams, since they need to closely follow a so-called conjugate curve network on the given freeform surface, a notion that is defined by the curvature behavior of the surfaces (see [do Carmo 1976] and [Liu et al. 2006]).

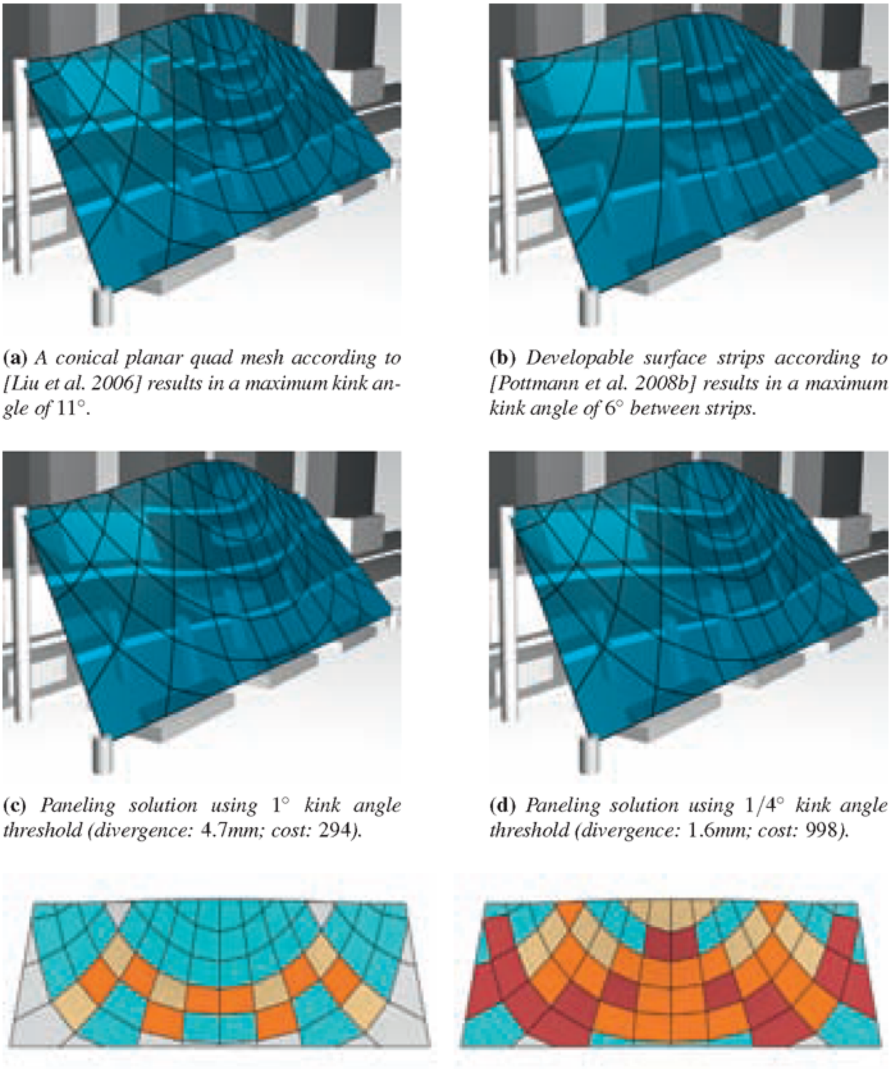
The optimization leading to a paneling solution is obtained by controlled deviation of the reference surface to increase the mold reuse. This is similar in spirit to symmetrization [Mitra et al. 2007; Golovinskiy et al. 2009] proposed to enhance object symmetry, i.e., repetitions, by controlled deformation of the underlying meshing structure.

## 2 Panels and Fabrication

Table 1 gives an overview of the state-of-the-art in architectural panel production. Curved panels are either produced using specially fabricated *molds* with the cost of mold fabrication often dominating the panel cost, or the panels require unique machine configurations, which drive cost by means of machining time. There is thus a strong incentive to reuse the same mold or machine configuration for the production of multiple panels to reduce the overall cost. In the following we use the term *mold* to also refer to machine configuration.

The choice of panel types depends on the desired material and on the available manufacturing technology. The paneling algorithm does not depend on materials: they may be transparent or opaque, include glass, glass-fibre reinforced concrete or gypsum, metal, wood, etc. Currently the algorithm supports five panel types that possess different cost to quality tradeoffs: planes, cylinders, paraboloids, torus patches, and general cubic patches (see Figure 4). If these types cannot approximate a surface segment within the required tolerances, a custom general double curved panel is used.

Planar panels are easiest to produce, but result in a faceted appearance when approximating curved freeform surfaces, which may not satisfy the aesthetic criteria of the design. A simple class of curved panels are cylinders, a special case of single-curved (developable) panels. Naturally, such panels can lead to a smooth appearance only if the given reference surface exhibits one low principal curvature. General free-form surfaces often require double-curved panels to achieve desired quality specifications prescribed in terms of tolerances in divergence and kink angles (see Section 3 for details). The paneling algorithm currently supports three instances of such panels: paraboloids, torus patches, and cubic patches. Paraboloids and tori are important



**(a)** *A conical planar quad mesh according to [Liu et al. 2006] results in a maximum kink angle of* $11^\circ$.

**(b)** *Developable surface strips according to [Pottmann et al. 2008b] results in a maximum kink angle of* $6^\circ$ *between strips.*

**(c)** *Paneling solution using* $1^\circ$ *kink angle threshold (divergence: 4.7mm; cost: 294).*

**(d)** *Paneling solution using* $1/4^\circ$ *kink angle threshold (divergence: 1.6mm; cost: 998).*

**(e)** *Panels colored by type of corresponding mold.*

**(f)** *Panels colored by type of corresponding mold.*

**Figure 3:** *Comparison with state-of-the-art rationalization algorithms on a freeform facade design study. (a, b) Rationalization using a planar quad mesh and developable surface strips, respectively. (c-f) Rationalization using the paneling algorithm with* $1^\circ$ *and* $1/4^\circ$ *kink angle thresholds, shown along with visualization of respective mold types (using glass cost ratios listed in Figure 4). A detailed overview of mold reuse for (e) is shown in Figure 8.*

*Panel types*

| | | | |
|---|---|---|---|
| *plane* | ⬜ | 🟧 | *torus* |
| *cylinder* | 🟦 | 🟥 | *cubic* |
| *paraboloid* | 🟨 | ⬛ | *custom* |

*Costs per mold and per panel*

| | *glass* | | | | | | *metal* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ⬜ | 🟦 | 🟨 | 🟧 | 🟥 | ⬛ | ⬜ | 🟦 | 🟨 | 🟧 | 🟥 | ⬛ |
| *mold* | - | 2 | 18 | 24 | 30 | - | - | 3 | 6 | 6 | 6 | - |
| *panel* | 1 | 2 | 5 | 5 | 5 | 35 | 1 | 3 | 6 | 6 | 6 | 12 |

**Figure 4:** *The panel types currently supported by our algorithm and two typical cost sets.*

because they are special classes of translational and rotational surfaces and carry families of congruent profiles (parabolae and circles, respectively). This typically simplifies mold production (see Table 1 and Figure 6). Although cubic panels do not have any such advantage for manufacturing, they offer the highest flexibility and approximation power. Thus a small number of cubic or more general double-curved molds are often indispensable to achieve a reasonable quality-cost tradeoff.

Mold reuse is a critical cost saving factor. In order to compute paneling solutions with mold reuse in reasonable time one needs to restrict the search space and parameterize panel types using a few parameters only. The paneling algorithm, therefore, uses the restricted panel types paraboloids, tori and cubics instead of the much more general translational, rotational and general double-curved surfaces. Paraboloid, torus, and cubic are defined by 2, 3 and 6 shape parameters, respectively (please refer to [Eigensatz et al. 2010] for details). In Section 6 we discuss the possibility of adding other panel types.

## 3 Paneling Architectural Freeform Surfaces

We review both the specification of the paneling problem and the optimization approach presented by Eigensatz and coworkers. For a more detailed description, in particular with respect to mathematical and algorithmic aspects, we refer the reader to [Eigensatz et al. 2010].

### 3.1 Problem Specification

Let $F$ be a given input freeform surface, called *reference surface*, describing the shape of the design. The goal is to find a collection of *panels*, such that their union approximates the reference surface. Since the quality of the approximation strongly depends on the position and tangent continuity across panel boundaries, Eigensatz and coworkers identify two quality measures (see Figure 5):

- **divergence:** quantifies the spatial gap between adjacent panels and,
- **kink angle:** measures the jump in normal vectors between adjacent panels.

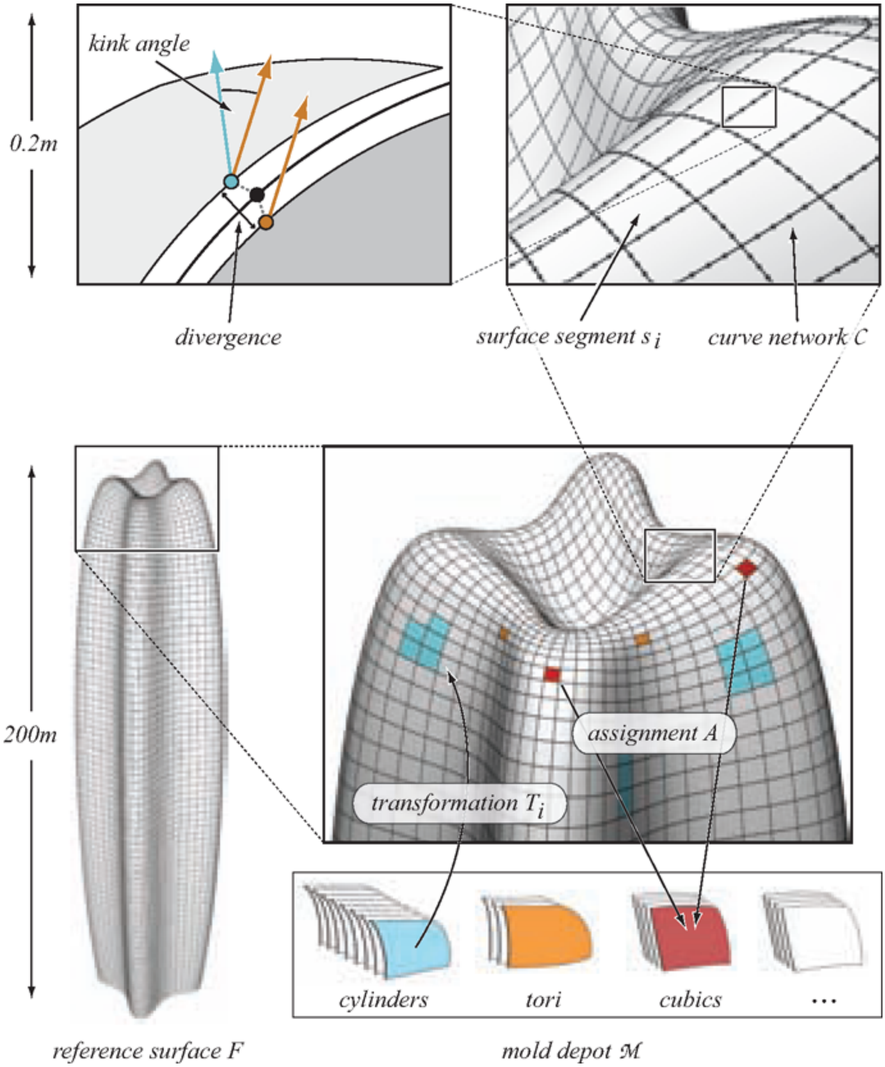


**Figure 5:** *Terminology and variables used in the paneling algorithm. The reference surface $F$ and the initial curve network $\mathcal{C}$ are given as part of the design specification. The optimization solves for the mold depot $\mathcal{M}$, the panel-mold assignment function $A$, the shape parameters of the molds, the alignment transformations $T_i$, and the curve network's normal displacement. Figure taken from [Eigensatz et al. 2010].*

While divergence is strongly related to the viability of a paneling solution, the kink angles influence the visual appearance, since they are related to reflections. Hence one can allow higher kink angles in areas not or only barely visible to an observer. We will elaborate on this possibility in Sections 4.2 and 5.2.

The intersection curves between adjacent panels are essential for the visual appearance of many designs (see Figure 2) and typically reflect the structure of the building, as they often directly relate to the underlying support structure. An initial layout of these curves is usually provided by the architect or engineer as an integral part of the design. While small deviations are typically acceptable in order to improve the paneling quality, the final solution should stay faithful to the initial curve layout and reproduce the given pattern as good as possible by the intersection lines of adjacent panels. The collection of all panel boundary curves (strictly speaking panel intersection curves) forms the *curve network*, which splits the given input freeform surface into *segments*. Each segment, in general polygonal, of the curve network has to be covered by a panel.

The *paneling problem* is formulated as follows: Approximate a given free-form surface $F$ by a collection of panels of selected types such that pre-defined thresholds on divergence and kink angle are respected, the initial curve network is reproduced as good as possible, and the total production cost is minimized. The production cost of a panelization comprises the following terms: the production cost of each employed mold and the cost of producing each panel from its assigned mold (see Figure 4 for two typical cost sets and Figure 8 for an illustration).

### 3.2 Paneling Algorithm

A paneling solution can be computed using the optimization algorithm described in [Eigensatz et al. 2010]. This algorithm takes as input the reference surface $F$, the initial curve network, and global thresholds on maximal kink angle and divergence, along with a permitted deviation margin of the final paneled surface from the reference surface. As output, the algorithm computes the parameters that determine the shape of the fabrication molds and the alignment transformations that position the panels in space. These parameters are computed in such a way that the reference surface is approximated as good as possible, while the kink angle and divergence thresholds are satisfied everywhere. At the same time, the cost of fabrication is minimized by favoring panels that are geometrically simple and thus cheaper to manufacture wherever possible, and maximizing the amount of mold reuse.

In order to achieve these conflicting goals, the paneling optimization is formulated as a mixed discrete/continuous optimization that simultaneously explores many different paneling solutions (see [Eigensatz et al. 2010] for details). From all these different alternatives, the solution of minimal overall fabrication cost is selected that satisfies the kink angle and divergence thresholds. An essential ingredient in this optimization is controlled deviation of the paneling from the initial design surface. By allowing the curve network to move away from the reference surface, panels can fit

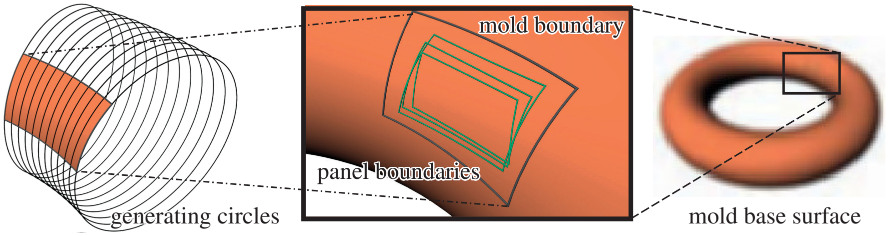


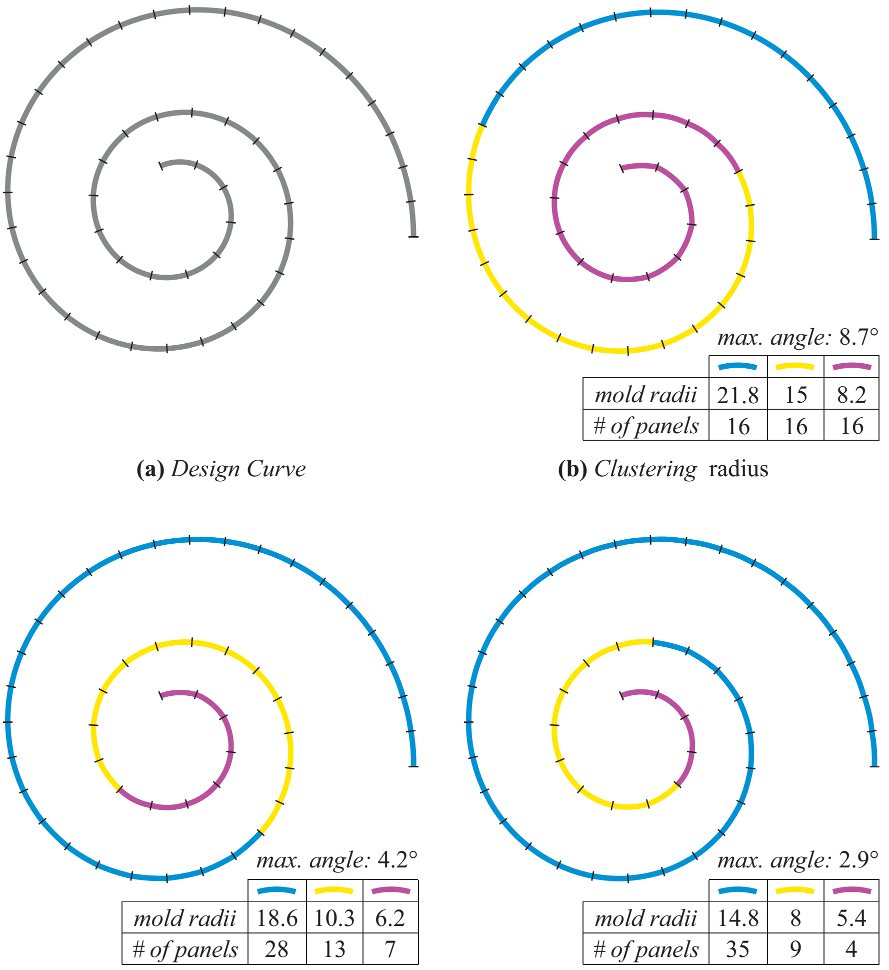
**Figure 6:** *Example of mold reuse. Panel boundary curves are in general not congruent. However, several panels may be closely grouped together on the same mold base surface. In that case the same mold or machine configuration, which embraces all affected panels, may be used to manufacture the panels. This figure further illustrates how the congruent profiles of a rotational or translational surface, in this case the circles generating a torus, can be exploited for mold fabrication.*

together with smaller kink angles and divergence, simpler and thus cheaper panels can be used in certain regions, and the amount of reuse of molds can be increased. Figure 7 demonstrates the effectiveness of the discrete optimization presented by [Eigensatz et al. 2010] on an illustrative example, comparing different techniques to enable mold reuse.

The results shown in [Eigensatz et al. 2010] include solutions to the paneling problem for large-scale architectural freeform designs that often consist of thousands of panels. Typically, these paneling solutions consist of patches of flat, single and double curved panels as shown in Figure 3, therefore partly generalizing the approaches introduced in [Liu et al. 2006] and [Pottmann et al. 2008b] to include double curved panels. The main innovations of the paneling algorithm can be summarized as follows:

- Given a table of mold and panel production costs, the paneling algorithm computes a panelization with minimal cost while meeting predefined quality requirements.
- The algorithm is adaptable to numerous production processes and materials.
- The possibility to explore diverse quality requirements and cost tables provides valuable information to guide design decisions.
- The rationalized 3D models produced by the algorithm may be used for visual inspection, prototype panel manufacturing, quality control, and the final production of freeform surfaces.
- Interference with the architects design intent is minimized.

The original paneling algorithm provides a general framework and is extensible in various ways. We propose and investigate two specific extensions in Section 4 and discuss further extension possibilities in Section 6.



**(a)** *Design Curve*

*max. angle:* 8.7°

| | 🟦 | 🟨 | 🟪 |
|---|---|---|---|
| *mold radii* | 21.8 | 15 | 8.2 |
| *# of panels* | 16 | 16 | 16 |

**(b)** *Clustering* radius

*max. angle:* 4.2°

| | 🟦 | 🟨 | 🟪 |
|---|---|---|---|
| *mold radii* | 18.6 | 10.3 | 6.2 |
| *# of panels* | 28 | 13 | 7 |

**(c)** *Clustering* (1 / radius)

*max. angle:* 2.9°

| | 🟦 | 🟨 | 🟪 |
|---|---|---|---|
| *mold radii* | 14.8 | 8 | 5.4 |
| *# of panels* | 35 | 9 | 4 |

**(d)** *Discrete Optimization*

**Figure 7:** *Illustrative comparison of different techniques for mold reuse. The curve should be approximated with circle arcs of varying radii. This can be understood as a simple paneling with cylinders of varying radii, where the figure shows an orthogonal cross section. The input design curve shown in* ***(a)*** *consists of nicely aligned circle arcs with decreasing radii from* 25 *to* 5*. The method shown in* ***(b)*** *clusters these radii (using k-means clustering) to obtain* 3 *molds and assigns the best mold to each segment. The colors indicate the segments sharing the same mold. The method shown in* ***(c)*** *does the same, but performs a clustering of* (1/*radius*) *instead of clustering the radius itself, which is a much better distance approximation for cylinders as shown in [Eigensatz et al. 2010] and therefore the maximal kink angle is already much lower compared to* ***(b)****. The method shown in* ***(d)*** *performs the full discrete optimization presented in [Eigensatz et al. 2010] and leads to an even better mold depot that enables a paneling with only* 3 *molds but very low kink angles. The differences presented on this schematic example become even more prominent if more complex surfaces and/or panel types are involved.*

# 4 Extensions

In this section we discuss algorithmic extensions to the method of Eigensatz and coworkers [Eigensatz et al. 2010] that broaden its applicability.

## 4.1 Sharp Features

The algorithm introduced by Eigensatz and coworkers assumes that the input reference surface is smooth everywhere. Sharp feature lines, however, are used in architectural freeform designs to highlight strong characteristic features and to enhance the visual appeal of a design. We therefore propose an extension of the paneling algorithm to incorporate sharp features.

| *mold type*<br>*cost* | **plane** | **cylinder** | **parab.** | **torus** |
|---|---|---|---|---|
| cost per **mold** dependent on type | plane 1 | cylinder 1<br>cylinder 2<br>cylinder 3<br>cylinder 4-6<br>cylinder 7-8 | paraboloid 1-3 | torus 1<br>torus 2 |
| cost per **panel** dependent on type | 18 panels | 16 panels<br>8 panels<br>6 panels<br>4 panels each<br>1 panel each | 2 panels each | 6 panels<br>2 panels |

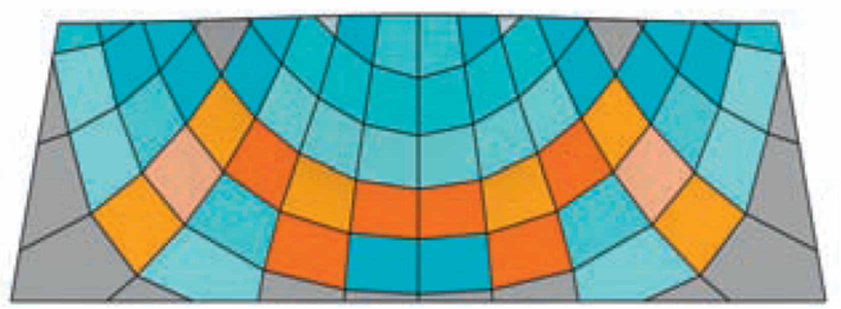


**Figure 8:** *Illustration of the mold depot and the cost model by means of the example shown in Figure 3(e). The colors of panels are saturated according to mold reuse. Figure 4 lists the glass cost ratios used for this example.*

Sharp feature lines can either be specified by the designer as specially marked lines of the initial curve network, or automatically computed by detecting sharp creases on the design surface. To support sharp features we adapt the original paneling algorithm such that

- kink angle thresholds are not applied along the curves describing sharp features and
- the tangent continuity between two panels on opposite sides of a sharp feature is not optimized.

Figure 14 demonstrates how this extension enables paneling freeform surfaces with sharp features.

### 4.2 Adaptive Control of Paneling Quality

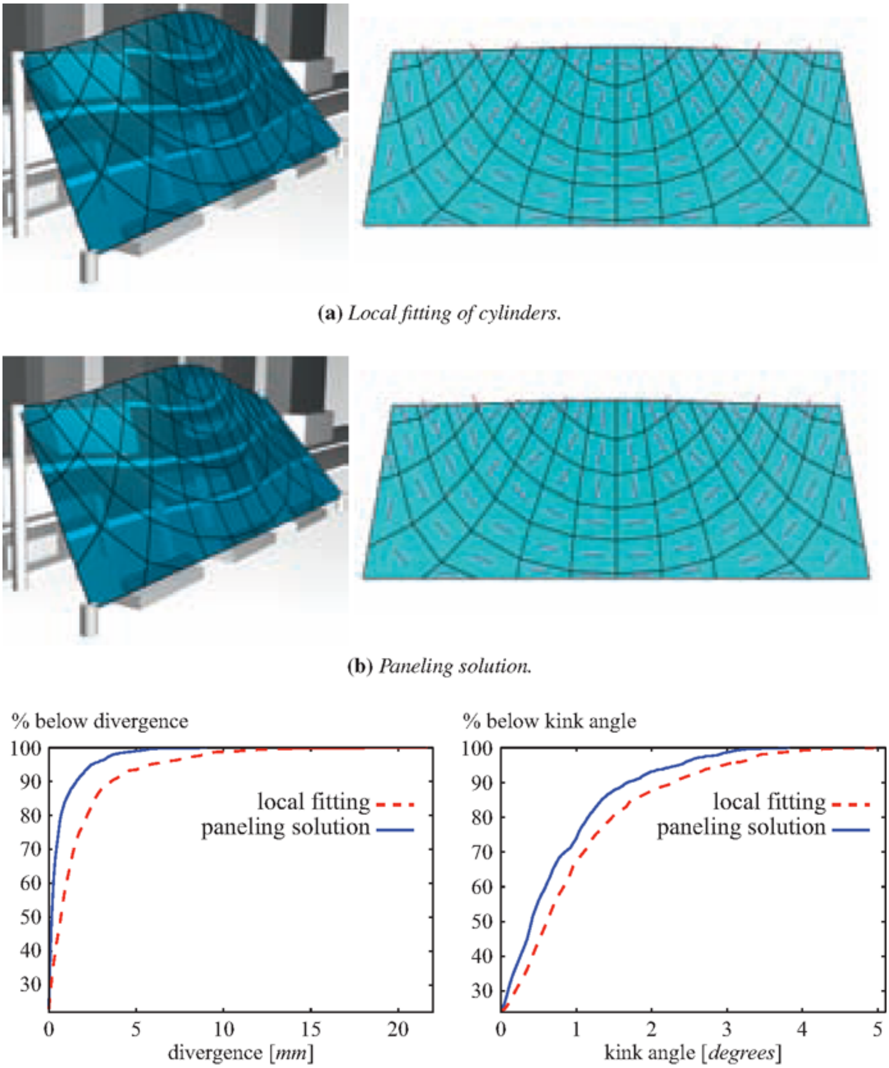
The paneling algorithm introduced in [Eigensatz et al. 2010] guarantees compliance with user-specified tolerance thresholds on divergence and kink angle. These thresholds are specified globally for the entire surface. In practice, however, the quality requirements might vary for different regions of the design. For regions not visible from certain view-points, for example, higher kink angles might be acceptable to reduce manufacturing cost. We therefore extend the original paneling algorithm to optimize the paneling quality with respect to a spatially adaptive importance function on the design surface.

As shown in Figure 10 this importance function can, for example, be computed using a visibility calculation that computes the visibility for every point on the design surface, if the design is viewed from a path or street around the building. This importance function is then an additional input to our extended paneling algorithm to

- adaptively specify a separate kink angle threshold for every point on the curve network and
- focus the tangent continuity optimization on important regions.

Figures 10-13 demonstrate how this adaptive quality control directs the use of expensive panels towards regions where they are needed most, leading to an improved paneling quality at similar or lower costs compared to globally specifying thresholds. Achieving the same quality at the important regions with the original paneling algorithm using global thresholds requires a much more expensive paneling.

The same technique can be used to adaptively control the divergence or the deviation from the original design surface.



**(a)** *Local fitting of cylinders.*

**(b)** *Paneling solution.*

**(c)** *Cumulative histograms of divergence and kink angles for the above solutions.*

**Figure 9:** *The paneling algorithm restricted to cylindrical panels. Here we compare a result on the Facade Design Study computed using simple local fitting of cylinders (a) to a paneling solution using only cylinders (b). For both results we show the axis directions of cylinders colored in magenta and the cumulative histograms of resulting divergences and kink angles.*

# 5 Case Studies

In this section we demonstrate the performance of the paneling algorithm on three case studies. Specifically we compare our solutions with state-of-the-art rationalization alternatives, study the preservation of sharp features, and compare the cost trade-offs for global kink angle specifications versus spatially adapted ones.

## 5.1 Facade Design Study

We compare several rationalization possibilities for a freeform facade. For this case study we use glass mold cost ratios as listed in Figure 4.

Figure 3a shows a rationalization result using a conical planar quad mesh, which implies very favorable properties for simplifying the substructure, cf. [Liu et al. 2006; Pottmann et al. 2007]. Naturally this approach leads to a facetted result with kink angles up to 11°. A further option makes use of the close relation between planar quad meshes and developable strip models ([Pottmann et al. 2008b]): Refining the planar quad mesh in one direction and keeping the faces planar leads to a rationalization using single-curved strips. Clearly this results in a much smoother representation of the surface as can be seen in Figure 3b (maximum 6° kink angle), while one could still make use of a planar quad mesh for the substructure. The deformation of glass to general single-curved panels, however, requires molds to be built, a possibility that was ruled out because of budgetary issues. Therefore the paneling algorithm was used to proof feasibility for the competition, making use of cylindrical panels only. The superiority of such a restricted paneling solution to results that are achievable using local fitting of cylinders is documented in Figure 9. Figure 3 compares further paneling solutions with respect to cost and paneling quality, making use of the complete set of mold types.

## 5.2 Skipper Library

Initially issued by Texxus, the skipper library is a feasibility study also picked up by Formtexx for freeform metal cladding. The case study demonstrates our extension of the paneling algorithm allowing adaptive control of the paneling quality, as well as the ability of the paneling algorithm to handle arbitrary panel layouts. The presented panel layout was created using the dual mesh of a circle packing mesh (cf. [Schiftner et al. 2009]), which leads to a panel layout consisting mainly of hexagonal panels combined with a torsion free support structure. Our motivation to adaptively control the paneling quality is given by the following:

Due to various constraints imposed by surrounding buildings, restricted access paths, neighboring trees and foliage, different sections of architectural buildings have different visibility. This can be exploited to reduce the manufacturing cost of such buildings by allowing larger kink angles in less visible regions. As described in

Section 4, we generalize the paneling algorithm proposed in [Eigensatz et al. 2010] to allow spatially variable kink angle specifications as opposed to a global maximum kink angle threshold. Figures 10-13 compare the results on manufacturing cost for a global threshold versus two spatially adapted threshold specifications. The local importance functions are computed based on visibility of the reference surface when moving along the specified access paths (see Figure 10). For this case study we use metal mold cost ratios as listed in Figure 4. The middle row in Figure 1 shows a paneling solution with 1° global kink angle threshold.

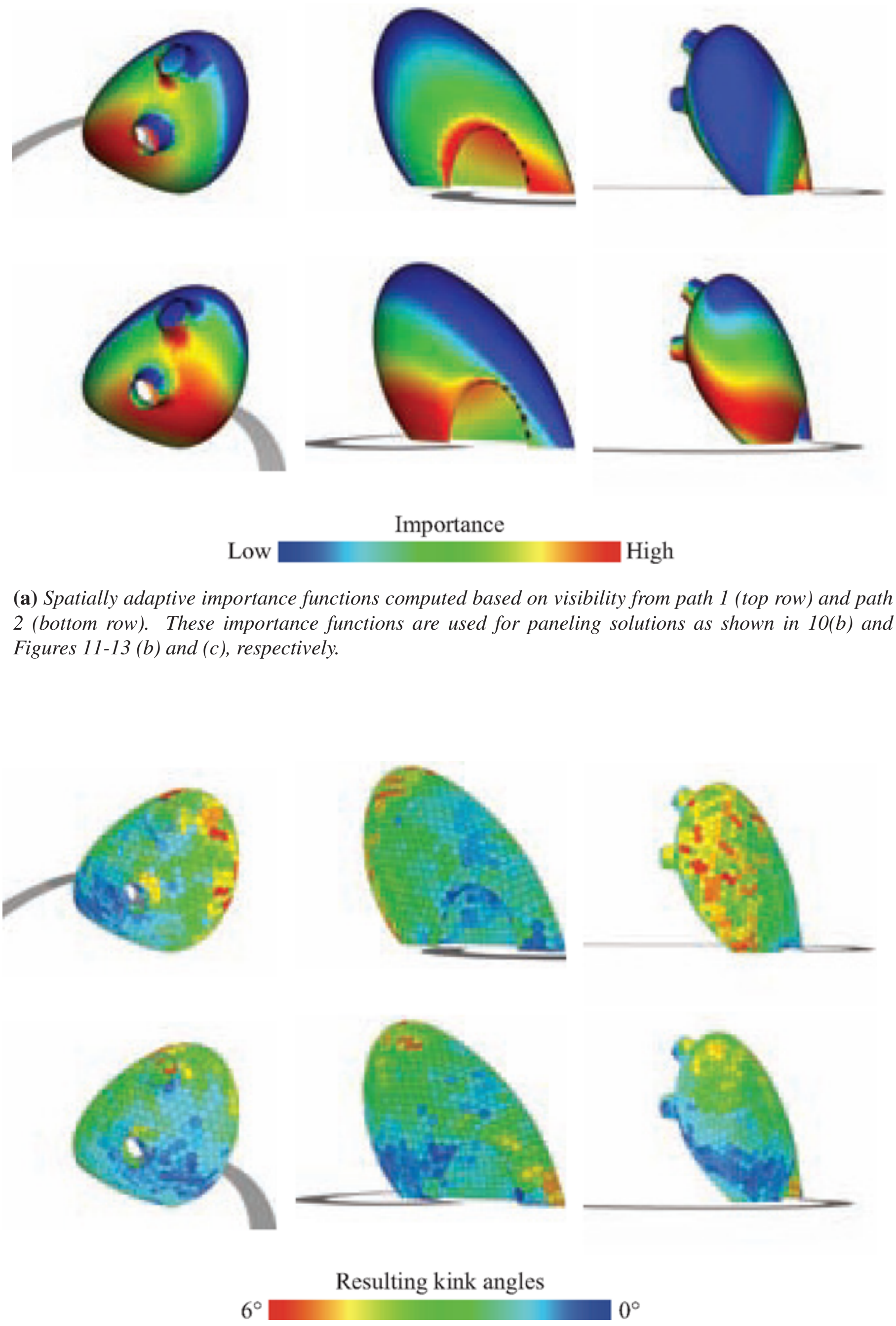
### 5.3 Lissajous Tower

Lissajous Tower is an example skyscraper specifically created for illustrating our extension to the paneling algorithm for handling sharp features. The surface contains large nearly flat and single-curved parts as well as small highly curved parts, which can not be approximated by cylinders within realistic tolerances. Figure 14 compares two paneling solutions produced by the paneling algorithm with maximum kink angle thresholds of 1° and 3°, respectively. While both solutions preserve the characteristic sharp feature line of the design, the production cost is significantly reduced (by 40%) for a slight relaxation in the maximum kink angle constraint. For this case study we use glass mold cost ratios as listed in Figure 4.

## 6 Discussion

*Limitations.* The input to the paneling algorithm is a design surface and a set of curves (panelization seams) that define how the surface is divided into panels. We consider both the surface and the panelization seams as design intent and thus aim to change them as little as possible. This approach leads to the following implications:

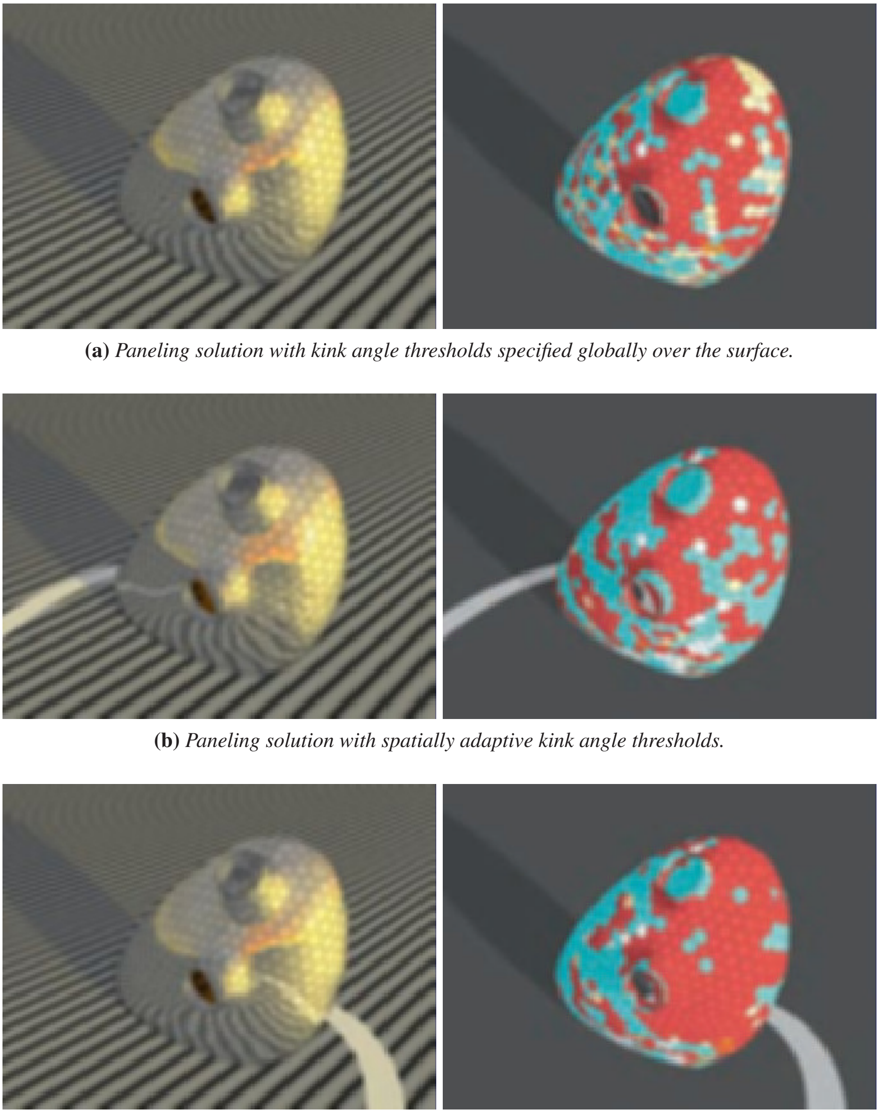
- If design surface or seams inherently violate the limits of a certain material or production process, for example with respect to maximum panel sizes, then the paneling algorithm will not eliminate this.
- When computing minimum cost solutions the paneling algorithm cares about cost of panel production only. This is reasonable because it just minimally changes the design surface and panelization seams, and therefore does not influence the cost of further parts like the substructure.



**(a)** *Spatially adaptive importance functions computed based on visibility from path 1 (top row) and path 2 (bottom row). These importance functions are used for paneling solutions as shown in 10(b) and Figures 11-13 (b) and (c), respectively.*

**(b)** *Kink angles of two paneling solutions (top and bottom rows) using adaptive thresholds based on the two importance functions shown in 10(a). Further renderings of the results are shown in Figures 11-13.*

**Figure 10:** *Adaptive quality control.*

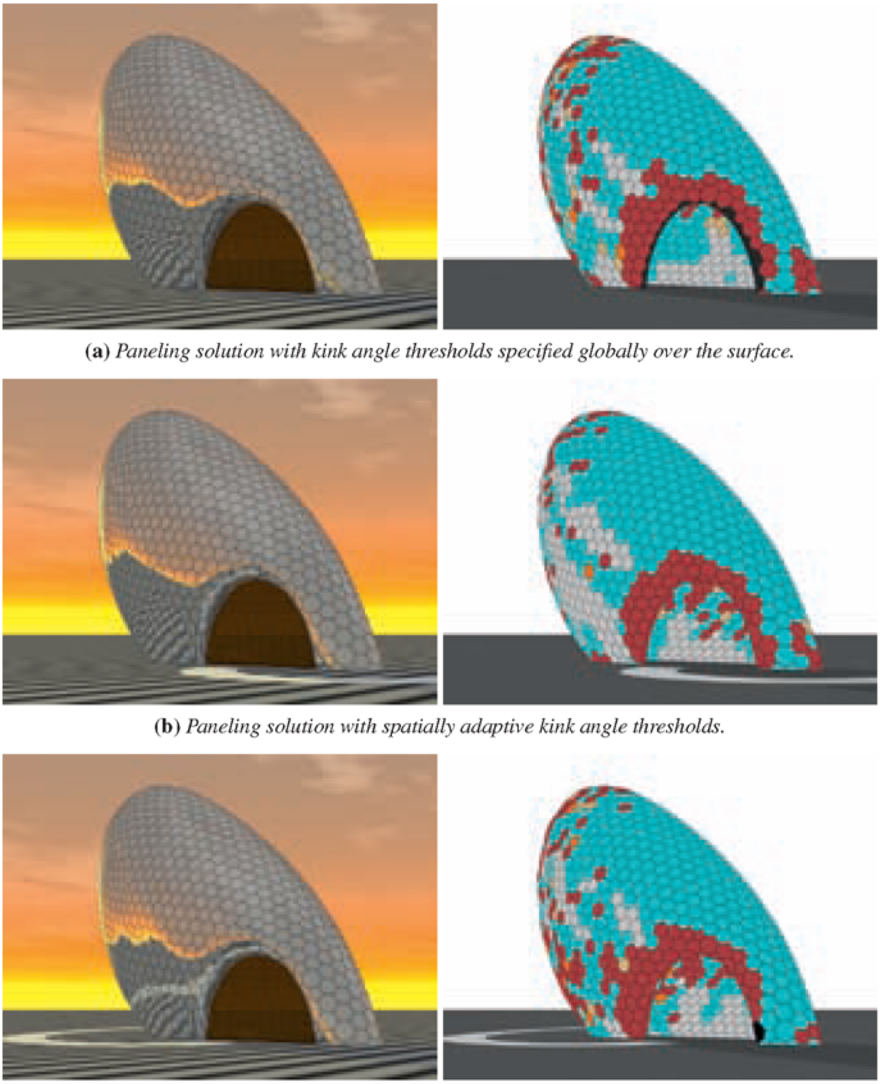


**(a)** *Paneling solution with kink angle thresholds specified globally over the surface.*

**(b)** *Paneling solution with spatially adaptive kink angle thresholds.*

**(c)** *Paneling solution with another set of spatially adaptive kink angle thresholds.*

**Figure 11:** *Effect of global vs spatially varying kink angle specifications on the Skipper Library dataset. Paneling solutions using a global kink angle specification (a) and using adaptive kink angle thresholds computed based on the extent of visibility while moving along the indicated ground paths (b, c). Left column images show the reflection lines on paneled surfaces, while right column images show the mold types for individual panels (color convention same as in Figure 1). Figures 12 and 13 show the same solutions from two other views. Figure 10 shows the spatially varying kink angle thresholds used in (b) and (c).*



**(a)** *Paneling solution with kink angle thresholds specified globally over the surface.*

**(b)** *Paneling solution with spatially adaptive kink angle thresholds.*

**(c)** *Paneling solution with another set of spatially adaptive kink angle thresholds.*

(a) global *cost:* 5946

| | ⬜ | 🟦 | 🟨 | 🟧 | 🟥 | ⬛ |
|---|---|---|---|---|---|---|
| *molds* | - | 38 | 15 | 2 | 119 | 32 |
| *panels* | 102 | 622 | 84 | 11 | 349 | 32 |

*divergence:* 6mm
*max angle:* 3°

(b) path 1 *cost:* 5810

| ⬜ | 🟦 | 🟨 | 🟧 | 🟥 | ⬛ |
|---|---|---|---|---|---|
| - | 73 | 8 | 1 | 169 | 22 |
| 152 | 683 | 17 | 5 | 321 | 22 |

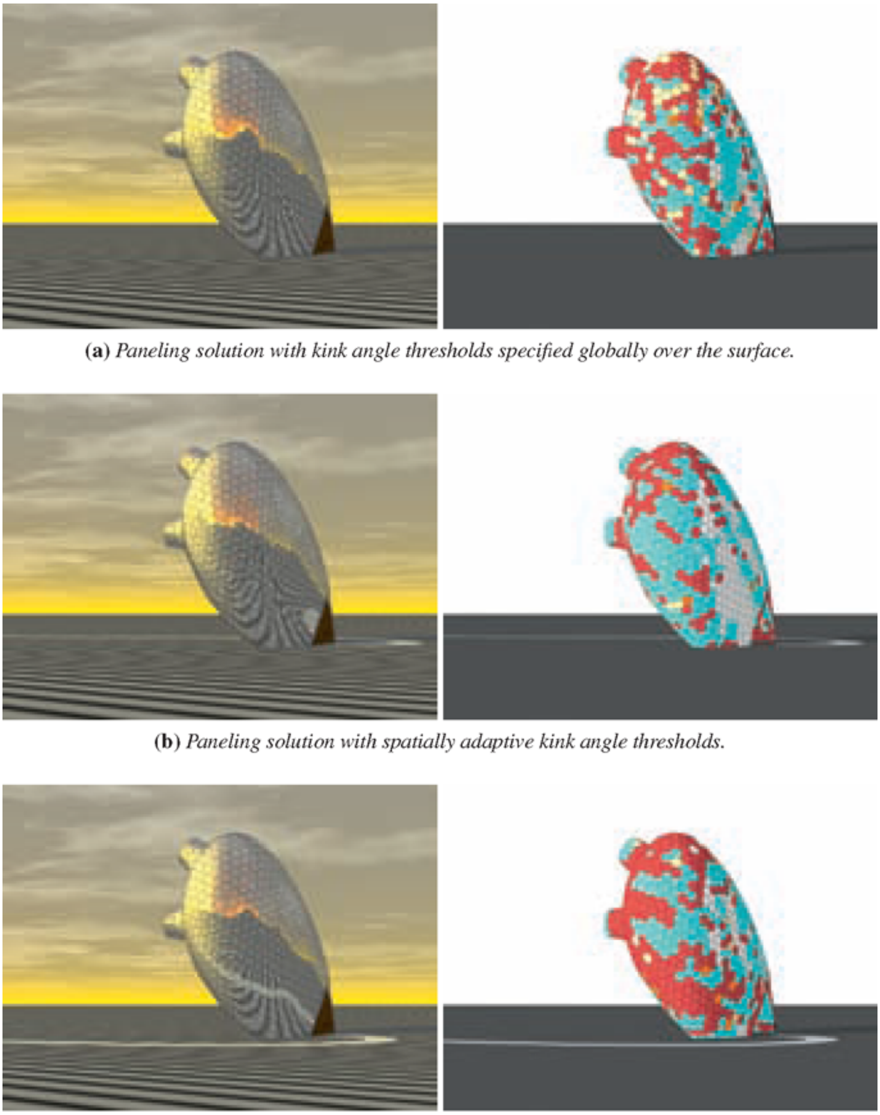
*divergence:* 6mm
*max angle:* 1°-6° (adaptive)

(c) path 2 *cost:* 6265

| ⬜ | 🟦 | 🟨 | 🟧 | 🟥 | ⬛ |
|---|---|---|---|---|---|
| - | 45 | 7 | 5 | 191 | 15 |
| 97 | 631 | 17 | 13 | 427 | 15 |

*divergence:* 6mm
*max angle:* 1°-6° (adaptive)

**Figure 12:** *Effect of global vs spatially varying kink angle specifications on the Skipper Library dataset, along with statistics for corresponding paneling solutions (see also Figure 11).*

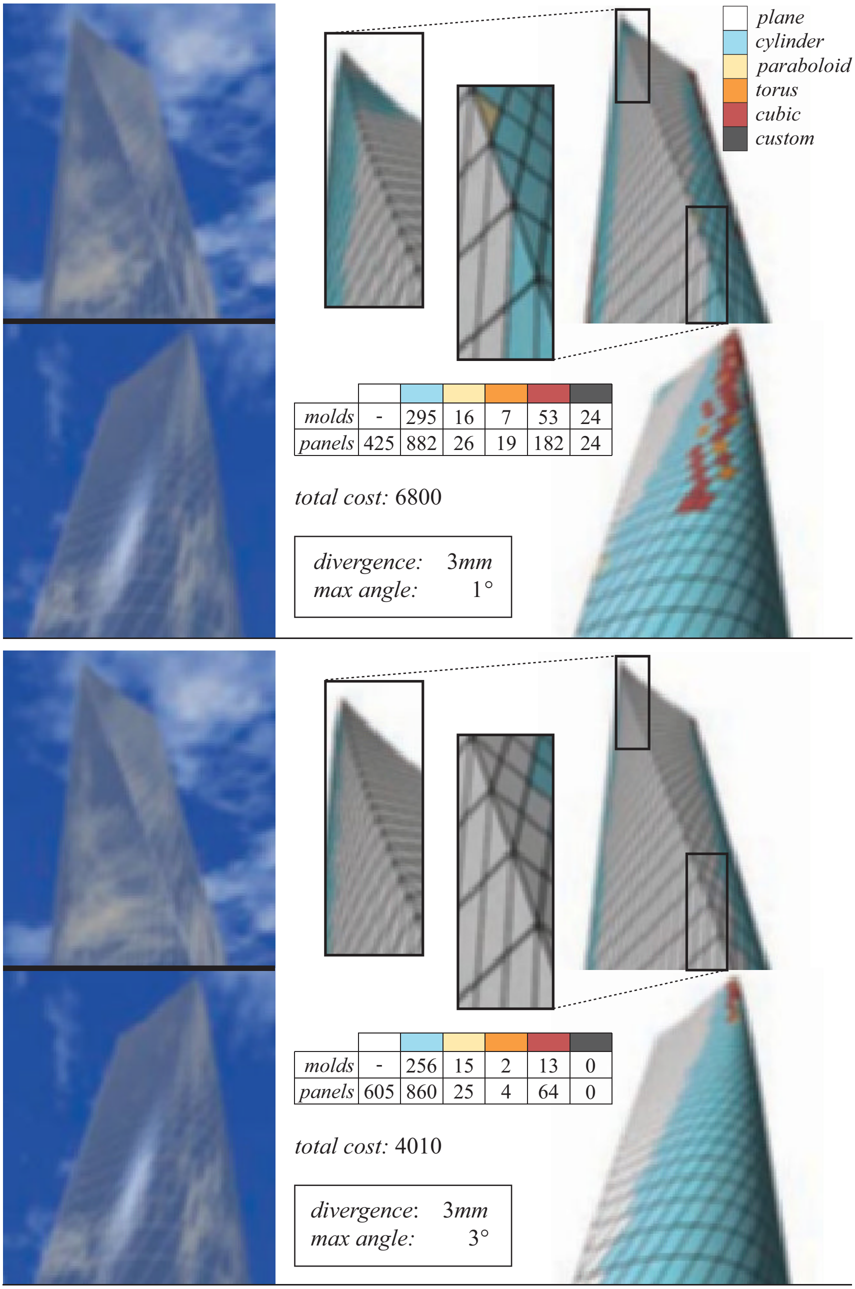


**(a)** *Paneling solution with kink angle thresholds specified globally over the surface.*

**(b)** *Paneling solution with spatially adaptive kink angle thresholds.*

**(c)** *Paneling solution with another set of spatially adaptive kink angle thresholds.*

**Figure 13:** *Effect of global vs spatially varying kink angle specifications on the Skipper Library dataset. Please refer to Figure 11 for details.*



| | ⬜ | 🟦 | 🟨 | 🟧 | 🟥 | ⬛ |
|---|---|---|---|---|---|---|
| *molds* | - | 295 | 16 | 7 | 53 | 24 |
| *panels* | 425 | 882 | 26 | 19 | 182 | 24 |

*total cost:* 6800

*divergence:* 3mm
*max angle:* 1°

| | ⬜ | 🟦 | 🟨 | 🟧 | 🟥 | ⬛ |
|---|---|---|---|---|---|---|
| *molds* | - | 256 | 15 | 2 | 13 | 0 |
| *panels* | 605 | 860 | 25 | 4 | 64 | 0 |

*total cost:* 4010

*divergence:* 3mm
*max angle:* 3°

**Figure 14:** *Paneling solution respecting crease line(s) on the input model. The characteristic sharp feature line of the Lissajous Tower is preserved in our paneling solution.*

*Future Work.* There are a few desirable extensions to the paneling algorithm leading to challenging problems for future research.

Figure 3 compares the paneling algorithm with rationalization approaches given by planar quad meshes and developable strip models. The latter include favorable geometric properties for the layout of substructure. It is natural to ask for possibilities of combining these approaches with the paneling algorithm. This motivates an adaption of the paneling algorithm towards the incorporation of optimization goals for the curve network, for example with respect to offsets and supporting structures.

For the three presented case studies, the Facade Design Study, the Lissajous Tower, and the Skipper Library, the paneling solutions are obtained in roughly 10 minutes, 1 hour, and 10 hours, respectively. In future, we plan to explore both algorithmic and computational changes to speed up the process in order to enable interactive and simultaneous exploration of reference surface design, curve network layout, and paneling solutions.

An obvious possibility for extending the paneling algorithm concerns the support of further mold types. We plan to include simple additional types like cones, but also more general surface types like general ruled surfaces.

*Conclusion.* This paper presents improvements of the paneling algorithm introduced by Eigensatz and coworkers [Eigensatz et al. 2010] to enable the preservation of sharp feature lines and the adaptive control of tolerance margins, allowing advanced exploration of cost effective rationalizations of architectural freeform surfaces. In our case studies on cutting edge architectural designs we evaluate the various modes of control enabled by our extended paneling algorithm and demonstrate the effectiveness of the algorithm with new examples, focusing on practical aspects complementary to the ones presented in [Eigensatz et al. 2010].

## Acknowledgments

We would like to thank Yves Brise, Peter Kaufmann and Sebastian Martin for their help with the project. Special thanks to Formtexx for providing the architectural datasets and to RFR for fruitful comments. This work was supported by the Swiss National Science Foundation, the Austrian Research Promotion Agency grant number 824197, and the European Community's 7th Framework Programme under grant agreement 230520 (ARC). Niloy Mitra was partially supported by a Microsoft outstanding young faculty fellowship.

## References

BOBENKO, A., AND SURIS, YU. 2008. *Discrete differential geometry: Integrable Structure*. No. 98 in Graduate Studies in Math. American Math. Soc.

DO CARMO, M. 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall.

EIGENSATZ, M., KILIAN, M., SCHIFTNER, A., MITRA, N. J., POTTMANN, H., AND PAULY, M. 2010. Paneling architectural freeform surfaces. *ACM Trans. Graphics 29*, 3.

GLYMPH, J., SHELDEN, D., CECCATO, C., MUSSEL, J., AND SCHOBER, H. 2002. A parametric strategy for freeform glass structures using quadrilateral planar facets. In *Acadia 2002*, ACM, 303–321.

GOLOVINSKIY, A., PODOLAK, J., AND FUNKHOUSER, T. 2009. Symmetry-aware mesh processing. *Mathematics of Surfaces 2009 (invited paper)*. to appear.

LIU, Y., POTTMANN, H., WALLNER, J., YANG, Y.-L., AND WANG, W. 2006. Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graphics 25*, 3, 681–689.

MITRA, N. J., GUIBAS, L. J., AND PAULY, M. 2007. Symmetrization. *ACM Trans. Graphics 26*, 3, # 63.

POTTMANN, H., LIU, Y., WALLNER, J., BOBENKO, A., AND WANG, W. 2007. Geometry of multi-layer freeform structures for architecture. *ACM Trans. Graphics 26*, 3, #65,1–11.

POTTMANN, H., HOFER, M., AND KILIAN, A., Eds. 2008. *Advances in Architectural Geometry*. Vienna.

POTTMANN, H., SCHIFTNER, A., BO, P., SCHMIEDHOFER, H., WANG, W., BALDASSINI, N., AND WALLNER, J. 2008. Freeform surfaces from single curved panels. *ACM Trans. Graphics 27*, 3, #76,1–10.

SCHIFTNER, A., HÖBINGER, M., WALLNER, J., AND POTTMANN, H. 2009. Packing circles and spheres on surfaces. *ACM Trans. Graphics 28*, 5. Proc. SIGGRAPH Asia.

SHELDEN, D. 2002. *Digital surface representation and the constructibility of Gehry's architecture*. PhD thesis, M.I.T.

# Tiling Freeform Shapes With Straight Panels: Algorithmic Methods.

**Johannes Wallner,**[1,2] **Alexander Schiftner,**[2,3] **Martin Kilian,**[2,3] **Simon Flöry,**[2,3] **Mathias Höbinger,**[2,3] **Bailin Deng,**[2] **Qixing Huang,**[4] **Helmut Pottmann** [2,5]

[1] TU Graz [2] TU Wien [3] Evolute [4] Stanford University [5] KAUST

**Abstract.** *This paper shows design studies with bent panels which are originally rectangular or at least approximately rectangular. Based on recent results obtained in the geometry processing community, we algorithmically approach the questions of an exact rectangular shape of panels; of watertightness of the resulting paneling; and of the panel shapes being achievable by pure bending. We conclude the paper with an analysis of stress and strain in bent and twisted panels.*

## 1 Introduction

This paper is concerned with panels of wood or metal, which are mounted on freeform surfaces, and which in their flat state are rectangular (or can at least be cut from rectangles). Figure 1 gives an impression of the kind of example we have in mind. In particular we deal with a mathematical formulation and algorithmic approach to this topic. Such patterns occur in the cladding of general freeform (double curved) shapes, for instance applied to interior surfaces. An experimental example, which is taken from [Spuybroek 2004], is shown by Figure 2.

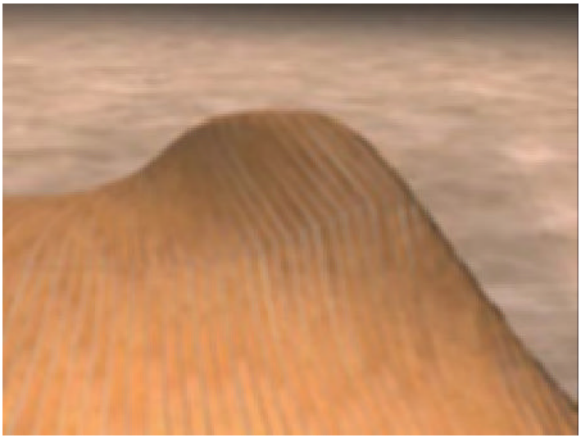


Figure 1: This image gives an impression of rectangular panels mounted on a freeform shape in an optimized pattern: Gaps are deliberately left open in order to illustrate how little the panel widths would have to be modified in order to achieve watertight paneling (cf. Figures 5, 14).

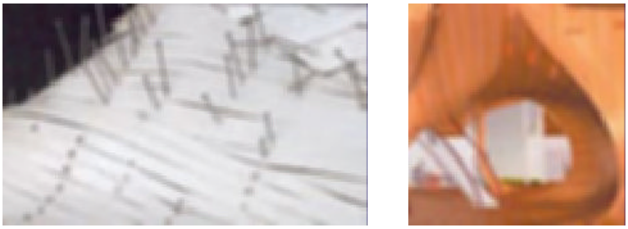


Figure 2: Experimental cladding using paper strips (left) results in an office space design by NOX Architects (right, see [Spuybroek 2004]).

In order to understand the *geometry* which governs the behaviour of panels, we discuss the various issues which arise when trying to cover freeform shapes with rectangular panels. There are several properties of the resulting patterns which one would like to have — each property being derived from practical considerations and giving rise to its own mathematical theory. Unfortunately only in rare instances we can have all of these properties at the same time. Usually a compromise will have to be found.

*The geodesic property.* Long and thin panels easily bend about their weak axis and may twist a bit, but for all practical purposes they do not bend about their strong axis. This translates into the mathematical statement that such a panel, if laid onto a surface, follows a *geodesic curve*. These curves are equally characterized by having zero geodesic curvature, and by being the shortest curves which connect different points of a surface. For more information on geodesics, the reader is referred to textbooks of differential geometry such as [do Carmo 1976].

*The constant width property.* We think of panels whose original, unfolded shape is a rectangle (see Figure 2, where those panels are represented as strips of paper). Only special shapes can be covered by such panels in a seamless and non-overlapping way: basically the only way in which this can happen is that the entire surface is itself a *developable surface*. For all other surfaces, assuming we have no gaps or overlaps, panels are not exactly rectangular when unfolded. In any case it is very important for the practical fabrication of such panels that they can be cut from a rectangular shape without too much waste. Mathematically this leads us to the requirement that the geodesic curves which guide the panels must have approximately *constant distance* from their neighbour curves.

*The developable (or 'pure bending') property.* The process of bending a surface changes the distances of points only by a very small amount, if those distances are measured inside the surface. A certain amount of twisting, as opposed to pure bending, is present in the applications we have in mind. While the previous two properties actively influence all our algorithmic approaches, the developable property is present in only one of them.

The issues discussed above lead to the following questions:

**Problem statement 1.** *We look for a system of geodesic curves in a freeform surface which are at approximately constant distance from their neighbours, and which can serve as guiding curves for the bending of rectangular wooden panels. Those panels are to cover the surface with only small gaps and no overlaps.*

**Problem statement 2.** *We look for a system of geodesic curves in a freeform surface which serve as the boundaries of wooden panels whose development is approximately straight and which can be cut from a rectangular shape. Those panels are to cover the surface without gaps.*

*Previous work.* Questions of this kind and generally the layout of geodesic patterns on surfaces have recently attracted great interest in the geometry processing community. [Kahlert et al. 2010] study the tiling of a surface by strips of controlled width which are bounded by geodesics. They employ an evolution method, starting from a single geodesic and proceeding from there until the surface under consideration is exhausted. [Pottmann et al. 2010] investigate general and multiple patterns of geodesics on freeform surfaces. They propose a mixture of methods (evolution, level set, geodesic vector fields), and it is that paper which our work is mainly based on.

— *Related work: Computing geodesics.* The theory of geodesics is found in textbooks of differential geometry such as [do Carmo 1976]. For computational purposes, shapes are represented as triangle meshes, and their geodesics are represented as polylines in meshes which are the shortest connections between points. That definition is usually sufficient but may lead to ambiguities which can be resolved by the concept of “straightest geodesics” [Polthier and Schmies 1998] which we use in our algorithms. Finding the truly shortest geodesic paths requires the computation of distance fields, for which several efficient algorithms have been developed, see for instance [Chen and Han 1996] or [Kimmel and Sethian 1998], or the later paper [Surazhsky et al. 2005].

— *Related work: Timber constructions and geodesics.* Geodesic curves have made their appearance in freeform architecture in another context, namely in the supporting structures of curved shells. [Pirazzi and Weinand 2006] show the design of freeform timber rib shells which are composed of screw-laminated beams. If such beams are considered as curves in the surface they support, then they have zero geodesic curvature, i.e., they are geodesics.

— *Related work: Rationalization of freeform surfaces by developable strips.* Early research on the cladding of freeform surfaces with developable panels evolved from the architecture of F. Gehry [Shelden 2002]. That work however does not deal with the decomposition of general shapes into developable strips, which problem was algorithmically solved by [Pottmann et al. 2008]. Already in that paper a notion of *geodesic strips* was defined: we discuss them later. The authors emphasize that in general any decomposition of a surface into developable strips must be such that the strip boundaries stay away from the *asymptotic directions* in the saddle-shaped regions of the surface. Differential-geometric issues of that kind will also be present in our work.

## 2 The Design of Patterns of Geodesics.

As a prerequisite for solving Problems 1 and 2 we first discuss patterns of geodesic curves in surfaces and methods to create them. Subsequent sections translate the geometric information stored in these curve patterns into actual paneling.

Let us rehearse the various properties of geodesics: They are the curves in a surface with zero geodesic (i.e., sideways) curvature. They are uniquely determined by an initial point and tangent. Mathematically, if a point $p(t)$ is moving in time $t$ with unit speed, then it moves along a geodesic if and only if the second derivative vector $p''(t)$ remains orthogonal to the surface. Also the shortest connections between points in the surface are geodesics.

### 2.1 Design by Parallel Transport.

In this section we describe how to find patterns of geodesics where either the maximum distance or the minimum distance between adjacent curves occurs at a prescribed location. This method is briefly described by [Pottmann et al. 2010].

Differential geometry knows the notion of *parallel transport* of a vector $V$ along a curve $s$ contained in a surface. It means moving that vector along $s$ such that it remains tangent to the surface, but such that it changes as little as possible (i.e., $\|V'(t)\|$ is minimal). It is known that the length of that vector remains unchanged [do Carmo 1976]. If, for computational purposes, a surface is represented as a mesh and a curve is represented as a polyline with vertices $P_0, P_1, P_2, \ldots$, we emulate parallel transport along that polyline by a simple step-by-step procedure explained in Figure 3.

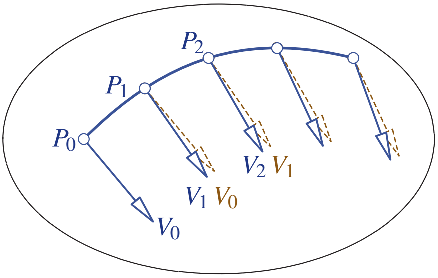


Figure 3: Parallel transport of a vector $V_0$ attached to the vertex $P_0$ along the polyline $P_0P_1P_2\ldots$ is algorithmically realized as follows: $V_i$ is found by orthogonal projection of $V_{i-1}$ onto the tangent plane of $P_i$, and subsequent re-normalizing.

Parallel transport has the following property relevant to the design of patterns of geodesics: Suppose a curve is sampled at points $P_0, P_1, \ldots$ as shown by Figure 3 and that geodesic parallel transport yields vectors $V_0, V_1, \ldots$ attached to these points. Consider the geodesic rays which emanate from the point $P_i$ in direction $V_i$ and $-V_i$ (two such rays together make one unbroken geodesic). Figure 4 shows an example of that. *Then extremal distances between adjacent geodesics occur near the chosen curve.*

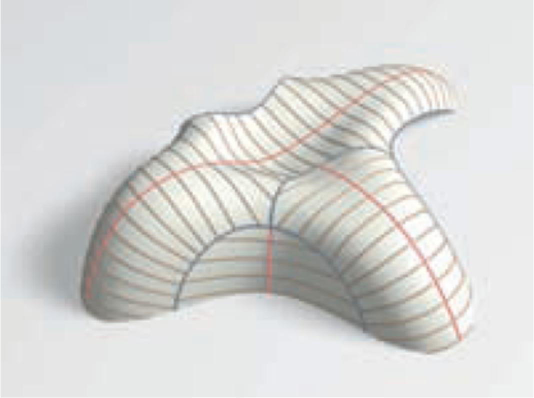


Figure 4: Designing a sequence of geodesics by choosing the locus (red) of minimum distance or maximum distance between neighbours. This is done by the *parallel transport method*. In this particular example, the method is applied not to the entire surface, but to previously selected patches.

### 2.2 Design by Evolution and by Segmentation

We first briefly rehearse the evolution method proposed by [Pottmann et al. 2010]. Starting from a *source geodesic* somewhere in the given surface, we evolve a pattern of geodesics, iteratively computing 'next' geodesics, each having approximately constant distance from its predecessor. This is not possible in an exact way on general surfaces, and if the deviation from a predefined width becomes too great one might to have to introduce breakpoints and proceed further with piecewise-geodesic curves. Figure 5 illustrates how this procedure works; for algorithmic details we refer to [Pottmann et al. 2010].

Another method employed by [Pottmann et al. 2010] is based on the concept of piecewise-geodesic vector fields. We cannot attempt to describe it here, but we mention that it performs *segmentation* of the given freeform shape into parts which are nicely coverable by a pattern of geodesic lines. Both Figure 4 and Figure 6 show an example of this. For Figure 4, the single patches which emerge after segmentation have been treated with the parallel transport method. For Figure 6, the evolution method has been used.

## 3 Panels from Curve Patterns.

Panels as we consider them are originally flat, and when mounted onto a surface they are bent (and twisted if necessary). We investigate two different ways of mathematical representation of such panels: One which produces almost exactly developable shapes which are achievable by pure bending, and another method where we check for the amount of twisting only afterwards. Unfortunately the first method is hindered by obstructions of a fundamental nature.

The exact relation between the ideal design surface $\Phi$ to be covered by the panels on the one hand, and the panels themselves on the other hand, needs clarification. One possibility is that we model the panel surfaces so that they are tangentially circumscribed to $\Phi$ along given geodesic curves; and this is what we do.

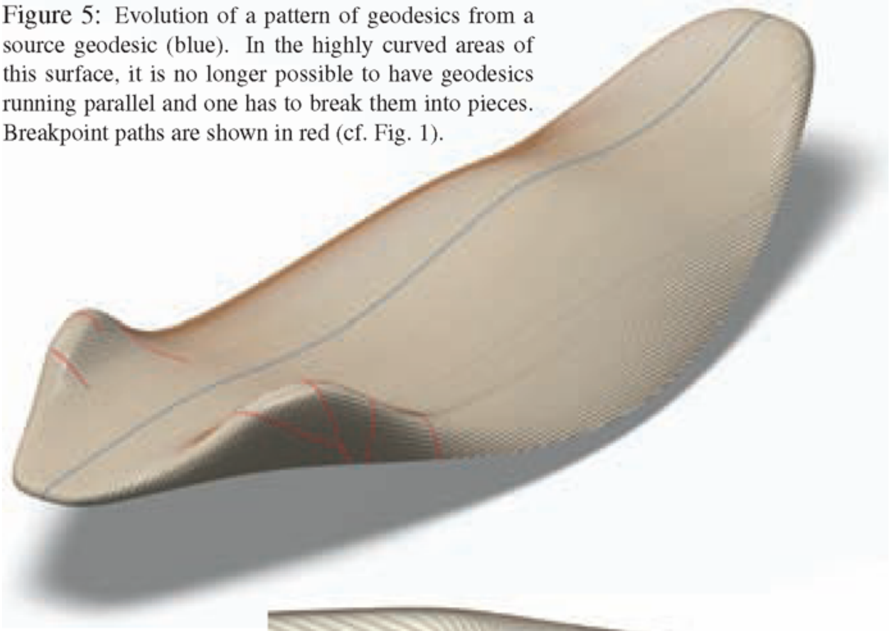


Figure 5: Evolution of a pattern of geodesics from a source geodesic (blue). In the highly curved areas of this surface, it is no longer possible to have geodesics running parallel and one has to break them into pieces. Breakpoint paths are shown in red (cf. Fig. 1).

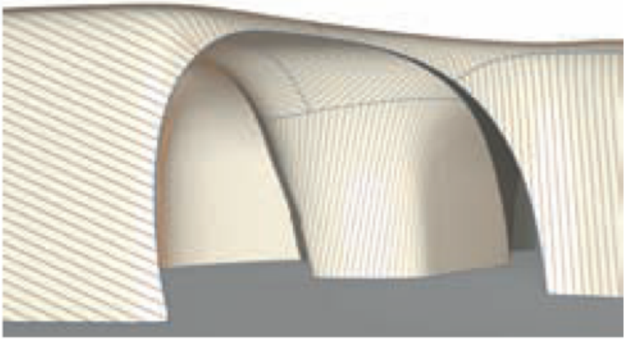


Figure 6: Segmenting a surface into pieces which can nicely be covered by a sequence of geodesic lines. For the covering, the evolution method was employed.

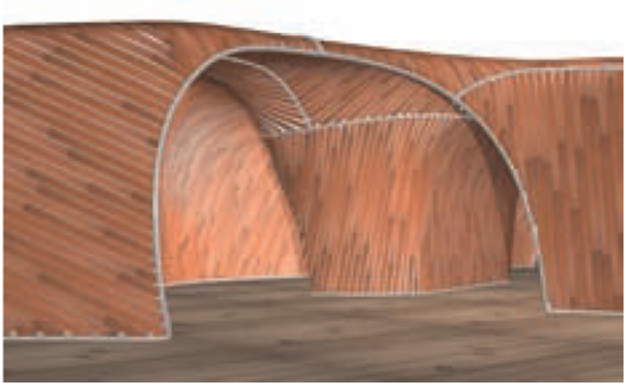


Figure 7: This design with bent rectangular panels is based on Figure 6.

Another idea is that the panel surfaces are *inscribed* into the design surface. For instance we could connect two neighbouring geodesics by a developable surface which is subsequently used for the panel. Algorithmically this is not easy [Rose et al. 2007] and anyway we would rather have a geodesic running in the center of the panel (which is achieved with the idea of circumscribed panels).

### 3.1 Panels with Pure Bending: the Tangent Developable Method.

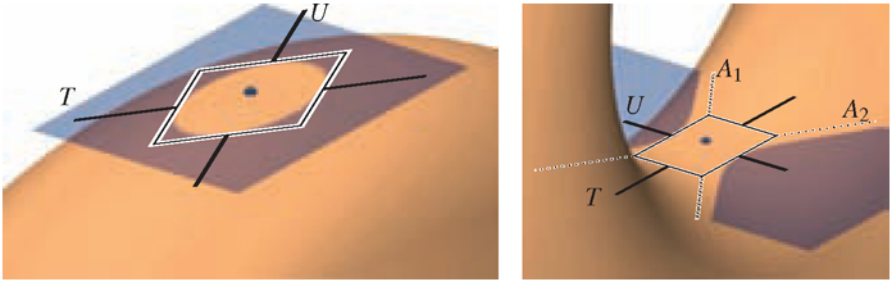


Figure 8: Illustration of asymptotic directions $A_1$,$A_2$ and conjugate directions $T$, $U$: Parallel translation of a tangent plane (blue) by a small amount and intersection with the surface yields a curve which approximates a conic section (the *Dupin indicatrix*). In negatively curved areas this is a hyperbola, whose asymptotes $A_1$,$A_2$ define the *asymptotic directions*. Any parallelogram tangentially circumscribed to the indicatrix defines two conjugate tangents $T$, $U$. It is known that $A_1$,$A_2$ are diagonals of any such parallelogram. Obviously choosing $T$ determines $U$. For both figures, the base surface is a torus.

For smooth surfaces the notion of *conjugate tangents* is defined; they are explained by Figure 8. Mathematically vectors $v, w$ which are expressed in a coordinate system whose basis are principal curvature vectors are conjugate, if and only if $v^T \operatorname{diag}(\kappa_1, \kappa_2) w = 0$, where $\kappa_1, \kappa_2$ are the principal curvatures. Algorithmically, curvatures and conjugate tangents can be computed from triangle meshes by well known methods of geometry processing, see e.g. [Cazals and Pouget 2003].

Conjugate tangents play an important role here because they can be used to create a developable surface $\Psi$ which is tangentially circumscribed to a given surface $\Phi$ along a curve $s$ (see Figure 9). That *tangent developable* even has the nice property

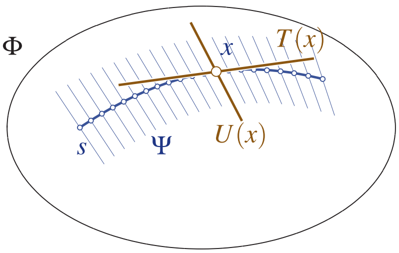


Figure 9: Consider a point $x$ in a geodesic $s$ which lies in the surface $\Phi$. If $T(x)$ is tangent to the geodesic, compute $U(x)$ as being conjugate to $T(x)$. Then the union of all tangents $U(x)$ is a developable ruled surface $\Psi$ which is tangentially circumscribed to $\Phi$ along the curve $s$.

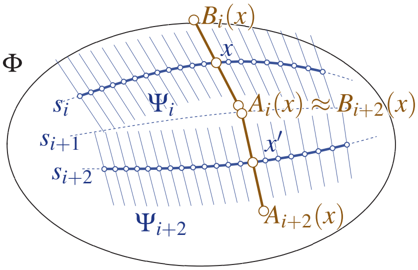


Figure 10: Developable surfaces $\Psi_i$ associated with geodesics with *even* indices $i$ are trimmed by geodesics with odd index.

that $s$ is a geodesic not only for $\Phi$, but also for $\Psi$. Thus, when $\Psi$ is unfolded into the plane, $s$ becomes a straight line.

This geometric information suggests the following algorithm to create panels: First, for all geodesics $s_i$ in a given geodesic pattern compute the tangent developable $\Psi_i$ according to Figure 9. Trim those surfaces along the intersection curves with their respective neighbours. Unfolding the trimmed $\Psi_i$'s yields the flat state of panels.

Unfortunately this does not work in practice. One reason is that the rulings of the tangent developables may behave in weird ways. Another reason is that the intersection of neighbouring $\Psi_s$'s is often ill-defined, so trimming as suggested will not work. We therefore have chosen the following modified procedure:

1. For the geodesics $s_i$ where $i$ is an even number compute the tangent developable $\Psi_i$ according to Figure 9. That is, for a dense sample of points $x$ on $s_i$ we compute the rulings $U_i(x)$ which are conjugate to the tangent $T_i(x)$.

2. Delete all rulings $U_i(x)$ of $\Psi_i$ where the angle enclosed with the tangent $T_i(x)$ is smaller than some threshold (say, 20 degrees) and fill the holes by interpolation (this is a standard procedure).

3. On each ruling $U_i(x)$ determine points $A_i(x)$ and $B_i(x)$ which are closest to the geodesics $s_{i-1}$ and $s_{i+1}$, respectively (see Figure 10). This serves for trimming the surface $\Psi_i$.

4. Optimize globally the positions of points $A_i(x)$ and $B_i(x)$ such that trim curves are smooth, such that $A_i(x)$ and $B_i(x)$ are close to geodesics $s_{i-1}$, $s_{i+1}$, and such that the ruling segments $A_i(x)B_i(x)$ lie close to $\Phi$. For this optimization we need the distance fields of $\Phi$ and of the single geodesics. We only change the surface a little bit and hope not to lose too much developability.

Figures 11, 12 and 13 illustrate panelizations of freeform shapes obtained by this method. The degree of developability which is achieved can be evaluated by measuring the Gauss curvatures of panel surfaces, such as done by Figure 16. The Gauss curvature vanishes for exact developability. The exact values for the panelizations of Figures 11 and 14, which work with the same design surface and comparable strip width can be seen in the table at the end of Section 4.

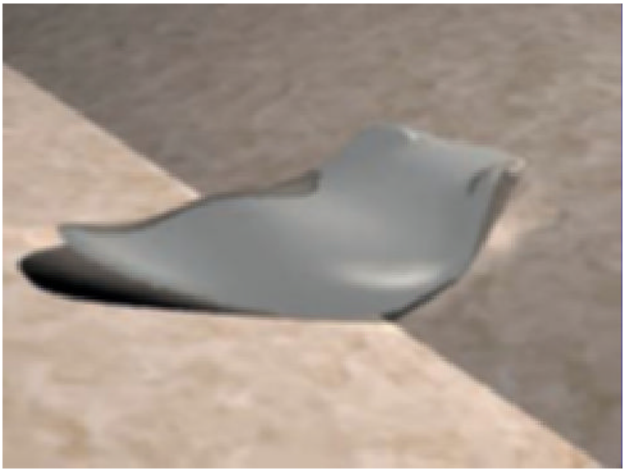


Figure 11: Almost-developable strips constituting a watertight surface.

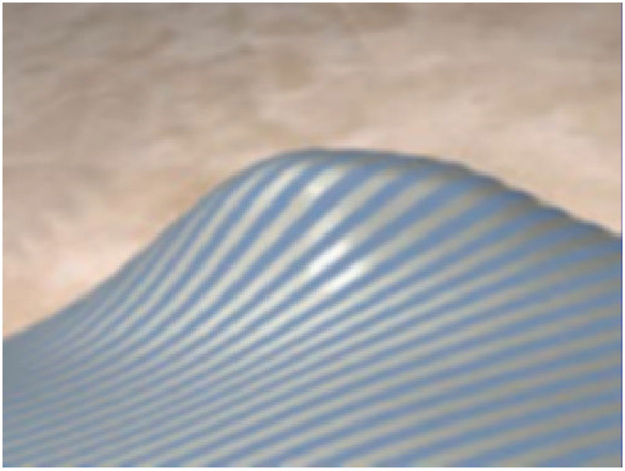


Figure 12: Detail of Figure 11. The gaps in between panels which occur in highly curved areas are hardly visible. The maximal strip width is 0.4% of the entire design's bounding box diagonal.

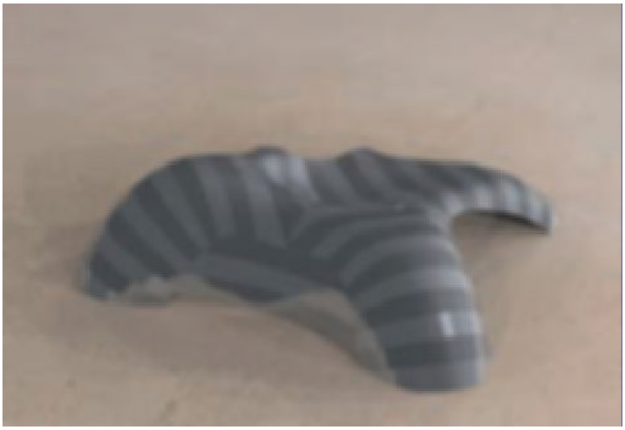


Figure 13: Watertight panels based on the segmentation and parallel transport methods. See also Figure 4. The intrinsic curvature of the rather broad panels is too high to make this design practicable: its purpose is to illustrate the parallel transport method.

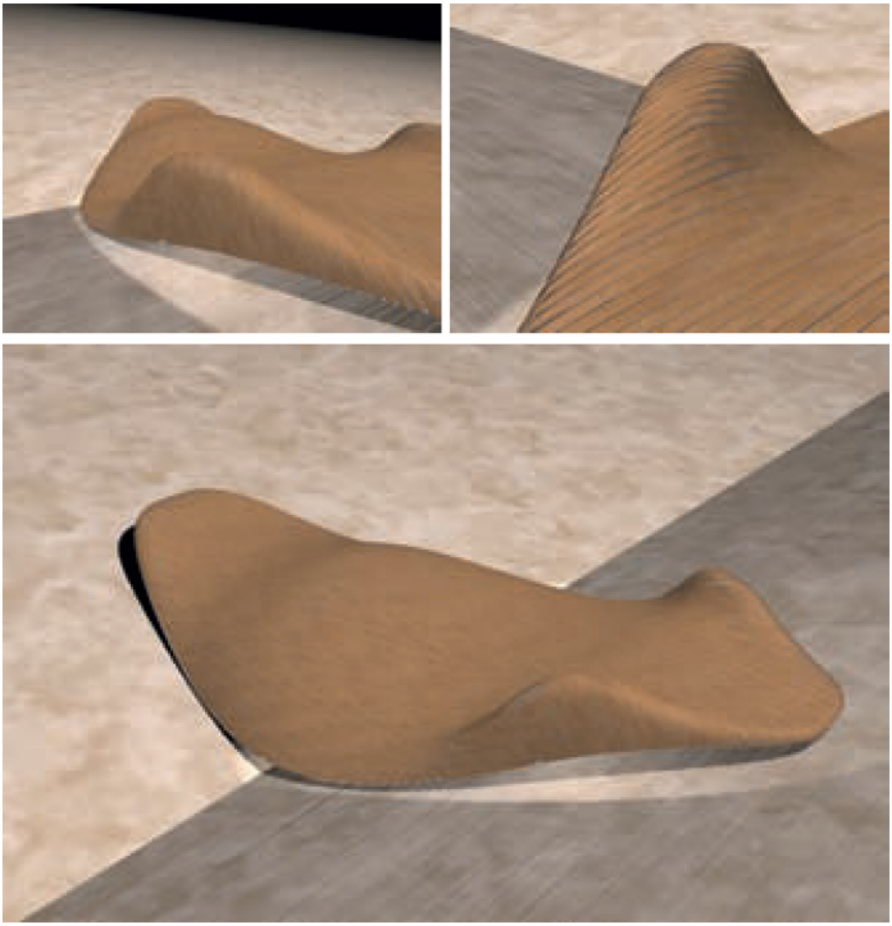


Figure 14: *Below:* A surface is covered by wooden panels of constant width. This is achieved by the 'evolution method' illustrated by Figure 5: The pattern of panels evolves from a well-placed source geodesic as long as the requirement of constant panel width is satisfied up to certain thresholds. If the panel width deviates too much from the desired value, the geodesics are broken. Subsequently panel surfaces have been created by the 'binormal method'. *Above:* Details. A further detail is shown by Figure 1.

## 3.2 The Binormal Method.

Our second method of defining panels (after a pattern of geodesics in the surface $\Phi$ has been found) works directly with the geodesic curves.

Assume that such a geodesic $s$ is traversed by a point $P(t)$ moving with unit speed, where $t$ is a time parameter. For each time $t$ we have the velocity vector $T(t)$, the normal vector $N(t)$ of the surface $\Phi$ in the point $P(t)$, and a third vector $B(t)$ (the *binormal vector*) which makes $T, N, B$ a moving orthogonal right-handed frame.

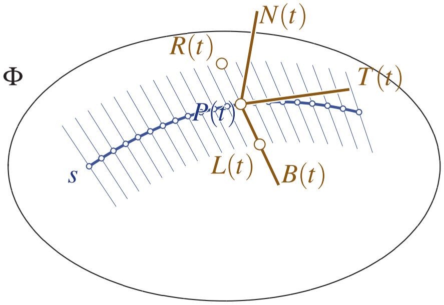


Figure 15: The binormal method defines a ruled panel surface from a central geodesic $s$ via its Frenet frame $T,N,B$: The ruling passing through the central point $P(t)$ on the geodesic is indicated by the binormal vector $B(t)$. The endpoints of the ruling segment are points $L(t)$ and $R(t)$ whose distance from $P(t)$ is half the intended panel width.

For computational purposes, the surface $\Phi$ is represented as a triangle mesh and $s$ is given as a polyline. Numerically the computation of the frame $T,N,B$ is stable if performed in the way described above, despite the fact that it is actually the Frenet frame of $s$ which usually exhibits numerical deficiencies (this connection with the Frenet frame follows from the geodesic property).

For each geodesic, the associated panel surface is constructed according to Figure 15. Panelizations of freeform surfaces which have been achieved with this method are shown by Figures 7 and 14.

### 3.3 Discussion

The previous two subsections proposed two different methods of defining ideal and mathematically abstract surfaces which are to be followed by panels. The ‘tangent developable’ method tries to produce panel surfaces which are achievable by pure bending (in fact the tangent developable is the only surface with this property which is also tangent to the original design surface). Thus the mathematical goal of developability is corresponding to a natural manufacturing goal. It seems reasonable to let actual panels exactly follow the surfaces proposed by this algorithmic method.

The situation is slightly different for the second suggested way of defining panel surfaces (the ‘binormal’ method). From a mathematical viewpoint it is a simple and obvious way of defining panel surfaces, but it is unclear that this surface should be the shape of a panel after it has been forced to follow a geodesic on the surface $\Phi$. Of course such a shape is subject to the existing constraints, but one would assume that panels rather assume shapes achievable by pure bending. The purpose of the binormal method is mainly to pin down a mathematically exact surface, for the practical purpose of having shapes exactly defined. Anyway the following section shows that the panel shapes defined by the binormal method are admissible from the viewpoint of stresses and strain.

## 4 Stress and Strain in Panels.

This section investigates the deformation a rectangular strip of elastic material experiences when it is bent into the shape of a ruled surface $\Psi$ such that the central line

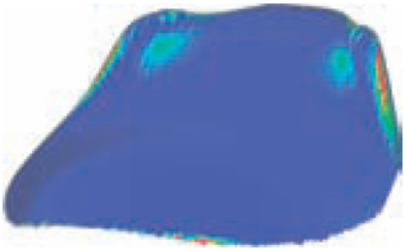


Figure 16: Visualization of Gaussian curvature of the design shown by Figures 11, 12. Blue corresponds to zero, red to the maximum value $-0.02$ (this means $\rho = 7.07$). The bounding box diagonal of this object is 188.

$m$ of the strip follows a 'middle geodesic' $s$ in $\Psi$. This applies to both our methods of defining panel surfaces. It seems a reasonable assumption that the central line is only bent, but not stretched. Due to the saddle shape (negative Gaussian curvature) of all ruled surfaces, the lines parallel to $m$ at distance $d/2$ are not only bent, but also stretched. It is known that after introducing the *radius of Gaussian curvature* $\rho = 1/\sqrt{|K|}$, the relative increment in length (the strain) of the strip boundaries is given by

$$\varepsilon = \frac{1}{2}(d/2\rho)^2 + \cdots,$$

where the dots indicate terms of higher order in $d$. We are first concerned with tensile stress due to this stretching; for other stresses due to bending and shear see the end of this section. A rough estimate, expressing stress by $\sigma = E\varepsilon$, yields

$$d/2\rho \leq C, \quad \text{with} \quad C = \sqrt{2\sigma_{\max}/E},$$

where $\sigma_{\max}$ is the maximum admissible stress and $E$ is Young's modulus. The approximative nature of our computation implies using a suitable safety factor when choosing $\sigma_{\max}$. The value $C$ is a material constant which yields an upper bound $d_{\max} = 2\rho_{\min}C$ for the maximum strip with. With sample values for $\sigma_{\max}$ we get

| material | Young modulus<br>$E$ $[N/mm^2]$ | maximum stress (sample values)<br>$\sigma_{\max}$ $[N/mm^2]$ | constant<br>$C = \sqrt{2\sigma_{\max}/E}$ |
|---|---|---|---|
| steel | 200000 | 250 | 0.05 |
| wood | 13000 | 80 | 0.11 |

Strip widths and their admissibility for models shown in this paper are collected in the following table. Since these examples have been selected mainly with a view towards visualization, some are not admissible. However they can easily be made so by choosing narrower panels. The choice of units in this table is arbitrary.

| Figure No. | material | actual panel width $[m]$ | $\lvert K\rvert_{\max}$ $[m^{-2}]$ | $\rho_{\min}$ $[m]$ | bounding box size $[m]$ | admissible width $[m]$ | admis-ible? |
|---|---|---|---|---|---|---|---|
| 1, 5, 14 | wood | $d = 0.7$ | 0.1 | 3.16 | 188 | 0.7 | yes |
| 4, 13 | steel<br>wood | $d \leq 0.1$ | 5 | 0.44 | 2.8 | 0.04<br>0.1 | no<br>yes |
| 11, 12 | steel | $d \leq 0.8$ | 0.02 | 7.07 | 188 | 0.71 | almost |

*Bending and shear stress.* Both bending stress and shear stress for a panel with thin rectangular cross-section depend on the panel thickness $h$, but not on the panel width $d$ if $h/d \ll 1$; the maximum values of these stresses (denoted by $\sigma$, $\tau$ in this paragraph) occur on the outer surface of the panel. These values depend on the curvature $\kappa$ of the panel's central geodesic and the rate of torsion $\theta$ of the panel (we have $\sigma = E\kappa h/2$ and $\tau = hG\theta$, where $G$ is the shear modulus). Clearly the panel surfaces obtained by the 'tangent developable' method experience less shear than the ones created by the 'binormal' method. It is a standard matter to combine all stresses (tension, shear, bending) and use this information for checking if the panel's dimensions are admissible.

It is interesting to know how the rate of torsion $\theta$ (twist angle per panel length) is related to the Gaussian curvature of the panel: It is known that $\theta$, measured in arc per meter, does not exceed $\sqrt{|K|} = 1/\rho$, where the maximum value occurs in case the central geodesic's tangent happens to be an asymptotic direction of the panel surface [do Carmo 1976].

## 5 Conclusion.

This paper treats paneling of freeform surfaces with rectangular (or almost-rectangular) panels, which are known to follow geodesic curves. For the layout of a system of geodesics several methods have recently been published. We survey some of them in this paper, especially those which produce geodesics running approximately parallel to each other. We further discuss the panel surfaces themselves under the viewpoint of panel shapes achievable by pure bending and a watertight overall panel surface, and we demonstrate our methods by means of some examples. Finally we discuss tensile and shear stresses in panels which occur when they are mounted on freeform surfaces.

*Future research.* The connection between geometry and mechanics is a very important and at the same time most challenging issue in any freeform design. One topic of future research therefore is to combine geometric considerations with simple aspects of mechanics – our way of expressing stresses by Gaussian curvature already points in this direction.

Panelization poses many geometric questions whose systematic investigation would be rewarding: For instance, panels in the shape of generalized cylinders which are important for bent glass; and more generally special shapes of panels which are relevant for certain manufacturing techniques and specific applications in building construction. Our aim must generally be to find *construction-aware design tools* which do not generate shapes first and lets us think about manufacturing afterwards, but tools which actively, during the design phase, incorporate the side conditions engendered by manufacturing constraints.

## References

CAZALS, F., AND POUGET, M. 2003. Estimating differential quantities using polynomial fitting of osculating jets. In *Symp. Geometry processing*, Eurographics, L. Kobbelt, P. Schröder, and H. Hoppe, Eds., 177–178.

CHEN, J., AND HAN, Y. 1996. Shortest paths on a polyhedron. I. Computing shortest paths. *Int. J. Comput. Geom. Appl. 6*, 127–144.

DO CARMO, M. 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall.

KAHLERT, J., OLSON, M., AND ZHANG, H. 2010. Width-bounded geodesic strips for surface tiling. *Vis. Computer*. to appear.

KIMMEL, R., AND SETHIAN, J. A. 1998. Computing geodesic paths on manifolds. *PNAS 95*, 8431–8435.

PIRAZZI, C., AND WEINAND, Y. 2006. Geodesic lines on free-form surfaces: optimized grids for timber rib shells. In *Proc. World Conference on Timber Engineering*. 7pp.

POLTHIER, K., AND SCHMIES, M. 1998. Straightest geodesics on polyhedral surfaces. In *Mathematical Visualization*, Springer, H.-C. Hege and K. Polthier, Eds., 391–409.

POTTMANN, H., SCHIFTNER, A., BO, P., SCHMIEDHOFER, H., WANG, W., BALDASSINI, N., AND WALLNER, J. 2008. Freeform surfaces from single curved panels. *ACM Trans. Graphics 27*, 3, #76, 1–10. Proc. SIGGRAPH.

POTTMANN, H., HUANG, Q., DENG, B., SCHIFTNER, A., KILIAN, M., GUIBAS, L., AND WALLNER, J. 2010. Geodesic patterns. *ACM Trans. Graphics 29*, 4, #43,1–10. Proc. SIGGRAPH.

ROSE, K., SHEFFER, A., WITHER, J., CANI, M., AND THIBERT, B. 2007. Developable surfaces from arbitrary sketched boundaries. In *Symp. Geom. Processing*, A. Belyaev and M. Garland, Eds. 163–172.

SHELDEN, D. 2002. *Digital surface representation and the constructibility of Gehry's architecture*. PhD thesis, M.I.T.

SPUYBROEK, L. 2004. *NOX: Machining Architecture*. Thames & Hudson.

SURAZHSKY, V., SURAZHSKY, T., KIRSANOV, D., GORTLER, S., AND HOPPE, H. 2005. Fast exact and approximate geodesics on meshes. *ACM Trans. Graphics 24*, 3, 553–560. Proc. SIGGRAPH.

# Freeform Rigid-Foldable Structure using Bidirectionally Flat-Foldable Planar Quadrilateral Mesh

**Tomohiro Tachi**

The University of Tokyo

**Abstract.** *This paper presents a computational design method to obtain collapsible variations of rigid-foldable surfaces, i.e., continuously and finitely transformable polyhedral surfaces, homeomorphic to disks and cylinders. Two novel techniques are proposed to design such surfaces: a technique for obtaining a freeform variation of a rigid-foldable and bidirectionally flat-foldable disk surface, which is a hybrid of generalized Miura-ori and eggbox patterns, and a technique to generalize the geometry of cylindrical surface using bidirectionally flat-foldable planar quadrilateral mesh by introducing additional constraints to keep the topology maintained throughout the continuous transformation. Proposed methods produce freeform variations of rigid-foldable structures that have not been realized thus far. Such a structure forms a one-DOF mechanism with two possible flat states. This enables the designs of deployable structures useful for packaging the boundary of architectural spaces, space structures, and so on.*

## 1 Introduction

A polyhedral surface composed of rigid facets connected by rotational edges forms a kinetic mechanism: a rigid folding mechanism. Several collapsible structures are proposed using this type of kinetic mechanism, such as Miura-ori [Miura 1980] and eggbox patterns [Brunner 1965] (Figure 1). Such a rigid-foldable and flat-foldable surface is useful as a deployable and transformable structures in an architectural context because of the following advantages.

1. The existence of a collapsed state enables compact packaging of the structure.
2. The synchronized complex folding motion produced by constrained rotational hinges can be controlled with simple manipulation.
3. The transformation mechanism that does not rely on the flexibility of materials can be made out of thick rigid panels and hinges.
4. The watertightness of the surface maintained throughout the transformation is potentially suitable for the envelope of a space, a partition, and the facade of a building.

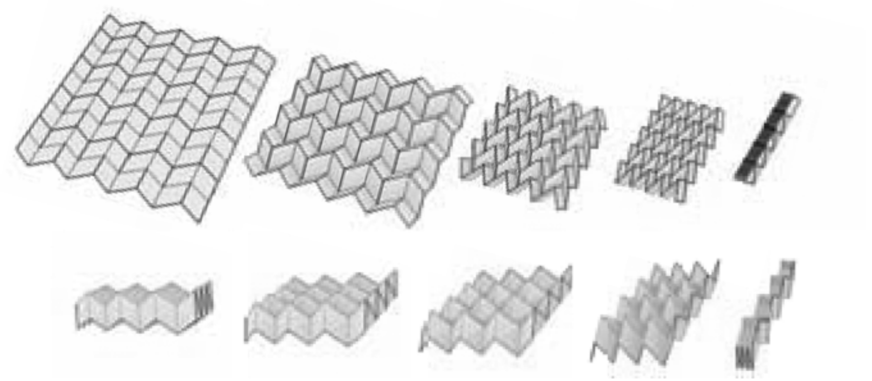


Figure 1: Folding motions of Miura-ori (Top) and eggbox pattern (Bottom).

In order to apply such kinetic surfaces to design purposes, it is required that we can control the three-dimensional shape of the surface and its behavior so that they are consistent with the design requirements. Design requirements in architectural context include environmental conditions, functional requirements, configuration of surrounding buildings and structures, and user's preference. However, since the rigid-foldability of known surfaces strongly relies on the symmetry of the pattern, designing such structures could not be done in an ad-hoc approach; the geometric constraints must be generally investigated to solve an inverse problem of obtaining a pattern from the resulting form and behavior. The objective of our study is thus to freely design kinetic forms using design methods based on computational geometry.

As the first step to achieve this goal, the author proposed a freeform design method of kinetic structures based on origami (developable surface) to provide design variations of Miura-ori through an interactive design system in which a user can deform the surface freely while sustaining the rigid-foldability of the surface ([Tachi 2009a]). This method yields variations of rigid-foldable origami models that have never been achieved otherwise, however, the method was not general enough for design applications, especially of architecture, since the developability condition used is not always an essential condition when we construct structures from multiple parts. In particular, this developability condition disables the design of rigid-foldable collapsible cylinders, realization of which can contribute the designs of surrounding of a volume of a space, collapsible containers, compound surface structures, and so on. This limitation mainly comes from the fact that a cylindrical surface cannot have a globally developed state under the conventional definition.

This paper provides a novel method that enables a freeform variation of such rigid-foldable structure, not restricted by the developability of the surface or the disk topology. We will solve this design problem by extending and combining the ideas of generalized Miura-ori [Tachi 2009a] (or flat-foldable 4-valent-vertex origami) and generalized eggbox pattern (or discrete Voss surface) [Schief et al. 2007] to produce bidirectionally flat-foldable planar quadrilateral mesh. Such a mesh has nice generalized characteristics: it is bidirectionally flat-foldable and produces one-DOF mechanism. Required design condition is loose compared to purely developable surface and it allows asymmetric and cylindrical variations. We will show a novel design method to obtain rigid-foldable collapsible cylinders by deforming symmetric rigid-foldable cylinders of [Tachi 2009b] into asymmetric ones.

## 2 Geometry of Rigid-Foldable Quadrilateral Mesh Disk

This section shows the geometry of 4-valency rigid-foldable mesh with two flat states that combines flat-foldable origami surface and discrete Voss surface. A 4-valency mesh or a quadrilateral-mesh surface in general does not enable a continuous rigid-folding motion because an overconstrained system is constructed. This is because the configuration of a rigid-foldable structure homeomorphic to a disk can be represented by the folding angles of edges, while these variables are constrained at each vertex by 3 equations as it is used in the simulation of rigid origami [Tachi 2009c]; if we are to construct a quadrilateral mesh, then the number of constraints exceeds the number of variables only by making $3 \times 3$ array. However, Miura-ori [Miura 1980] and eggbox patterns [Brunner 1965] are known to produce singular one-DOF mechanisms (Figure 1) because of their redundant constraints. Miura-ori is a flat-foldable origami surface composed of parallelograms used for packaging of large membranes in the space. The surface produces a synchronized kinetic motion of expanding in $x$ and $y$ directions at the same time from a collapsed state to a completely developed state (Figure 1 Top). Eggbox pattern is a polyhedral surface similarly composed of parallelograms but is not a developable surface. This has two flat-folded states and produces a different one-DOF kinetic behavior from that of Miura-ori; the surface expands in $x$ direction when it collapses in $y$ direction (Figure 1 Bottom). While Miura-ori is a surface with foldlines folded at the same time, eggbox pattern is a surface that has two groups of foldlines collapsed at the same time.

Since Miura-ori and eggbox patterns are surfaces composed of congruent parallelograms, the redundancy of constraints required for the mechanism seem to originate in its global repeating symmetry. However, the condition to produce such redundancy actually is a result of its intrinsic symmetry, i.e., the angles coincidence at each vertex, which also contribute to bi-directional flat-foldability of the surfaces (Miura-ori is developable and flat-foldable. eggbox pattern is flat-foldable in two directions). The intrinsic symmetries of Miura-ori and eggbox pattern, yield generalized forms of rigid-foldable origami surface [Tachi 2009a] and discrete Voss surface [Schief et al. 2007], respectively. In addition, these surfaces can be merged into one generalized hybrid surface, i.e., bidirectionally flat-foldable planar quadrilateral mesh, based on the fact that the behaviors of the two surfaces can be represented using a common form.

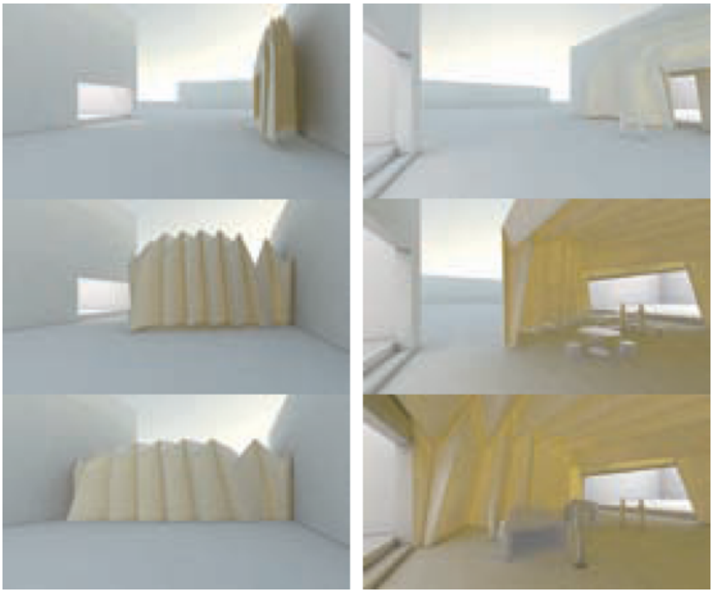


Figure 2: A corrugated vault used as a transformable architecture that connects two existing buildings.

### 2.1 Generalized Miura-ori

The rigid-foldability condition of generalized Miura-ori is presented by [Tachi 2009a]: consider a planar quad mesh surface of disk topology whose vertices are incident to 4 foldlines; when the surface is developable and flat-foldable, finite rigid-foldability of the surface is equivalent to the existence of one valid semi-folded state. This sufficient condition for rigid-foldability yields rigid-foldable generalization of Miura-ori. The condition also applies to any other degree-4 vertex based patterns such as assymetric vault pattern shown in Figure 2.

The behavior of the surface is represented by the configuration of fold angles. A vertex with 4 foldlines, and thus with 4 sector angles $\theta_i (i = 0, 1, 2, 3)$, form a one-DOF mechanism (Figure 3 Top). In the case of developable and flat-foldable surface, condition of origami (i.e., developability) and flat-foldability given by $\sum_{i=0}^{3} \theta_i = 2\pi$ and$\sum_{i=0}^{3} (-1)^i \theta_i = 0$, respectively, force the sector angles to satisfy

$$\theta_0 = \pi - \theta_2 \quad \text{and} \quad \theta_1 = \pi - \theta_3. \tag{1}$$

According to [Tachi 2009a], the fold angles $\rho_i$ and $\rho_j$ incident to the vertex are related as follows:

$$\tan \frac{\rho_i}{2} = \begin{cases} A_{i,j} \tan \frac{\rho_j}{2} & (i - j = 1 \text{ or } 3 \mod 4) \\ \pm \tan \frac{\rho_j}{2} & (i - j = 2 \mod 4) \end{cases} \tag{2}$$

where the latter represents that pairs of opposite foldlines have an equal absolute folding angles, and $A_{i,j}$ is the coefficient between these two equivalent pairs determined by $\theta_0, \cdots, \theta_3$, i.e., intrinsic measure in the crease pattern independent from the folding angles. Specifically, if $|\rho_0| = |\rho_2| > |\rho_1| = |\rho_3|$,

$$|A_{0,1}| = \sqrt{\frac{1+\cos(\theta_0 - \theta_1)}{1+\cos(\theta_0 + \theta_1)}}.$$

Note that this relationship is a special case of the generalized form of origami vertex presented by [Huffman 1976].

This gives an explicit configuration of folding angles of all foldlines globally connected via degree-4 vertices:

$$\left\{\tan\frac{\rho_i(t)}{2}\right\} = \left\{\tan\frac{\rho_i(t_0)}{2}\right\}\frac{\tan\frac{t}{2}}{\tan\frac{t_0}{2}}, \tag{3}$$

where $t$ $(0 \leq t \leq \pi)$ is the parameter that defines the folding amount ($t = 0$ and $t = \pi$ indicate developed and flat-folded states respectively) and $\left\{\tan\frac{\rho_i(t_0)}{2}\right\}$ is an arbitrary semi-folded state $(0 < t_0 < \pi)$. Generalized Miura-ori thus shows a kinetic motion of expanding in $x$ and $y$ directions at the same time.

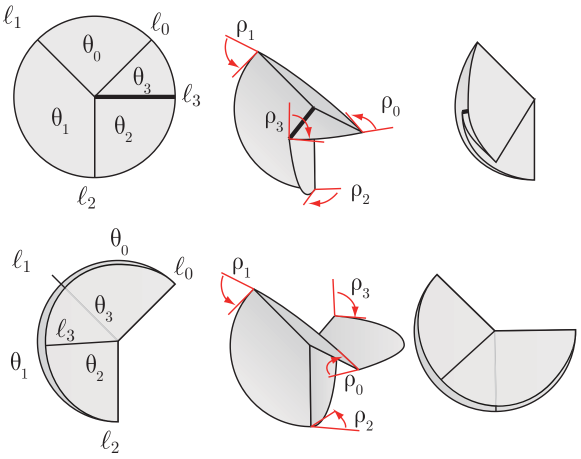


Figure 3: The folding motions of vertices of origami (Top) and discrete Voss surface (Bottom). Note that set of planes used is identical.

## 2.2 Generalized Eggbox Pattern

Eggbox pattern can be generalized as a discrete Voss surface; the rigid-foldability of discrete Voss surface is proved by [Schief et al. 2007]. Even though the term discrete Voss surface implies a mesh pattern that represents a smooth surface, corrugated surfaces such as eggbox pattern is also included in the family, since it is defined by the intrinsic angular condition that the opposite angles are equal. Every interior vertex on a discrete Voss surface is incident to 4 edges (or foldlines) and sector angles $\theta_i$ $(i = 0,1,2,3)$ that satisfy the following:

$$\theta_0 = \theta_2 \quad \text{and} \quad \theta_1 = \theta_3. \tag{4}$$

Note the similarity to (1): in fact, the set of planes that form an origami vertex and eggbox vertex are identical (Figure 3). According to [Schief et al. 2007], the folding angles $\rho_i$ and $\rho_j$ are similarly written as the following form:

$$\tan\frac{\rho_i}{2} = \begin{cases} A_{i,j}\cot\frac{\rho_j}{2} & (i-j = 1 \text{ or } 3 \mod 4) \\ \tan\frac{\rho_j}{2} & (i-j = 2 \mod 4). \end{cases}$$

If we measure the folding angles of one of the opposite pairs by their complementary angles $\rho_i' = \pi - \rho_i$, this becomes

$$\tan\frac{\rho_i}{2} = \begin{cases} A_{i,j}\tan\frac{\rho_j'}{2} & (i-j = 1 \text{ or } 3 \mod 4) \\ \tan\frac{\rho_j}{2} & (i-j = 2 \mod 4). \end{cases} \tag{5}$$

Therefore the kinematics of Miura-ori and discrete Voss surface are essentially identical as represented by Equation (3).

### 2.3 Hybrid Surface: Bidirectionally Flat-foldable Planar Quadrilateral Mesh

Here, we introduce a new type of functional patterns, i.e., BDFFPQ mesh (defined later), based on combining the geometry of origami and discrete-Voss surface We introduce the concept of *complementary foldlines*, which are edges whose folding angles are measured by their complementary angles, thus a complementary foldline folds from ±π to 0 when an ordinary foldline folds from 0 to ±π. By using complementary foldlines (CFLs) along with foldlines (FLs), we can produce a 4-valency network that consistently joins the vertices of origami and discrete Voss surfaces, so that one-DOF kinetic motion exits.

In order to understand this behavior, we define extended "developed state" and "flat-folded state" as follows:

Developed state: A flat state in which every edge has rotational angle of 0 (thus foldlines are unfolded and complementary foldlines are folded).

Flat-folded state: A flat state in which every edge has rotational angle of ±π (the sign shows mountain or valley).

Here, notice that complementary foldlines and foldlines are interchangeable by calling a developed state a flat-folded state and vice versa.

Here we define *bidirectionally flat-foldable planar quadrilateral mesh (BDFFPQ mesh)* a 4-valency network mesh surface that follows the following conditions.

- The surface is polyhedral, i.e., composed of planar facets, and every edge is either FL or CFL.

- The surface has developed and flat-folded states, where the intersection of the overlapped facets is ignored.
- Each interior vertex's incident edges are either 4 FLs, 4 CFLs, or 2 FLs + 2CFLs in which a pair of CFLs or FLs is an opposite pair.

Rigid foldability of such a surface is represented as follows when the surface is homeomorphic to a disk.

**Theorem 1** *If and only if BDFFPQ mesh homeomorphic to a disk with more than one interior vertex has one intermediate folded state, the surface is finitely rigid-foldable.*

**Proof:** Rigid-foldability of a polyhedral disk surface can be represented by the existence of a continuous solution in the configuration space. From the developability and flat-foldability condition, it follows that subnetwork composed of only FLs (CFLs) must divide the surface completely into one or multiple regions each of which is assigned with either $+1$ or $-1$ such that the signs of adjacent regions are opposite. The sign indicates the orientation of facet in the flat-folded (developed) state. Then the developability and flat-foldability of the network for each vertex is represented as follows:

$$\text{developability:}\quad \begin{cases} \sum_{i=0}^{3}\sigma^{\text{dev}}(i)\theta_i = 0 & \text{for 4CFL or 2FL + 2CFL vertex} \\ \sum_{i=0}^{3}\theta_i = 2\pi & \text{for 4FL vertex,} \end{cases} \tag{6}$$

$$\text{flat-foldability:}\quad \begin{cases} \sum_{i=0}^{3}\sigma^{\text{ff}}(i)\theta_i = 0 & \text{for 4FL or 2FL + 2CFL vertex} \\ \sum_{i=0}^{3}\theta_i = 2\pi & \text{for 4CFL vertex,} \end{cases} \tag{7}$$

where $\sigma^{\text{dev}}(i)$ and $\sigma^{\text{ff}}(i)$ indicate the assigned signs of the facet incident to the sector angle $i$ in the developed and flat-folded states respectively.

A 2FL+2CFL vertex satisfies $\sum_{i=0}^{3}\sigma(i)\theta_i = 0$ for both $\sigma$ in developed and flat-folded states ($\{\sigma(0),\sigma(1),\sigma(2),\sigma(3)\} = \{1,1,-1,-1\}$ and $\{1,-1,-1,1\}$); therefore $\theta_0 = \theta_2$ and $\theta_1 = \theta_3$. Hence, the vertex is a discrete Voss vertex and the folding motion follows (5). A 4FL or 4 CFL vertex is essentially a vertex of Miura-ori and the folding angles of incident edges follow the kinetic motion represented by (3). Since we represent the folding angles of complementary foldlines by their complementary angles, these relations can be represented in the following single form:

$$\tan\frac{\rho_i}{2} = A'_{i,j}\tan\frac{\rho_j}{2},$$

where $A'_{i,j} = 1/A_{i,j}$ for 4CFL vertices. Since all the foldlines are connected to each other, the transformation follows (3). Therefore if and only if there exists one intermediate configuration $\left\{\tan\frac{\rho_i(t_0)}{2}\right\}$, there exists a continuous valid configurations $\left\{\tan\frac{\rho_i(t)}{2}\right\}$. □

## 3 Design Variations of Rigid Foldable Surface

For obtaining design variations, we adopt the perturbation based approach used in [Tachi 2009a]. We first obtain a valid existing BDFFPQ mesh in a flat state; this can be done quite easily, e.g., by using a square grid and assigning mountain, valley, complementary mountain, and complementary valley to the edges. Here, we can roughly design the kinetic behavior since an origami vertex tries to expand the surface in two directions at the same time, while a discrete Voss vertex tries to expand in one direction while collapsing in the other direction.

Then we deform the pattern while sustaining the developability and flat-foldability conditions. The configuration of the mesh surface is represented by the coordinates of vertices $\mathbf{x}$ forming a triangular mesh whose edges are either triangulation edges or complementary or ordinary foldlines. This triangular mesh satisfies the planarity condition for each facet and flat-foldability (7) and developability (6) conditions for each interior vertex. Thus an infinitesimal deformation of mesh configuration must satisfy:

$$\mathbf{c}(\mathbf{x}) = \begin{bmatrix} \mathbf{c}^{\text{dev}}(\mathbf{x}) \\ \mathbf{c}^{\text{ff}}(\mathbf{x}) \\ \mathbf{c}^{\text{planar}}(\mathbf{x}) \end{bmatrix} = \mathbf{0}, \tag{8}$$

where $\mathbf{c}^{\text{dev}}(\mathbf{x})$ is a $N^{\text{Vint}}$-vector ($N^{\text{Vint}}$ is the number of interior vertices) whose element is

$$\begin{cases} \sum_{i=0}^{3} \sigma^{\text{dev}}(i)\theta_i & \text{for 4CFL or 2FL + 2CFL vertex} \\ 2\pi - \sum_{i=0}^{3} \theta_i & \text{for 4FLvertex,} \end{cases}$$

$\mathbf{c}^{\text{ff}}(\mathbf{x})$ is a $N^{\text{Vint}}$-vector whose element is

$$\begin{cases} \sum_{i=0}^{3} \sigma^{\text{ff}}(i)\theta_i & \text{for 4FL or 2FL + 2CFL vertex} \\ 2\pi - \sum_{i=0}^{3} \theta_i & \text{for 4CFL vertex,} \end{cases}$$

and $\mathbf{c}^{\text{planar}}(\mathbf{x})$ is a vector with dimension of the number of triangulating edges, and its element is its corresponding folding angle $\rho_i$.

Since the number of variables exceeds the number of constraints for a quadrilateral mesh structure with multiple boundary edges, the Jacobian matrix of the constraint $\left[\frac{\partial \mathbf{c}}{\partial \mathbf{x}}\right]$ is a rectangular matrix whose number of columns exceeds the number of rows. An infinitesimal solution space of this constraint can be calculated as:

$$\Delta\mathbf{x} = \left(\mathbf{I} - \left[\frac{\partial \mathbf{c}}{\partial \mathbf{x}}\right]^{+} \left[\frac{\partial \mathbf{c}}{\partial \mathbf{x}}\right]\right) \Delta\mathbf{x_0}, \tag{9}$$

where $\left[\frac{\partial \mathbf{c}}{\partial \mathbf{x}}\right]^{+}$ is the pseudo-inverse (Moore-Penrose generalized inverse) of the Jacobian matrix, and $\Delta\mathbf{x_0}$ represents an arbitrary vector. Equation (9) calculates the valid perturbation closest to $\Delta\mathbf{x_0}$ by orthogonal projection to the solution space; therefore, the user input such as the drag motion of vertices through GUI can be used as $\Delta\mathbf{x_0}$ in the implementation system of the method. A larger deformation of the shape can be achieved by accumulating small deformation using Euler integration while eliminating the residual **c** by the Newton-Raphson method. Figure 4 shows an example process of design in a design system that can interactively solve Equation 9 while displaying the 3D configuration, flat-folded state, and developed state of the surface. Figure 4 shows an example process of design in a design system that can interactively solve Equation 9 while displaying the 3D configuration, flat-folded state, and developed state of the surface.

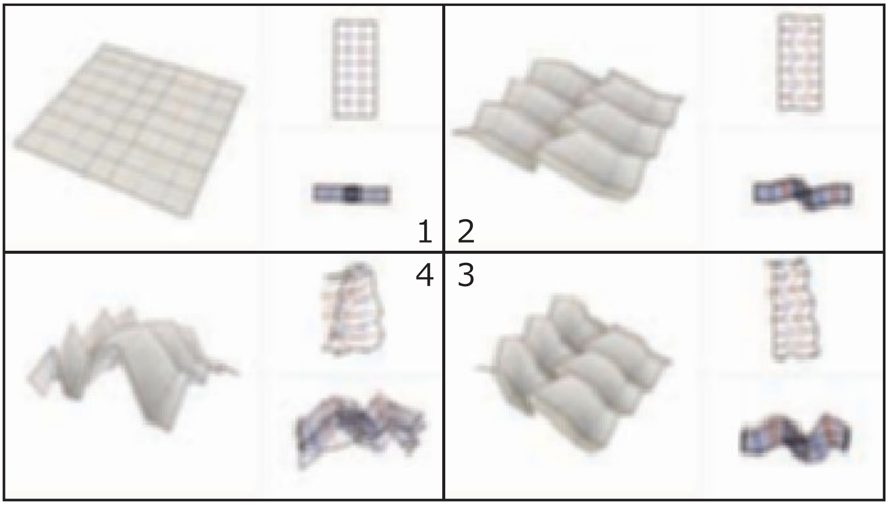


Figure 4: The design process of hybrid pattern obtained by deforming a square grid. Red and dark blue lines indicate mountain and valley foldlines, and pink and light blue indicate complementary mountain and valley foldlines. Each screen shows 3D configuration (left) developed state (upper-right) and flat-foldable shape (lower-right).

The proposed method succeeded in generalizing eggbox pattern and hybrid BDFFPQ mesh. Figure 5 shows an example of a generalized eggbox pattern that has “globally” positive curvature surface, and Figure 6 is an example complex foldable structure based on hybrid surfaces.

## 4 Rigid Foldable Cylinder

In an architectural context, collapsible cylindrical surfaces are a significant design target since a cylinder qualitatively surrounds a volume of space. Here, note that a cylindrical surface is one of the best possible solutions in this direction since a

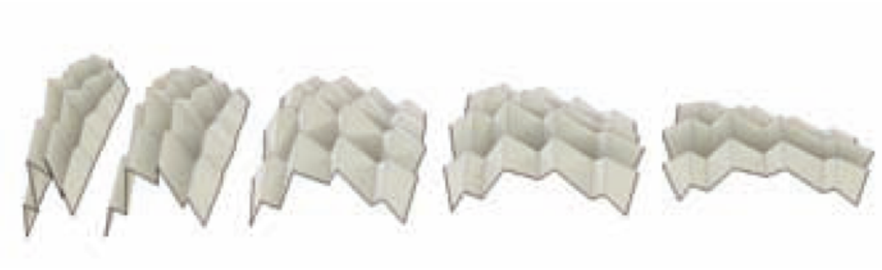


Figure 5: The folding motion of a generalized eggbox pattern.

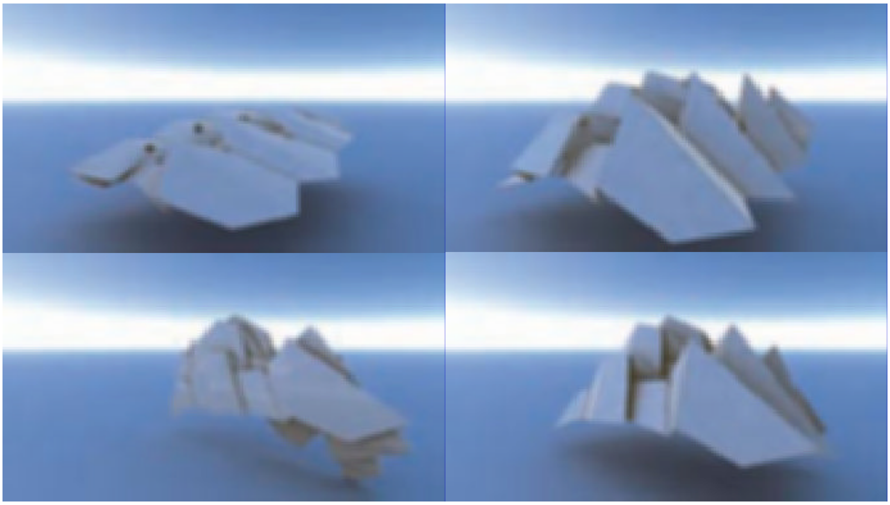


Figure 6: The folding motion of hybrid BDFFPQ mesh.

topologically closed polyhedron, which is ideal in the sense of enclosure of a space, is actually known to forbid a rigid folding motion with volume change as the bellows conjecture [Connelly et al. 1997] states. Although several designs are proposed to enable collapsible and non-rigid-foldable cylinders such as triangulated cylinders [Guest and Pellegrino 1994] and single curved corrugated surface [Hoberman 1993], these exiting designs relied on the flexibility of each facet for the folding motion; they were not applicable to smooth kinetic mechanisms that work with stiff materials and mechanical hinges often required for architectural-scale structures.

[Tachi 2009b] gives a family of cylindrical surfaces and compounds of cylinders that are rigid-foldable and flat-foldable, by solving the angular identity equations via constructing a symmetric modular cylinder based on rotational symmetry and then array copying the obtained module in an axial direction. This successfully produces

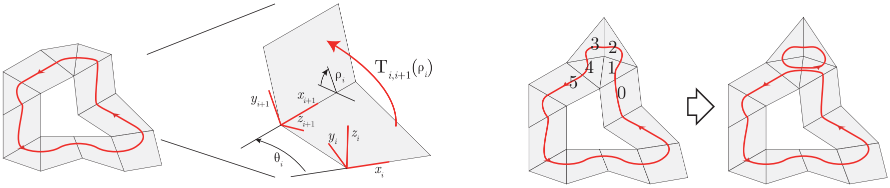


Figure 7: A closed strip of facets (Left). A loop can continuously shrink or expand using conditions around vertices (Right).

a folding motion that does not separate the structure into an open disk. However, the derived surface can only have a constant section along an axis in this approach, and the design variations are limited. Producing freely designed cylindrical surface is the goal of the work described in this section.

The biggest issue in the design of cylindrical surface is that the local condition around each interior vertex is not enough because loops surrounding a hole can geometrically separate by folding; and therefore cylindrical BDFFPQ mesh is not proved to be rigid-foldable in general. In general, it is required that for any closed strip consisting of $k$ facets, the strip does not separate by the folding motion of hinges. We assign a local coordinate to each facet $i$ and denote the local affine transformation from $i$-th to $i+1$-th coordinates by $4 \times 4$ matrix $\mathbf{T}_{i,i+1}(\{\rho_i\})$ (Figure 7 Left). This transformation is regular and the inverse can be described as $\mathbf{T}_{i,j}^{-1} = \mathbf{T}_{j,i}$. The condition thus is represented as, *there exists a vector function $\rho_i(t)$ ($0 \leq t \leq \pi$) that satisfies the following for any loop of facet strip:*

$$\mathbf{T}_{0,1}\mathbf{T}_{1,2}\cdots\mathbf{T}_{i,i+1}\cdots\mathbf{T}_{k,0} \equiv \mathbf{I}. \tag{10}$$

If the facet fan around each vertex is ensured to be rigid-foldable, loop condition around the vertex modifies the condition along a loop to another homologous loop. For example, in Figure 7 right, the transformation from facet 0 to 5 can be simplified as:

$$\mathbf{T}_{0,1}\mathbf{T}_{1,2}\mathbf{T}_{2,3}\mathbf{T}_{3,4}\mathbf{T}_{4,5} \equiv \mathbf{T}_{0,1}\mathbf{T}_{1,4}\mathbf{T}_{4,5}, \tag{11}$$

using the condition around an interior vertex: $\mathbf{T}_{1,2}\mathbf{T}_{2,3}\mathbf{T}_{3,4}\mathbf{T}_{4,1} \equiv \mathbf{I}$. Therefore, the rigid-foldability of a disk with a hole (or $n$ holes) can be represented by the combination of the local rigid-foldability around each interior vertex and the condition around the hole (or the $n$-holes).

Since BDFFPQ mesh satisfies local rigid-foldability conditions around vertices, the problem of obtaining a rigid-foldable cylindrical BDFFPQ mesh turns into a problem of obtaining a state with one rigid-foldable loop connected to the surface. We transform an existing rigid-foldable cylinder while ensuring the rigid-foldability of one loop satisfied. In order to do this, we use an isotropic type of rigid-foldable cylinders from [Tachi 2009b] as an initial state, which are symmetrically repeating structures constructed using the combination of degree-4 vertices termed *folds* and *elbows* producing a valid loop motion, where a fold and an elbow are symmetric cases of origami vertex and discrete Voss vertex, respectively, in our interpretation. Therefore the cylinder is a symmetric type of rigid-foldable cylindrical BDFFPQ.

Since the exact rigid-foldability condition around a loop is still not fully investigated, we guarantee the rigid-foldability of the whole model by using a sufficient condition: one loop around the hole is unchanged from the original symmetric cylinder. We pick up a strip loop along one of the boundaries that exactly produces a one-DOF motion and rigidify this part using a rigid bar model, while we deform other parts under the local conditions of developability, flat-foldability, and planarity of polygons. The developability and flat-foldability keep the sector angles incident to vertices between rigid strip and flexible BDFFPQ strips fixed and enable a one-DOF motion.

We obtained asymmetric form variations from one symmetric cylindrical structure using the design system that solves BDFFPQ mesh and extra constraints. The design process is shown in Figure 8, and Figure 9 shows the derived crease pattern of the design. The pattern shows the location of foldlines in the developed state (note that the lines are drawn on multiple layers since we use extended definition of developed state). Here, we can notice sets of parallel lines remaining in the generalized pattern. This implies the limitation of our method since this globally symmetric behavior forbids cylinders to change their overall radius; this may restrict the design applications of the cylinders. Since this behavior presumably originates in the fixed boundary of the surface, investigating the exact rigid-foldability condition around a loop and loosening the design constraints can contribute to more flexible designs of rigid-foldable cylinders, which still remains to be a future work.

## 5 Materialization

Because of the developability of the surface, the structure applied for a small scale object can be manufactured from sheets of material such as paper and plastics. The pattern can be first perforated by cutting plotter, laser cutter, or other CNC 2-axis machines on sheets of thin and hard materials, then pasted together, and folded to form a three-dimensional shape. Careful folding control is needed only at the beginning of the fold; once it is semi-folded, the surface folds automatically since the pattern produces a one-DOF motion. Figure 10 shows a folded model of a rigid-foldable cylinder designed using the proposed method.

For a larger scale architectural structure such as shades, retractable roofs, and transformable partitions, the following thickening method can be used to produce a robust kinetic structure. Each facet is first substituted by two layers of constant thickness panels whose contact plane is the ideal geometric surface. In order to avoid the intersection of panels by the folding motion of $0 \leq \rho_i \leq \pi - \delta_i$ for each foldline, the boundary of the panel in the valley side of the surface is relocated on an offset of the foldline by $t \cot \frac{\delta}{2}$, where $t$ is the thickness of the panel (Figure 11). This yields

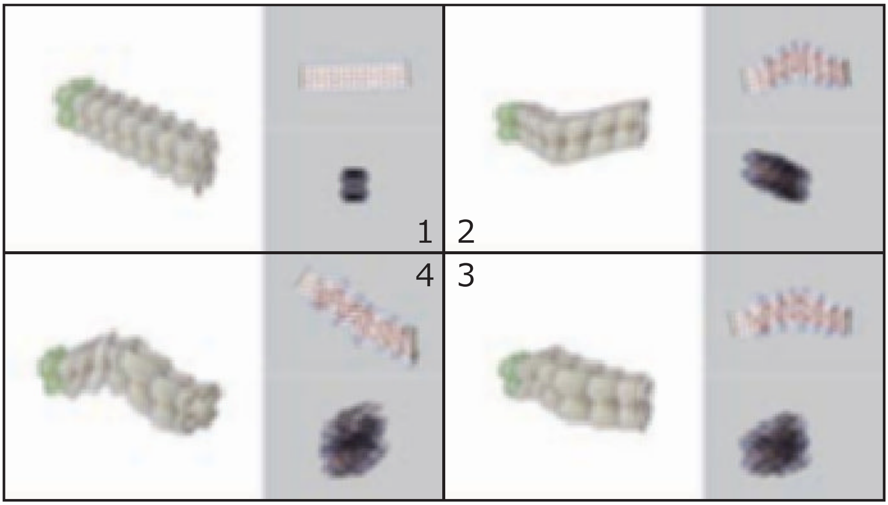


Figure 8: A design process of rigid-foldable cylinder. The initial state is a cylinder by [Tachi 2009b]. The surface is deformed keeping the condition of BDFFPQ mesh while a loop strip (two strips in this redundant case) is maintained to be rigid, as indicated by green rigid bars.

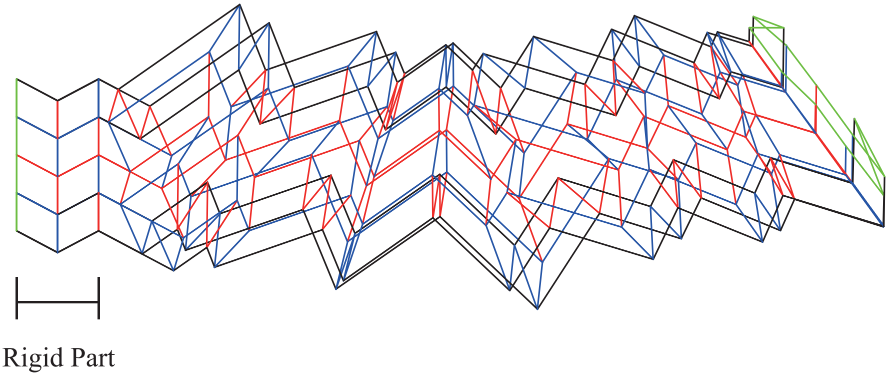


Figure 9: A crease pattern of generalized cylinder.

designs of kinetic structure composed of rigid panels using their edges as rotational hinges as shown in Figures 12 and 13. The hinges can be constructed mechanically or using a sheet or cloth of negligible thickness between panels; in the latter method, the cloth also becomes a watertight covering.

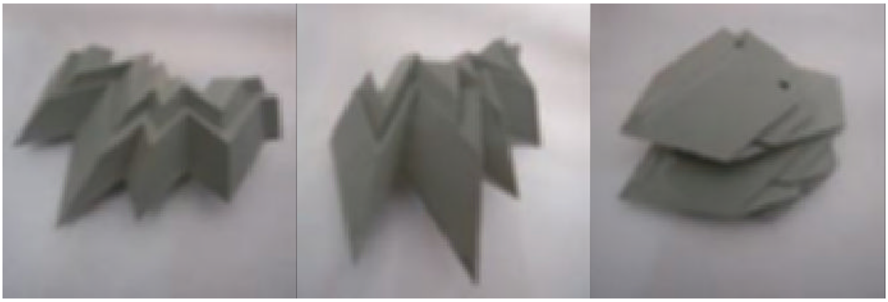


Figure 10: Folded model of a rigid-foldable cylinder. Left: in the developed state, Middle:Intermediate state, Right: flat-folded state.

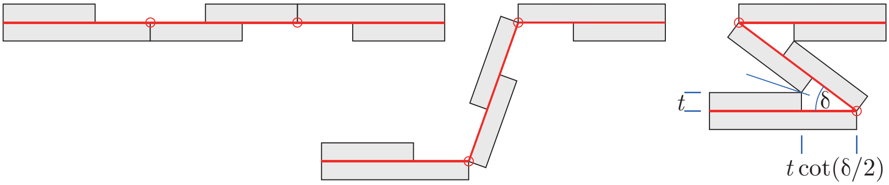


Figure 11: Thickening using two layers of constant thickness panels.

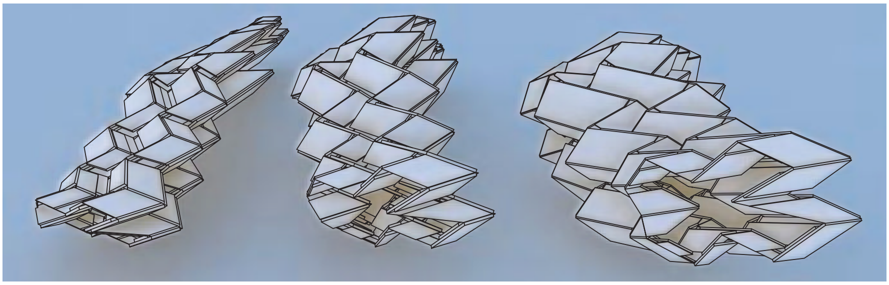


Figure 12: Cylindrical rigid-foldable BDFFPQ mesh consisting of thick panels.

## 6 Conclusion

In this paper, we presented a design method of rigid-foldable flat-foldable disks and cylinders by introducing bidirectionally flat-foldable planar quadrilateral (BDFFPQ) meshes and their generalization method. The concepts of BDFFPQ mesh, complementary foldlines, and extended definitions of developed and flat-folded states successfully unify and generalize the flat-foldable origami and discrete Voss surfaces. This generalization enables the perturbation based method that can

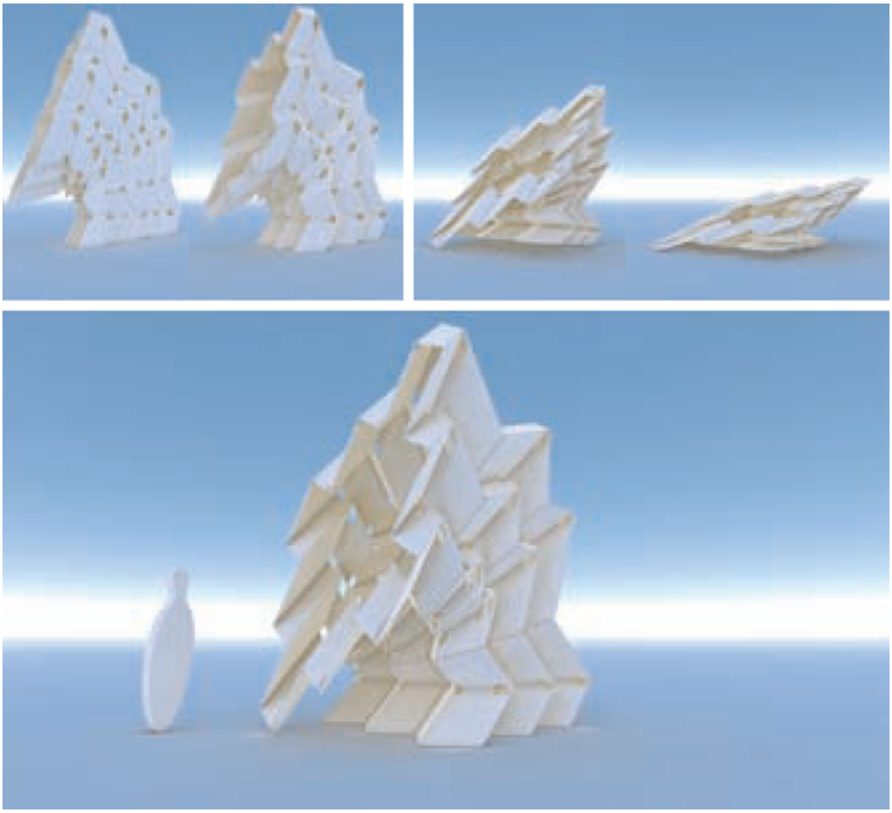


Figure 13: Folding motion of cylindrical structure using thick panels.

freely control the shape of the surface in a direct way; the method yields various designs of rigid-foldable and bidirectionally flat-foldable polyhedral disk. We also investigated the rigid-foldability of a cylindrical surface and its sufficient condition to design a rigid-foldable cylinder. A design method by modifying a cylindrical BDFFPQ mesh with one fixed loop copied from a known rigid-foldable cylindrical structure is introduced. The method enables us to obtain a novel generalized form of rigid-foldable flat-foldable surfaces that can be implemented for architectural design purposes.

It should be noted that we used sufficient condition of the rigid-foldability in this paper and did not provide the exact condition, which still remains to be an unsolved problem as it is seen in [Stachel 2010]. The sufficient condition for enabling cylindrical topology also made the deformation less flexible, the effect of which can be observed in the globally parallel edges. Deriving a more general condition of the rigid-foldability of cylindrical surface and thus enabling a less constrained design is one of the future works of this study.

## References

BRUNNER, A., 1965. Expansible surface structure. United States Patent 3,362,118.

CONNELLY, R., SABITOV, I., AND WALZ, A. 1997. The bellows conjecture. *Contributions to Algebra and Geometry 38*, 1, 1–10.

GUEST, S. D., AND PELLEGRINO, S. 1994. The folding of triangulated cylinders, Part I: Geometric considerations. *ASME Journal of Applied Mechanics 61*, 773–777.

HOBERMAN, C., 1993. Curved pleated sheet structures. United States Patent No. 5,234,727.

HUFFMAN, D. 1976. Curvature and creases: a primer on paper. *IEEE Transactions on Computers C-25*, 10, 1010–1019.

MIURA, K. 1980. Method of packaging and deployment of large membranes in space. In *31st Congress of the Internatinal Astronautical Federation.*

SCHIEF, W. K., BOBENKO, A. I., AND HOFFMANN, T. 2007. On the integrability of infinitesimal and finite deformations of polyhedral surfaces. In *Discrete Differntial Geometry (Oberwolfach Proceedings)*, 67–93.

STACHEL, H. 2010. A kinetic approach to Kokotsakis meshes. *Computer Aided Geometric Design*, 27, 428–237.

TACHI, T. 2009. Generalization of rigid-foldable quadrilateral-mesh origami. *Journal of the International Association for Shell and Spatial Structures 50*, 3 (December), 173–179.

TACHI, T. 2009. One-DOF cylindrical deployable structures with rigid quadrilateral panels. In *Proceedings of the IASS Symposium 2009*, 2295–2306.

TACHI, T. 2009. Simulation of rigid origami. In *Origami$^4$: The Fourth International Conference on Origami in Science, Mathematics, and Education*, A K Peters, R. Lang, Ed., 175–187.

# The Sphere Project - Negotiate geometrical representations from design to production

**Klaus Bollinger**

Bollinger + Grohmann Ingenieure

**Manfred Grohmann**

Bollinger + Grohmann Ingenieure

**Oliver Tessmann**

Bollinger + Grohmann Ingenieure

**Abstract.** *The Sphere is a sculpture designed by Mario Bellini for the lobby of the Deutsche Bank head office in Frankfurt, Germany. A spherical network with a diameter of sixteen meters spans between the twintowers. This paper describes the ollaborative design process of architects, engineers and contractor. The various geometrical implications from design synthesis to fabrication and the exchange of information between the different parties are discussed. The concept of a spherical network was transformed into a series of sixty intersecting rings that act together as one sculptural and structural object. The ring configuration is derived from an evolutionary design process that includes geometrical and structural fitness criteria. Evolutionary design, structural analysis and fabrication demanded different geometrical representations that addressed the various constraints emerging during design and construction.*

## 1 Introduction

The Sphere is a sculpture designed by Mario Bellini for the lobby of the Deutsche Bank head office in Frankfurt, Germany. The sculpture is one element of a large-scale refurbishment that is set to turn the headquarters into an eco-friendly high-rise building.  In the competition phase the spherical network of threads was represented by a texture mapping in a rendering. The image successfully conveyed the design intent of Mario Bellini and suggested a spatial quality within the entrance area.

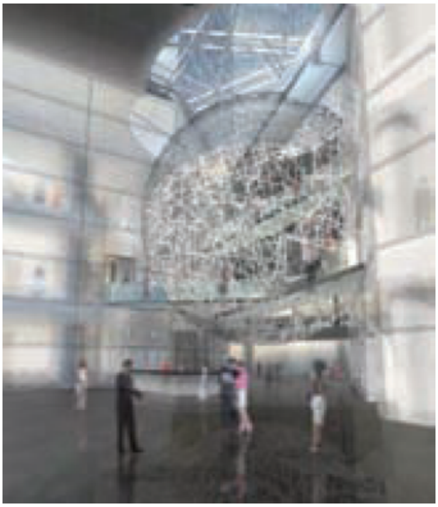


Figure 1: Early rendering with mapped structure. (Source: Mario Bellini Architects)

Nevertheless it had to be transferred into actual geometry which works as the basis for a structural and material system. In close collaboration with Mario Bellini Architects Bollinger + Grohmann developed a geometrical approach of a series of rings with various radii that populate the surface of the sphere.

## 2 Geometry generation

The distribution of rings on the Sphere's surface was the objective of a Genetic Algorithm. The goal of the evolutionary design process was a configuration of sixty rings that serve as a structural system, while at the same time providing an even distribution of rings that do not intersect with two bridges that penetrate the spherical surface. These requirements served as fitness criteria in a process of automated generation of geometrical and structural models and their evaluation, ranking and selection.

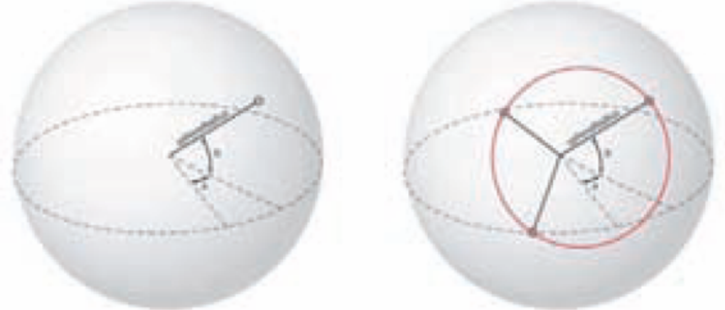


Figure 2: Every ring is based on a circle that is described by three spherical coordinates. (Source: Bollinger + Grohmann)

## 2.1 Geometric principles

The genome of the evolutionary process carried the information of all coordinates of every circle. It was fed into the 3D modeling application Rhinoceros© via a custom made Visual Basic application. Here a copy of the circle is offset towards its center point which defines the depth of the ring (Figure 3, left). The ring surface is extruded to form a volume which represents the steel profiles.

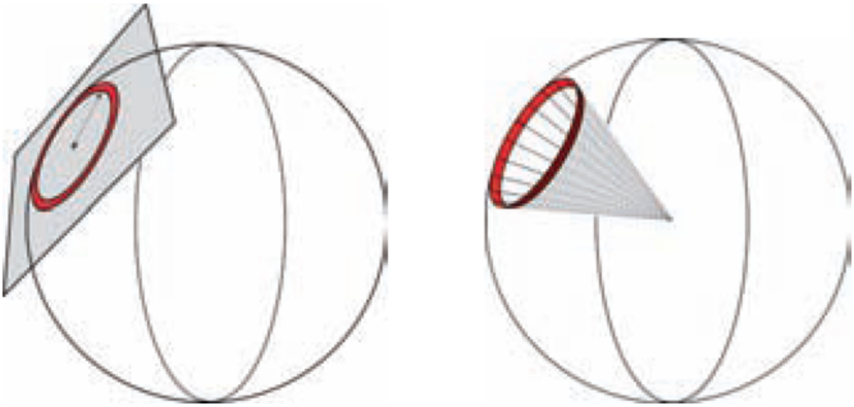


Figure 3: Two different ring orientations were investigated. Left: A ring which is derived from a planar curve offset of the circle. Right: A ring as a cone segment derived from a cone between the sphere center point and the circle. The former version was chosen because it could be easily fabricated from sheet material whereas the second version requires unrolling the single-curved surface. Nevertheless the different ring orientations of Version 1 created more complex ring intersections (Figure 4, Figure 5). (Source: Bollinger + Grohmann)

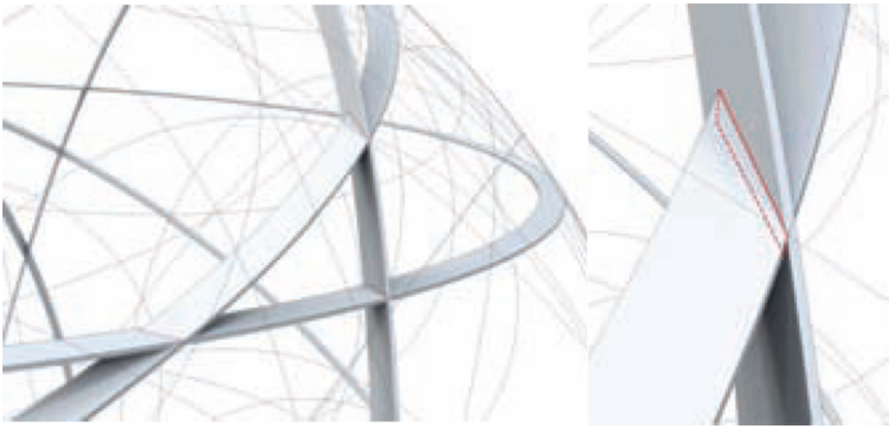


Figure 4: Planar rings never fully intersect because of different plane orientation. (Source: Bollinger + Grohmann)

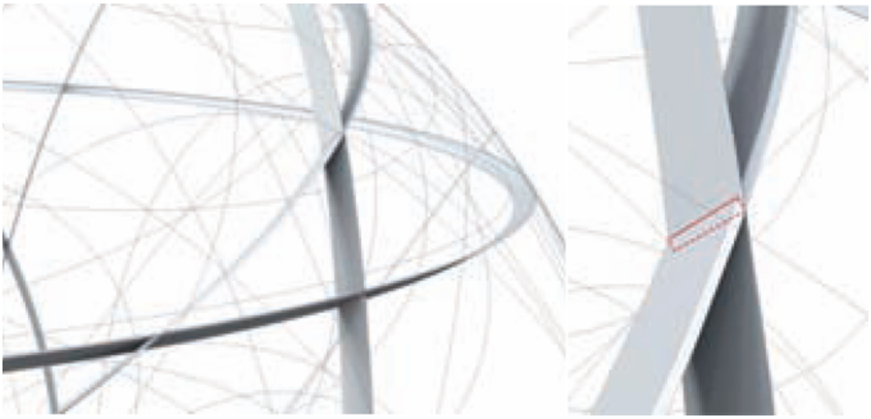


Figure 5: Similar topology of circles but different ring orientation. Cone segments fully intersect because every element is orientated towards the sphere's center (Source: Bollinger + Grohmann).

The 3D model served as a phenotype which could be evaluated according to the three fitness criteria. The geometrical aspects like the even ring distribution and possible conflicts with the bridges could be evaluated in the Rhinoceros© model. The structural performance evaluation required a neutral fiber model with circles converted into polygons.

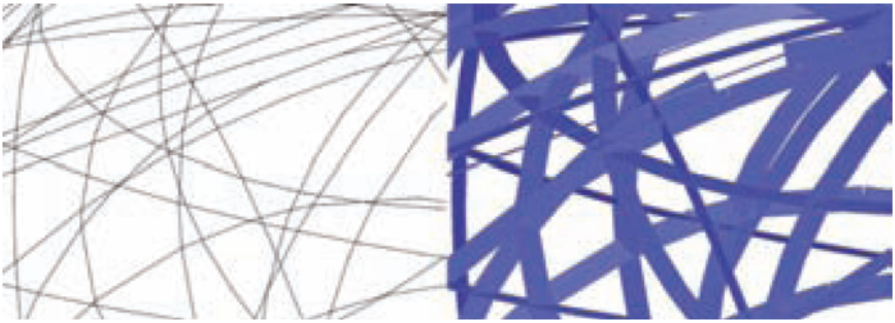


Figure 6: Polygonized rings in RSTAB©. The flat steel bars are oriented towards their circle center (Source: Bollinger + Grohmann).

Thus the circle intersection points were detected and the arcs between those intersections were converted into polylines which guaranteed a sufficient approximation of the original geometry. Nevertheless for the cross-section orientation of the linear segments it was very important to maintain the information of the initial circle, the element it is derived from  since the flat bar profiles were oriented towards the circle center point.

### 2.2 Fitness criteria for the evolutionary process

The structural systems and the sculpture coalesce into one single geometrical construct. No additional load bearing elements are necessary. The entire sphere is supported by the two towers it connects and intersects with. In the evolutionary process the overall deflection of the system is used as a fitness criteria to evaluate the structural performance.

An even distribution of rings across the sphere is a requirement which opposes structural demands. Hanging or standing arcs that span between the towers would serve as a proper structure but result in clustering elements in one zone of the sphere. This contradictory requirement became the second fitness criteria. It was quantified by measuring the angles between the circle planes. All angles were summed-up. A large number meaning wide angles between the rings and therefore a better distribution across the sphere.

The third criteria related to the context of the sphere. Two bridges penetrate the sculpture and improve the connection between the two towers. The rings should not interfere with the required clearance for construction and use of the bridge. Truncated rings would form threatening blades next to the circulation area. Thus the number of rings that intersect the clearance were counted and bad fitness values were assigned to individuals with a high number of collisions. The Genetic algorithm delivered a suitable ring configuration which was approved by the architects and became the basis for the subsequent design phases.

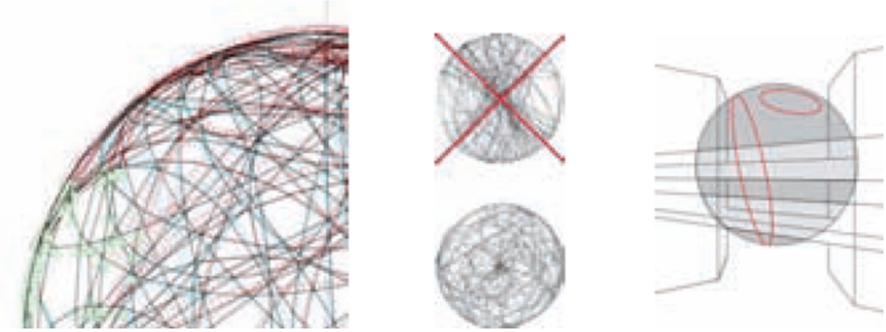


Figure 7: Three fitness criteria. Left: Deflection of the structure, Middle: Distribution of rings, Right: Collisions btw. rings and bridge clearance (Source: Bollinger + Grohmann).

## 3 Geometry refinement for fabrication

The evolutionary process yielded a sphere configuration which adapted to the various requirements of structure, context and geometry. Steel was chosen as the appropriate material which performs best in terms of structure/weight ratio, budget and fire resistance. The flat steel bars could be cut from sheet material, forming or bending was not necessary. No additional coating or cladding was intended, so that visitors can experience the ambiguity of a single sphere assembled by multiple rings.

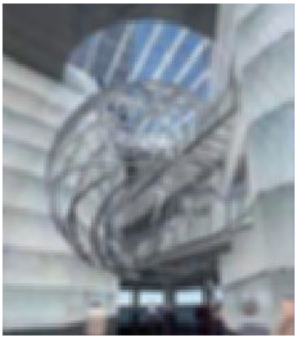


Figure 8: The Sphere with a configuration of sixty rings. Compared to the initial visualization the Sphere is now comprised of fewer but  more ordered elements. The generative rules are recognizable (Source: Mario Bellini Architects).

The first full-scale mock-up made from flat steel bars revealed a geometrical problem which did not seem so relevant in the digital model: The planar ring surfaces were extruded in both normal directions to generate solids. Thus the ring solids partially exceed the sphere volume. To avoid possible collisions the geometry generating script was revised to limit the extrusion direction of the rings towards the center of the sphere.

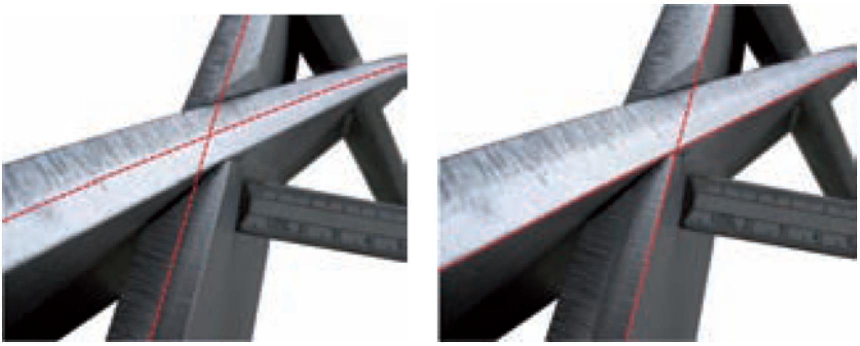


Figure 9: Different extrusion procedures from initial circle. Left: Extrusion in both directions, Right: Extrusion in one direction (Source: Bollinger + Grohmann).

Every intersection of rings was analyzed individually. The primary and the secondary rings were determined which means a primary ring stay continuous and cuts the secondary ring.

The ring segments were laser-cut from sheet material. The trimmed edge was subsequently miter milled to match the angle in which both rings intersect. Every intersection is then welded to serve as a rigid connection within the overall structure.

During workshop design stage the contractor set up his own geometrical model as the basis for construction. Thus three different models had to be maintained and updated simultaneously. RhinoScript served as an interface between the different platforms and formats which enabled the engineers at Bollinger + Grohmann to quickly analyze geometrical coherence of a thousands of nodes and elements.

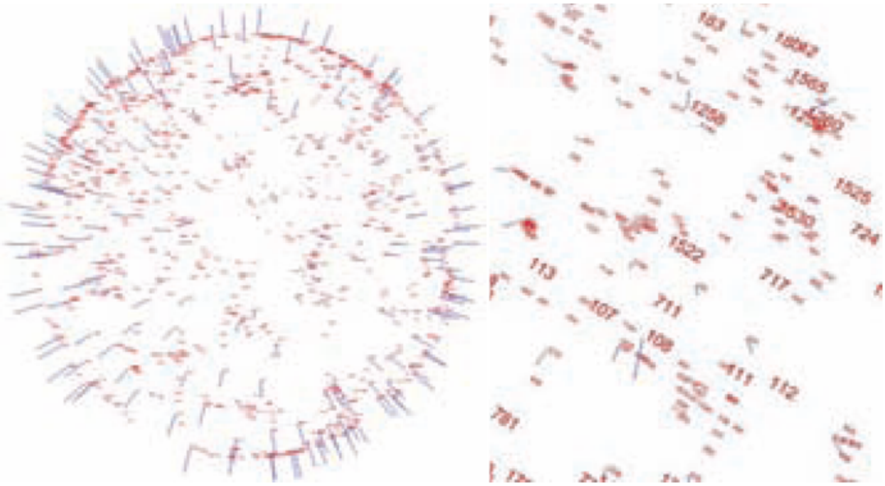


Figure 10: Mapping of nodes that need a threaded connection because of limited dimensions of sheet material. Those nodes had to be identified in the structural model to analyze the local forces. A RhinoScript analyzed the spherical markers of the contractor model and searched for the equivalent nodes in the analytical model. Node number and surface normal were placed at every node in question and transferred into the analytical model (Source: Bollinger + Grohmann).

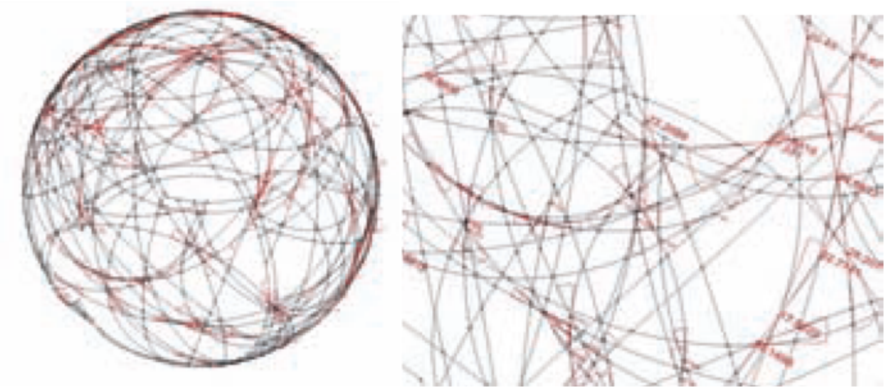


Figure 11: Automated analysis of angles between intersecting rings. Since welding of small angles is difficult and sometimes impossible every intersection with an angle smaller than 30° got tagged. Eight different welding seam types had to be developed to suit the various structural requirements (Source: Bollinger + Grohmann).

The full-scale mock-up exemplified the importance of the welding seam position. To achieve the visual effect of sixty rings assembled rather than merged the welding seam had to be placed exclusively at the longitudinal edge of the intersection. Thus the continuity of the rings is underlined rather than the node itself.

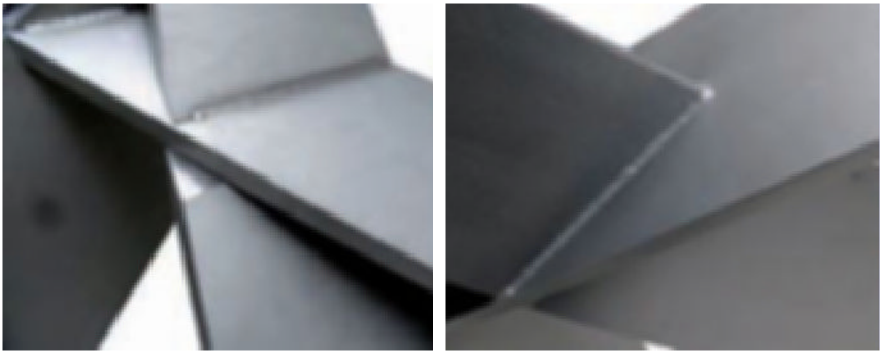


Figure 12: Two full-scale mock-ups: Detail of ring intersection with welding seam. The left proposal was favored for better embodying the idea of assembled rings (Source: Bollinger + Grohmann).

The sphere is currently under construction. After cutting the ring segments and a test-run to assemble and weld a large scale segment at the contractor's workshop of the contractor the sphere will be fully installed on site. Prefabrication of sphere segments was not considered because of transport, weight and possible tolerances on site.

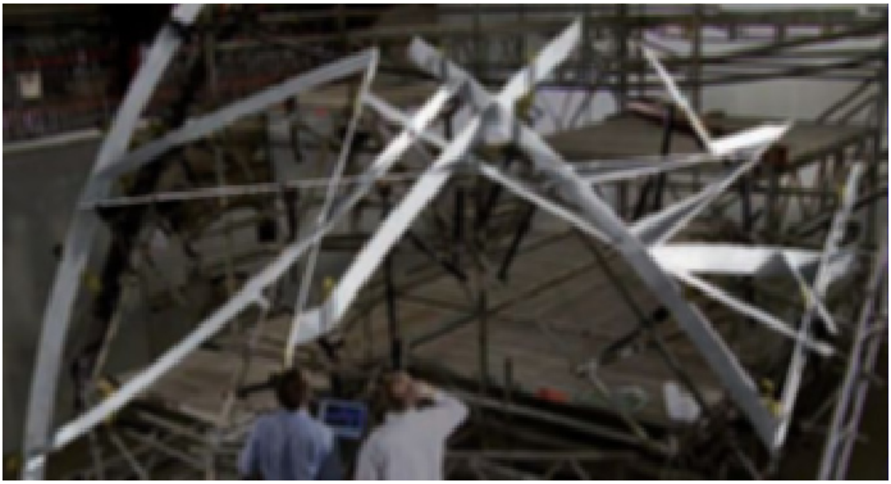


Figure 13: Test assembly by the steel contractor: Massive scaffolding and permanent surveying are needed to bring all elements into the final position (Source: Bollinger + Grohmann).

# 4 Conclusion

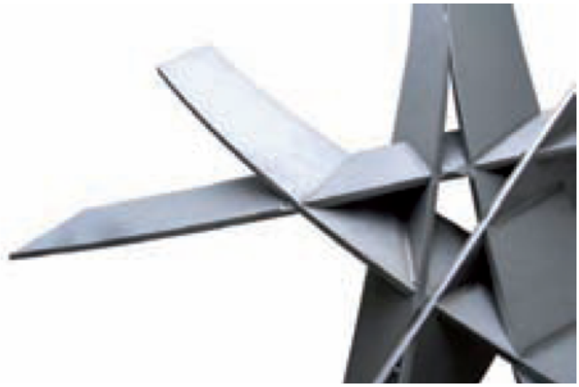


Figure 14: Mock-up of complex node (Source: Bollinger + Grohmann).

Collaborative and integrative design does not necessarily mean that all parties involved work in one single geometrical model. Different workflows, requirements, corporate infrastructure and skills of staff lead to the use of various software applications and geometrical representations. Exchange formats provided by the industry often fail to transfer the information which are relevant to the specific project. Thus interfacing various working environments, abstracting models and at the same time storing the initial information becomes part of the design process. Scripting served as a helpful approach to access the core data of the different models and to transfer them into the required format. Beyond mere automation of data exchange the algorithmic procedures allowed searching through vast solution spaces in this case by the use of Genetic Algorithms. Balancing the different requirements proved successful in this project. The subsequent refinement of the geometry for analysis and construction revealed further requirements which could have become the objective of an evolutionary design process. Due to the time-consuming character of this procedure these insights will be implemented in future projects.

## Acknowledgements

Special thanks to Markus Schein from the Kunsthochschule Kassel for the conceptual and technical support in the development of the Genetic Algorithm.

# Towards Teaching Generative Design in Architecture

**Ilija Bentscheff**

UdK Berlin

**Christoph Gengnagel**

UdK Berlin

**Abstract.** *The research presented in this paper is a consideration of the development of the discipline of architecture in terms of the emergence of digital design tools and the integration into the academic discourse and teaching. The paper focuses on three intellectual models or in this case three initial design strategies as a base for a comprehensive model for teaching and criticism.*

## 1 Introduction

The practice of architecture is increasingly defined by information exchange, communication and the possibilities of digital design and simulation systems as in the design process. Another aspect is image; the growing interest in complexity of built form, which often finds its origins in nature and biological systems, either as an analogy or as a model for generative design systems. These references and considerations relate to the question of performance in terms of structure, flow of forces, systemic integration, or its material properties and efficiencies. The employment of these systems has been subject of research for quite some time. The translation from physical model studies into structural rules (and into architectural space), for example, has been the project of Frei Otto.

The capacity of today's and tomorrow's softwares shifted this process into the realm of the digital. The pure ability of computational form-finding and performance simulation is developing a recursive phenomenon resulting in an increasing interest in even more complex forms and therefore in advanced formal-, performative- and conceptual models. Iterative design studies contain the danger of disregarding the classical notion of the relationship between form and meaning and therefore the relevance of a form towards an architectural (and sociological) context and criticism. So to say; it divorces its very own product (architectural form) from its meaning. This condition is raising the questions of form, meaning, representation and ornament. The inherent separation and focus on the form-finding and form reasoning process, by generating countless iteration of an object or form, tends to produce objects representing their own generation process and let them become dominantly self-referential to their design process. Since an architectural program can only suggest a form, but the generation process has the tendency to be so evident, it is articulating in the very Adolf Loos' definition of the ornament; the representation of the act of making it. This is followed by the notion of scale and scale-simulation and representation in a virtual environment. This raises the question how to criticize those condition or parameter driven forms and how do they inform an object-subject relationship and its phenomenological or sensual experience in the design process.

Another evident shift in this state of the discipline's development is the shift from architecture as an autonomous practice into the realm of the production of the physical project; a separation in the Albertinian sense between the designer and the builder. In today's words, design-to-fabrication; we talk about the shift from the architect and his product; the project, communicated to the fabricators via drawings towards the architect, communicating directly to the (for example robotic) fabrication devices. The immense amount of data from other disciplines (geological data, environmental surveys, structural simulation and optimization results, etc.) is feeding an information model and is recursively informing the design process and the design-object or project, and fundamentally the role of the architect.

This condition is in its complexity and variety is the base for a development of a comprehensive teaching model.

## 2 Towards Teaching

As aforementioned, the emergence and inhabitation of parametric and digital design tools and software in the design process is being described as a paradigmatic shift in the discipline of architecture. Iterative design processes and parameterized design information does not only change the act of designing, but also re-positions the design and the designer or architect in this process. We talk about a novel order in information structure, logistics and communication. The design and development process is to be seen as a non-linear process, which is being informed at any stage in any direction. Next to cultural and traditional criteria; performative, structural and logistical criteria drive and inform a project and its design process. This virtualization is redefining the evolution of a project; from design to fabrication.

We have selected three design projects which stand explanatory for three initial design concepts. The first model is the utilization of software as a design agent to develop a formal articulation. The second model is the refinement or post-parameterization of a design intend, based on a digital Finite Element Model and material simulation. The third model is the either by software or material/physical model based intend, developed in regard to geometrical, material and fabrication constraints. The differentiation and distinction of the three models is necessary for the formulation of a conceptual model and criticism, and therefore for the further development of an architectural project. This consideration is to be seen as a description of conceptual models. This condition requires an understanding of the canon of software, tools, intent and author.

The three introduced projects stand as models for their own specific approach, but show a clear thoroughly exercised development in terms of the three conceptual models.

# 3 Three Conceptual Design Models

## 3.1 Model One

The Digital Material

The introduced project is an example of the first conceptual design model; a design strategy within the digital environment.

Project: Manchester High Rise by Ilija Bentscheff (student).
Studio: The Office, Prof. P. Preissner, R. E. Somol, Critics: S. Whiting.
2008, University of Illinois at Chicago

This project addresses a conceptual investigation towards a novel office environment by developing a structure capable to contain various types of company and office typologies, an investigation on the relationship between the office (private) and the lobby (gradient of public) as the main organizing device. The lobby as the mediating space, is matured to a broad field of operations and political zones; a gradient between public and private; a space of connectivity and separation. From the clear appointed entry to the workplace, lobbies have expanded their operative field and programs, including retail, cultural venues and intermediate office functions like (press) conferences and meeting spaces of various kinds. The office itself remains generic floor plate efficiency.

Our investigation targets a fusion of the expanded operative field of the lobby and the remaining typology of the office. The office building will become a dynamic organizational system of spaces, a continuous field of rather dynamic zones of lobby, activities and office structures. The proposal may be seen as an entire zone of office and lobby-environments, providing increased possibilities for future businesses.

The initial strategy to develop the shape is based on a constellation or figuration of platonic or topological primitives as a base unit and programmatic distribution. The term Digital Material is being used to conceptualize the notion of inherent properties of platonic base geometries and their inherent topological properties which inform the evolutionary modeling process.

According to the local position of each program in the buiding, the base units are being fused and connected. This morphologic and, out of the inherent topological properties, morphogenetic process defines the overall articulation of the internal structure and its various spatial conditions like densities, hybrid spaces, haptic conditions and connection types.

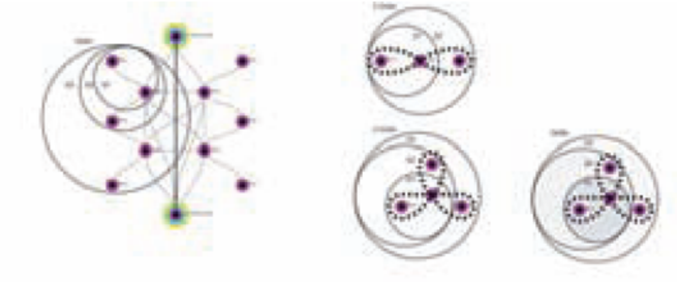


Figure 1: Internal Circulation Diagrams

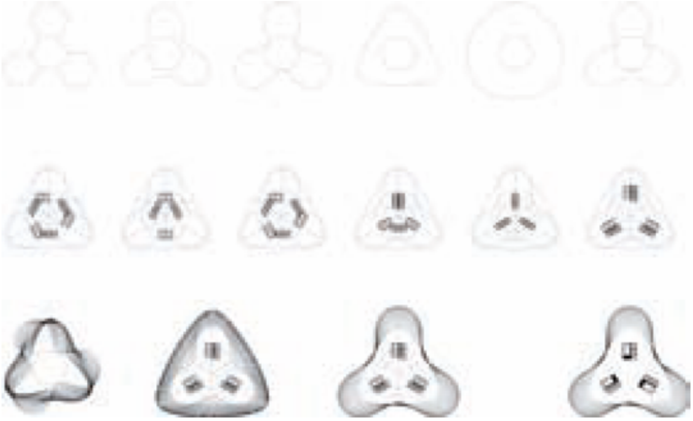


Figure 2: Floor plan Profile Evolution

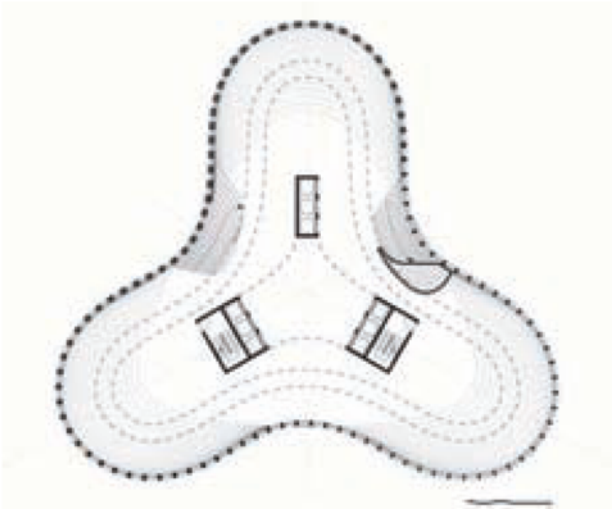


Figure 3: Developed Floor plan

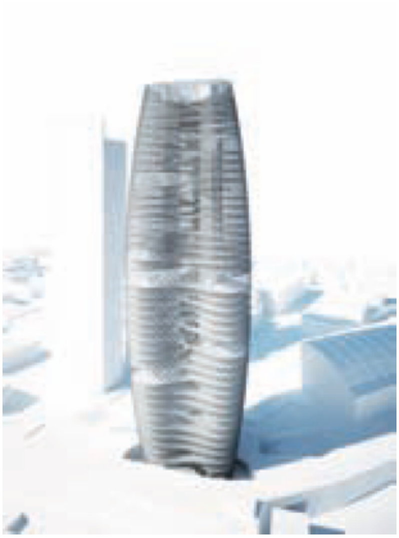


Figure 4: Tectonics of the Floor plates

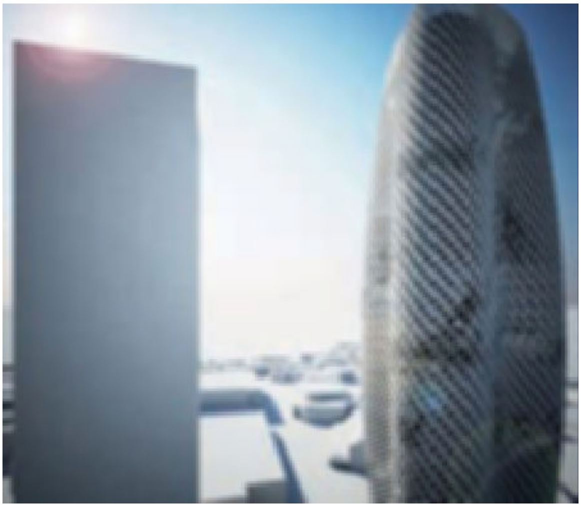


Figure 5: Rendering of the final Design

### 3.2 Model Two

Post Parameterization

This project is an example of the second model or strategy. The use of the computer and the software is essentially a tool for post-parameterization and optimization of the design task rather than a design agent (first model) and establishes a clear hierarchy between author and tool.

Project: Roof structure for the Ruin Garden of the UdK Faculty of Architecture by Felix Heisel (student)
Studio: Strukturelles Entwerfen, Prof. Dr.-Ing. C. Gengnagel, Dr.Ing. H. Alpermann and Cand. Arch. Ilija Bentscheff (instructor, Generative Design)
2009, University of the Arts Berlin

The light roof structure for the courtyard of the Ruin Garden provides a slight shadow. Its projection can be seen as an image of a shadow from the leaves of a tree in an almost kaleidoscopic agglomeration. The structural concept is a tensegrity structure. A simple twist-element is connecting a membrane between two networks of cables and is held in place by compression members. The overlay of the two twisted membrane networks is forming a moiré-like effect of shaded, semi-shaded and light-exposed areas on the ground and the walls of the courtyard, by maintaining a view through the roof structure to the sky and the surrounding trees. After the refinement of the basic concept, the development continued in Rhino-Grasshopper, by defining rules and the geometric and formal articulation, followed by the development and design of the parts, components and details. Based on the dimensions of the framing rectangle, the distance to the perimeter is driving the deformation of the lower cable network and the dimension of the membranes in each component. Each element is being evaluated and articulated according to the performance criteria; the dimensions of the compression members, the angles for the connection details, etc. The digital model has been frequently evaluated in a (digital) finite element model and a physical model in the scale of 1:10. Each component and detail in its formulation is a consequence of the initial base parameter; the base rectangle dimensions, the number of elements and the distance rule.

The initial rectangle is being subdivided into a hexagonal grid as the base pattern for the position of the tensegrity elements. The distance to the perimeter of each grid-point is defining an offset-value, which is also defining a normal vector other than purely perpendicular. The distance-value drives the dimensions of the triangular membrane components. The compression elements between the upper and lower point-grids are derived by a selection pattern, connecting the upper and lower membrane components in a twisted manner. Their section profile dimension is the consequence of their length. A further point selection by bi-nomic values in the upper and lower grid sorts the corner points for the smaller sub-membranes. The connection points are essentially Mero-System knots, detailed according to their position in the system. The knot or connection detail for the compression members went through an optimization process which affected the overall design.

The testing of the model in force-simulation software gave the initial values for the dimensions and material properties. Based on the evaluation, the knot was developed, but the detail itself did not satisfy in its necessary dimensions in relation to the design approach. It was simply too large and with a weight of eight kilograms too heavy to be considered as a successful solution. It became clear that it was a geometric problem and after various versioning attempts of the knot itself, we concluded to start exploring slight changes in the overall design of the parametric model. A slight rotation, of the more than four hundred connection points through their local center points of fifteen degrees, resulted in a significant size and weight reduction.

The development of the details, particularly of the connection detail; the knot, is explanatory for the non-linearity of the design process.

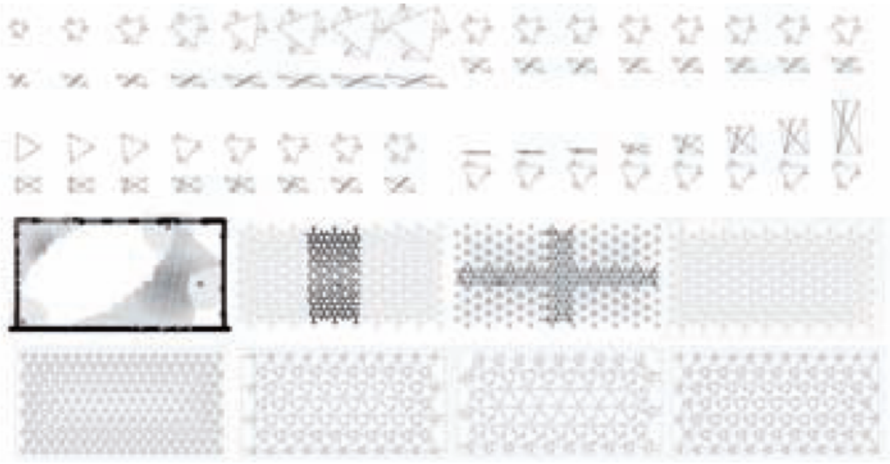


Figure 6: Scale, Twist, Height of the Elements and Configurations

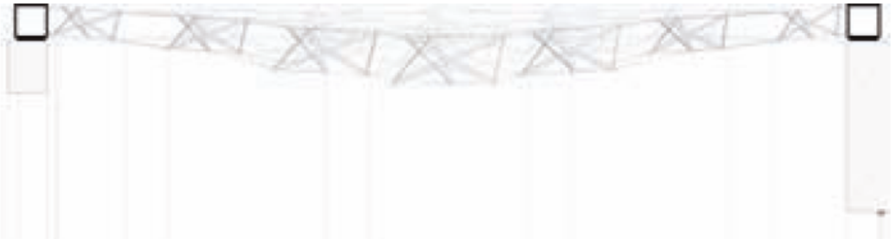


Figure 7: Cross Section of the Tensegrity Roof

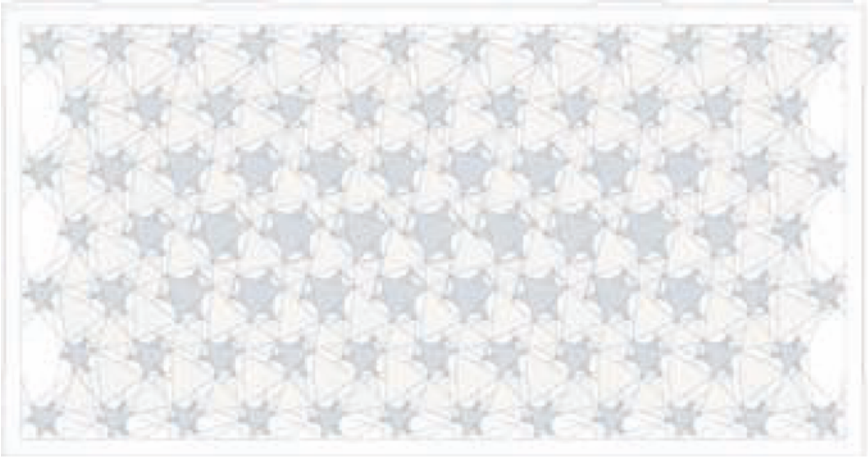


Figure 8: Top Elevation of the Tensegrity Roof

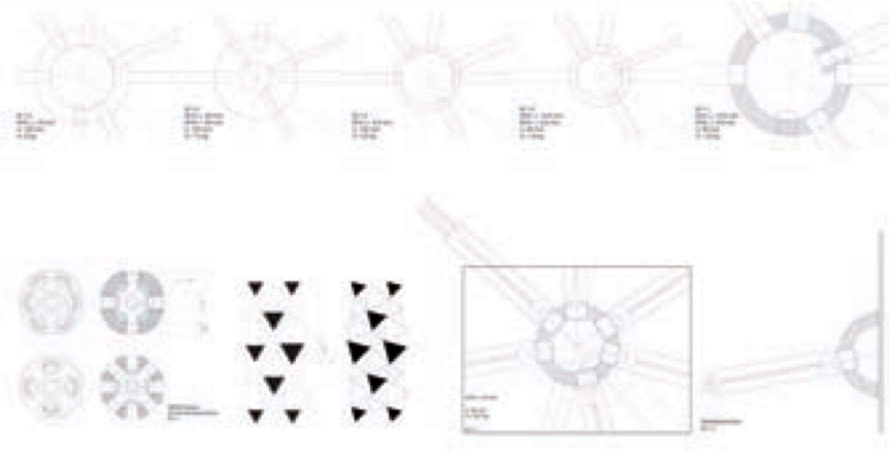


Figure 9: Detail Optimization

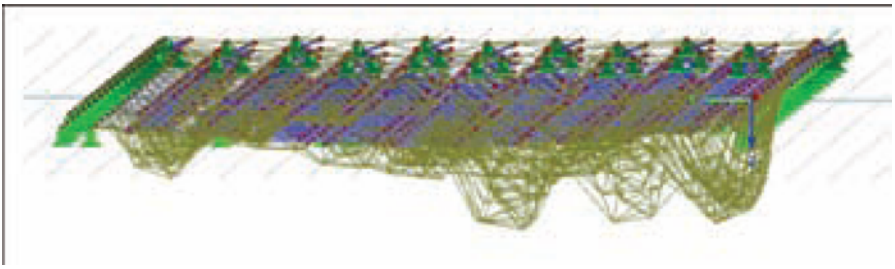


Figure 10: FEM Analysis

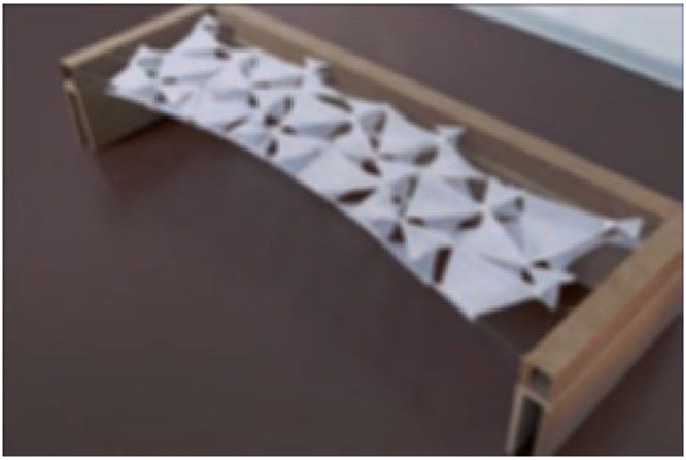


Figure 11: Physical Model, Scale 1:10

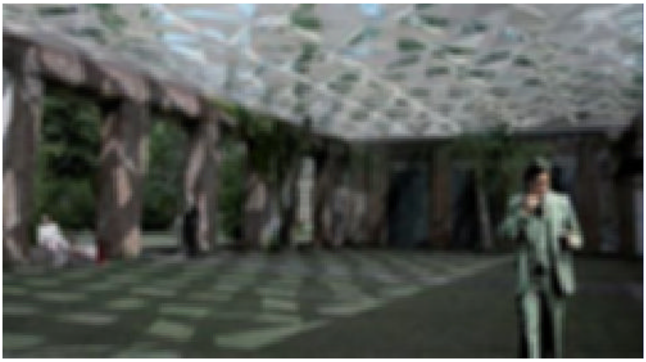


Figure 12: Rendering of the Roof covering the Courtyard

### 3.3 Model Three

Constraints and Performance

The Light Column as a project is an example for the third design model; a process of continuous optimization and recursive design information out of a digitally evolved design approach; a collaborative or canonical design process between software, material and production constraints; and most importantly, the author.

Project: The Light Column by Ilija Bentscheff (student)
Studio: TU Rule Based Design, Christophe Barlieb, Norbert Palz and Dimitrie Stefanescu, Martin Tamke.
2009, Technical University Berlin

Sand dunes and canyons are fascinating and sensational not only by their figuration; they are informed by tectonic movements and erosion. The varying densities of their sedimentary layering and the information and deformation through natural forces tell us about the climate, vegetation and events of the past. They are documentation or an atlas of history.

The lobby of the building as a place of arrival, introduction, exhibition and movement is the place for this spatial intervention. Events and movements, analogue to natural forces, inform and deform the column. It leaves the observer in the role of the interpreter of this documentation - the indexically traced past. The installation is a light column informed by the movement of people in the lobby. The column, as a tectonic element of public space, is the element with the closest association to the human body. Located at the joint between the library and the lecture hall of the architecture department of the Technical University Berlin, it places a "missing" column in the two overlaying column grids. The shape and the surface are computed in Rhino Grasshopper and are driven by multiple parameters establishing a relationship between moving or traced movements of subjects to the static, but formed and deformed object.

Movement vectors of the students and visitors of the lobby are the main forces of the deformation of the column, which is based on the platonic base geometry (topological primitive) of the cylinder; a directional generic form, as Peter Eisenman states in his dissertation towards The Formal Basis of Modern Architecture. The horizontal layering of the column refers to the concept image, the sedimentary layering of a dune. The varying distance from layer to layer is derived from the degree of deformation towards the direction of the main circulation vector from the main entrance to the elevators.

To address the fabrication, the further generation of the components is evolved as developable surfaces; a design constraint set up to direct towards the fabrication and material limitations for the prototype production. This constraint informed the design process by the deformation-degree which drives articulation of the rings, which diffuse the light coming from the inside of the column. The varying layering of the rings articulates in a light pattern of various intensities and forms a relationship between the overall shape and its surface condition. This blurred light pattern is creating an atmosphere of calmness and leaves the spectator in an unmediated relationship to the object and its meaning; a representation of documentation or a trace of a past presence.

Figure 13: Conceptual Image

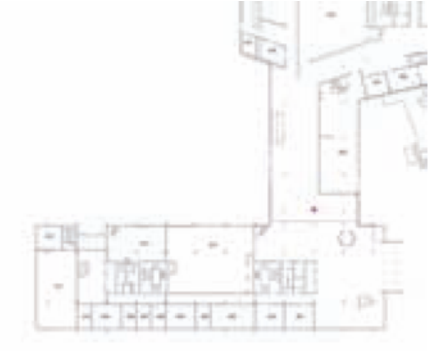


Figure 14: Lobby Plan

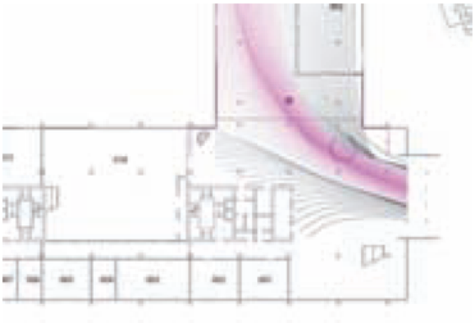


Figure 15: Movement Vectors

Figure 16: Fabrication Plan

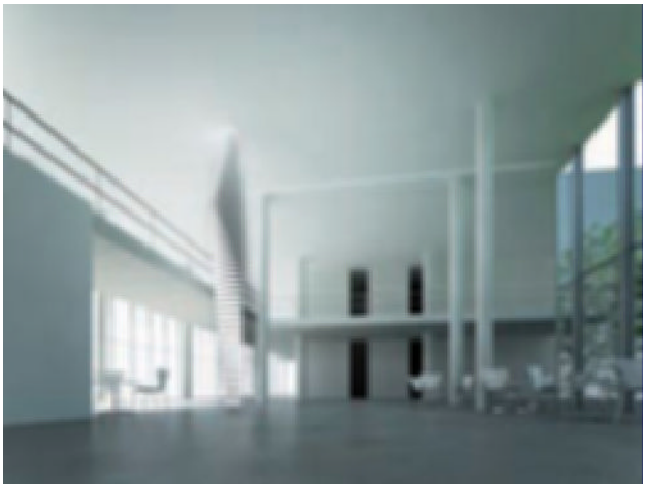


Figure 17: Rendering of the Light Column

# 4 The Generative Design Seminar

The Generative Design Seminar focuses on introduction of several digital modeling and design strategies. Case studies and the presentation of the aforementioned conceptual models gave the introduction to the class.

The dominating software in the seminar is Rhino Grasshopper, since Rhino is part of the school's curriculum. The focus after several on-screen tutorials showing three modeling concepts (mathematic form finding, global component logic and local component logic; figures 18-22) is the establishment of the three major models in the form of self-chosen design proposals by the students. The Roof structure for the Ruin Garden of the UdK Faculty of Architecture one of the proposed student works, which have been part of the seminar.

The second part of the seminar is focused on the development and refinement of the design projects in terms of structural performance and fabrication. Design decisions affect, or are being done, in relation to material-, structure- and/or fabrication constraints.

## 4.1 Mathematic Form-Finding

The onscreen tutorial shows the students a basic top-down approach. Based on a parametric sine-curve we developed three spans and connection iterations through variously sorted lists of the division/location points on the spans.

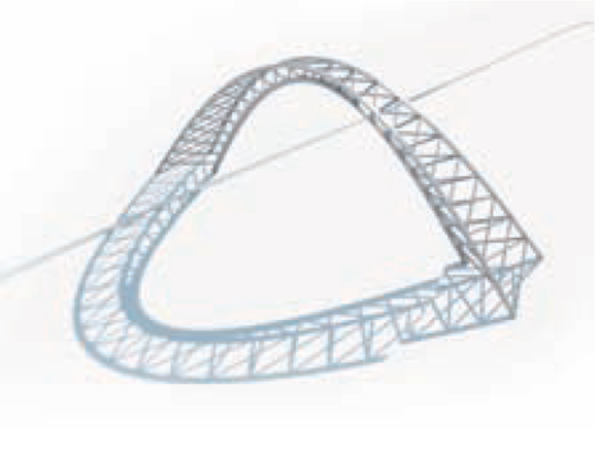


Figure 18: Sine Curve Bridge

## 4.2 Global Logics

The initial parameters are six curves, which define the overall shape of the bench profiles. The intent was to build the object out of ribs by defining the number of ribs and applying various features to generate the bench and the fabrication plan. All parameters and features can be edited at any level of the development to inform the design process and its criteria. The orientation of the ribs is driven by an external reference curve. The reference curve provides the position of a number of perpendicular planes, which serve as intersection references to create the outline of each rib. After the application of features, like fillet radii and section profiles, we modified the section profiles to have a gradually changing thickness towards the mid-point of the bench's length. This is also a clear top-down approach, since the overall design definition is driven by the initial parameters.

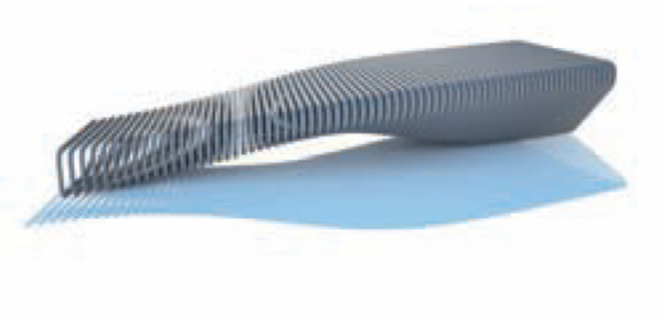


Figure 19: Bench Design

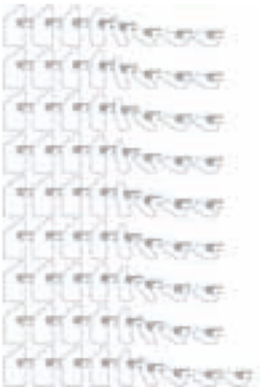


Figure 20: Bench, Top View Figure 21: Bench, Fabrication Plan

### 4.3 Component Logics

The tutorial covered the issue of a part-to-whole relationship of a component, its inherent logics and references to a whole assembly. We essentially developed a simple component based on one reference plane and distributed that particular component over a free-form surface using controlled local subdivisions as the application reference planes. The shown figure is a version where we linked the degree of the opening to the local position in terms of the Z-axis in the model space. The initial concept of this tutorial can be seen as a bottom-up approach. This continues even in the global positioning logic, which drives the degree of the openings, because the locally defined articulation of the opening is inherent in the component.

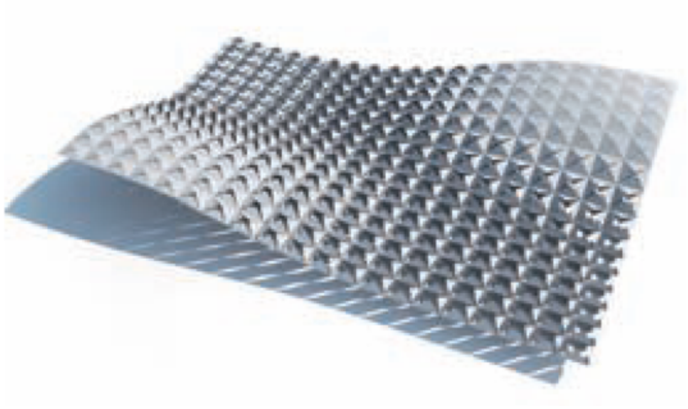


Figure 22: Component Logics

## 5 Conclusion

The work presented in this paper is to be seen as a start of an ongoing academic discourse at the University of the Arts. The expanding practice or discipline of architecture is demanding a broader conceptual understanding of the design process and design production and fabrication. This demand is subject to the development of a comprehensive teaching model which includes the understanding of intellectual models, as well as an introduction and utilization of the tools and softwares. A major concern is the immense expansion of the practice, its tools and possibly its expertise in terms of complexity requires moreover a specialization in fields within the practice, dividing the practice into various specialized fields. The concern is to provide a broad, but also deep coverage of an ever expanding and technologically advancing discipline into the school's curriculum. This means giving an introduction into the aspects of the expanded practice, by maintaining the recognition of architecture as a social practice and the "something" more than the pure science of planning and building, which make the phenomenological sensation of that what we call architecture.

## References

GENGNAGEL, C. (Ed): *Proceedings of the Design Modeling Symposium Berlin*, Universität der Künste Berlin 2009

EISENMAN, P.: *The Formal Basis of Modern Architecture*, Lars Mueller Publishers 2006

GLEITER, H.J.: *Architektur Theorie Heute,* Transcript 2008

HENSEL, M., MENGES, A.: *Morpho Ecologies,* AA Publications 2006

RIETHER, G., BEARLECKEN, D.: *Digital Girih, Proceedings of the Design Modeling Symposium Berlin*, Universität der Künste Berlin 2009

NEUMAYR, R., BUDIG, M.: *Associative Processes, Proceedings of the Design Modeling Symposium Berlin*, Universität der Künste Berlin 2009

*http://www.arch.udk-berlin.de/gengnagel/*

*http://www.ilijabentscheff.com/*

# Architectural Acoustics for Practitioners

**William Bergeron-Mirsky**

The Product Architecture Lab at Steven's Institute of Technology

**Jason Lim**

The Product Architecture Lab at Steven's Institute of Technology

**John Gulliford**

The Product Architecture Lab at Steven's Institute of Technology

**Aditi Patel**

The Product Architecture Lab at Steven's Institute of Technology

**Abstract.** *While much ink has been spilled over the benefits of intelligent models, information flow in the design process and the authoring of data sets there has been, in our opinion, little impact on the day-to-day practice of most architects. This paper discusses work by a team of graduate researchers to develop acoustic software tools that embody engineering fundamentals in support of the architectural design process.*

## 1 Introduction

We are interested in the design of software tools which, when embedded with engineering fundamentals can be integrated into the architect's existing design workflow at the correct stage and environment. These tools support the architectural practitioner in crafting design solutions while also gaining an appreciation of the engineering effects of his/her geometric choices. Ultimately, this interaction between practioner and software leads to a more deeply informed decision-making process. This paper details the development of such a tool, designed to be used in the domain of acoustics.

Acoustics is a fundamental design-performance criteria in architectural space and has been an area of extensive research [1]. However, it is our contention that this fundamental knowledge has not migrated fully into architectural practice such that acoustical performance is thought of as a critical quality of all designed space from the outset. In our opinion, this is due not to the complexity of acoustical science but the manner in which such knowledge is conventionally represented - in mathematical equations rather than geometric representation a form more intuitively understood by architects. We have created two acoustic software tools that are integrated into popular parametric design environments utilized by architects in order to address such a problem. The first tool is a plug in for Autodesk's Revit that calculates reverberation time in spaces; it is based on a statistical approach to acoustical analysis focusing on the ratio between volume (regardless of shape) and the room absorption area. Reverberation time is a global parameter of the space that does not vary from seat to seat [2]. The second tool is a custom raytracer and library of components for use in Grasshopper - the parametric interface of McNeel's Rhinoceros platform. Our raytracer is based upon a geometrical approach to acoustic analysis and is to be used for form-finding and shape tuning activities. This tool will be the subject of this paper as it highlights the close relationship between geometry and acoustics.

## 2 Current Practice

In architectural spaces where acoustics is an important performance criteria, geometry plays an important role in determining the eventual acoustic qualities. Additional measures of acoustical performance such as envelopment, loudness, intimacy and clarity are local parameters that vary per seat. They are heavily affected by the shape of the space [3]. When such spaces deviate from the conventional, whether it be in terms of complexity of form (concert halls) or specifics of program (experimental musical performances), the problem of developing geometric solutions that ensure high-performance becomes problematic. Lacking tools to perform even basic acoustic analysis on schematic designs, most architectural practices engage acoustic engineers for consultation and to troubleshoot design flaws. Acoustic engineers utilize specialized acoustic software packages such as CATT-Acoustic and ODEON Room Acoustics which are able to provide detailed simulations as well as being powerful enough to perform auralization. Software tools that could perform such analysis would be invaluable to an architectural practice at the design development stage when fundamental design choices have lasting and cascading effects on the rest of the design process.

Our raytracing tool is designed to reproduce the simulation capabilities of these specialized acoustics packages, but simplified so that features which may not be useful to architectural practitioners are removed. It is designed to be integrated into the existing design workflow and to represent the analysis results both geometrically and graphically. Equipped with a tool that has both rigorous methods and ease of use, an architectural practitioner gains the ability to develop a fundamental understanding of the relationship between geometry/form and acoustic performance. Thusly engaged, an architect can share a common language and the ability to generate standard metrics with which to engage an acoustician creating a more productive dialogue and superior outcomes. While this project does not seek to replace traditional acoustic consultation, the raytracing tool provides an alternate means of design exploration as well as a basis for communication, discussion and coordination of team-based efforts.

## 3 Design of Raytracing tool

The Raytracing tool is comprised of two, custom, VB.NET libraries. The first is built upon the Rhinoceros .NET SDK and encapsulates the logic that enables raytracing simulation to be performed on geometry within the Rhinoceros modeling environment. The second library extends the Grasshopper graphical editor plugin to provide customized components in that environment. We are integrating our software tool into McNeel's Rhinoceros/Grasshopper modeling package in order to leverage the robust geometric capabilities of Rhinoceros, the parametric capabilities of Grasshopper and the large installed base of architectural practicioners familiar with these tools.

Acoustics simulation using raytracing techniques was first outlined in a paper by Krokstad, Strǿm and Sǿrsdal in 1968[4]. Additional simulation methods including radiosity, image source method, beam tracing, hybrid method etc. have been developed since then in order to address the weaknesses of a purely raytracing approach [5]. Our tool is based on an implementation of the stochastic raytracing algorithm described by Michael Vorländer [6] . This type of geometric approach reduces the description of the sound field to a model of energetic rays [7] and has several implicit assumptions. Firstly, the room being tested is large relative to the wavelengths. Secondly, the source is a broadband signal [8]. Finally, our simulation assumes pure specular reflections and does not account for either diffusion/scattering or edge diffraction. Our object model for the raytracing simulator includes classes for stochastic emitters, surfaces with absorptive coefficients, energetic rays and detectors. The simulation cutoff is the order of reflections permitted as well as number of rays emitted. We believe, and are in the process of validating, that our simulator achieves accuracy sufficient for the purposes of providing feedback to an architect during design development.

## 4 Tool in Action

The general workflow for using the raytracing tool is: 1) develop a 3D model of the proposed designed space using Rhinoceros/Grasshopper platform, 2) link the custom raytracer components to the geometry of the model, 3) simulate and analyze the results and 4) iterate through design variations based on feedback from these results. The described workflow is self-contained within a single modeling environment and geared towards a design process based on iterative design and testing. Since external analysis software is not introduced into the workflow, this removes all the problematic issues of interoperability. Coupled with the near realtime simulation, made possible by a highly optimized algorithm (space partitioning), the raytracing tool delivers a seamless, fast and interactive experience that allows a designer to develop an intuitive understanding of the relationship between specific geometries and acoustic performance

As an illustration of our proposed workflow, we have applied these tools to an auditorium with a generic shoebox design. We've applied acoustically absorptive material to the seats (NRC – 0.7) and reflective (hard) materials to the walls and floors (NRC 0.1 -0.25). The auditorium will mainly be used for lectures and therefore its acoustic qualities should allow for clarity of speech.

### 4.1 Parametric Model

A model of the auditorium is built in the Rhinoceros environment (Figure 1). It is integrated with a Grasshopper definition (Figure 2) to enable quick variations of the geometry to be explored. The Grasshopper definition is built using custom components and is logically organized into three parts: i) Geometry inputs, ii) Simulation engine and iii) Visualization of data.

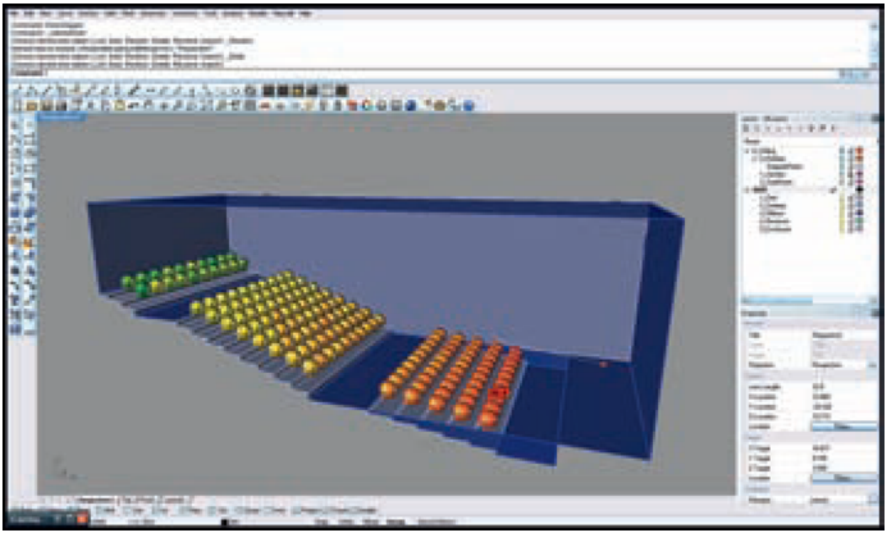


Figure 1 – Model of space in Rhinoceros

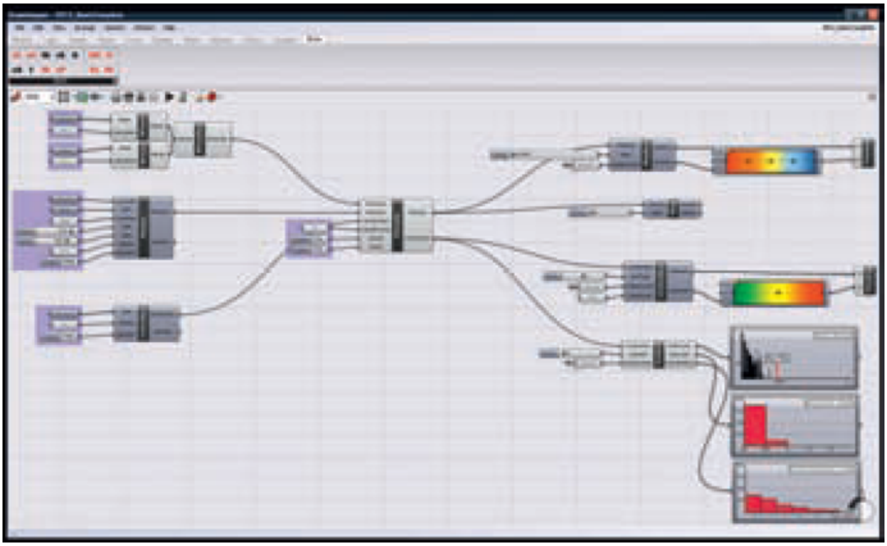


Figure 2 – Grasshopper definition built using custom components.

## 4.2 Analysis and Inspection

Collectively, seats can be evaluated based on different criteria (Figure 3) including: i) Energy received (loudness), ii) Early-Late ratio (definition/signal to noise) and iii) Direct-lateral ratio (envelopment/spatial impression). Each individual seat can also be inspected to understand its hit signature (Figure 4) – the energy of rays that it receives at specific instances in time. This allows for comparisons between seats. This geometric approach to acoustics enables the designer to differentiate the performances of seats/receivers in terms of a field rather than simply an average.

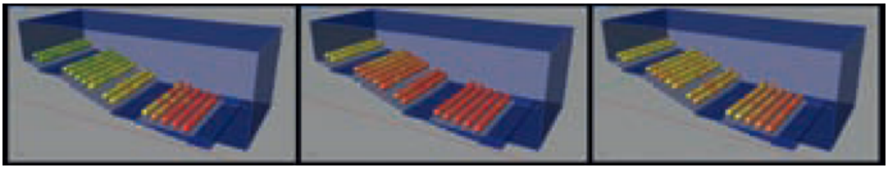


Figure 3 – Seats colored based on choice of evaluation methods.

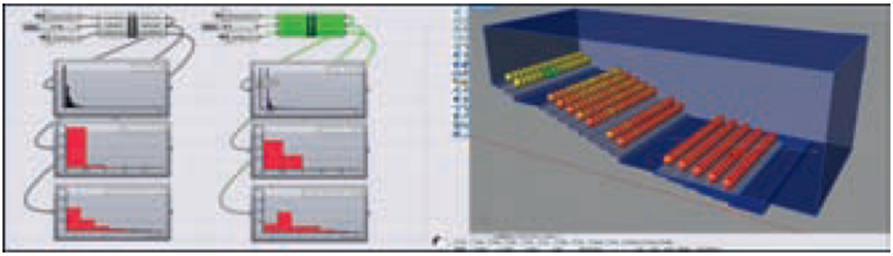


Figure 4 – Comparison of hit signatures between two seats

### 4.3 Visualization of Data

The data from the simulation can be visualized and interacted with in several ways. The rays can be visualized as particles colored per bounce (figure 5) in order to detect general patterns such as standing waves, echoes, focusing effects – this is a form of visual inspection. Individual rays can also be inspected (Figure 6) in order to analyze their paths and the materiality of the surfaces it comes into contact with. Such analysis is useful for dealing with specific rays e.g. problematic rays. Data is also visualized in terms of graphs (Figure 7) such as the energy histogram. These are standard metrics that allows architects and engineers to communicate using a common language. All data in the model can be further operated upon and manipulated using the standard grasshopper components. Visualizations can also be baked into geometry within rhino for documentation and comparison purposes.

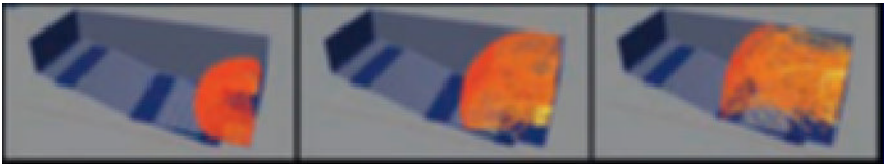


Figure 5 – Visualization of Rays as particles

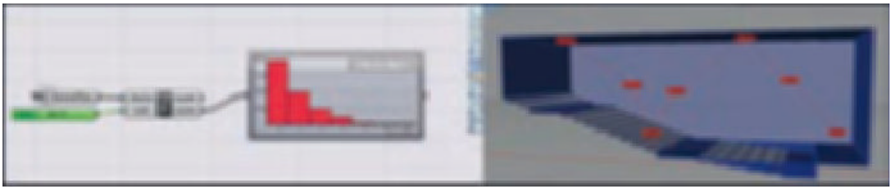


Figure 6 – Inspection of a single Ray

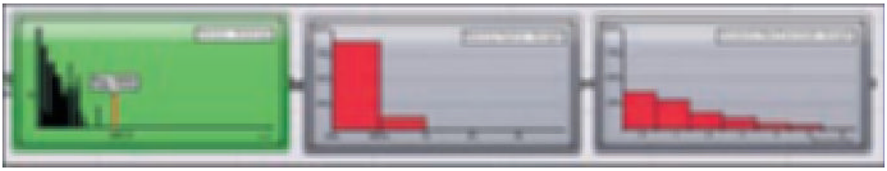


Figure 7 – Hit signature visualized as graphs

## 5 Conclusions

This work was undertaken as a case study in the embodiment of engineering fundamentals as software plug-ins rather than stand-alone tools. While addressing the specific field of acoustics, this model can be extended to other domains (structural, energy). Our conviction is that by providing tools directly to architectural practitioners at the appropriate moment (time and place) in the design process truly intelligent design will flourish.

## Acknowledgements

We would like to acknowledge the support and help of Professor John Nastasi (Product Architecture Laboratory), and Professor Marahelli Prasad and Rajavel Balaguru (Noise and Vibration Control Laboratory) at Stevens Institute of Technology.

## References

1. KROKSTAD, A. 2008. In: The Hundred Years Cycle in Room Acoustic Research and Design. Reflections on Sound, pp. 49 -76.
2. KROKSTAD, A. 2008. In: The Hundred Years Cycle in Room Acoustic Research and Design. Reflections on Sound, pp. 58.
3. BARTON, M. 1993. In: Auditorium Acoustics and Architectural Design. E & FN Spon, pp. 46.
4. KROKSTAD, A., STRØM, S., SØRSDAL, S. 1968. In: Calculating the acoustical room response by the use of a ray tracing technique. Journal of Sound & Vibration 8, pp. 118 -125.
5. SVENSSON, P. 2008. In: The Early History of Ray Tracing in Room Acoustics. Reflections on Sound, pp. 37-48.
6. VORLÄNDER, M. 2008. In: Auralization: Fundamentals of Acoustics Modelling, Simulation, Algorithms and Acoustic Virtual Reality. Springer-Verlag, pp. 175 -226.

7. Vorländer, M. 2008. In: Auralization: Fundamentals of Acoustics Modelling, Simulation, Algorithms and Acoustic Virtual Reality. Springer-Verlag, pp. 175.
8. Vorländer, M. 2008. In: Auralization: Fundamentals of Acoustics Modelling, Simulation, Algorithms and Acoustic Virtual Reality. Springer-Verlag, pp. 175.

# Wiggled Brick Bond

**Ralph Bärtschi, Michael Knauss, Tobias Bonwetsch, Fabio Gramazio, Matthias Kohler**

ETHZ

**Abstract.** *The wiggled brick bond is a generalized running bond which can be locally compressed. It was introduced to apply a bond onto two intersecting double curved bands. The wiggled bond is capable of shrinking and stretching with a constant gap between the bricks. Furthermore, the wiggled bond provides us with a generic crossing between two brick walls at an arbitrary angle. In this paper the mathematical techniques behind this bond are examined in detail.*

## 1 Introduction

Our research team has already done research with bricks assembled by an industrial robot [Bonwetsch et al. 2007]. The bonds described up to now were running bonds with the additional freedom of a rotation for each brick. This rotation had to be of little variation between neighboring bricks and lesser than 20 to 30 degrees relative to the bond direction. This limitation is due to the constructional demands of having a good interlocking between consecutive layers. In this paper we describe the wiggled brick bond which can handle bigger rotation while keeping good interlocking. The new bond provides us with a generic intersection between brick walls. Furthermore it can be compressed locally up to thirty percent while keeping constant gaps between the bricks. The new bond, while defined very easily, leads to some complex problems. How do we actually compress the bond? And how do we solve the intersections between the bricks on the same layer?

The architectural project we applied the new bond is described shortly in section 2. The wiggled bond is constructed in two subsequent steps. First the surface, the bond is applied to, defines the primary brick layers. Then the secondary layers are generated by the primary layers. In section 3 we describe this construction principle in greater detail. Section 4 formalizes all the aspects necessary to construct the primary layers. Section 5 describes the solution to the intersection problems in the secondary layers.

## 2 The Pike Loop project

Pike Loop is a 22m long brick installation built upon a pedestrian island in Manhattan. In changing rhythms, a loop lifts off the ground and intersects with itself at its peaks and valleys.

The continuous form and homogeneous expression of the structure can only be achieved through on site digital fabrication. The structure is built using the robotic fabrication unit R-O-B housed in a transportable freight container. In order to be able to build "Pike Loop" R-O-B was moved along the installation upon a truck trailer in 6 subsequent steps by 4.5 m.

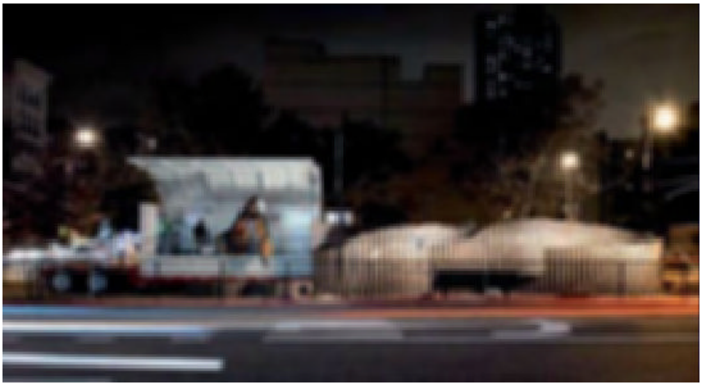


Figure 1: The onsite fabrication of a segment

## 3 The wiggled bond

The wiggled bond is based on the running bond. The running bond starts with a layer of evenly spaced bricks along a curve. The next layer is defined by the following construction procedure: Split two neighboring bricks into half bricks and connect their corresponding centers by a line segment. Take the middle of this line segment as the centre of a brick in the next layer and orientate it along the line segment.

We can apply the same construction procedure to more general starting condition. Starting with rotated bricks, we get bricks rotated in the opposite sense. (See Figure 8 left). This leads to a zigzag effect between the layers, from which the "wiggled bond" inherits its name. Depending on the rotation of the starting layer we can compress the bond (See Figure 5).

To apply this bond onto generic double curved wall surfaces we proceed as follows. We separate the layers into groups, the odd and even numbered. The odd numbered layers, called primary layers, are constructed according to the geometry. The even numbered layers, called secondary layers, are constructed according to the construction procedure. The bricks on the primary layer follow the generating surface. Their centers lie on curve segments and the gaps between the bricks are kept constant. How do we set the rotations of the bricks in the primary layers? The intersection between the curve segments defines the rotations at the boundary (Figure 2 right). The interpolation between the rotations at the boundaries is then parameterized and used to compress the brick course in order to match the length of the curve segment.

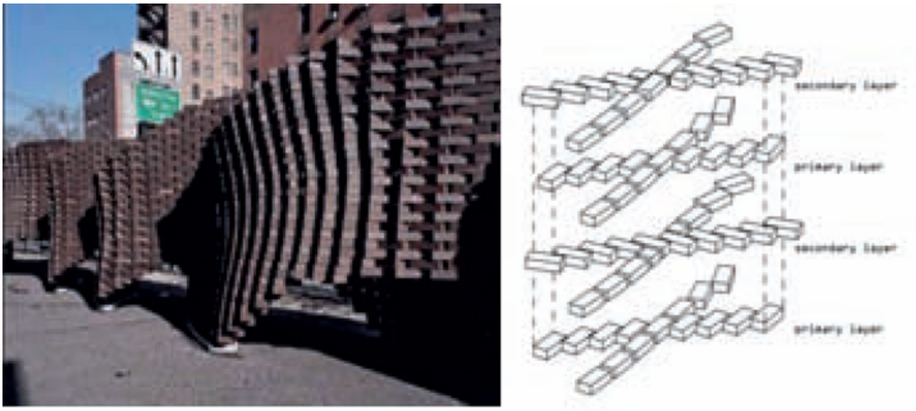


Figure 2: The final installation and a scheme of the intersection.

The straightforward construction procedure of the secondary layer, as described above, generally leads to intersections between the adjacent bricks. The solution of those intersections without compromising the visual appearance of the bond is described in detail in section 5.

We recapitulate the characteristics and the advantages of the wiggled bond. Like a harmonium, it has the ability to stretch and shrink with the same number of stones up to thirty percent. The wiggled bond enables us to build brick walls having surfaces with non constant section lengths and to define generic intersection between two walls. In the Pike Loop installation this compression capacity is used in a constructive, structural manner. The denser and thus more stable bond supports the lighter part at the crossing points, where the structure lifts off the ground.

## 4 The primary layer

The primary brick layers are defined by the geometry of the wall generating surface as well as by the boundary conditions at the crossings of the two wall segments. One end brick is parallel to its curve and the opposite end brick is orientated parallel to the crossing curve. The rotations of the bricks, in between those two ends, get orientations interpolated between those extreme rotations. The interpolation is parameterized to be able to resolve the boundary conditions while keeping a constant gap width between the bricks. The next subsection investigates the distances between touching bricks.

### 4.1 The distance map

We need to control the width of the gap between two rotated bricks. This can be simulated by two touching bricks which are enlarged by half the gap width. The bricks of the primary layer are then represented by touching rectangles (outlines of the bricks) along a curve. In this section we study the touching distance of two congruent orientated rectangles.

We can benefit from symmetry by assuming that one of the rectangles is centered at the origin and the other is laying on the positive x-Axis. To be able to control the touching of rectangles we need to investigate the following map:

Let $r(x, \alpha)$ be a rotated rectangle with centre at the coordinate (x, 0) along the positive x-Axis and rotated by $\alpha$. We define the map of the touching distance between two oriented rectangles with dimensions $w, l$ as

$$f_{dist}: S_1^2 \rightarrow [w, \sqrt{w^2 + l^2}]$$

$$f_{dist}(\alpha_1, \alpha_2) := \min\left(\left\{x \ \in [w, \sqrt{w^2 + l^2}] \,|\, r(0, \alpha_1) \cap r(x, \alpha_2) = \emptyset\right\}\right)$$

This definition could easily be extended to arbitrary 2D-polygonial shapes. Therefore we decided to implement the touching distance map numerically. We define the helping map $g_{dist}(x, \alpha_1, \alpha_2)$ to be the actual distance between the two rectangles if they are not touching and the square root of their intersection area if they are touching. We can now minimize $g_{dist}(x, \alpha_1, \alpha_2)$ for given $\alpha_1, \alpha_2$ by a naïve steepest gradient method [Pottmann et al. 2007]. For all numerical evaluations in this paper we used the rectangle dimension $w = 20, l = 10$.

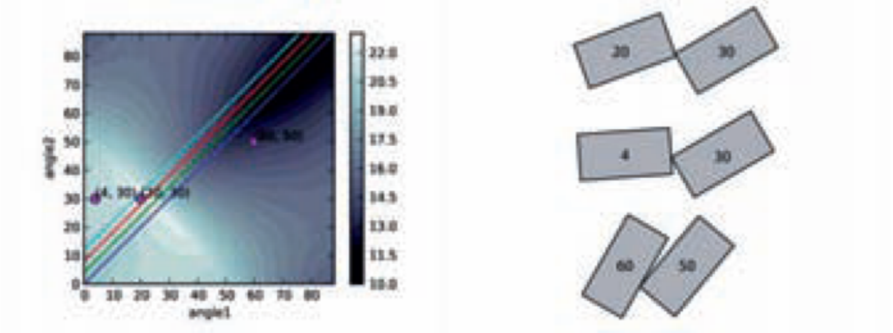


Figure 3: The touching distance map $f_{dist}(\alpha_1, \alpha_2)$. The dots correspond to the geometrical situations plotted on the right. The colored lines on the 2D-Plot are plotted in Figure 4.

As one can see from Figure 4 left, the section maps $f(\alpha) = f_{dist}(\delta + \alpha, \alpha)$ are not monoton. We wanted to increase the contrast between the compressed bond and the normal running bond. This is done by increasing the gap width for small angles. The distance map used in Pike Loop project is plotted in Figure 4 right. We call this modified map the truncated distance map.

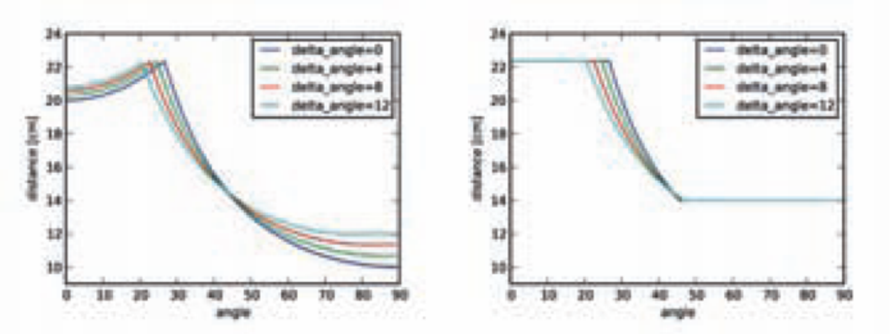


Figure 4: The distance maps with a constant relative angle between the bricks. On the right we plotted the truncated map which we used for the Pike Loop Installation.

## 4.2 The double exponent of interpolation

The primary layer is given by a curve, a number of bricks aligned on the curve according to a distance map, and the boundary conditions (the two relative angles $\beta_1$ and $\beta_2$ at the beginning and the end of the curve). To be able to solve the boundary condition, we introduced just one degree of freedom. The angle $\alpha_i$ of the $i$'th brick is given by the following interpolation formula

$$\alpha_i(\tau) := \big(1 - y(\tau, i)\big)\beta_1 + y(\tau, i)\beta_2,\ y(\tau, i) := x(i)^{1.3^{\tau}},$$

$$x(i) := \frac{\mathrm{i}}{(\mathrm{n}-1)},\ i \in \{0, \dots, n-1\}$$

The base 1.3 of the double exponent $\tau$ was chosen empirically in order to get a nearly linear behavior of the total length near an exponent of zero. This formula together with the distance map results in the following brick arrangements on straight lines with boundary conditions $\beta_1 = 0$ and $\beta_2 = 45$.

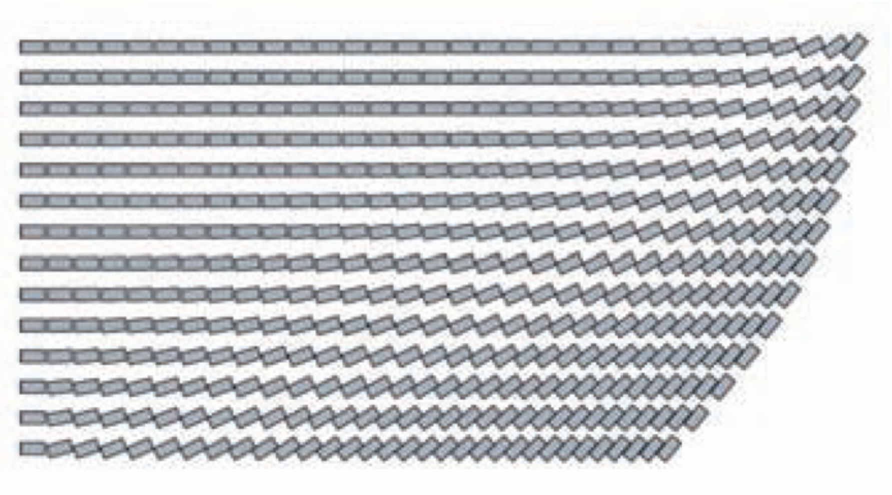


Figure 5: Interpolating the orientation of 32 bricks between 0 and 50 degrees. The double exponent $\tau$ increases from -4 up to 9.

Plotting the same situation as in Figure 5 for the interpolation exponent one observes quite a complicated microstructure. The green curve corresponds to the touching distance map while the blue curve corresponds to the monotone truncated distance map. For few bricks the microstructure is amplified.

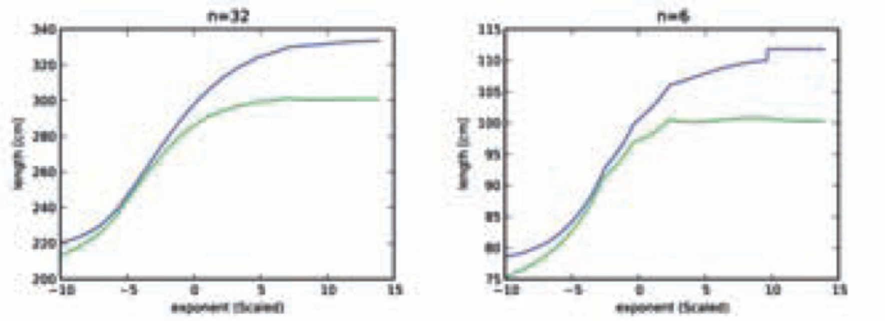


Figure 6: The length of some interpolated oriented bricks on a line depending on the interpolation exponent $\tau$. n represents the number of bricks. The two curves per plot correspond to the two distance maps plotted in Figure 4.

### 4.3 Solving the boundary condition for the primary layer on a curve

The solver for the boundary condition is again a naïve steepest descent algorithm with some empirically tuned step width modification. This local approach only works for the truncated distance map as can be seen from figure 6 where one can see clearly the local maximums of the un-truncated distance map.

In order to use the truncated distance map $\tilde{f}_{dist}$ we have to parameterize the brick's location and orientation. The orientation is given relative to the curve's direction. The position of the $i$'th brick is parameterized by $d_i$, the distance along the curve. Those distances $d_i$ are defined iterative and only depend on the curve, the boundary conditions, and the double exponent $\tau$. We are approximating the curve through straight line segments. If $d_i$ is bigger than the curve's length, we interpret the corresponding position as laying on a linear elongated curve.

$$d_i \cong \tilde{f}_{dist}(\alpha_{i-1}, \alpha_i + \delta_i) + d_{i-1}, d_0 = 0$$

$\delta_i$ is a correction term due to curvature of the curve. Let $\gamma(d)$ be the angle of the curve at distance $d$. Than $\delta_i$ is approximatively given by

$$\delta_i \cong \gamma(d_{i-1} + 20\cos(\alpha_i)) - \gamma(d_{i-1})$$

Let $L$ b the length of the curve. Solving the boundary problem is now equal to minimize a function in the double exponent:

$$\min_{\tau} g(\tau), \qquad g(\tau) \coloneqq |d_{n-1}(\tau) - L|$$

The solver algorithm starts with $e_0 = 0$ and than walks in the direction of the negative gradient of $g$ with a certain step size $s$.

$$\tau_i \coloneqq \tau_{i-1} + s\big(g(\tau_{i-1} + .05) - g(\tau_{i-1})\big)$$

If $g(\tau_i) > g(\tau_{i-1})$ we reduce the step size $s$ by a factor of 0.8 and restart the with a start value $\tau_{i-1}$.

The boundary angles are chosen such that we get a generic crossing bond for curves intersecting at angles $\gamma_1, \gamma_2$ we got the boundary condition $\beta_1 = 0, \beta_2 = \gamma_2$ for the first curve and $\beta_1 = \gamma_1, \beta_2 = 0$ for the other curve.

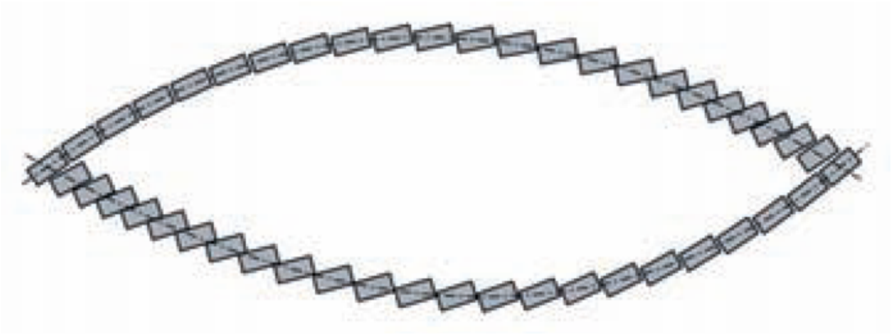


Figure 7: The primary layer for 24 bricks laying on different intersecting curves. The angles relative to the curves are truncated to a maximum of 45 degrees relative to the curves. This helps to prevent intersection problems of the secondary layer. If the segment gets too narrow, the bricks on different curves start intersecting. This can be resolved by gradient methods (see section 5.2).

# 5 The Secondary Layer

The secondary layers are the even numbered layers.  The secondary brick layer does not sample the actual wall geometry, but is derived from the upper and lower primary layer. First the bricks in the primary layer are divided in two half's. Then the conjunction of their centers defines the position of the brick on the secondary layer (figure 8 left). Unfortunately this simple construction logic does not guarantee for an intersection free brick layer (Figure 9).

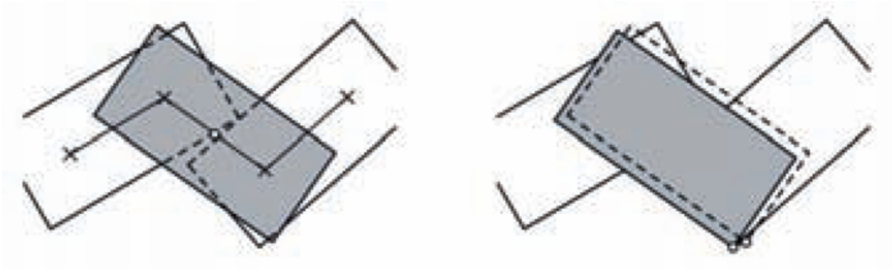


Figure 8: The construction of the secondary layer. Left: the original construction. Right: the cosmetic translation towards the outer corner.

### 5.1 The accentuation of the vertical lines

As the architectural design intends to accentuate the vertical lines in the final installation we had to align the visible corners of the bricks of the secondary layers with the nearest corners on the adjacent layers. We achieved this by translation of the secondary bricks the bricks on the secondary layers (figure 8 right). The translation to the outer corners fades in softly when the distance between the corners is smaller than 3cm.

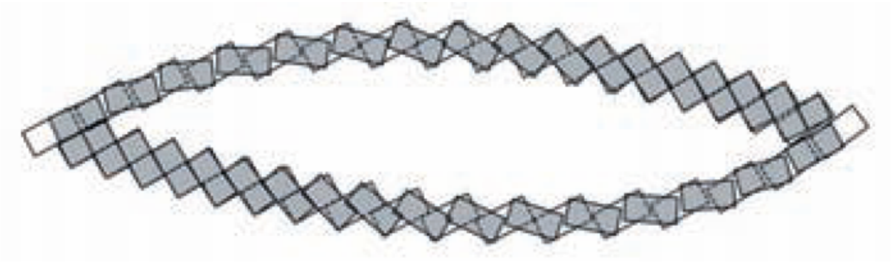


Figure 9: The secondary layer with the aligned outer corners. Remark the overlaps we already warned of. Alternating at the end or beginning we introduce a half-stone to get the crossing behavior of Figure 2 right.

### 5.2 Resolving intersection by rotation around the outer corners

The local search for the boundary problem has one degree of freedom. The intersections are resolved by a multi-dimensional local search. For the algorithm described here, we get one parameter per brick on a curve segment. We try to minimize the summed overlap area of the slightly oversized rectangles, by rotating the stone around the outer edge. The rotation around the outer corner is compatible with the correction for the accentuated vertical lines described in 5.2. To stay near the starting configuration we chose an upper bound for the maximum deviation of 10 degrees per brick. The algorithm is the same as described in 4.3 except that the gradient is a multi-dimensional vector. Since we separate this local search onto curve segments, it is necessary to freeze the first and the last rectangle. This steps does not resolve all the intersections between the bricks, but it reduces about 95% of the intersection area.

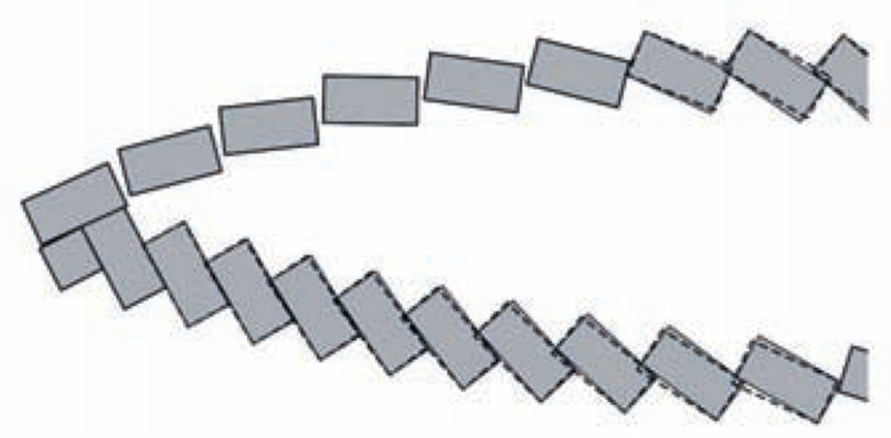


Figure 10: The dashed rectangles are the originals outlines of the secondary bricks. Nearly all the intersections are resolved by the rotational corrections. And the outer corners will be kept aligned between the layers.

### 5.3 Resolving the remaining intersection on both layers

We still have intersections not resolved by the rotational method of 5.2. The curve segments of the Pike Loop installation form lenses. If the lenses are too flat the bricks near the corners intersect. To solve this we translate the bricks and rotate them around their center. If the corrections needed are too big the results of the corrections are visually disturbing. Thus, it is important to start the design with surfaces which cross themselves in appropriate angles.

Again we use a steepest gradient method to minimize the summed overlap area. In order to reduce the dimension of the problem we resolve the intersection for each curve pair. Thus, we have to fix the corner bricks. For the other bricks we get three degrees of freedom per brick. There are two step sizes involved in the iteration step, one for the translational part of the gradient and one for the rotational part. If the overlaps at the start were too big this algorithm will produce visually noticeable deviations.

## 6 Conclusions

We presented a generalization of the common running bond. The wiggled bond has the ability to stretch and shrink with the same number of stones up to thirty percent. It has the potential to wiggle like a harmonium. This enables us to map it onto surfaces with non-constant section lengths. Additionally, the wiggled bond provides us with a generic crossing between two walls.

Further, we demonstrated that local search methods are important for design application. We used local search methods in four different settings for this project: calculating the touching distance map, solving the boundary problem for the primary layer, resolving the main intersections on the secondary layer by small rotation of the stones and finally resolving all the remaining intersections by translation and rotation. Local searches by steepest gradient are easy to implement. But they need some testing and tweaking for the step size and step size reduction, until they converge in a reasonable number of iterations. A previous formalization of the actual problem helps a lot in implementing them properly and generically.

## References

POTTMANN, H., ASPERL, A., HOFER, M., AND KILIAN, A. 2007. Architectural Geometry. Bentley Institute Press.

BONWETSCH, T., GRAMAZIO, F., KOHLER, M. 2007. Digitally Fabricating Non-Standardized Brick Walls. ManuBuild, conference proceedings, Rotterdam. D. M. Sharp, 2007, 191-196

# Geometric methods and computational mechanics for the design of stone domes based on Abeille's bond

**Maurizio Brocato**

University Paris Est

**Lucia Mondardini**

University of Bologna

**Abstract.** *We present an automatic procedure that, starting from fit design parameters, defines the geometry of stone vaults of a particular class and tests their mechanical performances. The considered vaults are spherical, but bonded adapting Abeille's 1699 design for flat vaults, thus promising esthetically and statically for new applications.*

*The procedure can help in the optimal design of these structures and provides information for their fabrication, solving all geometric issues, especially related to stereotomy. Examples are given of optimal choices of the design parameters.*

## 1 Introduction

At the end of the 17th century, flat vaults were invented by Joseph Abeille and Sébastien Truchet (patented in 1699, studied in [Frézier 1737 (ed. 1980)]), probably inspired by the proposals of timber frames appearing previously in Villard De Honnecourt, Leonardo da Vinci, and Sebastiano Serlio and in the contemporary work of John Wallis, and called later Serlio's floors (see [Emy 1837; Yeomans 1997]).

These vaults are structures that partake of the nature of reciprocal frames (or 'nexorades' [Baverel 2000; Baverel et al. 2000; Baverel and Nooshin 2007]) and of traditional vaults; stones are arranged in them in such a way that they can discharge loads in compression as usually, but a bending resistance larger than standard is obtained thanks to the particular interweaving of stones, mimicking that of timbers in Serlio's. As a consequence, an increase of the strength of the construction was expected and is actually observed in experiments [Fleury 2009], together with an enhancement of the vault's appearance.

These advantages notwithstanding, the system was rarely implemented in the past and a few ancient realizations exist in Spain [Rabasa-Dìaz 1998]. Recently, the possibility of building high performance stone structures has attracted attention on flat vaults anew [Fallacara 2006; Fallacara 2009; Uva 2003; Sakarovitch 2006]. The original Abeille's and Truchet's bonds, designed for a flat roof, were then adapted to model curved surfaces of architectural import [Etlin et al. 2008]. The question

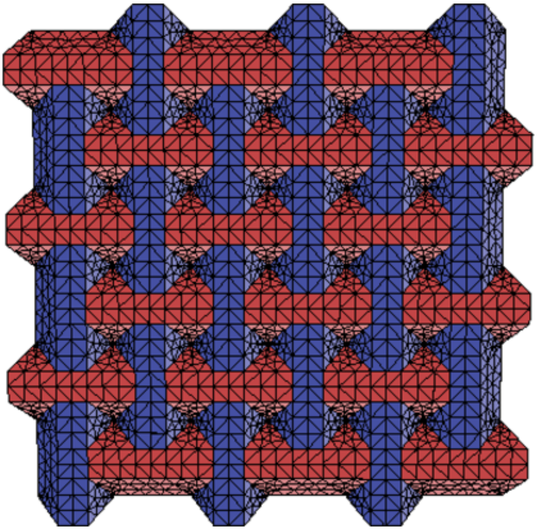


Figure 1: View of the intrados of an Abeille's flat vault. The bond is stable also if mounted upside-down.

arises then of the—perhaps optimal—design of the shape and arrangement of stones in relation with the shape of the vault.

Notice that Abeille's flat vault stability relies on thrust as in standard vaulted systems, but—contrary to what happens in these systems—here each stone withstand loads as a deep beam. The inclination of the stone's faces makes in fact contact forces act out of the plane of the vault, with a resultant orthogonal to that plane much more efficient, per unit of thrust, than standard. Actions in the stone are those typical of bending of deep beams, with a discharge arch interesting most of the stone's volume and a quasi-uniform high traction along the fiber opposite to the arch. The bending resistance of stones under such condition is sufficiently high for no records of tensile fracture in experiments on Abeille's vaults to exist.

In this paper we focus on the particular case of spherical domes and Abeille's bonding; we present an automatic procedure which, starting from a minimal number of geometrical parameters chosen by the designer, produces a rendering of the vault and all information needed for the cut and layout of stones and, via the necessary physical information on materials and loads, analyses mechanically the structure. The feedback obtained from this procedure is thus about the morphology, fabrication process and structural performance of a candidate design. Design optimization can then be sought comparing tentatives.

The procedure was implemented using Wolfram Mathematica version 5.0 for the geometrical computations and visualizations and Cast3M, 2009 release (a general purpose finite element computer code developed by the French Atomic Energy Commission CEA), for the mechanical computations. It can be divided into four parts, each being presented separately in the following sections of this paper:

- Definition of a geodesic dome
- Construction of an Abeille's bond
- Mechanical computations
- Stone cut and layout information

## 2 Definition of a geodesic dome

The first step for the design of an Abeille's dome is the definition of a set of points on the surface of a sphere that can be used as nodes of a mesh that covers that surface or vertices of a polyhedron approximating the sphere with given precision. The issue is similar to that called for the design of geodesic domes and leading to the definition of geodesic spheres [Kenner 1976].

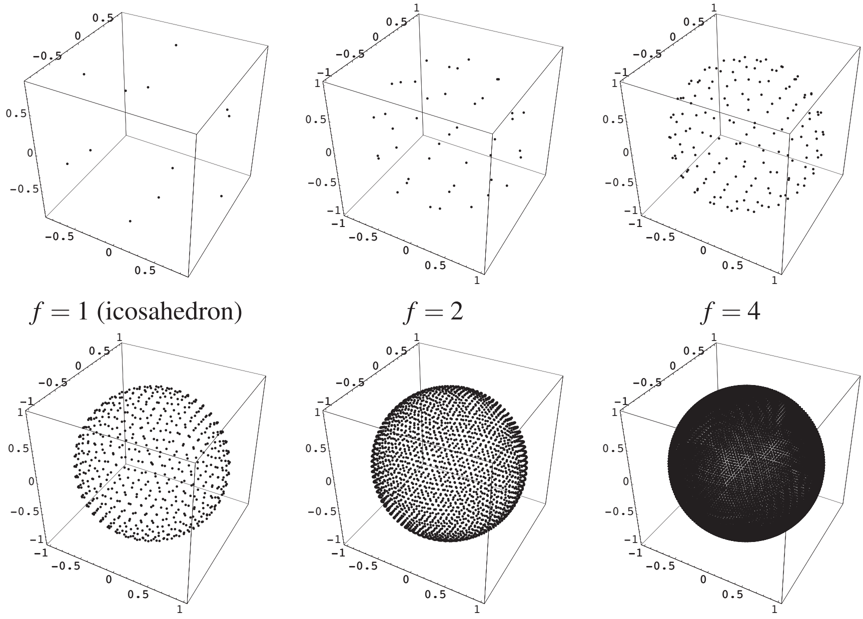
A geodesic sphere is a polyhedron whose vertices lay on the surface of a sphere and edges lay into planes that contain the centre of the sphere. It can be obtained starting from a regular polyhedron inscribed in the sphere: a spherical mesh can be obtained projecting the edges of the polyhedron from the centre of the sphere to its surface. This result can be refined—following several procedures—subdividing the faces of the polyhedron by triangulation to get denser meshes. The higher the refinement the larger the integer number called 'frequency' of the resulting geodesic sphere, counting the number of subdivisions applied to the reference polyhedron. Thus the $n$-th order 'geodesation' operation replaces each face of the polyhedron by the projection onto the sphere of the order-$n$ triangulation of that face.

The function 'Geodesate' exists in Mathematica 5.0 'Polyhedra' package that produces the geodesic sphere of wanted frequency starting from any of the polyhedra primitively known by the code or obtained from them through stellation or truncation.

We have used a slightly different algorithm: starting from an icosahedron, build its geodesate of frequency 2 (by projecting on the surface of the sphere the mid points of all edges of the icosahedron); this newly defined polyhedron replaces the icosahedron and the geodesation is applied on it, with frequency 2 again, generating a denser geodesic sphere; the procedure applies recursively as many times as needed.

This procedure leads to geodesic spheres more regular than when the tessellation is built at once before the projection: points are created, at each step, on faces defined in the previous step and closer to that of the sphere than the faces of the starting icosahedron, whence a smaller deformation due to the projection. As a drawback, only geodesic spheres with frequencies that are integer powers of 2 can be obtained.

In figure 2 we show the result of 5 iterations starting from the icosahedron (frequency $2^0 = 1$ to $2^5 = 32$). For the purposes of the present paper we will present all further results based on the $f = 4$ geodesic sphere, as in this case the ratio between the typical edge of the geodesic sphere and its radius is close what expected in some presently envisioned applications.



$f = 1$ (icosahedron) $f = 2$ $f = 4$

$f = 8$ $f = 16$ $f = 32$

Figure 2: Vertices of a geodesic sphere of frequency $f$.

Notice that at step $n$ we get the geodesic sphere of frequency $f = 2^n$ that has $V$ vertices, $F$ faces, and $E$ edges all depending on $f$:

$$V = 12 + (f-1)30 + (f-1)(f-2)10\,,\; F = 2V-4\,,\; E = 3V-6\,.$$

This geodesic sphere is given as the list of coordinates of the vertices $(x_i, y_i, z_i)$, $i = 1, \ldots V$; its faces are automatically generated and stored in a list of length $F$ of triples of vertex indices; its edges are similarly generated and stored in a list of length $E$ of couples of vertex indices, with the vertex of smaller index always first in the couple.

## 3 Construction of an Abeille's bond

The main issue in constructing an Abeille's bond is based on a procedure expedient for the design of nexorades. Given a bi-dimensional tessellation it is possible to define the beam's axes of the corresponding nexorade by rotating each side of the tessellation about its mid point of the same amount about an axis orthogonal to the plane (a procedure leading to the dual tessellation if the rotation is $\pi/2$). If the tessellation lays on a curved surface the rotation will be imposed in the plane tangent locally to the surface [Baverel 2000; Baverel et al. 2000; Baverel and Nooshin 2007].

Following this idea, for all edges of the geodesic sphere, the mid point is identified and the normal to the sphere through this point is taken; than a rotation of an angle $\alpha$ (an amplitude defined by the designer between 0 and $\pi/2$, ends excluded) about this normal is applied to the edge to obtain the axis of what we will call in the following part of this paper a 'nexor' (usually the basic element of structures called nexorades or reciprocal frames).

After such operation a list of $E$ nexor's axes is obtained in the form of couples of triples of coordinates. Actually three local coordinates axes are given per nexor, taking the nexor's axis $t_i$, the normal to the sphere in the mid point of the nexor $n_i$, and their vector product taken in this order $m_i$. Furthermore, a topological information must be recovered, to decide which nexors cross each other, a task that is accomplished checking the distances between the end points of the segments representing the nexor's axes. Figure 3 shows the result of the step for an emisphere.

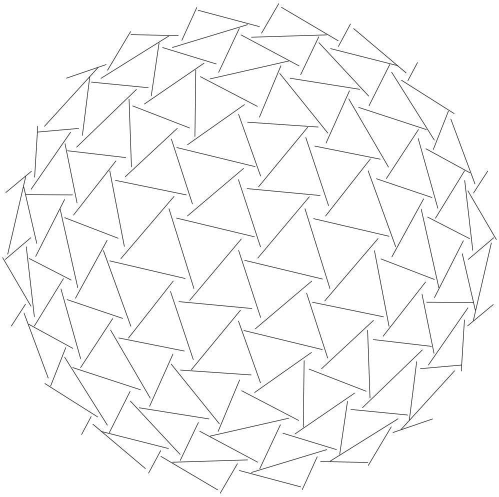


Figure 3: Nexorade sketch obtained from half a geodesic sphere rotating nexors of an angle $\alpha = \pi/10$ about the normal to the surface of the sphere passing by their mid point. Only segments aligned along the nexor's axes are represented.

The local coordinates are useful to define the shape of the stones that are then placed as nexors. All stones are obtained from right prisms having the same trapezoidal base and the height aligned with the nexor direction $t_i$; their bases are then cut away at an appropriate angle to get the proper contact conditions with the neighboring stones.

The base trapezoid, an isosceles one, is defined through the following parameters: $b$, the height measured along $n_i$, $a_0$, the length of the mid-segment measured along $m_i$, and $\phi$, the angle of the inclined sides on the normal direction $n_i$; the two bases are then

$$a_1 = a_0 - b \tan\phi \, , \; a_2 = a_0 + b \tan\phi \, .$$

It is assumed that the axis $t_i$ passes through the mid point of the mid segment of the trapezoid. For the examples that will follow we have taken $b = 0.0857r$, and $a_0 = 0.07875r$, where $r = 10$ m is the radius of the sphere, while variations of $\phi$ where explored. All these parameters can anyway be chosen by the designer to enhance the performance and esthetics of the construction.

Knowledge of the interweaving of nexors can then be used to compute intersections between the parallel edges of a prism and the surfaces identified by the sides of the neighboring prisms (due to the curvature of the approached surface, edges do not necessarily reach the faces of the neighboring prism). Furthermore the ideal print of stones on adjacent stones is computed for fabrication information.

Figure 4 shows a resulting Abeille's dome view from the extrados and the intrados; figure 5 shows a set of possible bonds obtained by changing the design parameter $\alpha$; figure 6 and 7 show the geometries that were explored by mechanical analysis, obtained changing both design parameters $\alpha$ and $\phi$, the second being positive in the set shown in figure 6 and negative in figure 7. Notice that this change of sign corresponds, mechanically, to the passage from end-supported stones bearing their neighbors between supports to center-supported stones bearing their neighbors as cantilevers and, from the esthetic point of view, from a smooth to a carved intrados.

## 4 Mechanical computations

Ad hoc interfaces where programmed to export geometries from Mathematica to Cast3M as a basis for a finite element model of the structure. The vault can then be modeled as a discontinuous set of stones in contact with each others.

Each stone is meshed into finite elements for the computation and considered as a linear elastic solid block (different mechanical behaviors being applicable if needed). Interfaces are treated as follows. As only the geometry of stones is constructed and exported from Mathematica, without topological information on the bond, Cast3M has to detect, after meshing the stones, all occurring contacts between them and adapt the mesh to match across such interfaces. These tasks are performed by routines included in the standard release of the code [Pegon et al. 2001]. Interfaces are thus modeled as joints with Mohr-Coulomb dilatant behavior: compression forces and tangential forces limited by friction can be exchanged across them; stones can separate and/or glide on each other under the action of contact forces.

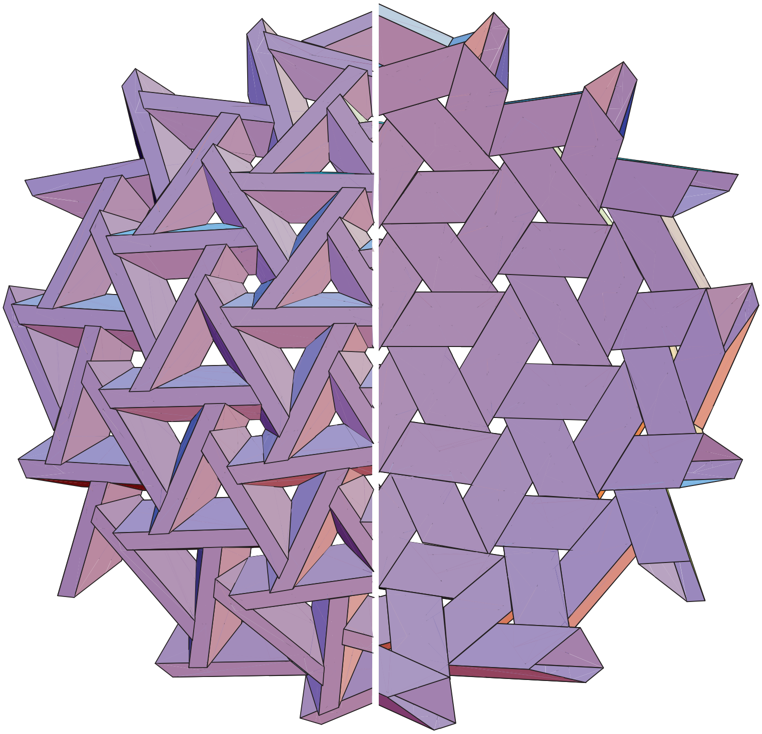
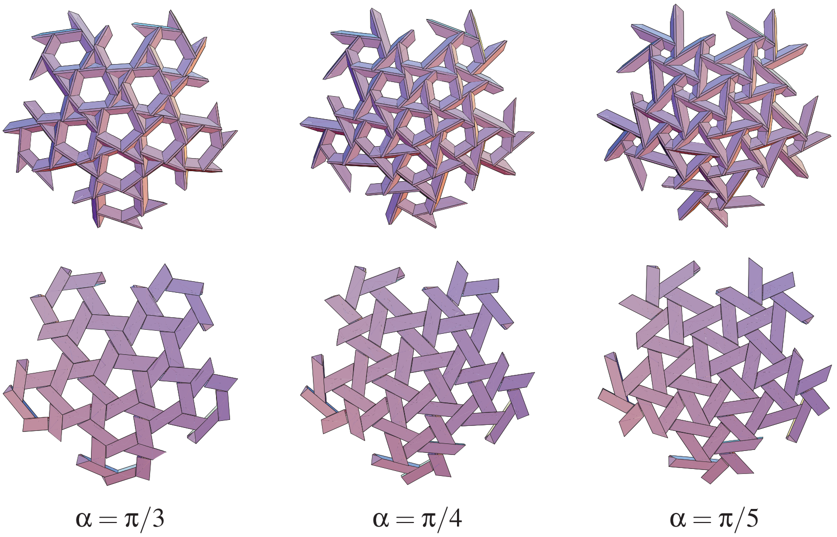


Figure 4: Extrados (left) and intrados (right) of an Abeille’s dome.

To complete the information needed for a mechanical analysis, boundary and load conditions and all material parameters must then be set. The boundary conditions are set starting from the knowledge of the boundary of the vault, as given by the end faces of the stones of the outer ring, which is exported from Mathematica to Cast3M. These faces are then considered in calculations resting on unilateral Coulomb contacts on a rigid confinement ring.

The remaining mechanical information (loads and materials) must be defined by the operator directly on Cast3M and can be adapted to any circumstances. Loads in particular can be assigned as functions of time, to perform non linear computations and detect the possible occurrence of consolidation. For the purposes of the present paper we have considered a self-weight loading oscillating between its natural value and 1.5 times this value on 3 charge-discharge cycles. The material parameters are listed in the following table.



$\alpha = \pi/3$ $\alpha = \pi/4$ $\alpha = \pi/5$

Figure 5: View of the extrados (above) and of the intrados (below) of spherical Abeille's bonds obtained for different values of $\alpha$.

| | | | |
|---|---|---|---|
| Stones | density | 2000 | kg/m$^3$ |
| | Young modulus | 23.8 | GPa |
| | Poisson coefficient | 0.27 | |
| Joints | cohesion | 0.00 | |
| | friction angle | $\pi/4$ | |

Analyses where carried out to evaluate the optimal bond under two mechanical criteria: (i) minimal deflection under the given load after the quoted series of loading-unloading cycles consolidating the structure; (ii) minimal thrust at the springing with respect to the overall weight borne by the vault. A typical output is shown in figure 8 for two different vaults charged cyclically and in figure 9 for the vertical displacement in one case.

The maximum deflection under self-weight is 4 cm, i.e. 1/400 of the span. Stresses are relatively uniform and low (less than 2 MPa in von Mises terms), with the irrelevant exception of a few contact points at the boundary, where the rigid constraint hardly represents the effect of a real confinement structure.

That particular investigation led to the preference of vaults with $\alpha = 23\pi/120$ and $\phi = 2\pi/15$.

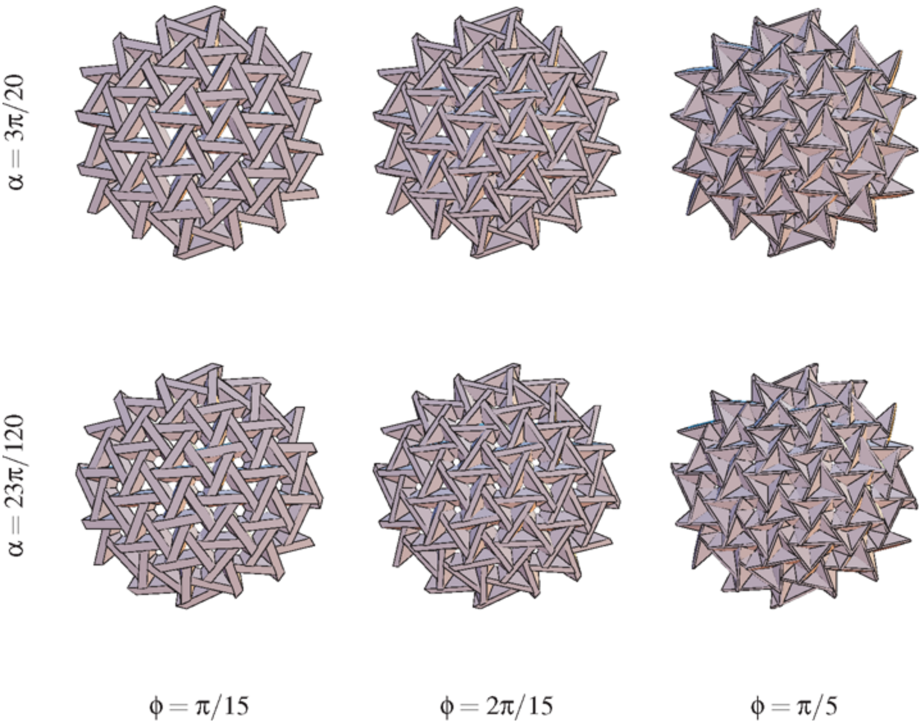


Figure 6: View of the extrados of spherical Abeille’s bonds obtained for different values of $\alpha$ and (positive) $\phi$.

## 5 Fabrication information output

For computer aided fabrications Mathematica results can be exported in any necessary format; nevertheless we have focussed particularly on traditional fabrication output, useful in experiments. Furthermore a classification of stones according to their geometry is automatically performed for standardization purposes and a layout is produced to locate individual pieces in the vault.

Stones are first classed into families of equal shape and dimensions, then the net of each family polyhedron is produced. This net reproduces also the print of neighboring stones, to help assembling. Figure 10 shows a typical output. In the particular cases that where studied, and thanks to the good regularity of the tessellation of the sphere obtained by geodesation, only 8 families are needed to class all 240 stones that compose an hemispherical vault.

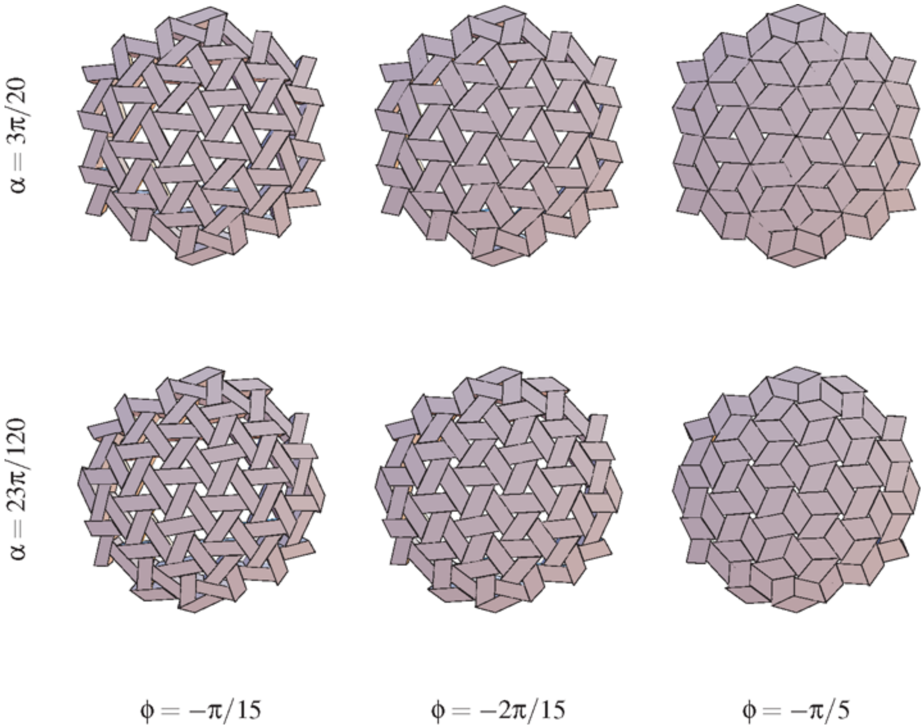


Figure 7: View of the extrados of spherical Abeille's bonds obtained for different values of $\alpha$ and (negative) $\phi$.

## 6 Conclusion

The procedure described in this paper makes a new family of structures accessible to the designer, solving some geometrical issues related to the accommodation of a flat geometry on the surface of a sphere and giving tools for the evaluation of the mechanical performances and the fabrication tasks. Extensions of the algorithms to surfaces geometrically more complex then the sphere can be envisioned.

The type of structures that can be designed, relying only on compressive strength and unilateral contact conditions, can prove advantageous in the next future, especially if a reduction of embodied energy is sought.

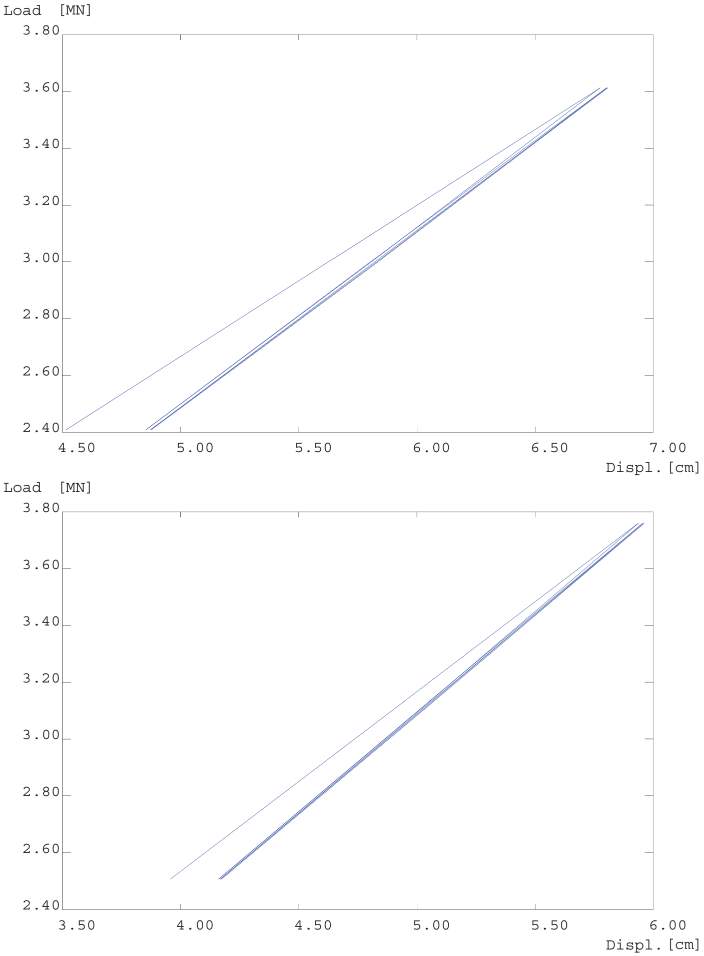


Figure 8: Load vs deflection curves under cyclic load for a $\alpha = \pi/6$, $\phi = 4\pi/45$ vault above and $\alpha = 11\pi/60$, $\phi = 2\pi/15$ below. The latter shows a better response, with smaller maximal deflection and consolidation. Deflections are measured from the unloaded configuration, the minimal represented load is the weight of the vault.

## References

Baverel, O., and Nooshin, H. 2007. Nexorades based on regular polyhedra. *Nexus Network Journal 9*, 281–298.

Baverel, O., Nooshin, H., Kuroiwa, Y., and Parke, G. A. R. 2000. Nexorades. *International Journal of Space Structures 15*, 155–159.

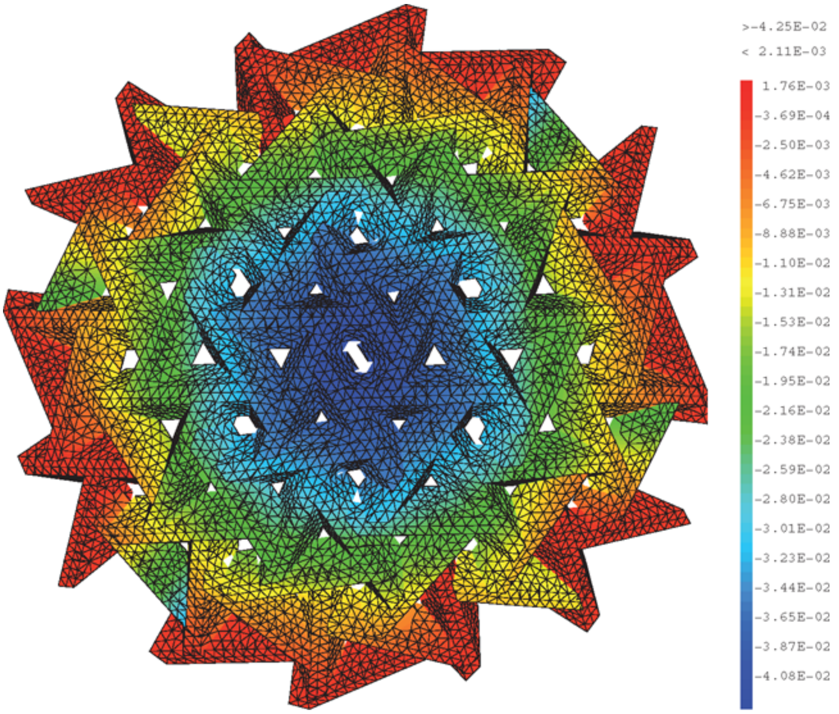


Figure 9: The vault's vertical displacement under self-weight load for $\alpha = 11\pi/60$ and $\phi = 2\pi/15$. Measures in [m].

BAVEREL, O. 2000. *Nexorade: A family of interwoven space structure*. PhD thesis, University of Surrey.

EMY, A. R. 1837. *Traité de l'art de la charpenterie*, vol. 1. Anselin and Cabilian-Gœury, Paris.

ETLIN, R., FALLACARA, G., AND TAMBORERO, L. 2008. *Plaited Stereotomy – Stone Vaults for the Modern World.* Aracne Editrice, Roma.

FALLACARA, G. 2006. Digital stereotomy and topological transformations : Reasoning about shape building. In *Proc. Second Int. Congress Construction History*, M. Dunkeld, Ed., Queen's College Cambridge, 1075–1092.

FALLACARA, G. 2009. Toward a stereotomic design: Experimental constructions and didactic experiences. In *Proc. Third Int. Congress on Construction History*, S. Huerta, Ed., BTU Cottbus, 553–559.

FLEURY, F. 2009. Evaluation of the perpendicular flat vault inventor's intuitions through large scale instrumented testing. In *Proc. Third Int. Congress on Construction History*, S. Huerta, Ed., BTU Cottbus, 611–618.

FRÉZIER, A.-F. 1737 (ed. 1980). *La théorie et la pratique de la coupe des pierres et des bois pour la construction des voutes et autres parties des bâtiments civils et militaires*. Jacques Laget L.A.M.E., Nogent-le-Roy.

KENNER, H. 1976. *Geodesic math and how to use it*. University of California Press, Berkeley.

PEGON, P., PINTO, A. V., AND GÉRADIN, M. 2001. Numerical modelling of stone-block monumental structures. *Computers & Structures 79*, 2165–2181.

RABASA-DÌAZ, E. 1998. La bóveda plana de abeille en lugo. In *Actas del Segundo Congreso Nacional de Historia de la Construcción*, F. Bores, Ed., Instituto Juan de Herrera, Universidad de A Coruña, 409–415.

SAKAROVITCH, J. 2006. Construction history and experimentation. In *Proc. Second Int. Congress Construction History*, M. Dunkeld, Ed., Queen's College Cambridge, 2777–2791.

UVA, G. R. 2003. Learning from traditional vaulted systems for the contemporary design. an updated reuse of flat vaults : Analysis of structural performance and recent safety requirements. In *Proc. First Int. Congress Construction History*, S. Huerta, Ed., Instituto Juan de Herrera, 2015–2021.

YEOMANS, D. 1997. The Serlio floor and its derivations. *Architectural Research Quarterly 2*, 74–83.

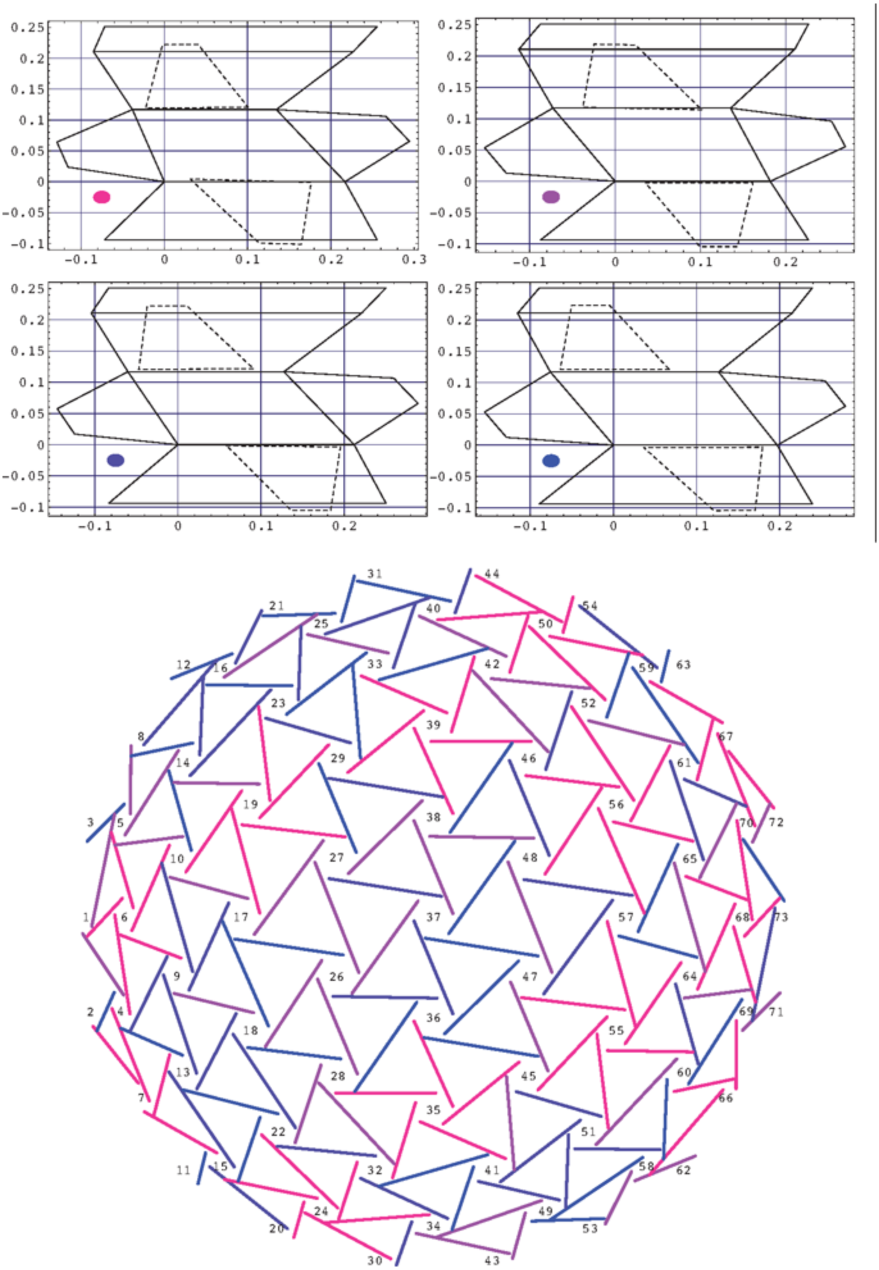


Figure 10: Output of the net of four of the eight families of stones that compose the vault (above) and layout of the structure representing stones as nexors (below). Nets are labelled with a colored spot that corresponds to the color of nexors in the layout. Dashed lines in the net figures represent the contour of the end-face of the stones that rest on sides of the portrayed one; the correct position of any stone is thus determined making its end-faces match such prints on the neighboring ones.

# Louvre Abu Dhabi 1/33 – Fabrication of a large-scale physical light-test model

**Benjamin S. Koren**

1:One | Computational Geometry

**Abstract.** *Physical scale models have not only been indispensible to architects as working tools for spatial design, but also for engineers as instruments for the accurate testing and verification of a variety of physical characteristics, such as light and sound. The following paper aims to document the development and construction of a large-scale building model used for scientific testing purposes: a light-test model of the Louvre Abu Dhabi, designed by the architect Jean Nouvel, at scale 1/33. As in its full-scale counterpart, building this model to detailed specifications, coupled with an inherent level of geometric complexity, large-dimensions and high-levels of precision required the development and use of a set of advanced methods in computation and fully-integrated CAM techniques: from the parametric generation of non-standard components, geometric optimization algorithms for efficient manufacturing of parts, to automated tasks of production.*

## 1 Introduction

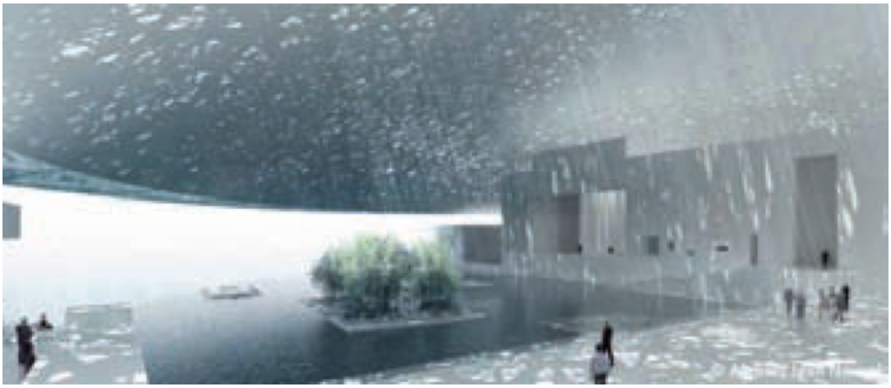


Figure 1: Design of the Louvre Abu Dhabi (© Ateliers Jean Nouvel)

The Louvre Abu Dhabi, to be completed in 2013, will be part of the world's largest concentration of cultural institutions within the Saadiyat Island Cultural District in Abu Dhabi. It will be the first universal museum in the Arab world and will showcase fine arts, decorative arts and archeological artifacts featuring the artistic achievements of different cultures that will be collected from all over the world. It is being designed by Jean Nouvel, who has a subtle and ingenious vision for the project: a gigantic dome, 180 meters in diameter, perforated by layers of superimposed cladding patterns that will filter the blazing desert sun onto the museum gallery buildings below, not only controlling the microclimate on the plaza beneath, but creating a dynamic, virtuoso light-effect, which the architect calls his “rain of light”. This aspect of the project was considered to be of such high importance that it had been decided, during the course of design, to have a large-scale model of the project constructed, to test and verify the lighting conditions.

In order to conduct scientific experiments and record accurate measurements, the specifications for such a model far surpassed those required in most regular architectural model-making: First, the model had to be constructed at a sufficiently large scale, at 1/33 measuring 5.5m in diameter. Secondly, a large quantity of elements of the dome directly affecting the lighting conditions had to be built to the highest level of precision, namely nearly 15,000 individual components comprising the domes structure and approximately 1,000 tiles of the light-filtering cladding layers mounted on top and bellow (fig. 2). Lastly, the experiments had to be conducted under the real light conditions in Abu Dhabi and thus withstand significant temperature differences, severely limiting the choice and mix of materials to be used, and consequently, the means of its production.

## 2 STRUCTURE

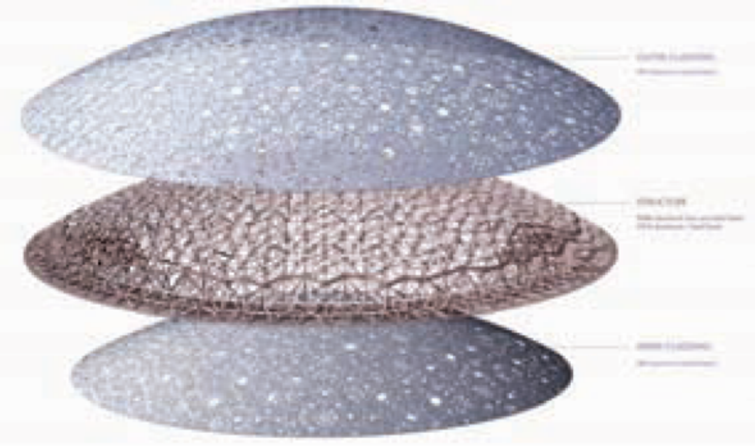


Figure 2: Schematic diagram of the model construction: Structure (middle), cladding layers (top and bottom).

The geometry of the structure was fixed, had been provided for by the architects and engineers, and thus exhibited a high level of sophistication. It was, however, lacking any details necessary to build it at the required scale. The strategy was such as to break down the geometry at declining scales, taking advantage of its geometric characteristics accordingly, from (1) the whole dome, (2) its building parts, (3) the individual modules down to (4) the individual bars and knots.

## 2.1 Whole Dome

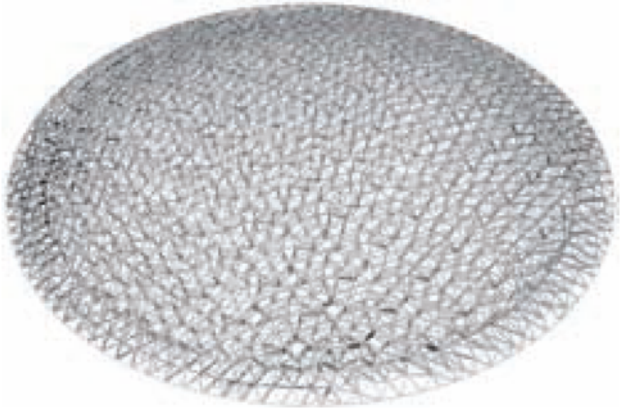


Figure 3: The structure of the dome design

The structure of the whole dome (fig. 3) consists of a space frame, the top and lower chords lying on perfect spherical segments. While the pattern of the elements could be defined using a square tiling of the Euclidean plane, and thus would be repetitive, once mapped, or rather projected, onto the surface of a sphere, the geometry of the structure distorts towards the edge of the dome, thus resulting in unique members throughout. As with any dome, the edge of the model had to be strengthened so as to resist the outward thrust of the dome. Finally, it rests on only four points, creating the additional problem of having to counter the sagging of the cantilevering edge. The model at scale 1/33 was of a sufficient size and mass as to require the analysis of its structural behavior (fig. 4) using the proposed materials, aluminum and stainless steel.

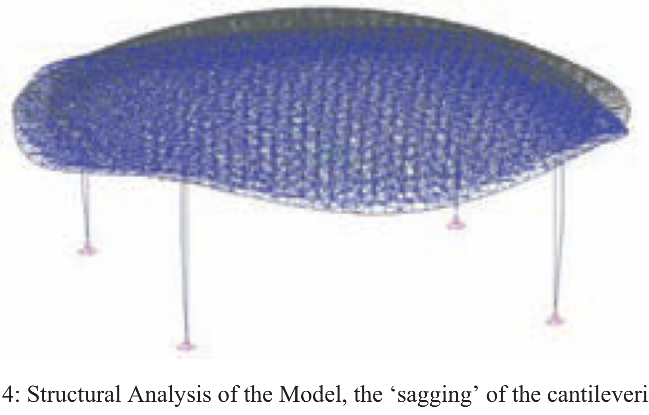


Figure 4: Structural Analysis of the Model, the ‘sagging’ of the cantilevering edge

Due to the outward thrust, and the sagging of the cantilever, it was therefore decided to pursue two detailing methods in the structure of the model: the interior part would consist of separate elements (knots and bars) mechanically pinned together, while the edge would be welded out of steel tubes, forming a tension ring along its perimeter. Basic structural analysis verified the structural behavior of this strategy.

## 2.2 Building Parts

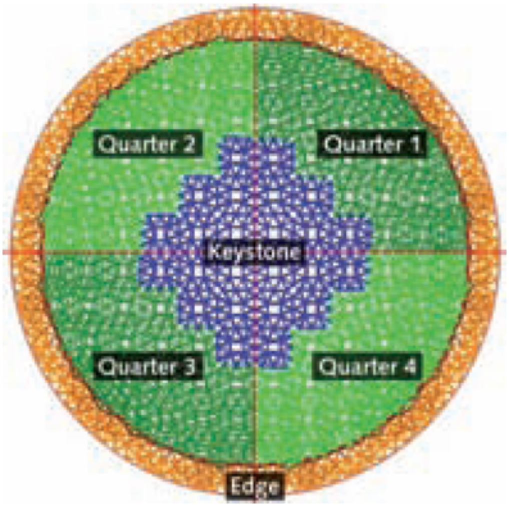


Figure 5: Building Parts: 4 quarters (green), a keystone (blue), including the welded tension-ring along the edge (red).

Resting on four supports, the building engineers working with Nouvel have rationalized the structure to the point where it is point-symmetrical, each quarter being identical (mirror and rotational symmetry), each quarter in turn exhibiting bilateral symmetry. Having had to pre-assemble the model in Germany, shipping it in parts to Abu Dhabi via air-freight to have it re-assembled on site, prompted the models division into 5 building parts: four quarters plus a segment around the zenith, the ‘keystone’ (fig. 5), due to air-freight size restrictions. The data for one quarter then served as the basis for developing further strategies for generating and producing the structure.

## 2.3 Modules

For reasons of assembly logistics, the building part was further divided into smaller modules to be preassembled in parallel and subsequently joined into a larger whole (fig. 6). Naming of the modules also served as a basis for indexing and labeling of parts. The structure for an entire quarter had to be computer generated in one step to ensure the highest level of precision.

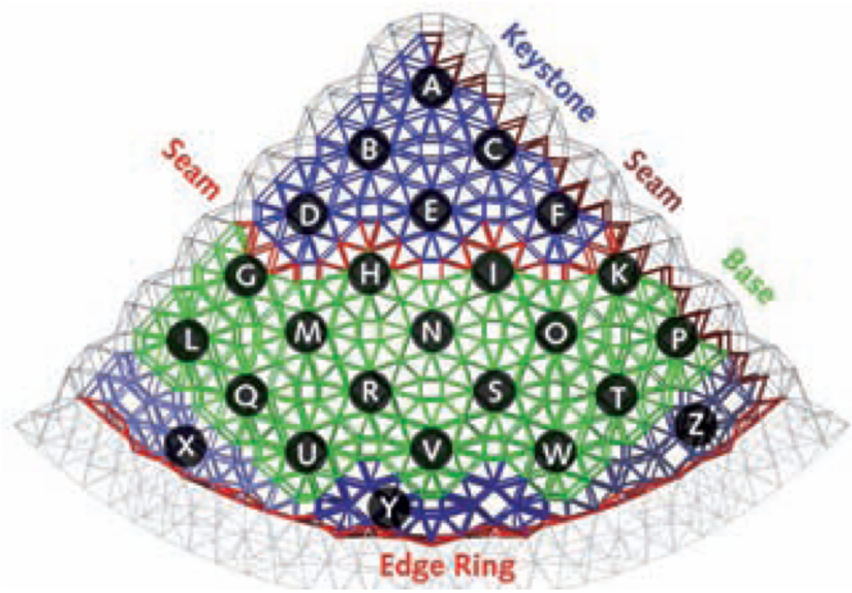


Figure 6: Modules

## 2.4 Bars and Knots

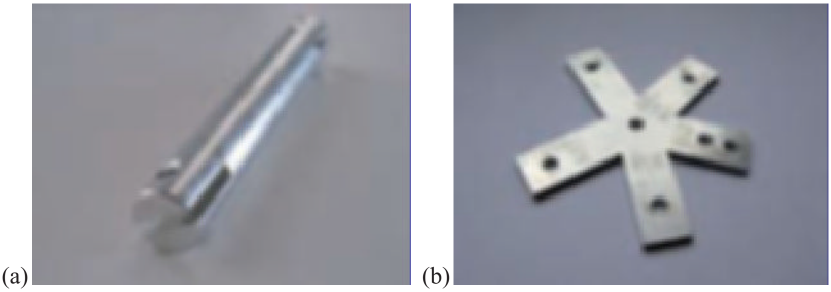


Figure 7: Bar and Knot

The main detail that served as the basis for the generation of the structure, and its subsequent optimization, was a combination of bars and knots. An initial material concept was followed using custom laser-sintered knots out of Alumide, an aluminum filled polyamide, a material chosen for its high stiffness, favorable thermal properties and metallic appearance. As sintered parts are mainly used for prototyping, however, some of Alumide's mechanical properties have not been published, most crucially the shear modulus was unavailable (EOS). A simple mockup confirmed the concern for the materials inadequate capacity to withstand even relatively weak shear forces empirically. The initial concept had to be dismissed for an all-metal alternative: non-standard stainless-steel knots, laser-cut out of flat metal sheets, and lathe-manufactured solid aluminum bars (fig. 7).

The entire structure first had to be generated in place in 3-D. For the purpose of generating the structure, a Plug-In was developed for Rhinoceros 3-D (fig. 8). A centre-line model of the dome was taken as an input to create an associative data-structure, both bars and knots as simple components were parametrically defined, and included such properties as thickness, length, arm-length, number of pins, ID and the IDs of connecting elements.  Each element was then custom-generated in place for an entire quarter. Due to the effects of distortion, the generated structure included close to no standard elements: each bar occurring only twice within each quarter due to the bilateral-symmetry, each knot occurring only once, resulting in 1250 different types of bars and 1079 different types of knots for one quarter alone. Since such a variety of non-standard elements proved to be impossible for the purposes of constructing the model, an optimization algorithm was developed, to increase the number of standardized elements. Since manufacturing non-standard knots proved to be less problematic, as each was individually laser-cut out of sheet metal, the focus was on reducing the number of bar types.

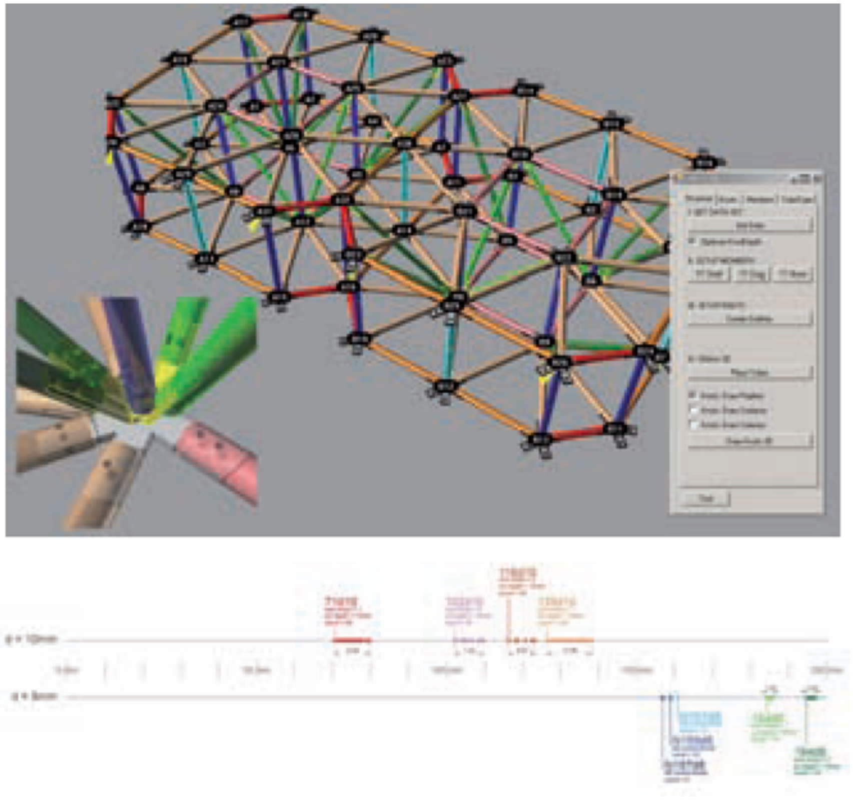


Figure 8: Structure Generation Plug-In (above), bar analysis and optimization (below).

Reducing the number of bar types was accomplished by first analyzing the distribution of maximum bar lengths for each diameter, strategically defining the length of standard bar lengths. Members with a similar length were included in a group if its length was between 0-15mm less than that of the standard element's length. Within each group, all members were then replaced by that standard bar type. As a result of reducing the length of each local bar, the arms of the knots, being parametrically defined, would grow to compensate for the difference (fig. 9). The optimization algorithm was integrated as a second step into the Plug-In. As a result, the 1250 different bar types were reduced to 44 standard elements, greatly reducing the cost of production and aiding the task of assembly.

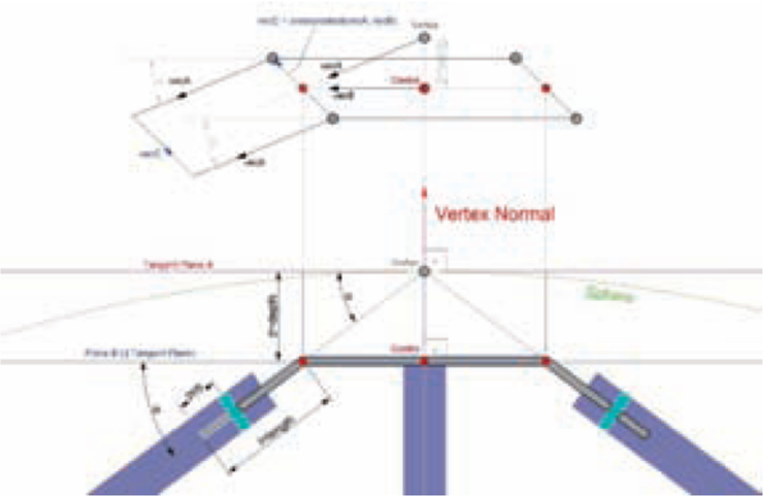


Figure 9: Geometric and Parametric definition of bars and knots

The knots, as mentioned, were each custom laser-cut, automatically flattened, labeled and nested to be cut out of flat metal sheets (fig. 10). All relevant information was engraved on each individual knot, the knot ID, the connecting bar types and neighboring knot numbers, greatly aiding the task of assembly.

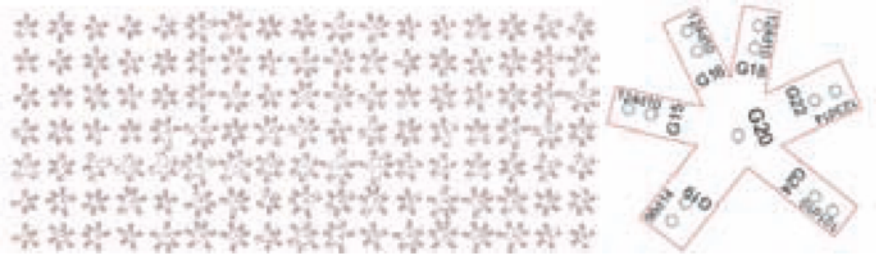


Figure 10: Automatically flattened, labeled and nested knots

Another point of concern was ensuring that the arms were bent to a precise angle, the angles for the arms being slightly different due to the effects of distortion and the varying lengths of connecting members. The range and differences in angles of every arm of every knot was analyzed. The result of the study, however, was that the effects were negligible, the angles ranging from approximately 0.5°-1.0° for knots lying on the surface of the spheres and approximately 55° for diagonal members (fig. 11). The flat knots were consequently pre-bent to a standard 1° and 55° respectively, slight variations in angles were compensated for.

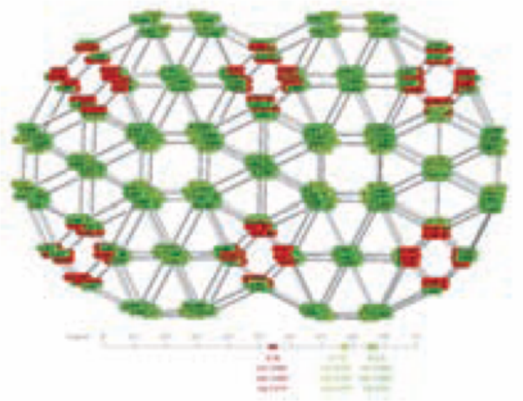


Figure 11: Knot Angle Analysis

Once the parts were generated, optimized and produced, the entire structure was eventually assembled element by element, module by module, and quarter by quarter into the final structure of the whole dome (fig. 12).

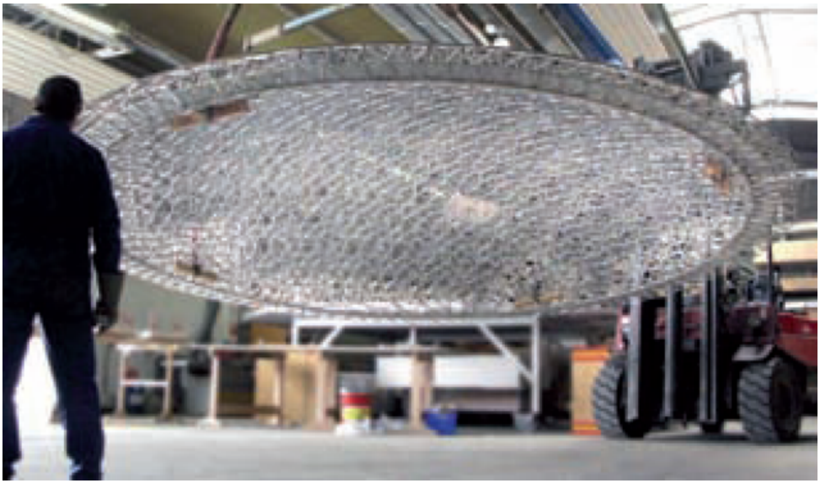


Figure 12: Finished Structure of the entire Dome

# 3 CLADDING

In addition to the development and assembly of the structure, the task of preparing the data for and producing the domes cladding had to be solved in parallel. The design by Nouvel envisiones the cladding of the dome to consist of an overlay of multi-layered (5 layers on top and 5 below) strips of varying widths.

Producing the cladding for the dome proved to be another challenge, as the model had to withstand temperature differences of approximately 60-70°K, while being used during the testing phase in Abu Dhabi; due to heat expansion, it was therefore decided to use the same material as the structure: The cladding was produced of layers of spherically shaped aluminum sheets, AlMg3, H111.

## 3.1 Cladding Data

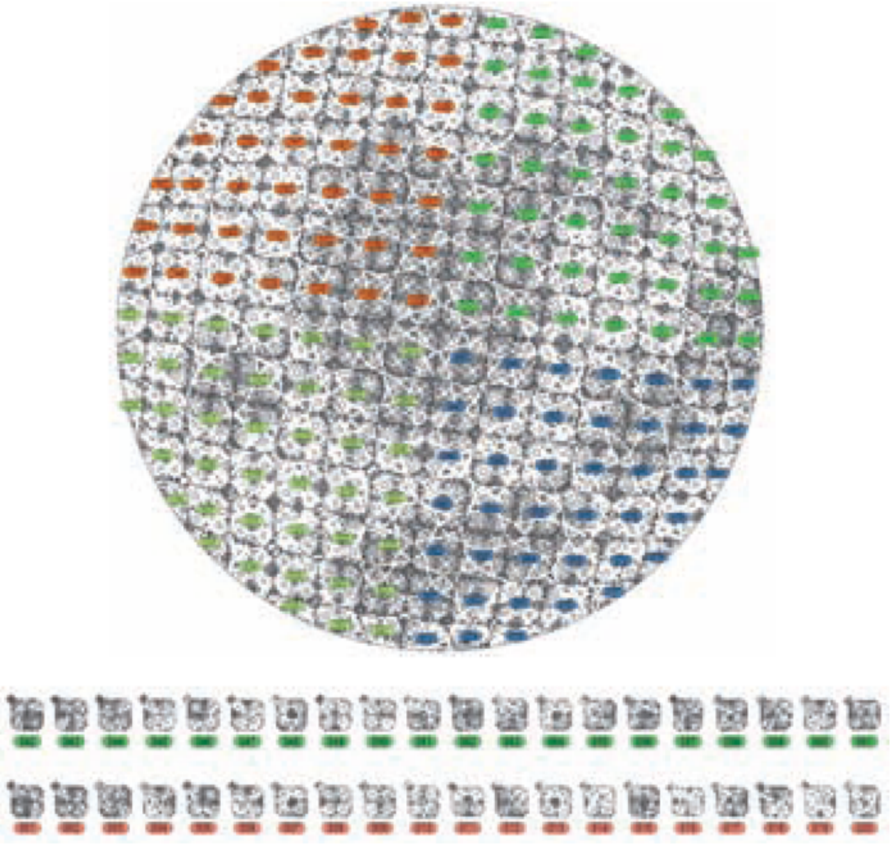


Figure 13: Cladding: Tiling Modules, assembled (above), automatic production file generation (below).

As with the structure, the data for the cladding was provided for by the architects, as a multitude of single quad-strips, lacking any detail for construction at the scale of the model. Proving impossible to realize the cladding down at the small scale of the strips, a repetitive tiling pattern was sought for, whose tiles could ideally be produced out of a single raw format, and that would follow the structure, in order to be fixed to the knots. A regular, square tiling pattern could not be used, as it would follow the structure not in all places. The result was a tile at the scale of the structural modules, that was based on a truncated square tiling pattern, that perfectly followed the centerlines of the structure, essentially and octagon with a square "nose" in one corner (fig. 13).

All geometric operations regarding the cladding, such as defining the outlines, and gaps between modules, had to be carried out in a flat plane. For that reason, the 3-D data provided for by the architects had to be projected onto a plane. An orthographic projection could not be used, however, as the cladding for the model had to be separated into different levels, due to the thickness of the aluminum sheets and the merging of layers, for reasons of cost, which would have distorted the pattern of the cladding markedly. All operations that were developed for preparing the cladding, also as part of the Rhino Plug-In, were based on gnomic projections (i.e. towards the centre of the sphere of the dome), the tangent plane being the XY world-construction plane.

## 3.2 Cladding Tile Production and Assembly

As with the structure, the cladding modules were first generated in 3-D, and automatically flattened and exported to be cut. The cladding modules consisted of small, spherically formed tiles of approx. 600 x 600mm, three layers each (fig. 14), with a gap of 0,8mm between each neighboring module to allow for heat expansion. The raw sheets for the cladding tiles were pressure formed (fig 15.a). As the formed aluminum has a tendency to spring back and rebound, it was difficult to achieve a plastic deformation in the aluminum sheets that would shape

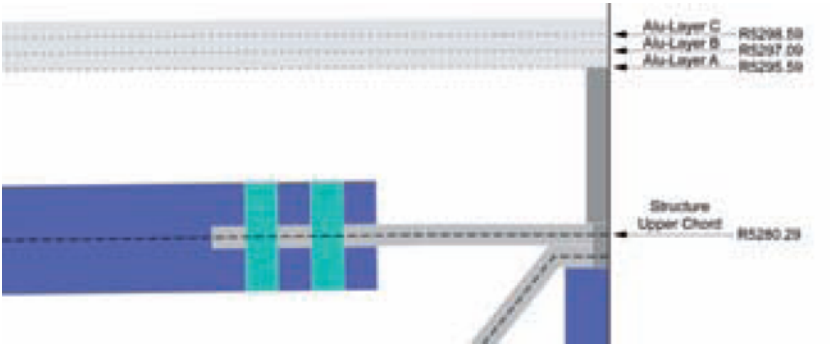


Figure 14: Outer Cladding: Three Aluminum Layers held in place above the knots by distance holders.

and hold the curvature of each module at the precise radius, due to the very slight curvature of the surface of the cladding. For that reason, the press tool was milled to a substantially smaller radius, to allow for the sheet to rebound to a radius aimed for. As there are no mathematical means available to anticipate the behavior of the metal sheet accurately, it had to be determined empirically, taking two trials, working closely with the metal workshop producing the sheets.

Once the raw modules were pressure formed, the pattern of each layer of each tile was water-cut (fig. 15.b), each module being different. The three layers of each module were consequently glued together, pressed in between two forms with the precise radius, allowing for the glue to dry, which would result in an even greater approximation of the final radius. The prepared cladding modules were then assembled onto the structure, covering its entire top and bottom area (fig. 16).

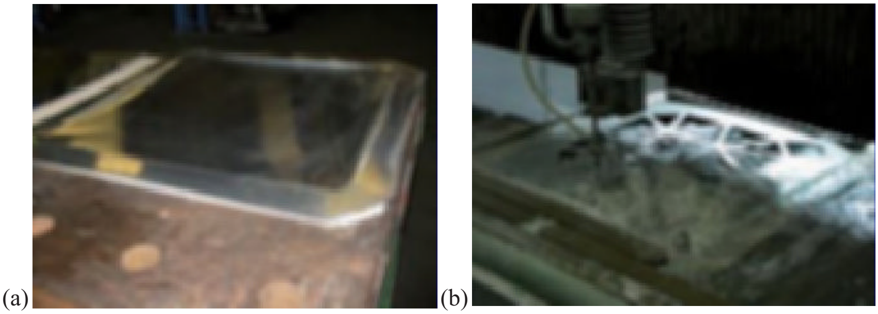


Figure 15: Pressed aluminum sheet and watercutting of the pattern

Once the structure of the dome had been completed and fully clad, it was disassembled, transported in five parts to Abu Dhabi and reassembled on site to be used for the light-tests.

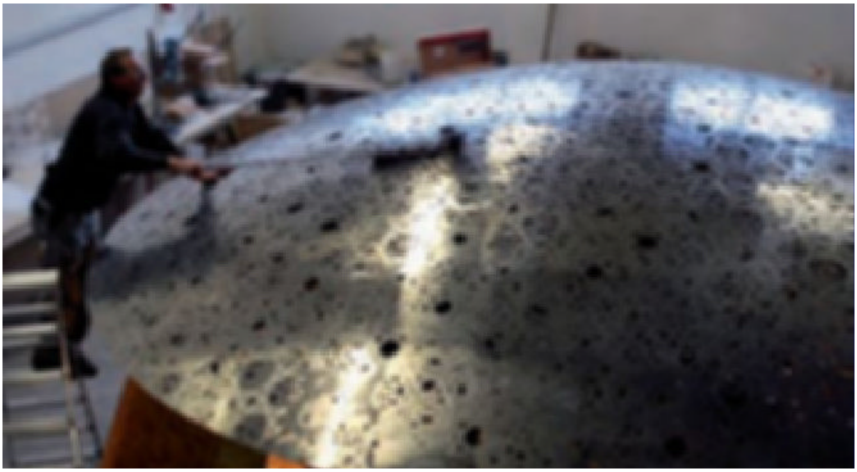


Figure 16: Fully clad model

## 4 Conclusion

The construction of the light-test model of the Louvre Abu Dhabi at scale 1/33 not only served its purpose as a light-test model but also as a precursor to the forthcoming full-scale dome erection, as many of the issues that will be relevant in construction had to be addressed for the first time. It confirmed the need of using advanced methods in computation and fabrication as well as an engaging an interdisciplinary team to be able to realize a project of such geometric complexity.

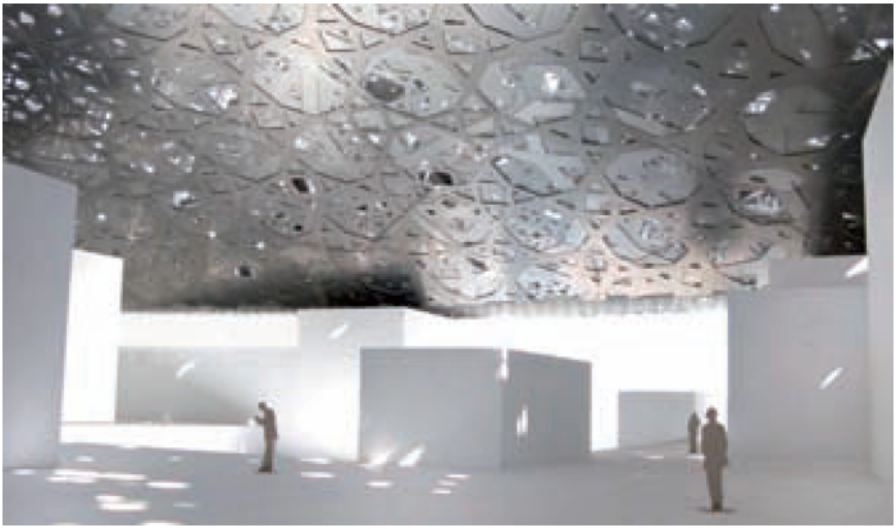


Figure 17: Interior view of the finalized model under the Abu Dhabi sun

## Acknowledgements

Published with kind permission by the Tourism Development & Investment Company (TDIC) and Ateliers Jean Nouvel.

Construction of the 1/33 model has been cooperation between the companies George Ackermann GmbH, Honkahe Interior+Furniture and 1:One.

## References

ELECTRO OPTICAL SYSTEMS (EOS) [Internet]. [04/2004]. Alumide Material Data Sheet. Available from *http://www.crdm.co.uk/technical-pdf/Alumide.pdf*.

# Ortho-Pictures: 3D Objects from Independent 2D Data Sets.

Gershon Elber[1]

Dept. of Computer Science
Technion – IIT
Haifa 32000, Israel

**Abstract.**

*This work portrays several ways in which a pair of completely independent 2D data sets, such as regular 2D color pictures or height fields, could be merged into one 3D object, creating an* Ortho-picture*. The* Ortho-picture *is an object that portrays the first input set from one view and will identify with the second independent set from an orthogonal view. While techniques to reconstruct 3D geometry from several 2D data sets of the* same 2D models *are well known in image processing, herein we strive to merge pairs (or even triplets) of* completely independent *2D input data sets into one 3D object.*

*The end result of this effort is a regular 3D object that is automatically synthesized from two (or even three) completely independent pictures or 3D objects (converted into 2D height fields via a Z-buffer). This result is in line of the artwork of conceptual artists such as Shigeo Fukuda [Fukuda] and Markus Raetz [Raetz]. We present several, fully automated, ways to create* Ortho-pictures*, and show some examples.*

**keywords:** *Ortho-pictures, 3D statues, 3D dithering, 3D depth maps, Non realistic modeling (NRM), Art in science.*

## 1 Introduction

Working in the field of Non Photo-realistic Rendering (NPR), computer graphics researchers have always sought different ways to automatically create and emulate images that are, traditionally, produced by human artists, and beyond. In the same way, in [Sela and Elber 2007], we have used an approach, which we coined *Non-Realistic Modeling (NRM)*, to create visually pleasing 3D models for mostly artistic purposes. Our effort in [Sela and Elber 2007] was geared towards creating objects that resemble work of artists such as Shigeo Fukuda's "Duet" [Fukuda]. The "Duet" sculpture looks like a pianist playing the pianos from one view, and from another view like a violinist playing a violin. One fairly well known simple 3D model that resembles different shapes from different views is the 3D combination of a SQuare,

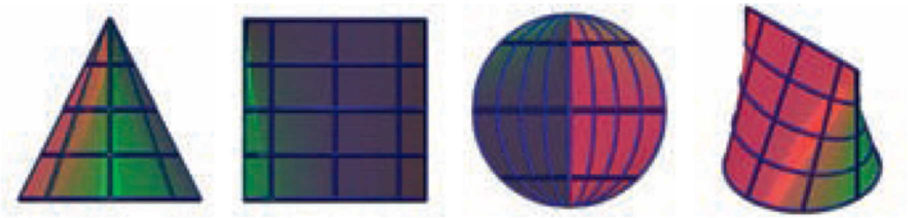


Figure 1: The sqriancle (SQuare, tRIANgle and cirCLE, following [Sela and Elber 2007]) is an object that resembles a square, a triangle and a circle when viewed from three different angles.

a tRIANgle and a cirCLE, coined *the sqriancle* in [Sela and Elber 2007] and is presented in Figure 1.

In this work, our (NRM) aim is to create 3D objects in $\mathbb{R}^3$, denoted *Ortho-pictures*, that resemble one 2D input set from one viewing direction, and another, completely different, 2D input set, from an orthogonal viewing direction to the first. In this NRM approach, the input includes two or three 2D data sets. These 2D data sets could simply be independent color (or gray-level) pictures, or, for example, be independent height fields. A height field to a 3D model, $M$, could be derived by extracting the $Z$ buffer's depth map of rendered object $M$.

Much like the pixel that is a primitive picture-element, hence after we will denote the primitive elements of our 3D objects as *objels*, for object-elements. By exploiting 3D objels that form a coverage for two/three 2D independent data sets from two/three orthogonal views, we are able to incorporate completely different 2D pictures and/or other 2D data sets and merge them into a one 3D object. The 3D objels forming the 3D object are quite unlimited as long as they satisfy proper mappings onto the domains of the input sets. Herein, we exemplify these mappings with 3D objels that are of minute size in all dimensions, much like a pixel of an image, and denote these objels as zero-dimensional. Further, we also consider objels that span one full dimension, as a curve, and denote such objels as one-dimensional.

The rest of this work is organized as follows. In Section 2, we provide some background and survey related work. In Section 3, we propose several algorithms to compute ortho-pictures and present a few examples, and finally, we conclude in Section 4.

## 2 Background and Related Work

The synergy between the sciences and the arts, and between geometric modeling and the plastic art in specific, is gaining a momentum in recent years. There is a growing recognition by artists that computers could serve in the creation of more compelling and intriguing geometries. Tools such as Maya [Maya] and 3D studio Max [3ds Max] are nowadays seen as fundamental tools to be studied in any art school.

Unfortunately, not much work can be found in the geometric modeling com-

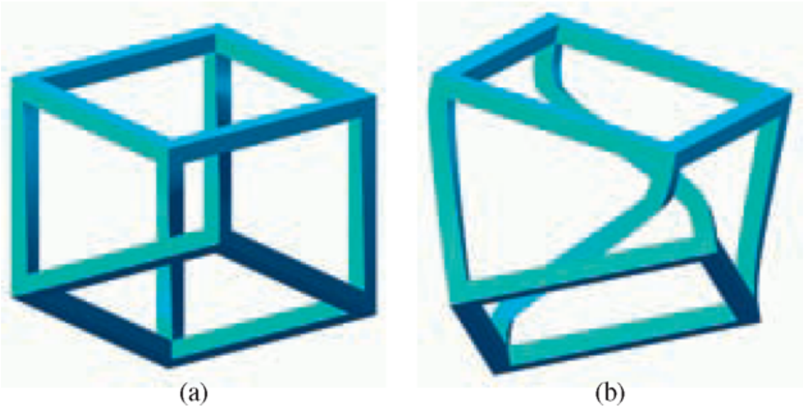


Figure 2: The “Escher for Real” (aka ’Necker’) Cube [Elber 2002b] as seen in the ‘Belvedere’ drawing of M.C. Escher. (a) shows the seemingly impossible cube as seen in the original ‘Belvedere’ drawing, while in (b) the model’s ‘impossibility’ is exposed from a general view. © Copyright Gershon Elber 2002.

munity on artistically-oriented modeling. In Fact, most contemporary geometric modeling packages are geared toward mechanical design. Yet, modeling efforts in the mechanical CAD/CAM fields focus on the creation of 3D models that satisfy design and/or manufacturing needs.

Traditional CAD/CAM design postulates high accuracy and precision needs that are not as significant in the arts, in general. Sketch based design is also an emerging area in both the industry, i.e. Sketchup [Sketchup], and academia, i.e. Teddy, by Igarashi et. al. [Igarashi et al. 1999], which allows end users to model simplistic 3D shapes using a few mouse silhouette sketches.

In the “Escher for Real” project [Elber 2002b], tangible 3D models that resemble the so called “impossible drawings” of M.C. Escher’s are created. The 3D models appear to be identical to the original 2D drawings from a single viewing direction. From any other direction, they are revealed to be a valid yet deformed 3D geometric model; See Figure 2 for a simple example. Similarly, the “Beyond Escher for Real” project [Elber 2002a] follows other artists beyond Escher and also includes, among other things, a few models that portray two completely independent shapes from two orthogonal views. For example, see Figure 3: A Menorah (the state emblem of Israel) is fused with the Star of David into one model. These models, of [Elber 2002a] and [Elber 2002b], were all created manually.

Another unique and quite impressive set of so called “impossible models”, that are physically realized, is made by [Sugihara]. His models are all manually crafted from paper and yet exploit depth misperceptions to the extreme.

Given two or even three outlines of some shapes, and under certain assumptions, one can extrude these outlines in $X$, $Y$, and $Z$, only to intersect the extrusions in $I\!R^3$ and create a (non-smooth) 3D object in $I\!R^3$ that projects in three different directions to these three outlines. Clearly not every outline could be fused with any other

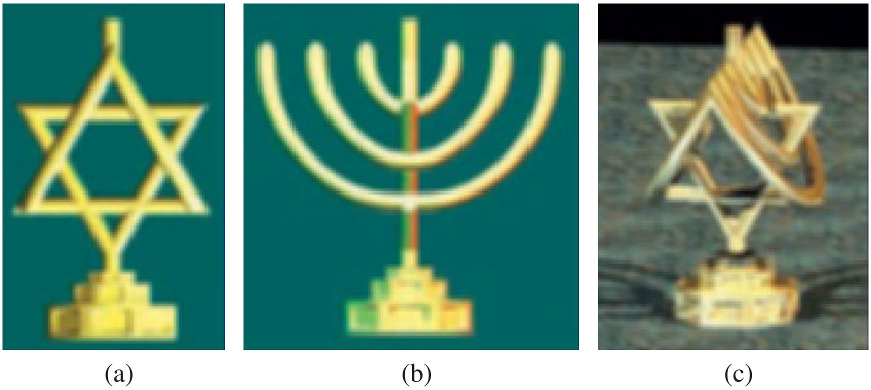


Figure 3: The “Beyond Escher for Real” Star-of-David/Menorah [Elber 2002a] model. (a) shows the view of the Menorah model while (b) presents the view of the Star-of-David. In (c), a general (raytraced) view of this model is presented. © Copyright Gershon Elber 2002.

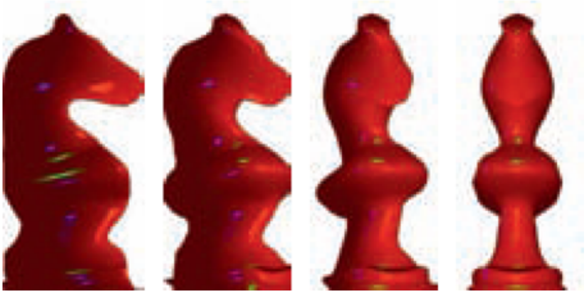


Figure 4: One example of the ‘knishop’ - a combination of a ‘knight’ (left) and and a ‘bishop’ (right) as one merged smooth 3D model, is presented in a few different views. Computed using a silhouettes’ based deformation, following [Sela and Elber 2007].

outline. Consider, for instance, one outline that converges to a line (or even a point). In the recent publication of [Mitra and Pauly 2009], outline shadows are used as a source to try and synthesize 3D objects. Since the solution does not always exist, the problem is posed as an optimization problem.

In [Sela and Elber 2007], the problem of creating a 3D model that resembles two different shapes is posed as an optimization problem of deforming a given object to follow the silhouettes of a different object. Given two objects, *A* and *B*, we seek one merged smooth object that looks like *A* from one view and like *B* from an orthogonal view. The desired silhouettes of *A* and *B* are extracted and a smooth deformation is computed to deform (the silhouettes of) *A* to look like (the silhouettes of) *B* from one view. Then, the outlines of *A* will follow the silhouettes of *B* from that one view and will remain as originally was, from an orthogonal view. Figure 4 shows one result of [Sela and Elber 2007].

In 1938, Victor Vasarely drew his ‘Zebra’ picture. This drawing includes a set

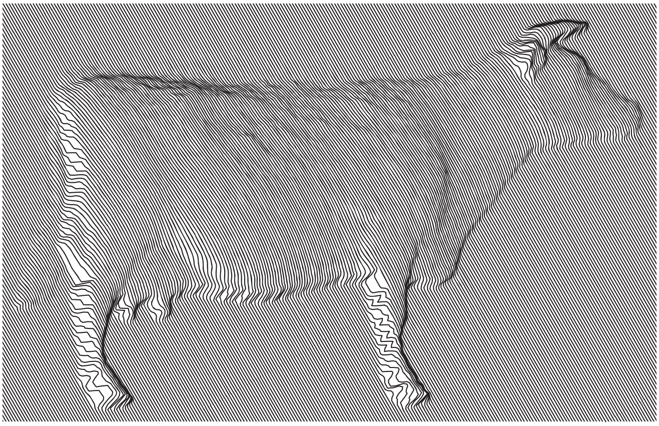


Figure 5: An algorithm that automatically synthesizes a drawing in the style of the ‘Zebra’ drawing by Victor Vasarely, from a given Z map’s height field of a 3D model, is presented in [Elber 2001].

of parallel stripes, that are shifted in the plane to create the perception of depth. The more these stripes are shifted, the deeper that regions seems to be. In [Elber 2001], we presented a simple yet automatic algorithm to take a 3D model and convert it into a depth-map using a Z-buffer, only to be used as a source for local shifting/bending of some parallel stripes, following the style of the original ‘Zebra’ picture. See Figure 5 for one example.

In contrast to the little work in the geometric modeling field on NRM, some NRM examples in the art world do exist. Shigeo Fukuda [Fukuda] created such works as “Duet”, “Love Story” and “Cat/Mouse”, all resembling two different objects from two orthogonal directions. Another relevant artist is Markus Raetz [Raetz], who also sculpts such works, for example “Metamorphose”, and also a series of pieces showing one word from one direction, and another word, usually the antonym, from another. Examples are “Yes/No” (in many languages), “This/That” and one interesting piece titled “Same/Same” [Raetz], which reads ”Same” from the front, and “Same” read backwards from the side. Hence, its reflection also reads “Same” on a well positioned mirror. Francis Tabary [Tabary] is another artist that creates such 3D- and wire-statues that projects into two independent words, in two orthogonal projections.

Yaacov Agam [Agam] is yet another well known artist for the creation of 3D kinetic statues that look completely different from different views. Agam also uses technologies such as lenticular printing to create his agamographs that looks completely different from different views.

# 3 Constructing Ortho-Pictures

In this section, we start by describing two different algorithms to compute Ortho-pictures. In the first, in Section 3.1, we extend the idea of automatically synthesizing Vasarely style pictures (Following [Elber 2001]) for two images in 3D. Then, in Section 3.2, we present a different way to synthesize an ortho-picture from two (or three) given 2D images. Finally, in Section 3.3, we consider several possible extensions.

## 3.1 Vasarely style Ortho-Pictures

Drawings of parallel stripes in the style of Figure 5 can create artistically appealing 2D drawings that portray depth. One can extend this idea and combine two such drawings, that are completely independent, into one 3D object. Consider a one-dimensional 3D tube objel, parallel to the $Z$ axis. Then, one has two independent degrees of freedom to bend each tube. One bending degree of freedom is around the $X$ axis and the other is to bend the tube around the $Y$ axis. Assuming an infinite vertical (along $Z$) tube, a small bending of the tube around the $X$ axis does not affect the shape of the tube as projected into the $XZ$ plane but affects it in the $YZ$ projection plane. Similarly, bending the (infinite) tube around the $Y$ axis does not affect the shape of the tube as projected into the $YZ$ plane while affecting the projection into $XZ$ plane

This simple idea stems from the complete independence (due to the orthogonality) of the bending in the two, $X$ and $Y$, axes. Let $Perm(N)$ be some permutations vector of size $N$ of the integers between zero and $N-1$ (or stated in computer science terms, a *perfect hasing* of integers to themselves). We can now create an arrangement above the $XY$ plane, in $I\!R^3$, of $N$ vertical tube objels that project to $N$ parallel and equally spaced (thick) lines in the $XZ$ plane and also to $N$ parallel and equally spaced (thick) lines in the $YZ$ plane. In other words, these $N$ objels form a uniform coverage of both the $XZ$ and $YZ$ planes. See Algorithm 3.1, steps 1-5.

In [Elber 2001], A depth map is created using a Z-buffer to a given 3D scene. Then, a deformation in the plane is used to bend and shift the stripes or tubes based on the depth. The deeper some location is, the greater the $XY$-created shift, resulting in drawings such as Figure 5. With an arrangement of $N$ tubes in $I\!R^3$, we now have two different yet completely independent problems of bending and shifting tubes in a plane, once in the $XZ$ plane and once in the $YZ$ plane. Each of these two planar problems is independently solved following and using [Elber 2001]. This Vasarely style bending process, that is extended to $I\!R^3$, is summarized in Algorithm 3.1. In [Elber 2001], the bending function is expressed as a self-intersection free deformation mapping from the plane to itself. The set of parallel tubes (lines in [Elber 2001]) undergoes the bending via the application of the computed deformation. In steps 6-8 of Algorithm 3.1, two such independent planar deformation mappings, denoted $\mathcal{DM}$, are computed and employed, following [Elber 2001], one for the $XZ$ plane and one for the $YZ$ plane.

Clearly, the permutations of $Perm(N)$ in lines 1 and 2 of Algorithm 3.1 could be anything. Random ordering of the $N$ numbers would yield a random placement

**Algorithm 3.1 (Vasarely Style 3D Ortho-Pictures)**
**Input:**
*N: Number of desired vertical tubes in the 3D Ortho-Picture;*
$I_1$, $I_2$*: Two square depth–images of the same size* $(N \times N)$*;*
**Output:**
*A 3D Ortho-Picture of N vertical tubes mimicking* $I_1$ *and* $I_2$ *from two orthogonal views, Vasarely style.*
**Algorithm:**

```
$V_1 \leftarrow Perm(N)$;
$V_2 \leftarrow Perm(N)$;
for $i = 0$ to $N - 1$ do
    $\mathcal{T}_i \leftarrow$ A vertical tube at XY coordinates $(V_1[i], V_2[i])$;
end for
$\mathcal{DM}_{XZ} \leftarrow$ Planar deformation mapping, Vasarely style, using Algorithm [Elber
  2001] for $I_1$;
$\mathcal{DM}_{YZ} \leftarrow$ Planar deformation mapping, Vasarely style, using Algorithm [Elber
  2001] for $I_2$;
$\{\tilde{\mathcal{T}}_i\} \leftarrow$ Deform $\{\mathcal{T}_i\}$ in the XZ plane following $\mathcal{DM}_{XZ}$ and in the YZ plane
  following $\mathcal{DM}_{YZ}$, $\forall\ i$;
Emit $\{\tilde{\mathcal{T}}_i\}$;
```

of the vertical tubes in the $XY$ plane[2]. This is the result of the example shown in Figure 6. If we assign $V_j$, $j = 1, 2$, with an ordered set, as $V_j[i] = i$, the constructed tubes (and the created 3D Ortho-Pictures) will all be along the $X = Y$ diagonal plane. See Figure 7 for an example of this case.

### 3.2 Ortho-Pictures from two 2D Images

Expanding on the ideas presented in Section 3.1, shading effects could be locally achieved by painting the local geometry in the local colors or gray levels of images $I_i$, $i = 1, 2$. If color is to be supported[3] the geometry of the ortho-picture could be locally painted to convey the proper color from the $X$ viewing direction from image $I_1$ and the proper color from the $Y$ viewing direction from image $I_2$. Because the geometry could locally be facing both the $X$ and the $Y$ viewing direction, such a solution is not completely independent for a generally shaped geometry. More on this later on.

However, gray-levels could also be achieved by controlling the local coverage of bright covering objels over a dark background, much like dithering. One such possibility is to exploit bright partially-covering objels of varying sizes over dark background. Locally scaling the objels down will expose more of the dark background while a full sized objel will appear locally white. Here, we will start with

[2] if $Perm(N)$ is random, any new invocation of $Perm(N)$ is assumed to yield a different result.

[3] There already exist layered manufacturing processes, such as by ZCorp (www.zcorp.com) or Objet (www.objet.com), that support fabrication of 3D geometry with colors or gray-levels

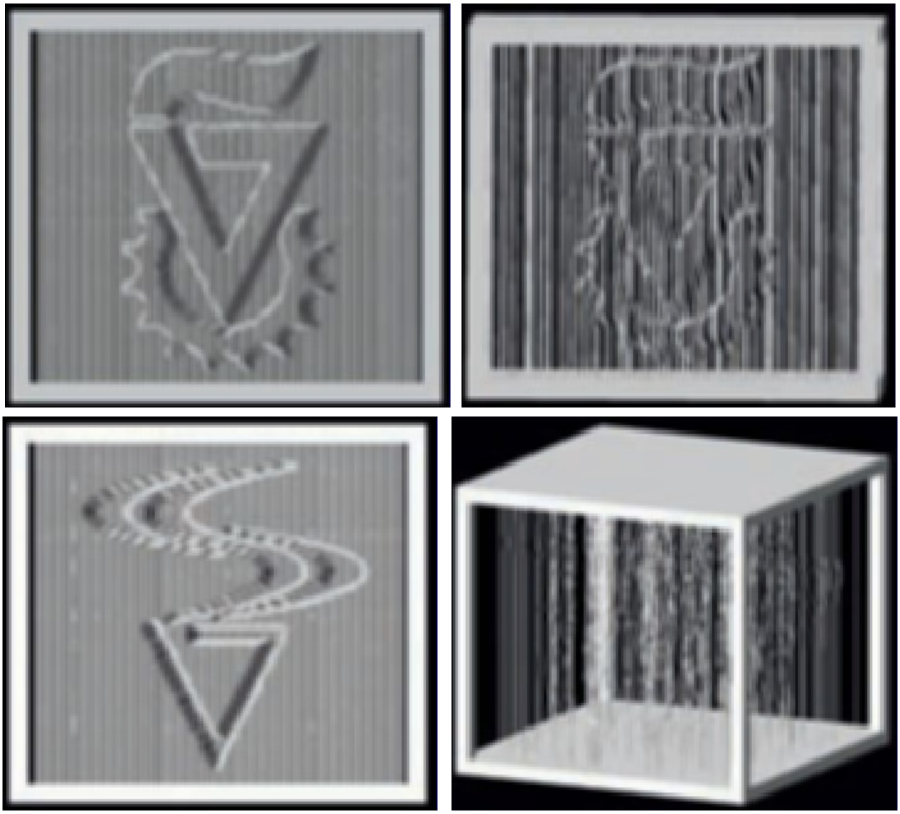


Figure 6: An example of randomly spread parallel 3D tubes. The two input models are the logos of the Technion university (top left) and the logo of its CS department (bottom left). The right figures show general views of the same ortho-picture, where the images blend. 160 vertical tubes are used. Compare with Figure 7.

some 3D zero-dimensional objels, also denoted *blobs*.

Consider two 2D input images of size $(N \times N)$. Randomly spread $N^2$ unit size blobs in a 3D cubical space of size $(N \times N \times N)$ so that from both the $X$ viewing direction and from the $Y$ viewing direction all $(N \times N)$ pixels are uniquely covered by exactly one blob objel. This distribution could be easily achieved as follows: For each of the $Z = 0$ to $N-1$ layers (of $N$ objels each), place $N$ objels at coordinates $(Perm(N) \times Perm(N))$. This holds since these $N$ blob objels will be projected and cover all $N$ (0 to $N-1$) slots in the $X$ and/or the $Y$ directions as we have all instances of $\{0, 1, 2, \cdots, N-1\}$ in the $Perm(N)$ sets, much like in Algorithm 3.1.

With this geometric arrangement of $N^2$ properly spread blob objels in an $(N \times N \times N)$ volume, we can independently scale each (bright) blob in $X$ and/or $Y$ (over a dark background) based on the two independent gray-levels of the two original

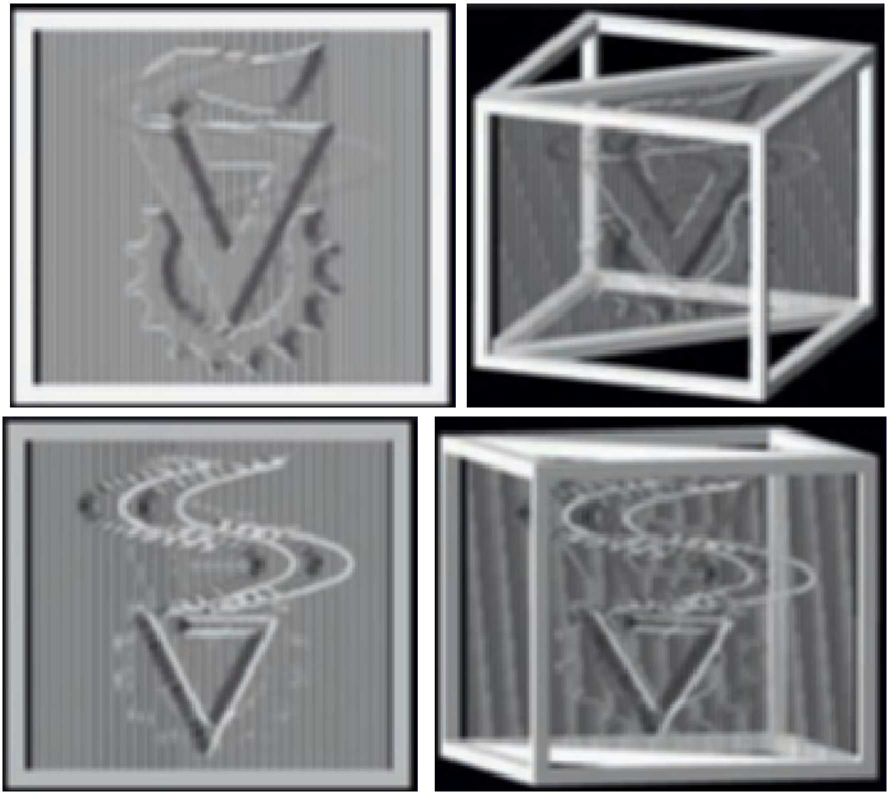


Figure 7: An example of parallel 3D tubes along a diagonal plane. The two input models are the logos of the Technion university (top left), and the logo of its CS department (bottom left). The right figures show general views of the same ortho-picture, where the images blend. 160 vertical tubes are used. Compare with Figure 6.

images at that location, creating shading effects. This whole process is presented as Algorithm 3.2.

Line 8 of Algorithm 3.2 performs the non-uniform scale of each blob objel to its proper local (bright over dark background) coverage for both the *X* and the *Y* viewing directions. Line 9 of Algorithm 3.2 then places the blob in its proper, 3-space, position.

The input image could be either a gray-levels image or a colored image. In the latter case, the colored image could be easily converted into gray-levels using the well accepted CIE conversion [Foley et al. 1990] of,

$$Gray = Red * 0.3 + Green * 0.59 + Blue * 0.11.$$

Finally, and almost needless to say, the need to have a square, similarly sized, images is artificial for the simplicity of the presented algorithms. These constraints

**Algorithm 3.2 (Shaded Style 3D Ortho-Pictures)**
**Input:**
*N: Number of desired spherical blob objels in the 3D Ortho-Picture along an axis;*
$I_1$, $I_2$*: Two square regular shaded (color or B&W) pictures of same size* $(N \times N)$*;*
**Output:**
$N^2$ *blob objels forming a 3D Ortho-Picture mimicking* $I_1$ *and* $I_2$ *from two orthogonal views;*
**Algorithm:**

```
for $Z = 0$ to $N-1$ do
    $V_1 \leftarrow Perm(N)$;
    $V_2 \leftarrow Perm(N)$;
    for $i = 0$ to $N-1$ do
        $\mathcal{B}_i^Z \leftarrow$ a unit size blob objel, at the origin;
        $I_1^Z \leftarrow$ Intensity of image $I_1$ at coordinate $(V_1[i], Z)$;
        $I_2^Z \leftarrow$ Intensity of image $I_2$ at coordinate $(V_2[i], Z)$;
        $\mathcal{B}_i^Z \leftarrow \mathcal{B}_i^Z * ScaleX(I_1^Z) * ScaleY(I_2^Z)$;
        $\mathcal{B}_i^Z \leftarrow \mathcal{B}_i^Z * Translate(V_1[i], V_2[i], Z)$;
    end for
end for
Emit $\{\mathcal{B}_i^Z\}$, $\forall i, Z$;
```

could be easily relaxed via the application of a proper image resizing to any desired size.

Figure 8 shows one example of the Clintons' couple with 10,000 randomly placed blob objels ($N$ = 100). Figure 9 presents another example of Herzl and Ben Gurion, two famous Jewish leader, using 10,000 blob objels that are placed along a diagonal (using the trivial permutations of $V_j[i] = i$).

### 3.3 Possible extensions

While Algorithm 3.1 assumed vertical tubes (along the $Z$ axis) nothing actually demands this and one can equally well exploit parallel tubes at any angle as long as a one-to-one and onto coverage is established between the tubes and the domains of the two inputs. Moreover, the initial set up does not even require the starting tubes to be linear. In [Elber 2001], Vasarely style drawings of parallel circles or even parallels (offsets) of general curves were exemplified. Nothing herein prevents one from the use of general parallel curved tubes (and their parallel offsets).

We showed two types of bijective mappings, using one-dimensional tube objels, extending the Vasarely's original drawing idea into 3D, in Algorithm 3.1, and zero-dimensional blob objels, in Algorithm 3.2. Clearly, other types of bijective coverings could be established. As one example, consider the different shapes of blobs that one can use. Figures 8 and 9 employ spherical objels that are scaled into ellipsoids following the desired shading level from either the $X$ or the $Y$ viewing directions, by properly and independently scaling each bright spherical blob in $X$

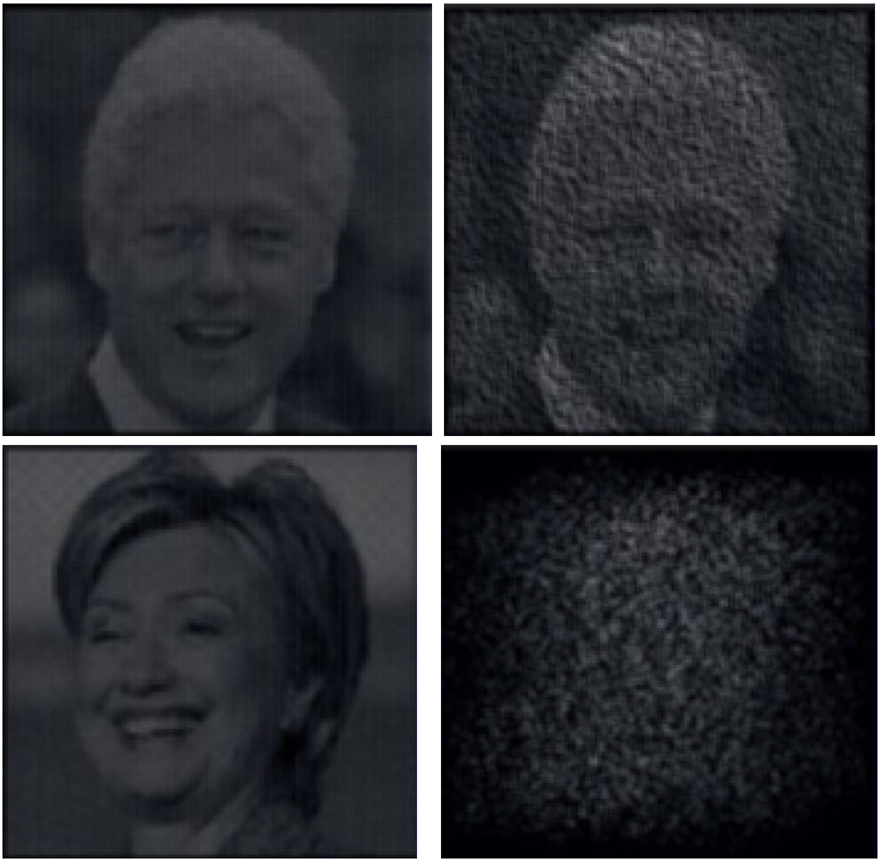


Figure 8: An example of an ortho-picture of the Clintons' couple using a covering with bright blob objels scaled to impose the necessary shade, over a dark background. The right figures show off-orthogonal view directions of this 3D object. 10,000 blobs are randomly spread in a $(100 \times 100 \times 100)$ volume. Compare with Figure 9.

and in $Y$ over a dark background. Figure 10 shows a few options of different shapes of blob objels. The box–shaped blob (Figure 10 (a)) has the benefit that it can (independently for $X$ and $Y$) change its local coverage from zero to 100%. A spherical blob (Figure 10 (b)) yields a more smoothed shape, having no sharp corners. Yet, the maximal local coverage of a spherical blob will be $\pi/4$, the ratio of the areas between a circle and its bounding square. A cross shaped blob (Figure 10 (c)), formed out of two orthogonal polygons and hence denoted the *2-cross*, has the advantage that only two polygons are used to represent a blob, making this objel the most memory-efficient blob's shape.

The 2-cross shaped blob also offer full covering abilities from zero to 100%. As stated, blobs such as the sphere can only yield a coverage up to $\pi/4$ if confined

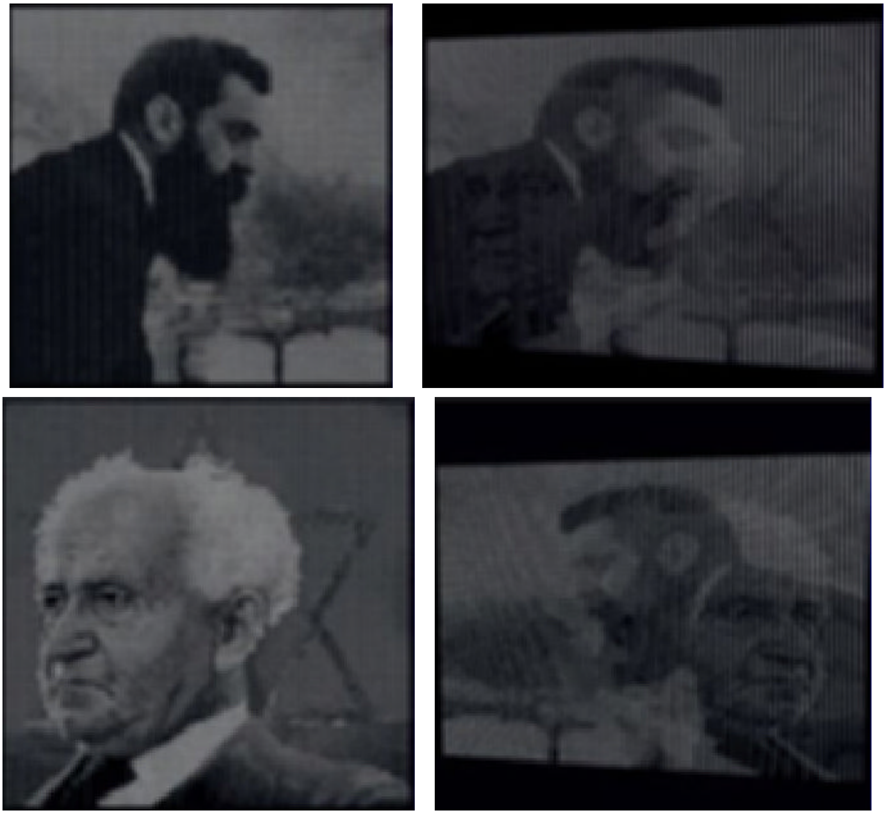


Figure 9: An example of an ortho-picture of Herzl (top left) and Ben Gurion (bottom left), two famous Jewish leaders, using a covering with bright blob objels scaled to impose the necessary shade, over a dark background. The right figures show off-orthogonal view directions of the 3D object. 10,000 blobs are spread along a diagonal plane in a $(100 \times 100)$ diagonal grids. Compare with Figure 8.

to their cell. However, if we do allow neighboring blobs to intersect, the maximal coverage might improve at the cost of some interference, affecting regions of neighboring blobs as well.

The box and the 2-cross blob have a different advantage, due to the fact that no polygon of the box and/or the 2-cross blob is simultaneously facing both the $X$ and the $Y$ viewing direction. These two types of blobs only present polygonal faces parallel to the $XZ$ and the $YZ$ planes which make them highly suitable for actual shading. If a colored blob is supposed to be locally red from the $X$ viewing direction (for the $YZ$ projection plane) and is supposed to be locally green from the $Y$ viewing direction, one can independently paint in red and green the relevant faces of the box or the two faces of the 2-cross (See Figure 10 (d)). In Figure 11, the cross–shaped blobs are shaded and colored to follow the pixel they represent in the

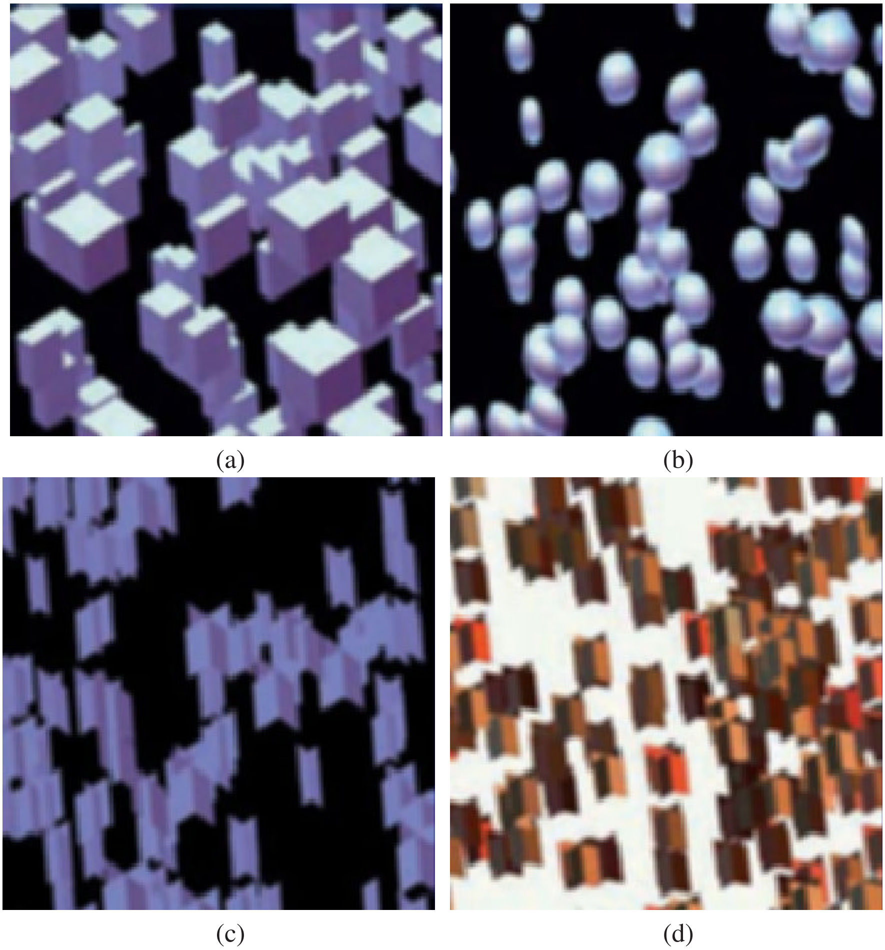


Figure 10: A few examples of possible blob objels that could be used in the coverage. A box (a), a sphere (b), or a simple 2-cross of two polygons (c), including in color (d).

two original input images. Each blob conveys two independent shades or colors. If a blob such as a sphere was employed, shades and/or colors were forced to be blended between the *X* and *Y* axes, achieving an inferior result.

An interesting question is how can one manufacture these objects, especially in color? Potentially and however difficult, the object with Vasarely style tube objels could be manufactured using wire bending. In contrast, having 10,000 blobs spread in some volume poses a major construction challenge if manufactured individually. Fortunately the layered manufacturing technology is getting to a point where different materials could be mixed in the same object allowing for a transparent box in which the opaque blobs could be embedded. Figure 12 shows two such tangible ob-

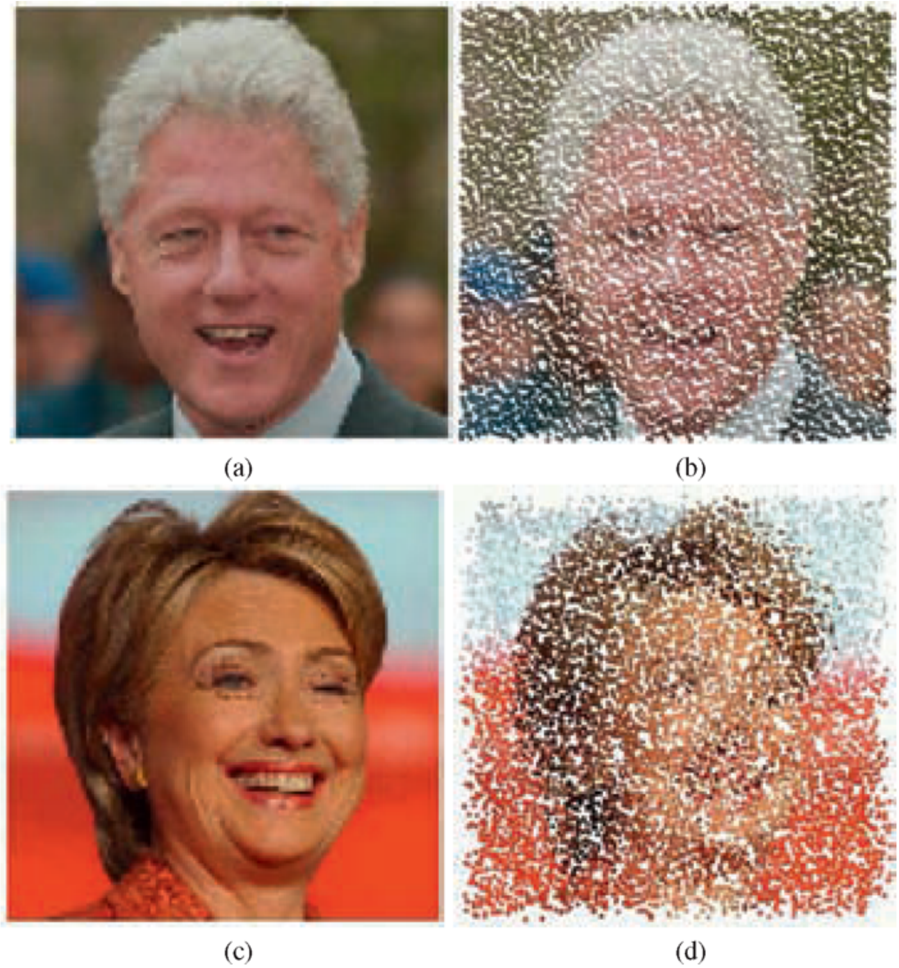


Figure 11: An example of the Clintons' couple, this time in color. The left side shows the orthogonal views while the right side show off-orthogonal view directions of the 3D object, that, again, blends the two input images. The object uses 10,000 blobs, each of which is shaped as a 2-cross, and painted in two independent colors (See also Figure 10 (d).

jects of the model in Figure 6 and the model in Figure 9 using technology from Objet (www.objet.com). The realization of these objects is not perfect. Transparency is not complete and interference of light between the two views does happen. Yet, the results are highly promising and in the right direction. Moreover, some layered manufacturing technologies also support shading and even start to process coloring of individual locations. Another interesting alternative to consider is the possibility of laser imprinting in glass, a technology one can find in every shopping mall nowadays, having glass imprints in key-chains, presents, etc.

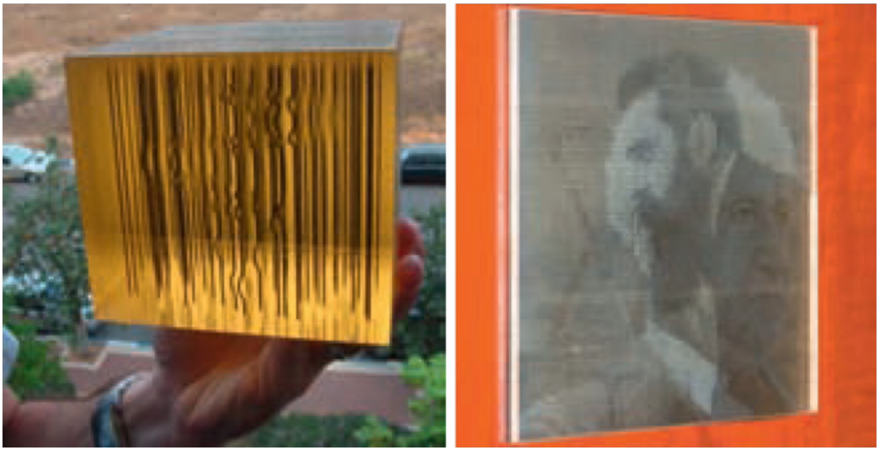


Figure 12: Examples of realized objects from Figures 6 and 9, realized using modern layered manufacturing technology from Object (www.objet.com).

In Algorithm 3.2, shading was achieved by scaling blobs. One can similarly consider adding shading abilities by locally thickening the tubes from Algorithm 3.1, based on the desired gray-level, in the two viewing direction. Again, due to the orthogonality, one can locally thicken the 3D tubes independently in $X$ or in $Y$, getting an independent shading for these two views. In other words, instead of dealing with 3D tubes of a circular cross section, we will now have elliptically-varying cross sections that depends on the local gray-levels at that location from the $X$ and $Y$ viewing directions. This shading option could be applied with or without the Vasarely tube-bending option. Alternatively, the original color and/or gray-level could also be added by painting the local tube's area with the desired color/gray-level. In fact, instead of circular cross section tubes, one can employ rectangle cross section tubes to achieve full independence in the use of coloring in $X$ and in $Y$, as is done for the box and 2-cross blobs. In summary, we can build sets of parallel tubes that would convey either depth by bending or convey color or gray-levels by painting or by scaling the cross sections, or both. Placing such colored straight rectangle cross section tubes along a diagonal, like in Figure 7, converges to art created by [Agam].

In general, one independent degree of freedom (axis) is needed to scale the bright blobs over dark background. Hence, in $\mathbb{R}^n$ one can present only $n-1$ independent data sets. However, we can create a singular blob, denoted *3-cross*, that presents no dependency between its *three* axes in $\mathbb{R}^3$, as a blob of three orthogonal square polygons in the $XY$, the $XZ$ and the $YZ$ planes. $N^2$ 3-cross blobs could be spread in a $(N \times N \times N)$ domain to form a coverage in all three axes of view. One should recall that while virtually such an object is simple to construct, in reality a compromise must be made when such an object is manufactured as some minimal thickness is to be expected. Figure 13 shows one example using this interesting yet singular 3-cross case, having the blobs spread along two diagonal triangles. Random spread

**Algorithm 3.3 (3-cross random spread)**
**Input:**
*N: Number of desired blobs in each axis ($N^2$ blobs in a 3D Ortho-Picture);*
**Output:**
*M: A matrix of size $(N \times N)$ of Z values to place the $(i, j)$ blob at $(i, j, M[i][j])$, achieving a coverage in three orthogonal directions;*
**Algorithm:**

```
U ← Perm(N);
V ← Perm(N);
for i = 0 to N − 1 do
    s = { k | U[i] = V[k] };
    M[i] ← V cyclically shifted s locations;
end for
```

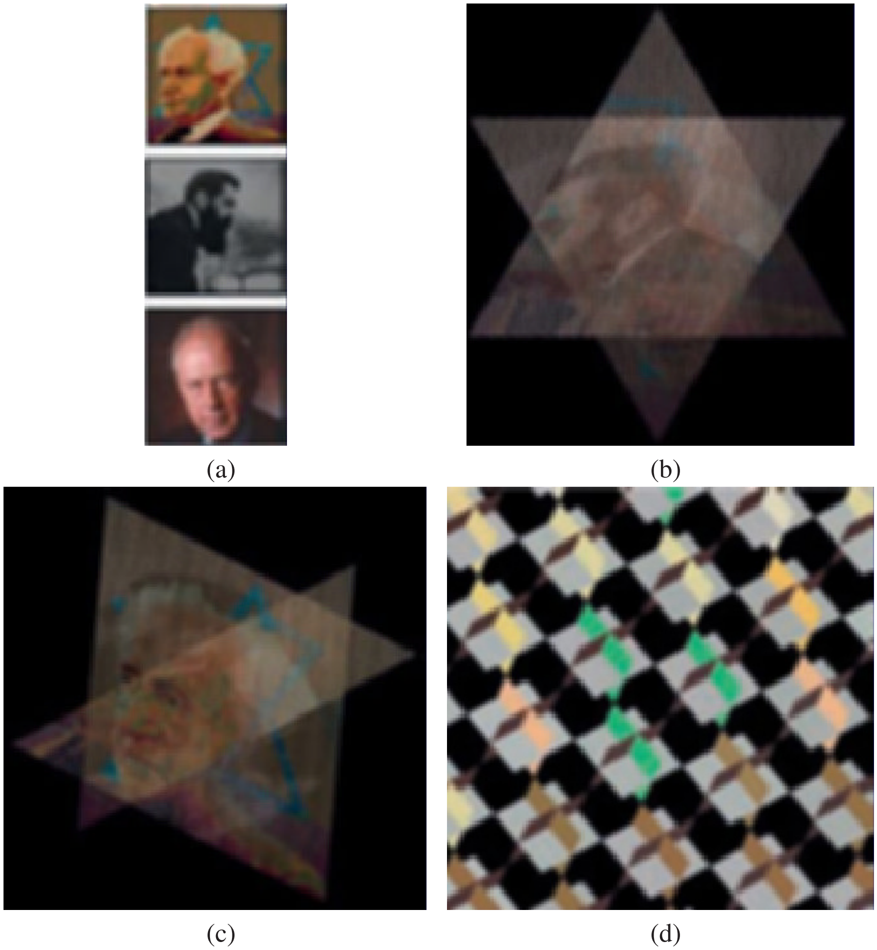
of blobs that satisfies these covering requirements can also be achieved as is shown in Algorithm 3.3.

Clearly, $Perm(N)$ forms a coverage in the first dimensions, by definition. However, Algorithm 3.3 terminates with $M[i][k] \neq M[j][k]$, $i \neq j$, $\forall k$, as they hold the same permutation vector $V$ but shifted differently, ensuring a coverage in the second dimension. Finally, $M$ is a 2D matrix that spans all the $XY$ domain, forming a coverage for the third dimension.

In this work, we emphasized the need for orthogonality throughout the presentation. Yet, this orthogonality is not really mandatory. A simple convincing argument toward this end works as follow. Create a regular 3D ortho-picture object that is orthogonal, with viewing directions along the $X$ and $Y$ axes. Then, apply some $(XY)$ skewing matrix transformation to the 3D ortho-picture object. The skewing matrix will have the effect of changing the view directions of the ortho-picture from $X$ and $Y$ to any two different directions in (not just!) the $XY$ plane.

Many of the objects reconstructed here create illusions of straight lines that are not straight and of flat regions that are actually bent. Users inspecting such objects expect the lighting of such regions to be uniform. Two alternative remedies exist. In the first, we can assign planar normals to the non planar regions, in essence further deceiving the user. The second alternative is to design a special lighting configuration that alleviate the lighting discrepancy. The first solution fits well virtual, computerized, representations whereas the second will have to be used in case of real, tangible, objects.

The presented solutions are all about coverages. Two types of objels were presented as covering elements, the tubes and the blobs. In $\mathbb{R}^3$, for a 2D picture, zero-dimensional objels (blobs) and one dimensional objels (tubes) are the only types one can expect to use. However, in $\mathbb{R}^n$, higher order objels could possibly be considered as well, for higher dimensional input sets. Further, any covering scheme that uses zero- and/or one-dimensional objels could be considered for ortho-pictures in $\mathbb{R}^3$. Drawing from NPR, an adaptive coverages that is based on curves and is presented in [Elber 1995] is one additional possibility.



(a) (b)

(c) (d)

Figure 13: An example of an ortho-picture of three different 2D images (Ben Gurion ((a) top), Herzl ((a) middle), and Rabin ((a) Bottom)) fused together into one 3D object in $\mathbb{R}^3$. The ortho-picture is formed out of blob objels oriented along two triangles seen in (b)-(c) in two general views. The 3D object consists of close to 8,000 blobs, each of which is shaped as a 3-cross, as seen in (d) which is a zoom-in on the blobs.

## 4 Conclusions

We have presented algorithms to merge two (or even three) 2D input sets into a one 3D object, in $\mathbb{R}^3$. Supported 2D input includes depth and color and/or gray-level images that by positioning bright geometry over dark background (or vice versa) could yield effective shading. These algorithms are all fully automatic and create 3D objects that could be manufactured using modern technologies such as layered manufacturing and/or laser imprinting in glass.

## Acknowledgment

I would like to thank Yossi Abu, from Objet, for his help in making the parts in Figure 12.

## References

3DS MAX, A.
http://www.autodesk.com/3dsmax

AGAM, Y.
http://en.wikipedia.org/wiki/Yaacov_Agam

ELBER, G. 1995. Line art rendering via a coverage of isoparametric curves. *IEEE Transactions on Visualization and Computer Graphics 1*, 3, 231–239.

ELBER, G. 2001. Rendering with parallel stripes. *Computer Graphics and Applications 21*, 3, 44–52.

ELBER, G., 2002. "Beyond Escher for Real" project.
http://www.cs.technion.ac.il/~gershon/BeyondEscherForReal

ELBER, G., 2002. "Escher for Real" project.
http://www.cs.technion.ac.il/~gershon/EscherForReal

FOLEY, J. D., VAN DAM, A., FEINER, S. K., AND HUGHES, J. F. 1990. *Fundamentals of Interactive Computer Graphics*. Addison-Wesley Publishing Company, second edition.

FUKUDA, S. "works".
http://neuro.caltech.edu/~seckel/mod/fukuda.htm

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: A sketching interface for 3D freeform design. In *SIGGRAPH*, 409–416.

MAYA, A.
http://www.autodesk.com/maya

MITRA, N. J., AND PAULY, M. 2009. Shadow art. In *SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers*, ACM, New York, NY, USA, 1–7.

RAETZ, M. ”metamorphose”.
www.crownpoint.com/artists/raetz/index2.html

SELA, G., AND ELBER, G. 2007. Generation of view dependent models using free form deformation. *The Visual Computer 23*, 219–229.

SKETCHUP, G. http://sketchup.google.com

SUGIHARA, K.
http://home.mims.meiji.ac.jp/~sugihara/hobby/hobbye.html

TABARY, F. http://www.francistabary.com

# Geometry of Structural Form

**Lorenz Lachauer**
ETH Zürich

**Toni Kotnik**
ETH Zürich

**Abstract.** *This paper describes a precise geometric method for the inscription of structural constraints into architectural form. Based on techniques from graphic statics, the force distribution in building structures is  visualized using geometric diagrams. This diagrammatic representation allows a formal description that shows the relationship between the force flow and the structural form. The formal character of this description enables the direct implementation of a parametric truss geometry that maintains major structural behavioral characteristics under deformation. An interactive model of a structural freeform roof is developed through this link between a parametric truss definition and a design-driving NURBS surface. This allows for an intuitive exploration of the constrained design space in real time. Formal explorations and the comparison with built examples demonstrate the effectiveness of this approach.*

**Keywords:** Architectural Freeform Surface, Structural Geometry, Interactive Design, Reciprocal Diagrams, Parametric Modeling, Graphic Statics

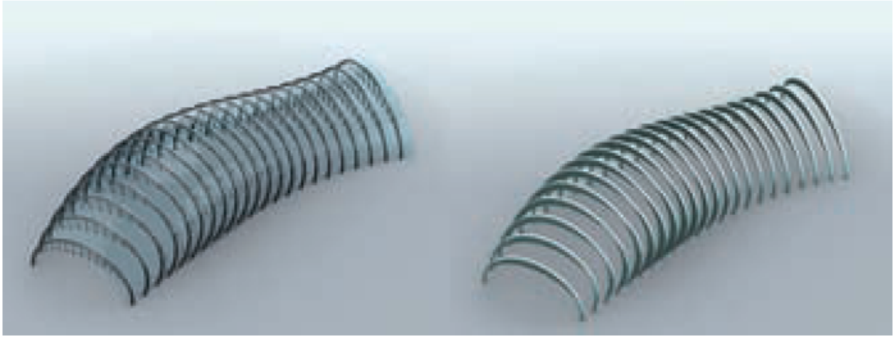


Figure 1: These freeform roof structures have efficient load-bearing capabilities and were designed by the direct integration of structural constraints into form. The methods used in the design of these structures are presented in this paper.

# 1 Introduction

Since the first use of digital design tools in architecture during the 1990s, the popularity of geometric modeling techniques has constantly increased. Originally coming from the airplane and film industries, these design tools were eventually adapted and embedded into architectural design software. In the last decade, the emergence of parametric modeling and scripting techniques in architectural CAD applications has enabled a new level of sophistication in freeform design.

With growing formal complexity in design, the realization of such forms becomes increasingly challenging. Substantial research has been done on the problem of geometric constraints of architectural freeform surfaces [Schober 2002], [Shelden 2002], [Pottmann 2007], [Pottmann 2008], whereas research on their structural behavior remains fragmentary. For this reason, it is necessary to develop strategies for the integration of structural constraints in the design of architectural freeform surfaces.

Structural limitations have to be integrated in the design process of freeform surfaces in order to reduce the amount of building material without decreasing the stability and usability of a structure. Methodologically, two different directions can be identified.

On the one hand, optimization approaches, that are limited to a post-rationalization of shape. They focus on using numerical methods to enhance a given structure, by minor changes in geometry, in a late design phase [Bollinger et al. 2005], [Sasaki 2007], [Tessmann 2008].

On the other hand, direct approaches, which integrate structural constraints into the design process in an early phase. The first attempts in this approach were based on physical models as a possibility for a direct integration of structural constraints with architectural form [Gass 1990], [Kotnik 2010]. The recent development of digital simulations of the physical behavior of hanging models has successfully integrated structural constraints in the digital design process of compression-only surfaces [Kilian 2004]. In direct approaches, the shape is the direct result of the flow of forces through the material. The visualization and construction of this force flow, therefore, can be seen as the starting point for the early integration of structural constraints into the design process.

Graphic statics is a geometric method for the representation of the force flow in structures that was developed in the mid-19th century [Culmann 1875], [Maurer and Ramm 1998]. This technique can be used to either analyze the internal forces in a structure for a given load condition, or to design a structural form for a given force distribution. As an example, the geometry of the Eiffel tower was designed using graphic statics [Charlton 1982]. The visual, diagrammatic character of graphic methods allows an intuitive exploration of different design alternatives and their structural implications [Muttoni 2004], [Schwartz 2009], [Allen and Zalewszky 2009].

Recent approaches combine graphic methods with interactive web applications for educational purposes [Greenwold and Allen 2001], [Block and Ochsendorf 2005]. The development of associative geometric modeling tools for architects, for instance *Grasshopper* and *Generative Components*, allows for a direct integration of geometric relations into an interactive parametric model.

The topic of this research is the integration of structural constraints in a parametric model using graphic statics. This paper is structured as follows: Section 2 briefly summarizes the geometric foundations of graphic statics. Section 3 shows a graphical design method for trusses. Section 4 reveals the concept of a structural relationship derived from the invariance of force-flow. To conclude, Section 5 describes the setup of the interactive parametric model of a freeform roof. The generation of examples of roof forms illustrates the power of this approach.

## 2 Graphic Statics and Reciprocal Diagrams

The method of graphic statics is based on the *reciprocal* relationship between two diagrams, that was first described for planar cases by Clerk Maxwell and Luigi Cremona [Maxwell 1864], [Cremona 1890]. Later, this reciprocal or *dual* relationship was extended to fully three dimensional cases [Crapo 1979]. One diagram, the *form diagram*, represents the geometry of the structure and the location of the applied loads. Its reciprocal figure, the *force diagram*, represents the vectorial force distribution in the structure [Muttoni 2004], [Schwartz 2009] [Allen and Zalewszky 2009]. In general, there is not a unique reciprocal diagram for a given form diagram. The force diagram is always scalable, it does not represent the absolute values of the forces, but rather the ratios between the forces. Irrespective of scaling, there is only one force diagram for a statically determinate structure. The question of degrees of freedom in the reciprocal diagram for indeterminate form diagrams has been studied recently [Block 2009]. Reciprocal figures have following properties:

- Each line $L$ in the force diagram represents one dual line $L^*$ in the form diagram and vice versa.
- Corresponding lines in form and force diagram are parallel.
- The length of the lines in the force diagram is proportional to the forces in the structure.
- If one connects the supports and the load vectors in the form diagram with an outer point, then the dual graph of this figure has the same topology as the force diagram. (Fig. 2)

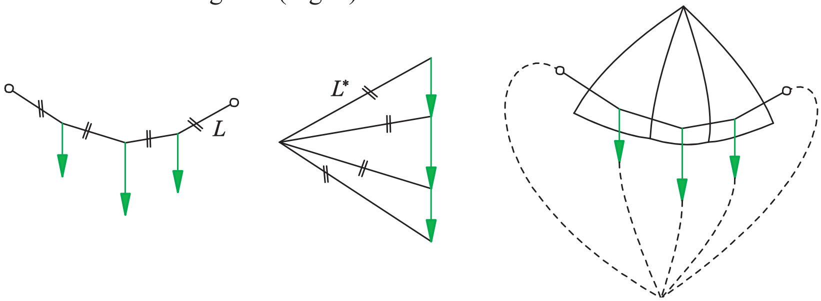


Figure 2: A *funicular polygon*: the form of a hanging string under the influence of forces, its force diagram and the visualization of their topological relationship.

The methods of graphic statics are applications of discrete differential geometry in two dimensions. For instance, in the construction of the *funicular polygon,* the form of a hanging string under the influence of forces corresponds to the repeated integration of the load vector field due to a graphical algorithm [Maurer and Ramm 1998]. With an increasing number of segments in a funicular polygon, the directions of the segments become tangents to a curve and its shape eventually attains that of the continuous curve of a string under the influence of a uniform load. (Fig. 3)

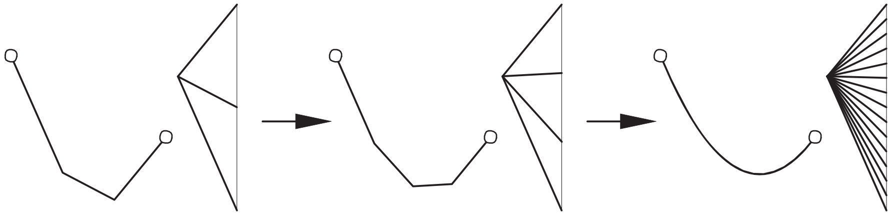


Figure 3: The funicular polygon leads to the form of a continuous curve of a hanging string with an increasing number of segments.

## 3 Geometry of an Efficient Truss

In order to demonstrate the method of graphic statics and its application to design, a geometric method for the procedural construction of a planar truss will be presented. The technique is based on a design method for the *constant chord force truss* [Allen and Zalewsky 2009]. This method generates a truss form with the top chord in pure compression and the bottom chord in pure tension for dead load[1]. Additionally, the tension forces in the bottom chord are all equal. While Edward Allen and Waclaw Zalewsky describe the application of this method for specific top chord shapes, this paper explores the possibility of this method for arbitrary top chord forms.

The truss form is constructed from a given discrete curve $S$ consisting of the segments $S_1, S_2, \ldots, S_n$, defining the geometry of the top chord, and chord force $F$. For each node of the top chord, $S$, a dead load component, $Q_i$, is assumed. The first step is to construct the reciprocal diagram from the chord segments $S_i$ , the nodal weights $Q_1, Q_2, \ldots, Q_{n-1}$,, and $F$. The second step is to construct the bottom chord of the truss.

The construction of the force diagram is straight forward: The nodal loads $Q_i^*$ in the force diagram are graphically added. The support forces $A$ and $B$ are derived either by the lever rule or graphically by a trial funicular [Schwartz 2009]. The circle $C$ is then constructed around the tip of the force vector $A^*$, with radius $F$.

[1] In structural design, permanent loads, especially the self weight of the building components, are called *dead loads*. All changing loads like wind, snow, movable objects, and people are called *live loads*.

The absolute value of $F$ must be large enough such that the reciprocal load components $Q_i^*$ are entirely located inside the circle. Next, construct the rays $S_i^*$ in the direction of the top chord segments $S_i$. (see Fig. 3) The connecting lines between the intersection points $I_i = C \cap S_i^*$ between the circle and the rays are the reciprocal representations $P_i^*$ of the truss members connecting the top and the bottom chord $P_i$. The representation of the force vectors $W_i^*$ in the bottom chord are constructed by the connection of the intersection points $I_i$ on the circle with the center of $C$. To construct the geometry of the bottom chord in the form diagram, start at support A and continue from left to right to the successive intersection of rays in the direction of $P_i^*$ and $W_i^*$, which are the nodes of the chord.

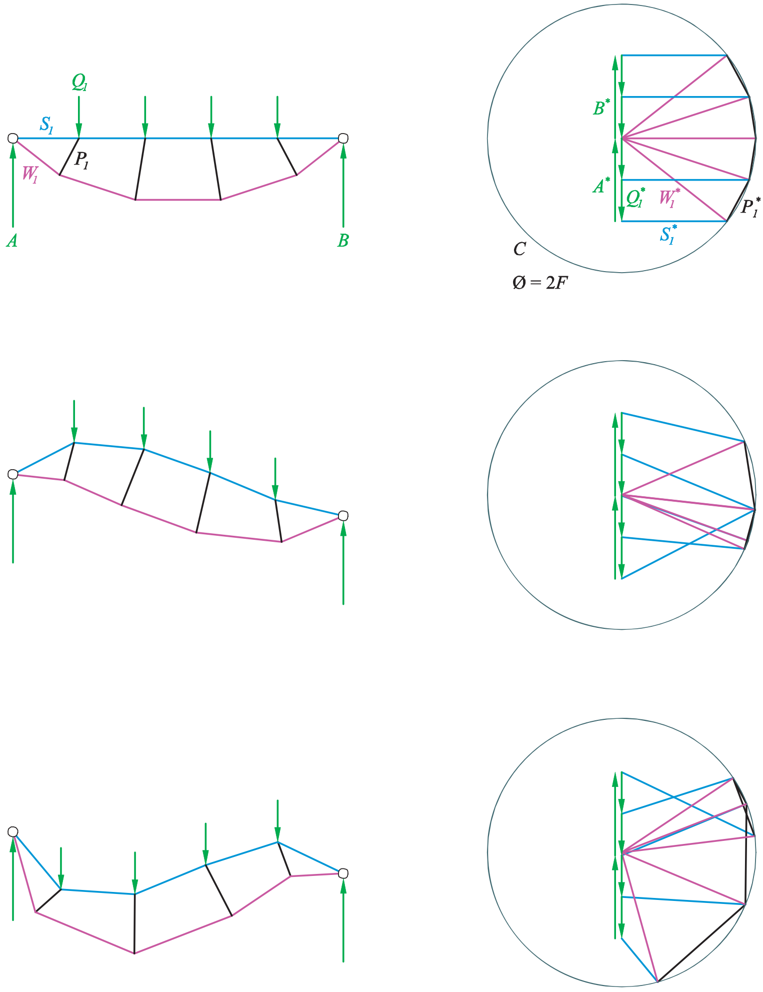


Figure 4: Form and force diagram for a constant chord force truss with five segments is shown in the first example. Implications of the deformation of the top

chord on the reciprocal diagram and the truss form can be seen in the two following examples.

# 4 Invariants in Force-Flow and Structural Classification

The above example of constant chord force trusses shows that the relationship between form and forces is fluid. However, there are properties that remain unchanged for transformations of the top chord S. These invariant properties enable a classification of truss geometries by similarities in the inner force distribution, independent of the morphological appearance of the truss.

The force diagram (Fig. 4) clearly shows that all dual representations of forces in the chord segments $W_i$ are radii, thereby demonstrating that are all inner forces equal:

$$\overline{W_i^*} = F \quad (1)$$

The dual representations of all truss members in the force diagram are contained by the circle, so all inner forces in all truss members are less or equal the circle diameter:

$$\overline{S_i^*} \leq 2\,F \quad \text{and} \quad \overline{P_i^*} \leq 2\,F \qquad (2)$$

The forces in the top chord segments $S_i$ and in $P_i$ are all compression, while in the bottom chord the segments $W_i$ are all tension. Together with (1) and (2), this allows a precise estimation of the dimensions of the truss elements, independent of the exact geometry of the truss.

For each transformation $f{:}\,S \to S'$ that maps the segmented curve $S$ to a curve $S'$, a mapping $g{:}\,T(S) \to T'(S')$ exists, which maps truss $T$ to $T'$. Without changing the chord force, $F$, the relations (1) and (2) are invariant for $g$. These invariant properties of the force distribution in truss geometries allow a classification of structural behavior based on properties of the force-flow. Typical structural classifications are based on morphology, e.g. the terms “beam,” “arch,” and “frame” point to a specific shape, more than to a specific structural behavior. All truss geometries that fulfill (1) and (2) may be seen as different shapes of a common class of related truss geometries. (Fig 5.) The examples demonstrate that the integration of structural constraints results in a coupling of design parameters. This calls into question the traditional differentiation of structural systems by shape. This definition of structural classes, which is based on geometric invariants of the force diagram as opposed to similarities of the form diagram, opens up a new understanding of building structures and the continuous relationship among structural forms.

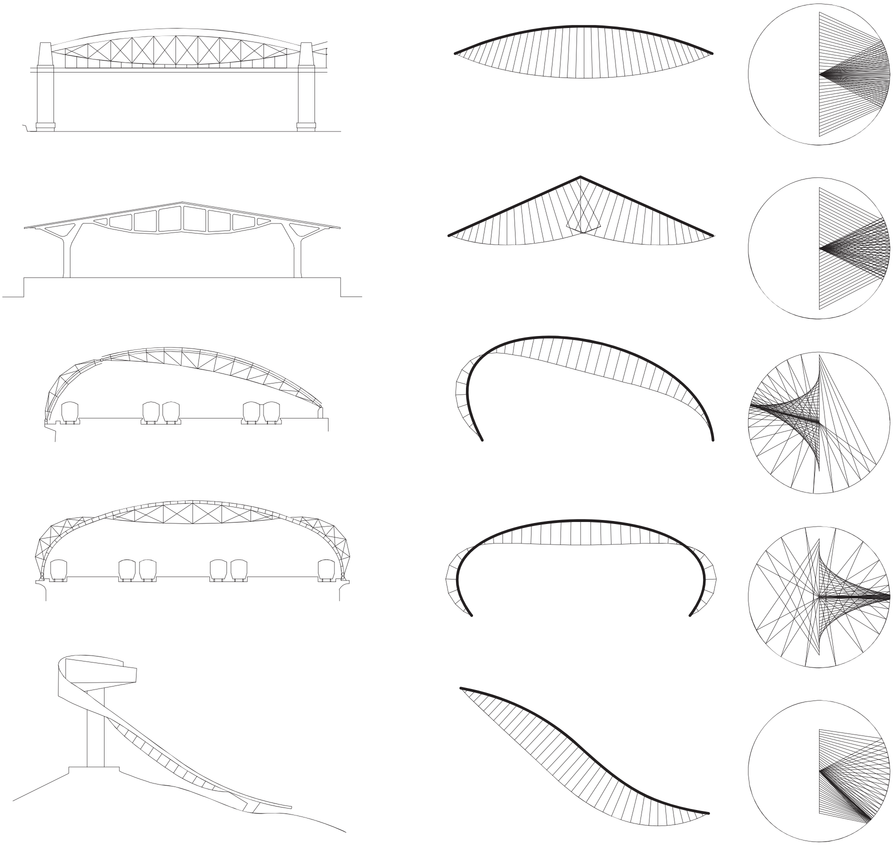


Figure 5: Building examples on the left, and examples of constant chord force truss geometries on the right. Their similarity in structural behavior becomes obvious in the force diagrams. From above: Luis Brunel: Royal Albert Bridge 1859; Robert Maillart: Chiasso Shed 1924; Grimshaw Architects: Waterloo Station 1993; Gerkan Marg + Partner: Lehrter Bahnhof 2002; Zaha Hadid Architects: Ski Jump 2001.

## 5 Parametric Model of an Efficient Freeform Roof Structure

This section describes an interactive parametric model of a freeform roof structure based on the above discussion. In the model, the gradually changing sections of the roof geometry are treated with the same structural principle without using the same truss morphology.

The parametric definition is built up using the associative modeling plug-in, *Grasshopper*, for the NURBS modeling CAD software, *Rhinoceros*. The definition has two input geometries: the freeform roof surface and a guiding curve  in the XY plane. The positions of the trusses are given by the guiding curve that is divided in segments of equal length. Straight lines normal to g define the truss axes. The top chord geometries of the trusses are derived by the segmentation of the vertical section curves through the roof surface. The self weight of the structure at each node of a truss is calculated by a polygonal approximation of the neighboring area around the node. (Fig 6.) Additionally, the model has four numerical input parameters: number of trusses, segments per truss, chord force, and a scale factor for the dead load.

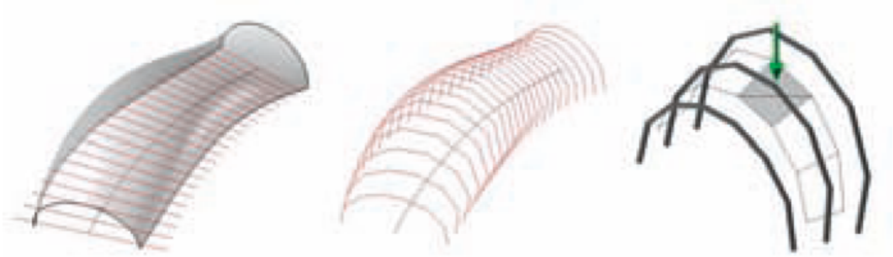


Figure 6: The generation of top chord geometry and loading from the input surface and guiding curve.

This parametric model enables the intuitive exploration of design alternatives for freeform structures supported by two edges. The interactive modification of input geometry works in real time because the direct procedural approach of geometry generation does not depend on computationally intensive algorithms. This allows for instant feedback to study the structural implications of formal choices.  In the example of a roof model with 50 trusses and 50 segments, changes in the surface geometry are updated within approximately one second using an Intel Core Duo Processor with 2.8 GHz. The use of a NURBS curve and a surface as input geometries provides full freedom in the design process. The input surface can either be manually sculpted by dragging the control points or generated by another script or algorithm. (Fig. 7)

It must be noted that truss geometries generated by the method described in Section 3 are in equilibrium only for the designing dead load. In order to resist additional loads such as wind and snow, the trusses have to be further stiffened. This could be achieved by the installation of diagonal braces or by a stiff top chord girder. (Fig. 1)

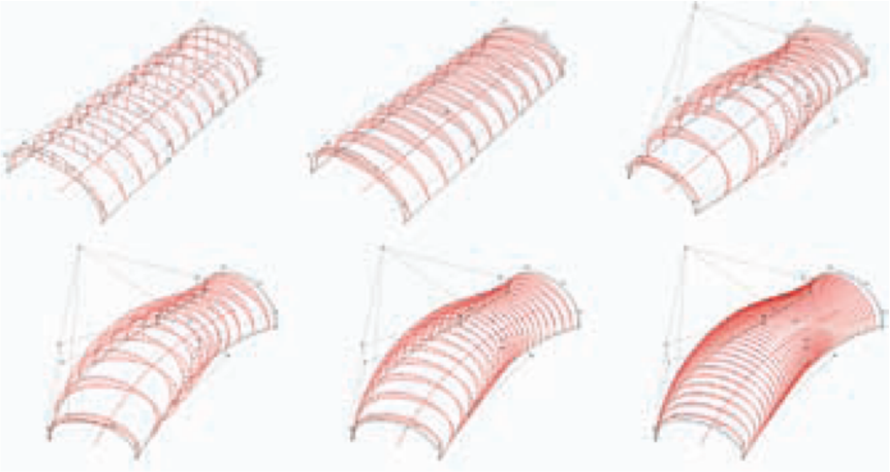


Figure 7: Design parameters of parametric roof are truss segmentation, truss number, surface form, and the form of the guiding line of the truss axes.

## 6 Conclusion and Future Work

As shown in this paper, the methods of graphic statics are easily applied to parametric CAD systems. These drafting-based techniques can be directly implemented using the rich variety of geometric operations provided by associative modeling environments. This allows for the direct integration of structural constraints in an interactive model that is based on the definition of geometric relationships and does not rely on additional computational techniques or software packages. Through graphic statics, structural problems can be treated with geometric means and may become questions of form.

This paper not only shows a new possibility for the development of interactive design tools, but also opens up a formal way of structural classification. The suggested classification is based on the inner force-flow and goes beyond the classical morphologic typologies of load bearing elements. It introduces a continuous relationship between structural forms, as it has been described a few years ago [Reiser and Umemoto 2006].

This is a first step towards parametric structural design based on graphical methods, shown for one specific class of truss geometries. Extensions are possible in several directions; a rich source may be historic collections of graphical methods [Malcolm 1914], [Wolfe 1921]. This work provides a basis for future exploration in the application of this method for different support conditions in two dimensions, such as a cantilevering beam a continuous beam. Another direction may be the extension of this truss to a spatial system, based on three dimensional extensions of the reciprocal relationship [Crapo 1979], [Micheletti 2008].

## Acknowledgements

We thank Philippe Block for inspiring discussions and Mallory Taub for editing.

## References

ALLEN E. AND ZALEWSKI W. 2010. *Form and Forces*. John Wiley and Sons, 275-300.

BLOCK P., AND OCHSENDORF, J. 2007. Thrust Network Analysis: A new methodology for three-dimensional equilibrium. *Journal of the International Association for Shell and Spatial Structures 48* (3), 167-173.

BLOCK P. 2009. *Thrust Network Analysis: Exploring Three-dimensional Equilibrium*. PhD thesis, MIT Cambridge. 72-82.

BOLLINGER K., GROHMANN M., PFANNER D., AND SCHNEIDER J. 2005. Tragkonstruktion der BMW-Welt in München. *Stahlbau, 74* (7), 483-491.

CHARLTON T. M. 1982. *A History of Structures in the Nineteenth Century*, Cambridge University Press.

CRAPO H. 1979. Structural Rigidity. *Structural Topology*, 1, 26-45.

CREMONA L. 1890. *Graphical Statics* (English Translation), Oxford University Press.

CULMANN C. 1875, *Die graphische Statik*. (2., neubearbeitete Auflage) Verlag von Meyer & Zeller.

GASS S. 1990. *Experimente*. Institut für leichte Flächentragwerke (IL) No. 25, 1.1-1.8.

GREENWOLD S. and ALLEN E. 2003. *Active Statics*. Available from: http://acg.media.mit.edu/people/simong/statics/data/index.html.

KILIAN, A. 2004. Linking digital hanging chain models to fabrication. *AIA/ACADIA* 23 Proc. 110-125.

KOTNIK T. 2010. Das Experiment als Entwurfsmethodik: Zur Möglichkeit der Integration naturwissenschaftlichen Arbeitens in der Architektur, in Moravansky, A. & Kirchengast, A. (eds.): *Experiments in Architecture*, Jowis, in press.

MALCOLM C. W. 1914. *A Text book on Graphic Statics*. Myron C. Clark Publishing Co.

MAURER B., AND RAMM E. 1998, *Karl Culmann und die graphische Statik*. Verlag für Geschichte der Naturwissenschaft und der Technik.

MAXWELL J. C. 1864. On Reciprocal Figures and Diagrams of Forces. *Philosophical Magazine 4*, 250-261.

MICHELETTI A. 2008. On Generalized Reciprocal Diagrams for Self-Stressed Frameworks. *International Journal of Space Structures 23* (3), 153-166.

MUTTONI, A. 2004. *L'art des structures: Une introduction au fonctionnement des structures en architecture.* Presses polytechniques et universitaires romandes.

POTTMANN H., LIU Y., WALLNER J., BOBENKO A., AND WANG W. 2007. Geometry of multi-layer freeform structures for architecture. *ACM Trans. Graphics, 26/3,* Proc. SIGGRAPH.

POTTMANN H., SCHIFTNER A., BO P., SCHMIEDHOFER H., WANG W., BALDASSINI

N., AND WALLNER J. 2008. Freeform surfaces from single curved panels. *ACM Trans. Graphics, 27/3,* Proc. SIGGRAPH.

REISER J. 2006. *Atlas of Novel Tectonics*. Princteon Architectual Press.

SASAKI M. 2007. *Morphogenesis of Flux Structure*. AA Publications, 101-109.

SCHOBER H. 2002. Geometrie-Prinzipien für wirtschaftliche und effiziente Schalentragwerke. *Bautechnik* (1), 16–24.

SCHWARTZ J. 2009. *Tragwerksentwurf I*, Skript zur Vorlesung, ETH Zürich.

SHELDEN D. R. 2002. *Digital surface representation and the constructability of Gehry's architecture*. PhD thesis, MIT Cambridge. 119-201.

TESSMANN O. 2008. *Collaborative Design Procedures for Architects and Engineers.* PhD thesis, Universität Kassel.

WOLFE W. S. 1921. *Graphical Analysis*. McGraw-Hill Book Company.

# Realizing Formal and Functional Complexity for Structurally Dynamic Systems in Rapid Computational Means: *Computational Methodology based on Particle Systems for Complex Tension-Active Form Generation*

**Sean Ahlquist**

University of Stuttgart, Institute for Computational Design

**Prof. Achim Menges**

University of Stuttgart, Institute for Computational Design

*Surface articulation provides a geometric vehicle to accomplish specific adaptation to conditions of materiality and spatiality. With tension-formed pre-stressed cable and textile systems, articulation is a matter of simultaneously arranging geometry and structure. The notion of a "deep surface" provides depth in a tension-active system in the arrangement of multiple membrane layers, with a resulting capacity for sophistication in the modulation of climatic performance. This imagines a structural and spatial system comprised of cellular elements of varying depth maintaining, through interdependent tensioned elements, structural continuity.  To manage the formulation of such complex integrated systems, the design process must work first with geometric instances guided by the rules of surfaces formed by the application of tension, and second, in a generative manner functioning rapidly; this being captured within the initial stages of architectural design formulation. Specifically, this methodology works with a constrained computational design environment that reflects the basic behaviors and rules of tension-active forms.*

*It will be described in this paper how the use of particle systems, particularly the use of particles and springs, can define such a constrained design space while maintaining a generative and iterative framework. Specific knowledge of the parameters of springs and functioning of the solver for tension equilibrium forms, in both particle systems and engineered-based solvers provides a coherent bridge to further analysis and form specification. This produces an informed computational process relevant to architectural design, built upon basic principles of material behavior and logics for computational form-finding and analysis. These relationships in process and physical structural behavior will be discussed in this paper, and shown how such knowledge and techniques can allow for the pursuit of advanced geometries within the highly complex structural and spatial parameters of multi-layered tension-active systems. This will be exemplified in both computational experiments and fabrication of physical prototypes of varying scale.*

# 1 Introduction

Particle systems are being commonly used as a computational form-making device in architectural design. In principle, they function by organizing geometry through the application and negotiation of various forces: gravity, magnetism, collision, drag, and springs. As such, they enable the rapid visualization of various physically-based organizational behaviors. They operate on low-level, idealized principles regarding these types of forces, allowing efficiency in computational effort, even in the cases of large datasets. Looking specifically at the use of spring force, based on the principle of Hooke's law of linear elasticity, a particle system can serve as a general form-finding tool for tensioned anticlastic surfaces. An anticlastic surface is defined by the conditions where at any point on the surface the principles curvatures are opposing, one being concave and the other convex, as shown in Figure 1. When defined by tension forces, the opposing curvature produces structural stability and stiffness in the surface.

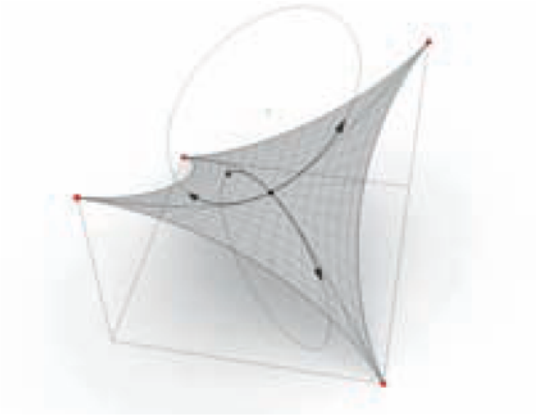


Figure 1: Definition of an anticlastic surface where the primary curvatures are opposite, shown on a typical "saddle" form. If the curvatures are opposite and equal then the resulting Gaussian curvature is 0.

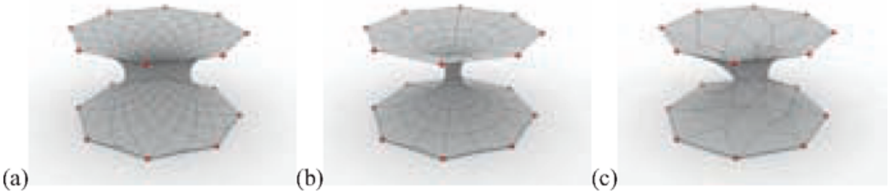


Figure 2: Geometries generated from various springs topologies: (a) dia-grid, (b) quad-grid, and (c) random arrangement. All values are the same for the springs and they always connect to form a "cylinder"; only the pattern in which springs are connected varies.

The topology is critical with springs in defining anticlastic tensioned surfaces, as each spring defines a single vector and a magnitude of force in space. The spring's dimension depends upon the forces acting upon it. A rearrangement of springs will result in a rearrangement of forces, and thus a rearrangement of geometric position. This is shown in Figure 2 where the boundary conditions remain the same, but the topology from one to the next changes. Flexibility in deriving the topology of spring elements, driven by algorithmic procedures to dictate how the springs associate with one another, allows for an advanced formal complexity and varying types of structural systems to be formulated. [Kilian and Ochsendorf 2005] With this computational design environment, rapid generation, variation, and iteration can be focused on investigation of integrated complex geometries and structural systems within the vocabulary of anticlastic surfaces.

Examining the fundamental parameters of springs regarding topology, spring length (termed *rest length*), and stiffness (sometimes termed *strength*), this general method for the form-finding of doubly-curved surfaces can be focused more acutely to producing a design space that respects the material behavior for pre-stressed textile and cable-net structures, namely following the formation of a more specific breed of anticlastic surfaces – those with a constant mean curvature tending towards zero. The *design space* here is one where the scope of geometries generated follows a particular range of values related to parameters of a particle system.

An individual spring can exhibit tension or compression force based upon its properties and the forces exerted upon it. This is described in Hooke's law by the following equation, where *F* is the resultant force, *k* is a spring constant, $x_0$ is the length of the spring at an equilibrium state, and *x* is the degree of displacement from its equilibrium position:

$$F_s = -k(x - x_0)$$

The resultant force if negative describes a spring exhibiting pulling force (tension), and if positive, the spring elicits a pushing force (compression). This presents a simple understanding of the amount of force exerted by a spring as other forces act upon it. Within a network of other spring forces (the topology), this depicts the local types and magnitudes of structural interactions amongst neighboring elements. The overall visualization of relational forces within the network produces a general notion of the force distribution and its degrees and locations of linearity and non-linearity; whether they follow a uniform or non-uniform distribution of force. Where rest length defines the equilibrium state, or $x_0$, and the spring strength defines *k*, these variables provide the vehicle to directly exert or inhibit force within a network of springs. As force is directly being modeled within the particle system environment, a change in spring topology will of course induce a shift in the characteristics of the overall force behavior, and resulting geometric definition.

The ability to localize control over force characteristics, and drastically shift the algorithms which define spring topology within the form generation process sets this particle system based method apart from design processes which utilize engineered-oriented software, or finite element based methods, for tension-active structures. While there may be a freedom in searching a broad design space for such a structural system using particle systems, there is still a necessity to examine the results in a more precise manner. Computational form-finding for pre-stressed lightweight cable and textile structures is done primarily via three different methods: Force Density, Dynamic Relaxation and Modified Stiffness. [Moncrieff 2005] The difference in "pre-planning" between particle systems and engineering-oriented methods is significant. But, constraining particular aspects of the spring-based process, most notably the topology, can allow for a particular information alignment, producing a fluid bridge between the differing computational methods. Such a bridge to examine the relation of form, force and geometric organization, has particular relevance in the early design phase where variability, rapid generation, and viability is most critical.

Membrane structures have continued to evolve in terms of expanding scale and techniques for transformability. The intimate interrelation of materiality and structure, as a continuous surface with uniform stress, directs such systems to produce, for the most part, homogeneous spaces. Utilizing particle systems and springs within the early stages of design formation allows for advanced geometric complexities to be pursued while providing depictions of the inherent structural dynamics, giving a relative accuracy towards the material structures in which it is describing. This computational approach supplies an avenue in which integrated tensioned textile and cable-net structures can be developed through a cellular-based method, articulating structure as a series of interconnected, and geometrically differentiated components which accumulatively define a system of *deep (multi-layered) surfaces*. In a cell-based approach, the varying of topology and geometric description between each cell enables localized differentiation eliciting, within a structurally continuous system, a materially and spatially heterogeneous architecture. A heterogeneous architecture is one where the conditions of environment (modulation of light, climate, dimension, etc.) amongst others can be varied by the articulation of a continuous system.

## 2 Topology – Material Distinction in Computation

In pre-stressed structures, cable-nets (or referred to computationally, in this paper, as "meshes") and tensioned textiles or membranes (referred to as "surfaces") are two systems that behave quite differently. Both sit within the family of non-rigid spatial structures and depend upon curvature to gain structural stiffness in the overall system. [Bechthold 2008] The major difference is in how closely that curvature tracks towards the definition of a minimal surface. A minimal surface is a surface, given a particular edge profile, which has, at any point, principle curvatures which are equal and opposite. [Pottman 2007] Such a condition defines a constant mean curvature of zero, and as a structural surface, contains forces which are uniform in all directions – the ideal parameters for pre-stressed a textile (membrane) structure. Membrane surfaces do not need to match this idealized definition to be structurally sound. But, as they move away from conditions of zero mean curvature, the amount of pre-stress in the warp and weft directions (the primary longitudinal and latitudinal axes at which the threads of the textile travel) becomes non-uniform. This produces a condition where additional loading will cause further uneven distribution of stresses. [Lewis 2003]

Tensioned cable-nets, on the other hand, have much more freedom in the explicit geometric arrangement of elements while following the rule for curvature with equal and opposite principle radii. In the ability to directly affect the geometry (more precisely – the topology), conditions of varying local curvature can be developed. There is still the relationship between degree of anticlastic curvature and structural stability – the flatter the form, the more stress necessary to gain stiffness, and the less efficient the solution. But, where the mesh clearly describes the paths at which the force will flow, there is more opportunity to investigate the relation of form, force-flow, and topological arrangement, on a micro and macro scale. [Otto 1975] Because of the uniform continuity and relative inelasticity of textile materials (fabrics and foils) used in pre-stressed structures, such local manipulation in curvature, and consequently in the material, causes peaks in stress values and uneven "ageing" of the material. In cable-nets, force is resolved at the node between the cables which define direction and degree of force. In a membrane, force is constantly trying to distribute across the entire surface. It is these critical distinctions that are necessary to embed into the particle system environment to elicit a form-finding process which produces geometry and force information akin to the effective functioning of the material structures of cable-nets and membranes.

This distinction between mesh and surface in the computational particle system environment can only exist notionally. That is, the spring, a representation of a force vector, can work as a line in a linear array, a mesh in a two-dimensional array, or a spatial network in a three-dimensional array. It cannot act as a surface with an infinite definition of points and vectors in space within the boundaries of such surface. In this particular case, the distinction of what is envisioned as a tensioned "mesh" or a tensioned "surface" comes down to the articulation of the topology of each continuous network of springs. As described before the topology of a cable-net is quite open ended. Therefore, the mesh is an entity at which the topology can vary greatly and represents a very direct depiction and alignment of the location of material and the flow of force. The variation and possible topological complexity with a "mesh" in tension equilibrium is shown in Figure 3.

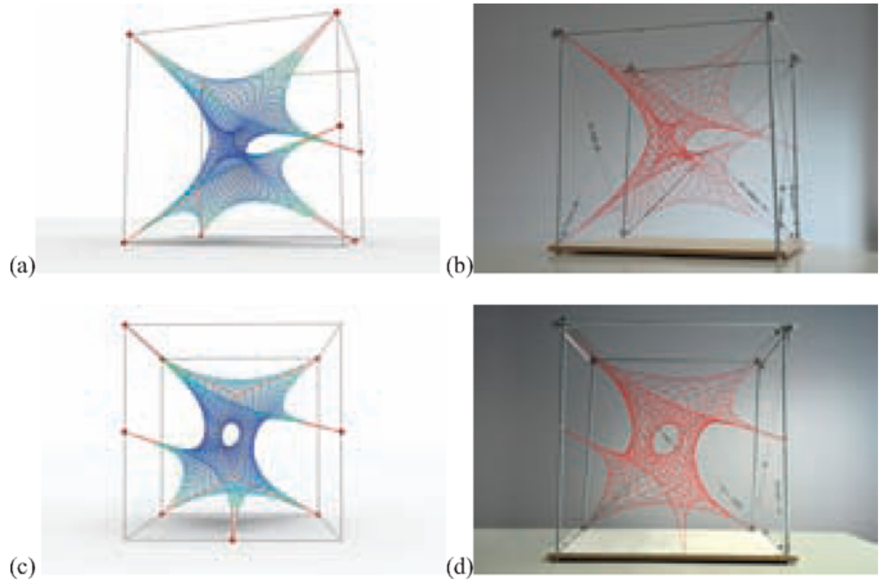


Figure 3: Tensioned mesh of continuous complex topology. Left: Computational model from particle system simulation with visualization of force distribution. Right: Scale model of tensioned mesh with information for fabrication provided by computational model.

The membrane is a more specific condition for relating topology with force and materiality. The specific topology is also a repercussion of the way in which geometry is managed in engineering-based methods for the form-finding of tensioned surfaces. Primarily, to follow the trajectories of the warp and weft direction of the materiality, a specific "quad" topology of springs is necessary, as shown in Figure 4. This is traditionally consistent with the way in which tensioned membrane surfaces are analyzed. Specifically, in plug-ins to Rhino such as RhinoMembrane and MPanel, the input geometry is a quad-based mesh. The resolution of the mesh is something that is variable and can only be resolved with the balance of geometry and curvature. These particularities will be explained further in the next section when describing the methods for deriving specific classes of forms within the family of anticlastic surfaces.

While the entities intended to define surfaces are constrained rather considerably, those which are classified as meshes can vary greatly both globally amongst other mesh entities (a particular collection of connected springs defining an entity) and locally within a single mesh. Springs which define a surface provides a bi-direction control, following the U and V direction of such an implied surface. Springs which define a mesh allow for multi-axial assembly and manipulation of geometry. This opens up the opportunity to pursue complex global geometric arrangements, and utilize local manipulations in topology and spring parameters to articulate such forms. In particular, this sets up a situation in which a series of layered or offset meshes can be organized.

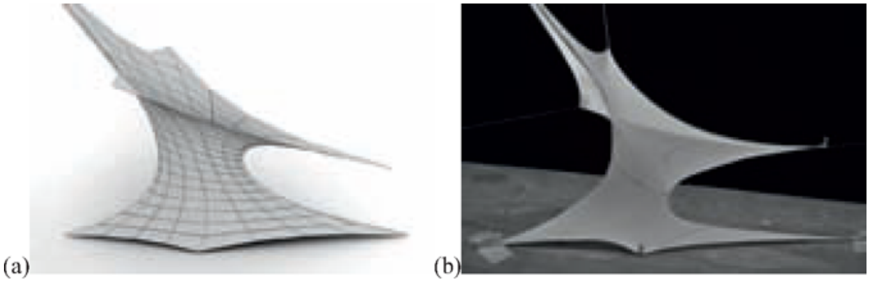


Figure 4: Tensioned “cylindrical” surface. Left: Tension equilibrium surface generated with particle system. Right: Fabrication of tensioned cylinder using elastic material where information for cutting pattern was extracted from the computational model.

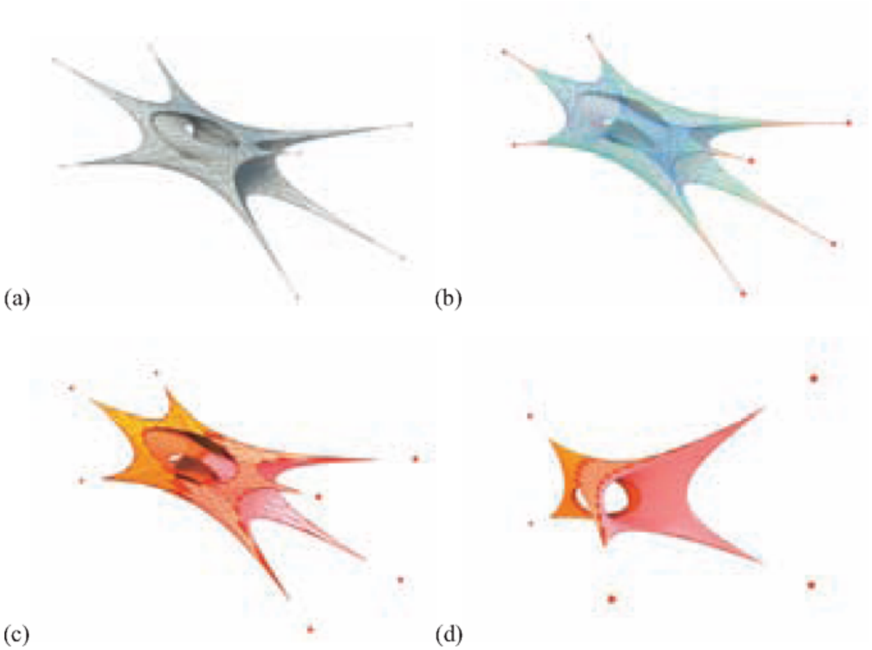


Figure 5: (a) Varied topology and (b) range of force characteristics to develop highly complex geometries with multi-layered “deep surface” offset conditions. The overall form is constructed of 4 “cylindrical” mesh entities, shown in (c) and (d).

The relationship between discreet geometry, curvature and offsets is often extremely challenging to solve. In the case of particle systems, there is the opportunity to use a procedural negotiation of force and topology to allow for such complex offsets to be determined. With cable-nets, as the topology can vary in the computational meshes, this variability can be taken most advantage of in helping to match particular criteria that may be established for the offset conditions, such as depth or alignment between cells of the offset meshes. Studies in complex geometries and variable spring characteristics to produce offset surfaces are shown in Figure 5.

## 3 From Anticlastic to Minimal Surface

For a generative design process to function efficiently, the design space in which it produces, however broad it may be, should logically work within a particular realm of materiality and structural behavior. If the computational process is not initially driven by material specificity, it should allow for such specificity to arise from basic principles regarding force, elasticity, and mass, to name a few. In the case of tension-active systems, materiality is a critical consideration even within the initial stages of design formation. Form, as an active system not a static object, is realized through the negotiation of forces within a material. A change in material will change the dynamics of the forces producing shifts in the overall form. This entails that a generative process for such a system must be able to recognize basic principles for materiality and its relationship to characteristics of force.

One method in which the materiality of a surface condition is embedded in the particle system is by organizing springs in quad-based topologies to signify the definition of a surface. The quad topology recognizes, generally, the warp and weft direction of a material – the principal axes in which force is analyzed. This also supplies a logical bridge to the methods in which membranes are form-found in finite-element based solvers. This alignment is critical in two ways. First, this allows for an iterative verification to take place that the forms generated within the particle system environment are relatively valid. Secondly, this allows for, at the stages of further development and design specificity, the geometric information to flow smoothly into environments which can perform further precise analysis.

The key factors for defining a tension-active surface are the amount of force at the “anchor points” and along the edges of the surface. As a result of “form-finding” the geometric definition of the surface is an emergent condition based primarily on these two inputs. In the particle system, the springs which are set at the boundaries of a surface are defined uniquely so that they can withstand additional amounts of tension force. This is controlled in the parameter of *spring strength*. Depending on the complexity of the topology, the approximate length of the boundary springs may not be known until the solution is generated. Comparatively, finite-element solvers allow for two conditions in terms of the edges – to form-find the edge based on a pre-set desired length or to input the exact edge profile to which the tensioned surface should match. In either case, a certain amount of pre-planning is necessary. In the first case, the geometry of the edge will occur in the “form-finding” process. In both cases, a certain understanding of the amount of curvature in the edge condition is necessary. The amount of curvature along a single edge is defined by the degree of force in the surface, which can be roughly calculated by measuring the distance between the fixed nodes and the radius of the edge curvature. [Lienhard 2009] Knowing this information geometrically can allow for the force parameters at the edge to be produced, as shown in Figure 6.

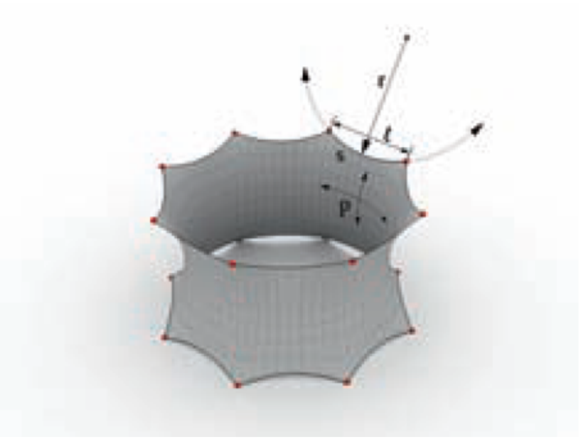


Figure 6: Approximating method for finite-element calculation of tensioned surface (form-finding done in Rhinomembrane). The calculation is predicated in the geometric description of the 'length' value between fixed anchor points. Pre-stress in the cable can be determined through the following two equations, where *s* is the desired cable pre-stress value, and *p* is the pre-stress in the membrane which can be, by default, considered as 1:

$$r = 1.3 \times l$$

$$s = p \times r$$

In finite-element methods, such pre-defined geometric information is generally necessary to translate the geometric problem into values of pre-stress. The process is oriented to finding a specific solution for the surface given specific inputs for the boundary conditions. In a lower-level generative process, it is the intention to develop the surface and the boundary conditions simultaneously, understanding that there are hierarchical relationships between the two. Nonetheless, it is necessary to understand this basic mathematical rule so that the particle system geometry is assessed in terms of real force units, of which those values can be fed into the finite-element analysis for further verification and specification of the overall geometric and structural arrangement. A comparison between the parameters of the boundary conditions and resulting geometries amongst different solvers is shown in Figure 7.

With the internal arrangement of forces within the surface, there is some possibility to influence the geometric formation of the surface, though the manipulation of stresses will always occur along the warp and weft, or U and V, directions. Within the particle system, the comparative parameters of spring rest length and strength in the springs that define the U and V directionality of the surface define the general ratio between stresses. This is most easily controlled if the numerical array of springs is uniform in both directions. It is often not a 1:1 relationship, where the springs attached directly to the boundaries (the V direction) will have the most force moving through them. The opposing springs (the U direction) must have more significant strength values so as to counteract those forces, and return the anticlastic curvatures to the overall form. The actual stresses will only be realized once the external forces are applied – in the boundary conditions comprised of the edges and anchor points. The numbers of springs which define the surface have a significant influence on the forms generated. This is in distinct contrast to finite-element methods where any mesh resolution will produce the same result given the same input parameters. Depending on the spring parameters, the more springs within a surface, the more ability there is to overwhelm the external forces at the boundaries. This will be examined later as a critical consideration in the calibration of integrated tensioned surfaces and layered offset meshes.

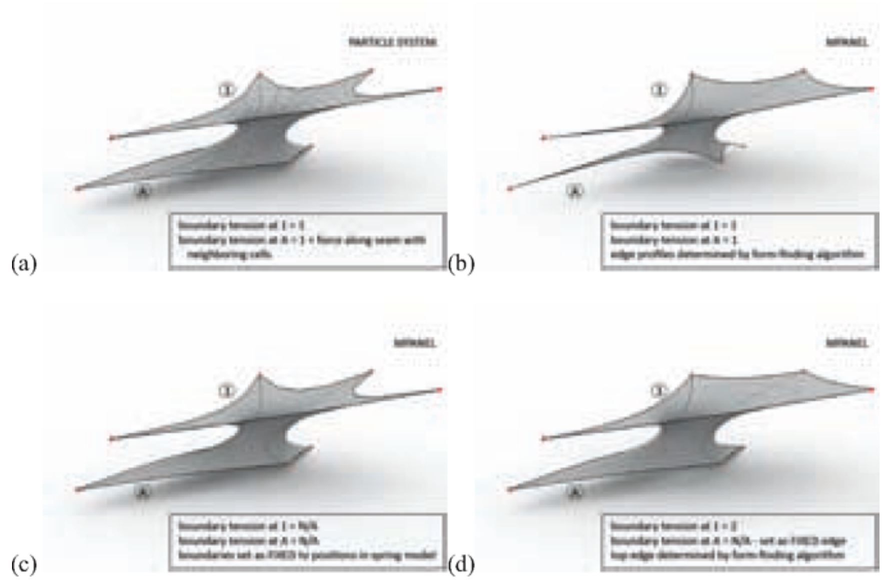

PARTICLE SYSTEM
1
A
boundary tension at 1 = 1
boundary tension at A = 1 + force along seam with neighboring cells
(a)
MPANEL
1
A
boundary tension at 1 = 1
boundary tension at A = 1
edge profiles determined by form-finding algorithm
(b)
MPANEL
1
A
boundary tension at 1 = N/A
boundary tension at A = N/A
boundaries set as FIXED by positions in spring model
(c)
MPANEL
1
A
boundary tension at 1 = 2
boundary tension at A = N/A - set as FIXED edge
top edge determined by form-finding algorithm
(d)

Figure 7: Form-finding with variation in tension of boundary conditions generated with particle system (a) or MPanel (b) – (d).

The internal stresses within a surface can be manipulated in finite-element form-finding processes as well, though it is primarily done numerical rather than topologically. Figure 8 displays the manipulation of stress parameters in different form-finding environments in an attempt to match and define the validity of the results from the particle system model. The values of warp and weft can be varied to increase the non-uniformity of the forces across the surface. This is a linear change though across the entire surface. An exception to this case is in Rhinomembrane, for example, where a quadratic change in stresses can be applied. This is specifically available for the case of "conical" membranes which often exhibit flat areas at the outer edges of the surface, when under uniform stress. With a change in the stress ratio moving towards the tip of the cone, a steeper incline can be achieved. With either springs or finite-element solvers, the variability of this method for stress distribution should be tempered with an understanding how the resulting material form will function. As mentioned previously, forms with non-uniform behavior do not typically make for effective structures. The general desire is to still constrain the methods to develop surface which approach a zero mean curvature.

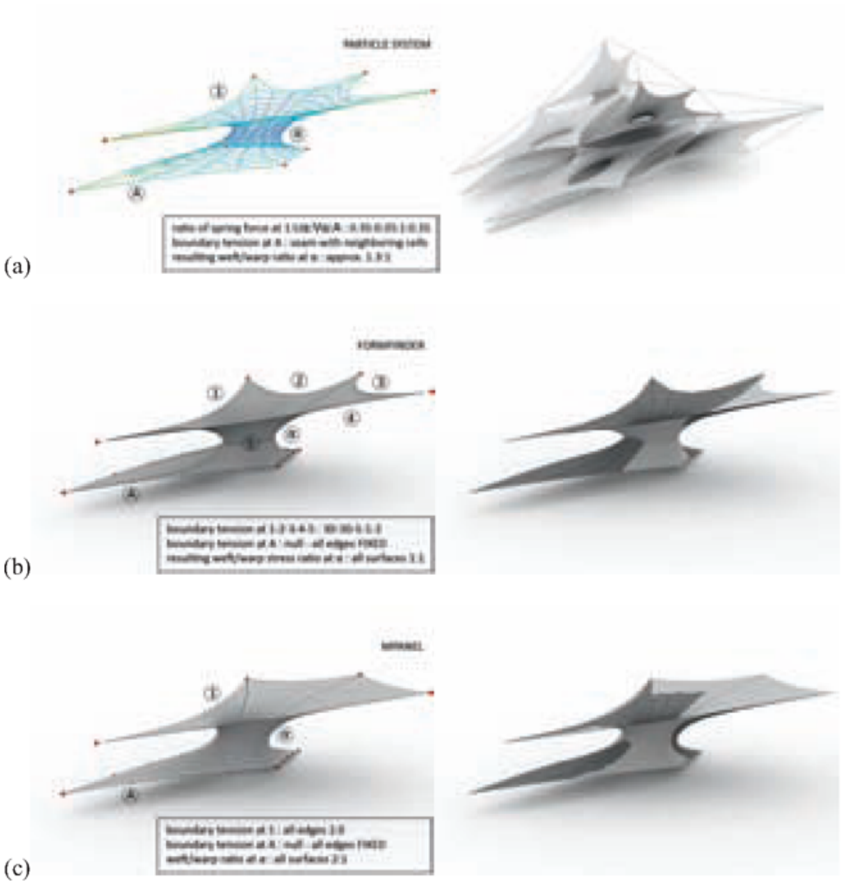

PARTICLE SYSTEM
1
A
(a)
FORMFINDER
1
2
3
4
A
boundary tension at A : null - all edges FIXED
(b)
MPANEL
1
A
boundary tension at A : null - all edges FIXED
(c)

Figure 8: Comparison of form-finding results between (a) particle system, (b) Form-Finder, and (c) MPanel. The geometry for the initial form is generated in the particle system (a) through the negotiation of neighboring tension-active surfaces and meshes. The resulting force diagram is shown at left. The forms in (b) and (c) are generated in isolation, attempting to match the same surface configuration and boundary condition through the manipulation of internal stress ratios and variation in boundary tensions. The comparison of surfaces (in light grey) to the initial spring model (in dark grey) is shown at right, displaying the discrepancies between the results of the form-finding methods.

It is possible that a self-regulating algorithm could be applied within the particle system environment to check mean curvatures within a surface and adjust accordingly the local spring values so that mean curvatures values could be pushed towards zero. Unlike finite element based methods for form-finding membrane surfaces, the particle system does not have a built-in mechanism for achieving minimal surfaces. But, it should be considered, though, in the application of such a constraining algorithm that within the particle system environment, there will be other criteria pressuring the formation of the spring-based surfaces. The most significant errors in the particle system geometry, driving it away from minimal surface formation, arise from the spring's ability to infinitely stretch, and that it does so with only linear behavior. This produces distortion in the surfaces and subsequent areas with high degrees of non-uniform stress, often at the anchor points where the highest degree of forces converge.  As shown in the overlaid renderings of surface geometries in Figure 8, the moment at the extreme lower left point where there is the highest concentration of force, the form-finding solvers in MPanel and Formfinder and not able to match the degree of distortion in the spring-based surface.

## 4 Advanced Complexity – the *Deep Surface*

The notion of a deep surface, in relation to tension-active systems, is a re-configuration of what is normally a single surface structure – textile sheets tensioned with boundary cables and anchoring points – to a scenario where the textile or the "surface" runs amongst a series of independent surfaces between two distinct tensioned cable-net meshes, as depicted in Figure 9. A global "surface" is implied by the meshes, while the textile surfaces run perpendicular – between the two meshes. Cable-net structures are often "componentized" so that the system can provide enclosure. They do so with elements that are relatively structurally inert – sitting within the cells of the mesh rather than adding resistance to the forces in the overall structure. By re-orienting the cells from running parallel to the mesh to working between meshes, a depth of varying dimension is accomplished. Such depth can allow for spatial characteristics to be further articulated – apertures for focused or dispersed light transmission, openings related to curvature for controlling air flow and ventilation, variable depths to control heat absorption and dissipation, exterior curvatures to respect water run-off being distinctly controlled in comparison to interior curvatures shaped for particular functionality and scale.

This type of cellular-based system is unique in that the forces are significantly influential within and across the hierarchies of the system. While there is some autonomy within the individual cell or component – the textile will exhibit force upon the cable-net which will then have repercussions across the entire mesh and into other cells. The calibration between these conditions is as much a design instrument as it is a negotiation of the structural material dynamics. The question lays in the balance of how much force is in the meshes to properly tension the surfaces that lie between them, what form do the arrangement of forces define, and how does that respect material characteristics and the various levels of functioning to which the entire system is prescribed. Where all the elements of the system will want to fall into their most minimal energy, the computational process for both initial design generation and further analytical specification must recognize these deep interrelations of form, force, materiality, and functioning.

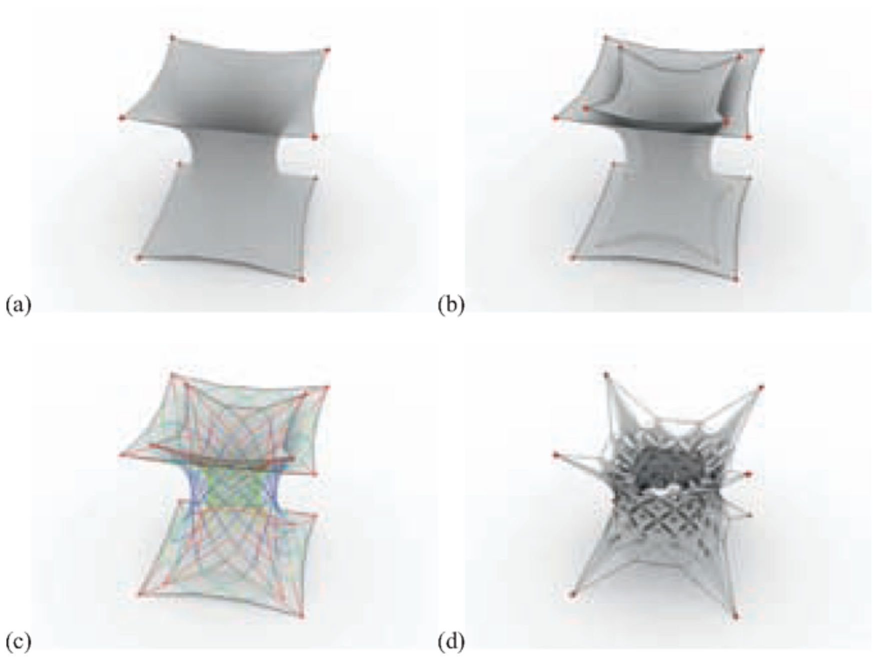


Figure 9: This sequence shows the distinction between: (a) single surface system, (b) multi-layer surface configuration, (c) multi-layer open mesh configuration, and (d) cellular "Deep Surface" constructed of integrated tensioned mesh and surface elements. The tensile action of the cells against the 2 open meshes is evident in the drastic change in morphology between (c) and (d).

In a particle system, the primary input is topology. Since most characteristics of the particle and spring elements, which constitute the topology, can only be realized *in* the process of form-finding, the presets for spring rest length and strength are something to be adjusted during the process of determining a force equilibrium state. For instance, knowing the amount of force at which a particular cellular component will pull upon the mesh and what form it will take cannot be determined in advance. Once the general forces within the mesh are realized, then the forces within the components have to be adjusted to either pull more on the mesh, to contract the overall form, or the springs within the component have to be weakened to allow expansion between the two meshes, as shown in Figure 11(a) and 11(b). This is a fundamental consideration for the materiality and overall structural significance of the system. As form with tension-active systems is derived by force and topology, that information, provided approximately with the particle system environment, must be carried over into subsequent analysis and portrayed so that such dynamics can be realized in the final material form.

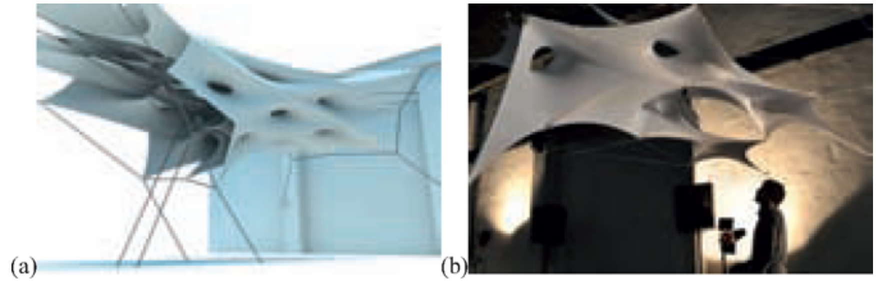


Figure 10: “Deep Surface” Prototype. Left: Computational model of integrated tensioned cable and textile system generated with particle system based form-finding process. Right: Physical prototype, approx. 7m long, constructed of “cylindrical” cells of elastic Lycra fabric interconnected with two “cylindrical” meshes consisting of 2mm steel cable.

# 5 Conclusion

Particle systems sit some distance away from being a solver which specifies the complex intricacies of structure, fabrication, detailing and assembly for lightweight tension-active structures. It has been shown that with a certain methodology, they can provide a unique design avenue for formulating complex structural arrangements. But, this sits within a larger design process framework of exchange and development between a rapid generative design environment and a precise analytical production environment (or multiple environments for design development). For such early simulation-based approximating methods to be useful, logics and fundamental principles must be drawn from other stages of design and embedded into the principles of the early generative processes. [Attar, et al. 2009] Some of the engineering of tensioned cable-net and membrane structures have been briefly discussed in this paper. This was, by no means, an exhaustive explanation – as it was not intended to be. The use of particle systems or any material behavior based method can serve ideally to transform the initial abstract design concepts into measurable formal systems. If these initial formal systems sit within a material and structural constraint space, then the integration with and feedback from further specification works in a logical and fluid manner. Computational processes in design function ideally to *generate* information and advance specificity through iterative means. The *means*, in this case, are proposed as a collection of algorithms to simulate, at various levels of precision and abstraction, the behavior of material under differing structural forces.

This paper has discussed a generative design process dealing primarily with the pre-stressed aspect of force dynamic structures. Further significant analysis is necessary to understand the additional loading and the potential implications of non-linear and non-uniform stresses as the additional differential loads are induced. To look at this in detail, such loads can be, arguably, dealt with by adjusting the pre-stressing – but ideally the overall form should be reconfigured to account for loads beyond the internal tension stress of the system. Such an approach demands a feedback mechanism between initial design formulation and precise engineering analysis on a definite form and its related structural parameters. The method elaborated in this paper begins to establish such a link and alignment of geometric information between environments of design generation and analysis. The particular challenge is in making such a process successfully iterative so as to supply the precise analysis to proper adjustments of the geometry.

All of this proposes the make up for an informed, constrained design space based on fundamental material behaviors, physical principles, and computational linkages between environments of form generation and structural analysis. What has not been additionally accounted for is the other aspects of design, in this case the multiple capacities possible with a “deep surface” lightweight structure, which sees the overall form as a series of interacting systems and functions. To test certain functional capabilities of the system, a geometric form and behavior has to be produced. When the discussion of design is beyond pure form-finding of structural ideals, then the necessity for the study of a multitude of geometric, spatially articulated, and environmentally functional systems is increasingly essential. The continuation of this research is focusing on the formulation of a collection of computational methods to act as a meta-process which can manage the measure and integration of such capacities. The computational framework intends to avoid the common approach of generalizing criteria to maintain expediency but rather concentrate on a process by which the production and specification of information can gradate with the articulation of form. Such a process still clearly exists within the realm of architecture but functions upon an enhanced integration of design-specific engineering principles and methods.

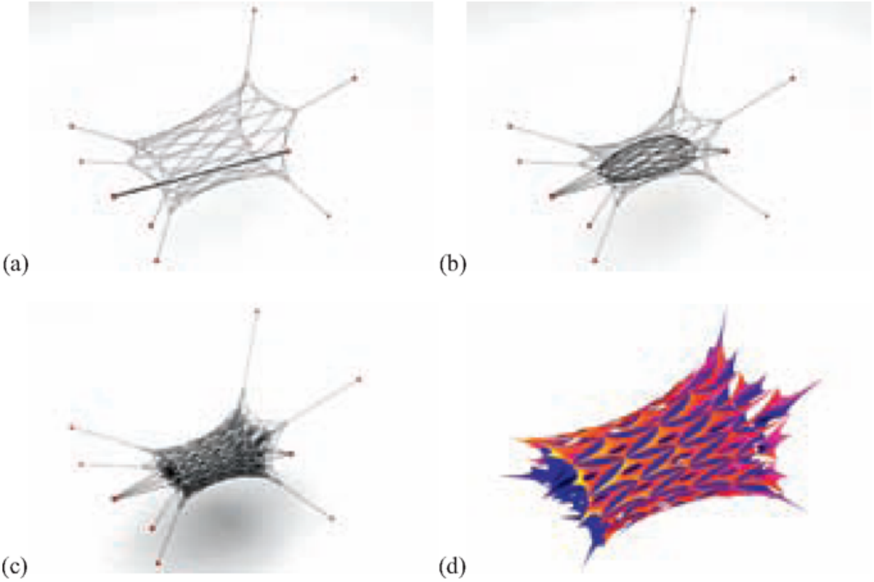


Figure 11: Sequence of form-finding for integrated cable mesh and textile surface. (a) Isolated tensioned meshes. (b) Form of meshes when tension from textile surfaces (not shown) is introduced. (c) Rendering of both cable meshes and textile surfaces. (d) Approximated analysis for solar incidence on textile surfaces.

## References

Attar, Ramtin, et al. "Physics-Based Generative Design." *Joining Languages, Cultures and Visions: CAADFutures 2009.* Montreal, 2009. 231-244.

Bechthold, Martin. "Lightweight design - new directions." In *Innovative Surface Structures: Technologies and Applications*, by Martin Bechthold, 2-15. Abingdon: Taylor & Francis, 2008.

Kilian, Axel, and John Ochsendorf. "Particle-Spring Systems for Structural Form Finding." *Journal of the International Association for Shell and Spatial Structures*, 2005.

Lewis, W.J. *Tension Structures: Form and Behavior.* London: Thomas Telford Publishing, 2003.

Lienhard, Julian. *Lecture: Fundamentals of Membranes and FE Calculation* (10. December 2009).

Moncrieff, Erik. "Systems for Lightweight Structure Design: the State-of-the-Art and Current Developments." In *Computation Methods in Applied Sciences, Volume 3*, 17-28. Springer Netherlands, 2005.

Otto, Frei. *IL 8: Nets in Nature and Technics.* Stuttgart: Institute for Lightweight Structures, 1975.

Pottman, Helmut. "Curvature of Surfaces." In *Architectural Geometry*, by Helmut Pottman, Andreas Asperl, Michael Hofer and Axel Killian, 487-501. Exton: Bentley Institute Press, 2007.

# Statics-Sensitive Layout of Planar Quadrilateral Meshes

**Alexander Schiftner**

TU Wien, Evolute

**Jonathan Balzer**

KAUST

**Abstract.** *Rationalization of architectural freeform structures using planar quadrilateral (PQ) meshes has received rising interest in recent years, facilitated mainly by the introduction of algorithms which are capable of generating such. These algorithms involve an optimization which is, up to now, motivated purely geometrically and accounts for aspects of feasibility, visual appearance, and approximation of the architectural design. Practitioners would wish to add stiffness to the objectives of the layout process. This paper presents a simple but effective statics-aware initialization procedure for the layout of PQ meshes approximating a given freeform surface. We focus on the class of surface structures with membrane-like load bearing behavior, quite regularly encountered in architecture. By compliance analysis of two representative examples, we demonstrate that this specific type of initialization has indeed favorable impact on the mechanical properties of the final PQ mesh.*

## 1 Introduction

### 1.1 Motivation

A common approach for rationalizing architectural freeform surfaces using a collection of flat panels is by means of triangle meshes, like the example in Figure 1. Recent research opened up the possibility of approximating architectural freeform surfaces by PQ meshes that exhibit sufficient quality for the architectural application in terms of visual appearance and practical size requirements. Besides the observation that PQ meshes admit torsion-free nodes and may therefore be realized more light-weight, little has been said so far about their load-bearing behavior. The aim of this contribution is to close this gap.

### 1.2 Outline of the Proposed Method

It is well known that the geometric constraints on the prospective layouts of a PQ mesh are substantially more profound than those of a triangular one. We thus believe that the yet complex system of different objectives could well cloud if not entirely

Figure 1: Roof over the Great Courtyard of the British Museum, London, in classical realization by triangular steel/glass panels (Source: http://carpelondinium.wordpress.com). We present a rationalization by a planar quadrilateral mesh with optimized statics.

compensate the effect of an additional stiffness-enforcing component. This is even more true as inconsistencies among the objectives must be expected. Luckily, the present work founds on an algorithm which operates in the spirit of classical perturbation approaches insofar that, roughly speaking, the non-planar quad mesh is shifted towards a proximate local minimizer of improved planarity. It is therefore reasonable to assume that the planarization result will, to some extent, inherit the mechanical properties of the initial mesh. This immediately implies the following two-step strategy:

1. Instead of optimizing the network of rods as part of the remeshing step explained next, we treat the given reference geometry as a continuum. Although the "discrete" shape of the structure (e.g. the sizing or density of rods) is discarded early in process, this presents the only possibility to perform the desired decoupling of statical optimization and mesh layout. The outcome of this "homogenization" is a so-called *shell*, which is, loosely speaking, a regular surface endowed with a selection of mechanical properties. We compute its strain state for the case that only gravitational forces induced by self-weight act upon the structure. After subsequent principal component analysis, a vector field of maximal loading directions can be passed to the meshing module.

2. We lay out a (quasi)-planar quad mesh which is aligned with the precomputed guidance vector field. Regular quad meshes approximating a freeform surface are characterized by two families of edge polygons which may be considered as discrete versions of smooth curves on the underlying

freeform surface. Requiring the quads to be planar poses a constraint on the two families of edge polygons respectively curves. They have to resemble a so-called *conjugate curve network*, see [Liu et al. 2006] for a definition. We use the guidance vector field resulting from the mechanical preprocessing phase as an input for computing a function defined on the given freeform surface, whose level set curves are aligned closely with the vector field. Furthermore, we compute a second function whose level set curves are conjugate to the level set curves of the first function. Having computed such functions, we may extract a quad-dominant mesh (a mesh consisting mainly of quads) from them, whose faces are not exactly planar but close to. Subsequent optimization of these meshes towards planarity of faces can be expected to be successful as we illustrate by examples. Furthermore, the resulting PQ meshes are likely to exhibit improved load bearing behavior compared to other PQ meshes approximating the same freeform surface.

### 1.3 Related Work

*Mechanics of Thin-Shell Structures.* Shell structures and their accurate as well as computationally tractable modeling remain an active area of research. The relevant sources are far too numerous to provide an exhaustive list here. Instead, we point the reader to the pioneering works by Koiter [Koiter 1970], Naghdi [Naghdi 1972], Reisner [Reissner 1947], Kirchhoff, and Love [Love 1888] as initial references. Recently, there has been an increased interest in thin shells in the graphics and computational geometry communities, mainly motivated by the aim to create physically sound deformation models for surface meshes, see e.g. [Grinspun et al. 2003] and in particular the survey [Botsch and Sorkine 2008]. The paper [Winslow et al. 2008] is probably the one most closely related to ours. The authors point out the connection between macroscopic grids in architectural geometry and – at a microscopic scale – fibres in a reinforced polymer composite. As the present contribution, they rely on a homogenization approach. But although, for static computations, they treat quadrilateral cells of the target grid as plates, i.e. planar shells, they do not seem to enforce a planarity condition in their multi-criterial grid optimization framework.

*Planar Quadrilateral Meshes.* Early applications of PQ meshes for architectural purposes can be seen in projects by Schlaich-Bergermann. These PQ meshes are discrete versions of translational, rotational, and a few further types of surfaces, refer to [Glymph et al. 2002]. More recent research initiated by [Liu et al. 2006] deals with the possibility of approximating general freeform surfaces by PQ meshes. The authors introduce an optimization method which perturbs the vertex positions of an initial quad mesh such that the faces become planar (so-called *planarization*). In order to generate a suitable initial quad mesh, we use a surface parametrization or level set method. Numerous papers have been published on this topic in previous years. The method we use is described in detail in [Zadravec et al. 2010] and closely related to [Kälberer et al. 2007].

## 2 Statics-Sensitive Preprocessing

We will now explain our simple mechanical model of the geometry to be remeshed. Section 2.2 discusses the numerical solution of the resulting partial differential equation (PDE), Section 2.2 how it interacts with the subsequent meshing step. All the while, it is not our intent to estimate the true stress and strain state of the surface under consideration as accurately as possible but rather provide a qualitative answer to the question how the multiply constrained meshing problem can be influenced positively in a statical sense.

### 2.1 Thin-Shell Model

A *shell* is a three-dimensional mechanical structure, but it is slender in the sense that its thickness is small compared to its lateral dimensions. It may be seen as a stack of two-dimensional layers, all distinguished by their distance to the so-called *mid-surface*. We associate with the mid-surface of a shell the given geometry that we want to approximate by a PQ mesh. The underlying theory is well-established and can be reviewed in the survey [Botsch and Sorkine 2008] among numerous other sources. In short: The constitutive equation of shell deformation derives from a simple conservation law. In static equilibrium, the internal elastic energy $E_{\mathrm{el}}$ equals the mechanical work $W$ applied to the body by external forces $\boldsymbol{f}$. The former dissolves into two parts: one which is induced by shearing or stretching the structure in tangential directions (the so-called *membrane energy*) and the other one which is due to bending. In what ratio the two deformation modes appear in a structure depends mainly on its bearing, i.e. the boundary conditions, and sometimes the prevailing load case, see Figure 2. A linearized deformation model, i.e. a vector field $\boldsymbol{v}$ of displacements between the original shell and its deformation, is found by solving the Euler equation of the corresponding energy residual functional $E_{\mathrm{el}} - W$:

$$-\lambda_{\mathrm{m}}\Delta\boldsymbol{v} + \lambda_{\mathrm{b}}\Delta^2\boldsymbol{v} = \boldsymbol{f}. \tag{1}$$

The proportionality constants $\lambda_{\mathrm{m}}$ and $\lambda_{\mathrm{b}}$ are material parameters describing deformation resistance or stiffness with respect to membrane actions respectively bend-

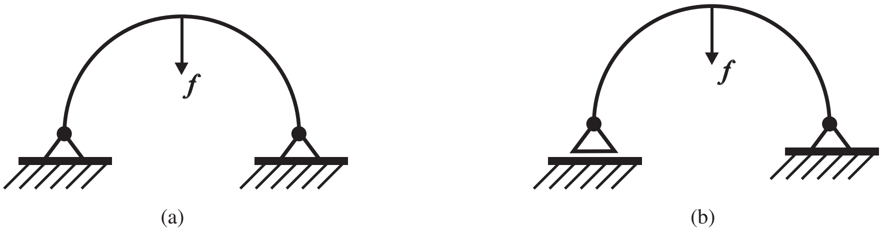


Figure 2: (a) A pre-stressed bearing fosters tangential force flow i.e. bending-stiff membrane behavior. (b) Introduction of a floating bearing significantly reduces the resistance to bending.

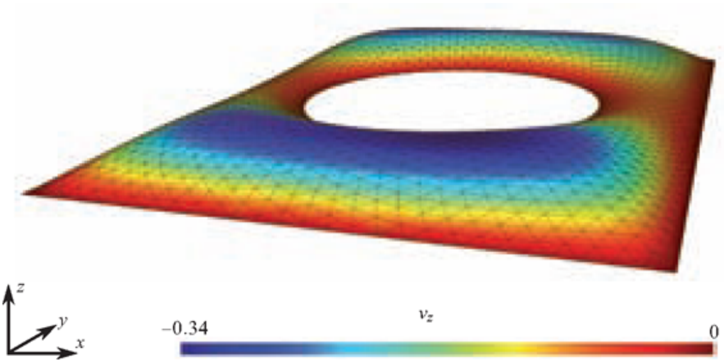


Figure 3: For mechanical preprocessing, the reference geometry is modeled as a membrane shell. The $z$-component $v_z$ of $\boldsymbol{v}$ owing to self-weight is obtained by solving, in this special example, a Dirichlet problem (fixed boundary). The color encodes its distribution over the undeformed surface.

ing. The Laplace and bi-Laplace operator $\Delta$ respectively $\Delta^2$ each represent one of the two deformation modes.

## 2.2 Implementation Issues

*Estimating the Strain State* For a discretization of the reference geometry a manifold mesh is used, i.e. a triangulation without self-intersections, hanging edges, t-junctions, or isolated vertices. This choice is very favorable with regard to the finite-elements (FE) formulation of (1): Locking effects i.e. insufficient convergence rates, typical complications of thin shell numerics, are not to be expected for this particular type of element, cf. [Hughes 2000]. Galerkin projection on its nodal basis situates Equation (1) in the finite-dimensional space of functions whose restriction to each individual triangle is linear. The outcome is a sparse linear system of algebraic equations, which can be solved rather cheaply with the help of standard software, e.g. the CHOLMOD library, featuring supernodal column ordering and sparse Cholesky decomposition [Davis and Hager 2009].

In theory, our method includes bending-dominated deformations, which cause, however, significant numerical complications arising at discretization of the bi-Laplacian $\Delta^2$ in (1). One could be tempted to simply square an approximation of the Laplacian $\Delta$. This practice is somewhat questionable as fourth-order PDEs necessitate designing a set of $C^1$-continuous shape functions, which the piecewise linear basis is clearly not. For a remedy, one would either have to increase the element function order, i.e. perform a $p$-refinement, or rely on different discretizations such as e.g. subdivision surfaces, cf. [Cirak et al. 2000], alternatively B-spline patches like in isogeometric analysis [Hughes et al. 2005]. Realization of any aforementioned strategy is well beyond the scope of the present work. Consequently, we omit the fourth-order term for the time being, also see the concluding remarks in

Section 5.

The current implementation features the possibility to assign either a Dirichlet or Neumann condition to every boundary node. As depicted in Figure 4, the former restricts translational, the latter rotational degrees of freedom. Although in practice, a building is subject to a variety of different forces, we confine ourselves to the case of dead loading[1]. Inclusion of more general scenarios is straightforward. The force field $\boldsymbol{f}$ is concentrated on the vertex set of the mesh by means of a simple mass lumping. Observe that, since in this case the $x$- and $y$-components of $\boldsymbol{f}$ are zero and, by the maximum principle, also those of $\boldsymbol{v}$, the system (1) reduces to a single scalar equation in the $z$-component of the latter.

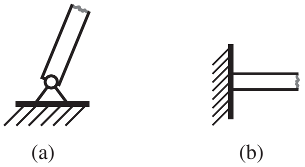


Figure 4: Available boundary conditions: (a) Setting $\boldsymbol{v} = \mathbf{0}$ on the boundary amounts to a fixed but limp bearing. (b) Enforcing $D\boldsymbol{v}\hat{\boldsymbol{o}} = \mathbf{0}$, where $\hat{\boldsymbol{o}}$ is the outer normal of $\Gamma$, suppresses rotations around the boundary curve but does not necessarily limit translational degrees of freedom.

*Principal Strains* Once $\boldsymbol{v}$ has been recovered from (1), we evaluate the so-called *Green Lagrange tensor*

$$\boldsymbol{\varepsilon}_{\mathrm{m}} := \frac{1}{2}(DT^{\top}DT - \mathbf{P}) \tag{2}$$

in a face of the mesh as follows: First, the displacement $\boldsymbol{v}$ is converted into the vector-valued mapping $T$ between deformed and original configuration of the shell. In the discrete setting, a scalar function in the piecewise linear FE space basis is uniquely defined by its values on the node set. It is thus straightforward to compute its gradient, which is constant on each single face, from the values on the three incident vertices. Concatenating the face-based gradients of all components of $T$ then approximates the Jacobian $DT$. All operations are performed with respect to a fixed Eulerian world coordinate system. Hence, the Green-Lagrange tensor $\boldsymbol{\varepsilon}_{\mathrm{m}}$ acquires the shape $3 \times 3$ and a nontrivial kernel, spanned by the current unitary face normal $\hat{\boldsymbol{n}}$. For this particular choice of basis, the projection matrix $\mathbf{P} := \mathbf{I} - \hat{\boldsymbol{n}}\hat{\boldsymbol{n}}^{\top}$ restricted to the tangent space corresponds to the identity there. After putting intermediate results together, it is illustrative to verify that, given a direction $\boldsymbol{t} \in \mathbb{R}^3$, intra-surface length discrepancy, i.e. membrane strain, is quantified via $\boldsymbol{t}^{\top}\boldsymbol{\varepsilon}_{\mathrm{m}}\boldsymbol{t}$.

By eigenvalue decomposition of $\boldsymbol{\varepsilon}_{\mathrm{m}}$, we get the orthonormal and symmetric frame field $(\boldsymbol{e}_{\max}, \boldsymbol{e}_{\min})$. It is ordered and labeled by magnitude of the associated

[1] One usually differentiates between dead and live loads. The former is attributed to pure self-weight. The latter include acceleration forces due to non-stationary winds, earthquakes, or similar as well as time-dependant mass surpluses in the form of snow, rain water, traffic if the structure is e.g. a bridge.

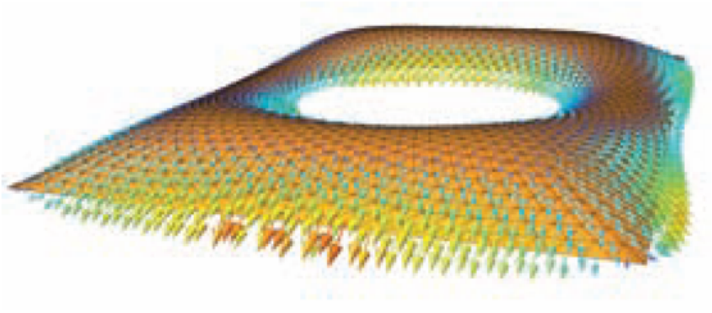


Figure 5: Example from Figure 3 continued: The Green-Lagrange tensor is uniformly positive-definite symmetric and can thus be brought to diagonal form with respect to its orthogonal eigenvector basis. Shear strains vanish in this particular basis so that the directions of minimal respectively maximal normal strain can be determined by sorting of eigenvalues. The latter serve as guidance field during initialization of the actual remeshing routine.

eigenvalues. These are called *principle strains* and equal the maximal respectively minimal absolute normal strain value encountered in a surface point.

A little care has to be taken concerning its orientation: If $\boldsymbol{e}_{\max}$ is an eigenvector of $\boldsymbol{\varepsilon}_{\mathrm{m}}$ corresponding to the eigenvalue greatest in modulus, then so is $-\boldsymbol{e}_{\max}$. A continuous sign distribution can be achieved, though, by aligning the guidance vector field $\boldsymbol{d} : \Gamma \mapsto T\Gamma$ with the projection of displacements $\boldsymbol{v}$ onto the tangent bundle $T\Gamma$ of $\Gamma$:

$$\boldsymbol{d} := \begin{cases} \boldsymbol{e}_{\max} & \text{if } \langle \boldsymbol{e}_{\max}, \mathbf{P}\boldsymbol{v} \rangle > 0, \\ -\boldsymbol{e}_{\max} & \text{else.} \end{cases}$$

Let us point out a significant strength of this specific choice of $\boldsymbol{d}$: Most quadrilateral meshing methods align isoparametric curves with one of the principle curvature directions. In umbilic areas, however, the latter are not well-defined and thus, some sort of smooth continuation is required. The mechanically motivated directions in turn are defined densely on the entire surface except for sets of zero measure, see Figure 5.

## 3 PQ Meshing Step

A typical output of the statics-sensitive preprocessing is illustrated in Figure 6(a), where integral curves of the guidance vector field are shown. Given such an output, we want to compute a PQ mesh fulfilling the following requirements:

- One family of mesh polygons shall be closely aligned with the guidance vector field,
- resulting quads shall be rather equally sized, and

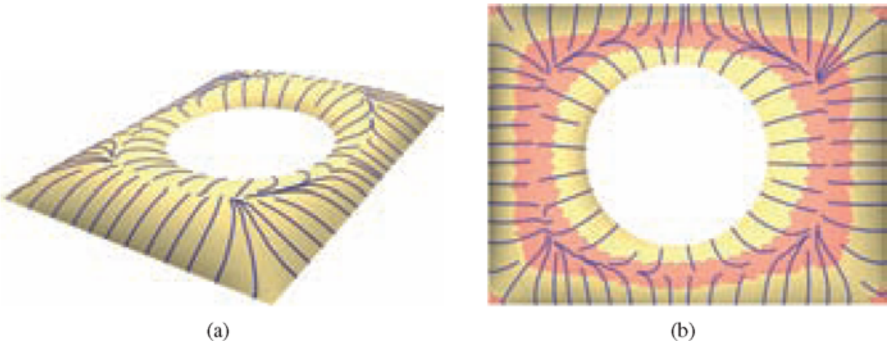


Figure 6: Integral curves of the guidance vector field. The structure with fixed boundary has been subject to dead loading. It can clearly be seen that these integral curves are not suitable for quad remeshing: While they are nearly equally spaced along the boundaries, they become very close to each other and eventually converge to four points (the sources of the guidance vector field) in the interior. When laying out a quad mesh, we therefore do not care about alignment to the vector field in regions with relatively low maximum principal strain. Those regions are colored in brown in (b). The example is a model of the Great Court Roof by Foster and Partners, British Museum, London, compare Figure 1.

- resulting quads shall be planar within close tolerances.

Looking at the example in Figure 6(a), it becomes immediately clear that a resulting PQ mesh can not fulfill all of the above requirements at the same time: While the integral curves shown are nearly equally spaced along the boundaries, they become very close to each other and eventually converge to four points (the sources of the guidance vector field) in the interior. This behaviour depends on the specific load case and boundary conditions: In the example, the boundary is fixed, while gravitation imposes a uniform force distribution in the direction of the negative $z$-axis. Clearly, the deformation is maximal in the very interior of the surface. As strain arises from deformation by one differentiation, these regions are exactly where the zeros or, more precisely, the sources of the guidance vector field are located.

In order to find a reasonable solution fulfilling the requirements, we therefore relax the first one. We use the absolute maximum principal strain as a measure for the importance of alignment with the maximum loading directions and do not care about alignment in regions of the surface where it is below a predefined threshold. This is illustrated in Figure 6(a).

### 3.1 Level Set Method and Quad Meshing

In the following, we give an overview on how we create a quad-dominant mesh (a mesh consisting mainly of quads, except at the boundaries), which exhibits a family of edge polygons that is closely aligned with the guidance vector field, and whose quads are close to planar. We refer to [Zadravec et al. 2010] for the mathematical

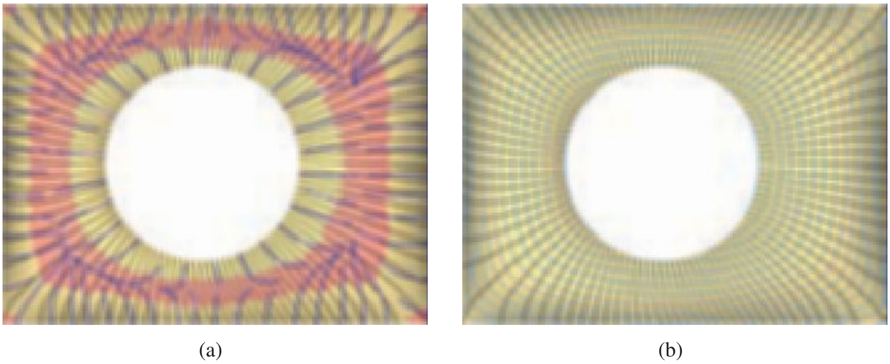


Figure 7: (a) Integral curves of the guidance vector field as shown in Figure 6(a) overlayed with level set curves (black) resulting from an optimization for approximate alignment with the guidance vector field as well as equal spacing. It is easy to recognize that there is a tradeoff between these two optimization goals. In the brown region, alignment was not taken care of. (b) Transverse level set curves (blue) which have been optimized to become conjugate to the aligned level set curves as well as equally spaced. All curves are used for extraction of the quad-dominant mesh.

details. This remeshing task can be decomposed into the following steps:

1. **Edge polygons aligned with guidance vector field.** In this step we compute a family of curves on the reference surface that will later serve as input to the quad meshing. A straightforward way to compute curves that are perfectly aligned with the guidance vector field is shown in Figure 6(a): We could simply choose a set of starting points (e.g. equally spaced on the boundary as shown there) and trace integral curves of the guidance vector field from them. This would not allow us to control equal spacing of the curves. Therefore, we choose the following approach which was introduced in [Zadravec et al. 2010]: We compute a function defined on the reference surface by means of optimization such that its level sets (curves of constant function value) become equally spaced and closely aligned with the guidance vector field. We formulate these requirements as optimization goals in a least-squares sense, i.e. we do not require the goals to be fulfilled exactly but rather make use of the possibility to trade off between them. Moreover, we give up alignment with the guidance vector field in parts of the surface as described above. Figure 7(a) shows an example. We do not give details of the optimization here but refer the reader to [Zadravec et al. 2010].

2. **Edge polygons aligned with conjugate directions to guidance vector field.** There is a close relation between PQ meshes and conjugate curve networks on smooth surfaces (refer to [Liu et al. 2006] and [Zadravec et al. 2010]). Therefore, we aim at aligning the second family of edge polygons to curves

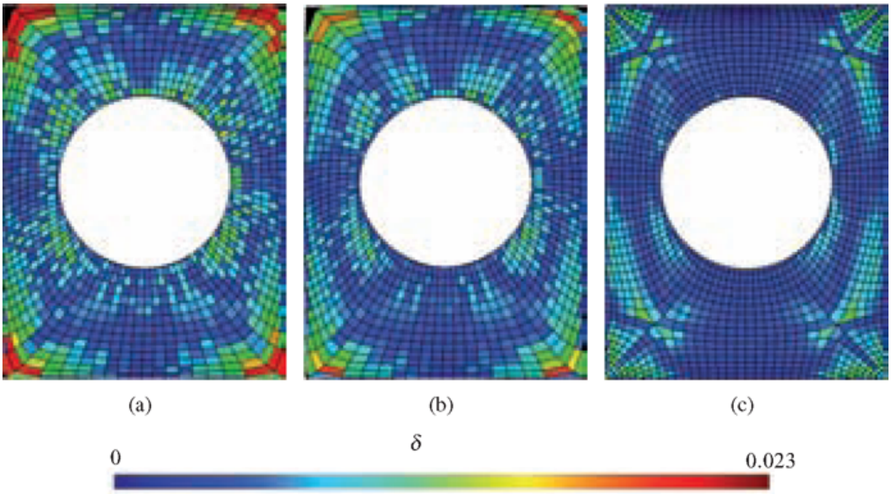


Figure 8: The Great Court Roof quad mesh (a) before and (b) after planarization. Planarity of quads is shown rainbow color coded: Blue corresponds to exact planarity, red corresponds to $\delta = 0.023$. For comparison, a conical PQ mesh is shown in (c). For a definition of conical mesh, see [Liu et al. 2006]. They offer the best possible choice for approximating a smooth surface by a PQ mesh, which is reflected by a low maximum $\delta = 0.015$. The connectivity is determined by the network of principal curvature lines and in general can not be used for our approach.

that are conjugate to the level set curves which we computed in the first step. We do this by computing a vector field that is conjugate to the guidance vector field. Then, we again employ the level set method as described above to compute a function whose level sets are aligned with this vector field. An example is shown in Figure 7(b).

3. **Quad meshing.** We use the level set curves computed using above steps to generate a quad mesh. We do this by intersecting the curves with each other and the boundary curves, and connecting neighboring intersection points. The resulting mesh will exhibit mostly quadrilateral faces, except at the boundaries as can be seen in Figures 9 and 13.

### 3.2 Planarization

The faces of meshes resulting from above procedure can be expected to be nearly planar because we use conjugate curve networks to generate them. This is illustrated by Figures 8(a) and 13(a). Nevertheless, we improve the planarity of the faces by subjecting the mesh to an optimization similar to [Liu et al. 2006], which slightly perturbs the positions of the vertices such that the faces become planar. Mathematical details can be found in [Zadravec et al. 2010]. The quality measure $\delta$, used in

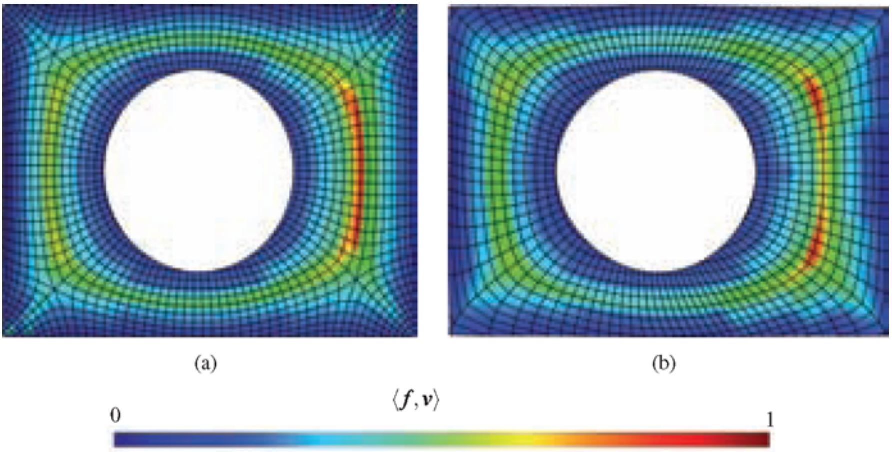


Figure 9: Compliance of the statically-optimized grid shell (b) is about 8 % lower than of the reference structure (a) .

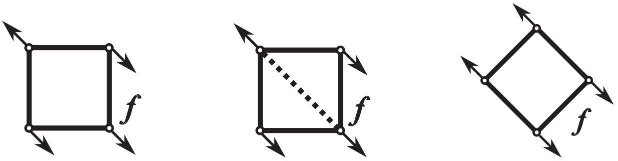
the following for planarity of quad faces, corresponds to the distance between the diagonals of a quad divided by the mean length of its diagonals. For $n$-gons with $n > 4$, we consider planarity for all possible quads that are spanned by the vertices of the $n$-gon.

## 4 Results

Above all, the following discussion should assess the resulting quad meshes with respect to their mechanical properties. Particularly, we would like to oppose those which underwent the first phase of our algorithm to those which did not. A suitable performance measure is the local compliance, which is – as the scalar product of force and displacement fields $\langle \boldsymbol{f}, \boldsymbol{v} \rangle$ – inversely proportional to stiffness. The lower the compliance of a structure, the more favorable its statical behavior will be for a certain load case. Primarily, we must pass from the continuous viewpoint, i.e. from the microscopic triangle mesh serving as reference, to the discrete edge set of the resulting PQ mesh. As the former is assumed to be a membrane shell and thus non-resistant to bending, it translates to a *truss* in the discrete domain. Deformations of such derive from a similar kind of equilibrium condition as (1). Again, the system adopts the state of minimal potential $E_{\mathrm{el}} - W$. In contrast to above, $E_{\mathrm{el}}$ now equals the (weighted) sum of elastic energies stored in every separate rod caused by extension/compression along its longitudinal axis. Let us remark that the discrete surface structures are sometimes called *grid shells*. This naming nicely illustrates their continuous origin.

### 4.1 Great Court Roof of British Museum

The first example geometry has already been introduced above. PQ approximations resembling the network of principal curvature lines are shown in Figures 8(b) (example taken from [Zadravec et al. 2010]) and 8(c). They differ mainly in the presence of four *singular vertices* or *branch points*, where more than four quads meet. This concept of *branched coverings* for improving the alignment with the guidance vector field is introduced in [Kälberer et al. 2007]. Nevertheless in this example, there is a tradeoff between the requirements as illustrated in Figure 7(a).



(a) High twisting moment

(b) Stiffening by triangulation

(c) Torsion-free alignment

Figure 10: Quadrilateral frame.

A compliance reduction becomes notable, mostly in the central regions where the deformation is highest (in modulus), as can be seen in Figure 9. In average, the reduction is about 8.0 %. The following supplies a heuristic explanation for why this should be so: Consider one grid cell in the discrete domain. Obviously, such a quadrangular frame is per se not very resistant to torsion, see Figure 10(a). In practice, supporting beams are inserted between two opposing corners to increase torsion stiffness, see Figure 10(b). Geometrically speaking, this would mean reverting to a triangular mesh. The sole remaining alternative is to orient the frame according to the known load case as indicated in Figure 10(c). This is roughly what our algorithm does, however, in a more abstract way and starting from a continuous perspective on the problem.

### 4.2 Neumuenster Abbey Court Roof Study

The given surface was originally designed by means of a combined structural and geometrical form finding process. Figure 11 shows that it was possible to fulfill the alignment with the guidance vector field very well while keeping the variation of spacing within tolerable bounds. We make the important observation that, like other e.g. curvature-based guidance fields, also the directions of principle strain may require the introduction of branch points into the quad mesh. In Figure 12, they appear where curves of maximal and minimal strain interchange roles. Figure 13 confirms that sufficient planarity was achieved.

We use a PQ mesh already shown in [Zadravec et al. 2010] for comparison of compliances. The impression one gets from Figures 14(a) and (b) is that the compliance *distribution* is less desirable in the mechanically motivated result. Still as predicted, its peak value of 0.8 is around 9 % lower than 0.88 in the reference model.

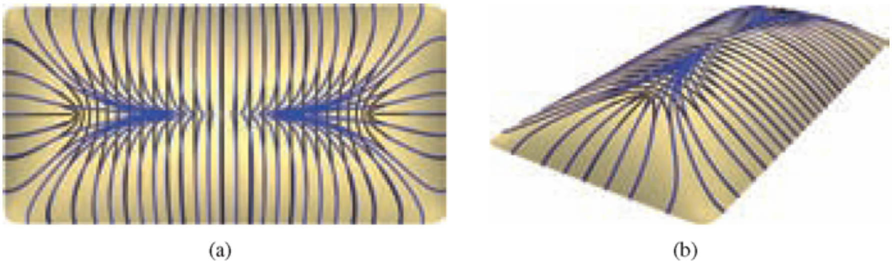


Figure 11: Integral curves of the guidance vector field (blue) and resulting level set curves (black) for the Neumuenster Abbey Court Roof example. Loading and boundary conditions are the same as in Section 4.1. In contrast to the example shown in Figure 6(a), the guidance vector field exhibits only one source. The level set curves are aligned very well to the guidance vector field except in the central region, where maximum principal strains are lower and integral curves would obviously not be usable for quad meshing.

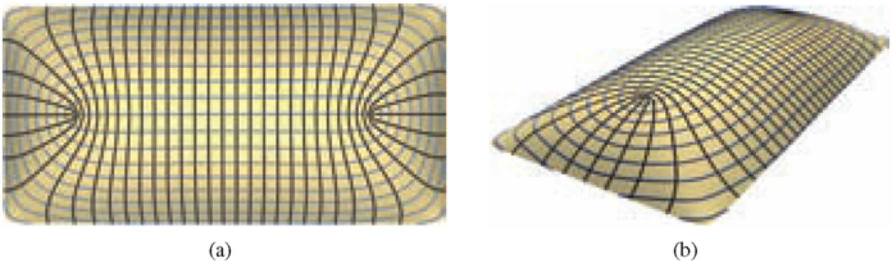


Figure 12: Level set curves aligned with the guidance vector field (black) and transverse level set curves (blue) optimized to become equally spaced and conjugate to the black curves. Quad meshing has been done based on these curves, see Figure 13.

## 5 Conclusion

We put forward a method for the layout of quadrilateral meshes which are almost planar and, beyond that, optimized with respect to their statics. Therein, physical and geometric computations are decoupled as stringently as possible to reduce conflicts among the multiplicity of involved objectives and constraints: Using a shell model of the reference surface, a first step yields the direction of maximal principle strain in every point. Then, a not necessarily planar quad-dominant mesh is generated with one family of isoparameter lines integrating the maximal principal strain vector field wherever possible. The algorithm concludes with an established planarization procedure, preserving the initial mechanical properties in large part. A compliance analysis suggests superiority of the statically optimized layouts over those arising from purely geometrical reasoning.

One future extension of the presented framework is obvious: While the basic

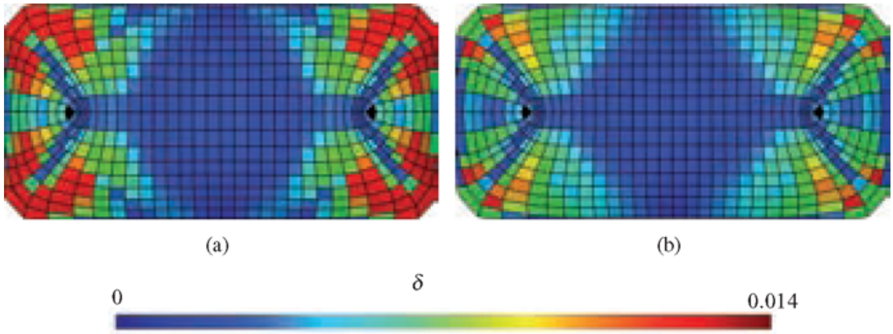


Figure 13: The Neumuenster Abbey Court Roof quad mesh (a) before and (b) after planarization. Planarity of quads is shown rainbow color coded: Blue corresponds to exact planarity, red corresponds to $\delta = 0.014$.

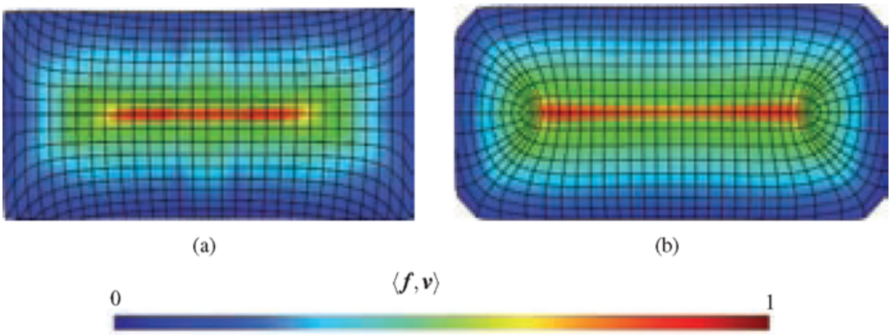


Figure 14: Remeshing result (a) with and (b) without mechanical preprocessing.

principle is well demonstrated under a membrane assumption, only a FEM implementation suitable for fourth-order PDEs would accommodate sufficiently general deformation modes like strong bending. Moreover, we plan to incorporate a shape optimization step, in which now the geometry itself is treated as variable – of course within tolerances describing the architect's willingness to accept modifications of the original design.

## Acknowledgments

The research leading to these results has received funding from the European Communitys Seventh Framework Programme under grant agreement No. 230520 (ARC), and grant No. 813391 of the Austrian research promotion agency (FFG). The authors would like to thank Waagner-Biro for providing the digital model of

the British Museum courtyard roof, and RFR for providing the Neumuenster Abbey court roof mesh.

## References

BOTSCH, M., AND SORKINE, O. 2008. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics 14*, 1, 213–230.

CIRAK, F., ORTIZ, M., AND SCHRÖDER, P. 2000. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering 47*, 12, 2039–2072.

DAVIS, T., AND HAGER, W. 2009. Dynamic supernodes in sparse Cholesky update/downdate and triangular solves. *ACM Transactions on Mathematical Software 35*, 4, 1–27.

GLYMPH, J., SHELDEN, D., CECCATO, C., MUSSEL, J., AND SCHOBER, H. 2002. A parametric strategy for freeform glass structures using quadrilateral planar facets. In *Acadia 2002*, ACM, 303–321.

GRINSPUN, E., HIRANI, A., DESBRUN, M., AND SCHRÖDER., P. 2003. Discrete shells. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

HUGHES, T., COTTRELL, J., AND BAZILEVS, Y. 2005. Isogeometric analysis: CAD, finite elements, nurbs, exact geometry and mesh refinement. *Computational Methods in Applied Mechanical Engineering 194*, 4135–4195.

HUGHES, T. 2000. *The Finite Element Method*. Dover Publications.

KÄLBERER, F., NIESER, M., AND POLTHIER, K. 2007. QuadCover – surface parameterization using branched coverings. *Computer Graphics Forum 26*, 3, 375–384. Proc. Eurographics.

KOITER, W. 1970. On the foundations of the linear theory of thin elastic shells. In *Proc. Kon. Ned. Ak. Wet.*, 169–195.

LIU, Y., POTTMANN, H., WALLNER, J., YANG, Y. L., AND WANG, W. 2006. Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graphics 25*, 3, 681–689.

LOVE, A. 1888. On the small vibrations and deformations of thin elastic shells. *Philosophical Transactions of the Royal Society*, 179–491.

NAGHDI, P. 1972. *Handbuch der Physik*. Springer, ch. The theory of shells and plates, 425–640.

REISSNER, E. 1947. On bending of elastic plates. *Quart. Appl. Math. 55*, 45–68.

WINSLOW, P., PELLEGRINO, S., AND SHARMA, S. 2008. Multi-objective optimization of free-form grid structures. In *Proceedings of the International Conference on Engineering Optimization*.

ZADRAVEC, M., SCHIFTNER, A., AND WALLNER, J. 2010. Designing quad-dominant meshes with planar faces. vol. 29. Proc. Symp. Geometry Processing.

# List of Contributors

**Sean Ahlquist** (Univ. Stuttgart)

*Address:* Sean Ahlquist
Institute for Computational Design
Universität Stuttgart
Keplerstrasse 11
70174 Stuttgart
Germany

*sean.ahlquist@icd.uni-stuttgart.de*

**Jonathan Balzer** (KAUST)

*Address:* Jonathan Balzer
King Abdullah University of Science and Technology
Thuwal 23955-6900
Kingdom of Saudi Arabia

*jonathan.balzer@kaust.edu.sa*

**Ralph Bärtschi** (ETH Zürich)

*Address:* Ralph Bärtschi
Institute for Technology in Architecture
ETH Hänggerberg, HIL F56
Wolfgang-Pauli-Str. 15
8093 Zürich
Switzerland

*baertschi@arch.ethz.ch*

**Ilija Bentscheff** (UdK Berlin)

*Address:* Ilija Bentscheff
Student, Universität der Künste Berlin
Hardenbergstrasse 33
10623 Berlin
Germany

*i.bentscheff@gmail.com*
*http://www.ilijabentscheff.com*

**William Bergeron-Mirsky** (Stevens Institute of Technology)

*Address:* William Bergeron-Mirsky
Student, Product Architecture and Engineering Lab
Stevens Institute of Technology
Castle Point on Hudson
Hoboken NJ 07030-5991
U.S.A.

*bill@bergeronmirsky.net*
*http://www.bergeronmirsky.net/*

**Klaus Bollinger** (Bollinger + Grohmann Ingenieure)

*Address:* Klaus Bollinger
B+G Ingenieure Bollinger und Grohmann GmbH
Westhafenplatz 1
60327 Frankfurt
Germany

*kbollinger@bollinger-grohmann.de*

**Tobias Bonwetsch** (ETH Zürich)

*Address:* Tobias Bonwetsch
Institute for Technology in Architecture
ETH Hänggerberg, HIL F56
Wolfgang-Pauli-Str. 15
8093 Zürich
Switzerland

*bonwetsch@arch.ethz.ch*

**Maurizio Brocato** (Univ. Paris Est)

*Address:* Prof. Maurizio Brocato
École nationale superieure
d'architecture de Paris-Malaquais
Laboratoire Géométrie Structure et
Architecture
14, rue Bonaparte
75272 Paris Cedex 06,
France

*maurizio.brocato@paris-malaquais.archi.fr*

**Cristiano Ceccato** (Zaha Hadid Architects)

*Address:* Cristiano Ceccato
Zaha Hadid Architects
10 Bowling Green Lane
London EC1R 0BQ
U.K.

*cristiano.ceccato@zaha-hadid.com*

**Barbara Cutler** (Rensselaer Polytechnic Institute)

*Address:* Barbara Cutler
Rensselaer Polytechnic Institute
Materials Research Center
110 8th Street Troy, NY 12180-3590
U.S.A.

*cutler@cs.rpi.edu*

**Bailin Deng** (TU Wien)

*Address:* Bailin Deng
Technische Universität Wien
Wiedner Hauptstr. 8–10/104
1040 Wien
Austria

*bldeng@gmail.com*

**Mario Deuss** (EPFL Lausanne)

*Address:* Mario Deuss
Laboratoire d'informatique graphique et
géométrique
École Polytechnique Fédérale de
Lausanne
CH 1015 Lausanne
Switzerland

*mario.deuss@epfl.ch*

**Michael Eigensatz** (ETH Zürich, EPFL, and Evolute)

*Address:* Michael Eigensatz
Evolute GmbH
Schwindgasse 4/10
1040 Wien
Austria

*eigensatz@evolute.at*

**Gershon Elber** (Technion)

*Address:* Prof. Gershon Elber
Dept. of Computer Science,
Technion – IIT Haifa
32000 Haifa
Israel

*gershon@cs.technion.ac.il*

**Simon Flöry** (Evolute GmbH and TU Wien)

*Address:* Simon Flöry
Technische Universität Wien,
Wiedner Hauptstr. 8–10/104
1040 Wien
Austria

*floery@evolute.at*

**Christoph Gengnagel** (UdK Berlin )

*Address:* Prof. Christoph Gengnagel
Universität der Künste Berlin
Hardenbergstrasse 33
10623 Berlin
Germany

*gengnagel@udk-berlin.de*

**Fabio Gramazio** (ETH Zürich)

*Address:* Prof. Fabio Gramazio
Institute for Technology in Architecture
ETH Hänggerberg, HIL F56
Wolfgang-Pauli-Str. 15
8093 Zürich
Switzerland

*gramazio@arch.ethz.ch*

**Manfred Grohmann** (Bollinger + Grohmann Ingenieure)

*Address:* Manfred Grohmann
B+G Ingenieure Bollinger und
Grohmann GmbH
Westhafenplatz 1
60327 Frankfurt
Germany

*mgrohmann@bollinger-grohmann.de*

**John Gulliford** (Stevens Institute of Technology)

*Address:* John Gulliford
Student, Product Architecture and
Engineering Lab
Stevens Institute of Technology
Castle Point on Hudson
Hoboken NJ 07030-5991
U.S.A.

*gullifordjohn@gmail.com*

**Lars Hesselgren** (PLP Research)

*Address:* Lars Hesselgren
PLP Architecture
Carlow House, Carlow Street
London NW1 7LH
U.K.

*lhesselgren@plparchitecture.com*

**Mathias Höbinger** (Evolute GmbH and TU Wien)

*Address:* Mathias Höbinger
Technische Universität Wien
Wiedner Hauptstr. 8–10/104
1040 Wien
Austria

*hoebinger@evolute.at*

**Qixing Huang** (Stanford University)

*Address:* Qixing Huang
Stanford University
Clark Center, S294. 318 Campus Drive
Stanford, CA 94305
U.S.A.

*huangqx@stanford.edu*

**Hauke Jungjohann** (Knippers Helbig Inc.)

*Address:* Hauke Jungjohann
Knippers Helbig Inc.
134 Spring Street, Suite 601,
New York, NY 10012
U.S.A.

*h.jungjohann@knippershelbig.com*

**Martin Kilian** (Evolute GmbH and TU Wien)

*Address:* Martin Kilian
Technische Universität Wien,
Wiedner Hauptstr. 8–10/104
1040 Wien
Austria

*kilian@evolute.at*

**Michael Knauss** (ETH Zürich)

*Address:* Michael Knauss
Institute for Technology in Architecture
ETH Hänggerberg, HIL F56
Wolfgang-Pauli-Str. 15
8093 Zürich
Switzerland

*knauss@arch.ethz.ch*

**Matthias Kohler** (ETH Zürich)

*Address:* Prof. Matthias Kohler
Institute for Technology in Architecture
ETH Hänggerberg, HIL F56
Wolfgang-Pauli-Str. 15
8093 Zürich
Switzerland

*kohler@arch.ethz.ch*

**Benjamin S. Koren** (One-to-One GmbH)

*Address:* Benjamin S. Koren
One-to-One GmbH
Zeppelinallee 25
60325 Frankfurt
Germany

*koren@1-to-one.com*

**Toni Kotnik** (ETH Zürich)

*Address:* Toni Kotnik
Institute for Technology in Architecture
HIL E 43.2, Wolfgang-Pauli-Str.15,
8093 Zürich
Switzerland

*kotnik@arch.ethz.ch*

**Lorenz Lachauer** (ETH Zürich)

*Address:* Lorenz Lachauer
Institute for Technology in Architecture
HIL E 43.2, Wolfgang-Pauli-Str.15,
8093 Zürich
Switzerland

*lachauer@arch.ethz.ch*

**Jason Lim** (Stevens Institute of Technology)

*Address:* Jason Lim
Student, Product Architecture and
Engineering Lab
Stevens Institute of Technology
Castle Point on Hudson
Hoboken NJ 07030-5991
U.S.A

*jasonlimteckchye@gmail.com*

**Achim Menges** (Univ. Stuttgart)

*Address:* Prof. Achim Menges
Institute for Computational Design
Universität Stuttgart
Keplerstrasse 11
70174 Stuttgart
Germany

*achim.menges@icd.uni-stuttgart.de*

**Niloy J. Mitra** (KAUST and IIT Delhi)

*Address:* Prof. Niloy Mitra
King Abdullah University of Science and Technology
Thuwal 23955-6900,
Kingdom of Saudi Arabia

*niloym@gmail.com*

**Lucia Mondardini** (Università di Bologna)

*Address:* Lucia Mondardini
Università di Bologna
Facoltà di Ingegneria
Dipartimento di architettura e pianificazione territoriale
Viale del Risorgimento 2
40135 Bologna
Italy

*mondardini.lucia@gmail.com*

**Joshua Nasman** (Rensselaer Polytechnic Institute)

*Address:* Joshua Nasman
Rensselaer Polytechnic Institute
Materials Research Center
110 8th Street Troy, NY 12180-3590
U.S.A.

*nasmaj@cs.rpi.edu*

**Aditi Patel** (Stevens Institute of Technology)

*Address:* Aditi Patel
Student, Product Architecture and Engineering Lab
Stevens Institute of Technology
Castle Point on Hudson
Hoboken NJ 07030-5991
U.S.A.

*aditi1110@gmail.com*

**Mark Pauly** (EPFL Lausanne)

*Address:* Prof. Mark Pauly
Laboratoire d'informatique graphique et géométrique
École Polytechnique Fédérale de Lausanne
1015 Lausanne
Switzerland

*mark.pauly@epfl.ch*

**Helmut Pottmann** (KAUST and TU Wien)

*Address:* Prof. Helmut Pottmann
Technische Universität Wien
Wiedner Hauptstr. 8–10/104
1040 Wien
Austria

*pottmann@geometrie.tuwien.ac.at*

**Mette Ramsgard Thomsen** (The Royal Danish Academy of Fine Arts)

*Address:* Mette Ramsgard Thomsen
The Royal Danish Academy of Fine Arts,
School of Architecture
Philip de Langes Allé 10
1435 Kopenhagen K
Denmark

*mette.thomsen@karch.dk*

**Jacob Riiber** (The Royal Danish Academy of Fine Arts)

*Address:* Jacob Riiber
The Royal Danish Academy of Fine Arts,
School of Architecture
Philip de Langes Allé 10
1435 Kopenhagen K
Denmark

*jacob.riiber@karch.dk*

**Alexander Schiftner** (Evolute and TU Wien)

*Address:* Alexander Schiftner
Evolute GmbH
Schwindgasse 4/10
1040 Wien
Austria

*schiftner@evolute.at*

**Tomohiro Tachi** (The University of Tokyo)

*Address:* Tomohiro Tachi
The Universiity of Tokyo 3-8-1
Komaba, Meguro-Ku, Tokyo 153-8902
Japan.

*tachi@idea.c.u-tokyo.ac.jp*

**Martin Tamke** (Royal Danish Academy of Fine Arts)

*Address:* Martin Tamke
The Royal Danish Academy of Fine Arts,
School of Architecture
Philip de Langes Allé 10
1435 Kopenhagen K
Denmark

*martin.tamke@karch.dk*

**Oliver Tessmann** (Bollinger + Grohmann Ingenieure)

*Address:* Oliver Tessmann
B+G Ingenieure Bollinger und Grohmann GmbH
Westhafenplatz 1
60327 Frankfurt
Germany

*otessmann@bollinger-grohmann.de*

**Johannes Wallner** (TU Graz)

*Address:* Prof. Johannes Wallner
Technische Universität Graz
Kopernikusgasse 24
8010 Graz
Austria

*j.wallner@tugraz.at*